Republic of Iraq
Ministry of Higher Education and Science Research
University of Babylon
Collage of Science for Women
Department of Computer Science

Classification of Macular Degeneration Based on Deep Learning Method

A Thesis
Submitted to the Council of College of Science for woman,
University of Babylon in Partial Fulfillment of the Requirement for
Degree of Master of Science in Computer

By
Saja Mahdi Hussein

Supervised by
Dr. Enas Hamood Al-Saadi
Dr. Ali Yakoob Al-Sultan

2021 A.D. **1442 A.H**

بسم الله الرحمن الرحيم

قَالُواْ سُبْحَانَكَ لاَ عِلْمَ لَنَا إِلاَّ مَا عَلَّمْتَنَا

إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ

صَدَقَ اللهُ العَلِيُّ العَظِيم

سورة البقرة(٣٢)

Supervisor Certification

I certify that project entitled **"Classification of Macular Degeneration Based on Deep Learning Method"** was prepared at the Department of computer Sciences/ College of Science for Women/ University of Babylon, by (Saja Mahdi Hussein) as partial fulfilment of the requirements for the degree of Master in Computer Science.

Signature:

Name: Dr. Enas Hamood Al-Saadi

Date: / /2021

Address: Department of Mathematical, College of Education for Pure Sciences, University of Babylon, Iraq.

Signature:

Name: Dr. Ali Yakoob Al-Sultan

Date: / /2021

Address: Computer Science Dept. College of Science for Women, University of Babylon, Babylon, Iraq.

The Head of the Department Certification

In view of the available recommendations, I forward the research entitled "**Classification of Macular Degeneration Based on Deep Learning Method**" for debate by the examination committee.

Signature:

Name: Dr. Farah M. Hassan, Ph.D.

Date: / / 2021

Address: University of Babylon/College of Science for Women

Dedication

To the spring of tenderness, My Dear Parents

and all my family

Saja Mahdi Hussein

2021

Acknowledgments

Praise be to Allah whose will and guidance enable me to complete my graduate study.

I would like to submit my sincere gratitude to my supervisors Dr. Enas Hamood Al-Saadi and Dr. Ali Yakoob Al-Sultan for their advices, support, encouragement, and supervision throughout this work to be in the best manner.

I would also like to thank them for giving me the opportunity to work in the Artificial Intelligence and Deep Learning fields.

I would also like to acknowledge University of Babylon - College of Science for Women - Department of Computer Science for providing support in conducting this thesis.

In addition, sincere appreciation and love go to all my family members whose have encouraged me to go through this study and provides me with love and pure affection.

Finally, I would like to thank the people who gave me help and advice.

Abstract

Age Macular Degeneration (AMD) diseases is a widely spread eye condition that infects senior people. An automatic method of diagnosing the disease will help the doctors in monitoring the cases and patients, to avoid advanced bad eye condition.

In this proposed method, a model based on the Convolutional Neural Network (CNN) was trained on a dataset of eye images with multiple levels of AMD conditions and normal eyes. ODIR dataset is used for both the training and testing phases of the model. The goal is to develop a deep learning model, which will classify each image based on its AMD condition.

The training stages included two phases: first phase is to build and train a model to classify the images into two classes (Normal, and AMD). After the sufficient results obtained from that model, its design and training parameters are developed further to classify three classes of AMD disease, which the AMD class is extended to (Early, Intermediate, and Late). Which are the three stages of the disease AMD beside to the normal stage.

High accuracy results are obtained, and the proposed model is compared with other cutting edge approaches on the same dataset. Moreover, advanced and popular models are trained on the same dataset, the results revealed better performance for the proposed model over the popular models.

In the two classes proposed system, total accuracy of (0.98) is obtained, (0.975) Area Under the Curve, (0.985) Specificity, and (0.965) Sensitivity.

The four classes proposed system, total accuracy of (0.985) is obtained, (0.964) Area Under the Curve, (0.928) Specificity, and (1.00) Sensitivity.

To further evaluate the real performance of the system, case study is conducted. With total 50 images of real patients, retinas are collected from local vision center. And they are tested on our proposed system. The obtained results are (0.98) accuracy, (0.983) Area Under the Curve, (0.966) Specificity, and (1.00) Sensitivity.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

Abbreviation	*Meaning*
AHE	Adaptive Histogram Equalization
AI	Artificial Intelligence
AMD	Age-related Macular Degeneration
ANN	Artificial Neural Network
AUC	Area Under Curve
BP	Backpropagation
CDF	Cumulative Distribution Function
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
DA	Data Augmentation
DL	Deep Learning
DNN	Deep neural network
DT	Decision Tree
FC	Fully Connected
FFNs	Feedforward Neural Networks
FN	False Negative
FP	False Positive
KNN	K Nearest Neighbour
ML	Machine Learning
MLP	Multi-Layer Perceptron
ODIR	Ocular Disease Recognition
PNN	Probabilistic Neural Network
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

CHAPTER ONE

INTRODUCTION

1.1 Overview

Due to the increase in vision loss in elderly people, and since this is a very important problem to be dealt with, the researchers found that one of the main reasons for blindness in elderly people is a disease called Age-related Macular Degeneration (AMD). This disease has developed because of a material called "drusen" that starts to accumulate in the eyes, causing damage to the vessels and tissues of the eye. This causes blurry and unclear vision, eventually leading to vision loss [1].

It is caused by degeneration of the macula, the central portion of the retina responsible for crisp, sharp vision. In AMD, yellow deposits known as drusen (accumulation of extracellular proteins and lipids) gradually accumulate in the macula (a part of the retina). Over time, it is expected that this accumulation will cause damage to the retina. The term "macular" means it affects a part of your eye called the macula. “Degeneration” means the type of damage happening to the eye[2]. This disease currently affects approximately 8% of the world's population, or about 196 million people in 2020, and is anticipated to reach 288 million people by 2040 [3] [4].

AMD can be categorized up to three stages (early stage, intermediate stage, late stage) based partially on the extent (size and number) of the drusen [5]. The last (i.e. late) stage is additionally divided into "dry" and "wet" forms.

It is critical to diagnose and treat ocular disorders promptly to avoid permanent vision loss. Fundus photography in color is an efficient and cost-effective method of screening funds [6]. Due to the fact that limited indications are apparent during the early stages of illness, automated and effective diagnostic algorithms based on color fundus images are urgently needed.

To address this problem, this work proposes a system that automatically recognize the AMD, based on the fundus images of retina. The system should be able to read the fundus image of retina, and based on its features (drusen presence, size, and number). It should give an acceptable level of accurate diagnosis of the condition in the eye and whether the eye is healthy of having AMD. Furthermore, the stage of AMD.

The approach to building this system is based on Deep Learning (DL) techniques. Due to the efficient performance of Deep Learning algorithms in Computer Vision (CV) and image classification [7]. This means that this work will necessitate the collection of a dataset of normal and AMD-affected eye retinas in order to train the Deep Learning model on it.

In traditional Machine Learning methods, the feature engineering is time consuming, and require longer time during the process of prediction especially with images data. [8]

In Deep Learning, feature extraction and engineering is carried out by the model itself. And generally, it perform much better comparing with the manual feature extraction and engineering.

However; the major challenges in Deep Learning, are to configure the suitable architecture, optimization methods, and the training parameters for the Deep Learning model training.

Convolutional Neural Networks (CNN), is a Deep Learning algorithm, and most popular categories of neural networks, especially for high-dimensional data (e.g., images and videos). [9] Convolutional Neural Networks (CNNs) are deployed to learn the characteristics of AMD in this work.

1.2 Aims of Thesis

The objectives of the current study are as follows:

1. Conducting a literature review to summarize and compare modern machine learning trends to solve this problem.

2. Designing, building, upgrading, and training a Deep Learning model to extract features and classify the fundus retinal images based on their AMD condition.

3. The CNN model should be capable of distinguishing between healthy and AMD-affected eyes, with high-level accuracy.

5. Testing CNN performance through performance measures (e.g. accuracy, sensitivity, specificity, area under the curve)

6. The ability to classify the three stages of AMD (early stage, intermediate stage, and late stage) in addition to the normal stage

7. Comparing the performance of the proposed architecture and training parameters with other famous CNN architectures such as AlexNet and VGGNet on the same dataset.

1.3 Main Contributions of Thesis

The main contribution of the proposed system is the following:

1. Build a system that can classify the images of AMD and normal eye fundus images, with higher accuracy than the previously proposed methods.
2. Adopting fewer parameters than the most modern Deep Learning models (Convolutional Neural Network).

3. Higher results for the system compared with other techniques and also with the most common architectures using the same dataset.

4. Fast predicting time about 0.28 second per image.

5. Extending the binary classification system (AMD and normal classification classes), to cover all the three stages of AMD, in addition to the normal status of the eye.

1.4 Related Works

A number of studies have been conducted to cover the most related work and to provide a summary of the work for the classification of macular degeneration related to age. The suggested age-related macular degeneration classification system can be implemented using the bypass neural network (CNN) algorithm. In the following, the description of this research:

In 2020, Junjun and others proposed a new approach, they used ODIR dataset. They proposed an attention-based unilateral and bilateral feature weighting and fusion network (AUBNet) to automatically classify patients into the corresponding disease categories. Specifically, AUBNet is composed of a feature extraction module (FEM), a feature fusion module (FFM), and a classification module (CFM). The FEM extracts two feature vectors from the bilateral fundus photographs of a patient independently. With the FFM, two levels of feature weighting and fusion are proceeded to prepare the feature representations of bilateral eyes. Finally, multi-label classifications are conducted by the CFM. The higher results achieved by their model was AUC (0.934), F1 score (0.913), kappa (0.6402) [10].

In 2020, Ram and other researchers used ODIR dataset too. They used the deep convolutional neural network topology with N-Way fully connected layers. This investigation's main emphasis was on the classification of normal, cataract, AMD, and myopia. As the network's feature extraction component (i.e. the convolutional net) is trained, the feature mapping component (i.e. the linear net) of the network is also trained to different specifications. The greatest level of accuracy obtained was (0.819), and the highest level of specificity was (0.663), with a sensitivity of (0.714), and a specificity of (0.663) [11].

In 2019, Islam and other researchers used ODIR dataset to create a new technique for identifying malignant tumours. The convolutional neural network (CNN) has been used to diagnose eight different kinds of eye disorders, and the performance of the CNNs has been assessed. Some standard pre-processing is carried out before the data is transmitted to the network for rigorous categorization to be carried out. The greatest level of accuracy was obtained with an F-score of (0.85), a Kappa score of (0.31), and an AUC value of (0.80) [12]. This work is the most similar to our approach, in the researches showed in related work.

Dihao and Liping in 2020, used Vessel-Net to the ODIR-2019 dataset. This research offers a new vessel-aware ensemble network for the identification of retinal diseases. The global image together with the local medical-related areas, such as the local disc region and the local vessel regions, combine to create the global network. The results are formed by combining multiple results from several data sources. Their findings show that this technique is good for the state-of-the-art approaches, as well as demonstrating the significance of vessel-related regions. The maximum results obtained was: accuracy (0.84), sensitivity (0.87), and specificity (0.85) [13].

Junjun and others in 2020, used the ODIR Dataset too. They use a convolutional neural network-based patient-level multi-label ocular illness classification model considering the complexity of both patient-level diagnoses associated with bilateral eyes and multi-label disease classification. A dense correlation network (DCNet) is utilized to solve the problem. The backbone CNN extracts features, while the spatial correlation module is responsible for correlating features, and the classifier generates classification scores. CNN uses a single backbone that analyzes left and right color field pictures to extract two very different groups of attributes. Once the spatial correlation module has determined the correlations between the two feature sets, it gathers the pixel-by-pixel correlations between them. A processed representation is subsequently created using the processed attributes, which are then blended together to provide a patient-level representation. The greatest accuracy values found were Kappa=(0.63), F1 score=(0.91), and AUC=(0.93) [14].

Jing and other authors use the ODIR Dataset in their method one or more fundus diseases may be diagnosed based on CNN-style model imaging of fundus images that does not need any extra labeling information. The first half of the solution relies on an efficient net-based feature extraction network, while the second half is a customized classification neural network that is suitable for multi-label classification scenarios. Finally, in order to determine the final recognition result, multiple models' output probabilities are merged. The additional information is that it was trained and assessed using the provided dataset from the ODIR 2019. Accoding to the researchers, experimental findings demonstrate that their model may be trained on a smaller number of data sets while still producing satisfactory results. (0.89) Accuracy, Recall is (0.58). AUC is (0.73). and Precession is (0.63) [15].

In the year 2020, Neha and Pritee collaborate on a new project. The ODIR Dataset is used by the authors in this method. Multi-class multi-label fundus picture classification of ophthalmologic diseases are compared using two different approaches: one that utilizes transfer learning and CNN techniques to classify pictures, and another that does so using transfer learning and CNN methods. It is found that the VGG16 pre-trained CNN architecture with the SGD optimizer is best for classifying multi-class multi-label fundus photos. The best obtained results are F1 score (0.85), and AUC was (0.84) [16].

In 2018, Tan and other researchers, at the Kasturba Medical College (KMC) Ophthalmology Department, used Deep Learning (DL), and Convolutional Neural Network (CNN). Their model utilized neural networks of 14 layers, Seven convolutional layers, four max-pooling layers, and three fully linked layers comprise this image. The primary objective of the convolution process is to extract unique characteristics from the input fundus picture. The pooling process lowers the output dimension and guarantees that the output is of a fixed size. The output layer of the fully-connected layer is activated using a softmax function. The primary objective is to determine if the picture in the input file is normal or AMD. Backpropagation was used to train the CNN model with a batch size of 50. Additionally, the 'Adam' method was used to improve the CNN model's learning parameters. Results are (0.95) accuracy, (0.96) sensitivity, and (0.93) specificity [17].

Liu and others in the year 2019, the authors gathered pictures for their dataset by working with their university's medical department. They used a Deep Learning (DL) strategy. Their work consists of two stages. To begin, it uses deep learning to extract the pictures' discriminative characteristics. Second, it employs several instances of machine learning to build a classifier using highresolution pictures of AMD symptoms in unknown places. Results revealed total accuracy of (0.7), sensitivity of (0.67), and specificity of (0.7) [18].

Yan and others in 2020 conducted this research using the Age-Related Eye Disease Study (AREDS) dataset. They used Deep Learning (DL) technique. By combining genotypes and fundus pictures, a modified deep convolutional neural network is used to predict if an eye has advanced to late AMD. Their findings indicated that fundus pictures in combination with genotypes might accurately predict late AMD development with a mean AUC of (0.85). The mean area under the receiver operating characteristic curve for fundus pictures alone was (0.81) [19].

Floriano and others in 2019, in this study the authors used STARE dataset. They preprocessed the images by 1-mathematical morphology to detect the dilation and erosion. 2-thresholding used to distinguish between objects of interest and the background. After that, Support Vector Machine (SVM) model was trained to classify the images. Highest performance achieved on full dataset was Accuracy: (0.83), Precision: (0.841), Recall: (0.836), F-measure: (0.834) [20].

1.5 Problem Statement

In this section, the problem that would be tackled by this project will be described.

The problems could include that sometimes the previously proposed system is not able to recognize the features, due to noise and poor image quality. This is captured by the poor classification accuracy.

Morover the discrimination limitations proble. Which although features are expected to vary greatly between images, there may be significant similarities in this data's feature sets. Thus, each algorithm has a higher theoretical limit in terms of its ability to distinguish.

And ahe stages of the disease are similar in their appearance. Since there is a thin distinguish between the number and size of the drusen, especially in the Early and Intermediate stages.

Furthermore low number of publicly available dataset to train and test the model on.

1.6 Thesis Outline

The thesis consists of five chapters and their description is briefly given below. In addition to this chapter, the remaining chapters of this thesis are distributed among the following chapters:

In Chapter Two: Background Theory, an introduction about the Machine Learning and Deep Learning, their types and the difference between them, and the reasons behind selecting this approach are explained.

In Chapter Three: Proposed System, a thorough explanation of the Convolutional Neural Network (CNN), and the selected algorithm of the Deep Learning to tackle our problem are presented. This chapter contains all the principles of Convolutional Neural Networks, besides the related researches and published works and codes that may be useful and insightful for our understanding on how this technique is working. Moreover, the design of our work and algorithms are explained. In addition, all the different pathways (the two developed models) are examined during this project and different configuration values are discussed in details.

In Chapter Four: Results, the different results of all the examined pathways are displayed and examined in order to understand the importance and effectiveness of each of them. Furthermore, a comparison is made between them. Finally, the most promising result are promoted.

In Chapter Five: Conclusions, a conclusion of the work and the related results are discussed. Besides, further works to improve this work are suggested

CHAPTER TWO
THEORITICAL BACKGROUND

2.1 Introduction

Machine Learning (ML) field is about developing algorithms that learn automatically from datasets. These learned models will make predictions in the future based on what they learned during the training process [21].

In this chapter a deeper investigation about AMD, Machine Learning, Deep Learning, image pre-processing, and their algorithms.

.

2.2 Age-related Macular Degeneration

Age related Macular Degeneration (AMD) is an eye disease that can blur the sharp, central vision you need for activities like reading and driving. “Age-related” means that it often happens in older people. “Macular” means it affects a part of your eye called the macula. “Degeneration” means the type of damage happening to the eye. This disease effect about 8% of people globally [4].

Age-related Macular Degeneration (AMD) is eye condition that effect old people. In addition, it is one of the prime reasons for blindness or blurred vision for senior citizens.

AMD is a chronic eye condition that affects the central vision of the eye. It is due to the degeneration of the macula, the central part of the retina that promote clear and sharp vision.

In AMD there is a progressive accumulation of yellow deposits, called drusen (build-up of extracellular proteins and lipids), in the macula (a part of the retina). This accumulation is believed to damage the retina over time [22].

AMD can be categorized into three stages (early stage, intermediate stage, and late stage) based partially on the extent (size and number) of drusen.

The three stages of AMD can be seen as follows [5][23]:

- **Early stage:** medium drusen deposits size, and no change on retina layers, and no vision loss.
- **Intermediate stage:** large drusen deposits size, and/or change on retina layers and in rare cases a mild vision loss.
- **Late stage:** large drusen deposits size, and changes on retina layers, which leads to what known as Geographic Atrophy (GA). Which is the loss of some tissues in the layers of retina, because of the advance in drusens. This will may lead to vision loss. This stage could be developed further, that the large drusens will change the size of to the eye blood vessels abnormally, making them bleeding and leaking.

Figure 2.1 shows a normal eye. The diagnose could be explained judging there is no drusens, and the texture of the retina is clear, and the blood vessels are healthy with no damage.

Figure 2.2 shows an eye with AMD in early stage. Where some drusens are visible on the macula of the eye. However, they don’t form a hard and large clusters.

Figure 2.3 shows intermediate stage in AMD disease. Where drusens are visible. Larger number of them comparing with the early stage, and larger clusters.

While Figure 2.4 shows the late stage. Where the central drusen cluster are visible, besides one large spot of the drusen on the macula, and apparent damage to the retina.

In Figure 2.5 shows the late stage, with advanced damage. Where blood leaking from the vessels is visible. With large damage to the retina.

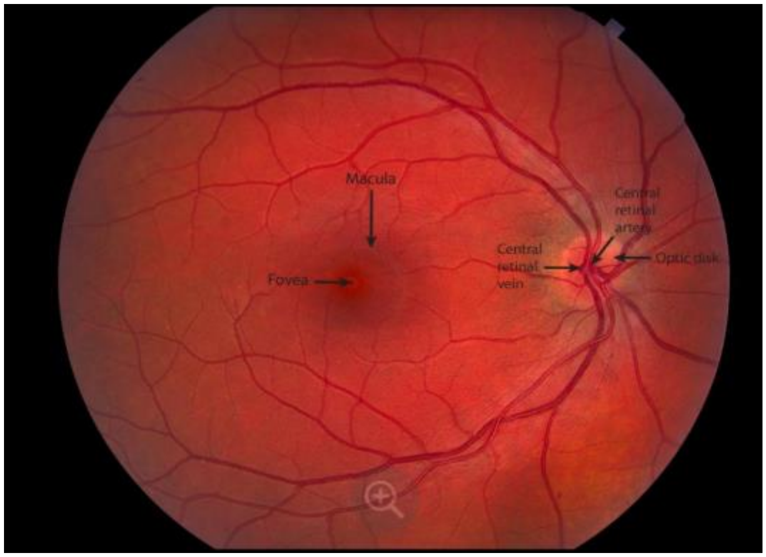


Figure (2.1): Normal retina [24]

Figure (2.2): AMD early stage [24]

Figure (2.3): AMD intermediate stage [24]

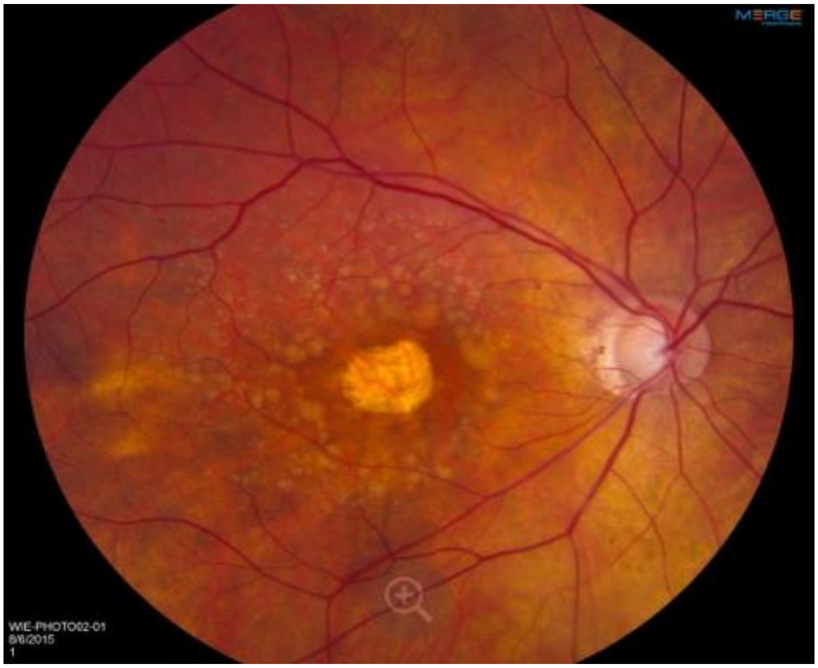


Figure (2.4): AMD – late stage [24]

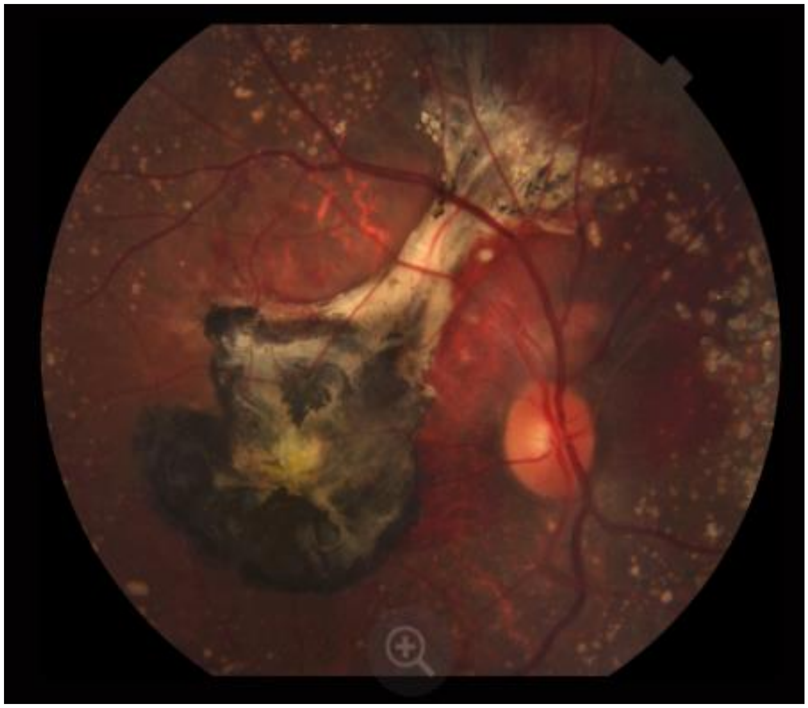


Figure (2.5): AMD – advanced late stage [24]

2.3 Machine Learning

It is a subfield of "Artificial Intelligence" that equips computers with the ability to execute skilled tasks via the use of intelligent software. Machine Learning methods have been used to a broad range of applications, including computer vision and email filtering, when it is impractical or impossible to enhance traditional algorithms to do the necessary job. The field of Machine Learning (ML) is about developing algorithms that learns automatically from datasets. These learned models will make predictions in the future based on what they learned during the training process. The idea behind these algorithms is that they create a statistical inference from the data they are learning from, and use this statistical model in order to predict future values. For example, assume we want to develop an algorithm to predict if there is storm coming in specific area, the data of the weather in that area for some time will be collected.

In addition, the Machine Learning algorithm will build a model to understand how the different variables and features of that data are combined together to get the right results. [25].

2.4 Machine Learning Approaches:

Although all the Machine Learning models share similar goals (i.e. predicting the right result for a specific problem), and although sufficient amounts of data are required to train the different types of Machine Learning problems, there are different approaches to getting this result Figure (2.6) shows the different types of Machine Learning approaches. The types of Machine Learning are [25]:

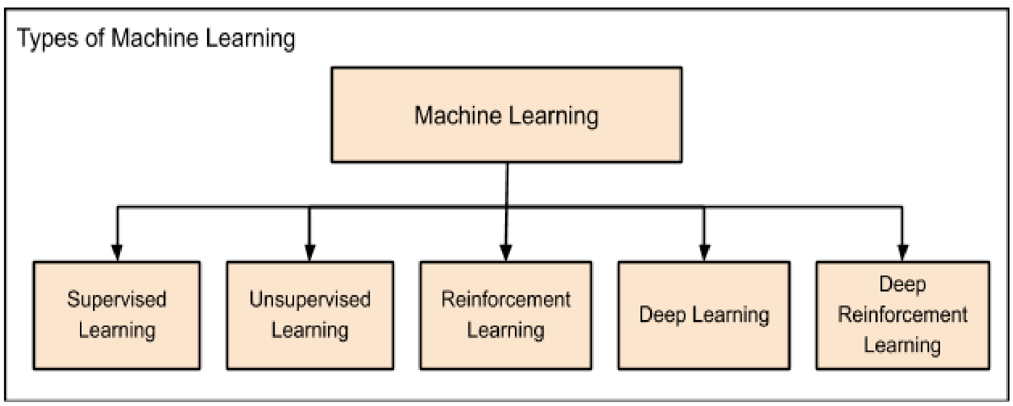


Figure (2.6). Types of Machine Learning [26]

A. Supervised Machine Learning

The data are labelled in supervised learning. This implies that the Machine Learning algorithm is trained on data with known input values (features or variables, denoted as x) and output values (target, written as y). The supervised algorithm's job is to match the inputs in a function to produce a precise output (y=f(x)). The objective is to achieve the greatest possible match between x and y, such that when a new value is input into the model, it produces the desired result. Supervised learning is one of the most popular learning methods. It may also be split into two categories: [25]

- **Classification:** in which the output or the target (y) is categorical. For example, if we want to know if the patient is sick or not, or whether or not it will rain a week later. Alternatively, if the customers prefer more red or blue colours.
- **Regression:** In regression, the target value (y) is numerical. It is difficult, for example, to estimate the value of the stock market. Or how much it will rain a week later. Or how many items the customer will buy.

Some common Machine Learning algorithms are Linear Regression, which is used for regression problems. Support Vector Machines are often utilized for categorization issues. In addition, Random Forest is used for classification and regression.

Notice that it is called supervised because the algorithm can be guided towards the correct answer during the training phase by using the target values (y) in the dataset.

B. Unsupervised Machine Learning

In this category, the data is not labelled. Therefore, the algorithms have only the input variable value (x), without the target value (y). And it will try to find the distribution and the underlying pattern of the data in order to build a model that gives better understanding and prediction of the data [27]. Unsupervised learning could be divided into:

- Clustering: is a type of algorithm that divides data into distinct groups. This division of different groups is based on some inner properties that the data has. For example, the clustering of website users is based on their different behaviour.

- Association: is the type of algorithm where it tries to connect or relate some values to another value. The ecommerce recommendation engine is an example of this; when a user purchases one item, it will recommend another item that is typically purchased together.

C. Semi-Supervised Machine Learning:

This approach is when some of the data is labelled. So, for some of the data, the target value (y) is known, while for the other parts of the data, the target value is unknown [28].

D. Reinforcement Learning:

This approach is based on rewards. The algorithm will get the inputs (x), and perform the function, and it will return an output. This output will include a value (award) that indicates how good the model was at its previous behaviour [29]. Generally, there are two types of awards:

Positive: occurs when the algorithm's behaviour toward the input results in stronger behaviour.

Negative: this happens when the algorithm behaviour the input will result in unwanted sequence.

2.5 Deep Learning

In the classical Machine Learning approach, the model receives inputs from predetermined features. Meaning, only the important and relevant features are selected or crafted. This is done manually by the programmer or the engineer.

However, this does not always lead to the best results. Since there are many phases with different approaches to deal with them. Moreover, manual feature selection and engineering do not always lead to the best results. In Deep Learning, this problem is solved by doing the feature extraction automatically by using deep layers to do the feature extraction [30].

2.5.1 The Techniques of Deep Learning:

It is a subcategory of Machine Learning that utilizes hierarchical structures to allow the computer to learn abstract concepts from data. . It is a developing approach and has been commonly implemented in conventional artificial intelligence domains, like transfer learning, Natural Language Processing, Semantic Parsing, Computer Vision and others. here are three main reasons for the growth of deep learning today: firstly, greatly increased chip processing capabilities (e.g. GPU units), secondly, the lower cost of computer hardware and, finally, the rapid advancement of machine learning algorithms [31].

2.5.2 Deep Neural Networks (DNN)

A network such as DNN is essentially an ANN with several hidden layers. MLP is among the most often utilized ANN architectures to build DNN. Since neural networks are defined as shapes, they may be considered figures. A machine-learning neural network is diagrammed in an abstract manner. The DNN, due to the high number of weights included in the network, needs very long computing times and substantial amounts of data input to training stages [32]. In the past decade, many new variables have developed, enabling us to teach this kind:

- The exponential growth in computer and GPU processing power.

- Big data analysis is facilitated by new data-oriented cultures, such as data mining and machine learning, which have developed in the past decade.

- New DNN training techniques emerge like "fast learning algorithm, ReLU layers, and dropout regularization". [33] [34]

2.5.3 Convolutional Neural Networks (CNN)

Some Deep Neural Networks (DNN) researchers believe that Convolutional Neural Networks (CNNs) are one of the most often used computer vision models in DNNs, which are also built using multilayer perceptron's (MLP) and back propagation methods. Traditional MLPs may perform feature extraction followed by classification, while this new model combines a number of locally linked layers for feature extraction and one large interconnected layer for classification. Multi-stage training Neural Network Architectures are known as convolutional neural networks. As each of these phases has the kinds of layers mentioned below, they are all included in each of these stages [35].

This type will be elaborated in the next section, since it is the cornerstone of this approach. However, notice that the most important of the CNN is the convolution operator (Figure 2.7), where the features are learned through these kernels. [36]

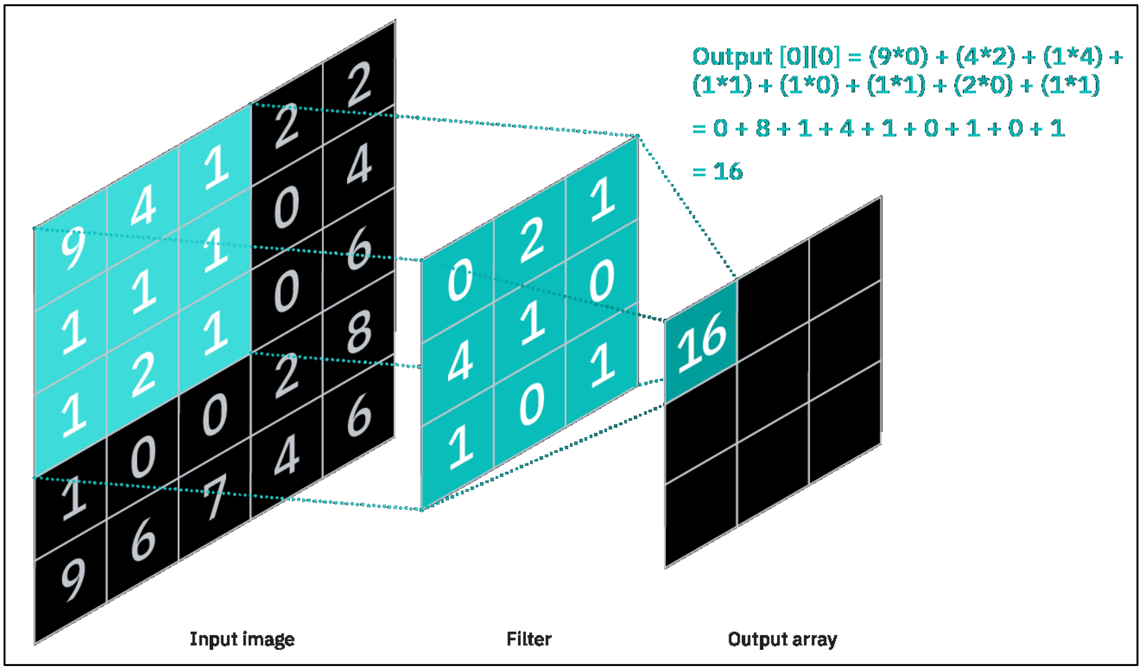


Figure (2.7): Sample of convolution layer operation [37]

2.5.4 Training Algorithms

Typically, while trying to reduce a deep learning problem's loss, we first set up a loss function and then use an optimization method in order to minimize it. A loss function is often defined as “the objective function of an optimization problem”.

2.5.4.1 Loss function

Sometimes called (objective function) or (cost function) or (error function). Since the general idea of Machine Learning and Deep Learning is to develop a method to set the values of the input dataset into correct output. This goal is achieved mainly by configuring the suitable weights of the Artificial Neural Network for this purpose. However, in practice, calculating the exact right value for each weight is unachievable. Since there are many unknowns values. So instead, the problem of finding the right values for the weights is set as optimization problem. In this sense, the algorithm will search the possible sets of weights that will lead to the expected and satisfying results. These expected results are recorded as lowest error value as possible. So all the aspect of the model performance should be turned into single value by the loss function. By using this single value, the amount of changes to the weights are derived by the optimization function. [38]

There are many functions that are available as loss function. Each one of them works in specific way to achieve specific goals. Generally, the loss functions are divided based on the type of problems they are working on. So there are functions suitable to regression problems, others are suitable for binary classification than regression problems.

A. **Regression Loss functions:** these types of loss functions are suitable for regression problems. Since regression is concerned in predicting numeric values. The most prominent and widely used loss functions in regression problems are:

1. **Mean Squared Error Loss:** or usually (MSE) is the default function used in regression problems. The value of it is calculated by subtracting the values of expected results and the real results get by the training observations. After that, these values are squared (to eliminate the negative value impact). Then calculating the average of these values. Therefore, the value will be always positive, since squaring negative values will result in positive result. Additionally, the squaring operation will means larger the error get, larger the model is pushed to modify its weights. And the perfect value that will not make any change will be zero (because the different between the expected value and what we get is the same, in other words our model get us exactly what we want) [39]. The mathematical equation for the mean square error is as follow:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad \ldots(2.1)\ [39]$$

2. **Mean Squared Logarithmic Error Loss:** or generally (MSLE) in this method, the natural logarithm value of both the predicted vale and the expected value are calculated. Then the same approach for the Mean Square Error is used. By subtracting both values and calculate the average of them. This method is necessary when there are a wide variance in the values of the target. So we don't want the model to make large changes when there are large differences between the expected and calculated values [40]. The equation for the Mean Squared Logarithmic Error is as fellow:

$$MSLE = \frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\widehat{y_i} + 1))^2 \qquad \ldots(2.2)\ [40]$$

Notice that there are one is added to each of the expected and predicted values. This is because its mathematically impossible to have a logarithm value for zero. In addition, the logarithm value for one is zero.

3. **Mean Absolute Error Loss:** or (MAE). In this approach, similar steps are taken as in Mean Square Error. But instead of using the square value for difference between the predicted values and expected values, an absolute value is taken instead. This method is preferable when there are many outliers in the target values (many target values are far from the mean value) [41]. The equation of this approach is as fellow:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}| \qquad \ldots(2.3)\ [41]$$

B. **Binary Classification Loss Functions**: in these types of problems, where there are two classes available for the target value (usually either 0 or 1). There are specific type of effective loss function for these types of problems.

1. **Binary Cross-Entropy:** this method is the default and most efficient methods used when the problem involves binary classification. It assume

and works when the target values are either 0 or 1. The value it calculates summarizes the average difference between the expected value probability distribution, and the calculated value probability distribution [42]. The optimal value for this method is zero.

$$h(x) = -\log(P(x)) \quad \ldots(2.4)\ [42]$$

$$h(X, y) = \begin{cases} -\log P(Y = 1|X) & if\ y = 1 \\ -\log P(Y = 0|X) & if\ y = 0 \end{cases} \quad \ldots(2.5)\ [42]$$

C. Multi-class Classification Loss Functions: these types of loss functions works with classification problems of two or more classes. The classes are usually labelled as integer numbers starting from zero. And the problem can be phrased as predicting the probability of each class. Here are some methods of calculating the loss function in multi-class classification [43].

1. **Multi-Class Cross-Entropy Loss:** this approach is the default and commonly used in the problems with multi-class classification. It is used when there are two or more classes. The labels of the classes are set as integers starting from zero. In this approach a value is calculated to represents the average different between the expected output and the actual output [44].

2. **Sparse Multiclass Cross-Entropy Loss:** in the case when there are very large number of labels to classify (for example, in the problems of prediction words in Natural Language Processing in vocabulary have hundreds or thousands categories) that the model should predict, there will be need for high memory and computation power. Notice that in Machine Learning, each label of the dataset are stored as One hot Encoding. So this means each label is a vector with hundreds or thousands length, all of them are zeros, only one is one. Sparse Multiclass Cross-Entropy Loss will address this problem by using the same cross-entropy operation to calculate the error, but

instead of using One Hot Encoding to represent the label, it will use an integer number to replace that representation [45] [43].

$$CCE(p,t) = -\sum_{c=1}^{C} t_{o,c} \log(p_o, c) \qquad \ldots(2.6)\ [45]$$

3. **Kullback Leibler Divergence Loss**: also called (relative entropy) or (KL Divergence). This method measures how different is the probability distribution from another probability distribution. If the difference between the two distributions is zero (i.e. the probability distributions are identical), this means there is no changes required, and the output meets the expected values. It is somehow similar to the previous Cross-Entropy method. In its main motivation, it calculates how far is the current distribution from the expected distribution [46].

$$KLD(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \qquad \ldots(2.7)\ [46]$$

2.5.4.2 The Activation Functions

This function is a fundamental component of an artificial neuron, and it can be nonlinear or linear in nature, displaying the degree of neuron activation. The normal range is (-1,1) or (0,1). The sigmoid function, which is the usual activation function for the calculation of probability using the equation formula (2.2), is the usual choice for the activation function in hidden layers.

$$g(z) = \frac{1}{1+e^{-z}} \qquad \ldots(2.8)$$

Another popular activation function is Rectified Linear Unit (ReLU): is a piece-wise linear activation function, the following equation represents its formula (2.3)

$$g(z) = \max(z, 0) \qquad \ldots(2.9)$$

The output layer must make educated guesses about the likelihood of different classifications. As a result, the SoftMax activation function is the most frequently employed activation function in the output layer, as shown in the equation (2.4).

$$\sigma(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \qquad \ldots(2.10)$$

Where, σ is the softmax function, $\vec{z}$ is the input vector, e^{z_i} is the standard exponential function for input vector, k is the number of classes in the multi-class classifier, e^{z_j} is the standard exponential function for output vector.

The activation function of the Rectified Linear Unit (ReLU) has two advantages: The first is that the ReLU function is simple to measure, and this criterion provides an advantage in terms of computational speed. The second is that the ReLU gradient is constant, which means that for any numerical input, the ReLU output value is zero or in the positive area [47].

2.5.4.3 Optimizer

Optimization in its general term refers to developing a procedure that will minimize or maximize the output of function, giving it the same input. By modifying and tweaking some changeable variables. This is tricky task since there are large and high dimensions space to seek the optimal values for these variables to minimize or maximize the output of the function. In practice, the variables that the optimizer controls are the weights and biases of the network. There are many different types of optimizers, each one have its advantages and disadvantages [48]. The most popular ones are:

1. **Batch Gradient Descent**: it is the most simple and widely used optimizer in machine learning. Mainly used in regression and classification problems. It uses the first order derivative of the loss function to detect the direction of the weights to be modified (increase or decrease the weight) so the

model will reach the minimum loss value. The way it works is that it's update the parameters after one epoch. In other words, it goes through all the observations to calculate the error, and after that it will update the parameters. The advantages of this optimizer is in its simplicity. So it is easy to implement and understand, and relatively computational efficient. Since the changes are happening only after a whole epoch is done. Moreover, it gives smother convergence than other gradient descent family of optimizers. However, it has the potential to stuck at local minima or saddle point, and having slower learning since it apply the changes only after whole epoch [48]. Figure (2.8) shows how the Batch Gradient Descent behave.

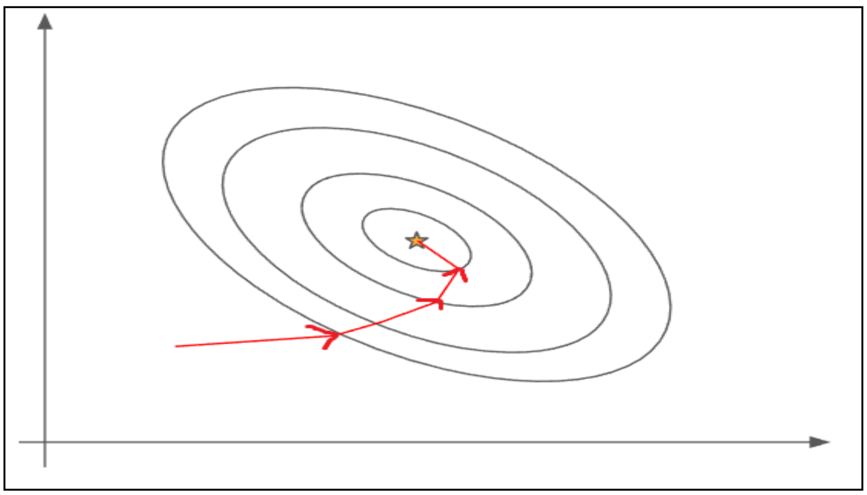


Figure (2.8): Batch Gradient Descent [48]

2. Stochastic Gradient Descent: or usually referred to as (SGD). Unlike Batch Gradient Descent, the Stochastic Gradient Descent consider only one observation before applying the changes in each time. Therefore, instead of going through all the dataset before calculating how to change the parameters, it will do the calculations after each observation is selected. This will lead to lower memory required since only one observation is selected to calculate how the changes will be, and frequent updates will make the optimizer less likely to stuck at local minima or saddle point. However, this frequent changes can lead it away from the global minima, and frequent changes are computationally expensive [48] [49]. Figure (2.9) shows the behaviour of SGD.

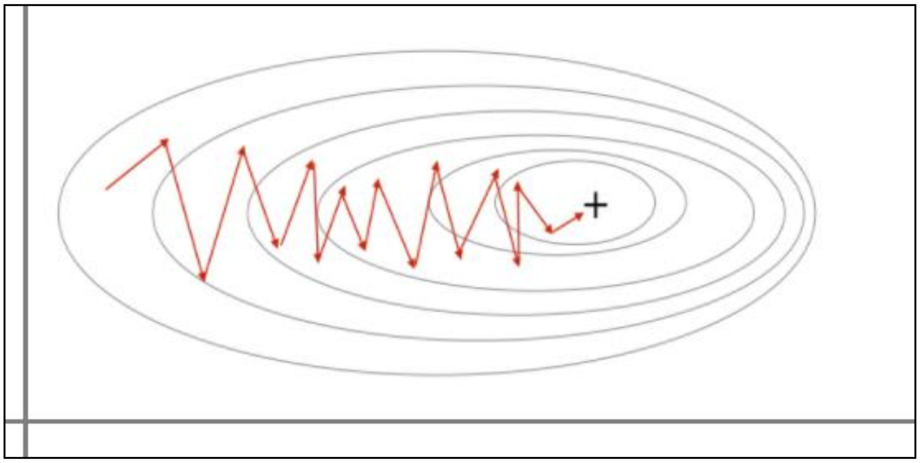


Figure (2.9): Stochastic Gradient Descent

3. **Mini-Batch Gradient Descent**: it is similar to the Batch Gradient Descent and Stochastic Gradient Descent. However, instead of updating the parameters after a whole epoch, it will update after running on a part of the observations in the dataset. So the iteration will be on a pre-define number of the observations and not all of them. This gave the Mini-Batch Gradient Descent a faster learning, since it will not require going through all the dataset in order to update the parameters. Still computationally efficient. However, it require to define the Mini-Batch size hyper parameter (the number of observations to consider in each iteration) [48] [50]. Its behaviour is showed in Figure (2.10).

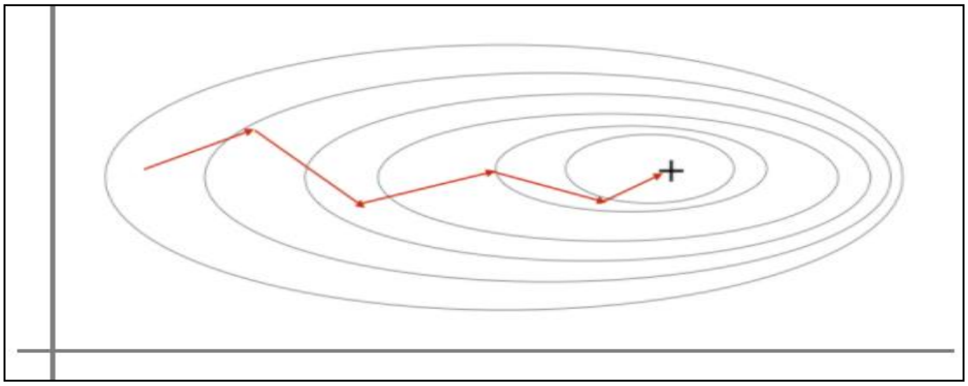


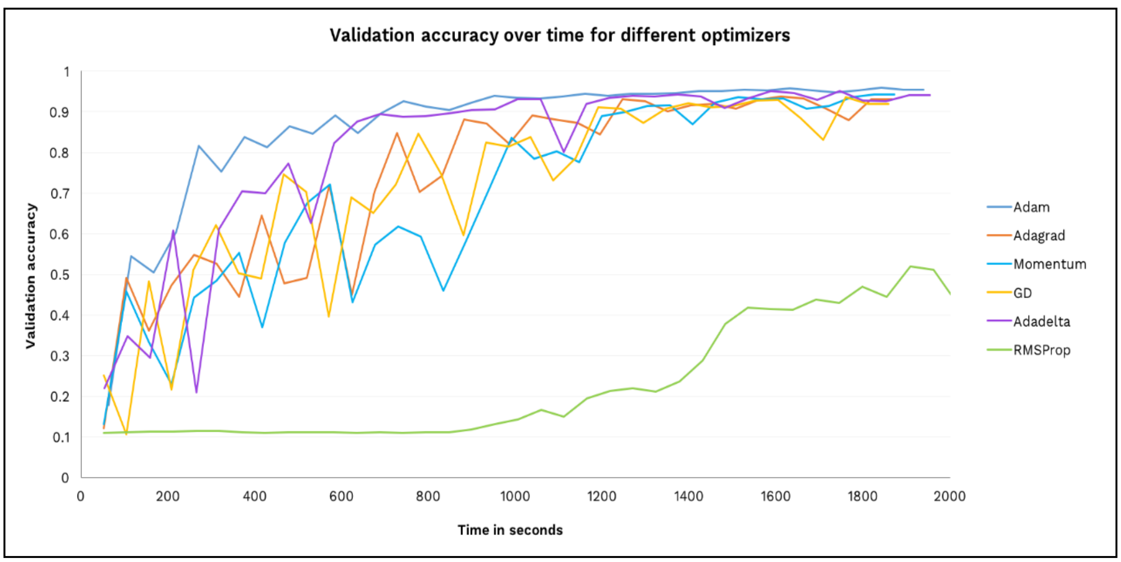
Figure (2.10): Mini-Batch Gradient Descent

4. Adam: Or called also (Adaptive Moment Estimation). It is so far the most prominent and popular optimizer that giving better results than others optimizers. It is a first order gradient, which is based on the stochastic cost function. The motivation behind Adam is to avoid missing the minima. Therefore, the optimizer velocity is reduced when it is required, while searching so it will not miss the minima. However, this reduction is not all the time of the search process, so it will not take too long to find the minima. This is achieved by keeping the squared decaying average values of the previous gradient like AdaDelta and RMSprop. Moreover, Adam holds the previous decaying average of past gradients. This leads to the advantage of speed convergence, and fix the problem of vanishing learning rate. However, it is computationally expensive [51] [52]. Algorithm 2.1 shows how Adam algorithm works.

Algorithm (2.1): Adam algorithm	
Input:	α: stepsize $\beta_1, \beta_2 \in [0,1)$: Exponational decay rates for the moment estimate $f(\theta)$: stochastic objective function with parameter θ θ_0: initial parameter vector
Output: optimized model parameters	

Begin
Step1: $m_0 = 0$ (initialize 1st moment vector)
Step2: $v_0 = 0$ (initialize 2nd moment vector)
Step3: $t = 0$ (initialize timestep)
Step4: while θ_t not converged **do**:
Step5: $t = t + 1$
Step6: $g_t = \nabla_\theta f_t(\theta_{t-1})$
Step7: $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
Step8: $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
Step9: $\widehat{m}_t = \frac{m_t}{(1-\beta_1^t)}$
Step10: $\hat{v}_t = \frac{v_t}{(1-\beta_2^t)}$
Step11: $\theta_t = \theta_{t-1} - \alpha \cdot \frac{\widehat{m}_t}{(\sqrt{\hat{v}_t}+\epsilon)}$ (update parameter)
Step12: End while
End.

Figure (2.11) shows a general comparison between different optimizers in their performance. Revealing that Adam is the best one of them. Which will be selected in this project.



Figure(2.11): Comparison of different optimization methods [53]

2.5.5 Regularization of Neural Networks

There are many techniques that have been suggested during the training process to prevent over-fitting. It is called regularization.

1- **Overfitting Problem:** Deep neural networks have a high capacity to represent learning. Lack of control over the learning process in deep neural networks can cause over-fitting problems. Over-fitting means that models have a poor generalization capability. That results in the poor predictability of test data, although a high output of training or validation data is achieved by the model. It happens when there is a large gap between the errors in training and the errors in testing. The overfitting problem is depicted in the Figure (2.12). This state suggests that the deep neural network model is equipped to have a strong ability to match the training data rather than know the data patterns [54].

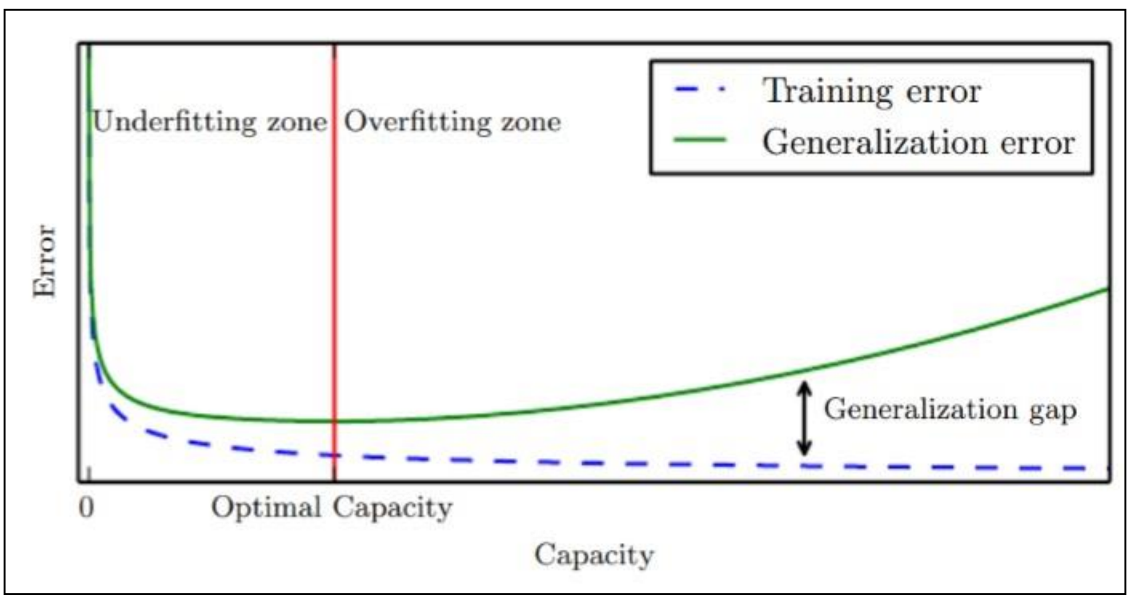


Figure (2.12) Overfitting in deep neural networks [55]

There are plenty of regularization techniques available to improve deep generalization capability.

2- **Dropout:** It is a regularization method for DNN models to deal with the issue of overfitting. The Dropout method is known to be a way of partially combining several different neural networks in an efficient way. It is clarified to drop out the neurons in the hidden layers of the neural network. Dropping out of neural neurons means that neurons and their related incoming and outgoing neural network connections are immediately lost. The Figure (2.13) depicts the concept of a dropout's life. When the dropout technique is applied to the multi-layer perceptron in the hidden layers, there is a likelihood that the neurons will be skipped in the hidden layer [56][57].

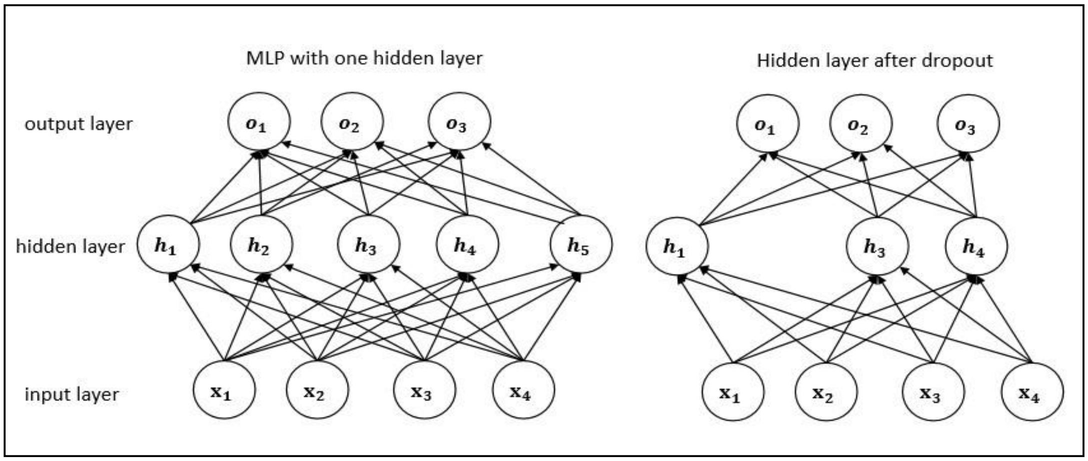
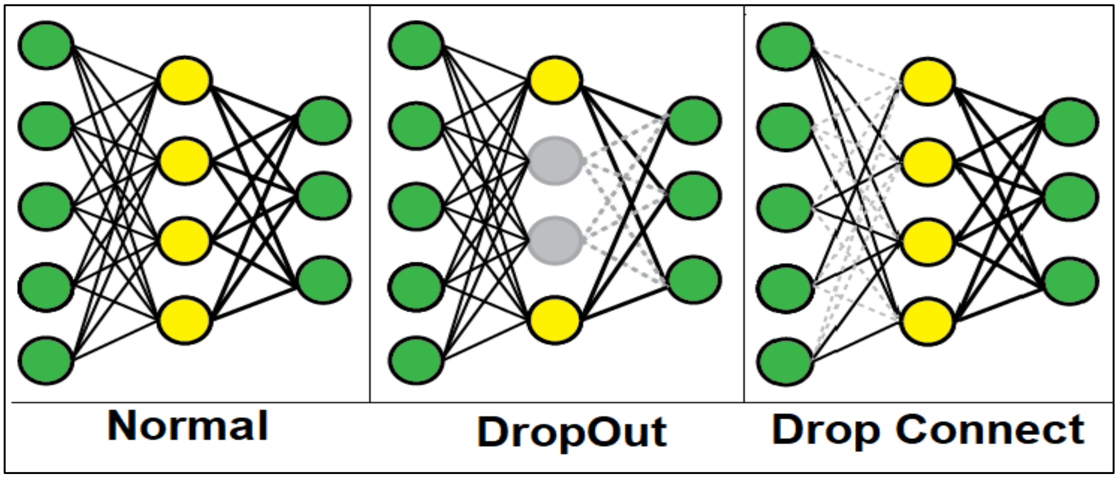


Figure (2.13) Comparison of MLP with and without dropout method [58]

3- **Drop Connect:** is a dropout generalization for the regularization of large, completely connected layers within neural networks that groups arbitrarily nominated weights to zero during preparation, data argumentation that operates the input data, and early iteration stoppage [59].

Figure (2.14) shows a comparison between the ANN in the normal, Dropout, and Drop Connect statuses.



Figure(2.14): Comparison between Normal, Dropout, and Drop Connect in ANN

4- **Early Stopping:** When CNN network is large, and the size of the training dataset is large too, there are a high opportunity of overfitting. Which means that the model will stop generalizing, and be limited to the training dataset, and will not perform well on the test dataset, neither in real life. So the goal is to train the model long enough to capture the general pattern of the dataset, without going through the problem of overfitting. Therefore, the idea of early stopping is to stop the training process before reaching the point when the problem of overfitting start to happen. This is done by validating the model performance during training phase with the validation dataset. And the training process will stops when the accuracy of the model will decrease against the validation dataset [60]. Figure (2.15) shows when the early stopping happen.

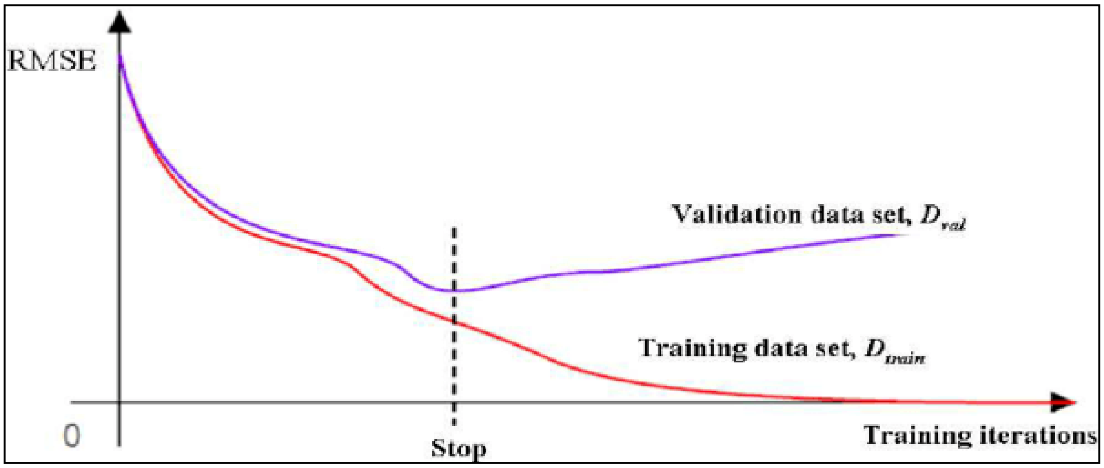


Figure (2.15): Early Stopping [61]

2.6 Convolutional Neural Network Concepts

Convolutional Neural Network (CNN) or Convolutional Network is proposed by LeCunn in 1989 [62]. Is a special kind of Artificial Neural Network that is specialized in processing grid structure data (such as images). CNN performance is considered successful when it's deployed in real applications. From its name, it is based mainly on the mathematical operation "convolution". It's main success key in its ability to combine a number of locally connected layers used for the extraction of features and followed by a number of completely interconnected layers used for classification. CNN are known as multi-stage training Neural Networks architectures well designed for Classification tasks. The early layers are responsible for extracting simple features (such as edges). The deeper the layers are, more complex and sophisticated the extracted features are. The deeper layers combine the output of the previous layers to produce complex and sophisticated features. Figure (2.16) shows that deeper the layers, more sophisticated the features are.

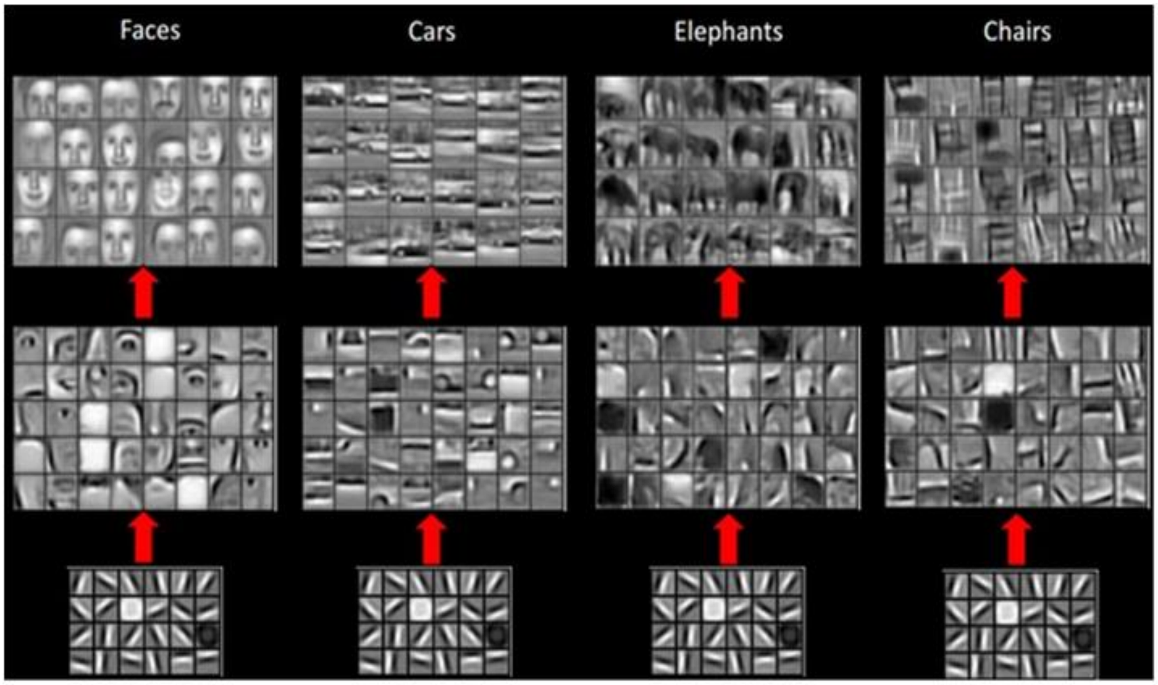


Figure (2.16): Illustration showing that deeper the layers, more sophisticated the features extracted are [63]

2.6.1 CNN layers:

In CNN, there are different types of layers. Each layer work in a specific way, for a specific task. These layers are:

A.Convolution layer: Is the essential layer to build a CNN model. It includes a number of kernel matrices that implement a convolution on the input and generate an output matrix of features (i.e. Feature map). Features in the convolution window are learned through the Convolution Operation. Additionally, the shared parameter method is used to decrease the number of parameters when the convolution is working [64]. Figure (2.17) shows convolution operation.

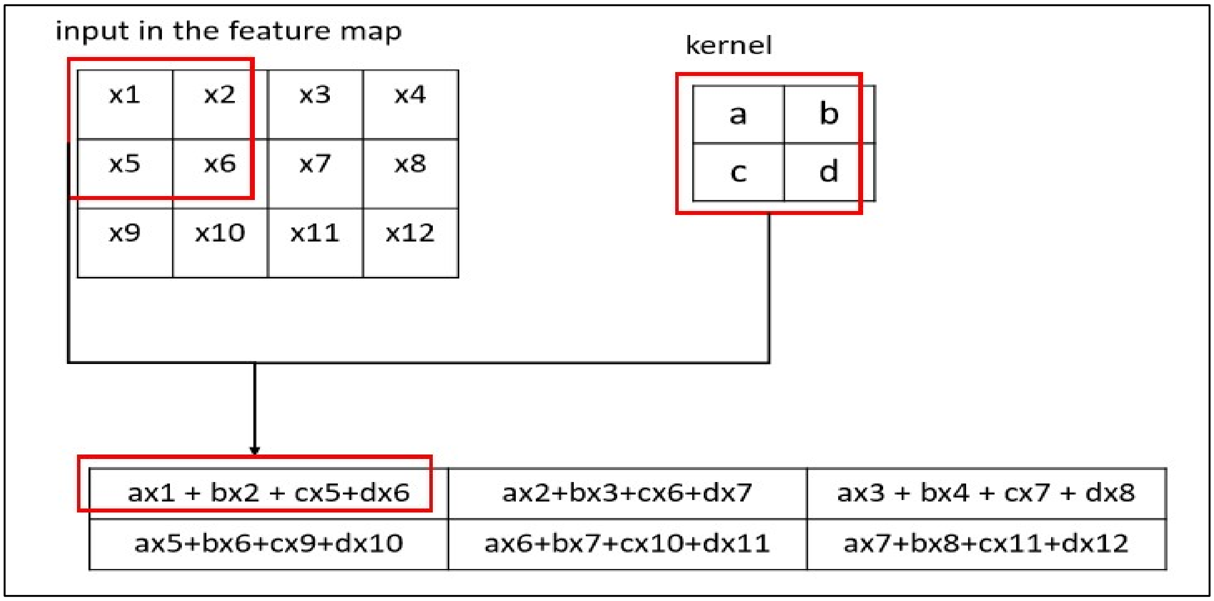


Figure (2.17): Convolution operation

$$a_{ij} = \sigma((W * X)_{ij} + b)$$ ...(2.11) [65]

Equation (2.11) shows how the equation of convolution operation. Where, X is the input provided to the layer, W is filter or kernel which slides over input, b is the bias, * representing the convolution operation, and σ is non-linearity introduced in the network.

B. Pooling layer: It is a fundamental layer to the CNN. Performance of dimensionality reduction of the input feature images is the purpose of a pooling layer. Pooling layers perform subsampling to convolutional layer outputs that linked with neighboring features. The average pooling and max pooling operations are two common operations. The average pooling operation uses all values within the convolution window and calculates the average value out of the window, meaning that the operation takes into account all values. While, in max pooling operation the largest value in each convolutional window is chosen [66]. Figure (2.18) shows the pooling operation.

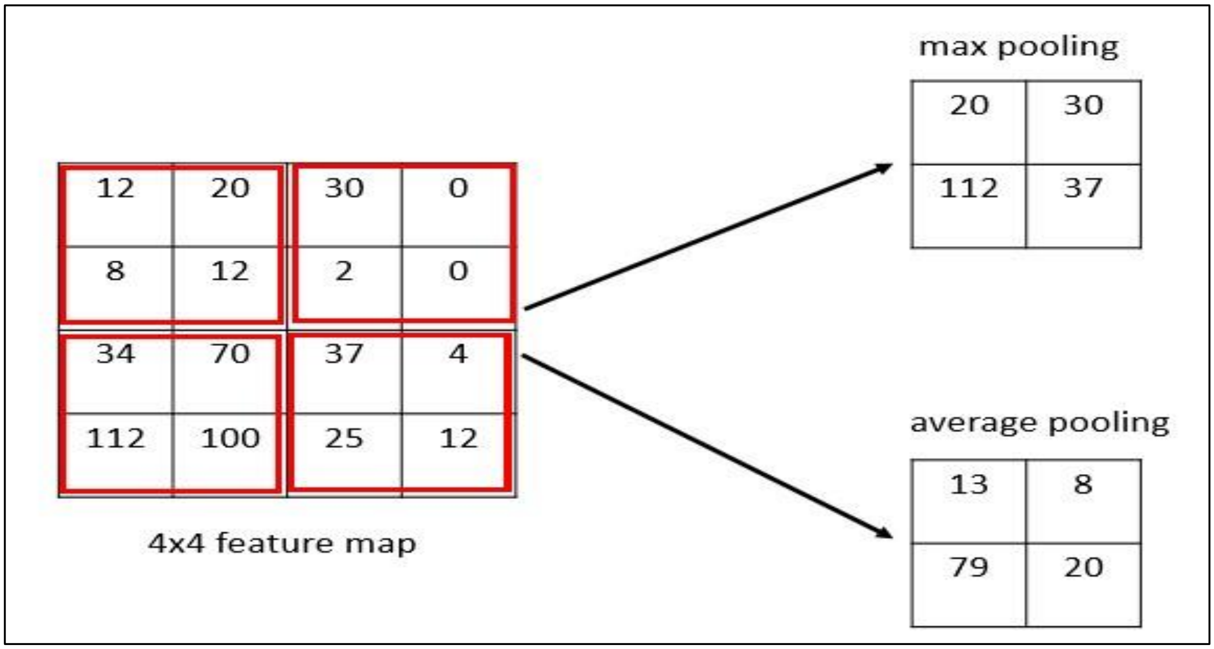


Figure (2.18): Pooling operation

C. Flatten Layer: The produced feature maps are normally flattened, meaning that is changed into a One-dimensional (1D) array of vector (or numbers), and connected to the one or more entirely connected layers (normal Artificial Neural Network). The main motivation of Flatten layer is to make is suitable for the coming signal from the previous layers of feature extraction to be sent to the input layer of the classification part of CNN [67]. Figure (2.19) shows how the flatten operation is carried out.

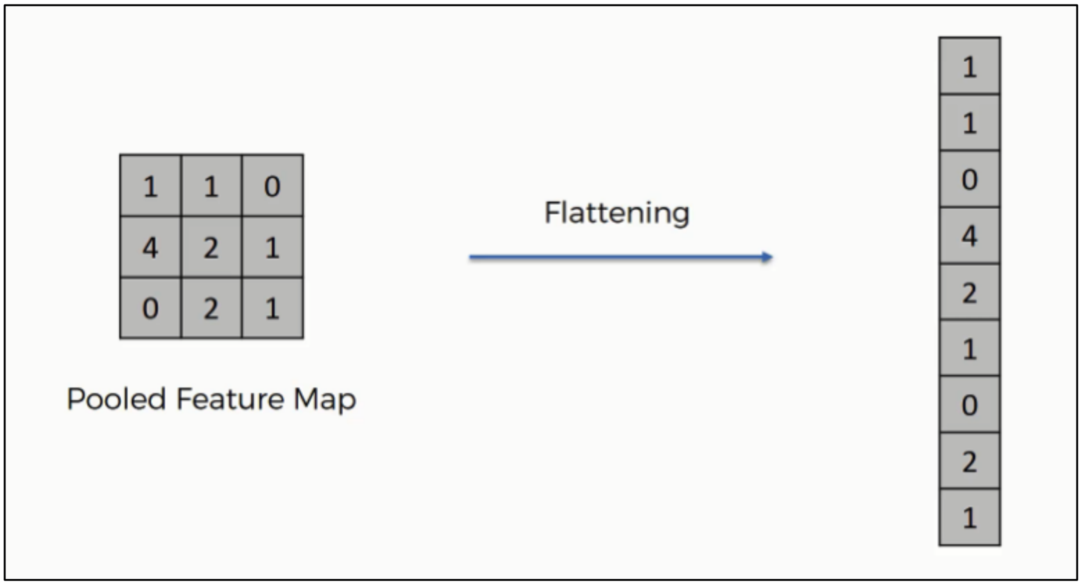
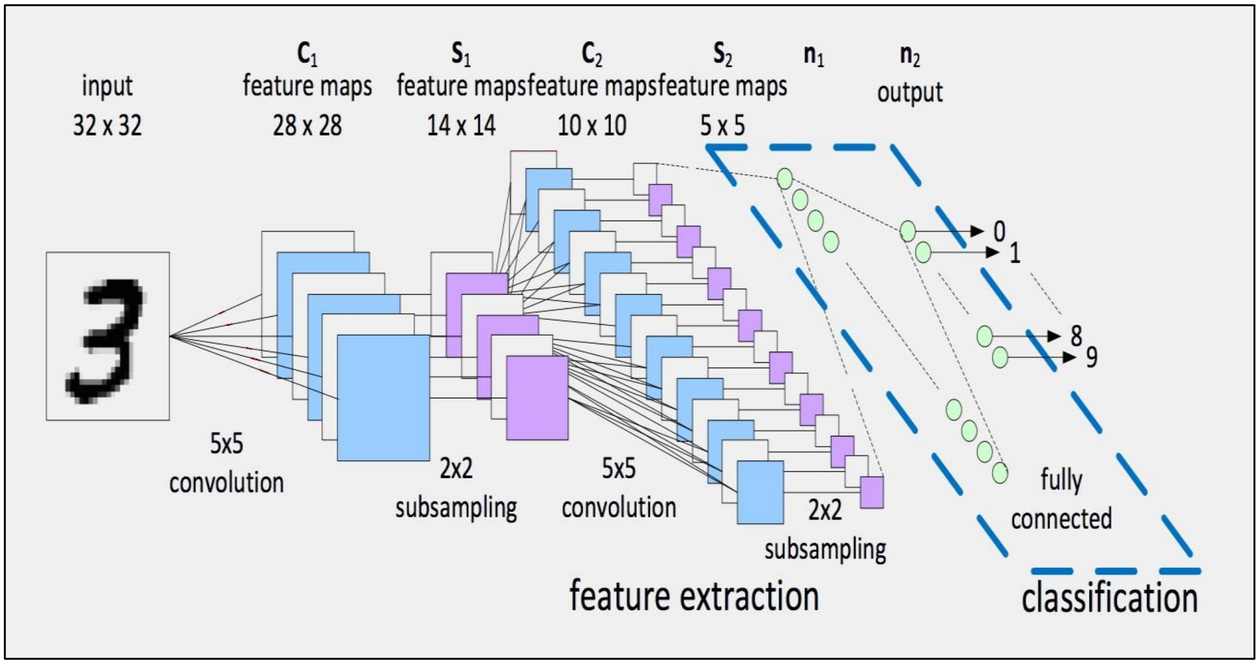


Figure (2.19): Flatten operation

D.Dropout Layer: Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel. Dropout has the effect of making the training process noisy, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs. This conceptualization suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model more robust. This will reduce the problem of overfitting in CNN [56].

E. Dense Layer: Or fully connected layer. In this layer, each input is connected to each output through a learnable weight. It's the classic Feed Forward Neural Network (FNN) hidden layer. Which is described in details in the previous section (ANN). At Once the features extracted via the convolution layers and down sampled by the pooling layers are performed. The last fully connected layer usually has same number of output nodes as the number of classes [68]. Figure (2.20) shows general hierarchy of CNN.

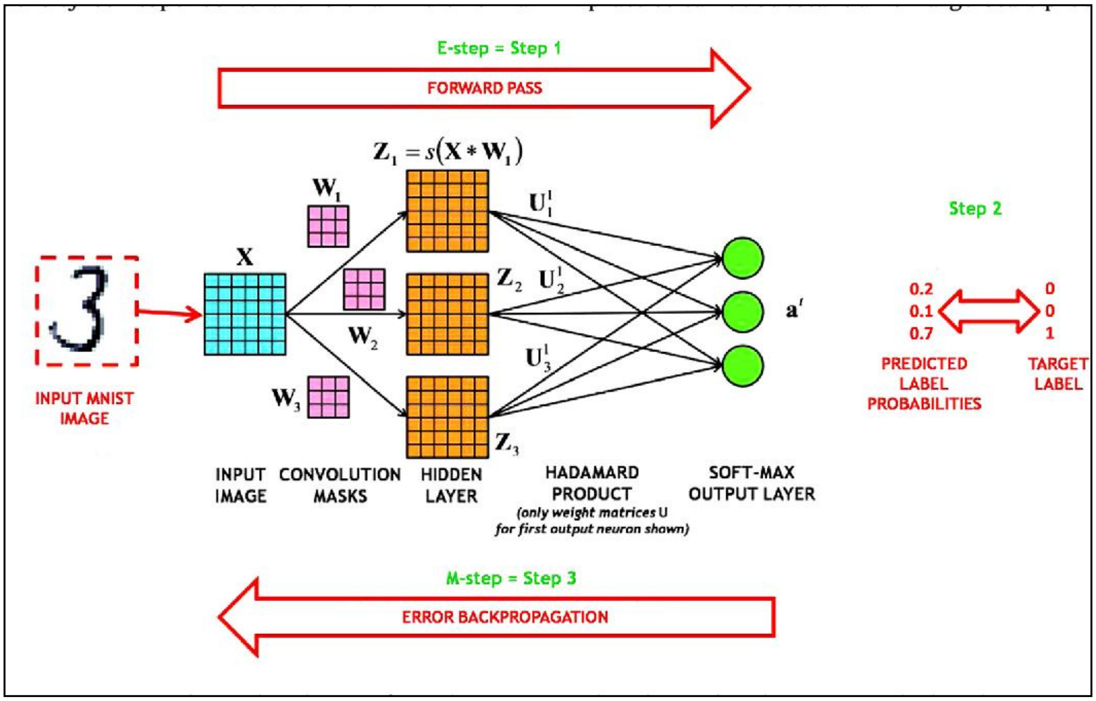


Figure(2.20): General hierarchy of CNN [69]

2.6.2 Training the CNN

Back-propagation has two stages: the forward and backward phases. Back-propagation algorithm of MLP is explained in section (2.3.2). The issue is more complex at the other levels of CNN.

In computing the outcome of an operation, the forward propagation computes the intermediate results in the memory before computing the final result. While the chain rule is used in the backward direction to calculate the gradient of the loss function with respect to the inputs, it is a one-sided chain rule that does not utilize the outputs.



Figure(2.21): Training the CNN [70]

This illustration in figure (2.21) shows the difference between feed forward and feedback. The goal is to discover how the gradient in a convolutional layer propagates backward. The objective of backpropagation is to discover the db, dx, and dw using the $\frac{dL}{dZ}$ and administer the gold rules chain.

The definition of the forward pass goes like this: It is possible to receive a string of n data points using c channels, each of which has a height of h and a width of W. N distinct filters are applied to each input, each of which covers all c channels and is tall and wide.

Algorithm 2.1 shows how the backpropogation works.

Algorithm(2.2): Backpropogation [71]	
Input:	*X*: training dataset, with size *m***n* *y*: labels of the observations (rows) in *X* *w*: weights of the respective layers

l: number of layers in neural network
Output: trained model
Begin **Step1:** calculate the error $D_{ij}^{(l)}$ **Step2:** for all *l*,*i*,*j* set $t_{ij}^{(l)} = 0$ **Step3:** for *i*=1:*m* **Step4:** $a^l = feedforward(x^{(i)}, w)$ **Step5:** $d^l = a(L) - y(i)$ **Step6:** $t_{ij}^{(l)} = t_{ij}^{(l)} + a_j^{(l)} \cdot t_i^{l+1}$ **Step7:** end for **Step8:** if $j \neq 0$ **Step9:** $D_{ij}^{(l)} = \frac{1}{m} t_{ij}^{(l)} + \lambda w_{ij}^{(l)}$ **Step10:** else **Step11:** $D_{ij}^{(l)} = \frac{1}{m} t_{ij}^{(l)}$ where $\frac{\vartheta}{\vartheta w_{ij}^{(l)}} J(w) = D_{ij}^{(l)}$ **End.**

Figure (2.22) shows the difference between the Forward pass and Backward pass mathematical operation on the level of the neuron.

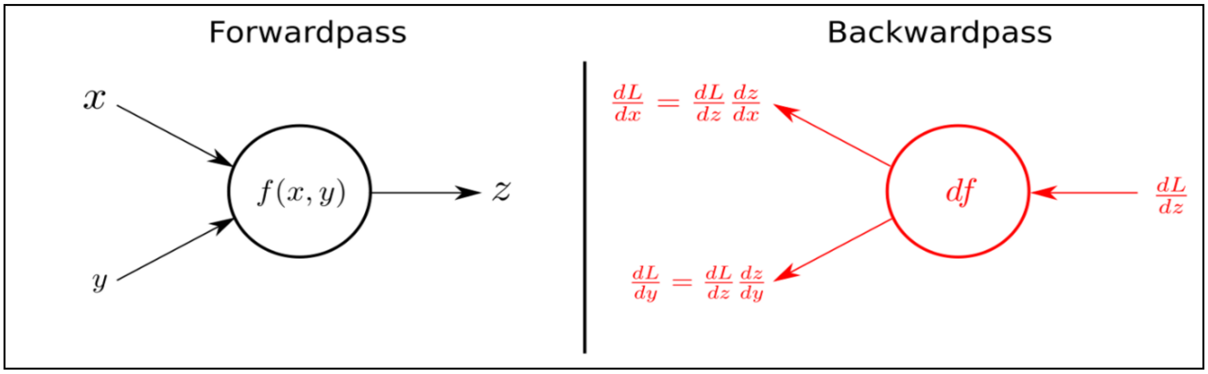


Figure (2.22): Forward pass and Backward pass in artificial neuron

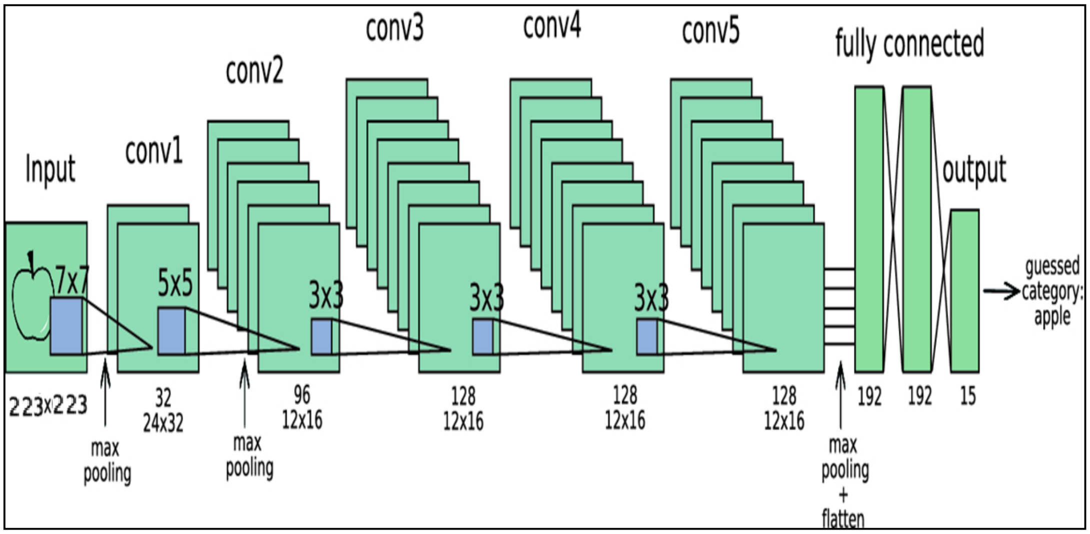
2.6.3 Famous CNN Networks

A. AlexNet: ImageNet contest is a contest where there are more than 15 million images in over 22 thousand categories. The goal of the contest is to build a program that is able to classify the images correctly. In 2012 [72], Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton developed what is known as AlexNet. Their proposed design includes eight layers in total. Five of them are convolutional layers, and three are fully connected layers. The new approaches used in this network, which made them competitive, are:

-ReLU: "Rectified Linear Units (ReLU)" are used as an "Activation Function" instead of a normal function. When compared to CNN using the tanh, ReLU reduced training time by six times.

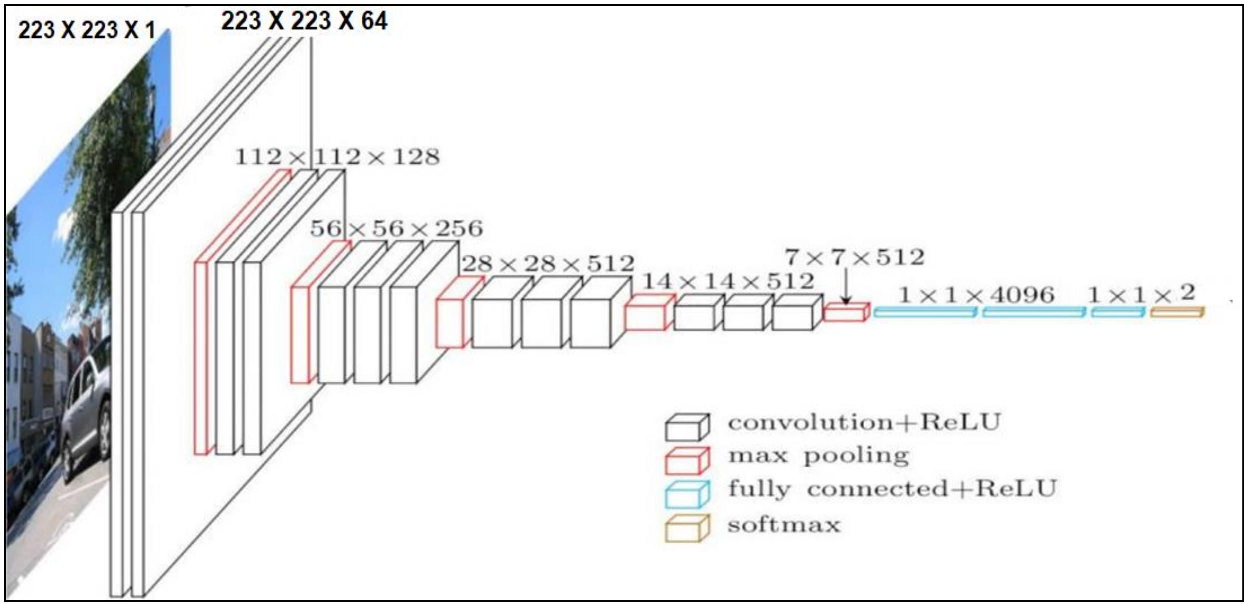
-Overlapping Pooling: The traditional method of pooling is used as a single pooling operation on each layer. However, in AlexNet, the concept of overlapping pooling is introduced. This led to a 0.5% reduction in the error rate. And made the model harder to fit.

Figure (2.23) shows a general design of AlexNet.



Figure(2.23): AlexNet design [73]

A. VGG: The VGG was introduced by K. Simonyan and A. Zisserman of the University of Oxford [74]. According to ImageNet, their model achieves an accuracy of 0.927, putting it among the top five most accurate models. It outperforms AlexNet by progressively substituting large kernel-sized filters. Figure (2.24) shows the general design of the VGG16. However, there are many configurations of the VGG network that results in slightly different design that is suitable for specific problems.



Figure(2.24) : VGG16 architecture [75]

2.7 The Selected Approach

As shown previously in this chapter, there are many approaches and methods to build and train a model based on the dataset. In order to select the appropriate approach for this project, there are two aspects to be considered.

2.7.1 Theoretical justifications

In theory, the traditional Machine Learning algorithms require the process of feature selection. Which is concerned with detecting and considering only the most important and effective features of the dataset to be trained on the model [76].

This is crucial since the traditional Machine Learning algorithms are very sensitive to the noise and irrelevant data [77].

Moreover, complex data (such as images) are difficult to extract their features manually and their importance manually. Thus in Deep Learning, the features are learned by the model itself based on the data. Therefore, the extracted features will be the true and real features with high accuracy [78].

2.7.2 Practical justifications

In order to select the right approach, recent research with different approach working on the same problem (AMD classification) are compared. So the performance of each different approach is evaluated.

Table (2.1) shows different approaches, with their results.

As seen in Table (2.1), we can see that most of the traditional Machine Learning classifiers have relatively low performance (such as k-NN, DT, and others). However, the SVM showed the highest level of performance among other traditional Machine Learning algorithms. On the other hand, CNN (Deep Learning algorithm) showed much higher performance (almost perfect) comparing to the traditional Machine Learning.

This observation shows a practical evidence that Deep Learning models (especially CNN) outperforms the different types of traditional Machine Learning classifiers and approaches, in classifying images. In addition, especially images related to AMD cases.

Table 2.1: Performance comparison between different classifiers

Researchers	Year	Dataset	Classifier	Results
Priya and Aruna [79]	2014	Aravind Eye Hospital and Postgraduate Institute of Ophthalmology	SVM	Accuracy: 0.96 Specificity: 0.94 Sensitivity: 0.96
			Naïve Bayes	Accuracy: 0.92 Specificity: 0.88

				Sensitivity: 0.93
			PNN	Accuracy: 0.88 Specificity: 0.82 Sensitivity: 0.90
Mookiah and others [80]	2014	ARIA dataset	Naïve Bayes	Accuracy: 0.83 Sensitivity: 0.90 Specificity: 0.73
			k-NN	Accuracy: 0.82 Sensitivity: 0.86 Specificity: 0.75
			PNN	Accuracy: 0.86 Sensitivity: 0.92 Specificity: 0.78
			DT	Accuracy: 0.80 Sensitivity: 0.86 Specificity:0.71
			SVM	Accuracy: 0.95 Sensitivity: 0.96 Specificity:0.93
		STARE dataset	Naïve Bayes	Accuracy: 0.78 Sensitivity: 0.90 Specificity: 0.59
			k-NN	Accuracy: 0.74 Sensitivity: 0.76 Specificity: 0.66
			PNN	Accuracy: 0.76 Sensitivity: 0.90 Specificity: 0.55
			DT	Accuracy: 0.79 Sensitivity: 81 Specificity: 75
			SVM	Accuracy: 0.95 Sensitivity: 0.96 Specificity: 0.93
García-Floriano and others [20]	2017	STARE dataset	SVM	Accuracy: 0.83 Precision: 0.84 Recall: 0.83 F-measure: 0.83
Mookiah and	2014		Naïve	Accuracy: 0.83

others [81]			Bayes	Sensitivity: 0.85 Specificity: 0.82
			k-NN	Accuracy: 0.87 Sensitivity: 0.85 Specificity: 0.89
			PNN	Accuracy: 0.89 Sensitivity: 0.81 Specificity: 0.97
			DT	Accuracy: 0.84 Sensitivity: 0.84 Specificity: 0.84
			SVM	Accuracy: 0.93 Sensitivity: 0.91 Specificity: 0.96
Tan and others [17]	2018	Ophthalmology Department of Kasturba Medical College (KMC)	CNN	Accuracy: 0.95 Sensitivity: 0.96 Specificity: 0.93
Matsuba and others [82]	2018	Tsukazaki Hospital	CNN	AUC: 0.99 Sensitivity: 1.0 Specificity: 0.97
Keel and others [83]	2019	Different hospitals	CNN	AUC: 0.99 Sensitivity: 0.96 Specificity: 0.96 Accuracy: 0.96

Combining both the theoretical and practical justifications, the selected approach to tackle this problem is to use Deep Learning instead of the traditional Machine Learning. Especially the CNN approach due to its previously stated advantages of it.

2.8 Image Processing Techniques

This section discusses the image processing methods that are used in this thesis for the purpose of preprocessing and improving the input pictures.

2.8.1 Type Of Image

The types of image used in this project can be classified into [84]:

A. Gray Scale Image

It is called a monochrome image. A function of two-dimensional light intensity, f (x, y), is defined in that each of the two variables is a spatial coordinate and the value of f is proportional to the brightness (or gray level) of the picture at each point in time (x, y). Each pixel in a gray-level image has its own brightness. Zero normally stands for black, and white represents the highest possible value. For example, in an eight-bit image, the overall unsigned value of eight bits is 255 and that is the value used for white. To calculate the value of each pixel in the new image by weighted summation of the three channels (RGB) of the original image. Equation (2.12) shows the configuration of the CV library in calculating the greyscale image.

$$Y = 0.299*R + 0.587*G + 0.114*B \qquad \text{…(2.12)}$$

Where: Y is the new greyscale image. R, G, and B are the pixel values for the red, green, and blue channels respectively [112].

B. Color Image

In a color image, every pixel has its own color and brightness, which is typically depicted as a triple component; red, green and blue intensities

2.8.2 Image Resize

The size of some of the pictures recorded by the camera and supplied to our AI system varies. As a result, a base size must be generated for all pictures given into the prposed system. The regular size maintains the clarity and sharpness of the visual elements and saves time when compared to the bigger size.

2.8.3 Remove Noise (Denoise)

At this stage, the images are cleared of the standard known noises and defects. During the capture and transmission of images, there is noise. Image enhancement is the main and primary step in the image processing concept. It is used to develop the quality and brightness of an image. A filter was used, and it was very effective in reducing image noise [85].

A. Types of noises: noise in image is usually a variance in brightness and colour intensity among the pixels of the image. It occur usually during the process of capturing the image or scanning it due to artificial effects (such as flashlight) or natural effects (such as high sun bright). These defects carries a large impact on the process of extracting features from the images and classifying them. Some of the most common noises are:

1. **Salt and Pepper Noise**: Sometimes it's called impulse noise, or spike noise, or random noise, or independent noise. This noise can be caused by sharp and sudden disturbances in the image signal. It presents itself as separated white and black pixels [86]. As showed in Figure (2.25). On the left side, there are two normal retina images. On the right side, the same images with salt and pepper noise.

Figure (2.25): Salt and Pepper Noise

2. **Gaussian Noise:** is also called white noise, or normal noise. This is because the noise have similar density to the normal distribution (i.e. Gaussian distribution). In practice, the occurrence of this type of noise is usually caused by poor illumination [87]. Figure (2.26) shows the effect of Gaussian noise. On the left side, two normal retina images. On the right side, the same images with Gaussian noise.

Figure (2.26): Gaussian Noise

B. Noise filters: in order to enhance the quality of the images, and reduce the noises effect, filters (sometimes-called kernel) must be applied to the images. Filter is relatively small matrix. Which is multiplied with the image itself to reduce the noise.

Average filter (mean filter): One of the main defects in images is intensity variation between the pixels in the images. This problem is addressed by applying the Average Filtering (Mean Filter). The mean filter is a linear type filtering method. Which smooth the image data. Each pixel mask's performance is averaged together to make distinct pixel from other pixels; hence, it is called an average filter. Mainly in photographic images (i.e. In fundus photographic images), the grain noises are removed using this mean filter. Equation (2.13) shows the Average filter mathematical bases [88].

$$\hat{f}(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t) \qquad ...(2.13)$$

Where m and n represemnts the dimention of the kernal. The $g(s,t)$ represents the pixel value of the original image. s and t represents the coordinates of the kernal.

2.8.4 Adaptive Histogram Equalization with Limited Contrast

First, we must look into adaptive histogram equalization in order to get a comprehensive grasp of CLAHE (AHE). A technique for improving the contrast of images in computer image processing, it is known as contrast enhancement. Because it differs from the more traditional histogram equalization method, the adaptive technique produces a number of histograms, each representing a distinct portion of the image, and then uses them to disperse brightness values across the image. Consequently, it is well-suited for enhancing local contrast and sharpening the definition of edges in each region of a photograph [89].

In regions of an image where the noise is usually homogeneous, AHE, on the other hand, has a tendency to amplify the noise. It overcomes this by limiting the amplification in the case of contrast limited adaptive histogram equalization (CLAHE), a kind of adaptive histogram equalization (AHE).

Because of the highly concentrated histogram in these places, ordinary AHE has a propensity to exaggerate the contrast in parts of the image that are almost constant in contrast. As a result, AHE may result in the amplification of noise in regions with a high degree of consistency. Converse Restricted Adaptive Histogram Equalization (CLAHE) is a kind of adaptive histogram equalization in which contrast amplification is restricted in order to alleviate the problem of noise amplification. It is one of the most often used types of adaptive histogram equalization.

In CLAHE, the contrast amplification in the vicinity of a given pixel value is determined by the slope of the transformation function. This is proportional to the slope of the neighbourhood cumulative distribution function (CDF), and therefore to the value of the histogram at that particular pixel value in the histogram. By clipping the histogram at a preset value before computing the CDF, CLAHE prevents overamplification of the signal from occurring. This constrains the slope of the CDF and, therefore, the slope of the transformation function. Because the clip limit is reliant on the normalization of the histogram and, as a result, the size of the neighbourhood area, it is important to understand how the clip limit is calculated and what it means. The resulting amplification is usually restricted to three or four times the original volume.

Rather than eliminating the part of the histogram that exceeds the clip limit, it is better to distribute it equally over all histogram bins rather than discarding the whole histogram [90].

Figure (2.27) shows how the CLAHE filter behave on each type of the images in our dataset.

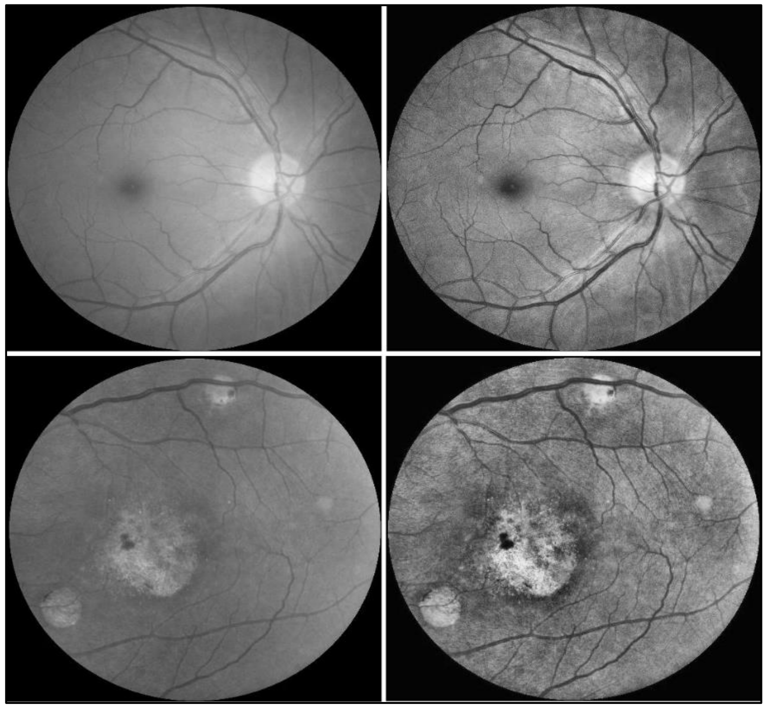


Figure (2.27): Retina fundus images on the left hand side, and the same images on the right hand side after applying CLAHE.

2.8.5 Image Normlization

Normalization is a method used in image processing that change the range of pixel intensity values that may be seen. Photos with poor contrast due to glare are one example of this phenomenon. The terms "normalization" and "contrast stretching" are often used interchangeably, as is the phrase "histogram stretching." Dynamic range expansion is the term used in more general fields of data processing, such as digital signal processing, to refer to this technique.

For a variety of applications, the goal of dynamic range expansion is often to place an image or other kind of signal into a more familiar or normal range for the senses, thus the term normalization. When dealing with a collection of data, signals, or images the aim is often to preserve the dynamic range of the collection in order to avoid mental distraction or fatigue. For example, a newspaper will make an effort to ensure that all of the photographs in an issue have a grayscale range that is similar [91].

Equation (2.13) shows how the normalization works in mathematical formula.

$$X_{Normlization} = \frac{X - \min(X)}{max(X) - \min(X)} \qquad \ldots(2.14)[92]$$

Where: $X_{Normlization}$ is the new normalized image, X is the original image, $max(X)$ is the upper value in the new normalized image, $\min(X)$ is the lower value in the new normalized image.

2.9 Data Augmentation

In computer vision, unbalanced datasets are a common issue that impede classification. A lack of images in each class can lead to under-fitting and over-fitting, which has a significant impact on the DCNNs' performance.

To address this issue, a data augmentation strategy within the ODIR datasets is presented to improve classifier performance.

Data augmentation is a method that increases the number of images used for training the neural network. That is, creating new data for categories that have fewer numbers in the data set. This process succeeds the constraining effect on data to prevent an asymmetric representation and successfully escape overfitting complications. Appropriate data augmentation techniques can help improve deep learning model strength [93].

Many Computer Vision tasks have shown that deep convolutional neural networks perform exceptionally well.

However, in order to avoid overfitting, these networks rely largely on massive data.

Deep neural networks are extremely complicated, requiring millions of weights to be optimized and matched to the training data. As a result, it is not only critical to have a decent model for the problem at hand, but it is also critical that the data collection in question is of appropriate quality and size. This complexity will require a larger dataset to feed and configure larger and more complex model, that is able to deal with harder and real life problems. In most deep learning applications, however, the number of parameters in a neural network exceeds the number of data points in the dataset, sometimes with a large margin [94].

A variety of simple data augmentation techniques will be discussed such as zooming, flipping, shifting, rotation, add noise, and transformation to be implemented to the original images.

A. **Horizontal and Vertical Flip**: Both the methods of vertical and horizontal flipping are very effective and popular for data augmentation. The augmented data is one of the simplest applications in terms of execution, which has been proven to be useful in the dataset. At the same time, flipping the horizontal axis is regarded to be more common than flipping the vertical axis because it is more convenient for most projects due to the high likelihood of finding images inverted horizontally, as shown in Figure(2.28)(b,c).

B. **Random Contrast**: In order to give the images more variation, and make the model more ruboust to changes. One main change that usually happens in real images is the contrast variation. Therefore, one of the most important data augmentation technique is contrast. Which changes the image contrast randomly in pre-determined range. Figure (2.28)(d) shows contrast data augmentation example.

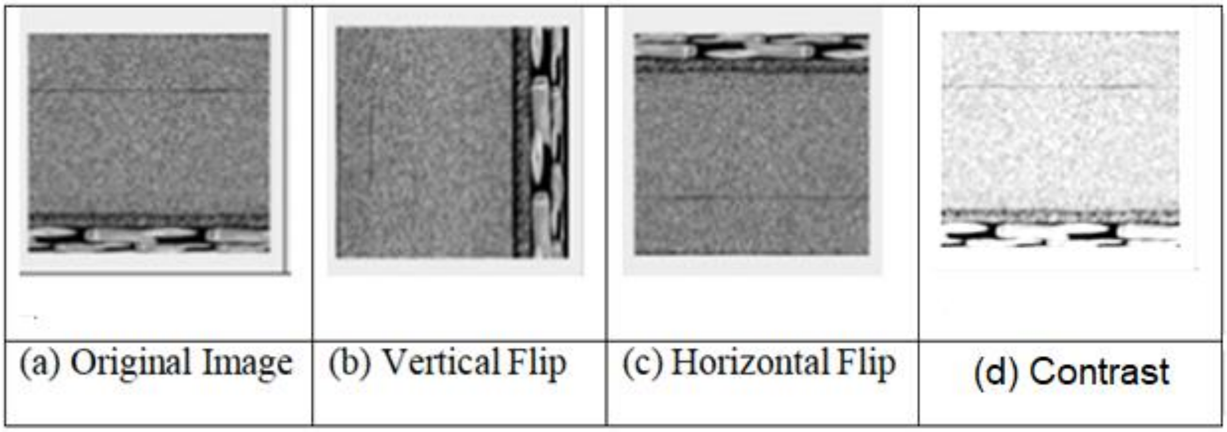


Figure (2.28) Different data augmentation techniques [95]

2.10 Evaluation Metrics

The different evaluation methods are described in this section. The need for various methods of assessing the performance of the model is essential. Since one evaluation method is not enough to capture the different parts of the system. Accuracy, recall, precision, and the F1-measure are all useful methods for assessing classification algorithms.

In order to determine the computations for these techniques, you will need to reference a matrix of computing confusion. This matrix consists of two class hierarchies: Documents create two separate class hierarchies, one for what actually predicted, and one for the expected prediction. See table (2.2).

Table (2.2): Confusion matrix

		Predict	
		Positive	**Negative**
Actual	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

A. **True positive (TP):** refers to occurrences of positivity that are categorized properly.

B. **False Negative (FN):** These are positive occurrences that have been categorized erroneously.

C. **False Positive (FP):** These are negative occurrences that are categorized erroneously.

D. **True negative (TN):** refers to occurrences of negativity that are identified properly.

These four metrics of confusion matrix are not enough the measure the quality of the model performance. Therefore, further operations and equations are held on these metrics to generate more robust and clear metrics.

1. **Confusion Matrix**: in the statistical learning and machine learning field, the confusion matrix (sometimes called error matrix) is a table showing the performance of algorithm by visualizing the results of it. It is usually used with supervised machine learning (in unsupervised machine learning, it is called matching matrix). In this table, the rows represents the real expected classes, and the columns represents the actual output of the algorithm, or vice versa as shown in Figure (2.29).

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure (2.29): Confusion Matrix

2. **Accuracy:** is the average correct prediction. Which is calculated by dividing the correct prediction over total predictions. Which will result in one value for entire network.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \qquad \ldots(2.15)$$

3. **Sensitivity:** (or Recall) the True Positive rate. Which represents the completeness of a model. Which is calculated by dividing the correctly detected phenomena (True Positive) over total true cases of that phenomenon (True Positive + False Negative). Each class/label has a single value of sensitivity.

$$Recall, Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negaive} \qquad \ldots(2.16)$$

4. **Specificity**: the true negative rate. Which represents the true negative values in the model. This is calculated by dividing the truly absence of the phenomena (True Negative) on the total number of true absence of the

phenomena (True Negative + False Positive). And each class/label has a single value of specificity. See Figure (2.30) for deeper understanding.

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \qquad \ldots(2.17)$$

5. **Precision:** represent the exactness of the trained model. Which means from the detected positive cases, how many are really positive.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \ldots(2.18)$$

6. **F1-score:** F1-score shows a compound of precision and sensitivity for computing a balanced mean output .

$$\text{F1 score} = \frac{True\ Positive}{True\ Positive + \frac{1}{2}(False\ Positive + False\ Negative)} \qquad \ldots(2.19)$$

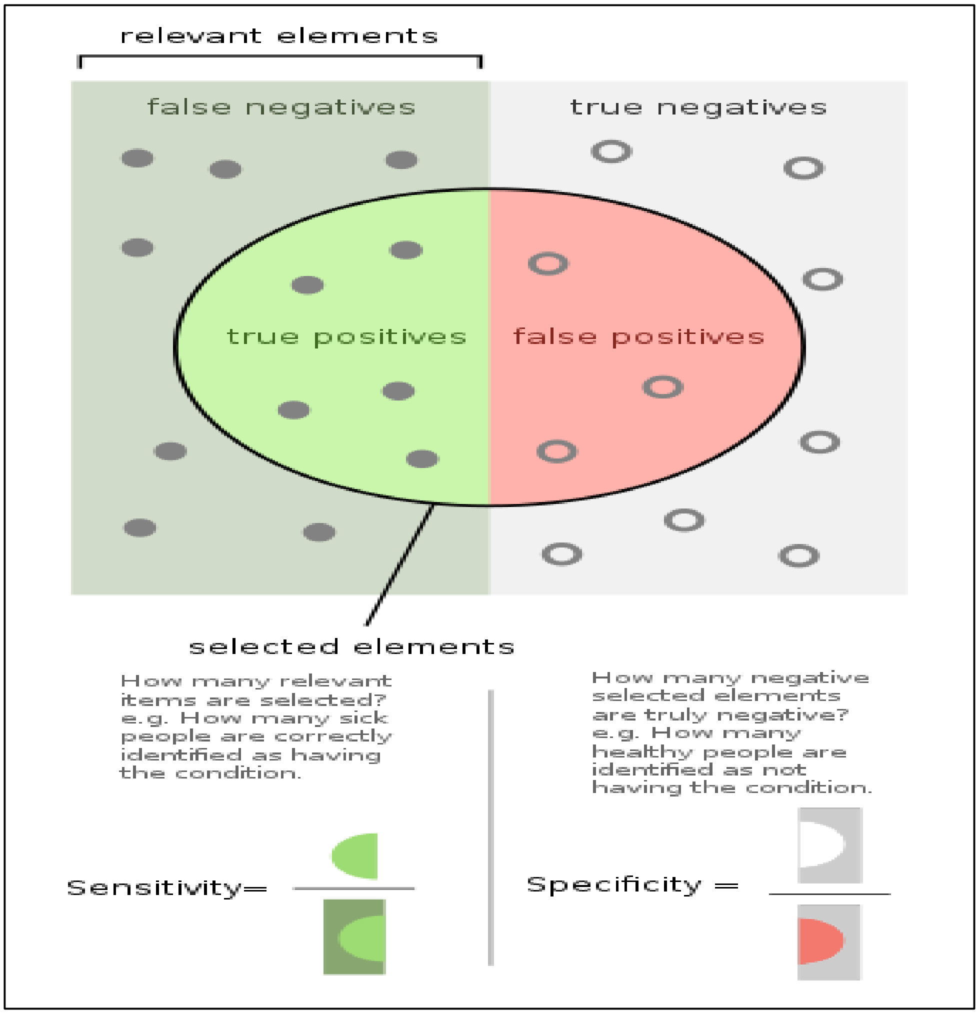
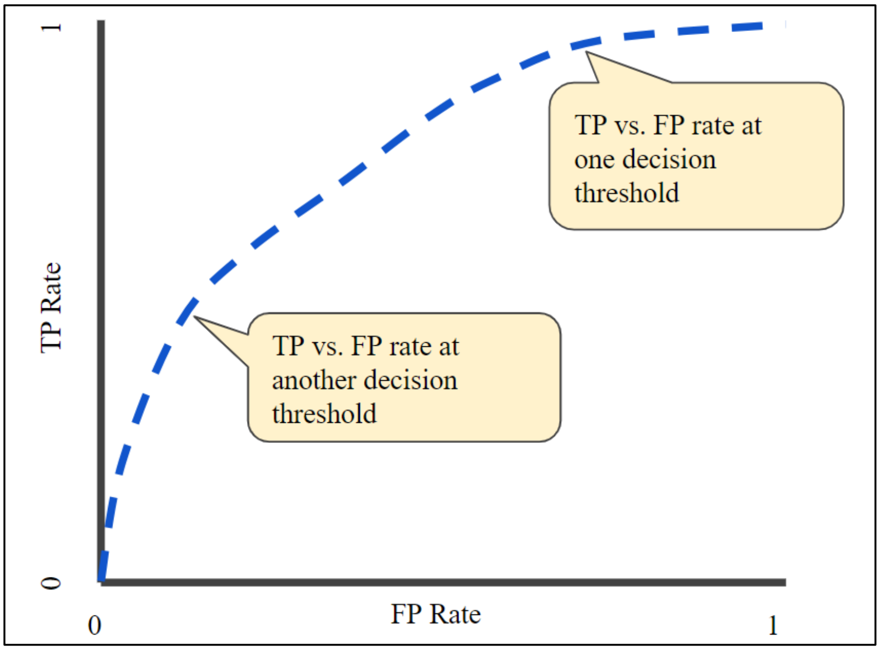


Figure (2.30): Visualization of the difference between sensitivity and specificity [96]

7. **ROC curve (receiver operating characteristic curve):** is a curve plotted on a graph to show the performance of classification model. This curve considers two parameters in its plot:
   - True Positive Rate
   - False Positive Rate

   The graph will plot the True Positive Rate against False Positive Rate at different classification threshold. If the threshold is lowered, that means more items are classified as positive. This will lead to the increase in the

False Positives and True Positives rates. Figure (2.31) shows how the ROC is plotted.



Figure(2.31): ROC plot graph [97]

8. **AUC (Area under the ROC Curve):** this curve will measure the area under the ROC curve. Which means it will calculate how much of the instance are classified correctly on different values of threshold. Which means the model ability to classify and distinguish between the different classes. In other words, AUC is a measurement of separability. It indicates how well the model can differentiate between classes. AUC value is ranked between 0 to 1. Where 0 is the lowest and represents 100% misclassification. And 1 is the highest and represents 100% correct classification of the model. It indicates how well the model can differentiate between classes. The larger the AUC, the more accurately the model predicts zero classes as zero and one classes as one. By example, the greater the AUC, the more accurate the model is in discriminating between patients

with and without illness [98]. Figure (2.32) shows the AUC plotted into a graph.

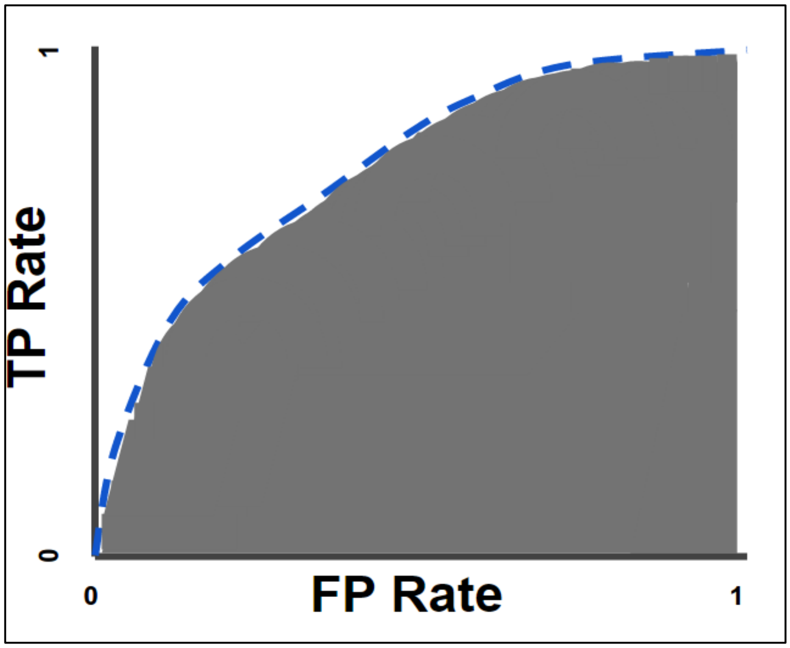


Figure (2.32): AUC curve plotted on graph [97]

For the purpose of this research, and since the problem is multiclass classification, the AUC curve will be used as the primary classification performance metric. However, the Specificity, Sensitivity, and Confusion Matrix will be used to get a detailed and rich insight on the model performance.

CHAPTER THREE

PROPOSED SYSTEM

3.1 Introduction

In this chapter, the proposed system design and development process is discussed. Besides the motivation behind it and the used tools. This includes the general architecture and all algorithms used to build and train the model. The model itself is an image classification system. Which classify the main stages of Age-related Macular Degeneration (AMD) explained in previous chapters. The images are classified based on their contents of into pre-defined categories (i.e. AMD stages). Information in images are the features that are used during the classification process. The image classification system is designed based on the contents of these images to achieve the desired output with high accuracy based on the model of deep learning Convolutional Neural Networks (CNN) algorithms.

This chapter explains the process of CNN Image Classification System. The classification contains two phases training and testing. The training phase revolves on built classifier structure from training about 70% from fundus retina images dataset that contains healthy and AMD images. Introducing images, and carrying out a number of steps to reach the correct category of unclassified images. Then after that, the testing process begins. Testing for the new unlabelled (unclassified) retina fundus image is performed using "ANN" and "CNN" classifiers that built before in the training phase. Then the retina fundus image will be classified as positive or negative. The dataset used is "ODIR" which includes 939 images of 496 AMD images and 443 Normal images. The accuracy of the system has been evaluated by utilizing popular evaluation mensuration such as "Accuracy", "Precision", “Sensitivity ", "Specificity" and Area under curve. The test phase classifies the unlabelled images into the correct category.

3.2 Dataset

The used dataset is the ODIR dataset. Which contains about 5000 coloured fundus images of both left and right eyes. These images are collected from *Shanggong Medical Technology Co., Ltd.* from different hospitals/medical centres in China. The labels of the images are set by professional physicist. The different categories of this dataset are: normal (N), diabetes (D), glaucoma (G), cataract (C), AMD (A), hypertension (H), myopia (M) and other diseases/abnormalities (O) [99].

In our project, we are interested in the Normal and AMD labels. Which make about 939 images combined for both healthy and AMD images (70% train, 30% test).

The AMD class is further divided into (116 images in Early stage, 254 in Intermediate stage, and 126 in Late stage).

Table (3.1) and Table (3.2) shows the detailed division of the dataset for binary and four class models respectively, in terms of the training and testing phases, and in terms of dataset classes.

Table (3.1): Dataset division based on the model phases and dataset classes for binary classes

	#Normal	#AMD	#Total (%)
Training	307	350	**657 (0.7)**
Testing	136	146	**282 (0.3)**
Total	443	496	**939 (1.0)**

Table (3.2): Dataset division based on the model phases and dataset classes for four classes

	#Normal	#Early	#Intermediate	#Late	#Total (%)
Training	307	88	175	87	**657 (0.7)**
Testing	136	28	79	39	**282 (0.3)**
Total	443	116	254	126	**939 (1.0)**

3.3 System Development Life-cycle

In order to develop a model that is capable of classifying images based the status of the eye, the steps showed in Figure (3.1) are followed.

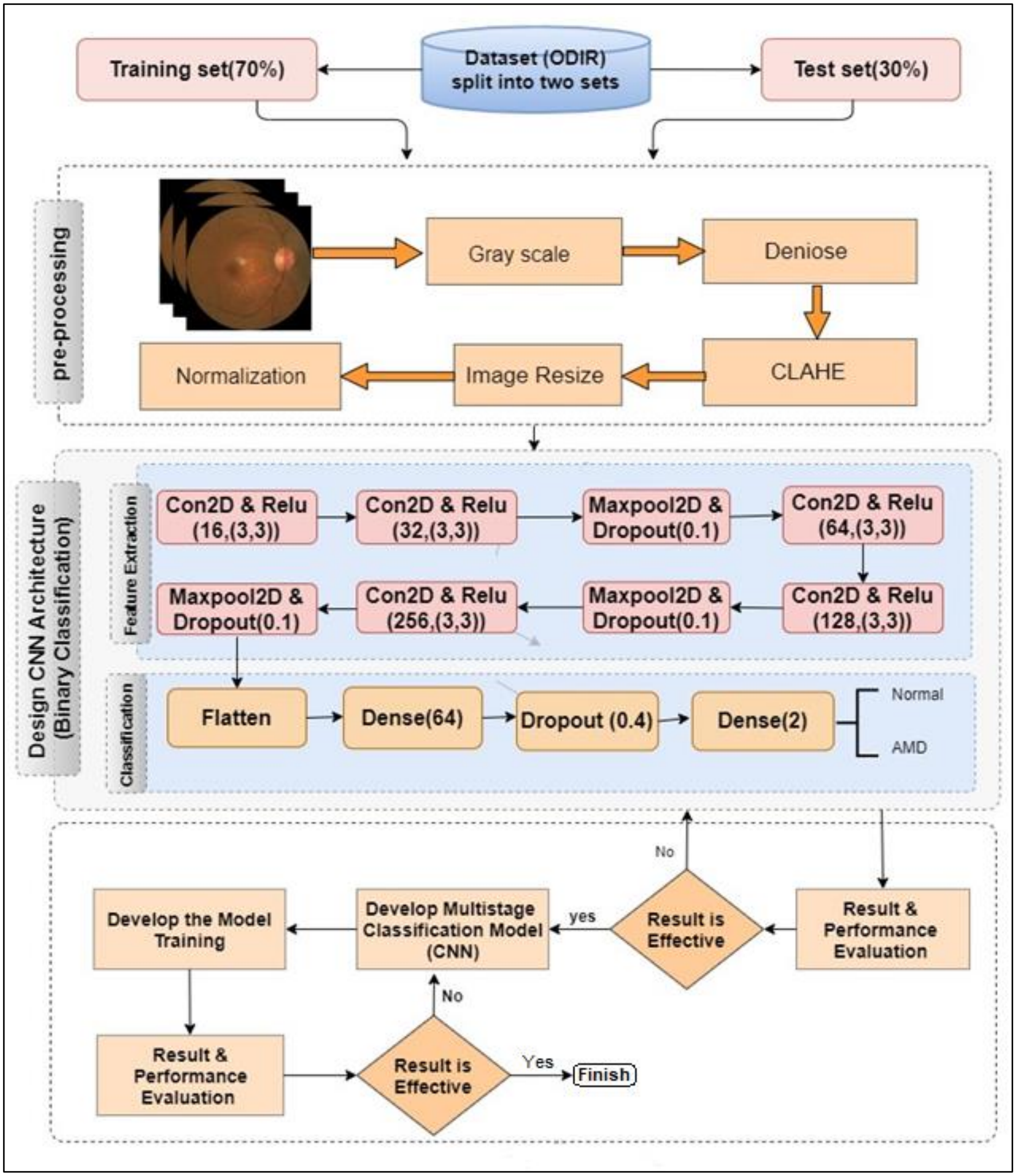


Figure (3.1): General graph for the proposed approach methodology

3.4.1 Data pre-processing

In the data pre-processing phase, which enhance and prepare the input images to the next step. To ensure their quality and validity. These steps will lead to image enhancement. These steps are consists of five processes (operations) to prepare the data to be valid to the model training. Algorithm 3.1 shows the steps of the pre-processing operations. These processes are:

A. Grey scaling: the original images are coloured images with three channels (RGB). In order to reduce the processing time, the images are converted into greyscale images (only one channel). The greyscale images keeps the important features (i.e. drusens, eye vessels and tissues, etc.) sharp and obvious. Therefore, the grey scaling operation didn’t damage or negatively affect the images features.

Since we are using CV library to convert the coloured images into greyscale, the followed method in CV library is the weighted grey scaling [100].

B. Denoise: in this stage, the images are cleared from standard know noises and defects. Images are noised during the capture and transmission.

In order to remove the types of noise described previously, Average filter (mean filter) is used (described in the previous chapter). The filter is a linear type filtering method. The image data is smoothed as a result of this. Each pixel mask's performance is averaged together to make a distinct pixel from other pixels, and, hence, it is called an average filter. Kernal size of (9*9) is used.

C. Image enhancement (CLAHE): The main goal of image enhancement techniques is to process a specific image so that the result is more appropriate than the original image, or this is done by increasing the distinction between the details in the image, noting that the improvement process is done after the image correction process is carried out by removing the noise in the image. In this step, the fundus images were pre-treated using the limited-contrast adaptive graph equation (CLAHE) technology. CLAHE is an effective contrast enhancement method that effectively increases image contrast.

D. Image resize: since the first layer require static size of the input image, all the training and testing images are resized to standard size. Notice that the original images in the dataset don't follow a standard size. The selected standard size is 223 * 223 pixels. The selected standard size ensure that the visual features are still clear and sharp, and reduce the training time comparing with larger size. The process of resizing the images are done in Python.

E. Normalization: Divide the image elements by 255 to make the values between 0 and 1 to speed up the processing process. As showed in equation (2.14) in chapter two.

Algorithm(3.1):Pre-processing steps
Input: Image Data-Set
Output: Pre-processed Images
Begin **Step1:** Load Image Data-Set **Step2:** Greyscale Images (Equation 2.12) **Step3:** Denoise by using Mean Filter (Equation 2.13) **Step4:** Image Enhancement (CLAHE) **Step5:** Images Resize (223 * 223) **Step6:** Image Normalization (Equation 2.14) End.

3.3.2 Convolutional Neural Network

In order to develop a Convolutional Neural Network (CNN) model, two stages are involved. First is model design. Second is training parameters.

A. CNN design: the design phase include arranging the different layers in order to get the best feature extraction process. Table (3.3) shows detailed description of the layers. the layers of the CNN are as follow:

1. **Layer 1 Convolutional layer (conv2D):** the first layer is convolutional layer. With 16 filters the layer will learn. With 3*3 kernel size, padding (same), and the used activation function is ReLU. Since this is the first layer in our model, the parameter *input_shape* is used. Which is set to *(s, s, 1)*. Where the *s* is the size of the image (223 for both the width and height of the image), and 1 for the single channel image. .
2. **Layer 2 Convolutional layer (conv2D):** the second layer of the model is convolutional layer too. However, the number of filters are 32, and have similar kernel size to layer 1 (3 * 3), padding (same), and ReLU activation function. Notice there is no need to determine the input shape since the network will detect it automatically from the previous layer.
3. **Layer 3 Max pool layer (maxpool2D):** the third layer is max pooling layer. With (2 * 2) size, and stride value of (2 * 2).
4. **Layer 4 Dropout:** layer 4 is a dropout regulation layer. With 10% value.
5. **Layer 5 Convolutional layer (conv2D):** the fifth layer is convolutional layer with 64 filters, and kernel size of (3 * 3), , padding (same), and reLU activation function
6. **Layer 6 Convolutional layer (conv2D):** the sixth layer is convolutional layer, with 128 filters, and kernel size of (3 * 3), padding (same), and reLU activation function.

7. **Layer 7 Max pool layer (maxpool2D):** the seventh layer is max pooling layer. With (2 * 2) size, and stride value of (2 * 2).
8. **Layer 8 Dropout:** the eighth layer is a dropout regulation layer with 10% value.
9. **Layer 9 Convolutional layer (conv2D):** the ninth layer is convolutional, with 256 filters, and kernel size of (3 * 3), padding (same), and reLU activation function.
10. **Layer 10 Max pool layer (maxpool2D):** the tenth layer is max pooling layer. With (2 * 2) size, and stride value of (2 * 2).
11. **Layer 11 Dropout:** the eleventh layer is layer is a dropout regulation layer, with 10% value.
12. **Layer flatten:** this layer will transform the previous feature map matrix into single column of values. So it will be appropriate to be entered to the next fully connected layer. The feature maps (three dimensions) will be converted to a vector (one dimension) so the output is $27 \times 27 \times 256 = 186624$.
13. **Layer 12 Dense layer:** In the twelfth layer, the number of units (size of output layer) is set to 64, and the activation function is ReLU.
14. **Layer 13 Dropout:** the thirteenth layer is a dropout regulation layer. With 40% value. To regulate and prevents the network from overfitting**.**
15. **Layer 14 Dense layer:** In the fourteenth layer, the number of units is set to 2 (to represent the two classes of *Normal* and *AMD*. And the activation function is Softmax.

Table (3.3): The proposed CNN structure

Layer (type)	Output Shape	Parameters number
layer_1 (Conv2D)	(223, 223, 16)	160
layer_2 (Conv2D)	(223, 223, 32)	4640
layer_3 (MaxPooling2D)	(111, 111, 32)	0
layer_4 (Dropout)	(111, 111, 32)	0
layer_5 (Conv2D)	(111, 111, 64)	18496
layer_6 (Conv2D)	(111, 111, 128)	73856
layer_7 (MaxPooling2D)	(55, 55, 128)	0
layer_8 (Dropout)	(55, 55, 128)	0
layer_9 (Conv2D)	(55, 55, 256)	295168
layer_10 (MaxPooling2D)	(27, 27, 256)	0
layer_11 (Dropout)	(27, 27, 256)	0
flatten (Flatten)	(186624)	0
layer_12 (Dense)	(64)	11944000
layer_13 (Dropout)	(64)	0
layer_14 (Dense)	(2)	130
Total Trainable Parameters:		**12,336,450**

B. **Training hyper-parameters:** in order to build an algorithm that will be able to train itself to classify the images of eyes with AMD disease, from normal images, many parameters and sub-techniques must be set. Here we will discuss deeply each parameter and what are the available options for them, and what option is choose for our approach, and why.

1. **Dataset split**: In this step, after the image pre-processing process is done, the dataset is uploaded to the proposed system. The dataset is divided into 70% training and 30% testing datasets.
2. **Loss function**: Since we are dealing with multiple class classification problem, and memory is needed to do the training fast, so we choose the Sparse Multiclass Cross-Entropy Loss (as described in the previous

chapter). Notice with this method, the labels provided to the loss function must be integer.

3. **Optimizer**: The optimization process is about finding the best parameters and values for the kernels and biases, to set the training operation correctly (as described in the previous chapter). Comparing all the optimizers available, to our approach the Adam optimizer is chosen (algorithm 2.1). Due to its high performance compared with other optimizers [101] [102]. The initial learning rate is set to 0.001, number of epochs is set to 25, and batch size is 64.
4. **Early Stopping:** Early stopping technique is used to regulate the training of models (as described in the previous chapter).

   In our model training, we set the validation dataset as the stopping criteria. And the value of *min_delta* which refers to the minimum amount of change to be considered is set to 0.005. In other words, the changes leading to less than 0.005 in accuracy will not considered as changes, for both improving and reducing the accuracy of the model. And the value of *patient* parameter is set to five. Meaning the model will runs five extra epochs after the training have no improvements, and then stops.

3.4.3 System training process

The system now have pre-processed images, which are suitable for training. And the CNN design and parameter are set.

The training parameters are:

- Initial learning rate: 0.001
- Epochs: 25

- Batch size: 64
- Decay: 0.00003 (which is the result of dividing the initial rate on 30)
- The early stopping will depend on the validation loss value. With patience value of 5, and minimum delta of 0.005, and verbose of 1.
- Learning rate reduction will depend on the value of validation loss too. With patience of 2, verbose of 1, and changing factor of 0.2, and minimum learning rate of 0.0001.

Algorithms (3.2) and (3.3) shows how the training is conducted in the proposed model.

Figure (3.2) shows a general diagram of the training process. Starting from loading the dataset, until testing the model.

Algorithm (3.2): Training phase of Convolutional Neural Network
Input: set of labelled and pre-processed images
Output: classified images
Begin **Step1:** Read the training images **Step2:** Build CNN model **Step3:** Set the training parameters **Step4**: Train CNN model (Algorithm 3.2) **End.**

Algorithm (3.3): Train CNN	
Input:	*X*: input images *Y*: image labels

	Epochs: number of epochs *Layers*: the model design
Output:	*W*: the new weights *K*: the new kernels

Begin

Step1: Read the dataset

Step2: Pre-process the dataset

Step3: divide the dataset (70% training, 30% testing)

Step4: build the model

Step5: For epoch = 1 to *Epochs*

For i =1 to N //where N is the number of iterations

Pass *X[i]*, *Y[i]*, *Layers* into equation (2.12) to produce *output*

Pass *output* into equation (2.9) and (2.10) to produce *error*

Pass *error* into algorithm (3.2) to update *K* and *W*

Next i

Next epoch

End.

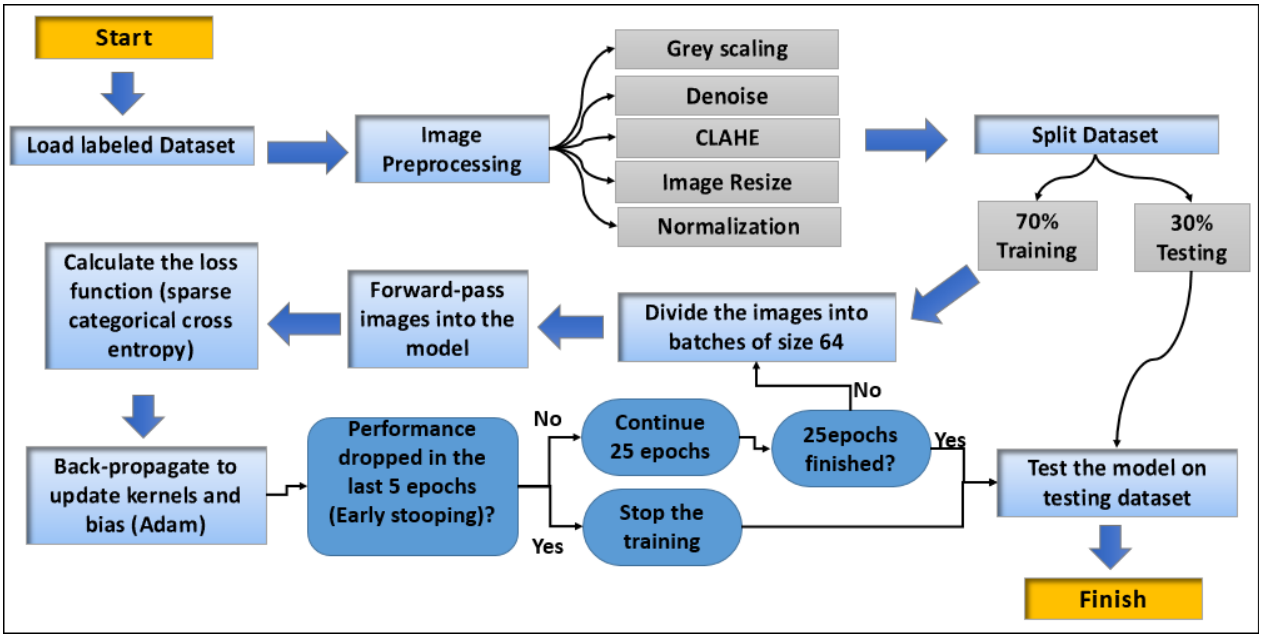


Figure (3.2): CNN training process

3.4.4 Extended System

In order to include all the stages of AMD (*Early*, *Intermediate*, and *Late*) cases, besides the *Normal*, the previous model is upgraded.

The new approach is similar to the previous one, with few changes in the design and training parameters, to perform on the full stages. For model extending to include all the stages, with higher results, improvements on the design and trailing parameters are conducted.

A. New design: the design is extended by adding new layers. The new layers are only convolutional layers.

- Layer2_1: this layer is inserted after layer_2. The size of this layer is (223, 223, 64).
- layer_5 and layer_6: layer_5 and layer_6 are replaced with 3 convolutional layers. Which are layer_5, with size (111, 111, 128), layer_5.1, with size (111, 111, 256), and layer_6, with size (111, 111, 512).

- layer_9: this layer is updated from the size (55, 55, 256), to the size of (55, 55, 640).

Figure (3.3) shows the updated extended design (four classes) of the model, and Table (3.4) shows detailed structure of the new layers.

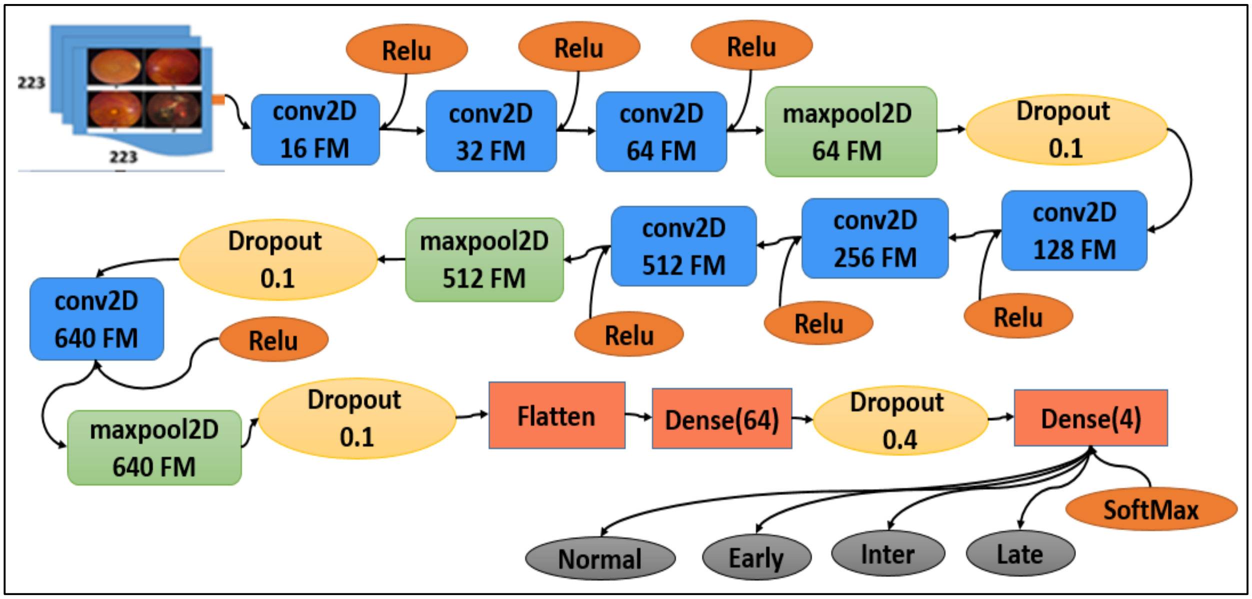


Figure (3.3): The developed architecture for 4 classes classification

Table (3.4): Detailed architecture for 4 classes classification

Layer (type)	Output Shape	Parameters number
layer_1 (Conv2D)	(223, 223, 16)	160
layer_2 (Conv2D)	(223, 223, 32)	4640
layer_2.1 (Conv2D)	(223, 223, 64)	18496
layer_3 (MaxPooling2D)	(111, 111, 64)	0
layer_4 (Dropout)	(111, 111, 64)	0
layer_5 (Conv2D)	(111, 111, 128)	73856
layer_5.1 (Conv2D)	(111, 111, 256)	295168
layer_6 (Conv2D)	(111, 111, 512)	1180160
layer_7 (MaxPooling2D)	(55, 55, 512)	0
layer_8 (Dropout)	(55, 55, 512)	0
layer_9 (Conv2D)	(55, 55, 640)	2949760
layer_10 (MaxPooling2D)	(27, 27, 640)	0

layer_11 (Dropout)	(27, 27, 640)	0
flatten (Flatten)	(466560)	0
layer_12 (Dense)	(64)	29859904
layer_13 (Dropout)	(64)	0
layer_14 (Dense)	(4)	260
Total Trainable Parameters:		**34,382,404**

B. New training parameters: to achieve the most of the new improvements, updated training parameters are needed. The new improvement on the training parameters are:

- Epochs: more epochs are required to fulfil the new goal. The new value of the epochs are raised from 25 to 30.

C. Testing phase of Convolutional Neural Network

- This part is the implementation of the test phase to predict several categories and the result of the test phase are classes to test the dataset.
- The features are extracted in the testing phase in the same way as in the training phase, but only in the forward direction, where the test images pass over the five blocks dedicated to extracting features in the structure of convolutional neural networks and then classifying these images into four stages using the trained weights in the fully connected layers and the trained kernel in the convolution layers that were stored in the training phase to be applied in the test phase.

CHAPTER FOUR
EXPERIMENTAL RESULTS

4.1 Introduction

In this chapter, a full review and comparison of the results yielded from the various developed systems, and the state of arts researchers are discussed.

First, a recap on the dataset and evaluation metrics previously will be discussed. In addition, the results of the pre-processing steps (image processing to clean the data) are descried, and a sample of their results are presented. After that, comparison with other research is conducted to evaluate the proposed approach. And finally, a real case study, which real AMD images are collected and tested on the proposed approach is described in details.

4.2 Software and Hardware Requirements

In order to carry the process of developing, training, and evaluating the system, specific software and hardware requirements are essential.

4.2.1.1 Hardware requirements:

The proposed AMD classification system operates by using personal computer hp that have specifications such us Intel(R) Core i7- 10510U @ 1.80 GHz 2.30 GHz for CPU, 8 GB, Windows10 of RAM and 64-bit Operating System.

4.2.1.2 Software requirements:

The proposed system done using Python version 3.6.9, and Colab notebook from Google.). Used with Python to easily implement CNN code. (TensorFlow framework (an open-source software library from Google that is focused on effective work with tensors), Keras (a neural network library based on TensorFlow written in Python, and also open-source), The system relies on open source libraries such as Open, Scikit Learn, and Pandas. These libraries are dedicated to dealing with machine learning and data analysis.

4.3 Preprocessing Results

In order to see the effect of the pre-processing operations on the images in the dataset, sample of eight images are selected to see the effect of the images pre-processing operations. These eight images are visualized first in the original form. After that, each operation of the pre-processing operation is visualized separately to all the eight images to see the effect of that operation.

Figure (4.1) shows the original shape and features of the images before pre-processing. After that, Figure (4.2) shows the images after applying average filter. Notice that images are generally clearer after this step. In Figure (4.3), the CLAHE operation is clearly made the features of the images sharper and very clear, without affecting the histogram or the balance of the images in general. After that, a resize operation is conducted. Which set the size of all images to 223 by 223 pixel. The results of images resizing are showed in Figure (4.4). However, since the window size of plotting the images is set, the effect of the resizing is not very obvious.

The last step is converting the images into greyscale; to make the image processing in the whole system is much faster, while maintaining the majority of the important and relevant features in the images sharp and obvious. The results of this operation are showed in Figure (4.5).

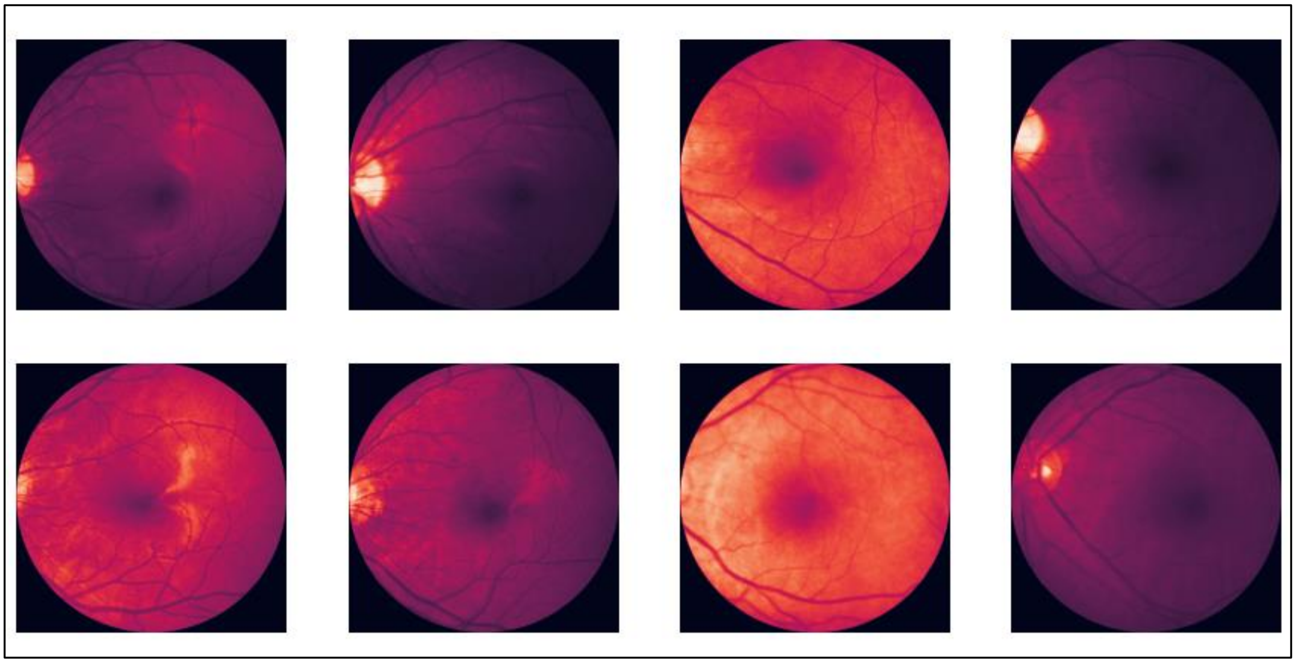


Figure (4.1): The original sample of the eight images

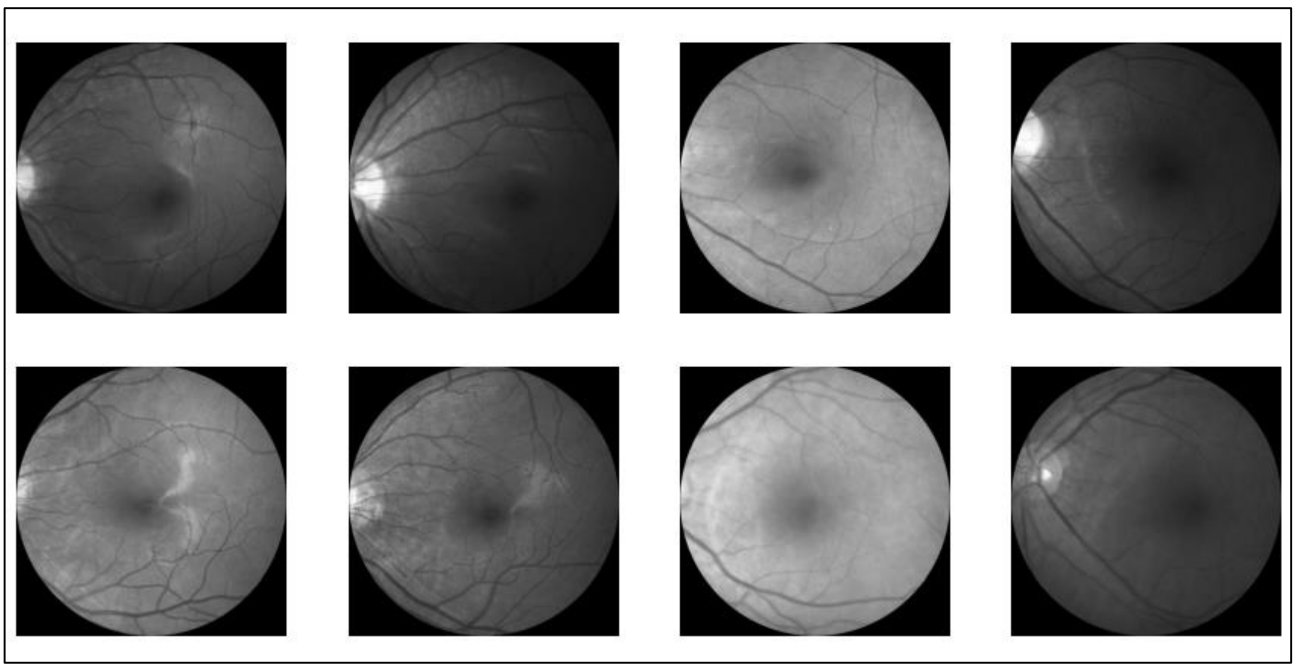


Figure (4.2): The images after converting them into greyscale

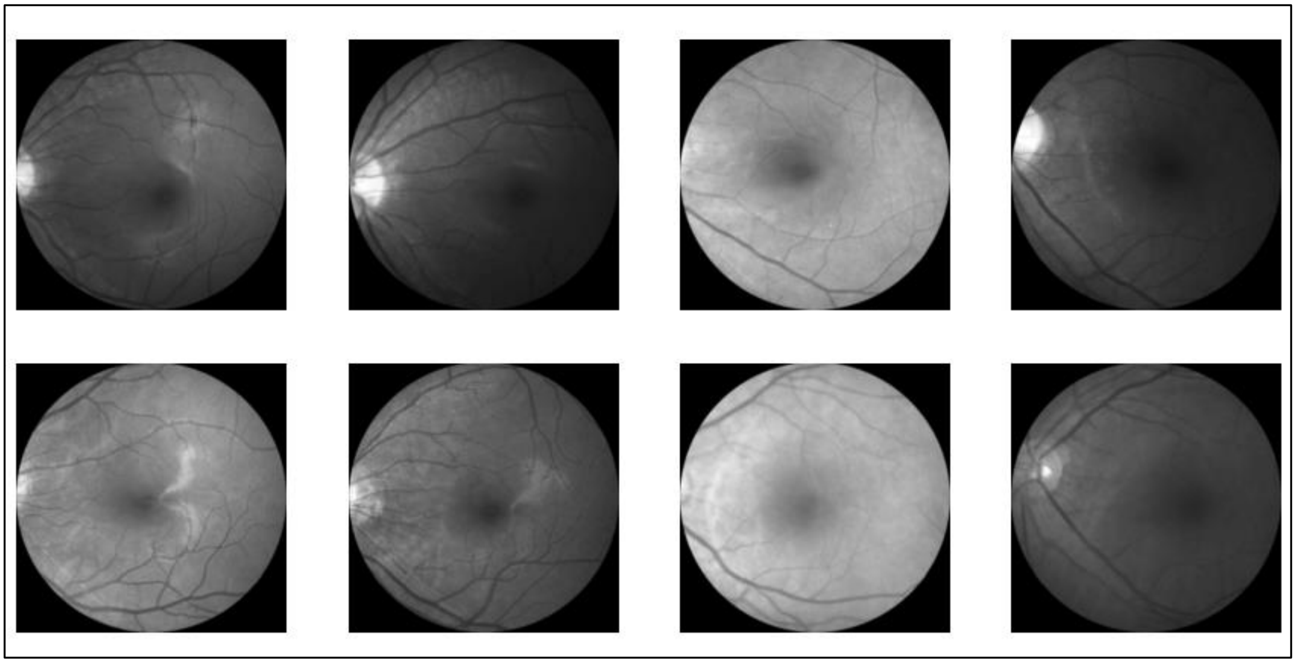


Figure (4.3): The images after applying Mean filter

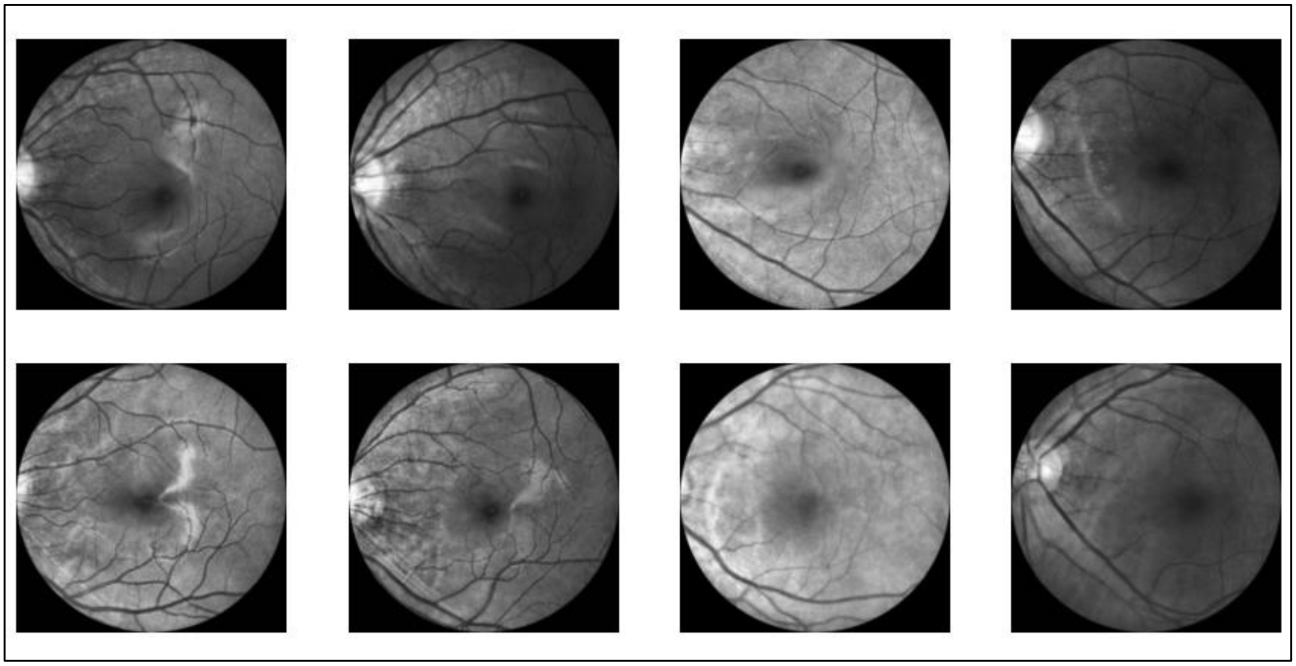


Figure (4.4): The images after applying CLAHE

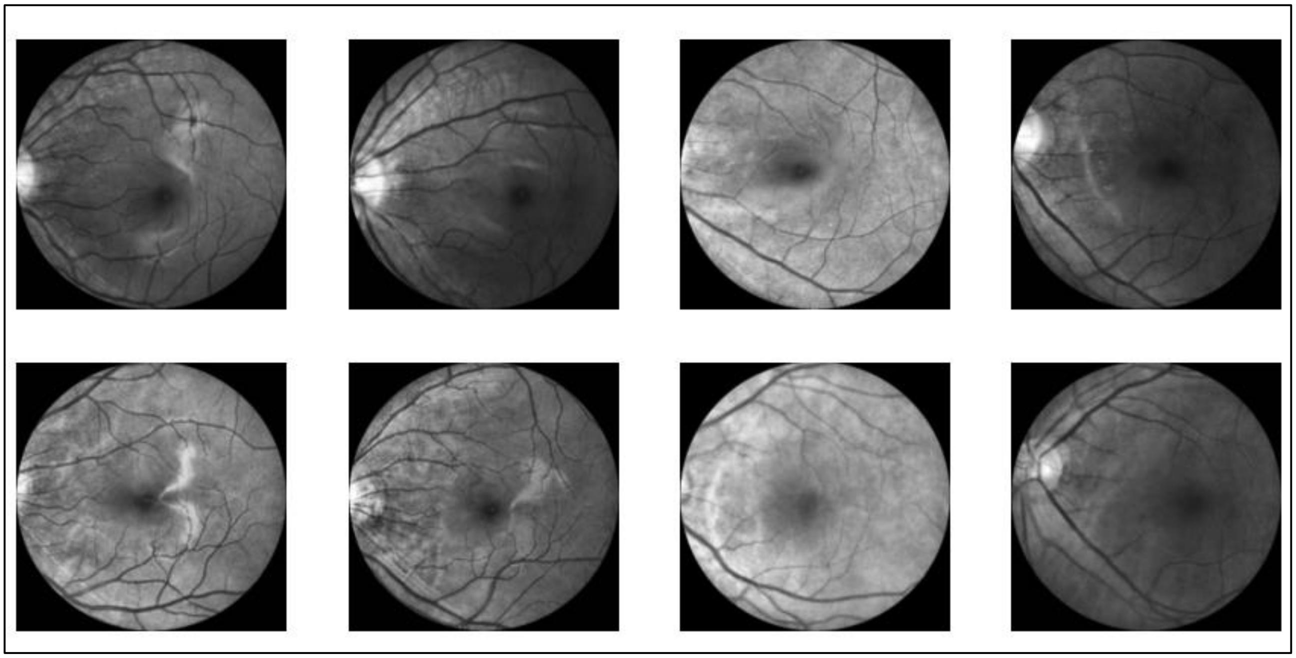


Figure (4.5): Final images after resizing and normalization

4.4 Binary Classification Results

In this stage of the system development, the number of classes are two. Which are *AMD* and *Normal*. During the training process, two measures are counted in order to check the model training performance. Figure (4.6) showed the model performance during epochs according to accuracy. While Figure (4.7) shows the training of the model performance for each epoch according to loss.

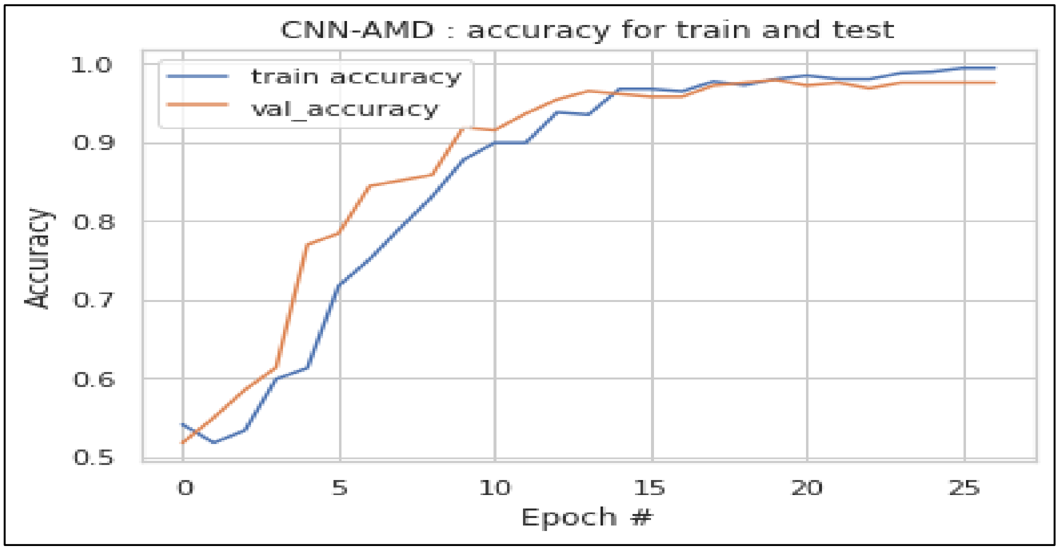


Figure (4.6): The accuracy of train and validation

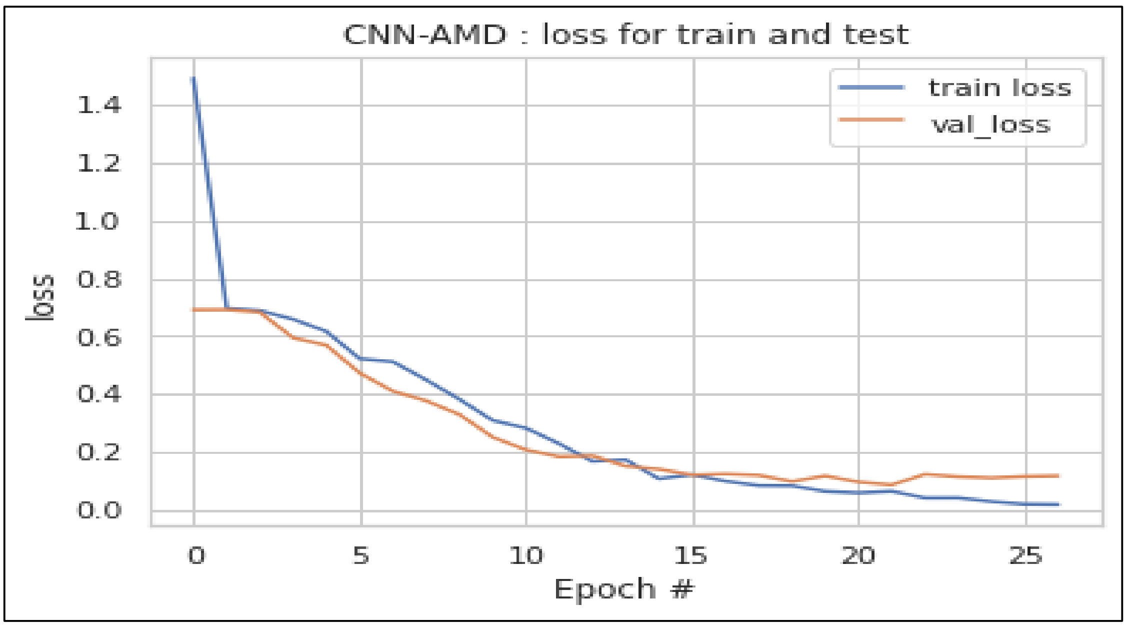


Figure (4.7): The loss of train and validation

This training led to high performance model when tested. Figure (4.8) shows the confusion matrix extracted by the test phase. In addition, Table (4.1) shows the model performance according to Sensitivity, Specificity, AUC, and Accuracy.

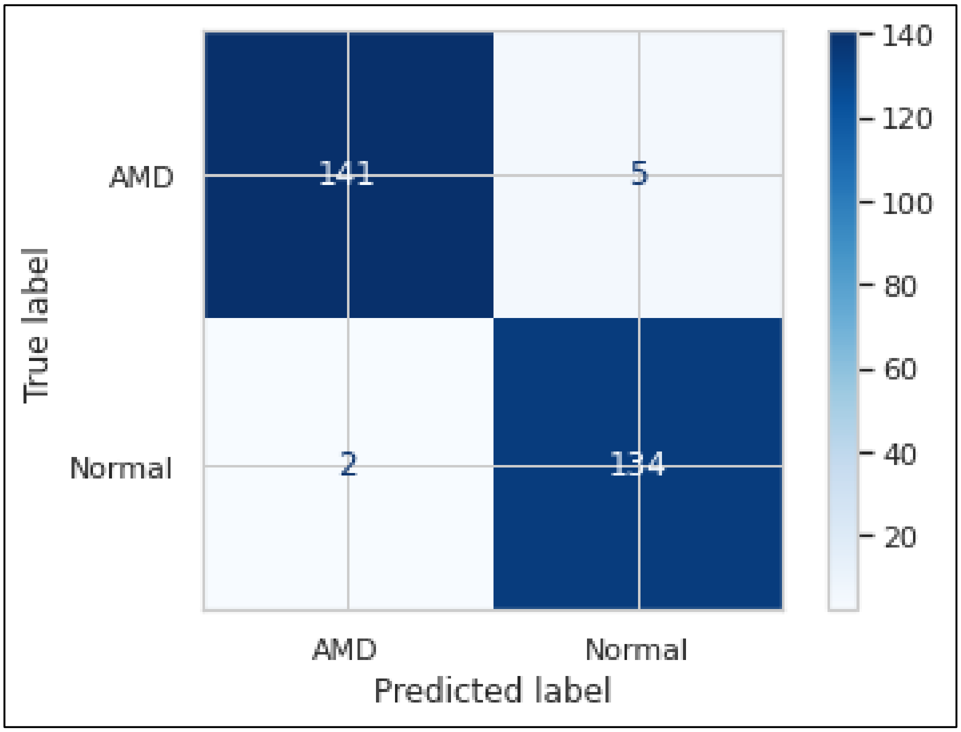


Figure (4.8) Confusion matrix extracted by the test phase for binary system

Table (4.1) Performance metrics for binary classification

Sensitivity	Specificity	AUC	Accuracy
0.965	**0.985**	**0.975**	**0.98**

These values shows high level of performance in binary classification using our approach.

Notice that the training process of two classes took (206) seconds to train the model.

4.5 Further System Validation

In order to validate the proposed system more robustly. Two methods are conducted. The first one is to train and test popular and efficient models and networks in the Deep Learning fields on the same dataset used in this system, and compare their performance. The second one is to compare the results obtained by this approach by other published researches worked on the same dataset.

4.5.1 Other models

The selected models to compare their performance on the same dataset are AlexNet and VGG16. Both of them are described in the previous chapters. Similar parameters described in section 3.4.3 are used in term of the training process for both AlexNet and VGG16. Here the performance on the ODIR dataset is displayed in Table (4.2)

Table (4.2): AlexNet vs. VGG16 performance

	AlexNet	VGG16
Accuracy	**0.5542**	**0.5462**
Sensitivity	**0.0**	**0.1441**
Specificity	**1.0**	**0.8696**
AUC	**0.5**	**0.5069**

The results showed in Table (4.2) shows that both models failed to classify the AMD images correctly. This could be justified that the architectures of these networks are not suitable for these types of images. Nonetheless, these architectures are proven to work very effectively in other types of images (such as cars, faces, etc.) [103][104][105].

4.5.2 Other researches

The second way to evaluate the results is to compare this approach finding with other researchers on the same dataset (ODIR). Summarized results are shown in Table (4.3).

Table (4.3): Comparison of the proposed method with other methods

Research Study	Approach	Parameters (M)	Sensitivity	Specificity	AUC	Accuracy
Ram and Reyes-Aldasoro, 2020 [11]	**CNN**	**-**	**0.714**	**0.859**	**-**	**0.819**
Gour and Khanna, 2021 [16]	**CNN**	**15.2**	**0.06**	**0.93**	**0.85**	**0.88**
He et al., 2021 [14]	**Dense Correlation Network (DCNet), CNN,**	**74.2**	**-**	**-**	**0.930**	**-**
He et al., 2020 [10]	**feature extraction module (FEM), feature fusion module (FFM), classification module**	**-**	**-**	**-**	**0.934**	**-**

	(CFM)					
Islam et al., 2019 [12]	**CNN**	**-**	**-**	**-**	**0.805**	**0.876**
Wang et al., 2020 [15]	**Ensemble model, CNN,**	**-**	**0.58**	**-**	**0.73**	**0.89**
Luo and Shen, 2020 [13]	**ResNet-50, U-Net model, Gaussian blur method, Ensemble Block**	**-**	**0.784**	**0.814**	**0.822**	**0.799**
Proposed method	**CNN**	**12.3**	**0.965**	**0.985**	**0.975**	**0.98**

The proposed methods by other researchers based on the same dataset, shows relatively lower performance than our model. Although that some of the proposed methods used group of different techniques and approaches together in their model.

4.6 Extended System

Since the original proposed system get high performance, the system classes are extended to include all three stages of AMD (Early, Intermediate, and Late), besides the Normal status of the eye. However, the results were not sufficient. So new extended system is proposed, which is based on the previous system (as discussed in the previous chapters), and new sufficient results are extracted.

4.6.1.1 Same system four classes: the results of the previous model with the four classes are showed in Figure (4.9) which is the confusion matrix, and in Table (4.4) which is the performance metrics.

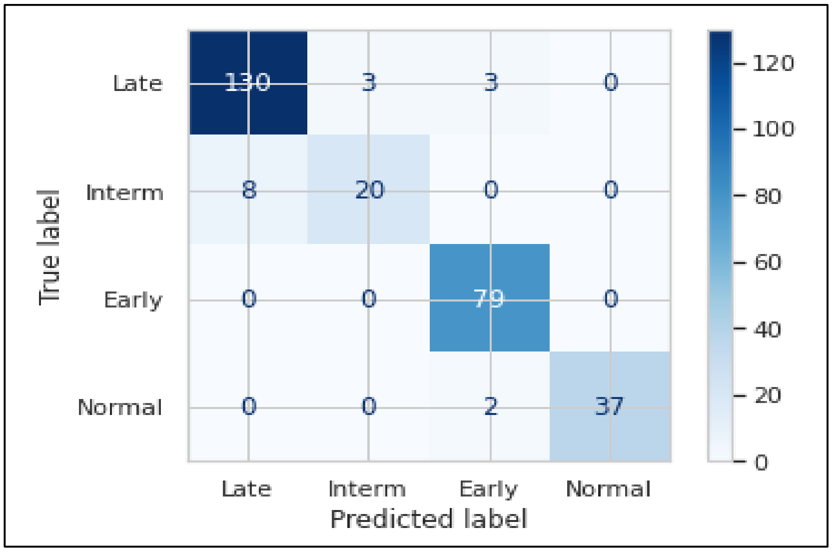


Figure (4.9): Confusion matrix for four classes' classification

Table (4.4): Performance metrics for four class's classification

Sensitivity	Specificity	AUC	Accuracy
0.977	**0.714**	**0.845**	**0.94**

The result shows good but not excellent performance. Therefore, the need to improve the current model is a necessity.

4.6.1.2 Extended system four classes: new extended network design and training process is developed in order to get better results.

Figure (4.10) shows the training process of the model during the different epochs compared with accuracy. While Figure (4.11) shows the training process of the model during the different epochs compared with loss function value.

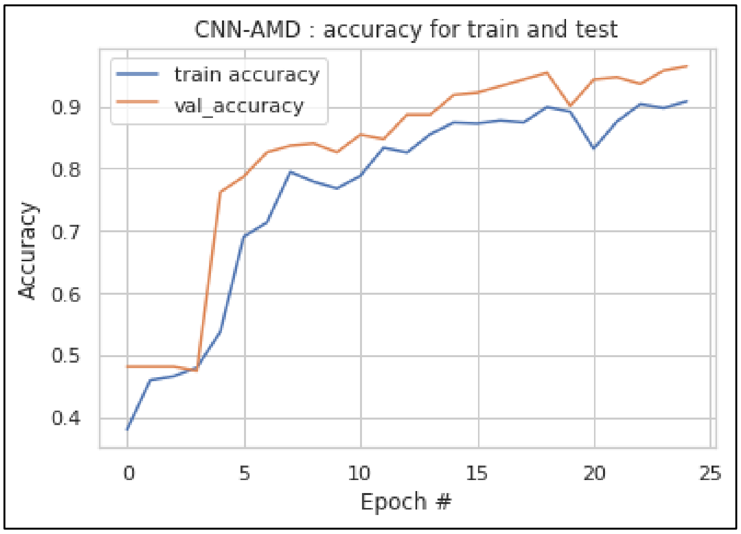


Figure (4.10): The accuracy of train and validation for four classes classification

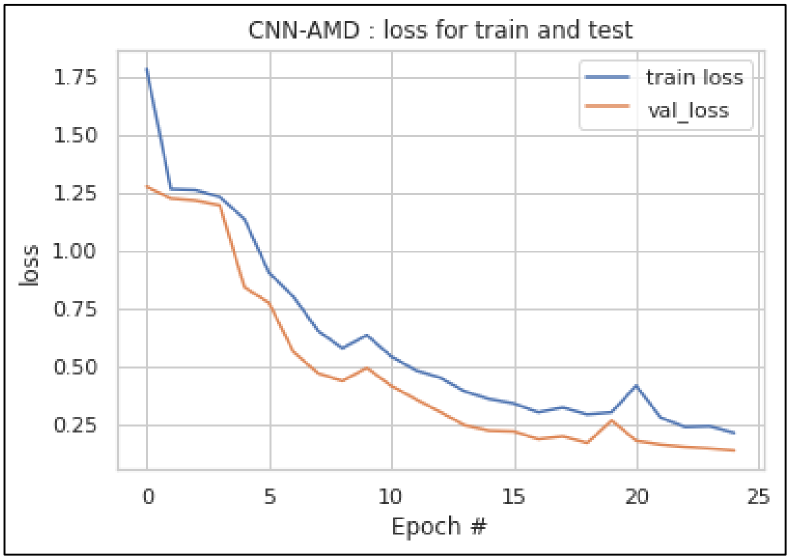


Figure (4.11): The loss of train and validation for four classes classification

The results of the new extended system are shown in Figure (4.12) as the confusion matrix, and Table (4.5) as the metrics performance.

Notice that the training process of four classes took (1237) seconds to train the model.

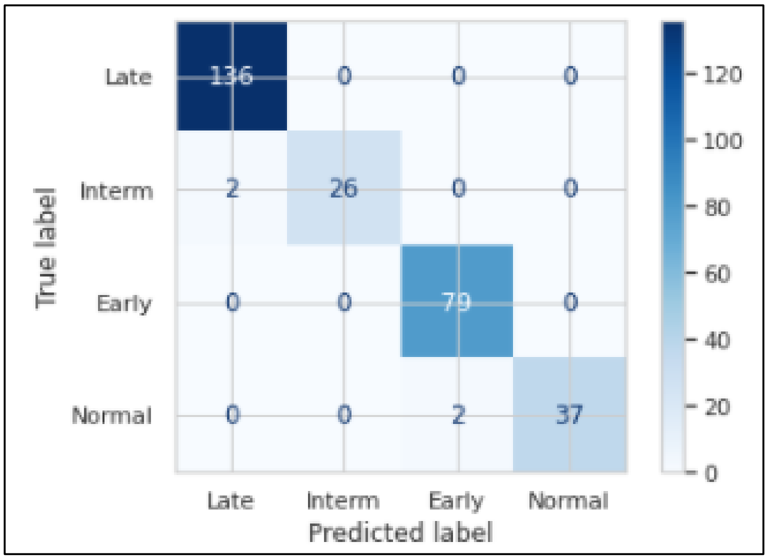


Figure (4.12): The confusion matrix for new proposed system on four classes classification

Table (4.5): Performance metrics for four classes classification in the extended system

Sensitivity	Specificity	AUC	Accuracy
1.00	0.928	0.964	0.985

4.7 Data Augmentation

In the Figure (4.13), sample of some original images before applying data augmentation. The observed effect of three common Data Augmentation techniques on the original training dataset only was presented, as follows:

1. **Horizontal and Vertical Flip:** You can flip or mirror an image so that the left side becomes the right side or the top becomes the bottom.

When you choose Flip, layers or selections are flipped in position. When you choose Mirror, layers or selections are flipped along their horizontal (left to right) or vertical (top to bottom) axis as shown in Figure (4.14). In addition, we can see the original images in figure (4.13) to compare with.

2. **Random Contrast:** contrast are modified randomly in pre-specific range which represent the lower and upper random contrast factor as shown in Figure (4.15).

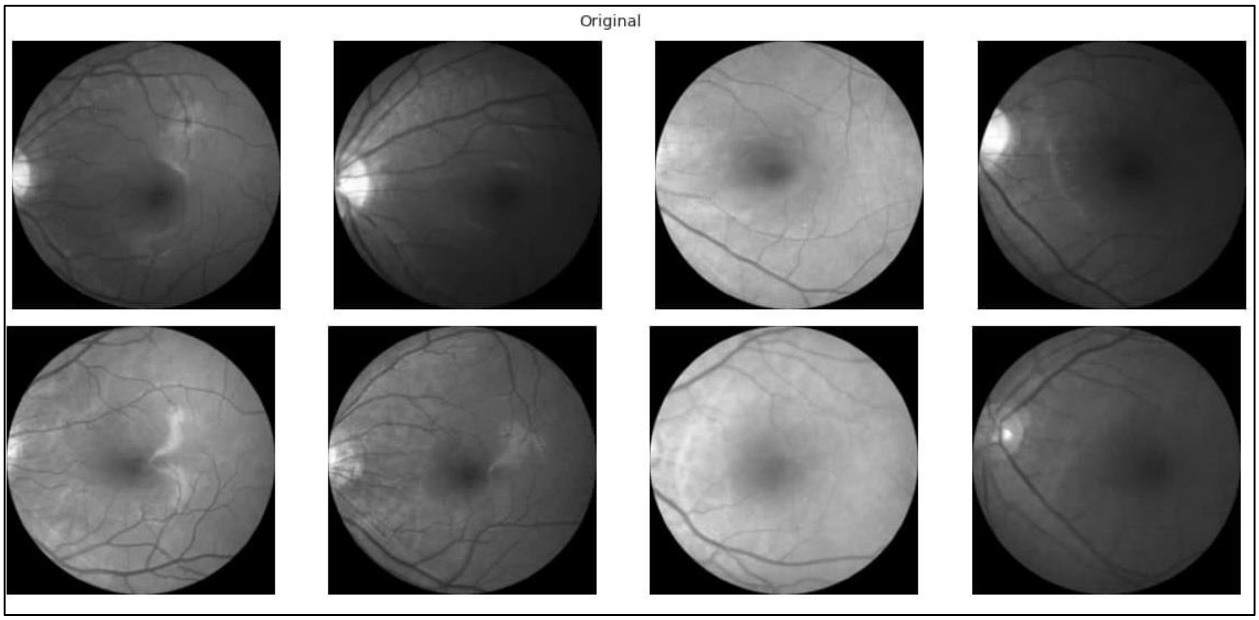


Figure (4.13): The original image sample before data augmentation

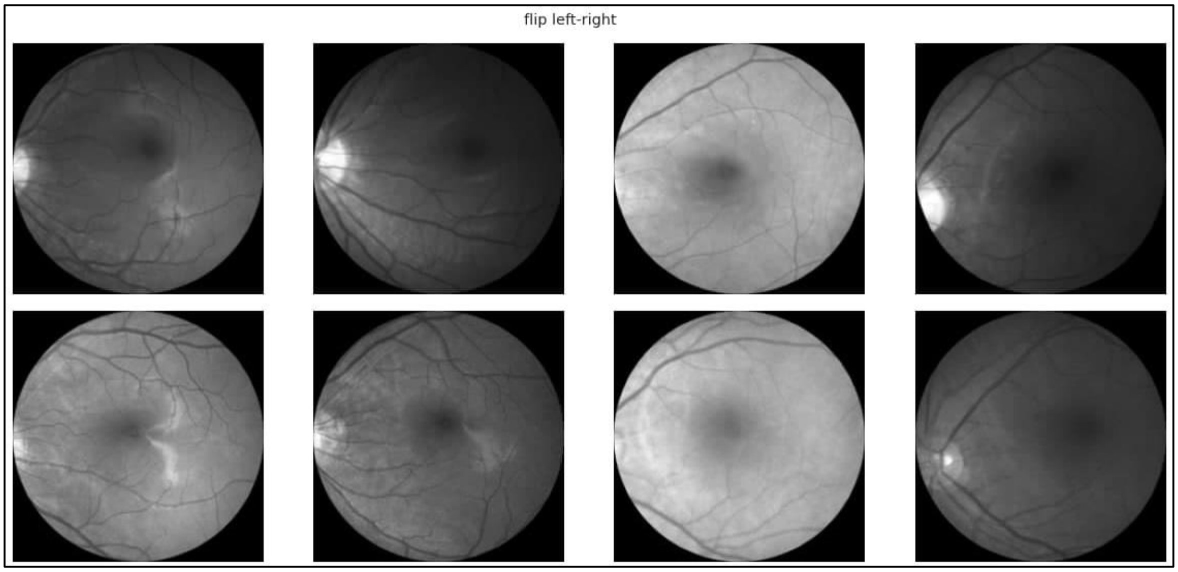


Figure (4.14): Horizontal and Vertical Flip

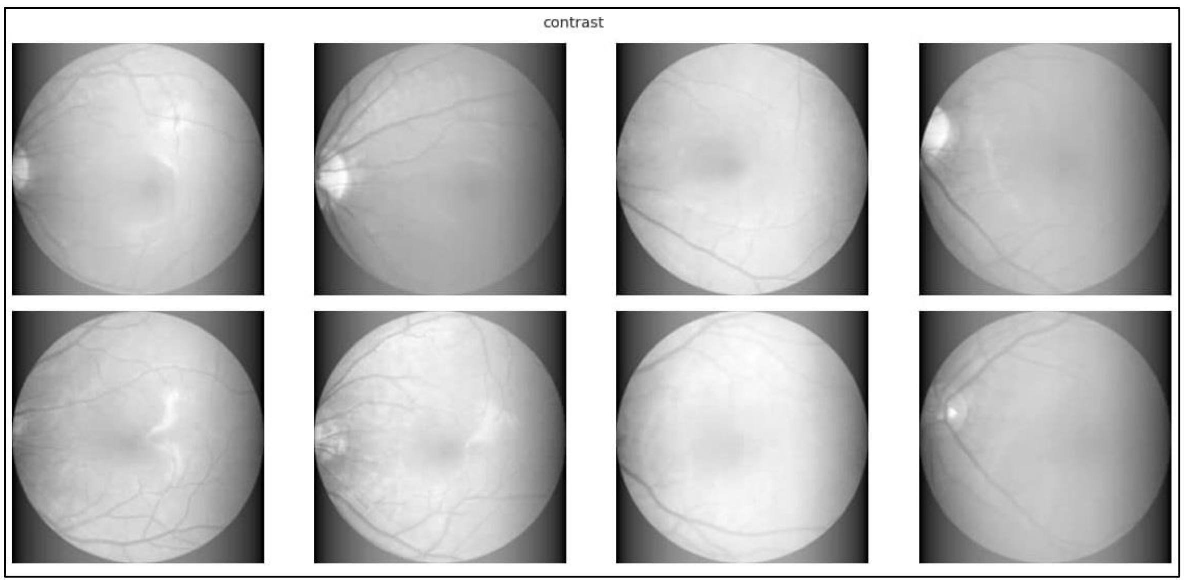


Figure (4.15): Random contrast

In order to check if our model could be improved any further, we used these two data augmentation techniques. Therefore, in case there is an overfitting problem with our model, or in case the training dataset are not sufficient to train the model in its best ability, there will be sufficient data to train our model.

However, the results after using the data augmentation techniques did not improve any further, comparing with the model training on the original dataset. This shows that our model does not suffer from the problem of overfitting. Moreover, its training gave the best results could be obtained from this dataset.

4.8 Case Study

In order to further validate the results of this study, real life images of AMD are tested on our proposed system. Real patients' data are collected from Ebsar Specialized Centre for Eyes, in collaboration with the medical doctors and technicians. While collecting these images, the full commitment of ethical codes for scientific research and data collecting. And the privacy of the patients is preserved.

The total number of 50 images are collected. 20 images with AMD condition and 30 images of normal and healthy retina.

The images are tested by our system. Figure (4.16) shows the confusion matrix of the case study. We can notice that the performance is almost perfect, with only one image is misclassified. Moreover, Table (4.6) shows different evaluation metrics of the results obtained from the results of the case study.

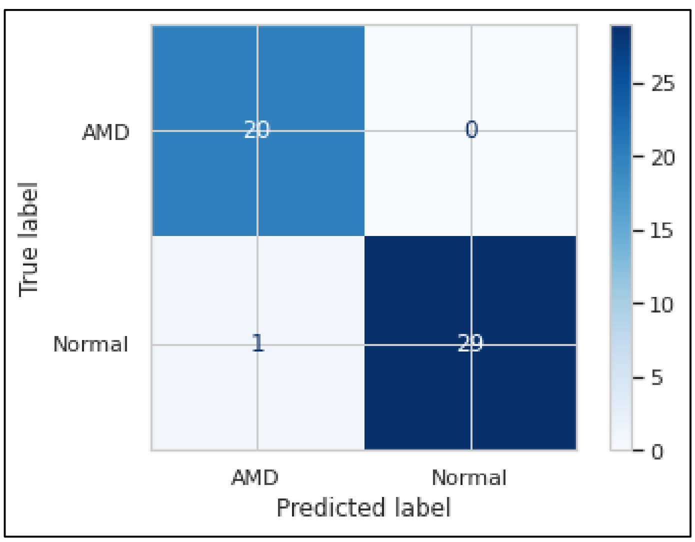


Figure (4.16): The confusion matrix of the case study

Table (4.6): Performance metrics of the case study

Sensitivity	Specificity	AUC	Accuracy
1.00	**0.966**	**0.983**	**0.98**

The whole process of reading these 50 images, preprocessing them, predict their class, calculate and plot the confusion matrix and evaluation metrics took (25) seconds and (273) mili seconds.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

Conclusion will be drawn from the developed system and the obtained results. Describing the main points on how the different parts of the system are developed and trained. And what results are drawn from it. These points are:

1. The proposed system tackle the problem of detecting AMD in fundus images of the eye as discussed in chapter 1.
2. The system was based on Deep Learning algorithm of Convolutional Neural Network (CNN). A unique CNN design and training parameters is developed based on intensive research and experiments conducted by the researchers.
3. The system was trained and tested to classify the images based on two classes, *Normal* and *AMD*. In which all the AMD images are summed together. The results of this system is as fellow: 0.965 sensitivity, 0.985 specificity, 0.98 accuracy, and 0.975 AUC.
4. The performance of this system is compared with others cutting edges researchers. Which showed the proposed system have higher performance than the previous researches.
5. AlexNet and VGG16 are trained and tested on the same dataset (ODIR). And the proposed system showed higher performance than both of them.
6. The same system is trained and tested on the original four classes (*Normal*, *Early*, *Intermediate*, and *Late*). The results were inefficient to consider it.
7. To tackle the previous performance limitation, the design and training of the proposed system is upgraded to include all the different classes. That led to increase in the performance of the system, with sufficient results of 1.0 sensitivity, 0.928 specifity, 0.985 accuracy, and 0.964 AUC. Comparing with other research on the same dataset, our model showed higher performance.

8. The real case study is carried out on real patients' images from the local vision centre, and high performance is achieved.

5.2 Future Works

The future work could include:

A. Develop the proposed system further to include more eye diseases (such as eye hypertension, cataract, glaucoma, and others) based on fundus images.

B. Develop a localization algorithm to identify the location of the features (i.e. drusens) used in the classification. R-CNN algorithms family could be utilized for this purpose.

C. Extend the suggested system's capabilities by using a big dataset of fundus images (thousands of photos).

REFRENCES

References

[1] "Age-related macular degeneration (AMD) - NHS," *NHS.UK*, 2021. https://www.nhs.uk/conditions/age-related-macular-degeneration-amd/ (accessed Jan. 27, 2021).

[2] C. A. Curcio, "Soft Drusen in Age-Related Macular Degeneration: Biology and Targeting Via the Oil Spill Strategies," *Investig. Opthalmology Vis. Sci.*, vol. 59, no. 4, p. AMD160, Oct. 2018, doi: 10.1167/iovs.18-24882.

[3] V. S. Makhijani, C. Ung, and D. Husain, "Dry Age-Related Macular Degeneration," in *Macular Disorders*, Springer, Singapore, 2020, pp. 1–12.

[4] W. L. Wong *et al.*, "Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis," *Lancet Glob. Heal.*, vol. 2, no. 2, pp. e106–e116, Feb. 2014, doi: 10.1016/S2214-109X(13)70145-1.

[5] P. Mitchell, G. Liew, B. Gopinath, and T. Y. Wong, "Age-related macular degeneration," *Lancet*, vol. 392, no. 10153, pp. 1147–1159, Sep. 2018, doi: 10.1016/S0140-6736(18)31550-2.

[6] Y. Kanagasingam, A. Bhuiyan, M. D. Abràmoff, R. T. Smith, L. Goldschmidt, and T. Y. Wong, "Progress on retinal image analysis for age related macular degeneration," *Prog. Retin. Eye Res.*, vol. 38, pp. 20–42, Jan. 2014, doi: 10.1016/j.preteyeres.2013.10.002.

[7] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[8] M. S. Nixon and A. S. Aguado, "Low-level feature extraction (including edge detection)," in *Feature Extraction & Image Processing for Computer Vision*, Elsevier, 2012, pp. 137–216.

[9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[10] J. He, C. Li, J. Ye, S. Wang, Y. Qiao, and L. Gu, "Classification of Ocular Diseases Employing Attention-Based Unilateral and Bilateral Feature Weighting and

Fusion," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 1258–1261, doi: 10.1109/ISBI45749.2020.9098525.

[11] A. Ram and C. C. Reyes-Aldasoro, "The relationship between Fully Connected Layers and number of classes for the analysis of retinal images," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.03624.

[12] M. T. Islam, S. A. Imran, A. Arefeen, M. Hasan, and C. Shahnaz, "Source and Camera Independent Ophthalmic Disease Recognition from Fundus Image Using Neural Network," in *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON)*, Nov. 2019, pp. 59–63, doi: 10.1109/SPICSCON48833.2019.9065162.

[13] D. Luo and L. Shen, "Vessel-Net: A Vessel-Aware Ensemble Network For Retinopathy Screening From Fundus Image," in *2020 IEEE International Conference on Image Processing (ICIP)*, Oct. 2020, pp. 320–324, doi: 10.1109/ICIP40778.2020.9190800.

[14] J. He, C. Li, J. Ye, Y. Qiao, and L. Gu, "Multi-label ocular disease classification with a dense correlation deep neural network," *Biomed. Signal Process. Control*, vol. 63, p. 102167, Jan. 2021, doi: 10.1016/j.bspc.2020.102167.

[15] J. Wang, L. Yang, Z. Huo, W. He, and J. Luo, "Multi-Label Classification of Fundus Images With EfficientNet," *IEEE Access*, vol. 8, pp. 212499–212508, 2020, doi: 10.1109/ACCESS.2020.3040275.

[16] N. Gour and P. Khanna, "Multi-class multi-label ophthalmological disease detection using transfer learning based convolutional neural network," *Biomed. Signal Process. Control*, vol. 66, p. 102329, Apr. 2021, doi: 10.1016/j.bspc.2020.102329.

[17] J. H. Tan *et al.*, "Age-related Macular Degeneration detection using deep convolutional neural network," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 127–135, Oct. 2018, doi: 10.1016/j.future.2018.05.001.

[18] H. Liu, D. W. K. Wong, H. Fu, Y. Xu, and J. Liu, "DeepAMD: Detect Early Age-Related Macular Degeneration by Applying Deep Learning in a Multiple Instance Learning Framework," 2019, pp. 625–640.

[19] Q. Yan *et al.*, "Deep-learning-based prediction of late age-related macular degeneration progression," *Nat. Mach. Intell.*, vol. 2, no. 2, pp. 141–150, Feb.

2020, doi: 10.1038/s42256-020-0154-9.

[20] A. García-Floriano, Á. Ferreira-Santiago, O. Camacho-Nieto, and C. Yáñez-Márquez, “A machine learning approach to medical image classification: Detecting age-related macular degeneration in fundus images,” *Comput. Electr. Eng.*, vol. 75, pp. 218–229, May 2019, doi: 10.1016/j.compeleceng.2017.11.008.

[21] T. Mitchell, *Machine Learning*, 1st ed. McGraw Hill, 1997.

[22] L. S. Lim, P. Mitchell, J. M. Seddon, F. G. Holz, and T. Y. Wong, “Age-related macular degeneration,” *Lancet*, vol. 379, no. 9827, pp. 1728–1738, May 2012, doi: 10.1016/S0140-6736(12)60282-7.

[23] A. Stahl, “The Diagnosis and Treatment of Age-Related Macular Degeneration,” *Dtsch. Aerzteblatt Online*, Jul. 2020, doi: 10.3238/arztebl.2020.0513.

[24] S. Mehta, “Age-Related Macular Degeneration (AMD or ARMD),” *MSD Manual*, 2020. https://www.msdmanuals.com/professional/eye-disorders/retinal-disorders/age-related-macular-degeneration-amd-or-armd (accessed May 20, 2020).

[25] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. MIT Press, 2014.

[26] “Machine Learning - Categories.” https://www.hebergementwebs.com/machine-learning-tutorial/machine-learning-categories.

[27] R. Gentleman and V. J. Carey, “Unsupervised Machine Learning,” in *Bioconductor Case Studies*, New York, NY: Springer New York, 2008, pp. 137–157.

[28] X. Zhu and A. B. Goldberg, “Introduction to Semi-Supervised Learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, Jan. 2009, doi: 10.2200/S00196ED1V01Y200906AIM006.

[29] C. Szepesvári, “Algorithms for Reinforcement Learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, Jan. 2010, doi: 10.2200/S00268ED1V01Y201005AIM009.

[30] F. Shaheen, B. Verma, and M. Asafuddoula, “Impact of Automatic Feature Extraction in Deep Learning Architecture,” in *2016 International Conference on*

Digital Image Computing: Techniques and Applications (DICTA), Nov. 2016, pp. 1–8, doi: 10.1109/DICTA.2016.7797053.

[31] John D. Kelleher, "Introduction to Deep Learning," in *Deep Learning*, The MIT Press Essential Knowledge Series, 2019, pp. 1–37.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[33] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[34] C. Baral, O. Fuentes, and V. Kreinovich, "Why Deep Neural Networks: A Possible Theoretical Explanation," 2018, pp. 1–5.

[35] I. Goodfellow, Y. Bengio, and A. Courville, "Convolutional Networks," in *Deep Learning*, MIT Press, 2016, pp. 321–366.

[36] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

[37] IBM Cloud Education, "Convolutional Neural Networks," *IBM Cloud Learn Hub*, 2020. https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[38] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, "Continuous Optimization," in *Mathematics for Machine Learning*, Cambridge University Press, 2020, pp. 225–227.

[39] Zhou Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009, doi: 10.1109/MSP.2008.930649.

[40] M. Eppert, P. Fent, and T. Neumann, "A Tailored Regression for Learned Indexes: Logarithmic Error Regression," in *Fourth Workshop in Exploiting AI Techniques for Data Management*, Jun. 2021, pp. 9–15, doi: 10.1145/3464509.3464891.

[41] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 324, p. 012049, Mar. 2018, doi: 10.1088/1757-899X/324/1/012049.

[42] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Deconstructing Cross-Entropy for Probabilistic Binary Classifiers," *Entropy*, vol. 20, no. 3, p. 208, Mar. 2018, doi: 10.3390/e20030208.

[43] A. Demirkaya, J. Chen, and S. Oymak, "Exploring the Role of Loss Functions in Multiclass Classification," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2020, pp. 1–5, doi: 10.1109/CISS48834.2020.1570627167.

[44] A. Semenov, V. Boginski, and E. L. Pasiliao, "Neural Networks with Multidimensional Cross-Entropy Loss Functions," 2019, pp. 57–62.

[45] S. Suresh, N. Sundararajan, and P. Saratchandran, "Risk-sensitive loss functions for sparse multi-category classification problems," *Inf. Sci. (Ny).*, vol. 178, no. 12, pp. 2621–2638, Jun. 2008, doi: 10.1016/j.ins.2008.02.009.

[46] M. Togami, Y. Masuyama, T. Komatsu, and Y. Nakagome, "Unsupervised Training for Deep Speech Source Separation with Kullback-Leibler Divergence Based Probabilistic Loss Function," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 56–60, doi: 10.1109/ICASSP40776.2020.9054171.

[47] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 12, pp. 310–316, 2020.

[48] S. Ruder, "An overview of gradient descent optimization algorithms," Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

[49] N. Ketkar, "Stochastic Gradient Descent," in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 113–132.

[50] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 2880–2887, doi: 10.1109/CDC.2017.8264077.

[51] R. N. Singarimbun, E. B. Nababan, and O. S. Sitompul, "Adaptive Moment Estimation To Minimize Square Error In Backpropagation Algorithm," in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, Nov. 2019, pp. 1–7, doi: 10.1109/ICoSNIKOM48755.2019.9111563.

[52] N.-D. Hoang, “Image Processing-Based Spall Object Detection Using Gabor Filter, Texture Analysis, and Adaptive Moment Estimation (Adam) Optimized Logistic Regression Models,” *Adv. Civ. Eng.*, vol. 2020, pp. 1–16, Nov. 2020, doi: 10.1155/2020/8829715.

[53] D. Mack, “How to pick the best learning rate for your machine learning project,” 2018. https://medium.com/octavian-ai/which-optimizer-and-learning-rate-should-i-use-for-deep-learning-5acb418f9b2 (accessed Jan. 15, 2021).

[54] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2015.

[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” Jul. 2012, [Online]. Available: http://arxiv.org/abs/1207.0580.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014, [Online]. Available: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

[58] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dropout,” in *Dive into Deep Learning*, 2020.

[59] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, “Regularization of Neural Networks using DropConnect,” in *30th International Conference on Machine Learning*, 2013, pp. 1058–1066, [Online]. Available: http://proceedings.mlr.press/v28/wan13.html.

[60] Y. Yao, L. Rosasco, and A. Caponnetto, “On Early Stopping in Gradient Descent Learning,” *Constr. Approx.*, vol. 26, no. 2, pp. 289–315, Aug. 2007, doi: 10.1007/s00365-006-0663-2.

[61] R. Sanghvi, “Early Stopping with Neptune,” 2021. https://neptune.ai/blog/early-stopping-with-neptune (accessed Apr. 15, 2021).

[62] Y. Lecun, “Generalization and network design strategies,” 1989.

[63] Sam, “Why convolutional neural networks belong to deep learning?,” 2015. https://stats.stackexchange.com/questions/146413/why-convolutional-neural-networks-belong-to-deep-learning.

[64] I. Goodfellow, Y. Bengio, and A. Courville, “The Convolution Operation,” in *Deep Learning*, MIT Press, 2015, pp. 322–330.

[65] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach,” *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[66] I. Goodfellow, Y. Bengio, and A. Courville, “Pooling,” in *Deep Learning*, MIT Press, 2016, pp. 330–334.

[67] J. Jin, A. Dundar, and E. Culurciello, “Flattened Convolutional Neural Networks for Feedforward Acceleration,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.5474.

[68] C.-L. Zhang, J.-H. Luo, X.-S. Wei, and J. Wu, “In Defense of Fully Connected Layers in Visual Representation Transfer,” in *Advances in Multimedia Information Processing – PCM 2017*, 2018, pp. 807–817, doi: 10.1007/978-3-319-77383-4_79.

[69] M. Peemen, B. Mesman, and H. Corporaal, “Efficiency optimization of trainable feature extractors for a consumer platform,” in *ACIVS’11: Proceedings of the 13th international conference on Advanced concepts for intelligent vision systems*, 2011, pp. 293–304.

[70] K. Audhkhasi, O. Osoba, and B. Kosko, “Noise-enhanced convolutional neural networks,” *Neural Networks*, vol. 78, pp. 15–23, Jun. 2016, doi: 10.1016/j.neunet.2015.09.014.

[71] H. Guo, H. Nguyen, D.-A. Vu, and X.-N. Bui, “Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach,” *Resour. Policy*, p. 101474, Aug. 2019, doi: 10.1016/j.resourpol.2019.101474.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[73] A. Patino-Saucedo, H. Rostro-Gonzalez, and J. Conradt, “Tropical Fruits

Classification Using an AlexNet-Type Convolutional Neural Network and Image Augmentation," in *International Conference on Neural Information Processing*, 2018, pp. 371–379, doi: 10.1007/978-3-030-04212-7_32.

[74] A. Vedaldi and A. Zisserman, "VGG Convolutional Neural Networks Practical," *Oxford Visual Geometry Group*, 2017. https://www.robots.ox.ac.uk/~vgg/practicals/cnn/index.html#part-31-traini... (accessed Feb. 11, 2021).

[75] P. HUILGOL, "Top 4 Pre-Trained Models for Image Classification with Python Code," 2020. https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/ (accessed Feb. 12, 2021).

[76] M. DASH and H. LIU, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 1–4, pp. 131–156, 1997, doi: 10.1016/S1088-467X(97)00008-5.

[77] A. Atla, R. Tada, V. Sheng, and N. Singireddy, "Sensitivity of different machine learning algorithms to noise," *J. Comput. Sci. Coll.*, vol. 26, no. 5, pp. 96–103, 2011.

[78] D. Garcia-Gasulla *et al.*, "On the Behavior of Convolutional Nets for Feature Extraction," *J. Artif. Intell. Res.*, vol. 61, pp. 563–592, Mar. 2018, doi: 10.1613/jair.5756.

[79] R. Priya and P. Aruna, "Automated diagnosis of age-related macular degeneration using machine learning techniques," *Int. J. Comput. Appl. Technol.*, vol. 49, no. 2, p. 157, 2014, doi: 10.1504/IJCAT.2014.060527.

[80] M. R. K. Mookiah *et al.*, "Automated diagnosis of Age-related Macular Degeneration using greyscale features from digital fundus images," *Comput. Biol. Med.*, vol. 53, pp. 55–64, Oct. 2014, doi: 10.1016/j.compbiomed.2014.07.015.

[81] M. R. K. Mookiah *et al.*, "Decision support system for age-related macular degeneration using discrete wavelet transform," *Med. Biol. Eng. Comput.*, vol. 52, no. 9, pp. 781–796, Sep. 2014, doi: 10.1007/s11517-014-1180-8.

[82] S. Matsuba *et al.*, "Accuracy of ultra-wide-field fundus ophthalmoscopy-assisted deep learning, a machine-learning technology, for detecting age-related macular degeneration," *Int. Ophthalmol.*, vol. 39, no. 6, pp. 1269–1275,

Jun. 2019, doi: 10.1007/s10792-018-0940-0.

[83] S. Keel *et al.*, “Development and validation of a deep-learning algorithm for the detection of neovascular age-related macular degeneration from colour fundus photographs,” *Clin. Experiment. Ophthalmol.*, vol. 47, no. 8, pp. 1009–1018, Nov. 2019, doi: 10.1111/ceo.13575.

[84] H. Müller, J. Kalpathy-Cramer, D. Demner-Fushman, and S. Antani, “Creating a classification of image types in the medical literature for visual categorization,” Feb. 2012, p. 83190P, doi: 10.1117/12.911186.

[85] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” *Vis. Comput. Ind. Biomed. Art*, vol. 2, no. 1, p. 7, Dec. 2019, doi: 10.1186/s42492-019-0016-7.

[86] J. Azzeh, B. Zahran, and Z. Alqadi, “Salt and Pepper Noise: Effects and Removal,” *JOIV Int. J. Informatics Vis.*, vol. 2, no. 4, p. 252, Jul. 2018, doi: 10.30630/joiv.2.4.151.

[87] F. Luisier, T. Blu, and M. Unser, “Image Denoising in Mixed Poisson–Gaussian Noise,” *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, Mar. 2011, doi: 10.1109/TIP.2010.2073477.

[88] Jian-Jia Pan, Yuan-Yan Tang, and Bao-Chang Pan, “The algorithm of fast mean filtering,” in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, Nov. 2007, pp. 244–248, doi: 10.1109/ICWAPR.2007.4420672.

[89] L. Li, Y. Si, and Z. Jia, “Medical Image Enhancement Based on CLAHE and Unsharp Masking in NSCT Domain,” *J. Med. Imaging Heal. Informatics*, vol. 8, no. 3, pp. 431–438, Mar. 2018, doi: 10.1166/jmihi.2018.2328.

[90] Sonali, S. Sahu, A. K. Singh, S. P. Ghrera, and M. Elhoseny, “An approach for de-noising and contrast enhancement of retinal fundus image using CLAHE,” *Opt. Laser Technol.*, vol. 110, pp. 87–98, Feb. 2019, doi: 10.1016/j.optlastec.2018.06.061.

[91] K.-M. Koo and E.-Y. Cha, “Image recognition performance enhancements using image normalization,” *Human-centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 33, Dec. 2017, doi: 10.1186/s13673-017-0114-5.

[92] C. Liu, “Data Transformation: Standardization vs Normalization,” *kd nuggets*, 2020. https://www.kdnuggets.com/2020/04/data-transformation-

standardization-normalization.html (accessed Sep. 27, 2020).

[93] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[94] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[95] M. Nisa *et al.*, “Hybrid Malware Classification Method Using Segmentation-Based Fractal Texture Analysis and Deep Convolution Neural Network Features,” *Appl. Sci.*, vol. 10, no. 14, p. 4966, Jul. 2020, doi: 10.3390/app10144966.

[96] FeanDoe, “Sensitivity and specificity,” 2018. https://commons.wikimedia.org/wiki/File:Sensitivity_and_specificity.svg (accessed Mar. 15, 2021).

[97] “Classification: ROC Curve and AUC,” *Google Developers*, 2020. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (accessed Apr. 14, 2021).

[98] “Evaluating Classification Models,” in *Practical Statistics for Data Scientists*, 2nd ed., O’Reilly, 2020, pp. 219–229.

[99] “Peking University International Competition on Ocular Disease Intelligent Recognition (ODIR-2019),” *Peking University*, 2019. https://odir2019.grand-challenge.org/introduction/ (accessed Dec. 12, 2020).

[100] “Color conversions,” *OpenCV*, 2015. https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html (accessed May 17, 2021).

[101] S. Sun, Z. Cao, H. Zhu, and J. Zhao, “A Survey of Optimization Methods From a Machine Learning Perspective,” *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020, doi: 10.1109/TCYB.2019.2950779.

[102] V. Bushaev, “Adam — latest trends in deep learning optimization,” 2018. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c (accessed Apr. 15, 2021).

[103] J. E. Espinosa, S. A. Velastin, and J. W. Branch, “Vehicle Detection Using Alex

Net and Faster R-CNN Deep Learning Models: A Comparative Study," 2017, pp. 3–15.

[104] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.01164.

[105] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and Benchmarking of Machine Learning Accelerators," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–9, doi: 10.1109/HPEC.2019.8916327.

[1] "Age-related macular degeneration (AMD) - NHS," *NHS.UK*, 2021. https://www.nhs.uk/conditions/age-related-macular-degeneration-amd/ (accessed Jan. 27, 2021).

[2] C. A. Curcio, "Soft Drusen in Age-Related Macular Degeneration: Biology and Targeting Via the Oil Spill Strategies," *Investig. Opthalmology Vis. Sci.*, vol. 59, no. 4, p. AMD160, Oct. 2018, doi: 10.1167/iovs.18-24882.

[3] V. S. Makhijani, C. Ung, and D. Husain, "Dry Age-Related Macular Degeneration," in *Macular Disorders*, Springer, Singapore, 2020, pp. 1–12.

[4] W. L. Wong *et al.*, "Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis," *Lancet Glob. Heal.*, vol. 2, no. 2, pp. e106–e116, Feb. 2014, doi: 10.1016/S2214-109X(13)70145-1.

[5] P. Mitchell, G. Liew, B. Gopinath, and T. Y. Wong, "Age-related macular degeneration," *Lancet*, vol. 392, no. 10153, pp. 1147–1159, Sep. 2018, doi: 10.1016/S0140-6736(18)31550-2.

[6] Y. Kanagasingam, A. Bhuiyan, M. D. Abràmoff, R. T. Smith, L. Goldschmidt, and T. Y. Wong, "Progress on retinal image analysis for age related macular degeneration," *Prog. Retin. Eye Res.*, vol. 38, pp. 20–42, Jan. 2014, doi: 10.1016/j.preteyeres.2013.10.002.

[7] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[8] M. S. Nixon and A. S. Aguado, “Low-level feature extraction (including edge detection),” in *Feature Extraction & Image Processing for Computer Vision*, Elsevier, 2012, pp. 137–216.

[9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[10] J. He, C. Li, J. Ye, S. Wang, Y. Qiao, and L. Gu, “Classification of Ocular Diseases Employing Attention-Based Unilateral and Bilateral Feature Weighting and Fusion,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 1258–1261, doi: 10.1109/ISBI45749.2020.9098525.

[11] A. Ram and C. C. Reyes-Aldasoro, “The relationship between Fully Connected Layers and number of classes for the analysis of retinal images,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.03624.

[12] M. T. Islam, S. A. Imran, A. Arefeen, M. Hasan, and C. Shahnaz, “Source and Camera Independent Ophthalmic Disease Recognition from Fundus Image Using Neural Network,” in *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON)*, Nov. 2019, pp. 59–63, doi: 10.1109/SPICSCON48833.2019.9065162.

[13] D. Luo and L. Shen, “Vessel-Net: A Vessel-Aware Ensemble Network For Retinopathy Screening From Fundus Image,” in *2020 IEEE International Conference on Image Processing (ICIP)*, Oct. 2020, pp. 320–324, doi: 10.1109/ICIP40778.2020.9190800.

[14] J. He, C. Li, J. Ye, Y. Qiao, and L. Gu, “Multi-label ocular disease classification with a dense correlation deep neural network,” *Biomed. Signal Process. Control*, vol. 63, p. 102167, Jan. 2021, doi: 10.1016/j.bspc.2020.102167.

[15] J. Wang, L. Yang, Z. Huo, W. He, and J. Luo, “Multi-Label Classification of Fundus Images With EfficientNet,” *IEEE Access*, vol. 8, pp. 212499–212508, 2020, doi: 10.1109/ACCESS.2020.3040275.

[16] N. Gour and P. Khanna, “Multi-class multi-label ophthalmological disease detection using transfer learning based convolutional neural network,” *Biomed. Signal Process. Control*, vol. 66, p. 102329, Apr. 2021, doi: 10.1016/j.bspc.2020.102329.

[17] J. H. Tan *et al.*, “Age-related Macular Degeneration detection using deep convolutional neural network,” *Futur. Gener. Comput. Syst.*, vol. 87, pp. 127–135, Oct. 2018, doi: 10.1016/j.future.2018.05.001.

[18] H. Liu, D. W. K. Wong, H. Fu, Y. Xu, and J. Liu, “DeepAMD: Detect Early Age-Related Macular Degeneration by Applying Deep Learning in a Multiple Instance Learning Framework,” 2019, pp. 625–640.

[19] Q. Yan *et al.*, “Deep-learning-based prediction of late age-related macular degeneration progression,” *Nat. Mach. Intell.*, vol. 2, no. 2, pp. 141–150, Feb. 2020, doi: 10.1038/s42256-020-0154-9.

[20] A. García-Floriano, Á. Ferreira-Santiago, O. Camacho-Nieto, and C. Yáñez-Márquez, “A machine learning approach to medical image classification: Detecting age-related macular degeneration in fundus images,” *Comput. Electr. Eng.*, vol. 75, pp. 218–229, May 2019, doi: 10.1016/j.compeleceng.2017.11.008.

[21] T. Mitchell, *Machine Learning*, 1st ed. McGraw Hill, 1997.

[22] L. S. Lim, P. Mitchell, J. M. Seddon, F. G. Holz, and T. Y. Wong, “Age-related macular degeneration,” *Lancet*, vol. 379, no. 9827, pp. 1728–1738, May 2012, doi: 10.1016/S0140-6736(12)60282-7.

[23] A. Stahl, “The Diagnosis and Treatment of Age-Related Macular Degeneration,” *Dtsch. Aerzteblatt Online*, Jul. 2020, doi: 10.3238/arztebl.2020.0513.

[24] S. Mehta, “Age-Related Macular Degeneration (AMD or ARMD),” *MSD Manual*, 2020. https://www.msdmanuals.com/professional/eye-disorders/retinal-disorders/age-related-macular-degeneration-amd-or-armd (accessed May 20, 2020).

[25] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. MIT Press, 2014.

[26] “Machine Learning - Categories.” https://www.hebergementwebs.com/machine-learning-tutorial/machine-learning-categories.

[27] R. Gentleman and V. J. Carey, “Unsupervised Machine Learning,” in *Bioconductor Case Studies*, New York, NY: Springer New York, 2008, pp. 137–157.

[28] X. Zhu and A. B. Goldberg, “Introduction to Semi-Supervised Learning,”

Synth. Lect. Artif. Intell. Mach. Learn., vol. 3, no. 1, pp. 1–130, Jan. 2009, doi: 10.2200/S00196ED1V01Y200906AIM006.

[29] C. Szepesvári, "Algorithms for Reinforcement Learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, Jan. 2010, doi: 10.2200/S00268ED1V01Y201005AIM009.

[30] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of Automatic Feature Extraction in Deep Learning Architecture," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov. 2016, pp. 1–8, doi: 10.1109/DICTA.2016.7797053.

[31] John D. Kelleher, "Introduction to Deep Learning," in *Deep Learning*, The MIT Press Essential Knowledge Series, 2019, pp. 1–37.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[33] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[34] C. Baral, O. Fuentes, and V. Kreinovich, "Why Deep Neural Networks: A Possible Theoretical Explanation," 2018, pp. 1–5.

[35] I. Goodfellow, Y. Bengio, and A. Courville, "Convolutional Networks," in *Deep Learning*, MIT Press, 2016, pp. 321–366.

[36] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

[37] IBM Cloud Education, "Convolutional Neural Networks," *IBM Cloud Learn Hub*, 2020. https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[38] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, "Continuous Optimization," in *Mathematics for Machine Learning*, Cambridge University Press, 2020, pp. 225–227.

[39] Zhou Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009, doi: 10.1109/MSP.2008.930649.

[40] M. Eppert, P. Fent, and T. Neumann, "A Tailored Regression for Learned Indexes: Logarithmic Error Regression," in *Fourth Workshop in Exploiting AI*

Techniques for Data Management, Jun. 2021, pp. 9–15, doi: 10.1145/3464509.3464891.

[41] W. Wang and Y. Lu, “Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 324, p. 012049, Mar. 2018, doi: 10.1088/1757-899X/324/1/012049.

[42] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez, “Deconstructing Cross-Entropy for Probabilistic Binary Classifiers,” *Entropy*, vol. 20, no. 3, p. 208, Mar. 2018, doi: 10.3390/e20030208.

[43] A. Demirkaya, J. Chen, and S. Oymak, “Exploring the Role of Loss Functions in Multiclass Classification,” in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2020, pp. 1–5, doi: 10.1109/CISS48834.2020.1570627167.

[44] A. Semenov, V. Boginski, and E. L. Pasiliao, “Neural Networks with Multidimensional Cross-Entropy Loss Functions,” 2019, pp. 57–62.

[45] S. Suresh, N. Sundararajan, and P. Saratchandran, “Risk-sensitive loss functions for sparse multi-category classification problems,” *Inf. Sci. (Ny).*, vol. 178, no. 12, pp. 2621–2638, Jun. 2008, doi: 10.1016/j.ins.2008.02.009.

[46] M. Togami, Y. Masuyama, T. Komatsu, and Y. Nakagome, “Unsupervised Training for Deep Speech Source Separation with Kullback-Leibler Divergence Based Probabilistic Loss Function,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 56–60, doi: 10.1109/ICASSP40776.2020.9054171.

[47] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 12, pp. 310–316, 2020.

[48] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

[49] N. Ketkar, “Stochastic Gradient Descent,” in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 113–132.

[50] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, “Mini-batch gradient descent: Faster convergence under data sparsity,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 2880–2887, doi: 10.1109/CDC.2017.8264077.

[51] R. N. Singarimbun, E. B. Nababan, and O. S. Sitompul, “Adaptive Moment Estimation To Minimize Square Error In Backpropagation Algorithm,” in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, Nov. 2019, pp. 1–7, doi: 10.1109/ICoSNIKOM48755.2019.9111563.

[52] N.-D. Hoang, “Image Processing-Based Spall Object Detection Using Gabor Filter, Texture Analysis, and Adaptive Moment Estimation (Adam) Optimized Logistic Regression Models,” *Adv. Civ. Eng.*, vol. 2020, pp. 1–16, Nov. 2020, doi: 10.1155/2020/8829715.

[53] D. Mack, “How to pick the best learning rate for your machine learning project,” 2018. https://medium.com/octavian-ai/which-optimizer-and-learning-rate-should-i-use-for-deep-learning-5acb418f9b2 (accessed Jan. 15, 2021).

[54] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2015.

[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” Jul. 2012, [Online]. Available: http://arxiv.org/abs/1207.0580.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014, [Online]. Available: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

[58] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dropout,” in *Dive into Deep Learning*, 2020.

[59] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, “Regularization of Neural Networks using DropConnect,” in *30th International Conference on Machine Learning*, 2013, pp. 1058–1066, [Online]. Available: http://proceedings.mlr.press/v28/wan13.html.

[60] Y. Yao, L. Rosasco, and A. Caponnetto, “On Early Stopping in Gradient Descent Learning,” *Constr. Approx.*, vol. 26, no. 2, pp. 289–315, Aug. 2007, doi: 10.1007/s00365-006-0663-2.

[61] R. Sanghvi, “Early Stopping with Neptune,” 2021. https://neptune.ai/blog/early-stopping-with-neptune (accessed Apr. 15, 2021).

[62] Y. Lecun, “Generalization and network design strategies,” 1989.

[63] Sam, “Why convolutional neural networks belong to deep learning?,” 2015. https://stats.stackexchange.com/questions/146413/why-convolutional-neural-networks-belong-to-deep-learning.

[64] I. Goodfellow, Y. Bengio, and A. Courville, “The Convolution Operation,” in *Deep Learning*, MIT Press, 2015, pp. 322–330.

[65] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach,” *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[66] I. Goodfellow, Y. Bengio, and A. Courville, “Pooling,” in *Deep Learning*, MIT Press, 2016, pp. 330–334.

[67] J. Jin, A. Dundar, and E. Culurciello, “Flattened Convolutional Neural Networks for Feedforward Acceleration,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.5474.

[68] C.-L. Zhang, J.-H. Luo, X.-S. Wei, and J. Wu, “In Defense of Fully Connected Layers in Visual Representation Transfer,” in *Advances in Multimedia Information Processing – PCM 2017*, 2018, pp. 807–817, doi: 10.1007/978-3-319-77383-4_79.

[69] M. Peemen, B. Mesman, and H. Corporaal, “Efficiency optimization of trainable feature extractors for a consumer platform,” in *ACIVS’11: Proceedings of the 13th international conference on Advanced concepts for intelligent vision systems*, 2011, pp. 293–304.

[70] K. Audhkhasi, O. Osoba, and B. Kosko, “Noise-enhanced convolutional neural networks,” *Neural Networks*, vol. 78, pp. 15–23, Jun. 2016, doi: 10.1016/j.neunet.2015.09.014.

[71] H. Guo, H. Nguyen, D.-A. Vu, and X.-N. Bui, “Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach,” *Resour. Policy*, p. 101474, Aug. 2019, doi: 10.1016/j.resourpol.2019.101474.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[73] A. Patino-Saucedo, H. Rostro-Gonzalez, and J. Conradt, “Tropical Fruits

Classification Using an AlexNet-Type Convolutional Neural Network and Image Augmentation," in *International Conference on Neural Information Processing*, 2018, pp. 371–379, doi: 10.1007/978-3-030-04212-7_32.

[74] A. Vedaldi and A. Zisserman, "VGG Convolutional Neural Networks Practical," *Oxford Visual Geometry Group*, 2017. https://www.robots.ox.ac.uk/~vgg/practicals/cnn/index.html#part-31-traini... (accessed Feb. 11, 2021).

[75] P. HUILGOL, "Top 4 Pre-Trained Models for Image Classification with Python Code," 2020. https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/ (accessed Feb. 12, 2021).

[76] M. DASH and H. LIU, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 1–4, pp. 131–156, 1997, doi: 10.1016/S1088-467X(97)00008-5.

[77] A. Atla, R. Tada, V. Sheng, and N. Singireddy, "Sensitivity of different machine learning algorithms to noise," *J. Comput. Sci. Coll.*, vol. 26, no. 5, pp. 96–103, 2011.

[78] D. Garcia-Gasulla *et al.*, "On the Behavior of Convolutional Nets for Feature Extraction," *J. Artif. Intell. Res.*, vol. 61, pp. 563–592, Mar. 2018, doi: 10.1613/jair.5756.

[79] R. Priya and P. Aruna, "Automated diagnosis of age-related macular degeneration using machine learning techniques," *Int. J. Comput. Appl. Technol.*, vol. 49, no. 2, p. 157, 2014, doi: 10.1504/IJCAT.2014.060527.

[80] M. R. K. Mookiah *et al.*, "Automated diagnosis of Age-related Macular Degeneration using greyscale features from digital fundus images," *Comput. Biol. Med.*, vol. 53, pp. 55–64, Oct. 2014, doi: 10.1016/j.compbiomed.2014.07.015.

[81] M. R. K. Mookiah *et al.*, "Decision support system for age-related macular degeneration using discrete wavelet transform," *Med. Biol. Eng. Comput.*, vol. 52, no. 9, pp. 781–796, Sep. 2014, doi: 10.1007/s11517-014-1180-8.

[82] S. Matsuba *et al.*, "Accuracy of ultra-wide-field fundus ophthalmoscopy-assisted deep learning, a machine-learning technology, for detecting age-related macular degeneration," *Int. Ophthalmol.*, vol. 39, no. 6, pp. 1269–1275, Jun. 2019, doi: 10.1007/s10792-018-0940-0.

[83] S. Keel *et al.*, "Development and validation of a deep-learning algorithm for the detection of neovascular age-related macular degeneration from colour fundus

photographs," *Clin. Experiment. Ophthalmol.*, vol. 47, no. 8, pp. 1009–1018, Nov. 2019, doi: 10.1111/ceo.13575.

[84] H. Müller, J. Kalpathy-Cramer, D. Demner-Fushman, and S. Antani, "Creating a classification of image types in the medical literature for visual categorization," Feb. 2012, p. 83190P, doi: 10.1117/12.911186.

[85] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Vis. Comput. Ind. Biomed. Art*, vol. 2, no. 1, p. 7, Dec. 2019, doi: 10.1186/s42492-019-0016-7.

[86] J. Azzeh, B. Zahran, and Z. Alqadi, "Salt and Pepper Noise: Effects and Removal," *JOIV Int. J. Informatics Vis.*, vol. 2, no. 4, p. 252, Jul. 2018, doi: 10.30630/joiv.2.4.151.

[87] F. Luisier, T. Blu, and M. Unser, "Image Denoising in Mixed Poisson–Gaussian Noise," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, Mar. 2011, doi: 10.1109/TIP.2010.2073477.

[88] Jian-Jia Pan, Yuan-Yan Tang, and Bao-Chang Pan, "The algorithm of fast mean filtering," in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, Nov. 2007, pp. 244–248, doi: 10.1109/ICWAPR.2007.4420672.

[89] L. Li, Y. Si, and Z. Jia, "Medical Image Enhancement Based on CLAHE and Unsharp Masking in NSCT Domain," *J. Med. Imaging Heal. Informatics*, vol. 8, no. 3, pp. 431–438, Mar. 2018, doi: 10.1166/jmihi.2018.2328.

[90] Sonali, S. Sahu, A. K. Singh, S. P. Ghrera, and M. Elhoseny, "An approach for de-noising and contrast enhancement of retinal fundus image using CLAHE," *Opt. Laser Technol.*, vol. 110, pp. 87–98, Feb. 2019, doi: 10.1016/j.optlastec.2018.06.061.

[91] K.-M. Koo and E.-Y. Cha, "Image recognition performance enhancements using image normalization," *Human-centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 33, Dec. 2017, doi: 10.1186/s13673-017-0114-5.

[92] C. Liu, "Data Transformation: Standardization vs Normalization," *kd nuggets*, 2020. https://www.kdnuggets.com/2020/04/data-transformation-standardization-normalization.html (accessed Sep. 27, 2020).

[93] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[94] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[95] M. Nisa *et al.*, “Hybrid Malware Classification Method Using Segmentation-Based Fractal Texture Analysis and Deep Convolution Neural Network Features,” *Appl. Sci.*, vol. 10, no. 14, p. 4966, Jul. 2020, doi: 10.3390/app10144966.

[96] FeanDoe, “Sensitivity and specificity,” 2018. https://commons.wikimedia.org/wiki/File:Sensitivity_and_specificity.svg (accessed Mar. 15, 2021).

[97] “Classification: ROC Curve and AUC,” *Google Developers*, 2020. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (accessed Apr. 14, 2021).

[98] “Evaluating Classification Models,” in *Practical Statistics for Data Scientists*, 2nd ed., O’Reilly, 2020, pp. 219–229.

[99] “Peking University International Competition on Ocular Disease Intelligent Recognition (ODIR-2019),” *Peking University*, 2019. https://odir2019.grand-challenge.org/introduction/ (accessed Dec. 12, 2020).

[100] “Color conversions,” *OpenCV*, 2015. https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html (accessed May 17, 2021).

[101] S. Sun, Z. Cao, H. Zhu, and J. Zhao, “A Survey of Optimization Methods From a Machine Learning Perspective,” *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020, doi: 10.1109/TCYB.2019.2950779.

[102] V. Bushaev, “Adam — latest trends in deep learning optimization,” 2018. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c (accessed Apr. 15, 2021).

[103] J. E. Espinosa, S. A. Velastin, and J. W. Branch, “Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study,” 2017, pp. 3–15.

[104] M. Z. Alom *et al.*, “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches,” Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.01164.

[105] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner,

“Survey and Benchmarking of Machine Learning Accelerators,” in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–9, doi: 10.1109/HPEC.2019.8916327.

المستخلص

مرض الشيخوخة البقعي (AMD) هو مرض يصيب العين منتشر على نطاق واسع، يصيب كبار السن. ستساعد هذه الطريقة الالية الأطباء على تشخيص المرض ومراقبة الحالات والمرضى ، لتلافي تدهور حالة العين.

في هذا العمل المقترح ، تم تدريب نموذج قائم على الشبكة العصبية التلافيفية (CNN) على مجموعة بيانات لصور العين بمستويات متعددة من حالات مرض الشيخوخة البقعي (AMD)، إضافة للعيون الطبيعية. أستخدمت مجموعة بيانات ODIR لكل من مرحلتي التدريب والاختبار للنموذج. الهدف هو تطوير نموذج تعلم عميق يصنف كل صورة بناءً على حالة مرض الشيخوخة البقعي (AMD) الخاصة بها.

تتمثل الخطوة الأولى للمشروع في تنفيذ معالجة مسبقة لمجموعة بيانات الصورة بطرق معالجة مختلفة لجعلها مناسبة لتدريب النموذج. تضمنت الخطوة الثانية بناء النموذج الذي ينقسم إلى ثلاث خطوات (أ) وهي تطوير تصميم النموذج الذي سيكون قادرًا على تصنيف الصور بشكل صحيح. (ب) هو تحديد معايير التدريب المناسبة التي ستعطي نتائج مقبولة في وقت معقول. (ج) تضمنت الخطوة الثالثة التحقق من صحة النموذج.

تضمنت عمليات التدريب مرحلتين: المرحلة الأولى هي بناء وتدريب نموذج لتصنيف الصور إلى فئتين (عادي ، و مصاب). بعد النتائج العالية التي تم الحصول عليها من هذا النموذج ، تم تطوير التصميم ومعايير التدريب الخاصة به لتصنيف ثلاث فئات من مرض الشيخوخة البقعي (AMD)، وهذه الفئات تشمل (المرض المبكر والمرض المتوسط والمرض المتأخر). وهي المراحل الثلاث لمرض مرض الشيخوخة البقعي (AMD) بجانب المرحلة الطبيعية.

تم الحصول على نتائج عالية الدقة ، وتم مقارنة النموذج المقترح مع الأساليب المتطورة الأخرى المقترحة من باحثين اخرين في نفس مجموعة البيانات. علاوة على ذلك ، تم تدريب شبكات عصبونية قوية وشائعة على نفس مجموعة البيانات ، وكشفت النتائج عن أداء أفضل للنموذج المقترح مقارنة بالنماذج الشائعة.

في النظام المقترح لتصنيف فئتين من الحالات (عادي او مصاب) ، تم الحصول على دقة إجمالية قدرها (٠,٩٨) ، (٠,٩٧٥) AUC ، (٠,٩٨٥) خصوصية ، و (٠,٩٦٥) حساسية.

وفي نظام التصنيف المقترح الخاص بالفئات الأربع ، تم الحصول على دقة إجمالية قدرها (٠,٩٨٥) ، (٠,٩٦٤) المنطقة تحت المنحنى ، (٠,٩٢٨) خصوصية ، و (١,٠٠) حساسية.

لغرض التأكد من دقة النظام على البيانات الحقيقية ، تم إجراء دراسة حالة. حيث جمعت ٥٠ صورة لشبكية العين لمرضى حقيقيين ، تم جمعها من مركز معالجة بصر محلي. ويتم اختبارها على نظامنا المقترح. النتائج التي تم الحصول عليها هي (٠,٩٨) دقة ، (٠,٩٨٣) AUC ، (٠,٩٦٦) خصوصية ، و (١,٠٠) حساسية.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل-كلية العلوم البنات

قسم علوم الحاسبات

تصنيف التنكس البقعي بالاعتماد على طريقة التعلم العميق

رسالة مقدمة الى

مجلس كلية العلوم للبنات-جامعة بابل

وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

سجى مهدي حسين

بإشراف

د.ايناس حمود السعدي

د.علي يعقوب السلطان

١٤٤٢هـ **٢٠٢١م**