Ministry of Higher Education and
Scientific Research
University of Diyala
College of Science
Department of Computer Science

Deep Learning Methods Detection of Disease from Images

A Dissertation
Submitted to the Department of Computer Science\ College of Sciences\ University of Diyala in a Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

By

Ohood Fadhil Alwan

Supervised By

Prof. Dr. Dhahir Abdulhade Abdullah

2020

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ
وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ﴾

صدق الله العظيم

(سورة المجادلة :الآية11)

(Supervisor's Certification)

We certify that this research entitled “***Deep learning Methods detection of Disease from images*** was prepared by ***Ohood Fadhil Alwan*** Under our supervisions at the University of Diyala Faculty of Science Department of Computer Science, as a partial fulfillment of the requirement needed to award the degree of Master of Science in Computer Science.

(Supervisor)

Signature:

Name: **Dr. Dhahir Abdulhadi Abdulah**

Date:2/7/2020

Approved by University of Diyala Faculty of Science Department of Computer Science.

Signature:

Name: **Dr. Taha Mohammad Hassan**

Date:

(Head of Computer Science Department)

Dedication

I would like to dedicate this work to:

To my father and mother, may God have mercy on them

To My husband Nashwan

For his unlimited love, support,

endurance and encouragement

To my candle, my children

Ahmed and Amina.

To my brothers and sisters

To everyone who helped me from a friend or fellow…

Acknowledgements

My thanks are above all to God Almighty, who guided my steps towards the path of knowledge and without his help and blessing. This thesis would not be advanced or you will see the light.

I express my sincere appreciation to the supervisors of ***Prof. Dr. Dahir Abdul Hadi Abdullah*** *for providing him with ideas, inspiration and continuous support for me during my studies.*

I am very grateful to all members and professors of the Computer Science Department at Diyala University for their general support, especially ***Lecturer Hazem, Adel,*** **Zaidoon*****, Mohammad Flaih and Mustafa Flaih****.*

*Finally, I would have never been able to finish my message without the help of friends and especially mention my friends (****Noor Hassan, Shaimaa Salem and Zeinab Yassin****) and support my family and my husband.*

Thank you all!

Ohood

Abstract

Most deaths from skin cancer are caused by the malignant type. Therefore, one of the last types of cancer is considered a treatment that can detect the disease early by biopsy examining, so the best solution for improving the diagnosis of skin cancer is early detection. Computer-aided diagnosis (CAD) is one of the widely used imaging techniques for detection and classification of skin cancer. The automatic detection and classification of image is considered very important for tumors skin and very challenging task for medical images. Previously, this decision was taken manually by humans with the help of dermatologists by taking and analyzing a patient's biopsy of the patient's skin. However, these procedures require more time and the result may not be very accurate, that leads to inaccuracies. This thesis presents a proposed system for classification of skin cancer after its detection with the help of deep learning mechanisms and machine learning algorithms. In this work, a strong classification system for skin cancer is implemented using three suggested models. The First proposed model depends on Convolutional Neural Networks (CNN) classifier. The second proposed model uses on Naïve Bayes (NB) classifier. While the third proposed model relays on Support Vector Machine (SVM) classifier. And each model with applying preprocessing algorithm and without applying. The results show that the first proposed model using (CNN) without preprocessing had average accuracy 85.00%, while with preprocessing had accuracy 69.99%. The second proposed model using (NB) without preprocessing had average accuracy 70.15%, while with preprocessing had accuracy 69.69%. The third proposed model using (SVM) without preprocessing had Achieve accuracy 76.81%, while with preprocessing had accuracy 77.12 %.

List of Contents

Chapter Five: Conclusions and Suggestions for Future Work

References

List of Abbreviations

Abbreviations	*Description*
DNA	Deoxyribonucleic acid
CAD	Computer-aided diagnosis
ABCD	Asymmetry, Border, Color, Diameter
ML	Machine Learning
NNs	Neural Networks
AI	Artificial Intelligence
CNN	Convolutional Neural Network
SVM	Support Vector Machine
ISIC	International Collaboration Skin Imaging
ISBI	International Symposium on Biomedical Imaging
VGG	Visual Geometry Group
NB	Naïve Bayes
IM	Input Image
SE	structuring element
ReLU	Rectifier Linear Unit
SLFFNs	Single-Layer Feed Forward Networks
MLFFNs	Multi - layer Feed Forward Networks
FFNN	Feed Forward Neural Network
ANN	Artificial Neural Network
BP	Back Propagation
FS	Function Signal
ES	Error Signals
SSE	Sum of Squared Errors
3D	3- Dimensional
MAP	Maximum A Posteriori

NBC	Naïve Bayes Classification
M	Margin (hyper plain distance)
RBF	Radial Basis Function
AC	Accuracy
TP	True Positive
FN	False Negative
FP	False Positive
TN	True Negative
Conv2D	Convolutional Tow – Dimension
FC	Fully Connection

List of Tables

List of Figures

Chapter one

General Introduction

1.1 Introduction

The skin is a vital organ that covers the entire outside of the body, forming a protective barrier against pathogens and injuries from the environment. But because it is located on the outer part, the skin is prone to disease. One of these diseases is known as skin cancer. Skin cancer is an abnormality in skin cells caused by mutations in cells Deoxyribonucleic acid (DNA). One of the most dangerous types of skin cancer is melanoma cancer. It is a skin malignancy derived from melanocyte cells; the skin pigment cells that produces melanin. Because these cells are still able to form melanin, melanoma is mostly brown or black colored [1].

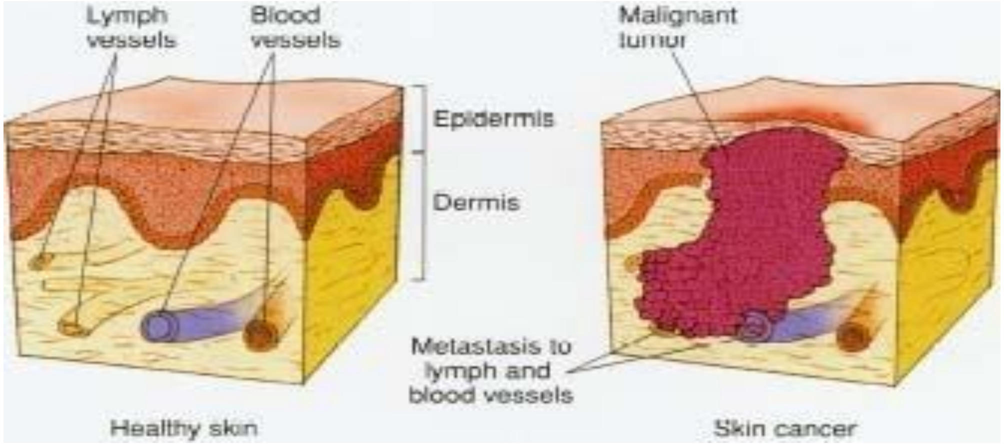


Figure (1.1): (left) Unaffected skin, (right) Affected skin [1].

More than 5,400 people worldwide die every month from malignant skin cancer, and estimates and statistics indicate that the number of new cases of melanoma cancer diagnosed in 2020 will increase by about 2 %. The number of skin cancer deaths is expected to decrease by 5.3 %. in 2020. Of these, 60,190 will be men and 40,160 will be women. In the past decade (2010-2020), the number of new diagnostic melanoma cases diagnosed annually increased by 47 % [**2**]. Skin cancer affects the men and the women at different ages, as shown in the figure.

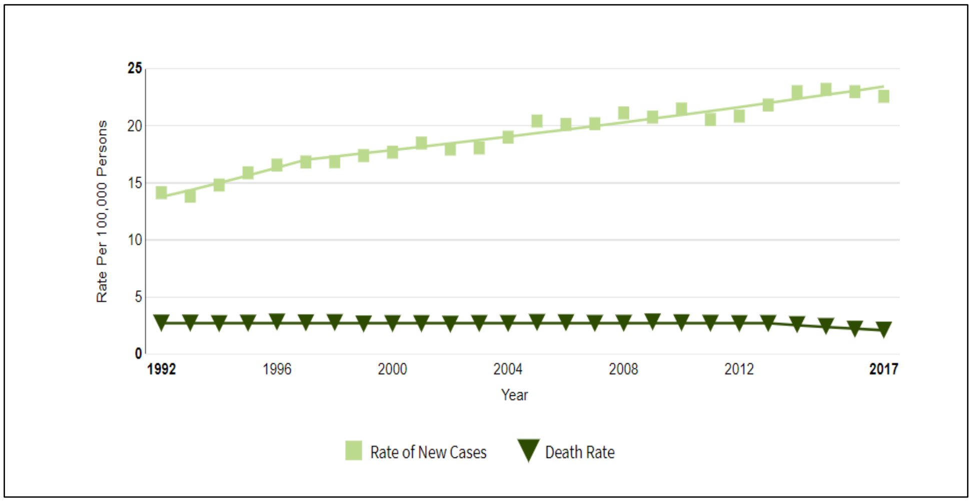


Figure 1.2: Skin Cancer Incidence and Death [2].

1.2 Skin Cancer Diagnoses

A- Traditionally Diagnoses

It is difficult to distinguish between types of skin cancer (melanoma and benign moles) at the beginning of its appearance is, even for experienced doctors [3]. The use of traditional methods to diagnose the disease by physical examination and biopsy. The biopsy is removed part or all of this spot and sent to the laboratory and the results may take a week to come through. This physical diagnosis is expensive, time-consuming, and may produce the wrong result for some reason. Therefore, sophisticated equipment and algorithms are required to assist decision makers. Various methods in dermatology such as “ABCD” (Asymmetry, Border irregularity, Color patterns, and Diameter) rule and the seven - points checklist [4].

B- **Computer-Aided Diagnosis (CAD)**

The concept of using computer vision to solve the task of identifying skin cancers arose recently. Automated pigmented lesion analysis has become an important research topic trying to improve or develop the diagnosis of computer-assisted skin cancer [**5**]. Medical image processing is an area of proven expansion and an interdisciplinary field of research and interest and various fields, computer science, engineering, applied mathematics, statistics, physics, medicine, and biology. Computer-assisted diagnostic treatment has occupied a remarkable space in the clinical routine and with the recent advances in high technology and the introduction of different methods and techniques leads to more challenge in the mechanism of dealing with the huge number of images. It provides a high-quality information that helps in diagnosing the disease [**6**]. The introduction of artificial intelligence methods as a method that helps doctors in diagnosing has become an increasing trend in dermatology. These methods generally utilize some procedure of machine learning (ML), which is a branch of Artificial Intelligence (AI) including approaches that enable machines to make the predictions based on their prior information and experiences [**7**].

1.3 Overview of Deep Learning Techniques

Machine and deep learning methods performance an important role to train computer systems as a professional prediction and decision making could be used. Machine learning is the field of study that give the computers the ability to learn without the need for complicated programs. Deep learning is one of the ways that gives the ability to understand the world, by arranging ideas and bringing intelligence to the computer, as extracting patterns and processing them becomes easy and it is one of the branches of machine learning **[8]**.

Automatic learning techniques that applied to images directly are not well efficient because they neglect or ignore the structure and composition of the image. Therefore, a deep learning solution is the place of automatic learning in many of the image processing tasks because it has the advantage of extracting features, which is part of the learning process **[9]**.

The request of computers to identify several features that can distinguish between the required data is under the idea of the basis of the work of many deep learning methods as it lies in the transfer of the image between different layers to give the result of a specific disease. These models or methods are used in processing big data to reach a size Interest required. The convolutional neural network is considered one of the most important models of deep learning in the field of image classification, that outperforms many automated machine learning algorithms. The following figure shows the main layers of the deep learning model **[10]**.

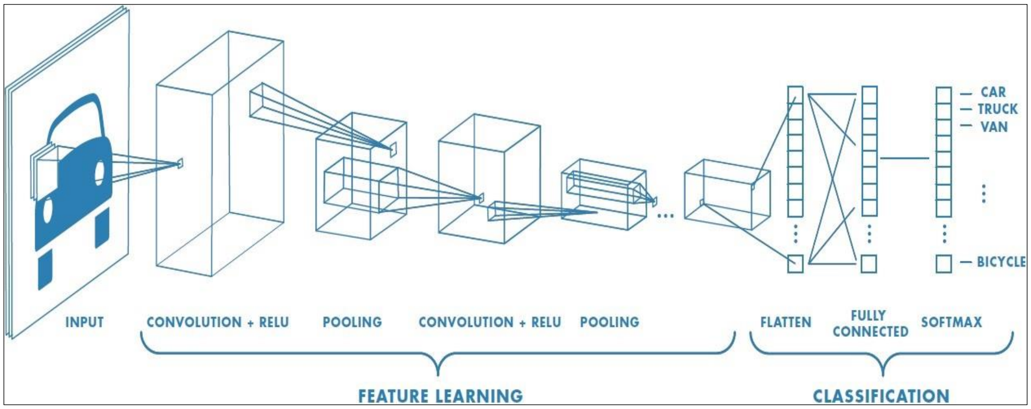


Figure 1.3: Sample of Main Layer in Deep Learning Model

1.4 Related Works

In this section the study reviews some of various styles and techniques that can used for detection skin cancer are presented:

- **Refianti R. et al 2019 [11]** Design a skin cancer image classification system to examine endoscopy. Convolutional neural network (CNN) with LeNet-5 geometry was used as a proposed method for the system in the classification of image data, as the number of testing data reached to 44 images. A classification accuracy of 93% for training and 100% for testing, which is a percentage that the model might expect in overfitting. Because of the small number of datasets, which consists only 176 images in the 100 epochs.
- **Albahar, M. A. 2019 [12]** Relied in proposed model on a technique that depends on the engineering of the convolutional network. The network is consisting of two convolution layers followed by one of max pooling layer with dropout layer to treat the overfitting, then the fully connection layer as it contains 128 neurons. The idea in this model is to include a regulator on each convolution layer to control the values of weights, which is a matrix of filter applied to each input, the model training 5600 images, but the proposed system faced problem of choosing an appropriate λ value that is difficult, because it is a continuous value and several attempts to select it are costly and takes time.
- **Sanket .K. Chandra J 2019 [13]** The researcher suggested a model to classify skin cancer in three ways, including SVM, KNN, Ensemble, used in preprocessing stage a hybrid method that are starts with wiener filter to remove noise the unwanted regions and then applied median filter to remove hair and after that used watershed algorithm with

morphological operation for segmentation to extract features from the images. The researcher relied on two types of data PH^2 contains of 200 images used for training and testing phase and ISIC dataset contains more than 30000 images of several types of cancer. SVM model has batter performance with accuracy 92% when compared to other methods like KNN, Ensemble algorithms. But he did not mention the time of the algorithm in training.

- **Mohan, K. et al. 2019 [14]** Classification of skin cancer was discussed using naive Bayes classifier with shear let transformation factors with three coefficients. Treated melanoma images for rank feature then applied naive bayes for classification. The results showed that the system achieved accuracy of 90% at levels 3 through 100. By using the PH^2 dataset contains 100 dermoscopic RGB images with melanocytic lesions with resolution is768x560 pixels. The researcher also explained that when applied shear let transformation on images and with other coefficients increases the complexity of the calculations and the required more time.
- **Md Ashraful 2018 [15]** Selected four approaches of the convolutional neural network such as SENet154, PNASNet-5-Large, InceptionResNetV2, InceptionV4to test the model in the classification of skin cancer images obtained from the (ISIC) 2018 Challenge data set. It contains more than 10015 pictures, after the pre-processing stage and testing the models, the results showed an accurate classification to PNASNet-5-Large with 76%, But it faced the problem of unbalanced data with a big change in all images make it difficult to generalize these features of skin lesions.

- **Lopez, A. et al 2017 [16]** Focus on the classification of skin cancer (benign or malignant), how to detect it early, and introduce deep learning methods to solve these problems. The researcher used the convolutional neural network model with VGG structure where that transfer learning model was used. The proposed method was tested on the "International Symposium- on Biomedical Imaging" (ISBI) 2016 data set (346 for training and 150 for testing). The model obtained a 78.66% classification rate.
- **Codella, N. C. et al 2017 [17]** Designed a system based on machine and deep learning techniques to detect and classify skin lesions (benign and malignant) using dataset released by the International Skin Imaging Collaboration (ISIC) for the 2016 International Symposium on Biomedical Imaging (ISBI 2016), where dataset was splatted into 900 for training and 379 for testing, and the researcher relied on multiple models of deep learning deep fully convolutional-Net architecture residual networks, convolutional neural networks, segmentation using to extracting features with the help of machine learning algorithms. Proposed model achieved classification accuracy76%.
- **Mustafa, S. et al 2017 [18]** Suggested a system which can a distinction between the skin lesions by using machine learning techniques such as SVM model and with use ABCD rule where he used color space by experimenting with luminance to increase the visualization for Grab Cut segmentation of image .Dataset that used 200 images 100 as benign and 100 as malignant .The algorithm can discovery the optimum line to separate the two classes with accuracy 80% but with

low sensitivity 71% and specificity 55%. The proposed method be facing a problem with a small dataset for training algorithm.

- **Nasr-Esfahan, E et al 2016 [19],** Applied a two-layer CNN was trained for the distinction of melanoma against benign nevi) built on clinical pictures. Only (136) images from dataset were used to train the model and the test dataset contained 34 images. The images were all from the public image archive of the "Department of Dermatology". The proposed method after preprocessing stage and tested model achieve accuracy of 81%, sensitivity of 81%, and a specificity of 80%. The tested images were very limited. However, the result can be improved when increased it.
- **Shoieb, D. et al 2016 [20]** Presents model for diagnosing the skin cancer by applying deep learning approaches. Enhanced segmentation is a stage the model applies to identify malignant skin cancer while the researcher used a network (CNN) to extract features from the images. The model was built on a multi-layered linear with SVM that was trained by features extracted from a(CNN) network. Despite the experimental results obtained by the system with accuracy 94%, but dataset that obtained from normal camera it faced additional effort in the pre-processing stages. Whereas, the total dataset 337 image 80% for training and 20% for testing.
- **Park, D. C. 2016 [21]** proposed a model explain how the naive Bayes Classifier could classify image of skin cancer and can be formed as the maximum posteriori of decision-making rule. The researcher relied on take advantage of concepts of the Naive Bayes probability classifier, in order to reduce the training time for the algorithm. Total images of

dataset 800 images divided in four class and each class contains 200 images. The proposed classifier reached accuracy 77.2 %.

1.5 Problem Statement

Skin cancer is the most dangerous type of cancer that requires an early detection to determine it, whether it is benign or malignant. Using neural networks has shown outstanding results, with high flexibility in different environmental conditions, but its limitations in image classification processes have led us to use deep learning methods to solve this problem as it has achieved impressive results in the field of medical image classification because early detection leads to rapid treatment.

1.6 Aim of the Thesis

The main aim of this thesis is to design a system to detect and classify skin cancer with different models, with applied proposed systems, the doctor can train the system on some known data and then apply this method to classify skin cancer. These models are:

1- Design and implement such a powerful structure by using Convolutional Neural Network (CNN) structure for the first classification approach.

2- The second model is Naïve Bayes (NB) that used for classification skin cancer for benign or malignant.

3- Support Vector Machine (SVM) is the third approach for classification approach Skin cancer of benign or malignant.

The objective of utilizing more meaningful information to improve skin cancer detection and help doctors and physicians in the clinical diagnosis with accurate detection of the disease and giving reliability in decision-making and rapid detection of skin cancer.

1.7 The Organization of the Study

This thesis consists four chapters in addition to the chapter one that was discussed above and it is organized as follows:

Chapter two describes the pattern classification system, design medical images analysis system, the concept of skin cancer with its types and overview of the method used to analysis and categorize skin images with their characteristic**.**

Chapter three present the details of the proposed detection and classification algorithms that are used to design the proposed system and the implementation of each one.

Chapter four gives the experimental results obtained from the implementation of proposed system.

Chapter five discusses results, conclusions and lists a number of suggestions for future studies.

Chapter Two

Theoretical Background

2.1. Introduction

This chapter provides an overview of the theoretical background of the main approaches used in this thesis, closing images for hair removal and median filter for image enhancement. Datamining approaches Convolutional Neural Network (CNN), Support Vector Machine (SVM), and Naive Bayes (NB) and which are the main methods that will be discussed in this thesis to perform classification.

Deep and machine learning procedures play an important role in the computer systems as a proficient which can be used further for prediction and decision of making. Machine learning is the field of study that provides computers the ability to learn without being explicitly programmed [**22**]. Deep learning is a type of machine learning that empowers systems to gain for a fact and comprehend the world regarding a pecking order of ideas. These fields bring intelligence into a computer that can extract the patterns according to the specific data and then process for automatic reasoning. Medical imaging is the rapidly growing research area that is used to diagnose a disease for early treatment. The function of image processing in the health domain is relative to the growing position of medical imaging [**23**].

2.2 Image Preprocessing

In order to progress the performance of the classifier, Preprocessing processes converts images into more convenient formats to facilitate the work and matching, which improves the quality of predictions provided by these classifiers. Removing hair from the image and using some filters to recover the image are instances of preliminary image processing.

2.2.1 Hair Removal Algorithm

In dermoscopic images are the most common artifact and necessary to remove the hair. Many methods and algorithms are presented in the literature to remove the hair when it is not shaved before the acquisition step. Therefore, the typical algorithm of hair removal methods is based on two main steps:

1. First use simple morphological closing operation with a disk-shaped structuring element. Based on the assumption that hair segments are thin structures, a simple morphological technique is applied.
2. Next, median filter applied to make the whole image the same degree of color and depends on the most prominent color, which is the (skin color). Each hair pixel from the resulted filter is replaced by an average mean of the neighbor's pixels [**24**].

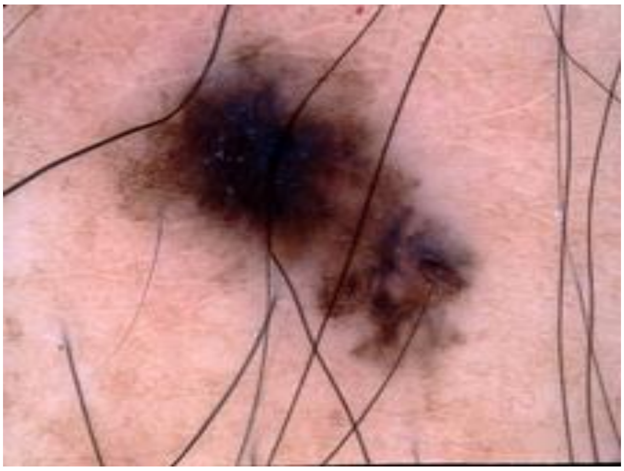


Figure (2.1) Hair is some Effects on skin

Morphological Opening and Closing Process

Together, the opening and closing processes constitute a method that manipulates erosion and dilation processes to gain a clearer image. Opening is a process that applies erosion and is followed by dilation on the input image (IM), while the closing process is the reverse of the opening process as shown in figure (2.2). The structuring element (SE) that is used for both processes is similar. The opening and closing operations are defined as below [**25**].

$$Opening = IM \, \Theta \, SE \; \oplus \; SE \qquad \textbf{(2.1)}$$

$$Closing = IM \; \mathit{\Theta} \; SE \; \oplus \; SE \qquad \textbf{(2.2)}$$

where $\ominus$ and $\oplus$ denote erosion and dilation respectively.

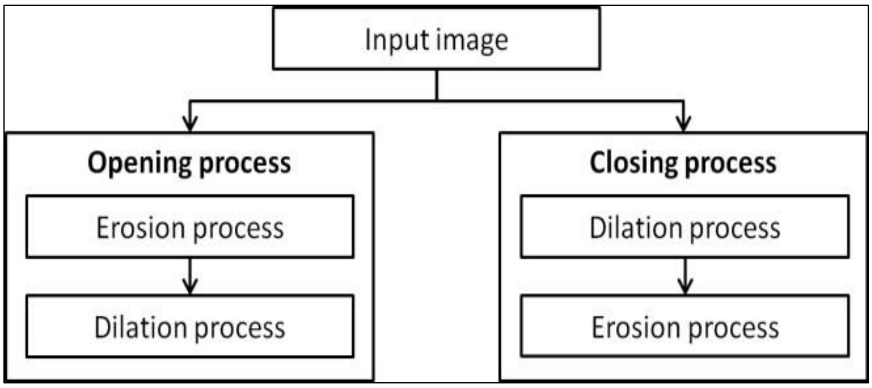
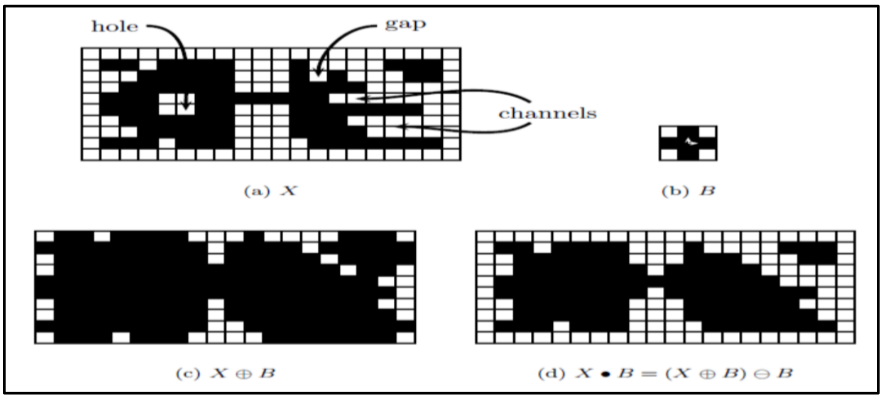


Figure (2.2): Flowchart of the Opening and Closing Processes

Closing Operation

Closing is an important operator from the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators it is normally applied to binary images, although there are gray level versions. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape **[26]**.



(a) X (b) B

(c) $X \oplus B$ (d) $X \bullet B = (X \oplus B) \ominus B$

Figure 2.3: Example of Closing Process

As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels **[26].**

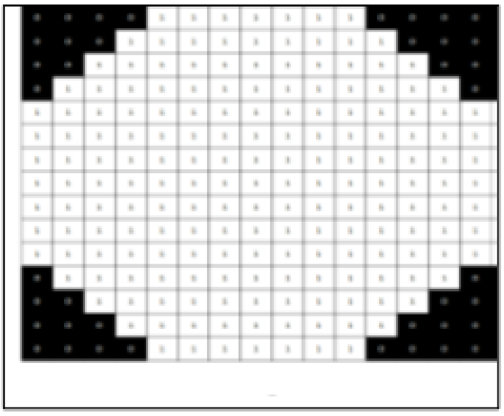


Figure 2.4: Structure Element Disk Shape.

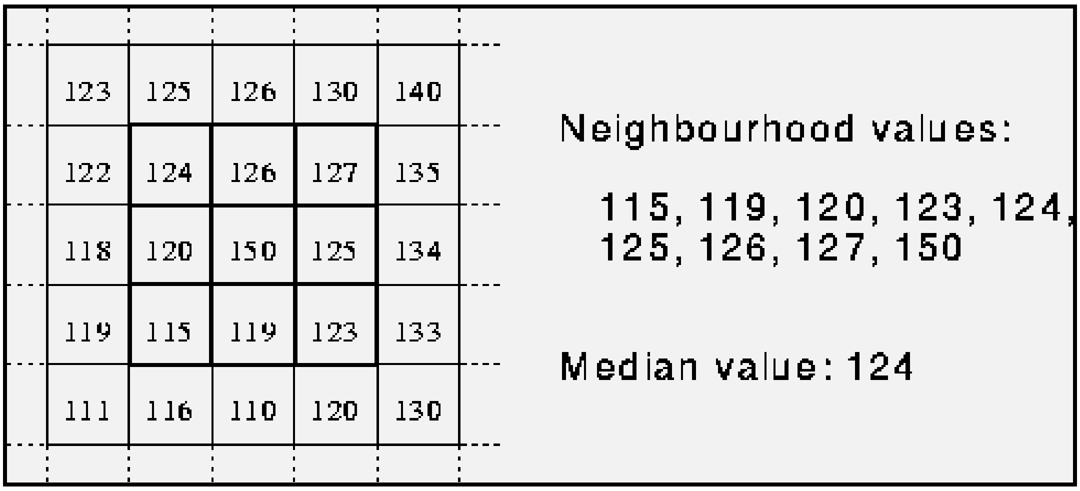
2.2.2 Image Enhancement

Image enhancement is one of the simplest methods and most popular in digital image preprocessing. Basically, the idea behind enhancement techniques is to highlight certain features of interest in an image or to bring out detail that is obscure. Image enhancement is applied in every field where images are ought to be understood and analyzed. For example, medical image, analysis of images from satellites etc. Various enhancement techniques are used for enhancing an image such as used filtering, image gray-scale manipulation, and histogram [**27**].

Median Filter

The median filter is a nonlinear digital filter used to image enhancement. In order to perform median filtering at a point in an image, values of the pixels in its neighbor, are first sorted, determine their median, and assign this value to that pixel. The median filter method is a simple and efficient technique and it is quite popular because for certain types of random noise it provides excellent noise reduction capability and it preserves edges [**28**].

A median filter is based moving a window over an image and computing the median value of the output pixel of the brightness within the input window. If the window is j×k in size the j*k pixels can be ordered in brightness value from smallest to largest. If j*k is odd then the median will be the (j*k+1)/2 entry in the list of ordered brightness. Figure (2.5) illustrates calculation the median value of a pixel neighborhood the central pixel value of 150 is replaced with the median value: 124 [**29**].



123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Figure 2.5: Example of Median Filter [29]

2.3 Artificial Neural Network (ANN)

The biological Neural Network consider is the main natural source principle of the Neural Network that has a proximally about 10 billion neurons connected via 100 trillion interconnections in the human brain. The Neural Networks (NNs) neurons process the information and the data. The particular neurons are communicated to each other by a connection is called synapses; where the synapses have set of variables such as weights. For this reason, the Neural Networks have parallel processing distributed system [**30**], Figure (2.6) shows biological neurons.

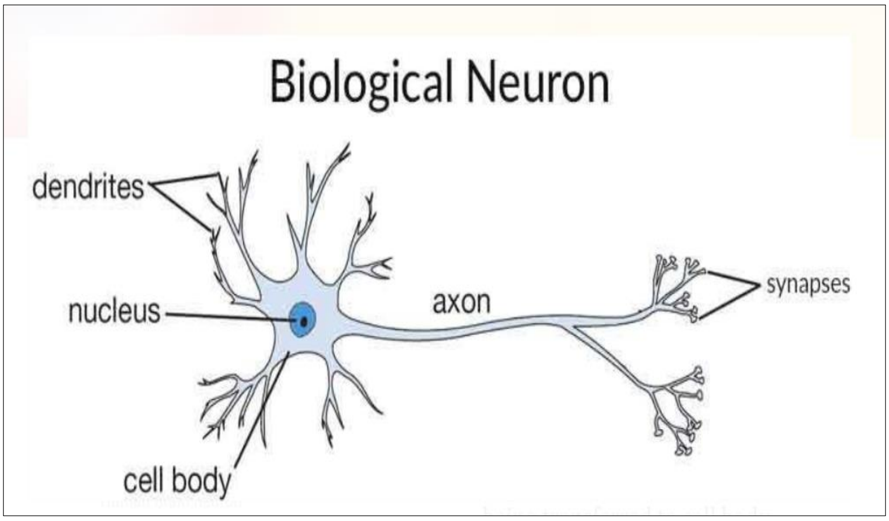


Figure 2.6: Biological Neuron

2.3.1 Neural Network Model

Suppose that the input vector $x = (x_{1,} x_2, x_3, \dots x_n) \in R^n$ of real numbers. The inputs are sent through the connections to the accumulator and the connections may carry some weight vector $w = (w_1, w_2, w_3, \dots, w_n) \in R^n$. These are applied to the inputs x_i by multiplying the input with its corresponding weight w_i. The products are then added and the information sent to the accumulator is of the form as given in equation (2.3) [**31**]:

$$(w, x) = \sum_{i=1}^{n} w_i \times x_i \qquad \textbf{(2.3)}$$

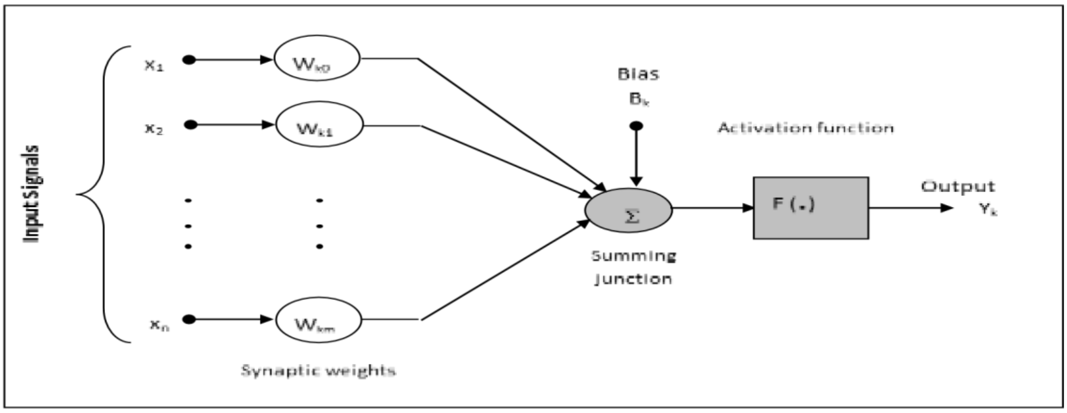


Figure (2.7): Neural Network Model

The neural network model in Figure (2.7) comprises an externally applied by bias, indicated by (b_k). The bias (b_k) has the impact of expanding or decreasing the net input (is the weighted input signal) of the activation function, which is relying on if it is positive or negative [**32**].

In mathematical terms, a neuron k may be described by the following equations:

$$u_k = (w, x) = \sum_{i=1}^{n} w_i \times x_i \qquad \textbf{(2.4)}$$

And

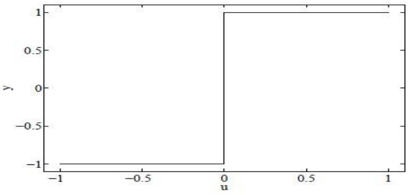
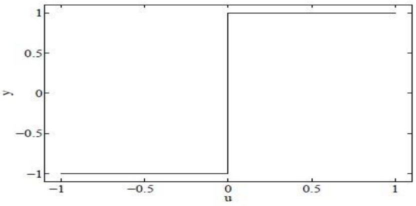
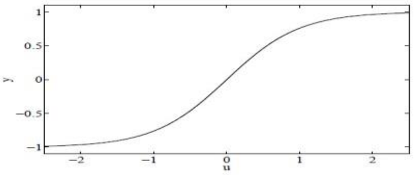
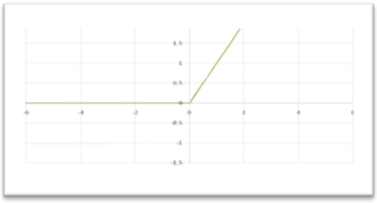
$$y = f[(w, x) = f(\sum_{i=1}^{n} w_i \times x_i) \qquad \textbf{(2.5)}$$

where x_i is the input signals, w_k is the weights synaptic of the neuron k, u_k is the combined output outstanding to the input signals, $f(u_k)$ is the activation function, y_k is the output signal of the neuron [**32**].

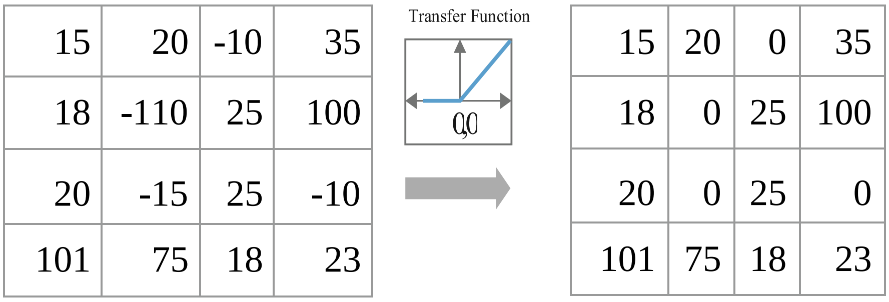
Activation Functions

Passing the result of the summation into an activation function provides the neuron with the ability of creating nonlinear boundaries for decision making. For example, if an activation function is not included in the computation of the neuron, the only possible boundary that a neuron can use to split the tuples in the dataset into classes is a linear boundary, which reduces the ability of providing more accurate predictions. Moreover, neurons located deeper in the neural network would have the ability of creating more complex boundaries for each class, which also improves the accuracy of the predictions provided by the entire neural network. In addition, the use of the bias values within each neuron can assist the creation of these complex boundaries by adjusting the locations of each part of the complex boundary, which is created by combining boundaries of neurons prior to that neuron [**33, 34**]. Some of the most common functions of activation listed in table (2.1).

Table (2.1) Type of Activation Function

Name	Equation	Figure
Unipolar Binary Function	$F(u) = \begin{cases} -1 & u < 0 \\ +1 & u \geq 0 \end{cases}$	
Bipolar Binary Function	$F(u) = \begin{cases} +1 & u < 0 \\ -1 & u \geq 0 \end{cases}$	
Bipolar Sigmoid Function	$g(u) = \tanh(u)$	
Rectifier Linear Unit (ReLU)	$ReLU(y) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases}$	

The widely used activation functions are Linear Rectifier (ReLU), neural networks with ReLU activation functions have shown significantly better performance than the other activation functions [**33**,**34**].



15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Figure 2.8: Example of ReLU Functionality

2.3.2 Neural Network Architecture

The neurons are arranged into layers where the forms of connection within and between the corresponding layers are named the network architecture. Generally, the neurons existed in a particular layer behave in similarly manner. The behaviors of neurons are determined by the weight of connections pattern where the signals send and receives and activation function. Within each layer, neurons usually have the same activation function and the same connections pattern to other neurons. In a lot of neural networks, the neurons within one layer are fully interconnected or not connected at all. In case of some neuron in a specific layer (for instance, the layer of hidden units) is associated to a neuron located in another layer (say, the output layer), then for each layer of hidden unit is attached to all output neuron [**35**] Neural Network can be classified according to number of layer to the following:

A. Single-Layer Feed Forward Networks (SLFFNs)

A Single-Layer Feed Forward Networks has a single of connection weighted layer. Usually, the units can be identified as input (input units), and used for receiving signals from the outside world, and output (output units), where the net response can be read. In the standard SLFFNs shown in Figure (2.9) fully connected between the input units and the output (output units) however are not connected to other input (input units), the output (output units) have no connection with other output units [**35**].

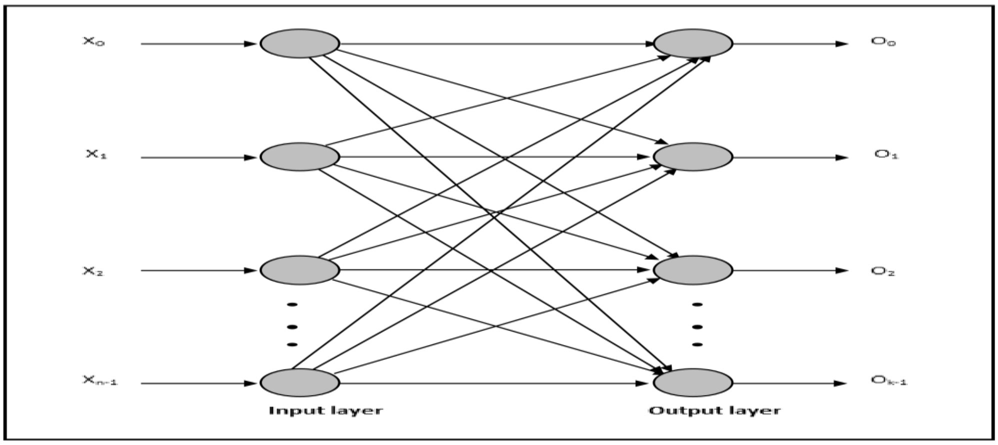


Figure 2.9: Feed Forward Network Single-Layer [36]

B. Multi - layer Feed Forward Networks (MLFFNs)

The second type of the ANN proposal is the Feed Forward Neural Network (FFNN). In this design, the ANN is presented by using input layer and one or more than one hidden layer as well as to the output layer. The main purpose of using the hidden layer is that to give ANN ability to the hidden layer to extract the higher order of the valuable statistic which is used to fine tune and adjust the whole parameters in the ANN. The architectural graph in Figure (2.10) shows the design for the case of a single hidden layer of a multilayer feed forward neural network [**35**].

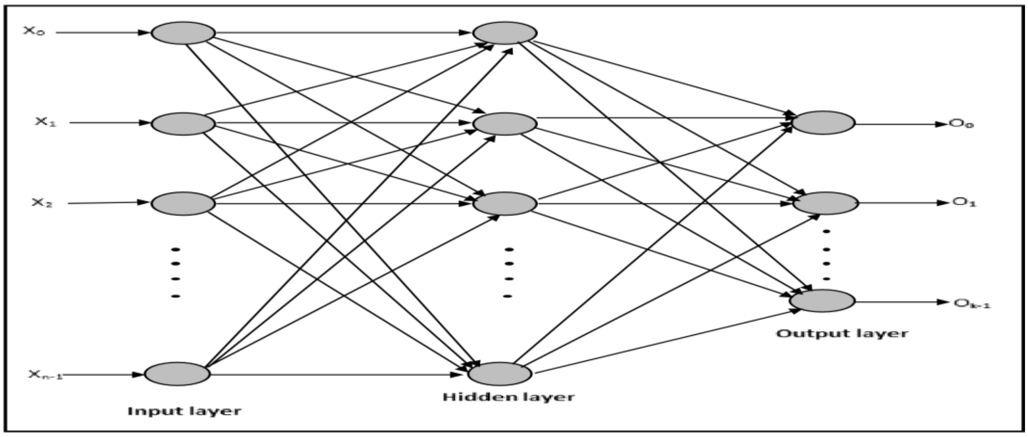
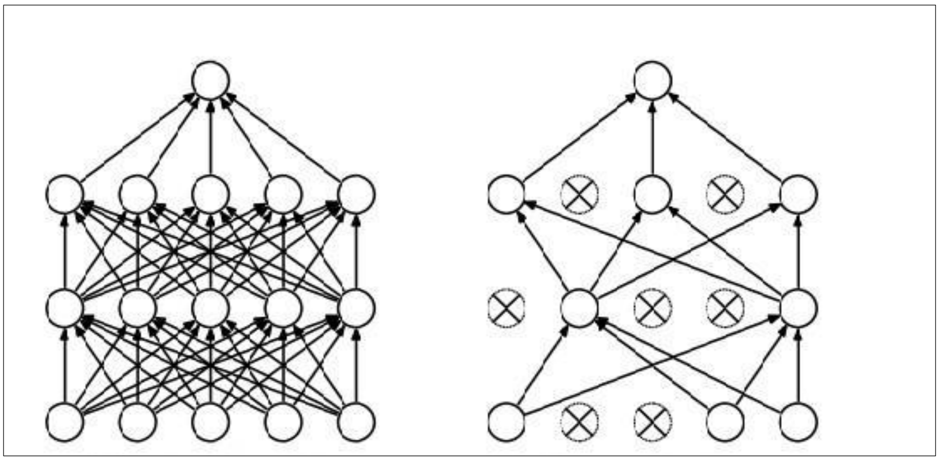


Figure 2.10: Feed Forward with One Hidden Layer [35]

2.3.2 Training Neural Network

Training is a process that is set weights contact. Most of the training systems begin by assigning random numbers for the weight's matrix. It is then adjusted on the basis of weights. The process of weight adjustment is repeated until the error limit in acceptable validation manner. The aim of the training is to produce the desired output (or at least consistent) when it is applied to a set of inputs to the network [**35**].

The main challenges faced by neural networks is the phenomenon of overfitting, where the predictions are based on definite features in the neural network, which makes these predictions very restrict to these features. Thus, any new inputs that may belong to that class but do not fire the neurons corresponding to these features are most probably are going be wrongfully classifies. The best solution for reducing overfitting is to obtain more training data. A model trained on a larger dataset typically generalizes better, though that is not always attainable in medical imaging. The other solutions include regularization with dropout or weight decay, batch normalization, and data augmentation, as well as reducing architectural complexity. To overcome such problems, a predefined ratio of the neurons in a hidden layer are randomly dropped per each iteration of the training phase, so that, the neural network is forced to find alternative paths to the same prediction and reduce the dependency on specific features. This approach is known as Dropout and has shown good improvement in the predictions providing by neural networks [**36**], it is shown in figure (2.11).



a- Neural Network b-After applying dropout

Figure 2.11: Dropout Neural Network [**36**]

Back Propagation Algorithm

Back propagation neural network is considered as the virtually used one in the arenas of the Neural Network models. The main design of the multi-layered feedforward neural network is based on backpropagation learning algorithm (design). This means that the design of (MLFFNs) is ordered in layers, then the output signal sends forward, then the errors of the whole networks is calculated and propagated again backward to the same layer.so, the network receives the signal by neurons in the input layer, and the output layer of the network is produced the output result of only neurons. The output may be back propagated by one or additional intermediate hidden layers. The notion of backpropagation learning algorithm is mainly used to fine training parameters where the error should be minimizing. The main parts of the backpropagation algorithm are described below [**35**] [**37**].

1- **Function Signals (FS):** The function signal here represents the incoming input signal that comes directly from the input end of the network, which is propagated again to forward in such (neuron by another) along the network. And then, it is emerged together at the output of the network and represents again as an input signal.

2- **Error Signals (ES):** Layer by another layer an error of the output neuron of the all network is computed and propagated backwardly to the first hidden layer throws the whole network [**35, 37**].

The calculation of a guess of the gradient vector which is the gradients of the error surface with respect to the weights associated to the inputs of a neuron), which is needed for the backward pass through the network as shown in Figure (2.12) [**35**].

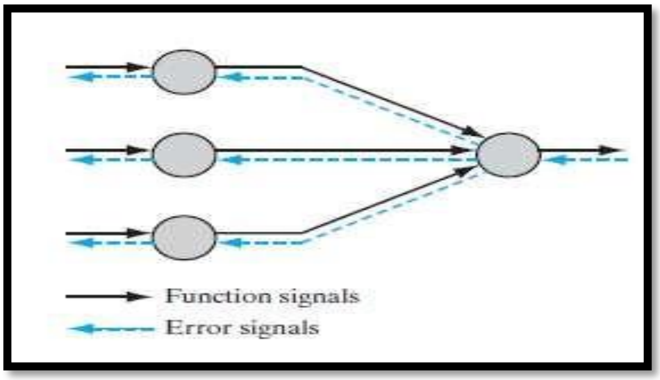


Figure 2.12: Illustration of Directions of Signal Flows [**35**]

The predictions provided by a neural network are the results of the computations performed in each neuron, per each layer starting at the input layer toward the output layer, which depends mainly on the weights and biases of the neural network, in addition to the input values of the input layer. As the input values are behind the control of the neural network, the training stage aims to adapt the weights and biases of the neural network in order to improve the accuracy of the predictions provided by these networks. This update is achieved by measuring the change between the predicted values and the required values, which is known as the cost using Sum of Squared Errors (SSE) function which shown in equation (2.6).

$$C = \sum_i \frac{1}{2}(y_i - \hat{y}_i)^2 \qquad \textbf{(2.6)}$$

where y is the predicted value and $\hat{y}$ is the actual required value for the i outputs in the output layer [**38**].

According to the calculated cost, the weights and biases are updated by back-propagating algorithm through the neural network and update their values based on the gradient descent algorithm. This algorithm calculates the partial first derivative of the calculated error with respect to each weight and bias in the previous layer. Then, the values computed from each derived is subtracted from the original value of that weight or bias. According the equation:

$$(weight)new = (weight)old - \eta \times \Delta \mathrm{w} \qquad \textbf{(2.7)}$$

Where w: is the weight if network, η: is learning rate, Δw: The change between weights

As the value computed from the partial derivative is positive when the effect of that parameter increases the error, the updated values ensure the decrease of the error in the output layer. Additionally, if the value of the derivative is negative, this indicates that increasing the value the error is increases, such that, combined with the subtraction in the update equation the new value is increased, which also reduces the error rate and advances the predictions provided by the neural network [**39**].

2.4 Convolutional Neural Networks

Convolutional Neural Network (CNN) model is considered as one of the most common modern types of the deep learning methods. Deep learning recently approach has shown a dressed performance in pattern recognition and computer vision tasks. It shows that the methods of deep learning are successfully used in several biomedical image analysis challenges, such as mitosis detection and brain image segmentation. Standard image analysis involves a series of steps including preprocessing, image segmentation, and careful selection of features, learning, and classification. Deep Learning performance of these methods is strongly trusting on the carefully chosen features and the accuracy of the previous stages such as feature dimension and fully connected method. [**40, 41**].

2.4.1 Architecture of Convolutional Neural Networks

Convolutional Neural Network (CNN) is a function signified by g that used for data x drawing, (e.g. an image) toward another output vector that signified by y. Then, the function g is the intensive combination of a sequence of simpler functions f_l, that we call layers or blocks of summation, $g = f_1 \circ \dots \circ f_L$.

Adopt the network input is $x_0 = x$, and the network outputs are, $x_1, x_2, \dots, x_L$. Where, each output $x_l = f_l(x_{l-1}; w_l)$ is deeply calculated from the previous output x_{1-1} by applying the function f_l with the parameters of w_l **[42].**

The data flowing through the network represents a feature field; $x_l \in R^{\mathrm{H}_l \times \mathrm{W}_l \times \mathrm{D}_l}$. Since the data x has a spatial structure, H_l and W_l are spatial coordinates, and D_l is depth of channels. The functions f_l act as local and translation invariant operators therefore the network is called convolutional.

CNNs are applied to discriminate between different classes by creating such as a vector of probabilities that denoted by $\hat{y} = f(x)$ for all tested image. If y is the true label of image x, CNN performance of true label y of image x is measured by a loss function $\ell_y(\hat{y}) \in R$ which gives a penalty to classification errors [**42**].

The main differences between Convolutional Neural Networks (CNN) and systematic Neural Networks (NNs):

- CNN input is accomplished to handle 3D volume or 2D and even 1D depending on application and supplies while NN input is a 1D array only.
- CNN has higher and large number of layers than NN.
- CNN associates for each node are local while NN is fully connected.

A CNN contain three main types of layers: convolution layers, Max-pooling layers, and a fully connected layer [**43**].

A. Convolutional layer

A convolution layer is an essential component of the CNN architecture that accomplishes feature extraction, which typically involves of a combination of linear and nonlinear operations, i.e., activation function and convolution operation.

Convolution

Convolution is a specific type of linear operation used for feature extraction, where a small array of numbers, called a "kernel", is applied on all the input, which is an array of numbers, called a "tensor". An element-wise produce between each element of the kernel and the input tensor is intended at each location of the tensor and summed to obtain the output value in the corresponding position of the output tensor, named a feature map. Two key hyperparameters that describe the convolution operation are size and number of kernels. The latter is arbitrary, and regulates the depth of output feature maps [**44**]. Figure (2.13) show convolution layer contain 4 filters.

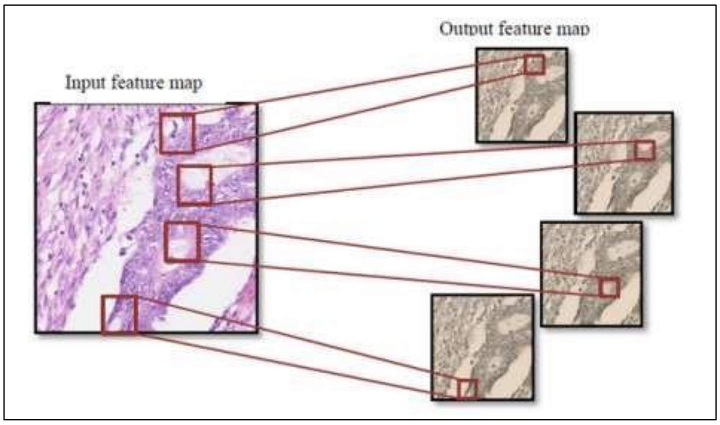


Figure 2.13: Convolutional Layer with 4 Filters.

Each filter is slipped through the height and width of the input image, creating a 2-dimensional activation map filter. The filters have the similar depth as in the input. The size of the output can be measured by three hyperparameters which are the zero-padding, depth, and stride.

1. **Zero-padding**: padding zeros around the boundaries of the input to preserve its size.

2. **Depth:** is fundamentally the number of filters that is apply to the input image.

3. **Stride:** number of pixels the filter jumps while gliding over the image. These filters distinguish structure such as edges, corners, blobs etc. as shown in Figure (2.14).

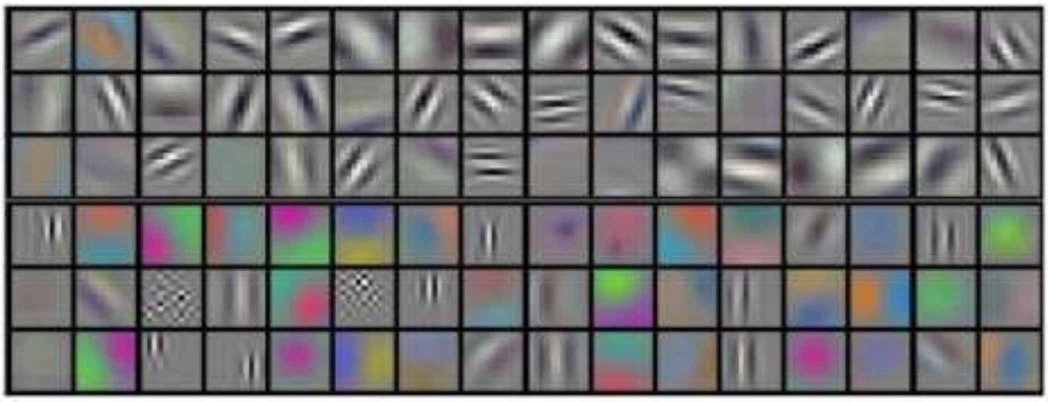


Figure 2.14: example of The Filters

Figure (2.15) A convolution operation with zero- padding so as to retain in-plane dimensions. Note that an input dimension of 5 × 5 is saved in the output feature map. In this instance, a kernel size and a stride are set as 3 × 3 and 1, respectively.

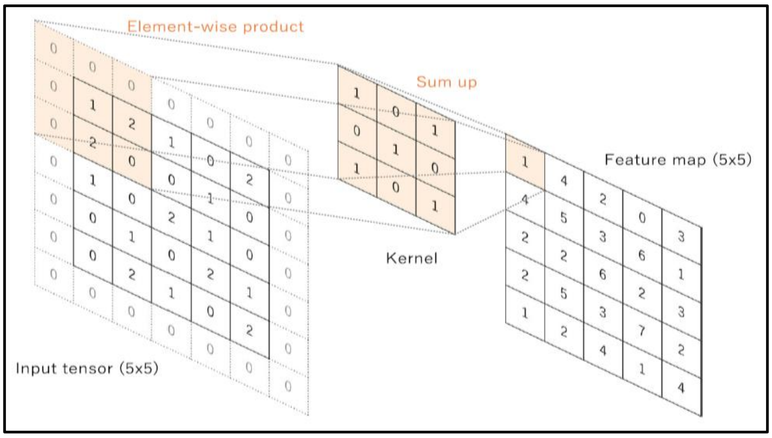


Figure 2.15: Sample Convolutional Layer Processing

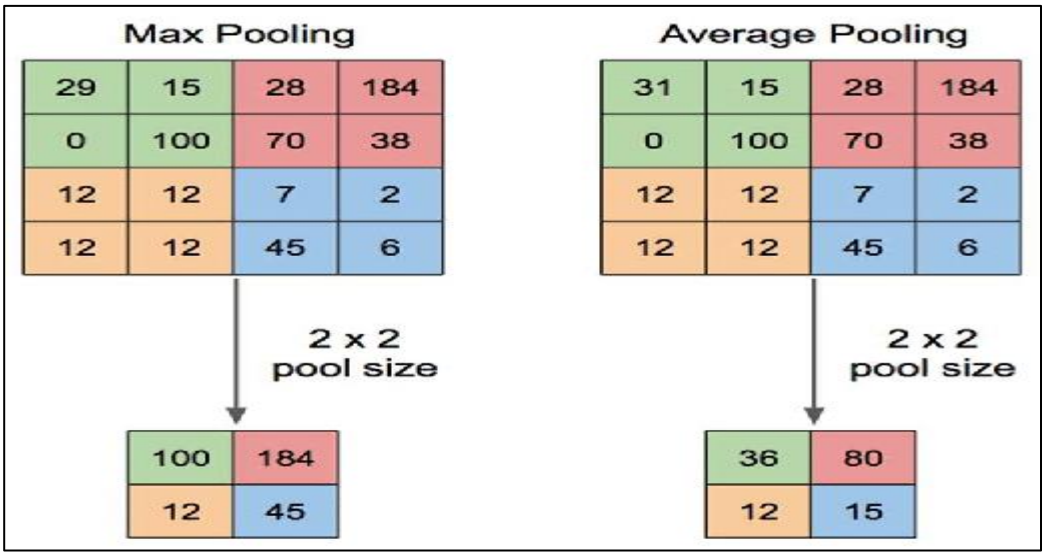
It should be noted that the number of parameters calculated during this layer is by the following equation

No. parameters=output channels* (input channels* window size + 1) **(2.8)**

B. Pooling layer

A pooling layer provides a typical down sampling process which reduces the in-plane dimensionality of the feature maps in order to present a translation invariance to small shifts and distortions, and reduction the number of subsequent learnable parameters. It is of note that there is no learnable parameter in any of the pooling layers, whereas filter size, stride, and padding are hyperparameters in pooling operations, like to convolution operations.

Max- pooling and average-pooling are the most common pooling operators (Figure 2.16). These operators calculate the maximum or the average value within a small spatial block. Pooling with filters size of 2 × 2 with a stride of 2 are considered ideal [**45**].



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

36	80
12	15

Figure 2.16: Max-Pooling and Average-Pooling

The most common form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the largest value in each patch, and rejects all the other values. Figure (2.17) explains max - pooling operation with 3 × 3 filters.

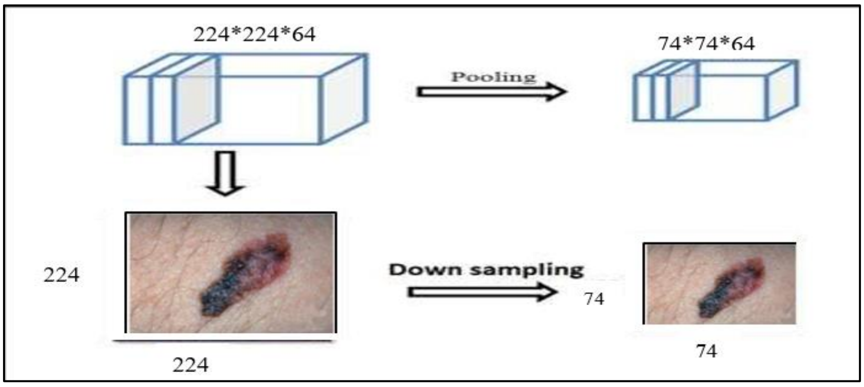


Figure 2.17: Max 3 × s3 Pooling Layer to Minimize the Spatial Size Image

C. Fully-connected layer

Fully-connected layer connects to all the neurons of the preceding layer as showed in (Figure 2.18). The output feature maps of the final convolution or pooling layer is naturally flattened, i.e., transformed into a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight. When the features extracted by the convolution layers and down sampled by the pooling layers are formed, they are mapped by a subset of fully connected layers to the last outputs of the network, such as the probabilities for each type in classification jobs. The last fully connected layer typically has the same number of output nodes as the number of types. Each fully connected layer is followed by a nonlinear function, such as ReLU, as described above [**45**].

Activation Function as Last Layer

The activation function applied to the latter fully connected layer is generally different from the others function. An appropriate activation function wants to be selected according to each task. An activation function apply to the multi class classification task is a soft max function which normalizes output real values from the last fully connected layer to target class probabilities, where each value ranges between 0 and 1 and all values sum to 1 [**46**].

A sample of fully connected layer is depicted in Figure below:

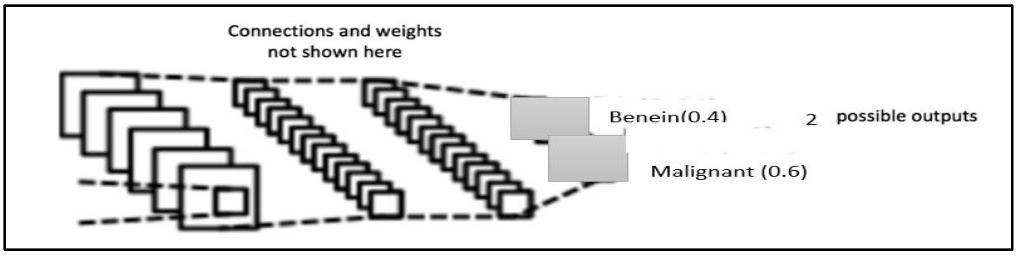


Figure (2.18): Fully Connected Layers

2.5 Naïve Bayes Algorithm

Naive Bayes technique is a classification scheme which owes its name to the British reverend Thomas Bayes (1702 - 1761). Despite its name (also known as "Idiot's Bayes") it is one of the most efficient and effective inductive learning algorithms for machine learning and data mining [**47**, **48**]. The Naive Bayes classifier is based on Bayes' theorem of probability. In Bayes' theorem, the conditional probability that an event *x* belongs to a class C can be calculated from the conditional probabilities of finding particular events in each class and the unconditional probability of the event in each class. That is, for given data, $x \in X$, and classes *C*, where *X* denotes a random variable, the conditional probability that an event *x* belongs to a class *(c)* can be calculated by using the following equation:

$$P(c_j|x) = P(c_j)\frac{P(x|c_j)}{P(x)} \qquad \textbf{(2.9)}$$

This equation shows that the computing of $P(c_j|x)$ is a pattern classification problem since it finds the probability that the assumed *x* data belongs to class j and we can decide the optimum class by selecting the class with the highest probability among all possible classes C, which can minimize the classification error. For doing so, we need to estimate $P(x|c_j)$ and assume that any particular value of vector x conditional on C_j is statistically independent of each dimension and can be written as follows:

$$P(X|C_j) = \coprod_{i=1}^{n} P(X|C_j) \qquad \textbf{(2.10)}$$

where X is a n-dimensional vector data X= (x_1, x_2, …, x_n).

The Naive Bayes classifier is based on equation (2.10) and assumes that each feature be statistically independent. This guess results in simpler calculation cost and efficient data processing. By combining equation (2.9) and equation (2.10), the Naive Bayes classifier can be summarized as the following equation:

$$J = argmax_c P(C_j) \coprod_{i=0}^{n} P(X_i|C_j) \quad \textbf{(2.11)}$$

where the denominator P(x) is omitted since the value is the same for all class.

Naive Bayes classifier is regularly referred as the maximum a posteriori (MAP) decision rule. Note that the assumption of statistically independence in each feature sometimes does not hold in certain cases and causes problems in some practical cases. However, various applications and experimental studies show that training schemes based on the MAP decision rule with the Naive Bayes assumptions yield an optimal classifier even when the assumption does not hold. **[49]**

Conventional classifiers compute the local classification decision probability and uses the information for parameters for creation a global classification decision. However, the Naïve Bayes classification computes the classification probability by using equation (2.14) directly when the model assumes a Naïve Bayes classifier as its local classifier. When the feature value is a continuous value, the proposed NBC guesstimates the probability that a feature vector element is classified as its class. Therefore, for estimating the probabilities of continuous feature value the following equation can be utilized and the training data are spitted by class and the mean and stander deviation of every class is calculated [**50**].

We can get to mean from the equation:

$$\mu = \frac{1}{n} \times \sum_{i=1}^{n} x_i \quad \textbf{(2.12)}$$

And stander deviation from equation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \quad \textbf{(2.13)}$$

So, the probability density function:

$$P(X_i|C_j) = \frac{1}{\sqrt{2\pi\sigma_{cj}}} e^{-\frac{(x_i-\mu_{cj})^2}{2\sigma_{cj}^2}} \quad \textbf{(2.14)}$$

where the probability density function is formed during the training stage of local classifiers with the mean and standard deviation of each two-class data for each feature vector component x_j.

Note that P(X|C$_j$) can be calculated by using equation (2.15) with each dimension independently because NBC adopts the Naive Bayes classifies as its local classifiers. The decision-making procedure for a given data is that the feature vectors are pass through corresponding local classifiers and the class for the given data is found by the following equation [**50**]:

$$Class(x) = argmax_k \frac{1}{N} \sum_{j=1}^{N} P(C_{ij}) \prod_{i=1}^{n} P(X_i|C_{jk}) \quad (2.15)$$

2.6 Support Vector Machines (SVM)

Supervised datamining Support Vector Machine (SVM) is an algorithm that used for both methods (regression or classification) tasks. Furthermore, it is often employed in the classification problems. Each data argument in this algorithm is represented by designed as an item in n-dimensional space (n is denoted as a features number or space). The value of a particular coordinate is molded by taking the value of each feature. Then, the classification is performed by determining the hyperplane (decision line) that discriminates the two classes effectively.

2.6.1 SVMs Classification of Linearly Separable for Two-Class Problems

Support Vector Machines (SVMs) are measured important approach as part of datamining procedures that was first developed and used by Vapnik [**51**]. They rely on designed a hyperplane (separating function) with the activation (objective function). It is used for different classes of data point separation. The separating hyperplane must have farthest distance as large as possible of each sample to the nearest data sample within each class for the whole data points.

A. The Primal Problem of the Linearly Separable Data

The fundamental idea of the primal problem of the linearly spreadable data of the SVMs is to rely on the linearly separable class problems to design the SVMs classifier. In this case, suppose we have the following set of training vectors as it defined

$D = \{(X_1, Y_1), \ldots (X_1, Y_1), X_i \in \{-1,1\}\}$, where $y_i \in \{-1,1\}$, and each one is denoted as a different class. Class 1 (positive sample points) and class 2 (negative sample points). In this case, if we consider that there exists a hyperplane function (H) with orthogonal to the hyperplane normal vector (w) and bias (b) and the distance $d(w, b;\ x)$ of a point x from a hyperplane (w, b) is defined by the following equation **[52]**:

$$d(w, b;\ x) = \frac{|\langle w, x\rangle + b|}{\|w\|} \qquad \textbf{(2.16)}$$

Where w: is signified vector orthogonal to the hyperplane

b : Controls the distance of hyperplane to the source point [**52**].

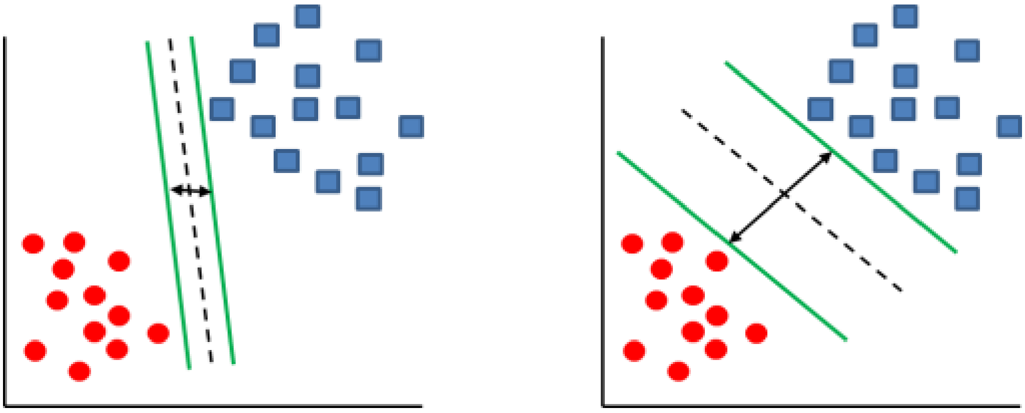
In this case, the Euclidean length which is clear as $\|w\|$, and it is clear as $\|w\| = \langle w, w\rangle$ by combining the set of inequalities that have been publicized in equation (2.16) gives the inequality as it defined by the following equation

$$y_i[\langle w, x_i\rangle + b] \geq 1, i = 1,2, \ldots, l \qquad \textbf{(2.17)}$$

Therefore, a separating hyperplane necessity satisfy this constraint. The maximum distance of a point to the separable (hyperplane) is also named the margin which is maximized subject to the same conditions as it defines in equation (2.18). In this case, the margin (hyper plain distance) denoted by M, which is defined by the following equation: [**51**]

$$M(w,b) = \min_{x_i:y_i=-1} d(w,b,x_i) + \min_{x_i:y_i=-1} d(w,b,x_i) \qquad \textbf{(2.18)}$$

Figure (2.19) explains an example of different margins of the "linearly separable" two-class problems. In Figure (2.19), (a) here the separated distance line between two classes (separable line) is small which reasons a small margin area. In contrast with Figure (2.19) (b) where the separated distance between the two classes (separable line) is large as possible and although reasons a large margin area. A large margin is more favorite because with max margin, few possibilities to isolated the data exist [**52**].



(a) **(b)**

Figure 2.19 (a) Small Margin (Hyperplane), (b) Large Margin (Hyperplane)

2.6.2 Kernels of Non-linear SVM

Non-linear separable for two type problem may not be completed in case of the data samples are too noisy. In this case, an method to overcome the problem for the two type noisy data can be traced back to the work that has been designed by "Vapnik and Cortes" In their style, the data samples are transferred to a "higher dimensional" space where in this field it will be linearly separated [**53**].

A. Higher Dimensionality Feature Space

Assume that the input vector $x = (x_1, \dots, x_n)'$, a transfer of the input vector into a feature space $\mathcal{F}$, which is achieved by the equation that is define in (2.19):

$$\varphi(x) = \varphi(x_1, x_2, \dots, x_n) \qquad (2.19)$$

Where x is represents the training data in input space.

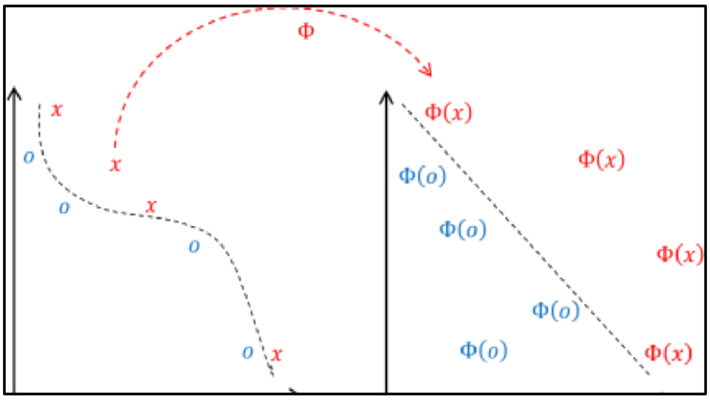


Figure 2.20 SVMs Feature Mapping

The mapping function is defined as $\mathbb{R}^n \longrightarrow \mathcal{F}$ where $\mathcal{F}$ is represented as an inner product function space. Figure (2.20) shows an example of mapping an input vector that has been moved into higher dimensional space. In this figure it is obviously to note that the mapping of the feature from input in form of two-dimensional space into a higher two-dimensional feature space where the linear separable line has not been performed in input space but happened in feature space [**53**].

In a binary classification problem, with the input feature vector x, the impartial is to specify a separating line after converting the vector to a feature space determining which of the two types each example point belongs. In this case, the data sample is mapped to another space such that $x \longrightarrow \varphi(x)$, where the classification function will build on the sign of the function $f(x) = \langle w, \varphi(x) \rangle + b$ [**53**]. Then, the orthogonal weight vector w has a new form that is defined in equation (2.20):

$$w = \sum_{i=1}^{l} \alpha_i y_i \varphi(x_i) \qquad (2.20)$$

The orthogonal weight vector w is performed in a new and higher feature space where data samples xi is plotted into a new space $\varphi(x_i)$.

B. Kernels and types

The feature mapping in a higher of dimensionality space of the non-linear separable class problem, to higher dimensional space is assisted to perform the linearly separable function. In this case, the input point $\langle x_i, x_j \rangle$ can be explicit in inner multiplication form $\langle \varphi(x_i), \varphi(x_j) \rangle$, it follows that shortest computations of inner product that happened in the feature space may be reached without the use of mapping functionφ. The function that performs like these computations is called a Kernel which is a function K that is clear in the equation (2.21):

$$K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \qquad \textbf{(2.21)}$$

Where φ: is a mapping from of an input space to feature space that is defined in the equation (2.22):

$$\varphi : \begin{cases} R^n \longrightarrow \mathcal{F} \\ x \mapsto \varphi(x) \end{cases} \qquad \textbf{(2.22)}$$

A feature map can be achieved in two steps. It involves a non-linear transformation of data into the feature space and then achieves linear classification in the feature space. Kernels can be used to plot data into a feature space. Training can take place in that space and does not need knowledge of the feature map [**53,54**].

C. Examples of kernels

The public example of the SVM higher dimensionality kernel that is commonly used for the SVMs classification are:

- Polynomial kernel which is describe:

$$K(x_i, x_j) = \left(\langle x_i, x_j \rangle + C\right)^d \qquad \textbf{(2.23)}$$

- Radial Basis Kernel Function (RBF) which is describes

$$K(x_i, x_j) = e^{\frac{-\|x_i, x_j\|^2}{2\sigma^2}} \qquad \textbf{(2.24)}$$

- Sigmoid kernel function which is describe

$$K(x_i, x_j) = tanh\left(K(x_i, x_j) + \Theta\right) \qquad \textbf{(2.25)}$$

Where d, σ and Θ are parameters of the specific kernels [**51**][**52**].

2.7 Evaluation Measures

To calculated evaluating act of proposed system may using three measures called Classification Rate, Detection Rate, and False.

The methods that are used for measurement computing are given as in equations blow.

A. Accuracy or Classification Rate:

The Classification Rate (Classification ratio or accuracy) is clear as the ratio between the actual numbers of true Classification to the total number of sample testing (total) that are applied during the training and testing. The formal formula of the Classification rate it is definite [**45**].

$$Accuracy = \frac{TP+TN}{Total\ number\ of\ test\ samples} * 100 \quad \textbf{(2.26)}$$

The estimating performance of the fully robotic skin cancer classification approach is measured by volume of how many true predictions that has been counted. The formal sensitivity, specificity and precision that have been used to measure the results of the proposed scheme are defined in equations blow.

B. Sensitivity or Recall:

Sensitivity is mainly calculated to define the ratio between the numbers of correct positive prediction and the total number of the positive prediction [**45**].

$$Sensitivity = \frac{No.of\ true\ positive\ prediction}{\sum Number\ of\ all\ positive\ assesment = TP+FN} \quad \textbf{(2.27)}$$

C. Specificity:

Specificity is denoted the ratio between the numbers of correct negative prediction and the all number of negative predictions [**45**].

$$Specificity = \frac{No.of\ true\ positive\ prediction}{\sum Number\ of\ all\ negative\ prediction = TN+FP} \quad \textbf{(2.28)}$$

D. Precision:

Precision mainly calculated to define the ratio between the total numbers of true prediction and relevant to the total number of correct cases during our training/testing performance [**45**].

$$Precision = \frac{No.of\ true\ positive\ prediction}{\sum Number\ of\ all\ positive\ assesment = TP + FP} \qquad \textbf{(2.29)}$$

Chapter Three

Proposed System Design

3.1 Introduction

This chapter presents a proposed skin cancer classification by using CNN, Naive Bayes and Support Vector Machine algorithms. It will start by presenting and discussing the general block diagram of the normal skin and with cancer for classification system proposed in a separate section. Moreover, the chapter will define the system building in details with the projected algorithms in the several stages of the projected system.

3.2 The Proposed System Design

The projected approach depends on three main steps to predict and classify the skin cancer. The input to the system is an RGB image, and the output is a classification of skin cancer. In this thesis, three types of systems were used, (the first system with (CNN), the second system with Naive Bayes and the third with Support vector machine (SVM), each system is divided into two type.

Each proposed system has many stages and each one has several phases that working together to accomplish the goals of first system. The first system divided in to two type which contain the following stages: image acquisition, preprocessing, and classification, while the second part consist of image acquisition, classification. Figure (3.1) shows the general block diagram of the proposed model to classification the skin cancer from images.

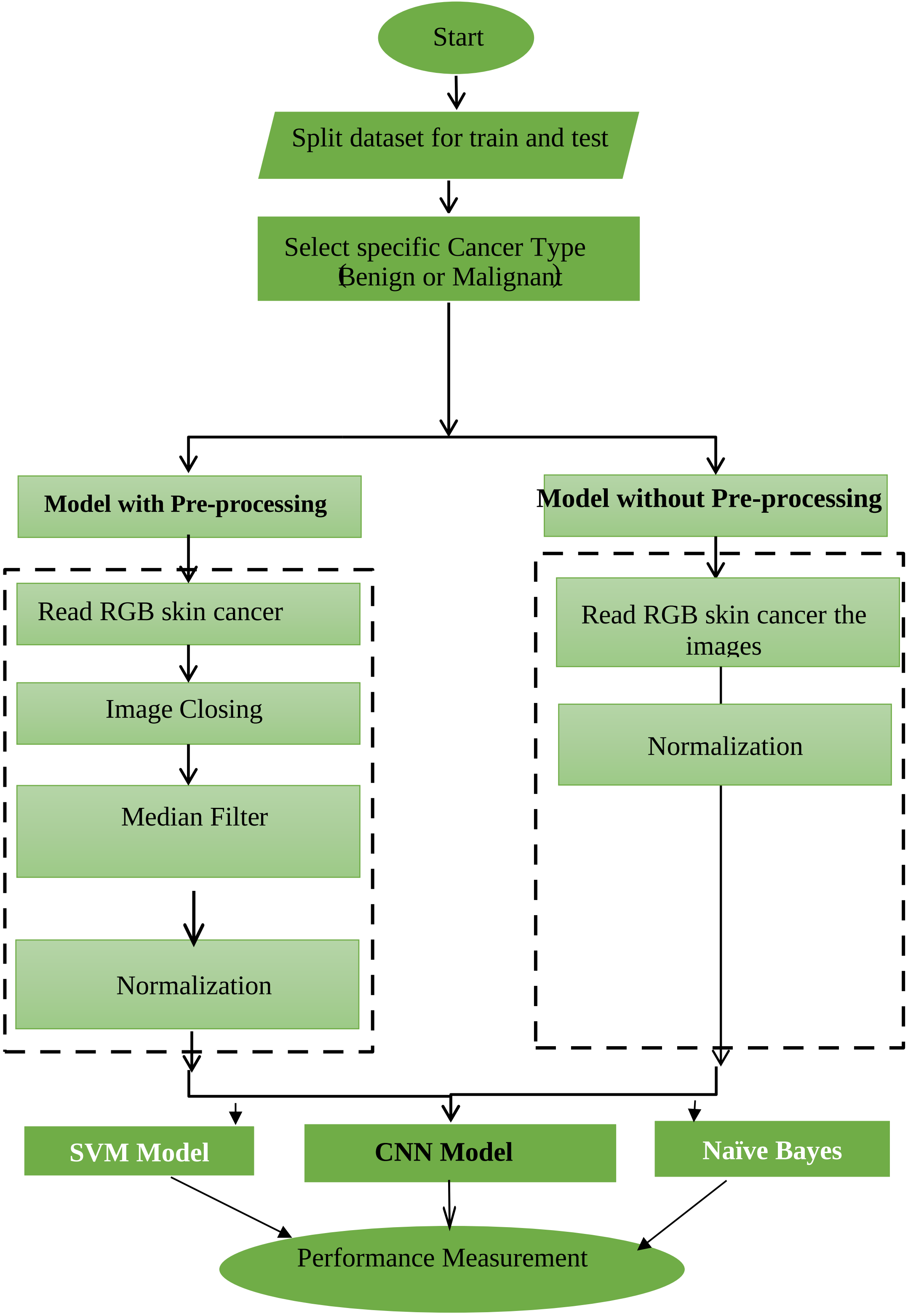
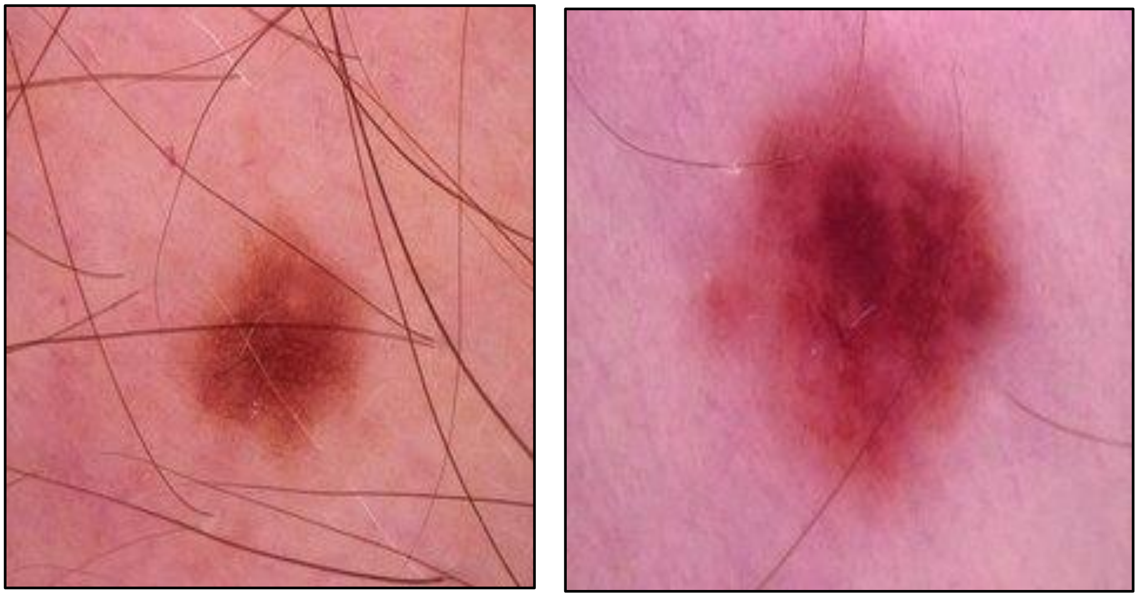


Figure 3.1: Block Diagram of General Proposed System

3.3 Image Acquisition Stage

In order to classify skin cancer images by using the proposed systems, dataset was collected from the source for different categories of the most common skin cancer. In this system three techniques were used for classification CNN, Naïve Bayes and SVM. The classification system was implemented on image dataset, gained from ISIC-Archive rights, as there is in the database is available from these types of skin cancer. Figure (3.2) presents the samples of these types of skin cancer.



Benign

Malignant

Figure (3.2): Samples of Tow Type Skin Cancer Images

3.4 Image Preprocessing Stage

In preprocessing image step the proposed system consists of multiple steps, they are:

3.4.1 Normalization for Image

Image normalization is a process in which change the range of pixel intensity values to make the image more familiar or normal to the senses, hence the term normalization. So, in our model we normalize all values of the pictures by dividing all the RGB values by 255.

3.4.2 Hair Removal Algorithm

- **Image Closing Operation**

Closing operation is an important operator from the field of mathematical morphology that can remove the thin hair from skin cancer image. Based on the assumption that hair segments are thin structures. As both erosion and dilation processes use the same structuring element (disk shape). First dilation process pulls the pixels, meaning the most numerous pixels are spread (the background or skin color), which weakens the hair color, or it expands the hair pixels and spreads them on the background. Then it is followed by the erosion process, which narrowing the pixels to mix with the background (skin color). Algorithm (3.1) is summarized the steps of image closing operation.

Algorithm (3.1): Closing Operation

```
Input: Image with size 224*224    L= 223, n=8
Output: RGB Skin cancer image
Begin
Step (1): Input an image
Step (2): The two-dimension window of size 3×3 is selected from image
Step (3): For i = 1 To L-1
Step (4): For j = 1 To L-1
Step (5): image closing for each layer (dilation then erosion)
      R layer
     G layer
    B layer
Step (6): Save result for next stage.
END
```

3.4.3 Enhancement Image by Median Filter

After the closing operation is performed on the skin cancer image, the median filter comes in to play that attempts to make the whole image the same degree of color and depends on the most prominent color, which is the (skin color). Median filter makes it suitable for the next step. the window size of the median filter selected is 3×3. Median filter algorithm is summarized by the algorithm (3.2).

Algorithm (3.2): Median Filter

Input: RGB Image from Image Closing

Output: Enhancement RGB Image by Median Filter skin cancer Image

```
Begin
Step (1): Input an image from image closing phase
Step (2): The two-dimension window of size 3×3 is selected from the image (for each layer R, G, B)
Step (3): For i=1 to width -1
Step (4): For j=1 to height -1
Step (5): Sort the pixels in the selected window in ascending order
Step (6): Set n ← pixel (i,j) // Choose the median value from the array as the new value of the image
Step (7): Set RGB (x,y) ← n // Replacing the pixel being considered with the median pixel value
          End For
          End For
END
```

To obtain the clear image for skin cancer, a median filter is applied after image closing operation. This operation is summarized by the algorithm (3.3).

Algorithm (3.3): Hair Removal

```
Input: RGB Image with size 224*224
       L= 223, n=8
Output: RGB Skin cancer image

Begin
Step (1): Input an image
Step (2): The two-dimension window of size 3×3 is selected from image
Step (3): For i = 1 To L-1
Step (4): For j = 1 To L-1
Step (5): image closing for each layer (from algorithm 3.1)
      R layer
    Apply median filter with size (3*3) from algorithm (3.2)
     G layer
    Apply median filter with size (3*3)
    B layer
    Apply median filter with size (3*3)
Step (6): Test the removal hair algorithm on image
             load the image
           Repeat from step 2 to 5
Step (7): Display tested image
END
```

3.5 Proposed Systems based on CNN and Naive Bayes and SVM

Our proposed approach combines three type CNN, Naive Bayes and SVM which used to improve the accuracy of the classification. The proposed algorithm consists of two parts used, pre-processing and without pre-processing.

3.5.1 Convolutional Neural Network (CNN) Algorithm

The Convolutional Neural Network is a class of deep artificial neural networks, and the most important characteristics of CNN are the local contact between the layers and the use of common weights between them; Therefore, it can learn the local features of the input image of skin cancer.

Figure (3.4) represents the block diagram of the first proposed model for both preprocessing and without for images skin cancer classification.

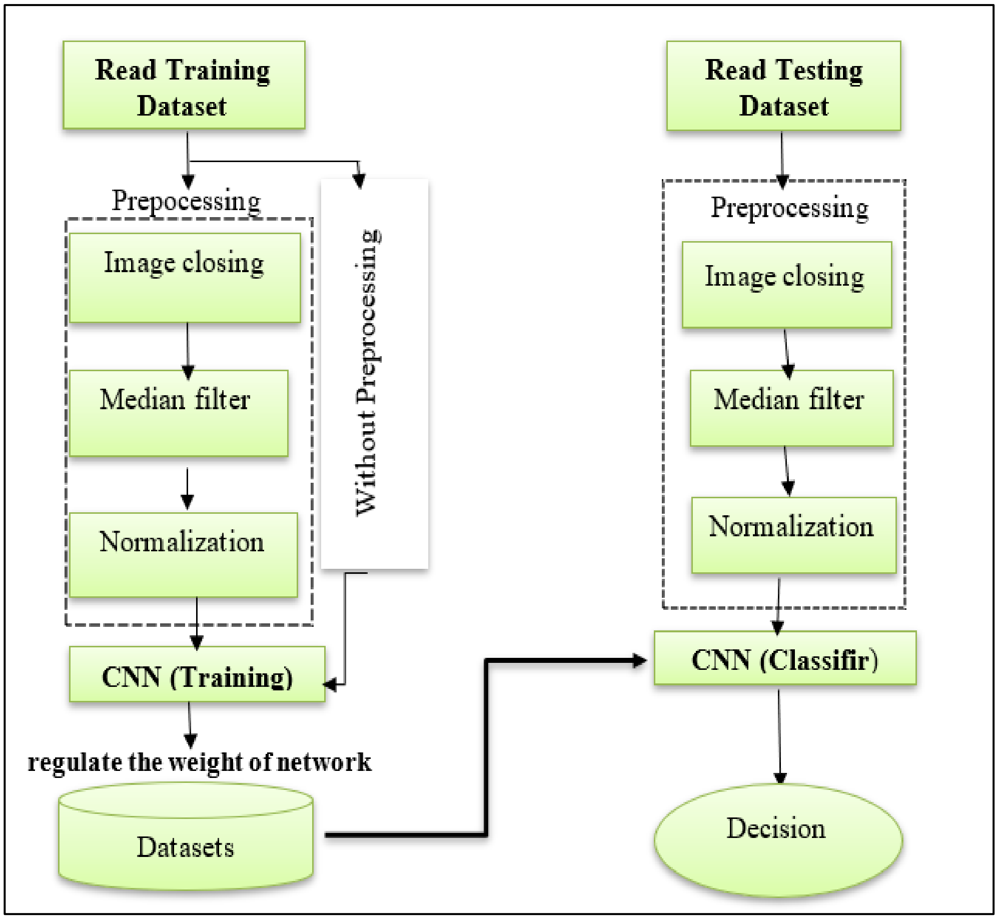


Figure (3.3): Block Diagram of First Proposed Model

Algorithm (3.4): The proposed CNN classifier without Preprocessing

Input: RGB Image of Skin Cancer with size 224*224

Output: A class Name

Begin

Training phase (80) % of datasets was taken for training.

Forward Propagation.

Step (1): Input layer: RGB image with 224*224 size

Step (2): Convolution layer: Multiple filters were used with size (3*3) with "zero" padding.
Input: $W_1*H_1*D_1$ //*W is the height, H is the width, D number of filters*
Output: Feature Map as Eq : $O \leftarrow 1+\frac{(N+2P-F)}{S}$ *//Where O is the output, N is the input size, F is the filter size, and S is the stride size, P is the number of the layers of the zero-padding*

Step (3): Activation layer (Nonlinear layer):$\mathrm{ReLU}(\mathrm{x}) \leftarrow \max(0, \mathrm{x})$
// *to adjust or cut-off the generated output.*
Step (4): Normalization layer: Using batch normalization.

Step (5): Pooling layer: Using max-pooling with (2*2) window
Input: activation map from convol layer of size $W_1*H_1*D_1$
Output: feature map of size $W_2*H_2*D_2$ with reduction of dim

$W_2 \leftarrow (w1-F)/S+1$

$H_2 \leftarrow (H1-F)/S+1$

$D_2 \leftarrow D_1$, *W is the height, H is the width, D number of filters, F is the filter size, and S is the stride size.*

Step (6): Dropout layer: During training, randomly ignore activations by probability ***p.*** by randomly setting the output for a given neuron to 0.
Step (7): Fully connected layer: feed forward neural networks
Input: The output from the final Pooling layer
Output: Flattened vector by calculate Eq : $g(Wx+b)$ where ***x*** *is the input vector with dimension* ***[p_l, 1]***, ***W*** *is the weight matrix with dimensions* ***[p_l, n_l]*** *where,* ***p_l*** *is the number of neurons in the previous layer and* ***n_l*** *is the number of neurons in the current layer,* ***b*** *Is the bias vector with dimension* ***[p_l, 1], g*** *Is the activation function, which is usually* ***ReLU***.

Step (8): SoftMax layer
Input: (logits vector) the output from last layer of neural network.
Output: is a probability distribution of all the label class candidates.

$S(Yi) = \frac{e^{yi}}{\sum_j e^{yi}}$ *where* $S(Yi)$ *is the softmax for numeric output of the last linear layer,* e^{yi} *is exponents of output neural network.*

Step (9): for k=1 to the number of neurons in the output layer find
Total Error By using equation (2.6).
End for

Back Propagation

Step (10): If (Error ≥ 1) back propagation for all layer n-1 to layer 1 update the weight by using Eq:$(w_{jk})new \leftarrow (w_{jk})old - η \times \Delta(w_{jk})$
Step (11): Repeat step (9) and (10) until gets the desired outputs
Testing phase (20) % of datasets was taken for testing

Step (12): Input (X)is the sample of testing dataset**.**

Step (13): Repeat same steps (2-8) and load Training model [only forward propagation]

Step (14): Find class Name

END

3.5.2 Convolutional Neural Network Working

After the preprocessing phase, RGB skin cancer image was taken with 224*224*3 to the first layer the convolutional (Conv2D) layer

- Applying set of learnable filters that chosen. We choose a set filters for the firsts conv2D layer. Each this filter moved a part of input image (defined 3*3 of kernel size) using this kernel filter.
- Used the second important layer (MaxPool2D) layer. This layer simply acts as a down sampling filter. It looks at the 2 neighboring pixels and picks the maximal value. These are used to reduce computational cost, and to

some extent also reduce overfitting. We have to choose the pooling size (i.e. the area size pooled each time)

- Dropout layer to regularization our network, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample. We chose 25% to dropped net, this technique also improves generalization and reduces the overfitting.

Repeated the above steps (convolution, activation, normalization, pooling and dropout) until we reach to the last convolutional layer.

- Finally, the flatten layer in our network is used to convert the final (feature map) into one single (1D) vector. The need for this step is to take full advantage of the connected layers after two layers convolutional and max-pool layers. Its associations all the local features that found of the preceding convolution layers. We get from last pooling layer output of 18*18*128 and that equal 41472 output shape that deals as a neural network, 41472 needed 1024 neurons. Then the SoftMax comes into play to calculate the probability of both classes malignant and benign.

Figure (3.4) shows the CNN architecture of the multi-channel input image to the output function exploiting full color information in RGB stained images, and includes a pooling layer which improves the classification.

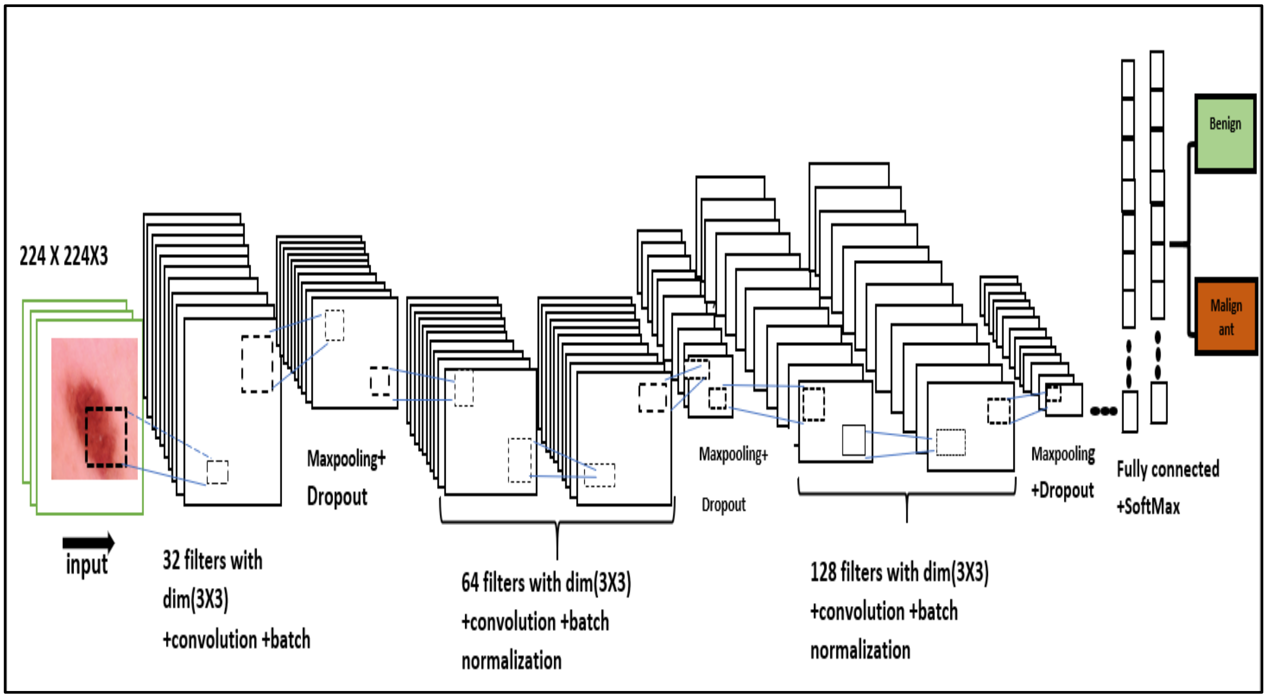


Figure (3.4): Represents the Architecture of CNN

The same steps were repeated to propose CNN classifier but with preprocessing. algorithm (3.5) clear these steps

Algorithm (3.5): The proposed CNN with Pre-processing

```
Input: Training data set TD,
Output: class name
Begin
Training phase (80) % of datasets was taken for training.
Step (1): Read RGB image with 224*224*3 size
Step (2): Apply Preprocessing on images as algorithm (3.3)
Step (3): Repeat the same steps (2 to 11) in algorithm (3.4) //For training stage.
Testing phase (20) % of datasets was taken for testing
Step (4): Repeat the same steps (12 to 14) in algorithm (3.4) //For testing stage.
END
```

Training Network

The training process starts by reading the model name, number of epoch and batch size received from the user. Then the system reads the dataset with benign and malignant categories. After that all the images from the dataset are preprocessing with (hair removal algorithm) and the dataset augmentation will be generated. The system starts to train the network as much the number of epochs inputted by the user earlier. The training will produce a probability value for the two classification classes, where the class with the greatest probability value is the classification class predicted by the program. The training results are then stored in the form of a model file. After completing the training, the system will save the model and plot from the results of the training.

In this training there are parameters that are run constantly throughout the procedure, namely learning rate, batch size and optimizer. The learning rate used is 0.0001, where this parameter states the constants for learning speed from the network layer used. While the batch size parameter serves to determine the total amount of data used in one batch of training, the batch size used is 64. Determination of batch size is considered from the memory capability of the device used to conduct the training process, and optimizer is (Adam).

3.5.2 Naïve Bayes Algorithm

The basic idea is to use the joint probabilities of features and classes to estimate the probabilities of classes given a feature. The naïve part of such a model is the assumption of feature independence.

Classifier based on the Naive Bayes algorithm. In order to find the probability for a label, this algorithm first uses the Bayes rule to express P (label |features) in terms of P(label) and P (features |label). The algorithm then makes the 'naive' assumption that all features are independent, rather than computing P(features) explicitly, the algorithm just calculates the numerator for each label, and normalizes them so they sum to one

Training step the classification algorithm was applied by selecting a class to be used for training and testing. The main purpose of the training step is computing prior probability of each class P(C_i) and likelihood p (X| C_i)). To build the feature vector of class *Ci* where *i* is the number of classes. Each class *Ci* has a set of training images *{P1, P2, P3, …Pn}* where *n* is the features number of in each class.

Prediction step: For any unseen test data, the method computes the posterior probability P (C_j| X) of that sample belonging to each class. The method then classifies the test data according the largest posterior probability.

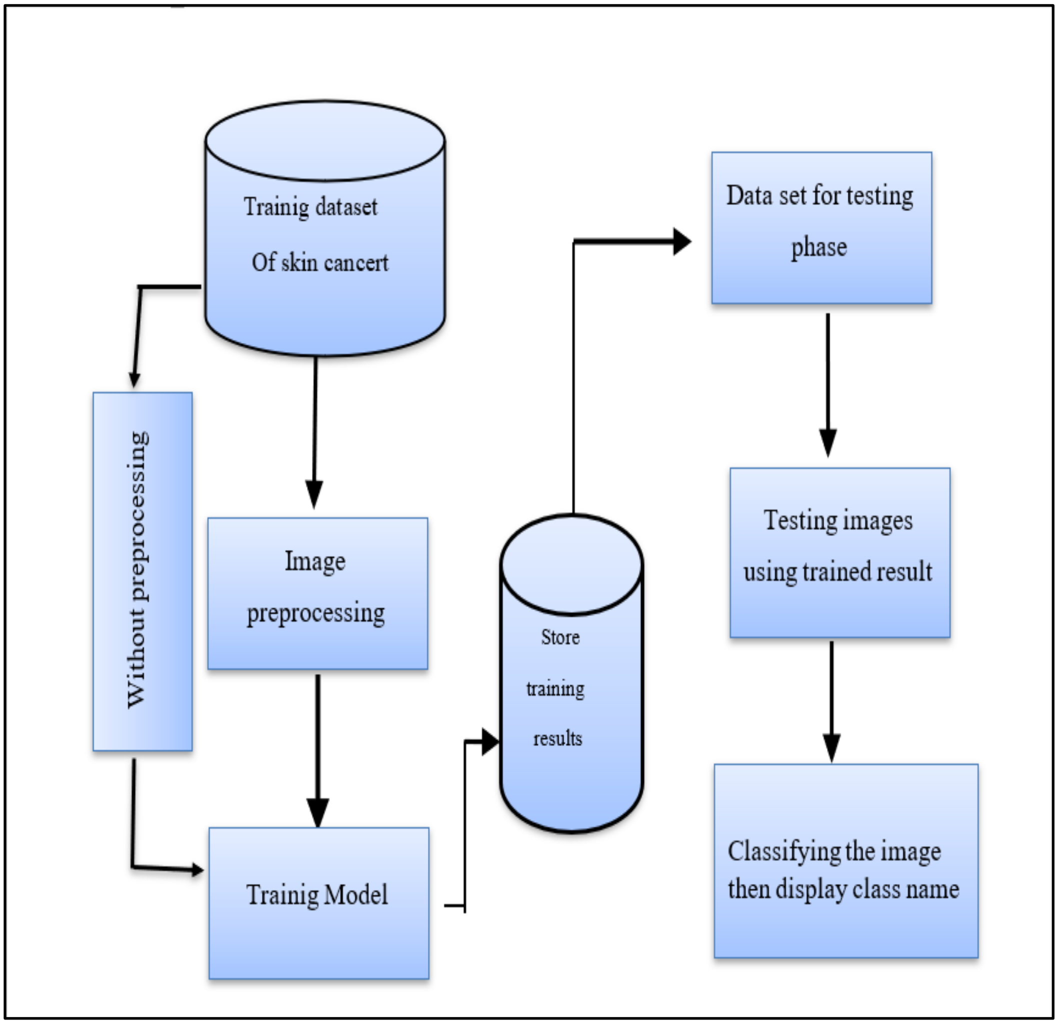


Figure (3.5) Represents Block Diagram of the of Naïve Bayes

Algorithm (3.6): Naive Bayes algorithm -without Preprocessing

```
Input: Training data set TD,
Output: Class Name

Begin
Step (1): Read training Dataset DS.
Training phase (80) % of datasets was taken for training.
Step (2): (Total) = all examples in training dataset.
$C_j$ refer to class in training DS.
Step (3): Calculate the Probability for each class.
P ($C_i$) = frequency ($C_i$) / total.
Step (4): Calculate the mean (to feature) (μ) and standard deviation (to feature) (σ) values for each feature in each class in TD and store the result.
Testing phase (20) % of datasets was taken for testing
Step (5): X is tested example in the testing DS.
Step (6): Calculate the probability density function (pdf) of X at $C_i$ for values of features of (X) exists in S, $P(X|C_i)$ by apply the equation (2.14).
Step (7): Calculate conditional probability of(X) at C j for values product from step (6), by apply equation. $P(X|C_i) = \coprod_{j=1}^{n} P(f_j|C_i)$
Step (8): Calculate posterior probability of(X), $P(C_i|X)$ that denote probability of example at $C_i$ by equation following:
  $P(C_i|X) = P\left(X\middle|C_j\right)p(C_j)$ --- probability of feature at ($C_i$).
Step (9): Select class label to the class (X) by choice maximization p ($C_j|X$).
Step (10): Return Class Name.
END
```

The same steps were repeated for the proposed Naïve Bayes classifier but with applying preprocessing. The steps are taken to clear these steps for Naïve Bayes algorithm are listed in the algorithm (3.7).

```
Algorithm (3.7): Naive Bayes algorithm -with Preprocessing

Input: Training data set TD,
Output: Class Name

Begin
Step (1): Read the training Dataset DS.
Step (2): Apply preprocessing on images as algorithm (3.3).
Training phase (80) % of datasets was taken for training.
Step (3): Repeat the same steps (2 to 4) in algorithm (3.6) //For the training stage
Testing phase (20) % of datasets was taken for testing
Step (4): Repeat the same steps (5 to 10) in algorithm (3.6) //For the testing stage.
END
```

3.5.3 Support Vector Machine Algorithm

SVM involves two phases: a training phase, and testing phase. The training phase learns a model by which to classify new test examples. The test phase applies the model to classify the test examples. SVM algorithm is based on the concept of ‘decision planes’, where hyperplanes are used to classify a set of given classes. It finds lines or boundaries that correctly classify the training dataset, then, from those lines or boundaries, it picks the one that has the maximum distance from the closest data points.

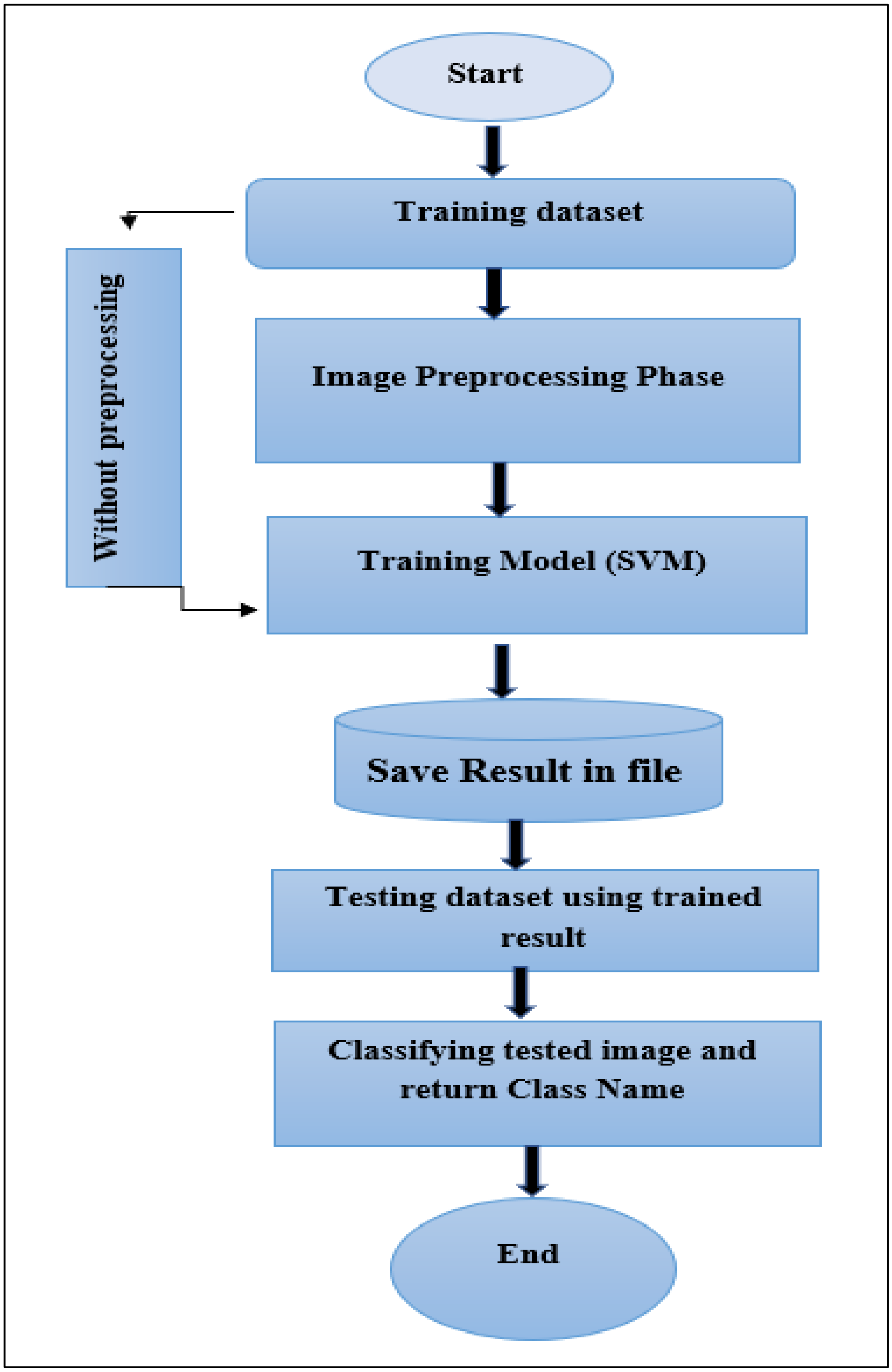

Start
Training dataset
Without preprocessing
Image Preprocessing Phase
Training Model (SVM)
Save Result in file
Testing dataset using trained result
Classifying tested image and return Class Name
End

Figure (3.6) Block Diagram of the of Support Vector Machine

Algorithm (3.8): SVM Algorithm without -Preprocessing

```
Input: Training data set TD,
Output: Class Name

Begin
Step (1): Split dataset for training and testing.
Training phase
Step (2): Read training dataset DS.
Step (3): Calculate the hyperplane $(w)$ for all Support Vector point as equation (2.13).
Step (4): Calculate the distance of a point (x) from a hyperplane for each Support Vector Points    as equation (2.15).
Step (5): Find maximum distance of a point to the hyperplane (margin) between two classes by apply (2.17).
Step (6): Repeat steps (2–5) for all points then store results.
Testing phase (20) % of datasets was taken for testing
Step (7): Read testing dataset, X is a feature vector.
Step (8):  Compute value $(w)$ for all tested dataset (X).
Step (9): Select class label to class (X) by choice maximization $M(w|x)$.
Step (10): Return Class Name.
END
```

The same steps were repeated for the proposed SVM classifier but with preprocessing. The steps are taken to clear these steps for SVM algorithm are listed in the algorithm (3.9).

Algorithm (3.9): SVM Algorithm with Pre-processing

Input: Training data set TD,
Output: Class Name

```
Begin
Step (1): Split dataset for training and testing.
Step (2): Apply Preprocessing on images as algorithm (3.3).
Training phase (80) % of datasets was taken for training.
Step (3): Repeat the same steps (2 to 5) in algorithm (3.8) //For the training stage.
Testing phase (20) % of datasets was taken for testing.
Step (4): Repeat the same steps (7 to 10) in algorithm (3.6) //For the testing stage.
END
```

3.6 The Proposed Classification System Stage

In the classification part, the training formula has an input consisting of examples in the form of vectors of the features with a label attached to them. The purpose of the classification algorithm is to learn how to determine the correct labels of samples. It is considered new and unseen to learning model task as shown in Figure (3.8).

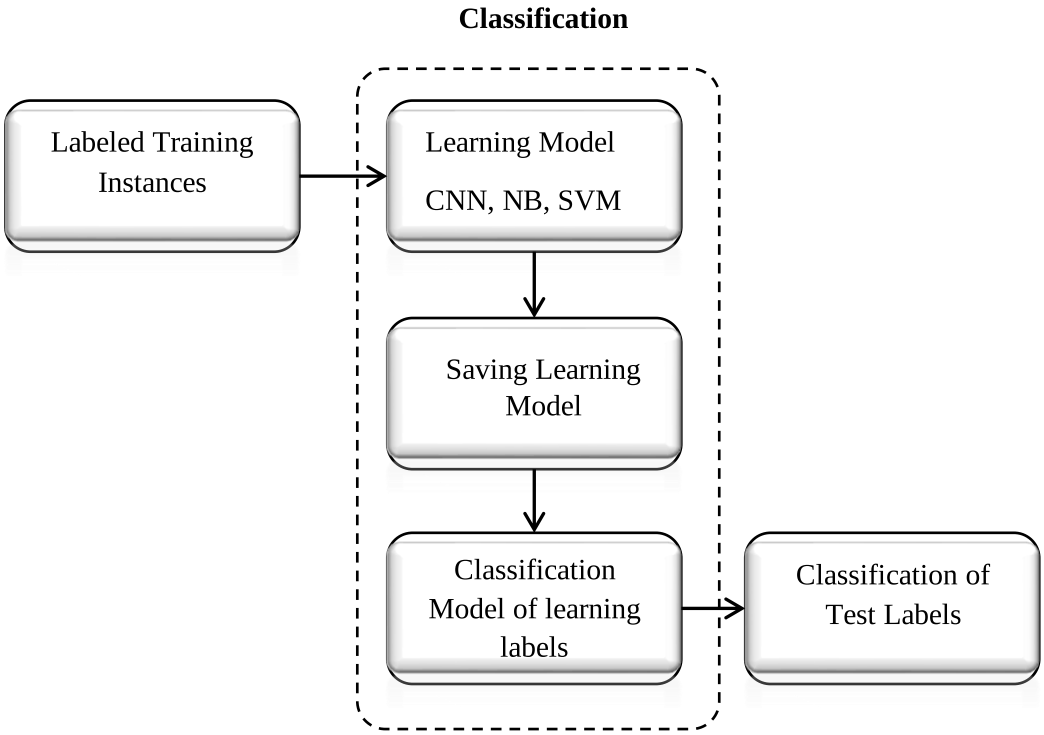

Classification
Labeled Training Instances
Learning Model
CNN, NB, SVM
Saving Learning Model
Classification Model of learning labels
Classification of Test Labels

Figure (3.7) Block Diagram of Classification Approach

Three proposed systems (the first proposed system (CNN), the second proposed system (NB) and the third Support Vector Machine (SVM) were used to classify the skin cancer images as illustrated in the following section. The main goal of the proposed methods is to classify skin cancer in a given image as benign or malignant. In order to better achieve this goal, then the model have to learn such powerful features that are able to recognize and classify the mole image to binary classification since we have two classes for the problem domain (Benign and Malignant).

Each algorithm in proposed scheme contain two main stages that are training stage and testing stage, for these stages the skin cancer image database folders will be divided into two parts the first (80%) is utilized to train skin cancer image and extract features vectors which are stored as numeric vectors in database and the second (20%) is utilized as test samples to test system matching performance.

In the training phase, a number of the samples have been utilized for training, these data will get through the stages of the proposed system, and finally saved as features vectors and stored in the database to be utilized within samples compared in the testing phase. Training of the proposed system is carried out on two types of skin cancer diseases (benign type and malignant type). The proposed system trains 1440 images of the benign type group and 1197 of the malignant type group. This means that the proposed system contains 2637 trained samples stored in the database.

In the testing phase, the sample that was left of each folder in training phase will be utilized to test the system performance those samples must pass the stages of the system to extract features vectors and compare with all database fields. Depending on pattern classification measure decision will be made to decide if the testing sample is accepted or rejected.

3.6.1 First Proposed System CNN Classification

The first proposed system was using CNN algorithm stage with and without preprocessing. The main goals of the proposed model are to classify skin cancer in a given image as benign or malignant.

The algorithm (3.10) proposed an algorithm for testing phase by (CNN) algorithm without preprocessing.

Algorithm (3.10): CNN classifier without Pre-processing

```
Input: Training dataset TD, Testing dataset TS.
Output: Accuracy of Testing Model

Begin
Step (1): input dataset (image for skin cancer with 224*224 dimension)
Step (2): Create labels for benign and malignant //zeroes for benign and
          ones for malignant.
Step (3): Shuffle data // for training dataset and testing dataset
Step (4): Categorical Labels // Turn labels into one hot encoding to deal
          with non-numerical values
Step (5): Normalization // Normalize in rang [0,1] all Value of image by dividing them RGB values by 255.
Step (6): Load Model Building using CNN
Step (7): Train the Model for new sample of skin cancer images
Step (8): Return Accuracy of Testing Model
END
```

While the second model with pre processing

Algorithm (3.11): CNN classifier with Preprocessing

```
Input: Training dataset TD, Testing dataset TS.
Output: Accuracy of Testing Model.

Begin
Step (1): Read dataset (image for skin cancer with size 224*224 dimension)
Step (2): Apply Preprocessing on images as algorithm (3.3).
Step (3): Create labels for benign and malignant //zeroes for benign and
          ones for malignant
Step (4): Shuffle data // for training dataset and testing dataset
Step (5): Categorical Labels // Turn labels into one hot encoding to deal
          with non-numerical values
Step (6): Normalization // Normalize in rang [0,1] all Value of image by
dividing them RGB values by 255.
Step (7): Load Model Building using CNN
Step (8): Test the Model for new sample of skin cancer images
Step (9):  Return Accuracy of testing Model.
END
```

3.6.2 Second Proposed System NB Classification

The second model used Naive Bayes were used to train the model and then testing in order to classify and its divided into two model, one without preprocessing and one with processing.

Algorithm (3.12): Naïve Bayes classifier without Pre-processing

```
Input: Training dataset TD, Testing dataset TS.
Output: Accuracy of Testing Model.
Begin
Step (1): Read dataset (images for skin cancer with 224*224 dimension)
Step (2): Create labels for benign and malignant //zeroes for benign and ones for malignant
Step (3): Shuffle data // for training dataset and testing dataset
Step (4): Categorical Labels // Turn labels into one hot encoding to deal with non-numerical values
Step (5): Normalization // Normalize in rang [0,1] all Value of image by dividing them RGB values by 255.
Step (6): Train model using Gaussian Naïve Bayes
Step (7): Calculate the mean (μ) and standard deviation (σ) values for each feature in each class in training dataset and store the result.
Step (8): Test the Model for new sample of skin cancer images
Step (9): Calculating Model Accuracy
Step (10):  Return Accuracy of Testing Model.
END
```

While the second model with pre processing

Algorithm (3.13): Naïve Bayes classifier with Pre-Processing

```
Input: Training dataset TD, Testing dataset TS.
Output: Accuracy of Testing Model.
Begin
Step (1): Read dataset (image for skin cancer with 224*224dimension)
Step (2): Apply Preprocessing on images as algorithm (3.3).
Step (3): Create labels for benign and malignant //zeroes for benign and ones for malignant
Step (4): Shuffle data // for training dataset and testing dataset
Step (5): Categorical Labels // Turn labels into one hot encoding to deal with non-numerical values
Step (6): Normalization // Normalize in rang [0,1] all Value of image by dividing them RGB values by 255.
Step (7): Train model using Gaussian Naïve Bayes
Step (8): Calculate the mean (μ) and standard deviation (σ) values for each feature in each class in training dataset and store the result.
Step (9): Test the Model for new sample of skin cancer images.
Step (10): Calculating Model Accuracy
Step (11):  Return Accuracy of Testing Model
END
```

3.6.2 Third Proposed System Support Vector Machine Classification

The third model used SVM were used to train the model and then testing in order to classify and divide into two types, one without preprocessing and other with preprocessing.

Algorithm (3.14): SVM classifier without pre processing

```
Input: Training dataset TD, Testing dataset TS.
Output: Accuracy of Testing Model.
Begin
Step (1): Read dataset (image for skin cancer with 224*224 dimension)
Step (2): Create labels for benign and malignant //zeroes for benign and ones for malignant
Step (3): Shuffle data   // distributed randomly for training dataset and testing dataset
Step (4): Categorical Labels // Turn labels into one hot encoding to deal with non-numerical values
Step (5): Normalization    // Normalize in rang [0,1] all Value of image by dividing them RGB values by 255.
Step (6): Train Model using (SVM).
Step (7): Calculating The highest margin hyperplane for each feature of each class in the training dataset then store results.
Step (8): Test the model for new sample of skin cancer images.
Step (9): Calculating model accuracy.
Step (10): Return accuracy of testing model.
END
```

While the same model to propose system SVM with preprocessing. Steps are summarized by the algorithm (3.15).

Algorithm (3.15): SVM classifier with pre-processing

Input: Training dataset TD, Testing dataset TS.

Output: Accuracy of Testing Model.

```
Begin
Step (1): input image (image for skin cancer with size 224*224*3)
Step (2):  Apply Preprocessing on images as algorithm (3.3).
Step (3): Create labels for benign and malignant //zeroes for benign and ones for malignant
Step (4): Shuffle data     //distributed randomly for training dataset and testing dataset
Step (5): Categorical Labels   //Turn labels into one hot encoding to deal with non-numerical values
Step (6): Normalization    // Normalize in rang [0,1] all Value of image by dividing them RGB values by 255
Step (7): Train Model using (SVM).
Step (8): Calculating the highest margin hyperplane for each feature of each class in the training dataset then store results.
Step (9): Test the Model for new sample of skin cancer images
Step (10): Calculating Model Accuracy
Step (11):  Return Accuracy of Testing Model
END
```

Chapter Four

Experimental Results and Discussion

4.1 Introduction

This chapter contains experimental results of the skin cancer classification, three classification algorithms are discussed in this approach. The first is the Convolutional Neural Network (CNN), the second is the Naïve Bayes (NB) and the third is the support vector machine (SVM).

4.2. Implementation Environment

Skin cancer images classification approach using CNN, NB, and SVM is implemented under a specific system requirement such as Windows-10 operating system, Hardware processor: Core i7- CPU 8550U, 200 GHz, and (8GB) RAM. Python 2018 (3.8 64-bit) programming language with TensorFlow backend, CNN programs implemented on Kaggle server.

4.3 Evaluation of Skin Cancer Systems

The experimental results system for the skin cancer classification approach using CNN, NB, and SVM has been tested by introducing the measures of validation for the results of classification.

A confusion matrix for the proposed system is represented as a matrix $[m * m]$, where m is referring to classes number. In general, confusion matrix that has been used as a performance measurement is technically done by the classification approach. The confusion matrix consists of the actual class as well as the predicated class for each classifier and shows that the predicted class wherever a corrected is classified or not. For this reason, it has been used to evaluate each classifier by measuring the performance results for each one. Table (4.1) shows a typical template of the confusion matrix.

Table (4.1) Confusion Matrix

Cancer case		**Predicted Class**	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Each of classifier (CNN, NB and SVM) would be evaluated by practicing various parameters such as the True Positive (TP), the False Negative (FN), the False Positive (FP) and finally the True Negative (TN). Each one has a specific meaning in the confusion matrix as it is shown below:

- **True Positive (TP):** These are cases in which predicted yes (they have the disease), and they do have the disease.
- **True Negative (TN):** predicted no, and they don't have the disease.
- **False Positive (FP):** predicted yes, but they don't actually have the disease.
- **False Negative (FN):** predicted no, but they actually do have the disease.

4.3.1 Performance Measure

The performance results of the proposed approach are based on two main phases the training and the testing phase. In the training phase, the scores model evaluation approach has been used as a training frame work in case to jump out the overtraining case that may the proposed system will face during the testing phase. The performance of a classification model relies on its effectiveness on recognizing the labels of unlabeled data. Then setting aside some of the labeled data for testing to evaluate the performance**.**

4.4 Skin Cancer Acquisition (Database)

The datasets that are used in proposed system has 3297. There are 1497 image cases of malignant skin caser type, and 1800 images cases for benign. The entire whole dataset material was collected from the ISIC (International Skin Image Collaboration) Archive [55]. The images of skin cancer have (24-bit) RGB color space where each (8 bits for each channel). Table (4.2) shows the distribution of skin cancer images.

Table (4.2) Distribution of Number Skin Cancer Images Dataset

Dataset	Training	Testing
Benign	1440	360
Malignant	1197	300

In this study, two types of skin cancer are used. First one is type of benign images that shown in figure (4.1) and malignant images shown in figure (4.2).

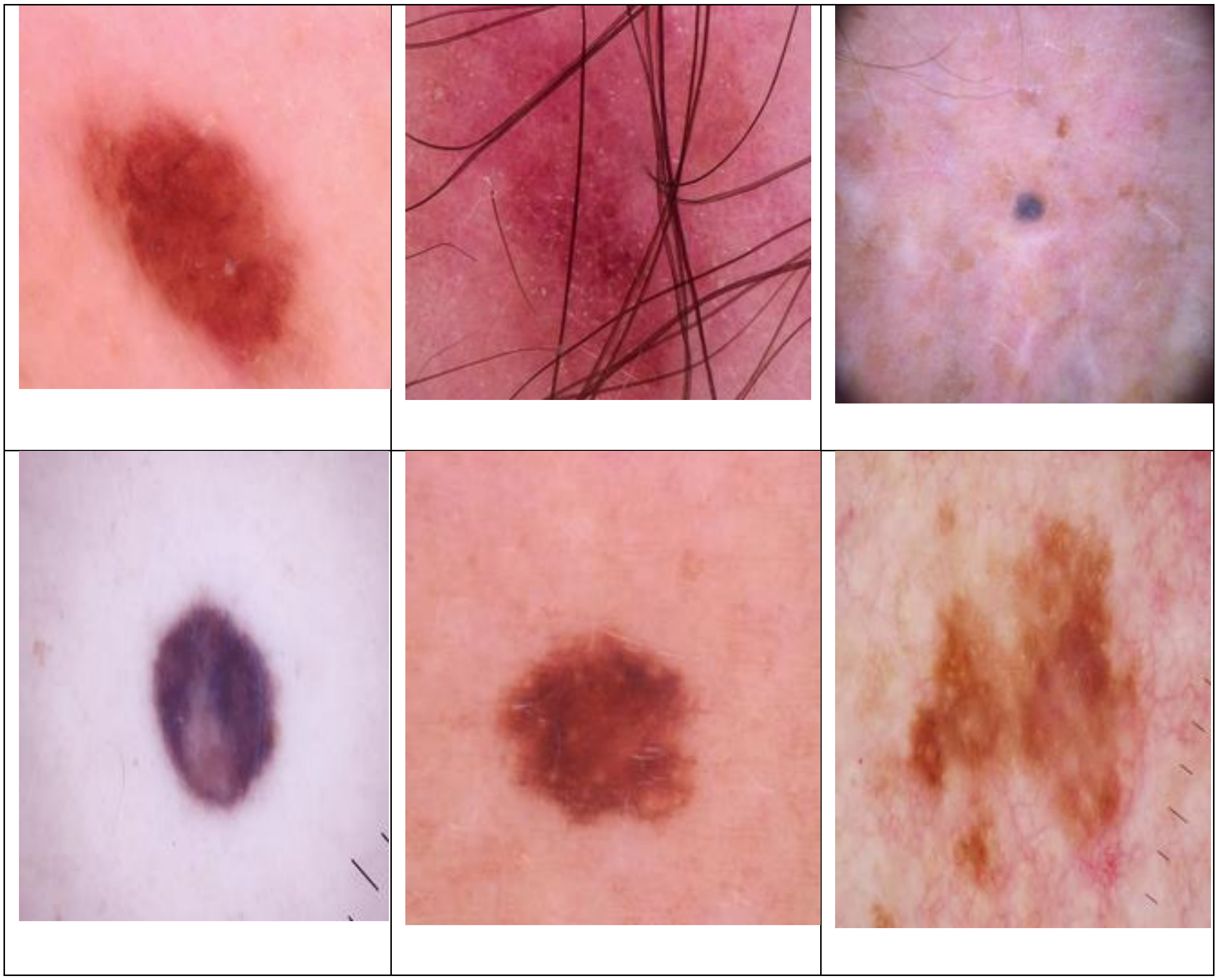


Figure (4.1) Benign Skin Cancer Images

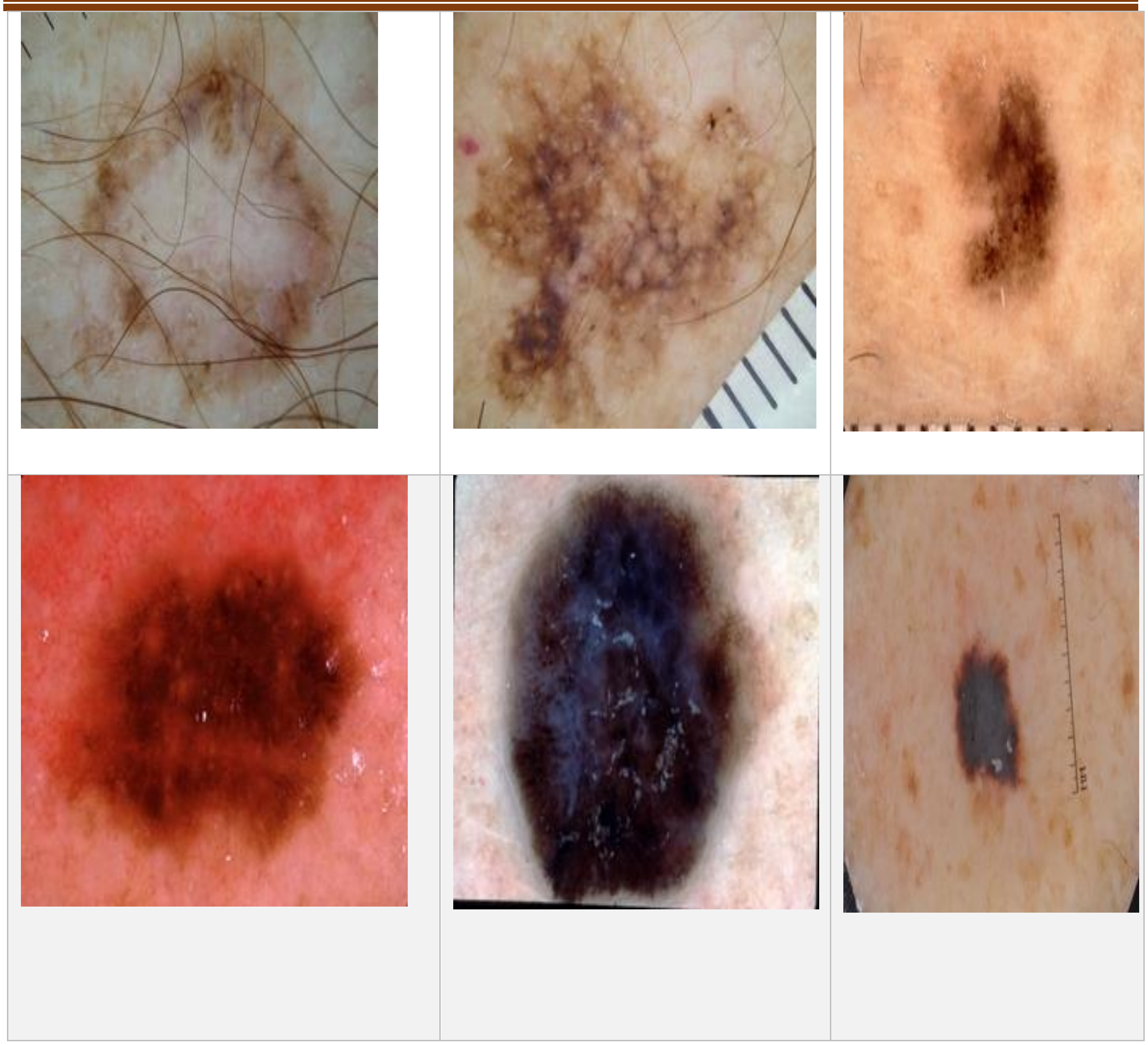


Figure (4.2) Malignant Skin Cancer Images

4.5 Skin Cancer Images Pre-processing Results

In this stage, two processes are performed on skin cancer images in order to make them ready for the next stages. These processes are performed using the algorithms that are explained in chapter three (section 3.4). The evaluation of these processes will be explained in the following sections:

4.5.1 Morphological Close Operation

The first step in the skin cancer classification system is applied closing operation to removing hair from image that is clear in algorithm (3.1). Figure (4.3) the original image before applied image closing operation.

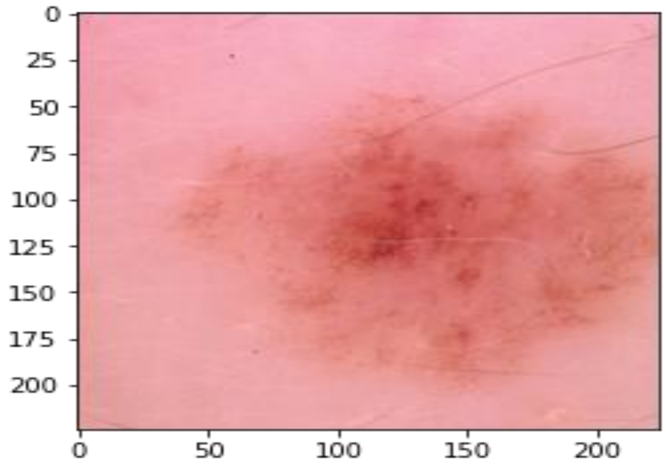


Figure (4.3): The Original Image before Preprocessing

4.5.2 Median Filter Result

Median filter applied after image closing operation to get on (hair removal algorithm) to detect the pixel containing natural hair similar to and each hair pixel from the resulted mask is replaced by an average mean of the neighbor's pixels like as algorithm (3.2) hair remove by image closing followed by median filter. Figure (4.4) show the image after applied the hair removal algorithm.

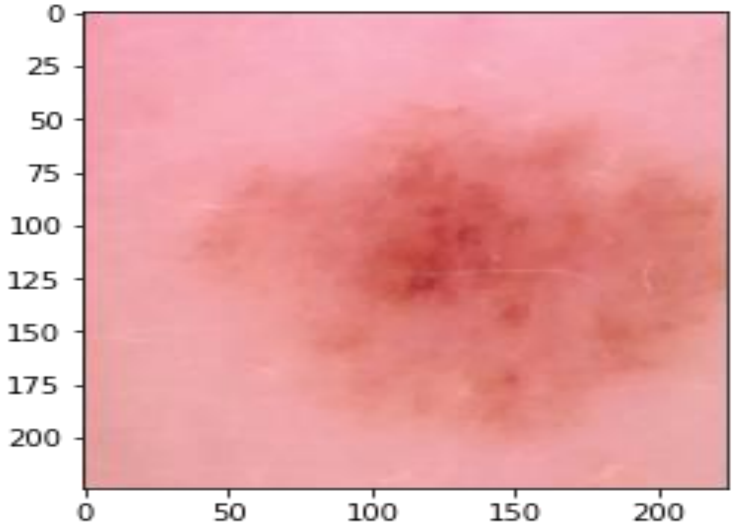


Figure (4.4): Image after Preprocessing

4.6 Skin Cancer Classification Models Results

4.6.1 First Model (CNN) Results

The results of the first model are explained, the proposed by CNN as shown in figure (3.3) and the procedure shown in algorithm (3.4). This section explores the performance results of CNN. Table (4.3) show the main CNN structure that has been used in the proposed system.

Table (4.3) Proposed Design of CNN Layers (Conv2D, Pooling, Full Connected)

Layer Number	Layer Type	Parameter				
		Dimension		Parameter No.	Padding Size	Kernel Size*stride
		Input size	Output size			
	Data(image)	224*224*3	-	-	-	-
Layer 1	Convolution1+ReLU	224*224*3	224*224*32	896	1	3*3*1
Layer 2	MaxPooling1+Dropout (0.25)	224*224*32	74*74*32	-	-	3*3*2
Layer 3	Convolution2 +ReLU	74*74*32	74*74*64	18496	1	3*3*1
Layer 4	Convolution3+ReLU	74*74*64	74*74*64	36928	1	3*3*1
Layer 5	MaxPooling2+Dropout (0.25)	74*74*64	37*37*64	-	-	2*2*2
Layer 6	Convolution4+ReLU	37*37*64	37*37*128	73856	1	3*3*1
Layer 7	Convolution5+ReLU	37*37*128	37*37*128	147584	1	3*3*1
Layer 8	MaxPooling3+Dropout (0.25)	37*37*128	18*18*128	-	-	2*2*2
Layer 9	FullyConnected+ReLU+Dropout (0.25) +Batch Normalization	18*18*128	41472	-	-	1*1*1024
Layer 10	Dense1+ReLU+ Dropout (0.25) +Batch Normalization	41472	1024	42468352	-	-
Layer 11	Dense2+Softmax	1024	2	2050	-	

- The first layer is convolution layer, consisting 32 filters each with three channels the filter window is 3*3 and the stride is 1, so every 3*3 square of image in an input to the filter. There is one unit of zero-padding, so the number of outputs of this layer equals the number of inputs, notice that we can computed the number of parameters in this layer according to equation (2.7).

  Conv1: number of input channels equal 3, the number of output channels equal 32.

  No. of parameters= 32*(3*(3*3) +1) =896.

  Conv2: number of input channels equal 32, the number of output channels equal 64.

No. of parameters=32*(32*(3*3) +1) =18496. The rest of Convolution layers applied the same formula.

- The output of first layer is fed to a max pool layer in which we divided the 224*224 array in to 3*3 squares with stride are 2. Thus, the 224*224 array has become a 74*74 array and there are still the same 32 filters. Notice that There is no parameter in this layer, as mentioned in chapter three. So, these processes repeated for layer (3 to 8).
- Finally, in layer 9 we flatten network takes inputs from the previous pooling layer 18*18*128= 41472, and that need 1024 node for first denes layer, while in second denes need 2 nodes for calculated classification by soft max function.

There were two section of training data, the first one training model without using preprocessing. Second section training model with using preprocessing.

➢ CNN Model without Preprocessing Results

The experiment was carried out by determining a different number of training epoch to get the best accuracy result. From figure (4.5) and table (4.4) can be seen that from epochs 0 to 10 shows that validation accuracy increased with the final result of 80.45% while validation loss has decreased with the final result of 0.39.and the total time is 54 minutes.

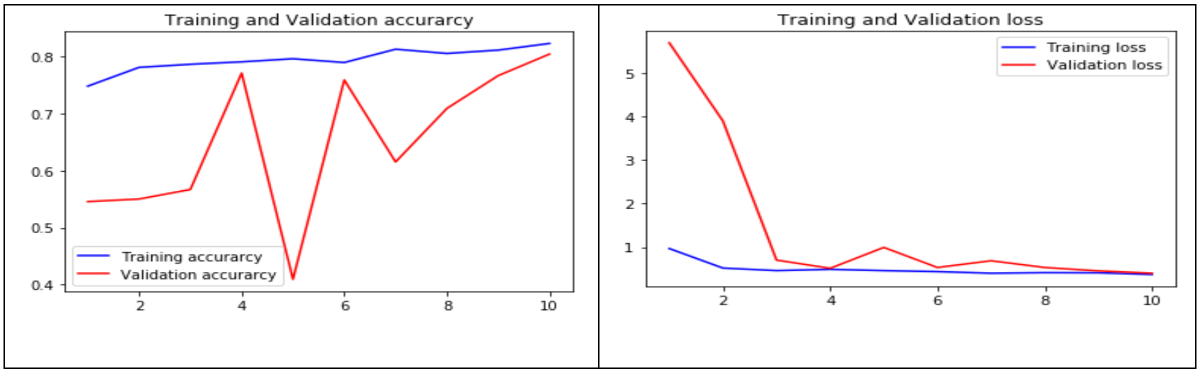


Figure (4.5): Accuracy and Loss Validation Change Against Training Epochs Using CNN Model(10-Epoch) Without Preprocessing.

Table (4.4) Accuracy and Loss for CNN without Pre-processing in 10-Epoch

Epoch	Time	Loss	Accuracy	Val _loss	Val -accuracy
1	293s 7s/step	0.9508	0.7482	5.6987	0.5455
2	294s 7s/step	0.5194	0.7812	3.9027	0.5500
3	291s 7s/step	0.4609	0.7866	0.7010	0.5667
4	290s 7s/step	0.4869	0.7909	0.5117	0.7712
5	292s 7s/step	0.4690	0.7963	0.9928	0.4091
6	291s 7s/step	0.4375	0.7897	0.5309	0.7591
7	291s 7s/step	0.3992	0.8131	0.6857	0.6152
8	290s 7s/step	0.4099	0.8057	0.5308	0.7091
9	290s 7s/step	0.4102	0.8115	0.4529	0.7667
10	295s 7s/step	0.3704	0.8232	0.3987	0.8045

Epochs 0 to epoch 30 shows figure (4.6) and table (4.5) that seen the validation accuracy has increased with the final result of 85.00% while validation loss has decreased with the final result of 0.3.and the total time is 2 hours and 20 minutes.

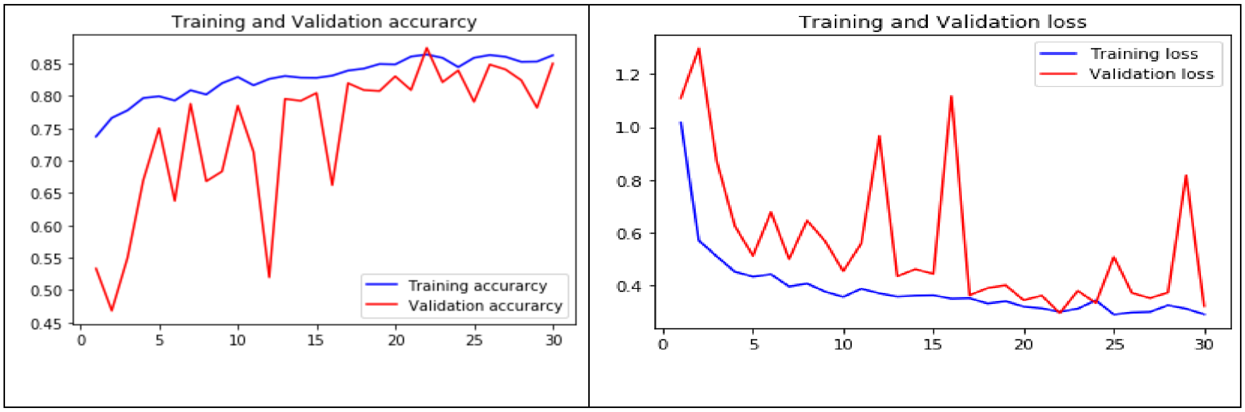


Figure (4.6): Accuracy and Loss Validation Change Against Training Epochs Using CNN Model(30-Epoch) Without Preprocessing.

Table (4.5) The accuracy and loss for each training in 30-Epoch

Epoch	Time	Loss	Accuracy	Val _loss	Val-Accuracy
1	318s 8s/step	1.0479	0.7373	1.1103	0.5333
2	319s 8s/step	0.5662	0.7660	1.2979	0.4682
3	318s 8s/step	0.5212	0.7777	0.8681	0.5500
4	316s 8s/step	0.4485	0.7967	0.6237	0.6697
5	318s 8s/step	0.4319	0.7995	0.5115	0.7500
6	318s 8s/step	0.4356	0.7928	0.6773	0.6379
7	316s 8s/step	0.4127	0.8088	0.4996	0.7879
8	318s 8s/step	0.4040	0.8022	0.6455	0.6682
9	317s 8s/step	0.3778	0.8197	0.5669	0.6833
10	323s 8s/step	0.3557	0.8293	0.4535	0.7848
11	313s 8s/step	0.3962	0.8164	0.5586	0.7136
12	320s 8s/step	0.3689	0.8263	0.9665	0.5197
13	324s 8s/step	0.3568	0.8308	0.4344	0.7955
14	317s 8s/step	0.3681	0.8282	0.4611	0.7924
15	315s 8s/step	0.3571	0.8279	0.4433	0.8045
16	318s 8s/step	0.3492	0.8313	1.1160	0.6621
17	323s 8s/step	0.3508	0.8392	0.3621	0.8197
18	321s 8s/step	0.3392	0.8422	0.3895	0.8091
19	315s 8s/step	0.3461	0.8493	0.4001	0.8076
20	323s 8s/step	0.3193	0.8487	0.3433	0.8303
21	315s 8s/step	0.3152	0.8608	0.3608	0.8091
22	323s 8s/step	0.2994	0.8639	0.2947	0.8742
23	312s 8s/step	0.3104	0.8588	0.3787	0.8212
24	319s 8s/step	0.3424	0.8445	0.3329	0.8394
25	317s 8s/step	0.2931	0.8589	0.5074	0.7909
26	324s 8s/step	0.2967	0.8632	0.3709	0.8485
27	318s 8s/step	0.3018	0.8605	0.3510	0.8409
28	317s 8s/step	0.3265	0.8527	0.3725	0.8242
29	316s 8s/step	0.3160	0.8531	0.8157	0.7818
30	317s 8s/step	0.2880	0.8628	0.3208	0.8500

Then, all the model resulted from the training were tested against 660 images of test data and calculated the percentage of accuracy, recall, specificity and precision from the results of the testing using confusion matrix as in figure (4.7).

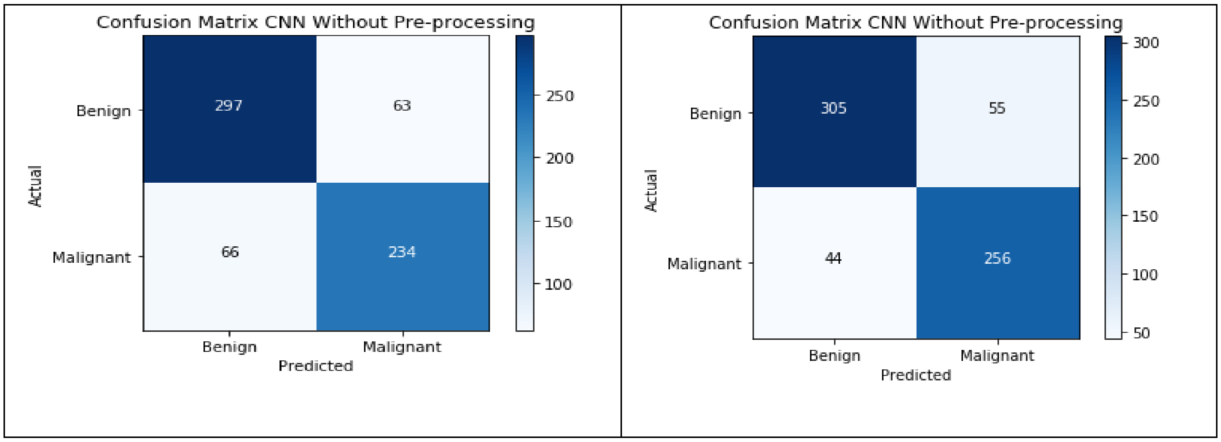


Figure (4.7): Confusion Matrix for the CNN Training without Preprocessing CNN Training. **Left**: In 10- Epoch; **Right**: In 30- Epoch.

$$Accuracy\ (\boldsymbol{CNN\ 10\ epoch}) = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100$$

$$\frac{297+234}{660} = 80.45\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{297}{297+66} = 81.81\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{234}{234+63} = 78.7\%$$

$$Precision = \frac{TP}{TP+FP} = \frac{297}{297+63} = 82.5\%$$

$$Accuracy\ (\boldsymbol{CNN\ 30\ epoch}) = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100$$

$$\frac{305+256}{660} = 85.00\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{305}{305+44} = 87.39\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{256}{256+55} = 82.31\%$$

$$Precision = \frac{TP}{TP+FP} = \frac{305}{305+55} = 84.72\%$$

Comparing figures (4.5) and (4.6) with the corresponding numbers in tables (4.4) and (4.5) show that when trained with 30 -epoch, the model gives better results and higher classification accuracy than trained in 10 -epoch. While it was noticed that the training time in 30-epoch takes more than the training time of 10-epoch.

- **CNN Model with Preprocessing Results**

CNN model is evaluated by using the same dataset, the values of the loss and accuracy of the second model (with preprocessing) during the 10 and 30 training epochs are illustrated in figure (4.8) and (4.9). The classification accuracy of the first proposed model with preprocessing and in 10-epoch, using the test images, is 69.99%, and the total time is 56 minutes.

Whereas the classification accuracy of the first proposed model with preprocessing and in 30-epoch, using the test images, is 83.03%, and the total time is 2 hours and 24 minutes.

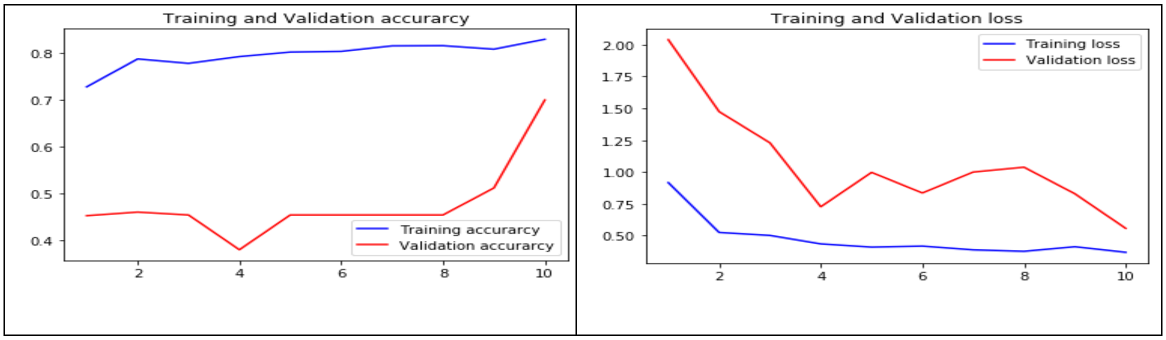


Figure (4.8): Accuracy and Loss Validation Change Against Training Epochs using CNN Model(10-Epoch) With Preprocessing**.**

Table (4.6) Accuracy and Loss for CNN with Preprocessing in 10-Epoch

Epoch	Time	Loss	Accuracy	val_loss	val_accuracy
1	346s 8s/step	0.9058	0.7279	2.0388	0.4530
2	322s 8s/step	0.5227	0.7874	1.4742	0.4606
3	334s 8s/step	0.5073	0.7781	1.2291	0.4545
4	332s 8s/step	0.4330	0.7925	0.7272	0.3803
5	344s 8s/step	0.4107	0.8022	0.9974	0.4545
6	319s 8s/step	0.4257	0.8037	0.8354	0.4545
7	341s 8s/step	0.3952	0.8154	1.0002	0.4545
8	318s 8s/step	0.3784	0.8158	1.0378	0.4545
9	335s 8s/step	0.4098	0.8084	0.8296	0.5121
10	330s 8s/step	0.3716	0.8294	0.5581	0.7000

The training process of this experiment was carried out using 30 epochs on 2637 train data which consist of 1440 images of benign and 1197images of malignant. The plot result from this training can be seen in figure (4.9).

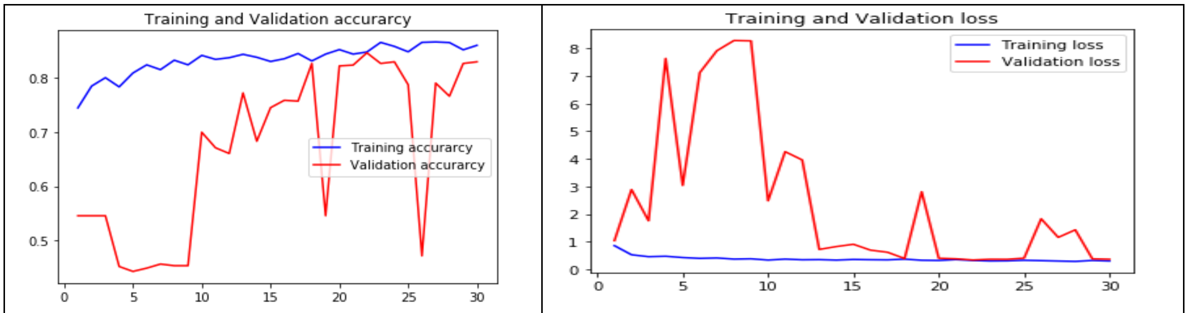


Figure (4.6) The Training and validation accuracy (30-epoch)

Figure (4.7) The Training and validation loss(30- epoch)

Figure (4.9): Accuracy and Loss Validation Change Against Training Epochs using CNN Model(30-Epoch) With Preprocessing.

Table (4.7) Accuracy and Loss for CNN with Preprocessing in 30-Epoch

Epoch	Time	Loss	Accuracy	Val _ loss	Val -accuracy
1	314s 8s/step	0.8577	0.7450	1.0363	0.5455
2	340s 8s/step	0.5260	0.7854	2.8886	0.5455
3	306s 7s/step	0.4520	0.8010	1.7552	0.5455
4	323s 8s/step	0.4714	0.7839	7.6366	0.4515
5	313s 8s/step	0.4227	0.8096	3.0361	0.4424
6	326s 8s/step	0.3916	0.8247	7.1229	0.4485
7	311s 8s/step	0.4065	0.8158	7.9149	0.4561
8	330s 8s/step	0.3690	0.8333	8.2854	0.4530
9	312s 8s/step	0.3755	0.8247	8.2729	0.4530
10	330s 8s/step	0.3337	0.8422	2.4790	0.7000
11	314s 8s/step	0.3927	0.8348	4.2600	0.6712
12	327s 8s/step	0.3489	0.8379	3.9604	0.6606
13	308s 8s/step	0.3464	0.8442	0.7187	0.7727
14	325s 8s/step	0.3365	0.8387	0.8228	0.6833
15	314s 8s/step	0.3520	0.8309	0.9062	0.7455
16	328s 8s/step	0.3537	0.8360	0.6934	0.7591
17	319s 8s/step	0.3413	0.8457	0.6164	0.7576
18	321s 8s/step	0.3773	0.8319	0.3890	0.8273
19	313s 8s/step	0.3291	0.8445	2.8096	0.5455
20	332s 8s/step	0.3189	0.8529	0.3993	0.8227
21	309s 8s/step	0.3441	0.8446	0.3795	0.8242
22	326s 8s/step	0.3191	0.8484	0.3370	0.8470
23	312s 8s/step	0.3013	0.8662	0.3624	0.8273
24	324s 8s/step	0.3028	0.8588	0.3607	0.8303
25	320s 8s/step	0.3323	0.8488	0.4031	0.7879
26	325s 8s/step	0.3203	0.8663	1.8288	0.4712
27	325s 8s/step	0.2996	0.8671	1.1552	0.7909
28	316s 8s/step	0.2843	0.8659	1.4289	0.7667
29	332s 8s/step	0.3191	0.8525	0.3777	0.8273
30	311s 8s/step	0.3014	0.8608	0.3639	0.8303

We classify with CNN and the confusion matrix in figure below lays out the performance of a learning algorithm by reporting the counts of the TP, FN, FP and TN of a classifier.

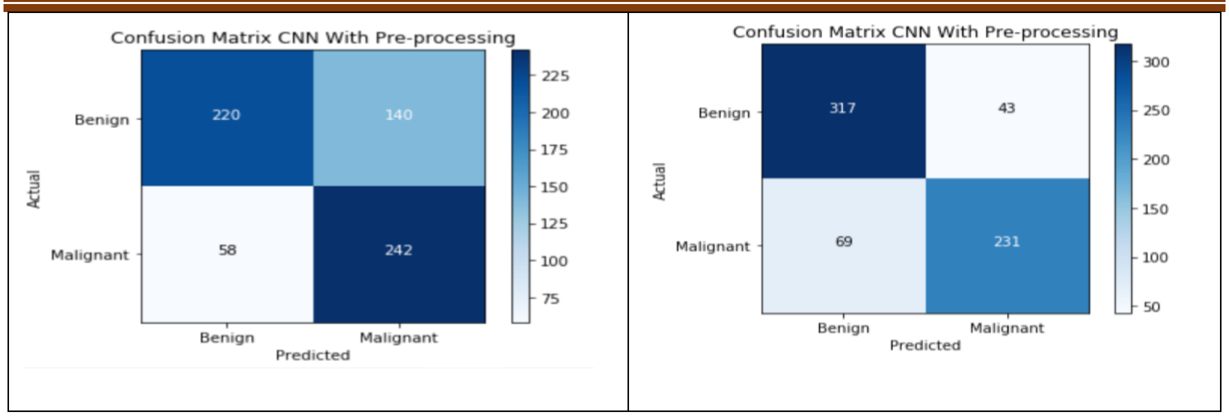


Figure (4.10): Confusion Matrix for CNN Training with Preprocessing CNN Training: **Left**: in 10- Epoch; **Right**: in 30- Epoch.

$$Accuracy\ (\boldsymbol{CNN\ 10\ epoch}) = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100$$

$$\frac{220+242}{660} = 69.9\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \quad \frac{220}{220+58} = 79.13\%$$

$$Specificity = \frac{TN}{TN+FP} = \quad \frac{242}{242+140} = 63.3\%$$

$$Precision = \frac{TP}{TP+FP} = \quad \frac{220}{220+140} = 61.11\%$$

$$Accuracy\ (\boldsymbol{CNN\ 30\ epoch}) = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100$$

$$\frac{317+231}{660} = 83.03\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \quad \frac{317}{317+69} = 82.12\%$$

$$Specificity = \frac{TN}{TN+FP} = \quad \frac{231}{231+43} = 84.30\%$$

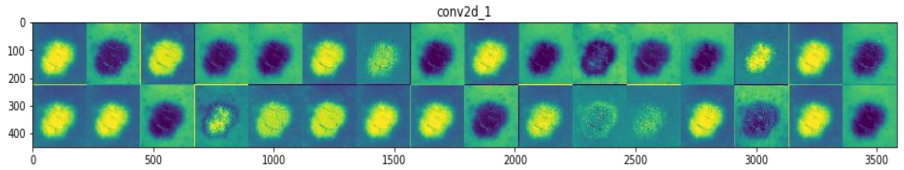
$$Precision = \frac{TP}{TP+FP} = \quad \frac{317}{317+43} = 88.05\%$$

The figures (4.5), (4.6), (4.8), and (4.9) were compared with the tables (4.4), (4.5), (4.6), and (4.7), notice that the first model with the preprocessing, the classification accuracy has decreased and fluctuation in loos function. As well as the training time, the model with the preprocessing takes more time than it is in model without preprocessing. Table (4.8) show the difference between first model CNN with and without preprocessing.

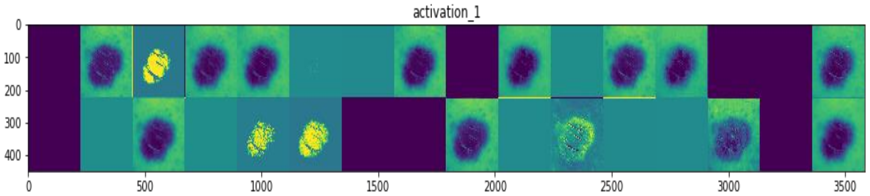
Table (4.8) Difference Between CNN Algorithm with and without Processing

Type	No. epoch	Test Accuracy	Total run Time
CNN without Pre processing	10 – Epoch	80.45%	54 minutes and 8 s on server
	30 -Epoch	85.00%	2 hours and 20m on server
CNN with Pre processing	10 -Epoch	69.99%	56 minutes and 8s on server
	30 -Epoch	83.03%	2 hours and 24 m on server

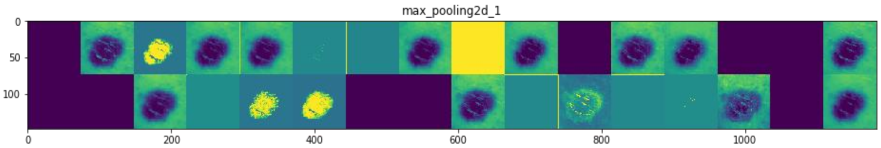
In (CNN with preprocessing) the main layers output show in figure (4.11) from the first layer (convolution layers) to the fifth with its activation, pooling and dropout parts.



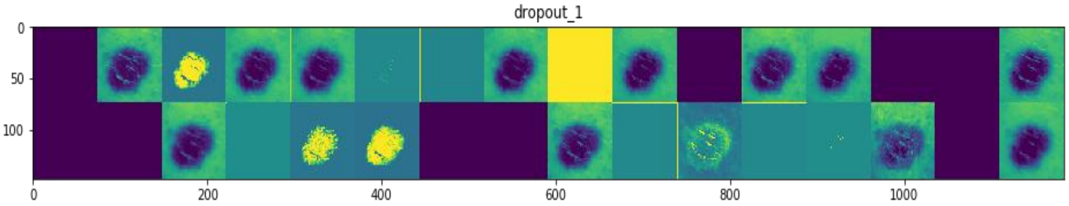
A-Output of First Conv2D Layer



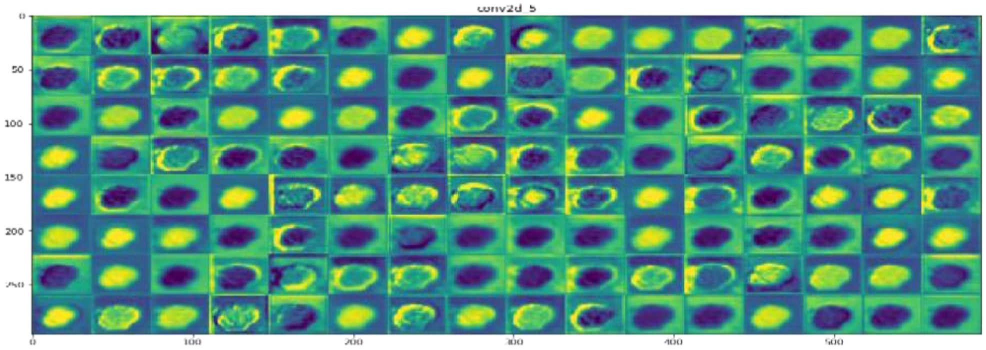
B- Output of Activation Layer (with 32-filters)



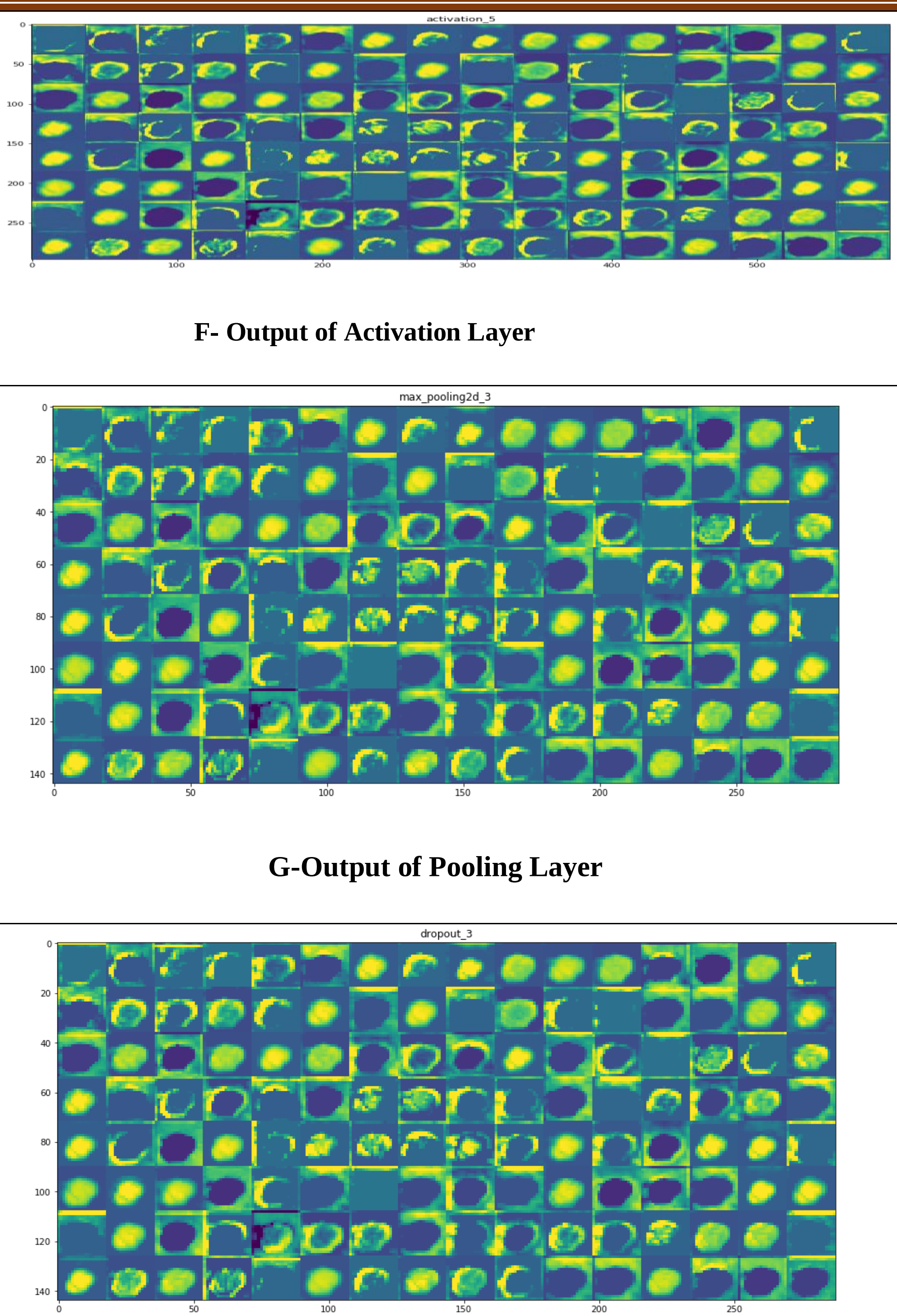
C- Output of Pooling Layer (with 32-filters)



D- Output of Dropout Layer (with 32-filters)



E- Output of last Conv 2D Layer (with 128-filters)



F- Output of Activation Layer

G-Output of Pooling Layer

H- Output of Dropout Layer

Figure (4.11): Result of Conv2D and Pool 2D Layers in CNN

When testing the model (CNN) in the classification of skin cancer images. Figure (4.12) show these images selected randomly form dataset.

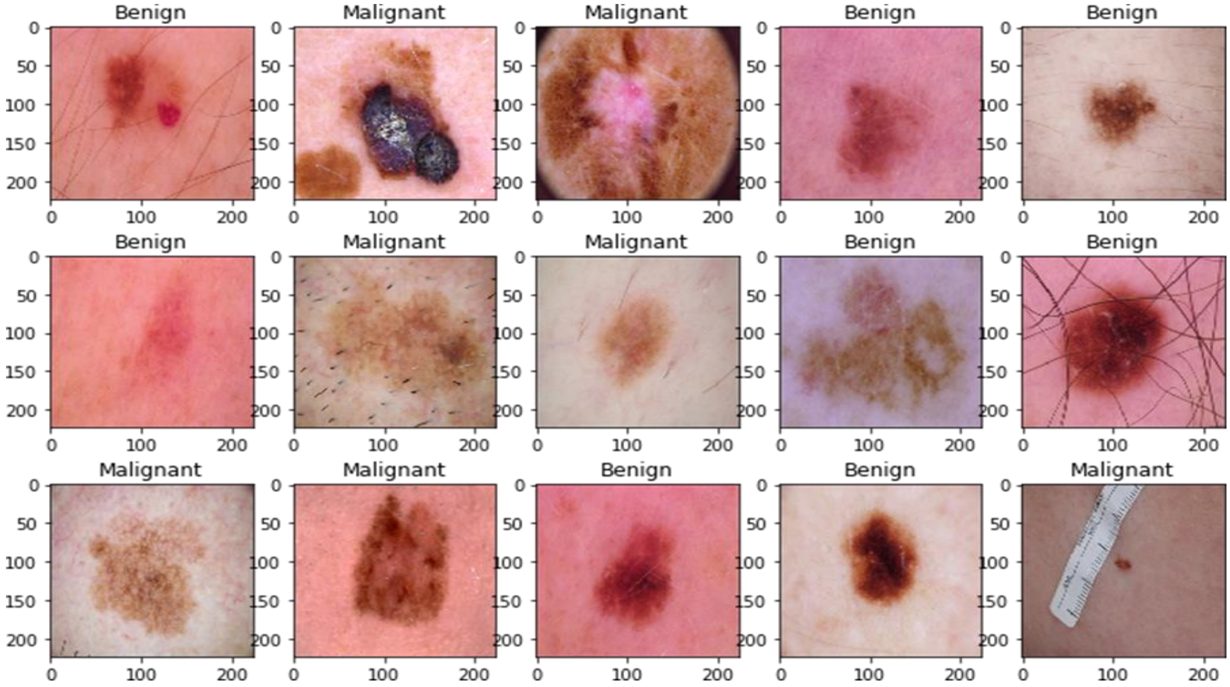


Figure (4.12) Skin Cancer Images with Labels

4.6.2 Results of Second Proposed Model (NB)

The second model consist of NB algorithm to classify the skin cancer images and it's also used in two cases, with and without preprocessing.

- The mean (μ) from equation (2.12), and standard deviation (σ) from equation (2.13), is formed during the training stage of local classifiers with of each two-class data for each feature vector component x_j.
- In testing phase and after obtaining the mean and standard deviation of each two class, we can apply the equation (2.14) to compute Probability density function $P\left(X_i \middle| C_j\right)$, and then select the maximum posterior probability of (X), $P\left(C_j \middle| X_i\right)$.

Figure (4.13) shows a two-dimensional model only of how the classes are distributed over the points, although they seem overlapping, but adding the rest of the dimensions, which is impossible, is what makes the classes separable.

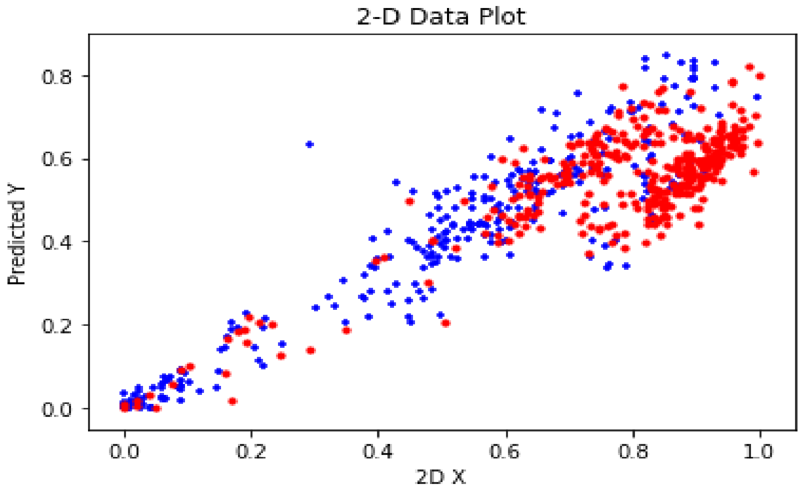


Figure (4.13) Dataset in Two-dimensional

Our proposed model includes two classes of benign and malignant skin cancer. In the figure above, points that have the same color indicate that they have the same class.

- **NB without Preprocessing Results:**

Figure (4.14) shows us a description of the performance of the classification model (or "Classifier") on a set of test data for which the real values are known. Where the confusion matrix itself is relatively easy to understand.

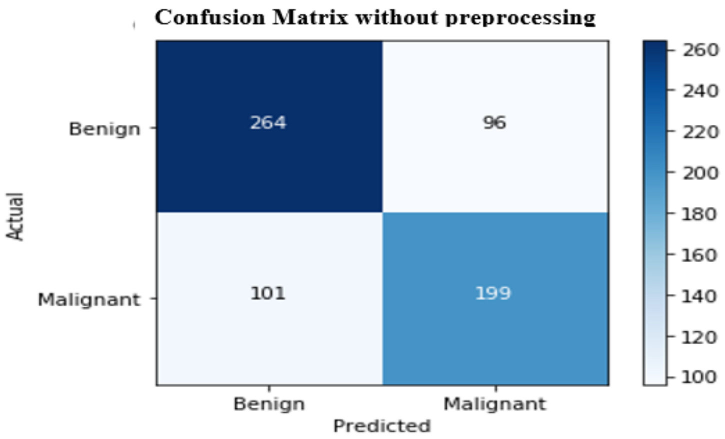


Figure (4.14) Confusion Matrix for Naïve Bayes without Preprocessing

For measuring Sensitivity (Recall), Precision, Specificity and Accuracy, we used equations (2.26) to (2.29)

$$Accuracy = \frac{TP+FN}{Total\ no.of\ test\ sample} \times 100 = \frac{264+101}{660} = 70.15\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{264}{264+101} = 66.33\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{199}{199+96} = 73.33\%$$

$$Precision = \frac{TP}{TP+FP} = \frac{264}{264+96} = 67.45\%$$

Table (4.9) Naïve Bayes Accuracy Without Preprocessing

Accuracy	70.15%
Sensitivity	66.33%
Specificity	73.33%
Precision	67.45%

Figure (4.14) and Table (4.9) illustrate the classification rate of our proposed system without preprocessing. In classification system for skin cancer images that presented by applying NB classifier is put forward. The algorithm NB does not require much training extreme of data compared to neural networks and their artificial structure. This makes it easier for the classifier to make a correct decision and with limited calculations. For minimizing the training time, our proposed classifier utilizes Naive Bayes classifier over the traditional classifiers while produce accurate in classification results by approve the advantage of NB algorithm and concepts of probability to the classifier.

- **NB with Preprocessing Results:**

The same model is evaluated by using the same dataset, the values of the accuracy, sensitivity, specificity, and Precision of the second model (with preprocessing) illustrated in figure (4.15) and table (4.16). The classification accuracy of the second proposed model with preprocessing and is 69.69%, and the total time is 2 minutes.

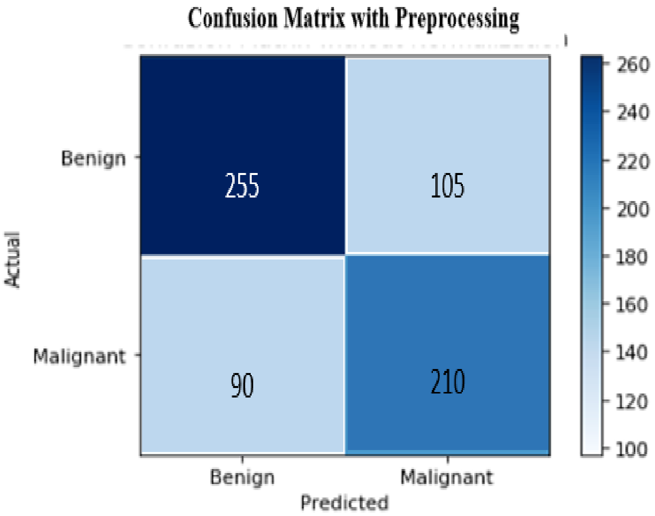


Figure (4.15) The Confusion Matrix for Naïve Bayes with Preprocessing

Table (4.10) Naïve Bayes Accuracy with Preprocessing

Accuracy	69.69 %
Sensitivity	70.00%
Specificity	70.08%
Precision	66.66%

From the tables above, it has been noticed a convergence in the rate of classification accuracy between the second model (with or without preprocessing) as the preprocessing did not affect the improvement of classification accuracy. As for the training time spent in the second model in the classification of skin cancer images, a naive Bayes algorithm that takes less time than CNN because this method does not need any too much training.

Table (4.11) Difference between (NB) Algorithm with and without Preprocessing

Type	Accuracy	Time
Naïve Bayes with Pre processing	69.69%	2 minutes and 6s
Naïve Bayes without Pre processing	70.15%	1 minutes and 4s

4.6.3 Results of Third Proposed Model (SVM)

Third model consist of SVM algorithm was used to classify the skin cancer images and it's also used in two cases, with and without preprocessing, skin cancer image samples are splatted into two subsets independent, i.e., part for training set and other part for the test set.

- During the training phase we computed the (w) values that represent the hyperplane between the classes, according on the equation (2.16).
- The largest margin is measured between the data entered in the testing phase, and on the basis of which the points belong to any class and through the formula are determined (2.18)

From the figure below, we can see the extent to which the data overlap being very large, as it becomes clear when represented by two-dimensional representation(2D), it becomes difficult to separate the two classes (benign, Malignant) linearly.

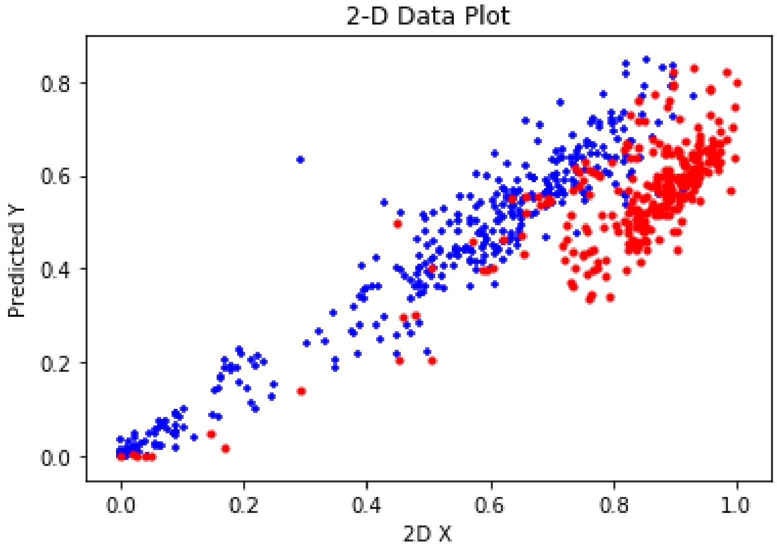


Figure (4.16) Dataset in Two-dimensional

-Proposed Model without Preprocessing

Figure (4.17) and table (4.18) indicate the accuracy of the classification of this model when classification of images without preprocessing.

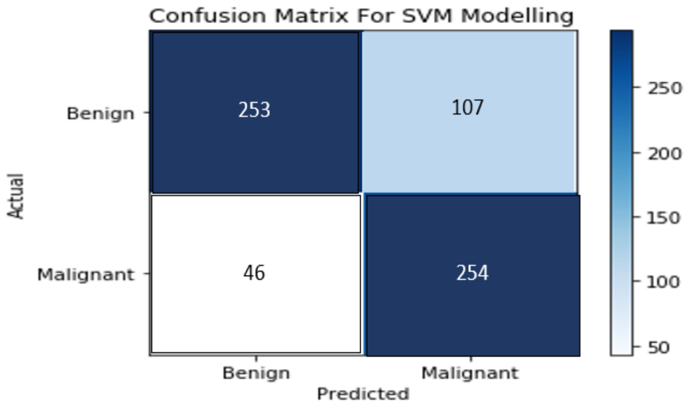


Figure (4.17) The Confusion Matrix for SVM without Preprocessing

By the indicators, accuracy, sensitivity, specificity that are shown in figure (4.17) and table (4.12) where the sensitivity (or recall) indicates the percentage of positives that were determined correctly (e.g., percentage of when samples that are correctly classify as having the skin cancer). by which the results are evaluated after obtaining them by our proposed method, where the results are as follows:

From equation (2.26) $Accuracy = \frac{TP+FN}{Total\ no.of\ test\ sample} \times 100$

$$\frac{253+107}{660} = 76.81\%,$$

Equation (2.27) $Sensitivity = \frac{TP}{TP+FN} = \frac{253}{253+46} = 84.66\%,$

Equation (2.28) $Specificity = \frac{TN}{TN+FP} = \frac{254}{254+64} = 70.27\%,$

and equation (2.29) $Precision = \frac{TP}{TP+FP} = \frac{253}{253+46} = 70.36\%.$

Table (4.12) SVM Accuracy without Preprocessing

Accuracy	76.81%
Sensitivity	84.66%
Specificity	70.27%
Precision	70.36%

-Proposed Model with Preprocessing

Same model is evaluated by using the same dataset, the values of the accuracy, sensitivity, specificity, and precision of the second model (with preprocessing) illustrated in figure (4.18) and table (4.13).

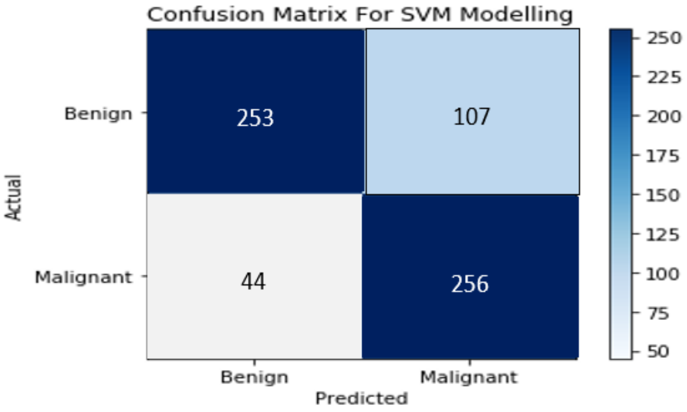


Figure (4.18) The Confusion Matrix for SVM with Preprocessing

Table (4.13) SVM Accuracy with Preprocessing

Accuracy	77.12 %
Sensitivity	85.3 %
Specificity	70.27 %
Precision	70.52 %

When talked about the time that algorithm took in training, it is much more than the time of training the naïve Bayes algorithm as shown in the table (4.14), and the reason for this time may be due to the nature of the work SVM algorithm , the kernel used RBF and the type of data used as the color skin cancer image contains three channels where we have a matrix with very high dimensions, which requires complex calculations that require a long training time.

Although the classification accuracy in the pre-treatment stage improved very slightly and the cause "*our images do not bear any signs other than the affected area, and they were taken closely, so any processing on these images loses information that might affect the classification process*". The difference between the two model is shown in table (4.14).

Table (4.14) difference between SVM with and without Preprocessing

Type	Accuracy	Time
SVM with Pre processing	77.1 %	37 minutes and 1s
SVM without Pre processing	76.8 %	22 minutes and 8s

Figure (4.19) shows the number of support vectors, where the red and blue dots represent the classes, while the yellow lines represent the support vectors, which are the lines separating the classes. Because the data are complex and has many dimensions, they do not appear clearly in drawing, whether it is two-dimensional or even three-dimensional.

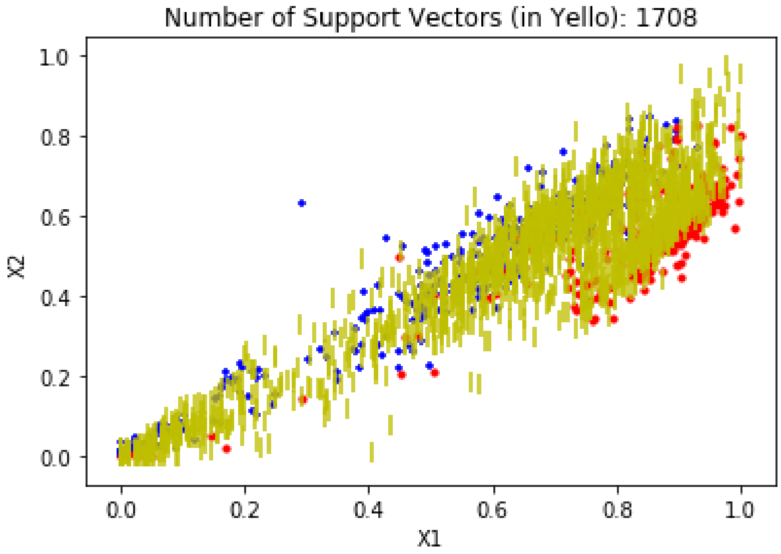


Figure (4.19) Number of Support Vectors

4.7 Result Analysis for the Proposed Models

The average performance of the methods evaluated during the study are summarized in table (4.15). The average accuracy measures, illustrated visually in figure (4.20) shows that the CNN method has a better performance, compared with NB and SVM, in both cases, with and without preprocessing. However, we also noticed the difference in calculating the time taken to train the data, as the NB algorithm exceled with the least time, then the SVM algorithm, and the CNN algorithm came with the longest time it took to train the data. Table (4.15) and (4.16): show the average performance measure of the proposed models (for two cases with and without) with training time for each model.

Table (4.15) Average Performance Measure of Proposed Models, CNN, Naïve Bayes and SVM Without Preprocessing

Type	Accuracy	Time
CNN byes without Pre processing	85 %	2 hours and 20 m
Naïve Bayes without Pre processing	70 %	1 m and 4 s
SVM without Pre processing	76%	22 m and 8 s

While table below shows the performance of the algorithms CNN, NB and SVM with the time taken to train each algorithm when applying the preprocessing on the images, the results are shown in table (4.16).

Table (4.16) Average Performance Measure of Proposed Models, CNN, NB and SVM with Preprocessing

Type	Accuracy	Time
CNN byes with Pre processing	83 %	2 hours and 24 m
Naïve Bayes with Pre processing	69 %	2 minutes and 6s
SVM without Pre processing	77 %	37 minutes and 1s

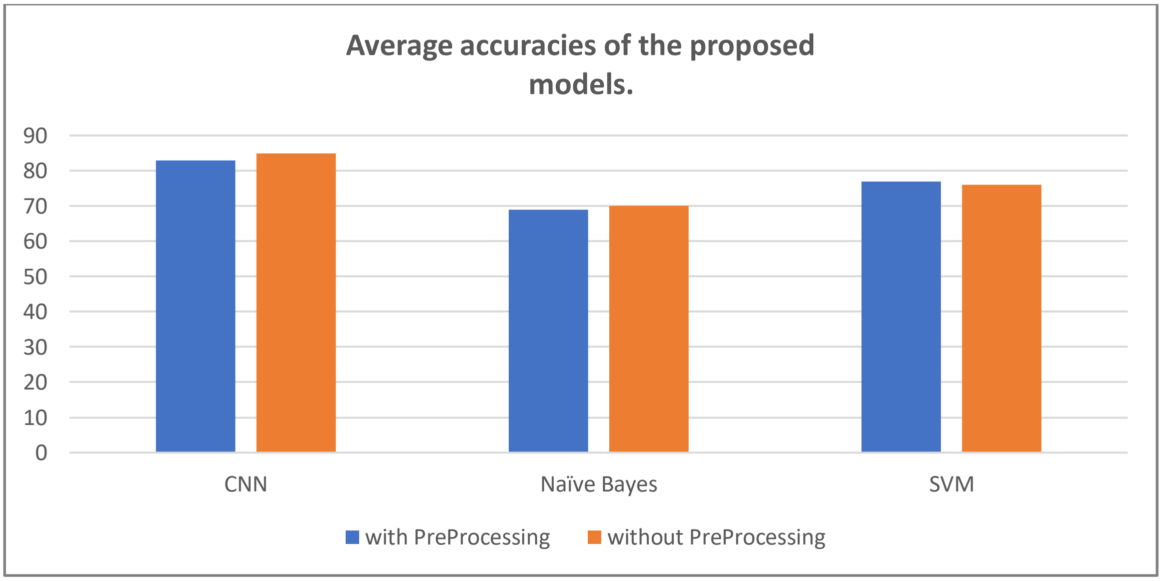


Figure (4.20): Illustration of Average Accuracies of Proposed Models.

CNN showed better performance and implementation from the SVM method and NB in terms of classification accuracy, but it is the longest in terms of training time, while the algorithm NB was able to take the least time with low accuracy , and finally the SVM algorithm where its performance was moderate in both aspects of accuracy and time.

4.8 Comparison to the Related Works

The comparison was made for the proposed system of the skin cancer. Table (4.17) shows a comparison with the related work **[15][16].**

Table (4.17) Comparison of Classification Accuracy with Earlier Studies

Approach	Accuracy	Methods
Md Ashraful [15] 2018	76.00 %	(CNN) PNASNet-5-Large model
Adria & Jack [16] 2017	78.66%,	VGG Net convolutional neural network architecture
proposed Models	85.00 % with (CNN) 76.8% with (SVM) 70.00 % with (NB)	Convolutional Neural Network &SVM & NB

Table (4.17) explains the results obtained from the performance of some of the proposed models in the same field of our proposed algorithms, the comparison shows that our proposed system CNN and its architecture, which differs from that in Md Ashraful & Adria & Jack, have achieved a very good classification accuracy, as it exceeds the previous two studies, where the development of two layers of the convolutional network followed by the Max-pooling layer has help a lot in extracting good features despite the time spent on training, which remains less than the algorithm PNASNet-5-Large model Ashraful that trained on Dataset was large, but accuracy remained less than the classification accuracy obtained by our proposed model (CNN). Also, Adria & Jack used VGG architecture as it was trained from scratch, but because the dataset few did not achieve a good rating accuracy compared to our model.

Chapter Five

Conclusions and Suggestions for Future Work

5.1 Conclusion

This chapter summarizes the evaluation of the study results and present the main benefit of models is proposed. it is the following conclusions are given:

1- This study demonstrated that the preprocessing phase of data does not necessarily improve the accuracy of the model used because, all images in our dataset do not own any signs other than the affected area, and they were taken closely, so any processing on these images loses information that might affect the classification process.

2- First methodology CNN without preprocessing that extracting features locally by (convolution and pooling layers) to classify skin cancer images achieved best performance compared with Naïve Bayes and SVM system.

3- The second model NB with preprocessing and without pre-processing noted that its classification accuracy was Close but less time for training, compared with CNN and SVM.

4- Although the classification accuracy in SVM with pre-processing improved it was very slight, unnoticeable, while the training time was less than the time of CNN and more than Naïve Bayes.

5- When increased number of training epoch as shown in figures (4.8) and (4.9), for CNN improves accuracy and reduces loss.

6- A comparison between tables (4.15) and (4.16) show that the convolutional neural network has been able to outperform the NB and SVM with every evaluation dataset. Despite the time spent training the data, this is due to the architecture of CNN used and the number of layers (convolution and pooling) used to extract the features.as shown in figure (3.4).

7- Comparing the first proposed system with other related works table (4.17) proves the proposed models is more effective over other related works that our model (**CNN**) accuracy 85% for without, and 83%for with processing), (**NB)** accuracy (70 %for without ,and 69%for with processing), and (**SVM**)model 76%for without and 77% for with processing).

5.2 Suggestions for Future Work

Some of the suggestions, which can be used in future work, are as follows:

1- Test other kinds of skin cancer dataset that contain a large number of skin cancer images.

2- Use other methods for pre-processing data and applied them on images then test their effect on classification accuracy.

3- Try to combine the CNN with another supervised learning approach for classification purpose instead the fully connected layer such Decision Tree (DT), K Nearest Neighbors (KNN) etc., and compare the obtained result with those classification algorithms.

4- Improving the performance of the (CNN) model, by increasing the number of filters as well as the size of the filter used like (5×5), (7×7) to improve the classification accuracy.

5- As a further development to the model we are targeting to expand the multi-platform capability through mobile support. Experimenting with datasets of collection from hospitals to enable doctors from given feedback about classification results.

References

[1] World Health Organization. 2018. Skin Cancers. Available https://www.who.int/news-room/q-a-detail/ultraviolet-(uv)-radiation-and-skin-cancer

[2] Cancer Facts and Figures 2020.(ACS) https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/

[3] Jerant, A. F., Johnson, J. T., Sheridan, C. D., & Caffrey, T. J., "**Early Detection and treatment of skin cancer**" American family physician, vol. 62, no. 2, pp. 357-386, 2000.

[4] Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E., & Delfino, M. "**Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comp. of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis**" Archives of Dermatology, vol. 134, no. 12, pp. 1563-1570, 1998.

[5] Korotkov, K., & Garcia, R. "**Computerized analysis of pigmented skin lesions: A review**" Artificial intelligence in medicine, vol. 56, no. 2, pp. 69- 90, 2012.

[6] Zhu H., "**Medical Image Processing Overview**", University of Calgary, pp.1- 27, 2003.

[7] Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. "**Machine learning: An artificial intelligence approach**". Berlin: Springer (2013).

[8] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. "**Deep learning**" book (Vol. 1). Cambridge: MIT Press. (2016).

[9] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. "**Deep learning applications and challenges in big data analytics,**" Journal of Big Data, vol. *2*, no. 2015.

[10] Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., & Nielsen, M.” **Deep feature learning for knee cartilage using a triplanar convolutional neural network**”. In International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp. 246-253. 2013.

[11] Refianti R., Benny B. M., Poetri R. Priyandini "**Classification of Melanoma Skin Cancer using Convolutional Neural Network**"(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, no. 3, 2019.

[12] Albahar, M. A. “**Skin lesion classification using convolutional neural network with novel regularizer**”. IEEE Access, vol.*7*, pp. 38306-38313. 2019.

[13] Sanket, K., Chandra, J.," **Skin Cancer Classification using Machine Learning for dermoscopy Images**" International Journal of Innovative Technology and Exploring Engineering (IJITEE) Vol.7, pp.2278-3075. 2019.

[14] Mohan, K., Ram, K., Gopalakrishnan, K.,” **Skin Cancer Diagnostic using Machine Learning Techniques – Shear let Transform and Naïve Bayes Classifier**” International Journal of Engineering and Advanced Technology (IJEAT) Vol.9 no.2, pp :2249 –8958. 2019.

[15] Md Ashraful Alam Milton” **Automated Skin Lesion Classification Using Ensemble of Deep Neural Networks in ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge**” arXiv preprint arXiv:1901.10802, 2019.

[16] Lopez, A. R., Giro-i-Nieto, X., Burdick, J., & Marques, O. " Skin lesion classification from dermoscopic images using deep learning techniques " In: 13th IASTED international conference on biomedical engineering (BioMed). IEEE, pp. 49-54.2017.

[17] Codella, N. C., Nguyen, Q. B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C., & Smith, J. R. “**Deep learning ensembles for melanoma recognition in dermoscopy images**. IBM Journal of Research and Development, 61(4/5), pp.5-1. 2017.

[18] Mustafa, S., Dauda, A. B., & Dauda, M.” **image Processing and SVM Classification for Melanoma Detection**” In: international conference on computing networking and informatics (ICCNI), IEEE. (pp. 1-5). 2017

[19] Nasr-Esfahani, E., Samavi, S., Karimi, N., Soroushmehr, S. M. R., Jafari, M. H., Ward, K., & Najarian, K. **Melanoma detection by analysis of clinical images using convolutional neural network**. In 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 1373-1376) 2016.

[20] Shoieb, D. A., Youssef, S. M., & Aly, W. M. **Computer-aided model for skin diagnosis using deep learning**. Journal of Image and Graphics, vol.4, no. 2, pp. 122-129. 2016.

[21] Park, D. C.” **Image Classification Using Naïve Bayes Classifier**” International Journal of Computer Science and Electronics Engineering (IJCSEE) Volume 4, no.3 pp 2320–4028 .2016.

[22] Valiant, L., G. “**A theory of the learnable**”. Communications of ACM, 27(11): pp. 1134–142, November 1984

[23] Robert, C., book, “**Machine learning, a probabilistic perspective**”. Pp 62-63 ,2014.

[24] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. **Deep learning applications and challenges in big data analytics**. Journal of Big Data, vol. 2, no.1:1. 2015.

[25] Li, N., Jia, L., & Zhang, P., “**Detection and volume estimation of bubbles In blood circuit of hemodialysis by morphological image processing**,” in IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), vol. 58, no 12, pp. 228–231.2015.

[26] Said, K. A. M., Jambek, A. B., & Sulaiman, N. “**A study of image processing using morphological opening and closing processes**”. International Journal of Control Theory and Applications, vol. 9, no 31, pp.15-21. 2016.

[27] Sarah A., and Anoop T., "**Brightness preserving and contrast Enhancement of various image Enhancement Techniques**: A review", International Journal of Electrical, Electronics ISSN No: 2277-2626, and Computer Engineering 6(1): 158-163, 2017.

[28] Maini R. and Aggarwal H., "**A Comprehensive Review of Image Enhancement Techniques**", Journal of Computing, Vol. 2, Issue No. 3, pp. 8-13, March, 2010.

[29] Janani R., "**Image Processing Using MATLAB GUI**", International Journal of Electronics, Electrical and Computational System, Vol. 6, Issue 1, 2017.

[30] Gurney K., "book", "**An introduction to neural networks**", CRC press, 1997.

[31] Šíma, J., & Orponen, P. "**General-purpose computation with neural networks: A survey of complexity theoretic results**". Neural Computation, vol.15, no.12, pp:2727–2778, 2003.

[32] Niriksha; Jain, N.; and Gankotiya, A. K.; "**A Detailed Study on Artificial Neural Networks**"; Discovery, Vol. 15, no. 41, Pp. 63-67, April 2014.

[33] Xu, B., Wang, N., Chen, T., & Li, M. "**Empirical evaluation of Rectified activations in convolutional network**," arXiv preprint arXiv:1505.00853, 2015.

[34] Ren, S., He, K., Girshick, R., & Sun, J. "**Faster R-CNN: towards real-Time object detection with region proposal networks**", In: Advances in neural information processing systems. p. 91-99. 2015.

[35] Kantardzic, M., "**Data Mining: Concepts, Models, Methods and Algorithms**", John Wiley & Sons -IEEE Press, 2011.

[36] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. "**Dropout: a simple way to prevent neural networks from overfitting**". The journal of machine learning research, vol.15 no.1, p:1929-1958.2014.

[37] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R. R. "**Improving neural networks by preventing co-adaptation of feature detectors**". arXiv preprint arXiv:1207.0580. 2012.

[38] Kingma, D. P., & Ba, J. “**Adam: A method for stochastic optimization**”. arXiv preprint arXiv:1412.6980.2014.

[39] Glorot, X., & Bengio, Y., "**Understanding the difficulty of training deep feedforward neural networks**" in Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256, 2010.

[40] Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S., & Shen, D. “**Deep convolutional neural networks for multi-modality isointense infant brain image segmentation**”. Neuroimage, vol.108. p:214–224, 2015.

[41] Tajbakhsh, N., Gotway, M. B., & Liang, J. “**Computer-aided pulmonary embolism detection using a novel vessel-aligned multi-planar image representation and convolutional neural networks**”. In International Conference on Medical Image Computing and Computer- Assisted Intervention (pp. 62-69). Springer, Cham.2015.

[42] Hatipoglu, N., & Bilgin, G. “**Classification of histopathological images using convolutional neural network**”. In International Conference on Image Processing Theory, Tools and Applications (IPTA), pages 1–6. IEEE, 2014.

[43] Ji, S., Xu, W., Yang, M., & Yu, K. "**3D convolutional Neural networks for human action recognition**." IEEE transactions on pattern analysis and machine intelligence vol. 35, no. 1, pp: 221-231. 2013.

[44] Lambin P, Rios-Velazquez E, Leijenaar R. “**Radiomics extracting more information from medical images using advanced feature analysis**”. European journal of cancer, 48(4), pp: 441-446. 2012

[45] Iftene, M., Liu, Q., & Wang, Y. “**Very high-resolution images Classification by fine tuning deep convolutional neural networks**”. In Eighth International Conference on Digital Image Processing, pages 100332D–100332D. International Society for Optics and Photonics, 2016.

[46] Lin M, Chen Q, Yan S. “**Network in network**”. arXiv preprint arXiv:1312.4400 2013.

[47] Duda, R., Hart, P., Storck, D. book” **Pattern classification**” (2nd Edition), Wiley Inter science, New York, 2000.

[48] Gorunescu, Florin. book “**Data Mining: Concepts, models and techniques**”. Vol. 12. Springer Science & Business Media, 2011.

[49] Jabbar, M. A., & Samreen, S. “**Heart disease prediction system based on hidden naïve bayes classifier**”, International Conference on Circuits, Controls, Communications and Computing (I4C), IEEE, pp. 1-5. 2016.

[50] Lowd, D., & Domingos, P. “**Naive Bayes models for probability estimation**,” In Proceedings. of the 22th International Conference on Machine Learning, pp. 529-536. 2005.

[51] Cortes, C., & Vapnik, V., “**Support-vector networks**,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[52] Masood, A., & Al-Jumaily, A. “**Differential Evolution based Advised SVM for Histopathological Image Analysis for Skin Cancer Detection**” 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2015.

[53] Farooq, M. A., Azhar, M. A. M., & Raza, R. H. “**Automatic Lesion Detection System (ALDS)for Skin Cancer Classification Using SVM and Neural Classifiers**”, IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE). IEEE, 2016.

[54] Fradkin, D., & Muchnik, I., **Support Vector Machines for Classification in A bello, J., and Carmode, G.** Discrete Methods in Epidemiology, pp. 13–20, 2006.

[55] ISIC Database (International Skin Image Collaboration) Archive.https://www.isic-archive.com/topWithHeader/wideContentTop/main

الخلاصة

تحدث معظم حالات الوفاة بسبب سرطان الجلد وخاصةً النوع الخبيث منه. لذلك ، يعتبر أحد الأنواع الأخيرة من السرطانات, يمكن علاجه باكتشاف المرض مبكرًا من خلال فحص الخزعة ، لذا فإن أفضل حل لتحسين تشخيص سرطان الجلد هو الكشف المبكر. يعد التشخيص بمساعدة الكمبيوتر (CAD) أحد تقنيات التصوير المستخدمة على نطاق واسع للكشف عن سرطان الجلد وتصنيفه. يعتبر الكشف عن الصورة وتصنيفها تلقائيًا مهمًا جدًا لبشرة والأورام ومهمة صعبة للغاية للصور الطبية. في السابق ، تم اتخاذ هذا القرار يدويًا من قبل البشر بمساعدة أطباء الجلد عن طريق أخذ وتحليل خزعة المريض من جلد المريض ومع ذلك ، تتطلب هذه الإجراءات المزيد من الوقت وقد لا تكون النتيجة دقيقة للغاية. مما يؤدي إلى عدم الدقة. تقدم هذه الرسالة نظامًا مقترحًا لتصنيف سرطان الجلد بعد اكتشافه بمساعدة آليات التعلم العميق وخوارزميات التعلم الآلي. في هذا العمل ، تم تنفيذ نظام تصنيف قوي لسرطان الجلد باستخدام ثلاثة نماذج مقترحة. يعتمد النموذج الأول المقترح على مصنّف الشبكات العصبية التلافيفية (CNN). يستخدم النموذج المقترح الثاني على مصنف Naïve Bayes (NB). في حين أن النموذج الثالث المقترح يعتمد على مصنف دعم آلة المتجه (SVM). وكل نموذج مع معالجة مسبقة وبدون معالجة مسبقة للبيانات. أظهرت نتائج هذه الأطروحة أن النموذج المقترح الأول باستخدام (CNN) بدون معالجة مسبقة كان متوسط الدقة ٨٥,٠٠٪. بينما مع المعالجة المسبقة كانت الدقة ٦٩,٩٩٪. والنموذج الثاني المقترح باستخدام (NB) بدون معالجة مسبقة كان متوسط الدقة ٧٠,١٥٪ بينما كان مع المعالجة المسبقة دقة ٦٩,٦٩٪. النموذج الثالث المقترح باستخدام (SVM) بدون معالجة مسبقة حقق دقة ٧٦,٨١٪ بينما مع المعالجة المسبقة حصل على دقة ٧٧,١٢٪.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية العلوم
قسم علوم الحاسبات

طرق التعلم العميق للكشف عن المرض من خلال الصور

رسالة مقدمة
الى كلية العلوم في جامعة ديالى وهي جزء من متطلبات نيل
شهادة الماجستير في علوم الحاسبات
تقدمت بها الطالبة

عهود فاضل علوان

بإشراف

أ.د. ظاهر عبدالهادي عبدالله

٢٠٢٠ م