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

Deep Learning Method for Classification of Skin Cancer Disease

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 Abdulhadi Abdullah

2020

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ 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 my supervisors ***Prof. Dr. Dahir Abdul Hadi Abdullah*** *for providing his 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 Noman, Adel Abed****,*

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

Thank you all!

Ohood

Abstract

Skin cancer is an abnormality in skin cells caused by mutations in cells Deoxyribonucleic Acid (DNA). 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. 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, where several steps are used in the form of stages, which are include, the image acquisition stage, image pre-processing, and the classification stage. The used dataset is obtained from the ISIC (International Skin Image Collaboration) Archive, it contains 3297 images. There are 1497 image cases of malignant skin caser type, and 1800 images cases for benign. In preprocessing stage, hair removal algorithm is using. The First proposed model depends on Convolutional Neural Networks (CNN) classifier. The second proposed model uses 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

List of Abbreviations

Abbreviations	***Description***
ABCD	Asymmetry, Border, Color, Diameter
AC	Accuracy
AI	Artificial Intelligence
ANN	Artificial Neural Network
CAD	Computer-aided diagnosis
CNN	Convolutional Neural Network
Conv2D	Convolutional Tow – Dimension
DNA	Deoxyribonucleic Acid
ES	Error Signals
FN	False Negative
FP	False Positive
FS	Function Signal
IM	Input Image
ISIC	International Collaboration Skin Imaging
ISBI	International Symposium on Biomedical Imaging
ML	Machine Learning
MAP	Maximum A Posteriori
MLP	Multi-Layer Perception
NB	Naïve Bayes
NBC	Naïve Bayes Classification
NNs	Neural Networks
RBF	Radial Basis Function
ReLU	Rectifier Linear Unit
SVM	Support Vector Machine
SE	Structuring Element

TP	True Positive
TN	True Negative
VGG	Visual Geometry Group

List of Tables

List of Figures

Chapter One

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].

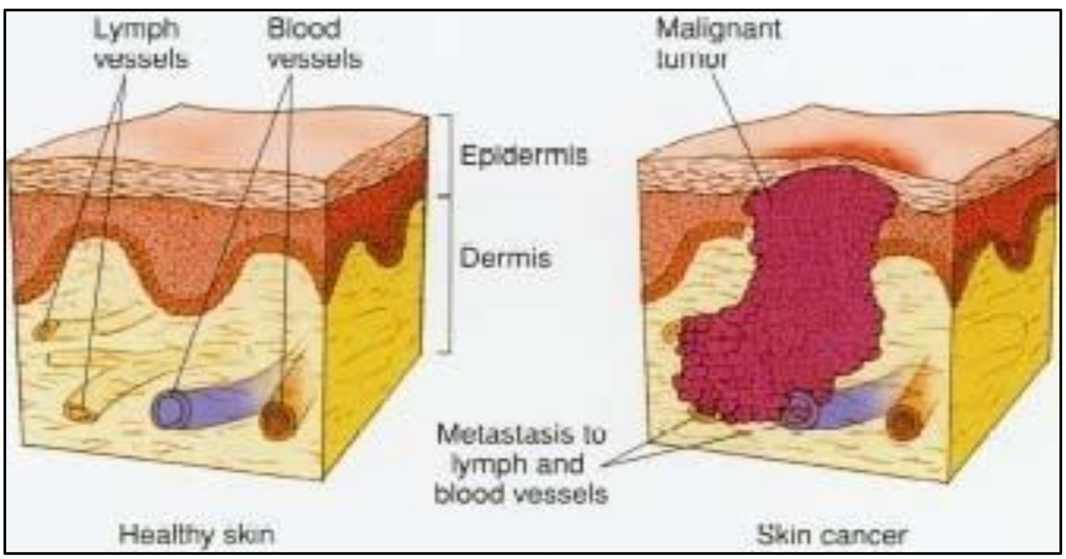


Figure (1.1): (left) Unaffected skin, (right) Affected skin [1].

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 cases will be men and 40,160 cases 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.

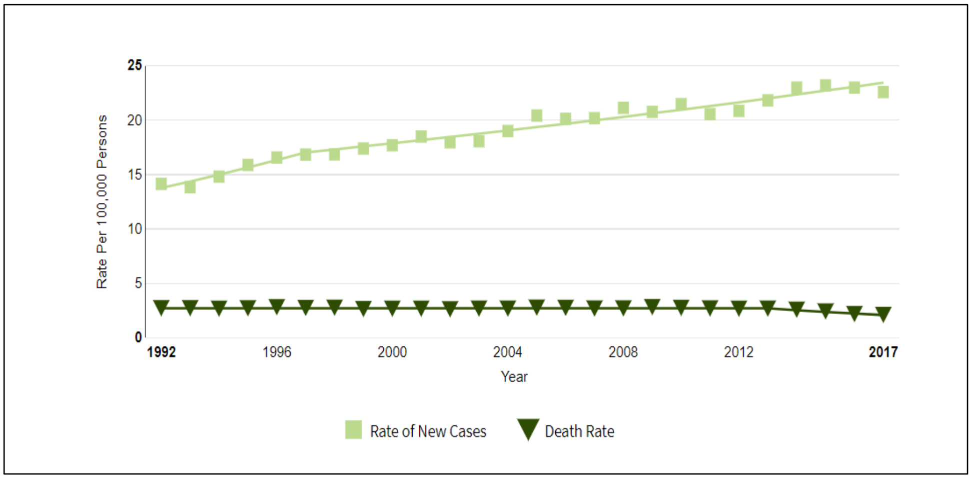


Figure (1.2): Skin Cancer Incidence and Death [2].

1.2 Skin Cancer Diagnoses

1.2.1 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].**

1.2.2 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 **[10]**.

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:

- **Park, D. C. 2016 [11]** proposed a model to 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 taking the advantage of concepts of the Naive Bayes probability classifier, in order to reduce the training time for the algorithm. The number of images in the dataset 800 images divided into four class and each class contains 200 images. The proposed classifier reached accuracy 77.2 %.
- **Shoieb, D. et al 2016 [12]** Presented 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.
- **Nasr-Esfahan. et al 2016 [13],** 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.

- **Mustafa, S. et al 2017 [14]** Suggested a system which can make distinction between the skin lesions 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 discover the optimum line to separate the two classes with accuracy 80% but with low sensitivity 71% and specificity 55%. The proposed method faces a problem with small dataset for training algorithm.
- **Codella, N. C. et al 2017 [15]** 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%.
- **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.

- **Md Ashraful 2018 [17]** 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.

- **Mohan, K. et al. 2019 [18]** Classification of skin cancer was discussed using naive Bayes classifier with shearlet 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.

- **Sanket .K. Chandra J 2019 [19]** 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 better 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

- **Refianti R. et al 2019 [20]** 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 [21]** 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.

1.5 Problem Statement

Skin cancer is a disease that requires 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 and machine 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 as 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 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 chapter one that was alredy discussed here 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 presents 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

Chapter Two

Theoretical Background

2.1. Introduction

This chapter provides an overview of the theoretical background of the main approaches used in this thesis, hair removal algorithm in preprocessing phase. Data mining 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). The median filter sums numbers at a specific perimeter of the pixel of interest, such as adding all the numbers in the adjacent pixels to a given pixel. Then the candidate arranges these numbers and takes the number that is in the middle of the list. This number that codes for a specific color is used for the relevant pixel, meaning that the relevant pixel takes that color [**24**].

Figure (2.1): Skin Cancer with Hair Effects [25]

A-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 [**26**].

$$Opening = IM \ominus SE \oplus SE \quad \textbf{(2.1)}$$

$$Closing = IM \oplus SE \ominus SE \quad \textbf{(2.2)}$$

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

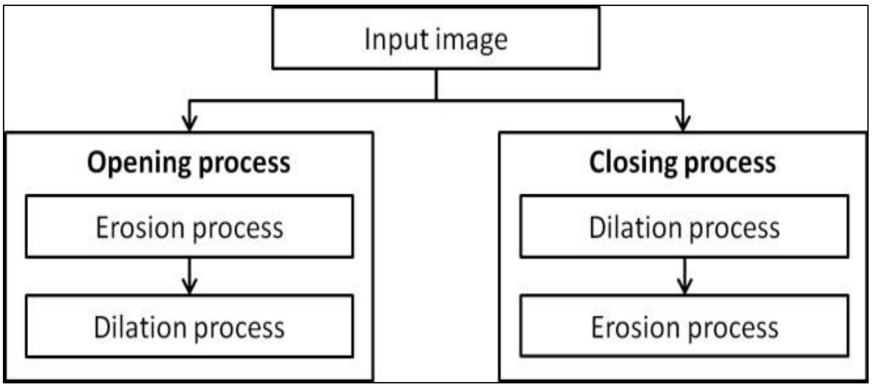


Figure (2.2): Flowchart of the Opening and Closing Processes [26]

B-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 **[27]**.

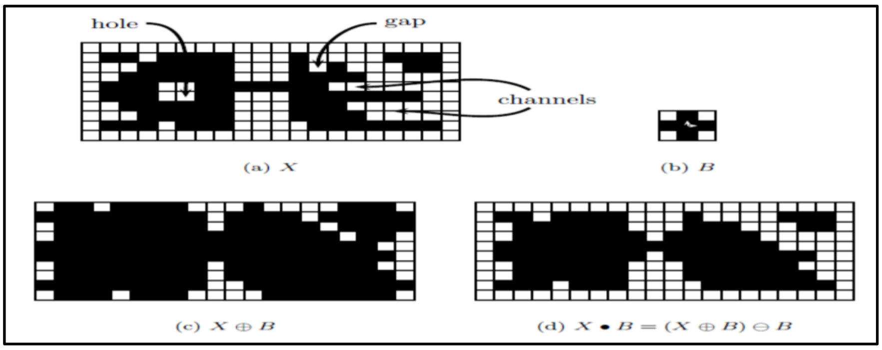


Figure (2.3): Example of Closing Process [27]

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].**

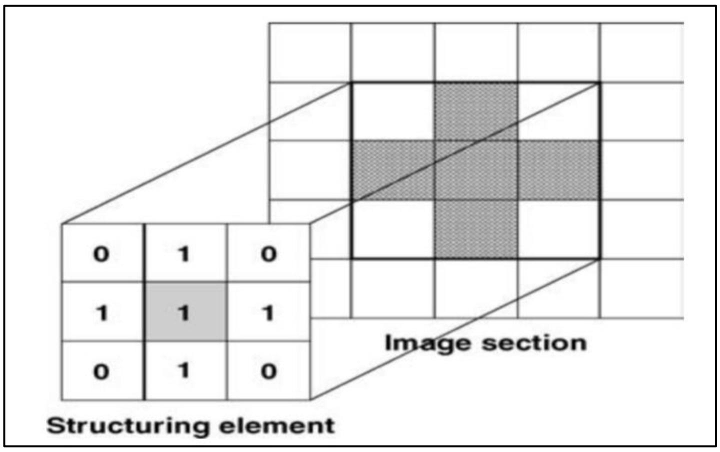


Figure (2.4): Structure Element with Disk Shape [26]

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 using filtering, image gray-scale manipulation, and histogram [**28**].

2.2.2.1 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 [**29**].

A median filter is based on 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 [**30**].

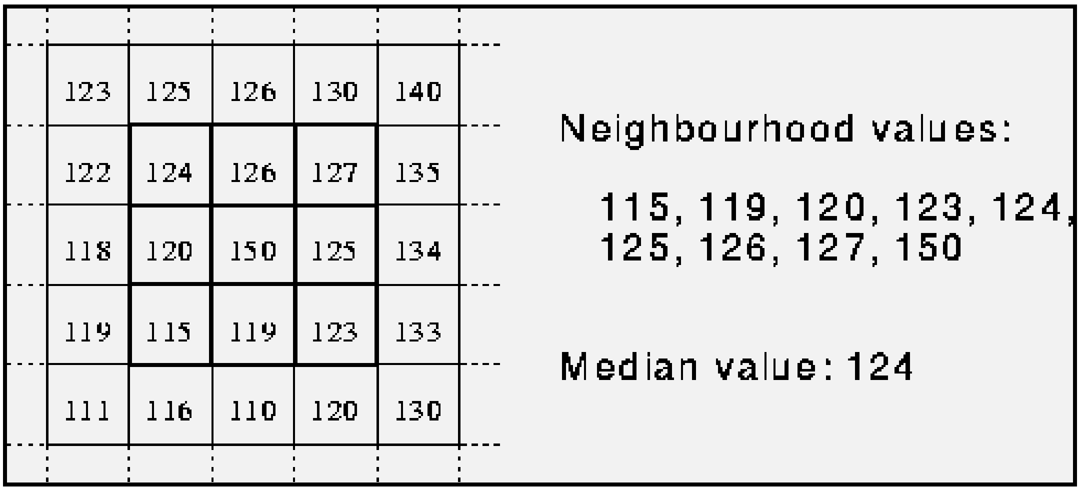


Figure (2.5): Example of Median Filter [30]

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 [**31**], Figure (2.6) shows biological neurons.

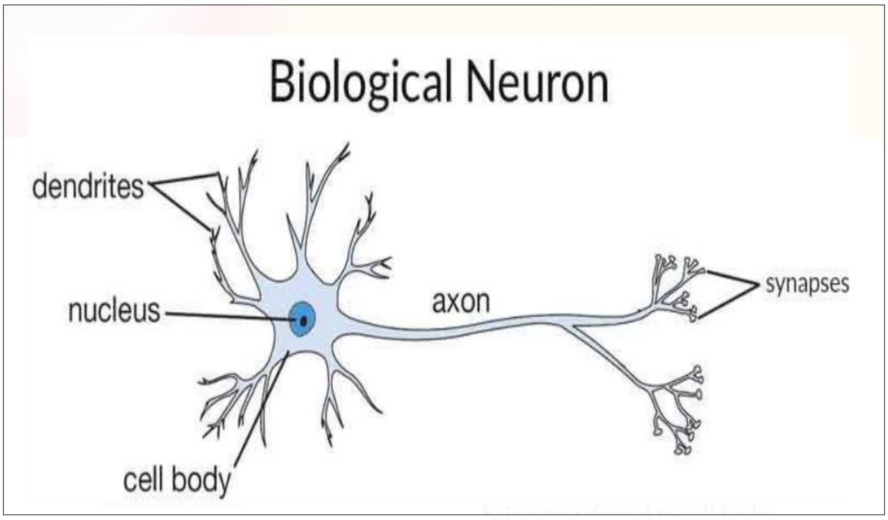


Figure (2.6): Biological Neuron [31]

2.3.1 Neural Network Model

Suppose that the input vector $x = (x_{1,}x_2, x_3, \ldots x_n) \in R^n$ of real numbers. The inputs are sent through the connections to the accumulator and the connections may carry some weight vector $w = (w_1, w_2, w_3, \ldots, w_n) \in R^n$. 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) [**32**]:

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

The accumulated sum of the inputs is compared to a value called the threshold such that if greater than the threshold, then the output is 1 and if less, it is 0. The threshold adds a charge known as the bias to the sum of the signals. The bias is denoted as b. The output produced is given in equation (2.4) [**32**].

$$y = f(w, x) + b \qquad \textbf{(2.4)}$$

Where y is an activation function and f is the output signal. There are three elementary components of the model neurons:

1. A set of synapses (connecting links), each one of synapses is identified by a weight. Specifically, a signal (x_j) at the synapse j input which is attached to the neuron k is multiplied by the synaptic weight (w_{kj}).

2. The adder is used for summing of the input signals (input), that is weighted by the respective synapses (connection) of the neuron.

3. The activation function used for limiting and controlling the amplitude of the output signal (output) of a neuron [**33**].

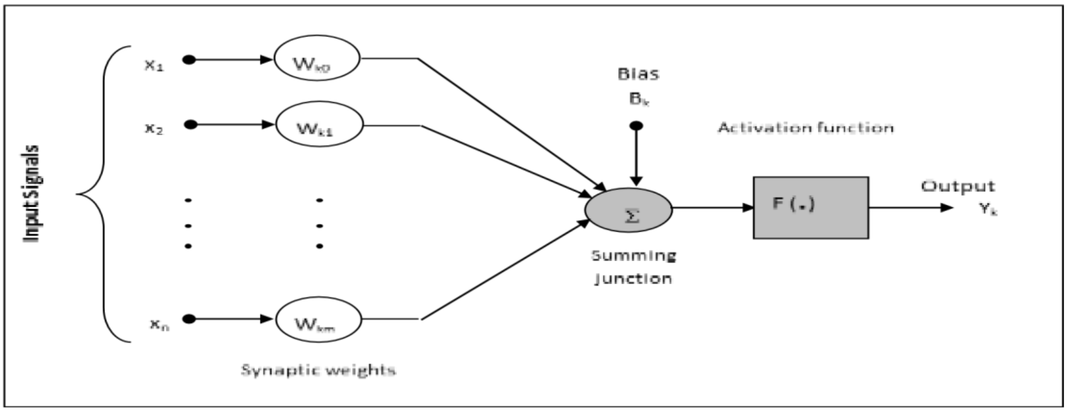


Figure (2.7): Neural Network Model [33]

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 [**34**].

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

$$u_k = (w,x) = \sum_{i=1}^{n} w_i \times x_i \quad \textbf{(2.5)}$$

And

$$y = f[(w,x) = f(\sum_{i=1}^{n} w_i \times x_i) \quad \textbf{(2.6)}$$

where x_i is the input signals, w_{kj} 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 [**34**].

2.4 Convolutional Neural Network (CNN)

CNN is mean the Convolutional Neural Networks is the most popular approaches in deep learning in which have many layers are robustly trained [**35**]. It has been observed to be extremely effective and is additionally the generality by various applications of vision computer. The general structure of the convolutional neural networks system is viewed in figure (2.8) **[36].**

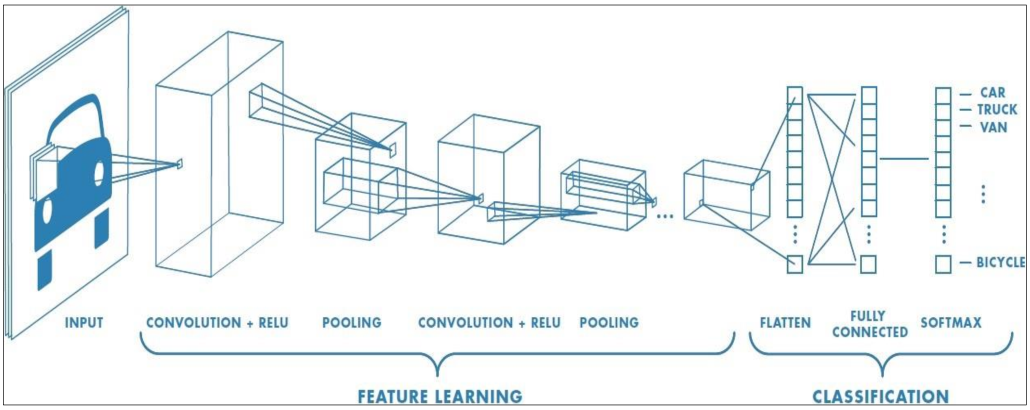


Figure (2.8): The General Structure of the CNN System [36].

Commonly, CNN consisting of three basic neural layers, convolutional layers, pooling layers, and layers that are fully connected. Various types of slices take various roles. The figure (2.8) is showed a general convolutional neural networks structure for classification of image is presented layer by layer **[36].** For train the network, there are two stages: forward and backward.

The forward stage's main objective is to specify the input image in every layer with the proposed parameters (weights and bias). Later the output of the prediction is used with the ground of the truth labels to calculate the estimate of loss. Second, relative to the cost of loss, the backward stage determines with chain rules the gradients of per parameter. All parameters are modified and prepared for the next forward calculation dependent on the gradients. After enough variations of the stages forward and backward, the learning network can be ended **[37].**

CNN and traditional Neural Networks (NNs) have major differences; the best is CNN, due to these differences, which are among the advantages of CNN:

1_ CNN input can handle either 3dimensions or 2D and 1D sizes focused on configurations and parameters, while NN input is just a 1D array

2_ CNN lists more layers than regular Neural Network.

2.4.1 Basic Structure of CNN

Convolution neural network is identical to artificial neural network, as both are composed of self-optimized neurons, introduced by inputs and executing non-linear transformation. Convolution neural network is commonly utilized in pattern recognition on objects relative to artificial neural networks, because it encodes image-specific features in system architecture. In the convolution neural network, there are five fundamental elements: the input layer, convolution layer, non-linear layer, pooling layer and fully connected layer [**37**].

1. Input Layer

The input layer contains the pixel values of the image that enter CNN.

2. Convolution Layers

A CNN uses different kernels in the convolutionary layers to convert the entire object as much as the optimal maps of feature, creating different feature maps as seen in figure (2.9) [**36**].

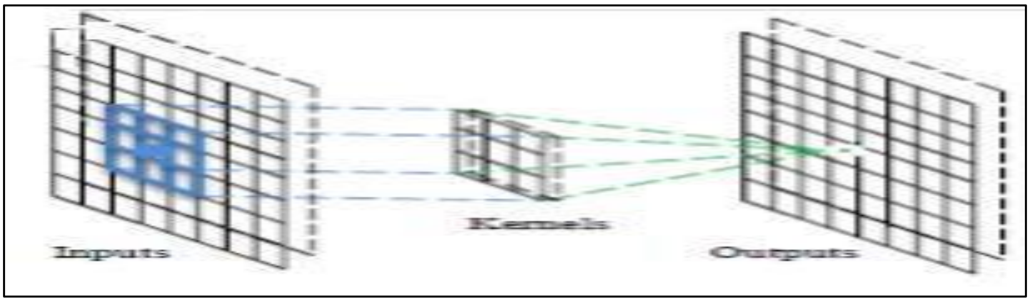


Figure (2.9): Convolutional Layer [36].

In another meaning a flip and gathering over the kernel. This is extended to two dimensions [**38**]:

$$S(i,j) = \sum_m \sum_n I(m,n)F(i-m, j-n) \quad \textbf{(2.7)}$$

In another meaning flip the filter (F) up-down and left-right and sum over all products are shown in an example about convolution layer in figure (2.10).

The convolution method has three key advantages [**39**]:

1) The weight distribution functions in a similar map of feature to decrease the number of parameters.

2) Spatial learns communication associations between nearby pixels.

3) Invariance to the object's position.

The primary idea is that the conventional replacement of convolution layer with a tiny multilayer softmax made up of completely many connected layers with functions nonlinear, substituting linear filtrate with neural networks nonlinear. This approach produces best results when classifying images.

An Example of Convolution Operation

Figure (2.10) shown a, b, c convolution example process with a kernel measurement of [3×3], with no padding and one stride, which the stride means the step scale to vertically and horizontally traverse. The kernel is extended around the insert tensor, and the component-wise function between each object of the tensor input and kernel is measured at every point and averaged up to produce the output amount in the related direction of the tensor output called the map feature [**40**].

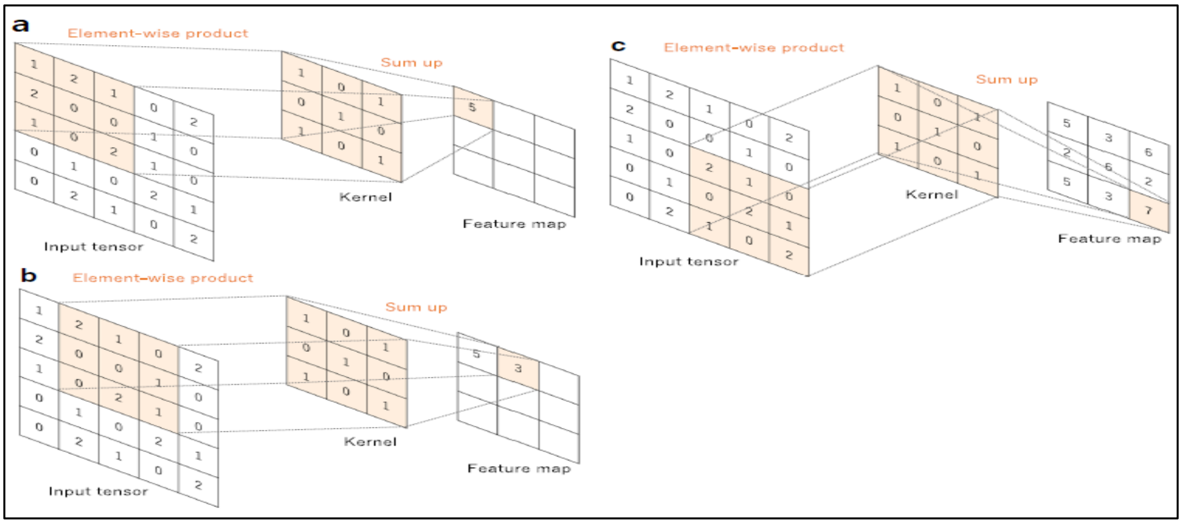


Figure (2.10): An example of convolution operation [40].

Number of parameters in convolution layer in same example can calculated by equation (2.8)

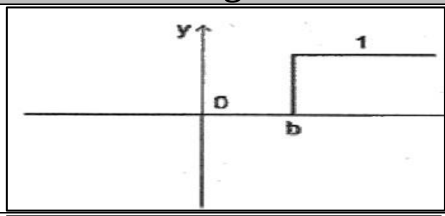
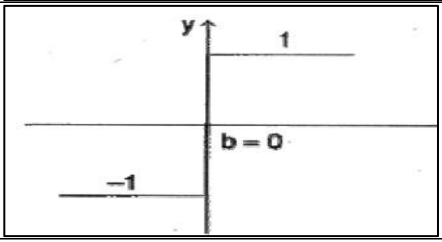
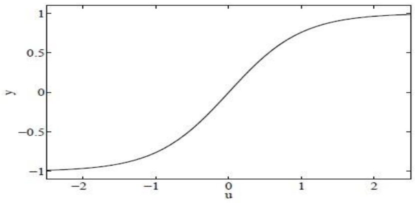
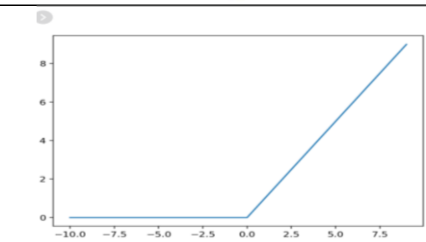
No. parameters=output channels* (input channels* window size + 1) **(2.8)**

Where output channels denoted to features maps that result from convolution layer, input channels denoted to previous layer, window size it means filter size, and 1 refer to stride size.

3. Non-linear Layer (Activation Function)

A non-linear transformation (or called activation function) also is applied to the input by the convolution neural network, the object of which is to classify the features within per hidden layer. The non-linear translation function in the artificial neural network is sigmoid or hyperbolic tangent. Though, if the sparsity of the data is greater, the outcome will be better for image processing. This layer uses many types of activation functions [**41**]. Some of the most common functions of activation listed in table (2.1).

Table (2.1): Type of Activation Functions [41]

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)	$ReLU(y) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases}$	

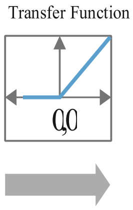
A- Rectified Linear Units (ReLU)

Rectified linear units are commonly used as non-linear transformation, for rectified linear unit, it applies a following equation [**42**].

$$y = \max\,(x, 0) \tag{2.9}$$

So, that the output is in the same size as the input. Rectified linear unit raises the decision function's non-linear properties and has no negative impact on the convolution layer's sensitive fields. The training rate of the rectified linear model is much higher compared to other non-linear units. The figure (2.11) an example of the Rectified Linear Units (ReLU) [**42**].

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 (2.11): An Example of ReLU Transformation [42].

4. Pooling layers

It's usually follows the layer of convolutional and used to minimize parameters of the network and the dimensions of feature maps. Pooling layers are also invariant in translation, alike to convolutional layers because their calculations take into account neighboring pixels. The most widely used approaches are average pooling and max pooling [**43**]. The next figure (2.12) describes the processes of both types of pooling.

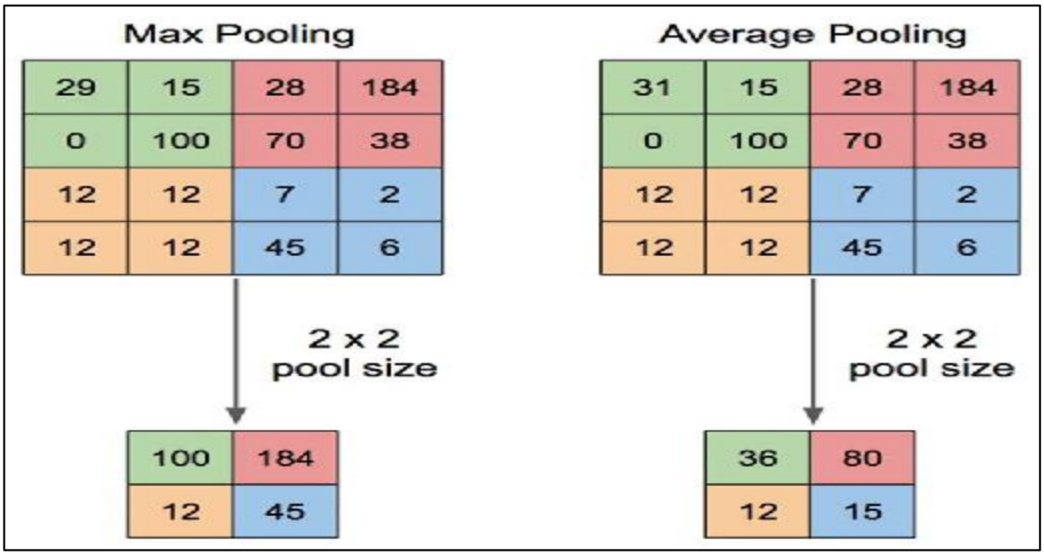


Figure (2.12): Two Classic Pooling Methods [43].

The pooling layer applies the max-pooling function with a 2 x 2 kernel and a stage of 2 along the spatial dimensions of the input, according to the reference cases. It is based on the care about the pooling layer's destructive functionality. Through this process, the feature map is reduced to 25 % of the previous size

while keeping the depth capacity at its standard size. It deducts the layer to use the intervening region to move through the whole spatial dimensionality of the data. If the stage is adjusted to 3 with a kernel size set to 3, the model's output will be effectively reduced [**43**]. Figure (2.13) explains max - pooling operation with 3 × 3 filters.

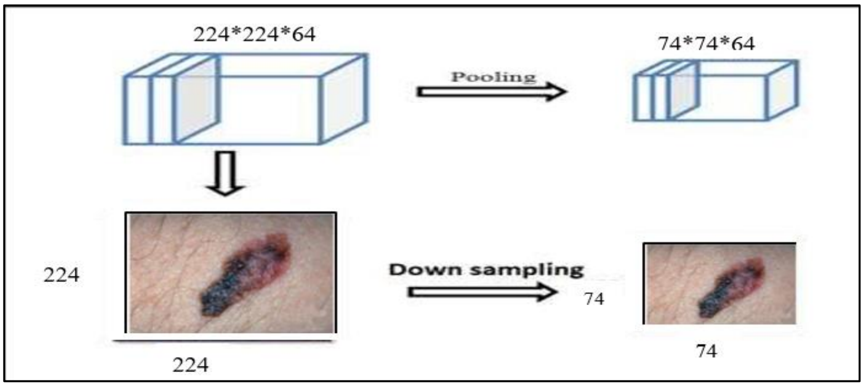


Figure **(2.13):** Max 3 × s3 Pooling Layer to Minimize the Spatial Size Image

5. Fully Connected Layer

The data arrives at the final layer of the convolution neural network, which is the fully connected node, after many iterations of the prior layers. In the two neighboring layers, the neurons are connected directly to the neurons within the fully connected network. The objective from this layer is, to sum up, the weights of the features from the previous layers and show the likelihood of per class. For example, if there is a neural convolution network for gender classification and the result matrix is a probability of [0.8, 0.2], this means that there is 80 % probability of male gender and 20 % probability of female gender **[44].** Also, there are several other concepts related to the CNN structure such as MLP, normalization layer and softmax function.

6. Multi-Layer Perception (MLP)

Finishing a CNN with a multi-layer perception is popular. In practice, this is just one pixel for every filter for convolutional layers. Also used in the term Deep Neural Networks which is the product of adding layers that expanding the network size. The decision boundary is no large limited to a hyperplane when

incorporating hidden layers but can lead to non-linear solutions as shown in figure (2.14) [**45**].

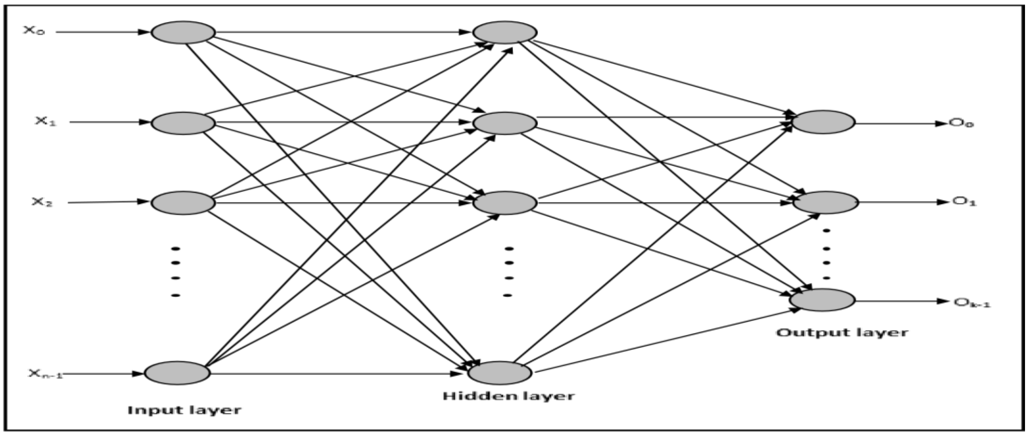


Figure (2.14): Schematic representation of an MLP [45].

7. Normalization Layer

The input layer is normalized by modifying and scaling the activations, for e.g., if it has features from 0 to 1 and also from 1 to 1000, also must normalize those features to accelerate learning. If it supports the input layer, why not do the alike for the values in the hidden layers that shift all the time and increase the training pace by 10 times or more. There are various types of normalization, the most famous of which is the batch normalization.

• **Batch normalization:** is a mechanism for enhancing the speed, execution, and stability of neural networks. In a neural network, batch normalization is achieved through a normalization step that fixes the means and variances of each layer's inputs, to compute the mini batch mean ($\mu\beta$) by the eq. (2.10) [**46**]:

$$\mu\beta = \frac{1}{m}\sum_{1}^{m} x_i \tag{2.10}$$

While computing the mini batch variance by the equation (2.11):

$$\sigma^2\beta = \frac{1}{m}\sum_{1}^{m}(x_i - \mu\beta)^2 \tag{2.11}$$

At the last, normalize the layer inputs by using the prior calculated batch statistics as in the equation (2.12) [**46**]:

$$\overline{x_l} = \frac{x_i - \mu\beta}{\sqrt{\sigma^2\beta + \epsilon}} \tag{2.12}$$

8. SoftMax

The network's performance can be difficult to interpret. It is normal to finish the CNN with a SoftMax function in classification issues. The following equation (2.13) expresses the SoftMax [**45**]:

$$yi = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_i}} \tag{2.13}$$

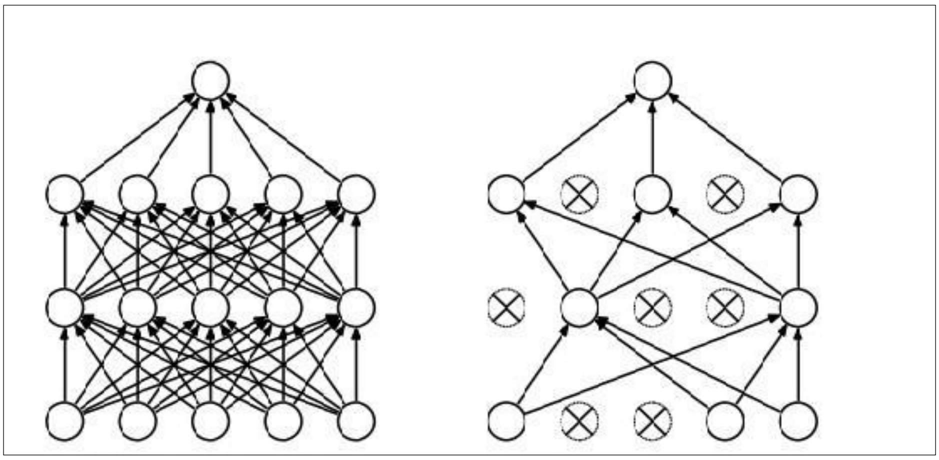
Where the N is the number of the input Z_i is the input layer value, this normalization of the total amount of outputs to one can be defined as the likelihood of the input belonging to class i and therefore the SoftMax output y_i [**45**].

2.4.2 Training a 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 [**47**].

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 [**48**], it is shown in figure (2.11).



a- Neural Network b-After applying dropout

Figure (2.15): Dropout Neural Network [48]

2.4.3 Back Propagation Algorithm

Back Propagation Neural Network is considered as the almost used one in the fields 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 organized in layers, then the output signal sends forward, then the errors of the whole networks is computed and propagated again backward to the same layer. In this case, 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 [**47**] [**49**].

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 [**47**] [**49**].

The computation of an estimate of the gradient vector which is the gradients of the error surface with respect to the weights connected to the inputs of a neuron), which is needed for the backward pass through the network as shown in Figure (2.16).

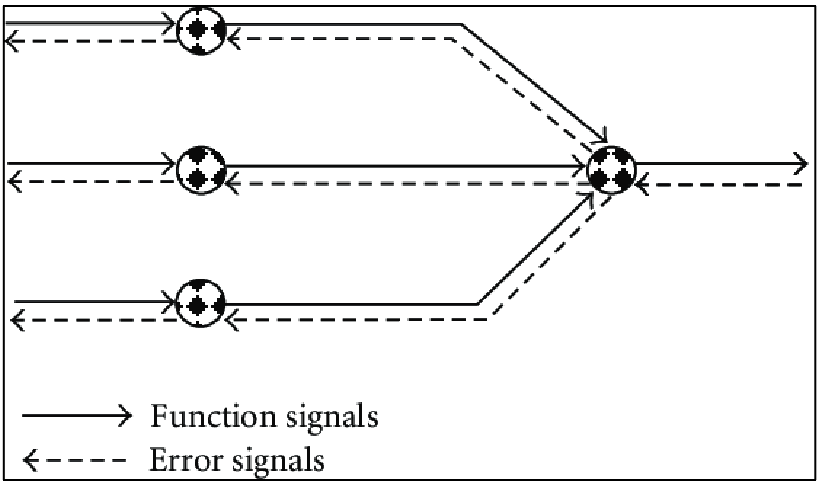


Figure (2.16): Illustration of Directions of Signal Flows [**49**]

2.4.3.1 Stochastic Gradient Descent with Adam (SGDA)

Stochastic gradient descent (frequently abbreviated SGD) is an iterative approach to optimize an actual function with appropriate smoothness (e.g. differentiable or sub differentiable) attributes. It can be considered as a stochastic estimate of gradient descent optimization as it substitutes the true gradient (obtained from the whole dataset) with an average of it (computed from a randomly picked data sub-set). Stochastic gradient descent in machine learning has become a significant method of optimization. The trouble of reducing a

function objective in the form of average is known by both statistical analysis and machine learning [**50**]

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} F_i(x) \qquad (2.14)$$

When the x parameter that decreases F (x) to be determined. Any function summand F (i) is generally joined in the i-th check in the set of data (used for training). A "batch" or standard tendency descent technique would carry the following equation [**51**].

$$x = x - \eta\nabla F(x) = x - \eta \sum_{i=1}^{n} \nabla F_i(x) / \mathrm{n} \qquad (2.15)$$

$\boldsymbol{\eta}$ represent a stage measure (sometimes named the rate of learning in machine learning) [**51**].

2.4.3.2 Loss Function

This function is known as an estimate function, calculates consistency through the network's output estimates by assigned ground truth labels and forward propagation. Binary Cross entropy is the widely utilized loss function for two class grouping, while mean squared error is usually extended to continuous values for regression. One of the hyper parameters is a form of loss function and it requires to be decided according to the tasks provided [**52**].

In deep learning and machine learning loss function is used to optimize a learning algorithm. The loss is calculated on training and validation and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization. An accuracy and validation accuracy is used to measure the algorithm's performance in an interpretable way. The accuracy of a model is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate your model's prediction is compared to the true data [**53**].

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 [**54**, **55**]. 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 *(k)* can be calculated by using the following equation [**55**]:

$$P(c_k|x) = P(c_k)\frac{P(x|c_k)}{P(x)} \qquad \textbf{(2.16)}$$

This equation shows that the computing of $P(c_k|x)$ is a pattern classification problem since it finds the probability that the assumed data *x* belongs to class *k* 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 $P(x|c_k)$ and assume that any particular value of vector *x* conditional on C_k is statistically independent of each dimension and can be written as follows [**56**]:

$$P(X|C_k) = \coprod_{i=0}^{n} P(X|C_k) \qquad \textbf{(2.17)}$$

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 assumption results in simpler calculation cost and efficient data processing. By combining equation (2.16) and equation (2.17), the Naive Bayes classifier can be summarized as the following equation:

$$k = argmax_k P(C_k) \coprod_{i=0}^{n} P(X_i|C_k) \qquad \textbf{(2.18)}$$

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. **[56]**

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 (2.21) 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 [**57**].

We can get to mean from the equation [57]:

$$\mu = \frac{1}{n} \times \sum_{i=1}^{n} x_i \qquad \textbf{(2.19)}$$

And stander deviation from equation [57]:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \qquad \textbf{(2.20)}$$

So, the probability density function [57]:

$$P(X_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{c_k}}} e^{-\frac{(x_i-\mu_{c_k})^2}{2\sigma_{c_k}^2}} \qquad \textbf{(2.21)}$$

where the probability density function is formed during the training stage of local classifiers with the mean, μ_{c_k} and standard deviation, σ_{c_k} of each two-class data for each feature vector component x_k.

Note that $P(X|C_k)$ can be calculated by using equation (2.22) 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 [**57**]:

$$Class(x) = argmax_k \frac{1}{N} \sum_{j=1}^{N} P(C_{ij}) \prod_{i=1}^{n} P(X_i|C_{jk}) \quad \textbf{(2.22)}$$

2.6 Support Vector Machines (SVM)

The algorithm was invention in (1995), introduced this algorithm to resolve the kind of specific regression and classification problems. This method is based on the learning statistical method [**58**]. Its work to determine the optimal hyperplane that maximizes the classes' margin. SVM has become one of the classification's most popular methods. It is one of the supervised classification methods in which the inputs series with their labels is given. The structures defined by the vector features of these inputs. This algorithm produces a hyperplane distinguishing the two groups in order to achieve total separation between these classes. In figure (2.17) there are two vectors parallel to the classifier that passes via several stages. The distance between these two parallel vectors is called the (margin). The points on boundary vectors are called support vectors [**59**].

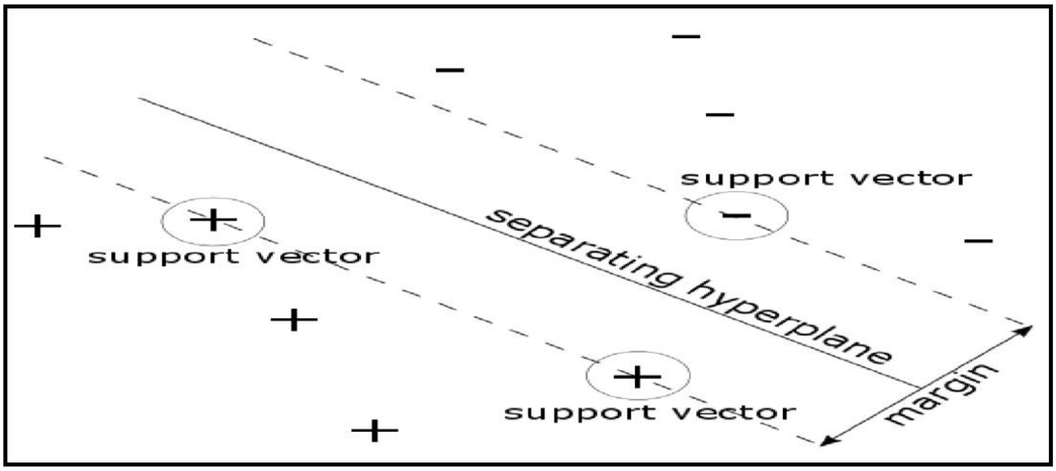


Figure (2.17): The SVM hyperplane between two classes [59]

The hyperplane is illustrated in figure (2.17) the separating data points from two groups as + and -. The vectors and margin of support are labeled. SVM algorithm is divided into two kinds Linear and non-Linear SVM that illustrate in the following section:

2.6.1 Linear SVM

If the linear SVM hyperplane, then the supper vector machine is defined as linear SVM, e.g. if z represents pairs training (x_i, y_i) when i=1,2...z, with category labels y (1, -1) the direct equation is defining the hyperplane [**60**]:

$$W.x + b = 0 \tag{2.23}$$

Where W is a vector of weight, W= {w1, w2, ... wn}, b is a bias, and x represent attributes. The data classification is considering as equation [**60**].

$$f(w.x.y) = sgn(w.x.b) \tag{2.24}$$

Where f(x) is the function of a hyperplane in m dimensions that is get as a series of every points x the (x ∈ Rm) that fulfils the equation f (x)= 0 such that the function hyperplane f(x) functions as a classifier linear predicting class y for each presented point x, depended on the subsequent rule decision [**60**]:

$$W^T.x + b \geq 1 = +1 \tag{2.25}$$

$$W^T.x + b < 0\, for\, y = -1 \tag{2.26}$$

Maximizing the margin is a constrained optimization trouble that Lagrange method can solve. A Lagrange multiplier (α_i – i) explains every training point x_i

$\alpha_i = 0 \longrightarrow$ $\boldsymbol{x}_i$ has no influence on the hyperplane.

$\alpha_i > 0 \longrightarrow$ $\boldsymbol{x}_i$ these points support vectors that are nearest to the hyperplane.

Also, can calculate weight and bias when obtaining the αi value, the weight calculation using the following formula [**60**]:

$$W = \sum \alpha_i x_i \tag{2.27}$$

Points with (α_i = 0) do not consider SVM, therefore the average of the SVM that this is (α_i) not equal to zero must take. When support vector with (α_i= 0) does not play any role in deciding [**60**].

2.6.2 Non-Linear SVM

In most cases, the linear classification does not consider the appropriate classification approach for the non-linear classification used in such situations, where a non-linear kernel function will be used [**59**]. Linear SVM are fast to train and implement, but with many training examples and not too many features they appear to underperform on complicated datasets. In many applications, non-linear SVM can be more consistent for quality across different problems and the preferred choice, although they lack critical power [**61**].

A- Kernel Function

To move testing samples and training to feature space high-dimensional feature space, functions kernel is implemented. This section, functions kernel is described to replace functions of mapping because the kernel computation is most effective than the function of mapping, computation period usually saved to exchange functions of mapping with the used of kernels [**59**]. SVM implements the kernel function, K (x_n, x_i), which transforms the raw data space into a higher-dimensional new space. It processes involves the dot product transformation function ϕ(x) (Equation (2.28). The target is the information that

can be feasibly extracted, that has been converted into a higher dimension. It is possible to write the hyperplane function in formula (2.29) [**62**].

$$K(x_n, x_i) = \emptyset\,(x_n)\emptyset\,(x_i) \tag{2.28}$$

$$f(x_i) = \sum_{n=1}^{N} \alpha_n y_n\, K(x_n, x_i) + \text{b} \tag{2.29}$$

Where α_n is a lagrange multiplie, x_n is support vector information, and y_n is a membership class label (+1, −1) with n = 1, 2, 3, ……., N,[**62**].

B- Examples of kernels

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

1. **Linear** which is describes as equation [**63**]**:**

$$K(x_n, x_i) = (x_n\ .\ x_i) \tag{2.30}$$

2. Polynomial kernel which is describe as equation [**63**]:

$$K(x_i, x_j) = (\langle x_n\ .\ x_i\rangle + C)^d \tag{2.31}$$

3. Radial Basis Kernel Function (RBF) which is describes as equation [**63**]:

$$K(x_i, x_j) = e^{\frac{-\|x_i, x_j\|^2}{2\sigma^2}} \tag{2.32}$$

4. Sigmoid kernel function which is describe as equation [**63**]:

$$K(x_i, x_j) = \tanh\,(k(x_i, x_j) + \theta \tag{2.33}$$

Where d, σ and Θ are parameters of the specific kernels [**63**].

2.7 Evaluation Measures

To calculated evaluating act of proposed system may using four measures called Accuracy, Sensitivity, Specificity and precision.

A. Accuracy or 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 that are applied during the training and testing. The formal of the accuracy it is definite [**64**].

$$Accuracy = \frac{TP+TN}{Total\ number\ of\ test\ samples} * 100 \tag{2.34}$$

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:

Sensitivity is mainly calculated to define the ratio between the numbers of correct positive prediction and the total number of the positive prediction [**64**].

$$Sensitivity = \frac{TP}{TP+FN} \tag{2.35}$$

C. Specificity:

Specificity is denoted the ratio between the numbers of correct negative prediction and the all number of negative predictions [**64**].

$$Specificity = \frac{TN}{TN+FP} \tag{2.36}$$

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 [**64**].

$$Precision = \frac{TP}{TP+FP} \tag{2.37}$$

Chapter Three

Chapter Three

Proposed System Design

3.1 Introduction

This chapter included the proposed skin cancer images detection and classification system by using machine learning and deep learning technique. It will start by introducing and discussing the general block diagram of the system. Furthermore, this chapter will present the architecture of the system in details with the proposed algorithms in the different stages of the system.

3.2 The Proposed System Design

The proposed system aims to identifying and classifying skin cancer image, the work includes three proposed systems. They are separated based on classification algorithm, each one is different from the other.

According to figure (3.1) below, the system will be divided into three parts depending on the classification method. Each proposed system divided in to two types (system with apply preprocessing and other without applying preprocessing). This system will classify skin cancer image and take into account the most common skin cancer types (benign or malignant).

Through this system and at the end, skin cancer will be detected and classified by applying the stages of the proposed methodology of the system. Figure (3.1) shows the general block diagram of the system for detecting and classifying skin cancer and its types will be explained later.

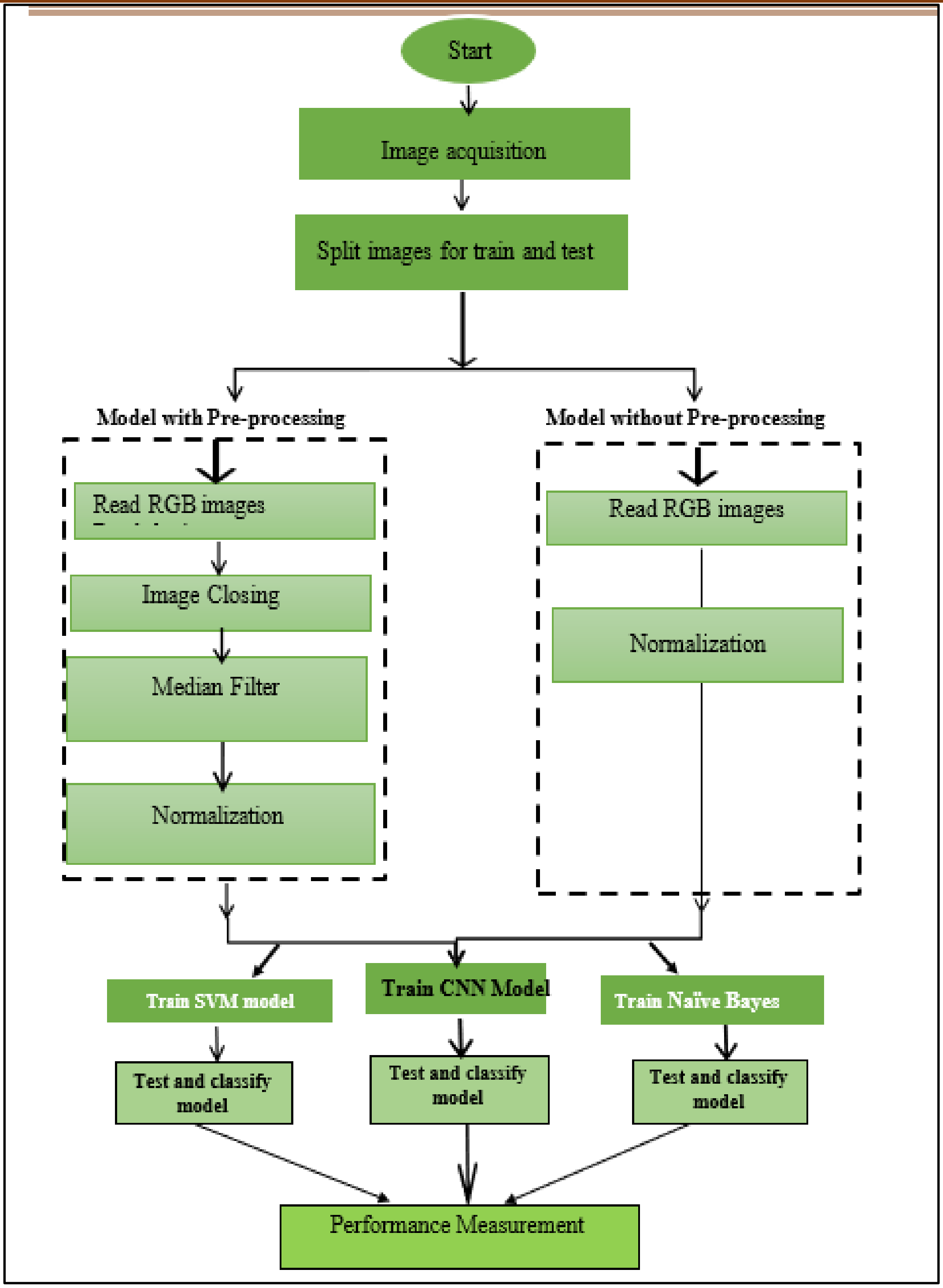
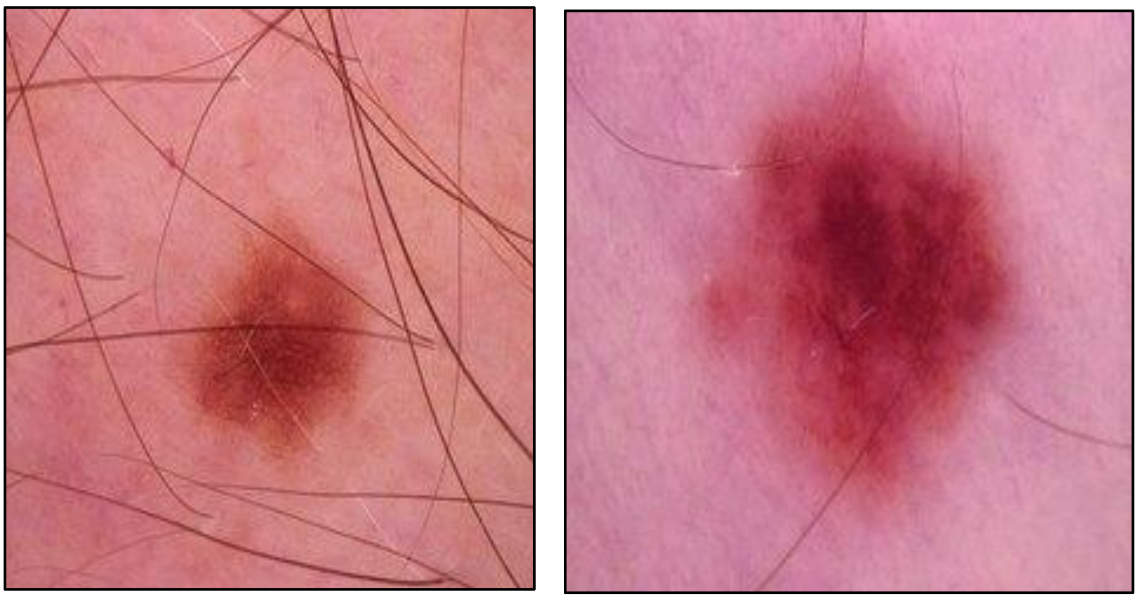


Figure (3.1): Block Diagram of General Proposed System

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 2016, 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 [25]

3.4 Image Preprocessing Stage

In preprocessing stage hair removal algorithm applying on images, this algorithm consists the morphological image closing and median filter.

3.4.1 Hair Removal Algorithm

A-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): Apply Image Closing Operation on image
Input: RGB Skin Cancer Image with size 224*224 **Output**: RGB Skin Cancer Image
Begin **Step (1):** Read RGB skin cancer 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):** Apply Dilation then Erosion for each layer R, G, B by using structure element of size (3x3) **Step (6):** Save result for next stage **Step (6):** End algorithm

B- 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): Apply Median Filter on Image
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 hair removal algorithm (3.3).

Algorithm (3.3): Hair Removal
Input: RGB Image with size 224*224 **Output**: RGB Skin cancer image
Begin **Step (1):** Read RGB skin cancer 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):** Apply Dilation then Erosion for each layer R, G, B by using structure element of size (3x3) … (from algorithm 3.1) **Step (6):** Apply median filter for each layer R, G, B with size (3*3) … from algorithm (3.2) **Step (7):** Test the removal hair algorithm on image load the image Repeat from step 2 to 6 **Step (8)**: Display tested image END

3.4.2 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. as it will be presented details in this chapter.

3.5 Classification Stage Using First Model (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.

The diagram for the first proposed system to classify and detect the skin cancer image will be displayed in figure (3.3). Then the algorithm (3.4) for CNN training will be showed. In this algorithm, the work of the first proposed system and the processes that occurred to detect and classify skin cancer with two types(with and without applying preprocessing) were presented starting from the first stage, which is pre-processing, as well as splitting dataset to training and testing, that have been chosen for the ratio It is 80 % for training and 20 % for testing.

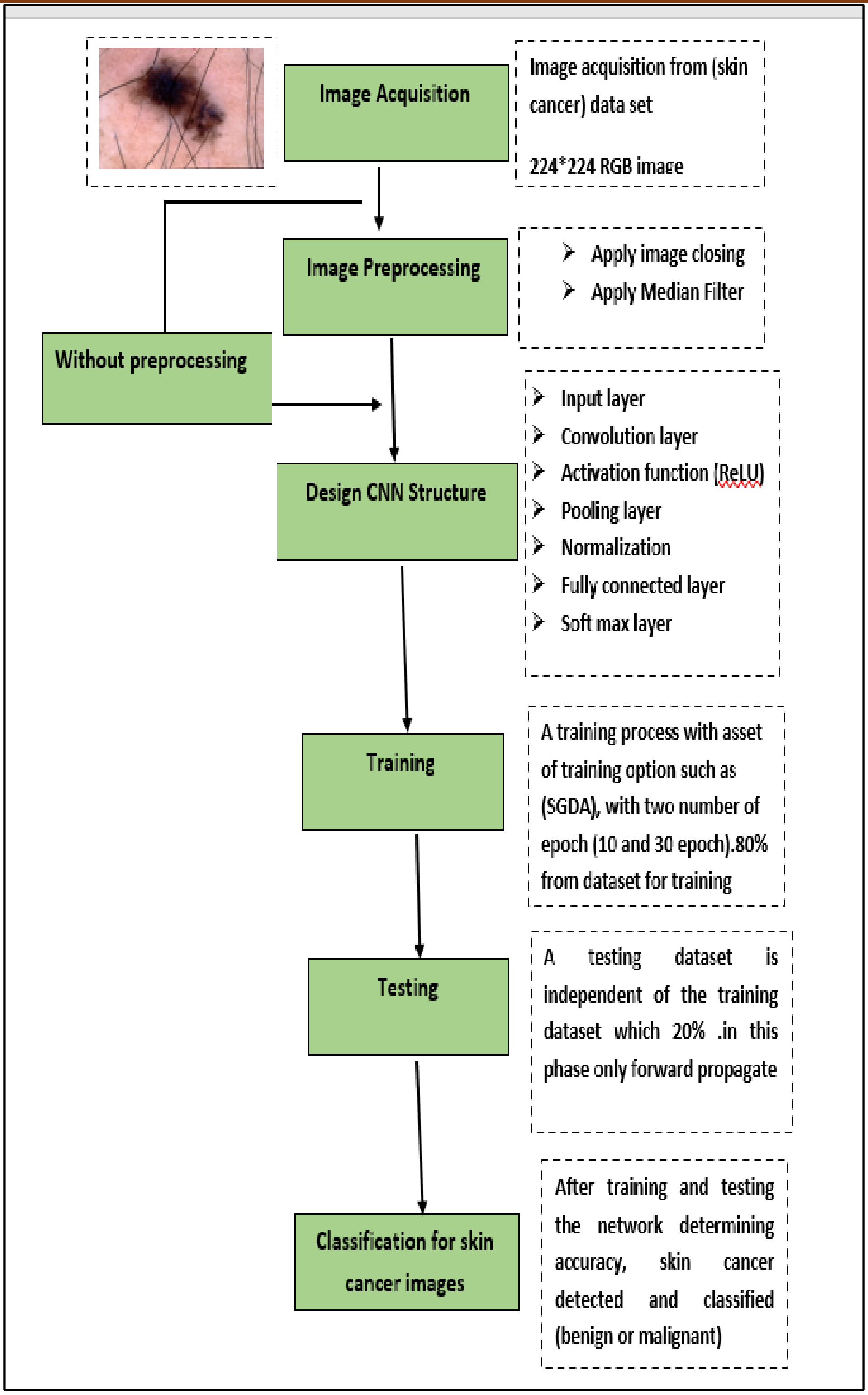


Figure (3.3): Block Diagram of First Proposed Model

Algorithm (3.4): CNN training algorithm to classify skin cancer images
Input: RGB Image of Skin Cancer with size 224*224 **Output**: CNN with the optimum weight

Begin

Step (1): Splitting the data into two parts, 80% for training and 20% for the testing.

Step (2):1. If network process image with hair removal algorithm, go to step (3).
2. Else, go to step (4).

Step (3): Pre-processing stage (hair removal …………as algorithm (3.3)

Step (4): Implement CNN algorithm without apply preprocessing

Step (5): CNN design which consisting of several layers:
a) Input layer: RGB image with 224 * 224 size
b) Convolution layer: Multiple filters were used with size 3*3
c) Nonlinear layer (Activation layer): Using Rectified linear units (ReLU) ……… as Equation (2.9)
d) Pooling layer: Using Max-pooling layer
e) Normalize layer: using batch normalization as algorithm(3.5)
f) Fully connected layer
g) Softmax layer ….by using Equation (2.13)

Step (6): For each pattern in the training dataset:
a) Input current pattern (input Image with Label)
b) Calculate the real output of the CNN through Softmax layer
c) Calculate the error rate by comparing the real output with the desired output.
d) Compare the performance goal with the error rate:
1. If the performance goal was not meet, change the connection weights by using the back-propagation learning algorithm.
2. Else, go to next image pattern.
e) Stop condition:
1. If the performance goal was meeting with validation data or the maximum iteration was achieved, go to step (7).
2. Else, repeat step (6).

Step (7): Return the CNN with the optimum weight.

Step (8): END

3.5.1 Design Convolution Neural network (CNN) Structure

The structure is designing in the figure (3.4) to suit the proposed system and after changing many of the parameters and testing it, by choosing this design for the network for obtaining the best results.

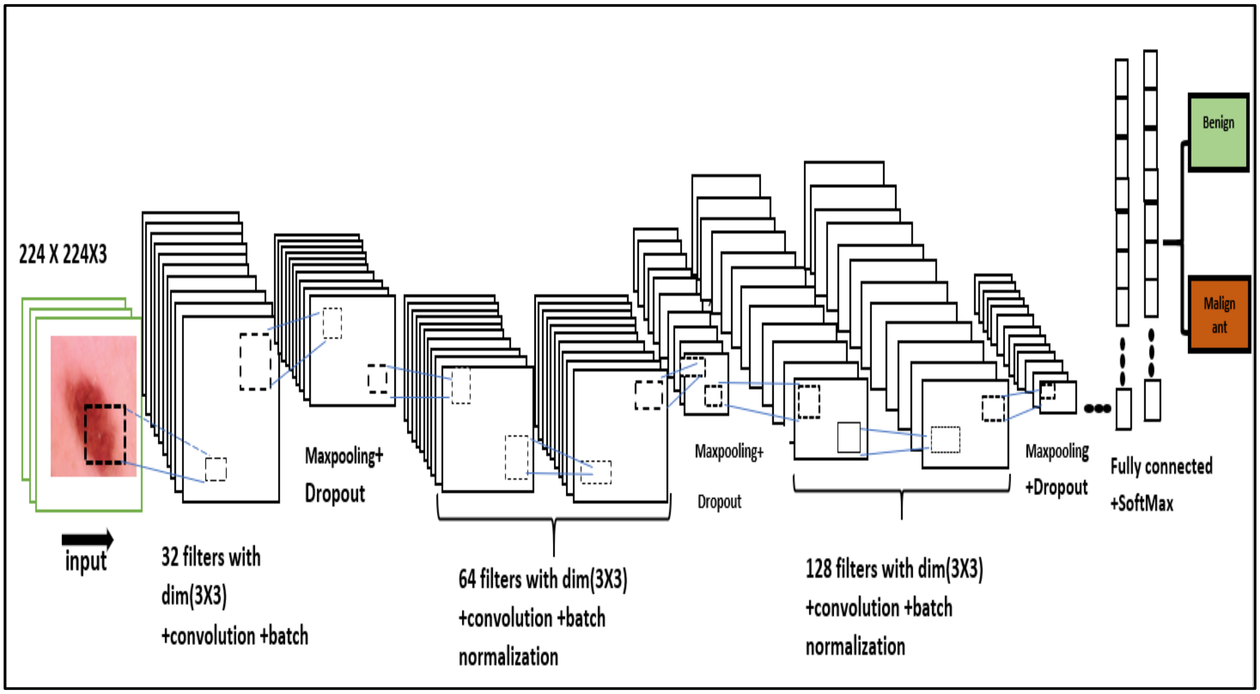


Figure (3.4): Structure of the CNN algorithm

As shown in the figure (3.4) above, the structure of the CNN algorithm consists of several layers for each layer a specific work and a different structure, the structure is designed as follows:

1. Input Layer

The input layer contains the input images and their pixel values. These images are the images of skin cancer that entered in matrix form, or in other words layers, whose size has been 224*224*3. The input layer consists of three layers because the dataset is RGB color images are red, green, and blue, so each color has a specific layer as in the example shown below in figure (3.5) that explains the input

image is an image of the affected skin and the process of converting it into matrixes to be dealt with it in the other stages.

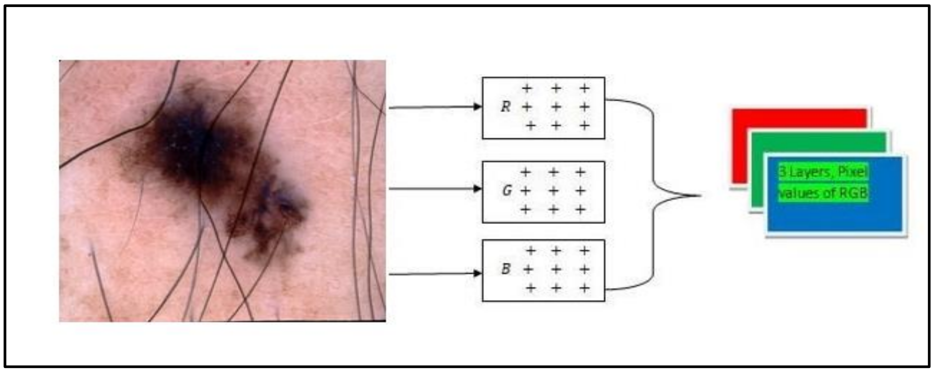


Figure 3.5: The input layer (input image).

2. Convolution Layer

A convolution Neural Network uses the different kernels, in the convolution layers to convert the entire object as well as the optimal feature maps creating different feature maps. Not only does CNN utilize kernel convolution, but it's also a staple of many smart vision algorithms in usual. It's a mechanism that takes a short array of numbers that named a kernel or filter; the filters are the shared weights and bias. These filters work on every section of the image it searching for the similar feature throughout the image.

The values are calculated according to the equation that was calculated in the second chapter in the part (2.4.1.2), which is the equation (2.7). Also, as in the example showed also figure (2.9) in the previous chapter also, that clarifies the work of the convolution layer. In the first proposed system, five convolution layers are used. In the first convolution layer, 32 filters were used with dimension 3*3 with “same padding”. The padding applied to the input along the edges, the 'same' means padding is set so that the output size is the same as the input size. In the second and third convolution layer, 64 filters were used with dimension 3*3 with "same" padding. In the fourth and fifth convolution layer, 128 filters were used

with 3*3 dimension with padding 'same', as shown in figure (3.4). Choosing the number of filters in each of the convolution layers, based on several experiments that prove the best result that obtained of these numbers that were used in each level.

3. Non-linear Layer (Activation Function)

A non-linear transformation is implemented to the input by the CNN, it is also called (Activation function), and the node activation function describes the node output provided by the input or input set. Rectified Linear Unit (ReLU), which was explained previously in chapter two, section (2.4.1.3.A). In this function, negative values in the matrix resulting from the previous step are converted to 0 and positive values remain the same. It also showed with its calculation in the previous chapter in the equation (2.9), and its work also has been clarified more in the example in figure (2.11). They are used in the proposed system after each level of the convolution layers were shown in figure (3.4).

4. Pooling Layer

Another building block to the CNN is a pooling layer; the purpose of this layer is to gradually decrease the spatial scale of the representation in order to reduce the number of parameters and the network calculation. A pooling layer often follows a convolutional layer and can be utilized to depreciate the dimensions of feature maps. Pooling layers are also invariant in interpretation, like to convolutional layers because their calculations take into account neighboring pixels were explained in detail in chapter two in the section (2.4.1.4). The most widely used approaches are average pooling and max pooling layer, in the proposed system the Max-pooling layer is applied. As explained in the previous chapter in the example was shown in figure (2.12), after performing the experiments, Max pooling was chosen because it gives better results. It takes the highest value in the matrix specified for pooling and passes the process to all the

values of the matrix and therefore has another matrix with fewer dimensions. In the proposed system 2*2 Max-pooling is used with two strides, the "stride" is mean “the step scale to vertically and horizontally traverse the input” is shown in figure (3.6) to explain Max-pooling layer process.

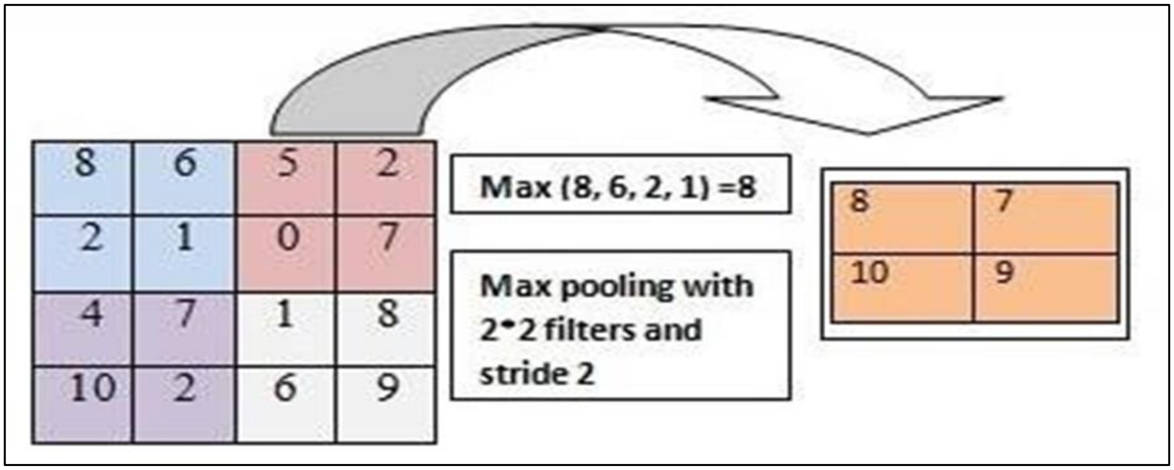


Figure (3.6): Max pooling layer

5. Normalization Layer

It is a very important layer that was explained in chapter two, section (2.4.1.7). Batch normalization layer is using in the first proposed system; it forms norms any channel by means of a mini-batch. This can help diminish sensitivity to variations in the results. It eliminates the fitting problem, as it has a slight effect on regularization. Batch normalization optimizes the yield of the prior activation layer, to calculate the batch normalization, at first calculate the mean and the variance for the layer that was inserted, and then do scaling and shifting to calculate the batch normalization, as shown in the algorithm (3.5).

Algorithm (3.5): Batch Normalization
Input: The matrix of the convolution layer, suppose C = (x_1 .... m)
Output: Batch normalization (Y_i =BN), β(xi)
Begin **Step (2):** Calculate the mini-batch mean ($\mu\beta$) of the layers input by using equation (2.10). **Step (3):** Compute the mini-batch variance of the ($\sigma^2\beta$) layers input by using equation (2.11). **Step (4):** Normalize the layer inputs utilizing the prior calculated batch statistics $\overline{x_\iota}$) by using equation (2.12). **Step (5):** Scale and shift to get output from the layer $Y_i \leftarrow \overline{x_\iota} + \beta \quad \equiv$ BN, (xi) **Step (6):** Return Batch Normalization **Step (7):** End algorithm

6. Fully Connected Layer

Often the last layers of a CNN are fully connected layers, in a traditional neural network; these would behave identically to the layers. The major difference is that the inputs would be in the form that CNN earlier stages would build. In the two neighboring layers the neurons were connected directly to the neurons within the fully connected network. So, what this implies is that every one of neurons in the fully connected layers can collect input data components over time that would help it to predict the right class value in the soft max layer afterwards are shown

in figure (3.4). The objective from this layer is to summarize the weights of the features from the prior layers, and show the value of per class, as was explained previously in the previous chapter in part (2.4.1.5). In proposed system two classes that will be produced from this process because according to the data used for training and determining the number of classes that have been extracted from this stage, these classes will be in the form of value for each class linked to the fully connected layer image and Softmax will be made for them in the last step as seen in figure (3.4).

7. Softmax Layer

The network's performance can be difficult to interpret, it is normal to finish the CNN with a soft max function in classification issues. The result values are between the [0, 1] range which is good for binary classification. The Softmax function equation was mentioned in the previous chapter in part (2.4.1.8) in equation (2.13). After extracting values of two classes of skin cancer in the fully connected step, a Softmax will be made for them, so that the class will be selected in each process and according to the features that were extracted through the previous layers that the images of plant diseases went through. Firstly, extracting the exponential for the inputs, and then dividing the values by the sum of the exponential values in order to obtain probabilities through which to classify them as shown in the algorithm (3.6). So, the greatest probability is the correct class that indicates the type of the skin cancer. According to these probabilities, the skin cancer type is classified between two classes, as shown in figure (3.4).

Algorithm (3.6): Softmax layer function
Input: Fully Connected Layer Values Z_i // Fully Connected Layers Values
Output: Softmax probabilities value for two classes (benign, malignant)
Begin **Step (1):** Calculate the exponential for every input in fully connected layer $\boldsymbol{e^{z_i}} \leftarrow \boldsymbol{e}$ **Step (2):** Calculate the exponential summation for two class of input fully connected layer $\sum_{j=1}^{2} \boldsymbol{e^{z_i}}$ **Step (3):** Calculate the soft max function (y_i) by using equation (2.13) for two classes after calculating the exponential for each class in step (2), and divide each of them by the sum of the two classes after calculating the exponential for them in step (3), to predicate a true class that has the highest probability. **Step (5):** Return softmax probabilities value for the classes of skin cancer (y_i) **Step (6):** End algorithm

3.5.2 CNN Training

Training a network is a procedure of obtaining kernels in convolution layers and weights in fully connected layers that reduce differences on a training dataset between output predictions and specified area truth labels, 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 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 for use later. 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) as explained in the second chapter in section (2.4.3.1).

The training stage, need three things for training, which is the training set that is obtained from the dataset, the layers in which the network was built, and the different training options that were made for training. Also, this structure has a very important function for evaluating the training process, which is the Loss function, which will also be used.

a. **Training Set**

The data is divided into two main sections, which are training data and testing data. The data is dividing using a function in the Python called " train _test _split ", this function divides the data according to the percentages determined by the user and randomly. It was randomly divided and not chained to obtain a better system and data is randomly taken from the dataset to make detection and

classification later better and stronger. In the proposed system the data was divided so that the training data would be 80% and the testing data is 20%, after testing all ratios, and this best ratio.

b. **Layers (designing)**

The layers are intended as the layers that were built in the construction of the CNN structure in the training stage, all of these layers that mentioned earlier will also be passed in order to extract the features and learn from them to be classified through these features. Training for all data for training will be passed through the layers that built.

c. **Training Option (training algorithm)**

In order for the training process to be completed, that need multiple options that were used in this process using Python program by parameters that are created in the training process these options are:

1. Stochastic Gradient Descent with Adam (SGDA)

The method is utilized in the data training process, due to its good properties that fulfill the purpose of training. It is an optimization method utilized to train CNN and machine learning systems. Adam is an optimization algorithm that can be used to update network weights iterative based in training data, and its tool that helps move vectors of gradients in the true directions and thus contributes to faster convergence. It is one of the most famous optimization algorithms and it is used to train several states of the art systems. They were explained in the previous chapter in section (2.4.3.1) and illustrated in equations (2.14) and (2.15).

2. Max Epoch

The “Epoch” is a metric of how many times all training vectors are once utilized to update the weights. Concurrently in one epoch in the learning algorithm, before weights are upgraded. The maximum number of epochs that used for training is 30 epochs. In addition, tried with 10 epochs before 30 epochs

in two type (CNN with preprocessing and without), and batch size is 64. As shown in next chapter.

3. Shuffle

The data is shuffled to have various data for per batch, this command is the training data is shuffled. There are many options for shuffle dataset in python, in CNN “random. Shuffle ()” is implemented and the dataset will be shuffled before each training epoch.

4. Validation Data

It’s the information to use throughout the training for validation. This may be an Image data store with categorical labels, a Mini-Batch data store with specified responses, and a table where unless image paths or images are in the first column, a cell array {X, Y} where X is a numerical array with the input data and Y is a return array. In the proposed system, this option was used (testing set, dataset for the testing set. labels) for validation data. The testing set mean is the group that was used for the later testing process, while the dataset is a set of datasets that have taken as a label form.

5. Validation Patience

The number of times the validation loss can be greater than or equal to the initially smallest loss before the network training is finished. There are two choices either an "Inf" choice or a "positive" choice. The definition of the "Inf" choice is the automatic ending of training the network, and this choice is the default choice for training the network. The other choice, which is positive, means specifying a "positive" integer for stop network training. In the proposed system that uses the "Inf" choice.

d. **Loss Function**

A loss function, also known as a cost function, can be calculated the consistency between the network's output estimates by forwarding propagation

and assigned area truth labels, it helps in the optimization of CNN parameters. The important purpose here is to reduce a convolution neural network's loss by optimizing its parameters (weights). The loss is measured utilizing loss function, by a convolution neural network combining the target (actual) result and the predicted value with errors. As explained in the second chapter in section (2.4.3.2), also the back-propagation that explained in the second chapter in section (2.4.3), the loss function and accuracy function was also calculated and displayed by using the confusion matrix as it will be presented in chapter 4.

3.5.3 CNN Testing

The testing is a dataset utilized to provide an impartial final design fit evaluation on the training set of data. In this stage, the system uses the groups that were trained in the previous step that was trained in CNN, and the features were extracted by learning the network when the dataset passes from skin cancer image on this network. The dataset also used that was allocated to the testing process when it was divided, based on this dataset and the training data for testing; the classification process for skin cancer image is done. Before the testing stage, there is the training stage; this means that the network is trained when inserting any image of skin cancer image. The type of skin cancer will be determined because the network was previously learned and trained. So, the main difference between training and testing is test data that are unlabeled, unlike training data that is labeled. In the proposed system, uses this function in Python, scores = model. Evaluate (X _test, y _test), this categorizes every row of the dataset to one of the training groups. Sample and training must be arrays with the same column size. Training the Group is a group variable and the specific values determine groups, and each component determines the group to which the associated training row belongs. Here we have reached to the last, which is the classification and detection of skin cancer image.

3.6 Classification Using 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.

The diagram for the second proposed system to classify and detect the skin cancer image will be displayed in figure (3.7). Then the algorithm (3.7) for NB training will be showed. In this algorithm, the work of the second proposed system and the processes that occurred to detect and classify skin cancer images with two types(with and without applying preprocessing) were presented starting from the first stage, which is pre-processing, as well as splitting dataset to training and testing, that have been chosen for the ratio It is 80 % for training and 20 % for testing.

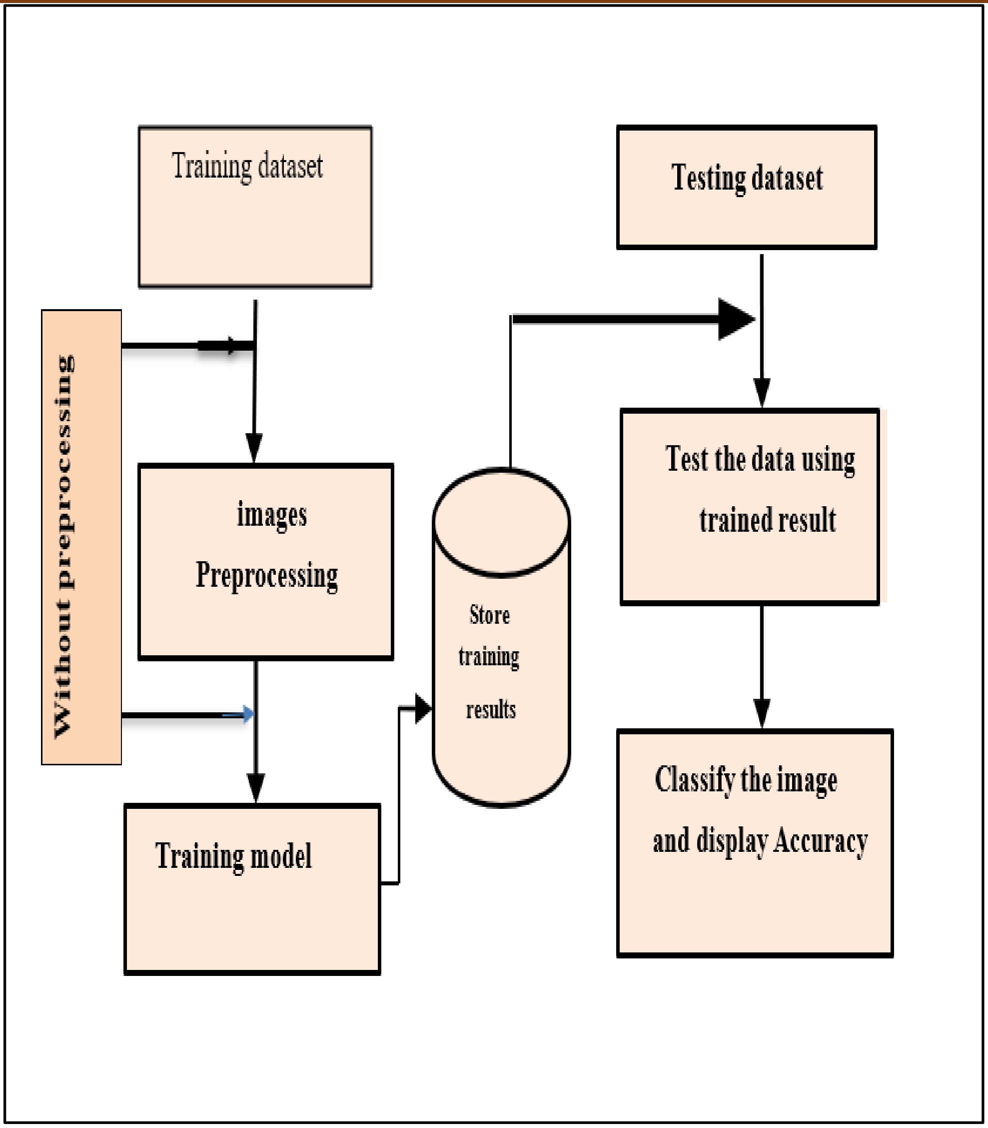


Figure (3.7): Block Diagram of the of Second Model Naïve Bayes

3.6.1 Training step the data is divided into two main sections, which are training data and testing data. In Python the function that use in train naïve byes algorithm called" gnb = GaussianNB ()". The main purpose of the training step is computing prior probability of each class $P(C_i)$ after computed (Total) 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. According to equations (2.19), (2.20), and (2.21) As shown in chapter two in section (2.5).

A-Training Option (training algorithm)

In order for the training process to be completed, that need multiple options that were used in this process using Python program by parameters that are created in the training process these options are:

1- **Priors = None:** that mean Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

2- **var_smoothingfloat, default=1e-9**: Portion of the largest variance of all features that is added to variances for calculation stability.

Figure (3.8) shows the parameter of gaussian naïve bayes during implemented this algorithm on dataset.

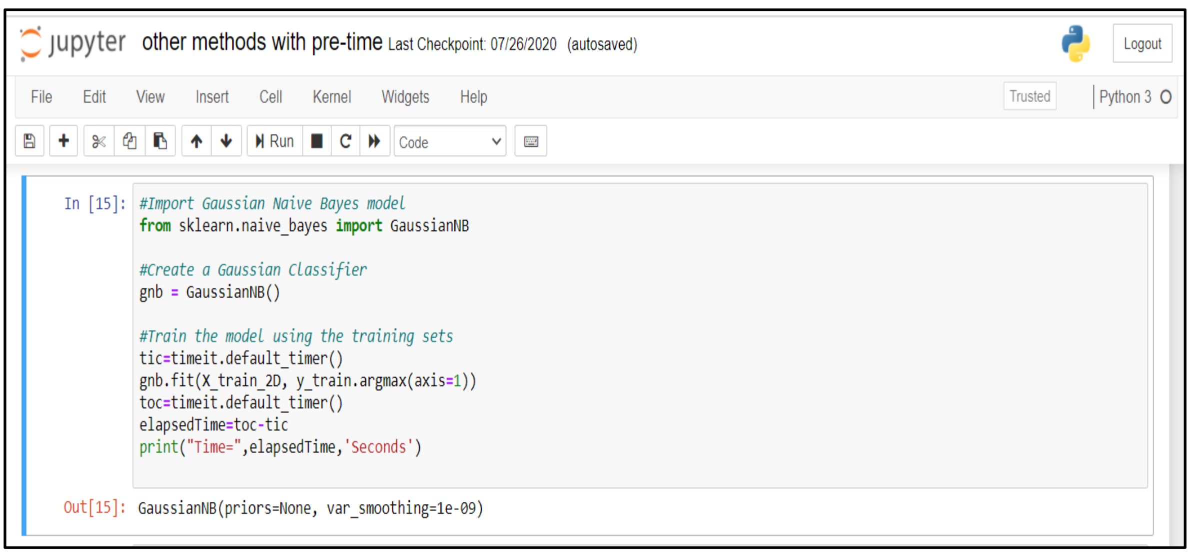


Figure (3.8): Naïve Bayes Parameters during the training process

3.6.2 Testing Phase: The trained NB model was used as the classifier; the classifier produced independent test dataset was categorized to the classifier's test accuracy, the method computes the posterior probability P (C_j| X) for new (x) after applying equation (2.22) as described in section (2.5) in second chapter. The method then classifies the test data according the largest posterior probability.

Algorithm (3.7): Naive Bayes algorithm to classify skin cancer images

Input: Skin cancer image RGB with size 224*224

Output: Class Name

Begin

Step (1): Splitting the data into two parts, 80% for training and 20% for the testing.

Step (2):1. If network process image with hair removal algorithm, go to step (3).
2. Else, go to step (4).

Step (3): Pre-processing stage (hair removal as algorithm (3.3))

Step (4): (Total) = all examples in training dataset.

C_k refer to class in training DS.

Step (5): Calculate the Probability for each class.

P (C_k) = frequency (C_k) / total.

Step (6): Calculate the mean (μ) … …... from equation (2.19)

and standard deviation (σ) from equation (2.20) then store the result.

Step (7): X is tested example in the testing DS.

Step (8): Calculate the probability density function (pdf) of X at C_i for $P(X|C_k)$ by apply the equation (2.21).

Step (9): Calculate conditional probability of(X) at C_k for values product from step (8), by apply equation. $P(X|C_k) = \coprod_{j=1}^{n} P(f_j|C_k)$

Step (10): Calculate posterior probability of(X), $P(C_k|X)$ that denote probability of example at C_k by equation following:

$P(C_k|X) = P(X|C_k)p(C_k)$ --- probability of feature at (C_i).

Step (11): Select class label to the class (X) by choice maximization p $(C_k|X)$.

Step (12): Return Class Name.

END

3.7 Classification Using SVM Algorithm

The classification process is the summary of the work through which the decision is made. The SVM algorithm is chosen, as it one of well-known traditional algorithms of supervised learning of machine learning algorithms. This algorithm was explained in detail with its equations in chapter two, section (2.6). The SVM algorithm as explained previously depends on the establishment of the hyperplane in order for the data to be separated, 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.

The diagram for the third proposed system to detect and classify the skin cancer image will be displayed in figure (3.9). Then the algorithm (3.8) for SVM training will be showed. In this algorithm, the work of the third proposed system and the processes that occurred to detect and classify skin cancer with two types(with and without applying preprocessing) were presented starting from the first stage, which is pre-processing, as well as splitting dataset to training and testing, that have been chosen for the ratio It is 80 % for training and 20 % for testing.

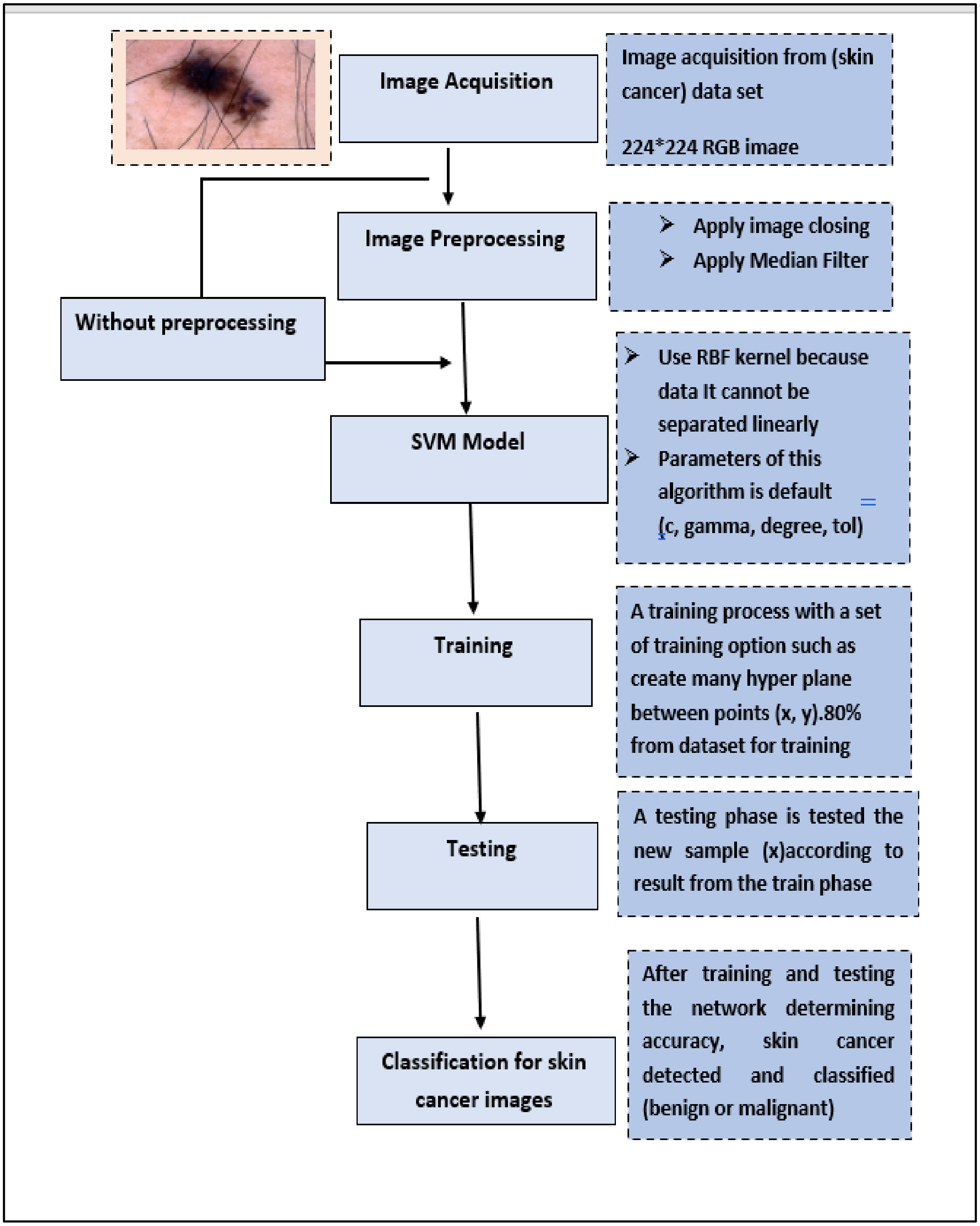


Figure (3.9): Block Diagram of the of Support Vector Machine

3.7.1 Training phase the implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using **sklearn.svm. LinearSVC**, the data is divided into two main sections, which are training data and testing data. In Python the function that use in train SVM algorithm called " svmModel = svm.SVC()". According to equation (2.32) as described in second chapter in section (2.6.2.B).

A-Training Option (training algorithm)

In order for the training process to be completed, that need multiple options that were used in this process using Python program by parameters that are created in the training process these options are:

1- **C float, default=1.0**

Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive**.**

2- **Kernel default='rbf'**

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used.

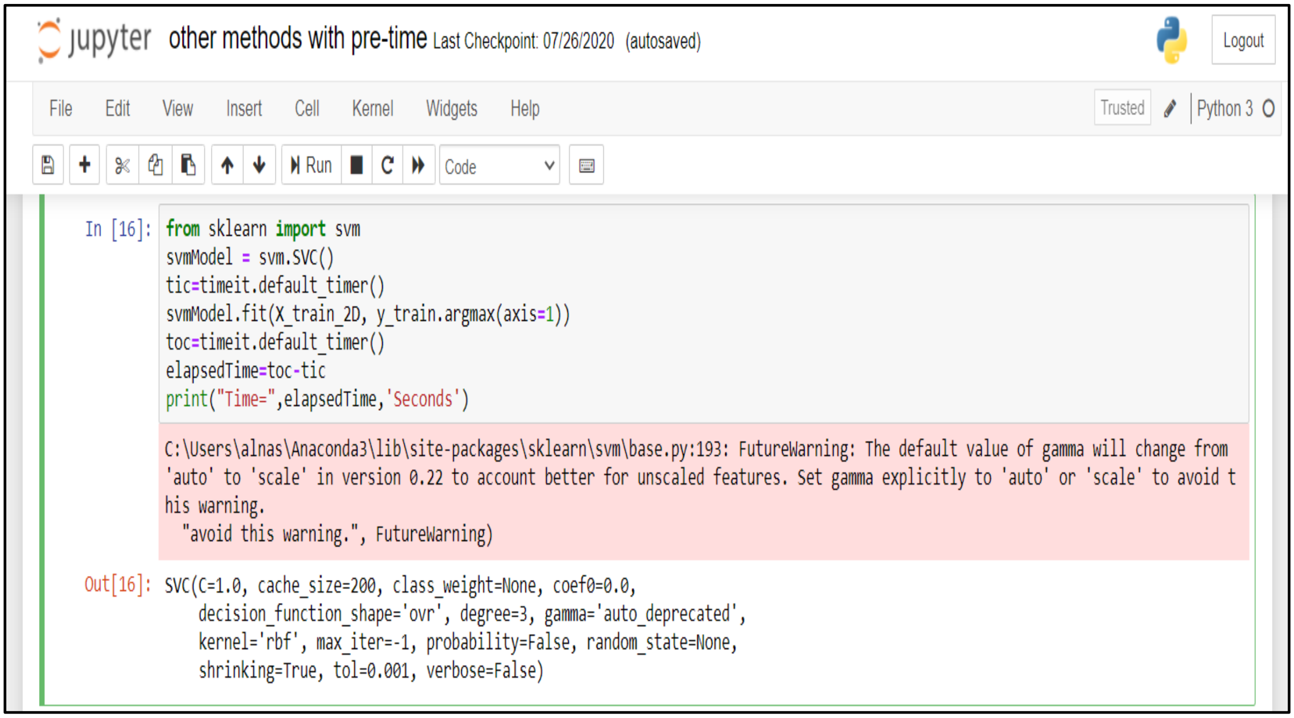
3- **Degree: int, default=3**

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.

4- **Tol: float, default=1e-3**

Tolerance for stopping criterion.

Figure (3.10) shows the parameter of gaussian naïve byes during implemented this algorithm on dataset.




```
In [16]: from sklearn import svm
svmModel = svm.SVC()
tic=timeit.default_timer()
svmModel.fit(X_train_2D, y_train.argmax(axis=1))
toc=timeit.default_timer()
elapsedTime=toc-tic
print("Time=",elapsedTime,'Seconds')

C:\Users\alnas\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from
'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid t
his warning.
  "avoid this warning.", FutureWarning)

Out[16]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

Figure (3.10): SVM Parameters during the training process

3.7.2 Testing Phase: The trained SVM model was used as the classifier; the classifier produced independent test dataset was categorized to the classifier's test accuracy, In order to estimate and evaluate the quality of the model training that has been proposed as a specific group is used randomly for this stage which is the testing stage. The classified system was created using the information provided in the training stage and in the features vector, and determined true class according to largest margin as described in chapter two.

Algorithm (3.8): SVM Algorithm to classify skin cancer images
Input: Skin cancer image RGB with size 224*224 **Output:** Class Name
Begin **Step (1)**: Splitting the data into two parts, 80% for training and 20% for the testing. **Step (2)**:1. If network process image with hair removal algorithm, go to step (3). 2. Else, go to step (4). **Step (3)**: Pre-processing stage (hair removal as algorithm (3.3) **Step (4):** Specify the kernel function ((RBF function), as calculated in the second chapter in the equation (2.32)) **Step (5)**: Applied to map the training set to kernel space by hyperplane computation, its calculation was also defined in the previous chapter in equation (2.29) **Step (6):** Find maximum distance of a point to the hyperplane (margin) between two classes. **Step (7):** Passes testing set on SVM depended on the hyperplane to identify the class name **Step (8):** Return Class Name. END

Chapter Four

Chapter Four

Experimental Results and Discussion

4.1 Introduction

In this chapter, the implementation and experimental results of the proposed system were presented and described. This proposed system detection and classification of skin cancer images using deep learning and machine learning techniques is divided into three different subsystems according to the used algorithms and each system implemented (with and without applying preprocessing algorithm) The results of these three systems will be presented and a comparison will be made between the results of the three systems. Moreover, a comparison with related works will be presented. The same dataset was used in the three systems.

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 $[2 * 2]$, where 2 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	TN	FP
	No	FN	TP

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.

4.4 Dataset Acquisition (Skin Cancer Images)

The datasets that are used in proposed system has 3297. There are 1497 image cases of malignant skin case 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).

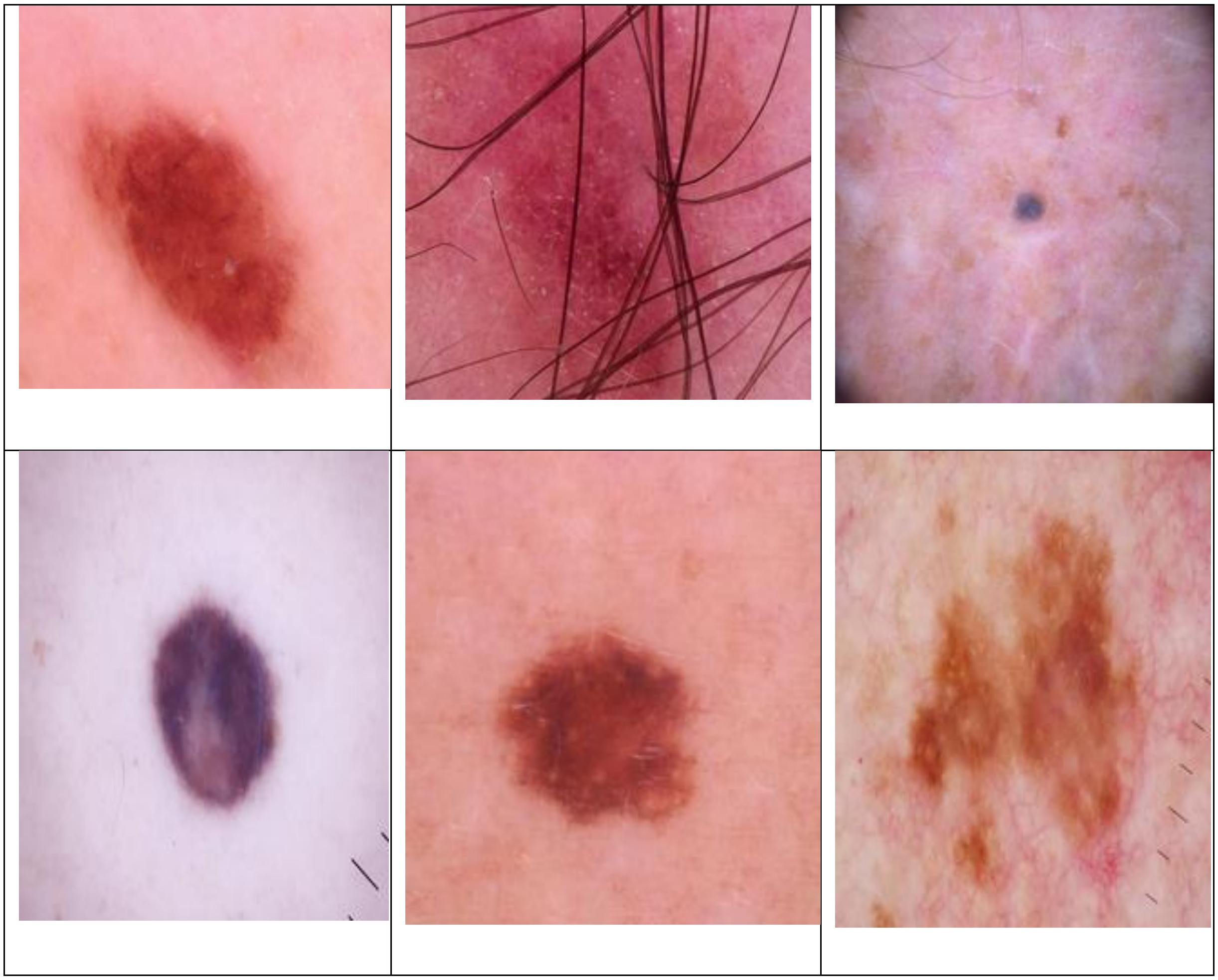


Figure (4.1): Benign Skin Cancer Images [25]

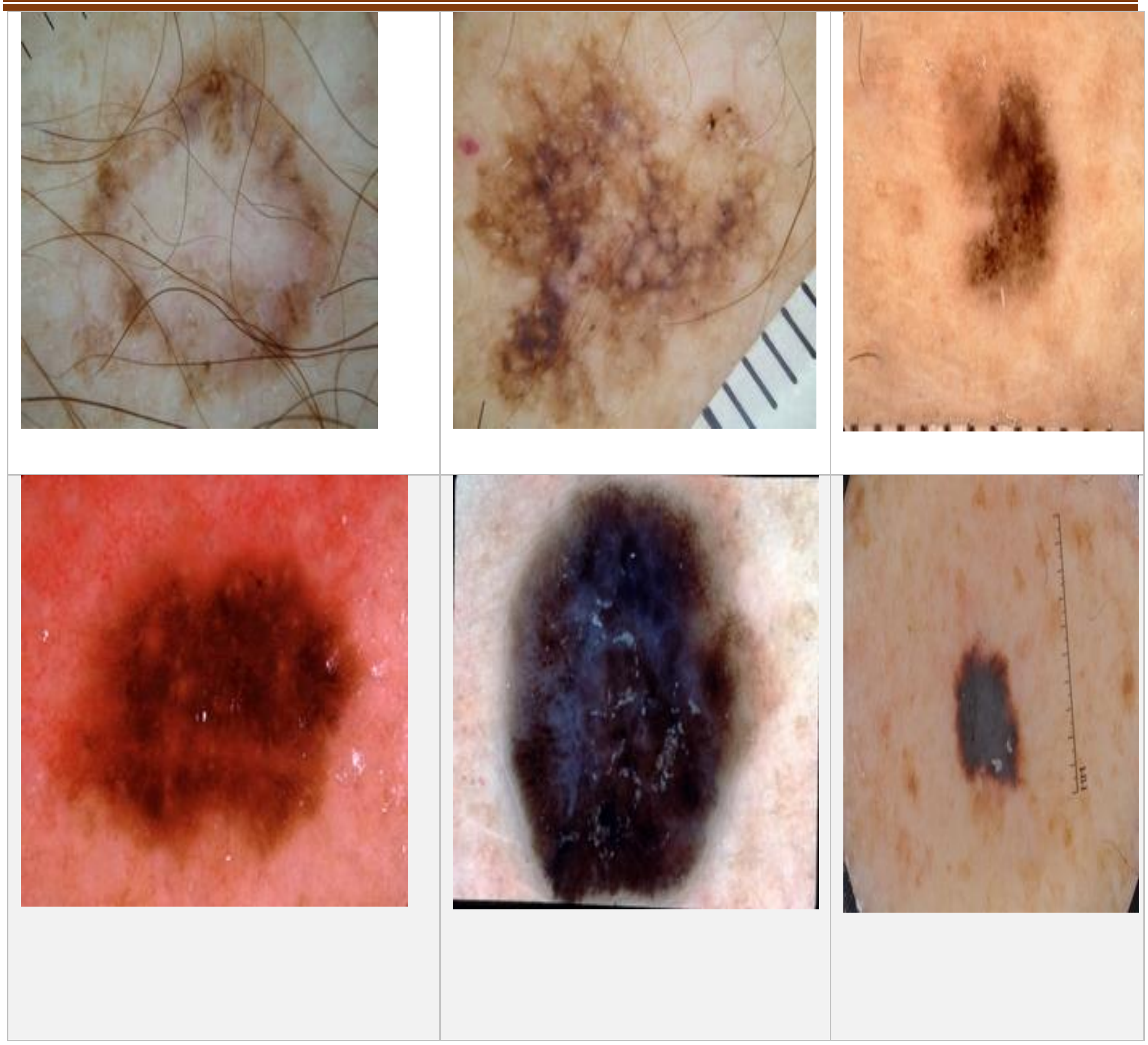


Figure (4.2): Malignant Skin Cancer Images [25]

4.5 Result of Image Pre-processing

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.

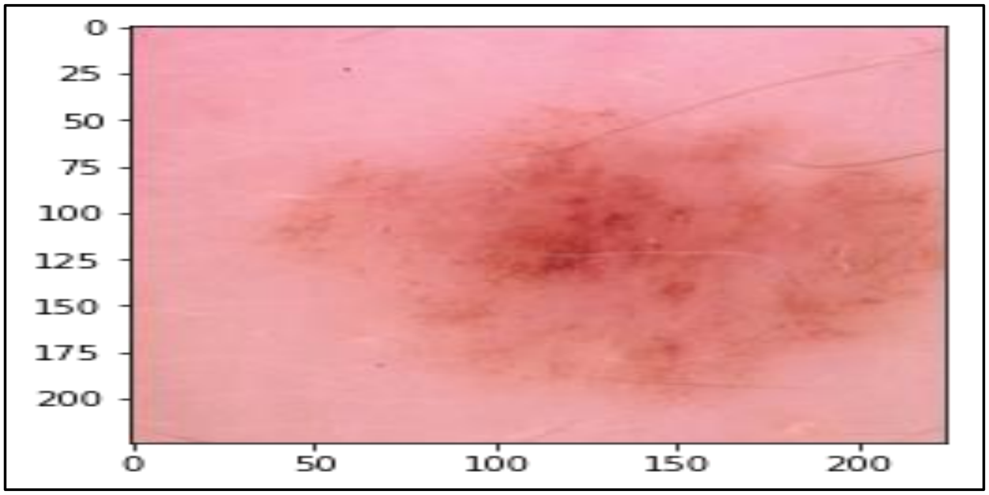


Figure (4.3): The Original Image before Preprocessing

4.5.2 Median Filter Result

A median filter is based on moving a window over an image and computing the median value of the output pixel of the brightness within the input window. After ordered in brightness value from smallest to largest, calculation the median value of a pixel neighborhood the central pixel value then replaced with the median value. As shown in algorithm (3.2). Figure (4.4) show the image after applied the hair removal algorithm.

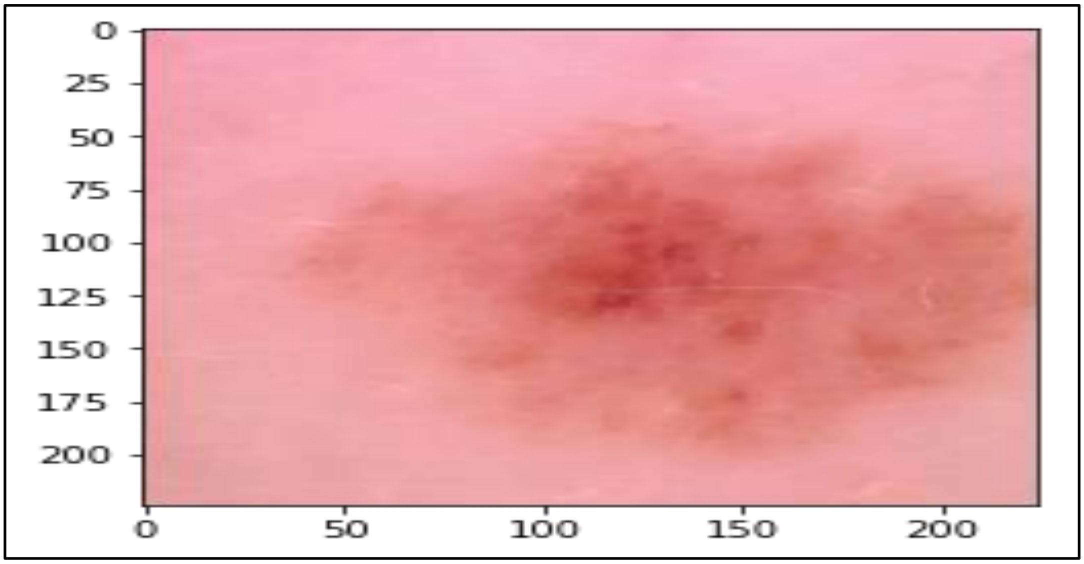


Figure (4.4): Image after Preprocessing

4.6 Evaluation of First Proposed System

The first proposed system that was explained in the third chapter, which consists of several stages, each stage and its results, up to the final results will be shown 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 (2.8).

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

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

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

  No. of parameters=64*(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, where 1024 is equal 2^{10} because propose system is binary classification, 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.

Accuracy, loss, validation accuracy and validation loss that act performance measurement of first model which explained previously in second chapter section (2.4.3.2), and here the results of the training and testing process will be presented. These results are presented in the tables, figures and confusion matrix that shows the results of each class of the two classes of skin cancer (benign or malignant).

4.6.1 Results of First Model without Preprocessing

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.

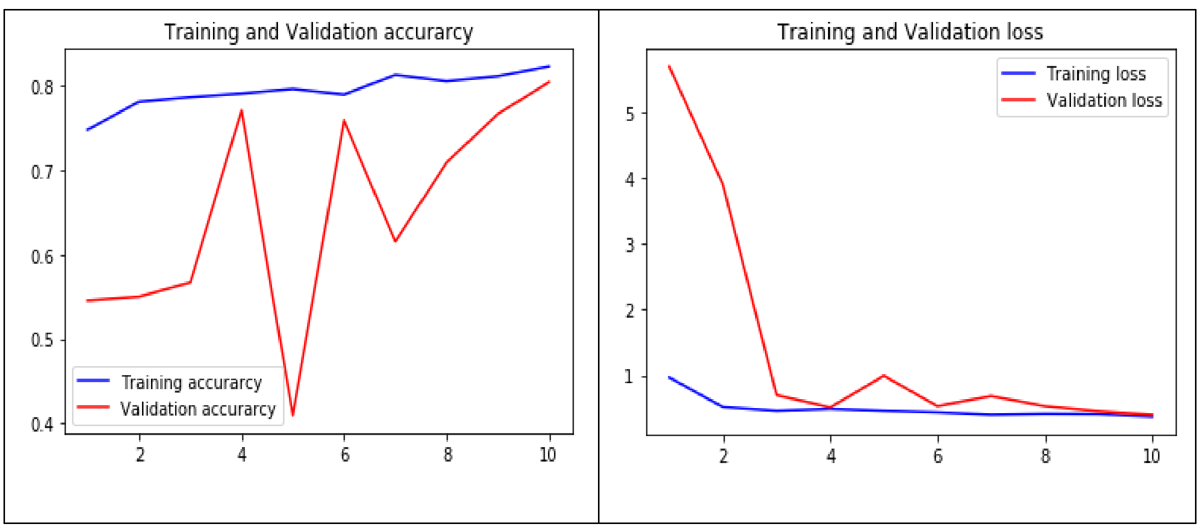


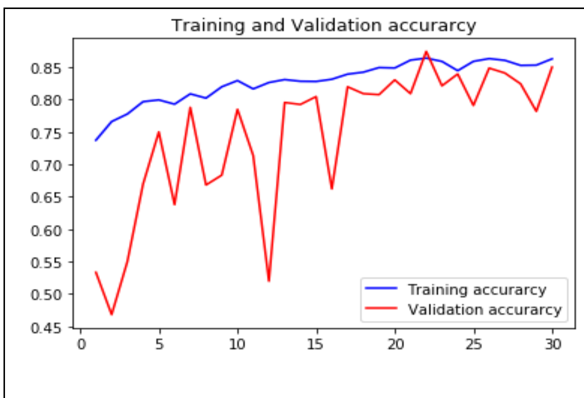
Figure (4.5): Accuracy and Loss Validation Change Against Training Epochs Using CNN Model(10-Epoch) Without Preprocessing.

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

The main problem that occurred during the training is that it takes a long time, and also the speed of the computer and its properties have a big role in the time spent on the training the network. In table (4.4) the column (time) represents the time that the network spends to training batch size of data in each epoch.

Epochs from 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.



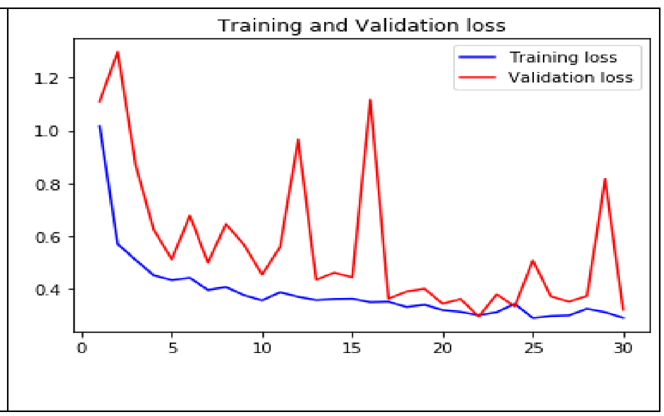


Figure (4.6): Accuracy and Loss Validation Change Against Training Epochs Using CNN Model(30-Epoch) Without Preprocessing.

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

As shown in table (4.5) the validation accuracy and validation loss in vacillation during training 30-epoch, in epoch 22 the validation accuracy is 87.12 and validation loss is 29.94, which is the least loss in first model without preprocessing for 30- epoch.

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

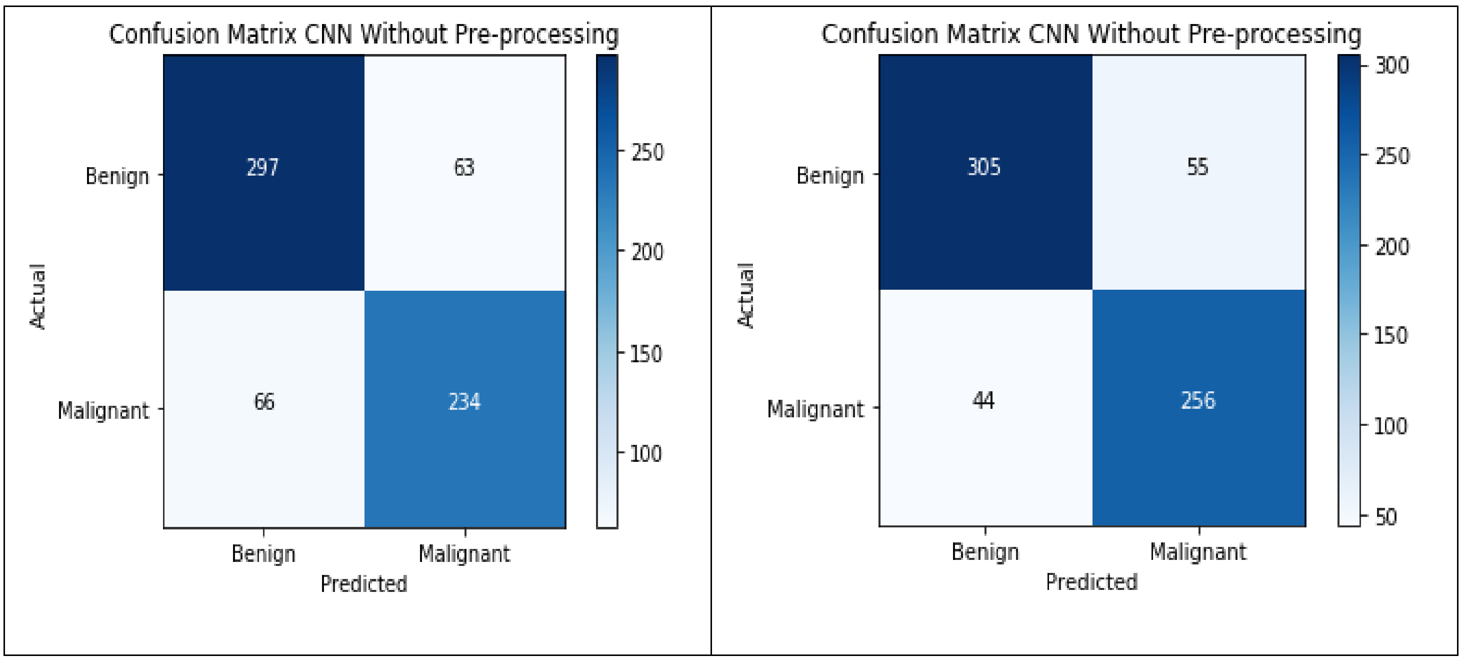


Figure (4.7): Confusion Matrix for the CNN Training without Preprocessing CNN Training. **Left**: In 10- Epoch; **Right**: In 30- Epoch.

According to equations (2.34), (2.35), (2.36), and (2.37) accuracy, sensitivity, specificity and precision computed for first model (without preprocessing).

$$Accuracy\ (\boldsymbol{CNN\ 10\ epoch}) = \frac{TP+TN}{TP+TN+FP+FN} \times 100 = \frac{234+297}{660} = 80.45\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{234}{234+66} = 78\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{297}{297+63} = 82.5\%$$

$$Precision = \frac{TP}{TP+FP} = \quad \frac{234}{234+63} = 78.78\%$$

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

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

$$Sensitivity = \frac{TP}{TP+FN} = \quad \frac{256}{256+44} = 85.53\%$$

$$Specificity = \frac{TN}{TN+FP} = \quad \frac{305}{305+55} = 84.47\%$$

$$Precision = \frac{TP}{TP+FP} = \quad \frac{256}{256+55} = 82.31\%$$

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.

4.6.2 Results of First Model with Preprocessing

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.

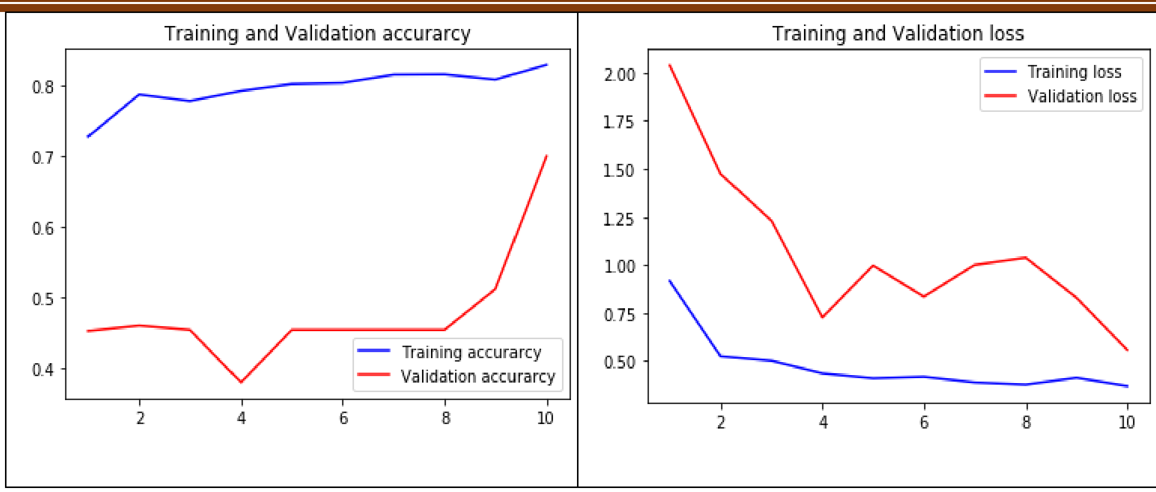


Figure (4.8): Accuracy and Loss Validation Change Against Training Epochs using CNN Model(10-Epoch) With Preprocessing.

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).

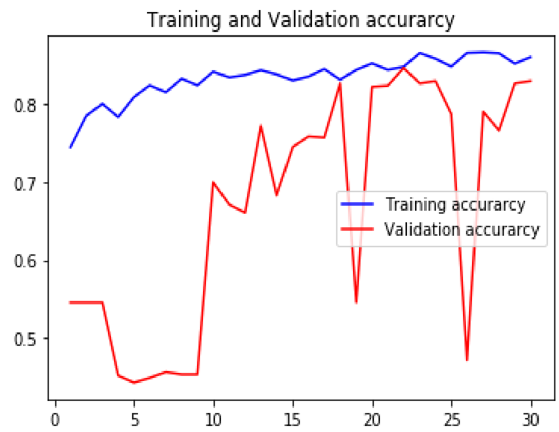


Figure (4.6) The Training and validation accuracy (30-epoch)

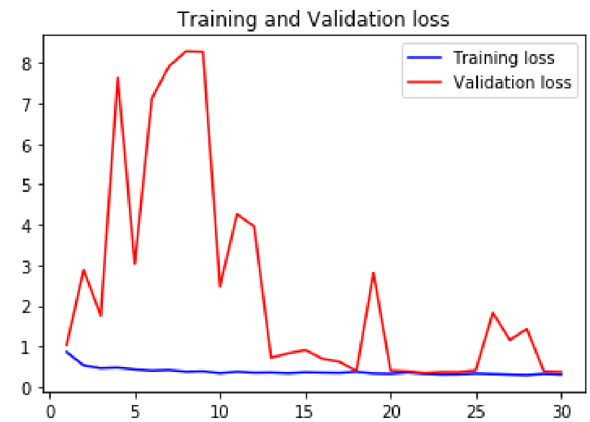


Figure (4.7) The Training and validation loss(30- epoch)

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

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.

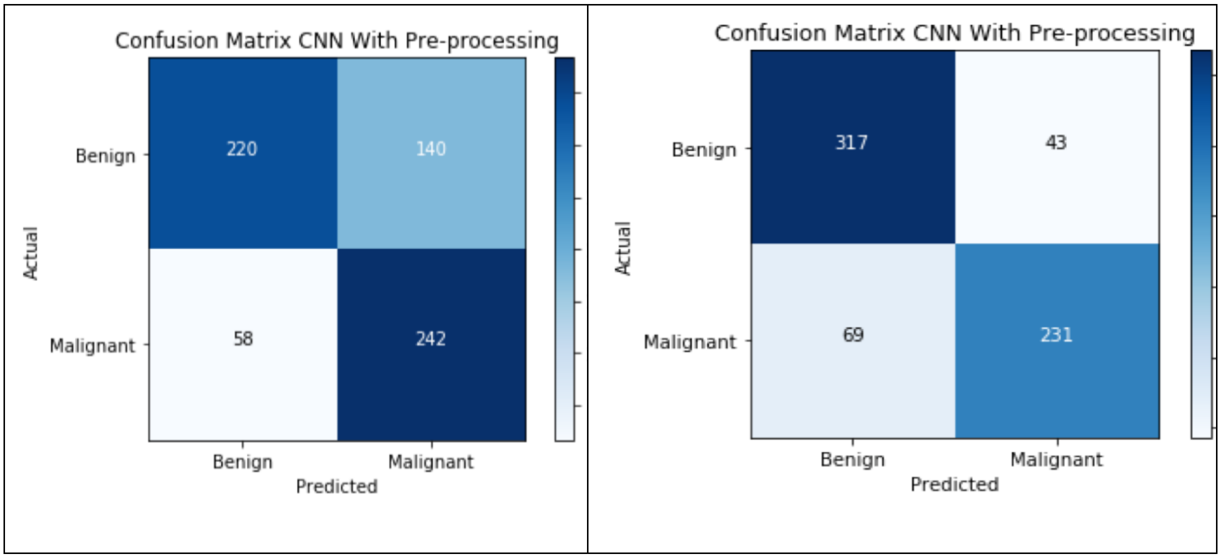


Figure (4.10): Confusion Matrix for CNN Training with Preprocessing
CNN Training: **Left**: in 10- Epoch; **Right**: in 30- Epoch.

According to equations (2.34), (2.35), (2.36), and (2.37) accuracy, sensitivity, specificity and precision computed for first model (with preprocessing).

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

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

$$Sensitivity = \frac{TP}{TP+FN} = \frac{242}{242+58} = 80.6\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{220}{220+140} = 61.1\%.$$

$$Precision = \frac{TP}{TP+FP} = \frac{242}{242+140} = 63.3\%.$$

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

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

$$Sensitivity = \frac{TP}{TP+FN} = \frac{231}{231+69} = 77\ \%.$$

$$Specificity = \frac{TN}{TN+FP} = \frac{317}{317+43} = 88.05\%.$$

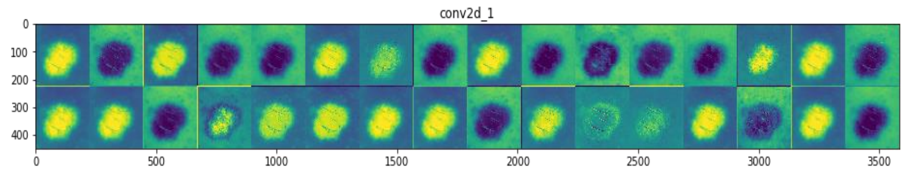
$$Precision = \frac{TP}{TP+FP} = \frac{231}{231+43} = 84.3\%.$$

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.

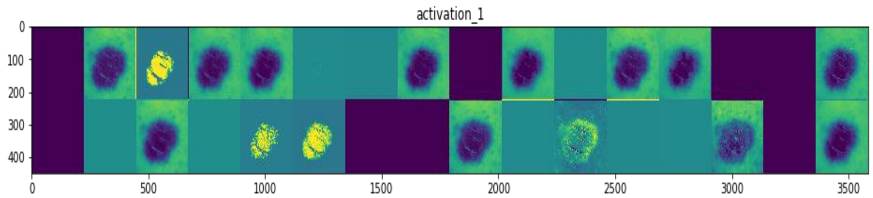
Table (4.8) Difference Between CNN Algorithm with and without Processing

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%	4 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

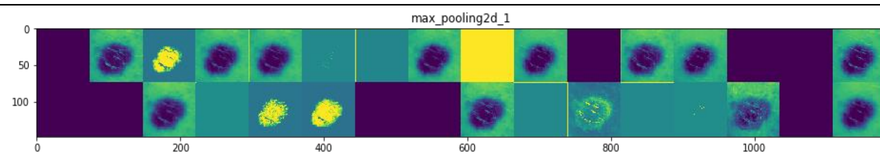
In (CNN with preprocessing) the main layers output show in figure (4.11) from the first layer (convolution layers) to the fifth with its activation, pooling and dropout parts.



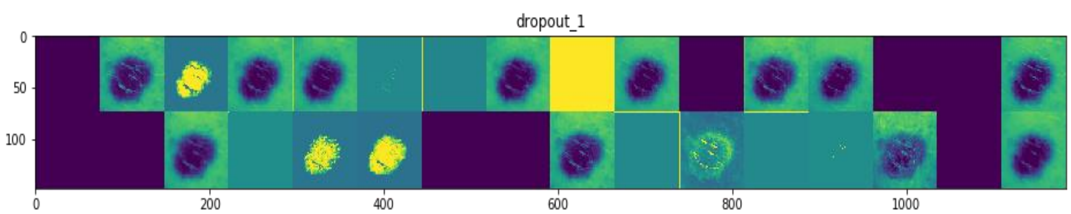
A-Output of First Conv2D Layer



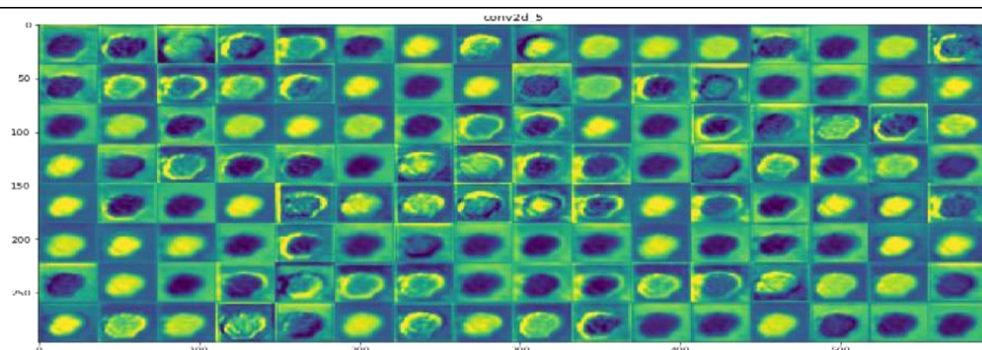
B- Output of Activation Layer (with 32-filters)



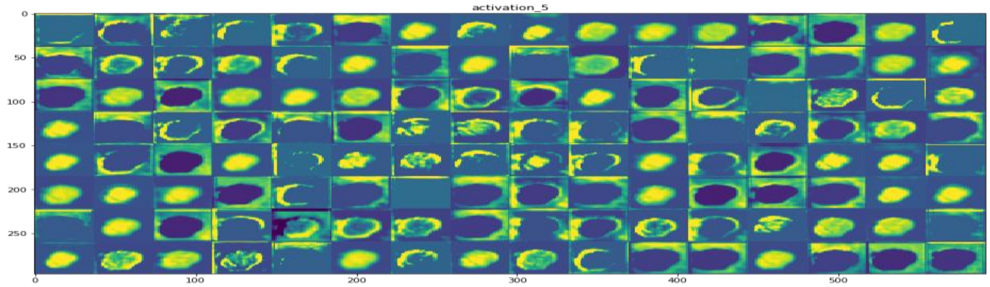
C- Output of Pooling Layer (with 32-filters)



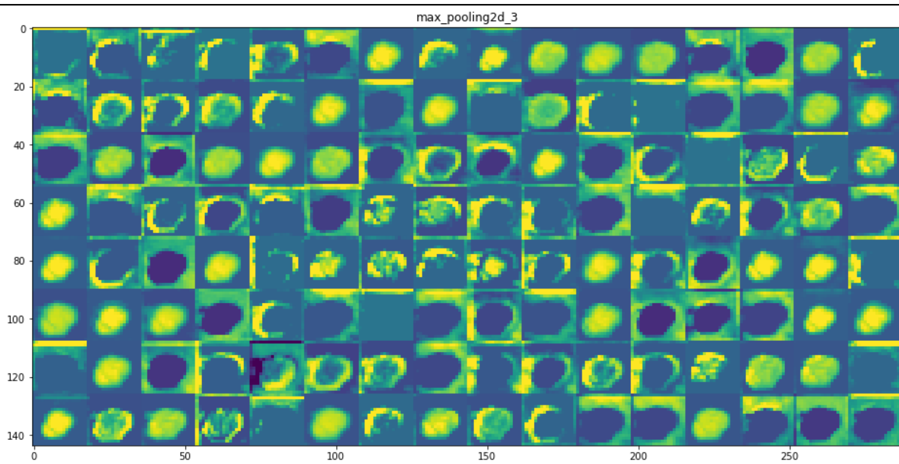
D- Output of Dropout Layer (with 32-filters)



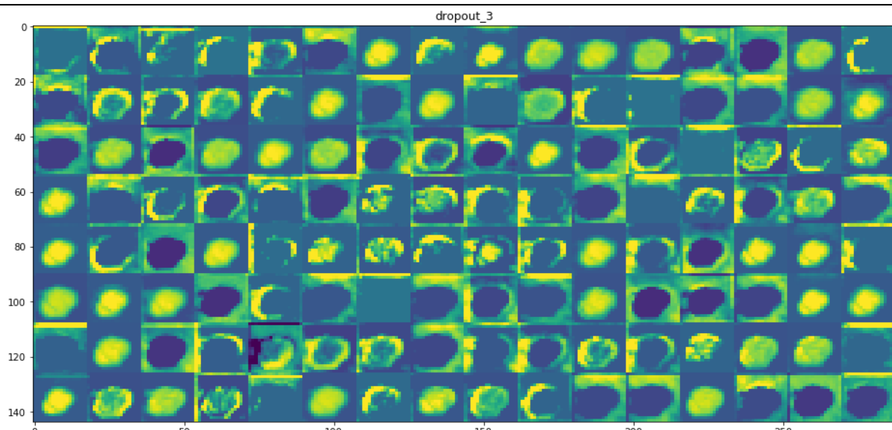
E- Output of last Conv 2D Layer (with 128-filters)



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.

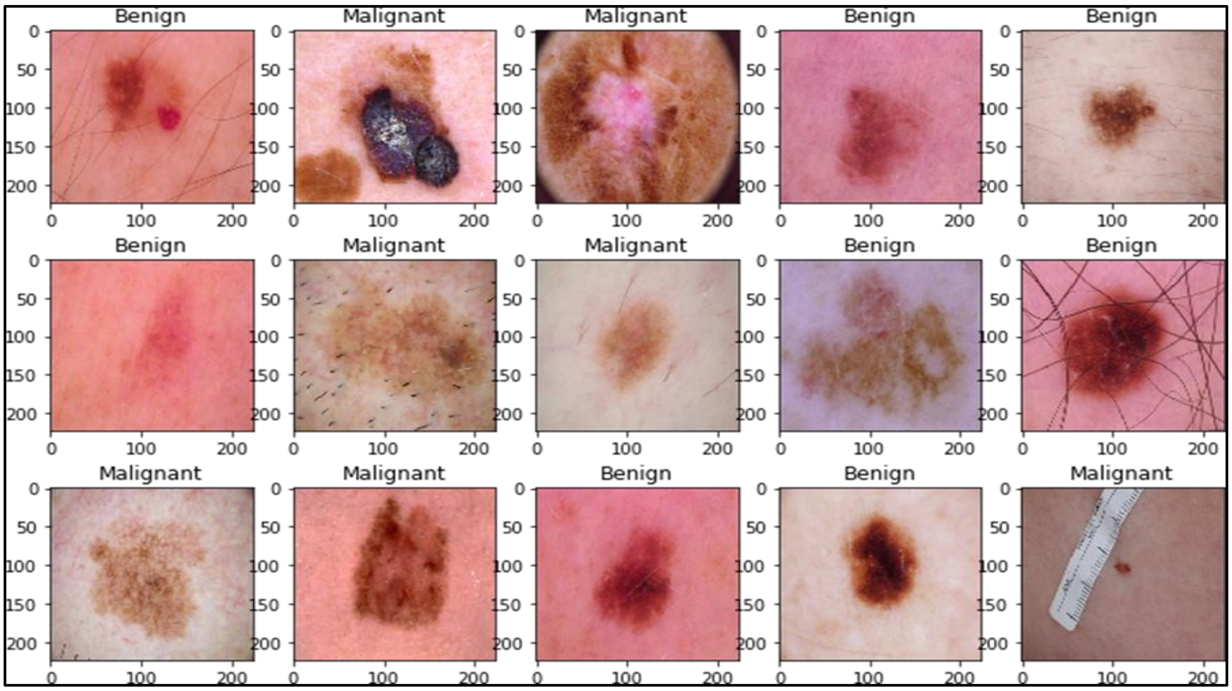


Figure (4.12): Skin Cancer Images with Labels

4.7 Results of Second Proposed Model (NB)

In this stage, the images of skin cancer are classified according to the NB algorithm was previously explained in the algorithm (3.7). The system needs two stages that have been explained previously, which are the training stage and the testing stage. The results of these two stages will be displayed in order to then detect and classify the image of skin cancer. The confusion matrix is used to display the accuracy of NB model.

4.7.1 NB Training

This operation was explained previously in section (3.6.1) where in the training process for images is done in a Naïve Bayes using GaussianNB. During the training phase computed the mean (μ) from equation (2.19), and standard deviation (σ) from equation (2.20) The performance measurement was calculated according to that was mentioned in the second chapter in section (2.7) and according to the equations (2.34), (2.35), (2.36) and (2.37).

4.7.2 NB Testing

This stage is an examination of the system by testing it with the remainder of the data that is not labeled in order to classify the images of skin cancer as explained in the previous chapter. At this stage also the results were presented in the form of a confusion matrix that shows the accuracy of each class in the testing stage for two case (model with applying preprocessing and without preprocessing).The display of the results using the confusion matrix as in figure (4.13) and table (4.9), which shows the accuracy of the testing results for model without preprocessing.

4.7.3 Result of Second Model without Applying Preprocessing:

Figure (4.13) shows description of the performance of the classification model (or "Classifier") on a set of test data. Where the confusion matrix itself is relatively easy to understand.

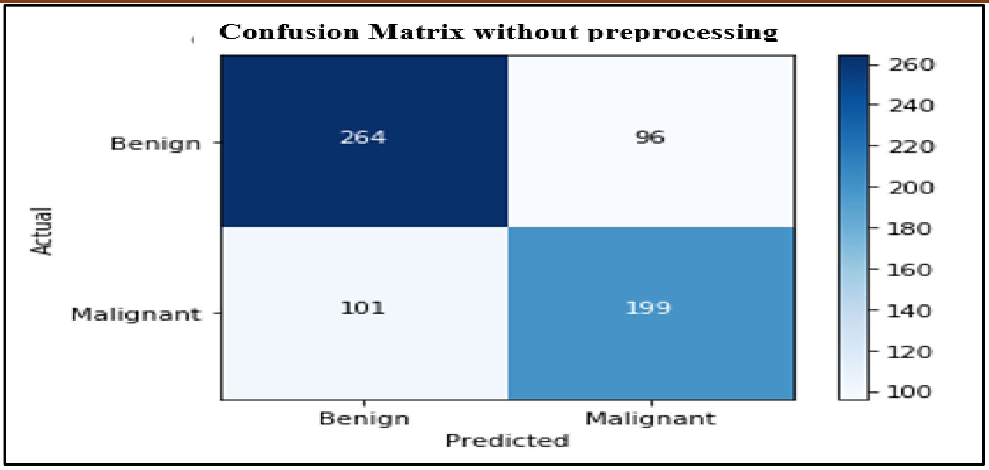


Figure (4.13) Confusion Matrix for Naïve Bayes without Preprocessing

For measuring Accuracy, Sensitivity, Precision, and Specificity, according of equations (2.34), (2.35), (2.36) and (2.37).

$$Accuracy = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100 = \frac{199+264}{660} = 70.15\%$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{199}{199+101} = 66.33\%$$

$$Specificity = \frac{TN}{TN+FP} = \frac{264}{264+96} = 73.33\%$$

$$Precision = \frac{TP}{TP+FP} = \frac{199}{199+96} = 67.45\%$$

Table (4.9) Naïve Bayes Accuracy Without Preprocessing

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

Figure (4.13) and Table (4.9) illustrate the classification rate of second 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, 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.

4.7.4 Result of NB Model with Applying Preprocessing:

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 (4.14) and table (4.10). The classification accuracy of the second proposed model with preprocessing and is 69.69%, and the total time is 2 minutes.

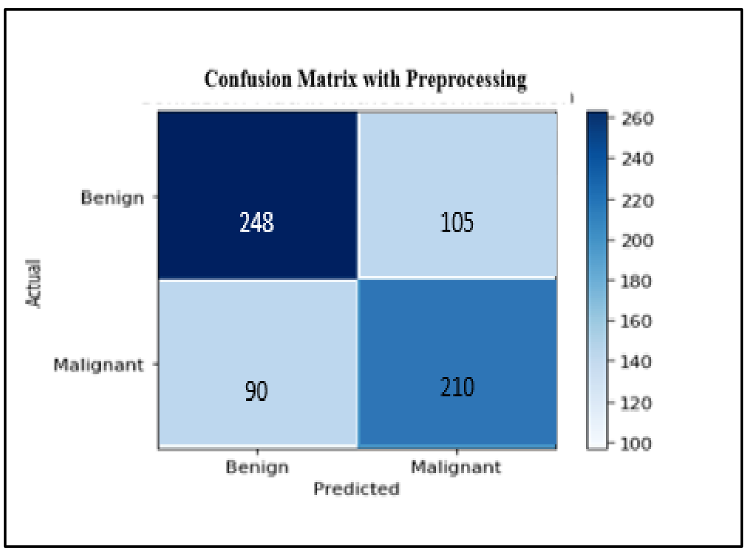


Figure (4.14) The Confusion Matrix for Naïve Bayes with Preprocessing

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 for two cases (with or without preprocessing) as the preprocessing did not affect the improvement of classification accuracy. 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 Two cases Naïve Bayes Model

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

Figure (4.15) shows a scatter plot which is type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis which is denoted to the predicted class (y) and the value of the other variable determining the position on the vertical axis which refers to value of (x). Proposed model includes two classes of benign and malignant skin cancer. In figure (4.16), points that have the same color indicate that they have the same class.

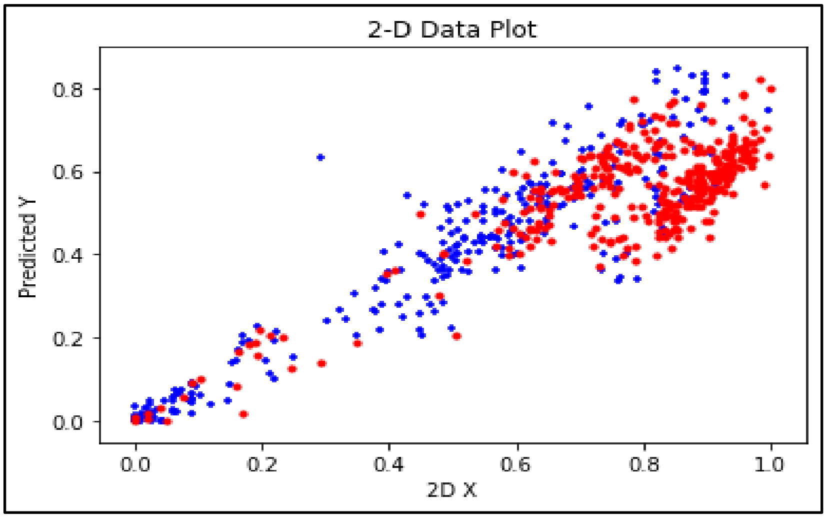


Figure (4.15): Scatter Plot for Dataset

4.8 Results of Third Proposed Model (SVM)

In this stage, the images of skin cancer are classified according to the SVM algorithm was previously explained in the algorithm (3.8). The system needs two stages that have been explained previously, which are the training stage and the testing stage. The results of these two stages will be displayed in order to then detect and classify the image of skin cancer. The confusion matrix is used to display the accuracy of model for each case (model with applying preprocessing and without preprocessing).

4.8.1 SVM Training

This operation was explained previously in section (3.7.1) where in the training process for images is done in a nonlinear (SVM) using RBF. During the training phase computed the (w) values that represent the hyperplane between the classes, according on the equation (2.29). The performance measurement was calculated according to that was mentioned in the second chapter in section (2.7) and according to the equations (2.34), (2.35), (2.36) and (2.37).

4.8.2 SVM Testing

This stage is an examination of the system by testing it with the remainder of the data that is not labeled in order to classify the images of skin cancer as explained in the previous chapter. At this stage also the results were presented in the form of a confusion matrix that shows the accuracy of each class in the testing stage for two case (model with applying preprocessing and without preprocessing).The display of the results using the confusion matrix as in figure (4.16) and table (4.12), which shows the accuracy of the testing results for model without preprocessing,

4.8.3 Result of SVM without Applying Preprocessing

Figure (4.16) and table (4.12) indicate the accuracy of the classification of this model when classification of images without preprocessing.

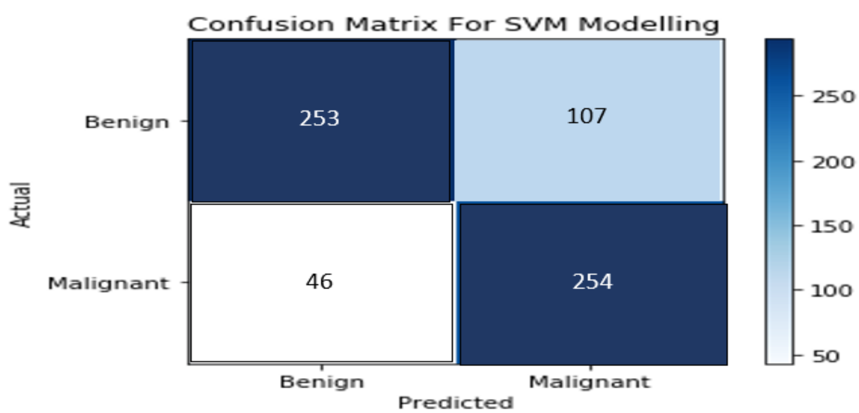


Figure (4.16) The Confusion Matrix for SVM without Preprocessing

By the indicators, accuracy, sensitivity, specificity that are shown in figure (4.16) and table (4.12) where the sensitivity 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 (2.34) $Accuracy = \frac{TP+TN}{Total\ no.of\ test\ sample} \times 100$

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

Equation (2.35) $Sensitivity = \frac{TP}{TP+FN} = \frac{253}{253+46} = 84.66\%,$

Equation (2.36) $Specificity = \frac{TN}{TN+FP} = \frac{254}{254+64} = 70.27\%,$

and equation (2.37) $Precision = \frac{TP}{TP+FP} = \frac{253}{253+46} = 70.36\%.$

Table (4.12) SVM Accuracy without Preprocessing

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

4.8.4 Result of SVM with Applying 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 (4.17) and table (4.13).

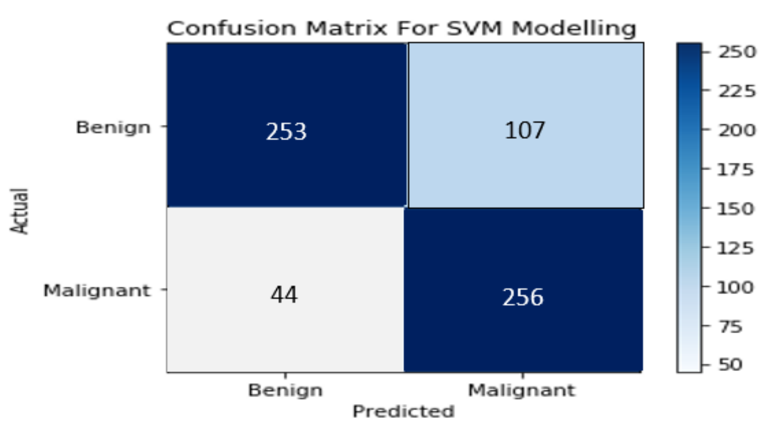


Figure (4.17) The Confusion Matrix for SVM with Preprocessing

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 cases for SVM model is shown in table (4.14).

Table (4.14) Difference Between Two Cases SVM Model

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

Figure (4.18) 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.

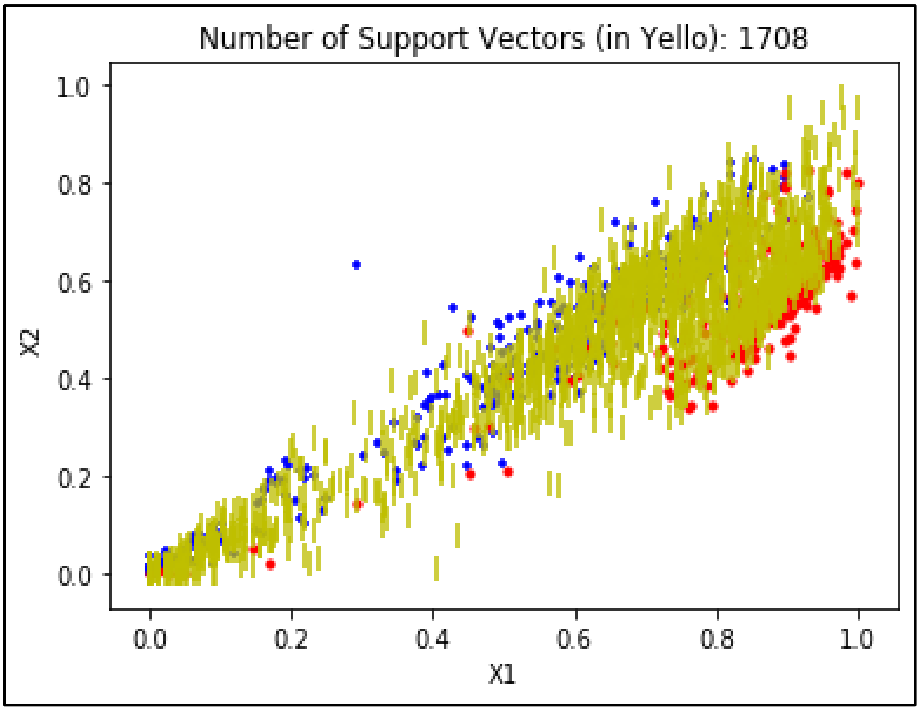


Figure (4.18): Number of Support Vectors

4.9 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 (4.19) 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 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 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

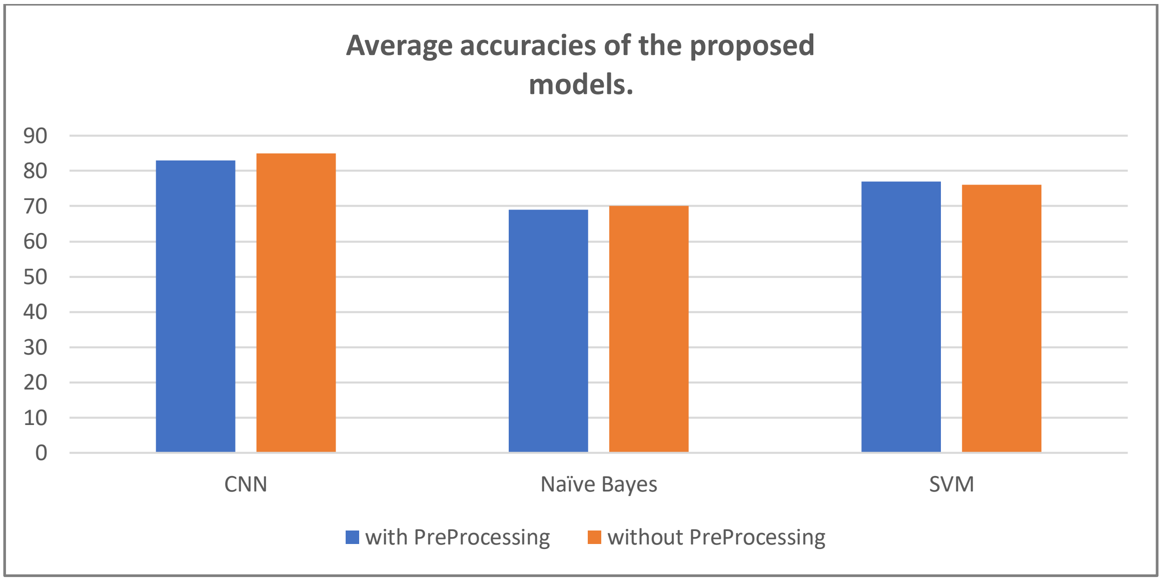


Figure (4.19): Illustration of Average Accuracies of Proposed Models.

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.

4.10 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 **[17][16].**

Table (4.17) Comparison of Classification Accuracy with Earlier Studies

Approach	Accuracy	Methods
Md Ashraful [17] 2018	76.00 %	(CNN) PNASNet-5-Large model
Adria & Jack [16] 2017	66.00%,	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

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 converge 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] 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.

[12] 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.

[13] 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.

[14] 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

[15] 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.

[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] 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.

[18] 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.

[19] 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.

[20] 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.

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

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

[**26**] 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.

[**27**] 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.

[28] 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.

[**29**] 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.

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

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

[**32**] 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.

[**33**] Chalmers, D. J. book "**The conscious mind: In search of a fundamental theory".** Oxford university press. (1996).

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

[35] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. **"Gradient-based learning applied to document recognition."** Proceedings of the IEEE 86, No. 11, PP 2278-2324, 1998.

[36] Guo, Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew, "Deep learning for visual understanding: A review.", Neurocomputing 187, PP 27-48, 2016.

[37] 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.

[**38**] Anna Gummeson “**Prostate Cancer Classification using Convolutional Neural Networks**” M.S.C thesis Lund Institute of Technology, 2016.

[**39**] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, “**ImageNet classification with deep convolutional neural networks**”, In Advances in neural information processing systems, PP 1097-1105, 2012.

[**40**] Zeiler, Matthew D, “Hierarchical convolutional deep learning in computer vision”, Ph.D Thesis, New York University, 2013.

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

[**42**] Yamashita, R., Mizuho N., Richard K. Gian Do, and Kaori T., “**Convolutional neural networks: an overview and application in radiology**”, Insights into imaging 9, PP 611-629, 2018.

[**43**] Iftene, M., Q. Liu, and Y. Wang. “**Very high-resolution images classification by fine tuning deep convolutional neural networks**” In Eighth International Conference on Digital Image Processing (ICDIP 2016), International Society for Optics and Photonics, vol.10033, PP 100332D. 2016.

[**44**] Demuth, Howard B., and Mark H. Beale., Orlando De Jess, and Martin T Hagan. “**Neural Network Design**”, Martin Hagan, 2014.

[**45**] LeCun, Y., Yoshua B., and Geoffrey H., “**Deep learning**”, Nature 521, No. 7553, PP 436-444, 2015.

[**46**] Ioffe, S., and Christian S. “**Batch normalization: Accelerating deep network training by reducing internal covariate shift**.”, *arXiv preprint arXiv:1502.03167*, 2015.

[**47**] 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.

[**48**] 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.

[**49**] 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.

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

[**51**] Bottou, L., and Olivier B**.**, "**The tradeoffs of large-scale learning**", In Advances in neural information processing systems, PP 161-168, 2008.

[**52**] Lin M, Chen Q, Yan S. "**Network in network**". arXiv preprint arXiv:1312.4400 2013.

[**53**] T. Hastie, R. Tibshirani, J. Friedman, "**The Elements of Statistical Learning**", 2nd edition, Springer, NewYork/Berlin/Heidelberg, 2008.

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

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

[**56**] 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.

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

[**58**] Jamal. M. AL-Tuwaijari and S. I. Mohammed, "**Face Image Recognition Based on Linear Discernment Analysis and Cuckoo Search Optimization with SVM**", International Journal of Computer Science and Information Security. ISSN 1947 5500, IJCSIS, Vol. 15, No. 11, November 2017.

[**59**] Kolluru, P., Kumar, "**SVM Based Dimensionality Reduction and Classification of Hyperspectral Data**", University of Twente Faculty of Geo-Information and Earth Observation (ITC), 2013.

[**60**] Ghaidaa W., and Jamal M. Al-Tuwaijari. "**Satellite Images Scene Classification Based Support Vector Machines and K-Nearest Neighbor**." *Diyala Journal for Pure Science* 15, No. 03, PP. 70-87, 2019.

[**61**] Ramon H., Fernando C., and Charles E., "**Nonlinear support vector machines can systematically identify stocks with high and low future returns**", Algorithmic Finance 2, PP 45-58, 2013

[**62**] Achirul N., Muhammad, K., Boro S., Dodi N., and Akhiruddin M., "**A comparison study of kernel functions in the support vector machine and its application for termite detection**", Information 9, No. 1, P 5, 2018.

[**63**] Soentpiet, Rosanna, "**Advances in kernel methods: support vector learning**", MIT press, 1999.

[**64**] Reem Majeed Ibrahim, "**Classification of Face Image Based on Gender Using Intelligent Method**", M.s.C electronic thesis, Computer Sciences, University of Technology, 2015.

الخلاصة

سرطان الجلد هو نموغير طبيعي في خلايا الجلد ناتج عن طفرات في الخلايا حمض (DNA). حيث تحدث معظم الوفيات الناجمة عن سرطان الجلد بسبب النوع الخبيث منه. لذلك يعتبر أحد أنواع السرطان الذي يمكن علاجه بالكشف المبكرعن المرض عن طريق فحص الخزعة ، لذلك 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 الاكتشاف التلقائي للصورة وتصنيفها مهمًا جدًا لأورام الجلد ومهمة صعبة للغاية بالنسبة للصور الطبية. تقدم هذه الرسالة نظام مقترح لتصنيف سرطان الجلد بعد اكتشافه بمساعدة آليات التعلم العميق وخوارزميات التعلم الآلي ، حيث يتم استخدام عدة خطوات على شكل مراحل والتي تشمل مرحلة الحصول على الصورة والمعالجة المسبقة للصورة ، ومرحلة التصنيف. تم الحصول على مجموعة البيانات المستخدمة من أرشيف ISIC (التعاون الدولي لصور البشرة) ، وهي تحتوي على ٣٢٩٧ صورة. يوجد ١٤٩٧ حالة مصورة من نوع كاشف الجلد الخبيث و ١٨٠٠ حالة صور حميدة. في مرحلة ما قبل المعالجة ، تستخدم خوارزمية إزالة الشعر. يعتمد النموذج الأول المقترح على مصنف الشبكات العصبية التلافيفية (CNN). يستخدم النموذج الثاني المقترح مصنف Naïve Bayes (NB). بينما يعتمد النموذج الثالث المقترح على مصنف Support Vector Machine (SVM). وكل نموذج مع تطبيق خوارزمية المعالجة المسبقة وبدون تطبيق. بينت النتائج أن النموذج الأول المقترح باستخدام (CNN) بدون معالجة مسبقة كان متوسط الدقة ٨٥,٠٠٪ ، بينما كانت دقة المعالجة المسبقة ٦٩,٩٩٪. النموذج الثاني المقترح باستخدام (NB) بدون معالجة مسبقة كان متوسط الدقة ٧٠,١٥٪ بينما كانت دقة المعالجة المسبقة ٦٩,٦٩٪. أما النموذج الثالث المقترح باستخدام (SVM) بدون معالجة مسبقة فقد حقق دقة تصل إلى ٧٦,٨١٪ ، بينما كانت دقة المعالجة المسبقة ٧٧,١٢٪.

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

طريقة التعلم العميق للتصنيف عن مرض سرطان الجلد

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

عهود فاضل علوان

باشراف

أ.د. ظاهر عبدالهادي عبدالله

٢٠٢٠ م