UNIVERSITY of GREENWICH

MSA UNIVERSITY
IN EGYPT SINCE 1996
Established by Dr.Nawal El Degwi
MSA UNIVERSITY
جامعة أكتوبر للعلوم الحديثة والآداب

MSA UNIVERSITY
OCTOBER UNIVERSITY FOR MODERN SCIENCES AND ARTS
FACULTY OF COMPUTER SCIENCE

MSA CS - Graduation Projects Fall 2020-Spring 2022

Undergraduate Publication list

Proceedings V 2.6

SUMMER RESEARCH INTERNSHIP, UNDERGRADUATE GRADUATION PROJECTS

This booklet was done under the supervision of Dr. Ayman Ezzat
First release, May 2022

Contents

1. Deans Word

I am really proud of the distinguished level of Faculty of Computer Science's Graduation projects 2022. They reflect serious thinking and scientific contributions, genius application of AI tools, and excellent coding skills. About 20% of the project's results were accepted and published as research papers in IT research magazines and conferences. We are aiming to push the publishing ratio to reach 80% of the projects in the coming few years. I would like to send a special thanks to Graduation project committees' members and to the projects' supervisors who gave time, patience and effort to help and encourage their students to get up to this level.

Special thanks are also going to the arbitration committees' members, the industry specialists and professors from British and Egyptian Universities who were really impressed with the projects' outstanding ideas, contributions and results, as well as the innovation spirits, oral and writing skills, and the matured personality of our newly graduates from the Faculty of Computer Science.

Ali El-Bastawissy

2. Book Chapters

2.1 Hyperparameters Optimization of Deep Convolutional Neural Network for Detecting COVID-19 Using Differential Evolution

Two students of Computer science Abdel Rahman Ezz Eldin and Mostafa Mohamed Saeed for publishing their second scientific paper with cooperation and under supervision of Dr. Ali Khater and Eng. Shereen El-Feky. Their paper was under title of "Hyperparameters Optimization of Deep Convolutional Neural Network for Detecting COVID-19 Using Differential Evolution" which has been published at International Series in Operations Research & Management Science book series (ISOR, volume 320) at Springer [15]

Chapter 18
Hyperparameters Optimization of Deep Convolutional Neural Network for Detecting COVID-19 Using Differential Evolution

Abdelrahman Ezzeldin Nagib, Mostafa Mohamed Saeed, Shereen Fathy El-Feky, and Ali Khater Mohamed

Abstract COVID-19 is one of the most dangerous diseases that appeared during the past 100 years, that caused millions of deaths worldwide. It caused hundreds of billions of losses worldwide as a result of complete business paralysis. This reason has attracted many researchers to attempt to find a suitable treatment for this dreaded virus.

The search for a cure is still ongoing, but many researchers around the world have begun to search for the safest ways to detect if a person carries the virus or not. Many researchers resorted to artificial intelligence and machine learning techniques in order to detect whether a person is carrying the virus or not.

However, many problems are arising when using these techniques, the most important problem is the optimal selection of the parameter values for these methods, as the choice of these values greatly affects the expected results.

In this chapter, Differential Evolution algorithm is used to determine the optimal values for the hyperparameters of Convolutional Neural Networks, as Differential Evolution is one of the most efficient optimization algorithms in the last two decades. The results obtained showed that the use of Differential Evolution in optimizing the hyperparameters of the Convolutional Neural Network was very efficient.

Keywords Convolutional neural network · COVID-19 · Differential Evolution · Hyperparameters Optimization

A. E. Nagib · M. M. Saeed · S. F. El-Feky · A. K. Mohamed (✉)
Faculty of Computer Science, October University for Modern Sciences and Arts (MSA), Giza, Egypt
e-mail: Abdelrahman.ezzeldin1@msa.edu.eg; Mostafa.mohamed52@msa.edu.eg; selfeky@msa.edu.eg; akhater@msa.edu.eg

S. A. Hassan et al. (eds.), *Decision Sciences for COVID-19*, International Series in Operations Research & Management Science 320,
https://doi.org/10.1007/978-3-030-87019-5_18

18.1 Introduction

Nowadays, COVID-19 became this ages' lesion after the new Chinese epidemic emergency which unfortunately considered from the most harmful diseases that sometimes cause death, COVID-19 is resulting from Coronaviruses that cause illnesses which are similar to flu with more severe clinical outcomes (da Costa et al. 2020). The workers of the healthcare sector have a very important role by providing healthcare for the COVID-19 patients at the front lines and make sure of prevention of the infection additional to the implementation of control (IPC) measures.

There are many clinical features of COVID-19 that are varying from case to case beginning from asymptomatic state to acute respiratory distress syndrome and multi Organ dysfunction, but there are many common clinical features as cough, sore throat, headache, fever (not in all cases), myalgia and breathlessness, in some cases, the infection can progress by the end of the first week to pneumonia then respiratory failure and death, that progress caused by the extreme rise in inflammatory cytokines (Singhal 2020).

Unfortunately, it is very hard to differentiate the COVID-19 Infection from all types of respiratory viral infections such as (Influenza, Parainfluenza, Respiratory syncytial virus, Non-COVID-19 Coronavirus, etc.) (Singhal 2020). The effective detection of COVID-19 patients is the most critical and important step towards the confrontation of the COVID-19 pandemic. Chest radiography is one of the most important detection approaches cause the radiography of the patients shows abnormalities in images which are used as characteristics of the infection (Wang et al. 2020).

The ground-glass opacity in the Chest X-rays has been observed once the COVID-19 has reached the lungs to differentiate between the COVID-19 patient and non-infected persons. Sample of CXR images for normal person, COVID-19 person, and pneumonia patient is presented in Fig. 18.1.

Although the radiological features from images are closely similar and overlapping those which have associated with SARS and MERS. The involvement of the lung bilateral is on initial imaging is more likely to be discovered as

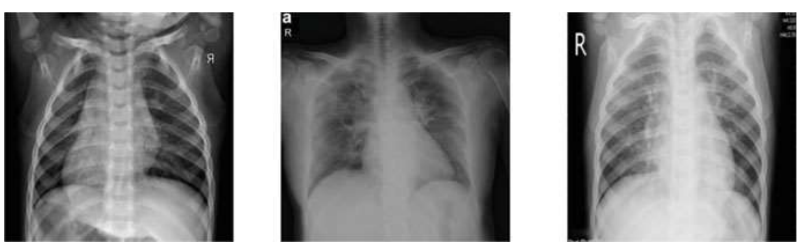


Fig. 18.1 Sample of CXR images

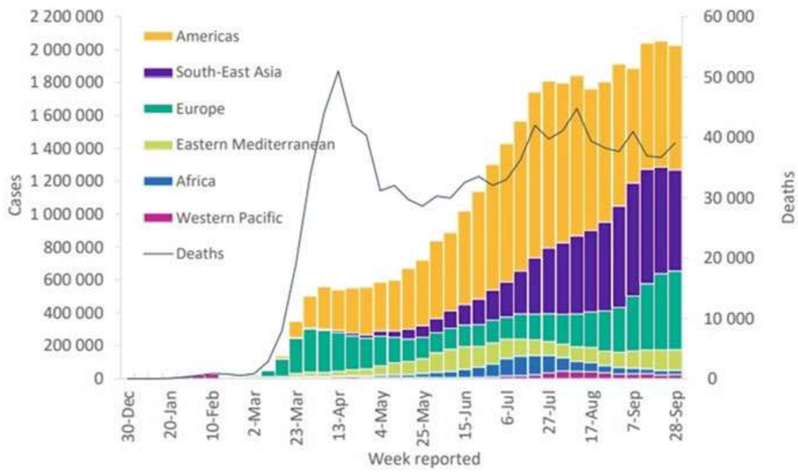


Fig. 18.2 Number of COVID-19 cases

COVID-19, as those associating SARS and MERS are more predominantly unilateral, that leads to the urgency need to expertise but need to fast diagnostic techniques that aligned with the pneumonic disease (Tahamtan and Ardebili 2020).

Over 37 million people have been infected by COVID-19 and one million deaths have been in the period by WHO Region from December 30, 2019 till October 11, 2020 as depicted in Fig. 18.2.

COVID-19 has a severe effect on all life fields as global aviation that has lost around 100 billion US dollars because of the travel restrictions which directly affect the world's economy accordingly and affect the individual's social life which causes economic crisis in addition to the health crisis, and that was the motivation for thousands of researches in all the fields after corona pandemic.

With the constant increase of infection rate, the call of the fast detection became an urgent need that led to the manifestation of the COVID-19 Auto-detection techniques which help with a rapid and automatic diagnosis through medical images processing using graphical processing units (GPUs).

Recently, the Convolutional Neural Network (CNN) presented optimistic results when used to classify radiological images. CNNs were used in many real-life applications including image classification. Therefore, this advantage was the main motive to present a CNN algorithm for diagnosing COVID-19 in the presented work.

Although CNNs networks have proposed a lot of breakthroughs but it seems like a black-box predictor as the hyperparameters of CNNs have a significant impact on the performance of the network and have a direct control on the training process.

The appropriate selection for the hyperparameters is very critical and directly affects the training process of the CNN network.

For example, the learning rate plays a vital role in network training, if it is too low, the network could lose the diversity. If it is too high, the model could converge so fast which leads to a premature convergence or stagnation (Nagib et al. 2020).

Therefore, the optimization of the hyperparameters of CNNs is considered a very challenging task and still an open area for research. A novel framework is presented in this chapter for optimizing the hyperparameters of CNN using Differential Evolution algorithm through CXR images. The proposed framework resulted in better performance metrices and avoid overfitting in the model training using the input images.

The main contributions of the chapter are:

1. Optimization of the hyperparameters of CNN using Differential Evolution algorithm for increased accuracy in diagnosing COVID-19 patients using CXR images.
2. The proposed framework is compared with other optimization algorithms used for tuning the hyperparameters of CNNs.
3. Ninety-nine percent accuracy is obtained for diagnosing COVID-19 patients using the presented framework.

The rest of the chapter is organized as follows: Section 18.2 presents related work. Differential Evolution and CNN are presented in Sect. 18.3. The material and methodology are presented in Sect. 18.4. The experiments are given in Sect. 18.5. Finally, the conclusion is presented in Sect. 18.6.

18.2 Related Work

Currently, there are many attempts that are being made to find a fast and effective way to detect the **COVID-19** virus with high accuracy using artificial intelligence techniques to end this global crisis. This section will review some of these attempts, which depend on the diagnosis of the patient through the CXR images.

Narin et al. (2020) proposed a detection method to detect COVID-19 infected patients using five pre-trained CNN-based models depends on Chest X-rays radiographs, binary classification with different three types have been implemented with an average of 99% accuracy between three datasets. El-Din Hemdan et al. (2020) have demonstrated the application of deep learning models for COVID-19 Classification based on X-ray images by using a new framework for deep learning called COVIDX-net, validation has been done on 50 X-Ray images with 25 confirmed COVID-19 positive cases, 80–20% evaluation has been done and tested on the seven different architecture of the COVIDX-net and shows that visual geometry group network and DenseNet showed similar or convergent performance with 0.89 *f*-score value for the automated COVID-19 and the normal COVID-19 is with 0.91 *f*-score value.

Khan et al. (2020) detected COVID-19 from Chest X-Ray images by using deep convolution and CoroNet Models and the trained CoroNet shows overall accuracy of

89.6%. Li et al. (2020a, b) developed deep learning model for COVID-19 autodetection, they developed convolutional neural network to extract biomarkers from chest scanning images, diagnostic performance was evaluated by the measurement of the area under the receiver operating characteristic curve and the results show the AUC (area under the receiver operating characteristic) with 0.96 value. Maghdid et al. (2020) proposed an artificial intelligence tool for COVID-19 autodetection by implementing dataset containing the scanning results of X-rays and chest scanning and using a convolutional neural network and the mechanism of deep learning for auto-detection Process, the results shows that the accuracy up to 98% via pre-trained network and 94.1% accuracy using the modified convolutional neural network.

Mahdy et al. (2020) produced the usage of the auto classifiers for COVID-19 detection that depends on X-ray images using deep features, they used the system of multilevel thresholding and support vector machine which presented a high classification of the COVID-19 patient by using the same size of all the images in JPEG format, and the results show that the average of the accuracy of used auto-classification model is 97.48%. Rehman et al. (2020) proposed the COVID-19 auto-diagnoses deep learning method by differentiating it into three categories (viral pneumonia, bacterial pneumonia, and normal cases), the results showed that the K-fold −10 produced 98.75% accuracy by using X-Rays and chest scanning. Abbas et al. (2020) made an adaptation for the chest X-ray images irregularities by using decompose, transfer, and compose (DETRAC) for the COVID-19 classification, the results that have been introduced shows that DETRAC has a great capability in COVID-19 detection using image dataset with an accuracy of 95.12%.

Afshar et al. (2020) provided the usage of handling small datasets by using alternative modeling techniques based on capsule network, it is used in COVID-19 detection, the used technique provides an advantage over the convolutional neural network-based models because it achieved an accuracy of 95.7%. Apostolopoulos and Mpesiana (2020a) provided evaluation for the performance of the classic CNN that has been used in medical detection sector by using the transfer learning which helps in many abnormalities detection process, they build their experiment using two datasets which collected from public medical centers that include X-ray images, this experiments showed that deep learning introduces very helpful biomarks which obtain 96.78% accuracy. Apostolopoulos et al. (2020) have been investigated extracted features importance by using Convolutional Neural Network Model called Mobile Net in diseases autodetection, the results shows that Convolutional Neural Network Training from scratch produce very helpful biomarkers for diseases do not end to COVID-19. That leads to reaching 87.66 accuracy of COVID-19 detection.

Xu et al. (2020) proposed a model to differentiate between COVID-19, influenza, and normal cases by using deep learning model which has been implemented on the chest scanning images data set, the results showed an overall accuracy of 86.7% from all CT datasets. Wang et al. (n.d.) used Artificial intelligence's deep learning techniques that could extract the Biomarkers of COVID-19 disease that will save the time of clinical diagnosis and for disease control; they made internal and external validation the results showed that internal accuracy is 89.5% and external accuracy is

79.3%. Kumar Sethy et al. (2020) proposed COVID-19 auto-detection by using deep learning-based mechanism depends on X-rays images; they used RESNET50 cooperated with Support vector machine classifiers and its results showed that the detection is with an accuracy of 95.52% from the repository of GitHub, Kaggle, and Open-I.

Li et al. (2020a, b) developed a deep learning neural network for COVID-19 auto-detection by visual features extraction from chest CTs by made binary classification and made validation by using independency, specify and independency tests that ends with the fact that deep learning methods detect COVID-19 accurately. Gozes et al. (2020) developed an artificial deep learning technique to detect, quantify, and track COVID-19 disease, by using 2D and 3D deep learning models; they proposed retrospective experiments for analyzing the attitude of the system in COVID-19 detection, the result of classification is 0.996 AUC. Shan et al. (2020) implemented deep learning for auto segmentation and quantification of infection locations beside the lung from chest screening images, the convolutional neural network used to detect infection regions, they used 249 COVID-19 cases, and validated using 300 COVID-19 cases.

18.3 Theory and Methods

18.3.1 Differential Evolution Algorithm

Differential Evolution (DE) presented by Storn and Price (Storn and Price 1995, 1997) is a stochastic population-based search method. DE proved an excellent ability to solve a wide range of optimization problems with different features from many fields and many real-world applications (Kenneth et al. 2005). The evolution process of DE uses mutations, crossover, and selection operators at each generation that must be used in order to obtain the global optima. DE is recognized as one of the most efficient evolutionary algorithms (EAs) currently in use. DE has many advantages including that it is very simple to implement, reliable, speed of convergence, and it is a robust algorithm. Therefore, it is applied in a wide range of applications to solve numerous numbers of real-world applications.

A brief summary of the basic Differential Evolution (DE) algorithm is presented. In simple DE, generally known as *DE/ rand* /1/*bin* (Storn and Price 1997), an initial population, denoted by P, is randomly initialized and consists of NP individual. The vector $x_{i=(x_{1,i},x_{2,i},\ldots,x_{D,i})}$ is used to represent each individual, where D is the number of dimensions in solution space. Since the population will be varied with the running of evolutionary process, the numbers of generation in DE are expressed by $G = 0, 1\ldots,$ G_{max}, where G_{max} is the maximum number of generations. The i$^{\text{th}}$ individual of the population at generation number G, is denoted by $x_i^G = \left(x_{1,i}^G, x_{2,i}^G \ldots, x_{D,i}^G\right)$. The lower bound and upper bound in each dimension is recorded by $X_L = (x_{1,\ L}, x_{2,\ L}\cdot\ldots, x_{D,\ L})$ and $X_U = (x_{1,\ U}, x_{2,\ U}\cdot\ldots, x_{D,\ U})$. The uniform random initial population P_0 is generated within the lower and upper boundaries (X_L, X_U). After the initialization process, those individuals evolve by using DE operators (mutation and crossover) to obtain a trial vector. The parent is compared to its child in order to select the fittest vector that will survive to the next generation. Detailed steps for DE are discussed as follows.

18.3.1.1 Initialization

DE start the optimization process by generating an initial random population P_0. This process is called initialization. Typically, the value of the jth dimension $(j = 1, 2 \ldots, D)$ of the ith individuals $(i = 1, 2 \ldots, NP)$ in the initial population P_0 could be obtained as follows:

$$x^0_{j,i} = x_{j,L} + rand(0, 1).\left(x_{j,U} - x_{j,L}\right) \tag{18.1}$$

where *rand* (0, 1) return a random number that follows uniform distribution between [0, 1].

18.3.1.2 Mutation

At generation G, a mutant vector v_i^G is generated for each target vector x_i^G using the following formula:

$$v_i^G = x_{r1}^G + F.\left(x_{r2}^G - x_{r3}^G\right), r_1 \neq r_2 \neq r_3 \neq i \tag{18.2}$$

where r_1, r_2, $r_3 \in \{1, 2, \ldots, NP\}$ represents three randomly chosen indices. F represents the mutation factor that is a real number that controls the amplification of the difference vector $\left(x_{r2}^G - x_{r3}^G\right)$. Storn and Price (1997) suggested that F in the range [0, 2] is a very good choice. In this work, if the value of any component of the mutant vector violates search space boundaries, then a new value is generated for this component by (1).

18.3.1.3 Crossover

There are two main crossover types, binomial, and exponential. We here elaborate on the binomial crossover. In the binomial crossover, the target vector is mixed with the mutated vector, using the following scheme, to yield the trial vectorU_i^G.

$$u_{j,i}^{G} = \left\{ \begin{array}{r} v_{j,i}^{G}, \text{if } \left(\text{rand}_{j,i} \leq \text{CR or j} = \text{j}_{\text{rand}}\right) \\ x_{j,i}^{G}, \text{otherwise} \end{array} \right\} \tag{18.3}$$

where $rand_{j,\ i}$, ($i \in [1, NP] and\ j \in [1, D]$) is a uniformly distributed random number in [0,1], $CR \in [0, 1]$ called the crossover rate that controls how many components are inherited from the mutant vector, j_{rand} is a uniformly distributed random integer in [1, D] that makes sure at least one component of trial vector is inherited from the mutant vector.

18.3.1.4 Selection

DE adapts a greedy selection strategy. If and only if the trial vector U_i^G yields as good as or a better fitness function value than X_i^G, then U_i^G is set to X_i^{G+1}. Otherwise, the old vector X_i^G is retained. The selection scheme is as follows (for a minimization problem):

$$x_i^{G+1} = \left\{ \begin{array}{r} u_i^G, f\left(u_i^G\right) \leq f\left(x_i^G\right) \\ x_i^G, otherwise \end{array} \right\} \tag{18.4}$$

A detailed description of standard DE algorithm is given in Fig. 18.3.

Parameters of differential evolution play a vital role in its performance and affect the convergence of the algorithm. There are three main control parameters in differential evolution:

1. Mutation factor (F).
2. Cross over rate(CR).
3. Population size (NP).

The control parameters are tuned manually, adaptively, or self-adaptively. (Mohamed and Mohamed 2019a, b) presented a novel scheme for changing the values of CR adaptively based on the probability of success, and a novel scheme for population size reduction is presented in Mohamed and Mohamed (2019a, b), Mohamed et al. (2018), Mohamed, Hadi and Mohamed et al. (2020) and it is used to solve constrained problems (Mohamed and Mohamed 2019a, b) and mixed-integer problems (Mohamed et al. 2019a, b).

From the literature, it has been shown that the mutation scheme in differential evolution has a great impact on the convergence of the algorithm. A complete study for the effect of mutation strategies on the convergence of the algorithm is presented in Mohamed, Hadi and Mohamed et al. (2021).

```
Generate an initial population
Evaluate the fitness of each individual
while the termination criteria are not satisfied do
      For i = 1 to NP do
        Select uniform randomly r1 ≠ r2 ≠ r3 = i
                    j_rand = rndint(1, D)
        For j = 1 to D do
            if rndreal_j(0,1) < CR or j = j_rand then
              u_i,j = x_r1,j + F.(x_r2,j − x_r3,j)
            else
              u_i,j = x_i,j
             End if
          End for
      End for
     For i = 1 to NP do
        Evaluate the offspring u_i
        If f(u_i) is better than or equal to f(x_i) then
             Replace x_i with u_i
        End if
     End for
End while
```

Fig. 18.3 Description of standard DE algorithm

18.3.2 Convolutional Neural Network (CNN)

Since the start of deep learning and the outstanding achievements achieved by the dense neural networks (ANN) that was first invented in 1958, since then dense neural networks have proved amazing results but in the 1980s the convolutional neural networks also known as CNNs or CovNet, which is a class of neural networks with a main specialty in processing the gird like topology data such as images. A digital image is a binary representation of visual data in which the images have a series of pixels that are arranged in a grid-like form that contains the pixel value to denote and show what color and how bright is that color in each pixel would be, since then the convolutional neural networks (CNN) has proved an outstanding performance in the field of deep learning in image classification. The CNN can successfully capture the temporal and spatial dependencies in an image through the application of relevant filters and the architecture performs better fitting to the dataset of images due to the reduction of parameters in the CNN; in other words, the CNN can be trained to understand the sophistication of the images really well.

The Convolutional Neural Network consists of Convolutional layers "as the name applies" in which the Convolutional layers are the layers in which filters are applied to the original image, or maybe to other feature maps in a deep convolutional neural network. This is where most of the user-specified parameters are in the network.

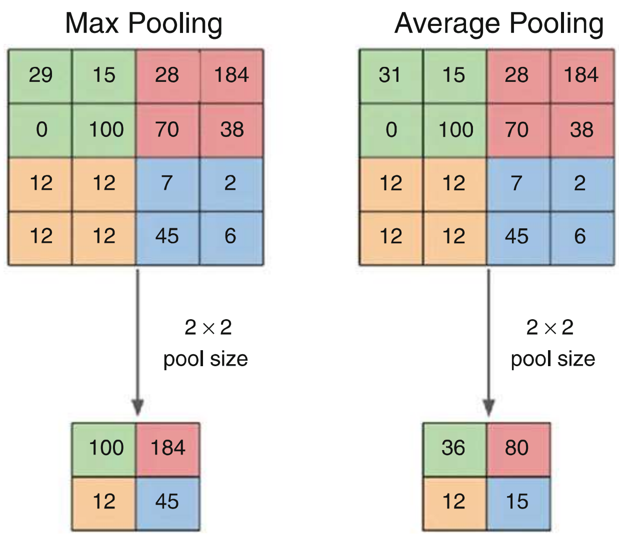


Fig. 18.4 CNN Pooling

Where the most important parameters are Features of a pooling layer and the number of kernels and the size of the kernels.

The CNN structure starts with its first secret ingredient that has made CNNs very successful; that is pooling. Pooling is a scalar transformation vector that operates on each and every local region of an image, but what makes it different from convolutions is that they do not have any filters and do not compute the dot products with the local region, they compute the average of the pixels in the region (Average Pooling) or simply select the highest intensity pixel and discard the rest of the pixels in the region (Max Pooling) (Fig. 18.4).

Another main ingredient In CNN is the kernel, the kernel is nothing but only a filter that is used to extract features from our images, we can say it is a matrix, which is sliding across the image and being multiplied with the input, in such a way that the output is improved in a such a noticeable desirable way (Fig. 18.5).

In summary, the CNN consists of 2 bases, the convolution base, and the classifier base.

The convolutional base has three main layers which are: the convolutional layers, the activation layers, and the pooling layers. These layers are used to discover the features of the input images, which is called a feature map. A feature map is constructed by performing convolution processes to the input image or prior features using a linear filter and merging a term called the bias. Then forwarding this feature map through a non-linear activation function such as Sigmoid or Rectified Linear Unit (RELU). In contrast, the classifier base has dense layers that are combined with

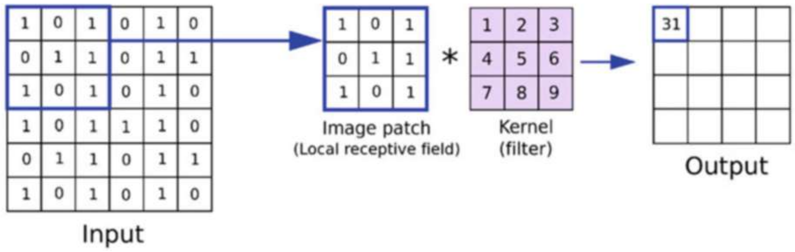


Fig. 18.5 Kernel in CNN

the activation layers to turn the feature maps into one-dimensional vectors to speed up and accelerate the task of classification using neurons.

18.4 The Proposed Framework

In image classification, it is usually better to convert images from RGB (3 color channels) to gray scale (1 color channel) in order to reduce complexity and the size of input for the artificial neural networks, so the presented framework starts by making the images of gray scale.

Another important feature in image classification is the width and height of the image, which is usually resized to smaller value to reduce the input shape for the artificial neural network and yet maintain high results because in the case of a 1080×1080 image of gray scale we get an input shape around 1,160,000. So, in the case of RGB not being gray scale then this number will be multiplied by 3.

Usually image resizing and the width and height of the image is decided randomly while creating the neural network; the presented framework in Fig. 18.6 works on using the differential evolution algorithm to find the best width and height for the images to be resized to according to given bounds that the width and height cannot be less than the minimum width nor more than the maximum height; yet in order to reduce the number of parameters that need to be optimized we decided to always make the width and height equal to each other so we do not have to consider the width as special parameter and the height as another special parameter because this will create an unimaginable size of search space given the other parameters that the Differential Evolution tries to optimize; it will be a very large combination; so for the sake of taking a step forward without including huge search space we treat the width and height as one single parameter in order to find the best number that fits them both at the same time. Then in the framework, we scale the input (the images) value to between 0 and 1.

All of these processes are done according to whether the developer wants only training or training, and testing, or training, testing and, validation.

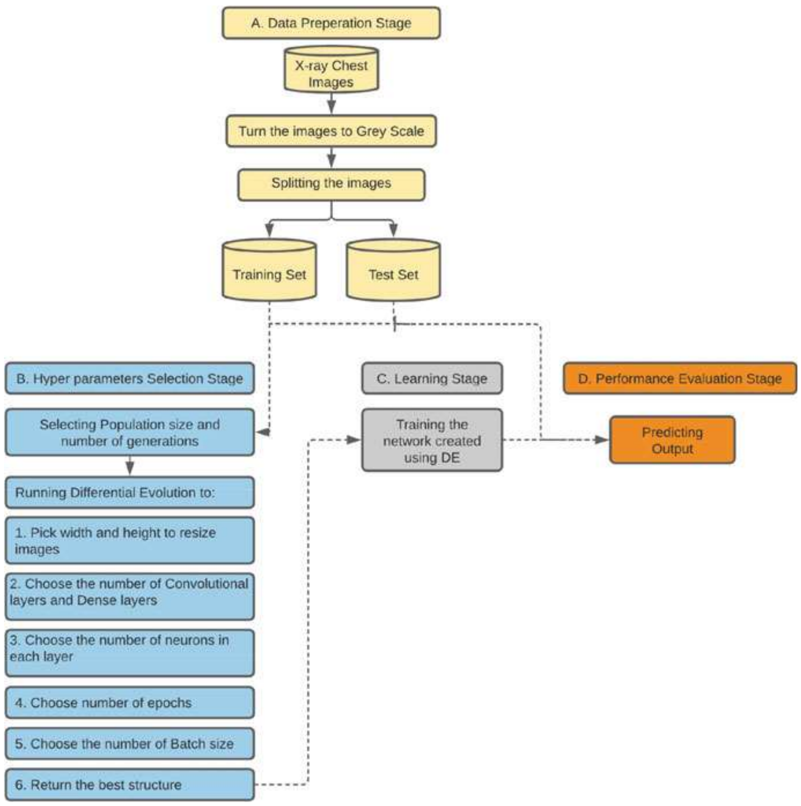


Fig. 18.6 The proposed framework

The previous process will be applied on the entire data and be split in the end according to a given parameter in the start that allows the developer to choose how he wants to split the data and to what ratios.

The differential evolution tries to optimize the number of layers given and the number of neurons in each layer as well. Since our Artificial Neural Network is a Convolutional Neural Network so we have three types of layers:

1. The Convolutional Layers.
2. The max pooling layers.
3. The dense layers.

we choose to consider the convolutional layers and the max pooling layers as one type of layers by which we create a max pooling layer of 2 * 2 after each convolutional layer (with a constant scanner of 3*3 for all the conv.) if it is possible due to the over pooling of the image might at some point cause some errors. So in order to prevent that problem from occurring we decided to handle it that once max pooling is not possible, we will stop creating convolutional layers and get on with creating the dense layers; yet this is not the way of how the number of convolutional layers and max pooling layers as picked; as the differential evolution algorithm will select a number of the convolutional layers as mentioned above and will then create one max pooling layer after each convolution layer and then create the dense layers.

In fact, the way the presented framework selects the best number of convolutional neural network for the mix of convolution layers (and max pooling) and the dense layers is by setting the number of minimum layers for the convolution layer to 1 and the minimum number of dense layers to 1 and a number is set as the maximum bounds for the total number of layers of both the convolution and dense layers combined to be set as not to exceed it.

For example, if the bounds of the number of layers for the model is 1 and 20; the layers can be 3 convolutional (with 1 max pooling after each) and 4 dense layers or 1 convolutional and 19 dense or 19 convolutional and 1 dense; so, the differential evolutional finds the best combination of the number of layers of the convolutional layers and the dense layers.

Actually, while creating the layers of each the convolutional and the dense layers, the differential evolution decides which is the best number of neurons in each layer of the convolutional and the dense layers according to some bounds that should not be less than the minimum bound and never exceed the maximum bound given in the framework.

Before creating the convolutional neural network model, the presented framework makes a few decisions according to the given data; the framework figures out the input shape for the input layer according to the images dimensions that is given, and thus the input layer is always dynamic and will work with any input shape as long as it is in gray scale, the model creates the layers and gives them an activation function of "ReLU" as the default for all the hidden layers, in the output layer we decide the number of outputs by examining the target values, if they are 2 classes then the output layer activation function will be set to 'sigmoid' and the loss function of the model will be "binary_crossentropy" but if they are more than 2 classes the output layer activation function will be set to "SoftMax" and the loss function will be set to "categorical_crossentropy" and the target data will be handled to be in the form of vectors (in case of more than 2 classes in the target data) if they are not in the form of vectors.

The presented framework handles the target data given that if its values are strings or characters, they are handled and turned into integer values for the sake of improvements and speed.

In order to measure the success and improvement of the different neural network structures and resized images created, the accuracy is considered as the default improvement measure but the improvement can be measured on any of the following as chosen while using the framework; the improvement can be done on: the training data alone, or the test alone, or the validation, or the training and test, or the training, test and validation or the training validation or the test validation in order to allow the user of the framework to focus on improving the model on all the kinds of data because one approach is to improve the test accuracy and the training accuracy will usually improve by default.

RMSE (root mean square error) is another measure that could be used instead of the accuracy. The presented framework gives the user the ability to use the accuracy and the RMSE combined as a measure of improvement.

In the end, the presented framework manages to find the best width and height that the images need to be resized to. The framework manages to find the best combination number of layers of convolutional layers to be used alongside with the dense layer as well as finding the best number of neurons for all the convolutional layers and the dense layers as well to create the convolutional neural network. Adam optimizer is used as the default optimizer in the framework.

18.5 Experimentation

This section represents the experimentation results for the presented framework presented in Sect. 18.4 and Fig. 18.6. Besides, experimentation results without using DE for hyperparameters selection is presented, in order to investigate the efficiency of using DE as a metaheuristic for hyperparameters selection.

18.5.1 CNN Using DE

The presented framework used a differential evolution algorithm as an optimization technique on Convolutional Neural Network on the COVID-19 detection using X-RAY images to optimize the hyperparameters of the network.

Data is composed of 2905 chest X-ray images, 1345 images are classified as normal cases, 1345 images are classified as Viral Pneumonia cases, 219 images are classified as COVID-19 cases, of size 1080 $\times$ 1080. Data was imported and the images were scaled to values between 0 and 1. Then, data were split to 80% for training and 20% for testing.

Differential Evolution is applied with a maximum number of iterations is set to 10 and population size =10 with the following constraints:

- Cost function is measuring the error in accuracy, so we try to reach the highest accuracy.
- Number of layers between 1 and 7.
- Number of neurons in each layer between 1 and 512.
- Batch size between 4 and 32.
- Number of epochs between 25 and 150.
- The width and height between 15 and 250.

Differential evolution algorithm returns the structure of the CNN neural network as follows:

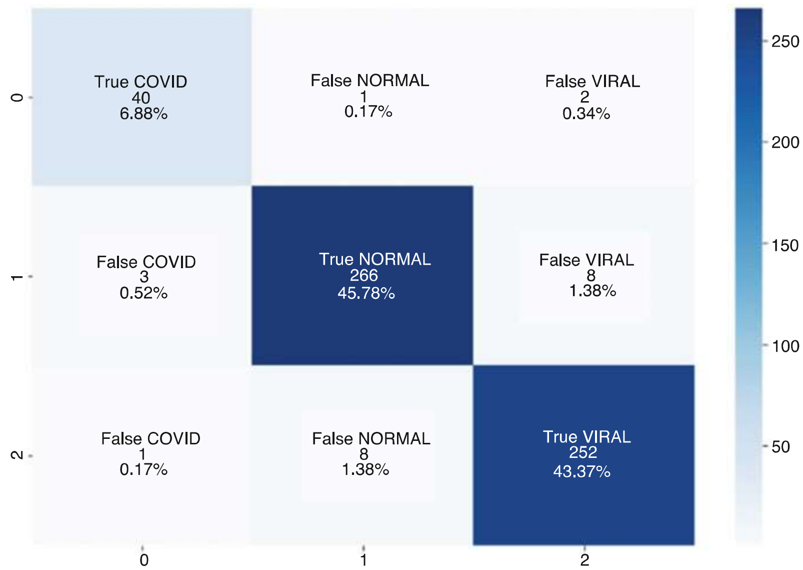


Fig. 18.7 Confusion matrix for CNN using DE

- One Conv2D layer for the input layer with shape **(width = 33 and height = 33)** with 64 Neurons with a 3 × 3 Scanner.
- One Max Pooling layer 2 × 2.
- One Conv2D Layer 134 neurons, 3x3 Scanner and activation function "ReLU".
- One Max Pooling layer 2 × 2.
- One flattens layer.
- One Dense Layer with 117 Neurons and activation function "ReLU".
- One output layer with 3 neurons and activation function SoftMax.

The total number of parameters of the current structure is 642,837 parameters, and the loss function of the model is categorical "crossentropy" and optimizer is Adam.

The Differential Evolution algorithm also returned a batch size of 72 and number of epochs equal to 125.

The resulting model was trained with the returned parameters and scored **100%** accuracy in the training and **97.25%** in the testing which proved that our differential evolution approach in the Convolutional Neural Network proved worthy by outperforming the compared various structures accuracy in the COVID-19 research area as well as it also proved to be better in finding a simpler structure (Fig. 18.7).

18.5.2 CNN without Using DE

Data is composed of 2905 chest x-ray images, 1345 images are classified as normal cases, 1345 images are classified as Viral Pneumonia cases, 219 images are classified as COVID-19 cases. Data were imported and the images were scaled to values between 0 and 1. Images were resized to 32×32 and split the data to 80% training and 20% testing.

The model structure is as follows:

- One layer of CNN with 15 neurons and scanner of 3×3 with the input shape of 32×32 (the width and height of the images) and an activation function "ReLU".
- One layer of max pooling of shape (2,2).
- One layer of CNN for the with 10 neurons and scanner of 3×3 and an activation function "ReLU".
- One layer of max pooling of shape (2,2).
- One layer of CNN with 8 neurons and scanner of 3×3 and an activation function "ReLU".
- One layer for flattening the neurons shape.
- One Dense layer with 16 neurons and activation "ReLU".
- One Dense layer for the output with 3 neurons.

The full structure is composed of 4353 parameters.

Class weights assigned of 1.3 weight adjustment for the COVID cases, 1 for the Normal, and 1 for the Viral Pneumonia.

"Adam" optimizer is used with learning rate equal to 0.0008, and the loss function is Sparse Categorical Crossentropy, and the model is trained for 150 epochs.

The model ends the final epoch with training accuracy of 99.14% and testing accuracy of 96.04%, where True COVID is 48, False COVID is 4, True NORMAL is 264, False NORMAL is 10, True Viral is 246, False Viral is 9, True COVID (92.30%), True Normal (96.35%), and True Viral (96.47%) (Fig. 18.8).

18.5.3 Comparison and Analysis

In this study, two models are presented to determine if the patient is affected by COVID-19 or not, one model using DE and the other without using DE. The performance of each model is evaluated and a comparative analysis with the most recent models from the literature (Khan and Aslam 2020) is presented in Table 18.1.

From Table 18.1, it could be easily observed that the proposed model outperformed other models in the literature except COVIDiagnosis-Net. Most of the previous research studies suffered from having a very limited number of COVID-19 images and imbalanced data.

The main contributions of the current research are:

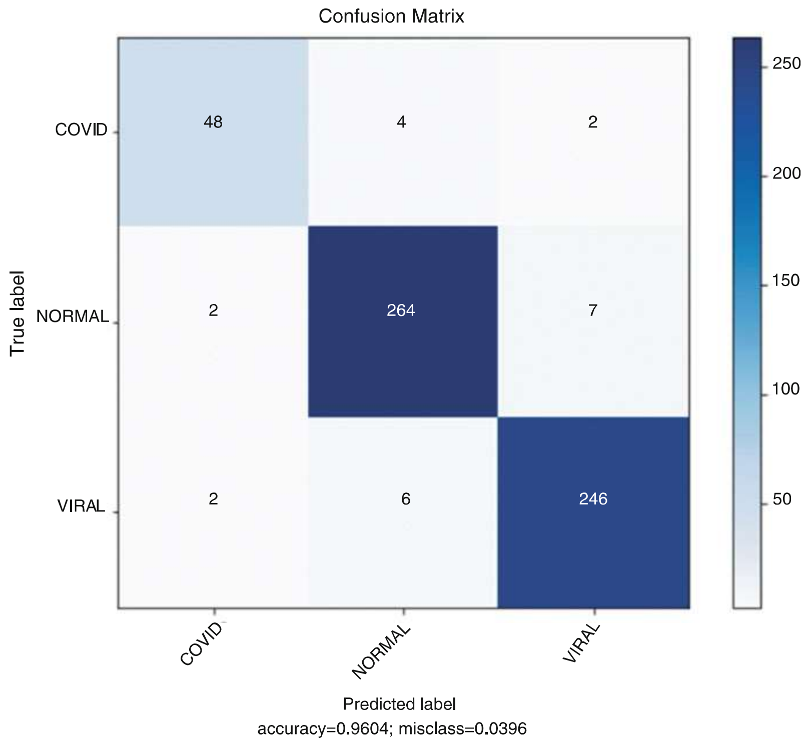


Fig. 18.8 Confusion matrix for CNN without using DE

1. The model does not suffer from imbalanced data.
2. The proposed model is a fully automated diagnosis method and does not require any separate feature extraction or annotation prior to the diagnosis.
3. Optimizing the hyperparameters of the model using Differential Evolution algorithm.
4. Less network structure as the number of parameters in the current suggested model is very simple, have very low number of parameters (296,000) compared to other complex models.
5. The model outperforms the previous models in the literature.

Table 18.1 Performance of the proposed models compared to models in the literature

Study	Number of samples	Technique	Testing accuracy	Sensitivity	Specificity
Hemdan et al. (2020)	50 (25 healthy, 25 COVID-19)	COVIDX-net	90%	–	–
Wang et al. (2020)	13,975 (normal, pneumonia, and COVID-19	Tailored CNN (COVID-net)	93.3%	91%	–
Narin et al. (2020)	100 (50 Normal, and 50 COVID-19)	ResNet50	96.1%	91.8%	96.6%
Farooq and Hafeez (2020)	2813 (1203 Normal, 931 bacterial pneumonia, 660 viral pneumonia, 19 COVID-19)	COVID-ResNet	96.23%	–	–
Ucar and Korkmaz (2020)	2839 (1203 Normal, 1591 pneumonia, and 45 COVID-19	COVIDiagnosis-net	98.30%	–	–
Apostolopoulos and Mpesiana (2020b)	1427 (224 COVID-19, 700 pneumonia, and 504 normal)	VGG19	93.48%	92.85%	98.75%
Proposed	2905 (1345 normal, 1345 viral pneumonia, and 219 COVID-19)	CNN	96.04%	99.24%	92.3%
		CNN-DE	97.25%	98.5%	93%

18.6 Conclusion

This paper presented a new approach called DE-CNN-COVID-19 that could be used to detect COVID-19 patients using chest images. The presented model starts with data preparation, followed by hyperparameters optimization using Differential Evolution algorithm, and ended with the learning phase and evaluation of performance.

Firstly, imbalanced data is handled, and the data set is divided into training and test sets. Secondly, Differential Evolution algorithm is used to optimize the hyperparameters of CNN and presenting the optimal structure of CNN. At last, the model is trained using the optimal structure obtained from Differential Evolution. The results obtained from the presented model were very promising and the model could be attaining 100% accuracy for training and 97.25% for testing.

The presented model was then compared with other models in the literature, and it proved the outperforming performance and the superiority over other models in detecting COVID-19.

References

Abbas A, Abdelsamea MM, Gaber MM (2020) 'Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network', medRxiv. medRxiv. Available at: http://arxiv.org/abs/2003.13815. Accessed: 16 January 2021

Afshar P et al. (2020) 'COVID-CAPS: a capsule network-based framework for identification of COVID-19 cases from X-ray Images', Pattern Recognition Letters. Elsevier B.V., 138, pp. 638–643. Available at: http://arxiv.org/abs/2004.02696. Accessed: 16 January 2021

Apostolopoulos ID, Aznaouridis S, Tzani M (2020) Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases. Journal of Medical and Biological Engineering 40(3):462–469. https://doi.org/10.1007/s40846-020-00529-4

Apostolopoulos ID, Mpesiana TA (2020a) Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine 43(2):635–640. https://doi.org/10.1007/s13246-020-00865-4

Apostolopoulos ID, Mpesiana TA (2020b) Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43 (2):635–640. https://doi.org/10.1007/s13246-020-00865-4

da Costa VG, Moreli ML, Saivish MV (2020) The emergence of SARS, MERS and novel SARS-2 coronaviruses in the 21st century. Arch Virol:1517–1526. https://doi.org/10.1007/s00705-020-04628-0

El-Din Hemdan, E., Shouman, M. A. and Karar, M. E. (2020) COVIDX-Net: a framework of deep learning classifiers to diagnose COVID-19 in X-Ray images

Farooq M, Hafeez A (2020) 'COVID-ResNet: a deep learning framework for screening of COVID19 from radiographs', arXiv. arXiv. Available at: http://arxiv.org/abs/2003.14395. Accessed: 5 April 2021

Gozes O et al. (2020) 'Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for Automated Detection & Patient Monitoring using deep learning CT image analysis', arXiv. arXiv. Available at: http://arxiv.org/abs/2003.05037. Accessed: 16 January 2021

Hemdan EE-D, Shouman MA, Karar ME (2020) 'COVIDX-net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images', *arXiv*. arXiv. Available at: http://arxiv.org/abs/2003.11055. Accessed: 5 April 2021

Kenneth VP, Rainer MS, Jouni AL (2005) Differential evolution–a practical approach to global optimization. Springer-Verlag, Berlin, Germany

Khan IU, Aslam N (2020) A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images. Information. Multidisciplinary Digital Publishing Institute 11(9):419. https://doi.org/10.3390/info11090419

Khan AI, Shah JL, Bhat MM (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Comput Methods Prog Biomed 196:105581. https://doi.org/10.1016/j.cmpb.2020.105581

Kumar Sethy P, Kumar Sethy P, Kumari Behera S (2020) Detection of coronavirus disease (COVID-19) based on deep features. https://doi.org/10.20944/preprints202003.0300.v1

Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B (2020a) Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest CT. Radiology:200905

Li L et al (2020b) Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy. Radiology 296(2): E65–E71. https://doi.org/10.1148/radiol.2020200905

Maghdid HS et al. (2020) 'Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms', arXiv. arXiv. Available at: http://arxiv.org/abs/2004.00038. Accessed: 16 January 2021

Mahdy LN et al. (2020) 'Automatic X-ray COVID-19 lung image classification system based on multi-level thresholding and support vector machine', medRxiv. medRxiv, p. 2020.03.30.20047787. https://doi.org/10.1101/2020.03.30.20047787

Mohamed AK, Hadi AA, Mohamed AW (2020) Generalized adaptive differential evolution algorithm for solving CEC 2020 benchmark problems. In: 2nd novel intelligent and leading emerging sciences conference, NILES 2020. Institute of Electrical and Electronics Engineers Inc, pp 391–396. https://doi.org/10.1109/NILES50944.2020.9257924

Mohamed AW, Hadi AA, Mohamed AK (2021) Differential evolution mutations: taxonomy, comparison and convergence analysis. IEEE Access. Institute of Electrical and Electronics Engineers Inc. 9:68629–68662. https://doi.org/10.1109/ACCESS.2021.3077242

Mohamed AK, Mohamed AW (2019a) Real-parameter unconstrained optimization based on enhanced AGDE algorithm. Studies in Computational Intelligence:431–450. https://doi.org/10.1007/978-3-030-02357-7_21

Mohamed AW, Mohamed AK (2019b) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybern 10(2):253–277. https://doi.org/10.1007/s13042-017-0711-7

Mohamed AW, Mohamed AK, Elfeky EZ, Saleh M (2019a) Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems. Int J Appl Metaheuristic Comp (IJAMC) 10(1):1–28. https://doi.org/10.4018/IJAMC.2019010101

Mohamed AK et al (2018) Enhancing AGDE algorithm using population size reduction for global numerical optimization. Advances in Intelligent Systems and Computing:62–72. https://doi.org/10.1007/978-3-319-74690-6_7

Mohamed AK et al (2019b) Solving constrained non-linear integer and mixed-integer global optimization problems using enhanced directed differential evolution algorithm. In: Studies in computational intelligence. Springer Verlag, pp 327–349. https://doi.org/10.1007/978-3-030-02357-7_16

Nagib,AE et al. (2020) 'Neural Network with Adaptive Learning Rate', in 2nd Novel Intelligent and Leading Emerging Sciences Conference, NILES 2020. Institute of Electrical and Electronics Engineers Inc., pp. 544–548. https://doi.org/10.1109/NILES50944.2020.9257880

Narin A, Kaya C, Pamuk Z (2020) 'Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks', arXiv. arXiv. Available at: http://arxiv.org/abs/2003.10849. Accessed: 16 January 2021

Rehman A et al. (2020) 'Improving coronavirus (COVID-19) diagnosis using deep transfer learning', medRxiv medRxiv, p. 2020.04.11.20054643. https://doi.org/10.1101/2020.04.11.20054643

Shan F et al (2020) Lung infection quantification of COVID-19 in CT images with deep learning. Med Phys. https://doi.org/10.1002/mp.14609

Singhal T (2020) A review of coronavirus Disease-2019 (COVID-19). Indian J Pediatr:281–286. https://doi.org/10.1007/s12098-020-03263-6

Storn R, Price K (1995) Differential Evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces

Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359. https://doi.org/10.1023/A:1008202821328

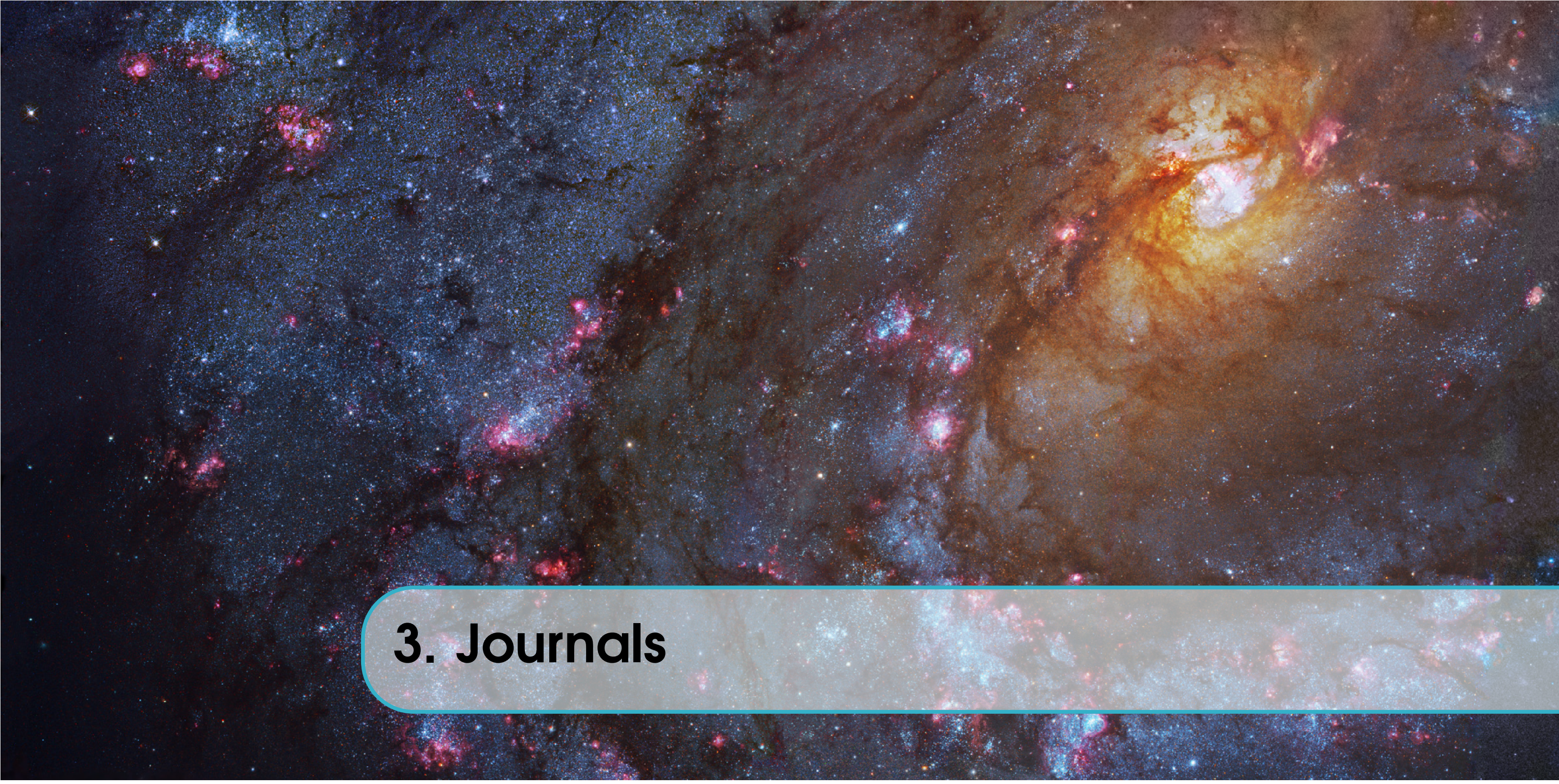
Tahamtan A, Ardebili A (2020) 'Expert Review of Molecular Diagnostics Real-time RT-PCR in COVID-19 detection: issues affecting the results'. https://doi.org/10.1080/14737159.2020.1757437

Ucar F, Korkmaz D (2020) COVIDiagnosis-net: deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. Med Hypotheses. Churchill Livingstone 140:109761. https://doi.org/10.1016/j.mehy.2020.109761

Wang L, Lin ZQ, Wong A (2020) COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep. Nature Publishing Group UK 10(1):1–12. https://doi.org/10.1038/s41598-020-76550-z

Wang S et al (n.d.) A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). https://doi.org/10.1101/2020.02.14.20023028

Xu X et al (2020) A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 6(10):1122–1129. https://doi.org/10.1016/j.eng.2020.04.010



3. Journals

3.1 Human Activity Recognition in Car Workshop (Q3)

Omar et al., presented project to reduce wasted time during work to solve the problem of human activity recognition within maintenance workshops and improve the system's ability to identify the time wasted by workers in activities. We start with the first problem, which is the human activity recognition, The recognition of human activity is a problem related to computer vision, and this makes it difficult to identify the human skeleton through different complex positions of the workers, such as while doing work in the car engine, the car's hood can block the view of the camera and if he was doing an activity under the car. [8]

Human Activity Recognition in Car Workshop

Omar Magdy
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA),
Giza, Egypt

Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA), Giza, Egypt

***Abstract*—Human activity recognition has become so widespread in recent times. Due to the modern advancements of technology, it has become an important solution to many problems in various fields such as medicine, industry, and sports. And this subject got the attention of a lot of researchers. Along with problems like wasted time in maintenance centers, we proposed a system that extracts worker poses from videos by using pose classification. In this paper, we have tested two algorithms to detect worker activity. This system aims to detect and classify positive and negative worker's activities in car maintenance centers such as (changing the tire, changing oil, using the phone, standing without work). We have conducted two experiments, the first experiment was for comparison between algorithms to determine the most accurate algorithm in recognizing the activities performed. The experiment was done using two different algorithms (1 dollar recognizer and Fast Dynamic time warping) on 3 participants in a controlled area. The one-dollar recognizer has achieved a 97% accuracy compared to the fastDTW with 86%. The second experiment was conducted to measure the performance of a one-dollar algorithm with different participants. The results show that a 1 dollar recognizer achieved an accuracy of 94.2% when tested on 420 different videos.**

Keywords*—*Machine learning; human activity recognition; pose identification; industry analysis

I. Introduction

Human activity recognition has gained importance in recent years because of its applications in various fields like health, security and surveillance, entertainment, and intelligent environments. Human activity recognition accounted a lot of researches in different approaches, like wearable devices [1][2][3][4][5],object-tagged[6][7][8], and device-free [9][10][11], to acknowledge human activities. Wasted time has become a prominent problem in various fields of work, and this problem affects the percentage of products and services required in our daily lives. The survey provided by Salary.com [12] found that 89% of workers admitted to wasting time at work every day. The survey showed that 61% claim to waste between 30 minutes to an hour a day. While this may not seem like much, it can add up to 5 hours a week or 260 hours a year/per employee. Maintenance places and factories are the most vulnerable to this issue. The survey depend heavily on human activity, which harms production rates and disrupts many required services. The field of car maintenance has become an important field in our daily life, and car workshops have spread greatly in the recent period, but difficulties have begun to appear inside the maintenance centers. The problem is that workers waste a lot of time while working by doing some negative activities. Fig. 1 shows a worker who uses the phone at work. Moreover, there are a lot of negative activities such as (eating, drinking, talking to others, standing without work...). Besides, it is difficult to have an employee responsible for monitoring the worker for no less than 12 hours.

Fig. 1. A Negative Wasting Time Activity (using Mobile while being in Work).

The maintenance workshop could have dozens of activities some of them are consider positive and other consider negative. However,some activities look very similar such as shown in Fig. 2(a) and Fig. 2(b). The posture of changing oil activity looks very similar in motion trajectory to engine rebuild activity.

(a) Changing oil (b) Engine rebuild

Fig. 2. A Sample for Positive Activity in Workshop.

The proposed system works by taking an input of recording videos of the worker's activities inside the maintenance center. The video is classified to determine the number of activities the worker performed such as (Changing the oil, changing the tire, or using the phone... Etc.). The classification was done by taking the path (x, y) of the important points in the skeleton as shown in Fig.3, Fig. 4 and comparing them with the points that were taken from the collector of the data set. The system can determine the type of activities that the worker performed in the video.

Fig. 3. Worker Changing Oil.

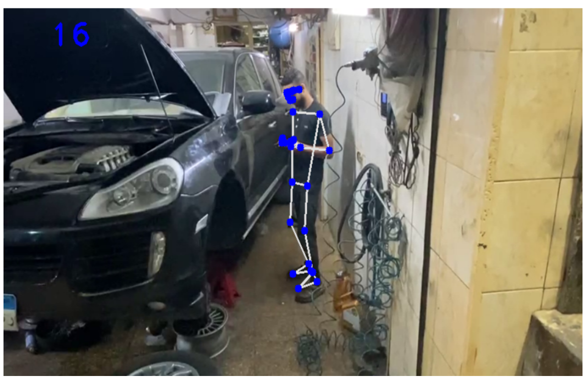


Fig. 4. Worker use Mobile.

The main contribution of this paper is to create a human activity recognition system that extract worker poses from the videos by using pose classification. The system used a dollar recognizer (1$) to detect worker positive and negative activities in maintenance workshop. As a result of the presence of many maintenance workshops, a problem occurred which is many workers neglect their work resulting in the occurrence of negative activities. Therefore, this paper is needed to detect the positive and negative activities and differentiate between them without the need for a person to monitor these activities and depend on the computer as an alternative. The purpose of this paper as well as to facilitate the reduction of the waste time done by each worker so they can be more productive. Moreover, positive activity is detected to reward the committed workers and to evaluate the performance of the workers in general.

II. Related Work

Due to the popularity of human activity recognition systems and with the rapid advancement of computer technology, lots of research efforts were dedicated to this subject. Congcong Liu et al. [6], has proposed a activity recognition method that can identify abnormal human activity recognition in surveillance video using a combination of Bayes Classifier and CNN (Convolution Neural Network) to detect the activities and also use KTH dataset as the input of Bayes Classifier and CNN. Another system that is able to recognize human activity is proposed by Bagate et al. [1], which identifies the activity using RGB-D sensors that is developed with deep learning model CNN (Convolution Neural Network) and using knight depth camera for capturing 3-D skeleton data. For Human detection and Motion tracking, Sandar et al. [8], used frame wise displacement and recognition is based on the skeletal model with the deep learning framework to understand human behavior in the indoor and outdoor environment. And in the field of Human activity recognition by sensors, Murat et al. [13], automatically identifies human activity using joint coordinates skeletons and uses two types of deep learning to make classification and use data set of multiple people in the images. Song et al. [5], propose (1D) Convolution Neural Network (CNN) -based method for recognizing the activities using collected accelerometer data from smartphones and this method gave high accuracy of 92.71. And using wearable devices, Tahera et al. [2], used the eSense accelerometer sensor to detect the matching of activity between the head and the mouth, from this collected data some activities of the head and mouth were identified and using the machine learning and deep learning for data classification. Nitin et al. [14], propose to use the TCN (temporal Convolutional Network) to recognize the activities because it better than other deep learning methods , it has strong ability to capture long-term dependencies. Godwin et al. [9], combined gyroscope sensors with accelerometers to detect human activity and perform analysis and recognition using ANN (artificial neural networks). Tsokov et al. [15], use of the 1D synaptic neural network (CNN) with accelerometer data to make recognition of human activity more accurate.

Isah et al. [3] collected hip motion from the different waist mounted sensors, and convert each signal into spectrum image and use them as input to the CNN (Convolution Neural Network). Nacer et al. [16], use entropy point estimate for 1D heat map to separate between human maps and animal maps to give high accuracy in human activity recognition. Selçuk et al. [10] used a novel design to reduce the number of sensors used for human activity recognition and detection by using (EMD) empirical mode decomposition.And Jiewen et al. [4], identified and interacted by focusing on two wearable cameras and the interactive activities that involve only two people. Peter Washington et al. [11], addressed the topic of identifying human activity in the treatment of autism and grouped movements with a handheld camera and used the classifier CNN (Convolution Neural Network) for detecting headbanging in home videos. Hristov et al. [17]proposed a method that classifies human activity by using 3D skeleton data and normalizing it beforehand and it was represented in 2 forms. They applied this method to the UTDMHAD dataset, the system has achieved a 92.4% accuracy rate. Heilym et al. [18] was opposed to the idea of wearing devices and sensors to determine human activities and pointed out that these devices could cause inconvenience to the bearer and could give false results if used in crowded places, so he relied on the camera and determining the activity through the human skeleton features. Salahuddin Saddar et al. [19]have an objective which is to compare some machine learning algorithms

TABLE I. COMPARISON WITH SIMILAR SYSTEMS

Paper	Activities	Algorithms	Methods	Area	Accuracy
[7]	stopping the furnace operating checking the solid fuel tank checking the gear motor and auger tightening the mounting screws of the gear motor	CNN CNN+SVM Yolov3	Image classification	controlled area	95.7%
[22]	grab tools hammers nail wrench use screwdriver	CNN	Image classification	controlled area	87%
This paper	changing tire changing oil use mobile stand without work	one-dollar Fastdtw	Pose classification	uncontrolled area	95%

that were used in human activity recognition such as (SVM, Decision Trees, Random Forests, XGBoost). They tested these algorithms with measurement sensor data that was recently released from the LARA dataset. The XGBoost has achieved the best accuracy with a rate of 78.6%. Ismael et al. [20] took the topic of identifying human activity in terms of reducing aggressive actions inside prisons and on the streets to reduce aggression and used "handcrafted/learned" as a hybrid feature framework that gave it very high accuracy rates. Yusuf Erkan et al. [21]used depth sensor to classify 27 different activities and by using long-short term memory, they analyzed skeleton data. It has achieved an accuracy rate 93%. Halikowski et al. [7], presented a system for monitoring activities inside the factory using (CNN, CNN+SVM, Yolov3) algorithms. They used some activities such as (stopping the furnace operating, checking the solid fuel tank, checking the gear motor and auger, tightening the mounting screws of the gear motor) and achieved an accuracy 94%. The work presented utilize deep learning for extracting features without considering human post estimation. Zhaozheng et al. [22] presented a system for detecting activities in smart manufacturing. They used some activities (grab tools, hammer nail, wrench use, rest lever, screwdriver). They captured these activities using IMU and sEMG signals obtained from a MYO armband. They extract feature using a convolutional neural network (CNN) model. The CNN model is evaluated on this data set and achieves 98% and 87% recognition accuracy in the half-half and leave-one-out experiments. All of the previous explained the importance of identifying human activity in solving some problems in various fields. Alghyaline et al. [23] has proposed system that detects different actions in the street such as (walking, running, stopping). They measure the movement type by using three different techniques which are (Yolo, Kalman filter, Homography). The method was tested by CCTV camera and BEHAVE dataset, it has achieved an accuracy of 96.9% for the Behave dataset and achieved 88.4% for the dataset that was collected by CCTV camera.Arzani et al. [24]proposed a structural prediction strategy proposed by this system to recognize the simple and complex actions by using probabilistic graphical models (PGMs). These activities require various model parametrization to be spanned, category-switching scheme is used to deal with this parametrization. Three datasets were used to cover the two action types which are (CAD-60, UT-Kinect, and Florence 3-D). This system could recognize simple and complex activities while the previous systems focused on only one type of these two. The system proposed by Archana et al.[25] recognized human activity with Resnet and 3D CNN without using the LSTM- attention model as the 3D CNN is achieved by modifying the 2D Resnet in order to achieve better accuracy, so that the development of detecting, and recognizing real-time human motion has been achieved.The system proposed by Zheng Dong et al. [26]resolves the issue of incomplete feature extraction by a new framework called CapsGaNet which proposed multi-feature extraction, and gated recurrent units (GRU) with attention mechanisms. The constructed dataset was a daily and aggressive activity dataset (DAAD). Moreover, the paper approved that CapsGaNet has efficiently improved the accuracy of recognition. Radhika V. et al. [27]proposed a system that used Random Forest Algorithm(RFA) to recognize human activity using Smartphones. RFA algorithm has different decision trees that is used in classification of the dataset. There were four various evaluation parameters used to measure the performance of the system such as F1 score, accuracy, precision, and sensitivity. The accuracy of the system achieved 98.34%. The system proposed by Navita et al. [28] detects the activity of aged people using the Internet of Things (IoT) monitoring model to monitor the activity of their health state. The SVM has attained 98.03%. The proposed system by Yin Tang et al. [29], a new CNN model that used hierarchical-split (HS) for a huge number of varieties in human activity recognition. Each one feature layer uses multi-scale feature representation by capturing a wide range of receptive fields of human activities. The proposed HS model can achieve high recognition performance compared to similar models complexities. The system achieved 94.10% SOTA accuracy on human activity recognition dataset.The proposed system by Maciej A. Noras et al. [30] discussed the topic of far-field electric field sensors, which accompany different physical events. The determination of activities in the proximity of the sensor is done by field signature signals. Moreover, the paper provided enhancements for electric field sensor usage and signal processing in human and animal motion recognition, perimeter monitoring, moving objects recognition, and electric power faults detection.

Table I shows a comparison between our system and different systems. The first system, Halikowski et al. [7] proposed this system to measure the performance of the worker in the factory, they recognized four different activities which are (stopping the furnace operating, checking the solid fuel tank, checking the gear motor and auger, tightening the mounting screws of the gear motor) in a controlled area using image classification method. They used more than one algorithm such as (CNN, CNN+SVM, Yolov3), their system has achieved a 95.7% accuracy rate. The second system which is proposed by Zhaozheng et al. [22] was used for qualification and evaluation of the workers. They also used image classification to detect four activities which are (grab tools, hammer nail, wrench use, rest lever, screwdriver) in a controlled area, and they achieved an accuracy rate of 87%. Our system was proposed to detect the activities of the worker inside the car maintenance center and differentiate between the negative and the positive activities. To help the workshop owner in measuring the worker performance, the system has used the pose classification method to detect four different activities which are (changing oil, changing tire, use mobile, stand without work). The system was used in an uncontrolled area and on different body characteristics such as(height and weight).

III. Proposed System

We presented a method that recognized and classified human activity in car workshops performed and captured from videos. It differentiated between the positive activities and the negative activities based on a comparison between input video and dataset stored in the templates. The proposed system used mediapipe for collecting key points of the skeletal joints and used the one-dollar and Fastdtw algorithms to classify the poses.

Fig. 5 shows the system overview, the system has two different way to input which is videos or live camera. The processing part starts with face recognition to differentiate between the workers because there is a large number of workers in the maintenance center. Then the mediapipe starts to extract the poses by calculating the path of each point in the skeleton. The mediapipe can extract 32 points, but this system focuses on extracting five important points which are (shoulder, elbow, wrist, hip, knee). The path of points was saved in a file to be ready for classification by the algorithms. The algorithms start to match these points with the points stored in the data set and send the results to the database to create a report that the managers and workshop owners can see.

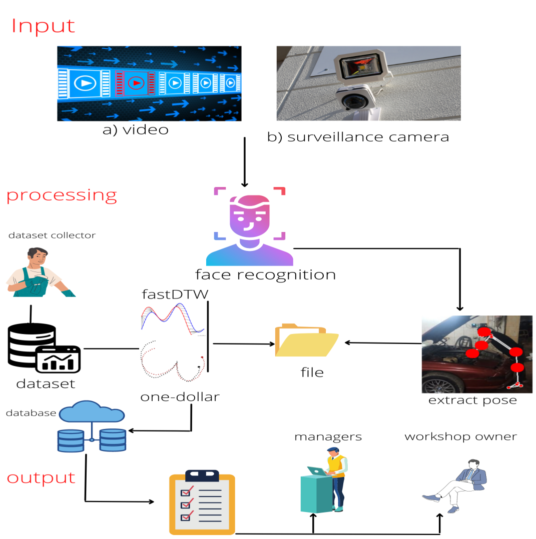


Fig. 5. System Overview.

The dataset was collected by recording videos of a specialist in the activities of the worker inside the maintenance workshop. These videos were entered into the mediapipe and OpenCV to extract the human pose from the videos showing the path of each point during the video. Each list of points has been saved in a file contains the points and the name of the activity. With the aid of a partner in the industry, we managed to collect 560 videos including the four activities which are (changing tires, changing oil, using a mobile, and standing without work), to start testing them and in order to have a huge number of data to help in testing.

A. FastDTW Algorithm

Fast Dynamic Temporal Warping (FastDTW) is a time series alignment algorithm that was initially designed for speech recognition. Its goal is to align two sequences of feature vectors by warping the time axis iteratively until an optimal match is discovered. The two sequences can be placed on the sides of a grid, one on top of the other, with the Y time-series axis on the top and the X time-series axis on the left. In each cell, a distance metric can be set, comparing the relevant elements of the two sequences. Path is detected through the grid by determining the best match alignment between the two sequences which are "I", and "j", and results in the minimum overall distance between each cell. Finding all different routes possibilities through the grid and computing the total distance regarding each one is the process for computing this total distance (D) as mentioned in Equation 1. The total distance is calculated by dividing the sum of the distances between individual items on the path by the weighting function's sum. This is achieved by making an equal number of points in each of the two series, then calculating the Euclidean distance between the first and each subsequent point in the first series.

$$D(i,j) = |t(i) - r(j)| + min \begin{cases} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{cases} \tag{1}$$

B. 1 Dollar Algorithm

The one-dollar is a geometric template matcher, the previously stored templates (T) are compared to the candidate strokes (C) resulting in the match that is the closest in 2-D Euclidean space as mentioned in Equation 2. Thus, we have exactly N points that will allow us to calculate the distance between C[k] to T[k] in which k= 1 to N. The most used pairwise point comparisons in the one-dollar algorithm are scale, rotation, and position invariant. One-dollar cannot differentiate between gestures whose identities rely on special orientations, ratios, and locations. One-dollar algorithm does not contain usage of time, as a result, gestures cannot be separated with respect to the speed.

$$d_i = \frac{\sum_{K=1}^{N} \sqrt{\left(C\left[K_x\right] - T_i\left[K_x\right]\right)^2 + \left(C\left[K_y\right] - T_i\left[K_y\right]\right)^2}}{N} \tag{2}$$

IV. Experiment

A. Experiment 1

The objective from conducted this experiment was to find out the most accurate algorithm in determining the activity performed by the worker inside the maintenance center. Besides, how it is estimated to differentiate between positive and negative activities.

This experiment was conducted on 3 participants by recording 3 video streams with a duration of 7 minutes for each video. The system has 84 videos of four different activities, including the positive and the negative activities in the different sequences. Each activity was repeated 7 times in each video 4 from the right side and 3 from the left side. The input to our system is split videos for each sequence of activities, the duration is 25 seconds for each video. consequently, After that, the videos were inserted into the system and extracted the important points in the skeleton to determine the path of each point (the X-coordinate and the Y-coordinate) using mediapipe, and save them in a file to be ready to be tested by the algorithms. The system calculated the accuracy of each algorithm in identifying and distinguishing between activities.

In this experiment, we were able to choose the best algorithm by testing each one of them separately, the one-dollar recognizer was able to give the best accuracy rate of 97% regarding the following activities: (changing tires, changing oil, using mobile, and standing without work). The rate of the fastDtw algorithm was 86% in the same previously mentioned activities as shown in Table II.

TABLE II. The Results of Experiment 1

Algorithms	Activites	Accuracy
One-dollar	changing tire changing oil use mobile stand without work	97%
Fastdtw	changing tire changing oil use mobile stand without work	86%

B. Experiment 2

This experiment was conducted to measure the performance of one-dollar algorithm with an increase in the number of participants.

We asked 15 participants to do a sequence of activities in different stream videos. The scenario of the activities was changing tire, use mobile, changing oil, stand without work and it was changeable from one to another. The average duration of the activity in the video was 30 seconds, Each activity was repeated 7 times 4 from the right side and 3 from the left side. The average age of the participants ranged from 19 to 23 years, and the characteristics of the body were also different. Afterwards, the videos were entered into the system, and the system was able to extract 420 videos for a range of different activities in uncontrolled environments. Through this experiment, we were able to measure the accuracy of the system in identifying activities in different conditions.

In this experiment, with large number of participants, experts in the field of mechanics, as well as workers from maintenance centers, the system was able to extract 420 videos, and the one-dollar algorithm started the stage of identifying activities. It achieved an accuracy rate of 94.2%.

C. Discussion

After the two experiments have been done, there's a discussion to explain why these results appear in both. The first experiment was to determine the most accurate algorithm, and the result was that one dollar was more accurate than the fastdtw, which is because the One dollar recognizer has been built over fast dynamic time warping (fastDTW), and both can determine path differences between two trajectories. But the difference between them is that one dollar recognizer reduces the noise in orientation by taking the angles into its calculations, which gives preference to one dollar in the accuracy ratios.The second experiment was to test the performance of the one-dollar algorithm in determining the activity of a large number of participants. The results showed that there was a significant difference in accuracy ratios between the activities. The activity of changing the tire was more subtle because this activity differed from the other activities in the motion trajectory, and the activity of using mobile and standing without work had high accuracy, but there were few similarities between them. The oil change activity has a similar motion trajectory with the two activities (use mobile, standing without work), which reduced its accuracy as shown in Fig. 6.

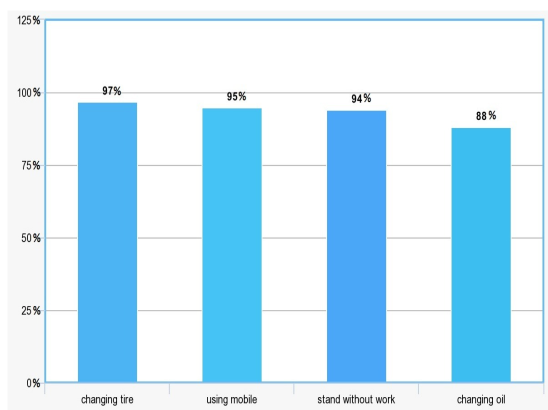


Fig. 6. The Results of the Activities.

The aim of this paper was achieved by using the method of pose classification to extract the pose points from the input video, after extracting the points, a one-dollar algorithm is used to match the extracted points with the collected dataset. As a result, we detect the activities and classify them whether they are positive activities or negative activities. We tested the previous method on 18 participants to prove the accuracy of the mentioned method which reached 94.2%.

V. Conclusion and Future Work

In this paper, we proposed a human activity recognition system in a car workshop that would help maintenance center owners to identify positive or negative worker activities correctly and efficiently. Our system used pose classification to extract workers poses and used two different algorithms to detect the activities. The system achieved an accuracy rate of 94.2% using this method. However, we are confident that the accuracy of this system will improve with more test videos and that we will increase the activities on which the experiment was conducted in the future. Our future work focus on calculating the wasted time of each worker. Besides, solving problems such as obstacles (engine hood) that appear while identifying activities and affecting the accuracy of the system.

REFERENCES

[1] A. Bagate and M. Shah, "Human activity recognition using rgb-d sensors," in *2019 international conference on intelligent computing and control systems (ICCS)*. IEEE, 2019, pp. 902–905.

[2] T. Hossain, M. S. Islam, M. A. R. Ahad, and S. Inoue, "Human activity recognition using earable device," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 81–84.

[3] I. A. Lawal and S. Bano, "Deep human activity recognition using wearable sensors," in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2019, pp. 45–48.

[4] J. Zhao, R. Han, Y. Gan, L. Wan, W. Feng, and S. Wang, "Human identification and interaction detection in cross-view multi-person videos with wearable cameras," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2608–2616.

[5] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 ieee international conference on big data and smart computing (bigcomp)*. IEEE, 2017, pp. 131–134.

[6] C. Liu, J. Ying, F. Han, and M. Ruan, "Abnormal human activity recognition using bayes classifier and convolutional neural network," in *2018 IEEE 3rd international conference on signal and image processing (ICSIP)*. IEEE, 2018, pp. 33–37.

[7] J. Patalas-Maliszewska, D. Halikowski, and R. Damaševičius, "An automated recognition of work activity in industrial manufacturing using convolutional neural networks," *Electronics*, vol. 10, no. 23, p. 2946, 2021.

[8] S. Win and T. L. L. Thein, "Real-time human motion detection, tracking and activity recognition with skeletal model," in *2020 IEEE Conference on Computer Applications (ICCA)*. IEEE, 2020, pp. 1–5.

[9] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proceedings of the 2018 10th international conference on machine learning and computing*, 2018, pp. 41–46.

[10] S. Sezer and E. Surer, "Information augmentation for human activity recognition and fall detection using empirical mode decomposition on smartphone data," in *Proceedings of the 6th International Conference on Movement and Computing*, 2019, pp. 1–8.

[11] P. Washington, A. Kline, O. C. Mutlu, E. Leblanc, C. Hou, N. Stockham, K. Paskov, B. Chrisman, and D. Wall, "Activity recognition with moving cameras and few training examples: Applications for detection of autism-related headbanging," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[12] Z. Hussain, Q. Sheng, and W. E. Zhang, "Different approaches for human activity recognition: A survey," 06 2019.

[13] M. Yalçın, N. Tüfek, and H. Yalcin, "Activity recognition of interacting people," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[14] N. Nair, C. Thomas, and D. B. Jayagopi, "Human activity recognition using temporal convolutional network," in *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*, 2018, pp. 1–8.

[15] S. Tsokov, M. Lazarova, and A. Aleksieva-Petrova, "Evolving 1d convolutional neural networks for human activity recognition," in *International Conference on Computer Systems and Technologies' 21*, 2021, pp. 49–54.

[16] A. Naser, A. Lotfi, J. Zhong, and J. He, "Human activity of daily living recognition in presence of an animal pet using thermal sensor array," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–6.

[17] P. Hristov, A. Manolova, and O. Boumbarov, "Deep learning and svm-based method for human activity recognition with skeleton data," in *2020 28th National Conference with International Participation (TELECOM)*. IEEE, 2020, pp. 49–52.

[18] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall detection and activity recognition using human skeleton features," *IEEE Access*, vol. 9, pp. 33 532–33 542, 2021.

[19] A. S. Syed, Z. S. Syed, M. S. Shah, and S. Saddar, "Using wearable sensors for human activity recognition in logistics: A comparison of different feature sets and machine learning algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110976

[20] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight recognition in video using hough forests and 2d convolutional neural network," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4787–4797, 2018.

[21] Y. E. Görgülü and K. Taşdelen, "Human activity recognition and temporal action localization based on depth sensor skeletal data," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 1–5.

[22] W. Tao, Z.-H. Lai, M. C. Leu, and Z. Yin, "Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks," *Procedia Manufacturing*, vol. 26, pp. 1159–1166, 2018.

[23] S. Alghyaline, "A real-time street actions detection," *International Journal of Advanced Computer Science and Applications*, vol. 10, 01 2019.

[24] M. Arzani, M. Fathy, A. Azirani, and E. Adeli, "Switching structured prediction for simple and complex human activity recognition," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–12, 01 2020.

[25] N. Archana and K. Hareesh, "Real-time human activity recognition using resnet and 3d convolutional neural networks," in *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*. IEEE, 2021, pp. 173–177.

[26] X. Sun, H. Xu, Z. Dong, L. Shi, Q. Liu, J. Li, T. Li, S. Fan, and Y. Wang, "Capsganet: Deep neural network based on capsule and gru for human activity recognition," *IEEE Systems Journal*, 2022.

[27] V. Radhika, C. R. Prasad, and A. Chakradhar, "Smartphone-based human activities recognition system using random forest algorithm," in *2022 International Conference for Advancement in Technology (ICONAT)*. IEEE, 2022, pp. 1–4.

[28] P. Mittal *et al.*, "Machine learning (ml) based human activity recognition model using smart sensors in iot environment," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2022, pp. 330–334.

[29] Y. Tang, L. Zhang, F. Min, and J. He, "Multi-scale deep feature learning for human activity recognition using wearable sensors," *IEEE Transactions on Industrial Electronics*, 2022.

[30] M. Noras, "Activity detection and recognition with passive electric field sensors," *IEEE Transactions on Industry Applications*, 2021.

4. Conferences

4.1 Summer 2020 - Fall 2020

4.1.1 Neural Network with Adaptive Learning Rate (Summer 2020)

Two students of Computer science Abdel Rahman Ezz Eldin and Mostafa Mohamed Saeed for publishing their paper with cooperation and under supervision Of Dr. Ali Khater and Eng. Shereen El-Feky. Their paper was under title of "Neural Network with Adaptive Learning Rate". [14]

Proceedings of NILES2020:
2nd Novel Intelligent and Leading Emerging Sciences Conference

Neural Network with Adaptive Learning Rate

Abdelrahman Ezzeldin Nagib[1], Mostafa Mohamed Saeed[1], Shereen Fathy El-Feky[1], Ali Khater Mohamed[1]
[1]Department of Computer Science, Faculty of Computer Science, October University for Modern Sciences and Arts (MSA),
6th October city, Giza 12451, Egypt
abdelrahman.ezzeldin1@msa.edu.eg
mostafa.mohamed52@msa.edu.eg
selfeky@msa.eun.eg
akhater@msa.eun.eg

Abstract*—** Over the last two decades, the neural network has surprisingly arisen as an efficient tool for dealing with numerous real-life applications. Optimization of the hyperparameter of the neural network attracted many researchers in industrial and research areas because of its great effect on the quality of the solution. This paper presents a new adaptation for the learning rate with shock (ALRS***) as the learning rate is considered one of the most important hyperparameters. The experimental results proved that the new adaptation leads to improved accuracy with a simpler structure for the neural network regardless of the initial value of the learning rate.

Keywords*— *Artificial Neural Network, Adaptive Learning Rate, Breast Cancer, Hyper-Parameter optimization.

I. Introduction

Since the invention of the neural networks and with the rapid increase in data everywhere and the recent increasing call of the terms of data analysis, classification, and forecasting. Artificial Neural Network (*ANN*) became widespread and commonly used especially in modern machine learning regarding its efficient results with high accuracy, which attracted a lot of researchers to use *ANN* for solving different types of problems in different fields such as engineering, medicine, and physics [1].

Even though the artificial neural network has proposed a lot of breakthroughs but it seems as a black-box Predictor, because of the increase of the number of its parameters which rely on the size of the neural network as (Number of Hidden Layers, Number of neurons of each layer, learning rate). *ANN* structure with a diversity of its parameters is very provoking rich environment for researches, especially in the direction of improving the performance out of practicing some modification applied on the *ANN* algorithm via lot of optimization techniques that might be applied on *ANN* that may improve its overall performance which affect directly its performance.

Optimization of the hyperparameters of the neural network starting from the number of hidden layers, number of neurons in each layer, the optimizers used, and the learning rate of the optimizer; is still a challenging area of research.

The learning rate plays an important role in the accuracy of the neural network during training. Many attempts have been made to obtain the optimal value of the learning rate. Either by trial and error or by adapting the learning rate.

Many trials to optimize the learning rate of the optimizers algorithm were done but through an exhaustive search by creating a big pool of learning rate and usually this learning rate is given as discrete values between 0 and 1 and then the model is trained using each learning rate to figure which is the best learning rate, this is considered computationally very expensive to try the entire pool of learning rate and will be computationally more expensive as the pool size increases, dataset size increases, complex neural networks, and the more generations the network have [2].

Thus making the optimization of the learning rate very expensive and time-consuming, while many others depend on setting the learning rate to a small value and counting on the momentum in the optimization algorithms to help escape local minimums yet it doesn't always converge to the global minimum and takes time to calculate and make an expense for calculating the momentum.

Others do an exhaustive search like the learning rate but with the momentum to find the best learning rate along with the best momentum for the optimization algorithm by creating two pools, one for the learning rate and one for the momentum, which leads to the very long processing time to do all the combinations among the two pools [2].

The remainder of the paper is organized as follows. section 2 presents a review on learning rate schedules from the literature. The proposed framework is presented in section 3. Section 4 represents the simulation. Results and discussion are presented in section 5. Finally, section 6 presents a work summary and conclusion.

II. Literature Review

Recently, different adaptive learning rate frameworks were presented. [3] Presented a learning rate scheme which is based on computing the Hessian of the loss function, the proposed framework is called AdaDelta. WNGrad is presented in [4], in which a weight normalization is used to adapt the learning rate. AdaBound and AMSBound are presented in [5], which depends on the moment for adapting the learning rate.

Learning rate schedules in the literature could be presented in four categories [6]:

1. Constant learning rate: in which the learning rate η is set initially to a constant number that is multiplied by the gradient and gradient may not be scaled during the learning process [7], [8].
2. Scheduled learning rate: where learning rate is not constant during the training, but a function that decreases as learning progress either step-based, or time-based [7], [9]–[11].
3. Adaptive optimization methods with the effect of changing learning rate: such as Adagrad and RMSProp [6], [12]–[18].

978-1-7281-8226-1/20/$31.00 ©2020 IEEE

4. Finally, combined adaptive optimization methods and adaptive learning rate [8], [19]–[21].

Unfortunately, despite the increased accuracy presented in many of researches, there is still some omission for the network complexity constraints which is a very important direction that should not be ignored, and this was the motivation for this research.

III. PROPOSED FRAMEWORK

Over the past years, a lot of research was presented to increase the efficiency of *NN*. One challenging task was the optimization of the hyperparameters of the neural network (number of hidden layers, number of neurons in each layer, the optimizers used, and the learning rate of the optimizer). One of the most important hyperparameters is the learning rate (*LR*).

A. Learning rate

LR is a positive scaler used in stochastic gradient descent (*SGD*) in order to determine the step length [22]. For the necessity of enhancing the accuracy, the *LR* must be adjusted manually during the training process [23]. In most cases, researchers have tried to optimize the learning rate by creating a big pool of learning rates that usually given as discrete values between 0 and 1. The model is trained using each learning rate to figure which is the best learning rate [2].

This is computationally very expensive to try the entire pool of learning rate and will also be computationally more expensive as the pool size increases, dataset size increases, complex neural networks, and the more generations the network has; thus making the optimization of the learning rate hyperparameter very expensive and time-consuming.

So, the choice of a good constant *LR* is a very important but tricky task. The choice of a proper constant *LR* will train the network to a passable but unsatisfactory accuracy.

LR schedule or varied *LR* over the training process is considered as an alternative choice for the fixed *LR*. In *LR* schedule, the performance or structure of the model controls the adaptation process of the *LR* supported by a learning algorithm [24].

For many researchers, linear *LR* decay is considered as the most popular choice for scheduling the *LR*, in which the *LR* is changed gradually based on time or step during the training process. The linear decay function for the learning rate is $LR = \frac{LR_0}{1+dt}$, where LR_0 and d represents the initial learning rate and the decay rate respectively [2].

Due to the aforementioned problems with the constant *LR* and directed by the adaptation importance of the *LR*, a new adaptation scheme for the learning rate is presented to enhance the accuracy with more simplified structure for the network.

B. Adaptive Learning rate

SGD is considered as the base for many updating approaches. In *SGD*, the parameters updated for each sample or mini patch by the following formula:

$$\theta_{t+1} = \theta_t - \eta.\nabla_{\theta t}E(\theta t; x^{(k)}; y^{(k)}) \qquad (1)$$

Where θ holds the *NN* parameters (weights and biases), $x^{(k)}$ and $y^{(k)}$ are input with training sample (k) and the label respectively, *E* is the loss function and η is the learning rate.

Increased value of the learning rate η increases the exploration and finding a new promising region for the optimal value for η but may not lead to converge as the convergence will be very slow. On the contrary, small values for η will speed up the convergence but may cause the loss of diversity and may lead to premature convergence or stagnation.

So, in order to avoid premature convergence or stagnation, most updating methods were designed to decrease the learning rate during training phase. Actually, this helped in narrowing the search space and start to explore the narrow region for the optimal value of η, but it becomes very difficult to reach the optimal as the small region to explore may contain lots of local optima.

Therefore, a combination of decreasing the learning rate and increasing the learning rate by shock is a novel promising idea for handling the problems of premature or stagnation and to be able to skip the locals in the search space and find the optimal value for η.

A new learning rate update is proposed on the basis of the loss function. In which the learning rate is initialized randomly by only one single value between 0 and 1, i.e. $\eta \in U[0,1]$, and the learning rate is updated during the training phase every predetermined *(t)* epochs and the model accuracy is less than a predetermined satisfactory level of accuracy *(a%)* in order to preserve the stability through the following equation:

$$\eta_{t+1} = \left|\frac{l_{t+1} - l_t}{l_t}\right| \qquad (2)$$

Where l_t and l_{t+1} are the values for the loss function in the previous epoch and the current epoch, respectively. This means that the value of η is initially random and decreased in the same decreasing rate of change in the loss function. In the early stages of the training process, the learning rate is high as the change in loss function is high and decreased in later stages as the rate of change in the loss function decreased.

Premature convergence implies stuck in a local optimum, whereas stagnation describes losing diversity. So, in both cases there is no improvement in the solution. Any of the two cases may happen due to the adaptation of the learning rate. Therefore, a novel mechanism is proposed in order to avoid the problems of premature and/or stagnation called (Shock). In Shock, an archive of size *(n)* contains the last values for the loss function in each iteration is created. The learning rate (η) is given a shock using the following equation:

$$\eta_{t+1} = \eta_t * \mathrm{Q} \qquad (3)$$

If the next condition holds:

$$\frac{\sum_{i=1}^{n} archive_i}{n} > l_{t+1} \qquad (4)$$

Where $archeive_i$ contains the values of the loss function in an archive of size *(n)*, *Q* is a certain threshold value (maximum value for η). Eqn.(4) indicates that the average for the last n values for the loss function is greater than the current value of the loss function, which means that the current value of η may cause a premature or stagnation. Therefore, shock is needed by Eqn.(3) to change the current value of η to a new range (between 0 and *Q*) of larger value in order to skip the local optima.

The experimentations proved that the proposed novel idea is very efficient in terms of stability, robustness, and accuracy of the model with much simpler architecture for the network.

C. Dimensionality Reduction

Dimensionality reduction is performed in this work as it is considered an important process required to increase the efficiency of the model.

One of the most important techniques for dimensionality reduction is Principal Component Analysis (*PCA*), in which singular value decomposition of the data is used in order to project the data to lower dimensional space by finding the components with maximum variance.

Given a statistical distribution of data in an L-dimensional space, *PCA* examines the distribution properties and obtained what so-called "main components", the components that maximize the variance, and are a linear combination of random variables that maximize the variance in relation with eigenvalues of the covariance matrix.

IV. SIMULATIONS

In order to test the efficiency and accuracy of the proposed framework (***ALRS***), the Wisconsin Breast Cancer dataset is used [25].

A. Problem scope

Recently, breast cancer has been become one of the highest causes of high death rate between women, A lot of researches said that between every eight women there is one woman diagnosed as a breast cancer patient. This was a powerful motive for us to apply our proposed model on breast cancer detection as a case study.

Breast cancer could be proposed as some of the rapidly growing cells that have been converted to a lump which converted to be a tumor. We can classify Tumors as malignant or benign.

B. Implementation details

The model is trained and tested on the Wisconsin Breast Cancer dataset [25] which consists of 569 records with 32 features. Dataset is divided to 90% for the training and 10% for the validation.

PCA is applied to the correlation matrix in Figure 1 in order to reduce the dimensionality to the 10 most important features with an eigenvalue greater than 1. The eigenvalues accumulate 94.19% of the total variance as presented in Table I.

TABLE I. TOTAL VARIANCE

Component	***Initial Eigenvalues***		
	Total	***% of Variance***	***Cumulative %***
1	*13.288*	*42.865*	*42.865*
2	*5.697*	*18.377*	*61.241*
3	*2.835*	*9.146*	*70.388*
4	*1.981*	*6.391*	*76.779*
5	*1.649*	*5.319*	*82.098*
6	*1.235*	*3.983*	*86.081*
7	*.978*	*3.156*	*89.237*
8	*.672*	*2.167*	*91.404*
9	*.461*	*1.486*	*92.890*
10	*.403*	*1.300*	*94.190*

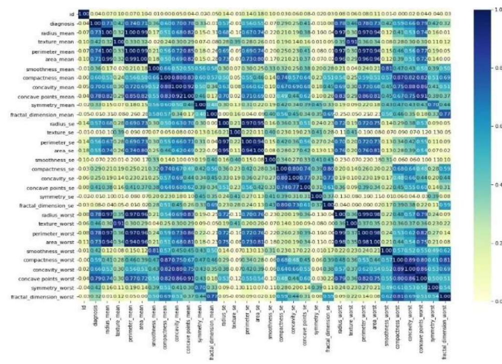


Figure 1. Correlation matrix

The structure of the *NN* consists of 10 neurons in the input layer (output from *PCA*), a dense hidden layer of 20 hidden neurons with the activation function *ReLU*, the output layer is a dense layer with 1 neuron and *sigmoid* activation function.

Learning rate η is randomly initialized between 0 and 1. The model is trained using *SGD* (Stochastic Gradient Descent) optimizer starting with the initial random learning rate and updated by Eqn. (2,3 and 4), with momentum = 0, the loss function used = 'binary_crossentropy', Q=2 and n=4. Number of epochs = 30, and number of iterations = 35.

Class weights are assigned to be 75% weight change in case of a Benign and 25% weight change in case it is a Malignant as classes are skewed and the Benign appears less than the Malignant.

The performance of the model is measured in terms of measurement indices which has been showing via the confusion matrix, accuracy, sensitivity, and specificity. Eqn. (5, 6 and 7) are used to obtain the metrices. Table II represents the confusion matrix.

TABLE II. CONFUSION MATRIX

		Predicted Class	
		Positive	***Negative***
Actual Class	***Positive***	*TP*	*FN*
	Negative	*FP*	*TN*

Where:
TP represents that observation is Correctly Identified
TN represents that observation is Incorrectly Identified
FP represents that observation is Correctly Rejected
FN represents that observation is Incorrectly Rejected

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

$$Specificity = \frac{TN}{FP+TN} \quad (6)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

Accuracy measures the classifier's ability to diagnosis accurately.

Specificity measures the model's ability to separate the target class.

Sensitivity measures the model's ability to accurately identify the occurrence of the target class.

RMSE (Root Mean Square Error) is used as another measure for estimating the accuracy of the model and to assure that no overfit in the proposed model.

All experiments were implemented and executed using python 3.7 on anaconda Jupiter notebook on a PC core i7-9750 hexa-core processor with 16 GB RAM and 6 GB Nvidia GTX 1660 TI graphics card.

V. RESULTS AND DISCUSSION

A. *Results of the proposed framework* ***(ALRS)***

The experimental results for breast cancer classification using the proposed model yielded a training accuracy **100%** as depicted in Figure(2) with validation accuracy **98.25%** which is very remarkable and proves the capability of the proposed model, as a remarkable accuracy is achieved with more simple *NN* architecture and adaptive learning rate with shock regardless of the initial value for the learning rate η. The optimal value obtained for the learning rate η is **0.315910**

RMSE (root mean square error) for both training and validation data are **0.0204** and **0.0823** respectively which excludes any chances of overfitting the model.

The performance of the proposed model is evaluated using the performance measurement indices through the confusion matrix. The results are presented in Table III.

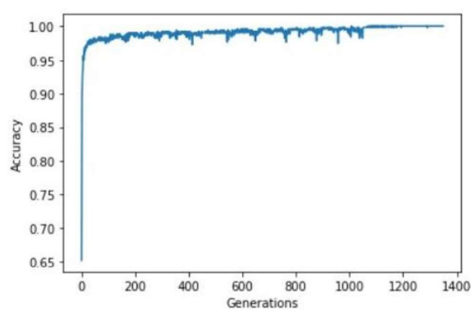


Figure 2. Training accuracy

TABLE III. PERFORMANCE MEASURE INDICES

Parameter	***measure***
Accuracy	*98.24%*
sensitivity	*95.45%*
Specificity	*100%*

B. *Comparison with other methods*

A comparison of the proposed model with other work from the literature [26] that applied to the Wisconsin breast cancer database [25] is presented in Table IV.

It is clear from Table IV that the proposed model ***(ALRS)*** is better than the other approaches in terms of the classification accuracy. Except for *GONN3* and *GONN4*, that used an *ANN* that is genetically evolved to an optimal architecture (structure and weight) through the concept of Genetic Programming with proposed crossover and mutation operators. We reached the same accuracy *100%* but with a simpler structure.

TABLE IV. COMPARISON WITH OTHER ALGORITHMS FROM LITERATURE

Method	***Classification Accuracy %***
C4.5	*94.74*
RAIC	*95.00*
Neuro-fuzzy	*95.06*
Fuzzy-GA	*97.36*
LSA machine	*98.80*
Supervised fuzzy clustering	*95.57*
Fuzzy-AIRS	*98.51*
SVM	*99.54*
LS-SVM	*98.53*
CFW	*99.50*
Real Coded GA	*96.50*
AMMLP	*99.26*
Decision Tree algorithms	*92.97*
ICA	*97.75*
RF-ANN	*98.05*
PSO (4-2)	*94.74*
GONN1	*98.24*
GONN2	*99.63*
GONN3	***100***
GONN4	***100***
ALRS (proposed model)	***100***

VI. CONCLUSION

The heuristic search for the optimal values of learning rate is proved to be computationally expensive. Thus, a novel framework ***(ALRS)*** for learning rate adaptation based on the loss function is presented in this research given that the initial value for learning rate is a uniform random number between 0 and 1 and the momentum value is zero. For a better balance between exploration and exploitation for the search space to find the optimal learning rate, a ***shock*** idea is proposed to prevent the premature convergence or stagnation. i.e. avoid poor local solutions. PCA is augmented within the framework for dimensionality reduction purposes.

The framework is tested on the Wisconsin Breast Cancer dataset. The experimental results proved that ***(ALRS)*** is able to achieve *100%* accuracy with *RMSE* 0.0204 and 0.0823 for training and validation respectively, and ***(ALRS)*** is capable of skipping the area leading to poor local solutions. It is also very stable as the model is able to achieve the desired accuracy regardless of the initial starting value for the learning rate.

The results are compared with 20 models from the literature that used the Wisconsin Breast Cancer dataset. The comparison confirmed that the proposed model can achieve higher accuracy using a very simple structure for the network.

REFERENCES

[1] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11. Elsevier Ltd, p. e00938, 01-Nov-2018.

[2] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," *arXiv preprint arXiv:2003.05689*, Mar. 2020.

[3] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv preprint arXiv:1212.5701*, Dec. 2012.

[4] X. Wu, R. Ward, and L. Bottou, "WNGrad: Learn the Learning Rate in Gradient Descent," *arXiv preprint arXiv:1803.02865,* Mar. 2018.

[5] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive Gradient Methods with Dynamic Bound of Learning Rate," *7th Int. Conf. Learn. Represent. ICLR 2019*, Feb. 2019.

[6] J. Park, D. Yi, and S. Ji, "A novel learning rate schedule in optimization for neural networks and it's convergence," *Symmetry (Basel).*, vol. 12, no. 4, p. 660, Apr. 2020.

[7] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," Springer, Berlin, Heidelberg, 2012, pp. 437–478.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[9] Y. Nesterov, "Introductory Lectures on Convex Programming Volume I: Basic course," *Lecture notes* 3.4. 1998.

[10] R. Ge, S. M. Kakade, R. Kidambi, and P. Netrapalli, "The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares," *Advances in Neural Information Processing Systems*. 2019.

[11] Z. Li and S. Arora, "An Exponential Learning Rate Schedule for Deep Learning," *arXiv preprint arXiv:1910.07454.* Oct. 2019.

[12] J. Duchi JDUCHI and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research* 12.7 2011.

[13] T. Tieleman, G. H. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." *COURSERA: Neural networks for machine learning* 4.2 (2012): 26-31.

[14] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv preprint arXiv:1212.5701* Dec. 2012.

[15] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[16] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, Apr. 2019.

[17] D. Yi, J. Ahn, and S. Ji, "An Effective Optimization Method for Machine Learning Based on ADAM," *Appl. Sci.*, vol. 10, no. 3, p. 1073, Feb. 2020.

[18] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2. Society for Industrial and Applied Mathematics Publications, pp. 223–311, 08-May-2018.

[19] N. Le Roux and A. Fitzgibbon, "A fast natural Newton method," *ICML*. Jan. 2010.

[20] J. Sohl-Dickstein, B. Poole, and S. Ganguli, "Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods." *International Conference on Machine Learning*. 2014.

[21] S. Becker, "Improving the Convergence of Back-Propagation Learning with Second-Order Methods" *Proceedings of the 1988 connectionist models summer school*. 1988.

[22] L. Yann, Y. Bengio, and G. Hinton, "Deep Learning." *nature* 521.7553 (2015): 436-444.

[23] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," Springer, Berlin, Heidelberg, 2012, pp. 437–478.

[24] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, 2017, pp. 464–472.

[25] "Breast Cancer Wisconsin (Diagnostic) Data Set | Kaggle." [Online]. Available: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data. [Accessed: 28-Jul-2020].

[26] A. Bhardwaj and A. Tiwari, "Breast cancer diagnosis using Genetically Optimized Neural Network model," *Expert Syst. Appl.*, vol. 42, no. 10, pp. 4611–4620, Jun. 2015.

4.1.2 A Comparative Study of Machine Learning and Deep Learning in Network Anomaly-Based Intrusion Detection Systems (Fall 2020)

Mohab Sameh et al., presented his paper "A Comparative Study of Machine Learning and Deep Learning in Network Anomaly-Based Intrusion Detection Systems", which discusses the application of artificial intelligence in the field of network refinement. [1]

2020 15th International Conference on Computer Engineering and Systems (ICCES) | 978-0-7381-0559-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCES51560.2020.9334553

A Comparative Study of Machine Learning and Deep Learning in Network Anomaly-Based Intrusion Detection Systems

Mohab Sameh Abdel-Wahab
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA), Egypt
mohab.sameh@msa.edu.eg

Ahmed M. Neil
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA), Egypt
aneil@msa.eun.eg

Ayman Atia
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA), Egypt
HCI-LAB, Faculty of Computers
and Artificial Intelligence,
Helwan University, Egypt.
aezzat@msa.eun.eg

***Abstract*—This paper presents a comparative study of Machine learning and Deep learning models used in anomaly-based network intrusion detection systems. The paper has presented an overview of the previous work done in the field of ML and DL IDS, then an overview of the used datasets in reviewed literature was presented. Moreover, ML and DL models were tested on the KDD-99 dataset, and performance results were presented, compared, and discussed. Finally, areas of future research of critical importance are proposed by the authors.**

***Index Terms*—intrusion detection, machine learning, deep learning, comparative**

I. Introduction

In the past decade, dependence on network-attached devices has continuously grown. With such dependence, the importance of securing network-attached devices has grown in importance thus, network security has been an ever-evolving field. Among other interests, research in network security heavily aims to find methods to prevent malicious attacks to be executed on networks. Such attacks can cause disastrous outcomes organizations, industrial entities, governments, and individual privacy. Intrusion Detection Systems (IDS) are systems that aim to detect such malicious attacks, and are often coupled with Intrusion Prevention Systems (IPS) to prevent such attacks to bypass the implemented security measures and affect the inter-connected devices within networks. Therefore, Intrusion Detection Systems subjects of high demand and development interest. Intrusion Detection Systems are mainly divided into two main categories Network-based Intrusion Detection Systems (NIDS) and Host-based Intrusion detection Systems (HIDS), where NIDS are attached directly to the network's infrastructure fabric at critical network gateways, on the other hand HIDS are installed on individual hosts that are connected to the network. Within both IDS types, IDSs can detect malicious attacks using two main detection methods: Signature-based Detection (aka misuse detection) and Anomaly-based Detection [1] [2]. Signature-based detection works by detecting a specific signature which resembles a specific action set, while Anomaly-based detection works by differentiating between what is considered a normal packet or an abnormal one [1] [3]. Therefore, Signature-based detection is very useful in extremely stable environments which are not prone to change in potential attack methods, while Anomaly-based detection is a much more suitable method of detection of novel malicious attacks [1]. Therefore this paper will focus on using Anomaly-based detection as it suits real-world scenarios, where malicious attack signatures, are continuously changing. Securing the network against any potential attacks that might compromise the victim infrastructure is important. However to achieve that, conventional techniques becomes very weak when it comes to sophisticated attacks such as zero day attacks. This has led the research community to work on anomaly network detection. To prevent malicious use or accidental damage to the network's private data, its users, or their devices. Many proposed works have shed the light on using Machine learning techniques in network anomaly detection in order to distinguish between malicious and benign packets and use such techniques in the context of detection of malicious packets in real time streaming networks [4]. Machine learning is a field in which computers are trained to predict the state of newly fed data based on adequate amounts of previously digested data given by humans as a verified fact. Computers can execute such operations using many different ML models, with each model usually having its advantages and can excel in different use-cases [4]. In the context of IDS, this paper reviews and compares many previous literature which use many different ML models aiming to find the most suitable model and methodology that can be used for effective Anomaly-based IDS.

This paper aims to present a comparative study of the reviewed existing literature in the field of machine learning

978-0-7381-0559-8/20/$31.00 ©2020 IEEE

and deep learning anomaly-based intrusion detection systems. Moreover, this paper tests a selected set of machine learning and deep learning models and illustrates a performance comparison between the utilized models.

II. Datasets

This section introduces different types of datasets which were used in other previous works and in this work. However, to the best of our knowledge and our research we found only these datasets which showed some drawbacks during our experiments due to the age of creation. Such drawbacks are further discussed by Divekar et al. [5], Sung et al. [6], and Janarthanan and Zargari [7].

A. KDD-99 & NSL-KDD Datasets

KDD-99 dataset has been released on 1999 and is based on a 7 week period of packet capture exports made by DARPA's IDS evaluation program in 1998. The KDD-99 dataset is composed of nearly 4.9 million records of network packets, some of which are normal packets and others which are malicious attack packets. Each packet is listed based on 41 features and a target which identifies whether the packet is a normal/attack packet. The dataset targets 5 different attack classes, which break down to 22 different attack subclasses [4]. The dataset is available as a complete version containing 4.9 million records or a 10% version as a less resource intensive alternative. The KDD-99 dataset, despite having some inherent problems such as uneven data, outdated attacks, redundant data, and high skewedness as discussed in [5], is still the most widely used dataset for ML IDS training and performance evaluation [8].

The NSL-KDD is an iteration over the KDD-99 dataset. The NSL-KDD dataset utilizes the same feature and target set, however, it aims to remove the inherent problems within the KDD-99 dataset. Some of which are the removal of redundant and duplicate records in order to avoid bias due to frequently occurring records, providing less records while maintaining relevance to improve performance on less resourceful training and testing environments, and selection of records within each difficulty level is based on an inversely-proportional relationship to the percentage of records in the KDD-99 dataset. [2]

B. UNSW-NB15

The UNSW-NB15 dataset has been developed by Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The dataset has been extracted and derived from 100GBs of raw pcap data using the IXIA PerfectStorm tool, and recognizes nine attack types being: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The dataset labels each vector based on 49 features with the target label. The UNSW-NB15 dataset is considered by some to be a proper advocate as an alternative for the popular KDD-99 dataset [5]. However, the UNSW-NB15 has shortcomings; for example, the dataset contains several features that are redundant and also affect the utilized models' accuracy [6] [7]. To add due to the dataset being relatively new [7], not much work has been done utilizing the dataset, therefore using the dataset as a benchmark for the tested models might lead to misleading results.

III. Related work

The related work of ML and DL in anomaly-based IDS will be reviewed and illustrated in the chronological order of ML and DL pipelines as seen in the reviewed literature, which is in the following order: Data collection, Pre-processing, and Classification [9]. Figure 1 illustrates a ML and DL IDS pipeline structure commonly used in reviewed literature discussed in this section.

A. Data Collection

Despite some inherent problems in the KDD-99 dataset, it is considered the default bench marking standard in ML and DL IDS performance evaluation [10]. It has been found that most of the reviewed literature utilizes the KDD-99 dataset in the data collection phase such as in [8] [11] [12] [3]. A closely-popular choice is the NSL-KDD dataset, which has been utilized in [2] [13] [14]. Moreover, other datasets has been used such as the ISCX dataset in [15] and UNSW-NB15 dataset as used in [5] [10]

B. Pre-processing

In the previous reviewed literature, it has been found that the pre-processing pipeline mainly consists of two stages: numericalization of categorical features, and standardization of highly variant data. To elaborate, the KDD-99 dataset has three categorical features: protocol_type, service and flag [4], all of which need to be labeled as numerical representations of such categorical features [2] [8]. In the reviewed literature, it has been observed consistent popularity of usage of One-Hot Encoders and Label Encoders in the numericalization of the categorical features in the used datasets such as in: [16] [3]. However, other methods of encoding has been used such as the proposition of a novel Non-Symmetric Deep Auto-Encoder (NDAE) in [8]. Moreover, in the next stage of pre-processing, standardization of highly variant data aims to scale data into numerically similar range around the mean of the distribution, and scaling by the standard deviation value. Standardization of highly variant features is a common practice among reviewed literature as observed in [5] [17], the authors of this paper believe that the need of standardization originates from the flexible nature of network usage spikes, which reflects onto packet features being highly variant.

C. Classification

This section illustrates the usage of different classification methods in the reviewed related literature.

Mainly, the utilization of ML classifiers and DL classifiers has been observed. Moreover, such observations will inspire this paper to further test a select set of highly-performing classifiers to be further tested and compared on a local testing environment [18].

Previous work has utilized classifiers such as Support Vector Machines (SVM) [19] [20], K-Nearest Neighbors (KNN) [21],

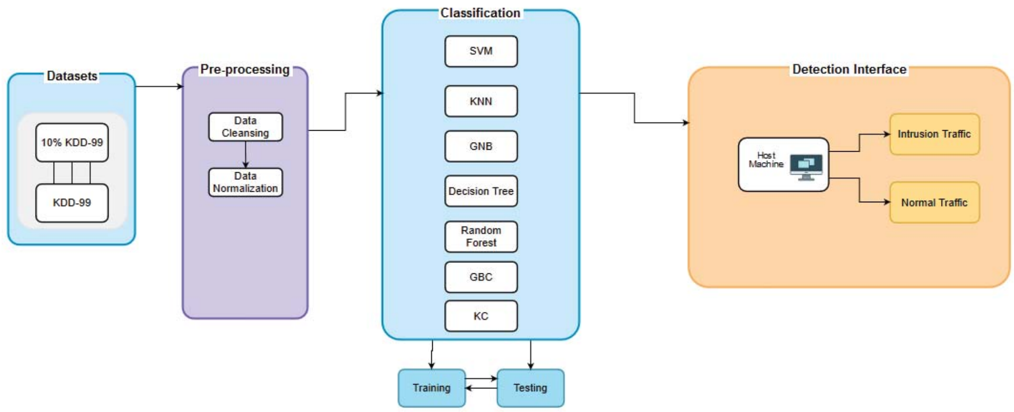


Fig. 1. ML and DL IDS pipeline structure commonly used in literature

Decision Trees (DT) [20] [22], Ensemble Classifiers [23] [4] [24] [5], Naive Bayes (NB) [25] [5] [4], and Deep Neural Networks (DNN) [26] [27] [10].

IV. Experiment

A review of the common promising ML and DL models from the previously reviewed literature will be recreated, analyzed, and discussed in detailed. Moreover, the most suitable select set of models will be indicated.

The experiment was done on a local testing bench with the following specifications: AMD Ryzen 3600 6 Core, 12 Thread CPU, 16GB 3200Mhz CL16 Memory, AMD Radeon RX 570 8GB GPU, Crucial MX300 256GB Sata SSD, Windows 10 Pro (v.1909), Anaconda3(1.9.12) w/ JupyterLab(1.1.4).

A. *Data Collection*

As illustrated by Shone et al. [8], the popularity of the KDD-99 makes it considered by some to be the de facto standard of IDS ML and DL testing datasets. Therefore the authors of this paper decided to test the use of the select ML and DL models on the commonly-used KDD-99 dataset in practice which can reflect the findings relative to a common benchmark.

The KDD-99 dataset is publicly available on University of California Irvine Knowledge Discovery in Databases Archive. A comparison of both the 10% and full corrected versions of the dataset will be denoted, therefore, both dataset versions will be attained in the data collection stage. Both versions of the KDD-99 datasets will be utilized, as it was illustrated in I the larger version of the dataset scales up the frequency of the common classes while keeping unchanged frequencies on the least 2 common classes. Therefore, it is of critical importance to test how various models perform in an environment which resembles high skewness in common class occurrence frequency.

B. *Pre-processing*

1) Data cleansing: Data cleansing is performed on the dataset, as the presence of non-cleansed data can cause undesirable effects on the final models' results. First, the authors of this paper verified the inexistence of null values within both dataset versions, and no nulls were found in both corrected KDD-99 dataset versions. To add, due to the high variance in the dataset features, standardization of highly variant data was performed on both datasets to improve classifier performance during the classification stage.

2) Analysis and Visualization: General analytics of the dataset is performed for visualization and further understanding of the dataset construction.

TABLE I
ATTACK CLASSES AND FREQUENCY OF OCCURRENCE IN 10% AND FULL DATASET

	10% KDD-99 Dataset	Full KDD-99 Dataset
Dos	391458	3883370
Normal	97278	972781
Probe	4107	41102
R2L	1126	1126
U2R	52	52

Table (I) illustrates each attack class and the frequency of its occurrence in both dataset versions is observed. This outcome is of critical importance as the high variation in class frequency can cause high detection difficulty in less common classes.

3) Feature selection: 2 illustrates a correlation heatmap of the dataset's features is display, which shall be used to identify extremely-correlated features which can be dropped. Since the authors have utilized the corrected version of the dataset, no extremely correlated features have been identified. However, the heatmap still is useful for further analysis of the dataset as it identifies features which are highly-uncorrelated and can affect the model's accuracy positively.

Moreover, there were 2 features which had less than 2 occurrences of unique values, which were the 'is_host_login' and 'num_outbound_cmds', therefore such features were considered redundant features and have been dropped to further decrease unnecessary performance overhead on the models.

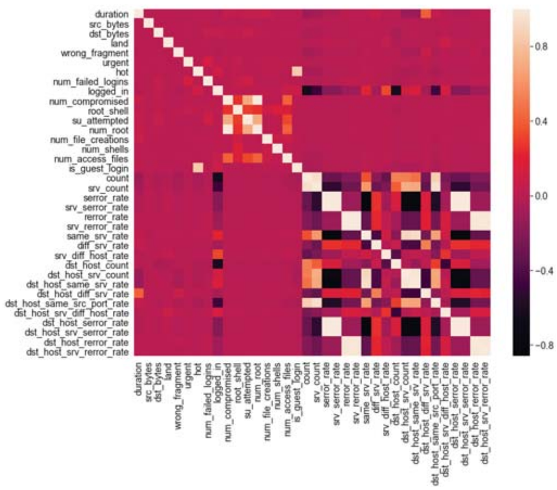


Fig. 2. 10% KDD-99 features correlation heatmap

4) Label encoding: In the label encoding stage, three categorical features 'protocol_type', 'flag', and 'service' were found and scikit-learn Label Encoding was utilized to encode those features' values into numerical representations.

5) Scaling and Splitting Dataset: Scaling has been done using the Min Max Scaler and splitting of the data is done using the scikit-learn python library. The dataset was split into training and testing datasets of a test-to-train ratio of 0.2.

C. Classification

A select choice of models which showed promising results in reviewed literature will be tested. The authors will utilize scikit-learn library to perform the models previewed within subsections (1-7), and the Keras classifier library will be utilized in subsection (8). Moreover, scores and metrics of all models will be illustrated on the basis of precision, recall, f1-score, and support metrics. A detailed descriptions of the metrics is presented below:

Precision:
The average precision (AP) scores are computed as a value between 0 and 1, with 0 referring to a model with 0% accuracy and 1 being a model achieving 100% accuracy. AP is defined as:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where TP is the number of positive items flagged as positive by the model and FP is the number of negative items flagged as positive by the model.

TABLE II
PERFORMANCE RESULTS OF UTILIZED MODELS ON BOTH 10% AND FULL VERSION OF KDD-99 DATASET

		SVM	KNN	GNB	DT	RF	LR	GBC	KC
10% KDD-99	Training Time (seconds)	66	263	0.5	1.1	7.11	12.73	353	300
	Testing Time (seconds)	6.5	226	0.38	0.01	0.39	0.02	1.2	0.5
	Training Score	0.99	0.99	0.88	0.99	0.99	0.99	0.99	0.99
	Testing Score	0.99	0.99	0.88	0.99	0.99	0.99	0.99	0.99
Full KDD-99	Training Time (seconds)	2472	-	5.39	15.8	119	146.9	4019	2900
	Testing Time (seconds)	202	-	3.41	0.64	4	0.21	10.1	4.3
	Training Score	0.99	-	0.88	0.99	0.99	0.99	0.987	0.99
	Testing Score	0.99	-	0.88	0.99	0.99	0.99	0.99	0.99

Recall:
Recall, also known as Sensitivity or True Positive Rate (TPR), calculates the ratio of all correctly detected vectors within the data to all vectors that should be detected in an ideal case.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where TP and FN are True Positives and False Negatives, respectively. True positives being the items that were correctly detected by the model, and false negatives being positive items that were not detected by the model.

F1-score:
F1-score is the harmonic mean of both, precision and recall, and reflects a score of the model's accuracy.

$$F1 = \frac{2 * TP}{2 * TP + FN + FP} \quad (3)$$

where R_n and P_n are the precision and recall at the nth threshold. With random predictions, the AP is the fraction of positive samples.

Support:
Support is the number of occurrences found of each class of items in the dataset.

V. Evaluation and Results

As illustrated in III, the various models were utilized and the performance was measured based on precision, recall, f1-score, and support. Performance was tested on both the 10% and full versions of the KDD-99 datasets.

TABLE III
RESULTS OF ALL TESTED MODELS ON BOTH THE 10% AND FULL VERSION OF KDD-99 DATASET

		10% KDD				Full KDD			
		Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
SVM	dos	1.00	1.00	1.00	78439	1.00	1.00	1.00	776980
	normal	1.00	1.00	1.00	19260	1.00	1.00	1.00	194138
	probe	1.00	0.98	0.99	872	1.00	0.98	0.99	8311
	r2l	0.93	0.91	0.92	223	0.87	0.75	0.81	247
	u2r	1.00	0.18	0.31	11	1.00	0.18	0.31	11
KNN	dos	1.00	1.00	1.00	78439	-	-	-	-
	normal	1.00	1.00	1.00	19260	-	-	-	-
	probe	1.00	0.99	0.99	872	-		-	-
	r2l	0.95	0.96	0.95	223	-	-	-	-
	u2r	0.83	0.45	0.59	11	-	-		-
GNB	dos	0.98	0.94	0.96	129106	0.98	0.95	0.96	1281483
	normal	0.97	0.64	0.77	32167	1.00	0.62	0.76	321021
	probe	0.09	0.99	0.17	1348	0.10	0.97	0.18	13594
	r2l	0.29	0.39	0.33	387	0.03	0.44	0.06	365
	u2r	0.01	0.74	0.01	19	0.00	0.90	0.00	20
DT	dos	1.00	1.00	1.00	129106	1.00	1.00	1.00	1281483
	normal	0.98	0.98	0.98	32167	0.98	1.00	0.99	321021
	probe	0.55	0.90	0.68	1348	0.99	0.64	0.78	13594
	r2l	0.00	0.00	0.00	387	0.00	0.00	0.00	365
	u2r	0.00	0.00	0.00	19	0.00	0.00	0.00	20
RF	dos	1.00	1.00	1.00	129106	1.00	1.00	1.00	1281483
	normal	1.00	1.00	1.00	32167	1.00	1.00	1.00	321021
	probe	1.00	0.99	0.99	1348	1.00	0.99	1.00	13594
	r2l	0.98	0.96	0.97	387	0.96	0.93	0.94	365
	u2r	0.93	0.68	0.79	19	1.00	0.20	0.33	20
LR	dos	1.00	1.00	1.00	129106	1.00	1.00	1.00	1281483
	normal	0.98	1.00	0.99	32167	0.99	1.00	1.00	321021
	probe	0.98	0.90	0.94	1348	0.98	0.90	0.94	13594
	r2l	0.84	0.82	0.83	387	0.14	0.02	0.03	365
	u2r	0.86	0.32	0.46	19	0.50	0.05	0.09	20
GBC	dos	1.00	1.00	1.00	129106	1.00	1.00	1.00	1281483
	normal	1.00	0.95	0.97	32167	1.00	1.00	1.00	321021
	probe	1.00	0.65	0.78	1348	1.00	0.98	0.99	13594
	r2l	0.13	0.77	0.23	387	0.70	0.66	0.68	365
	u2r	0.28	0.79	0.42	19	0.64	0.35	0.45	20
KC	dos	1.00	1.00	1.00	129106	1.00	1.00	1.00	1281483
	normal	1.00	1.00	1.00	32167	1.00	1.00	1.00	321021
	probe	0.99	0.98	0.99	1348	0.97	0.99	0.98	13594
	r2l	0.86	0.89	0.87	387	0.00	0.00	0.00	365
	u2r	0.00	0.00	0.00	19	0.00	0.00	0.00	20

VI. DISCUSSION

SVM and KNN models showed promising accuracy scores in the less common classes 'r2l' and 'u2r', but both models fell behind significantly in terms of training and testing times, which rises concerns whether SVM and KNN can be deployed in real time detection considering high bandwidth and time-critical networks. Moreover, the KNN model was unable to finish training during a 24-hour period, therefore the test was aborted and KNN was considered unsuitable for real-world usage.

Furthermore, LR showed satisfactory accuracy scores, excellent training time, and outstanding training time in the 10% dataset, however, the model fell behind significantly in the training time and accuracy of the full dataset in the less common classes; therefore, it shall further be investigated whether the model can be used in a real time detection on a non-time critical environments or whether the model can be utilized in an offline-learning ML IDS.

The GBC and KC model showed adequate accuracy scores in the common classes; however it fell behind significantly in the less common classes showed unsatisfactory training and testing time in both datasets.

Moreover, GNB and DT were able to achieve excellent training and testing time in both versions of the dataset; however, the shortcomings of both models in an IDS context is clear due to the overall low average accuracy scores achieved in the less common attack classes.

Finally, the RF model showed outstanding accuracy in all 5 classes, while achieving satisfactory training and testing times. Despite falling behind in training time of the larger dataset, RF was able to achieve extremely satisfactory results even in both dataset versions, which was remarkable considering the significant accuracy drop other models suffered from when being tested on the full dataset. Therefore, the RF model's results were considered extremely promising and further testing of the model is highly recommended.

VII. Conclusion and Future Work

This paper presents a comparative study of ML and DL models used in anomaly-based network intrusion detection systems. The paper has presented an overview of the previous work done in the field of ML and DL IDS, then an overview of the used datasets in reviewed literature was presented. Moreover, ML and DL models were tested on the KDD-99 dataset, and performance results were presented and compared. Various models showed specific advantages and disadvantages and no specific model was considered completely superior over other models, however, the RF model showed promising results and shall further be tested in a real-world IDS scenario.

Furthermore, it should be noted that the field of ML and DL IDS is relatively new and further research is extremely needed. Therefore this paper lays the ground for future work to be done especially in the areas of online-learning ML and DL IDS. To add, another recommended field of research is improving current datasets, as the datasets available showed inherent problems such as being dated and unrepresentative of modern network attacks. Finally, another important area of research is targeting mobile-specific network attacks, either by providing specialized mobile IDS network datasets or providing mobile-specific ML and DL IDS architectures.

References

[1] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, p. 227–261, Nov. 2000. [Online]. Available: https://doi.org/10.1145/382912.382914

[2] B. Ingre and A. Yadav, "Performance analysis of nsl-kdd dataset using ann," in *2015 International Conference on Signal Processing and Communication Engineering Systems*, 2015, pp. 92–96.

[3] W. Hu and W. Hu, "Network-based intrusion detection using adaboost algorithm," in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '05. USA: IEEE Computer Society, 2005, p. 712–717. [Online]. Available: https://doi.org/10.1109/WI.2005.107

[4] Y. Hamid, M. Sugumaran, and L. Journaux, "Machine learning techniques for intrusion detection: A comparative analysis," in *Proceedings of the International Conference on Informatics and Analytics*, ser. ICIA-16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2980258.2980378

[5] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: Kdd cup 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018, pp. 1–8.

[6] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *2003 Symposium on Applications and the Internet, 2003. Proceedings.*, 2003, pp. 209–216.

[7] T. Janarthanan and S. Zargari, "Feature selection in unsw-nb15 and kddcup'99 datasets," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1881–1886.

[8] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[9] G. Karatas, O. Demir, and O. Koray Sahingoz, "Deep learning in intrusion detection systems," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 2018, pp. 113–116.

[10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[11] G. Meena and R. R. Choudhary, "A review paper on ids classification using kdd 99 and nsl kdd dataset in weka," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017, pp. 553–558.

[12] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on k-means clustering and naïve bayes classification," in *2011 7th International Conference on Information Technology in Asia*, 2011, pp. 1–6.

[13] L. Hakim, R. Fatma, and Novriandi, "Influence analysis of feature selection to network intrusion detection system performance using nsl-kdd dataset," in *2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, 2019, pp. 217–220.

[14] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *2015 National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 339–344.

[15] S. Soheily-Khah, P. Marteau, and N. Béchet, "Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, 2018, pp. 219–226.

[16] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, 2014, pp. 247–252.

[17] H. Nkiama, S. Zainudeen, and M. Saidu, "A subset feature elimination mechanism for intrusion detection system," *International Journal of Advanced Computer Science and Applications*, vol. 7, 04 2016.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[19] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[20] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 2, 2002, pp. 1702–1707 vol.2.

[21] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary pso and k-nearest neighborhood algorithm," in *2017 11th International Conference on Information Communication Technology and System (ICTS)*, 2017, pp. 181–186.

[22] A. Zainal, M. Maarof, and S. M. Shamsuddin, "Ensemble classifiers for network intrusion detection system," *Journal of Information Assurance and Security*, vol. 4, pp. 217–225, 07 2009.

[23] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Int. Res.*, vol. 11, no. 1, p. 169–198, Jul. 1999.

[24] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, p. 832–844, Aug. 1998. [Online]. Available: https://doi.org/10.1109/34.709601

[25] H. Zhang, "The optimality of naive bayes," vol. 2, 01 2004.

[26] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, p. 85–117, Jan 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003

[27] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 258–263.

4.2 Summer 2021

4.2.1 Can pose classification be used to teach Kickboxing?

Two students Abdelaziz Ashraf and Eriny Wessa have published a research paper at the international conference ICECET hosted in South Africa. Their Paper proposed a system to teach kickboxing to beginners using pose estimation and classification [23].

Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET) 9-10 December 2021, Cape Town-South Africa

Can pose classification be used to teach Kickboxing?

Eriny Wessa
Faculty of Computer Science
October University for
Modern Sciences and Arts
Cairo Egypt
Email: eriny.yousrey@msa.edu.eg

Abdelaziz Ashraf
Faculty of Computer Science
October University for
Modern Sciences and Arts
Cairo Egypt
Email: abdelaziz.ashraf@msa.edu.eg

Ayman Atia
HCI-LAB,Faculty of Computers and
Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for
Modern Sciences and Arts (MSA)
Giza Egypt
Email: aezzat@msa.edu.eg

***Abstract*—Kickboxing is a combat sport, based on kicking ,punching, Knee and elbow strikes and defence moves. Every kickboxing technique needs to be preformed a specific way, As there is correct postures and wrong postures to every technique. In this paper, we offer a system that can facilitate the beginners trainees to learn kickboxing. The system uses a camera to estimate poses and then, classify them into ”correct techniques” and their common mistakes or ”wrong pose” using ANN. Live feedback is offered by the system. Whenever the classifier recognize a wrong pose, a message is shown to indicate how to correct the posture. Our hypothesis is that, when trainees have the ability to see and recognize their wrong posters, they learn faster. We evaluate the progress of the trainees based on the time it takes to complete a simple kickboxing exercise. Two types of experiments were conducted. The first calculated the progress of trainees everyday, the other calculated the progress of trainees through three training sessions in the span on two hours. Our results show that time taken by users to preform the moves decrease with each time they use our system. This paper focuses on 3 kickboxing techniques, which are slipping, jab and front kick.**

***Index Terms*—pose estimation, Neural Networks, Kickboxing, classification**

I. Introduction

Martial arts are a common combat tradition, That helps people learn self-defence and different styles of combat. Over the past fifty years, there has been a steady increase in the number of individuals that learn different types of martial arts. Martial arts have different types that focus on striking punches and Kicking, such as Karate, kickboxing, taekwondo, judo. Kickboxing is a popular martial-art, practised for fitness and self-defence. Despite the sport’s popularity, It can be dangerous to practise without supervision, As injuries are common. Furthermore, Kickboxing techniques need to be performed with a specific pose or a stance, which means that there are often correct poses and wrong poses for every technique.

In order to avoid injuries and guarantee the quality of learning, trainees seek supervision from professional trainers. Unfortunately, martial art classes tend to have a large number of students, and trainers find it difficult to pay attention to every student individually. Trainers must monitor each student and take care if any of them has a wrong posture. This process can be time consuming and error prune. Similarly, trainees might try to learn from online videos, which has no way to prove the quality of learning and might lead to injuries

(a) Jab Punch Correct Pose (b) Jab Punch Wrong Pose

Fig. 1: Jap Punch

(a) Front Kick Correct Pose (b) Front Kick Wrong Pose

Fig. 2: Front kick

Kickboxing has four categories of techniques. Which are Punching, Kicking, defence, Knee strikes and elbow strikes. This paper focuses on the Jab punch, the front kick and the slipping defence techniques. Each one of these techniques has a common mistake. Firstly, the Jap punch must be preformed by the lean hand while the dominant hand is protecting the

2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 978-1-6654-4231-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECET52533.2021.9698656

(a) Slipping Technique Correct Pose

(b) Slipping Technique Wrong Pose

Fig. 3: Slipping Technique

face as in fig 1 (a), many beginners and professionals forget to protect their face which leaves them vulnerable to an attack As shown in fig 1 (b). Secondly, when preforming a front kick it is a common mistake to not keep the hands close to the chin, which is important to protect the neck and head from a counter attack. furthermore, in order to keep balance, the other leg must be slightly bend as in fig 2 (a). Shown in fig 2 (b) are the wrong positions of front kick. lastly, in the slipping technique it is important to keep the hands in a blocking position and also to lower the body as in fig 3 (a). Shown in fig 3 (b) is a user standing to full height, which makes it easy for incoming punches to hit him. Therefore, it is important to stay low while preforming a slip and not to stand still.

The pose classification problem has been researched before, and different approaches have been implemented. One of the most well know ways to capture this type of technique is by sensors or video. [14] used a multi sensor and a camera to get the trajectory of the limbs preforming kickboxing techniques. [8] used a video as input to extract key frames. Another example [4] used sensors to record movement for karate to classify them as correct and wrong moves.

One of the Challenges of real time pose classification is that we need to be fast and reliable to classify these poses in live video. Real time pose classification and providing real time feedback is a time sensitive presses. Therefore, accruing high accuracy while maintaining the fast processing time is a challenge. Another challenge is the pose estimation of users of different heights. The height of the trainee makes a difference when extracting the key points of the pose. Furthermore, The colours of the outfit also is a challenge because darker colours makes it difficult to detect the key points of the trainee. The research question of this paper asks if we can teach kickboxing using pose classification. Therefore we must have a parameter by which we calculate the learning rate. In the scope of this paper, we measure the learning curve by the time it takes a user to perform the kickboxing exercise correctly to see if there any progress.

In this paper we propose a system to detect and estimate the user's pose using PoseNet, Then classify the correct posses into "slip", "Jab" or"front kick". Furthermore, the classification mode recognizes a common mistake for each pose. The classification model is made using ANN neural network and ML5. If the Wrong pose is detected,the system justifies the right position for the user. We used ANN neural network and ML5 to be able to run our system with less processing and less time. Moreover ANN has the ability to work with incomplete knowledge. Furthermore, given our small data-set, the ANN model can easily provide real time feedback. In the end, we used ANN to have a simple model to detect the common mistake for each pose. Finally, the system calculate the time taken by the user to perform each technique correctly and calculate the total time taken to perform all 3 techniques.

Two types of experiments were conducted. These experiments objective is to measure the learning time to preform a simple Kickboxing exercise. Therefore, In this paper we measure the exercise completion time of each participant . The first experiment monitors the progress of trainees throughout three days. The second experiment monitors the progress of trainees throughout three sessions on the same day. The main contribution of this paper is to measure the learning rate of the trainees, by comparing the time taken by them to complete a simple kickboxing exercises. Our hypothesis is that by pointing out constructive real time feedback to the trainee's common mistakes, it will make it easier for them to learn kickboxing.

II. Related work

Various fields could be related to our system. We categorized the previous works into three categories. Firstly, we discuss previous work that focuses on teaching activities with the help of pose classification. Secondly, we discuss previous work that focuses on pose detection. Finally, we discuss previous work that focuses on activity recognition.

A. *teaching and tracking progress*

P.Thiparpakul et al., [15] proposed a system to teach Muay Thai. Their system is an offline program that uses Microsoft KINECT 2.0 and Unity3D. They proposed a Check Enter Similarity (CES) algorithm to compare the student movements to that of the saved trainer 3D model. After conducting an experiment they reached the result that on average the score of students training with a trainer was 74.25% and the score of students that trained using the program was 86.75%. Wennrich et al., [19] used Virtual Reality and motion capture with a Kinect camera to train students in Karate. similarly, Trejo et al. [16] proposed a system for teaching yoga. They used kinect and Adaboost algorithm to detect yoga posses. The system also provide instructions to improve the student's pose. Their system reached a confidence value of 92%. kipp et al., [9] proposed an Artificial Neural Network model that can predict the height of a counter movement jump for an athlete given the resistance training volume load. They recorded 21 athlete's volume load and jump performance for 15 weeks. Their model was able to predict counter movement jump performance with average error of 1.47cm. Wang et al., [17] proposed a system to act as an AI coach system for skiing. Taking a video as input, the system detect subjects, estimate poses and classify "good poses" and "bad poses", highlighting bad posses so

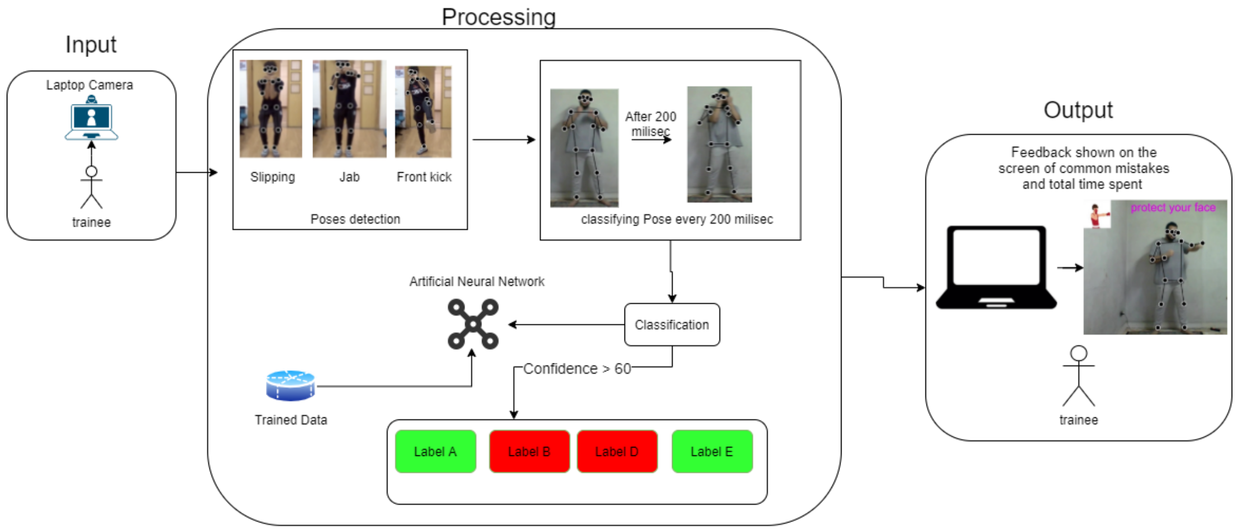


Fig. 4: System Overview

that the student can check them. They developed a human detector to track one person through out the whole video, this tracking system is similar to the R-CNN models used in [5] [13]. Regarding the classification module, the classified common mistakes like bending the hips, crossed snowboard and bending the knees with f1 score of 83.4, 24.8 and 76.4 respectively. Bassel Emad et al., [4] proposed a smart coaching system for karate. They used Kinect's infrared sensor to record player's movements, then classified these movements using Fast Dynamic Time Wrapping, Support Vector Machine, K Nearest Neighbours and Decision Tree. They classified movements included correctly preformed movements and common mistakes. A report is shown to the player at the end to highlight wrong moves. This type of personalized feedback is adapted in our system They reached accuracy of 91.07% on F-DTW, 81.25% on SVM, 73.21% on KNN and 63.39% on DT.

B. *pose detection and classification*

[14] were used multiple Devices (optical motion capture system,cameras and infrared markers) on 42 people. Data are recorded in 3d points. PCA used on trajectory data to reduce features and remove correlations. They used KNN and multi class linear SVC.The Accuracy of limb classification in KNN and SVC (98%,99%) respectively,and technique classification (86% , 86.7%) respectively.

Bezobrazov et al., [2] used PIQ ROBOT with LVQ-nets neural networks to recognize tennis gestures. Their classifier reached accuracy of 90%, although it had poor accuracy in detecting similar gestures.

Zhi-chao et al., [20] proposed a system to detect key poses in wight lifting. They used FCN to exclude background noise, then used CNN for key point classification. The classification model reached accuracy of 95.23%

Holatka et al., [7] manged to classify correct setting techniques in volleyball. They used inertial measurement unit and EMG mussel sensors. They used a LSTM Recurrent neural network model for classification. Their system reached F1 score of 0.74. Similarly, Jian et al., [8] proposed a system for extracting key frames in weightlifting videos. they used Fully Convolutional Networks to extract the frames and CNN for pose estimation. their algorithm ac hived 98.7% accuracy.

Choi et al., [3] proposed a system to classify posses in real time to some labeled activities like sleeping or writing...etc. They used a small camera and refined a ANN and deep learning algorithm to classify the posses. They compared between different numbers of hidden layers and the averaged accuracy of all of them was 82.03%. A similar system was developed by Park et al., [12]. They proposed a system for pose categorization based on an image, they also developed an algorithms that determines the activities using a frequency based method. The goal of both systems was to predict the best thermal environment and using the metabolic rate. they used DNN for pose classification.they reached accuracy of 98.9%

He et al., [6] proposed a system to detect and classify poses from a surveillance video, using a single camera. They divided the problem into two steps, the extraction of silhouettes and pose classification. For extracting silhouettes, they modified the algorithm proposed by [18] and used an approximate median filter [10]. Regarding the pose classification, they had two classes normal pose or hands raised. they used Star Skeletonization to classify posses. they reached accuracy of 95.15%.

C. *activity recognition*

Baumbach et al., [1] used smartphones and smartwatches to recognize sports activities. They compared several machine learning and deep learning algorithms. Some of the machine

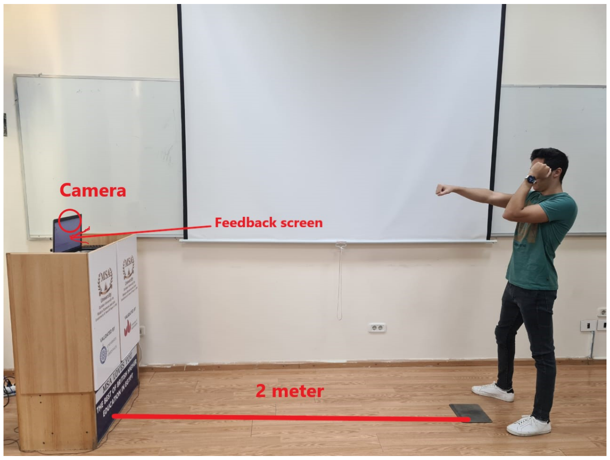


Fig. 5: Second Experiment Environment With A Trainee

learning algorithms used were Naive Bayes with Gaussian Kernel, linear SVM and decision trees. The machine learning algorithms reaches maximum accuracy of 80%. they Proposed a Deep neural network that reached accuracy of 92%.

Moran et al., [11] demonstrated a sensor based system to generate Representative data and to detect various actions like running jumping and landing. They used Discrete Wavelet Transform to extract data from sensors. The extracted data was then fed into a random forest classifier. Using acceleration sensors alone, the classifier reached 98% accuracy.

Throughout this section, we have demonstrated that this problem has been tackled by different approaches in the past. However, we need to highlight that most of the related work needed to use devices like sensors or smartwatches for pose and activity detection [14] [2] [7] [1] [11]. Although, some related work only used one camera like our proposed system [6] [3]. Moreover, our proposed system works in real-time. Meaning that the whole system works using a live camera for the input and shows real-time feedback as the trainee is training. This is an important point because students are more likely to learn quickly if they can see their mistakes as they are training. Also, real-time feedback is not common in related work to this problem, as highlighted before in the works of [15] [19] [9] [17] [4] [8].

III. Methodology

A. Data set

We collected data from 7 different participants that performed the right and wrong pose for each of the three techniques. The data-set was collected from different users with different heights ranging from 155 cm to 186 cm and ages ranging from 21 to 25, 6 participants were males and 1 participant were female. Moreover, While collecting the data set there were different lighting conditions and different setups. Furthermore, each participant in the data collection used their laptop to collect the data, to have different processing environments and different camera qualities. We collected the pose estimation data of the participants using ml5 PoseNet, performing each technique correctly in 15 sec and its common mistake in another 15 sec. Pose estimation data has 34 key points of each participant, which represent the correct poses and the common mistake. Finally, we were able to collect a data set that contains of 34 land markers points which represent the joints, and also the label for the classified pose.

B. Proposed system

We propose a system that estimate poses and recognize three kickboxing tetchiness, the system also classifies ”good techniques” and ”common mistakes.” The Artificial neural network classifies the incoming estimated poses into six categories, three correct posses and one common mistake for each pose. We used PostNet which is based on the MobileNetV1 architecture for poses estimation in both the data collection phase and also the deployed version. We then used the ANN model to develop a system. The ANN model is built-in ml5 neural network helper module that has a 34 inputs and 6 outputs. The ANN model has a 3 layers (input , one hidden layer , output). The 34 inputs represent the key points of the pose which the model takes and classify this poses into 6 outputs, these 6 outputs have 3 correct techniques and 3 common mistakes. The system tracks the time it takes to perform a simple beginner exercise which is slipping then Jap then a front kick. The system calculate time in mil-sec then convert it to seconds. The system records the time spent in one move and the total time spent to correctly perform all three moves.

When the system starts, it asks the user to stand far enough to show their whole body. Then once the whole body is visible, pose estimation using PoseNet starts. The ML5 neural network classifies 5 posses every second. The system prompts the user to perform the Three techniques Which are Slip, Jap and front kick. The trainee performs a technique in which his pose gets recorded, and then compares the Trainee Pose technique to a trained model by ml5 neural network. If the trainee Performed a wrong technique, The model detects it and shows them how to correct it. The system calculates the time spent by the trainee to perform each technique and time spent on performing all three techniques.

IV. Experiment and Results

For the purpose of this study, we have collected a Data set for the ANN classifier. Furthermore , we conducted two types of experiments, with different participants. The Two experiments used PoseNet for pose estimation, and ANN for pose classification. This section will discuss the two experiments to measure the learning progress across days and across same day. Experiment one was done online, meaning that participants were asked to use our system daily from their own homes and record their results. Experiment two was done is a controlled lab environment. Also, due to the pandemic few voluntaries accepted the invitation to join the experiment. However, using a small testing data, we were still able to reach results that worth discussing.

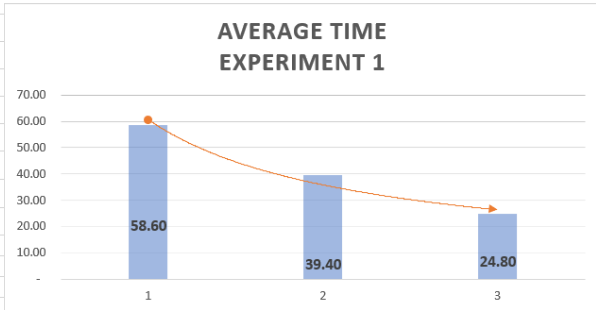


Fig. 6: Average Time Of First Experiment, Where Participants Are Asked To Use Our System Daily

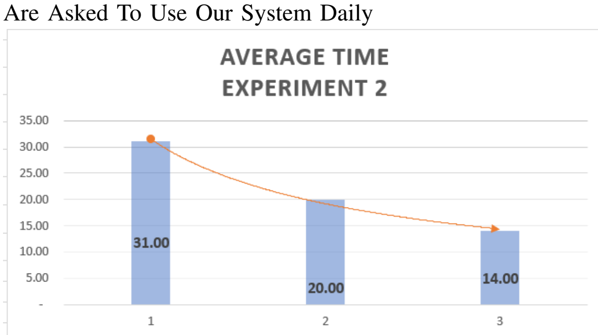


Fig. 7: Average Time Of Second Experiment, Where Participants Are Asked To Use Our System 3 Times In The Same Day

Fig. 8: Average Time Taken By Participants To Complete The Experiments

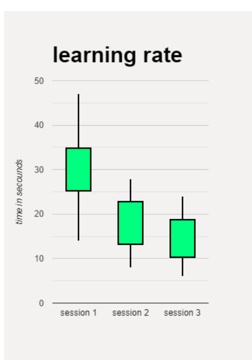


Fig. 9: Same Day Experiment. Time To Preform Exercise Correctly Decreases With Each Use Of Our System.

A. *Experiment one*

This section will discuss the first experiment to measure the learning progress across days.

1) Objective: The first experiment objective was to use our system for three consecutive days to track their progress.

2) Setup: The first experiment setup was 5 users aged from 21 to 26, one female and four male, every participant had different lighting conditions and different processing specs, and they tested our system with their personal computer, and the users were standing two meters away from the camera.

3) Method: The system prompts the users to perform a technique, these techniques are slipping, jab, and front kick. Whenever the user performs a technique correctly the system moves to the next. However, if the system classifies any of the techniques as "wrong poses", a message shows the user how to correct the technique. Some examples of the messages are "Protect your face." and "Get lower". Finally, the system shows the time taken by the user to perform all the techniques,

4) Result: The results from the first experiment shown in fig 6 show that the average time taken by users to preform the exercise correctly decreases each day. The average time of the third day is less than half that of the first Day. Proving that the personalized feedback provided by our system improves the users performance.

B. *Experiment two*

This section will discuss the second experiment to measure the learning progress across same day.

1) Objective: The second experiment objective was to use our system for three consecutive sessions with 30 minutes break in one day to track their progress in performing the three techniques.

2) Setup: The second experiment setup was sixteen users aged from 19 to 24, two female and fourteen male, This experiment was conducted in a controlled environment, with the same lighting conditions and the same processing power, and the same camera quality in figure 5, where the black spot is where the trainee stands and The laptop camera captures the trainee poses.

3) Method: The system prompts the users to perform a technique, these techniques are slipping, jab, and front kick. Whenever the user performs a technique correctly the system moves to the next. However, if the system classifies any of the techniques as "wrong poses", a message shows the user how to correct the technique. Some examples of the messages are "Protect your face." and "Get lower". Finally, the system shows the time taken by the user to perform all the techniques,

4) Result: The result of the second experiment, Which was conducted in a controlled environment, can be seen in fig 7. These results show that improvements in trainees performance was visible even one day of using our system. Furthermore, A box plot can be seen in fig 9 which highlights the progress in the learning rate of the participants. We observed that the top third users that preformed the exercise in the shortest time between 6 and 13 seconds, where between 155 and 175 in height. These users were also aged between 18 and 20 years old.

C. *Accuracy*

After conducting the experiment, we calculated the total predictions of users in experiment two. We have 16 users and each user preform the technique 3 times, so in total we have 144 prediction $16 \times 3 \times 3$. Furthermore, we observed and calculated number of predictions our system declassifies the pose and it was 15, so we calculated the number of successful predictions $144 - 15 = 129$.

$$\frac{No.\,of\;successful\,predictions}{No.\,of\;total\,predictions}$$

$$Accuracy\;=\;\frac{129}{144}\;=\;0.89$$

V. Conclusion and Future Work

In conclusion, pose classification can be used to teach a variety of activities, including kickboxing. Therefore, we proposed a system that teaches beginners how to kickbox. Our system estimates poses using PoseNet, which is based on the MobileNetV1 architecture. Moreover, our system uses ML5 for pose classification. The classifier can detect three kickboxing techniques, which are slip, jap and front kick. Furthermore, the classifier detects a common mistake for every technique, and the system displays a message to the user to help them correct their pose. We conducted two experiments to test this system ability to teach kickboxing. Participants were asked to use our system for three consecutive days in the first experiment. Meanwhile, participants were asked to use our system for three sessions on the same day in a controlled environment in the second experiment. The results of both experiments demonstrate that with each usage of our system, the average exercise completion time for participants decreases.

Because of the Corona pandemic, there were many Limitation, among these obstacles, the collection of data-set for training our model, and find participants to try our system and gather from them their learning process. Our aim in this paper is to detect the common mistakes of beginners and help them learn kickboxing. Future work for this paper includes detecting more advanced Kickboxing techniques and capturing the body moves due to its in continuous motion. Moreover, using Dynamic Time Warping for more complicated kickboxing techniques and body motion will result in more accurate outcomes. Furthermore, This methodology can be implemented in other applications, for example, teaching other forms of martial arts or dances. Some future work might also include an improved user interface, or more specified detection of common mistakes. For example, highlighting the body part responsible for the wrong pose.

References

[1] S. Baumbach, A. Bhatt, S. Ahmed, and A. Dengel, "Towards a digital personal trainer for health clubs-sport exercise recognition using personalized models and deep learning." in *ICAART (2)*, 2018, pp. 438–445.

[2] S. Bezobrazov, A. Sheleh, S. Kislyuk, V. Golovko, A. Sachenko, M. Komar, V. Dorosh, and V. Turchenko, "Artificial intelligence for sport activity recognition," *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2, pp. 628–632, 2019.

[3] E. J. Choi, Y. Yoo, B. R. Park, Y. J. Choi, and J. W. Moon, "Development of occupant pose classification model using deep neural network for personalized thermal conditioning," *Energies*, vol. 13, no. 1, p. 45, 2020.

[4] B. Emad, O. Atef, Y. Shams, A. El-Kerdany, N. Shorim, A. Nabil, and A. Atia, "ikarate: Karate kata guidance system," *Procedia computer science*, vol. 175, pp. 149–156, 2020.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] X. He, Z. Wang, B.-S. Shin, and R. Klette, "Detection of defined human poses for video surveillance," in *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, ser. IVCNZ '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 160â165. [Online]. Available: https://doi.org/10.1145/2683405.2683439

[7] A.-K. Holatka, H. Suwa, and K. Yasumoto, "Volleyball setting technique assessment using a single point sensor," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 567–572.

[8] M. Jian, S. Zhang, L. Wu, S. Zhang, X. Wang, and Y. He, "Deep key frame extraction for sport training," *Neurocomputing*, vol. 328, pp. 147–156, 2019.

[9] K. Kipp, J. Krzyszkowski, and D. Kant-Hull, "Use of machine learning to model volume load effects on changes in jump performance," *International journal of sports physiology and performance*, vol. 15, no. 2, pp. 285–287, 2020.

[10] N. J. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine vision and applications*, vol. 8, no. 3, pp. 187–193, 1995.

[11] K. Moran, A. Ahmadi, C. Richter, E. Mitchell, J. Kavanagh, and N. OâConnor, "Automatic detection, extraction, and analysis of landing during a training session, using a wearable sensor system," *Procedia Engineering*, vol. 112, pp. 184–189, 2015.

[12] B. R. Park, E. J. Choi, Y. J. Choi, and J. W. Moon, "Accuracy analysis of dnn-based pose-categorization model and activity-decision algorithm," *Energies*, vol. 13, no. 4, p. 839, 2020.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[14] K. M. Soekarjo, D. Orth, E. Warmerdam, and J. Van Der Kamp, "Automatic classification of strike techniques using limb trajectory data," in *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer, 2018, pp. 131–141.

[15] P. Thiparpakul and W. Limprasert, "New design system to learning martial art via kinect 2.0," in *Proceedings of the 5th International Conference on Information and Education Technology*, ser. ICIET '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 89â93. [Online]. Available: https://doi.org/10.1145/3029387.3029413

[16] E. W. Trejo and P. Yuan, "Recognition of yoga poses through an interactive system with kinect based on confidence value," in *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2018, pp. 606–611.

[17] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, "Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 374â382. [Online]. Available: https://doi.org/10.1145/3343031.3350910

[18] Z. Wang, B.-S. Shin, and R. Klette, "Accurate silhouette extraction of a person in video data by shadow evaluation," *International Journal of Computer Theory and Engineering*, vol. 6, no. 6, p. 476â483, 2014.

[19] K. Wennrich, B. Tag, and K. Kunze, "Vrte do: The way of the virtual hand," in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3281505.3281578

[20] C. Zhi-chao and L. Zhang, "Key pose recognition toward sports scene using deeply-learned model," *Journal of Visual Communication and Image Representation*, vol. 63, p. 102571, 2019.

4.2.2 Exam Cheating Detection System with Multiple-Human Pose Estimation

Two students of Computer Science Mohamed Amr and Youssef Maged for publishing their scientific paper to the IEEE Xplore as part of the IEEE International Conference on Computing (ICOCO). Mohamed and Youssef presented their paper "Exam Cheating Detection System with Multiple-Human Pose Estimation" which proposes a system that tracks cheating during exams with the presence of multiple students. The system can detect whether or not a student is cheating by continuously validating their head posture and hand movement conditions during the exam. The Malaysia-based conference took place virtually on November 18th, 2021. The presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community. [17]

Exam Cheating Detection System with Multiple-Human Pose Estimation

Mohamed Amr Samir
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Egypt.
mohamed.amr19@msa.edu.eg

Youssef Maged
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Egypt.
youssef.maged4@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University, Egypt.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Egypt.
aezzat@msa.edu.eg

2021 IEEE International Conference on Computing (ICOCO) | 978-1-6654-3689-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOCO53166.2021.9673534

***Abstract*—Cheating in exams is a persistent problem that contributes to academic dishonesty. In this paper we explore a variety of related work proposed as a solution for exam cheating, then we propose an exam cheating detection system that works for both on-site and online examinations. The proposed system applies Human Pose Estimation that includes both single-user and multiple-user tracking algorithms. Based on video footage, the system can detect whether or not a student is cheating by continuously validating their head posture and hand movement conditions during the exam. The system doesn't fully imply a student is cheating, instead, we use the term 'warning' for the output to indicate that the student has met an abnormal condition that is similar to cheating behavior. At last, we validate the system usage in real-life examination environments through two different experiments that resulted in accuracy numbers of 92%-97% in cheating detection.**

***Index Terms*—Exam cheating detection, multiple-human pose estimation, tracking on-site and online cheating**

I. Introduction

Exam cheating is a very common issue in many communities. It is considered the number one cause of inefficiency and unproductiveness that range from primary education to higher education [1]. While there is on-site cheating that occurs in places such as classrooms and lecture halls, with the COVID-19 pandemic, online cheating has increased dramatically because of the switch from offline to online learning at homes. Many students have acknowledged to cheating on online tests as a result of the stress caused by the COVID-19 quarantine and their concern about their GPAs [2].

Teachers and other professionals might have a difficult time spotting cheaters due to a variety of factors. Deceptive students cheating in exams are an example. Such students can manipulate the teacher questioning them in a way in which the teacher is left to believe that the student is not cheating when it is otherwise [12].

Exam cheating is defined as using the assistance of outside material provided in some way to solve exam questions [7]. Some of the issues attributed to tracking exam cheating are the affiliation of different cheating techniques and the need to observe more people in examination rooms. Casually, on-site exams have more people sitting in a room than online exams.

There are a variety of cheating techniques such as writing on arms and hands, but it is not preferable by all students. Most students prefer looking by the head and communication through gestures, signs, and trading papers, because it is easier and sometimes not as detectable as writing on body parts [3]. During pandemics, online exams become popular, but also cheating becomes easier as there is often no examiner watching over. Students can rely on searching for answers through notebooks prepared beforehand and several other techniques [18].

Head posture plays a big part in cheating, most students take it easy and look for answers around them, whether that is the exam paper of another person or a scattered note nearby. Some students might even take it far as trading papers or writing answers for each other.

Human Pose Estimation systems have been rapidly expanding with ever-increasing technology advancements, and have achieved tremendous progress in recent years by incorporating various forms of artificial intelligence [20]. A system applying pose estimation can easily track the key points and how their position changes in human skeletons. Pose estimation is a solid solution for exam cheating detection since body posture and movements play a big part in it [14].

This paper explores a variety of related research intended as solutions for exam cheating. Then we propose a system designed with Human Pose Estimation that can detect some of the cheating techniques. The main focus of this system is particularly on tracking abnormal head posture and hand movement that implies the presence of a possible cheating activity. The system is also designed for use on single or multiple student examinations.

The main goal of this research is to help in tackling a very persistent academic problem that is cheating. To indicate whether or not our proposed system is successful, we tested the system through two different experiments that included different students and environments.

II. Related Work

Nishchal et al. [4] developed a system that detects on-site cheating with a CCTV camera. The proposed system applies Human Pose Estimation (OpenPose) and emotion analysis. It can detect if a student turns back to cheat by calculating hip to head ratios from a side camera, as well as recognize any possible cheating emotions. The system works for only one student per camera observation. Chuang et al. [6] developed a tracking system that analyzes student behavior during online exams to find suspicious movements that relate to cheating. The main tracking technique is analyzing the head pose of students taking the exam using a real-time video camera, the footage is run through a machine learning model that detects abnormal head angel. Bali et al. [15] developed a system that detects body posture with artificial neural networks. The system will first capture a video divided into frames, then it applies object detection and neural network to learn possible suspicious postures.

Arinaldi et al. [5] developed a system for on-site cheating detection. The system follows 4 main steps to indicate cheating behavior; MOG to extract a cropped video chunk of potential abnormal gestures, feature extractions using 3D CNN, and a classifier based on XG boost. The last step is the interaction of one or more gestures and their corresponding text description. Yulistia et al. [8] used video cameras to track single-person on-site exam cheating by implementing background subtraction and pixel changes tracking in session footage. The occurring movements are classified with machine learning to extract cheating behavior. Zhao et al. [10] developed an on-site multiple cheating detection system with a camera. The system applies a moving objects detection algorithm, background subtraction, and artificial bee colony algorithm for optimization. The system is used for paper trading cheating detection.

Fakhroddin et al. [11] developed an image processing method to detect single person cheating with a camera. Images taken are automatically compared to reference images when the exam is over. Tomas et al. [16] developed a system that detects multiple people poses in 2D images. The system applies part affinity fields (PAFs) to learn body parts using 2D vector fields. Asadullah et al. [9] developed a system that detects whispering during exams. The proposed system contains four components: silence removal, energy estimation, time duration, and spectrum estimation. Firuz et al. [17] developed a system that detects cheating in exams using assessments. The system validates the scores of each student to detect abnormal changes. Machine learning techniques are used for abnormal assessment classification.

E-proctor is a technique used to help prevent cheating during online exams by obligating the students to authenticate their identity using a fingerprint scanner to access their exam. It also requires a camera to track the eye movements of the student when looking at the screen while solving the exam. Additionally, it requires the student to install software that locks the computer system on the exam which prevents students from switching between tabs or searching through the internet. [13]

III. Methodology

A. System Overview

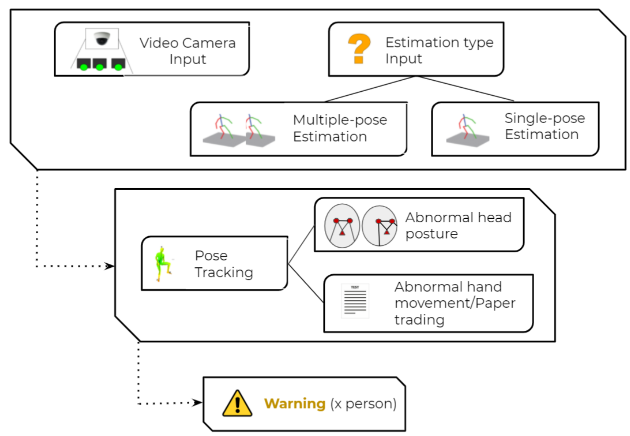


Fig. 1. System overview demonstrates the input and the flow of data between each system component.

The system starts by extracting the input footage and asking for estimation type (single or multiple), proceeding to pose classification where joints are tracked to determine cheating behaviors and display a corresponding output. Below, we demonstrate each section separately.

B. Data collection

The system requires a centered camera setting in front of the students taking the exam. In the below figure, the camera setting is demonstrated for both multiple and single user observation. A camera placed in front of multiple students should typically allow more view distance to cover the entire set of students setting in chairs. The camera must observe students from a straight point of view. Improper camera installation could lead to decreased detection accuracy.

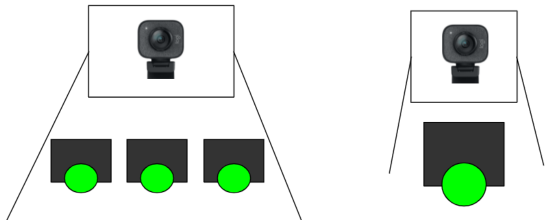


Fig. 2. Demonstrates camera setting in a multiple or single student examination.

C. Human Pose Estimation

The system applies the Tensorflow-PoseNet model on input footage. ”Pose estimation refers to computer vision techniques that detect human figures in images and video. The pose estimation model takes a processed camera image as the input and outputs information about keypoints. The keypoints detected are indexed by a part ID, with a confidence score between 0.0 and 1.0. The confidence score indicates the probability that a keypoint exists in that position.” [19]. In this system, we apply Multiple-pose Estimation. This algorithm is very useful for on-site cheating detection as it makes it possible to track multiple students per one frame, up to twenty students. Tensorflow-PoseNet is a machine learning-based model taught with human images and videos, in order to successfully track down students, the head and wrist of each student must be clearly visible as a requirement for the system to run without flaws.

Fig. 3. Shows the focal key points returned by the Tensorflow-Posenet model.

D. Head posture tracking

The system continuously checks for abnormal head posture by validating the head angel condition. The abnormal condition is met when the face is leaning sideways at a certain degree away from the center. Since we are applying pose estimation, the main focal key points used for validation are the nose, eyes, and ears. This method of tracking applies to both single and multiple student tracking.

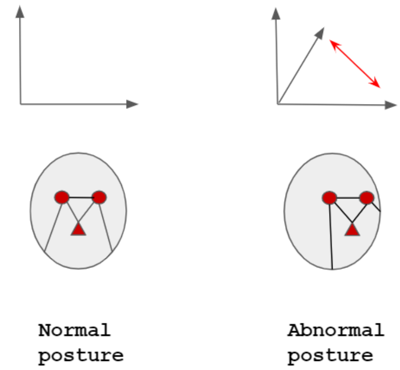


Fig. 4. Demonstrates face angle changing from a normal position into an abnormal position (cheating posture).

The size appearance for these key points scale with distance off-camera. In order to calculate a scaling face posture degree, the system applies an equidistance formula determined by the changing x-axis between both eye key points.

The system applies a default confidence value of 0.15 to generate the output, it is the best possible value to track key points precisely. However, this value can be manually changed within the system interface for adaptability. Considering the possibility of false positives, the system doesn’t fully imply a person is actually cheating, instead, we use the term ’warning’ as an output to avoid this kind of misunderstanding between reality and computer estimation.

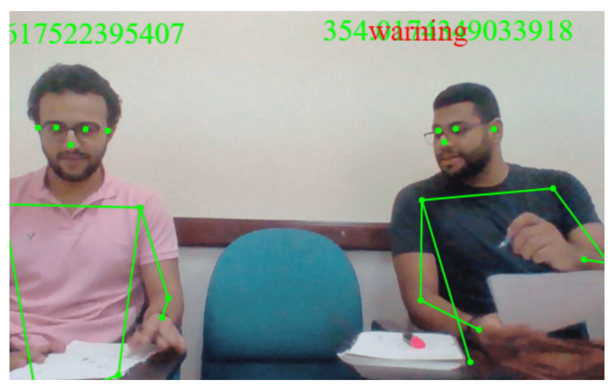


Fig. 5. Demonstrates the detection of an abnormal head angel that results in a warning.

E. Hand movement tracking

Analyzing hand movement can indicate when there is possible paper trading or someone writing in a paper that belongs to another student. The x-axis and y-axis of both wrist key points are tracked to validate the abnormal condition, also the head position is used for calibration. Abnormal hand position is concluded when the x-axis of either wrist key points is reaching far away from the head, however, the y-axis of wrist key points must not be exceeding that of the head, otherwise, the student can be doing something else like raising their hands. We apply the equidistance formula in order to determine the amount of reach required to conclude abnormal hand movement. The same default confidence value of 0.15 is used to generate the output.

Fig. 6. Demonstrates the detection of an abnormal hand movement that results in a warning.

IV. Experiment 1

The proposed system was tested in a real-life setting where cheating variables were manipulated to only address the detectable techniques built within the system algorithm. The focus of this experiment was to validate the system's capability of successfully tracking down cheating attempts with multiple subjects included.

A total of 3 experimental sessions were conducted with 9 participants including 7 males and 2 females within the age of 19 to 22 years old. The participants were distributed across groups of 3. Sessions were conducted in university rooms typically used for examinations, with subjects sitting within 20 cm close to each other in order to allow the possibility of cheating by head movement and trading papers. A video camera remained facing the subjects and recording within a distance of 1.2 meters.

Participants were instructed to make cheating attempts at different times and as many times as possible, noting that these attempts were limited to only head posture and paper trading techniques. The cheating attempts for all sessions have accumulated to 162, described as events. Ground truth visual extraction was done by the testers to accumulate the successful cheating detection warnings displayed by the system for each set of events. In conclusion, the system had an overall average detection accuracy of 94% and 97% for abnormal head posture and paper trading respectively.

The above results table displays cheating events detected as in abnormal head posture or hand movement for different sessions. Concluded accuracy is the weighted average of total events.

Accuracy flaws related to paper trading were mainly due to subjects using an unexpected way to trade papers with each other. However, this specific issue was addressed in a later release of the system. The tight distance between subjects applied in the experiment setup was a negative factor in abnormal head posture detection accuracy. To clarify, it occurred a few times that subjects have had to move their heads only slightly to get answers from very close papers, which went by undetectable for not passing the algorithm conditional standards for calculating abnormal head posture angel.

TABLE I
Experiment 1 - Results

Session	***No. Events***	***Head Posture***	***Paper Trading***
Session 1	36	24/25	10/11
Session 2	76	58/65	11/11
Session 3	50	40/40	10/10
Avg Accuracy %	162	94%	97%

Fig. 7. Sample image taken from the first experiment shows two detected abnormal head postures.

V. Experiment 2

The proposed system was also tested on pairs of students attempting to cheat in different environments. The focus of this particular experiment was to validate the system in different environments and backgrounds. Two sessions were conducted both at home and university examination room to insure that the system can work for online or offline exams. The experiment was organised with the same set of subjects, camera setup, and instructions used in experiment 1. Ground truth visual extraction was done to count successful cheating detection. For both abnormal head and hand movement, the system had an overall average detection accuracy of 95% and 92.3% respectively.

TABLE II
Experiment 2 - Results

Environment	***No. Events***	***Head Posture***	***Paper Trading***
Home	25	19/20	5/5
University Exam Room	28	19/20	7/8
Avg Accuracy %	52	95%	92.3%

The above table displays cheating events detected as in abnormal head posture or hand movement for two different environments. Concluded accuracy is the weighted average of total events.

From this experiment, we concluded that the system can run in different environments without flaws. The failed detection had the same causing factor reported in experiment 1.

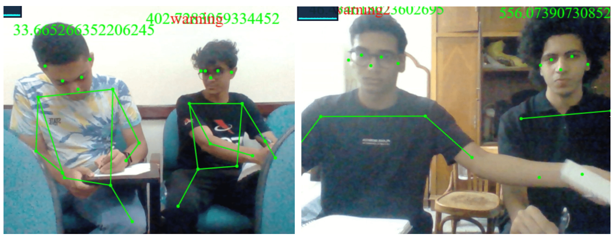


Fig. 8. Different environments used for the second experiment.

VI. Conclusion

This paper introduces some of the common cheating techniques and the role of Human Pose Estimation in cheating detection. The proposed system in this research focuses mainly on detecting abnormal head posture and hand movement cheating techniques. In the related work section, we explore a set of solutions proposed to tackle cheating, then we proceed to display our methodology in detail. Methods include data collection with video cameras, pose estimation algorithms, abnormal head posture and hand movement tracking, and output warnings. At last, the validity of the proposed system is tested through two different experiments.

VII. Future Work

Our goal for the system implementation is to increase the number of students the program is able to track per camera. Several other anti-exam cheating techniques such as whispering recognition and eye movement tracking can also be of great addition to our system.

References

[1] Ç. Semerci & Z. Saglam, “Attitudes and opinions of the police candidates regarding cheating; Elazig province example” Firat University Journal o f Social Sciences, 15 (2), 163-177, 2005.

[2] Abdelrahim, & Yousif. (2021). The Effects of COVID-19 Quarantine on Online Exam Cheating: A Test of COVID-19 Theoretical Framework. Journal of Southwest Jiaotong University. 56. 1-9. 10.35741/issn.0258-2724.56.2.1.

[3] D. A. Odongo, E. Agyemang, & J. B. Forkuor, ”Innovative Approaches to Cheating: An Exploration of Examination Cheating Techniques among Tertiary Students”, Education Research International, vol. 2021, Article ID 6639429, 7 pages, 2021. https://doi.org/10.1155/2021/6639429

[4] J. Nishchal, S. Reddy, & P. N. Navya, ”Automated Cheating Detection in Exams using Posture and Emotion Analysis,” 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020, pp. 1-6, doi: 10.1109/CONECCT50063.2020.9198691.

[5] A. Arinaldi & M. I. Fanany, ”Cheating video description based on sequences of gestures,” 2017 5th International Conference on Information and Communication Technology (ICoIC7), 2017, pp. 1-6, doi: 10.1109/ICoICT.2017.8074679.

[6] C. Chia., C. Scotty & F. John. (2017). Detecting probable cheating during online assessments based on time delay and head pose. Higher Education Research & Development. 1-15. 10.1080/07294360.2017.1303456.

[7] Stuart, Jillian & Barnes, Jeffrey & Deaton, Anna & Fulks, Kristopher and Ryan, Kristina & Rettinger, David. (2010). Imitation Is the Sincerest Form of Cheating: The Influence of Direct Knowledge and Attitudes on Academic Dishonesty. ETHICS & BEHAVIOR. 20. 47-64. 10.1080/10508420903482616.

[8] Y. K. Aini, A. Saleh & H. A. Darwito, ”Implementation of Background Subtraction Method to Detect Movement of Examination Participants,” 2020 International Electronics Symposium (IES), 2020, pp. 494-500, doi: 10.1109/IES50839.2020.9231922.

[9] M. Asadullah & S. Nisar, ”An automated technique for cheating detection,” 2016 Sixth International Conference on Innovative Computing Technology (INTECH), 2016, pp. 251-255, doi: 10.1109/INTECH.2016.7845069.

[10] Y. Lin & J. Zhou, ”Detection method for cheating behavior in examination room based on artificial bee colony algorithm,” 2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), 2015, pp. 122-125, doi: 10.1109/ICCSS.2015.7281161.

[11] Jalali, Khatereh & Noorbehbahani, Fakhroddin. (2017). An Automatic Method for Cheating Detection in Online Exams by Processing the Student‘s Webcam Images. doi: 10.1109/ICCSS.2015.7281161

[12] Marksteiner, Tamara & Reinhard, Marc-Andre & Dickhäuser, Oliver & Sporer, Siegfried. (2012). How do teachers perceive cheating students? Beliefs about cues to deception and detection accuracy in the educational field. European Journal of Psychology of Education - EUR J PSYCHOL EDUC. 27. 10.1007/s10212-011-0074-5.

[13] Bawarith, Razan & Abdullah, Dr & Anas. (2017). E-exam Cheating Detection System. International Journal of Advanced Computer Science and Applications. 8. 10.14569/IJACSA.2017.080425.

[14] Suryadevara N. (2021) Human Pose Estimation in the Browser. In: Beginning Machine Learning in the Browser. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6843-83

[15] Kale, Hrishikesh & Mandke, Prathamesh Mahajan, Hrishikesh & Deshpande, Vedant. (2018). Human Posture Recognition using Artificial Neural Networks. 272-278. 10.1109/IADCC.2018.8692143.

[16] Cao, Zhe & Simon, Tomas Wei, Shih-En & Sheikh, Yaser. (2016). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. doi: 10.1109/CVPR.2017.143

[17] Kamalov, Firuz et al. “Machine learning based approach to exam cheating detection.” PloS one vol. 16,8 e0254340. 4 Aug. 2021, doi:10.1371/journal.pone.0254340

[18] Adzima, & Kerry. (2021). Examining Online Cheating in Higher Education Using Traditional Classroom Cheating as a Guide. 10.34190/JEL.18.6.002

[19] Tensor-flow PoseNet. “Skeleton Keypoints” Tensorflow Blog , May 07, 2018, blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html.

[20] Khan, N. U., & Wan, W. (2018). A Review of Human Pose Estimation from Single Image. 2018 International Conference on Audio, Language and Image Processing (ICALIP). doi:10.1109/icalip.2018.8455796

4.2.3 Yoga Trainer for Beginners Via Machine Learning

Two students of Computer science Omar Tarek and Omar Magdy for publishing their scientific paper to the IEEE Xplore as part of the international Japan-Africa Conference on Electronics, Communication, and Computations (JAC-ECC) . Omar Tarek and Omar Magdy presented their paper "Yoga Trainer for Beginners Via Machine Learning " which propose a Yoga practicing system which can identify human body movement and detect difference yoga postures while simultaneously offering real-time feedback for the practitioner. In addition to measuring the learning rate for individuals after using the system. The conference took place virtually on December 13th, 2021. The presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community . [21]

2021 9th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC) | 978-1-6654-8292-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/JAC-ECC54461.2021.9691425

Yoga Trainer for Beginners Via Machine Learning

Omar Tarek
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
omar.tarek22@msa.edu.eg

Omar Magdy
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
omar.magdy10@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers
and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—Yoga is the practice for both mind and body as it has been proven scientifically on various occasions. Due to the modern advancements of technology, remote Yoga practice sessions have been increasing in popularity following the increase of demand for professional Yoga instructors. In order to tackle this problem, we proposed a system that uses machine learning techniques utilizing an ANN (Artificial Neural Network) model and a human pose tracking model to classify Yoga Hatha movements and detect Incorrect Yoga poses while providing real-time constructive feedback for practitioners to get them to maintain the correct posture for a specific Yoga Hatha pose. This system aims to enhance the learning experience and reduce the practice time for beginners while still retaining a versatile environment. In this paper, we managed to achieve a testing accuracy of 82.2% for our proposed model and were successfully able to reduce the average practice time by an average of 6.4 seconds when tested on 20 participants of different body features.**

***Index Terms*—Machine Learning, Human Activity Recognition, Pose Identification, Sports analysis**

I. Introduction

Yoga is an ancient spiritual practice based on the mental and physical discipline of focusing on the connection of mind and body to boost spiritual strength. Yoga first originated in ancient India over 5000 years ago but recently it grew ever so popular in the last decade. According to the World Health Organization yoga is a valuable tool to increase physical activity and decrease non-communicable diseases. A common predicament for yoga beginners is that it can be difficult to practice correctly without the supervision of a professional instructor. However, with the increase in popularity of online fitness programs, remote yoga practice has been more accessible with people now having access to practice yoga wherever they want. Due to the innovation of modern technologies like virtual tutoring applications, there are now countless ways to help us develop more intelligent and efficient tools for computer-assisted yoga practice.

Human activity recognition has many obstacles concerning computer vision which makes it a challenge to be able to recognize the human skeleton through various and complex human postures especially when some human activity involves being on the ground or in a position where part of the human body gets blocked by another. With that in mind, we present a machine learning model that focuses on classifying Yoga Hatha poses from video streams by detecting the human skeleton through a pose tracking model.

The proposed system works by taking an input of a real-time video stream from a webcam captured at 30fps. An individual then stands in frame with an uninterrupted view and enough room to do basic movement while making sure to stay inside the camera frame. Afterwards, the system proceeds to continuously capture multiple key-frames of the individual while tracking the skeletal joints of the human body in means of X, Y coordinates of each joint. After collecting the dataset required, the data is classified and then trained using Neural network machine learning. Hence, a generated model determines whether the individual practicing the yoga pose is correct or incorrect.

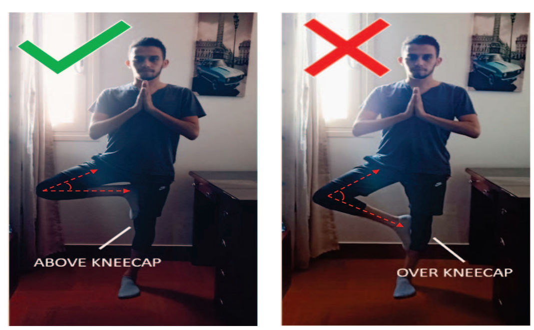


Fig. 1. Yoga Tree Pose

Figure 1 shows two postures for the same yoga pose that indicates both the correct and incorrect techniques used in yoga practice. This genuinely shows how real-time feedback can help the user improve their practice technique efficiently at yoga by avoiding previous mistakes and focusing on maintaining the correct posture in each attempt.

The main contribution for this paper is to create a Yoga practicing system that uses an ANN (Artificial Neural Network) model which can identify human body movement and detect different Yoga postures while simultaneously offering real-time feedback for the practitioner. In addition to measuring the learning rate for individuals after using our system.

II. Related Work

Due to the popularity of online tutoring applications and with the rapid advancement of computer technology, lots of research efforts were dedicated to this subject which concerns two hot topics (A) active pose recognition and identification, (B) human pose training and classification.

A. Active pose recognition and identification

In human pose detection and identification, Yadav et al. [1] has proposed a yoga recognition method that can accurately identify real-time yoga poses using a combination of a hybrid deep learning model that uses CNN (Convolution Neural Network) to extract features and key points and LSTM (Long Short-Term Memory) to give temporal predictions. For Monitoring human body movements and different yoga poses, Islam et al. [2] was able to use Microsoft Kinect Sensor to detect the joints of the human skeleton in real-time and compare them with ones captured with various angles and different yoga poses.

Additionally, Byeon et al. has proposed a human posture recognition using two deep learning methodologies CNN (Convolution Neural Network) and SAE (Stacked Auto Encoder) [3]. Thar et al. [4] has also proposed a yoga training system using Y-system to analyze the postures of a practitioner which includes dominant axes, skeleton-based feature points, and contour-based feature points. Iqbal et al. [5] has presented a pose tracking system that works by first detecting the multiple people in an image. Then, it is followed by a human pose estimation technique that targets each person individually. However, such approaches are only applicable if people in the image appear well separated and do not occlude each other.

Sajjad et al. [6] conducted a study about the learning of human pose recognition regarding the recent progress in computer vision technologies throughout the years and how it changed the way to what they are now with the help of machine learning algorithms and classification techniques by finding gaps in existing work and to give a direction on future work. Jessika et al. [7] conducted a study concerning part affinity fields for human pose estimation with regards to a deep neural network, the paper tackles this topic by investigating the implementation of human body pose estimation from images and videos under various body positions and poses. the human pose detection can be done when automatically recording key points annotations in the affinity map cluster which is done on several images and videos that contains single and multiple human posters.

Belagiannis et al. [8] presented a recurrent human pose estimation that utilizes a ConvNet model that can predict 2D human body poses in an image then regresses a heat map representation for keypoint, this is available through designing an architecture that combines a feed-forward module with a recurrent module where it is possible to run the recurrent model iteratively to improve performance. Moreover, the model can be trained end-to-end with the incorporation of auxiliary losses, investigations also take place to be able to predict key points visibility. Golda et al. [9] proposed a human pose estimation for real-word crowded scenarios is a common problem that is highlighted in this paper, the problem is addressed by taking into consideration multiple approaches which highlights the explicit detection of occluded body parts and also a data augmentation method used on the synthetic generated dataset JTA(3). Su et al. [10] proposed an estimation system of human postures with both physical and physiological constraints that makes full use of 3D information and medical information for improving body pose estimation.

B. Human pose training and classification

Regarding human pose identification, Agrawal et al. [11] has presented a similar system that uses TensorFlow-pose estimation algorithm for identification of yoga poses using a dataset of 5500 images where 80% of the dataset has been used for training various machine learning models and the other 20% were used for testing. Another classification of Yoga Asana poses proposed by Nagalakshmi et al. [12], which uses a 3D view of human poses by learning 3d landmark points from a single image then an encoder architecture is applied followed by a regression layer to estimate pose parameters so that it can be mapped to the 3D landmark points using SMPL (Skinned Multi-Person Linear) model.

Liaqat et al. [13] proposed a hybrid posture detection framework using deep neural networks and multiple machine-learning classifiers. Chen et al. [14] presented a computer-assisted yoga training system that works by instructing practitioners to perform yoga poses correctly by integrating computer-aided vision techniques. Another utilization of Microsoft Kinect Sensor by Jin et al. [15] for a virtual yoga personal trainer by capturing the user's actions then comparing it to standard actions to generate a fitness score representing an evaluation for users performance.

III. Proposed System

We present a method that tracks and classifies Yoga Hatha poses performed and captured from a real-time video stream then it provides positive/negative feedback based on how accurate the user gets compared to the correct pose. The proposed system uses a Simple ANN (Artificial Neural Network) model of the ML5 library from TensorFlow for collecting key points of the skeletal joints using PoseNet to train multiple yoga poses emphasizing the potential posing mistakes for each class.

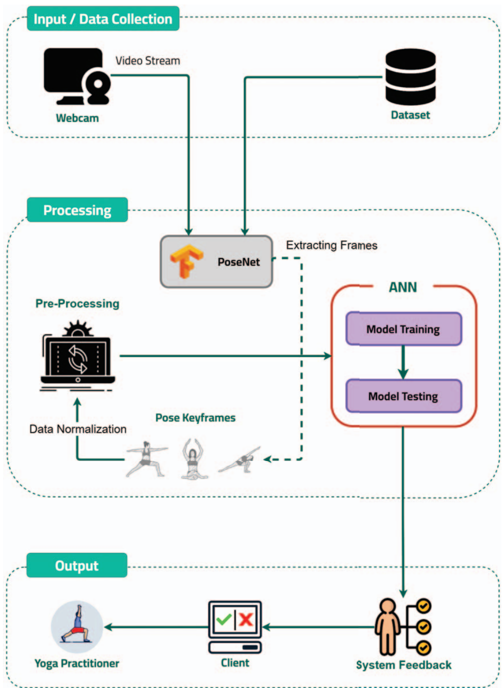


Fig. 2. System Overview

The first step is to identify and extract the yoga poses performed in front of a real-time video stream captured from a webcam of resolution [1280 x 720] at 30fps. Then, by using the PoseNet library, each pose is identified and tracked using the skeletal joints of the human body. On each frame, the output of the pre-trained model of the PoseNet library which contains X-coordinates and Y-coordinates of each joint is then captured and then fed into the ML5 Neural Network after being labeled to match a corresponding pose.

The data generated from our pose tracking model is gathered and stored in a list of multi-dimensional arrays where each array contains 17 elements representing the number of joints of the human body, each element (joint) contains another array that stores only 2 points which are the X-Coordinates and Y-Coordinates of a joint for a total of 34 input point per each frame captured.

In the second step, after collecting important key points from the pre-trained model, we used a Simple ANN classification model that consisted of 3 layers (1 Input Layer, 1 Hidden Layer, and 1 Output Layer) provided from the ML5 API which allows us to configure the neural network by controlling certain attribute (inputs, outputs, layers, task, epochs, etc.) to accommodate for the data of our pre-trained model to classify each yoga pose separately. The ANN classification model consisted of 34 inputs (keypoints collected from the pre-trained model) and 9 outputs (1 correct output and 2 incorrect output for each yoga pose for a total of 3 yoga poses).

IV. Experiment

We performed a total of two experiments involving 24 participants of different body types in order to test the system training capabilities.

A. Experiment 1

The purpose of this experiment is to test the system performance and accuracy when tested on participants with different body types regarding their height, weight, age, and gender. The experiment was conducted inside a well-lit room using a webcam to stream real-time video input into the system. The experiment consisted of 5 sessions and in each session, the participant was given 3 Yoga Hatha poses to practice. Afterwards, participants were asked after each session to state whether they were able to practice the poses successfully or not. After collecting each participant's results throughout the experiment, we measured each participant's progress of all the sessions with a rating from 1 to 5 for each pose (total of 3 poses). Then, we were able to calculate the total accuracy for the system and the average ratio of success between each pose.

B. Experiment 2

This experiment was conducted to test the teaching functionality of the system and its learning effect on different participants.

We gathered the participants in a controlled environment by setting up a webcam at a testing point which is placed at a constant distance from the webcam to ensure consistency of our results. The participants are then asked to stand at the marked position to try and imitate a yoga pose that appears on the screen. Afterwards, they are required to hold their stance in the correct position for a period of 3 seconds counting down from 3 to 1 on the screen in order for the system to progress through to the next yoga pose. If the participant failed to do the correct stance the countdown timer will pause and the system will let them know simultaneously to get back into the correct position. In addition to repeating the same steps for 2 more yoga poses for a total of 3 yoga poses which are done one after the other. Each participant is asked to repeat the same session 2 more times with a 30-minute gap between each session. For each session, we were able to measure the time each participant took to complete the whole session for a total of 3 readings per participant.

V. Results and Discussion

The objective of the experiments conducted was to help us reach 2 main goals which are (A) determining the system accuracy and (B) determining the learning rate, bearing in mind that goal (A) was fulfilled by the first experiment and goal (B) was fulfilled by the second experiment.

A. determining the system accuracy

We were able to calculate the system's accuracy by measuring the number of successful attempts and comparing it to the number of total attempts that were recorded by each participant to achieve an accuracy of 82.2%.

B. determining the learning rate

Regarding the second experiment, we found a substantial decrease in the practice time for the performed Yoga Hatha poses after calculating the average time taken by all participants to complete each session. Upon comparing the results of each session, we were able to measure the learning rate and successfully found an increase by an average of 6.4 seconds by the 3rd session as shown in figure 3.

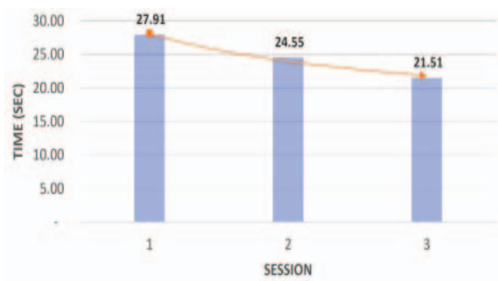


Fig. 3. Average Session Time

To ensure that the results stay uniform, we took into consideration how external factors (height, weight, clothes) might affect the classification process and noticed that participants having a height measurement between 165cm and 175cm were found to have a higher score than the rest of the participants which possibly sets the most desirable height that works best with our ANN model. We also find that clothes and weight do not affect the identification process since the classification system uses the PoseNet library for the identification of the human body movements which depends only on the skeletal view of the body.

No.	Classes	Methods	Algorithms	Participants	Accuracy	Controlled Env.	Related Ref.
1	Bhujangasana, Padmasana, Shavasana	Pose Classification	CNN, LSTM, OpenPose	12	98.92%	✓	[1]
2	Vrikshasana, Virabhadrasana II	Pose Classification	CNN, OpenPose	3	-	✓	[4]
3	Vrikshasana, Padmasana, Tadasana	Image Classification	Random Forest, SVM, KNN, Native Bayes	3	99.04%	×	[11]
Our system	Vrikshasana, Virabhadrasana I, Virabhadrasana II	Pose Classification	ANN, PoseNet	20	82.02%	✓	-

Fig. 4. Similar Systems

VI. Conclusion and Future Work

In this paper, we have proposed a Yoga training system for beginners which helps them practice Yoga Hath poses correctly and efficiently. Our system uses a machine learning model that utilizes an ANN model and a human pose tracking model. We were only able to achieve an accuracy of 82.2% by our ANN model. However, we are confident that the system accuracy could be improved by providing more test cases in the future in addition to calculating the precision, recall, and F1-score. On the bright side, we have successfully managed to provide a better learning experience and were able to reduce the practice time for Yoga beginners where they can practice Yoga more efficiently.

References

[1] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, "Real-time yoga recognition using deep learning," *Neural Computing and Applications*, vol. 31, no. 12, pp. 9349–9361, Dec 2019. [Online]. Available: https://doi.org/10.1007/s00521-019-04232-7

[2] M. U. Islam, H. Mahmud, F. B. Ashraf, I. Hossain, and M. K. Hasan, "Yoga posture recognition by detecting human joint points in real time using microsoft kinect," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 668–673.

[3] A. Mohanty, A. Ahmed, T. Goswami, A. Das, P. Vaishnavi, and R. R. Sahay, "Robust pose recognition using deep learning," in *Proceedings of International Conference on Computer Vision and Image Processing*, B. Raman, S. Kumar, P. P. Roy, and D. Sen, Eds. Singapore: Springer Singapore, 2017, pp. 93–105.

[4] M. C. Thar, K. Z. N. Winn, and N. Funabiki, "A proposal of yoga pose assessment method using pose detection for self-learning," in *2019 International Conference on Advanced Information Technologies (ICAIT)*, 2019, pp. 137–142.

[5] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4654–4663.

[6] F. Sajjad, A. F. Ahmed, and M. A. Ahmed, "A study on the learning based human pose recognition," in *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, 2017, pp. 1–8.

[7] Jessika, A. Handayani, I. Amanda, and H. M. Auliya, "A study on part affinity fields implementation for human pose estimation with deep neural network," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, 2019, pp. 391–396.

[8] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, 2017, pp. 468–475.

[9] T. Golda, T. Kalb, A. Schumann, and J. Beyerer, "Human pose estimation for real-world crowded scenarios," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–8.

[10] L. Su, J. She, and C. Xu, "Estimating human pose with both physical and physiological constraints," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 693–699.

[11] Y. Agrawal, Y. Shah, and A. Sharma, "Implementation of machine learning technique for identification of yoga poses," in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, 2020, pp. 40–43.

[12] C. Nagalakshmi and S. Mukherjee, *Classification of Yoga Asanas from a Single Image by Learning the 3D View of Human Poses*. Cham: Springer International Publishing, 2021, pp. 37–49.

[13] S. Liaqat, K. Dashtipour, K. Arshad, K. Assaleh, and N. Ramzan, "A hybrid posture detection framework: Integrating machine learning and deep neural networks," *IEEE Sensors Journal*, vol. 21, pp. 9515–9522, 2021.

[14] H.-T. Chen, Y.-Z. He, and C.-C. Hsu, "Computer-assisted yoga training system," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23 969–23 991, Sep 2018. [Online]. Available: https://doi.org/10.1007/s11042-018-5721-2

[15] X. Jin, Y. Yao, Q. Jiang, X. Huang, J. Zhang, X. Zhang, and K. Zhang, "Virtual personal trainer via the kinect sensor," in *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, 2015, pp. 460–463.

4.3 FALL 2021

4.3.1 Comparitive study for Shrimp Disease

Abdelaziz senior Student from the Faculty of computer science Abdelaziz Ashraf whose paper got published to the IEEE Xplore as part of the International Conference on Computer Engineering and Systems Abdelaziz Ashraf presented his paper "Comparative study between transfer learning models to detect shrimp diseases" which discussed using machine learning transfer learning models to detect shrimp diseases and whatever is it diseased or not. The Conference took place virtually on 15/12/2021, and the presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community. [3]

Comparative Study Between Transfer Learning Models to Detect Shrimp Diseases

AbdelAziz Ashraf
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
abdelaziz.ashraf@msa.edu.eg

Ayman Atia
HCI-LAB,Faculty of Computers and
Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for
Modern Sciences and Arts (MSA)
Email: aezzat@msa.edu.eg

2021 16th International Conference on Computer Engineering and Systems (ICCES) | 978-1-6654-0867-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCES54031.2021.9686116

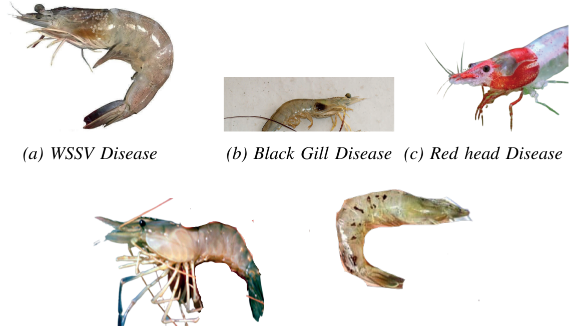
***Abstract*—Shrimps are one of the most important animals in aquaculture. Over the past fifty years, there has been a steady increase in shrimp production worldwide. Shrimp production reached 5.5 tonnes in 2021, and that many countries tend to increase their CAGR and production. Some major problems and challenges persist in shrimp production, such as feed quality and availability, production cost, seed quality, and diseases. There are types of diseases such as black gill and white spot disease. Any delay in the detection of the diseases can lead to the loss of shrimp and infection of other shrimp. In this paper, the authors used transfer learning models to detect two types of shrimp disease (white spot disease and black gill) and to detect diseased shrimp from normal shrimp. the authors aim to know the best transfer learning model that has the highest accuracy in the early detection of shrimp disease. Using five types of transfer learning, the model with the highest validation accuracy is MobileNetV1, with 95% in experiment one and 92.5% in experiment two.**

***Index Terms*—Shrimp diseases, Image processing, Transfer Learning, Machine learning**

I. Introduction

Shrimps are one of the important animals in the aquaculture species, which is characterised as the most popular seafood. Over the past fifty years there has been a steady increase in Shrimps production worldwide, According to FAO and GOAL surveys, the shrimp production Compound Annual Growth Rate (CAGR) resulted in 2.2% from 2012 to 2017, and increased to 5.4% from 2017 to 2021 worldwide. The shrimp's production reached 5.5 tonnes in 2021, which tells us the importance of shrimp, and that many countries tend to increase their CAGR and production. Regardless of the progress in this area, some major problems and challenges persist, such as feed quality and availability, production cost, seed quality, and diseases.

Diseases are considered the most challenging issue that faces shrimp owners and farmers. Since shrimp farming became a significant commercial entity, the disease has had a significant impact on the industry and shrimp owners. To reduce disease risks, shrimp farmers buy quality products from reputable hatcheries and prepare and disinfect ponds regularly. But farmers and shrimp owners make these risks based on their knowledge and without any medicine-based knowledge. Disease types include viruses, bacteria. The most famous types of this virus are black gill, black spot, white spot disease(WSSD), Infectious Myonecrosis Virus(IMV), and yellow head virus(YHV) and other disease such as in figure 1. The loss of the entire shrimp crop can be severe if disease detection and treatment are delayed. This disease makes farms vulnerable and makes them lose a lot of shrimp.



(a) WSSV Disease *(b) Black Gill Disease* *(c) Red head Disease*

(d) Nutritional deficiency Disease *(e) Black spots Disease*

Fig. 1: Shrimp Diseases

WSSV is considered a deadly virus that spoils and infects other shrimps if it has been detected early. The WSSV virus description is that white spots appear on the shrimp shell or discoloration of the shrimp body. YHV is also a deadly disease that kills shrimp in a matter of days, When the shrimp head turns yellow, that is an indication of YHV. Black spots and gill are black spots or lines on the body of the shrimp or the shell. Initial and speedy detection of this virus is an effective way to enhance shrimp farming and global shrimp health.

Previous studies in this area of research have reported using image processing and neural networks to identify and classify diseases in shrimp. Morimoto et al showed a system [13] using image processing techniques to track down the shrimp's movements and to find the moving distance feature to classify infected shrimp and non-infected shrimp. Infected shrimp tend to not eat at the time of feeding. They converted the video data

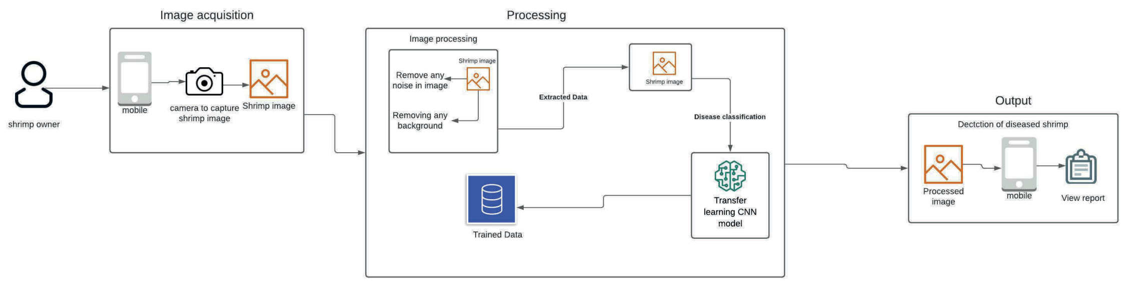


Fig. 2: System Overview

of shrimps into a sequence of frames.

Liu et al showed another system [10] used deep convolutional neural networks (Deep-ShrimpNet) and to normalise the images, they used multiple image processing techniques to differentiate between soft-shell shrimp and sound shrimp. Quach et al showed another system, [14]. In this paper, the authors take a different approach. They used several text classification algorithms (SVM, Random Forest), obtained a description of several diseases, and tokenized these to get the last final text source.

In this paper, our contribution is using transfer learning models to detect shrimp diseases and to know the highest accuracy model. Transfer learning is a new approach in deep learning that transfers knowledge of one or more tasks so that it can be used in a new task to enhance its accuracy. Machine learning models may now be applied to fresh data taken from sources that are completely different from the original data sources, thanks to this approach. By utilising already trained models on predefined big datasets, machine learning techniques avoid cold start issues, by using data from the source task. Transfer learning aims to increase generalisation capability in the target tasks. The difference between transfer learning and traditional machine learning is shown in figure 3

II. Related Work

Results of previous studies, [3] used to detect six types of shrimp disease, they used transfer learning models (inception-V3, MoblieNets-V1), they are CNN architecture based. Used this model on a collection of data set, they gathered it by launching a hub to collect shrimps images. The image was sent to experts to label them, and a background elimination was used on the shrimp images. They got an accuracy of 90% on inception-V3 model. This paper [15] develop a prototyping system to detect the prawns. The image of the prawns is captured by a camera, and they detect the prawns by representing their structure and comparing it to other prawns. Edge detection and K means clustering is used to detect the features and structure of the prawns, which resulted in prawns detection. This research [6] proposed a system to detect WSSV location and discoloration of the shrimp body. A set of 300 images were used to train their model, used a hybrid neural network that consists of ANN and fuzzy logic. Grayscale, edge detection, blob extraction are used as pre-processing on the images. Archived accuracy of 90% and Test-Retest Reliability Equation of 0.8. Another similar system [16], they detect three types of shrimp diseases. They are using Gaussian Filter to filter noise in the image, then using the grayscale and segmentation to extract features of the image. The classification process begins with KNN to classify the shrimp condition as normal, medium, abnormal. Achieved accuracy of 93% by using 30 samples for training and 10 samples for testing purposes. Their system [1] is detecting the infected fish and non-infected fish. Using pre-Processing techniques on images, such as resizing of images, contrast enhancement, RGB color to L*a*b color space, then used K means clustering segmentation to extract diseased area, then feeding it to SVM model. SVM performs 91.42 and 94.12 accuracy, with and without augmentation. Similar to the previous work, used noise filtering [17],then FCM and K-means to segment the diseased image. And to extract the features they used GLCM. Then feed these features to the SVM model to classify. By using the K-means, their accuracy is 97.9% and by using FCM was 96.48%. In this paper [12] they used machine learning to predict the AHPND which can have molarity up to 100% of the shrimp farms. they collected the data since 2010 from multiple farms. Logistic regression, artificial neural network, decision tree, and K-nearest neighbor are used to predict the AHPND. Logistic regression was more stable than the other algorithms with an accuracy of according to the hold-out test were 90.30% and 85.50%. The authors here [5] want to accurately identify and classify medicinal plants. they used transfer learning and deep learning combination to classify the medicinal plants. they collected 2300 samples from different growing areas. they used 16 different combinations of MobileNets. Best classification accuracy of 98.7% was achieved. Using transfer learning and convolutional neural network [4] to classify the Grain discoloration disease of rice. collected over 1000 samples, 566 samples are retained by experts. used an inception model to classify the discoloration. obtained 88.2% accuracy. Developed a model [2] which will classify the 8 types of skin disease using an image dataset. The implementation result of training obtains an accuracy of 96.63% with a dataset of 4000 images into eight classes. It

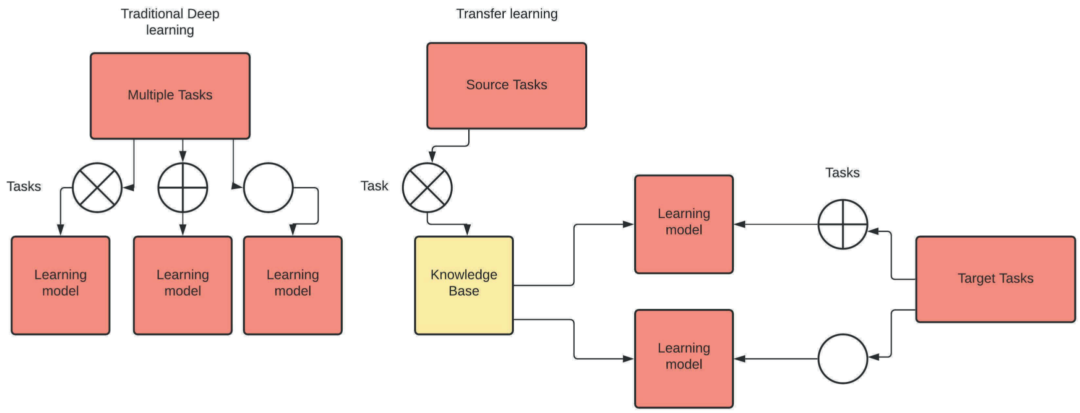


Fig. 3: The difference between transfer learning and traditional machine learning

works on convolutional neural networks (CNN) and fine-tuned transfer learning using the google net network. Developed a model called ShrimpNet [8], which is important to detect shrimps in fields of aquaculture. The proposed ShrimpNet has 95.48% accuracy in shrimp recognition. It includes two CNN layers and two fully-connected layers which are used to train and test the performance, with 16,132 collected samples. in this paper [7] the proposed artificial neural network techniques for identifying shrimp feed type, data set obtained in a shrimp farm. According to the inputs area, length, and weight features. The ANN model can recommend the most appropriate feed type with testing accuracy 97%. in this paper [18] combines emerging machine learning techniques based on a large number of crop pest and disease pictures, and introduces two kinds of convolutional neural networks AlexNet and GoogleNet. The detection accuracy rate of 98.48for the recognition of sick and healthy leaves [9]. They used deep learning for the automatic detection of plant disease is an innovative solution for agricultural applications. This paper presents a Convolutional neural network model based on VGGnet16 architecture for recognizing sick and unhealthy leaves. Several optimizers are tested to examine accuracy. The best results are obtained with Adadelta, With an accuracy of 90%. Develop a system [11] with accuracy 84%, 95% with a model (SURF-RootSIFT) and DNN VGG16 for automatic detection of 11 shrimp organs from histological images. There are 3 classes of organs (linfoid organ, pleopods, and epithelium stomach)

III. Methodology

This section going to discuss the Data set collection and the proposed system.

A. System Overview

Our System Overview figure 2 shows that a shrimp farmer or owner takes a photo of a shrimp using a mobile phone, then this photo will go through some manual preprocessing, such as removing any noise or background in the image, then using edge detection to detect the shrimp. The extracted photo will go through a transfer learning model.

That has been trained to classify 2 types of disease, then classify the shrimp as diseased or not.

After the processing, the photo of the shrimp will display on the mobile phone with a report that indicates if the shrimp are diseased or not and what type of disease and its percentage.

B. Data-set

the authors collected data-set images from many websites and Facebook fourms and YouTube videos. Collected 91 images total for white spot disease in shrimps and black gill and healthy shrimps, and then 200 images for diseased shrimp and healthy shrimps. These images had their background removed manually using a remove. bg tool. Any unwanted data in the background was removed. The dataset contains images that are of different resolutions and different sizes.

C. Model Approaches

the authors used 5 types of the Transfer Learning models (VGG16 , MobileNetv1 , ResNet50 , MobileNetv2 , InceptionV3) and one type of traditional Learning model. the authors going to talk about all models in detail next paragraph.

1) CNN Model: CNN model is a traditional learning convolutional neural network. Our model has 7 layers and. applying the architecture to images of size (320 * 240). the authors compile the model with a Adam optimizer and loss Categorical-Crossentropy and Binary-Crossentropy. The model has (input , conv2d(128) , maxpooling2d , conv2d(64) , maxpooling2d , flatten , dense(32) , dense). Model has 12,537,539 trainable params, and batch size is 4.

2) VGG16: A convolutional neural network is used in the VGG16 model. In ImageNet, a dataset of over 14 million images belonging to 1000 classes, the model achieves 92.7 percent top-5 test accuracy. At the 2014 ILSVRC, it was a well-known model that was shown. the authors used the pre-trained model without the top layers to apply the VGG16 architecture on photos of size (320 * 240). The model now has four layers after removing the top layer of VGG16. The VGG16 is built using an SGD optimizer and loss Categorical-Crossentropy and Binary-Crossentropy.

3) ResNet50: ResNet-50 is a deep convolutional neural network with 50 layers. The network can be loaded with a pre-trained version from the ImageNet database, which has been trained on over a million image. The network has already been pre-trained to detect 1000 different sorts of objects in images. The picture input size for the network is 224 224 pixels. the authors used the pre-trained model without the top layers to apply the ResNet50 architecture on images of size (320 * 240). The ResNet50 is built using an SGD optimizer using loss Categorical-Crossentropy and Binary-Crossentropy.

4) MobileNet-V1: MobileNet-V1 is a high-performance, portable deep convolutional neural network architecture that is presently being used in real-world applications. Instead of the conventional convolutions used in earlier architectures, MobileNets uses depthwise separable convolutions to construct lighter models. MobileNets now includes two additional global hyperparameters (width multiplier and resolution multiplier). In our MobileNet-V1 model, there are 91 layers. the authors used the pre-trained model without the top layers to apply the MobileNet-V1 architecture on images of size (320 * 240) as in figure 4. To develop the MobileNet-V1, the authors used an Adam optimizer and loss Categorical-Crossentropy and Binary-Crossentropy.

5) MobileNet-V2: MobileNetV2 is nearly identical to MobileNet, with the exception that it uses inverted residual blocks with bottlenecking features. It has a much smaller set of settings than the original MobileNet. MobileNets supports any image size greater than 32 by 32 pixels, with larger image sizes providing better performance. There are 160 layers in our MobileNet-V2 model. To apply the MobileNet-V2 architecture to photos of size (320 * 240), the authors used the pre-trained model without include the top layers. the authors use an SGD optimizer and loss Categorical-Crossentropy and Binary-Crossentropy to create MobileNet-V2.

6) Inception-V3: Inception-V3 is a 48-layer deep convolutional neural network. A pre-trained version of the network can be loaded from the ImageNet database, which has been trained on over a million photos. The network has been pre-trained to recognise 1000 different types of objects in photos. The network's picture input size is 224 × 224 pixels. The pre-trained model was employed, but without the top layers. There are 316 layers in our model. The Inception-V3 architecture will be applied to images with a resolution of (320 * 240). the authors use an SGD optimizer and loss Categorical-Crossentropy and Binary-Crossentropy to assemble the Inception-V3.

IV. Experiment and Results

The models were conducted on i7 9 Gen 16GB ram and GPU Nvidia GTX 1650ti 6GB. the authors experimented with

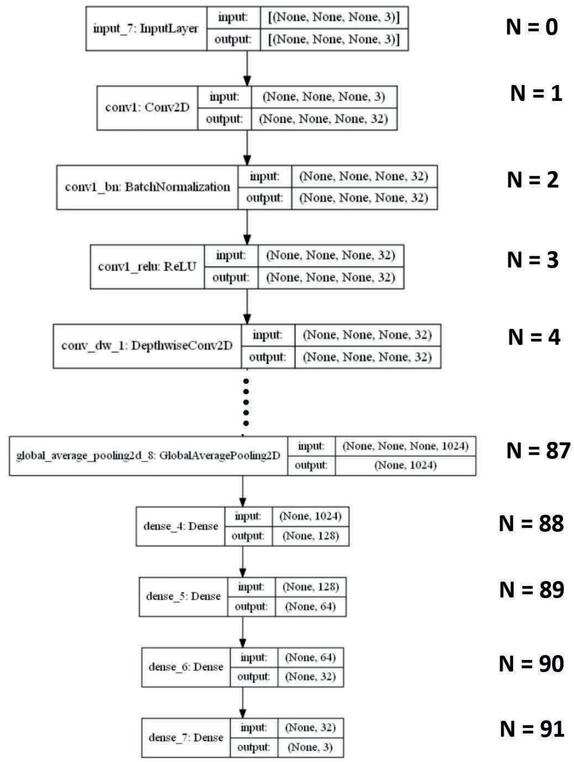
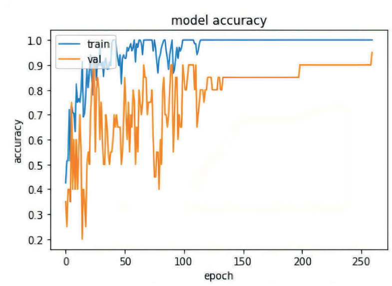
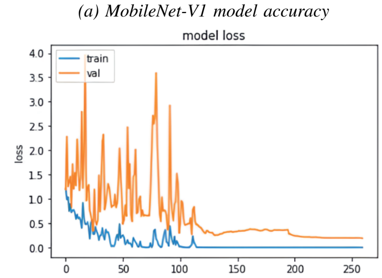


Fig. 4: Our MobileNet-V1 model architecture

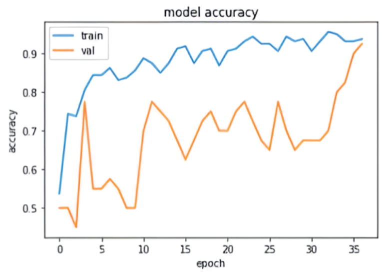


(a) MobileNet-V1 model accuracy

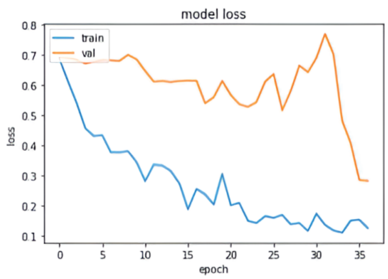


(b) MobileNet-V1 loss

Fig. 5: MobileNet-V1 model accuracy and loss in experiment one



(a) MobileNet-V1 model accuracy



(b) MobileNet-V1 loss

Fig. 6: MobileNet-V1 model accuracy and loss in experiment two

TABLE I: Number of images in dataset experiment one

Disease	Number of images in experiment one
Black gill	33
White spot	30
Healthy shrimps	28
Total	91

TABLE II: Number of images in dataset for experiment two

Disease	Number of images in experiment two
Healthy shrimps	81
Diseased shrimp	119
Total	200

TABLE III: Results experiment one

Models	Result in experiment one
CNN	70%
VGG16	80%
ResNet50	85%
InceptionV3	90%
MobileNetV2	90%
MoblieNetV1	95%

TABLE IV: Results experiment two

Models	Result in experiment two
CNN	75%
VGG16	75%
ResNet50	85%
InceptionV3	85%
MobileNetV2	85%
MoblieNetV1	92.5%

our 6 models with the collected dataset, which contains 91 images of 3 classes (White Spot, Black Gill, and healthy shrimp) as table I show and 200 images of 2 classes (diseased shrimp, healthy shrimp) as table II show. The images were scraped from the internet and community forums from Facebook. The images had their background removed. the authors used data augmentation on the training dataset where the authors re-scaled the images 1./225 and randomly horizontal flipped them, and the validation dataset images were re-scaled 1./225. The two datasets images were split into 80% for training and 20% for validation. the authors used an early stoping method to achieve the best out of the models and to reduce the overfitting of the models due to the small number of images.

1) Experiment One: In this experiment, the authors used images that contained 91 images of 3 classes (White Spot, Black Gill, and healthy shrimp) each class contains 30, 33, 28 images respectively, to detect the diseased shrimp and determine which type it was. In this experiment the authors used in our models loss as Categorical-Crossentropy as the authors have 3 class, so the authors treat our dataset as Categories, Moreover at the end, the model output is the type of disease and its accuracy percentage.

The results were VGG16 , MobileNetv1 , ResNet50 , MobileNetv2 , InceptionV3, CNN. Models reached accuracy as table III 80% ,95% ,85% ,90% ,90%, 70%. That shows model MobileNetv1 is the better classifier of the disease in shrimp in Experiment One, its loss and accuracy shown in figure 5.

2) Experiment Two: In this experiment, the authors used images that contained 200 images of classes (diseased shrimp,healthy shrimp) each class contains 119, 81 respectively. to detect the diseased shrimp from the normal shrimp, furthermore, the dataset contains multiple types of disease so that the authors can have diversity in our images. In this experiment the authors used in our models loss as binray-Crossentropy as the authors have 2 class, so the authors treat our dataset as binary, Moreover at the end, the model output is if the shrimp diseased or not .

The results were VGG16 , MobileNetv1 , ResNet50 , MobileNetv2 , InceptionV3, CNN. Models reached accuracy as table IV 75% ,92.5% ,85% ,85% ,85%, 75%. That shows model MobileNetv1 is the better classifier of the disease in shrimp in Experiment Two, its loss and accuracy shown in figure 6 and its confusion matrix in figure 7 .

V. Discussion

MobileNet-V1 archives the best accuracy in both experiments, due to the model have a low numbers of parameters which lead to better classification accuracy, moreover the model works better with latency, size ,low-power models parameterized to meet the resource limitations a differ of use cases. as far as our knowledge, [3] have reached 90% accuracy using transfer learning model ,their dataset contains 7 type of disease , each disease has class contains set of images. Also [6] reached 90% accuracy using Ann and fuzzy logic. their model detects one type of disease. the authors advice using

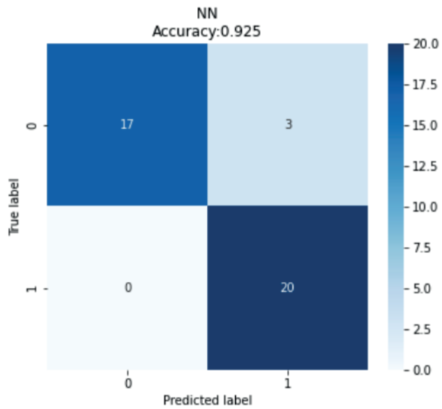


Fig. 7: Confusion matrix 0 indicate Healthy shrimps and 1 indicate Diseased shrimp

mobilenet-v1 to detect shrimp disease as it shown to have better accuracy in our created dataset, and May have better accuracy in different dataset.

VI. Conclusion and future work

The production of shrimp is very important for many countries. Shrimp farms face a lot of different challenges and issues. One of the most challenging issues for shrimp farms and owners is diseases. Diseases can cause molarity to reach 100% of the corpus. Early detection of this is critical. Early detection of this disease can help out the farm. the authors used two approaches to find out which was the better deep learning model to classify the disease. the authors used the transfer learning approach. the authors used five models to see and one traditional learning model to see which model achieved the best accuracy. The MobileNetv1 model achieves the best accuracy in both experiments, which is the best accuracy out of the five models. Our future work is to make our models more stable and get more data images to feed to our models.

VII. Acknowledgment

The authors want to thank Dr. Aleta C. Fabregas. Polytechnic University of the Philippines (PUP) Associate Professor and Nutrivet company, who helped in this study and dataset collection.

References

[1] M. S. Ahmed, T. T. Aurpa, and M. Azad, "Fish disease detection using image based machine learning technique in aquaculture," *ArXiv*, vol. abs/2105.03934, 2021.

[2] R. K. Bhadoria and S. Biswas, "A model for classification of skin disease using pretrained convolutional neural network," 2020.

[3] N. Duong-Trung, L.-D. Quach, and C.-N. Nguyen, "Towards classification of shrimp diseases using transferred convolutional neural networks."

[4] N. Duong-Trung, L.-D. Quach, M. Nguyen, and C.-N. Nguyen, "Classification of grain discoloration via transfer learning and convolutional neural networks," in *ICMLSC 2019*, 2019.

[5] N. Duong-Trung, L.-D. Quach, M.-H. Nguyen, and C.-N. Nguyen, "A combination of transfer learning and deep learning for medicinal plant classification," in *Proceedings of the 2019 4th International Conference on Intelligent Information Technology*, 2019, pp. 83–90.

[6] A. C. Fabregas, D. Cruz, and M. D. Marmeto, "Sugpo: A white spot disease detection in shrimps using hybrid neural networks with fuzzy logic algorithm," in *Proceedings of the 6th international conference on information technology: IoT and Smart City*, 2018, pp. 199–203.

[7] R. P. C. Gamara, P. J. M. Loresco, and R. Q. Neyra, "Artificial neural network-based decision support for shrimp feed type classification," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2019, pp. 1–5.

[8] W.-C. Hu, H.-T. Wu, Y.-F. Zhang, S.-H. Zhang, and C.-H. Lo, "Shrimp recognition using shrimpnet based on convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2020.

[9] F. Jakjoud, A. Hatim, and A. Bouaaddi, "Deep learning application for plant diseases detection," in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, ser. BDIoT'19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3372938.3372983

[10] Z. Liu, "Soft-shell shrimp recognition based on an improved alexnet for quality evaluations," *Journal of Food Engineering*, vol. 266, p. 109698, 2020.

[11] M. Mendieta, F. Panchana, B. Andrade, B. Bayot, C. Vaca, B. Vintimilla, and D. Romero, "Organ identification on shrimp histological images: A comparative study considering cnn and feature engineering," in *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, 2018, pp. 1–6.

[12] N. Minh Khiem, Y. Takahashi, D. Oanh, T. Hai, H. Yasuma, and N. Kimura, "The use of machine learning to predict acute hepatopancreatic necrosis disease (ahpnd) in shrimp farmed on the east coast of the mekong delta of vietnam," *Fisheries Science*, vol. 86, 05 2020.

[13] T. Morimoto, T. T. Zin, and T. Itami, "A study on abnormal behavior detection of infected shrimp," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2018, pp. 291–292.

[14] L.-D. Quach, L. Q. Hoang, N. Duong-Trung, and C.-N. Nguyen, "Towards machine learning approaches to identify shrimp diseases based on description."

[15] M. M. Sankar, C. N. Rao, G. Sailaja, N. Bhuvaneswary, and P. Gunasekhar, "White spot syndrome virus detection in shrimp images using image segmentation techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 9, 2013.

[16] R. S. Sankar, V. Jenitta, B. Kannapiran, and K. Valarmathi, "Diagnosis of shrimp condition using intelligent technique," in *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, 2017, pp. 1–5.

[17] J. Sikder, K. I. Sarek, and U. Das, "Fish disease detection system: A case study of freshwater fishes of bangladesh," 2021.

[18] J. Wu, B. Li, and Z. Wu, "Detection of crop pests and diseases based on deep convolutional neural network and improved algorithm," in *Proceedings of the 2019 4th International Conference on Machine Learning Technologies*, ser. ICMLT 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 20–27. [Online]. Available: https://doi.org/10.1145/3340997.3341010

4.3.2 Applying Deep Learning to Track Food Consumption

Mohamed Amr et al. , of Computer Science Mohamed Amr whose paper got published to the IEEE Xplore as part of the IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). Mohamed has presented his paper "Applying Deep Learning to Track Food Consumption and Human Activity for Non-intrusive Blood Glucose Monitoring" which proposes a system for predicting hyperglycemia/hypoglycemia in diabetic patients based on deep learning models that track food consumption and human activity. The Newyork based conference took place virtually on December 2nd, 2021. The presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community. [18]

2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) | 978-1-6654-0690-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/UEMCON53757.2021.9666662

Applying Deep Learning to Track Food Consumption and Human Activity for Non-intrusive Blood Glucose Monitoring

Mohamed Amr Samir
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Giza, Egypt.
mohamed.amr19@msa.edu.eg

Zeinab A. Mohamed
College of Pharmacy,
University of South Florida, Tampa, USA.
Nanotechnology Research Center,
The British University in Egypt, Cairo, Egypt.
zmohamed@mail.usf.edu

Mona Abdelmotaleb A. Hussein
National Institute of Diabetes and Endocrinology,
Cairo, Egypt.
Faculty of Dentistry,
October University for Modern Sciences and Arts (MSA),
Giza, Egypt.
mmotaleb@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University, Cairo, Egypt.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Giza, Egypt.
aezzat@msa.edu.eg

***Abstract*—Blood glucose monitoring is a wide area of research as it plays a huge part in controlling diabetes and many of its symptoms. A common human disease 'Diabetes Mellitus' (DM), which is characterized by hyperglycemia, has a number of harmful complications. In addition, the low glucose level in blood caused by hypoglycemia is correlated to fatal brain failure and death. In this paper, we explore a variety of related research to have a grasp on some of the systems and concepts that can assist in forming an autonomous system for glucose monitoring, including deep learning techniques. The proposed system in this paper utilizes non-intrusive Continuous Glucose Monitoring (CGM) devices for tracking glucose levels, combined with food classification and Human Activity Recognition (HAR) using deep learning. We relate the preprandial and peak postprandial glucose levels extracted from CGM with the Glycimc Load (GL) present in food, which makes it possible to form an estimation of blood sugar increase as well as predict hyperglycemia. The system also relates human activity with decrease in blood glucose to warn against possible signs of hypoglycemia before it occurs. We have conducted 3 different experiments; two of which are comparison between deep learning models for food classification and HAR with good results achieved, as well as an experimental result that we obtained by testing hyperglycemia prediction on real data of diabetic patients. The system was able to predict hyperglycemia with an accuracy percentage of 93.2%.**

***Index Terms*—Continuous Glucose Monitoring (CGM), Human Activity Recognition (HAR), Food classification, Deep learning, Hyperglycemia and hypoglycemia prediction, Glycemic Load**

I. Introduction

Consistent high blood glucose level referred to as 'Diabetes Mellitus' (DM), which is characterized by hyperglycemia, is the 7th leading cause of death in the United States and it is the number one contributor to kidney failure, lower-limb amputations, and adult blindness worldwide [1]. According to the International Diabetes Federation, there are about 463 million people (20-79 years) living with diabetes; this number is expected to rise to 700 million by 2045 [2].

Diabetes Mellitus is a chronic metabolic disorder that has two types; type 1, which is an auto-immune disease where the immune system attacks the pancreatic cells that release insulin, thus blood glucose level increases and the body cells cannot utilize the glucose molecules that are needed in all of the biological processes. While type 2- DM is caused by the partial destruction of the pancreatic cells that release insulin or by a condition called 'insulin resistance', where the cells are incapable of taking up insulin to utilize glucose and it also causes the blood glucose level to increase. There are several risk factors for DM such as aging, family history, obesity, Polycystic Ovary Syndrome (PCOS), metabolic disorders, and inactivity [1].

Hypoglycemia is another major disorder that is characterized by low blood glucose level and if the level continues to drop, the brain does not get enough glucose and stops functioning normally. Hypoglycemia leads to blurred vision, difficulty concentrating, confused thinking, slurred speech, numbness, and drowsiness, and if this condition stays for too long, the brain starves for glucose and this can cause seizures, coma and eventually death [3]. Hypoglycemia can be caused by high doses of insulin, skipping meals, excessive exercise, alcohol or some medications [4].

According to the National Institute for Clinical Excellence (NICE), normal blood glucose level should range between 4.0 to 5.4 mmol/L (72 to 99 mg/dL) when fasting, and increases up to 7.8 mmol/L (140 mg/dL) two hours after eating; a good control over blood sugar level in diabetic patients before meals should range from 4 to 7 mmol/L for people with type 1 or type 2 diabetes, and should be under 9 mmol/L after meals for people with type 1 diabetes and under 8.5mmol/L for people with type 2 diabetes [5].

Blood glucose levels are influenced heavily by the daily habits of people. One big contributor to blood glucose is food consumption. Food containing carbohydrates breaks down as sugar when digested, leading to an increase in blood glucose when it enters the blood [6]. In type 2 diabetes, exercise improves blood glucose management. On the other hand, excessive exercise could lead to developing symptoms of hypoglycemia [7]. Many studies suggest that balanced diet, exercise, and medications can lead to better diabetes control and avoiding many problems related to diabetes. [8].

Technologies for controlling diabetes has shown great advance in the last decade. One of the devices that assist patients with diabetes control is the Continuous Glucose Monitoring (CGM) device. "CGM works through a tiny sensor inserted under your skin, which tests glucose every few minutes. A transmitter wirelessly sends the information to a monitor." [9].

Tracking food consumption and human activity can be a great addition to a diabetes control regime. However, manual observation of such variables would pretty much get tiring and difficult to keep up with. Automated approaches for tracking these variables have been developing too fast in recent years; Machine learning techniques such as deep learning can help classify food and work even more challenging tasks like Human Activity Recognition (HAR). Many artificial intelligence (AI) apps and services rely on deep learning to improve automation by executing analytical and physical activities without the need for human intervention [10].

The proposed system in this paper utilizes the non-intrusive CGM device for tracking glucose levels, combined with food classification and Human Activity Recognition (HAR) using deep learning. We relate the preprandial and peak postprandial glucose levels extracted from CGM with the Glycimc Load (GL) present in food, which makes it possible to form an estimation of blood sugar increase and predict hyperglycemia. The system also relates human activity with decrease in blood glucose to warn against possible signs of hypoglycemia before it occurs.

Table[1] marks the similar ties with other external research. Reference papers are presented in the related work section.

TABLE I
RELATED WORK TIES

Paper Reference	***HAR***	***Food Classification***	***CGM Integrated***
[11], [12], [13], [14]	No	Yes	No
[15], [16], [17], [18]	Yes	No	No
[19], [20]	Yes	No	Yes
Our proposed system	Yes	Yes	Yes

II. RELATED WORK

Yang Gao et al. [11] developed a system that automatically recognizes eating activity through acoustic features. The proposed solution incorporates bluetooth headphones to gather acoustic data through the microphone. The data is then exposed to a deep learning technique that classifies the food type. S. Mezgec et al. [12] have developed a system that automatically recognizes images of food and beverages. The system applies deep learning on a dataset composed of freely downloadable images over the internet, combined with datasets from other dedicated research (in this case, Fake Food dataset). Abdulkadir Şengür et al.'s [13] system applies feature concatenation, feature extraction, and support vector machine (SVM) to classify food images. The deep learning models used are pre-trained VGG16 and AlexNet fed with three different datasets (Food-5k, food-11, and food-101). Md Tohidul Islam et al. [14] developed a system that applies convolutional neural network (CNN) to extract spatial features from food images. Pre-trained Inception V3 deep learning model trained on Food-11 dataset was used to classify the food images.

Daniele Rav et al. [15] developed an activity recognition application for low power devices. The implemented system applies multilayered deep learning models to classify data coming off mobile acceloreometer and gyro scope with respect to time. Mandal et al. [16] developed a platform that detects human activity based on motion patterns. The algorithm uses extracted acceloreometer sensor data and applies machine learning model SVM for binary data classification. Zhenguo Shi et al. [17] developed a human activity recognition system with enhanced channel state information deep learning neural networks. The system starts by extracting deep features using LSTM-RNN model then classifies the extracted features with softmax regression. Md. Atikuzzaman et al. [18] developed a system that applies HAR-feature based classifier to detect human poses in CCTV camera video. The machine learning model was trained on a custom dataset of 5648 images.

Weixi et al. [19] developed a multi-task deep learning system that predicts hypoglycemia and hyperglycemia based on daily habits of diabetic patients. The proposed system utilizes the CGM device for glucose level monitoring, and a manual approach of users providing daily information. The system resulted in accuracy number of 82.14% in predicting hyperglycemia/hypoglycemia. Zhou et al. [20] developed a mobile application that works as a temporary alternative to CGM devices when they become unwearable. To make full use of the available blood glucose information, the system employs an RNN model, an efficient learning paradigm. The newly constructed grouped input layers, combined with the usage of a deep RNN model, allow for the creation of blood glucose models for the general public based on restricted personal measurements from both single-user and grouped-user viewpoints.

III. Methodology

A. System Overview

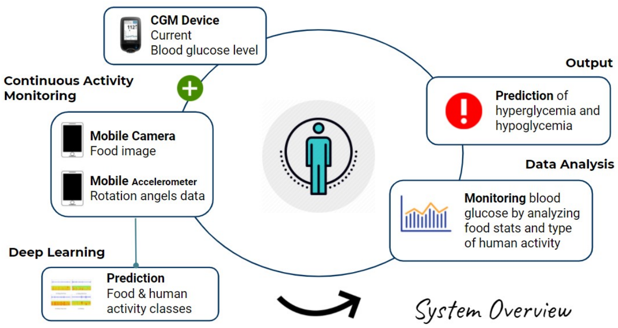


Fig. 1. System overview demonstrates the input and the flow of data between each system component.

Input data are current blood glucose level, food images, and accelerometer data. Deep learning models are used to classify food and recognize human activity. The output is fed into an analysis component where it attempts to make predictions. The patient has to provide a food image in order to make a prediction. The system then extracts preprandial blood glucose (blood glucose level before eating) in order to calculate peak postprandial glucose (blood glucose increase after meal) and test for hyperglycemia. Human activity data are retrieved from the accelerometer and fed into the model in order to recognize the type of activity, which is then used to predict hypoglycemia with the help of CGM. The Patient can see the results of monitoring and predictions through an output interface.

B. Food Image Classification

Deep learning techniques applied in the system make it possible to recognize food images provided by the patients. In order to train our system models, we used the freely available food-101 dataset which originally contains 101 food categories with 1000 images each. However, food classes are slimmed down to only address the most common ones. Other pre-processing techniques such as common size and data splitting for validation and testing are applied. We tested the same dataset with two different deep learning models for image classification. Results for each of these two models are discussed in the experiments section.

One of the applied deep learning models is a multi-layered Convolutional Neural Network (2D CNN) model for image classification. The model follows a multi-layered architecture in order to map to output neurons. Input image is segmented at each layer, with max pooling applied to determine the unique identifying value of every segmentation. A final dense layer with softmax activation function is used to estimate a probability for each class. Prediction of a certain class can be made based on the max value.

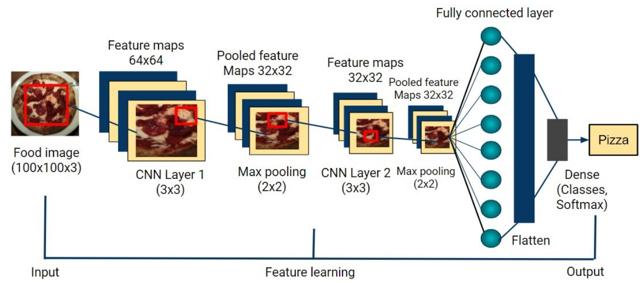


Fig. 2. Illustrates the applied 2D CNN deep learning model architecture.

MobileNetV2 is another transfer learning based model that we trained on the same dataset. Transfer learning is a machine learning technique in which a model created for one job is utilized as the basis for a model on a different task. Given the vast compute and time resources required to develop neural network models on these problems, it is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks [21].

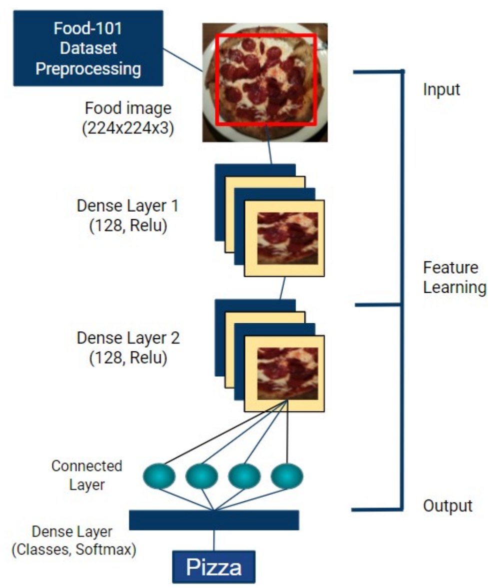


Fig. 3. Illustrates the applied pre-trained MobileNetV2 transfer learning model architecture. The dataset images are preprocessed and split as training, validation, and testing data. Two dense layers are used with relu activation then connected to a final softmax activated layer for prediction.

C. Human Activity Recognition Using Deep Learning

Automated human activity recognition is useful for diabetes control as it can help in predicting possible cases of hypoglycemia before it occurs. We trained our deep learning models with the WISDM (Wirless Sensor Data Mining) dataset [22], which offers 2 sets of recorded data for lab controlled and in-field human activity accelerometer data. The dataset was collected by different users and originally contains 6 activity classes (walking, jogging, upstairs, downstairs, standing, and sitting).

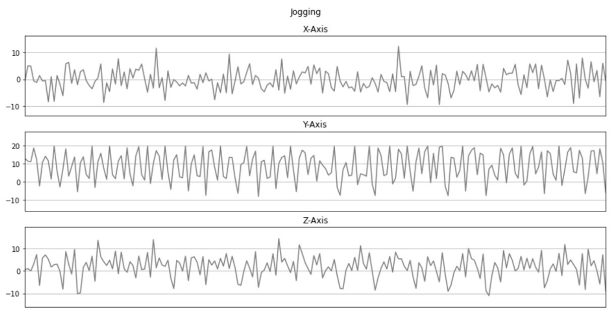


Fig. 4. Sample data visualization of an activity class (jogging) shows the extracted rotational values of lateral, vertical, and sagittal from the accelerometer.

One of the models we trained for human activity recognition is a 2D CNN model. Activity classes in the dataset have uneven data between each activity class which we have re-balanced according to the smallest data-containing class. Other pre-processing essentials such as standardization, labeling, and reshaping were applied before the data is fed into the input layer. The input layer is a 240-element vector that is a flattened representation of 80 time slices containing three accelerometer readings each. One additional layer is used to reshape the input into 80x3 matrix. For the classification task, the model applies 3 hidden layers with 100 nodes each. For the output layer, a final dense layer with softmax activation function is used to make a prediction.

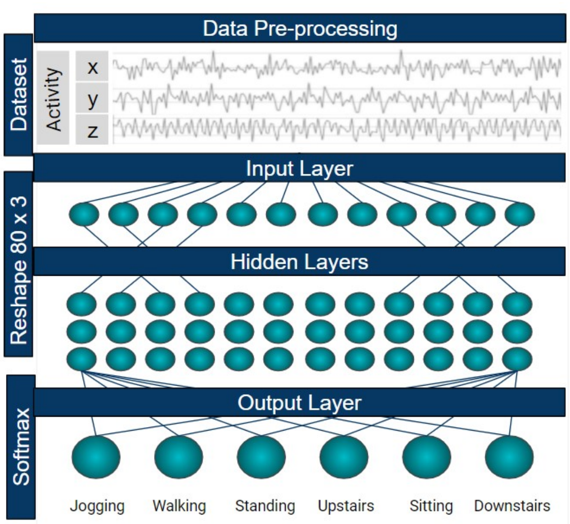


Fig. 5. Illustrates the applied 2D CNN model architecture for human activity classification.

We used several other deep learning models to train on the same dataset, including the Long Short-Term Memory (LSTM). Results from these models are discussed in the experiments section.

D. Blood Glucose Monitoring

Hyperglycemia and hypoglycemia are common harmful symptoms that occur when a person has diabetes. Predicting these symptoms can significantly help patients in controlling diabetes and avoiding any related complications. The proposed system supports blood glucose monitoring by combining the CGM device with the constructed deep learning components. With these resources available, the system can estimate how much blood glucose will rise after eating a certain meal (peak postprandial glucose), as well as predict hyperglycemia and hypoglycemia before they occur.

In order to estimate the effects of food on blood glucose, scientists have developed the two numerical terms 'Glycemic Index' (GI) and 'Glycemic Load' (GL). Many foods have pre-calculated GI and GL that help with food-related diabetes control, as they provide a generalized number for a certain portion of food to indicate how fast blood glucose will rise or how much it will increase after ingesting the contained carbohydrates. We rely on the glycemic load measurement specifically as it is derived from both the glycemic index and carbohydrates of a certain food, thus it is more beneficial for estimating peak postprandial blood glucose without the need of specifying a portion size for the food.

The way hyperglycemia prediction works is that CGM provides a current state of blood glucose level, when a food picture is provided, it gets classified and labeled with its corresponding GL. We apply the following equation to estimate peak blood glucose after consuming each food item;

$$PeakBG = CurrentBG + (GL \times GlucoseIncreasePerGL) \tag{1}$$

Equation (1) is used to estimate peak postprandial glucose level referred to as 'PeakBG'. 'CurrentBG' refers to the current blood glucose level. We estimate the glucose increase per GL based on the patient's weight class (source: [25]). However, a more generalized value is applied if patients do not specify their weights. Prediction is issued when peak postprandial glucose level is greater than the threshold of 11mmol/L for hyperglycemia level [23]. List of calculated glycemic load in food can be found at the International table of glycemic index and glycemic load values [24].

Hypoglycemia prediction is based mainly on human activity. Warning prediction is generated when monitored glucose levels are noticeably decreasing while the person is doing some sort of draining activity such as jogging. The warning levels are considered when blood glucose levels are falling and is estimated to reach below 3mmol/L (default level of hypoglycemia) after a certain time.

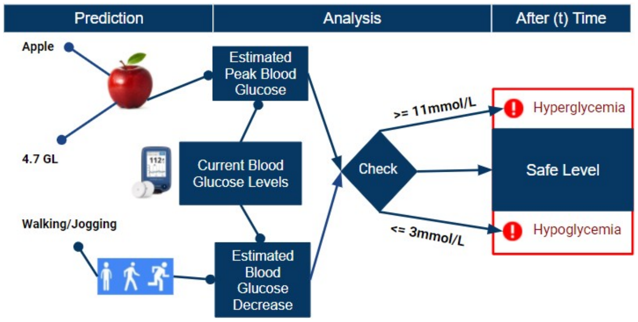


Fig. 6. Shows a sample prediction for a certain food and activity. The system estimates blood glucose increase or decrease after a certain amount of time by comparing with the current glucose level of the patient. It then checks if the output level falls within the warning threshold for hyperglycemia or hypoglycemia.

IV. Experiments

A. Comparison of Deep Learning Models in Human Activity Recognition

In this experiment we tested 4 different models for activity recognition on the same WISDM dataset. The goal from this experiment is to determine the best model for recognizing human activity in our system. We evaluate each model based on accuracy and the time it took for training and testing. Table (2) evaluates each of these models after 10 epochs of training.

TABLE II
Results of Human Activity Recognition Experiment

Model	*Training Time*	*Accuracy*	*Val Accuracy*	*Test Time*
2D CNN	12ms/step	0.9412	0.9065	3s
GRU	450ms/step	0.8118	0.7696	2m 7s
Simple RNN	138ms/step	0.7859	0.7037	42s
LSTM	221ms/step	0.8912	0.8546	1m 24s

Table (2) results show that 2D CNN has the highest accuracy value and lowest timing in human activity recognition for the applied dataset. Validation flaws in this particular model has been mainly the cause of confusion between the upstairs and downstairs classes as the values contained within the trained model are similar as shown in fig (7).

B. Comparison of Deep Learning Models in Food Classification

The trained models for food classification were tested and validated. Testing and validation data were derived from the food-101 dataset. The goal from this experiment is to determine the better model in classifying food images from the same provided dataset. Table (3) contains the evaluation of each model after 5 epochs of training.

Table (3) concludes that our MobileNetV2 model has better speed and accuracy when training on food images. Validation flaws has been mainly caused by confusion between similar looking food as shown in fig. (6).

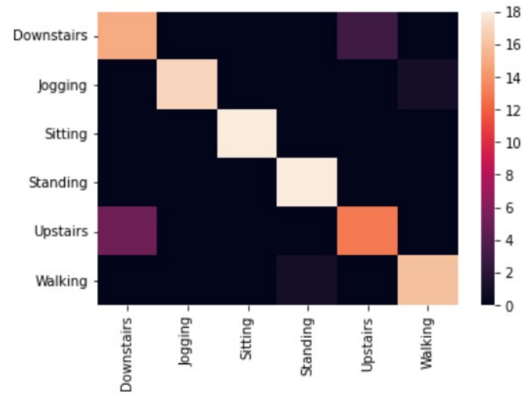


Fig. 7. Heatmap visualization of the six activity classes shows that there is noticeable confusion between the upstairs and downstairs classes. The columns and rows represent the actual and predicted classes resulted from the 2D CNN model validation.

TABLE III
Results of Food Classification Experiment

Model	*Training Time*	*Accuracy*	*Val Accuracy*	*Test Time*
2D CNN	21s 1s/step	0.8772	0.7852	4m
MobileNet	21s 1s/step	0.9887	0.8045	3m

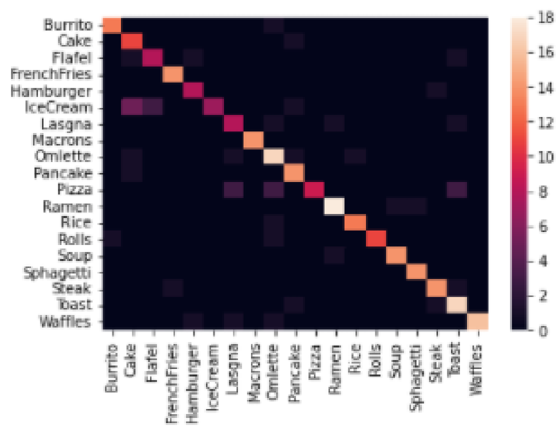


Fig. 8. Heatmap visualization of 19 sample food classes shows that flaws in validation is caused by confusion between similar looking food classes.

C. Experimenting Hyperglycemia Prediction

The final set of tests were conducted to validate the system's capability in predicting hyperglycemia. For the purpose of this experiment, diabetic patient data subset derived from the D1NAMO dataset [26] was used as it includes recorded data collected by diabetic patients using CGM devices while providing timed food images and accelerometer data. System inputs included glucose levels, food image, and the registered time. Classifying food was controlled and limited to food belonging to classes addressed within the trained model. Visual extraction for peak postprandial blood glucose was done to indicate how close was the estimated increase by the system to a hyperglycemia level of 11mmol/L, described as a total accuracy value. The accuracy result was derived from 10 recorded hyperglycemia cases that was provided for prediction. The system has achieved a prediction accuracy of 93.2 percent, noting that faults in prediction have mainly been the cause of undefined patient weights and insulin interference in some cases.

V. Conclusion

The system in this research proposes a solution for blood glucose monitoring by applying deep learning techniques to classify food and recognize human activity. By combining the output with CGM, we were able to construct an equation for hyperglycemia prediction based on the glycemic load present in food and detecting hypoglycemia based on human activity. In the related work section we explore a set of research papers that included several proposed deep learning systems for diabetes management, food classification, and human activity recognition. Finally, we have conducted 3 different experiments to validate our proposed system.

VI. Future Work

The system ideology can be improved by attaching more autonomous recognition of other daily patient habits. CGM values can be possibly predicted after the device becomes unwearable by tracking food habits and human activity.

References

[1] Centers for Disease Control and Prevention (CDC). Diabetes Fast Facts. https://www.cdc.gov/diabetes/basics/quick-facts.html

[2] International Diabetes Federation (IDF) (2020). About Diabetes. https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html

[3] American Diabetes Association (ADA). Hypoglycemia (Low Blood sugar). https://www.diabetes.org/healthy-living/medication-treatments/blood-glucose-testing-and-control/hypoglycemia

[4] National Health Service (NHS). Low blood sugar (hypoglycaemia). https://www.nhs.uk/conditions/low-blood-sugar-hypoglycaemia/

[5] National Institute for Health and Care Excellence. Diabetes. Available: https://www.nice.org.uk/guidance/conditions-and-diseases/diabetes-and-other-endocrinal–nutritional-and-metabolic-conditions/diabetes

[6] Harvard School of Public Health (nd.). Carbohydrates and Blood Sugar. Available: https://www.hsph.harvard.edu/

[7] Colberg SR, Sigal RJ, Fernhall B, et al. Exercise and type 2 diabetes: the American College of Sports Medicine and the American Diabetes Association: joint position statement executive summary. Diabetes Care. 2010;33(12):2692-2696. doi:10.2337/dc10-1548

[8] Diatribe (Dec, 2019). How Does Diabetes Affect Your Body?. https://www.webmd.com/diabetes/guide/risks-complications-uncontrolled-diabetes

[9] National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). Continuous Glucose Monitoring. https://www.niddk.nih.gov/

[10] International Business Machines (IBM). What is deep learning?. https://www.ibm.com/cloud/learn/deep-learning

[11] Y. Gao et al., "iHear Food: Eating Detection Using Commodity Bluetooth Headsets," 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016, pp. 163-172, doi: 10.1109/CHASE.2016.14.

[12] Mezgec, Simon Seljak, Barbara. (2019). Using Deep Learning for Food and Beverage Image Recognition. 5149-5151. 10.1109/BigData47090.2019.9006181.

[13] A. Şengür, Y. Akbulut and Ü. Budak, "Food Image Classification with Deep Features," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019, pp. 1-6, doi: 10.1109/IDAP.2019.8875946.

[14] M. T. Islam, B. M. N. Karim Siddique, S. Rahman and T. Jabid, "Food Image Classification with Convolutional Neural Network," 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018, pp. 257-262, doi: 10.1109/ICIIBMS.2018.8550005.

[15] Ravì, Daniele Wong, Charence Lo, Benny Yang, Guang-Zhong. (2016). Deep learning for human activity recognition: A resource efficient implementation on low-power devices. 71-76. 10.1109/BSN.2016.7516235.

[16] I. Mandal, S. L. Happy, D. P. Behera and A. Routray, "A framework for human activity recognition based on accelerometer data," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), 2014, pp. 600-603, doi: 10.1109/CONFLUENCE.2014.6949248.

[17] Z. Shi, J. A. Zhang, R. Xu and G. Fang, "Human Activity Recognition Using Deep Learning Networks with Enhanced Channel State Information," 2018 IEEE Globecom Workshops (GC Wkshps), 2018, pp. 1-6, doi: 10.1109/GLOCOMW.2018.8644435.

[18] M. Atikuzzaman, T. R. Rahman, E. Wazed, M. P. Hossain and M. Z. Islam, "Human Activity Recognition System from Different Poses with CNN," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), 2020, pp. 1-5, doi: 10.1109/STI50764.2020.9350508.

[19] Gu, Weixi. (2017). Non-intrusive blood glucose monitor by multi-task deep learning: PhD forum abstract. 249-250. 10.1145/3055031.3055049.

[20] Weixi Gu, Yuxun Zhou, Zimu Zhou, Xi Liu, Han Zou, Pei Zhang, Costas J. Spanos, and Lin Zhang. 2017. SugarMate: Nonintrusive Blood Glucose Monitoring with Smartphones. PACM Interact. Mob. Wearable Ubiquitous Technol. 1, 3, Article 54 (September 2017), 27 pages.

[21] J. Brownlee. A Gentle Introduction to Transfer Learning for Deep Learning. 2017. https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[22] Jennifer R. Kwapisz, Gary M. Weiss and Samuel A. Moore (2010). Activity Recognition using Cell Phone Accelerometers, Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10), Washington DC.

[23] The Global Diabetes Community. Diabetes and Hyperglycemia. https://www.diabetes.co.uk/Diabetes-and-Hyperglycaemia.html

[24] K. Foster-Powell, S. H. Holt, and J. C. Brand-Miller, International table of glycemic index and glycemic load values. 2002.

[25] How Does Diabetes Affect Your Body?. Dec, 2019. https://www.webmd.com/diabetes/guide/risks-complications-uncontrolled-diabetes

[26] F. Dubosson, J. Ranvier, S. Bromuri, J. Calbimonte, J. Ruiz, M. Schumacher, The open D1NAMO dataset: A multi-modal dataset for research on non-invasive type 1 diabetes management, Informatics in Medicine Unlocked. Volume 13, 2018, Pages 92-100, ISSN 2352-9148

4.3.3 Parallelization of One Dimensional First Fit Decreasing Algorithm

Eriny Wessa et al., whose paper got published to the IEEE Xplore as part of the International Conference on Computer Engineering and Systems Eriny Wessa presented her paper entitled " Parallelization of One Dimensional First Fit Decreasing Algorithm" which discussed the effect of parallel processing on the computational time and utilization rate of the bin packing algorithm. The Conference took place virtually on 15/12/2021, and the presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community. [24]

2021 16th International Conference on Computer Engineering and Systems (ICCES) | 978-1-6654-0867-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCES54031.2021.9686107

Parallelization of One Dimensional First Fit Decreasing Algorithm

Eriny Wessa
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Cairo, Egypt
eriny.yousrey@msa.edu.eg

Ayman Atia
HCI-LAB,Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Giza Egypt
Email: aezzat@msa.edu.eg

***Abstract*—Bin packing is an optimization problem defined as placing different sized objects into similar containers or bins to minimize the number of used bins. This problem has different variations based on the dimensions of the bins, placement limitations, and priority. This paper focuses on one-dimensional bin packing. Two algorithms are explored in this paper, which are First Fit, First Fit Decreasing. The contribution of the paper is to explore the effect of parallelization on the First Fit Decreasing algorithm regarding the processing time and the utilization rate. Furthermore, the effect of using different numbers of concurrent workers on the problem is also explored. We proved that just by parallelizing the sorting in the First Fit Decreasing algorithm, we can decrease the computation time by 4.73% without affecting the utilization rate.**

***Index Terms*—Bin Packing, Parallel processing, Energy efficiency.**

I. Introduction

The one-dimensional bin packing problem is an NP-hard optimization problem [8] [4] [12] defined as placing different sized objects into similar containers or bins to minimize the number of used bins. In other words, given a list of N objects where each object has a different size or weight, we are asked to pack them into identical bins with a fixed capacity while using the least amount of bins possible. Furthermore, numerous variants of the problem exist in practice based on the dimensions of the bins, placement limitations, and priority [14] [13].

One dimensional bin packing has various applications that effect Energy and recourse consumption. For example, bin packing can be used in memory and resource allocation. Moreover bin packing is also used in scheduling tasks on parallel processors to improve performance [2] [6] [11]. Furthermore, bin packing is also used in line balancing where all workstations have a predefined cycle time, the task is to assign a list of operations to a number of workstations while minimizing the number of workstations used [5] allowing for more efficient energy consumption.

The First Fit (FF), and the First Fit Decreasing (FFD) algorithms are variants of the bin packing approximation algorithms. Both algorithms place each item into the first bin that fits, However, the First Fit Decreasing algorithm sort the items in a decreasing order before placing them. These two algorithms are the focus of this study, As the worst case performance bounds of the FF algorithm is $17/10$, and the worst case performance bounds of the FFD algorithm is $11/9$ [7] [10] [12] [3].

The FF and FFD algorithms reach near-optimal results, however, they represent a challenging complexity of $O(n^2)$ for using a simple list structure or $O(n \log n)$ for using a tree structure. Moreover, The trade-off between the utilization rate and the processing time is an important challenge to balance. Utilization is represented by the number of bins used. Furthermore, the ratio between the used and unused area of the bin is an important parameter in measuring utilization.

This paper focuses on one-dimensional bin packing algorithms, specifically the First Fit (FF) and the First Fit Decreasing (FFD) algorithm. This paper compares the FF algorithm and two variants of the FFD algorithm, one variant where the sort is sequential and the other where the sort is parallel. The main contribution of this paper is to study the effect of parallel processing on the execution time and the utilization rate of the FFD algorithm. Furthermore, The relation between the degree of parallelism represented by the number of parallel workers working concurrently, and the speed up is explored.

II. Related work

The bin packing problem has been researched in the past from different angles and approaches. This section is therefore divided by the optimization approach taken by different researchers. Section II-A discusses the traditional optimization efforts in bin packing. Also, section II-B discusses the parallel approach to the bin packing problem.

A. Traditional optimization efforts

The motivation to use the FFD in this research was the work by Johnson et al. [9]. They argued that although the worst-case bound of the algorithm is to use 22.2% more space than the optimal solution, that the algorithm actually uses no more than 2% more spaces on average.

Zehmakan et al. [16] proposed two algorithms in the bin backing problem. The first Algorithm was a $3/2$ approximation algorithm had a similar time order to the FFD algorithm but used less space, as the FFD algorithm saves all bins during execution. The second algorithm is a modification to the FFD algorithm that decreases the time order to linear.

Balogh et al. [1] focused on the parametric case of the online bin packing algorithms algorithms. They offered mathematical proof that sorting the objects to be packed in a decreasing yields a lower bound of $54/47$. They also argued that if the size of the largest elements is in the interval between $8/29$ and $1/2$ then the FFD algorithm yields upper bound of $71/60$. Therefore, in this paper we generated the objects to be normally distributed, as to minimize the possibility of large objects.

B. Parallel processing optimization efforts

Varsamis et al. [15] implemented a MATLAB program to parallelize the best fit and the best fit decreasing algorithm. They aimed to reduce computation time by dividing the problem into sub-problems and applying the best fit decreasing algorithm to each sub-problem. They came to the conclusion that in order to divide the data into partitions, each partition must be representative of the whole data set to achieve the best results. They also noticed that in large data sets when the number of objects is 2^{16} and 2^{18}, the parallelization of the BFD algorithm decreased the time from 50% to 90%. Similarly, Ghosh et al. [7] implemented multiple Parallel algorithms for the bin packing problem. They demonstrated the usage of the task and data parallelism.

III. Methodology

This section discusses the implementation. Firstly, section III-A discusses the problem instances. Secondly, section III-B discusses the First Fit Algorithm and how it is implemented. Finally, the First Fit Decreasing Algorithm is discussed with its two variants, the sequential III-C and the Parallel III-D.

A. Problem instances

The problem instances in bin packing are defined by four parameters. Bin capacity (BC) is one of these parameters, which represent the maximum capacity of all the bins. The second parameter is the number of objects (n). The last two parameters are the lower (v_1) and upper (v_2) bounds, which are fractions between 0 and 1 such that the objects are between $v_1 \times BC$ and $v_2 \times BC$. Any bin backing problem instance can be represented by the tuple (BC, n, v_1, v_2). Throughout this paper, different tuples which correspond to different data sets are used. Furthermore, all the objects are randomly generated and normally distributed to minimize the chance of having really low or really high values.

B. First Fit Algorithm

The First Fit Algorithm is an optimization algorithm for bin backing. The algorithm simply places the objects into the first bin with enough space. If there was no opened bin with enough space, a new bin is added. Figure 1 shows a pseudo-code for the algorithm.

Algorithm 1 First-Fit

```
for All objects i = 1, 2, ..., n do
    for All bins j = 1, 2, ... do
        if Object i fits in bin j then
            Pack object i in bin j.
            Break the loop and pack the next object.
        end if
    end for
    if Object i did not fit in any available bin then
        Create new bin and pack object i.
    end if
end for
```

Fig. 1: Pseudo code for first fit algorithm

C. First Fit Decreasing Algorithm (sequential)

The First Fit Decreasing Algorithm starts with sorting the objects in decreasing order, Then the algorithm continues the same as the First Fit Algorithm. In the sequential version of this algorithm, an implementation of the merge sort was employed to sort the objects. The merge sort algorithm works in two steps. The first step is a recursive function that divides a list into two halves until it can no more be divided. The second step is to merge the smaller lists into a new ordered list. Figure 2 is an Example on the merge sort. The figure shows that the array is divided until it can no longer be divided. In this example, level 4 reaches a stage at which the array can no longer be divided. At this step of the algorithm, the left most sub array can now be merged. The merge occurs in ascending order to insure that the final result is a sorted array. Merge sort was chosen because it is stable and remains consistent with a variety of problem sizes. Furthermore, the merge sort has a worst-case time complexity of $O(n \log n)$.

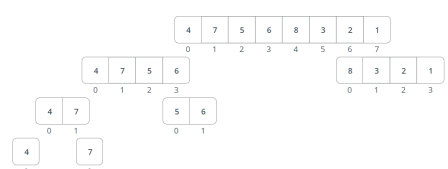


Fig. 2: An example of how merge sort works.

D. First Fit Decreasing Algorithm (Parallel)

The parallel version of the First Fit Decreasing algorithm implements a parallel merge sort. Since merge sort already uses the divide and concur approach, it benefits from penalization. Each worker divides the objects list into two halves, this process is repeated recursively until each worker holds only one element. Then, each worker compares and merge two elements together in decreasing order until the entire array is sorted. Figure 3 Shown an example on parallel merge sort.

The figure shows that each worker divides the array into 2 halves and checks if the sub-array can divided further. Then each worker merges two sorted sub-arrays into one array. This process repeats until the whole array is sorted. Notice that in this example the sort is ascending while the implementation is descending. Furthermore, We added a parameter to the parallel implementation of FFD algorithm that controls the level of parallelism. The level of parallelism is the maximum number of allowed workers to run concurrently.

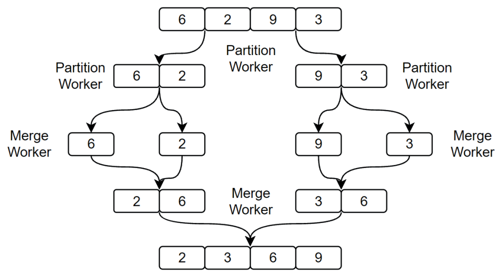


Fig. 3: An example of The Parallel merge sort.

IV. Experiment

This section discusses two experiments and the environment in which they were executed.

A. Setup environment

In order to study the effects of parallelism on the performance of the bin packing problem, We must set up a standardized environment. All the experiments performed in this paper had the following environment: Processor Intel Core i7-10750H 2.60GHz and 16 GB main memory.

B. Experiment one

In this experiment, we explore the effect of using different numbers of workers on the processing time of the FFD algorithm.

1) Objective: This experiment was performed to analyze some questions. Firstly, how can we parallelize the sorting step in the FFD algorithm to affect the computing time? Does it affect the utilization? Secondly, parallelization is known to cause unnecessary overhead for small problem instances, then at which point does the parallelization becomes effective?

2) Method: In order to answer those questions, we need to keep a few factors from affecting the experiment. Therefore, in this experiment, all the bins have bin capacity of 500, lower bound 0.2 and upper bound 0.8. The size of the problems is from 10000 to 60000. The objects are randomly generated between the lower bound and the upper bound. This experiment is focused on the parallel FFD algorithm. Each of the problem instances is solved with the FFD parallel algorithm with a different number of

number of objects	***number of bins***	***total spaces***	***average waste***
10000	5018	335459.37	66.85120
15000	7536	676891.59	89.82107
20000	10009	356514.4	35.61938
25000	12476	272495.59	21.84158
30000	15155	1692539.83	111.68194
35000	17570	1013370.2	57.67616
40000	20181	2195508.82	108.79088
45000	22591	1697240.31	75.12904
50000	25123	1330406.11	52.95570
55000	27593	1323925.87	47.98049
60000	30139	1704587.67	56.55753

TABLE I: The utilization rate of the FFD parallel algorithm. The average waste is the number of total unused spaces divided by the number of bins used.

concurrent workers from 2 workers to 12 workers. For each (problem size, number of workers) pair, the average time of 10 trails is measured. Indeed, the processing time is the main point that we are focusing on in this paper. However, we also tested the utilization rate of the algorithm as well. Parameters like the number of used bins, the number of unused spaces and average empty space per bin are computed.

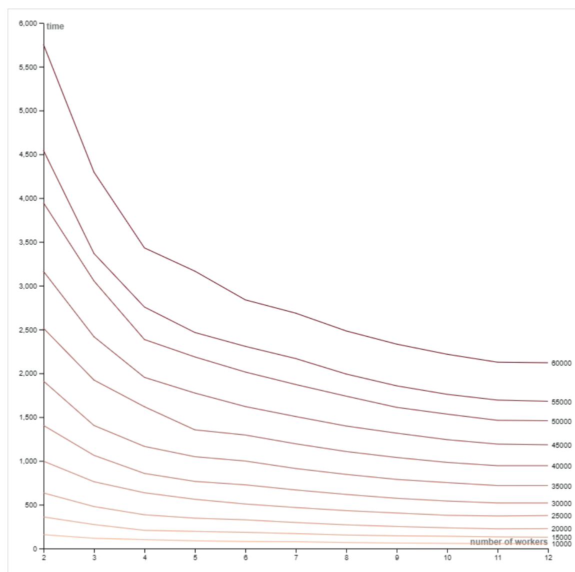


Fig. 4: The X axis represent the number of concurrent workers. The Y axis represent time in milliseconds. Each line is a problem instance with different size.

3) Result: Regarding the effect of parallelization on the processing time, Figure 4 shows the relation between the number of workers and the time in milliseconds. Overall, it is clear that by increasing the number of workers, the processing time decreases. Furthermore, the relation between the problem size and the speedup is also clear. All the problem sizes in the experiment have benefited from parallelization, However, the speed up differs from one instance to another. For example, for the smallest problem size 10000, the processing time for 2,3 and 12 workers is 157.57, 116.7 and 55.139 milliseconds respectively. However, for the biggest problem size 60000, the processing time for 2,3 and 12 workers is 5752, 4295.9 and 2120.3 milliseconds respectively. Furthermore, it is safe to say that the pattern shown in figure 4 is expected to be expanded for larger problem sizes. This proves that the FFD algorithm can benefit greatly from any level of parallelization in large problem sizes.

This benefit of decreased processing time has no negative effect. Meaning that the utilization rate of the FFD is not affected. That is because the original sequential FFD algorithm also sorts the objects to be packed before processing them. Therefore, it is not surprising that the results in I are the same for both the sequential and the parallel versions on the FFD. The table describes the number of used bins, total empty spaces and the average waste per bin. The number of used bins is the main attribute for evaluating a bin backing algorithm. Table I shows that for the problem instance provided, the number of used bins are always about half that off the number of objects. Therefore, for these problem instances, the relation between the number of used bins and the problem size appears to be liner.

C. Experiment Two

1) Objective: In this experiment, we compared The three algorithms FF, FFDS (Sequential) and FFDP (Parallel) to find the processing time and number of used bins for each algorithm.

2) Method: In this experiment, several data sets were used. All the data sets follow the tuple $(500, n, 0.2, 0.8)$, where that the number of objects n are randomly generated from 100000 and 160000. This gives us a chance to focus on the size of the problem, as the differences between each data set in the number of objects needing to be packed. Each data set was fed into each of our 3 algorithms 10 times.

3) Result: Figure 5 is a graph representing the average time in seconds for each data set. The parallel FFD algorithm shows the best results. This proves that just by parallelizing the sorting in the FFD algorithm, we can save a lot of time. The FFDP algorithm is 21.9% faster than the FF algorithm and 4.73% faster than the FFDS algorithm. The number of workers in this experiment was 12, due to the set up environment mentioned above. However, if using 12 workers achieved 4.73% speed up, then by the relation proved in the previous experiment, the potential of this approach seems promising.

Regarding the utilization rate of every algorithm, figure II shows the relation between the size of the data set as a number of objects and the utilization as the number of used bins. The FFDP and FFDS algorithms have the same number of bins as the only difference between them is the sorting

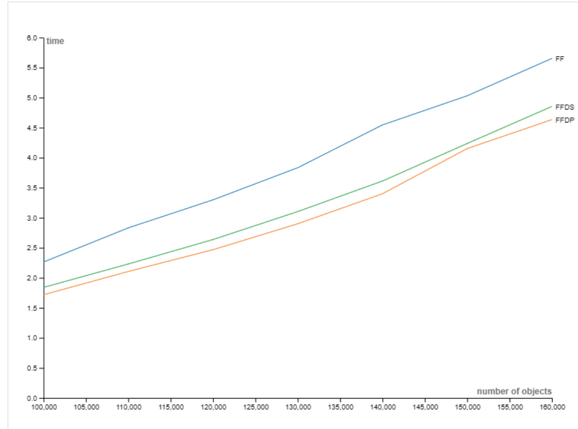


Fig. 5: A graph representing the Average time in seconds for each problem instance. Each instance has a different number of objects. The parallel FFD algorithm shows the best results.

type	used bins	total spaces	average waste	problem size
FF	50775	389331.76	7.66778454	100000
FFDS	50126	64831.76	1.293375893	100000
FFDP	50126	64831.76	1.293375893	100000
FF	56002	473080.16	8.447558301	110000
FFDS	55206	75080.16	1.36	110000
FFDP	55206	75080.16	1.36	110000
FF	60915	448929.99	7.369777395	120000
FFDS	60143	62929.99	1.046339391	120000
FFDP	60143	62929.99	1.046339391	120000
FF	65810	442565.92	6.724903814	130000
FFDS	64982	28565.92	0.439597427	130000
FFDP	64982	28565.92	0.439597427	130000
FF	71035	528659.9	7.442245372	140000
FFDS	70086	54159.9	0.772763462	140000
FFDP	70086	54159.9	0.772763462	140000
FF	76123	549749.39	7.221856601	150000
FFDS	75164	70249.39	0.934614842	150000
FFDP	75164	70249.39	0.934614842	150000
FF	81121	558161.48	6.880604036	160000
FFDS	80132	63661.48	0.794457645	160000
FFDP	80132	63661.48	0.794457645	160000

TABLE II: This table shows the utilization of the three algorithms for different problem sizes. The average waste is the total number of empty spaces divided by the number of used bins.

process. Moreover, the FFD algorithms use less bins than the FF algorithm. The average waste results show how the average bin is utilized and how much of it is wasted with empty space. Also, the FFD utilizes 99.84% of the used bins when the problem size is 60000. Meaning that only 0.16% of each bin is empty.

V. Conclusion and Future Work

In conclusion, this paper proposed a parallel approach to the FFD algorithm. We proved that just by parallelizing the sorting in the First Fit Decreasing algorithm, we can decrease the computation time by 4.73% without affecting the utilization rate. Future work may include parallelizing a two-dimensional bin packing algorithm. Furthermore, the proposed algorithm can be optimized to use less space, as the FFD algorithm saves all bins in memory.

REFERENCES

[1] J. Balogh, J. Békési, and G. Galambos, "New lower bounds for certain classes of bin packing algorithms," *Theoretical Computer Science*, vol. 440-441, pp. 1–13, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397512003611

[2] E. G. Coffman, M. R. Garey, and D. S. Johnson, "An application of bin-packing to multiprocessor scheduling," *SIAM J. Comput.*, vol. 7, pp. 1–17, 1978.

[3] G. Dósa and J. Sgall, "First Fit bin packing: A tight analysis," in *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), N. Portier and T. Wilke, Eds., vol. 20. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 538–549. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2013/3963

[4] L. Epstein and A. Levin, "Asymptotic fully polynomial approximation schemes for variants of open-end bin packing," *Information processing letters*, vol. 109, no. 1, pp. 32–37, 2008.

[5] E. Falkenauer, A. Delchambre *et al.*, "A genetic algorithm for bin packing and line balancing." in *ICRA*, 1992, pp. 1186–1192.

[6] M. R. Garey, R. L. Graham, and J. D. Ullman, "Worst-case analysis of memory allocation algorithms," in *Proceedings of the fourth annual ACM symposium on Theory of computing*, 1972, pp. 143–150.

[7] S. Ghosh and A. H. Gebremedhin, "Parallelization of bin packing on multicore systems," in *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*, 2016, pp. 311–320.

[8] D. S. Hochbaum, Ed., *Approximation Algorithms for NP-Hard Problems*. USA: PWS Publishing Co., 1996.

[9] D. S. Johnson, "Near-optimal bin packing algorithms," 1973.

[10] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," *SIAM Journal on computing*, vol. 3, no. 4, pp. 299–325, 1974.

[11] W. Leinberger, G. Karypis, and V. Kumar, "Multi-capacity bin packing algorithms with applications to job scheduling under multiple constraints," in *Proceedings of the 1999 International Conference on Parallel Processing*. IEEE, 1999, pp. 404–412.

[12] E. C. man Jr, M. Garey, and D. Johnson, "Approximation algorithms for bin packing: A survey," *Approximation algorithms for NP-hard problems*, pp. 46–93, 1996.

[13] A. Martinez-Sykora, R. Alvarez-Valdés, J. A. Bennell, R. Ruiz, and J. M. Tamarit, "Matheuristics for the irregular bin packing problem with free rotations," *European Journal of Operational Research*, vol. 258, no. 2, pp. 440–455, 2017.

[14] A. Ntanjana, "Two and three-dimensional bin packing problems: an efficient implementation of evolutionary algorithms," Ph.D. dissertation, 2018.

[15] D. Varsamis and F. Chanlioglou, "A parallel approach of best fit decreasing algorithm [j]," *Wseas Transactions on Computers*, vol. 17, no. 9, pp. 79–85, 2018.

[16] A. N. Zehmakan, "Bin packing problem: Two approximation algorithms," *arXiv preprint arXiv:1508.01376*, 2015.

4.4 Spring 2022

4.4.1 Human Activity Recognition in Maintenance Centers to Reduce Wasted Time

Omar Magdy Tawfik for publishing his scientific paper to the IEEE Xplore as part of the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference (MIUCC). The paper entitled “Human Activity Recognition in Maintenance Centers to Reduce Wasted Time" proposed a system that extracts workers’ poses from the live cam and video clips. This system aims to detect and classify the positive and negative activities of the worker in car maintenance centers such as (changing the tire, changing the oil, using the phone, and standing without work) by calculating the time of each activity to measure the worker’s performance and determining the lost time correctly and properly. The presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community.[9]

Human Activity Recognition in Maintenance Centers to Reduce Wasted Time

Omar Magdy
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
omar.magdy10@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers
and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781695

***Abstract*—This paper proposed a system that extracts workers' poses from live cam and video clips using mode classification. In this paper, we tested two algorithms to detect worker activity. This system aims to detect and classify the positive and negative activities of the worker in car maintenance centers such as (changing the tire, changing the oil, using the phone, standing without work) by calculating the time of each activity to measure the worker's performance and determining the lost time correctly and properly. Two experiments were conducted , the first experiment was conducted to measure the performance of the dollar algorithm with different participants. The results showed that the 1 dollar recognizer achieved 94.2% accuracy when tested on 364 different videos. The second experiment was conducted to measure the accuracy of the system in recognizing real-time activities from the live camera. It was conducted on 5 participants in a controlled environment. The system achieved an accuracy rate of 93.3%.**

***Index Terms*—Machine Learning, Human Activity Recognition, Pose classification, industry advancement**

I. Introduction

The importance of human activity recognition has been gained in recent years, because of its applications in various fields like health, security and surveillance, entertainment, and intelligent environments. Lots of research have been accounted in many different approaches, such as wearable devices.ref [1]–[5],object-tagged ref [6]–[8], and device-free ref [9]–[11], to acknowledge human activities. In different fields of work, the wasted time has become an outstanding problem, moreover, the percentage of products and services needed in human's daily lives are affected by this problem.

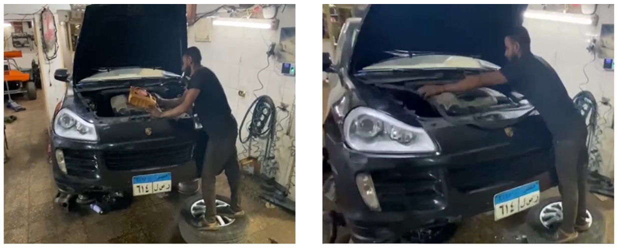
The survey that was provided by "Salary.com" [12] found that 89% of workers are exposed to wasting time at work every day. It was claimed by the survey that 61% waste between 30 minutes to an hour a day. It may not seem like much until it can add up to 5 hours a week or 260 hours a year/per employee. Maintenance places and factories are the most exposed to this issue. The survey has a huge dependency on human activity. In which the production rates are harmed, and many required services are disrupted.

Recently, in human's daily lives, the field of car maintenance and car workshops have spread extremely and became important, but difficulties have begun to appear inside the maintenance centers. The problem specifies that as a result of doing some negative activities by the workers, they waste a lot of time while they are working. A worker who uses the phone at work is illustrated in Figure 1. In addition to many other negative activities like eating, drinking, talking to others, standing without work, … etc. Moreover, having an employee who is responsible for monitoring the worker for no less than 12 hours is difficult.

Fig. 1. A negative wasting time activity (using mobile while being in work)

The maintenance workshop could have dozens of activities some of them are considered positive and other considered negative. However,some activities look very similar such as shown in Figure 2 (a) and Figure 2(b). The posture of changing oil activity looks very similar in motion trajectory to engine rebuild activity.

978-1-6654-6677-6/22/$31.00 ©2022 IEEE



a) Changing oil

b) Engine rebuild

Fig. 2. A sample for positive activity in workshop

The proposed system works by taking input by recording videos or a Live-stream camera of the worker's activities inside the maintenance center. The system has classified the input to determine the number of activities the worker performed such as (Changing the oil, changing the tire, using the phone... Etc.) by taking the path (x, y) of the important points in the skeleton as shown in Figure(3), Figure(4) and comparing them with the points that were taken from the collector of the data set. The system can determine the type of activities that the worker performed in the video or the Live-stream and calculated the wasted time by the worker.

Fig. 3. Worker changing oil

.

A. 1 dollar algorithm

Many papers have used the one-Dollar algorithm in recognition of gestures and achieved satisfying results, so as a result, this algorithm was used in this paper as mentioned in [13], moreover, high accuracy results have been achieved using this algorithm. A geometric template matcher called the One-Dollar operates in which previously saved templates (T) and candidate strokes (C) are both compared to each other. The result is observed in the match that is the nearest in the 2-D Euclidean space as referred to in Equation 1. Therefore, after having N points in the 2-D Euclidean space, it will be able to calculate the distance between C[k] to T[k] where k= 1 to N. Pairwise point comparisons most used in the one-dollar algorithm are scale, rotation, and position invariant. The one-Dollar algorithm can not find the difference between gestures whose identities rely on special orientations, ratios, and locations. The one-Dollar algorithm does not involve time use, hence, gestures can not be separated concerning the speed.

Fig. 4. Worker use mobile

$$d_i = \frac{\sum_{K=1}^{N} \sqrt{\left(C\left[K_x\right] - T_i\left[K_x\right]\right)^2 + \left(C\left[K_y\right] - T_i\left[K_y\right]\right)^2}}{N} \tag{1}$$

The main objective of this paper is to propose a human activity recognition system to help reduce the wasted time by the worker by extracting worker poses from the videos or Live-stream camera by using pose classification and it recognized the worker by face recognition. The system used a dollar recognizer (1$) to detect workers' positive and negative activities in the maintenance workshop.

II. Related work

A lot of research efforts were specified on this subject, due to the popularity of recognition of human activity systems, and the rapid development of computer technology. Congcong Liu et al . [6],has raised an activity recognition method that can identify abnormal human activity recognition in surveillance video by using a combination of Bayes Classifier and CNN (Convolutional Neural Network) to detect the activities. Moreover, by the use of the KTH dataset for the input of Bayes Classifier and CNN. Bagate et al. [1], proposed another system that is able to recognize human activity, the activity is identified by using RGB-D sensors that are advanced with the deep learning model CNN (Convolutional Neural Network), as well as using a Knight Depth camera for capturing 3-D Skeleton data. Frame-wise displacement and recognition were used by Sandar et al. [8], for Human Detection and Motion Tracking, which is based on the skeletal model with the deep learning framework in order to understand human behavior in the indoor, as well as outdoor environment.

Regarding the field of Human Activity Recognition by sensors, Murat et al. [14], by using joint coordinates, human activity was automatically identified, used 2 types of deep learning to make classification, and multiple people in images were used as a data set. (1D) Convolution Neural Network (CNN) -based method was proposed by Song et al. [5], for recognizing the activities by using the collected accelerometer data from smartphones, the previous method gave high accuracy of 92.71%. eSense accelerometer was used by Tahera et al. [2], for the wearable devices to detect the matching of activity between the head and the mouth, some activities of the head and mouth were identified from these collected data, and for data classification, the machine learning and deep learning were used. TCN (temporal Convolutional Network) was proposed to be used by Nitin et al. [15], in order to recognize the activities, because it is better than other deep learning models, as it is strongly able to capture long-term dependencies.

Gyroscope sensors with accelerometers were combined through Godwin et al. [9], for detecting human activity and performing analysis and recognition by using ANN (Artificial Neural Networks). In order to make recognition of human activity more accurate, Tsokov et al. [16], used the 1D Synaptic Neural Network (CNN) with accelerometer data. Hip motion from the different waist-mounted sensors was collected by Isah et al. [3]. then each signal was converted into a spectrum image, and finally, used as input to the CNN (Convolutional Neural Network). The entropy point estimate for the 1D heat map was used by Nacer et al. [17], as to separate between human maps and animal maps in order to give high accuracy in human activity recognition.

Selc¸uk et al. [10] claimed to use a novel design to decrease the number of sensors used for recognition and detection of human activity by using Empirical Mode Decomposition (EMD), Jiewen et al. [4], identified and interacted by focusing on two wearable cameras and the interactive activities that involve only two people. The topic of identifying human activity in the treatment of autism, and movements were grouped with a handheld camera, and the classifier CNN (Convolution Neural Network) was used for detecting headbanging in home videos, the previous work was addressed by Peter Washington et al. [11], The method in which classifies human activity by using 3D skeleton data but normalizing it before and it was represented in 2 forms that were proposed by Hristov et al. [18]The previous method was applied to the UTDMHAD dataset, and the system recorded a 92.4% accuracy rate. The idea of wearing devices and sensors to determine human activities was opposed to Heilym et al. [19]it was indicated that inconvenience to the bearer could be caused by these devices, so if it is used in crowed places it could give false results, as a result, he depended on the camera and on determining the activity through the human skeleton features.

An objective was proposed by Salahuddin Saddar et al. [20]in which it compares some machine learning algorithms that have been used before in human activity recognition (SVM, Decision Trees, Random Forests, XGBoost). The previous algorithms were tested with measurement sensor data. The sensor data was recently released from the LARA dataset. The best accuracy was achieved by the XGBoost with a rate of 78.6%. The topic of identifying human activity in terms of reducing aggressive actions inside prisons and on the streets to reduce aggression was taken by Ismael et al. [21] took the topic of identifying human activity in terms of reducing aggressive actions inside prisons and on the streets to reduce aggression and used ”handcrafted/learned” as a hybrid feature framework that gave it very high accuracy rates.

Yusuf Erkan et al. [22]to classify 27 various activities, and long-short term memory was used as well. They analyzed skeleton data, and it has achieved an accuracy rate of 93%. A system for monitoring activities inside the factory was presented by Halikowski et al . [7], the system used (CNN, CNN+SVM, Yolov3) algorithms. Moreover, some activities were used as the furnace operating stop, the solid fuel tank check, the gear motor and auger check, the mounting screws of the gear motor tight, and an accuracy of 94% was achieved. The work presented the use of deep learning for feature extraction without taking into consideration human post estimation.

A system for detecting activities in smart manufacturing was presented by Zhaozheng et al. [23] some activities were used such as grabbing tools, hammer nailing, wrench using, rest lever, and screwdriver. By using IMU and sEMG signals that are obtained from an MYO armband, they were able to capture these activities. Convolutional Neural Network (CNN) model was used for extracting features. An evaluation of the CNN model on this dataset is held and achieves recognition accuracy of 98% &, and 87% in the half-half and leave-one-out experiments. The importance of identifying human activity in solving some problems in different fields is explained in all of the previous.

A system that detects different actions in the streets like walking, running, and stopping was proposed by Alghyaline et al. [24] The movement type is measured by the usage of three different techniques which are (Yolo, Kalman filter, and Homography). CCTV camera and BEHAVE dataset was used to test this method, an accuracy of 96.9% was achieved for the BEHAVE dataset, and an accuracy of 88.4% was achieved for the dataset that was collected by CCTV camera.

Table 1 shows a comparison between our system and different systems is shown in Table 1. The first system, this system was proposed by Halikowski et al. [7] in order to measure the worker's performance in the factory. In a controlled area, 4 different activities were recognized which are making the furnace operating stop, checking the solid fuel tank, checking the gear motor and auger, and tightening the mounting screws of the gear motor by using an image classification method. More than one algorithm has been used (CNN, CNN+SVM, Yolov3), and an accuracy of 95.7% was achieved by their system. Zhaozheng et al. [23] proposed the second system that was used for workers' qualifications and evaluation. Image classification was used to detect 4 activities which are grabbing tools, hammer nailing, wrench using, rest lever, and screwdriver in a controlled area. An accuracy rate of 87% was achieved. Our system was proposed in order to detect the worker's activities inside the car maintenance center and to differentiate between the negative and positive activities. The pose classification method was used in the system to detect four different activities which are changing oil, changing tires, using mobile, and standing without working in order to help the workshop owner in measuring the worker performance. The system was used in an uncontrolled area and on different body characteristics such as height and weight.

III. Proposed system

This paper presented a method that recognized and classified human activity in car workshops performed and captured from videos or live-stream cameras. It differentiated between the positive activities and the negative activities based on a comparison between the input and dataset stored in the templates. The proposed system used mediapipe for collecting key points of the skeletal joints and used the one-dollar and Fastdtw algorithms to classify the poses it used face recognition to differentiate between the workers and calculated the time of each activity to determine the wasted time by workers.

The dataset was collected while visiting a real car workshop by recording videos of real workers and specialists in the activities of the worker inside the maintenance workshop. It contains 560 videos including the 4 activities which are (changing tires, changing oil, using a mobile, and standing without work). These videos were entered into the media pipe and OpenCV to extract the human pose from the videos showing the path of each point during the video. Each list of points has been saved in a file containing the points and the name of the activity.

Figure 6 shows the system overview, the system has two different way to input which is videos or live camera. The processing part starts with face recognition to differentiate between the workers because there is a large number of workers in the maintenance center. Then the mediapipe starts to extract the poses by calculating the path of each point in the skeleton. The mediapipe can extract 32 points, but this system focuses on extracting 5 important points which are (shoulder, elbow, wrist, hip, knee) as shown in Figure 5. The algorithms start to match the input points with the points stored in the data set and send the results with calculated the time of each activity and the name of each worker to the database to create a report that the managers and workshop owners can see and help them to measure the performance of the workers.

Fig. 5. Pose extraction

IV. Experiment

A. Experiment 1

The objective of conducting this experiment was to measure the performance of the one-dollar algorithm, but with increasing the number of participants. Thirteen participants were asked to do a sequence of activities in different stream videos. The scenario in which the activities were held were changing tires, using a mobile, changing oil, and standing without work, it was changeable from one to another. 30 seconds was the average duration of the activity in the video. Each activity is repeated in each video 7 times, 4 from the right side, and 3 from the left side. The average range of the participant's ages was from 19 to 22 years. The characteristics of the body were different as well. Then, the videos were inserted into the system, and 364 videos were extracted by the system for a range of different activities in uncontrolled environments. We were able to measure the accuracy of the system in identifying activities in different conditions through this experiment. The system was able to extract 364 videos in this experiment, with a huge number of participants, experts in the field of mechanics, and workers from maintenance centers. Moreover, the one-dollar algorithm initiated the stage of identifying activities and an accuracy rate of 94.2% was achieved.

TABLE I
COMPARISON WITH SIMILAR SYSTEMS

paper	Activities	Algorithms	Methods	Area	Accuracy
[7]	stopping the furnace operating checking the solid fuel tank checking the gear motor tightening the mounting screws of the gear motor	CNN CNN+SVM Yolov3	Image classification	controlled area	95.7%
[22]	grab tools hammers nail wrench use screwdriver	CNN	Image classification	controlled area	87%
This paper	Changing tire changing oil using mobile stand without work	1$	Pose classification	uncontrolled area	94.2%

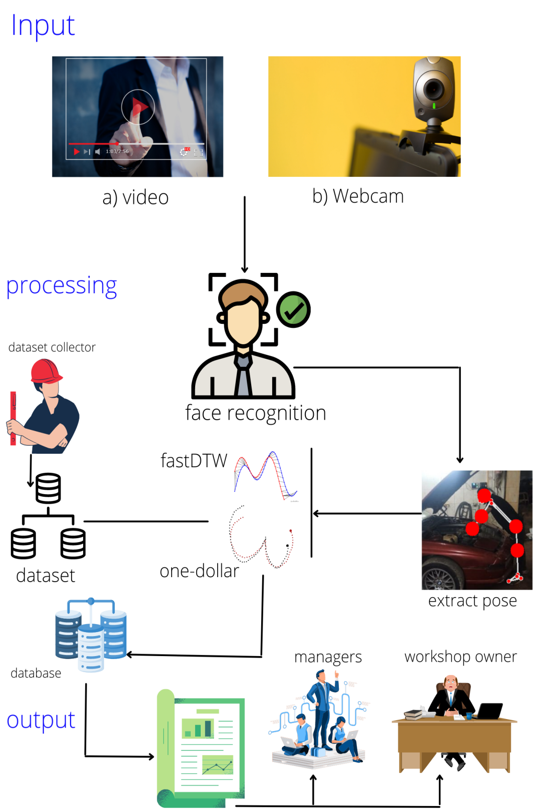


Fig. 6. System overview

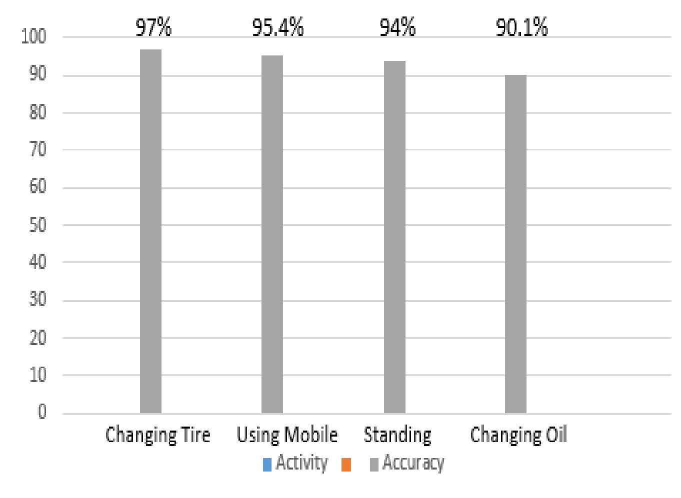


Fig. 7. The results of Experiment 1

B. Experiment 2

This experiment aims to test the accuracy of the system in giving immediate results by using it on live broadcast cameras. This experiment was conducted in a controlled environment on five participants with different physical characteristics in terms of height and weight. We set up a webcam at a testing point which is placed at a constant distance from the webcam to ensure consistency of our results.We asked each participant to make a sequence of activities, this sequence lasts for 3 minutes. Each activity they did was from the right and left sides. Each activity in the sequence contains a range of 26 to 30 activities. This sequence combined the activities mentioned in this system and other activities that have not been mentioned. The scenario of this sequence was changing tires, using mobile, changing oil, and then standing without work. The previous scenario was combined with some activities that were not mentioned in the system, such as sitting, eating, and sleeping. Moreover, the sequence varies from one person to another in order to see the accuracy of the system in real-time identification of activities.

In this experiment, after performing a set of sequences of activities, the participants were able to perform an average of 26 different activities, and the system was able to identify an average of 22 of them correctly, and an average false detection of 4 activities. The system achieved an accuracy rate 93.3%.

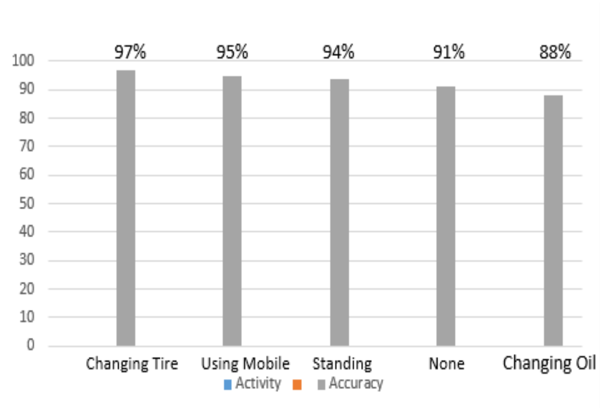


Fig. 8. The results of Experiment 2

C. Discussions

The first experiment was to test the performance of the one-dollar algorithm in determining a large number of participants' activities. There was a significant difference in accuracy ratios between the activities as the results have shown. The activity that was subtle as it differed from the other activities in the motion trajectory was changing the tire, the highest accuracy was given to the activity of using mobile and standing without work, however, there were few similarities between them. A similar motion trajectory was between the oil change activity, usage of mobile, and standing without work, as a result, its accuracy was reduced as shown in Figure 7. The second experiment was testing the accuracy of the system in detecting the activities in real-time by using the live camera. The reason for these results is that there are also some similarities between the different activities, as happened in the second experiment, carrying out some activities that are not proposed in the system such as eating, sleeping, and sitting as shown in Figure 8.

V. Conclusion and Future Work

In this paper, a human activity recognition system in a car workshop was proposed to help maintenance center owners identify positive or negative worker activities and measure worker performance correctly and efficiently. Our system used a live camera and recorded videos as input and used pose classification to extract workers' poses and used two different algorithms to detect the activities and also it recognized the face of the worker to differentiate between them and calculated the wasted time of each worker. The system achieved an accuracy rate of 94.2% using this method. However, we are confident that the accuracy of this system will improve with more test videos and more tests on a live camera. Our future work will focus on solving problems such as obstacles (hood) that arise while defining activities and affecting system accuracy. As well as increasing the data set to reduce the problem of similarity between activities. Besides, we will increase the activities in which the experiments were conducted in the future with more than one algorithm being tested and compare it with the one-dollar algorithm to increase the accuracy.

References

[1] A. Bagate and M. Shah, "Human activity recognition using rgb-d sensors," in *2019 international conference on intelligent computing and control systems (ICCS)*. IEEE, 2019, pp. 902–905.

[2] T. Hossain, M. S. Islam, M. A. R. Ahad, and S. Inoue, "Human activity recognition using earable device," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 81–84.

[3] I. A. Lawal and S. Bano, "Deep human activity recognition using wearable sensors," in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2019, pp. 45–48.

[4] J. Zhao, R. Han, Y. Gan, L. Wan, W. Feng, and S. Wang, "Human identification and interaction detection in cross-view multi-person videos with wearable cameras," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2608–2616.

[5] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 ieee international conference on big data and smart computing (bigcomp)*. IEEE, 2017, pp. 131–134.

[6] C. Liu, J. Ying, F. Han, and M. Ruan, "Abnormal human activity recognition using bayes classifier and convolutional neural network," in *2018 IEEE 3rd international conference on signal and image processing (ICSIP)*. IEEE, 2018, pp. 33–37.

[7] J. Patalas-Maliszewska, D. Halikowski, and R. Damaševičius, "An automated recognition of work activity in industrial manufacturing using convolutional neural networks," *Electronics*, vol. 10, no. 23, p. 2946, 2021.

[8] S. Win and T. L. L. Thein, "Real-time human motion detection, tracking and activity recognition with skeletal model," in *2020 IEEE Conference on Computer Applications (ICCA)*. IEEE, 2020, pp. 1–5.

[9] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proceedings of the 2018 10th international conference on machine learning and computing*, 2018, pp. 41–46.

[10] S. Sezer and E. Surer, "Information augmentation for human activity recognition and fall detection using empirical mode decomposition on smartphone data," in *Proceedings of the 6th International Conference on Movement and Computing*, 2019, pp. 1–8.

[11] P. Washington, A. Kline, O. C. Mutlu, E. Leblanc, C. Hou, N. Stockham, K. Paskov, B. Chrisman, and D. Wall, "Activity recognition with moving cameras and few training examples: Applications for detection of autism-related headbanging," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[12] Z. Hussain, Q. Sheng, and W. E. Zhang, "Different approaches for human activity recognition: A survey," 06 2019.

[13] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, 2007, pp. 159–168.

[14] M. Yalçın, N. Tüfek, and H. Yalcin, "Activity recognition of interacting people," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[15] N. Nair, C. Thomas, and D. B. Jayagopi, "Human activity recognition using temporal convolutional network," in *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*, 2018, pp. 1–8.

[16] S. Tsokov, M. Lazarova, and A. Aleksieva-Petrova, "Evolving 1d convolutional neural networks for human activity recognition," in *International Conference on Computer Systems and Technologies' 21*, 2021, pp. 49–54.

[17] A. Naser, A. Lotfi, J. Zhong, and J. He, "Human activity of daily living recognition in presence of an animal pet using thermal sensor array," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–6.

[18] P. Hristov, A. Manolova, and O. Boumbarov, "Deep learning and svm-based method for human activity recognition with skeleton data," in *2020 28th National Conference with International Participation (TELECOM)*. IEEE, 2020, pp. 49–52.

[19] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall detection and activity recognition using human skeleton features," *IEEE Access*, vol. 9, pp. 33 532–33 542, 2021.

[20] A. S. Syed, Z. S. Syed, M. S. Shah, and S. Saddar, "Using wearable sensors for human activity recognition in logistics: A comparison of different feature sets and machine learning algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110976

[21] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight recognition in video using hough forests and 2d convolutional neural network," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4787–4797, 2018.

[22] Y. E. Görgülü and K. Taşdelen, "Human activity recognition and temporal action localization based on depth sensor skeletal data," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 1–5.

[23] W. Tao, Z.-H. Lai, M. C. Leu, and Z. Yin, "Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks," *Procedia Manufacturing*, vol. 26, pp. 1159–1166, 2018.

[24] S. Alghyaline, "A real-time street actions detection," *International Journal of Advanced Computer Science and Applications*, vol. 10, 01 2019.

4.4.2 Exploring Brain Tumor Classification Using Deep Learning

Habiba Mohamed Mousa on publishing a Conference paper on his graduation project, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The paper entitled “Exploring Brain Tumor Classification Using Deep Learning " proposed a system that build a reliable means and appropriate method for classifying human brain cancers that uses magnetic resonance imaging (MRI) to distinguish between the many forms of Glioblastoma, malignant tumors, and gland tumors. [12]

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781767

Exploring Brain Tumor Classification Using Deep Learning

1st Habiba Mohamed
Faculty of computer Science
October university for Modern Sciences and Arts(MSA)
Cairo, Egypt
habiba.mohamed10@msa.edu.eg

2nd Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of computer Science
October university for Modern Sciences and Arts(MSA)
Cairo, Egypt
aezzat@msa.edu.eg

***Abstract*—Diagnosis at a beginning period and recognition of the type of cancer can assist doctors and health experts in determining the most appropriate treatment. The target of this is to research is to build a reliable means and appropriate method for classifying human brain cancers that uses magnetic resonance imaging (MRI) to distinguish between the many forms of Glioblastoma, malignant tumors, and gland tumours are examples of brain tumors. In order to enhance and achieve accurate results can make preprocessing methods like resize MR images, cropping and data augmentation to avoid over fitting. By using deep learning pre-defined models as ResNet, VGG16, MobileNet and Inception. And transfer-based learning CNN that supported with calculation of dice, sensitivity and specificity we founded that by using dice with CNN model the achieved accuracy was 99.9%.**

***Index Terms*—Brain Tumor,Data Augmentation , Deep Learning, CNN,**

I. Introduction

A brain tumour is defined as the uncontrolled expansion of neural cells. Non-cancerous or malignant cells can form brain tumors. cancerous cells are categorized into two types [17] the first one is primary which means that tumor start in brain, The second one is Secondary which means that tumor start in elsewhere in body then move to brain. The potential negative effects of a brain tumour differ widely based on the tumor's shape, position, and pattern of increase. The most common indications are headaches, Problems with walking or balance, Problems with memory or thinking, feeling lazy or sleepy, mood swings or changes in your attitude, alterations in your ability to effectively communicate, hear, or see, Paroxysmal which means Seizures and Slightly out of focus vision, double vision, or a lack of mind's eye are all examples of visual issues [15]. Their exist several types for brain tumor First, "Gliomas" Is cancer that start either the brainstem or the central nervous system. Second, Tumors that grow in the membranes that surround the brain and spinal cord are known as "meningiomas' (meninges). Third ,The "pituitary" gland is a little shaped organ region found at the center of your brain, between your ears and slightly behind your nose. The gland, although its modest shape, has an impact on practically every aspect of your brain. These are the most frequent brain tumor types that was proved by statistics are Meningiomas, Gliomas and Pituitary [16]

Over the last 30 years, the United nations stated according to the Brain Tumor Organization (NBTF), death rates in various countries have increased by 300%. [19]. An approximated 83,570 people will be successfully treated with brain tumors in the Us in 2021, with 18,600 dying from the disease [21]. To collect information regarding tumours (tumor kind, form, volume, placement, and so on) that is essential for their diagnosis, a variety of medical imaging techniques are used PET, CT scan Tomography, MRS, and MRI are all imaging techniques.

These methods can be utilized to provide more specific information about tumors. MRI, on the other hand, is the most widely utilised method because of its numerous advantages. [21]. Deep learning has recently been used to classify a variety of diseases and cancers.Nyoman Abiwinanda et al. [22] created a CNN that can automatically classify the three most common forms of BTRs, namely Glioma, Meningioma, and Pituitary, without the need for any training data.for a request for pre-processing depending on geography They made use of a BTR.

Quick diagnosis of brain tumours is important for successful treatment planning and healthcare. Manually classifying items with comparable features is a time-consuming process. The main idea underlying transfer learning is to speed up the learning of by moving data from a previously learned or trained model to a new task. Many scientists have used AI algorithms and Deep Learning find out the ways to make more progress, but tracking or identifying accuracy could still be improved.

This paper tries to enhance classifying of brain tumor types. The paper try to explore different types of deep learning algorithms and transfer learning over dataset in order to reduce miss identification of brain tumor types. Deep Learning strategy for brain tumor classification that uses pre-trained models.(VGG16, Inception, ResNet, and MobileNet) and CNN with transfer learning on a bench marking dataset for the types of brain tumor was explored.

II. Related Work

There is a need for a quick and reliable method to determine whether the tumour present is cancerous (LGG orHGG). For the classification of tumorous images into LGG or HGG, a two-stage deep learning architecture is proposed and they achieved accuracy 92.54%. [5]

To address this challenge, The MRI picture must first be kept at the image's boundary by median filtering to distinguish brain tumor tissues. The following phase in the tumour segmentation process is to use a median filter, which is repeated until the greatest region is obtained with 90% accuracy. [1]

According to recent research, deep learning methods outperform machine learning methods in image classification tasks and provide higher accuracy.Deep learning is a subtype of machine learning that eliminates the need for manual feature extraction, providing it an advantage over other methods. Using fully connected neural networks and Convolution Neural Networks, Paul et al. developed a generalised method for brain tumor classification that achieved 91.43 100 % accuracy. (CNN). [2].

The reference dataset utilised by the majority of the researchers is Brats 2015. They next compare multiple CNN-based transfer learning models for brain tumour classification in MRI images, as well as extract characteristics.It demonstrates how deep learning algorithms may be used to detect brain malignancies from MRI images of the brain and classif them utilising a number of approaches for brain diagnosis en hancement, segmentation, and classification, including medica image processing, pattern analysis, and computer vision. [3]

Establish a robust and efficient method for classifyin brain tumours using MRI that is based on the transfer learn ing technique. Deep features from brain MRI require pre trained models like Xception, NasNet Large, DenseNet121 and InceptionResNetV2. For accurate and fast training, th dataset was first cropped, preprocessed, and augmented. Thre different optimization algorithms were used on an MRI datase (ADAM, SGD, and RMSprop). According to the experimenta results, the CNN model discovered that the ADAM optimize outperforms the other three proposed models. [4]

A CNN able to detect and identifying whether or not patient has BTR. The suggested technique The model accurac is compared to a pre-implemented CNN model, — in othe words, layer combinations Convolutional layer 2D and ReLU by combining the ReLU activation function with a convolutio 2D layer Convolutional layer 2D and ReLU With 78.57 % of the voting, the suggested model was approved. [6] Fo segmentation techniques such as Authors utilize architecture threshold, and a simple constrained container. Transfer learn ing is used to classify AlexNet and VGG-19, two pre-traine convolutional neural networks. [7]

Polly et.al [8] K-means clustering, DWT, and SVM were combined to create a classifier. The accuracy was found to be 99 %. The system was not designed for large data sets, despite its high accuracy.

Kerala [9] discusses a robust deep learning method for predicting brain tumours from MRI images that employs a convolution neural network. The model, which has been evaluated on two datasets, shows good performance metrics on both, making it acceptable for therapeutic diagnostics. In the study, two datasets with 99 % and 95 % accuracy were used.

III. Methodology

In our approach, the input dataset is first divided into training and testing as shown in figure 1. The images are then subjected to preprocessing operations, the pre-processing images are scaled.

Transfer learning is used to create the CNN architecture that are composed of 6 layers ,the model was created using an Adam optimizer and a loss function Categorical Crossentropy and Binary-Crossentropy. (input, conv2d(128), maxpooling2d, conv2d(64), maxpooling2d, fatten, dense(32), dense) is the model.The model has a total of 7,533,956 trainable parameters, with a batch size of 32., and the hyperparameters are modified to achieve the desired output. The final model is obtained once the network has been trained using the training and testing data. This model is used to predict whether a subject has a tumour or not and which type of tumor if it exists.

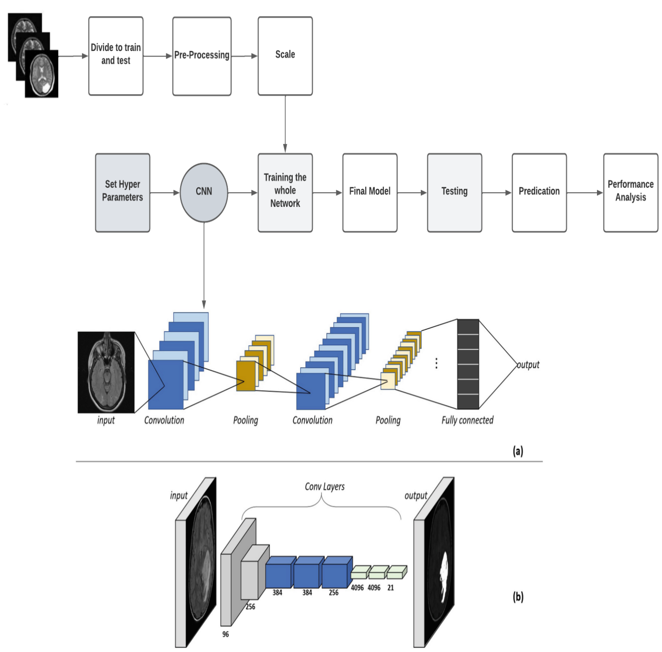


Fig. 1. System overview

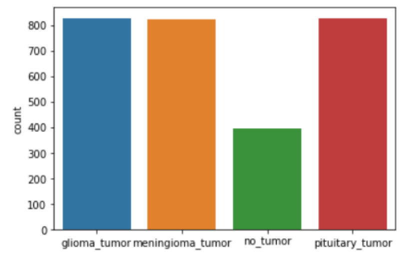


Fig. 2. Number of images per each class

A. *Dataset*

We have obtained dataset from Kaggle (Brats 2018) [12] composed of MR images of 3264 files all images have the same size and they classified into testing and training. Each one classified into 4 classes Meningioma, Glioma, Pituitary and No-Tumor. In the file training, first class (Meningioma) has 822 images, the second class (Glioma) has 826 images, the third class (Pituitary) has 827 images and last class (no tumor) has 395 images. On the other hand, in the testing file, first class (Meningioma) has 115 images, the second class (Glioma) has 100 images, the third class (Pituitary) has 74 images and last class (no tumor) has 105 images. Figure 2 shows the distribution of samples per each class.

B. *Models Approaches*

A. CNN

The CNN model is a type of convolutional neural network that is used to learn, CNN has four main layers convolutional layer, pooling layer, RelU and Fully connected layer. Our model is made up of 6 levels. Using the architecture to create graphics with a size of (200*200). The model was created using an Adam optimizer and a loss function Categorical Crossentropy and Binary-Crossentropy. (input, conv2d(128), maxpooling2d, conv2d(64), maxpooling2d, flatten, dense(32), dense) is the model.The model has a total of 7,533,956 trainable parameters, with a batch size of 32.

B. Inception v3

Inception Modules are utilised in Convolutional Neural Net works to enable more efficient computation and deeper net works by lowering dimensionality with stacked 11 convolu tions. The modules were intended to solve a variety of dif ficulties, including computational expense and overfitting. I summary, rather than stacking them sequentially, the approac is to use multiple kernel filter sizes within the CNN and orde them to work on the same level.

C. VGG16

The VGG-16 model is a straightforward CNN architecture with only a few hyper parameters. This model utilizes a 3 x 3 filter size throughout the design, In addition, there is one step in the convolution layer and two steps in the pooling layer, both with the same spacing. It's known as VGG-16 since it has 16 layers, including a softmax convolution layer.

D. Mobile-Net

Mobilenet is a model that filters images using the same convolution as CNN, but in a different method than CNN before it. It uses the depth and point convolution concepts, which are different from the usual convolution used by traditional CNNs.

E. ResNet

A sort of artificial neural network known as ResNet is a type of artificial neural network (ANN). Residual neural networks use skip connections, also known as shortcuts, to leap over some layers. Double- or triple-layer skips with nonlinearities (ReLU) and average pooling are common in ResNet models.

IV. Evaluation and Experiments

In order to evaluate the performance of each algorithm on the dataset we have conducted two experiments. The first experiment aim was to compare the accuracy of different classifiers on the datasets. In the second experiment, we explored different parameters for the CNN hyper parameters in order to achieve highest accuracy.

A. *Experiment1*

Table V show the accuracy of different algorithms. The results obtained after training and validating the proposed pre trained models (Inception , VGG16, Mobile-Net, ResNet) and the basic preimplemented CNN model.

TABLE I
Results of the pre-trained models

Result	Accuracy
CNN	96%
Inception	80%
VGG16	78.1%
Mobile-Net	70%
ResNet	68%

We have trained our models with 40 epochs and to make sure there is no over-fitting we plotted the model loss for training versus validation loss. Figure 3 shows the validations versus training accuracy for the trained model.

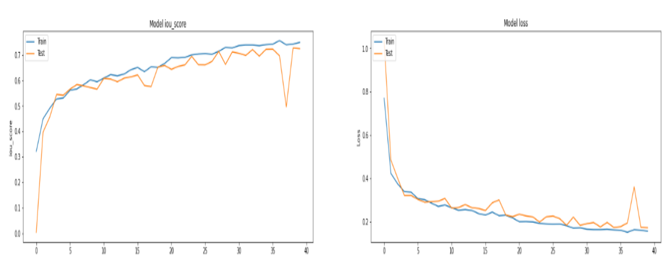


Fig. 3. Resnet training, validation accuracy and loss

Figure 4 shows the validations versus training accuracy for the Inception model.

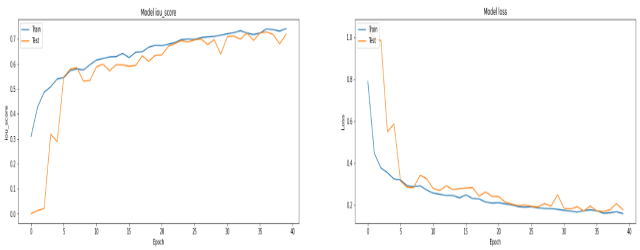


Fig. 4. Inception training, validation accuracy and loss

Figure 5 shows the validations versus training accuracy for the MobileNet model.

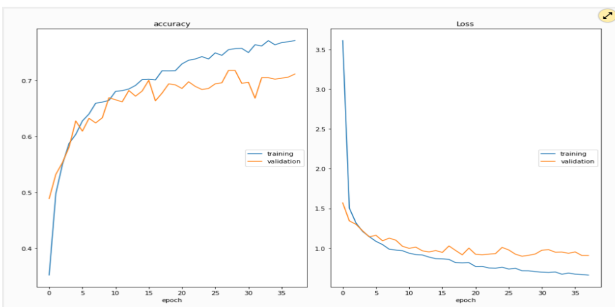


Fig. 5. MobileNet training, validation accuracy and loss

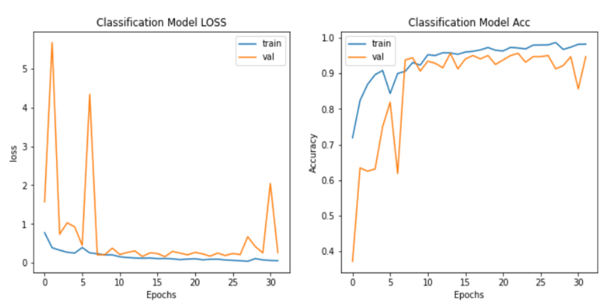


Fig. 6. VGG16 training, validation accuracy and loss

B. *Experiment 2*

The main objective of this study is to examine the the CNN model with different kernel sizes 5x5 and kernel size 7x7. In addition to different batch sizes 16,32 and 64. The model was also tested with different optimizes (ADAM, SGD, RMSProp, Adadelta). We have tested our results in terms of Accuracy, Dice, Sensitivity and Specificity.

2. Batch size 16, 32 or 64

TABLE II
DIFFERENT KERNEL SIZE

Kernel Size	5x5	7x7
Accuracy	94.6%	95%

TABLE III
DIFFERENT BATCH SIZE

Batch Size	16	32	64
Accuracy	96.3%	96.34	96.9%

3. Testing models using optimizers Adam, SGD, RMSProp, Adadelta

TABLE IV
DIFFERENT BATCH SIZE

Optimizers	Adam	SGD	RMSProp	Adadelta
Accuracy	96.3%	96.34	96.9%	96.9%

4. The model is also compiled with calculation of dice, sensitivity and specificity with optimizer SDG

$$Dice = \frac{2 * \sum(TP * Ppred)}{\sum(TP)^2 + \sum(Ppred)^2 + \in} \tag{1}$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{FP + TN} \tag{3}$$

A true positive (TP)is when the model predicts the positive class properly. A real negative(TN), on the other hand, is a result in which the model correctly predicts the negative class. A false positive(FP) occurs when the model forecasts the positive class inaccurately. A false negative(FN)is an outcome in which the model forecasts the negative class inaccurately and $\in\ isatinyconstantthatpreventsdivisionbyzero$. The Dice score is a statistic that assesses how closely ground truth labels and predictions overlap, with a higher number indicating better performance and for the correct classification of tumor types, sensitivity and specificity were assessed.

TABLE V
MEASUREMENTS

Accuracy	Dice	Sensitivity	specificity
Result	99.9%	99.11%	99.8%

After making a these comparison founded that the best model with Kernels size 7x7, Batch size = 32, Number of epochs = 30 and Gradient Descent Optimizer: SGD, then visualize the model with these methods.

After that, we make predictions and classification report for the validation of dataset. Finally, Confusion matrix for the conducted model

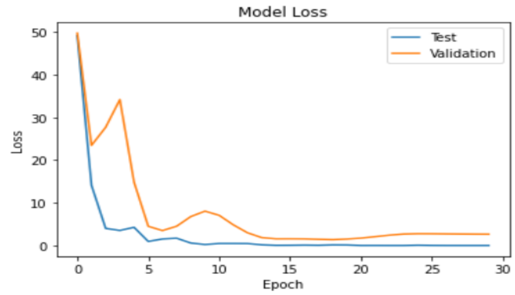


Fig. 7. CNN training, validation accuracy and loss

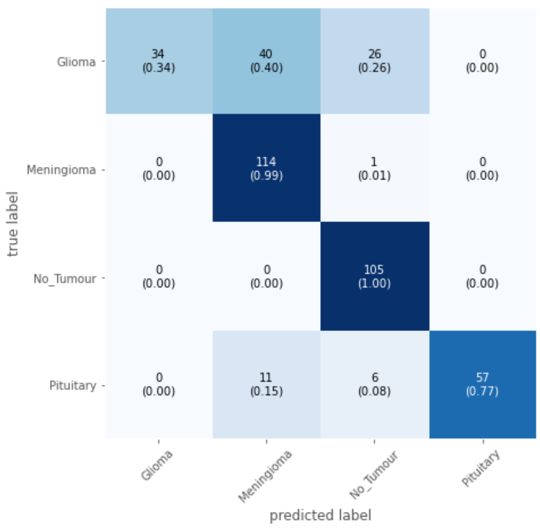


Fig. 8. Confusion Matrix For CNN Model.

After the confusion matrix we found that there exist a conflict between classes in the tumor types (meningioma Tumor,pituitary tumor and no tumor classes). Figure 9 shows samples of the dataset for different classes causing confusion as they looks close to each other.

Conclusion

We suggest in this study a technique for classifying brain MRI into four separate classes that can be utilized successfully. Pre-processing, processing, and classification of MRI scans are the three key stages of the model. Using models that have already been trained This technique applies the learning accomplished by the MRI images categorization using a pre-trained algorithm for natural image classification. This model can be used to detect sickness in other parts of the body in the future. We founded that by using dice with CNN model the achieved accuracy was 99.9%.

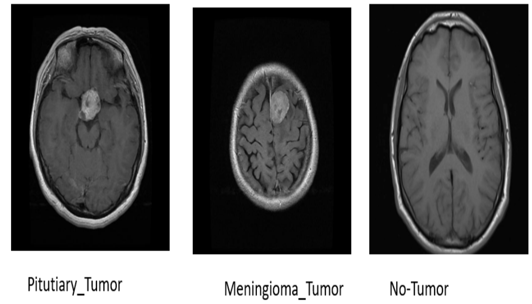


Fig. 9. Sample from tumor classes

References

[1] Arora, Shaveta & Sharma, Meghna. (2021). Deep Learning for Brain Tumor Classification from MRI Images. 409-412. 10.1109/ICIIP53038.2021.9702609.

[2] Paul, Justin & Plassard, Andrew & Landman, Bennett & Fabbri, Daniel. (2017). Deep learning for brain tumor classification. 1013710. 10.1117/12.2254195.

[3] Arora, Shaveta & Sharma, Meghna. (2021). Deep Learning for Brain Tumor Classification from MRI Images. 409-412. 10.1109/ICIIP53038.2021.9702609.

[4] Polat, Ozlem & Güngen, Cahfer. (2021). Classification of brain tumors from MR images using deep transfer learning. The Journal of Supercomputing. 77. 1-17. 10.1007/s11227-020-03572-9.

[5] Chandra, Gaurav & Shambhavi, & Paritosh, Pushp & Saha, Sriparna. (2021). Segmentation and Classification of Brain Tumors using Deep Learning Techniques. 1-7. 10.1109/INDICON52576.2021.9691513.

[6] Maghiar, Catalina & Chifu, Viorica & Pop, Cristina Bianca & Chifu, Emil. (2022). Brain Tumors Detection Using Convolutional Neural Networks. 10.1007/978-3-030-93564-1_24.

[7] Gull, Sahar & Akbar, Shahzad & Shoukat, Ijaz. (2021). A Deep Transfer Learning Approach for Automated Detection of Brain Tumor Through Magnetic Resonance Imaging. 1-6. 10.1109/ICIC53490.2021.9692967.

[8] Kabir, M.A., 2020. AUTOMATIC BRAIN TUMOR DETECTION AND FEATURE EXTRACTION FROM MRIIMAGE. GSJ, 8(4).

[9] MK, Husna & L, Mredhula. (2021). Prediction of Brain Tumor on MRI Images Using Deeplearning. 1-6. 10.1109/ICMSS53060.2021.9673656.

[10] Abiwinanda N., Hanif M., Hesaputra S.T., Handayani A., Mengko T.R. (2019) Brain Tumor Classification Using Convolutional Neural Network. In: Lhotska L., Sukupova L., Lacković I., Ibbott G. (eds) World Congress on Medical Physics and Biomedical Engineering 2018. IFMBE Proceedings, vol 68/1. Springer, Singapore. https://doi.org/10.1007/978-981-10-9035-6_33

[11] Kumar, Gulshan & Kumar, Puneet & Kumar, Deepika. (2021). Brain Tumor Detection Using Convolutional Neural Network. 1-6. 10.1109/ICMNWC52512.2021.9688460.

[12] https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri?select=Testing

[13] Anaya-Isaza, Andres & Mera Jiménez, Leonel. (2022). Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging. IEEE Access. 10. 1-1.

10.1109/ACCESS.2022.3154061.

[14] Anaya-Isaza, Andres & Mera Jiménez, Leonel. (2022). Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3154061.

[15] Kaur, Ashmeet & Singh, Shailendra. (2021). Diagnosis of Brain Tumor using Automated Deep Learning Model. 1-9. 10.1109/ICRITO51393.2021.9596536.

[16] https://my.clevelandclinic.org/health/diseases/6149-brain-cancer-brain-tumor#: :text=Mutations

[17] Amin, Javaria & Sharif, Muhammad & Haldorai, Anandakumar & Mussarat, Yasmin & Nayak, Ramesh. (2021). Brain tumor detection and classification using machine learning: a comprehensive survey. Complex & Intelligent Systems. 10.1007/s40747-021-00563-y.

[18] RAI, HARI & Chatterjee, Kalyan & Gupta, Apita & Dubey, Alok. (2020). A Novel Deep CNN Model for Classification of Brain Tumor from MR Images. 134-138. 10.1109/ICCE50343.2020.9290740.

[19] M. K. Abd-Ellah, Ali Ismail Awad, Ashraf AM Khalaf, Hesham FA Hamed, "Classification of brain tumor MRIs using a kernel support vector machine," in International Conference on Well-Being in the Information Society, 2016, pp. 151-160: Springer.

[20] Siegel RL,Miller KD,Fuchs HE,Jernal A. Cancer statistics,2021.CA Cancer J Clin.2021;71:7-33.

[21] Díaz-Pernas, Francisco & Martínez Zarzuela, Mario & Antón-Rodríguez, Míriam & González-Ortega, David. (2021). A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network. Healthcare. 9. 153. 10.3390/healthcare9020153.

[22] Abiwinanda N., Hanif M., Hesaputra S.T., Handayani A., Mengko T.R. (2019) Brain Tumor Classification Using Convolutional Neural Network. In: Lhotska L., Sukupova L., Lacković I., Ibbott G. (eds) World Congress on Medical Physics and Biomedical Engineering 2018. IFMBE Proceedings, vol 68/1. Springer, Singapore.

4.4.3 IRats: Intelligent System for Rat Behavior Analysis

Andrew Zaky Naguib for publishing a scientific paper to the IEEE Xplore as part of the 2022, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The project was part from the cross disciplinary research work that was done with colleges from the faculty of Pharmacy towards the digital transformation. The paper entitled "IRats: Intelligent System for Rat Behavior Analysis" proposed a system to make data extraction from pharmaceutical experiments on rats. Detection and classification of the trajectory of rats and mice will help scientific researchers speed up the development of new drugs for the community. This paper proposes and objectively uses image processing and computer vision to analyze one of the eminent rat behavior analysis experiments, which is the Morris Water Maze (MWM). The paper aims to reduce the time taken by the doctors to extract the information about the rat and its analysis and increase the accuracy of the classification recognition of the rat's trajectory. [25]

IRats:Intelligent system for rat behavior analysis

Andrew Zaky
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Giza, Egypt.
andrew.zaky@msa.edu.eg

Mona M Saber
Department of Pharmacology and Toxicology, Faculty of Pharmacy,
Giza, Egypt.
ORCID: 0000-0001-7916-7863
Mona.magdy@pharma.cu.edu.eg

Reham M. Essam
Department of Pharmacology and Toxicology, Faculty of Pharmacy,
Giza, Egypt.
ORCID: 0000-0004-8882-8330
reham.essam@pharma.cu.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University, Cairo, Egypt.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA),
Giza, Egypt.
aezzat@msa.edu.eg,

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781727

***Abstract*—Rats' behavior analysis is fundamentally important in the medical field and pharmaceutical industry. Drugs and chemical compounds are given to mice and rats to measure their therapeutic effects. If the drug's effect is promising, it will be investigated in humans. Data extraction from experiments and classifying the trajectory of rats and mice will help scientific researchers speed up the development of new drugs for the community. We used image processing and computer vision to analyze one of the eminent rat behavior analysis experiments, which is Morris Water Maze (MWM). In the current work, we implemented two experiments, the first was testing more than one object detection and tracking algorithm on rats and mice behavior analysis. Second, we used three tracking algorithms, which were $P Point-Cloud Recognizer, Dynamic Time Warping (DTW), and FastDTW, to track the rat's path and classify its movements. The result of the first experiment was that a combination of CSRT (Channel and Spatial Reliability Tracking) and optical flow sparse would be more accurate in extracting the data. In addition, the second experiment showed the highest tracking algorithm was FastDTW, with an accuracy of 81%**

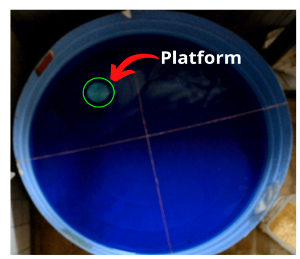
***Index Terms*—Morris Water Maze, Rats, Trajectory detection, Rat behaviours**

I. Introduction

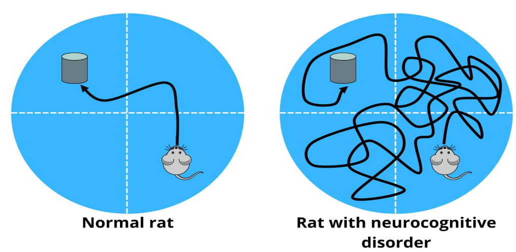
More than one biomedical study has shown that rats and mice are similar to humans in the genetic, physiological, and anatomical aspects [1]. Performed biological processes on rats or mice under similar conditions reflect the same results as that performed on man [2]. Rats and mice occupy about 80%–90% of the total number of animals used in biomedical researches [3]. Rodents (rats) are preferred by researchers due to their tiny size, having a large genetic pool, being easy to handle, and having a short lifetime [4]. Various experiments can be performed on them, but in this study we focused on the Morris Water Maze (MWM) experiment.

Fig.1(a) shows the real pool image where we did our experiments and recorded the video datasets.

As shown in Fig.1(b), normal rats can directly reach the platform while rats suffering from neruocognitive disease wander around before finding it [5].



(a) Real pool of experiment



(b) Noraml rat vs. rat with neruocognitive disorder in Water

Fig. 1. Water Maze experiment

MWM is one of the most commonly used experiments on rats and mice to study spatial learning, memory, and neurocognitive diseases like Alzheimer's disease [6]. The MWM is also used for evaluating cognitive impairments in mice following Traumatic brain injury (TBI) [7]. MWM was established by Richard G. Morris in 1981 [8]. The mice in the experiment are supposed to escape the water by finding a submerged, invisible platform, climbing on it, and staying on it until they return to their home. Water (26°C) was added to a depth of 25 cm in a circular water tank made of white polypropylene, about 50 cm deep and 150 cm in diameter. Four spots around the tank edge, 90 degrees apart, were labelled north (N), south (S), east (E), and west (W), offering four different starting points. A spherical, transparent escape platform with a rough surface (6 cm in diameter) was placed in the middle of one of the four quadrants, submerged 0.5 cm below the water level. A video camera was installed 1.4 metres above the centre of water tank to capture all swimming trials. [9], [10]

Various outputs could be extracted from the MWM experiment. The most common outputs used by researchers in the experiment are the latency time to enter the quadrant, the number of entries, and the time spent in the target quadrant [11], which we worked on extracting. First, the latency time to enter the quadrant is the time taken from the start of the experiment till the first time the rat enters the target quadrant where the platform is placed. Next, the number of entries is the number of times that rat enters the target quadrant. Finally, the time spent in the target quadrant is the time taken inside the target quadrant from the start of the experiment until the end. While for the classification of the swimming rat path, we worked on five moving paths, which are incursion, scanning, focused search, chaining response, and self-orientation as shown in Fig.2.

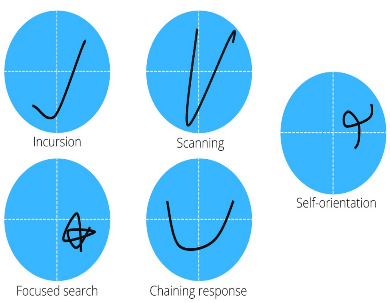


Fig. 2. Classification of the swimming rat path

Studying rat behavior in a water maze has to be done visually or by watching a previously-recorded video to extract the results. These results are divided into measured data (such as latency time , entries number, and time spent in target quadrant) and rat trajectory (classification swimming path). This process consumes a lot of time to extract the appropriate basic information. Besides, it demands more effort, and the results may be inaccurate because of human error. Also, classifying swimming paths is sophisticated and complex to be determined and extracted without a program [12].

Finally, in this paper, we worked on reducing the time spent by the researchers to extract the information from the experiments. Moreover, trying to increase the accuracy of the classification recognition of the rats and mice trajectory.

II. Overview

The researchers start putting a video camera over the pool to record the experiments. Then we start tracking the path and trajectory of the rat. After that, we upload the video into the program to capture the trajectory of the rat, and then time analysis, frames comparison , and image subtraction is done. Various techniques for analyzing rat data in MWM experiments has been presented throughout the years. These approaches range from simple performance assessments to more advanced classification techniques that divide the animal's swimming route into behavioral categories called exploratory strategies [11]. This is followed by saving the extracted data in the database. Then we apply data analysis techniques. Finally, the program will generate a report of the results for the researcher, as shown in Fig.3.

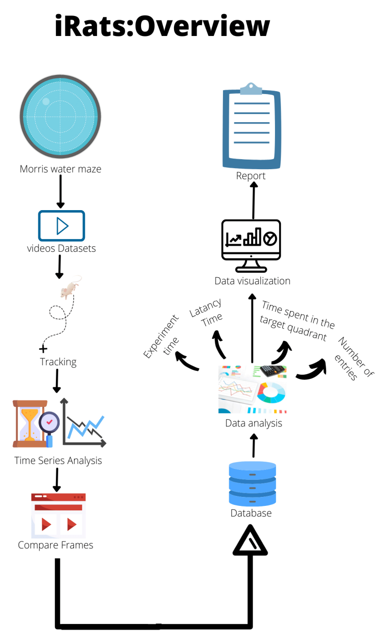


Fig. 3. Overview of system input and output

III. Related Work

In this section, we carefully collected academic research studies that focuses on the detection and tracking of animals and the automation of MWM experiments. We began with research studies that used different methods to detect the rat. Next, we analysed how they reduced the noise during the experiments. Then, the tracking and classifying of the swimming path techniques were used. Finally, the automation of the MWM experiment made it easier to utilise.

A. *Rat detection methods*

Rat detection is important in this experiment as the location of the rat should be determined in each frame. P. Khunarsar, et. al. [13], reconstructed the background model to be a grey-scale image and applied it to the foreground extraction Gaussian Mixture. Then, started to segment the moving object from the background to detect the rat's location. This method was tried on a dataset consisting of 11,655 images and got an 83.53% accuracy rate. S. Yavuzkiliç, M. Aslan, and A. Şengür [14] used threshold value for rat recognition. T. Jin and F. Duan [15] integrated YOLOv3 and a neural network to detect the rats. They found that with the increase of the training dataset, the detection accuracy gradually improved, and the final average accuracy was 89.09% by using 1300 images for training.

B. *Noise reduction*

One of the critical challenges that are faced during the experiments are water motion, light reflection, and brightness change. Z. T. Pennington, et.al [16] used Med Associates, which is an infrared camera that was used during the experiment to prevent light reflection. As infrared does not fetch the stimulation wavelengths that come from light-source. G. Miramontes de Leon , et. al [17] used Kalman filter technique which demonstrated that it performs well in the removal of significant noise data values, and that it was critical to include measurement noise as a mean to compel the algorithm to underestimate measurements with considerable mistakes. The findings showed that the algorithm's improved ability to predict noise and reduce its impact.

C. *Trajectory tracking*

Rat trajectory tracking is essential in water maze experiments. Trajectory tracking is one of the main steps used to determine the rat position and could predict the next position. According to Y.-J. Chen et. al [18], they used the FPGA-based digital analysis and image tracking system in their study. Their results showed that they can process up to 700X480 pixels images with high quality in real-time speed. J. Patman, et. al [19] applied Adaptive Kalman Filter to help them in the tracking process and predict the next step later on. R. Farah, et. al [20], suggested a reliable approach for tracking animals and determining their mobility patterns. The used approach is intended to function in uncontrolled laboratory environments. Their approach is divided into two steps: the first uses a fixed size window and four features to track the animal coarsely, and the second refines the tracked area's borders for a better fit on the animal. Combining the two steps achieves an average tracking error of less than 5%.

D. *Classifying swimming paths*

The experimental rats' reactions and behaviours during the swimming are highly powerful markers for their emotional state in biomedical investigations. Rodents' (especially rats') exploring behaviours are indicative of their stress levels [20]. Also, each rat path has meanings and indications to the scientists. V. Gehring1, et. al [12] made a classification of the path on two types of rats; normal and stressed rats. The findings demonstrated that stressed animals have substantially longer average pathways than non-stressed animals, as predicted, but the increase is non-uniform across the different tactics.

E. *Automation MWM experiment*

M. Forero, et. al [21] worked on automation of detecting the location of the pool and its demotions without selecting it using the Hough transformation. Furthermore, they used mathematical morphology techniques to determine the rat's position. The findings reveal that the method works in every scenario, minimizing analysis time and user interaction. Although, A. Higaki, et. al [22] used data collected over four days in normal mice and vascular dementia model animals and created an artificial neural network (ANN) system to predict the ultimate result in MWM. They used Chainer, an open-source deep learning framework, to create a multiple-layer perceptron (MLP) as a prediction model. The results showed that comparing the measured value with the predicted value and the accuracy of the predictions did not change considerably.

In conclusion, some of the research studies focused on detecting the rat in the pool and how to reduce the noise, and the other research studies worked on tracking and classifying the path. In this paper, we combined the two types of research studies and compared the results.

IV. METHODOLOGY

A. *Data collection*

We have conducted 18 experiments for detecting and extracting different rat behaviours. During these experiments, a camera was attached to the top of the swimming pool. The recorded videos data-set for MWM consists of 18 videos, each lasting about 60 seconds (1 min). Each video has a 3840x2160 resolution and a 60-frame rate.

B. *Algorithms*

Algorithm 1 Detecting and Tracking Rat

Ensure: $ratPostion = False$
Ensure: $postionPoints = none$
  $trackingPoints = []$
  while $True$ **do**
    $Ratpostion, postionPoints \leftarrow RatDetection()$
    if $Ratpostion == True$ **then** ▷ //Ratpostion is True if rat was detected
      $M \leftarrow cv2.moments(postionPoints)$
      $center = (int(\frac{M["m10"]}{M["m00"]}), int(\frac{M["m01"]}{M["m00"]}))$
      $combinationTracking \leftarrow CSRTandOF(center)$
      $trackingPoints.append(combinationTracking)$
    end if
  end while
  $data \leftarrow saveDatatoFile(trackingPoints)$

In Alg.1, the program processes the video and applies detection techniques to allocate the rat position in the water pool. Then, determine the center using the ready-made moments function in the OpenCV library. They calculated it using Green's formula. Next, we start to path the determined center to the tracking algorithms, which are a combination of CSRT and Optical flow sparse algorithms. Finally, we save the tracking points of the rat in a points list and save the data into a file.

Algorithm 2 classification of rat swimming path

Ensure: $trackingPoints \leftarrow readDataFromFile()$
Ensure: $results = []$
  for each $point \in trackingPoints$ **do**
    $classifyPath \leftarrow FastDTW(point)$
    $results.append(classifyPath)$
  end for
  $data \leftarrow saveDatatoFile(results)$

In Alg.2, we start to read the points which are the output from Alg.1. This tracking points will start to fire the FastDTW algorithm to start comparing the rat's swimming paths. Finally, save the data and generate a report for the user.

C. Detection and tracking problems

The detection and tracking techniques of the rat for most of the videos were clear, but in some videos more than one challenge was faced. One challenge relied on the camera stabilization and the other on recorded videos that should not contain any moving objects except the rat for some techniques as image subtraction. Furthermore, the video may contain a light reflection in the water, holding up the detection as shown in Fig. 4. Moreover, some problems related to the HSV method. HSV requires that the recorded video should not contain any other object that has the same range of color as the rat. In MWM, the rat's color changes in some parts of the pool due to the brightness change.

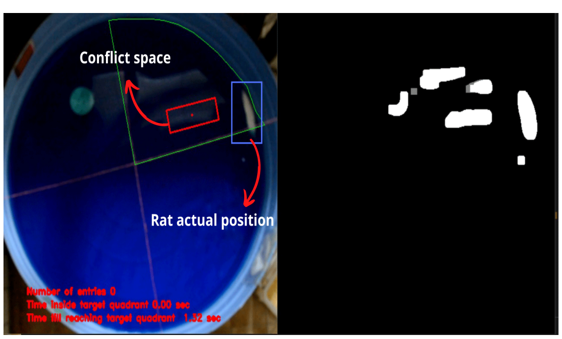


Fig. 4. Image-Subtraction Rat detection problems

V. EXPERIMENT

We implement two experiments that aimed to find high-performance animal detection and tracking techniques for getting the results from the videos.

A. Experiment 1

In the first experiment, we worked on the detecting and tracking of the rat. We implemented more than ten tracking techniques in which most of them successfully worked and some of them failed. The successful techniques are K-nearest neighbours Background segmentation (Image Subtraction), HSV, combining image subtraction and HSV, Boosting, CSRT, MIL, Optical flow sparse, combining CSRT, and Optical flow sparse. While the techniques that failed are KCF, TLD, MEDIANFLOW, MOSSE, Meanshift, and camshaft. Finally, we calculated the Root mean square error (RMSE) for each of the extracted data values.

B. Experiment 2

In the second experiment as shown in Fig. 5, we tried to classify five swimming paths for the rat. The swimming paths are incursion, scanning, focused search, chaining response, and self-orientation. We used three classifying algorithms, which are $P Point-Cloud Recognizer, Dynamic Time Warping (DTW), and FastDTW. We saved the templates of points containing the correct swimming rat paths for each of the used algorithms. Next, we used the collected tracked points list and started to call each classifying algorithm and send each 70 points from the list. After that, we compared the results with the ground truth. Finally, we calculated the accuracy for each of them.

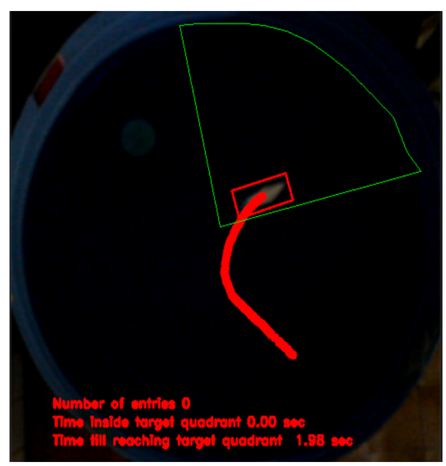


Fig. 5. Rat Trajectory

VI. RESULTS

Table 1 shows the results of Exp. 1, which are RMSE results extracted from each of the eight successful techniques on the three required outputs.

TABLE I
COMPARISON FOR DIFFERENT TRACKING ALGORITHMS IN MEANS OF ROOT MEAN SQUARE ERROR (RMSE)

	Time spent inside target quadrant (sec)	Number of entries	Time till reaching target quadrant (sec)
Image subtraction	1.1393	0.816	1.67511
HSV	1.1178	0.3333	1.6869
Combining Image Subtraction and HSV	1.1301	0.3333	1.6676
BOOSTING	1.2387	0.236	1.89682
CSRT	0.8669	0.4082	0.6804
MIL	0.9645	1	1.7524
Optical flow Sparse	0.9922	0.2357	0.9797
Combining CSRT and Optical flow sparse	0.8669	0.2357	0.6804

We found that all the successful tracking techniques got RMSE of lower than 2, as shown in Table I,. So, we decided to make it more visible by extracting the grouped bar chart shown in Fig.6. We found that CSRT got the lowest error rate in time spent inside the target quadrant and time till reaching the target quadrant. While for the number of entries, optical flow sparse is the lowest error value. Therefore, We have combined the CSRT and the optical flow sparse to get the lowest RMSE.

Fig 6, shows a grouped bar chart for the results. We found that CSRT was the best technique for extracting time spent inside the target quadrant and time till reaching the target quadrant, with values of 0.8669 and 0.6804, respectively. While for the number of entries, we found that optical flow sparse was better with a value of 0.2357. Thus, we decided to make a combination of the two techniques to get the highest performance results.

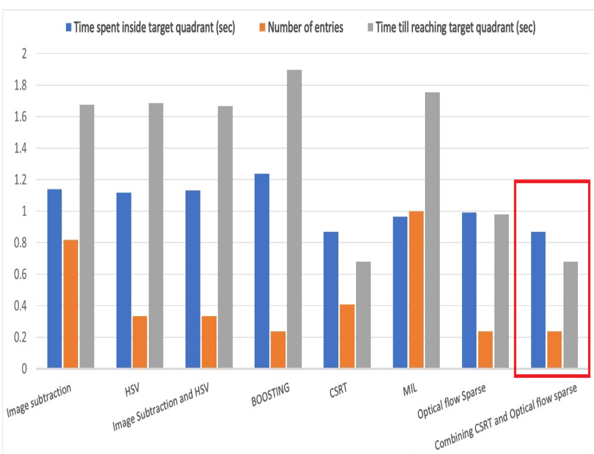


Fig. 6. Grouped bar chart for the tracking techniques

Table 2 shows the results of Exp. 2, where we worked on ten videos, and each video has a different number of events (swimming paths) depending on the time of the experiments. First, we collected the ground truth for each event that occurred during the experiments. Next, we collect the predicted values from each technique.

TABLE II
ACCURACY FOR EACH CLASSIFYING TECHNIQUES

VideoNumber	Number of events	$P Point-Cloud	DTW	FastDTW
1	10	80%	70%	90%
2	7	57%	71%	86%
3	8	75%	75%	75%
4	9	67%	78%	67%
5	14	64%	86%	86%
6	7	71%	86%	86%
7	23	43%	70%	74%
8	17	65%	82%	76%
9	26	65%	85%	88%
10	14	79%	86%	79%
Avarage	-	67%	79%	81%

Then, we start comparing the results from the technique with the ground truth and calculate the success rate for each technique. Finally, we calculate the average success rate resulting from all the experiments for each technique. FastDTW was the best result, with an accuracy of 81%

VII. DISCUSSION

Conducting the two experiments revealed that the combination of the CSRT and Optical flow sparse tracking algorithms showed better extraction of the three experiment outputs than other tracking techniques, namely; the time spent in the target quadrant, the number of entries, and time till reaching the target quadrant with RMSE 0.8669, 0.2357, and 0.6804, respectively. Moreover, similar results were obtained with the DTW and FastDTW for the classification of the rats' swimming paths due to applying the same algorithm, but FastDTW has a higher rate. Therefore, FastDTW is recommended for the classification of the rat's swimming path as it displayed an accuracy level of 81% for each 70 saved points considering the number of events.

VIII. CONCLUSIONS AND FUTURE WORK

In conclusion, for the detection and tracking of the rat MWM experiments, a combination of CSRT and optical flow will be satisfactory with the lowest error rate compared to the ground truth results. Also, for classifying the swimming path of the rat, using FastDTW was the best compared to other algorithms. In the future, we will work on increasing the amount of output data extracted from the video dataset and widen the classification classes of the swimming rat path. This would be done by feeding the models with more templates and trying to find the optimal number of points to be tested on this model using deep learning. Moreover, trying to solve the problems faced during the experiment by combining two or three alternative tracking techniques; if one fails, the other will be replaced. Finally, we would try other behavioural experiments such as Open field and Y-maze experiments.

REFERENCES

[1] Florin Rotaru, Silviu-Ioan Bejinariu, Mihaela Luca, Ramona Luca, and Cristina Nita. Video processing for rat behavior analysis. In *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4, 2017.

[2] Carine Poussin, Carole Mathis, Leonidas G Alexopoulos, Dimitris E Messinis, Rémi H J Dulize, Vincenzo Belcastro, Ioannis N Melas, Theodore Sakellaropoulos, Kahn Rhrissorrakrai, Erhan Bilal, Pablo Meyer, Marja Talikka, Stéphanie Boué, Raquel Norel, John J Rice, Gustavo Stolovitzky, Nikolai V Ivanov, Manuel C Peitsch, and Julia Hoeng. The species translation challenge—A systems biology perspective on human and rat bronchial epithelial cells. *Scientific Data*, 1(1):140009, 2014.

[3] Jianghui Hou. Chapter 9 - paracellular channel as drug target. In Jianghui Hou, editor, *The Paracellular Channel*, pages 175–199. Academic Press, 2019.

[4] Elizabeth C Bryda. The mighty mouse: the impact of rodents on advances in biomedical research. *Mo. Med.*, 110(3):207–211, May 2013.

[5] Jorge Alberto Quillfeldt. Behavioral methods to study learning and memory in rats. *Rodent Model as Tools in Ethical Biomedical Research*, page 271–311, 2016.

[6] Kelley Bromley-Brits, Yu Deng, and Weihong Song. Morris water maze test for learning and memory deficits in alzheimer's disease model mice. *Journal of Visualized Experiments*, (53), July 2011.

[7] S W Scheff, S A Baldwin, R W Brown, and P J Kraemer. Morris water maze deficits in rats following traumatic brain injury: lateral controlled cortical impact. *J. Neurotrauma*, 14(9):615–627, September 1997.

[8] Charles V Vorhees and Michael T Williams. Morris water maze: procedures for assessing spatial and related forms of learning and memory. *Nat. Protoc.*, 1(2):848–858, 2006.

[9] Joseph AU Nunez. Morris water maze experiment. *JoVE*, (19):e897, Sep 2008.

[10] Xijun Wang, Ying Han, Aihua Zhang, and Hui Sun. Chapter 22 - metabolic profiling provides a system for the understanding of alzheimer's disease in rats post-treatment with kaixin san. In Xijun Wang, Aihua Zhang, and Hui Sun, editors, *Chinmedomics*, pages 347–362. Academic Press, Boston, 2015.

[11] Avgoustinos Vouros, Tiago V. Gehring, Kinga Szydlowska, Artur Janusz, Zehai Tu, Mike Croucher, Katarzyna Lukasiuk, Witold Konopka, Carmen Sandi, and Eleni Vasilaki. A generalised framework for detailed classification of swimming paths inside the morris water maze. *Scientific Reports*, 8(1), October 2018.

[12] Tiago V. Gehring, Gediminas Luksys, Carmen Sandi, and Eleni Vasilaki. Detailed classification of swimming paths in the morris water maze: multiple strategies within one trial. *Scientific Reports*, 5(1):14562, Oct 2015.

[13] Pattranis Khunarsar, Nontapat Benjathum, Theekapun Charoenpong, and Ampom Jariyapongskul. A method of swimming rat detection in morris water maze by using image processing. In *2018 International Electrical Engineering Congress (iEECON)*, pages 1–4, 2018.

[14] Semih YAVUZKILIÇ, Muzaffer ASLAN, and Abdulkadir ŞENGÜR. A matlab tool for morris water maze test implementation. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–4, 2018.

[15] Tianlei Jin and Feng Duan. Rat behavior observation system based on transfer learning. *IEEE Access*, 7:62152–62162, 2019.

[16] Zachary T. Pennington, Zhe Dong, Yu Feng, Lauren M. Vetere, Lucia Page-Harley, Tristan Shuman, and Denise J. Cai. ezTrack: An open-source video analysis pipeline for the investigation of animal behavior. *Scientific Reports*, 9(1), December 2019.

[17] Gerardo Miramontes-de León, Hamurabi Gamboa-Rosales, Arturo Moreno-Báez, and Claudia Sifuentes-Gallardo. Kalman filtering of noisy video tracking data from morris water maze tests. In *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, 2018.

[18] Yu-Jen Chen, Yan-Chay Li, Ke-Nung Huang, and Ming-Shing Young. The implementation of a stand-alone video tracking and analysis system for animal behavior measurement in morris water maze. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 1766–1768, 2005.

[19] Jon Patman, Sabrina C.J. Michael, Marvin M. F. Lutnesky, and Kannappan Palaniappan. Biosense: Real-time object tracking for animal movement and behavior research. In *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8, 2018.

[20] Rana Farah, J.M. Pierre Langlois, and Guillaume-Alexandre Bilodeau. Rat: Robust animal tracking. In *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 65–70, 2011.

[21] Manuel Forero, Natalia Hernandez Riaño, Cristian Morera, Laura Baquedano Santana, and Luis Aguilar. New automatic method for tracking rats in a pool for medication studies. page 85, 09 2019.

[22] Akinori Higaki, Masaki Mogi, Jun Iwanami, Li-Juan Min, Hui-Yu Bai, Bao-Shuai Shan, Masayoshi Kukida, Harumi Kan-no, Shuntaro Ikeda, Jitsuo Higaki, and Masatsugu Horiuchi. Predicting outcome of morris water maze test in vascular dementia mouse model with deep learning. *PLoS One*, 13(2):e0191708, February 2018.

4.4.4 The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches

Youssef Maged on publishing a Conference paper on his graduation project, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The paper entitled “The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches" The objective of this research paper is to try out different machine learning models on Diabetic patients in order to predict their blood glucose level based on the heart rate values along with extra features coming from the smart watch.[10]

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781730

The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches

1st Youssef Maged
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA).
Cairo, Egypt
youssef.maged4@msa.edu.eg

2nd Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—The objective of this research paper is to try out different machine learning models on the patients from the D1NAMO data-set in order to predict blood glucose level based on the heart rate values along with extra features coming from the smart watch, The glucose readings will be measured by the CGM along the 2 weeks duration of the calibration between smartwatch and CGM device. Our proposed System will be applied per patient, the patient will use the CGM device along with the smart watch and calibrate the 2 readings together, after the CGM expires the data will be taken and processed together, any noise that interfered with the ECG signal will be removed in order to have a clean noise that can be combined with the Glucose readings to be in the exact timestamp, as an output the user will have the ability to depend only on the smart watch in knowing the blood sugar level for a duration of time.**

***Index Terms*—Continuous Glucose Monitoring (CGM) , Re-sampling , Blood Glucose (BG), Electrocardiogram (ECG) , Mean squared error (MSE) , Mean absolute error (MAE) , Root mean squared error (RMSE)**

I. Introduction

Diabetes is a disease that happens when the blood glucose level is too high, the glucose which comes from the portion of food that the patient intakes, is responsible for supplying the body with the energy needed in order to do his/her daily routine with the sufficient energy, there is a hormone that the body produces that is secreted by the pancreas is called insulin which controls the glucose levels, so the diabetic person suffers from the lack of insulin in their body which leads to the stack of glucose in the blood and having an excessive amount of glucose in the blood can cause medical issues [1]. According to IDF Diabetes Atlas, 537 million adults have been suffering from diabetes over the few years [4].

There are 3 types of diabetes. First type, is when the body stops producing insulin which causes the increasing of the blood sugar, it is often developed at a fast pace and is generally diagnosed in young-age people, daily insulin shots needed for those who have type 1 diabetes in order to survive. Second type, is that the body doesn't use the insulin produced well as a result the body can't keep a normal blood sugar levels and it is normally diagnosed at old-age and it is not noticeable so monthly blood sugar test is needed. Third type, is the Gestational Diabetes which is developed in pregnant women and it puts the baby at high risk for health problems and it ordinarily disappears after the child is conceived yet expands the danger for type 2 diabetes sometime along the way [2].

There are alot of devices out there that have the capability to measure the blood glucose level whether on time or every few minutes, they have different price ranges depending on the quality of the product and its effectiveness, First type of products is the finger-picking devices which is not preferable by everyone as that the patient have to finger-pick himself in order to get a blood sample to put it in the device and let it process the data and estimate the blood glucose level [3].

The other type is the glucose monitoring devices that keeps track of the blood glucose level continuously throughout the day, from these type of devices it enables the patients to have the ability to review the change of glucose level, this review enables the user of the device to keep track of his/her health life and take decisions based on those reviews, decisions like the most effective method to adjust the food's portion intake, active work and medicine intake [5].

In this paper, we will be focusing on the second type of devices which are the continuous glucose monitoring devices (CGM), those devices are sensors that are placed onto a different body parts like upper arm or stomach.

CGM device has a needle that is inserted inside the body to penetrate the skin and then a tiny filament is left there in order to take glucose readings throughout the day, those readings are then sent to the user's smartphone in order to check out the glucose values and these devices are supported with features that enables it to send alerts if the blood sugar is higher or lower than the target range set by the user [6].

As there are pros for having CGM device there are also cons, some people complained about the painful sensor insertions, there might be a lost signals due to any circumstances also its high price and its need to be changed every two weeks may be considered as a problem for some people who cant afford buying the device for several weeks [7].

The objective of this paper is to replace the CGM device with the smart-watch that is capable of measuring heart rate values. This can be achieved by proposing a machine learning

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

model that can use the heart rate values along with different features in order to predict the glucose values, this model will help solve time gap delay that exists between the measured glucose level by the devices and the actual level, it will also save people from buying costly CGM device.

II. Related Work

Calbimonte et al. [15] discussed their goal which was focused mainly on D1namo Project that targeted non-invasive diabetes monitoring, they used different wearable sensors to collect the data, as an example they used sensor belt in order to extract reading for accelerometer along with ECG, they created two models one for heart beat classification in glycemic conditions and one for prediction of blood glucose levels of the patient at a certain time, they prepared the data by cleaning any noise that interfered with the ECG signal.

Shaqiri et al. [9] proposed a deep learning model to predict the glucose levels based on the heart rate variability by using two main HRV parameters which are SDNN and RMSSD, also they build an architecture of three hidden layers in addition to Adam optimizer and tanh for activation function, they used Autokeras in order to pick the best model with the best performance and they achieved 91.96% validation accuracy.

Rashitan et al. [10] They proposed a machine learning framework its purpose was to Extract different features from heart-rate that correlates strongly with the extracted features from CGM data, at first They applied Feature Extraction on the data after representing the values as time series and then they used Canonical Correlation Analysis (CCA) in order to find the maximally correlated latent representations. their goal was to provide a technique to grow the efficiency of heart-rate data to be used for diabetes identification for patients and medical services suppliers.

Cichosz et al. [11] presented their objective which was to use the previous and current readings to predict glucose levels in the future based on those past readings, they used used an artificial neural network regression algorithms (NN). In order to achieve high results they used 3 different data-sets, one for training and testing and the other two for validation.

Shaqiri et al. [8] attempted to sort out which heart-rate variability parameter that can give a high correlation between those parameters and the Glucose values of a patient, after trying many parameters they concluded that the Root Mean Square Successive Differences parameter (RMSSD) concluded better results in short term continuous measurements and for the long term measurements they concluded that Standard Deviation of Normal to Normal intervals (SDNN) is better.

Cichosz et al. [16] the objective of this study was to predict and detect hypoglycemia based on the relation between the heart-rate and CGM, they have done this study on 16 patients with 16 hypoglycemic events, they started by applying feature extraction on both of the data and then they ranked the features based on the highest correlation and they used the high correlated subset of features, as a result they managed to predict all the 16 hypoglycemic events without any false-positives.

Gusev et al. [17] they elaborated a comprehensive overview of the progress that has been reached until now by different researchers to use different approaches and techniques based on machine and deep learning algorithms to conduct a relation between HRV and CGM, they also discussed the usage of ECG sensor in identifying the heart-rate and correlate it with other features.

Charamba et al. [18] they started by collecting the data then they started processing on it by cleaning both of the readings, they used expert opinion to determine the range of measurements that weren't compatible, and they also cleaned signal loss. their purpose was to study the relation between the glucose and QTc and they discovered that the glucose have a positive relationship with the QTc.

Gusev et al. [19] they designed a system that is capable of measuring glucose based on ECG readings along with different HRV parameters, they collected the ECG signal through a wearable sensor and its data is collected by a smartphone and it is analyzed by using sophisticated machine learning algorithms, the ECG signal is digitalized and cleaned from any noise and it is processed through the models along with the parameters extracted from HRV.

Vishinov et al. [20] the objective of this research was to investigate the different parameters that can be extracted from the HRV in order to correlate it with the glucose readings, they used two methods to correlate which were Pearson and Spearman, they divided the HRV parameters into two sections which are time-domain parameters like SDNN , ASDNN and the other section was non-linear parameters like SD1. They concluded that SD1/SD2 got the strongest positive correlation.

III. Methodology

In this section we will be talking about the data-set that was used in the training process of the different machine learning models, also we will be mentioning the data preprocessing techniques used in cleaning, combining and the feature extraction of the data.

A. System Overview

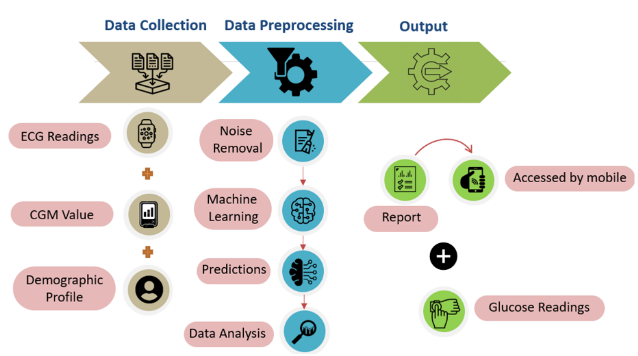


Fig. 1: System Overview illustrates the inputs and the flow of data throughout the system.

Figure 1 shows the proposed system will be applied per Patient, the system starts with the data collection process where the patient will have to get the CGM device along with the smart watch firstly and we will calibrate them together during the 2 weeks duration of the CGM device, after the CGM expires the data will be taken as an input for the system then the 2 readings will be processed together.
After that we move onto the Data Preprocessing phase where any noise that interfered with the ECG signal will be removed in order to have a clean noise that can be combined with the glucose readings to be in the exact timestamp.
After the data has been cleaned and calibrated together it will be used to train the different machine learning models with, as an output the patient will have the ability to depend only on the smart watch in knowing the blood sugar level for a duration of time along with a report with the previous readings that can be accessed by mobile phone, monthly calibration would be preferable in order to have.

B. Data Collection

As shown in Table I, few data-sets were checked to conclude whether they contained the values that were needed or not to be used in the training process that will be used to satisfy our objective.

TABLE I: Example of the datasets

Dataset	Glucose	ECG	Activity
D1NAMO [13]	✓	✓	✓
CITYPublicDataset [14]	✘	✓	✓
Diabetes UCI [12]	✓	✘	✘

The data-set that was used is called D1NAMO data-set, it was a public data-set and it contained 29 patients (9 Diabetic Patients and 20 healthy ones), the data were in csv format containing the ECG signals readings along with the glucose readings and extra features that will be used in the future work, the data were collected during 4-6 days duration.
The data wasn't coordinated due to time gaps between the 2 readings of ECG and glucose, also the ECG signal was found to be noised. That's why data preprocessing needed to be made in order to prepare the data to be used to train the model with.

C. Data Preparation

a) Noise Removal: The data contained noise which happened during the transmission and collection of the signal, this noise affected the values of the ECG as shown in Figure 2.

The noise that was discovered to be found in the signal was high frequency noise specifically Motion Noise that happens due to the electrical activity that occurs during the different movement of the muscle along with changes in skin temperature.
In order to remove the noise, a set of filters were applied on the signals, IIR Notch filters were used to remove the motion artifact interference along with FIR Filters that were used to specify the suitable frequency for the ECG data.

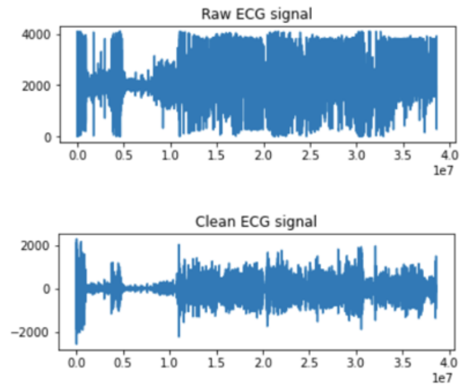
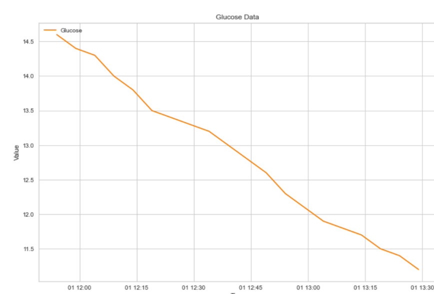
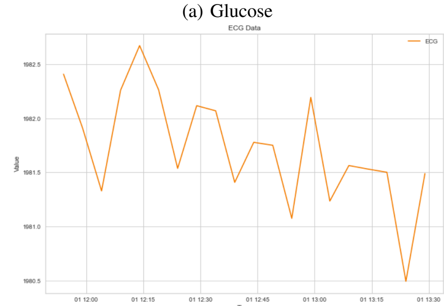


Fig. 2: ECG signal before and after applying IIR and FIR filters on the signal.

b) Time Interval: There was a different time interval between the ECG values and the glucose values, as the glucose were measured every 5 minutes and the ECG were measured every 1 milliseconds, which made a huge time gap between the two measurements, so we managed to combine these two readings together by using time-window averaging which is capable of capturing the average values of the ECG measurements in every 5 minute duration of the glucose values as shown in Figure 3, so now we have the glucose value in corresponding to the average of the ECG values in the specified duration, which makes the data ready for us to train the model with.



(a) Glucose



(b) ECG Signal

Fig. 3: Data Visualization

c) Feature Selection: The D1NAMO data-set came with some extra features that can be used along with the ECG readings, feature selection techniques were applied on those features in order to deduct which one has the highest importance in predicting the glucose values. as shown in Figure 4.

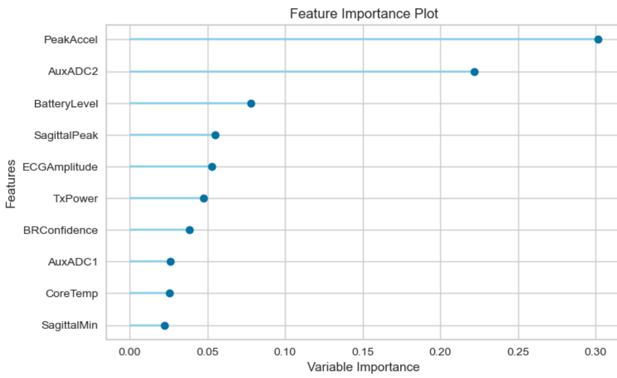


Fig. 4: Feature Importance Plot

Different tools were used in order to deduct the highest correlated features with glucose values and three features were chosen from this plot in order to be used in the prediction of glucose values.

d) Refining Data: After the data has been cleaned from all the noise using the previous filters mentioned above and after extracting the most important features to be used with the ECG signal in the training process, different regression models were used in order to conclude different results and visualize those results to conclude the high result model among the different models that can used in order to predict the glucose values based on the heart rate along with different features.

IV. Experiment 1

Our objective in this experiment is to predict the glucose readings based on two methods, the first one is to predict the glucose based on heart rate only and the second method is to use extra features along with the heart rate. We conducted this experiment on 1 patient only during his 4 days of data collection and Table II shows the difference in Number of records between both values.

TABLE II: Difference between data

Data	No of Records
Glucose	1438
ECG Signal	38,621,500

At the first method, we prepared the data by combining the 2 readings of glucose and ECG signal together without any extra features in order to have the 2 readings at the same time, we used Moving Averages in the ECG signal in order to decrease The readings in order for it to be compatible with the glucose values in time and Table III shows the results.

TABLE III: Exp1's First Method Results

Model	MAE	MSE	RMSE	R2
Gradient Boosting Regressor	3.8105	23.4060	4.8357	0.0158
Light Gradient Boosting Machine	3.8226	23.4706	4.8423	0.0131
Ridge Regression	3.8230	23.8178	4.8786	-0.0018
Least Angle Regression	3.8230	23.8178	4.8786	-0.0018
Orthogonal Matching Pursuit	3.8230	23.8178	4.8786	-0.0018

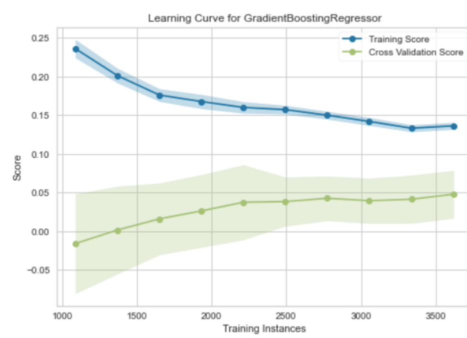
In the second method of this experiment, extra features were used in order to be added along with the ECG signal in order to deduct the difference between those two methods.

TABLE IV: Exp1's Second Method Results

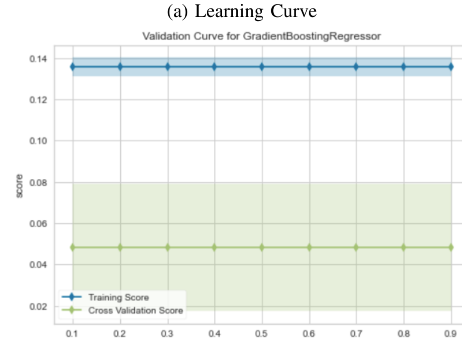
Model	MAE	MSE	RMSE	R2
Gradient Boosting Regressor	3.7472	22.4390	4.7351	0.0482
Light Gradient Boosting Machine	3.8226	23.0307	4.7975	0.0223
Ridge Regression	3.7702	23.0708	4.8015	0.0216
Least Angle Regression	3.7702	23.0708	4.8015	0.0216
Orthogonal Matching Pursuit	3.7795	23.1761	4.8125	0.0171

The combined data of the ECG and the extra features were used to train different models with it, as shown in Table IV, those are the top 5 models that got the least MAE and MSE among the models, Gradient Boosting Regressor scored the Highest results among the others.

Figure 5. The Learning curve of the model shows a slowly decreasing rate in the training score along with the faster increasing rate in the cross validation score which shows the different training instances that were made. While the Validation curve shows an constant rate in the training score.



(a) Learning Curve



(b) Validation Curve

Fig. 5: Learning,Validation Curves For The Gradient Boosting Regressor Model in the Second Method

V. EXPERIMENT 2

We had the same objective in this experiment which is to predict the glucose readings based on the heart rate only in one method and in the other method while using extra features, but we wanted to conduct it on higher number of patients, so we conducted this experiment out on 5 different patients during their 4 days of data collection. This time we had much larger data which took huge time in order to process it, as shown in Table V:

TABLE V: Difference between data

Data	No of Records
Glucose	4,606
ECG Signal	206,023,500

And we used different regression models on the data that we collected and processed and the results that was conducted from the models are shown in Table VI:

TABLE VI: Exp2's First Method Results

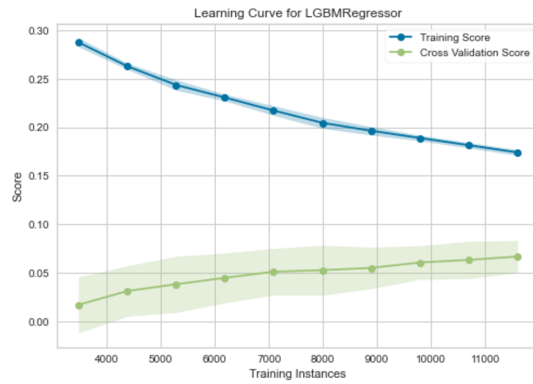
Model	MAE	MSE	RMSE	R2
Light Gradient Boosting Machine	3.6140	20.5731	4.5350	0.0840
Gradient Boosting Regressor	3.6310	20.5854	4.5362	0.0838
AdaBoost Regressor	3.8473	22.1800	4.7089	0.0126
Elastic Net	3.7947	22.4679	4.7390	0.0002
Orthogonal Matching Pursuit	3.7948	22.4679	4.7390	0.0002

In the second method we used the features that were chosen in experiment 1 in order to identify the difference in the results while using those extra features.

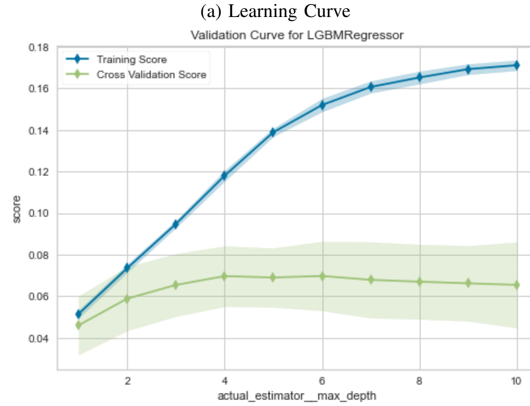
TABLE VII: Exp2's Second Method Results

Model	MAE	MSE	RMSE	R2
Light Gradient Boosting Machine	3.5965	20.5443	4.5323	0.0664
Gradient Boosting Regressor	3.6021	20.6187	4.5404	0.0632
AdaBoost Regressor	3.7519	21.3686	4.6221	0.0291
Linear Regression	3.7162	21.5673	4.6436	0.0201
Ridge Regression	3.7164	21.5689	4.6438	0.0201

The data was used to train different machine learning models and Table VII shows the results of the top five models that scored the highest results among each other and as shown the Light Gradient Boosting Machine got the better results at 3.5965 Mean absolute error. Learning curve and validation curve are shown in Figure 6. The Learning curve shows a decreasing rate in the training score along with the increasing rate in the cross validation score which shows the different training instances that were made. While the Validation curve shows an increasing rate in the training score.



(a) Learning Curve



(b) Validation Curve

Fig. 6: Learning,Validation Curves For The Light Gradient Boosting Machine Model in the Second Method

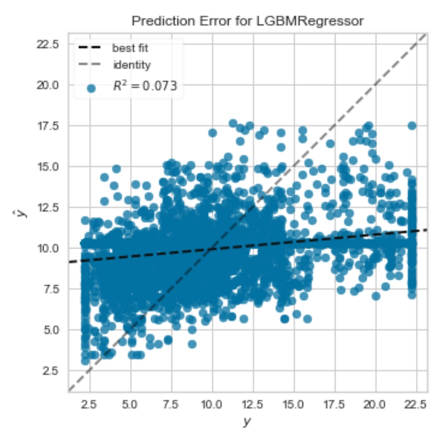


Fig. 7: Prediction Error Plot

Figure 7, shows the Prediction Error Rate for the Light Gradient Boosting Machine Model and the plot here shows the best fit line that was conducted during the testing process.

VI. Conclusion

The Gradient Boosting Regressor got the highest results scoring 3.7472 MAE & 22.4390 MSE when dealing with small scale of data like data of one patient as shown in Experiment IV, However when dealing with Huge scale of data like five patients or more as in Experiment V, Light Gradient Boosting Machine handled the data well by scoring 3.5965 MAE & 20.5443 MSE. Those results were achieved using the ECG signal, glucose and combined with different features, those features that were included in the experiments were chosen from the feature importance plot that were extracted and based on certain features were chosen. As a conclusion, There exists a correlation between the heart-rate and the blood glucose levels along with other extracted features.

VII. Future Work

We will be exploring different modalities in order to improve the results, one of the modalities that can be used is the food tracking system that is capable of identifying the glycemic load that comes from the food and how will it impact the blood glucose level of the patient, another modality that can be used is the identification of glucose by identifying different behaviour performed by the patient.

References

[1] U.S. Department of Health and Human Services. (n.d.). What is diabetes? National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved from https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes.

[2] Centers for Disease Control and Prevention. (2021, November 16). What is diabetes? Centers for Disease Control and Prevention. Retrieved from https://www.cdc.gov/diabetes/basics/diabetes.html.

[3] Olansky, L., & Kennedy, L. (2010). Finger-stick glucose monitoring: issues of accuracy and specificity. Diabetes care, 33(4), 948–949. https://doi.org/10.2337/dc10-0077

[4] Tenth edition. IDF Diabetes Atlas. (n.d.). from https://diabetesatlas.org/.

[5] U.S. Department of Health and Human Services. (n.d.). Continuous glucose monitoring. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved from https://www.niddk.nih.gov/health-information/diabetes/overview/managing-diabetes/continuous-glucose-monitoring.

[6] What is a CGM? flash glucose monitoring. South Africans with Diabetes. (2020,November 5). Retrieved from https://sweetlife.org.za/.

[7] Pros and cons of CGM described by parents of children with diabetes. diabetes.medicinematters.com. (2019, August 8). Retrieved from https://diabetes.medicinematters.com/continuous-glucose-monitoring/children/cgm-parents-children/17049042.

[8] Shaqiri, Ervin & Gusev, Marjan & Poposka, Lidija & Vavlukis, Marija. (2020). Correlating Heart Rate Variability to Glucose Levels.

[9] Shaqiri, Ervin & Gusev, Marjan. (2020). Deep Learning Method to Estimate Glucose Level from Heart Rate Variability. 10.1109/TELFOR51502.2020.9306609.

[10] Rashtian, Hootan & Torbaghan, Solmaz & Rahili, Salar & Snyder, Michael & Aghaeepour, Nima. (2021). Heart Rate and CGM Feature Representation Diabetes Detection From Heart Rate: Learning Joint Features of Heart Rate and Continuous Glucose Monitors Yields Better Representations. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3085544.

[11] Cichosz, Simon & Jensen, Morten & Hejlesen, Ole. (2021). Short-term prediction of future continuous glucose monitoring readings in type 1 diabetes: Development and validation of a neural network regression model. International Journal of Medical Informatics. 151. 104472. 10.1016/j.ijmedinf.2021.104472.

[12] Aitbayev, A. (2021, July 2). Diabetes UCI Dataset. Kaggle. Retrieved December 25, 2021, from https://www.kaggle.com/alakaaay/diabetes-uci-dataset

[13] Fabien Dubosson, Jean-Eudes Ranvier, Stefano Bromuri, Jean-Paul Calbimonte, Juan Ruiz, Michael Schumacher, the open D1NAMO dataset: A multi-modal dataset for research on non-invasive type 1 diabetes management, Informatics in Medicine Unlocked,2018, https://doi.org/10.1016/j.imu.2018.09.003

[14] JCHR. Jaeb Center for Health Research. (n.d.). Retrieved December 25, 2021, from https://public.jaeb.org/datasets/diabetes

[15] Calbimonte, Jean-Paul & Ranvier, Jean-Eudes & Dubosson, Fabien & Aberer, Karl. (2017). Semantic representation and processing of hypoglycemic events derived from wearable sensor data. Journal of Ambient Intelligence and Smart Environments. 9. 97-109. 10.3233/AIS-160420.

[16] Cichosz, Simon & Frystyk, Jan & Hejlesen, Ole & Tarnow, Lise & Fleischer, Jesper. (2014). A Novel Algorithm for Prediction and Detection of Hypoglycemia Based on Continuous Glucose Monitoring and Heart Rate Variability in Patients With Type 1 Diabetes. Journal of diabetes science and technology. 8. 10.1177/1932296814528838.

[17] Gusev, Marjan & Poposka, Lidija & Spasevski, Gjoko & Kostoska, Magdalena & Koteska, Bojana & Simjanoska, Monika & Ackovska, Nevena & Stojmenski, Aleksandar & Tasic, Jurij & Trontelj, Janez. (2020). Noninvasive Glucose Measurement Using Machine Learning and Neural Network Methods and Correlation with Heart Rate Variability. Journal of Sensors. 2020. 1-13. 10.1155/2020/9628281.

[18] Charamba, Beatrice & Liew, Aaron & Coen, Eileen & Newell, John & O'Brien, Timothy & Wijns, William & Simpkin, Andrew. (2021). Modelling the relationship between continuously measured glucose and electrocardiographic data in adults with type 1 diabetes mellitus. Endocrinology, Diabetes & Metabolism. 4. 10.1002/edm2.263.

[19] Gusev, Marjan & Gusheva, E. & Poposka, Lidija. (2020). Design of a Non-invasive ECG-based Glucose Measurement System. 303-308. 10.23919/MIPRO48935.2020.9245403.

[20] Vishinov, Ilija & Gusev, Marjan & Vavlukis, Marija & Poposka, Lidija. (2020). Correlating Short-Term Heart Rate Variability and Instantaneous Blood Glucose Measurements. 1-4. 10.1109/TELFOR51502.2020.9306517.

4.4.5 Fruit Diseases Identification and Classification using Deep Learning Model

Mohammed Ahmed Matboli on publishing a Conference paper on his graduation project, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The paper entitled ”Fruit Diseases Identification and Classification using Deep Learning Model” proposed a system that extracts (Classifies.) the fruit diseases to help in the early detection of any illness, which is useful in the exporting process in Egypt. This paper proposes and objectively evaluates an image management system for disease detection and economic grouping in natural fruit products. The model implemented in this paper was capable to beat the more extraordinary accuracy in the citrus fruit. [11]

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781688

Fruit Disease's Identification and Classification using Deep Learning Model

Mohammed Ahmed Matboli
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Cairo, Egypt
mohamed.ahmed184@msa.edu.eg

Ayman Atia
HCI-LAB,Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Email: aezzat@msa.edu.eg

***Abstract*—Now days Fruits is being produced from alot of countries as the global fruit production reached up to 2914.27 production in thousand metric tons and in the upcoming years a lot of countries want to increase the production. However, some challenges and problems persist to exist through the fruits production like the quality of the fruit, the cost of the production, the quality of the seed and the illness of the fruit itself. There are types of recognized Diseases in the apples such as blotch, scab, and rotten diseases, and for the citrus, Black spot, Scab Citrus, and Citrus Canker.In this project, Our aim is to identify the best transfer learning model that is able to achieve the most extraordinary accuracy through the early detection of fruit diseases. Five different types of transfer learning models are presented, and they are being used in this proposed solution as the customized CNN model achieved the highest accuracy that reached up to 99.16%.**

I. Introduction

The conventional strategy for the recognition and detectable detection of diseases from organic substances depends on the human eye of specialists. Some agricultural places like to get help from specialists which is extremely expensive and cumbersome because remote areas are hard to be accessible. The process of localizing the illness of the crop is fundamental to consistently differentiate the indications of diseases just in time as they appear in the fruit under development. Infections caused by natural fruit products can lead to a significant number of losses in the yield, and in this way the quality will be affected likewise. During the harvest, to recognize which is causing this problem, it needs to be controlled within a year to escape the losing that is happening because of the diseases. Those infections also can produce damage during the import and export process which will increase the loss of the production of the fruits.

Some types of organic diseases also contaminate separate parts of the tree by typically causing infections to the branches, twigs, and leaves. For example, in apples, there are standard types of illnesses which are scabbed, blotch and rotten as shown in the fig 1. Another fruit like pomegranate, its frequent illnesses in common are Rotten, Anthracnose, and Bacterial blight. Also in Citrus, ordinary infections are black spot, scabbed, and citrus fruit canker. It's complex to accurately detect those diseases by the human eye without being able to get mistaken. Can those possible errors be reduced or even able to not be caused by any potential mistakes?

Fig. 1. Common diseases found In an organic apple products: (a) Apple scab, (b) Apple rot, and (c) Apple Blotch.

Machine learning is one of the important tools that can help us in many Fields in our life, that can be produced easily in a device for helping the human in their daily tasks. It is applied with a machine learning technique that can predict and give a trusted selection to make the human more confident while making a choice in their work for example. Machine learning has many kinds to work with, yet every type has its own properties that work with various problems. For example, classification, Regression, clustering, Dimensional reduction, Model selection, and pre-processing. In the classification, can be employed in some applications like image recognition and spam detection which will be extremely helpful in those areas. Some algorithms used in the classification are SVM, Near-est Neighbors, Random Forest and more.

This paper presented a solution that supports the employees. To possess a second accurate eye while sorting the fruit lane in import and export facilities or for the farmers. The article degraded the image from any external noises that can appear on the image. The paper was able to classify the ordinary diseases that appear on the fruits. The structure will try beating the existing accuracy to possess an accurate system that produces no error while classifying on more than one disease by producing a comparative study between the previous work and this paper.

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

II. Related work

Based on C.senthilkumar et al., [1], in this paper, they talk about citrus fruit and its importance in agriculture, as the citrus is an excellent source of vitamin C. Basically, their objective was to recognize and classify the diseases of the citrus fruit. Their structure was based on various processes, and they are pre-processing Hough Transform, optimal weighted segmentation, and (RFANN) rough fuzzy artificial neural network that depends on the classification. The model proposed in this paper is called HT-RFANN. The diseases utilized to classify them were Greening, Scab, Canker, and Blackspot. The total number of images in the data-set was 150 images that includes healthy and affected images of citrus. Their weaknesses were the data-set they used however was extremely small and wasn't using deep learning in the feature extraction phase. The proposed model was HT-RFANN and was the best one to achieve accuracy up to 96.93%.

Sharifah Farhana et al [2] presented a system that discusses how important the citrus is and how the citrus is critically valuable in economics. As citrus is produced in 140 countries it has an enormous importance to the community, there are some common diseases that can produce an enormous loss in the production of the citrus. In this paper, they work on the leaf of the citrus. They work in two stages to allow early detection of the citrus disease and to classify using the images that have of the leaf details. In stage one, they propose a way to discover the diseased area using a proposed network. Stage two is to classify the target area employing a classifier to classify the three types of citrus diseases. The data-set used in this paper was a citrus data-set available on Kaggle for the experiment. The classes of the data-set represent three types, and they are (Huanglongbine, Canker, and Blackspot) with the total number of images of 477 and the image size is 256 x 265. Their strength in their proposed solution is in proposing a potential target for the diseased areas. The proposed model achieved accuracy up to 94.37% in detection and an average precision of 95.8%.

Asmaa Ghazi Alharbi et al., [3], presented a system that worked on apple diseases. Their objective was to classify the apple illness. There were four classes of Apple diseases to work with which are scabbed apples, rotten apples, blotch apples, and ordinary apples. The data-set of the apples turned out to be bought from the market nearby. They achieved an accuracy of 99.17% implementing a CNN efficient model for predicting apple diseases. P. Kola Sujatha, et al. [4] , presented a system that worked on apple diseases. Their classification was into three classes which are apple scab, apple blotch, and apple rot diseases. Their technique of apple diseases contribution was in several steps. In step one, they applied segmentation. Step two used k-means clustering. Step three was based totally on the shape, size and surface of the apple. Their image acquisition technique was by photographing the apple into classes to get the image from various angles. Their achieved accuracy was 96%.

Akshatha Prabhu et al., [5] presented a system that indicates the performance of hyperspectral imaging system, segmentation algorithms, feature extraction methods spectral and the texture, color of the fruit. They applied a comparative study between classifiers in more than one fruit. The fruits applied were Apples, Citrus, orange, papaya, peach, grapefruit, and pomegranate. Their most excellent accuracy was 100% using PNN among KNN and SVM. Liu Zixi et al., [6] presented a system that worked on apple illnesses. Their objective was to classify whether the apples are diseased or not. The combined samples represent 265 samples consisting of two classes, diseased one and healthy one. Six classifiers were employed to establish their models. While applying their experiments, it proved to them that SMOTE-Tomek Link stays the one maintaining the data distribution. The most excellent classifier was the GBDT classifier with an accuracy of up to 95.97%. Shiv Ram et al., [7] presented a system to be used in localization of apple products. They achieved their contribution in steps. In step one they used the K-means to help them in the segmentation. Step two is by getting the feature extraction from the segmented image. The last step is by classifying using a multi-class support vector machine. They used three types of diseases they were apple rot, apple scab, and apple blotch. The accuracy of the proposed solution was 93%.

Ayaz et al., [8] presented a system with a contribution on the apple fruit. They used three classes with three diseases they are scab, rot, and blotch. Their objective was to classify the three diseases. They achieved accuracy up to 99.99% using the DCGAN-DCNN model. They tried some other models and assigned a comparison study between them. The models used in this area were 2D CNN, MiniVggNet, SqueezeNet, ResNet50, etc. Zhang et al., [9] presented a system that worked on apple disease position. They used Fuzzy C-means and (FCM-NPGA) Nonlinear Programming Genetic Algorithms as techniques in their contribution. There are steps they generated in that contribution. First, the image was enhanced using fractional differentiation. Then they remove the noises and edges too as to obtain the important texture information. Then they use the FCM-NPGA to segment with it. Their objective was to detect the defects of an apple by analyzing it. They worked with 2000 images as an experiment to achieve an accuracy of 98%. Samajpati et al., [10] presented a system that works on fruit diseases. They employed the segmentation using the K-means clustering techniques. They depend on the texture features and the color of the fruit by extracting them. Then, in the end, they fust two or more texture features and the color as CCV, LTP, LBP, CLBP, and GCH. They used a random forest classifier in their classification contribution. Khan, Muhammad Attique et al., [11] presented a system using lesion segmentation, prominent features selection, and spot contrast stretching in it. They employed SCP in enhancement and segmentation in their contribution. They used M-SVM as a classifier and achieved accuracy of 92.9%, 94.30%, and 97.20% (Test 1). They contributed by three tests, used in (Test 2) PCA as a feature reduction, and it (Test 3) they show the proposed features selection technique. Their data were about leaves they consist of four classes Scab, Rust, healthy, and Black rots. Yogesh et al., [12] presented a system that uses the segmentation technique on the fruit to grades its disease. They separate the image into several areas then they focus on the size, color, texture, and shape of the fruit. They aim to replace the manual techniques as it Is hard to sort the pleasant fruits. They used more than one method which is K-means, fuzzy c-means, watershed and Otsu. Then a comparative study is generated by them to discover the defective areas in the fruits.

K. Swetha et al., [13] presented a system that depended on the right value of the threshold. Thresholding is used frequently as a technique of segmentation its idea is for setting the most excellent value of the threshold to obtain the best results. Their objective is to predict the injured area and give a percentage for It. They used the perimeter and region as parameters to make it more accurate while detecting the affected area. Dahua Li et al., [14] presented a system that focused on the green apples. In their proposed system they tried to combine color features, shape features, and texture features to solve the segmentation problem to focus on the target and in the background of the apple-picking robot. They used the (SVM) support vector machine to aid them in segmenting the image prelaminar also for the shape and color they are merged to achieve the accuracy of the segmentation. The algorithm they implemented in this paper achieved a better recognition rate and improvements in speed than other algorithms. They achieved a 90.08% accuracy using their algorithm. Jolly et al., [15] presented a system that works on describing a various techniques and approaches for the identification and detection of diseases in an apple by the assist of computer vision. Their approach in their paper was to analyze the surface of the apple for any abnormal behavior in the texture, color, and image features. They employed the segmentation to achieve their goal for this they used (ROI) Region of interest, also the K-means is used. A comparative study is applied that shows the difference between kernel PCA, Local binary patterns, and Haralick features to test their performance through Gabor features. The classification had been performed with K-nearest neighbors and (SVM) support vector machine. The accuracy respectively was 96.9%, 96.25% using Gabor+LBP and Gabor+Haralick. Ayyub et al., [16] presented a system that proposed an image processing technique to achieve their purpose using shape, texture and color in the feature extraction. They achieved it by implementing steps they were using image segmentation, extracting the feature extraction from an image (shape, texture, and color), then they combine the features, and, in the end, they classify the apple disease using a multi-class (SVM) support vector machine. Their output was to classify if it is diseased or normal class. Their proposed approach achieved up to 96% accuracy. Assunc et al., [17] proposed a system that they had a problem with the number of images to use and to prepare the model with, so they proposed an efficient and tiny convolution neural network that can work in mobile devices. Their objective was to help the farmer to have a device that helps him while working to classify the healthy and diseased peach fruits. Their model achieved the F1 score of 96% accuracy. The model was capable to compute every class correctly and not be able to misclassify any of the corrupt ones. This paper was able to achieve a little model to work with small data and deliver the farmer a trusted device that couldn't misclassify any type of disease. Their dataset represents four classes they were (healthy, Rot, Mildew, and Scab) with a combined number of images 313 in the training phase grabbed 249 images and, in the testing, phase took 64 images.

III. IMPLEMENTATION FRAMEWORK

The structure overview in Fig 2 shows that the user takes a photo of fruit utilizing a camera, then pre-processes image augmentation as removing any unnecessary image background or any noise and optimizing it, applying the normalization process to the image, completely connecting the fruit layers to a machine learning model that is trained on the three types of the disease of fruit and four classes including the healthy part. A dropout is involved and then classify them and finally, the system shows the user a report whether the fruit for example, if it has either black spot, citrus scab, citrus canker, and is in good condition and with what percentage It is infected by.

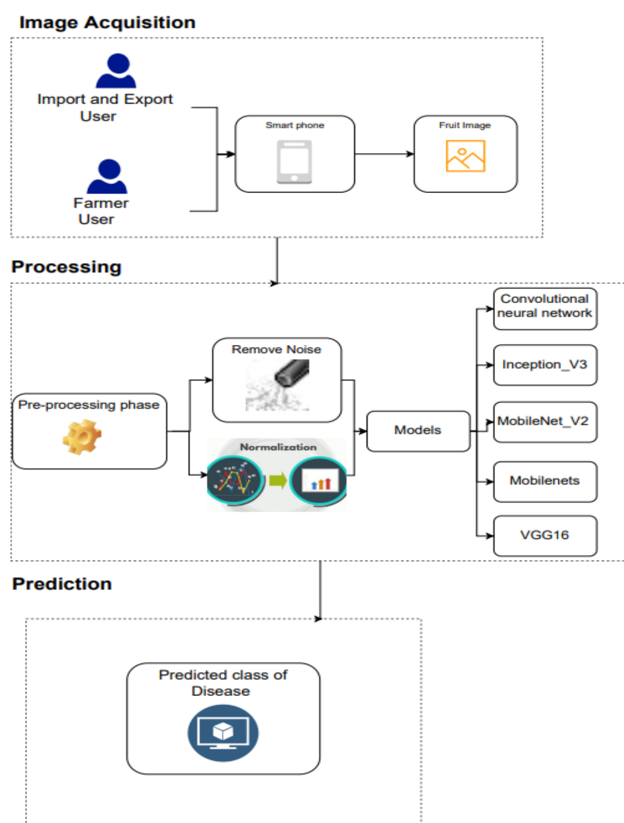


Fig. 2. System overview

A. Dataset

There are two datasets for two various fruits in the fig 3 , 4. The fruits consumed were citrus and apple. In the citrus dataset, it is obtained from the Kaggle website for various datasets it consists of two primary categories they are (test and train). The train part consists of four classes, and they are (black spot, citrus scab, citrus canker, and healthy) with an aggregate number of images of 100 to work with. The test part consists also of four classes same as the train section in the citrus dataset but with an aggregate number of images of 50 to work with. The apple dataset was acquired from the Kaggle website also has two samples (train and test). The train part consists of four classes, and they are (Blotch apple, Normal apple, Rotten apple, and Scab apple) with a combined number of images 4500 to work with. The test section consists also of four classes same as the train section in the apple dataset but with a combined number of images 500 to work with. All the images are in the JPEG format in the two datasets. The apple dataset size is 1.06 GB, while in the citrus dataset it is 2.31 GB. There is some sample of the apple 3 and citrus 4 datasets.

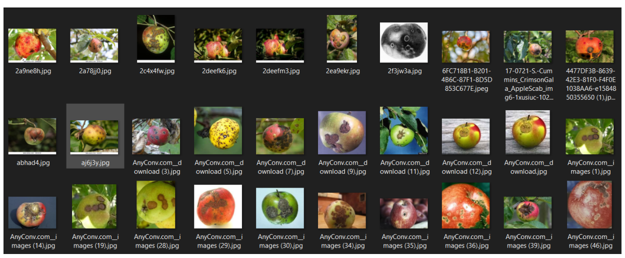


Fig. 3. Apple Dataset samples

Fig. 4. Citrus Dataset samples

TABLE I
NUMBER OF IMAGES PER CLASS IN APPLE DATASET

Disease	Images Number
Bloch	1250
Scab	1250
Rotten	1250
Healthy Apples	1250
Total Images	5000

B. Model Approaches

This paper used five various types of Transfer Learning models (CNN, VGG16, Inception-V3, MoblieNet-v2, moblienets) an observation for the difference between each transfer learning technique and their accuracy.

1) CNN: A CNN model, [18] is used to approach this problem as it has been extremely useful and accurate in the prediction of an illness CNN model is a convolutional neural network model. Using a data-set of four various classes of apple disease which are Blotch apples, Normal apples, Rotten apples, and Scab apples and citrus diseases dataset that contains four classes black spot, citrus scab, citrus canker, and healthy. To apply the CNN model architecture to images of (320 * 320). The model includes 9 layers (experimental, Conv2D, MaxPooling2D, Flatten, Dense) after eliminating the top layer of the model plus the normalization phase. A 9 layers used which are respectively 1 Conv2D Layer, 1 Max poolting 2D Layer, 1 Conv2D Layer, 1 Max poolting 2D Layer, 1 Conv2D Layer, 1 Max poolting 2D Layer, Flatten Layer, dense, and dense In the end, used 2 Dense Layers wit total parameters 13,131,557 in fig 5.

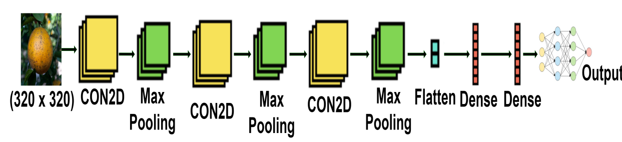


Fig. 5. System Architecture for CNN Model

2) Inception-V3: Another model which is the Inception-V3 model, [19] is used to approach this problem. The Inception-V3 model architecture was applied to images of (224 * 224). The model includes four layers which contain dense layers. To classify for a more encouraging result, it is preferred to use Dense layers.

3) VGG16: Another Model that is being popularly used efficiently is the VGG16, [20] in common is a convolutional neural organization model. It efficiently was a notable model that was equitably distributed at ILSVRC2014. Without topmost layers to properly apply VGG16 to images of size (320 * 320). The model maintains four layers. The VGG16 was integrated with an SGD streamlining the agent and unfortunately Categorical Cross-entropy.

4) MoblieNet-v2: Another developed model which correctly is the MoblieNet-v2 model, [21] used to carefully approach this economic problem. By applying the MoblieNet-v2 model architecture to images of (224 * 224). The model maintains four layers which consist of Dense layers. To classify for a more encouraging result, it is preferred to use Dense layers.

5) Moblienets: Another developed model which is the moblienets model, [22] to approach this economic problem. By properly applying the moblienets model architecture to images of (224 * 224). The model maintains four layers which consist of the Dense layer. To classify for a more encouraging result, it is preferred to use Dense layers.

IV. Experiment and Result

1) Experiment 1: An experiment is being applied with five developed models using apple data-set, that includes two distinct sections (train, test) within them contains four classes which are (normal apples, blotch apples, Rot apples, and Scab Apples) diseases that every class contains a number of images respectively (1250, 1250, 1250 and 1250). step two which removes any noises from the image that is not necessary by decreasing the number of parameters that is used in a training phase and increasing the accuracy number it has been divided into 75% for the trained data and 25% for the tested data, with using batch size equal to 32, image height equals to 320 and image width equals to 320 then the prediction phase is applied as a photo is being inserted to the model of the apple fruit and then and the model was accurate to detect the following diseases (Blotch, Scab and Rotten) by (100.00%, 88.41%, 99.30%) and the Healthy apple by 99.98%. The developed CNN model remains steadfastly the most accurate predictable model of all developed models. CNN Model achieved overall accuracy up to 96.97% compared to the other models as in the table III.

TABLE II
ACCURACIES OF DIFFRENT MODELS USING THE APPLE DATASET

Models	Accuracy Score
Inception_V3	88%
MoblieNet_v2	80%
moblienets	83%
VGG16	78%
CNN	96.97%

2) Experiment 2: Another experiment is applied with CNN developed model using citrus data-set, that includes two distinct sections (train, test) within them contains four classes which are (black spot, citrus scab, citrus canker, and healthy) diseases that every class contains a total number of images 150. step two which removes any noises from the image by applying AUTOTUNE to the image then NORMALIZE the image. Also decrease the number of parameters that is used in a training phase and the training split is being divided into 75% for the trained data and 25% for the tested data, with using batch size equal to 32, image height equals to 320 and image width equals to 320 then the prediction phase is applied as a photo is being inserted to the model of the citrus fruit and then and the model was accurate to detect the following classes (black spot, citrus scab, citrus canker, and healthy) by an overall accuracy equals to 99.16% wish exceeds the accuracy of [1] using the same dataset.

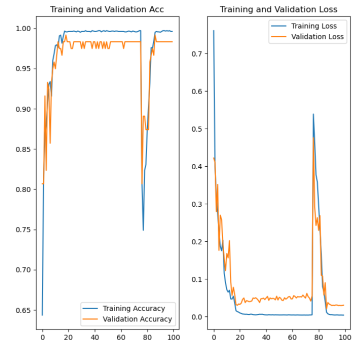


Fig. 6. Training and validation accuracy and loss For CNN model on Citrus dataset

V. Discussion

In our proposed solution as in Fig 2, the user takes a photo of the fruit using a mobile phone, then pre-processes image augmentation as removing any unnecessary image background or any noise and optimizing it, apply the feature extraction process, then fully connect fruit layers to a transfer learning model that has been trained on the three types of the disease for apples in experiment one and also three types of diseases for the citrus in experiment two then dropout then classify them and finally, the system shows the user a report whether the apple has either blotch, scab or rotten disease or it is a normal apple and same for the citrus experiment and with what accuracy the model is sure while classifying apple or citrus diseases. The developed CNN model was the best model of all the models. Because of using 9 layers which are respectively 1 Conv2D Layer, 1 Max pooling 2D Layer, 1 Conv2D Layer, 1 Max pooling 2D Layer, 1 Conv2D Layer, 1 Max pooling 2D Layer, Flatten Layer, two Dense Layers In the end, used 2 Dense Layers with total parameters 13,131,557. Which made the CNN model more accurate and be able to detect without producing errors in the testing phase.

VI. Conclusion

This paper proposes and objectively evaluates an image management system for disease detection and economic grouping in natural apple products. The proposed approach consists essentially of three stages. The first step is processing the image. In the second step, an experiment is being preformed by employing five models and comparing the accuracy between each model on an apple dataset. Another experiment is performed using a CNN classification method in experiment two and our model achieved better accuracy than [1]. The third step is preparation and grouping and predictions. Explicitly, in our data set three types of apple diseases used and they are: Blotched Apples, Rotten Apples, Scabbed, and Apples as a discourse investigation in the first dataset. In the second dataset, it has been employed in it three types of citrus diseases: black spot, citrus scab, citrus canker. Our test results show that the proposed arrangement can in principle maintain

TABLE III
GAP ANALYSIS

References	Algorithm	Classifier	NO. Of Diseases	Accuracy	No. Of Classes	Dataset	Dataset Size
[3]	CNN	BPNN	3	99.17%	4	Local Market	3200
[4]	Decision Tree and K-means	MSVM and BDT	3	96%	4	Collected Images of Apple Disease	2000
[9]	K-means	GA, SVM and FCM-NPGA	Not Specified	98%	2	Not Specified	Not Specified
[1]	RFANN	HT-RFANN, M-SVM, W-KNN, EBT, DT and DT	3	96.93%	4	Kaggle	150
[2]	RPN	Not Specified	3	94.37%	3	Kaggle	477
Our Result	CNN	CNN	3	99.16%	4	Same Citrus Dataset in paper [1]	150

the programmed detection and disease characterization of natural apple products and citrus. In terms of our research, it tracked the typical apples and citrus that are effectively detectable in unhealthy apples and citrus. In experiment one, the CNN model achieved 96.96% and in experiment two achieved 99.16%

REFERENCES

[1] C. Senthilkumar and M. Kamarasan, "An optimal weighted segmentation with hough transform based feature extraction and classification model for citrus disease," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020, pp. 215–220.

[2] H. M. . P. M. Syed-Ab-Rahman, S.F., "Citrus disease detection and classification using end-to-end anchor-based deep learning model," p. 927–938, 2022.

[3] A. G. Alharbi and M. Arif, "Detection and classification of apple diseases using convolutional neural networks," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, 2020, pp. 1–6.

[4] P. K. Sujatha, J. Sandhya, J. S. Chaitanya, and R. Subashini, "Enhancement of segmentation and feature fusion for apple disease classification," in *2018 Tenth International Conference on Advanced Computing (ICoAC)*, 2018, pp. 175–181.

[5] A. Prabhu, K. V. Sangeetha, S. Likhitha, and S. Shree Lakshmi, "Applications of computer vision for defect detection in fruits: A review," in *2021 International Conference on Intelligent Technologies (CONIT)*, 2021, pp. 1–10.

[6] L. Zixi, H. Jinrong, Z. Yuanyuan, C. Yanan, and S. Aqing, "Nondestructive detection of apple mouldy core disease based on unbalanced dielectric data," in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 466–470.

[7] S. R. Dubey and A. S. Jalal, "Detection and classification of apple fruit diseases using complete local binary patterns," in *2012 Third International Conference on Computer and Communication Technology*, 2012, pp. 346–351.

[8] H. Ayaz, E. Rodríguez-Esparza, M. Ahmad, D. Oliva, M. Pérez-Cisneros, and R. Sarkar, "Classification of apple disease based on non-linear deep features," *Applied Sciences*, vol. 11, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/14/6422

[9] W. Zhang, J. Hu, G. Zhou, and M. He, "Detection of apple defects based on the fcm-npga and a multivariate image analysis," *IEEE Access*, vol. 8, pp. 38 833–38 845, 2020.

[10] B. J. Samajpati and S. D. Degadwala, "Hybrid approach for apple fruit diseases detection and classification using random forest classifier," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1015–1019.

[11] M. A. Khan, M. I. U. Lali, M. Sharif, K. Javed, K. Aurangzeb, S. I. Haider, A. S. Altamrah, and T. Akram, "An optimized method for segmentation and classification of apple diseases based on strong correlation and genetic algorithm based feature selection," *IEEE Access*, vol. 7, pp. 46 261–46 277, 2019.

[12] Yogesh, P. Singhal, A. K. Dubey, and A. Goyal, "A comparative approach for image segmentation to identify the defected portion of apple," in *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2017, pp. 604–608.

[13] K. Swetha, V. Venkataraman, G. Sadhana, and R. Priyatharshini, "Hybrid approach for anthracnose detection using intensity and size features," in *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, 2016, pp. 28–32.

[14] D. Li, M. Shen, D. Li, and X. Yu, "Green apple recognition method based on the combination of texture and shape features," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017, pp. 264–269.

[15] P. Jolly and S. Raman, "Analyzing surface defects in apples using gabor features," in *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 2016, pp. 178–185.

[16] S. R. N. M. Ayyub and A. Manjramkar, "Fruit disease classification and identification using image processing," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 754–758.

[17] E. Assunção, C. Diniz, P. D. Gaspar, and H. Proença, "Decision-making support system for fruit diseases classification using deep learning," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, 2020, pp. 652–656.

[18] C. Huang and J. Wu, "Multi-task deep CNN model for no-reference image quality assessment on smartphone camera photos," *CoRR*, vol. abs/2008.11961, 2020. [Online]. Available: https://arxiv.org/abs/2008.11961

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[20] H. Qassim, D. Feinzimer, and A. Verma, "Residual squeeze VGG16," *CoRR*, vol. abs/1705.03004, 2017. [Online]. Available: http://arxiv.org/abs/1705.03004

[21] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018. [Online]. Available: http://arxiv.org/abs/1801.04381

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

4.4.6 Dental implant recognition and classification with Convolutional Neural Network

Andrew Ayman Edward on publishing his paper in the Faculty of Computer Science, which was published in 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC-22). The paper entitled “Dental implant recognition and classification with Convolutional Neural Network" proposed a system that extracts the features from the dental implants and categorize them by using Convolutional Neural Network to help the dentists to identify the implants easily and get the company contacts. The project was part from the cross disciplinary research work that was done with colleges from the faculty of Dent at MSA towards the digital transformation. [4]

Dental implant recognition and classification with Convolutional Neural Network

Andrew Ayman
Faculty of Computer Science
October University for Modern Science and Arts
Faculty of Computer Science
University of Greenwich
Giza, Egypt
andrew.ayman1@msa.edu.eg

Shereen W.Arafat
Oral & Maxillofacial Surgery Department
October University for Modern Science and Arts
Faculty of Dentistry
Giza, Egypt
swagdy@msa.edu.eg

Ahmed M. Hossam Eldin
Oral Radiology Department
October University for Modern Science and Arts
Faculty of Dentistry
Giza, Egypt
ahossam@msa.edu.eg

Ayman Atia
HCI-LAB,
Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for Modern Science and Arts
Giza, Egypt
aezzat@msa.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781715

***Abstract*—The dental implants market was worth over USD 7,222 million in 2020, and it's predicted to grow to USD 11,801 million by 2026, with a compound annual growth rate of 8.6 % over the forecast period of 2021-2026. This demonstrates that the number of dental implants will dramatically increase by 2026. These contributions will create a problem for dentists all over the globe in identifying the type of implant and getting the manufacturer's company contacts. The authors compared three CNN models: VGG16, Xception, and ResNet50V2, and applied transfer learning to train the models on them. The authors also proposed system architecture to simplify the identification process. The presented system identified four types of implants with acceptable accuracies.**

***Index Terms*—Convolutional Neural Networks, Image processing, Transfer Learning, Dental Implants, Data Augmentation.**

I. Introduction

A dental implant technically is an artificial tooth root that's placed into the jaw bone to hold a prosthetic tooth or bridge. Dental implants became a common treatment option for people who have lost one or more teeth due to periodontal disease, injury, or other reasons and who prefer not to wear dentures [1]. However, dental implants might fail to owe to mechanical issues, such as screw loosening, or biological issues, such as peri-implant infections [2]. Dental professionals are increasingly likely to see patients with implant-supported restorations or prostheses as the number of patients with dental implants grows [3] The broad availability of dental implant systems with different designs worldwide presents a challenging problem for dental professionals to detect the inserted implant type by radiographic means without available records. [4], [5] In such cases, precious clinical time is typically spent performing detective work, utilizing whatever information the dentist has, clinical knowledge from colleagues, and assistance from implant manufacturer personnel, among other resources, to help identify the system which is time-consuming and costly [6].

Dental radiography, panoramic radiography, and computed tomography were all used to identify implant systems based on radiographic images. Panoramic radiography pictures have the benefit of being standardized to a certain degree regardless of the patient. The drawback is that the implant forms are ambiguous when they overlap with a shadow or are excessively short or slanted. This may cause misdetection or wrong classification in the results.

[6] Although the pictures used in this algorithm's learning method are square, the original panoramic radiography images are rectangular. As a result, during the learning phase, panoramic radiography pictures are compressed laterally, as are the forms of the implants. As a result, implant details may become unclear, thus lowering learning performance. Cropping the original panoramic image into a square shape that incorporates implants beforehand might improve learning performance.

The problem is that the dentist faces some difficulties in identifying the implant because of multiple problems, closure of the manufacturing company. The patient could have come from another clinic in another city, and the local clinics don't have information about this implant or the patient. so, the

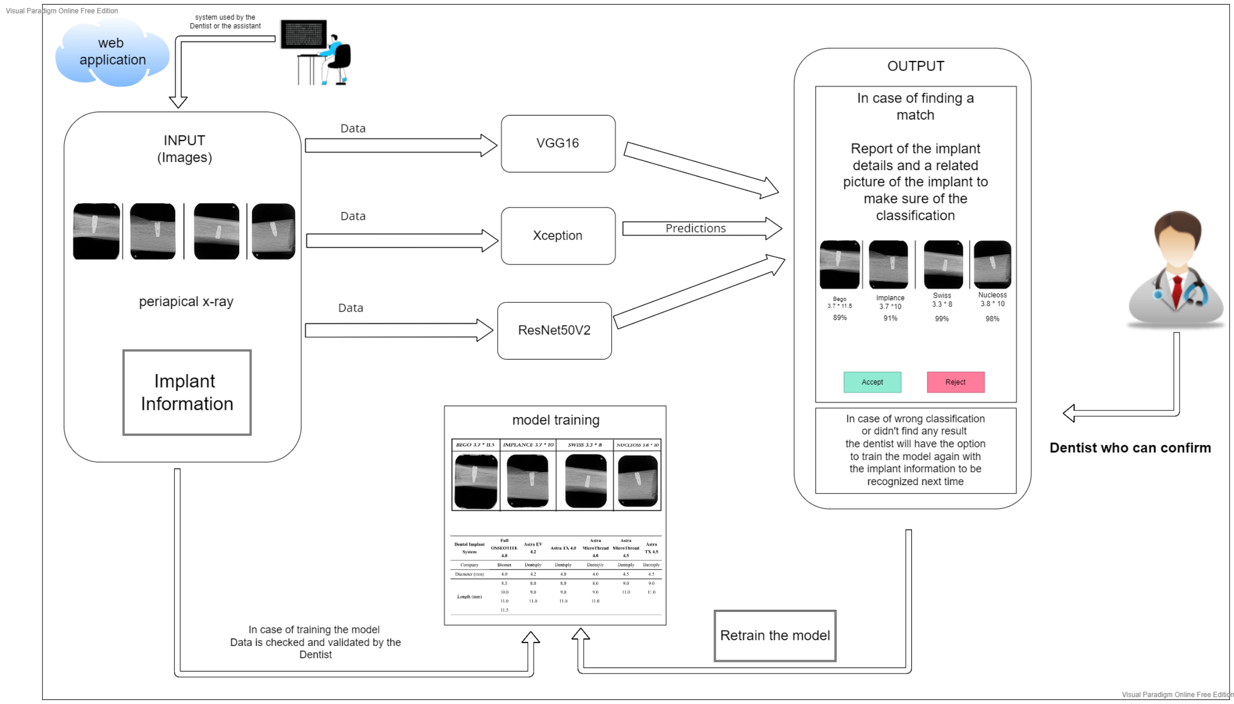


Fig. 1: System Overview

Dentist uses the traditional solution to identify the implant by making a forum that may be online or offline to ask the colleagues about the implant which will take much time to know the type of the implant. The second solution is to replace the entire implant with a new one which is very costly and painful to the patient. This paper will show that CNN was able to identify four implant types with high accuracy, and this will help doctors identify the dental implants easily and know their specifications so that they can fix the problem without replacing the implant. This will reduce the cost of the new implant and make the process of identifying the implant faster.

II. Related work

A. Teeth detection and segmentation

Using various forms of radiography pictures such as bitewing, periapical, and panoramic imaging, several methods for tooth segmentation have been created in the field of dental informatics. Mircea Paul Muresan et al [7] utilized the CNN to automatically recognize and segment the teeth in a panoramic X-Ray and diagnose dental abnormalities like tooth fillings. They presented one solution and two additional approaches with an accuracy of 0.89 percent, 0.87 percent, and 0.68 percent respectively.

According to specific prior domain information, H. Chen et al [8] offered three post-processing strategies to enhance the baseline quicker R-CNN. To eliminate overlapping boxes identified by a faster R-CNN for the same tooth, a filtration algorithm is first built. Second, to find out whether there are any missing teeth. The next step is to employ a neural network model. Finally, a rule-based module based on the tooth numbering scheme is presented. The results show that both precisions and recalls are greater than 90 percent, and the mean value of the IOU between detected boxes and ground facts is also greater than 90 percent.

B. classification with deep neural network

Shintaro Sukegawa et. al [9] employed three CNN structures in their research: a basic CNN with three convolution layers, VGG16, and VGG19. To locate and distinguish the dental implant from the panoramic dental X-Ray, transfer learning and fine-tuning were used. They created an implant type report that included the implant name and model number. With an accuracy of 0.935 percent for the VGG16 and 0.927 percent for the VGG19, they were able to recognize eleven different types of dental implants. Their work does not automatically partition the X-Ray.

T. Takahashi et al. [6] built an object detection method using (Yolov3) algorithm, TensorFlow, and Keras deep-learning frameworks to identify the six implant systems from X-Rays by three manufacturers. They applied the algorithm to a dataset of 1282 panoramic radiograph images with implants, and the result was that at least 240 instances of each implant system were detected in the panoramic radiographic images, with MK III/IIIG (1919 instances) being the most common and Genesio being the least common (240 instances).

Deep neural networks can recognize four distinct types of implants on intraoral radiographs, according to Jee Hwan Kim et al [10]. A confusion matrix was used to determine the accuracy, precision, recall, and F1 score for each network. The test accuracy of all five models was greater than 90 percent. The accuracy of SqueezeNet and MobileNet-v2 was roughly 96 percent and 97 percent, respectively.

III. System Overview

The proposed system is a web application that is divided into three sections as shown in fig 1. First, The input by which an authorized Dentist Radiologist uploads an implant X-Ray and its data through a web interface to let the CNN model train on the new implant to be recognized in the future. The second, Is to recognize the dental implant by uploading the X-Ray to the system where the model will identify the implant and give a detailed report about the implant type, Company name and contacts, and the implant specification. The first part of the processing is to train the model on the new data that is uploaded to the system to be recognized in the future. The second part of the processing is to make the CNN model identify the uploaded X-Ray through the CNN models by extracting features from the X-Ray and categorizing the implant according to the features extracted. Finally, in the output section, after the system is done with the classification by the module, it generates a report which contains the prediction percentage and a similar implant image to be visually compared and verified by the dentist and an option to retrain the module on the implant if he got its information in case of the wrong classification.

IV. Methods

A. Annotation of implants

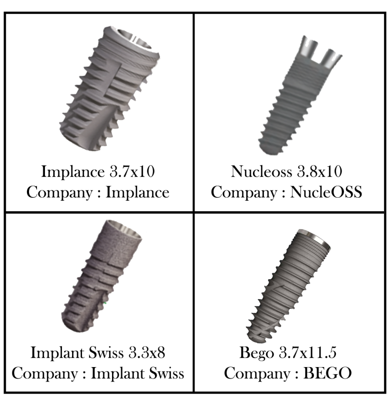


Fig. 2: Implants used in our study

Sixteen implant sites were drilled and prepared in bovine ribs followed by implant placement. The four implant systems used in the current study were the Nucleoss dental implant system, Implant Swiss system, Implance system, and Bego dental implant system as shown in fig 2 different implants for each system were used. The placed implants were used for all subsequent radiographs.

B. Data Collection

The study was conducted with Periapical radiographs acquired using size two VistaScan photostimulable storage phosphor plates (Durr Dental AG, Bietigheim Bissingen, Germany) and an Xgenous DC radiography unit (De Gotzen, Via Roma, Italy) at 70 kV, 8 mA, and 31 cm source-to-object distance. For each implant, different radiographs were acquired in the bovine bone. A single direct exposure resembling periapical with no vertical or horizontal shift, also four other exposures: two with mesial and two with distal shifts. The horizontal shift was made at 10 and 20 degrees from the original exposure, with no vertical angle change. Thus, acquiring five periapical exposures for each implant with different horizontal angles. This was made to capture different views of the implant without affecting its dimensional appearance as shown in fig 3.

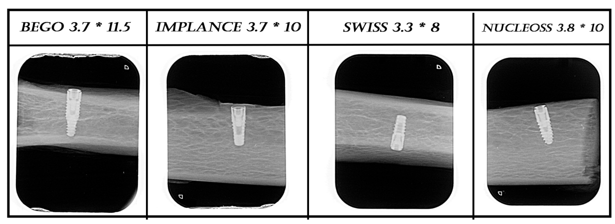


Fig. 3: These are the implant samples and names

C. Performance Metrics

Precision, recall, and F1 score, which measure the relation between the data's positive labels and those given by the classifier, are calculated using the testing dataset and a confusion matrix and determined as follows.

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

TP denotes true positive, FP denotes false positive, FN denotes false negative, and TN denotes true negative. These equations are used in the comparison shown in the tables I,II.

D. Convolution Neural Network

Resnet50V2, Xception, and VGG16 were the three CNN architectures evaluated in this study. The VGG16 offers a weighted-layer depth of 13 and 3 entirely connected layers This model was developed using a dataset of over 14 million photos from 1000 different classes. [11] Xception is a 71-layer convolutional neural network. Google's xception model was developed using over a million photos from the ImageNet database. The network can categorize photos into 1000 different object categories, including mice, Keyboards, pencils, and a variety of animals. As a result, the network has learned a variety of rich feature representations for a variety of pictures. The network accepts images with a resolution of 299 by 299 pixels. The model showed better performance than the Inception V3, although the Xception architecture has the same amount of parameters as Inception V3, the performance increases are due to more effective use of model parameters rather than greater capacity [12]. ResNet50V2 is a modified version of ResNet50 that performs better on the ImageNet dataset than ResNet50 or ResNet101. The propagation formulation of the links between blocks was changed in ResNet50V2. ResNet50V2 is a 50-layer deep convolutional neural network. Despite a large number of layers, the layered architecture made model network training easier, making the layers easier to tune, and this architecture produced better results than standard VGG nets [12].

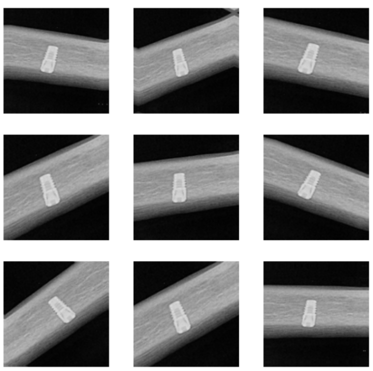


Fig. 4: Data Augmentation

For the several stages of learning done in this investigation, the datasets were separated into 70 percent training and 30 percent Validation, and Testing. we removed the upper pre-trained layers to give us a significant boost in performance [13], Adam was the optimization algorithm utilized for all the CNN models. Transfer learning was used to train the model on the datasets. Contour is applied on the X-Rays to cut the white border of the X-Ray to focus on the implants only without the border noise. Random Flip, Random Rotation, and Random Zoom are used to generate the images in data augmentation. Data augmentation is adding slightly changed copies of current data or creating new synthetic data from existing data to expand the quantity of data available in our study as shown in fig 4. When training a machine learning model, it works as a regularizer and helps prevent overfitting [14]. ModelCheckpoint and EarlyStopping were used also to get the best result from the model while training it and to reduce the chance of overfitting. All the processes were repeated for each CNN architecture.

V. Results

The Categorical crossentropy loss function was utilized to train the CNN models employed in this work. The following tables show the performance of each of the three CNN models examined in this study.

	Xception	ResNet50V2	VGG16
Bego 3.7 * 11.5	1.00	1.00	0.80
Implance 3.7 * 10	1.00	1.00	0.86
Impllant Swiss 3.3 * 8	0.89	1.00	1.00
Nucleoss 3.8 * 10	0.86	1.00	0.86

TABLE I: F1-Score Table of the three models

	Xception	ResNet50V2	VGG16
Bego 3.7 * 11.5	1.00	1.00	0.67
Implance 3.7 * 10	1.00	1.00	0.75
Impllant Swiss 3.3 * 8	1.00	1.00	1.00
Nucleoss 3.8 * 10	0.75	1.00	0.75

TABLE II: This is the Recall Table

As shown in the table I the highest model according to this study was the ResNet50V2 followed by the Xception and then the VGG16. The three models showed reasonable results on a relatively low number of images. The models were conducted on i7 9th Gen, 16GB ram, and GPU Nvidia GTX 1050ti 4GB. The features of the Training, Validation and testing batches are Extracted and saved on a bottleneck file to train the CNN model on them.

After applying the techniques to reduce the chance of overfitting as shown in the Convolution Neural Network section the comparison between the three models showed high accuracy in training and testing. The tarefAccuracy shows the accuracy of training and testing in each model.

	Training Accuracy	Testing Accuracy
Xception	75.00 %	93.00 %
ResNet50V2	81.25 %	100 %
VGG16	75.00 %	93.00 %

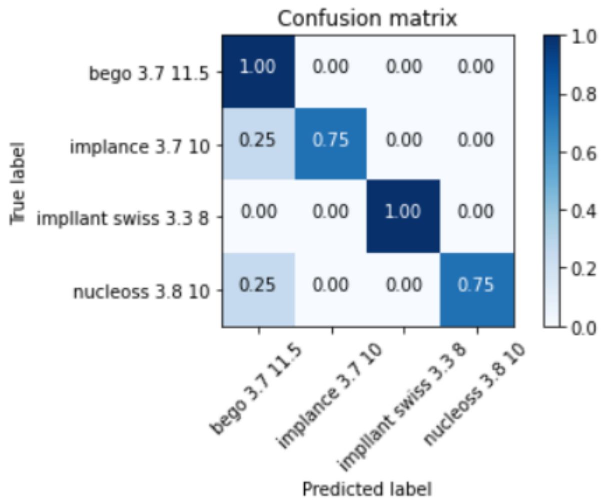
TABLE III: Training and Testing Accuracy

There was confusion in 2 classes in the VGG16 model and confusion in one class in the Xception model because some implants have common features like the connection type as shown in5a.

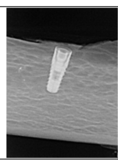
VI. Discussion

Despite the small number of datasets, the three CNNs studied were able to categorize four dental implant systems retrieved from periapical X-ray pictures. These findings were important in using deep learning to categorize dental implant brands from periapical radiographs. CNN was able to do

Normalized confusion matrix



(a) VGG16 Confusion Matrix

Implance	Bego
	

(b) Two classes making confusion

Fig. 5: VGG16 confusion matrix and similar classes

image classification with acceptable accuracy while utilizing relatively limited picture datasets by applying suitable transfer learning and fine-tuning to the pre-trained deep CNN architectures. The VGG16 and the Xception had the weakest categorization results. The results from the ResNet50V2 model were the best after using several approaches to increase the model's accuracy and avoid overfitting. Fine-tuning certain convolutional blocks in the deep CNN layers can improve image classification performance. Deep CNN models based on large natural image datasets using pre-trained deep neural networks are generally good for general image categorization.

The capacity to identify teeth and jaw-related items concurrently is the primary benefit of panoramic radiography [15]. Despite the availability of pictures accessible, little research [16]–[18] has used CNNs to classify and diagnose their findings. Panoramic radiographs were often employed in studies involving diseases of the jawbone [19] . Because panoramic radiographs have varying distortions depending on where they are taken, most diagnoses have relied on periapical radiographic images, whereas CNNs have been employed for tooth-related classifications and diagnoses. [20], [21].In our research, we discovered that CNNs employing periapical radiographic pictures recognize implant systems better than panoramic radiographic images.

The compatibility of dental implants varies depending on the system [22]. Some systems are incompatible with those of other manufacturers, while others are. These elements, as previously stated, have a direct impact on the maintenance of implant prostheses. Patient implant maintenance, on the other hand, will continue as long as the device is in the patient's mouth. It's critical to collect current data and apply what's been learned to the next generation of products. Even though it is difficult to get information on discontinued implant systems, dentists must arrange those systems so that they can readily obtain and respond to implant data that has been gathered thus far.

This study demonstrated that deep neural networks are capable of categorizing the dental implants included in the study. There is potential to apply them to more implants on real datasets from human bone and setups with different devices for image acquisition. Every region of the globe has its own major dental implant systems and radiography technologies. As a result, it is required to first develop an accurate database for each and then use the information to generate an accurate classification using a deep neural network. We expect that cross-sectional investigations conducted by different universities throughout the world will contribute to the development of a more robust dental implant categorization technique based on CNN.

There were three flaws in this study. The first was the limited number of CNN models available. The models VGG16 and Xception, as well as ResNet50V2, were utilized. Deep-learning algorithms with deep and broad layers, as well as those with modified stratification approaches (e.g., VGG19, CapsNet and inception CNN models), are constantly being developed [35]. In the future, it will be important to research this and additional CNN models. Second, all of the X-ray pictures utilized in this study for categorization were obtained with the same panoramic X-ray machine. The picture quality and magnification produced by different panoramic X-ray equipment would differ. As a result, future work will need to estimate the findings of large cross-sectional research encompassing a variety of panoramic radiographs and picture quality. Third, we only looked at periapical X-ray pictures. A more important route for future research would be to create a network that can detect implants using panoramic photos or that can apply algorithms to detect several implants at the same time. and to use this technology through a web interface and deploy it as a web service that can be accessed from anywhere in the world.

VII. Conclusion And Future Work

The ability to recognize dental implants is critical to the treatment procedure. Dentists confront a variety of concerns and obstacles and Identifying the type of implant is one of those obstacles. The identification method, with the use of (CNN) models, will save time and money, as well as lessen the patient's surgical discomfort. To determine whether the deep learning model was superior for classifying dental implants, the scientists explored three techniques. The authors employed a transfer learning strategy. In the experiments, the ResNet50V2 model obtains the highest accuracy, which is the best accuracy among the three models. Future work will focus on improving the stability of our models, obtaining more data pictures to feed our models, and experimenting with new and alternative CNN models.

References

[1] F. Maryam, T. Malek Mohammadi, M. Mohammadi, and R. Gudarzi, "Oral health related quality of life in patients undertaking implant treatments: A review of literature," *Family medicine*, vol. 16, 02 2018.

[2] D. Hashim, N. Cionca, C. Combescure, and A. Mombelli, "The diagnosis of peri-implantitis: A systematic review on the predictive value of bleeding on probing," *Clinical Oral Implants Research*, vol. 29, pp. 276–293, 10 2018.

[3] A. Warreth, "Dental implants: An overview," *Dental update*, vol. 44, 07 2018.

[4] I. Sahiwal, R. Woody, B. Benson, and G. Guillen, "Radiographic identification of threaded endoseous dental implants," *The Journal of prosthetic dentistry*, vol. 87, pp. 552–62, 06 2002.

[5] I. G. Sahiwal, R. D. Woody, B. W. Benson, and G. E. Guillen, "Radiographic identification of nonthreaded endoseous dental implants," *The Journal of Prosthetic Dentistry*, vol. 87, no. 5, pp. 552–562, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022391302000136

[6] T. Takahashi, K. Nozaki, T. Gonda, T. Mameno, M. Wada, and K. Ikebe, "Identification of dental implants using deep learning-pilot study," *International journal of implant dentistry*, vol. 6, p. 53, 09 2020.

[7] M. P. Muresan, A. R. Barbura, and S. Nedevschi, "Teeth detection and dental problem classification in panoramic x-ray images using deep learning and image processing techniques," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2020, pp. 457–463.

[8] H. Chen, K. Zhang, P. Lyu, H. Li, L. Zhang, J. Wu, and C.-H. Lee, "A deep learning approach to automatic teeth detection and numbering based on object detection in dental periapical films," *Scientific Reports*, vol. 9, 03 2019.

[9] S. Shintaro, K. Yoshii, T. Hara, K. Yamashita, K. Nakano, N. Yamamoto, H. Nagatsuka, and Y. Furuki, "Deep neural networks for dental implant system classification," *Biomolecules*, vol. 10, p. 984, 07 2020.

[10] J. H. Kim, N.-E. Nam, J. Shim, Y.-H. Jung, B.-H. Cho, and J. Hwang, "Transfer learning via deep neural networks for implant fixture system classification using periapical radiographs," *Journal of Clinical Medicine*, vol. 9, p. 1117, 04 2020.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," vol. 9908, 10 2016, pp. 630–645.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," pp. 4510–4520, 06 2018.

[14] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[15] A. Ribeiro, R. Keat, S. Khalid, S. Ariyaratnam, M. Makwana, M. do Pranto, R. Albuquerque, and L. Monteiro, "Prevalence of calcifications in soft tissues visible on a dental pantomogram: A retrospective analysis." *Journal of stomatology, oral and maxillofacial surgery*, vol. 119 5, pp. 369–374, 2018.

[16] M. Murata, Y. Ariji, Y. Ohashi, T. Kawai, M. Fukuda, T. Funakoshi, Y. Kise, M. Nozawa, A. Katsumata, H. Fujita *et al.*, "Deep-learning classification using convolutional neural network for evaluation of maxillary sinusitis on panoramic radiography," *Oral radiology*, vol. 35, no. 3, pp. 301–307, 2019.

[17] Y. Ariji, Y. Yanashita, S. Kutsuna, C. Muramatsu, M. Fukuda, Y. Kise, M. Nozawa, C. Kuwada, H. Fujita, A. Katsumata *et al.*, "Automatic detection and classification of radiolucent lesions in the mandible on panoramic radiographs using a deep learning object detection technique," *Oral surgery, oral medicine, oral pathology and oral radiology*, vol. 128, no. 4, pp. 424–430, 2019.

[18] J.-S. Lee, S. Adhikari, L. Liu, H.-G. Jeong, H. Kim, and S.-J. Yoon, "Osteoporosis detection in panoramic radiographs using a deep convolutional neural network-based computer-assisted diagnosis system: a preliminary study," *Dentomaxillofacial Radiology*, vol. 48, no. 1, p. 20170344, 2019.

[19] P. Chu, C. Bo, X. Liang, J. Yang, V. Megalooikonomou, F. Yang, B. Huang, X. Li, and H. Ling, "Using octuplet siamese network for osteoporosis analysis on dental panoramic radiographs," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 2579–2582.

[20] J.-H. Lee, D.-h. Kim, S.-N. Jeong, and S.-H. Choi, "Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm," *Journal of periodontal & implant science*, vol. 48, no. 2, pp. 114–123, 2018.

[21] J.-H. Lee, D.-H. Kim, S.-N. Jeong, and S.-H. Choi, "Detection and diagnosis of dental caries using a deep learning-based convolutional neural network algorithm," *Journal of dentistry*, vol. 77, pp. 106–111, 2018.

[22] M. Karl and A. Irastorza-Landa, "In vitro characterization of original and nonoriginal implant abutments." *International Journal of Oral & Maxillofacial Implants*, vol. 33, no. 6, 2018.

4.4.7 Extractive Summarization of Scientific Articles

Rana Reda Waly on publishing a Conference paper on her graduation project, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The paper entitled “Extractive Summarization of Scientific Articles" proposed a model that generate a summary of a given Scientific Article automatically according to a new approach in the Extractive Summarization concept. And this will be very useful, especially after the gigantic number of Scientific Articles in any field. This automated model was capable of achieving high results compared to more complex models that used the same dataset, which is the "CL-SciSumm 2019" dataset. [22]

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781726

Extractive Summarization of Scientific Articles

Rana Reda Waly
Faculty of Computer Science
MSA University
Giza, Egypt
rana.reda2@msa.edu.eg

Wael H. Gomaa
Faculty of Computer and Artificial Intelligence, Beni-Suef University
Faculty of Computer Science, MSA University
Egypt
Wahassan@msa.edu.eg

***Abstract*—Scientific articles are continuously growing on a daily basis, which makes it difficult to keep up with the latest updates. Hence, this problem can be solved by summarizing these articles. The methodology used aims to summarize single documents and apply the extractive approach to generate the summary using GloVe as an embedding technique. This methodology was applied to the CL-SciSumm 2019 dataset with results of a 0.124465 F-score measure of ROUGE-2.**

***Index Terms*—summarization, scientific articles summarization, CL-SciSumm 2019, Extractive Approach Summarization.**

I. Introduction

The number of internet resources has been increasing rapidly on a daily basis, and these resources may now be found in a variety of formats, including documents, books, news, stories, scientific publications, websites, social media blogs, and more. As a result, the amount of time people take to find the information they require has increased.Users are unable to read all of the retrieved search results, which may contain duplicates or irrelevant information. Hence, summarizing and shortening textual data has become a necessary and crucial task in Natural Language Processing (NLP) [1]. Automatic Summarization of Scientific Articles is a well-known problem in NLP. In the Scientific Articles summarization problem, we aim to reduce the time to review the full article to be an informative summary which gives a meaningful preview about the information within the article. The Scientific Articles structure is what makes our task a difficult one. As There's a huge difference between Scientific Articles summarization and generic text summarization which are [2]:

1- Scientific articles have a defined structure which usually includes an abstract, introduction, background, related work, methodology/approach, results, and conclusion.

2- Scientific articles are significantly lengthier than generic text documents. As a result, one of the biggest concerns surrounding this task is the lack of gold-standard summaries of scientific articles.

Usually there are two ways of generating a summary for the scientific articles. The first is single document summarization, in which each input article is given a summary. The second is multi-document summarization, in which different articles but at the same time related ones are summarized by giving the main topic and essential information. One of the most important questions that everyone may have is why do we need to summarize an article? Why isn't the article abstract sufficient if one is available? There are numerous reasons why summarising an article is vital and why the abstract is insufficient. To begin with, the information provided by the author in the abstract does not always correspond to the substance found in the entire text. Second, the abstract is clearly written by the author, indicating that the author has a personal opinion about his work. As a result, the author may be prejudiced towards his own characteristics and work. Finally, and most importantly, the abstract does not show off the paper's impacts or contributions; rather, it just indicates what the author wants to emphasise and focus on more.

Human Communication? Computer Communication? Natural Language? Natural Language Processing? Can all this expression be related to each other? [3] Humans can communicate with each other through lots of methods talking, listening, sign languages, gestures, etc. and many other different ways, this is called Natural Language which is text. [4] As for Natural Language Processing it is the process of deriving computers to learn and understand Natural Language text to be able perform required tasks. Therefore, NLP take a major role in lots of applications as chatbot, search engines, sentiment analysis, speech recognition, spell checking, information extraction, text summarization and many more. Natural language understanding (NLU) and natural language generating (NLG) are the two components of NLP.

1. Natural language understanding (NLU) is a subset of natural language processing that uses syntactic and semantic analysis of text and speech to identify the meaning of a sentence. NLU also generates a relevant ontology, which is a data structure that defines the connections between words and sentences.

2. Natural language generation (NLG) is the process of writing a human-readable response based on some data input. This text can also be converted into a spoken format using text-to-speech services. Text summary capabilities are also included in NLG, which generates summaries from in-put documents.

Our interest is in Automatic Summarization Generally there are two approaches that are used in generating summaries, Extractive and Abstractive approaches [5].

A. Extractive Summarizer

This means it extracts the most important sentences from the input and creates a summary of these input sentences as in figure 1.

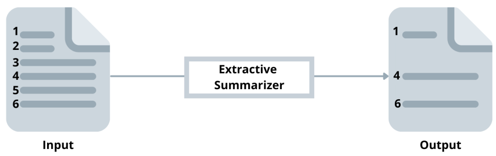


Fig. 1: Illustration of the Extractive Approach Summarizer

B. Abstractive Summarizer

This means it re-represents the input sentences and produces an output summary with sentences that differ from the original input sentences as in figure 2.

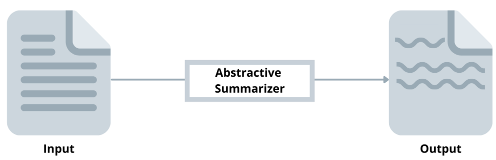


Fig. 2: Illustration of the Abstractive Approach Summarizer

C. General Architecture of Automatic Summarization

Although the different approaches and algorithms and methods introduced to generate a summary, they all share a general architecture as in figure 3 [5].

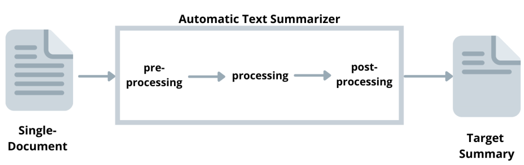


Fig. 3: General Architecture of Automatic Summarizer [5]

The remaining sections will be organized as follows. Related Work section that shows the previous work in the scientific article summarization. Dataset section that shows the dataset used and its details. Methodology that shows the proposed solution used in this paper for solving the problem. Experiments and results section that shows the experiments conducted and the results of these experiments. Discussion section and finally The Conclusion and Future work section.

II. Related work

The author in [6] proposed a SDS (Scientific Document Summarization) system, for summarizing scientific documents due to its continuous growth. The author used a mixture of two datasets on the Computational Linguistic domain, which are CL-SciSumm 2016 and CL-SciSumm 2017. The author proposed system is divided into two phases: Phase 1: Extraction of reference Text-spans which means we are only interested about the sentences in the reference paper (RP), then using similarity to compare sentence in the citation context with the ones in the reference paper and then selecting top 5 similar sentences for each sentence ordering them with respect to the first appeared in the RP. Phase 2: Generating the summary and its optimization, where the input in this phase will be the sentences collected from phase 1. MOOSciSumm (Multi Objective Optimization SciSumm) base is the differential evolution DE framework, step 1 is to begin with random summary solutions in the binary space which we can call them population as each solution represents a possible solution, then evaluate the quality of each possible solution using an objective function. Then step 2 after the values in the objective function is deliberated.

The approach the author represented in [7] is a framework for summarizing of scientific articles using citation of text in the article and find it's relevant in the cited paper. The author presents three approaches for the cited texts which are query reformulation, word embeddings and supervised learning. After that the author trained a model to know the discourse facets for every citation. Lastly, the output is a summary of scientific articles based on the discourse facets of each citation texts and its alternative contexts. The author presents its approach to a combination of two datasets a Biomedical and Computational Linguistic dataset which are CL-SciSumm 2016 and TAC 2014.

The author in [8] tried to generate a summary that is as much as similar to the abstract of the article. The author approach was applied in a collection of a biomedical domain scientific articles.The author presents its approach using articles from PubMed dataset, it's a Biomedical dataset that contains 9978 articles collected from the original dataset. The methodology of the author was using a combination of two ranking algorithms which are TextRank and WordRank to get the best results of them both rather than using a single method.

The [9] represented the Poli2Sum approach for solving the Cl-SciSumm 2019 shared task. Hence, The author presents its approach using the CL-SciSumm 2019 dataset. The methodology used by the author is represented in the Poli2Summ system and its architecture consists of the following stages: 1. Parsing and preprocessing: preparing the raw data to be used in the next stage. 2. Cited text span identification 3. Citation classification 4. Reference paper summarization: in this step the author ranks the cited text spans that were generated from stage 3, the system uses supervised models to predict the level of the text spans based on what it learned from the training set.

The author in [10] represented the NaCTem-UoM (National Center for the Text Mining- University of Manchester)system for solving the CL-SciSumm 2019 shared task. The methodology of the system in the summarization task (task 2), the authors transacted with the summarization problem as a classification problem, they divided it into two classes included or not-included, then those sentences are to be ranked, from the obtained ranked list the sentences are transferred to the final summary list taking into consideration that their are no trigram overlapping between the sentences in the final summary list and the ones to be added to them from the ranked list.

The author in [11] represented the IRTM-NJUST system for solving the CL-SciSumm 2019 shared task. The methodology of the system in the summarization task is that Presentation generation can be divided into two steps. First is to group sentences into different clusters. Second is using ranking features to extract sentence from each cluster. We use linear sum of Jaccard, IDF and TF-IDF similarities to rank sentences. We want to split identified text span into groups based on some logical order. Since keywords can represent more meaningful information, we extract keywords and calculate the Jaccard similarity with abstract. Then, we choose the first sentence from each group based on the ranking score to build summary until the length of summary reaches 250 words.

III. Dataset

A. Dataset Overview

The chosen dataset is the CL-SciSumm 2019 where SciSumm is a shared task in the automatic summarization of scientific articles. This dataset has many different versions which are CL-SciSumm 2016, CL-SciSumm 2017, CL-SciSumm 2018 and CL-SciSumm 2019. The reason for choosing this version is because it has larger number of documents than the other ones. In addition to that this version of the data set each document has it's summary (human summary) so I can compare the generated one (my summary) with it.

TABLE I: Dataset Structure

Point	Description
Dataset Name	CL-SciSumm 2019
Dataset Domain	Computational Linguistic
Dataset Source	Downloaded
Dataset Size	1008 article
Dataset Format	XML Files and JASON Files

B. Dataset Details

As mentioned in the above table, this dataset consists of a 1008 folders each folder contain the following: 1) XML File that contains the scientific article. 2) JASON File That contains the annotated cited sentences. 3) Summary Folder that contains the Gold-summary (Human-summary).

IV. Methodology

As we mentioned before any Automatic Summarization generally have the general architecture as in figure 4. So the proposed solution for the problem will also follow the general architecture manners. It starts with the pre-processing phase followed by processing phase and lastly followed by post-processing phase. So now let's discuss these 3 phases in details.

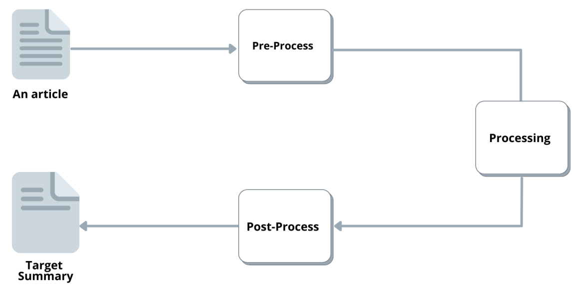


Fig. 4: General Architecture of the proposed solution

A. Data Cleaning and Preparation

Data cleaning and preparation is an important process for detecting and correcting corrupted or inaccurate records in the data Since the dataset cl-scisumm was in the xml format Which were difficult to work with and iterate over, we changed the presentation of the data from XML files to CSV files that was more easier to work with and iterate over as the header tags that holds the sections of the paper are now columns Ana the sentences in the tags are the rows. After transforming the data into csv files, there were observations such as empty xml files And other xml documents that doesn't contain sections.

B. Pre-processing

The pre-processing phase 5 involves applying linguistic approaches to the row data in order to make it easier to work with, which are: 1. Tokenization: tokenizing has two types, sentence tokenization and word tokenization. Where sentence tokenization is the process of breaking down paragraphs into individual sentences and same applies on work tokenization where it is the process of breaking down sentences into words or we refer to it as tokens. 2. Stemming: is the process of getting the root form(stem) of a given word by cutting its ends or beginnings. 3. Lemmatization: is a modified version of the stemming where it takes in consideration the dictionary when getting the root form(lemma) of the word. 4. POS Tags: POS Tags identify weather the word in the sentence is Noun, Adverb, Verb, Adjective, etc. 5. NER: this technique is an information extraction task to identify named entities in the given text. Pre-processing phase is the one of the most challenging phases as its the ground base for the next upcoming phases. In this process our goal is to prepare the row data so that we can work on it easily and benefit from all of it, in other words cleaning the data so it's ready to work with. This phase we use some of the Linguistic Techniques we discussed previously. After this phase our data is ready to be taken to the next phase which is the processing phase.

C. Processing

In phase 2, the Processing phase 6the input will be the pre-processed article that we generated in phase 1. Now we are ready to apply our proposed solution, first we will start by embedding the words in the article, where embedding means

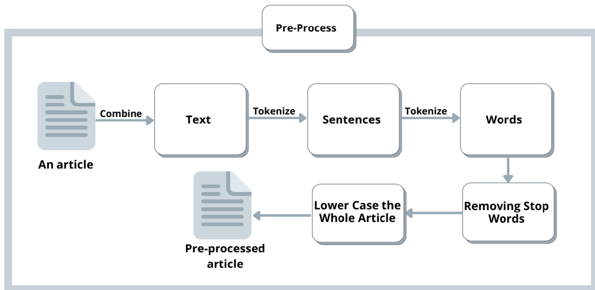


Fig. 5: The Pre-processing stage

representing the words of the article by encoding them with a real value vector. By using a combination of the different embedding techniques as Glove model, Bert model, Fasttext model, etc. then we will conduct the similarity matrix, then the summary will be the top ranked sentences in the similarity matrix.

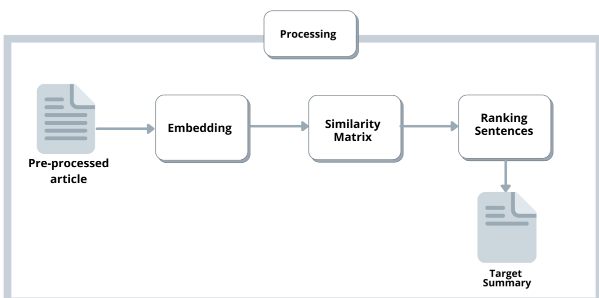


Fig. 6: The Processing stage

D. Post-processing

In this stage, the last stage after generating the summary, the aim here is to make sure of the quality of the generated, such as analyzing the generated summary and see if there is anything that enhances this summary or not and this summary will be the final output.

E. Evaluation

The evaluation method used is the ROUGE Metrices, which is one of the most popular evaluation methods in the summarization task. ROUGE stands for Recall Oriented Understudy for Gisting Evaluation and refers to a set of evaluation criteria for automatic text summarization and machine translations. The metrics, in essence, compare an autonomously generated summary to a reference summary or a set of reference summaries.

It includes the five evaluation metrics listed below: 1) ROUGE-N: The n-gram overlap between the automatically generated and reference summary is calculated. The value of N in n-grams can range from 1 to n, although the cost of computation increases as the number of n increases. The most frequent n-gram metrics are uni-gram and bi-gram.

2)ROUGE-L: Determine the reference and candidate (machine generated) summaries' longest common sub-sequence. Each sentence is handled as a series of words in a summary. Two summaries that have a longer common word sequence are more similar than two summaries that have a shorter common word sequence.

3) ROUGE-S: In the reference summary and candidate summary, ROUGE-S measures the co-occurrences of skip bi-grams. The order of the bi-grams is crucial. ”Any pair of words in sentence order is referred to as a skip bi-gram.” Any arbitrary gaps are permissible.”

4) ROUGE-SU: ROUGE-flaw's is that it only considers bi-grams. If a sentence does not contain any bi-grams that overlap, it will be given no weight. ROUGE-SU is a ROUGE-S extension that also considers uni-grams with bi-grams to solve this problem.

V. Experiment and Results

The experiments were run on a RYZEN 7 9 Gen 16GB RAM system with an Nvidia RTX 3060 6GB GPU. All of the tests were carried out on the same dataset (CL-SciSumm).

1) Experiment One: In the first experiment, first the pre-processing stage of text where removing special characters and stop words the generated text is to be taken for the next stage, the next stage is the word embedding. The embedding technique used is GloVe. GloVe [12] is a technique for generating word vector representations that is based on unsupervised learning. The resulting representations highlight intriguing linear substructures of the word vector space, and the training is based on a corpus's aggregated global word-word co-occurrence statistics. GloVe consists of 6 billion tokens and 400k vocab, it has multiple dimensions to represent the vectors starting from 50d, 100d, 200d and 300d, the one used in the experiments was 300 dimensional vector representation. Then a similarity matrix was created, which kept track of the cosine distance between each sentence and the other sentences. Last but not least, the Text rank method is based on Google's well-known Page Rank algorithm. The Page Rank algorithm is based on the idea that more important websites are more likely to be linked to by other websites. It counts the quantity and quality of links to a page to get a general idea of the website's importance. Similarly, the text rank creates a graph with a set of text sentences as vertices. The edges between the text sentence vertices are based on a measure of similarity. From the similarity matrix generated all the sentences in the scientific article have scores, Then the sentences are sorted in decreasing order of their score. for the first experimented the top 10 sentences were taken to generate our summary. The table II shows the results of this experiments.

2) Experiment Two: For the Second experiment, after analyzing the results of experiment 1, it was noticed a huge difference between the length of the reference summary and the length of the generated summary(our summary), so the

TABLE II: Experiment 1 results

Embedding Technique	ROUGE	Precision	Recall	F-score
Glove	Rouge_1	0.20803	0.443693	0.275603
Glove	Rouge_2	0.067975	0.067975	0.092138

same concept was applied, but instead of taking the top 10 ranked sentences, the top 5 sentences were taken to generate the summary. The table III shows the results of this experiments.

TABLE III: Experiment 2 results

Embedding Technique	ROUGE	Precision	Recall	F-score
Glove	Rouge_1	0.252247	0.335316	0.278336
Glove	Rouge_2	0.075544	0.075544	0.08166

3) Experiment Three: For this experiment the aim was to mix between the two previous experiments by finding the difference in the length of the top 10 and top 5 generated summaries with the reference summary and then the summary that gives us the smallest difference was taken. The table IV shows the results of this experiments.

TABLE IV: Experiment 3 results

Embedding Technique	ROUGE	Precision	Recall	F-score
Glove	Rouge_1	0.249339	0.33956	0.279741
Glove	Rouge_2	0.073937	0.073937	0.082427

4) Experiment Four: In this experiment, from the observations and analysis of the previous experiments, the idea of mixing the length of the generated summary. In this experiment, the summaries generated from top 1 to top 11 ranked sentences for all the documents, after that the highest rouge result for every summary is taken, then finally the final result is the mean of all the documents rouge result. According to the table V and figure 7 that shows the difference in how the results increased.

TABLE V: Experiment 4 results

Embedding Technique	ROUGE	Precision	Recall	F-score
Glove	Rouge_2	0.121481	0.122481	0.124465

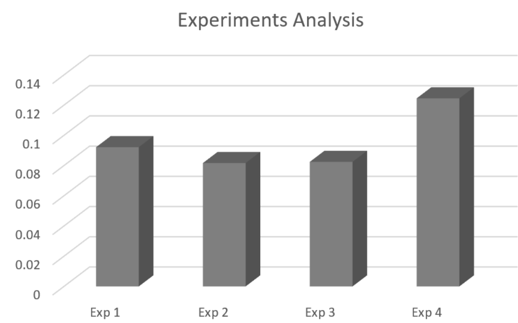


Fig. 7: Analysis of the experiments

VI. Discussion

The findings of the experiments, particularly the third and forth experiment, reveal that each document is unique, and adopting the same summary length to all documents will not be ideal because each document is different in length and structure. However, when comparing the simplicity of the proposed method in this paper to the results, it is clear that combining different generated summary lengths yields better results. On the other hand when comparing our proposed model to the other approaches mentioned in the related work section we can see that the other approaches achieved a good results but the methodology was very complex in compare to the proposed model in this paper. Also the difficulties of evaluating based on a single gold human summary as each human have different perspective, so other summaries can be added by employing paraphrasing tools on this human summary. to generate different replicas of the same single gold summary. As described in the future work, this will be the initial stage to be modified further.

the table VI is the results of the proposed model in compare with other approaches mentioned in the related work section.

TABLE VI: Results

Ref.No.	ROUGE-2
[7]	0.302
[6]	0.27
[10]	0.265
[11]	0.237
[9]	0.218
[8]	0.1670
Our Proposed Model	0.1244

VII. Conclusion and future work

Summarization of Scientific articles is a very important and common task in the Natural Language Processing field. its importance lies in how much time that researchers, students, teachers and many more will save, which will motivate them to read more and gain more information.

Our future work is to apply different embedding techniques, different similarity measurements and also try different machine learning and deep learning algorithms in the extractive approach summarization of scientific articles. In addition to the aim of merging the two approaches the extractive and the abstractive approaches by taking the top ranked sentences from the extractive phase then give it to the abstractive phase, and the abstractive phase will be heavily based on deep neural networks. And in order to enhance the accuracy and the quality of the generated summary, Different data sets will be applied on our model to enhance it and train the model more to make sure of its accuracy and quality.

References

[1] Gregory César Valderrama Vilca and Marco Antonio Sobrevilla Cabezudo. A study of abstractive summarization using semantic representations and discourse level information. In *International Conference on Text, Speech, and Dialogue*, pages 482–490. Springer, 2017.

[2] Simone Teufel and Marc Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.

[3] R Kibble. Introduction to natural language processing. *London: University of London*, 2013.
[4] Zehra Taskin and Umut Al. Natural language processing applications in library and information science. *Online Information Review*, 2019.
[5] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021.
[6] Naveen Saini, Sushil Kumar, Sriparna Saha, and Pushpak Bhattacharyya. Scientific document summarization using citation context and multi-objective optimization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4290–4295. IEEE, 2021.
[7] Arman Cohan and Nazli Goharian. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2):287–303, 2018.
[8] Kaykobad Reza, Rifat Rubayatul Islam, Sadik Siddique, Md Mostofa Akbar, M Sohel Rahman, et al. Automatic summarization of scientific articles from biomedical domain. In *Proceedings of International Joint Conference on Computational Intelligence*, pages 591–602. Springer, 2020.
[9] Moreno La Quatra, Luca Cagliero, and Elena Baralis. Poli2sum@ cl-scisumm-19: Identify, classify, and summarize cited text spans by means of ensembles of supervised models. In *BIRNDL@ SIGIR*, 2019.
[10] Chrysoula Zerva, Minh-Quoc Nghiem, Nhung TH Nguyen, and Sophia Ananiadou. Nactem-uom@ cl-scisumm 2019. In *BIRNDL@ SIGIR*, pages 167–180, 2019.
[11] Shutian Ma, Heng Zhang, Tianxiang Xu, Jin Xu, Shaohu Hu, and Chengzhi Zhang. Ir&tm-njust@ clscisumm-19. In *BIRNDL@ SIGIR*, pages 181–195, 2019.
[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

4.4.8 A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers

Mena Hany Presented a work of natural language processing (NLP), paraphrase detection is a highly common and significant activity. Because it is involved in a lot of complicated and complex NLP applications like information retrieval, text mining, and plagiarism detection. The proposed model finds the best combination of the three types of similarity techniques that are string similarity, semantic similarity and embedding similarity. Then, inputs these similarity scores that range from 0 to 1, to the machine learning classifiers. This proposed model will be benchmarked on "the Microsoft research paraphrase corpus" dataset (MSRP) and from this approach for paraphrase detection problem, the accuracy acquired is 75.78% and F1-Score of 83.01%. [7]

A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers

Mena Hany
Faculty of Computer Science
MSA University
Giza, Egypt
mena.hany@msa.edu.eg

Wael H. Gomaa
Faculty of Computer and Artificial Intelligence, Beni-Suef University
Faculty of Computer Science, MSA University
Egypt
wahassan@msa.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781678

Abstract—In the realm of natural language processing (NLP), paraphrase detection is a highly common and significant activity. Because it is involved in a lot of complicated and complex NLP applications like information retrieval, text mining, and plagiarism detection. The proposed model finds the best combination of the three types of similarity techniques that are string similarity, semantic similarity and embedding similarity. Then, inputs these similarity scores that range from 0 to 1, to the machine learning classifiers. This proposed model will be benchmarked on "the Microsoft research paraphrase corpus" dataset (MSRP) and from this approach for paraphrase detection problem, the accuracy acquired is 75.78% and F1-Score of 83.01%.

Index Terms—Paraphrase Detection, Text Similarity, Machine Learning, Natural Language Processing

I. INTRODUCTION

NLP is the field of making the computer understand, use and produce the human language and it is used in many things like chatbots and plagiarism detection systems. NLP has many topics and ideas for projects that can help people automate their day-to-day lives. One of these topics is paraphrase detection. The ability to determine if two statements have the same semantic meaning or not is known as paraphrase detection and paraphrase detection is one of the basic operations in NLP that is involved in many more complicated NLP operations like information retrieval, text mining and plagiarism detection, although it is a basic operation of many NLP tasks. It is not an easy or trivial problem to solve and many old researchers faced many problems in solving the paraphrase detection to use it in other NLP operations. And one day-to-day life example of systems that use paraphrase detection is Turnitin to help people like teachers and professors [1]. And for example when teachers are revising the research or assignments of their students. They could wish to see if this research or homework is plagiarized from an online source or if the students are plagiarizing from one another. And here comes the great use of paraphrase detection which is the most important step in plagiarism detection. The paraphrase detection model will be formed from the combination of NLP techniques and machine learning techniques. This paper will mainly focus on paraphrase detection between two sentences. The method that will be utilized is to compare two sentences and then use machine learning techniques to determine whether one of the statements is paraphrased from the other. There are generally three main techniques of NLP that are used to measure the likeness of two sentences:

First, the string similarity techniques usually work on measuring the similarity score by comparing the characters of the sentence with the other sentence or by comparing the whole sentence with the other sentence [2].

Second, the semantic similarity techniques usually work on measuring the similarity score by comparing the meaning of the whole sentence with the other one. This comparison usually returns to a dictionary or something like Wikipedia or WORDNET to take the meaning of the sentences and make the comparisons [2].

Third, the embedding similarity techniques usually work on measuring the similarity score by converting the two sentences to two vectors after that the technique compares those two vectors with each other to get the similarity score [3]. There are many machine learning techniques like classic machine learning like classifiers, deep learning with neural network and finally the pre-trained models [4]. The second section will then give a literature review as well as background information on prior research in the same field. The third section will talk about the methodology and the system architecture of the project. The fourth section will discuss the data, tools and environment that will be used in the paper, also will talk about the setup of the experiments, the experiments themselves and the results of those experiments. The fifth section will be a discussion of the acquired results.

II. RELATED WORK

Mainly these previous works are divided into two categories which are unsupervised and supervised and those previous works worked on MSRP dataset. The unsupervised usually targets the problem of paraphrase detection by using different similarity techniques. The other approach of targeting the problem of paraphrase detection always uses machine learning and deep learning techniques to solve this problem and the supervised has labeled data to use. The two usual evaluation scores used by these two approaches are the accuracy score and the F1-Score. Accuracy score is how many correctly classified instances of the data while the F1-Score is a measure of the test's accuracy [5].

A. Unsupervised Methods

Some researchers used unsupervised methods to tackle the paraphrase detection problem by using LSA, WORDNET similarity like (lesk, Lin, jcn and other) and some semantic similarity techniques like Resnik, J & C and L & C and also uses cosine similarity and tf-idf weighting [6]. Hassan [7] used the encyclopedic method to get the similarity of the words by using Wikipedia and used also the LSA (latent semantic space) method, the SSA (salient semantic space) and the ESA (explicit semantic space). Rus et al [8] used a method that maps the two sentences into a graph in

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

three stages. First preprocess (tokenization, lemmatization, parsing) the two sentences, second the actual mapping from the text to the graph is called dependency graph creation and it uses the information from the parsing in the preprocessing stage and the last stage is final graph generation which uses the dependency graph and does on it some refinements to get the final graph. After mapping the text into a graph they use the Graph Subsumption method to get the decision if whether these two sentences are paraphrased or not. Islam and Inkpen [9] introduced a new model by using string and semantic similarity techniques together they named this model Semantic Text Similarity (STS) and they used string similarity techniques like the longest common subsequence (LCS) and normalized longest common subsequence (NLCS). In semantic similarity, the semantic similarity matrix between words. Milajevs et al [10] their model convert the two sentences into two vectors and then compare these two vectors with each other by the cosine similarity method and if the similarity value exceeds a certain threshold. These two sentences are considered paraphrased. They tried different thresholds with and without lemmatization. Fernando and Stevenson [11] used the different WORDNET similarity methods like ("JCN", "LCH", "LESK", "LIN", "RES" and "WUP") in the matrix similarity approach.

In table I, a list of unsupervised previous works was summarized to show different paraphrase detection methods on the MSRP dataset. The table is divided into four columns: the first column lists the previous work's reference number, the second column lists the paraphrase detection method, and the third and fourth columns list the accuracy and F1-Score, respectively.

Table I: Different models of unsupervised methods on MSRP dataset

Ref. No.	Methods	Accuracy	F1-Score
[6]	Vector based similarity (Baseline) – MCS	65.4% - 70.3%	75.3% - 81.3%
[7]	ESA – LSA – SSA	67.0% - 68.8% - 72.5%	79.3% - 79.9% - 81.4%
[8]	RMLMG	70.6%	80.5%
[9]	STS	72.6%	81.3%
[10]	Vector-based similarity	73.0%	82.0%
[11]	**matrixJcn**	**74.1%**	**82.4%**

B. *Supervised Methods*

Kozareva and Montoyo [12] the researchers used a combination of string similarity techniques as they used the "n-grams", "skip-gram" and "longest common subsequence (LCS)" and semantic similarity techniques like WORDNET similarity methods to extract features from the two sentences and then fed these features to machine learning techniques as they used three machine learning modules that are "SVM (support vector machine)", "K-NN (K Nearest Neighbors)" and Maximum Entropy (MaxEnt). Qiu et al [13] introduced a new model by combining the sentence similarity and dissimilarity. Their model has three phases and the first phase is preprocessing the sentences by using Charniak parser and assert and get the predicates of the sentences and the second phase they get the similarity between the predicates by using Thesaurus and the final phase is to input these predicates into dissimilarity classifier to judge if the two sentences are paraphrased or not. Zia and Wasif [14] their model has three stages. The first stage is the preprocessing by the "part of speech (POS)" tagging and followed by removing the stop words. The second stage is feature extraction by the longest common subsequence (LCS) and WordNet semantic heuristics, and the last stage is inputting the features to the weka tool that has many classifiers and specifically they used the logistic regression model. Blacoe and Lapata [15] their model represents the sentences in three types of vectors. The first type represents the pair of input sentences via concatenation or subtraction, and the second type represents a vector of encoding the words of the sentence, and the last type represents a vector of four informational items that are the cosine similarity of the sentence vectors, the length of the first sentence, the length of the second sentence and lastly the unigram overlap of the two sentences. After that, these three vectors are fed with different combinations to four composition models that are the "distributional memory (DM)", "the neural language model (NLM)", the recursive auto encoder (RAE) and the simple distributional semantic space (SDS) that gives the best results. Ji and Eisenstein [16] they introduced a new weighting method that is called TF-KLD and used a combination of feature set that has the unigram and bigram similarity methods and other methods that they called the fine-grained features. They used the weighting method with the fine-grained features and input them into the support vector machine. El Desouki et al [17] they approached the problem of paraphrase detection in three steps. In the first step, they inputted the sentences into two types of text similarity methods and the first is the string similarity algorithms and the second is the semantic similarity algorithm which is called skip-thought. Through these methods, they get similarity values and then in the second step they input the similarity values to the weka tool that has a lot of classifiers and the last step they used the select attribute method from the weka to select the best combination of the text similarity algorithms and it resulted in 7 algorithms that they called the CombineBest. They inputted the CombineBest into weka again and got the best results with the VotedPerceptron. Finch et al [18] there model has three steps. First, they tokenized the sentences and after that, they used stemming technique to return the word to its original root. Finally, in the last step, they used the output from the machine transition evaluation techniques that are WER, BLEU, PER and NIST. The output of those techniques is used to feed the support vector machine classifier to judge if the sentence is paraphrased or not. Wan et al [19] they used many features to feed different classifiers from the weka tool. The features are different "N-gram" techniques, "Dependency relation techniques", "Dependency Tree edit distance techniques" and surface techniques. To give a total of 17 used techniques to feed 5 classifiers of the weka that are the NavieBayes, a clone of the C4.5 decision tree classifier that is called J48, a support vector machine with a polynomial kernel that is called SMO, K-Nearest Neighbor that is called IBK and lastly the baseline technique that is called ZeroR but they only reported the result from the support vector machine as it outperformed other classifiers.

In table II, a list of supervised previous works summarized to show different paraphrase detection methods on the MSRP dataset. The table is divided into four columns: the first column lists the previous work's reference number, the second column lists the paraphrase detection method, and the third and fourth columns list the accuracy and F1-Score, respectively.

Table II: Different models of supervised methods on MSRP dataset

Ref. No.	Methods	Accuracy	F1-Score
[12]	KM	76.6%	79.6%
[13]	QKC	72.0%	81.6%
[14]	ParaDetect	74.7%	81.8%
[15]	SDS	73.0%	82.3%
[16]	**TF-KLD**	**80.4%**	**85.9%**
[17]	CombineBest	76.6%	83.5%
[18]	FHS	75.0%	82.7%
[19]	WDDP	75.6%	83.0%

III. METHODOLOGY

Fig 1 shows the process of our proposed model and it has 3 processes the preprocessing of the dataset and after that, the preprocessed data is inputted to the three text similarity techniques to get the similarity scores and lastly these similarity scores will be inputted into machine learning to get the result of the data that the data is paraphrased or not and the next subsections will explain the overview in details.

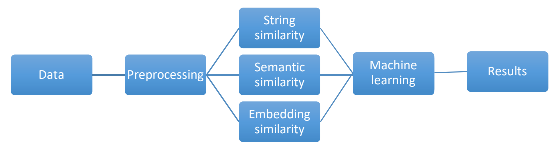


Fig 1: System Overview

A. *Preprocessing*

First is the step of preprocessing where we remove special characters, change the sentence to lowercase, use stemming, use lemmatization and remove stop words. Where the stemming and the lemmatization are to convert the words of the sentence to their root form.

B. *Text similarity techniques*

The second step will input the preprocessed sentences into three similarity techniques that are:

First, the string similarity techniques that take the two sentences and compare the difference in the characters and words in the two sentences and the abydos library in python has over 200+ string similarity techniques like Levenshtein similarity and Damerau-Levenshtein-similarity and other more string similarity techniques.

Second, the semantic similarity techniques that take the two sentences and compare the difference in the meaning of the two sentences and the nltk and spacy libraries in python have different semantic similarity techniques like wup and lin and other more semantic similarity techniques.

Third, the embedding similarity techniques that take the two sentences and convert the two sentences to their vectors and then compare these vectors with each other to get the similarity score and the sentence transformers library in python has many pre-trained models that make the embedding similarity techniques like bert-base-nli-mean-tokens and all-mpnet-base-v2 and other more pre-trained models. These similarity scores that are produced are between 1 and 0. 1 being 100% percent similar to each other and 0 being has no similarity with each other.

C. *Hybrid model*

The third step is to take these similarity scores with different combinations and all of them to be fed to different machine learning classifiers from the skleran library in python to decide whether the two sentences are paraphrased or not and these classifiers are like Logistic Regression and Extra Tree Classifier and other classifiers.

IV. EXPERIMENT AND RESULTS

The tests were carried out on an i7 9th Gen processor with 16GB of RAM and an NVIDIA GTX 1660ti 6GB GPU. All the experiments were done on the same dataset with a 10-fold K-fold validation method.

A. Data-set

The proposed model will use "the Microsoft research paraphrase corpus" (MSRP) dataset. This dataset contains 5801 pairs of sentences and these sentences are extracted from news sources on the web and from each given news article, only one sentence has been taken. Each pair of sentences is combined with human annotations to indicate whether the pair of sentences are semantically equivalent or not and 3900 pairs (67%) of the original 5801 pairs were annotated as paraphrased pairs. In table III, there are some examples of the dataset.

Table III: Dataset examples

#1 String	#2 String	Quality
Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.	Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	1
Around 0335 GMT, Tab shares were up 19 cents, or 4.4%, at A$4.56, having earlier set a record high of A$4.57.	Tab shares jumped 20 cents, or 4.6%, to set a record closing high at A$4.57.	0
The Nasdaq had a weekly gain of 17.27, or 1.2 percent, closing at 1,520.15 on Friday.	The tech-laced Nasdaq Composite .IXIC rallied 30.46 points, or 2.04 percent, to 1,520.15.	0
Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.	With the scandal hanging over Stewart's company, revenue the first quarter of the year dropped 15 percent from the same period a year earlier.	1

B. Experiments

In table IV, the different experiments done on the MSRP dataset with lowercase, remove stop words and lemmatization as preprocessing stage and the text similarity stage consist of three different types of text similarities which consist of different combinations of 168 string similarity techniques, 8 bert models for embedding similarity techniques and lastly 3 wordnet algorithms and 1 spacy algorithm for semantic similarity techniques and those similarity scores are inputted to 3 classifiers that are Linear SVC, Logistic RegressionCV and Ridge ClassifierCV. From table IV, the Linear SVC classifier scored the best accuracy score with 75.78% and the best F1-Score with 83.01%.

Table IV: Different experiments of the proposed model

Name	Accuracy	F1-Score	Classifier
168 string + 8 bert + wordnet	**75.78%**	**83.01%**	**Linear SVC**
168 string + 8 bert + spacy + wordnet	75.75%	82.88%	Logistic RegressionCV
168 string + 8 bert + spacy	75.64%	82.97%	Ridge ClassifierCV
168 string + 8 bert	75.59%	82.76%	Logistic RegressionCV
168 string + spacy	74.71%	82.19%	Logistic RegressionCV
168 string + spacy + wordnet	74.66%	82.19%	Logistic RegressionCV
168 string + wordnet	74.38%	81.99%	Logistic RegressionCV
168 string	74.14%	81.96%	Linear SVC

V. DISCUSSION

The combination of the 168 string similarity techniques from the abydos library in python and 8 bert models from the sentence transformers library as the embedding similarity techniques and 3 of wordnet as the semantic similarity techniques. This combination is better than any other combination as it got 75.78%. It is clear from these trials that the more features and similarity algorithms added, the better the outcomes will be. It is concluded from table V that the combination of the different similarity techniques is much better than single text similarity techniques alone and the selected algorithms in table V are the top 20 algorithms with the Gradient Boosting classifier as it is the best classifier for single techniques. It is concluded from table VI that the approach of combining the similarity techniques with the machine learning techniques is giving better results than using the unsupervised approach that concentrates on using the text similarity techniques alone. And from table IV and table V that the improvement range is 1.26% between the worst combination and the best single text similarity technique and 2.9% between the best combination and the best single text similarity technique and 1.64% between the best combination and the worst combination.

Table V: Single text similarity algorithms bench-marked with the MSRP dataset

Methods	Accuracy	F1-Score
all-MiniLM-L12-v2	**72.88%**	**81.36%**
all-MiniLM-L6-v2	72.83%	81.27%
all-mpnet-base-v2	72.59%	81.30%
Dennis	72.36%	81.05%
Kuhns VII	72.36%	81.05%
Kuhns IX	72.34%	81.22%
ms contingency	72.34%	81.22%
Pearsons Chi-Squared	72.34%	81.22%
Pearson Phi	72.34%	81.22%
Goodman & Kruskals Tau A	72.33%	81.27%
BaroniUrbaniBuserI	72.26%	80.65%
Doolittle	72.24%	81.17%
Pearson II	72.24%	81.17%
Unknown B	72.24%	81.17%
Harris & Lahey	72.19%	81.14%
GilbertWells	72.17%	80.89%
Kuder & Richardson	72.15%	81.13%
Maxwell & Pilliner	72.15%	81.13%
all-distilroberta-v1	72.14%	80.90%
Kuhns XI	72.09%	80.88%

Table VI: Comparing the proposed model with previous models

Ref. No.	Methods	Accuracy	F1-Score
Proposed Model	**Linear SVC**	**75.78%**	**83.01 %**
[6]	Vector based similarity (Baseline) – MCS	65.4% - 70.3%	75.3% - 81.3%
[7]	ESA – LSA – SSA	67.0% - 68.8% - 72.5%	79.3% - 79.9% - 81.4%
[8]	RMLMG	70.6%	80.5%
[9]	STS	72.6%	81.3%
[10]	Vector-based similarity	73.0%	82.0%
[11]	matrixJcn	74.1%	82.4%

VI. CONCLUSION AND FUTURE WORK

Paraphrase detection is one of the most basic and common tasks in the field of Natural language processing. Yet, it is very important as it is involved in many more complex natural language processing tasks like text mining, plagiarism detection, data retrieval and it is used also in academic writing as a lot of people copy from other people's work and research and this is considered as a big crime. Also, it is concluded from the experiments done in this paper that the approach of combining many different similarity techniques is giving much better results than the use of only one type of similarity techniques. it is concluded that the combination of string similarity, semantic similarity and embedding similarity with the help of machine learning classifiers are better than the use of only a single type of similarity technique alone with help of machine learning classifiers and also better than the use of unsupervised methods that use the text similarity techniques with a threshold only and also the proposed model is better and simpler than some previous supervised methods as they used some complex and deep machine learning neural networks.

Our future work is to try different preprocessing techniques like stemming, removing special characters, removing numbers and without any preprocessing and also different combinations of similarity techniques and to add more similarity techniques and to use other different machine learning techniques like neural networks and LSTM

REFERENCES

[1] Meo, S. A., & Talha, M. (2019). Turnitin: Is it a text matching or plagiarism detection tool?. Saudi journal of anaesthesia, 13(Suppl 1), S48.
[2] Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. international journal of Computer Applications, 68(13), 13-18.
[3] Shahmohammadi, H., Dezfoulian, M., & Mansoorizadeh, M. (2021). Paraphrase detection using LSTM networks and handcrafted features. Multimedia Tools and Applications, 80(4), 6479-6492.
[4] Sah, S. (2020). Machine learning: a review of learning types.
[5] El Desouki, M. I., & Gomaa, W. H. (2019). Exploring the recent trends

of paraphrase detection. International Journal of Computer Applications, 975(S 8887).
[6] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
[7] Hassan, S. (2011). Measuring semantic relatedness using salient encyclopedic concepts. University of North Texas.
[8] Rus, V., McCarthy, P. M., Lintean, M. C., McNamara, D. S., & Graesser, A. C. (2008, May). Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In FLAIRS conference (pp. 201-206).
[9] Islam, A., & Inkpen, D. (2009). Semantic similarity of short texts. Recent Advances in Natural Language Processing V, 309, 227-236.
[10] Milajevs, D., Kartsaklis, D., Sadrzadeh, M., & Purver, M. (2014). Evaluating neural word representations in tensor-based compositional settings. arXiv preprint arXiv:1408.6179.
[11] Fernando, S., & Stevenson, M. (2008, March). A semantic similarity approach to paraphrase detection. In Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics (pp. 45-52).
[12] Kozareva, Z., & Montoyo, A. (2006, August). Paraphrase identification on the basis of supervised machine learning techniques. In International conference on natural language processing (in Finland) (pp. 524-533). Springer, Berlin, Heidelberg.
[13] Qiu, L., Kan, M. Y., & Chua, T. S. (2006, July). Paraphrase recognition via dissimilarity significance classification. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 18-26).
[14] Ul-Qayyum, Z., & Altaf, W. (2012). Paraphrase identification using semantic heuristic features. Research Journal of Applied Sciences, Engineering and Technology, 4(22), 4894-4904.

[15] Blacoe, W., & Lapata, M. (2012, July). A comparison of vector-based representations for semantic composition. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (pp. 546-556).

[16] Ji, Y., & Eisenstein, J. (2013, October). Discriminative improvements to distributional sentence similarity. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 891-896).
[17] El Desouki, M. I., Gomaa, W. H., & Abdalhakim, H. (2019). A hybrid model for paraphrase detection combines pros of text similarity with deep learning. Int. J. Comput. Appl, 975, 8887.

[18] Finch, A., Hwang, Y. S., & Sumita, E. (2005). Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In Proceedings of the third international workshop on paraphrasing (IWP2005).

[19] Wan, S., Dras, M., Dale, R., & Paris, C. (2006, November). Using dependency-based features to take the'para-farce'out of paraphrase. In Proceedings of the Australasian language technology workshop 2006 (pp. 131-138).

4.4.9 An Ensemble-Based Model to Improve the Accuracy of Automatic Short Answer Grading

Mostafa Mohamed Saeed on publishing a Conference paper on his graduation project, which was published at the 2022 International Mobile, Intelligent and Ubiquitous Computing Conference international conference. The paper entitled “An Ensemble-Based Model to Improve the Accuracy of Automatic Short Answer Grading" proposed an automated system that evaluate the student answers and give it an accurate grade automatically according to a given model answers of these questions without the participation of any manual evaluator, which is very useful specially after COVID-19 pandemic when most of the quizzes, assignments and exams became online. The Hybrid model that is implemented in this paper was capable enough to score high results comparing to much more complex models that have been applied on the same Dataset which is “Texas Data Structure” Dataset. [16]

An Ensemble-Based Model to Improve the Accuracy of Automatic Short Answer Grading

Mostafa Mohamed Saeed
Faculty of Computer Science
MSA University
Giza, Egypt
mostafa.mohamed52@msa.edu.eg

Wael H. Gomaa
Faculty of Computer and Artificial Intelligence, Beni-Suef University
Faculty of Computer Science, MSA University
Egypt
Wahassan@msa.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781737

***Abstract*—Since 1966 much research has been done on the automatic grading of student answers task, and it was divided into short answer grading and essay scoring. In this paper, on the short answer grading challenge, we are working with text similarity approaches that are being classified into string, semantic (corpus and knowledge-based), and embedding text similarity approaches are the three types of text similarity techniques. On the Texas data-structure data set, different experiments were examined individually before being merged to give a maximum correlation result of 0.65.**

***Index Terms*—Automatic Scoring, Natural Language Processing, Machine Learning, Text Similarity**

I. Introduction

The intersection or mixture of computer science, linguistics, and machine learning is known as natural language processing (NLP). This area focuses on computer-human natural language communication, and NLP is all about training computers to interpret human language as effortlessly as people do. NLP has benefited greatly from recent advances in machine learning, particularly deep learning methods. The field is organized into several areas, the most prominent of which are: Natural Language Understanding (NLU) refers to a computer's capacity to understand what we say as clearly as humans, whereas Natural Language Generation (NLG) refers to the ability of a computer to generate natural language. Our major focus will be Natural Language Understanding, and these two components can perform a variety of tasks including automatic translation, named-entity recognition, summarization, connection extraction, sentiment analysis, audio recognition, and topic segmentation. NLP is a challenging field. There are a lot of factors that make this process more difficult, including the fact that there are hundreds of natural languages, each with its unique set of syntactic rules. Words can be ambiguous, altering their meaning depending on the context. Furthermore, ambiguous sentences may be a prevalent difficulty in NLP. Sentences and phrases that can be construed in two or more ways are said to as ambiguous. Reading a sentence without the context of the surrounding language is challenging for humans, so imagine how tough it is for a machine to understand this. POS (part of speech) tagging is one NLP approach that may be employed. Humans are easily able to rectify spelling errors. We have the capacity to distinguish between a misspelt word and its correctly spelled equivalent. A machine, on the other hand, will have a tougher problem comprehending a phrase that contains several spelling mistakes. To complete this demanding task, natural language processing methods were used. Another issue is identifying each individual sentence since phrases and sentences are made up of words that are combined. Phrase detection may be tough, and it's not as simple as looking for periods at the end of a sentence. Periods can be used in a variety of locations where they should represent the conclusion of a sentence, but they are often used as a shorthand for something else, such as Mrs. or Mr., or decimal numbers like 12.834. Furthermore, there are other issues that make working in the NLP field challenging and difficult. Text similarity measures play an effective and important role in all NLP related research and applications. Finding word similarity is a crucial aspect of text similarity, which is then utilized as a starting point for comparing sentences, paragraphs, and documents. Words can be compared to each other from two ways semantical and lexical. String based similarity is a metric that measures the distance between two or more text strings. From the most popular algorithms in string-based similarity Longest Common Substring (LCS) that calculate the similarity according to the length of characters that exist in both strings, Dameraue Levenshtein that is based on counting the minimum operations needed to transform one string to another [1], [2], Cosine similarity that is based on calculating the cosine angle between two vectors to determine their similarity. and Jaccard Similarity that is calculated according to the number of shared terms over the number of terms in the two comparable words [3]. For semantic similarity, approaches there are two types, Corpus-Based Similarity and Knowledge-Based Similarity. Corpus-Based Similarity is based on the similarity between two words that exist in a corpus. There are much algorithms for corpus-based similarity like "Hyper Analogue to language (HAL)" [4], [5] and "Latent Semantic Analysis (LSA)" [6]. Knowledge-Based similarity is based on calculating the semantic degree between two words according to a large semantic network like WordNet [7] database that groups each word in a unique sysnet that is an interlinked with all the information needed about each word and also its conceptual semantic

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

and lexical relations. There are different semantic similarity techniques using WordNet. There are three algorithms that are based on information content which are ”Resnik algorithm (res)” [8], ”Lin algorithm (lin)” [9], and ”Jiang Conrath algorithm (jcn)” [10]. The other three metrics are based on path length, ”Leacock Chodorow algorithm (lch)” [11], ”Wu Palmer algorithm (wup)” [12], and ”Path Length” algorithm. Over the last few years, the educational system has undergone significant adjustments, particularly in the aftermath of the COVID-19 pandemic. As a result, the educational phases have shifted from being only between the student and the academic staff to be between them both but with technology that helps them in their overall education processes. As a result, the idea of implementing an automatic grading system has gained traction, owing to the ever-expanding educational community. Moreover, this approach will undoubtedly provide numerous benefits, such as assisting academics in decreasing their burden and allowing them to focus on other tasks. In addition to that, it will make the process of sending the real grades to the students more promptly rather than waiting for manual examiners to grade their answers one by one, and it will obtain unbiased grading results, ensuring that grading is always done in a standard formal manner. This study proposes an automatic scoring system for short answers that calculate an efficient and quick grade for the student answer based on different text similarity techniques. Automatic grading comprises reviewing and grading any student answer and awarding a specific grade depending on the contents. There are two sorts of automatic grading: essay scoring and short response grading, with each having its own system or methodology. ES is a challenging process to perform successfully and efficiently since it involves grading long essay questions based on content, grammar checking, correct run-ons, punctuation, and more other concepts of linguistics. The majority of ES approaches are used in the English language; however, ES may be characterized by other terms such as automated essay evaluation, automated essay evaluations, and automated writing scoring. Its execution necessitates the training of a model on hundreds of essay replies assessed by experts, after which the computer extracts certain particular aspects from these essays that will aid him in developing a model capable of predicting the human manual grade. Several software systems have been developed, including Project Essay Grader, which uses proxy metrics to assess essay quality, Intelligent Essay Assessor, which compares the semantic similarity of textual paragraphs information, and E- rater, which was developed by the Educational Testing Service and relies primarily on linear regression. The next sections will cover some related works for the same topic, data set description and example, our approach (which comprises five stages), implemented experiments and their outcomes, analysis of our results, and finally conclusion and future work.

II. RELATED WORK

Various approaches for implementing an automatic short response grading system have been presented in the previous ten years.

(Tulu et al.) [13] This research has proposed a new technique using a deep learning approach, which is MaLSTM and sense vectors obtained by SemSpace. The student answer is represented by a sysnet-based embedding of WordNet, and the reference answer is inputted into a parallel LSTM model, which leads to the transformation of these two representations into vectors and the use of Manhattan similarity to measure the similarity between the two vectors (student answer and reference answer). This technique was utilized and evaluated on the Texas data structure dataset, and they were able to reach a correlation of 0.95 by separating the data into train and test for each question. When they operate on the dataset as is, however, they get a correlation of 0.15. Furthermore, they said that LSTM does not take into account long-range dependencies, in contrast to the transformers that perform admirably in the automatic short response grading assignment.

(Süzen et al.) [14] This study focuses on automatic scoring of the students' answers since it is common in the United Kingdom, and they would want to provide feedback on the students' answers whether they are incomplete or incorrect. This study suggested using clustering, regression analysis, and hamming distance to find the similarity between the student answer and the given reference answer to solve the automatic short answer grading problem, but they didn't go into detail about the technicalities involved in getting their results. The authors evaluated their strategy on the Texas data structure data set, achieving a correlation of 0.81, and the authors said that semantics or synonyms were not taken into consideration in this approach.

(Gomaa and Fahmy) [15] In this study, the Ans2vec approach was suggested as a simple and efficient model for scoring brief answers. To convert the model and student responses into vectors. They make use of Ans2vec (Skip-thought vector technique). Skip-thought vectors are a sentence-level version of word2vec, in which the surrounding words is being predicted by the word2vec and the surrounding sentences is being predicted by the skip thought vectors. According to the authors, they evaluated their model on the Texas Data Structure data set and obtained a correlation of 0.63. They stated that working with the Texas dataset is tough because most researchers struggle to produce correlation results greater than 0.7, as can be demonstrated in this study.

(Pribadi et al.) [16] The first stage of this research will propose (Maximum Marginal Relevance (MMA), which will generate a reference answer from the given student answer, and the second stage will propose ”GAN-Longest Common Subsequence (GAN-LCS)”, which is an extension of the LCS that works on computing the similarity between two sentences of different lengths, with the output coefficient being the student grade. This approach was tested on the Texas Dataset, which yielded a 0.468 correlation score.

(Hassan et al.) [17] This study proposes using a paragraph embedding approach for both the student and reference answer, then computing the cosine similarity between them and using the result as a feature in a regression model to predict the student grade. The vector representation of the student response and the reference answer might be generated in two ways. The first method was to create the vector by summing the word vectors in our text. To construct our vectors, we used the second way of training a deep learning model. All word embedding was done with Glove, Word2Vec, and fasttext, while all paragraph embedding was done with Skip- thoughts, Doc2Vec, and InfraSent. The author utilized the Texas dataset to test his method, and they were successful.

(S. Roy et al.) [18] The focus of this study was on vector-based approaches. They began by deleting all stop words from their data before computing the seven to seven similarity measures: block distance, JingConarth, DISCO, Lesk, Glove, Sense Vectors, and Word2Vec. They also employed sentence-to-sentence similarity metrics such as the text-to-text model, the Min-Max additive model, and the Vector Summa- tions Model. Finally, all of these methods were evaluated on the Texas dataset, with the greatest correlation value of 0.586.

(Magooda et al.) [19] His system is offered as a three-module sequence in this study. He began by pre-processing his data by eliminating stop words and doing lemmatization or stemming. Second, he will begin calculating certain similarity measures using the word-to-word approach, which will include string similarity techniques such as block distance. In addition to knowledge-based techniques like JiangConrath and Lesk algorithm, corpus-based techniques like DISCO, and the last type of similarity, which is the Vector representation, such as Word2Vec toolkit, Glove pre-trained model, and sense aware vectors, there is also the Vector representation, which includes Word2Vec toolkit, Glove pre-trained model, and sense aware vectors. Finally, the scaling module assigns a grade to the student between 0 and 5 on a scale of 1 to 5. This method was tried on a dataset from Texas, and he was able to reach a correlation of 0.550.

III. Data set

TABLE I: Example for a question and its model answer

Sample Question
In one sentence, what is the main idea implemented program by insertion sort?
Model Answer
Taking one array element at a time, from left to right, it inserts it in the right position among the already sorted elements on its left.

The ASAG task uses six datasets, starting with the CREG and CREE datasets (Meurers et al. 2011), the ASAP dataset that is from a Kaggle competition (Hewlett 2012), the SciEntsBank and Beetle datasets (Dzikovska et al., cited in Galhardi Brancher, 2018), and the Texas dataset from (Meurers et al.) The Texas dataset was chosen as the major source for implementing the ASAG out of those six datasets. The reason for choosing the Texas dataset is because it is the most difficult dataset to use as the main source. The Texas Dataset (Mohler 2011) contains 87 questions chosen from 10 assignments (each assignment has from four to seven questions) and two tests (each with ten questions), all of which are connected to an introduction of computer science curriculum for undergrad students (Mohler et al. 2011). There are 2,442 replies from students. The data includes the average grades of the two manual graders that grades each answer on scale of 0 to 5. When it comes to grading, grader one is more tolerant than grader two. Grader one assigned a 4.43 average, whereas grader two assigned a 3.94 average. Because it was an augmentation of the dataset generated by the authors in 2009, this dataset is also known as Texas Extended Dataset (Mohler and Mihalcea 2009).

TABLE II: Two student answers with manual evaluators grade

Student 1 Answer	Grade/5
Take a number and choose a pivot point and insert the number in the correct position from the pivot point..	**3/5**
Student 2 Answer	**Grade/5**
Insertion sort removes an element from the data, and inserts it at the correct position in the already sorted list.	**5/5**

IV. Methodology

Our proposed model will pass through five stages as shown in fig (1). The first stage is the data set preparation. The second stage is the pre-processing stage. The third stage is the processing stage where multiple text similarity techniques will be implemented to compare the student answer with the same question given reference answer. The fourth stage is to grade the student answer. Then finally, the last stage is evaluating the proposed model grade according to the manual given grade by the examiner (educational staff).

A. Data set preparation

In the first stage as shown in fig (2), we have to extract everything separately and efficiently from our data. Starting with extracting the question itself, the student answer for each question, all the given reference answers for each question and finally the average grade of the two evaluators that will be our main target to reach.

B. Pre-Processing

In this phase as shown in as shown in fig (3), Most of the linguistic techniques are being applied. The first technique is converting all the letters of all given sentences (including the question itself or the question reference answer or the student answer) into lower case letters as in NLP the capital 'A' is not equal to a small 'a'. The second technique is performing the stemming which is converting each word is given into its root form or lemmatization which is the process of combining a different word's inflected forms so that they may be studied as a single item. Moreover, these two processes have to be done after splitting each sentence according to the spaces. The third technique is removing all stop words and special characters

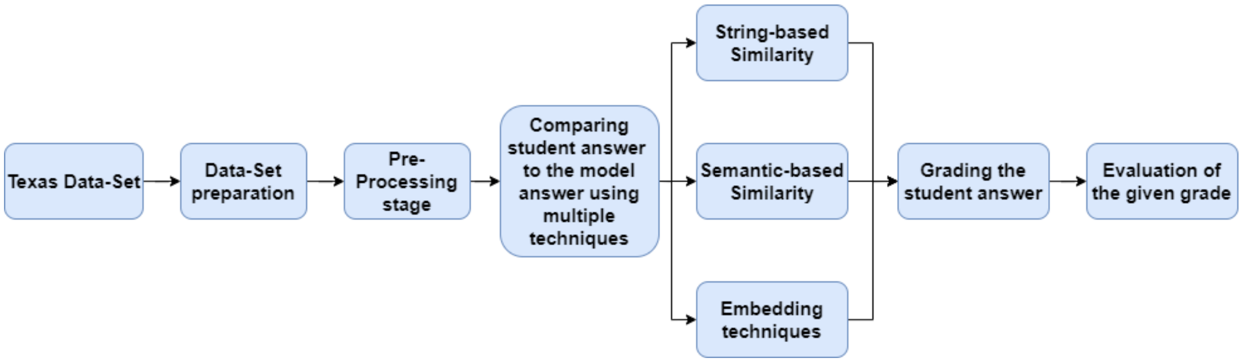


Fig. 1: General System Architecture

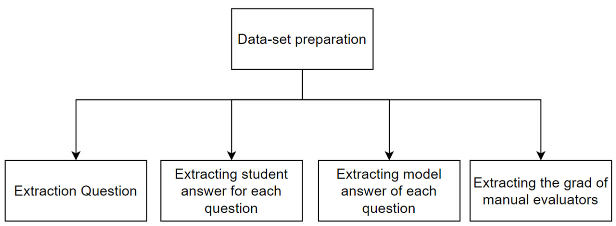


Fig. 2: Data set preparation processes

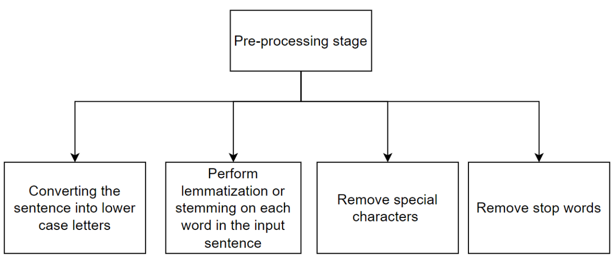


Fig. 3: Pre-Processing processes

in our sentences. And the final technique is converting all numeric numbers into alphabetical numbers.

C. *Processing*

In this phase, Multiple text similarity techniques and algorithms are going to be implemented and then combined together to form our Hybrid proposed model. For the string-based similarity algorithms, we will apply 168 string-based algorithm from Abydos library. For the Corpus-based similarity algorithms, Latent Semantic Analysis (LSA) algorithm will be implemented and for Knowledge-based similarity algorithms, five algorithms from WordNet similarity algorithms will be applied which are LI, LIN, WPATH and JCN. Finally, for Embedding techniques, we will be using Bert models, Glove models, Roberta, and more other models that convert our sentences into vectors and then apply some text similarity algorithms using their pre-trained models.

D. *Predicting the grade*

In this phase, our main target is to make our regression model got the ability to predict the quick and efficient grade for the student after merging all the similarity algorithms output that we have calculated while comparing the student answer to the reference answer.

E. *Evaluation*

In the final phase, the correlation metrics are used in this regression task to evaluate all the output grades predicted by our proposed model.

V. Experiment and Results

The environment that was used to implement this approach was on a laptop with core-i7 8th Gen, 16GB ram and GPU Nvidia GTX 1060 6GB.

TABLE III: Correlation results of some techniques without combination

Text-similarity technique	**Correlation Result**
Kuhns VI	0.488928
GiniII	0.487648
Iterative SubString coorelation	0.486732
Forbes II	0.481941
Fuzzy Wuzzy Token Set similarity	0.477305
Kuhns IV	0.473229
Tulloss S	0.466162
Cosine	0.420157

1) Experiments: Text-similarity strategies (algorithms) were applied independently without being integrated with any other algorithm in the early studies, but the results were not the best as shown in Table.(III) and Fig(4).

Our technique delivers superior and comparable outcomes in experimental evaluation on Texas data-structure data set as shown. The implemented tests depicted in Table (IV) are the most significant of all since they introduce our fundamental concept and goal of creating an ensemble model. For the first experiment, the student answer and the reference answer were compared using string similarity approaches, with lower case

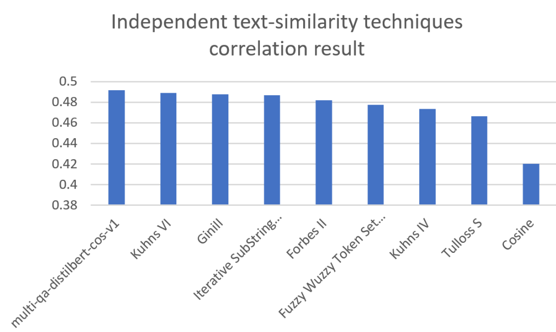


Fig. 4: Independent text similarity techniques correlation result

TABLE IV: Experiments

Pre-processing						Text-similarity				Corr.
Lower Case	Remove stop words	Remove special characters	Number to strings	Lemma	Stemming	String similarity	Semilar	Wordnet	Embedding models	Correlation results
√	-	-	-	√	-	√	√	√	√	0.6514
√	-	-	-	√	-	√	√	√	-	0.6105
√	-	-	-	√	-	√	-	-	-	0.5946
√	-	-	-	-	-	√	-	-	-	0.5923
√	-	-	-	-	-	√	-	-	-	0.5825
√	-	-	-	-	√	√	-	-	-	0.577

letters and stemming procedures applied to each word in both phrases, yielding a 0.577 correlation. The student answer and the reference answer were compared using string similarity methodologies, but without any processing techniques on the student answer or the manual reference answer, yielding a 0.5825 correlation. For the third experiment, the student answer and the reference answer were compared using string similarity methodologies, with lower case letters and stemming techniques used for each word in both phrases, yielding a 0.5923 result. And this demonstrates that the pre-processing stage is one of the most important aspects that must be applied well in this technique. The concept of building an ensemble model was used in the fourth and fifth experiments, where lower case letters technique and lemmatization were used for all sentences, but the fourth experiment was a combination of string and semantic similarity techniques with a correlation of 0.6105, while the fifth experiment with the addition of only some embedding text similarity techniques like "bert-base-nli-mean- tokens" model. This is a sentence transformers Bert model that has been pre-trained on one million sentence pairings with a batch size of 16 and a learning rate of 2e-5. It has a maximum sequence length of 128 characters and a word

embedding dimension of 728. which results in a correlation equal to 0.6514.

For the second experiment, The student answer and the reference answer were compared using string similarity approaches, but without any pre-processing technique on the student answer or the manual model answer, which result for 0.5825 correlation. In this approach K-fold was implemented as our validation technique, the data is separated into different k-subsets of data, with each k subset working as a test set and the k-1 subset acting as a training set, and a holdout process being performed k-times. (k=10 in our implementation). Texas data set. Moreover, different classifiers have been implemented to our data and the Random Forest classifier was the best one each time for the correlation final result.

2) Results: Our proposed hybrid approach has achieved better or more competitive results in the experimental evaluation on the Texas benchmarking data set as shown in Table (V) and fig (5). Our model achieved 0.65 correlation result compared to all the other systems.

TABLE V: Results of Texas data set

System	Correlation
Chaturvedi and Basak [20]	0.805
Our proposed Model	**0.65**
Gomaa, W. H., & Fahmy [15]	0.63
S. Roy et al [18]	0.57
Hassan et al [17]	0.569
Magooda et al [19]	0.55
Mohler et al [21]	0.52
Gomaa et al [22]	0.50
Basak et al [23]	0.365
Tulu et al [13]	0.15

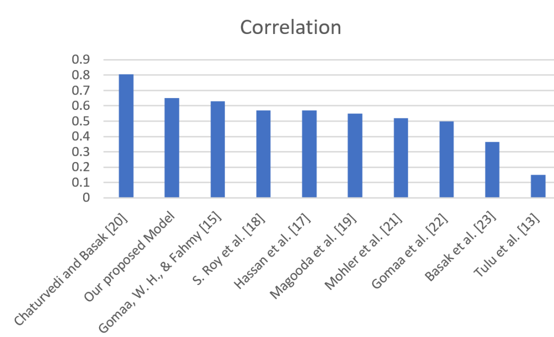


Fig. 5: Comparable Correlation Results

VI. Result Analysis

As shown in Table (III) the best text-similarity technique correlation was 0.488928 utilizing the Kuhns IV algorithm which indicates that employing text similarity, semantics, or even embedding techniques individually without combining them will not yield the best correlation results.

Moreover, as shown in Table (V), the combination between the 168 string similarity algorithm has increased the string similarity techniques' overall correlation results. In addition to that, adding standard semantic similarity techniques and embedding text similarity techniques improved our model outcomes, and ultimately, embedding text similarity techniques models increased our hybrid model's confidence and accuracy compared to other simple and complex approaches. The ensemble model delivers more competitive results since each individual algorithm is powerful in its own way, and by combining them, the model becomes much more efficient.

VII. Conclusion and Future Work

Automatic Short Answer Grading has been one of the most important tasks that need to be implemented in an accurate, efficient and quick way in real life. In comparison to other complicated implementations, this study has shown that using simple procedures which are string similarity or semantic or embedding similarity techniques or the combination between them in one hybrid approach without the usage of bert models can result for a very efficient correlation result.

Future work will be concerned with different approaches. Different data sets will be applied on our model like the Beetle and SciEntsBank data set or Corpus of Reading comprehension exercises in German data set (CREG) or the Corpus of Reading Comprehension Exercises (CREE) data set. From other approach which is testing our proposed model on different language data sets like Arabic data sets. Also, during analyzing Texas data structure data set, it was noticed that there are many numbers, mathematical equations and programming code which need handcrafting techniques or deep learning approaches either using transformers or deep neural networks or even both merged together that will lead for more efficient results in this field specifically.

References

[1] Patrick AV Hall and Geoff R Dowling. Approximate string matching. *ACM computing surveys (CSUR)*, 12(4):381–402, 1980.

[2] James L Peterson. Computer programs for detecting and correcting spelling errors. *Communications of the ACM*, 23(12):676–687, 1980.

[3] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[4] Kevin Lund. Semantic and associative priming in high-dimensional semantic space. In *Proc. of the 17th Annual conferences of the Cognitive Science Society, 1995*, 1995.

[5] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208, 1996.

[6] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[7] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[8] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[9] Dekang Lin. Extracting collocations from text corpora. In *First workshop on computational terminology*, pages 57–63. Citeseer, 1998.

[10] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[11] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[12] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*, 1994.

[13] Cagatay Neftali Tulu, Ozge Ozkaya, and Umut Orhan. Automatic short answer grading with semspace sense vectors and malstm. *IEEE Access*, 9:19270–19280, 2021.

[14] Neslihan Süzen, Alexander N Gorban, Jeremy Levesley, and Evgeny M Mirkes. Automatic short answer grading and feedback using text mining methods. *Procedia Computer Science*, 169:726–743, 2020.

[15] Wael Hassan Gomaa and Aly Aly Fahmy. Ans2vec: A scoring system for short answers. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 586–595. Springer, 2019.

[16] Feddy Setio Pribadi, Adhistya Erna Permanasari, and Teguh Bharata Adji. Short answer scoring system using automatic reference answer generation and geometric average normalized-longest common subsequence (gan-lcs). *Education and Information Technologies*, 23(6):2855–2866, 2018.

[17] Sarah Hassan, Aly A Fahmy, and Mohammad El-Ramly. Automatic short answer scoring based on paragraph embeddings. *International Journal of Advanced Computer Science and Applications*, 9(10):397–402, 2018.

[18] Shourya Roy, Sandipan Dandapat, Ajay Nagesh, and Yadati Narahari. Wisdom of students: A consistent automatic short answer grading technique. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 178–187, 2016.

[19] Ahmed Ezzat Magooda, Mohamed Zahran, Mohsen Rashwan, Hazem Raafat, and Magda Fayek. Vector based techniques for short answer grading. In *The twenty-ninth international flairs conference*, 2016.

[20] Bhuvnesh Chaturvedi and Rohini Basak. Automatic short answer grading using corpus-based semantic similarity measurements. In *Progress in Advanced Computing and Intelligent Engineering*, pages 266–281. Springer, 2021.

[21] Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 752–762, 2011.

[22] Wael H Gomaa and Aly A Fahmy. Short answer grading using string similarity and corpus-based similarity. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 3(11), 2012.

[23] Rohini Basak, Sudip Kumar Naskar, and Alexander Gelbukh. Short-answer grading using textual entailment. *Journal of Intelligent & Fuzzy Systems*, 36(5):4909–4919, 2019.

4.5 Spring 2022 - Accepted and waiting for publication

4.5.1 Exploring and Classifying Beef Retail Cuts Using Transfer Learning

MSA University and Faculty of Computer Science would like to congratulate the Senior Student from the Faculty of computer science Abdallah Ayman Abuzaid whose paper got published to the IEEE Xplore as part of the 2022 IEEE 9th International Conference Science of Electronics, Technologies of Information and Telecommunications (SETIT). Abdallah Abuzaid presented his paper entitled "Exploring and Classifying Beef Retail Cuts Using Transfer Learning" which proposed an evaluation of the deep learning neural network in artificial intelligence (AI) technologies to provide a rapid recognition and immediate proper classification of the different beef retail cuts (Liver, Roast Beef, Beef Chuck, Beef Round, Strip-Lion, Round Fillet, Beef Flank) to classify them accordingly. The 2022 IEEE SETIT conference was held as and hybrid event from 28th to 30th May 2022., and the presentation was followed by a QA discussion where participants shared their interest in the topic and how it can affect the scientific community. [2]

Exploring and Classifying Beef Retail Cuts Using Transfer Learning

1st Abdallah Abuzaid
Faculty of Computer Science
October University for Modern Sciences and Arts
Giza, Egypt
abdallah.ayman1@msa.edu.eg

2nd Ayman Atia
HCI-LAB, Faculty of Computers and
Artificial Intelligence, Helwan University.
Faculty of Computer Science, October University
for Modern Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—An evaluation of the deep learning neural network in artificial intelligence (AI) technologies is proposed to provide a rapid recognition and immediate proper classification of the different beef retail cuts (Liver, Roast Beef, Beef Chuck, Beef Round, Strip-Lion, Round Fillet, Beef Flank) to classify them accordingly. The problem is that many of the modern consumers face difficulties in recognizing the different retail beef cuts. Thus, a solution was created through collecting a dataset for retail cuts and creating an algorithm to classify them. A dataset, which is available for public, of 7 different beef retail cuts was proposed. This dataset includes colored images from our own image library, a total of 1638 images for validation testing and training are used for this project. The deep learning neural network algorithm-based model was able to identify specific beef retail cuts. 5 models were used in this paper to reach the highest accuracy for the classification of our dataset (MobileNet, ResNet50, InceptionV3, EfficientNetB0 and our customized model). EffecientNetB0 pre-trained model is one of the best and easiest pretrained models in Keras CNN. The employment of this model, after training and data augmentation techniques, was able to achieve the highest accuracy by a 99.81%. Based on our trained model and the huge results, deep learning technology evidently showed a promising effort for beef cuts recognition in the meat science industry.**

I. Introduction

Food industry operations include the supply and usage of food items and services. It is vital to any country's economic growth. It is one of the most active economic sectors on the planet. Meat is an essential element of the human diet, having significant health and economic and cultural ramifications across the world. Meat production is governed by convenience and availability. It involves a number of domestic species and is influenced by religious and cultural beliefs [1]. For many people all around the world, meat is an essential source of sustenance. Meat production has increased by more than 50 percent in the last 50 years which shows that global demand is increasing. Almost 340 million tonnes are produced worldwide each year. Meat is a good source of protein and it has an essential nutritional value. However, it rots easily and quickly due to its sensitivity which is caused by it's high nutritional value. Eliminating hunger is one of the main goals of sustainable development in order to achieve a high level of food security and saving resources. [2].Food fraud is a serious problem for most retail food industries. The huge difference between the prices and the quality of the retail beef cuts is a common problem among many consumers. [3], hence, researchers have increased their scrutiny of classifying beef cuts. Classifying beef cuts is a common issue for those who don't have an experience in detecting the beef cuts.

Knowing the difference between the beef cuts is extremely important, since there are many consumers who haven't the experience to differentiate between them, so knowing the difference is essential to avoid commercial cheat, as there is a big difference in price and quality of each beef cut.

The classification of beef cuts is also indispensable for meat lovers and meat shops who like to make food, especially meat dishes. They need to classify beef cuts, which is a big problem. It takes time to classify and at the end the accuracy result is low because it needs a huge experience. The central challenges of this study shows that the extraction of the features of beef is extremely hard because of the common similarities between beef cuts like color extraction, muscle fibre detection and distribution of fats in each cut. These features are the ones where we were going to implement a method to extract them and differentiate between the retail beef cuts. The aim of this paper to is facilitate the process of differentiating between the retail beef cuts and type of meat to make it easier for the customers who have no experience in meat and to avoid commercial cheat, speeding up the production process and reduce human faults.

AI has been used to recognise a variety of targets, including text/words, illness expression, food recognition, and identity authentication systems [4].Deep learning techniques have been used for recognition and classification of many things like recognition of face, pattern, disaster and voice, and classification of food like bacon and food species.CNN is a well-known deep learning method that has been frequently used in the issues classification. The most notable benefit of CNN is its capacity to learn from an input image without having to extract features [5].

II. Related Works

The exploration of industrial classification was limited to 3 popular techniques of Electronic nose, Image processing, and Deep learning neural network [6]. However, the introduction of electronic noses paved the door for a new type of complicated sample analysis that was not based on the traditional analytical chemistry method of component separation, but rather on a synthesis of global chemical information [7]. Image processing is the process of transforming an image into digital form and performing operations on it in order to enhance the image and extract useful information. Image processing is described mathematically as the computer's processing of a two-dimensional image, i.e., an image is defined as a function of two real variables, such as t(x, y), with an amplitude such as the brightness of an image at a coordinate point (a, b). A picture or a set of attributes or qualities connected to the image might be the result of image processing [8]. Deep Learning is a relatively new machine learning subject that has sparked a lot of attention in recent years. It has been widely used in a variety of applications and has shown to be an effective machine learning technique for a variety of challenging issues like image classification in animal science [9].

A. E-Nose detection

The electronic nose is one of the most accurate and efficient technologies for food quality detection. It implemented a method using SR SNR maximums (SNRmax) linear fitting regression to detect freshness of beef strip lions. It reached a high level of accuracy. Consequently, questions have been raised about the highness of this accuracy level. (Xiao et al, 2014) [10]. The presented work subdivided the degree of freshness into 3 levels which are Fresh, Sub Fresh and putrid. The detection of freshness degree using the electronic nose method was applied on pork meat, mutton meat and beef meat. The results of the experiment showed freshness degree of 89.5% for pork, 84.2% for beef, and 94.7% for mutton. However, the accuracy level of the experiment is incomplete as it did not include the type of the beef cut. The Electronic nose was reported in the later models of Xiaohui Weng et al. [11]. The models evaluated a new method by using E-nose, computer vision, and artificial tactile all together to detect the freshness of meat. It added a new feature. It succeeded in predicting the exact number of days of safe food storage. The results of the study showed accuracy levels of 93.47% for pork meat Unfortunately, the experiment did not include beef meat. The impact of this technology on beef meat is yet to be discovered.

B. Image processing techniques

In the last five years, there has been an increase in using image analysis on animal sciences. A study reported that the merging between computer vision, digital image processing, and digital image analysis is extremely important in applications concerning animal sciences (Fernandes et al) [12]. This merging of technologies is applicable on high-throughput challenges and precise livestock farming applications. (Andrei et al) [13] implemented a system with fuzzy logic based classifier to score meat marbling. They also proved that artificial intelligence has been shown to be a useful tool in meeting the ever-increasing demands of technology advancement. The study's overall goal was to provide a new creative approach of analysing typical events that would benefit the majority of people. Kusworo Adi et al [14] have introduced a new methodology to determine the quality of the meat using phone's camera. In addition to they implemented an automated system to classify the quality of the meat using image processing digital images. They obtained 90% accuracy for training and 84% for testing. Regardless of the progress in the area of image processing for food quality detection, the major problem of thresholding persists. According to (Asmara et al,2018) [15] they demonstrated a new technology using Grey Level Co-Occurrence Matrix method to classify pork and beef with color and texture extraction. Beside They used back-propagation algorithm to identify the digital image of the meat, achieving 89.57% of accuracy results. Despite its long clinical success, there are another types of meat rather than pork and beef which need to be identified.

C. Deep Learning Neural Network Techniques

According to (Narit and Smith,2018) [16], they used deep learning neural network techniques to generate a CNN model predicting the Thai fast food, they trained the data set on 3,960 images using GoogleNet framework. In addition to they reached average accuracy 88.33%. However they proved that neural network is one of the most essential techniques to classify food. Food image recognition was reported later using convolutional neural network(Xin et al,2019) [17] they implemented a method to avoid commercial cheat for rice in Malaysia, they achieved high accuracy of 92.6%. Moreover they proved that CNN algorithm is the best model to be trained to classify types of food. Since researchers have increased their scrutiny of knowing how well a deep learning neural network is, Saidul et al [18] demonstrate that past research to prove this method by the usage of two pretrained models from CNN (VGG16 and ResNEt V2). Beside they used these models for only 7 cuts with accuracy reached 98.6% for the first model and 95.7% for the second one. However, these rapid changes are having serious effects on the field of animal science applications.

State of Art	Our Work
1. The dataset consists of 7 classes of only types of steak not all beef cuts which are (bone in rib eye steak, boneless rib eye steak, chuck steak, flank steak, New York strip, short rib, and tenderloin). 2. They only used 2 models.	1. We have collected a dataset of different retail beef cuts which are (Liver, Roast Beef, Beef Chuck, Beef Round, Strip-Lion, Round Fillet, Beef Flank). 2. We have used 5 CNN models to classify this beef cuts and to differentiate the performance between them. 3. We showed the effect of data augmentation on the performance of model in multi class classification.

Fig. 1. Comparison between our work and state of art

III. Methodology

The proposed system used deep learning technique to classify beef cuts using CNN. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input picture, assign significance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them. When compared to other classification methods, the amount of pre-processing required by ConvNet is significantly less. While basic techniques need hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training [19].

A. Architecture

We have used in this paper 5 architectural models to train and test our data (MobilNet, ResNet50, InceptionV3, EfficientNetB0 and our customized model), using sequential model API. The sequential model is suited for a simple stack of layers with precisely one input tensor and one output tensor for each layer [20]. According to (Atila, 2021) [21] MobileNet, ResNet50, InceptionV3 and EfficientNetB0 are the highest accuracy in the pretrained models for CNN. The performance of this proposed architectures compared to other image classification models is extremely high.

We have used other existing neural network pretrained models to increase the confidence of the result, hence we will explain what is special about every model.

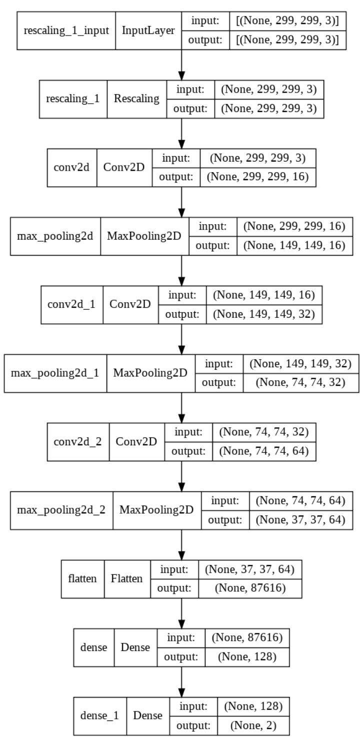


Fig. 2. Customized Model Layers

Figure 2 shows the layers of our customized model which consists of 11 layers. One of the most important phases in the pre-processing of data before developing a machine learning model is feature scaling in machine learning. The reason for using feature scaling is that few algorithms converge much faster with it than without it [22]. Maximum pooling is a pooling operation that computes the highest, or biggest, value within every patch of each convolution layer [23].Conv2D is a 2D Feature Extraction that produces a set of outputs by winding a mapping function with the layer's input [24]. The reason we used flatten layer that a tensor flatten operation reconfigures the convolutional layer to get a form equal to the number of components in the convolution excluding the the batch dimension [25]. While Dense layer was used because it aids in modifying the dimensions of the previous layer's output such that the model to readily establish the relationship between the values of the data wherein the model is operating [26].

MobileNet does not employ ordinary convolutions, but rather Depth wise separable convolutions, which cost one-eighth as much to compute. To address the aforementioned issue, a lighter model known as MobileNet was developed, which has fewer parameters and takes less time to compute, but rather depth wise separable convolutions, which cost one-eighth as much to compute. There are two hyper parameters in MobileNet: width multiplier and resolution multiplier [27].

According to ResBlocks, ResNet50 is separated into four components. After each shallow segment, a classifier is set, along with a bottleneck layer and a fully connected layer that are only used in training and may be deleted in inference. The major reason for including the bottleneck layer is to reduce the effect of each shallow classifier and to incorporate L2 loss from hints. During the training phase, all shallow portions with associated classifiers are taught as student models by distillation from the deepest part, which may be thought of as the teacher model [28]. Inception V3 lowers the number of convolutions by restricting the maximum filter size to three, enhances the network depth, and employs a better feature combining approach for each inception module [29].

EffecientNetB0 utilises a compound scaling search based on depth, width, and input size in EfficientNet. EfficientNet's fundamental design concept is to deconstruct the modules with various object detection functions, then scale the picture size, width, #BiFPN layers, and #box/class layer. SpineNet, which is primarily focused at the overall architecture of fish-shaped object detector for network architecture search, is another design that employs the NAS idea. This design approach might result in a scalepermuted structure at the end [30].

We also want to demonstrate the best features of each model to show why we chose these models. MobileNet is one of the best pretrained models because it reduces the size of the model after training. It's very light and has high speed in computation [31]. To reduce time and processing power, the EfficientNet design leverages transfer learning. As a result, it outperforms other well-known models in terms of accuracy. This is because some smart scaling at the depth, breadth, and resolution levels [32]. The Inception model is extremely effective in classifying remote-sensing photos. In a typical convolutional neural network, we must decide whether to use a pooling operation or a convolution operation, as well as the filter size. All of these procedures may be done simultaneously with Inception [33]. ResNet extracts features from pictures using a neural architecture, then uses fully-connected classifiers to give labels to images based on the retrieved features [34].

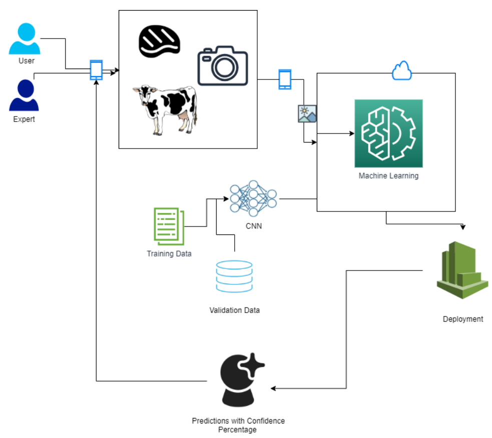


Fig. 3. System Overview

Figure 3 shows the detailed implementation of our system. The system takes a photo of the beef cut from a device then it sends the photo to the cloud using machine learning algorithms. The model categorizes the inputted photo according to features extracted compared by the training data. After the prediction of the type of the beef cut, the deployment process starts by sending the results and predictions to the user.

B. Beef cuts collection

In the dataset collection, 7 different beef retail cuts were used, (Liver, Roast Beef, Beef Chuck, Beef Round, Strip-Lion, Round Fillet, Beef Flank). 1638 images collected from our image library where used for training and validation, 80% was for training and 20% was for validation and testing. All the images were extracted from 18 videos using ezgif tool and using 10 frame per second rate then filtering them manually. Using unknown data set with pretrained model will get persuasive outcomes, but sometimes you need to create your own model. Utilizing the weights of a pretrained model, which carries out information on millions of pictures from ImageNet, is one of the most practical aspects of using transfer learning [35]. This method not only saves time while training, validating, and testing the model, but it also increases the overall accuracy of prediction and classification. [36]

Retail Beef Cuts Dataset. (2022, January 30). [Dataset]. Abdallah Abuzaid. https://www.kaggle.com/abdallahabuzaid/retail-beef-cuts-dataset

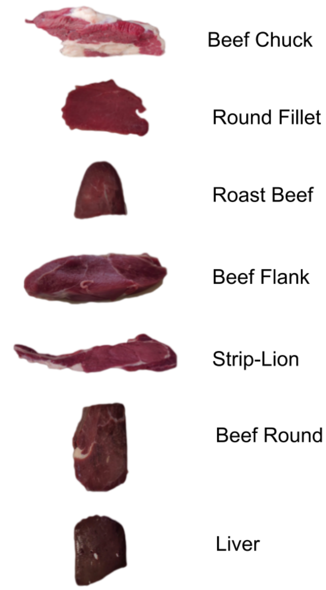


Fig. 4. Dataset Samples

IV. Results

In this research paper, as we show in the below table, the selected models have reached high accuracy. The highest accuracy we reached was by EffecientNetB0 after data augmentation, then the accuracy reached 99.81% for training and 85% for validation after data augmentation while before augmentation, the accuracy was 99.43% for training and 79.37% for validation. For our customized model the accuracy reached 99.89% before data augmentation then 97.34% after data augmentation. The InceptionV3 model reached 99.05% after data enhancement, and it reached 98% before, which is a very high rate. Resnet50 model is the perfect model for multiple class classification, as it reached 99.34% accuracy for training before augmentation of data. Our customized model reached 97.34% accuracy.

Data augmentation was used to avoid overfitting but it doesn't seem to be a good approach in some models as it decreases the accuracy. Deep learning models are data hungry. If the model is unable to detect patterns due to a lack of data, it will attempt to remember the dataset. Because they are large enough, larger models tend to remember data rather than identifying patterns. When a model memorises training data, it will perform admirably on the training set but badly on the validation set. The approach for regularising data is data augmentation. Your model weights are penalised higher during regularisation to ensure that they don't overfit. As a consequence, your model may not perform well on the training set (depending on how much regularisation is done), but it will attempt to uncover generic patterns in the dataset, which will aid validation [37]. That's why data augmentation may lead to decrease the accuracy of training and increase the accuracy of validation in some models.

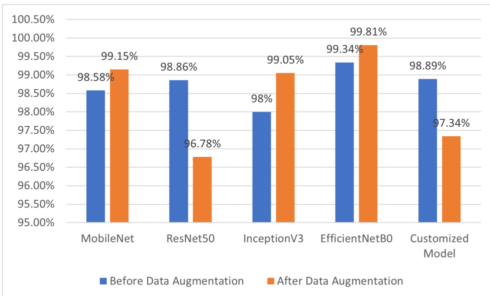


Fig. 5. The Accuracy Results of 5 Models

Fig. 6. Accuracy and Loss curve of EffecientNetB0 model

Model	Before Data Augmentation	No of epochs	After Data Augmentation	No of epochs
Resnet50	98.86%	10	96.78%	15
MobileNet	98.58%	10	99.15%	15
EfecientNetB0	**99.34%**	**10**	**99.81%**	**15**
InceptionV3	98.0%	10	99.05%	15
Customized Model	99.89%	10	97.34%	15

Table 1. Accuracy Results

Figure 7 shows the classification report of the highest accuracy CNN model. The accuracy reached 86% for precision, recall and f1-score. The number of correct positive divided by the total number of positive predictions is referred to as precision. The recall is determined as the ratio of Positive samples that were properly categorised as Positive to the total number of Positive samples. The recall metric assesses the model's ability to detect Samples were positive. While f1 Score is calculated as the weighted average of precision and recall.

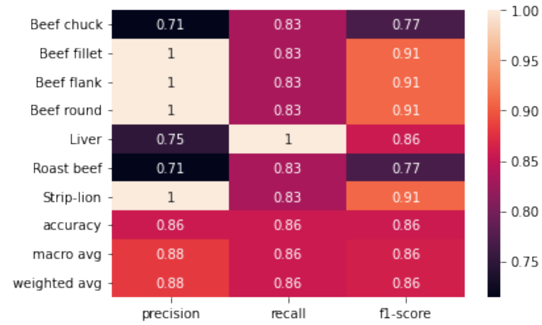


Fig. 7. Classification Report of EffecientNetB0

Figure 8 shows the confusion matrix of our proposed system. A confusion matrix is a summary of classification problem prediction outcomes. The number of right and wrong predictions is evaluated with count numbers as divided by class.

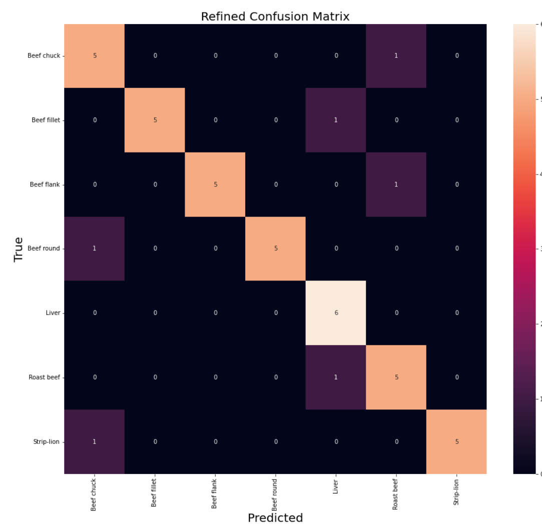


Fig. 8. Confusion Matrix of EffecientNetB0

V. Conclusion

The increase of meat consumption was a motivation which developed this system to make it easier for the consumers when differentiating between the retail beef cuts. Meat is essential part in one's diet and rich source of protein. Despite the fact that it's a high nutritional content renders it's susceptible to blight and decay. Eliminating hunger and achieving a high degree of food security since retaining nutrition is one of the sustainable development goals. The main point was to collect a dataset of the popular retail beef cuts and try to classify them using CNN models. We aimed to prove that convolutional neural network is a good approach to use in the classification of meat for the industry. We have collected 18 videos for 7 different retail beef cuts (Liver, Roast Beef, Beef Chuck, Beef Round, Strip-Lion, Round Fillet, Beef Flank). The extracted 1638 digital images from these videos were used as 80% for training the model and 20% for validation. After using 5 models (MobileNet, ResNet50, InceptionV3, EfficientNetB0 and our customized model), we have reached huge results with a great accuracy for the pretrained model EfficientNetB0 which reached 99.81% after data augmentation. This proved that CNN demonstrated a promising effort in the meat science business to recognise retail beef cuts.

VI. Future Work

After the huge results ,the convolutional neural network models have motivated us for completing our work by collecting more dataset for more retail beef cuts to increase the number of classes. Also, our aim is to recognize the origin of the beef by detecting if it's local or imported to help our objective of avoiding commercial cheat. The last thing we aim to do is to differentiate between beef and other types of meat like mutton and pork meat which will help in the production of meat industry .

References

[1] G. Paredi, M.-A. Sentandreu, A. Mozzarelli, S. Fadda, K. Hollung, and A. M. de Almeida, "Muscle and meat: New horizons and applications for proteomics on a farm to fork perspective," *Journal of Proteomics*, vol. 88, pp. 58–82, 2013, special Issue: New Horizons and Applications for Proteomics [EuPA 2012]. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874391913000596

[2] G. Rasul, "Managing the food, water, and energy nexus for achieving the sustainable development goals in south asia," *Environmental Development*, vol. 18, pp. 14–25, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211464515300646

[3] E. Giampietri, A. Finco, and T. Del Giudice, "Exploring consumers' behaviour towards short food supply chains," *British Food Journal*, vol. 118, no. 3, pp. 618–631, Jan 2016. [Online]. Available: https://doi.org/10.1108/BFJ-04-2015-0168

[4] I. Kumar, J. Rawat, N. Mohd, and S. Husain, "Opportunities of artificial intelligence and machine learning in the food industry," *Journal of Food Quality*, vol. 2021, p. 4535567, Jul 2021. [Online]. Available: https://doi.org/10.1155/2021/4535567

[5] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[6] B. G. Kermani, S. S. Schiffman, and H. T. Nagle, "Performance of the levenberg–marquardt neural network training method in electronic nose applications," *Sensors and Actuators B: Chemical*, vol. 110, no. 1, pp. 13–22, 2005.

[7] C. Di Natale, R. Paolesse, A. Macagnano, A. Mantini, A. D'Amico, A. Legin, L. Lvova, A. Rudnitskaya, and Y. Vlasov, "Electronic nose and electronic tongue integration for improved classification of clinical and food samples," *Sensors and Actuators B: Chemical*, vol. 64, no. 1-3, pp. 15–21, 2000.

[8] B. Basavaprasad and M. Ravi, "A study on the importance of image processing and its applications," *IJRET: International Journal of Research in Engineering and Technology*, vol. 3, p. 1, 2014.

[9] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Classification using deep learning neural networks for brain tumors," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 68–71, 2018.

[10] Y. Xiao, J. Jiaojiao, H. Guohua, Y. Fangyuan, W. Minmin, H. Jie, Y. Xiaoguo, and D. Shanggui, "Determination of the freshness of beef strip loins (m. longissimus lumborum) using electronic nose," *Food Analytical Methods*, vol. 7, no. 8, pp. 1612–1618, Sep 2014. [Online]. Available: https://doi.org/10.1007/s12161-014-9796-8

[11] X. Weng, X. Luan, C. Kong, Z. Chang, Y. Li, S. Zhang, S. Al-Majeed, and Y. Xiao, "A comprehensive method for assessing meat freshness using fusing electronic nose, computer vision, and artificial tactile technologies," *Journal of Sensors*, vol. 2020, p. 8838535, Sep 2020. [Online]. Available: https://doi.org/10.1155/2020/8838535

[12] A. F. A. Fernandes, J. R. R. Dórea, and G. J. d. M. Rosa, "Image analysis and computer vision applications in animal sciences: An overview," *Frontiers in Veterinary Science*, vol. 7, p. 800, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fvets.2020.551269

[13] A. Andaya, E. Arboleda, A. Andilab, and R. Dellosa, "Meat marbling scoring using image processing with fuzzy logic based classifier," *International Journal of Scientific & Technology Research*, vol. 8, pp. 1442–1444, 08 2019.

[14] K. Adi, S. Pujiyanto, O. Dwi Nurhayati, and A. Pamungkas, "Beef Quality Identification Using Thresholding Method and Decision Tree Classification Based on Android Smartphone," *Journal of Food Quality*, vol. 2017, p. 1674718, 2017. [Online]. Available: https://doi.org/10.1155/2017/1674718

[15] R. A. Asmara, R. Romario, K. S. Batubulan, E. Rohadi, I. Siradjuddin, F. Ronilaya, R. Ariyanto, C. Rahmad, and F. Rahutomo, "Classification of pork and beef meat images using extraction of color and texture feature by grey level co-occurrence matrix method," *IOP Conference Series: Materials Science and Engineering*, vol. 434, p. 012072, dec 2018. [Online]. Available: https://doi.org/10.1088/1757-899x/434/1/012072

[16] N. Hnoohom and S. Yuenyong, "Thai fast food image classification using deep learning," in *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*, Feb 2018, pp. 116–119.

[17] M. J. Nordin, O. Xin, and N. Aziz, "Food image recognition for price calculation using convolutional neural network," 02 2019, pp. 80–85.

[18] S. GC, B. Saidul Md, Y. Zhang, D. Reed, M. Ahsan, E. Berg, and X. Sun, "Using deep learning neural network in artificial intelligence technology to classify beef cuts," *Frontiers in Sensors*, vol. 2, p. 5, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fsens.2021.654357

[19] A. F. Al-Daour, M. O. Al-Shawwa, and S. S. Abu-Naser, "Banana classification using deep learning," *International Journal of Academic Information Systems Research (IJAISR)*, vol. 3, no. 12, 2020.

[20] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.

[21] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021.

[22] E. T. Bekar, P. Nyqvist, and A. Skoogh, "An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study," *Advances in Mechanical Engineering*, vol. 12, no. 5, p. 1687814020919207, 2020.

[23] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[24] V. Sineglazov, M. Omelchenko, and V. Hotsyanivskyy, "Comparative characteristics of keras and lasagne machine learning packages," *Electronics and control systems*, no. 3, pp. 106–111, 2017.

[25] Y. Shi, U. N. Niranjan, A. Anandkumar, and C. Cecka, "Tensor contractions with extended blas kernels on cpu and gpu," in *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. IEEE, 2016, pp. 193–202.

[26] Z. Zhang and Y. Dong, "Temperature forecasting via convolutional recurrent neural networks based on time-series data," *Complexity*, vol. 2020, 2020.

[27] D. Sinha and M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 0280–0285.

[28] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3713–3722.

[29] S. Kulhare, X. Zheng, C. Mehanian, C. Gregory, M. Zhu, K. Gregory, H. Xie, J. M. Jones, and B. Wilson, "Ultrasound-based detection of lung abnormalities using single shot detection convolutional neural networks,"

in *Simulation, image processing, and ultrasound systems for assisted diagnosis and navigation*. Springer, 2018, pp. 65–73.

[30] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-yolov4: Scaling cross stage partial network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 029–13 038.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[32] G. Marques, D. Agarwal, and I. de la Torre Díez, “Automated medical diagnosis of covid-19 through efficientnet convolutional neural network,” *Applied soft computing*, vol. 96, p. 106691, 2020.

[33] N. Devi and B. Borah, “Refining the features transferred from pre-trained inception architecture for aerial scene classification,” *International Journal of Remote Sensing*, vol. 40, no. 24, pp. 9260–9278, 2019.

[34] D. McNeely-White, J. R. Beveridge, and B. A. Draper, “Inception and resnet features are (almost) equivalent,” *Cognitive Systems Research*, vol. 59, pp. 312–318, 2020.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[36] J. Yim, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 7130–7138.

[37] A. Hernández-García and P. König, “Data augmentation instead of explicit regularization,” *arXiv preprint arXiv:1806.03852*, 2018.

4.5.2 Forensic Handwritten Signature Identification Using Deep Learning

Forensic Handwritten Signature Identification Using Deep Learning [20]

Forensic Handwritten Signature Identification Using Deep Learning

Omar Tarek
Faculty of Computer Science,
October University for Modern
Sciences and Arts (MSA).
Giza, Egypt
omar.tarek22@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers and
Artificial Intelligence, Helwan University.
Faculty of Computer Science, October University
for Modern Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—Forgery is a type of fraud defined as the act of forging a copy or an imitation of a document, signature, or banknote which is considered a form of illegal criminal activity. In this paper, we are focusing on the identification and detection of handwritten signature forgeries inside documents. The proposed system uses contemporary methods that utilize a deep learning approach of CNNs (Convolutional Neural Networks) for binary image classification and aims to help forensic examiners measure the genuineness of handwritten signatures. We considered using a number of five different classification models of CNN which are, VGG-16, ResNet50, Inception-v3, Xception, and Our CNN model. The purpose for using these different CNN models is to determine and study which model is best at identifying images containing text data containing similar resemblances. Upon comparing these CNN models, we concluded that the ResNet50 model was able to reach the highest score at identifying handwritten signatures with an accuracy of 82.3% and 86% when tested on datasets of 300 images and 140 images respectively. Regarding future work, this is a required step that determines what model to focus on for more in-depth analysis and classification of the characteristics of handwritten signatures.**

***Index Terms*—Forensic Document Examination, Handwritten Signatures, Forgery Detection, Image Classification, CNN Architectures**

I. Introduction

Forensic analysis is the application of scientific methods for the identification and examination of scientific evidence during a criminal investigation. FDE (Forensic Document Examination) is a forensic science that focuses exclusively on examining documents concerning emails, business letters, transactional documents, financial reports, bank signatures, or other criteria. Forgery is the number one crime that is most commonly associated with FDE. It involves the recreation of a false document, signature, or other imitation of an object of value to be used with the intent to deceive another. And it is stated by Interpol that document fraud is considered a form of illegal criminal activity. In this paper, we will be focusing on the identification of handwritten signatures inside documents by using state-of-the-art technologies that utilize artificial intelligence and deep learning techniques.

Handwritten signature forgery can be identified by a set of features and characteristics that most forensic document examiners are familiar with. However, many of these features can cause confusion for the examiners especially when a forgery is performed by a skilled perpetrator. The proposed system solves this problem by using deep image classification methods that focus on comparing multiple combinations of comprehensive signature features.

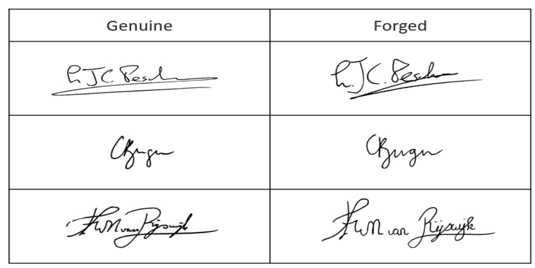


Fig. 1. Genuine Signatures vs Forged Signatures

The handwritten signature samples shown in figure 1 represent the level of challenge encountered by our classification models by showing close similarities between each class. To tackle this problem, we will be using a combination of five different CNN models for the binary image classification of handwritten signatures. With that in mind, the classification models must go through a sufficient amount of training, validating, and testing to reach acceptable results. The dataset used consisted of a sample containing 300 images of handwritten signatures (150 Genuine Signatures + 150 Forged Signatures).

The main contribution of this paper is to explore and compare different CNN models to determine the best-performing model in terms of classification of images containing text data which aims to help forensic examiners with identifying the genuineness of the handwritten signatures. The system uses five different CNN models which are, VGG-16, ResNet-50, Inception-v3, Xception models, and our Custom CNN Model.

II. Related Work

Because of the rapid rise of fraudulent activities with fraudsters now taking advantage of contemporary computer software for illegal behaviors and illicit purposes, lots of research efforts were dedicated to this subject for the following topics which are (A) forged document detection, and (B) handwritten signature identification.

A. forged document detection

In document forgery detection, Praba et al. [1] has presented a system proposing two methodologies for detecting forged documents, the first is by scanning a QR-code of the document to determine its source, the second way is by using a machine-learning algorithm to train, test, and classify the questioned document to determine whether it is forged or not. Alternatively, James et al. [2] have used a data-driven approach that utilizes OCR (Optical Character Recognition) graph features to detect document manipulations on a constructed dataset of real business documents containing slight forgery imperfections. Yoosuf et al. [3] has proposed a method that applies cognitive techniques to identify and detect document forgery in an information management system by using an automatic document verification model utilizing CNN then applying OCR and LBP (Linear Binary Pattern) to extract textual information and regional edges before using ORB (Oriented fast and Rotated Brief) to extract images from the scanned documents, the MIDV-500 dataset which contains 256 Azerbaijani passport images was used to train the CNN model that uses sliding window operations layers to evaluate authenticity. Roy et al. [4] has presented a system that conducts forensic analysis on handwriting for identifying forgery owing to word alteration. The proposed system uses a Multilayer Perceptron classifier adopted to classify data instances computed by extracting color-based statistical features, the system managed to achieve an accuracy of 83.71% for blue pen data and 78.18% for black pen data.

For copy-move forgery detection and localization, Abdalla et al. [5] has used a deep learning approach that utilizes deep Convolutional learning algorithms, this paper investigates copy-move forgery detection by utilizing a fusion processing model of GANs (General Adversarial Networks) and CNN on four different datasets with an accuracy approximate of 95%. Khan et al. [6] have presented an automated deep learning for forgery detection of ink mismatch in hyperspectral document images using CNN-friendly image format after extracting ink pixels from a hyperspectral document image to be fed to the CNN for classification, the proposed method identifies different inks in hyperspectral document image with an effective accuracy of 98.2% for blue ink and 88% for black ink on the UWA Writing Ink Hyperspectral Images (WIHSI) database.

Ranjan et al. [7] has proposed a framework for image forgery detection and classification using machine learning, the paper lays a foundation for the investigation of digitally manipulated images by providing a solution to distinguish between such images with an accuracy of 96.4%. Adi et al. [8] has presented a system for combining perceptual hash and OCR for securing and authenticating printed documents from forgery attacks. Bunk et al. [9] has presented a system that uses two deep learning methods for detection and localization of image forgeries by utilizing resampling features. The first method uses overlapping image patches to compute the radon transform of resampling features before using deep learning classifiers and Gaussian conditional random field model to create a heatmap to then identify tempered regions using the Random Walker segmentation method. In the second method, what differs is that the computed resampling features on overlapping image patches pass through an LSTM (Long Short Term Memory) network for classification purposes. Vieira et al. [10] has proposed an information system for automation management of counterfeited document images using a two-fold approach that utilizes the OpenCV framework which is used to compare images, match patterns, and analyze textures/colors tested on a Portuguese citizen card. Ghosh et al. [11] has presented a system for detection and localization of image and document forgery using CNN classifier.

Shang et al. [12] has presented a document print and copy forgery detection by analyzing different printer types (laser printers, inkjet printers, and electrostatic copiers), the proposed method can distinguish between each type based on features extracted from characters in the documents which was able to achieve an accuracy of 90% that also works with JPEG compression. Bertrand et al. [13] has presented a system that is based on finding intrinsic features for detection of fraudulent documents by proposing a method which is based on detecting outlier characters in a discriminant feature space and the detection of strictly similar characters, each character then is classified as a genuine one or a fake one.

B. handwritten signature identification

Other than checking for document authenticity, forged handwritten signatures is one of the most common things that would be found on a document, must not be overlooked. Poddar et al. [14] has presented a system that uses a deep learning approach to recognize and detect forged handwritten signatures using CNN, Crest-Trough method, SURF algorithm, and Harris Corner detection algorithm, the proposed system managed to attain an accuracy of 90%-94% for recognizing a valid signature and an accuracy of 85%-89% for detecting forged signatures. Kao et al. [15] has proposed a method based on a single known sample for offline signature verification and forgery detection using an explainable deep learning approach that utilizes DCNN (Deep Convolutional Neural Network) backed by a unique local feature extraction method, this system was capable of achieving an accuracy between 94.37% add 99.96% with false rejection rate (FRR) between 5.88% and 0% and a false acceptance rate (FAR) between 0.22% and 5.34% when used on the Document Analysis and Recognition (ICDAR) 2011 SigComp dataset. Wei et al. [16] has presented IDN (Inverse Discriminative Network) for verification of handwritten signatures that aims to differentiate between genuine signatures and forged signatures, this model was used on a Chinese signature dataset of 29,000 images from 749 individuals to test on different datasets of different languages as well which are: CEDAR, BHSig-B, and BHSig-H. Pokharel et al. [17] has proposed a deep learning system for handwritten signature recognition that uses a CNN architecture, they used a transfer learning method to retrain the GoogleNet model on 25 classes of a signature image dataset where each class contains 85 signatures to achieve a mean testing precision of 95.2%.

III. Proposed Methodology

We proposed a binary image classification system that uses a combination of different CNN models to classify these images of handwritten signatures. The goal of this system is to find the most effective classification method that works best with text-based images by testing each method separately and comparing the attained outcome. This system works by taking an input of a scanned image containing the signature and using binary image classification models in order to validate the processed image.

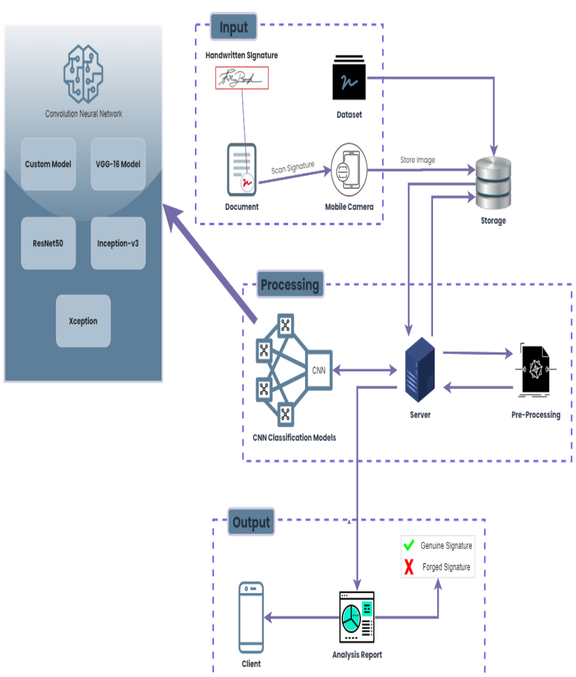


Fig. 2. Overview

A. Convolutional Neural Network

A CNN is a type of artificial deep learning neural network that is used most in image recognition/classification and is commonly associated with computer vision. It contains multiple layers called convolutional layers which are based on the convolution operation that works by combining two functions (input image, image filter) to produce a third function.

Convolution Operation:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

$(f * g)(t) =$ functions that are being convoluted
$t =$ real number variable of functions f and g
$g(t) =$ convolution of the function $f(t)$
$d\tau =$ first derivative of $g(\tau)$ function

An image consists of a set of arrays of different numbers representing squares of pixels arranged in rows and columns, which is called an Image Matrix. The CNN works by taking an input image in the form of an image matrix to extract features from the image and classify them based on learned features.

B. Pre-Processing

Before introducing the classification models, a pre-processing phase must take place. The methods used in this phase are simple but rather very important in order to allow the classification models to perform most effectively on the signature images. Each image matrix is rescaled to grayscale and resized to a default resolution of 320 X 240, which is going to be the same as the input layer (first layer of the classification model in Fig. 3) for all the different classifiers.

C. CNN Models

The proposed system uses the following 5 CNN image classification models, which consist of our Custom CNN model and four other pre-trained models (VGG-16 model, ResNet50 model, Inception-v3 model, and Xception model).

First, the Custom CNN classification model is of type sequential, which consists of 3 convolutional layers that use RELU (Rectified Linear Activation Function), three max-pooling layers, and two dense layers with the output layer using SOFTMAX activation function for binary classification.

Second, to configure the proposed pre-trained models which were trained on the ImageNet dataset through a process of Transfer Learning, we were able to fine-tune each model by removing its current input and output layers so that we can replace them with our own inputs and outputs. For the input, we added a new layer with our desired input data. As for the output, we added a new flatten layer to flatten the pre-trained model into a 1 Dimension followed by a dense layer of 512 neurons that uses RELU activation function before adding the final output dense layer for binary classification that consists of the Softmax activation function and 2 categorical classes which are later defined as either a forged signature or a genuine signature.

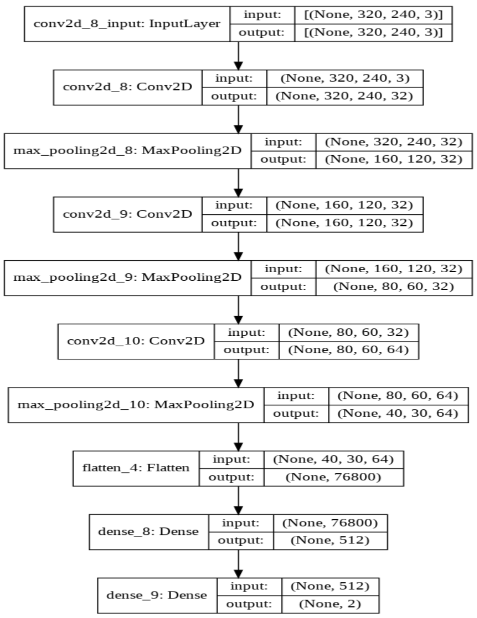


Fig. 3. Our CNN Model

IV. Experiment

We conducted a total of two experiments. The purpose of the first experiment is to test all of our CNN models to determine which model has the best results of the classification task on a dataset of 300 images of handwritten signatures. As for the second experiment, the objective is to use the model that achieved the highest score from the first experiment to test it on a newly collected dataset that contained 130 images of handwritten signatures.

A. *experiment 1*

In this experiment, we set up each of our five models for training and testing on a dataset of 300 images[1] for handwritten signature samples that consisted of 150 genuine signatures and 150 forged signatures. The main objective of this experiment is to compare the outcome of the different models and determine which model is best at classifying text-based images.

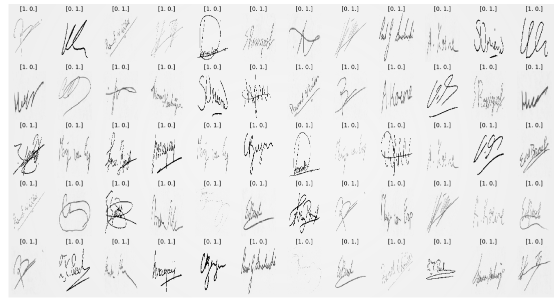


Fig. 4. Handwritten Signature Dataset 1

B. *experiment 2*

The experiment objective is to test the model that achieved the highest score from the first experiment on a new dataset of 140 images[2] to see if it can deliver better results. First, we worked on creating a new dataset by gathering new handwritten signature samples from different individuals, each individual providing five samples of their handwritten signature. Then we attempted to forge the gathered signatures to best resemble their genuine signature as a skilled perpetrator would likely do in an event of a forgery.

Fig. 5. Handwritten Signature Dataset 2

For the experiment, we used our created dataset which provides the input data of 140 handwritten signature samples belonging to 14 different individuals where half of them are forged. Both of the forged and genuine signatures are then introduced to the classification model to being the training phase. We obtained 10 signature samples per individual where the used dataset ratio of training to testing was 6 : 4. For each 10 signature samples, we used only 6 samples for training the model and the remaining 4 samples for testing the trained model. In the training phase, the 6 samples are categorized into 3 genuine signatures and 3 forged signatures. As for the testing phase, the 4 samples were categorized into 2 genuine signatures and 2 forged signatures.

[1] Handwritten Signatures Dataset 1: https://www.kaggle.com/divyanshrai/handwritten-signatures

[2] Handwritten Signatures Dataset 2: https://github.com/OmarTIbraheem/FDE/tree/main/Datasets/Handwritten%20Signatrue%20Dataset%202

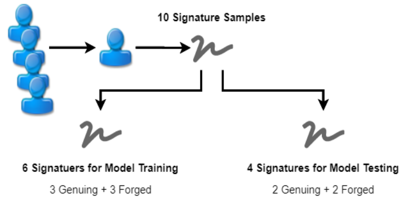


Fig. 6. Breakdown of Collected Signature Samples

V. Results and Discussion

Concerning the first experiment, we were able to reach the following scores for each model as shown in table 1. The ResNet-50 model was capable to perform the most accurate of the 5 models by reaching an accuracy of 82.3%.

Model	Accuracy	Precision	Recall	F1 Score
CNN	0.75	0.80	0.67	0.73
VGG-16	0.75	0.70	0.87	0.78
ResNet50	**0.82**	**0.81**	**0.83**	**0.82**
Inception-v3	0.77	0.72	0.87	0.79
Xception	0.70	0.64	0.93	0.76

Table 1. Classification Models Report

We used the following mathematical relations to calculate each of the Accuracy, Precision, Recall, and F1 Score. It is also worth knowing that: TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative. Keeping in mind that we will be using our CNN model result values as an example.

$$TN = 0.83 \qquad FN = 0.33$$
$$\text{FP} = 0.17 \qquad \text{TP} = 0.67$$

Precision:

$$P = \frac{TP}{TP + FP} = \frac{0.67}{0.67 + 0.17} = 0.80$$

Recall:

$$R = \frac{TP}{TP + FN} = \frac{0.67}{0.67 + 0.33} = 0.67$$

F1 Score:

$$F1 = 2 \times \frac{Precision \times recall}{Precision + recall} = 0.73$$

The following figures show the classification results through visual representations of the confusion matrix and reveal where the confusion occurs between both false positive and false negative predictions for each model. The reason for these false predictions is due to the not having enough images of the same type of signature as it is expected to have much lower false predictions if provided.

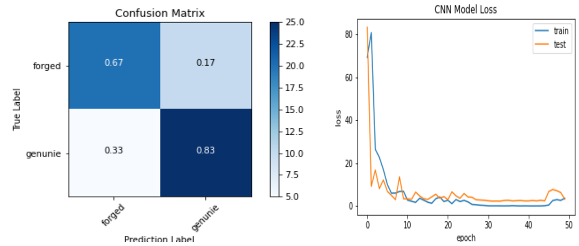


Fig. 7. Our CNN Model Confusion Matrix and Loss Curve

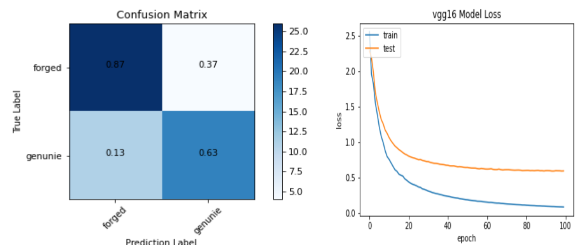


Fig. 8. VGG16 Confusion Matrix and Loss Curve

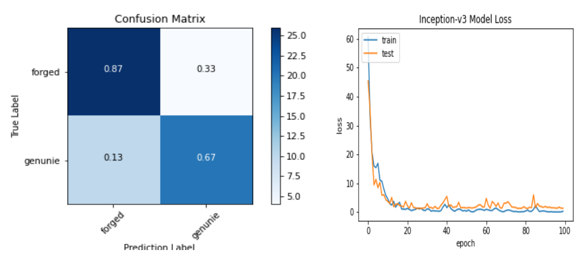


Fig. 9. Inception-v3 Confusion Matrix and Loss Curve

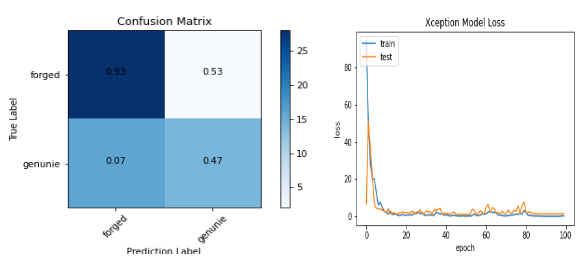


Fig. 10. Xception Confusion Matrix and Loss Curve

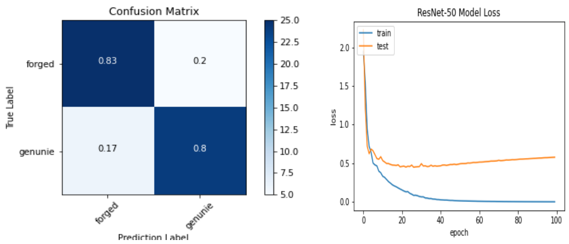


Fig. 11. ResNet50 Confusion Matrix and Loss Curve for **Experiment 1**

As for the second experiment, we used the model with the highest rating from the first experiment, which is the ResNet50 model. After testing the ResNet50 model on the newly collected dataset, which was able to reach a higher precision score since the model is performing on a lower number of signature images from the second dataset. The accuracy reached by this model is 86% which is relatively close compared to a recent work presented by Poddara et al. [14] on using CNN combined with Crest-Trough method, SURF algorithm, and Harris Corner detection algorithms to achieve an accuracy of 85% - 89% for forgery detection of offline signatures.

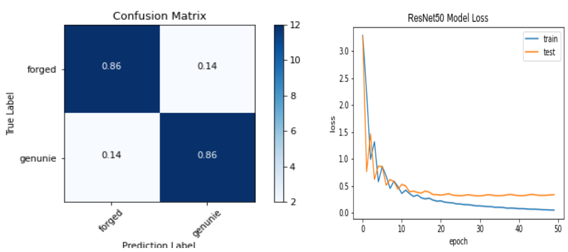


Fig. 12. ResNet50 Confusion Matrix and Loss Curve for **Experiment 2**

VI. Conclusion and Future Work

In this paper, we have proposed five different CNN classification models to determine which is the best-performing model for binary image classification of handwritten signatures inside documents. After testing each model, we concluded that the ResNet50 model has achieved the best results compared to the other CNN models regarding the classification of text-based images where it was able to reach an accuracy of 82.3% and 86% conducted on two datasets which contained samples of 300 and 140 handwritten signature images respectively.

Regarding future work, this paper provides useful information for suggesting the ResNet50 model and encourages further analysis that includes not binary but categorical classifications of handwritten signatures by utilizing different signature aspects and characteristics.

References

[1] G. C. Praba, E. Jeevitha, A. Abitha, A. Shalini, and B. Swetha, “Fake education document detection using image processing and deep learning,” *International Journal of Engineering Research Technology*, vol. 9, no. 5, Mar 2021. [Online]. Available: https://www.ijert.org/research/fake-education-document-detection-using-image-processing-and-deep-learning-IJERTCONV9IS05018.pdf, https://www.ijert.org/fake-education-document-detection-using-image-processing-and-deep-learning

[2] H. James, O. Gupta, and D. Raviv, “OCR graph features for manipulation detection in documents,” *CoRR*, vol. abs/2009.05158, 2020. [Online]. Available: https://arxiv.org/abs/2009.05158

[3] M. Yoosuf and R. Anitha, “Forgery document detection in information management system using cognitive techniques,” *Journal of Intelligent Fuzzy Systems*, vol. 39, pp. 8057–8068, 12 2020.

[4] P. Roy and S. Bag, “Forensic performance on handwriting to identify forgery owing to word alteration,” 01 2019, pp. 1–9.

[5] Y. Abdalla, M. T. Iqbal, and M. Shehata, “Copy-move forgery detection and localization using a generative adversarial network and convolutional neural-network,” *Information*, vol. 10, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2078-2489/10/9/286

[6] M. Khan, A. Yousaf, A. Abbas, and K. Khurshid, “Deep learning for automated forgery detection in hyperspectral document images,” *Journal of Electronic Imaging*, vol. 27, p. 053001, 09 2018.

[7] S. Ranjan, P. Garhwal, A. Bhan, M. Arora, and A. Mehra, “Framework for image forgery detection and classification using machine learning,” 06 2018, pp. 1872–1877.

[8] P. Adi and M. Luthfi, “An authentic and secure printed document from forgery attack by combining perceptual hash and optical character recognition,” 10 2019, pp. 157–162.

[9] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, “Detection and localization of image forgeries using resampling features and deep learning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Jul 2017, p. 1881–1889. [Online]. Available: http://ieeexplore.ieee.org/document/8014969/

[10] R. Vieira, C. Silva, M. Antunes, and A. Assis, “Information system for automation of counterfeited documents images correlation,” *Procedia Computer Science*, vol. 100, pp. 421–428, 2016, international Conference on ENTERprise Information Systems/International Conference on Project MANagement/International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916323468

[11] A. Ghosh, D. Zou, M. Singh, and Verisk, “Detection and localization of image and document forgery : Survey and benchmarking,” 2016.

[12] S. Shang, N. Memon, and X. Kong, “Detecting documents forged by printing and copying,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 140, Sep 2014. [Online]. Available: https://doi.org/10.1186/1687-6180-2014-140

[13] R. Bertrand, P. Gomez-Krämer, O. R. Terrades, P. Franco, and J.-M. Ogier, “A system based on intrinsic features for fraudulent document detection,” in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 106–110.

[14] J. Poddar, V. Parikh, and D. Bharti, “Offline signature recognition and forgery detection using deep learning,” *Procedia Computer Science*, vol. 170, pp. 610–617, 01 2020.

[15] H.-H. Kao and C.-Y. Wen, “An offline signature verification and forgery detection method based on a single known sample and an explainable deep learning approach,” *Applied Sciences*, vol. 10, no. 11, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/11/3716

[16] P. Wei, H. Li, and P. Hu, “Inverse discriminative networks for handwritten signature verification,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5757–5765.

[17] S. Pokharel, S. Giri, and S. Shakya, “Deep learning based handwritten signature recognition,” 02 2020.

4.5.3 Multi-Modal Online Exam Cheating Detection

Multi-Modal Online Exam Cheating Detection [6]

03/06/2022

ACCEPTANCE LETTER

Dear ***Ahmed Abozaid, Ayman Atia,***

Thank you for your submission to the ICECET 2022 conference. We are pleased to inform you that your paper entitled "**ID-467 Multi-Modal Online Exam Cheating Detection**" has been accepted as a full paper for **oral presentation** by the conference committee of *International Conference on Electrical, Computer, and Energy Technologies (ICECET)*. The event will take place in Prague, Czech Republic on 20-22 July 2022 **online** and **physically.**

We strictly follow "no podium, no paper" policy and only the papers that are presented at the conference will be submitted to IEEE Explore for publication. **At least one author** of an accepted paper must register (as a full participant) and participate in ICECET 2022 online or physically for the paper to be included in the proceedings. If you have not yet registered online (using the credit card or bank transfer options), at least one author of each paper should register to the conference via the online registration page at https://www.ecres.net/icecet. If you have already registered, please do not make another registration. Kindly note that your registration becomes valid only after your payment.

According to the conference regulations, only those papers which have been duly registered and presented on the conference day are considered for submission to IEEE Explore. The conference program will be communicated in due course.

We look forward to seeing you for a fruitful research and innovation event and for a great time in the wonderful environment of Prague

Yours sincerely,

Dr. Simon Winberg
Chair

Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET 2022)
20-22 July 2022, Prague-Czech Republic

Multi-Modal Online Exam Cheating Detection

Ahmed Abozaid
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Cairo, Egypt
ahmed.hamada1@msa.edu.eg

Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—Online exams have been the standard approach adopted by universities and institutes because of the COVID-19 pandemic which forces the world to go towards distance learning and online exams. But with this approach come the challenges such as online exam proctoring which is considered one of the most difficult challenges to solve. It is a must to ensure the academic honesty and credibility of the online exam. Existing proctoring techniques require a few proctors to observe a huge number of students to detect cheating students, and due to it is time-consuming and labor-intensive, we implemented multi-modalities to detect the student's activity during the online exam using a webcam and sent a report to the proctor for the suspected student. Those modalities are head-pose, object detection and eye-gaze estimation. This proposed solution is tested and evaluated on 29 students with a total of four exam sessions to ensure the effectiveness of our proposed solution. The events' detection accuracy of the multi-modalities experiment was 95.69%.**

***Index Terms*—Cheating Detection, online exams, multi-modalities ,proctoring, object detectoin, head pose, eye gaze.**

I. Introduction

In education, Online exams and distance learning have grown into popularity; they are approaches adopted by MOOCs (Massive Open Online Courses) like EDX, Udacity, Coursera, and many other platforms that focus on educating students and evaluating them using assignments, Quizzes, and online exams. Students have to pass those online assessments by getting the course certificate to certify that a student has passed the course. Changes in education that occurred during the pandemic affected 1.5 billion (90.1%) of total enrolled students [1]. From these changes, distance learning and online exams have became a popular approach adopted by universities, institutes, and schools. It seems beneficial to use information technology in education, but with this approach came some challenges. From these challenges is how to proctor the online exams in an efficient, convenient, and reliable manner.

According to Darwin and Carl [14] 74% of students in a survey said that it is easier to cheat in online exams than in traditional exams. In addition to 32.7% of those students admitted they cheated in online exams. Also, In our survey of 16 students asked them about the ease of cheating in online exams and the result was that 62.6% of the students said it is easy to cheat the online exams. The first and the second survey support the idea that proctoring is a must to ensure the academic honesty and credibility of the exams. In online exams, students use many methods to cheat, as they use their mobile phones to send messages, call their friends or even search online. They also use their books or ask someone else to help them during the exam. During the online exams, some institutes use a third-party conference call application to proctor the students. They open the students' cameras and mics to monitor them during their exam which is time-consuming and difficult for the proctor to monitor a large number of enrolled students at the same time. According to a survey by Li et al., existing online exam proctoring approaches are categorized into three groups: fully-automated proctoring, semi-automated proctoring, and manual proctoring. The most common approach is manual proctoring which is time-consuming and labor-intensive, as it requires few proctors to watch students' videos during the exam and detect any cheating behavior. The other approach is the fully automated proctoring which uses machine learning techniques on the students' recorded data to detect any cheating behavior [8]. However, this approach is very difficult to get a high accuracy result. The last approach which is the semi-automated approach is used to overcome the issues of the fully automated approach, which combines the manual approach and machine learning techniques to help the proctor focus only on the suspected students and review any reported unusual behavior. In this paper, we introduce a semi-automated approach that monitors students' activity to help proctor the online exams and detect any abnormal behavior. Our approach relies on multi-modalities combined together, Head-pose detection, object detection, and eye gaze estimation. The usage of multi-modalities are more effective than using one modality. If the student is focusing on the screen but there is a phone in his/her hand if there is a head pose modality only it won't detect that behavior as cheating behavior. So it is important to use more than one modality. In head pose, it detects the student's head direction whether the student is focusing on the screen or looking to the right or left. In object detection, it detects any forbidden object that shouldn't be used during the exam like cell phones or books; in addition to whether or not more than one person may be appearing in the camera. In the eye gaze modality, it detects the gaze direction if the student is focusing on his screen or left or right. The student's camera extracts a frame every 10 seconds and starts a process of modality extraction. Each modality process starts working sequentially with the received image and returns a report if it detects any abnormal behavior to be combined with other modalities report and sent to the human proctor to help on detecting cheating students easily.

II. Related Work

There are many methods for cheating in online exams. Therefore, many approaches are using different tools and techniques to detect cheating. This section provides different approaches to cheating detection and most of them use multi-modalities to increase cheating detection in online exams. In an approach proposed by Chirag et al, using their student's webcam, the system categorizes the student's Visual focus of attention by capturing head pose and eye gaze estimation using only a webcam during the online exams to detect cheating. They provide alerts to human proctors if a certain student has a visual focus of attention value greater than a certain specified threshold. The default model is the eye gaze model which decides the visual focus of attention. If it fails, it uses the head pose model. They use FSA-Net for head pose detection and a pre-trained Dlib model for gaze estimation; they achieved a total accuracy of 96.04% in attention metric classification. [12]. Another approach, taken by Ozgen et al. used recorded videos of students' webcams to apply a pipeline of cheating detection to detect if there is another person, the student's absence status, and the usage of any electronic device. The pipeline contains object detection, face detection and recognition, and face tracking. They used mobile-SSD for object detection. For face detection, they implemented two main algorithms which are CNN and Histogram of Oriented Gradients based SVM detector. The result overall cheating performance precision with 0.90 and recall 0.86 [19]. Y.atoum et al. proposed an automated multimedia system that depends on a microphone, wearcam, and a webcam. Their system modalities are user verification, gaze estimation, phone detection, text detection, voice detection, and active window detection. By collecting the visual and audio data of their subjects, they apply several techniques to detect their system modalities and apply an SVM classifier to continuously make a decision of cheating as the result showed overall 87% detection rate [3]. Li et al. have proposed a system that consists of three main phases, the first phase is the Automatic Cheating detector, the second is the peer cheating detector and the third is the final review committee. First the automatic monitoring of students that detects the gaze of the students using KNN, EEG brain signal reading to detect cheating using Binary Gaussian Naive Bayes. The peer cheating detector reviews the incidents flagged by the automated phase and then the final review committee decides which punishment should the student take. The automated approach only using multi-modalities achieved 81.1% and with the total phases, it achieved 92.7% [16]. Haotian et al focus more on helping human proctors to monitor students by detecting abnormal head movements like head rotation and face disappearance using R-CNN, also monitoring mouse events. Then they generated views for the proctor with the students that have possible cheating incidence to decide whether the student cheated or not. They provide better accuracy and time than the standard method of manual proctoring. [15]. Jalali and Noorbehbahani proposed an automated method to detect cheating using only webcam images. In their system, they capture a normal image for students and compare the threshold with images taken during the exam to detect any abnormal behavior. On 10 students the average accuracy value is 78%. [13]. Chuang et al used head pose relevant to computer screen and time delay to detect cheating during the online exam and reached an accuracy of 75.6% on detecting whether the student is behaving in a manner that may be considered as cheating or not [7]. Bawarith et al. proposed a solution that depends on the authentication of students using a fingerprint, and then using eye gaze tracking to track students if they seem away from the screen and calculates that time to detect any cheating behavior. They reached an accuracy of 97.78% on their classifier whether it is cheating or not [4]. A survey study by Zaidi et, al. [22] stated that different models and datasets were used in object detection problem. MS-COCO and RetinaNet with ResNet backbone represent an easy approach to train and also introduce promising performance in run time and accuracy compared to two-stage detectors.

paper	Techniques	Accuracy
[12]	visual focus of attention, eye gaze, headpose	96.04%
[19]	object detection, face tracking	0.90 P & 0.86 R
[3].	detection of gaze, phone, text, voice, and active window	87%
[16]	Gaze, EEG	92.7%
[13]	Image threshold change	78%
[7]	head pose and time delay	75.6%
[4]	eye gaze with time consideration	97.78%

TABLE I: Previous approaches and accuracy

III. Methodology

A. System Overview

Our System overview as explained in figure 1 contains three phases, the first is the input which is a webcam that captures a student's image every 10 seconds, the second represents the processing phase that receives the image and passes it to the different modalities. Each modality processes the image and then returns its output prediction. The third phase is a combination report of all modalities to alert the human proctor if there is a suspected cheating behavior.

B. Head pose

1) Overview

Head pose estimation is one of the challenging problems in computer vision, which is capable of indicating the head orientation from a digital image. The head pose is considered

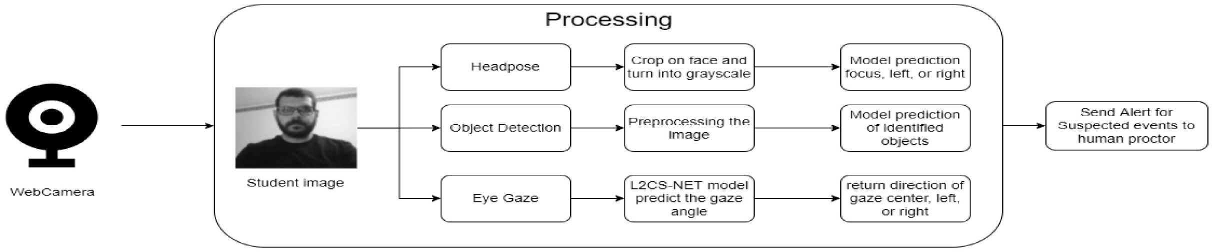


Fig. 1: System Overview with different modalities

an essential modality in detecting cheating in the online exam as the student should focus only on his screen, so if the student's head orientation on another direction then it is considered a probability of cheating.

2) Models

In our implementation of head pose, we used a transfer learning approach which is to use a pre-trained model that is trained on a similar type of problem and that is the imagnet dataset [9] then adjust its weights to use it in our problem. We tried five models which are VGG16, VGG19, RESNET50, Xception, and InceptionV3. We tried to compare those models to pick the best result to work with the head pose. All the above models show high accuracy in the imagenet dataset. VGG16 is a CNN model which is introduced in a paper by Simonyan and Zisserman [20], and it is called 16 referring to its 16 layers of DNN. The model is trained on Imagenet which contains 14 million images for 1,000 classes, and it achieves the top 5 test accuracy in this dataset with 92% accuracy. The input for this model is a 224x224 RGB fixed-size image. After the input layer, a stack of conv layers of them is followed by a max-pooling layer with stride 2, then three fully connected layers, in the first two layers contain 4,096 channels each. While the third contains 1,000 channels, which are one for each class, as it is a 1000-way ILSVRC classification. Then the final layer is the softmax layer. VGG19 is very similar to the VGG16 except that it contains 19 layers compared to the 16 layers of the VGG16 showing that VGG19 contains 3 conv layers more than the VGG16. This means that the VGG is a CNN model with its essential features which are the input layer, conv Layers, and the fully connected layers. The VGG19 carries the same architecture as the VGG16 but with the extra 3 conv layers. Resnet50 [11] is a CNN model that contains 50 layers. Like all the models used for the head pose in this paper, the Resnet50 is trained on the imagnet data, and the model is capable of classifying 1000 objects for this dataset. Resnet50 is a variant of the 48 Conv layers with one max pool and one average pool layer in the Resnet Model. The Resnet at first is classified as a model for image recognition tasks but it could also be used for non-computer vision tasks as it can achieve better accuracy. The architecture of the Resnet50 contains four stages. The input layer that takes a 224x224 size image, as in any Resnet architecture starts with a 7 x 7 conv and 3 x 3 max pool. Then, followed by stages which contain 3 blocks with each containing 3 layers, and then followed by the average pooling layer and the fully connected layer. InceptionV3 is a model that shows an accuracy of 78.1% on the imagenet dataset with its 48-layers. Like others, we used its pre-trained version on the imagnet dataset to be used in our task. InceptionV3 is introduced by Szegedy, et al. [21]. The model contains asymmetric and symmetric building blocks, which contain conv layers, max pool layers, avg pool layers, concat layers, dropout layers, and a fully connected layer. Its input size is 299x299 images and could be used in many tasks like general classification, Object detection, Image Classification, etc. Xception [6] is a 71-layer CNN model that like others is trained on the Imagnet dataset. The Xception model is known as the extreme version of the inception model as it uses the same architecture but with some variations. Hence, it achieves better accuracy than the inceptionV3 model with 94.5% in top-5 accuracy compared to 94.1% in inceptionV3. It takes a 299x299 size input image, while in inception, the original input is compressed by the 1 x 1 convolutions, and then different types of filters are used for each input space. But in Xception, this step is reversed to start with the filters first before the compression. The other difference between them is that the inception model after each of the first operations adds RelU non-linearity, but the Xception Model doesn't add non-linearity.

3) DataSet

In this modality, two datasets are combined to be used to train our model. The first dataset is the cropped version of pandora dataset [5] and the second dataset is by Gourier et al. [10]. The cropped pandora dataset contains 15,679 images of 10 males and 12 females from different head angles, and the second dataset contains 2790 images of 15 subjects also from various angles. In the data preparation phase, we combined both datasets and split them into three classes which are focus, left, and right. Then detect and crop images on the head only using cvlib so that there is no background in the image. After that, all images are converted to grayscale.

C. Object detection

1) Overview

Another modality implemented in our proposed solution is object detection. It is useful to get forbidden objects like cell phones or books, and could also detect people. So, if there is more than one person it is considered a cheating event.

Fig. 2: Head pose detection (focus,left, or right)

The usage of those forbidden objects no doubt indicate the student is cheating. There are many techniques and approaches in object detection, and in our approach we chose the retinanet model [17] to train and use it in our proposed solution.

2) model

Retinanet is a one-stage object detection model which runs directly over a dense sampling of locations. The one-stage models have high inference speed. Retinanet has proved that it works effectively with small-scale objects. The idea of retinanet is to improve single-stage object detection models by two things. The first is feature pyramid networks and focal loss. The Architecture of the retinanet consists of four parts. The first is bottom-up which is resnet as a backbone network that aims to take different scales and calculate its feature maps. The second is Top-down and lateral connections, which first from the higher pyramid levels it up-samples the spatially coarser feature maps, and then with the same spatial size, it merges the bottom-up layer and the top-down layer. The third is the classification sub-network which predicts the probability of the availability of objects for each object class and anchor box. Then the last is the regression sub-network for each ground-truth object. It regresses the offset of bounding boxes from the anchor boxes. Before prediction, a processing phase is triggered when reading the image, which is subtracting the imagnet dataset mean to normalize the input image, and then using the image after subtraction.

3) Dataset

One of the most popular datasets in object detection is Microsoft Common Object in Context (COCO) [18] The dataset contains nearly 328k images with 91 classes that are vailable with its annotations. We trained our model in only three classes which are cell phones, books, and person. The dataset contains images and annotation files for those images. Annotation basically contains metadata like width, height, and depth, and also the name and path of the image with objects on it with its location.

D. Eyegaze

1) Overview

Eye gaze is one of the most important modalities in cheating detection, as the student, without moving his head or using any object that could appear to the camera, has the ability to direct his/her eyes to an object that doesn't appear to the camera like another monitor, paper on the wall or many other cheating techniques, so eye gaze is an important modality to detect abnormal behavior from the student during the exam. The definition of eye gaze is the ability to predict the gaze angles of a person that indicates whether a person is staring or not. Therefore, it is an essential aspect of cheating detection to consider in our proposed solution.

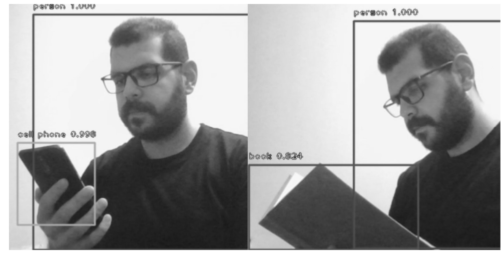


Fig. 3: cellphone and book detection

2) model

The Eye gaze implemented in this research is a model introduced by A.Abdelrahman. [2] L2CS-Net. This model captures an RGB image and estimates 3d gaze angles from it using multi-loss. The architecture of the model uses Resnet50 as the backbone. First, the input, which is a face image, passes through the Resnet50 model to be used in extracting the spatial gaze features from the image. At that time, each gaze passes to a fully connected layer to predict the angle of each gaze separately. Then, two loss functions are used to give every gaze an angle which is either the yaw or the pitch. Each loss function converts the network output logits into a probability distribution using a softmax layer, followed by a calculation of bin classification loss between target bin labels and output probabilities with cross-entropy loss. At that point, a probability distribution expectation calculation is applied to get gaze predictions. After that, a mean square error is calculated for the prediction to be added to the classification loss. The output of the model is represented in figure 4 A: eye at the center, B: looking right, and C: looking Left.

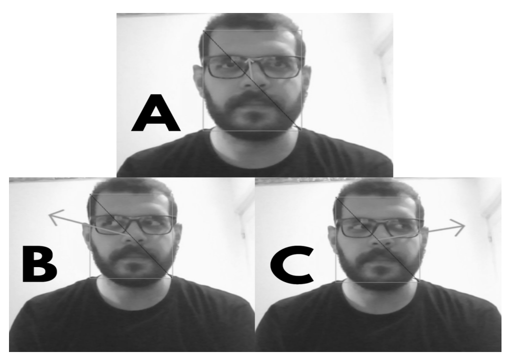


Fig. 4: Eye gaze estimation

E. Combining all modalities

After completing all the modalities explained above, it's time to combine them all in one system. The input of the system is a camera stream or a recorded video of a student solving his exam. The system takes an image frame every 10 seconds and this frame is passed to the three modalities functions. First is the head pose that crops the face and converts it to gray-scale, then pass the image to the model to predict the head direction and attaches it into a report if it includes Abnormal behavior. The second function is object detection that processes the image as explained above and includes the objects detected to the report if it includes forbidden objects. The third and last modality is the eye gaze model that predicts the pitch and yaw angles and then gets the direction of the gaze if it contains abnormal direction included in the report.

IV. Experiment and Result

A. General Setup

Three experiments have been constructed to evaluate our system. The experiments are done with the help of 29 students. For every experiment, a custom scenario and one controlled environment are set to evaluate our system. The average age of the students is 20.8 years. Figure 5 shows the experiment setup for our experiments. The student sits near the laptop camera by 70 cm. All experiments were done with normal room light conditions.

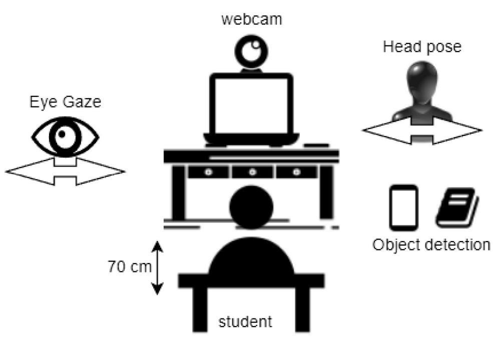


Fig. 5: Experiments setup

B. Experiment 1: Head pose Models

The objective of experiment 1 is to evaluate the created head pose models to detect the Higher model accuracy. The 29 students are asked to do certain poses each for 10 seconds to point their heads to the center, left, and right.

C. Experiment 1 Result

The accuracy of head pose models is calculated by extracting the frames from recorded videos of experiment 1 and then manually labeling those frames whether it is a focus, left, or right. Next, transmit all images to every model to predict and calculate the accuracy. In figure 6, the head pose models are compared together on extracted frames with a total of 8,696 labeled frames. The VGG16 showed a higher accuracy of 92.5%, therefore it is used in experiments 2 and 3.

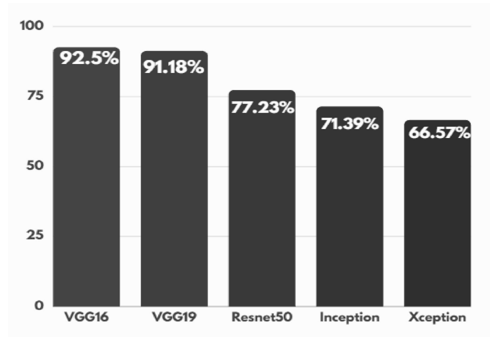


Fig. 6: Models Accuracy Comparison

D. Experiment 2: Basic modalities

The main objective of experiment 2 was to test if each modality can be detected independently and correctly. We have asked 15 subjects (two females and 13 males) to participate in this experiment. Six of them were wearing glasses (which might affect tracking eye gaze). This experiment was conducted with one session per subject. The subject sits with a laptop on the table in front of the student, where head and chest appeared on the webcam in its center. We have asked the 15 subjects to enter a sequence of five events, and each event is for an average of seven seconds. The sequence is ordered as follows (focus directly, face right, hold mobile, face left, and ask another person to help them in front of the camera).

E. Experiment 2 Results

The experiment result is evaluated according to the number of events detected according to the scenario with the accuracy of each modality alone shown in table I. The overall accuracy of total events detected was 96.66%.

Event	percentage
Cellphone	93.33%
head pose	96.67%
Another person	100%

TABLE II: Second experiment events detection accuracy

F. Experiment 3: Extended modalities with directions

The main objective of the experiment was to test the detection of certain cheating events using multi-modalities and increase the number of cheating events compared to experiment two. We have asked 14 subjects (five females and nine males) to participate in this experiment. Five of them were wearing glasses (which might affects tracking eye gaze). This experiment was done with three sessions per subject. The subject, also like experiment two, sits with a laptop on the table in front of the student, with head and chest appearing on the webcam in its center. To measure the accuracy of detecting multi-modal cheating events, we have asked the 14 subjects to enter a sequence of 10 events. The sequence is ordered as follows (focus directly, face right, face left, hold cell phone in front of the camera, look with your eyes looking only at the paper on your right, look with your eyes looking only at the paper on your left, hold a book in front of the camera, hold a cell phone with focusing right, hold a book with focusing left, asking another person to help them in front of the camera).

G. Experiment 3 Result

In experiment 3, there are more events of cheating in the scenarios and also more sessions so it provides the right indication for the accuracy of our system to detect the cheating events as proved in table II. The overall accuracy of cheating events detected is 95.69%.

Event	percentage
Cellphone front	94.44%
Cellphone right	94.87%
Book front	97.22%
Book left	86.20%
Another person	100%
Gaze right	100%
Gaze left	100%
Headpose	92.86%

TABLE III: Third experiment events detection accuracy

V. Discussions

The primary objective of our proposed solution was to support the proctor in discovering the student who cheat in the online examination easier, in less time, and with better accuracy. After completing the experiments, our proposed system could capture the cheating events accurately and could deliver the report to the proctor, but when combining multi-modalities in experiment three the system could detect more events and showed it is more effective to combine them all together to provide extra confidence to the proctor to discover the cheated student.

VI. Conclusion

There are many approaches to detect cheating in online exams. Some of them are using manual proctoring which has proven that it is time-consuming and high cost. The other approach is to make fully automated proctoring which is early to rely on the machine to decide whether the student has cheated or not. The approach that this paper proposed is to use the semi-automated approach to detect the cheating events and inform the human proctor to decide, which is better than the manual proctoring, costs less money and time and gets more accuracy when proctoring a large number of students. On top of that, from our experiments, it has been proved that using more than one modality is an add-on to cheating detection instead of implementing only one modality. In our experiments, the accuracy of detecting the events is 96.66% in experiment 2 and 95.69% in experiment 3 with multi-modalities.

References

[1] "Education: From disruption to recovery," Aug 2021. [Online]. Available: https://en.unesco.org/covid19/educationresponse

[2] A. A.Abdelrahman, "L2cs-net," 2022. [Online]. Available: https://github.com/Ahmednull/L2CS-Net

[3] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

[4] R. Bawarith, D. A. Basuhail, D. A. Fattouh, and P. D. S. Gamalel-Din, "E-exam cheating detection system," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080425

[5] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5494–5503.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 07 2017, pp. 1800–1807.

[7] C. Chuang, S. Craig, and J. Femiani, "Detecting probable cheating during online assessments based on time delay and head pose," *Higher Education Research & Development*, pp. 1–15, 03 2017.

[8] C. Y. Chuang, S. D. Craig, and J. Femiani, "Detecting probable cheating during online assessments based on time delay and head pose," *Higher Education Research & Development*, vol. 36, no. 6, pp. 1123–1137, 2017. [Online]. Available: https://doi.org/10.1080/07294360.2017.1303456

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[10] N. Gourier, "Head pose image database." [Online]. Available: http://crowley-coutaz.fr/Head\%20Pose\%20Image\%20Database.html

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[12] C. Indi, V. Pritham, V. Acharya, and K. Prakasha, "Detection of malpractice in e-exams by head pose and gaze estimation," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 08, 2021.

[13] K. Jalali and F. Noorbehbahani, "An automatic method for cheating detection in online exams by processing the student's webcam images," 03 2017.

[14] D. L. King and C. J. Case, "E-cheating: Incidence and trends among college students," *Issues In Information Systems*, Jan 2014.

[15] H. Li, M. Xu, Y. Wang, H. Wei, and H. Qu, "A visual analytics approach to facilitate the proctoring of online exams," *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, May 2021. [Online]. Available: http://dx.doi.org/10.1145/3411764.3445294

[16] X. Li, K.-m. Chang, Y. Yuan, and A. Hauptmann, "Massive open online proctor: Protecting the credibility of moocs certificates," ser. CSCW '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1129–1137. [Online]. Available: https://doi.org/10.1145/2675133.2675245

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017. [Online]. Available: https://arxiv.org/abs/1708.02002

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. [Online]. Available: https://arxiv.org/abs/1405.0312

[19] A. C. Özgen, M. U. Öztürk, and U. Bayraktar, "Cheating detection pipeline for online interviews and exams," *ArXiv*, vol. abs/2106.14483, 2021.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[22] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103514, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200422001312

4.5.4 Multi-modalities Analysis In Profiled Learning

Multi-modalities Analysis In Profiled Learning [13]

16/06/2022

ACCEPTANCE LETTER

Dear ***Mario Shoukry, Ayman Ezzat,***

Thank you for your submission to the ICECET 2022 conference. We are pleased to inform you that your paper entitled **"ID-997 Multi-modalities Analysis In Profiled Learning "** has been accepted as a full paper for **oral presentation** by the conference committee of *International Conference on Electrical, Computer, and Energy Technologies (ICECET)*. The event will take place in Prague, Czech Republic on 20-22 July 2022 **online** and **physically.**

We strictly follow "no podium, no paper" policy and only the papers that are presented at the conference will be submitted to IEEE Explore for publication. **At least one author** of an accepted paper must register (as a full participant) and participate in ICECET 2022 online or physically for the paper to be included in the proceedings. If you have not yet registered online (using the credit card or bank transfer options), at least one author of each paper should register to the conference via the online registration page at https://www.ecres.net/icecet. If you have already registered, please do not make another registration. Kindly note that your registration becomes valid only after your payment.

According to the conference regulations, only those papers which have been duly registered and presented on the conference day are considered for submission to IEEE Explore. The conference program will be communicated in due course.

We look forward to seeing you for a fruitful research and innovation event and for a great time in the wonderful environment of Prague

Yours sincerely,

Dr. Simon Winberg
Chair

Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET 2022)
20-22 June 2022, Prague-Czech Republic

Multi-modalities Analysis In Profiled Learning

Mario Nady
Faculty of Computer Science
October University for Modern Sciences and Arts
Faculty of Computer Science
University of Greenwich
Giza, Egypt
mario.nady@msa.edu.eg

Ayman Atia
HCI-LAB,
Faculty of Computers and Artificial Intelligence
Helwan University
Faculty of Computer Science
October University for Modern Science and Arts
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—The advances in information technologies offer a promising approach to leveraging effective and engaging learning experiences. The diversity of sensor-based technologies, such as facial expression analysis and gaze tracking, has introduced the opportunity to capture students' interactions with learning activities or assessments. In such context, captured data hold meaningful promise for gathering a deeper understanding of students' learning experience and informing an adaptive frame to support individualized learning needs. This paper introduces an analytic approach that incorporates students' eye tracking, facial expression, and mouse movement data to predict students' performance while resolving an English exam. The proposed system examined the degree to which different modalities captured from 53 students (aged 18 to 22) in an authentic learning environment were predictive of the students' exam scores. The analysis of the collected dataset shows that adding student features has effectively predicted his assessment score. The adoption of different regression models revealed that multi-modal data could accurately predict students' exam scores and hold significant potential for guiding a real-time adaptive environment.**

***Index Terms*—eye gaze, facial expression, mouse movement, learning performance.**

I. Introduction

Learning is one of the most important things that any country works on and tries every day to enhance the process of education to raise the educational state of the people. Through some experiments that have been done, it has been proved that there are different types of learning styles and each learner has a different style from the other [5]. This adds to the growing body of data that traditional teaching techniques should be reformulated for online classrooms since the approaches that correlate to learners' abilities are more difficult to identify [7]. The scientists discovered a new approach that helps the learner to meet up with the requirements that make him capable of learning with less time [18], with more benefits, and high quality, this approach is known as Adaptive Learning. Adaptive learning refers to the delivery of training or education that makes use of technology and data to provide students with an individually tailored learning program that intelligently adapts to their learning needs. There are partnerships between E-learning software companies and education publishers, and a large investment in adaptive learning in recent years and the concept of adaptable learning has grown in popularity. Adaptive learning became a field for research since digital transformation has begun. This paper discusses how some different modalities and behaviors of the learner lead to predicting his score in the assessment.

In recent times, the behavior analysis of learners is gaining considerable interest in the context of educational technology, and its rise has fueled a wave of educational information. Online learning became the gate to analyzing the behaviors of the students, and the performance of each student is being analyzed to predict his learning patterns. Because of their personality features, learning experiences, understanding of online learning, and degree of engagement with other students, students have diverse learning behaviors. There are many questionnaires that were used to figure out the learning styles of the learner but sometimes the learner can not answer the questionnaire fairly so the results may not be accurate enough [14]. This study's aim is to predict a student's academic achievement. Students' data is utilized to create a model that predicts a student's academic achievement based on their background information. This study's input data should be a student's behaviors dataset. Algorithms have been trained to generate the model that predicts the score. In this study, this study looks at the factors that affect students' performance using regression models.

II. Related Work

Several studies have examined students' interaction while interacting with learning activities or resolving an assessment to measure their performance using different data sources such as behavior traces via mouse movement, facial expressions of emotions, and eye-gaze. Such intelligent observations enable researchers and practitioners to develop and implement remedial measures in order to assist students' learning needs, adapt activities, and improve learning outcomes [8], [23].

Emerson et el. [3] developed a multimodal learning analytics strategy that uses data from students' gaming, eye tracking, and facial expressions to anticipate peak performance and engagement in a game-based teaching environment by using 65 students. Models that are used based on multimodal data

978-1-6654-7087-2/22/$31.00 © 2022 IEEE

either perform similarly well or outperform models based on unimodal data when predicting students' posttest performance and interest, according to the findings. They show how mixing modalities have a synergistic effect on predicting both students' interest and posttest performance. According to the findings, multimodal learning analytics can effectively predict students' posttest performance and interest during game-based learning, and they have a lot of promise for managing real-time adaptive scaffolding.

A. Eye gazing

In the field of education, research studies used eye-gaze data to examine and model students' knowledge acquisition. Examining patterns of eye movements was shown to be a good indicator of how students learned in interactive learning settings [17]. Eye gazing is a good indicator of knowing whether the student is paying attention or not, as the systems correlate the eye gazing and the engagement of the student whether in the learning process or the assessment. In this paper [1], they used a large dataset called MPIIGaze and it consists of 213,659 photos from 15 participants through their daily routine in an experiment that lasted for 45 days. The study implemented a CNN model achieving an accuracy of 70% at the x-axis and 60% at the y-axis. I. Joe Louis Paul and S. Sasirekha tried to implement the same idea in a virtual classroom of students [11]. They used EyeTribe Tracker Pro which tracks the user's eye movement by a camera to measure students' concentration on the screen by analyzing visual attention while attending video lectures. They used heat map attention analysis to figure out the fixation time of students on the screen. They presented different types of videos and provided an assessment at the end of the lecture to determine whether or not the students were paying attention.

B. Facial expression recognition

The development of online learning platforms gives researchers a chance to get closer to the behaviors of the students [4], [11]. Facial expression recognition is one of the behaviors that help us to know the patterns of learning of some students. Yang, Lei interpreted how the machine is able to recognize the emotions of the students and how it helps in the evaluation of learning effects [20]. They proposed their idea by using CNN and under two features which are basic features and learner features. They build their own dataset by taking 10 students for each emotion and then taking 27 frames of pictures from each student. They relied on 7 different types of emotions like interest, sleep, pleasure, etc. In their paper, they put estimation for quantitative features of expression for learners' three-dimensional emotions. Their work simulates the expressions and does not extract them from real-life applications for students.

Darapaneni et el. [2] used VGG16 besides deep CNN and transfer learning methodology to improve the performance of classification. They applied the model to a dataset that consists of 213 images, with different expressions for all of them. Their model got an accuracy of around 88.26%. They did data pre-processing which allowed them to have good results before implementing the algorithms. Hesham et el. [6], used a Support Vector Machine (SVM) on a real dataset to determine the education level of the learners. Their data set consists of 6 different levels of difficulties. They used a feature extraction module that was implemented by OpenCV library. The facial expressions were 5 different faces which are joy, surprise, natural, sadness, and anger. The SVM got the highest accuracy with 87%, DecisionTreeClassifer got 73%, and at last k-nearest neighbors (KNN) 53%. In this paper, the regression approach is used and proved to perform better than SVR in Mean Squared Error.

C. Mouse movement

Youssef et al., [21] proposed a system that uses machine learning techniques to predict students' cognitive types based on their behavior traits while they interact with an online lab experiment. In an online lab experiment employing MCQ, True or False, and fill-in-text questions, the recommended system was constructed to assess and analyze users' behavior while interacting with various components on the screen using a mouse or touchpad. The system keeps track of all mouse (x,y) trace events, including the time spent on each screen component and the number of hovers. After that, the data is gathered and recorded in a CSV file. The trials show that the KNN and SVM classifiers are accurate enough to predict the majority of cognitive types. The KNN, linear regression, neural network, KNN coupled, and SVM studies indicate an improvement in overall total RMS error in relation to the KNN.

Academic cheating is most common among college students due to the rapid increase in online education, especially in developing countries. H.Sokou et al., 2020 in a blended course, tested the developed mouse tracking program and the developed Moodle plugin in 20 percent examinations of the mid-term for detecting the potential cheaters [13]. The used machine learning algorithm is SVM to establish the connection between the target and independent variables for both regression and classification applications (the four different mouse event logging). To model, estimate, and classify the collected data, they used SVM with Linear Kernel. They discovered that observing learners' actual behaviors via mouse activities allow for early prediction of unethical behavior (cheating). The recommended model predicted 94% (sensitivity metric) of students executing unethical acts with an accuracy score of 89% percent at some time throughout the online mid-term assessment. Similarly, experimental results demonstrate a statistically significant difference in time spent in other labs between male and female students, indicating that male students are more likely than female students to participate in other activities.

Y. Li et al., 2021 proposed an evaluation approach of Focused Attention Level using a user's mouse operation records [9]. The Schulte table was used in the study, which is a frequent exercise used to improve children's and students' focused attention skills. Using functions extracted from mouse operation records to complete a Schulte Table task and a machine learning model to anticipate the subject's intended attention level. They assess the efficacy of proposed features as well as the overall performance of different category models. The accuracy of this approach reached as high as 80.9%, according to experimental results.

III. METHODOLOGY

In this section, the authors discuss the data set that was utilized in the training of the various machine learning models, as well as the data preparation techniques used in cleaning, merging, and extracting features from the data.

A. SYSTEM OVERVIEW

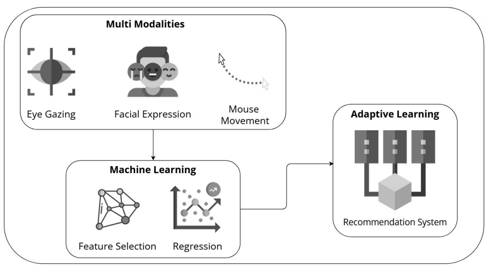


Fig. 1: System Overview

Figure 1 shows the proposed system that is applied to students' multi modalities. The system begins with the data gathering process, which requires the student to enroll in an English exam and monitor his interactions and engagements during the exam. Three modules are being developed to monitor his behaviors, such as eye gaze, facial expression, and mouse movement. The data is then preprocessed in order to be ready for analysis and training in various machine learning models. The authors attempt to predict the assessment score based on the features of students in this study.

B. Data Collection

Fig. 2: Students in the experiment.

The gathered data and some annotations process for the dataset that has been collected of different students who take English exams as shown in Fig2. Besides, their behaviors like eye-gazing, facial expressions, and mouse movements according to their score. Fifty-three students took place in our dataset and there is a video recorded and frame shots while solving the exam that has been made to cover all levels of difficulty. The exam has been prepared for them in a windows application made by Python and the video was recorded from a front webcam in the same condition as the light and calibration of the camera. Before the exam, the authors collected some information from them that helped in the prediction as indicated below: Gender, learning level (i.e., 1,2,3), GPA (i.e., less than 2, 2.33 to 3, and so on), Left-Handed or Right-Handed. Academic: subject(i.e English). Behavior: the time that each student took in each question.

The exam itself consists of five different questions in English subject and they were in multiple-choice questions (MCQs) style and they are made by a tutor in the subject to determine the level of their knowledge in English.

Table I shows the categories of each attribute and their description of them. The proposed system collected 53 videos from 53 students and each video works by 20 frames/second (FPS). Besides the videos, the module of facial expressions takes photoshoots for the student to analyze students' facial expressions.

It consisted of 53 records gathered throughout the evaluation. They received a maximum score of 5 points as a consequence of their efforts.

C. Eye gazing

Fig. 3: Eye gazing module

Eye-tracking is used to extract features and determine gaze direction [12]. Eye gazing is composed of 4 main columns such as blink, center, right, and left, each column indicates how many times the learner did move his eye as shown in fig 3. The mouse movement consists of X and Y coordinates of the mouse during the assessment. The proposed system used Dlib eye gazing model to detect their eye movements. We implemented Eye gazing by a webcam [10]. The output videos are captured at 20 frames per second using OpenCV. It gives good results in good conditions of light even using a low-cost webcam [8]. The eye gazing module has been tested before the experiment on each student to make calibration for the camera.

D. Facial expression recognition

The facial expression module has been implemented by using the deep-face library. This module is analyzing the face attributes like emotion, gender, age, and race. The experiments of this library show 97.53% accuracy for facial recognition and facial attribute analysis [16]. It provides 7 different emotions

TABLE I: DESCRIPTION OF THE COLLECTED DATA.

Category	Description	Attribute Name	Possible Values
Student Profile	Student's gender	Gender	Male or Female
	Student's age	Age	18 - 22
	The student's grade group in the university.	Level	1 -5
	Student Height.	Height	Height in cm
	Wearing glasses or not.	Glasses	Yes/No
	The used hand.	RH or LH	RH or LH
Academic Performance	Score of the exam	Score	0 - 5
	Detailed Score for each Question	Result_Q(*)	0 - 1
Behavioral Features	Looking Left (Eye Gazing)	Q(*)_Left	Numbers of Frames
	Looking Right (Eye Gazing)	Q(*)_Right	
	Looking Center (Eye Gazing)	Q(*)_Center	
	Blinking (Eye Gazing)	Q(*)_Blink	
	Timer Per Question	Timer	Time in seconds
	Dominant Facial Expression	D1_Q(*)	Natural - Sad - Surprise - Happy - Angry
	Mouse Movements (X,Y) per Question.	M_Q(*)	Coordinates [x,y] of the mouse.

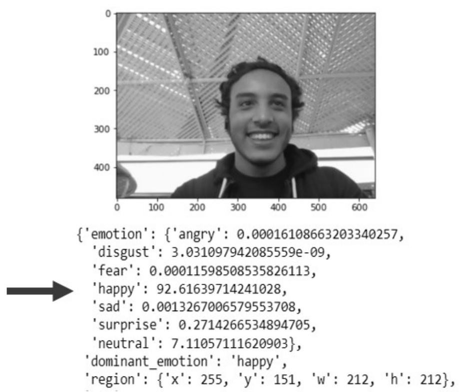


Fig. 4: Facial expressions recognition module

for the face such as anger, disgust, fear, happiness, sadness, surprise, and nature.

This library uses the VGG-Face model as a default configuration, it also has many different models like Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace. Sanun et el. [15] got 98.61% accuracy on LFW database and 99.17% for real-time videos. The proposed system took five photo shoots of each question to determine the dominant emotion. Then, the dominant emotion has been extracted and its accuracy from the analysis and saved into a CSV file as shown in fig 4.

E. Mouse movement

The proposed system extracted the unseen behavior of the student by tracking his mouse movements during the exam. Python library called Pynput has been used that allows monitoring the coordinates of input devices like the mouse. The cognitive state has been predicted student's behavior from his mouse movements [22]. At this time, they collected all the coordinates from the mouse during each separate question. We considered his movements as part of our analysis.

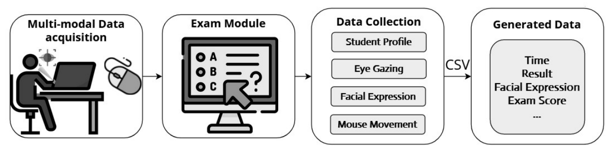


Fig. 5: Experiment Setup

F. Experiment Setup

The experiment setup starts with multi-modal data acquisition as shown in Fig5. The experiment took place in the university among different levels of university students from different faculties. They have different characteristics like age and GPA. The authors introduced the idea to them and took their permission for recording their experiment for academic purposes.

Exam Module

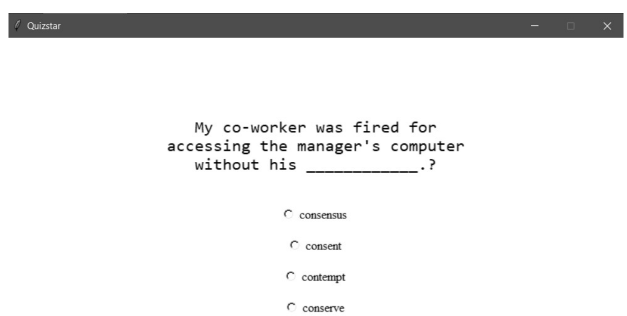


Fig. 6: Exam Module Interface

The exam module has two main sectors, first, they filled out a form for collecting demographic questionnaires. Second, they started to solve 5 questions in the form of Multiple Choice Questions (MCQ) as shown in fig 6. They were not obligated to finish the exam at a specific time but their average time was 5 minutes. In addition, the environment was prepared to let them focus on the exam module. Following the gathering of the dataset, pre-processing procedures have been used such as data cleansing and transformation such as handling missing values, removing duplicates, fixing spelling mistakes, and aggregation of eye gazing attributes per each question.

G. Feature Selection

The collected dataset came with extra features whether from student profile, Academic performance, or Behavioral features as mentioned in TABLE I. In this case, the most important features need to be identified as it affects the prediction. The most important features have been selected from highest to lowest. as shown in Figure 7.

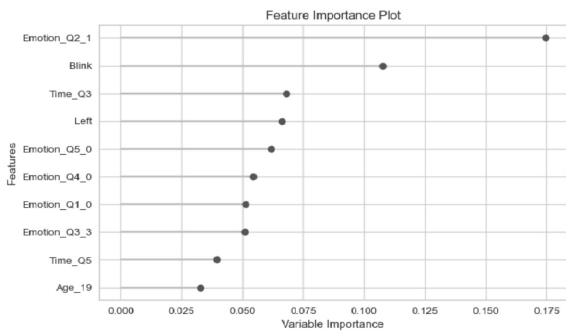


Fig. 7: Feature Importance Plot

The most associated features with the characteristics of the student were deduced using various techniques, and the first four features were chosen from this plot to be utilized in the prediction of the student's result. Pycaret library has been used to make feature importance which uses several supervised feature selection techniques.

The collected dataset has been divided into 70:30 training and testing subsets, with 70% used to train and validate the prediction model. Various regression models have been used to deduce the correlation between the behaviors and the result of the student. The linear and non-linear regression models are used to predict these types of problems [19].

IV. Experiment

After cleaning the data, the most relevant features have been extracted to be utilized in the training process. There were several regression models that have been used to deduce different results and visualize those results to conclude the high result model among the numerous models that have been used to predict the result as a target attribute based on the student behaviors.

The aim of this experiment is to predict the result of the student based on the behaviors that have been collected in the previous section. The collected dataset is analyzed to see how significantly the modalities are linked to students' behavior and how this information is used to help students set up a profile during the learning process.

Data Analysis: The authors constructed some statistical analysis for the data set in order to help in feature extraction. The data set consists of 53 students, 83% right-handed and 17% left-handed. Approximately, 77.4% were male students and 22.6% were female students. The mean of their age was 20 years old, and the majority were from levels 2 and 3 of the university.

TABLE II: Experiment Results

Abbreviation	Model	MAE	MSE	RMSE
ada	AdaBoost Regressor	0.7330	0.9544	0.8330
omp	Orthogonal Matching Pursuit	0.7536	0.9936	0.8834
et	Extra Trees Regressor	0.7740	1.1532	0.9319
gbr	Gradient Boosting Regressor	0.7916	0.8234	0.8592
rf	Random Forest Regressor	0.8655	1.0183	0.9469

Different models have been trained on the data, as shown in Table II, where the AdaBoost Regressor got the least (Mean Absolute Error) MAE and (Mean Square Error) MSE among the other models. We targeted the result attribute among the other attributes to predict the result of the students. Moreover, the other models give closer results at Mean absolute error with different values. The accuracy of the algorithm is tested using the 10-fold cross-validation approach in this study. The algorithm predicts the result of the student depending on the feature importance as shown in Fig 7.

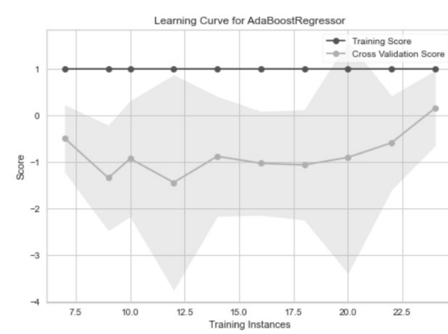


Fig. 8: Learning Curve

The model's learning curve depicts how the rate in the training score has been steadily growing due to the small dataset, as well as the faster variable increasing rate in the cross-validation score, which depicts the various training instances. This is shown in Figure 8.

The findings revealed that using variables such as students' gaze movement, Emotion of Question 2, Time to Finish Question 3, and Emotions of other questions does accurately predict students' performance. These findings assure the hypothesis that a fusion of student behaviors, such as eye gazing and facial expressions, might predict students' quality of learning.

Conclusion and future work

This study investigated the possibility of building a predictive machine learning model by monitoring students' behaviors during an online exam. The observed results contribute to the field of educational technology research by demonstrating how a machine learning pipeline based on students' multi-modal activities might aid in the development of an adaptable learning environment. It reported findings from a 53-student in-person experiment on student engagement and performance (aged 18-22 years). Additionally, evaluating eye-gaze, facial expressions, and mouse movements, gave insights regarding students' performance and interaction behaviors. There exists a correlation between students' behaviors and their performance in the education process. In order to enhance the results, other modalities will be investigated. One of the modalities that may be applied is the pose estimation of the student during the assessment or learning process.

REFERENCES

[1] Panteleimon-Evangelos Aivaliotis, Foteini Grivokostopoulou, Isidoros Perikos, Ioannis Daramouskas, and Ioannis Hatziligeroudis. Eye gaze analysis of students in educational systems. In *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*, pages 1–6, 2020.

[2] Narayana Darapaneni, Rahul Choubey, Pratik Salvi, Ankur Pathak, Sajal Suryavanshi, and Anwesh Reddy Paduri. Facial expression recognition and recommendations using deep neural network with transfer learning. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0668–0673, 2020.

[3] Andrew Emerson, Elizabeth Cloude, Roger Azevedo, and James Lester. Multimodal learning analytics for game-based learning. *British Journal of Educational Technology*, 51, 07 2020.

[4] Sai Manvitha Enadula, Akshith Sriram Enadula, and Rama Devi Burri. Recognition of student emotions in an online education system. In *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–4, 2021.

[5] Nico Herbig, Tim Düwel, Mossad Helali, Lea Eckhart, Patrick Schuck, Subhabrata Choudhury, and Antonio Krüger. Investigating Multi-Modal Measures for Cognitive Load Detection in E-Learning. *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, 2020.

[6] HabebaElRahman Hesham, Mourad Nabawy, Ola Safwat, Youssef Khalifa, Haytham Metawie, and Ammar Mohammed. Detecting education level using facial expressions in e-learning systems. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6, 2020.

[7] Ermira Idrizi, Sonja Filiposka, and Vladimir Trajkovik. The discourse on learning styles in online education. In *2019 27th Telecommunications Forum (TELFOR)*, pages 1–4, 2019.

[8] Ko Fong Lee, Yen Lin Chen, Chao Wei Yu, and Cheng Han Wu. The eye tracking and gaze estimation system by low cost wearable devices. In *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, pages 1–2, 2020.

[9] Yongming Li, Junfeng Niu, and Zhongmin Cai. Evaluating focused attention level using mouse operation data. In *2020 Chinese Automation Congress (CAC)*, pages 3679–3684, 2020.

[10] Lorenzo Lucignano, Sébastien Cuendet, Beat Schwendimann, Mina Shirvani Boroujeni, Jessica Dehler, and Pierre Dillenbourg. My hands or my mouse: Comparing a tangible and graphical user interface using eye-tracking data. 01 2014.

[11] I. Joe Louis Paul, S. Sasirekha, S. Uma Maheswari, K. A. M. Ajith, S. M. Arjun, and S. Athesh Kumar. Eye gaze tracking-based adaptive e-learning for enhancing teaching and learning in virtual classrooms. In *Information and Communication Technology for Competitive Strategies*, pages 165–176. Springer Singapore, August 2018.

[12] SV Sheela and K R Radhika. Feature based methods for eye gaze tracking. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1101–1107, 2020.

[13] Hamidullah Sokout, Fajar Purnama, Ahmad Nawid Mustafazada, and Tsuyoshi Usagawa. Identifying potential cheaters by tracking their behaviors through mouse activities. In *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 143–149, 2020.

[14] Boddeti Sravani and Myneni Madhu Bala. Prediction of student performance using linear regression. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–5, 2020.

[15] Sanun Srisuk and Surachai Ongkittikul. Robust face recognition based on weighted deepface. In *2017 International Electrical Engineering Congress (iEECON)*, pages 1–4, 2017.

[16] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[17] Michelle Taub, Nicholas V. Mudrick, Roger Azevedo, Garrett C. Millar, Jonathan Rowe, and James Lester. Using multi-channel data with multi-level modeling to assess in-game performance during gameplay with crystal island. *Computers in Human Behavior*, 76:641–655, 2017.

[18] Shuai Wang, Mingyu Feng, Marie Bienkowski, Claire Christensen, and Wei Cui. Learning from an Adaptive Learning System: Student Profiling among Middle School Students. *Proceedings of the 11th International Conference on Computer Supported Education*, 2019.

[19] Zhuojia Xu, Hua Yuan, and Qishan Liu. Student performance prediction based on blended learning. *IEEE Transactions on Education*, 64(1):66–73, 2021.

[20] Lei Yang and Chao Qi. Study on learner's facial expression recognition. In *2020 International Conference on Networking and Network Applications (NaNA)*, pages 174–179, 2020.

[21] Ahmed Mohamed Fahmy Yousef, Ayman Atia, Amira Youssef, Noha A. Saad Eldien, Alaa Hamdy, Ahmed M. Abd El-Haleem, and Mahmoud M. Elmesalawy. Automatic identification of student's cognitive style from online laboratory experimentation using machine learning techniques. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0143–0149, 2021.

[22] Jia Yue, Feng Tian, Kuo-Min Chao, Nazaraf Shah, Longzhuang Li, Yan Chen, and Qinghua Zheng. Recognizing multidimensional engagement of e-learners based on multi-channel data in e-learning environment. *IEEE Access*, 7:149554–149567, 2019.

[23] Wei Zhang, Xujun Huang, Shengming Wang, Jiangbo Shu, Hai Liu, and Hao Chen. Student performance prediction via online learning behavior analytics. In *2017 International Symposium on Educational Technology (ISET)*, pages 153–157, 2017.

4.5.5 Recognition of Butterfly strokes using different Machine Learning Models

Recognition of Butterfly strokes using different Machine Learning Models [19]

ICECET 2022
International Conference on
Electrical, Computer and Energy Technologies
20-22 July 2022, Prague

16/06/2022

ACCEPTANCE LETTER

Dear ***Salma Tamer, Ayman Atia***

Thank you for your submission to the ICECET 2022 conference. We are pleased to inform you that your paper entitled **"ID-1039 Recognition of Butterfly strokes using different Machine Learning Models "** has been accepted as a full paper for **oral presentation** by the conference committee of *International Conference on Electrical, Computer, and Energy Technologies (ICECET)*. The event will take place in Prague, Czech Republic on 20-22 July 2022 **online** and **physically.**

We strictly follow "no podium, no paper" policy and only the papers that are presented at the conference will be submitted to IEEE Explore for publication. **At least one author** of an accepted paper must register (as a full participant) and participate in ICECET 2022 online or physically for the paper to be included in the proceedings. If you have not yet registered online (using the credit card or bank transfer options), at least one author of each paper should register to the conference via the online registration page at https://www.ecres.net/icecet. If you have already registered, please do not make another registration. Kindly note that your registration becomes valid only after your payment.

According to the conference regulations, only those papers which have been duly registered and presented on the conference day are considered for submission to IEEE Explore. The conference program will be communicated in due course.

We look forward to seeing you for a fruitful research and innovation event and for a great time in the wonderful environment of Prague

Yours sincerely,

Dr. Simon Winberg
Chair

Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET 2022)
20-22 June 2022, Prague-Czech Republic

Recognition of Butterfly strokes using different Machine Learning Models

1st Salma Tamer
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA).
Cairo, Egypt
salma.tamer@msa.edu.eg

2nd Ayman Atia
HCI-LAB, Faculty of Computers and Artificial Intelligence,
Helwan University.
Faculty of Computer Science,
October University for Modern Sciences and Arts (MSA).
Giza, Egypt
aezzat@msa.edu.eg

***Abstract*—Swimming is a lifelong beneficial activity. It is an excellent training since it requires you to move your entire body against the water's resistance; however, by the time, these movements may not be in a right way. In addition, the wrong movements may lead to many pains such as shoulder pain, elbow pain and lower back pain especially in difficult strokes. The coach is the one who instructs the swimmers and tell them which is incorrect, and which is correct. However, he can't recognize all the incorrect movements, so this needs an instructor who can see all the stroke's mistakes. Hence our proposed system, which uses machine learning techniques, utilizes four different models which are Long short-term memory (LSTM), k- nearest neighbor (Knn), for time series 1-$ recognizer and Dynamic time wrapping (DTW) to detect the incorrect butterfly stroke. The system uses an accelerometer and gyroscope sensors to detect and evaluate correct and Incorrect swimming patterns in butterfly stoke. In addition to attaching a mobile application to the swimmer's wrist which gathers all data which allows the coach and the swimmer to know the incorrect strokes such as lifting the head too high, sweeping out after hand entry, and bending the arm. When an incorrect movement is recognized. DTW achieved the best accuracy among all classifiers which are 80.5%. The system helps in aiding the coaches to know all the swimmer's performance and all his performance, and also aid the intermediate swimmers to know more about his performance to enhance it.**

***Index Terms*—Machine learning, Butterfly stroke, Correct Technique, Incorrect technique, Sensors and Wearable device.**

I. Introduction

Swimming is an individual or team sport that involves moving through water. It can be done in pools or in open water such as lakes, rivers, or sea. Hence swimming sport helps in body endurance and losing weight and prevents getting any diseases or illnesses. Swimming is a motivating sport for people of all ages no matter what age. Each stroke in this sport requires a set of techniques which if it is made in an incorrect way it will cause injuries in some of the body parts such as shoulder, lower back, elbow pain, neck pain..etc. Although, those are the main injuries which a lot of the swimmers suffer from.

Competitive swimmers complete 2500 or more shoulder revolutions per day [1]. Competitive swimmers' most prevalent musculoskeletal concern is shoulder pain or injury [2] [3]. During practice, 72 percent of high school competitive swimmers used pain medication to control their shoulder pain [4]. During the training phase, the swimming coach is responsible for motivating a progressive change in a swimmer's performance.

The butterfly stroke is the most challenging stroke that demands both good technique and powerful muscles. Butterfly stroke works on arms, chest and upper back muscles to raise both of arms up out of the water and over the head. Swimmers are facing many challenges in butterfly's technique such as Lifting head too high, sweeping out after hand entry and bending arms or elbow. All of these challenges lead to many injuries and let swimmers never enhance their time in their training. Many coaches are suffering from not seeing the swimmer's techniques under water. As shown in figure 1. Those are two pictures that show the correct and incorrect technique for butterfly stroke. The left picture is the correct one because his arm entry is near to his head. The right picture is the incorrect technique because of the wrong arms entry to the swimming pool, for the reason that his arms are far away from his head. The swimming pool distance is so important which are 25m and 12.5m while making our second experiment.

We demonstrated a system that concentrates on recognizing and analyzing incorrect butterfly strokes. Lifting the head too high, sweeping out after hand entry, and bending the arms or elbow are all examples of classified incorrect strokes. First, "lifting head too high", it's right technique is that the head should be on the surface of the water not upper than that just to breath and get it back down. Second, "Sweeping out after hand entry", it's right technique is that the hands should enter the water wider than the shoulders and then pull straight down, and the hands will move little bit closer. Third, "Bending arms or elbow" it's right technique is that The arm should be stretched out in front of the body above the water's surface and let it into the water by using the top of their fingers. Despite all these wrong and right techniques, the coach recognizes only

Fig. 1. Butterfly stroke right and wrong strokes

the technique that is above the water. Therefore the system will help the coach recognize the techniques which are in the water. The project's contribution is to differentiate between the incorrect and correct technique through accelerometer and gyroscope sensor by using DTW, LSTM, KNN and 1$ recognizer algorithms for aiding coaches and preventing swimmers from injuries for improving their performance. We have conducted two experiments to evaluate our proposed system. The first experiment's objective focus on collecting the data from outside the water to determine which classifier has the best accuracy. The second experiment's purpose behind collecting the data inside the water is to evaluate the chosen classifier that will be accurate in the swimming pool and to be sure of that.

II. Related Work

A smart watch that collected the data from swimming stroke by using Bi-LSTM and it classified the data and made f1 score 91%. [5]

Estimated lap count, stroke, time in lap, and overall swimming time are all calculated using a wrist-worn sensors. Accelerometers were used to determine linear kinematics and slope in proportion to gravity: Magnetometers and gyroscopes are devices that measure the rotations and body position in respect to the Earth's magnetic field. Accelerometer are used as a 3D to classify stroke and lap counting. [6]

A piece of hardware that collects a set of signals based on the attitude and performance of swimmers of various skill levels. For data analysis, they used the heading reference system (AHRS) and machine learning. In the recognition of butterfly, breaststroke, and freestyle technique, the result was revealed with accuracy of 100 % and 86 % in swimming lap segmentation. They employed "swimbit" technology, which contains an IMU with a three-axis gyroscope, accelerometer, and magnetometer that is worn on the back of the swimmer and allows the data to be calculated. [7]

A head-worn sensor that collects kinematic data for analysis of swimming data. The data included two separate different lane-ending turn techniques, as well as the four most common swimming strokes: backstroke, breaststroke, butterfly, and backstroke. For style classification, a 95% classification rate was achieved. To obtain inertial sensor data, they used the SHIMMER sensor platform. A three-axis accelerometer was added to the main board, together with a three-axis gyroscope, resulting in raw data in six dimensions. [8]

Using the Bluetooth Low Energy(BLE) which is wireless communication protocol to modify an inertial measurement unit (IMU) and position it on the back of the swimmer's head to monitor the pitch and roll of the head during front crawl swimming.The data collected by an IMU can be seen in real time on an Android mobile device which is saved in the device's built-in memory, and then uploaded to Google Drive for further study. For breathing pattern analysis, an algorithm was created to count the number of breaths taken to the left and right. [9]

Swim styles may be distinguished and strokes can be counted in real time. A data processing phase and a stroke analysis phase make up the suggested approach. The data processing phase begins with the collection of linear acceleration values, the removal of floating data, and the identification of feature locations in the perceived sensory input, according to the information gathered during the data processing step. The stroke analysis step determines the swimmer's stroke style before employing the correlation coefficient idea to count strokes by using a waterproof android platform. [10]

This paper aims to know the swimmer's performance and help the coach by giving him a real time feedback. They used worn inertial sensors to collect the data with it. They mainly focus on the evaluation of the accuracy of different algorithms to analyse different phases of swimming, specifically starts, turns and freestyle stroke. [11]

ISwimCoach wearable device assists the coach to analyze and detect the swimmer's pattern by collecting all the patterns from the swimmer's stream. Although each swimmer's pattern (Stroke) is classified into correct and incorrect technique in freestyle stroke. In addition to that, all the strokes will be collected and classified in real time. Moreover, the proposed system capable to classify the stroke into four types which are correct strokes, wrong recovery, wrong hand entry and wrong high elbow. The System inform both the coach and the swimmer when an incorrect movement is occurred, and the system analyzes it. The system achieved 91% accuracy for detecting the incorrect stroke that is recognized with dynamic time wrapping algorithm. [12]

The key aim behind this research paper is that to solve the problem of the swimming recognition and lap counting by applying a wearable device to track and analyze all the swimmer's pattern. After that, the data was utilized to train a convolutional neural network to distinguish the four main swimming styles, as well as transition periods and lap turns. Their technique achieved a 97.4% F1 score for style recognition and a 99.2% F1 score for lap counting. [13]

Efficient Accelerometer-Based Swimming Exercise Tracking by pika et al. [14] is a research that focuses on tracking swimming exercises with 3D accelerometer data and shows that human activities may be accurately recorded even with low sampling rates. Swimming activity is measured in three stages: first, the swimming style and turns are determined, then the number of strokes are counted, and finally, the swimming intensity is estimated. In addition, two different sensor placements are investigated (wrist and upper back). An upper back-worn sensor is more accurate than a wrist-worn sensor when the swimming style is recognized, while vice versa when it is not recognized, the sensors keep track of the number of strokes and measure the degree.

They proposed system that developed a novel method for evaluating and providing feedback on performance of freestyle swimming stroke. During a 10-week training period, they looked at the effects of SmartSwim training on the performance of a group of swimmers. SmartSwim's delivers objective feedback during training sessions with a lightweight and by using IMU which was proven by the results. [15]

They proposed sports feedback systems which handle functions like gathering, evaluating, and representing data. These systems are designed to not only offer athletes and coaches with information on their performances, but also to assist athletes in learning new activities and controlling movements in order to avoid injuries and improve their performance. Designing mobile feedback systems, on the other hand, necessitates a high level of competence from researchers and practitioners in a variety of fields. They used Direct Mobile Coaching (DMC) as a design paradigm and model for mobile feedback systems as a solution to this challenge aside from feedback provisioning, which includes data recording, storage, and administration components. [16]

The proposed system is based on (IMU), and the main recognition method is dynamic time warping (DTW). They utilize optimized DTW in order to increase recognition accuracy. The capital alphabet (from 'A' to 'Z') achieved an accuracy 84.6% to be recognized. Furthermore, the recognition method based solely on the DTW algorithm is the type of user-dependent approach. [17]

Handwriting recognition by derivative dynamic time warping methodology via sensor-based gesture recognition by E. Tunce al. [18]. This research uses DTW to recognize handwritten characters, which is demonstrated by using acceleration signals collected from gesture sensors. Similarities are found by DTW after pre-processing. Then, when signals have acceleration and deceleration, they used DTW, which is stand for derivative dynamic time wrapping.

III. Dataset Collection

There are many challenges that faced us during searching for a public dataset that could fulfil our objective. The first challenge was that there was no csv file or text file containing only butterfly strokes. Another challenge was that even if the dataset contained butterfly stroke, it won't have the starting and ending point of the stroke [13]. Hence, new dataset is needed to be collected from swimmers in real time within coordinated sequence. As shown in figure 2, the data set was collected in two different ways which are outside the water and inside the water. A smartphone was worn on the swimmer's wrist and placed in a horizontal position. The dataset is collected by a swimmer and resulted as text files. The data set are collected from beginners and elite swimmers whom are from 16-18 years old. The swimmer's height are from 169-175.

Fig. 2. Inside & Outside the swimming pool

As shown in figure 3, Those are two samples of our collected dataset which shows the correct technique of the two collected datasets with the same gesture. The left picture indicates the incorrect stream inside the swimming pool while the right picture indicates the incorrect stream of the collected data outside the swimming pool.

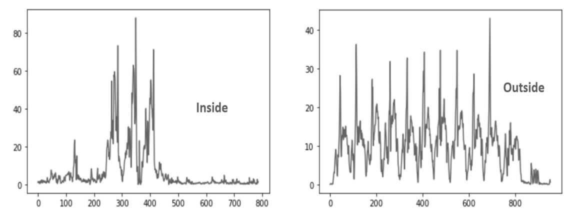


Fig. 3. Sample of the data Inside & Outside the swimming pool

IV. Proposed system

The proposed system classifies the technique of butterfly strokes, whether it detects the correct or the incorrect technique, and generates reports to the coach, swimmer, and parents. The system uses two sensors which are accelometer and gyroscope.

As shown in figure 4, data are collected by a mobile application. The data is collected by fixing a mobile on the wrist of the arm to collect the data which is X, Y, Z and Timestamp by two sensors that are accelometer and gyroscope. Since the system used two sensors, we made sensor fusion to add the attributes of the two sensors with each other. In addition, the data will be sent to the database by 4G when the swimmer's arms go up the swimming pool, on the other hand all the data will be uploaded to google cloud. The data set is collected as points x, y and z, so the data must be segmented.

The data is segmented to know the start and the end of the stroke by windowing the data, which is every 100 point that is approximately 2 seconds is for only one stroke as shown in figure 5. In addition to that, the Vector features are applied on segmented data which are vector magnitude as shown in equation

$$\sqrt{x^2 + y^2}$$

, hence there are only 3 points in the file which are magnitude, z and time stamp. However, there are two csv files with different sizes which are compared to each other, so the data must be re-sampled to be able to classify it as shown in figure

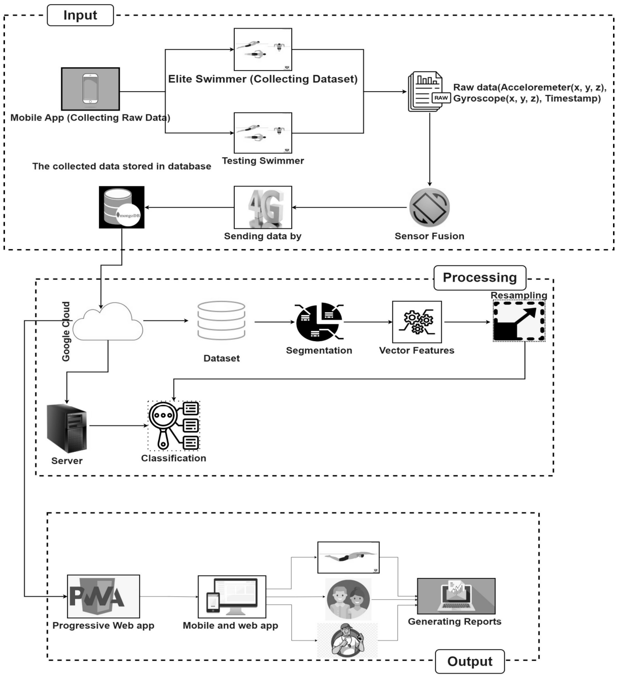


Fig. 4. System Overview

6. The left picture is before re-sampling and the right one is after re-sampling. The system uses four different models which are dynamic time wrapping, long short-term memory, 1-$ recognizer and k- Nearest neighbor, and then all the classified data will be sent to the server. Finally, the output is a progressive web app that the data will be generating as a report for every swimmer per day to the parents, coaches, and swimmers. Consequently, the reports will be full of many data about the swimmer like heart rate, his performance and his technique as shown in figure 7. In conclusion, the system will generate reports and it detects the wrong and right technique of butterfly stroke.

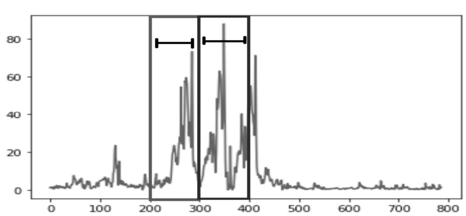


Fig. 5. Segmented data

V. Experiments

There are two experiments which are constructed to evaluate our proposed system. The objective of the first experiment is to test all the algorithms outside the swimming pool to know which model has the best accuracy. The classification phase

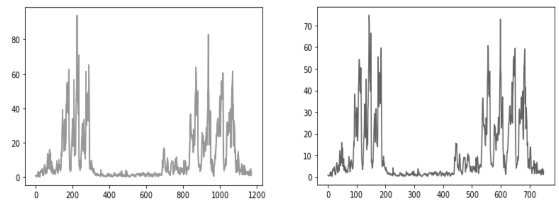


Fig. 6. After and Before Re-sampling

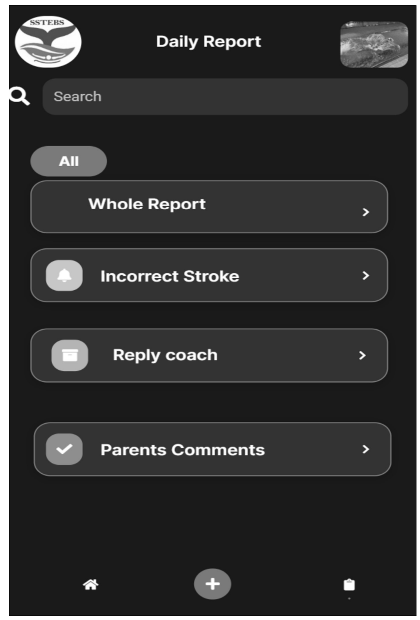


Fig. 7. Report prototype

was made on dataset consists of 10 correct and 10 incorrect strokes, also collect 20 correct and incorrect strokes from 8 swimmers. As for the second experiment, the main aim is to use the best algorithm that is performed from the previous experiment inside the water. In addition to that, the dataset which is used for the classification within a distance 12.5 and 25 meters was placed in an Olympic swimming pool during the training, it was collected by 5 elite swimmers stream and were inquired to swim 10 correct, incorrect, and mixed strokes. Each swimmer's data was gathered by asking them to collect their data by placing their smartphone on a horizontal position and swim with it. Each user recorded each stroke many times, giving us a total of many samples for each stroke. The dataset which are collected by the user are divided into two samples, in which it is utilised as training data, while the rest is used as testing data.

A. Experiment 1

The purpose of this experiment is to evaluate the classification models to determine which one will be the best to employ it in our system. Four Models were used to test our

TABLE I
MODEL'S ACCURACY

Models	Accuracy
Dynamic Time Wrapping	**80%**
K-Nearest Neighbour	**62.8%**
1$ recognizer	**73.2%**
Long Short Term Memory	**71%**

classification dataset which are LSTM, DTW, Knn, and 1$ - recognizer. The data is collected by 8 swimmers whom are 5 elite swimmers and 3 beginners. Although, those swimmers were inquired to record 10 correct movements and 10 incorrect movements separately. They also collected 20 strokes (correct and incorrect Strokes) outside the water. The results showed that the best model is dynamic time wrapping (DTW) as shown in TABLEI.

B. Experiment 2

This experiment involves determining the classifier's accuracy in classifying a stream of butterfly strokes. It took place in an Olympic swimming pool during the training phase. Also, the data was collected by wearable device and they were given two scenarios after gathering data from 5 elite swimmers, then we asked them to swim 10 correct, incorrect, and mixed strokes on a distance of 12.5 and 25 meters. Finally, the classifier will detect the incorrect techniques which achieves an accuracy 79%, 82% respectively.

VI. RESULTS AND DISCUSSION

The proposed system manipulates four different classifiers which are LSTM, KNN, 1$-recognizer and DTW. The first classifier is Long short-term memory which labeled the data and then tested the data with 150 epoch and reached an accuracy 71%. Regarding K-nearest neighbor, it uses three different K-values which are k = 1, k = 5, k = 3, however the K-value = 3 is the best accuracy which is 62.8%. Concerning 1$-recognizer, there are two templates which are the correct and incorrect techniques. Hence the templates were as two points which are vector magnitude(x&y) and z, and it reached an accuracy 73.2%. Therefore, We found that Dynamic Time Wrapping produces the best results within an average 80.5% among the evaluated classifiers. In conclusion, DTW is the only classifier that will be utilized by the proposed system. DTW is the only classifier that can distinguish between the correct and incorrect stroke. We make a questionnaire with a swimming coach and many swimmers, they told us that this system is extremely useful because if the swimmers know their incorrect techniques and solve it, hence, their performance will be improved and their time in the training and championships will be enhanced. As well, the system will aid the intermediate swimmers and their parents to know all their improvements using the reports that will be generated from the system.

VII. CONCLUSION AND FUTURE WORK

The proposed system analyzes the swimming butterfly stroke and can differentiate between correct and incorrect technique. The system used Dynamic time wrapping algorithm (DTW) as a classifier to know the difference between the correct and incorrect stroke with an accuracy 80.5%. Our future work is to know what type of the wrong technique is. The mobile will be replaced by a smartwatch with sdk and with accelerometer sensor will can collect the dataset with it. The system should inform the coach and the swimmer that there is a wrong technique by showing a notification (vibration) to the swimmer for enhancing and improving his style and his time inside the water.

REFERENCES

[1] M. L. Sein, J. Walton, J. Linklater, R. Appleyard, B. Kirkbride, D. Kuah, and G. A. Murrell, "Shoulder pain in elite swimmers: primarily due to swim-volume-induced supraspinatus tendinopathy," *British journal of sports medicine*, vol. 44, no. 2, pp. 105–113, 2010.

[2] H. Walker, B. Gabbe, H. Wajswelner, P. Blanch, and K. Bennell, "Shoulder pain in swimmers: a 12-month prospective cohort study of incidence and risk factors," *Physical Therapy in Sport*, vol. 13, no. 4, pp. 243–249, 2012.

[3] J. Marshall, "Smartphone sensing for distributed swim stroke coaching and research," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct. New York, NY, USA: Association for Computing Machinery, 2013, p. 1413–1416. [Online]. Available: https://doi.org/10.1145/2494091.2496036

[4] F. Struyf, A. Tate, K. Kuppens, S. Feijen, and L. A. Michener, "Musculoskeletal dysfunctions associated with swimmers' shoulder," *British journal of sports medicine*, vol. 51, no. 10, pp. 775–780, 2017.

[5] D. Tarasevičius and A. Serackis, "Deep learning model for sensor based swimming style recognition," in *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE, 2020, pp. 1–4.

[6] R. Delgado-Gonzalo, A. Lemkaddem, P. Renevey, E. M. Calvo, M. Lemay, K. Cox, D. Ashby, J. Willardson, and M. Bertschi, "Real-time monitoring of swimming performance," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 4743–4746.

[7] E. Ramos Félix, H. P. da Silva, B. H. Olstad, J. Cabri, and P. Lobato Correia, "Swimbit: A novel approach to stroke analysis during swim training based on attitude and heading reference system (ahrs)," *Sports*, vol. 7, no. 11, 2019. [Online]. Available: https://www.mdpi.com/2075-4663/7/11/238

[8] U. Jensen, F. Prade, and B. M. Eskofier, "Classification of kinematic swimming data with emphasis on resource consumption," in *2013 IEEE International Conference on Body Sensor Networks*, 2013, pp. 1–5.

[9] C.-C. Jeng, "A low-cost mobile real-time monitoring system for analyzing head position and breathing patterns in front crawl swimming," *Journal of Computers*, vol. 32, no. 2, pp. 8–22, 2021.

[10] M.-S. Pan, K.-C. Huang, T.-H. Lu, and Z.-Y. Lin, "Using accelerometer for counting and identifying swimming strokes," *Pervasive Mob. Comput.*, vol. 31, no. C, p. 37–49, Sep. 2016. [Online]. Available: https://doi.org/10.1016/j.pmcj.2016.01.011

[11] R. Mooney, G. Corley, A. Godfrey, L. R. Quinlan, and G. ÓLaighin, "Inertial sensor technology for elite swimming performance analysis: A systematic review," *Sensors*, vol. 16, no. 1, p. 18, 2016.

[12] M. Ehab, H. M. M. Ahmed, M. Hammad, N. ElMasry, and A. Atia, "Iswimcoach: A smart coach guiding system for assisting swimmers free style strokes." MeC'20 Workshop, 2020.

[13] G. Brunner, D. Melnyk, B. Sigfússon, and R. Wattenhofer, "Swimming style recognition and lap counting using a smartwatch and deep learning," in *Proceedings of the 23rd International Symposium on Wearable Computers*, ser. ISWC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 23–31. [Online]. Available: https://doi.org/10.1145/3341163.3347719

[14] P. Siirtola, P. Laurinen, J. Röning, and H. Kinnunen, "Efficient accelerometer-based swimming exercise tracking," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011, pp. 156–161.

[15] M. Hamidi Rad, V. Gremeaux, F. Massé, F. Dadashi, and K. Aminian, "Smartswim, a novel imu-based coaching assistance," *Sensors*, vol. 22, no. 9, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3356

[16] M. Dobiasch, S. Oppl, M. Stöckl, and A. Baca, "Direct mobile coaching as a paradigm for the creation of mobile feedback systems," *Applied Sciences*, vol. 12, p. 5558, 05 2022.

[17] Y. Luo, J. Liu, and S. Shimamoto, "Wearable air-writing recognition system employing dynamic time warping," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[18] E. Tuncer and M. Ünlü, "Handwriting recognition by derivative dynamic time warping methodology via sensor-based gesture recognition," *Maejo International Journal of Science and Technology*, vol. 16, pp. 72–88, 04 2022.

4.5.6 Prediction of Hypotension in Hemodialysis Session

Prediction of Hypotension in Hemodialysis Session [5]

ICECET 2022
International Conference on
Electrical, Computer and Energy Technologies
20-22 July 2022, Prague

08/06/2022

ACCEPTANCE LETTER

Dear ***Salma Gheith, Ahmed F. Al-Sadek,***

Thank you for your submission to the ICECET 2022 conference. We are pleased to inform you that your paper entitled "**ID-969 Prediction of Hypotension in Hemodialysis Session"** has been accepted as a full paper for **oral presentation** by the conference committee of *International Conference on Electrical, Computer, and Energy Technologies (ICECET)*. The event will take place in Prague, Czech Republic on 20-22 July 2022 **online** and **physically.**

We strictly follow "no podium, no paper" policy and only the papers that are presented at the conference will be submitted to IEEE Explore for publication. **At least one author** of an accepted paper must register (as a full participant) and participate in ICECET 2022 online or physically for the paper to be included in the proceedings. If you have not yet registered online (using the credit card or bank transfer options), at least one author of each paper should register to the conference via the online registration page at https://www.ecres.net/icecet. If you have already registered, please do not make another registration. Kindly note that your registration becomes valid only after your payment.

According to the conference regulations, only those papers which have been duly registered and presented on the conference day are considered for submission to IEEE Explore. The conference program will be communicated in due course.

We look forward to seeing you for a fruitful research and innovation event and for a great time in the wonderful environment of Prague

Yours sincerely,

Dr. Simon Winberg
Chair

Prediction of Hypotension in Hemodialysis Session

1st Salma Gheith
Faculty of Computer Science
October University for Modern Sciences and Arts (MSA)
Giza, Egypt
Email: salma.adel7@msa.edu.eg

Ahmed F. Al-Sadek
Faculty of Computer Science
October University for Modern Sciences and Arts
6th of October City, Giza, Egypt
Climate Change Information Center
Renewable Energy and Expert System (CCICRES)
Agricultural Research Center (ARC)
Giza, Egypt
Email: afarouk@msa.edu.eg

***Abstract*—There are 4.35 million people who receive the hemodialysis treatment or undergoing a transplant as they had suffered from the session's duration or the session's complications. Therefore, this research aims to solve a daily nephrology problem. The main problem is that how a hemodialysis patient suffer for several hours multiple time per week and may have complications during the session which may lead to severe risks. Hypotension is one of the critical problems that faces any dialysis patient. According to nephrologists and previous research papers showed how difficult it was to treat and detect the occurrence of hypotension. To achieve this target, a few steps were needed. Firstly, the data set is required to be from dialysis session to monitor the patient records from the dialysis machine. After preparing the artificial neural network (ANN) model to be able to predict, it needs some adjustment to make it efficient to use and a powerful model to handle the intradialytic event. In conclusion, the research aims to predict the occurrence of hypotension during the session using ANN.**

***Index Terms*—Artificial Neural Networks, Hemodialysis, Hypotension**

I. Introduction

Chronic kidney disease (CKD) is a long-term disease in which the kidney stops filtering the toxins and the waste that the body contains. The main symptoms are decrease in urine percentage, low mental alertness and daily variations in blood pressure. There are 840 million patients that suffer from CKD where they need renal replacement treatment immediately to avoid deterioration according to Jager, Kitty J et al [1]. The renal replacement treatment is divided into 3 types: peritoneal dialysis (PD), hemodialysis (HD) and continuous renal replacement therapies (CRRT). However, according to a statistical experiment that took place in America in 2021 [2], it had been discovered that nearly 500 thousand patients received hemodialysis and 230 thousand patients received a kidney transplant. The hemodialysis is the machine that filters the blood among multiple procedures inside the machine. Fig. 1. shows the hemodialysis procedure as shown by Hattersley, John [3]. This figure shows some essential attributes which are needed to help in prediction process such as arterial pressure monitoring.

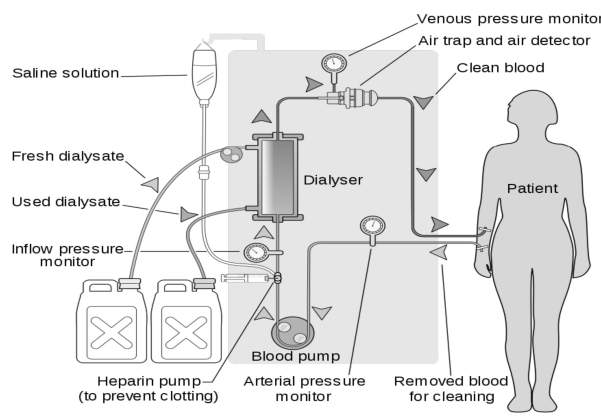


Fig. 1. Simplified process of hemodialysis

As hemodialysis includes blood filtration cycle, there should be monitoring for the patient for 4-6 hours during the session. As a result of the long duration of the sessions, there are a lot of complications that may occur. For instance, extreme variation in blood pressure (low or high), bone diseases, anemia and fluid overload. The 2 types of blood pressure variation which are hypotension and hypertension [4]. Furthermore, hypotension during hemodialysis might result in mortality and convulsions. As a result, hypotension is difficult to diagnose and cure and it is regarded as a significant intradialytic occurrence.

In this Research, it aims to handle the suffering of hemodialysis' patients. Recently, computer science contributed in medical field to solve many problems within medical procedures. The prediction of the hypotension is considered as a challenge whether in the medical field or the technical field. Therefore, the purposed solution will predict the occurrence of the hypotension during the session to minimize the intradialytic event's complications. Despite that many research papers used machine learning classifier such as support vector machine

978-1-6654-7087-2/22/$31.00 ©2022 IEEE

(SVM) and decision tree (DT), but both weren't suitable for neither imbalance nor large dataset. On the other hand, some used deep neural networks (DNN) [5] and recurrent neural network (RNN) for the prediction. However, multiple research papers highlighted ANN as a different approach for the detection field. The ANN model is powerful in the medical field and the monitoring of clinical parameters.

In conclusion, the model has 2 main problems which are data imbalance as the occurrence of hypotension is a minor event and the prediction of the occurrence of hypotension is considered a problem. Therefore, the model uses ANN and data re-sampling techniques.

II. Related work

Clearly, there are multiple attempts for computer science to contribute in solving a problem in kidney failure disease. Starting with the use of machine learning classifier to predict the kidney disease. The techniques used were decision tree, linear regression and k-nearest neighbors algorithm (KNN) with 3 different models used for training [6]. Moreover, there are multiple researches made for prediction of hypotension in hemodialysis session.

Firstly, a research focused on hypotension risk through the dialysis machine ultra-filtration rate and protein intake. The research used ANN to solve some mathematical problems using both protein intake (PCR) and urea clearance (Kt/v) [7]. In addition to, the research compared between linear regression model and ANN. The comparative study targeted the PCR, Kt/v and the hypotension risk to see the independence and which the model will have high performance. Secondly, the other research used machine learning classifier to predict the occurrence of hypotension through the session [8]. The research had used a database from a dialysis unit to monitor 6 hour dialysis sessions. Initially, they needed to pre-process the data then apply decision tree model. Also, the model used SVM to improve the performance of the model. Unfortunately, the research concluded that hypotension is hard to detect or treated. The latest research is real-time prediction of the occurrence of hypotension during the session using RNN and predicts the level of hypotension and its risk of occurrence [9].

In conclusion, there are multiple useful researches which helped in monitoring the intradialytic events which occurs unpredictably and might be uncontrollable. Therefore, the challenge of predicting the hypotension was the main target for this research.

III. Material and Methods

A. System Overview

Generally, Fig. 2 shows the proposed system which is composed of multiple stages. Firstly, the dataset is taken from dialysis machine after the session ends converting it into excel files. Secondly, due to dataset imbalance there are multiple techniques applied to re-sample the data. Finally, the decision whether the patient had hypotension or not will be taken after the model training stage to be able to construct the decision.

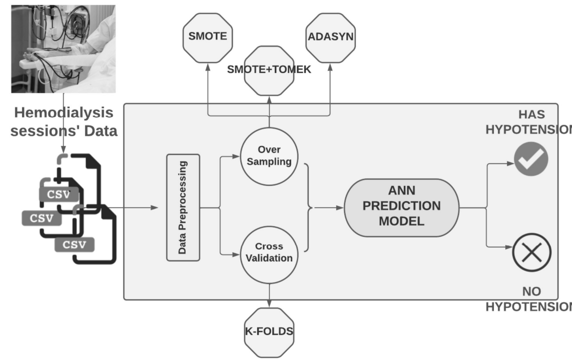


Fig. 2. Proposed system for the hypotension detection

B. Data

Firstly , the dataset is from dialysis database collected from Hospital Principe de Asturias, Madrid, Spain since 2016 till 2019 [10]. The dataset is composed of 11 files of hemodialysis sessions. Secondly, there are 24 features shown in Table 1 which are mostly clinical parameters fetched from the dialysis machine. The data didn't need much prepossessing but there was some attributes that were unnecessary.

TABLE I
Dataset parameters

Initials	Parameter
SEX	Sex
AGE	Age
DIA	Dialyzer
WDR	Weight(dry)
WPR	Weight(pre)
WPO	Weight(post)
IWG	Intradialytic weight gain
HBT	Hemodialysis bath temperature
VOL	Volume changes
KT	Urea clearance
BFR	Blood flow rate
HBF	Hemodialysis bath flow
HBC	Hemodialysis bath conductivity
BSC	Bicarbonate-based solutions conductivity
APR	Arterial pressure
VPR	Venous pressure
TMP	Transmembrane pressure
SBP	Systolic blood pressure
HRA	Heart rate
TUF	Total ultrafiltrations
BOT	Body temperature

C. Methodology

After the dataset was collected as excel files, it had 2 stages to pass through to solve the data imbalance and prepare it for the model. Firstly, the data preprocessing was the simple stage which had to filter the attribute according to the medical criteria. Moreover, the changes need for the dataset was applied and the dataset will be split into train and test. Secondly, the re-sampling phase will have its role. Each technique should be applied separately. Therefore, K-folds, synthetic minority oversampling Technique (SMOTE), adaptive synthetic algorithm (ADASYN) and SMOTE+TOMEK applied in separated experiments as each technique has it's effect on the dataset. Firstly, the oversampling techniques were applied on train data only and the second time with train and test data.

- SMOTE:
  This oversampling technique creates synthetic new instance for the minority as shown in Fig. 3 The generation of the synthetic examples are in the features' space [13]. This approach solves the over-fitting problem and the imbalanced dataset by using specific methodology. So, this methodology depends on interpolation.

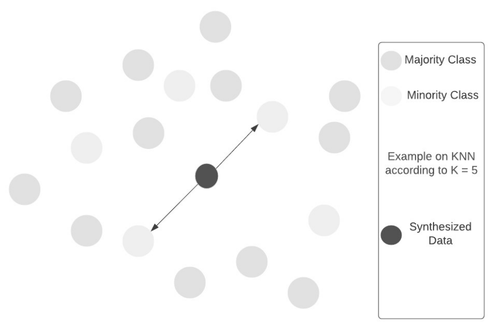


Fig. 3. Procedure of SMOTE synthetics creation

- ADASYN:
  This type of oversampling is a generalized technique of the SMOTE. Despite the similarity of the two techniques but the ADASYN which is shown in Fig. 4 uses the density distribution to calculate the synthetic instances amount for the minority class. Therefore the ADASYN is called the adaptive SMOTE due to its flexibility in deciding the minority class boundaries.

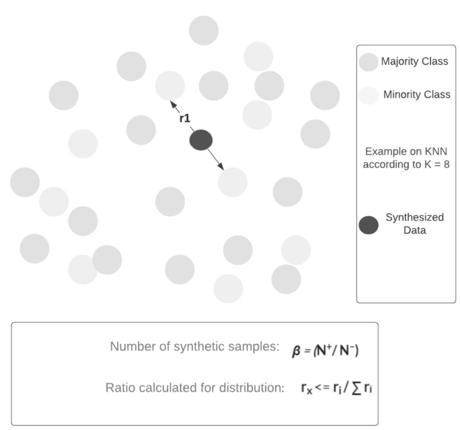


Fig. 4. ADASYN main process

- SMOTE+TOMEK:
  Firstly, the technique is a hybridization which combines the oversampling technique and the under-sampling as shown in Fig. 5. Moreover, the under-sampling is a technique which target the majority class to balance the data. Secondly, its aim is to have no overlapped data. Therefore, the SMOTE is applied in the first step then the TOMEK is applied on the over-sampled class.

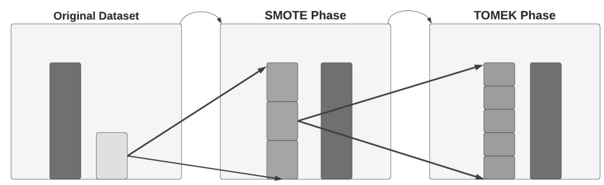


Fig. 5. A dataset shows before and after applying the SMOTE+TOMEK

- K-folds:
  The K-folds technique is one of the cross-validation types. It aims for splitting the dataset into parts to solve the imbalance problem without simulating new data points according to mathematical equation or specific theory. According to many previous researches the K-fold is a stable technique and flexible as it can decide the number of K (the number of splits for the data). Despite the high accuracy the k-fold technique may have but it's a time consuming technique when the dataset and the folds number increase [14]. The structure of K-fold is training the model according to the data split as shown in Fig. 6.

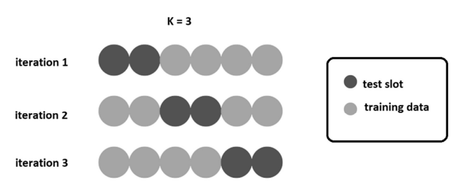


Fig. 6. Illustration for the K-FOLDS logic

  As a consequence, each iteration has its accuracy. The final accuracy is the average of performance of each iteration.

After applying each of the previous techniques separately, the effect on the dataset will be visualized to monitor the difference and time consumption. Obviously, increasing the data size will gradually consume more time and some techniques require changes to adjust to larger data.

The next step is to build the model after the imbalance problem is solved. Artificial neural network was the selected model as it shows the enhancement in prediction field especially prediction in medical field. The dataset was added file by file to test the model after the re-sampling. Due to increasing the trials, the model had been changed multiple times. The architecture of the model is shown in Figure 3. In another trial according to the previous research [15]. The use of decision tree wasn't suitable with the imbalanced data. The decision tree trial was approximately 68-70%. Despite the usage of ANN according to another research had proved its efficiency. According to multiple previous research in the hemodialysis field, the usage of ANN was suiting the purpose and helped in monitoring and predicting the clearance of toxins, the protein intake for the patients [16] and the blood pressure [17].

D. Architecture

Firstly, the simple architecture shown in Fig. 7 for the ANN model. The architecture of the model which all the dataset is used in it. The model had been developed by increasing the dataset capacity. There were 2 models' architecture. The first architecture is illustrated in Fig. 8.

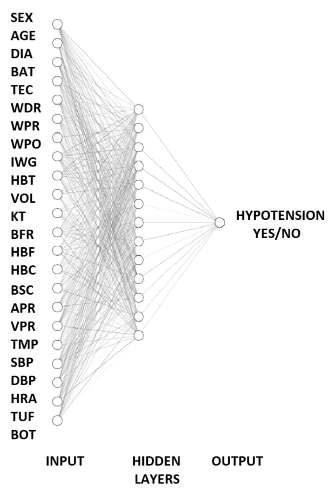


Fig. 7. Artificial neural network visualization

The first architecture is for the experiments with oversampling techniques and cross-validation with an increasing value of dataset. On the other hand, the other model is for the K-fold with a size bigger than the one used in the first model. The epochs number was 200 for the 2 models. In addition to, the optimizer used is Adam with the binary cross entropy loss function. The binary cross entropy was the most suitable loss function as it compares the probabilities of prediction to decide whether the data belongs to a class or belongs to another one.

Furthermore, the model will have different effects on each re-sampling technique due to its difference in the mechanism and structure. Each technique varies in time consumption, how it creates the synthesized data and the data re-sampled size.

The second model is developed on the success of K-folds. The previous model needed adjustments to adapt on large biased dataset. The 2 models select 5 folds with no other trials to decrease or increase the folds size. The differences are in the model architecture, the data size and the increase of time taken for the iterations to take place.

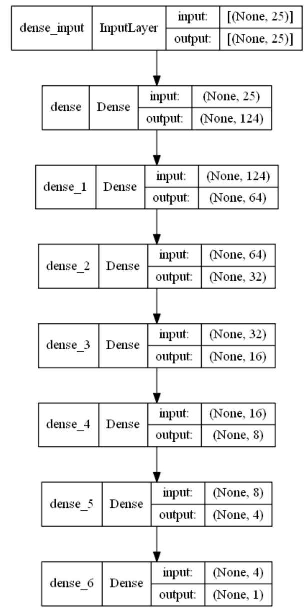


Fig. 8. The first model architecture

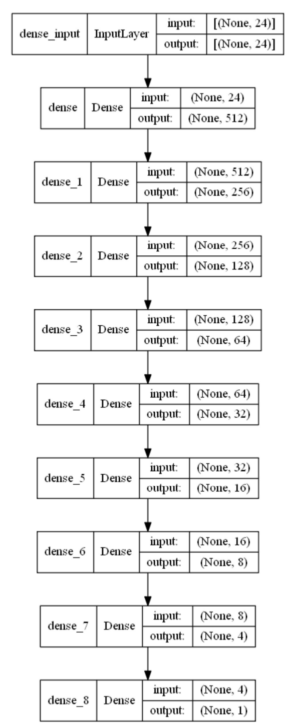


Fig. 9. The second model architecture

IV. EXPERIMENTS AND RESULTS

Indeed after building the structure of the model and understanding the methodology of every technique and its effect on the data, the experiments had taken place simultaneously.

- The First Model Experiments:

  The first model architecture had been built for each of the oversampling techniques and the cross-validation k-fold technique. The experiments are divided into 3 sections according to the data size incrementally as shown in Table 2. The data size is split to training and testing section. Firstly, with 284,008 patients' records each of the oversampling techniques had approximately the same accuracy while the k-fold technique had much higher accuracy relatively to theirs. Secondly, the increasing of the data will show that SMOTE and SMOTE+TOMEK nearly had the same effect as the 2 techniques are consisted of SMOTE logic. In addition to the increase of K-fold was only with 0.72%. The final experiment shows that the SMOTE had succeeded as it had the highest performance with this amount of data while the ADASYN and SMOTE+TOMEK had failed with the increasing of the data as shown in Fig. 10 and Fig. 11. Furthermore the K-fold had the highest performance due to its efficiency in solving the biased data.

  Despite the efficiency of the previous model, the extra increase of the data leaded to a major change to SMOTE performance and a minor change to K-fold performance as shown in Table 2. Hence the second model was developed to handle more data.

TABLE 2
THE RESULTS OF THE FIRST MODEL USED WITH DIFFERENT RESAMPLING TECHNIQUES

Testing Accuracy				
Techniques		**Data Size**		
		284,008	**376,359**	**464,624**
Oversampling	**SMOTE**	75%	76.34%	80.53%
	ADASYN	74%	75.9%	78.08%
	SMOTE+TOMEK	73%	76.98%	77.95%
Cross Validation	**K-folds**	81.13%	81.95%	82.25%

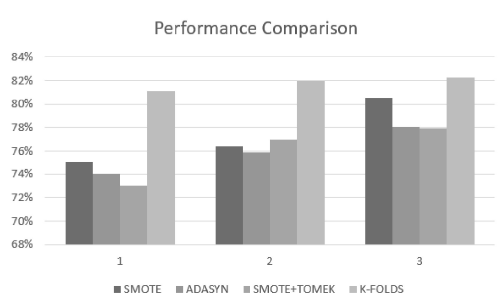


Fig. 10. Chart 1 to determine the accuracy (y-axis) of each technique in the first model experiment

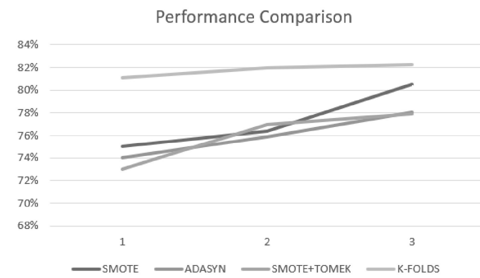


Fig. 11. Chart 2 to determine the accuracy (y-axis) of each technique in the first model experiment

- The Second Model Experiments:

  While the second model has been developed for the success of K-fold experiments and its high performance. In addition to a development for SMOTE technique due to its great performance. For these experiments the ratio of existence of hypotension increased from 1:9 (yes to no) to nearly 3:7. This increase helped in training the model in more efficient way as shown in Fig. 12 and Fig. 13.

  Although the SMOTE had proven how powerful and effective it is but the K-folds had approached in the most effective way for this biased data whether it contained low ratio of yes portion or high ratio as shown in Table 3. Therefore, the K-folds with the model improvements had reached its target.

TABLE 3
THE ACCURACY RESULTS OF THE SECOND MODEL USED WITH DIFFERENT RESAMPLING TECHNIQUES

Testing Accuracy			
	Data Size		
Techniques	**632,318**	**690,711**	**716,752**
SMOTE	92.58%	92.75%	93%
K-folds	92.7%	93.6%	98.29%

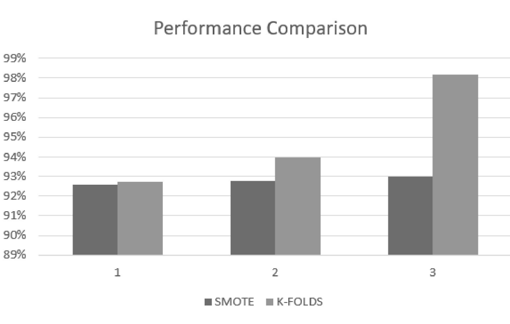


Fig. 12. Chart 1 to determine the accuracy (y-axis) of each technique in the second model experiment

V. DISCUSSION

In fact, a previous research which collected the dataset used in this research paper, used decision tree and SVM to

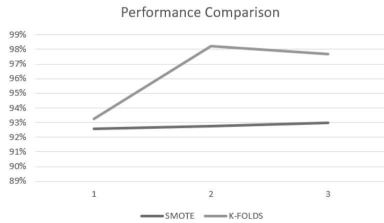


Fig. 13. Chart 2 to determine the accuracy (y-axis) of each technique in the second model experiment

predict the hypotension. The results were between 74-80%. While these machine learning classifiers weren't suitable for the data imbalance. This research aimed to solve the imbalance problem by using data pre-processing and data re-sampling techniques before using the ANN model. As ANN model can handle enormous data sets and can find complex nonlinear interactions between dependent and independent variables implicitly. Hence,the final accuracy was 98.29% which was achieved by k-folds with the ANN model.

VI. Conclusion & Future Work

For emphasis, there are 15-20% of death due to hemodialysis treatment. A percentage of the death happens during the session due to hypotension. This research follows the third section of the SDGs which is to ensure the health of human beings and take care of their treatment to avoid serious health issues. On the technical side, the BP monitoring during the hemodialysis sessions is difficult whether it's hypotension or hypertension. The hypotension is considered more difficult to detect and to treat as the drop occurs rapidly. The research aimed to firstly detect the occurrence of it by using the most sufficient way to deliver a solution for this problem. Initially, the ANN is a powerful to for detection the occurrence. Also, by using the k-folds technique as a cross-validation for the biased data. Both techniques had served the problem in an efficient way to develop a solution for detection. Therefore, the next step for the future work is to help in treatment or taking a fast real-time action during the dialysis by using fuzzy control system.

References

[1] Jager, Kitty J et al. "A single number for advocacy and communication-worldwide more than 850 million individuals have kidney diseases." Kidney international vol. 96,5 (2019): 1048-1050.

[2] Centers for Disease Control and Prevention. Chronic Kidney Disease in the United States, 2021. Centers for Disease Control and Prevention, US Department of Health and Human Services; 2021.
Nitta, K., Masakane, I., Hanafusa, N. et al. Annual dialysis data report 2017, JSDT Renal Data Registry. Ren Replace Ther 5, 53 (2019). https://doi.org/10.1186/s41100-019-0248-1

[3] Hattersley, John. (2022). Mathematical modelling of immune condition dynamics: a clinical perspective.

[4] The Egyptian Society of Nephrology and Transplantation ESNT Hemodialysis Guidelines 2019. J Egypt Soc Nephrol Transplant [serial online] 2020 [cited 2022 Apr 27];20:1-90. Available from: http://www.jesnt.eg.net/text.asp?2020/20/1/1/277816

[5] Chen, Jin Wu, Kuo-Chuan Moi, Sin-Hua Chuang, Li-Yeh Yang, Cheng-Hong. (2020). Deep Learning for Intradialytic Hypotension Prediction in Hemodialysis Patients. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2988993.

[6] Ifraz GM, Rashid MH, Tazin T, Bourouis S, Khan MM. Comparative Analysis for Prediction of Kidney Disease Using Intelligent Machine Learning Methods. Comput Math Methods Med. 2021 Dec 3;2021:6141470. doi: 10.1155/2021/6141470. PMID: 34899968; PMCID: PMC8664508.

[7] Luca Gabutti, Dario Vadilonga, Giorgio Mombelli, Michel Burnier, Claudio Marone, Artificial neural networks improve the prediction of Kt/V, follow-up dietary protein intake and hypotension risk in haemodialysis patients, Nephrology Dialysis Transplantation, Volume 19, Issue 5, May 2004, Pages 1204–1211, https://doi.org/10.1093/ndt/gfh084

[8] Gómez-Pulido JA, Gómez-Pulido JM, Rodríguez-Puyol D, Polo-Luque ML, Vargas-Lombardo M. Predicting the Appearance of Hypotension During Hemodialysis Sessions Using Machine Learning Classifiers. Int J Environ Res Public Health. 2021 Feb 28;18(5):2364. doi: 10.3390/ijerph18052364. PMID: 33671029; PMCID: PMC7967733.

[9] Lee H, Yun D, Yoo J, Yoo K, Kim YC, Kim DK, Oh KH, Joo KW, Kim YS, Kwak N, Han SS. Deep Learning Model for Real-Time Prediction of Intradialytic Hypotension. Clin J Am Soc Nephrol. 2021 Mar 8;16(3):396-406. doi: 10.2215/CJN.09280620. Epub 2021 Feb 11. PMID: 33574056; PMCID: PMC8011016.

[10] Gomez-Pulido, Juan A. (2021), "Dialysis database: sessions with valid data of clinical parameters", Mendeley Data, V1, doi: 10.17632/7kmtsmsgfw.1

[11] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo and J. Santos, "Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]," in IEEE Computational Intelligence Magazine, vol. 13, no. 4, pp. 59-76, Nov. 2018, doi: 10.1109/MCI.2018.2866730.

[12] More, Ajinkya. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets.

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (January 2002), 321–357.

[14] Aridas, C. (n.d.). Smote + tomek. SMOTE + Tomek - imbalanced-learn 0.3.0.dev0 documentation.

[15] 4. C.Decaro et al., "Machine Learning Approach for Prediction of Hematic Parameters in Hemodialysis Patients," in IEEE Journal of Translational Engineering in Health and Medicine, vol. 7, pp. 1-8, 2019, Art no. 4100308, doi: 10.1109/JTEHM.2019.2938951.

[16] Luca Gabutti, Dario Vadilonga, Giorgio Mombelli, Michel Burnier, Claudio Marone, Artificial neural networks improve the prediction of Kt/V, follow-up dietary protein intake and hypotension risk in haemodialysis patients, Nephrology Dialysis Transplantation, Volume 19, Issue 5, May 2004, Pages 1204–1211, https://doi.org/10.1093/ndt/gfh084

[17] Gómez-Pulido JA, Gómez-Pulido JM, Rodríguez-Puyol D, Polo-Luque ML, Vargas-Lombardo M. Predicting the Appearance of Hypotension During Hemodialysis Sessions Using Machine Learning Classifiers. Int J Environ Res Public Health. 2021 Feb 28;18(5):2364. doi: 10.3390/ijerph18052364. PMID: 33671029; PMCID: PMC7967733.

5. Cooperation Minutes of Understanding

Cooperation agreement to support the graduation project

To the Mr. / **Dean of the Faculty of Computer Science**

After Greeting..

Mazaraana company has the honor to cooperate with the College of Computer Science (October University for Modern Sciences and Arts) with the support of the graduation project submitted by the student / **Abdullah Ayman Abu Zaid**

And this is under the supervision of Dr. / Ayman Ezzat

Entitled (Exploring Local Meat Using Machine Learning)

The company provides the technical support and consultations necessary to test the initial model during its experimental stages with its knowledge.

Coordination will also be made for mutual visits between those in charge of the project and our concerned parties.

Mazaraana company

General Director

Samir Shoukry Riad

مزارعنا

س.ت، ٢٢١٨٠٣١

ب.ض، ٥/٤٢٣/١٥/٢١٢

اتفاقيه تعاون لدعم مشروع التخرج

الي السيد / **عميد كليه عاوم الحاسب**

تحيه طيبه و بعد ..

تتشرف شركه مزارعنا بالتعاون مع كليه علوم الحاسب (اكتوبر للعلوم الحديثه و الاداب) و ذلك بدعم مشروع التخرج المقدم من الطالب / **عبدالله ايمن ابوزيد**

و ذلك تحت اشراف دكتور / ايمن عزت

بعنوان (استكشاف اللحوم المحلية باستخدام التعلم الآلي)

و تقوم الشركه بتقديم الدعم الفني و التقني و الاستشارات اللازمه لتجربه النموزج الاولي خلال مراحله التجريبيه بمعرفته.

كما سيتم التنسيق لعمل زيارات متبادله بين القائمين على المشروع و الاطراف المعنيه لدينا.

شركه مزارعنا

المدير العام

سمير شكري رياض

مزارعنا

س.ت: ٢٢١٨٠٣٠

ب.ض: ٥/٤٢٢/١٥/٢١٢٠

NUTRIVET

إتفاقية تعاون لدعم مشروع التخرج

إلى عميد كلية علوم الحاسب

تحية طيبة وبعد،

تتشرف شركة نيوتريفت بالتعاون مع **كلية علوم الحاسب بجامعة أكتوبر للعلوم الحديثة الآداب**

لدعم مشروع التخرج المقدم من الطالب/ عبد العزيز أشرف

وتحت إشراف دكتور/ أيمن عزت

بعنوان : **"التعرف على الأمراض الأنماط السلوكية للجمبري "**

وتقوم شركة نيوتريفت بتقديم الدعم الفني والتقني الإستشارات اللازمة لتجربة النموذج الأولي خلال مراحله التجريبية بمعرفتها ، كما سيتم التنسيق لعمل زيارات متبادلة بين القائمين على المشروع و الأطراف المعنية لدينا.

شركة نيوترفيت

توقيع

NUTRIVET MISR

Villa 8, Elnabatat St., West Somid, 6 of October, Giza, Egypt.
+2 (02) 383 67 978 info@nutrivetmisr.com www.nutrivetmisr.com

اتفاقية تعاون لدعم مشروع التخرج

إلى السيد / **عميد كلية علوم الحاسب**

تحية طيبة وبعد ..

يتشرف مصنع البطل للملابس الجاهزة بالتعاون مع كلية علوم الحاسب (اكتوبر للعلوم الحديثة والآداب) وذلك بدعم مشروع التخرج المقدم من الطالبة / **ايريني يسري ويصا**

وذلك تحت اشراف دكتور / أيمن عزت

بعنوان (مؤازرة صندوق التعبئة احادى وثنائى الأبعاد)

ويقوم المصنع بتقديم الدعم الفني والتقني والاستشارت اللازمة لتجربة النموذج الأولي خلال مراحله التجريبية بمعرفته.

كما سيتم التنسيق لعمل زيارات متبادله بين القائمين على المشروع والاطراف المعنية لدينا .

مصنع البطل للملابس الجاهزة

المدير العام

محمد درويش

مصنع البطل
رخصة رقم ٢٠١٢/٤
ب.ض ٠٨٠-٥١٤-٣٠٧ أول
س.ت ٢٠١٣

Cooperation agreement to support graduation project

To the Dean of the College of Computer Science
After Greetings,

The Faculty of Pharmacy, department of Pharmacology & Toxicology, Cairo University has the honor to cooperate with the Faculty of Computer Science in October Modern Sciences and Arts to support the graduation project submitted by the student / Andrew Zaky Naguib
Under the supervision of Dr. Ayman Ezzat, entitled
"iRats: Intelligent system for rat behavior analysis"

And the consultant Lecturer. Dr. Reham Mohamed Essam provides the necessary technical support and
consultations to test the first model during its experimental stages with his knowledge, coordination will also be made for mutual visits between those in charge of the project and our concerned parties.

__Consultant ___

Signature

Reham Essam

Reham Essam

مركز العنان
لصيانة السيارات

اتفاقية تعاون لدعم مشروع التخرج

إلي عميد كلية علوم الحاسب

تحية طيبه وبعد،،

يتشرف مركز العنان لصيانة السيارات بالتعاون مع كلية علوم الحاسب **بجامعة أكتوبر للعلوم الحديثة و الأداب** لدعم مشروع التخرج

المقدم من الطالب / **عمر مجدي توفيق**

وتحت اشراف دكتور/ **أيمن عزت**

بعنوان "**التعرف على النشاط البشري في مراكز صيانة السيارات** "

ويقوم مركز العنان بتقديم الدعم الفنى و التقنى والاستشارات اللازمة لتجربة النموذج الاولى خلال مراحله التجريبية بمعرفته ،كما سيتم التنسيق لعمل زيارات متبادلة بين القائمين على المشروع و الاطراف المعنية لدينا.

مركز العنان لصيانة السيارات

توقيع

مركز العنان لصيانة السيارات
PORSCHE
س. ت: ٩٦٤٨٨

Abd El-Maguid Sidi, El Haram st, Giza , Egypt
+2(02)38512112 www.masrmotors.com

Cooperation agreement to support graduation project

To the Dean of the College of Computer Science

After Greetings,

Ass. Prof. Ahmed Mohamed Fahmy Yousef has the honor to cooperate with the Faculty of Computer Science in October Modern Sciences and Arts to su pport the graduation project submitted by the student / Mario Nady Shoukry Under the supervision of Dr. Ayman Ezzat, entitled "Multi-modalities analysis in profiled learning.".____________________

And the consultant provides the necessary technical support and consultations to test the first model during its experimental stages with his knowledge, coordinati on will also be made for mutual visits between those in charge of the project and our concerned parties.

__Consultant ___

Signature

Ahmed fahmi

__02/07/2022___

إتفاقية التعاون لدعم مشروع التخرج

إلى السيد / **عميد كلية علوم الحاسب**

تحية طيبة وبعد ...

يتشرف كل من الاستاذ "**هشام محمد على**" المنوط به رئيساً لقطاع أمن المعلومات ببنك الإمارات دبى الوطنى **Emirates NBD** و كلية علوم الحاسب بجامعة أكتوبر للعلوم الحديثة والأداب **MSA University** بالتعاون لدعم مشروع التخرج المقدم من الطالب / عمر طارق إبراهيم

وهو بعنوان *Forensic Identification of Handwritten Signature Using Deep Learning*

وتم هذا تحت إشراف الدكتور / أيمن عزت

وذلك دون تحمل ادنى مسئولية من قبل البنك.

HishamMohamed

Hisham Mohamed Aly
Chief Information Security Officer, Emirates...

Cooperation agreement to Support our graduation project

To Mr./ **Dean of the Faculty of Computer Science**

Greetings….

Egypt Swim Away academy under the supervision of coach **"Hazem Reda"** that has the honor to cooperate with October university for Modern science and Arts(MSA University) to support the graduation project which is submitted by the student/ **Salma Tamer**

This is under the supervision of Dr./ **Ayman Ezzat**

Under title: **Swimming system tracker for enhancing butterfly stroke by using machine learning**

Signature

c/ Hazem Reda
6/7/2022

Bibliography

[1] M. S. Abdel-Wahab, A. M. Neil, and A.Atia. A comparative study of machine learning and deep learning in network anomaly-based intrusion detection systems. In *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–6, 2020.

[2] Abdallah Abuzaid and Ayman Atia. Exploring and classifying beef retail cuts using transfer learning. In *International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages –, 2022.

[3] AbdelAziz Ashraf and Ayman Atia. Comparative study between transfer learning models to detect shrimp diseases. In *2021 16th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–6, 2021.

[4] Andrew Ayman, Shereen W. Arafat, Ahmed M. Hossam Eldin, and Ayman Atia. Dental implant recognition and classification with convolutional neural network. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 477–482, 2022.

[5] Salma Gheith and Ahmed F. Al-Sadek. Prediction of hypotension in hemodialysis session. In *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages –, 2022.

[6] Ahmed Hamada and Ayman Atia. Multi-modal online exam cheating detection. In *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages –, 2022.

[7] Mena Hany and Wael H. Gomaa. A hybrid approach to paraphrase detection based on text similarities and machine learning classifiers. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 343–348, 2022.

[8] Omar Magdy and Ayman Atia. Human activity recognition in car workshop. *International Journal of Advanced Computer Science and Applications*, 13(4), 2022.

[9] Omar Magdy and Ayman Atia. Human activity recognition in maintenance centers to reduce wasted time. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 118–124, 2022.

[10] Youssef Maged and Ayman Atia. The prediction of blood glucose level by using the ecg sensor of smartwatches. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 406–411, 2022.

[11] Mohammed Ahmed Matboli and Ayman Atia. Fruit diseases identification and classification using deep learning model. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 432–437, 2022.

[12] Habiba Mohamed and Ayman Atia. Exploring brain tumor classification using deep learning. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 483–488, 2022.

[13] Mario Nady and Ayman Atia. Multi-modalities analysis in profiled learning. In *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages –, 2022.

[14] Abdelrahman Ezzeldin Nagib, Mostafa Mohamed Saeed, Shereen Fathy El-Feky, and Ali Khater Mohamed. Neural network with adaptive learning rate. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 544–548, 2020.

[15] Abdelrahman Ezzeldin Nagib, Mostafa Mohamed Saeed, Shereen Fathy El-Feky, and Ali Khater Mohamed. *Hyperparameters Optimization of Deep Convolutional Neural Network for Detecting COVID-19 Using Differential Evolution*, pages 305–325. Springer International Publishing, Cham, 2022.

[16] Mostafa Mohamed Saeed and Wael H. Gomaa. An ensemble-based model to improve the accuracy of automatic short answer grading. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 337–342, 2022.

[17] Mohamed Amr Samir, Youssef Maged, and Ayman Atia. Exam cheating detection system with multiple-human pose estimation. In *2021 IEEE International Conference on Computing (ICOCO)*, pages 236–240, 2021.

[18] Mohamed Amr Samir, Zeinab A. Mohamed, Mona Abdelmotaleb A. Hussein, and Ayman Atia. Applying deep learning to track food consumption and human activity for non-intrusive blood glucose monitoring. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0319–0324, 2021.

[19] Salma Tamer and Ayman Atia. Recognition of butterfly strokes using different machine learning models. In *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages –, 2022.

[20] Omar Tarek and Ayman Atia. Forensic handwritten signature identification using deep learning. In *International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages –, 2022.

[21] Omar Tarek, Omar Magdy, and Ayman Atia. Yoga trainer for beginners via machine learning. In *2021 9th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, pages 75–78, 2021.

[22] Rana Reda Waly and Wael H. Gomaa. Extractive summarization of scientific articles. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 349–354, 2022.

[23] Eriny Wessa, Abdelaziz Ashraf, and Ayman Atia. Can pose classification be used to teach kickboxing? In *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6, 2021.

[24] Eriny Wessa and Ayman Atia. Parallelization of one dimensional first fit decreasing algorithm. In *2021 16th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–5, 2021.

[25] Andrew Zaky, Mona M Saber, Reham M. Essam, and Ayman Atia. Irats:intelligent system for rat behavior analysis. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 219–224, 2022.