MOHAMED JUMA ALI MOHAMED

ACUTE MYELOID LEUKEMIA MALIGNANCY IDENTIFICATION USING NEURAL NETWORKS

NEU 2018

ACUTE MYELOID LEUKEMIA MALIGNANCY IDENTIFICATION USING NEURAL NETWORKS

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
MOHAMED JUMA ALI MOHAMED

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronic Engineering

NICOSIA, 2018

ACUTE MYELOID LEUKEMIA MALIGNANCY IDENTIFICATION USING NEURAL NETWORKS

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
MOHAMED JUMA ALI MOHAMED

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronic Engineering

NICOSIA, 2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name:

Signature:

Data:

ACKNOWLEDGMENT

All praise and thanks to God the exalted, the Lord of the universe, who guided me alone and made all things easy and possible to be realized by His majesty.

I wish to express my sincere thanks and gratitude to my supervisor Assoc. Prof. Dr. Kamil Dimililer. For his guidance, continuous supervision and support throughout the research and the rest of the course of my study at Near East University. His enthusiasm and interest in the work have been very encouraging. I would never forget his attitude and efforts to overcome the obstacles of the research. This thesis would not have been possible without his guidance and help.

Special thanks go also to the Libya Ministry of Education for their trust and continuous financial support during the years of my graduate study at Master level.

A great thank go to my beloved parents, my first tutors in life, siblings for their continued support and prayers, and nephews and nieces for their love.

Gratitude is expressed to my supportive wife, Um Ali, and my wonderful children Ali, Warda, and Razan, who have made this hard time much more enjoyable. My wife's encouragement, patience, and helpfulness have motivated me to achieve my ambitions.

Last but not least, may Allah bless us with his love and make this work purely for his sake and bless all we do.

To my family and parents.....

ABSTRACT

Acute Myeloid leukemia is one type of cancer in which the body creates extensive quantities of strange cells regularly white blood cells (WBC). Artificial neural networks have been used in medical imaging and medical image classification. These intelligent models have shown a great efficacy in this field, although in some areas, neural networks couldn't perform well due to some data fuzziness. Acute Leukemia blood cells are microscopic images that contain different types of cells in addition to the cancerous cells, and this is what makes it a tedious task to classify the cancerous blood cells images from those which have no cancer. Hence, in this thesis a feedforward neural is designed and trained with backpropagation adaptive learning algorithm to be then used a an automatic system for the classification of blood cells into normal or Acute myeloid leukemia. Experimentally, the system uses some image processing techniques in order to enhance the quality of blood cell images before being fed into network, which may facilitates the learning process. Moreover, the patterns of cells in images are also extracted using edge detection techniques: Canny, Sobel, and Prewitt. A comparison between the network models which use different edge detection algorithm is carried out in this thesis. Overall, it is found that the Canny operator is the best in extracting the real patterns of the cancerous cells in a microscopic blood cell image. This is found by the highest accuracy the network achieved when the Canny edge detector is used.

Keywords: Neural network; Backpropagation; Acute Myeloid leukemia; Canny; Sobel; Prewitt

ÖZET

Akut Miyeloid Lösemi, vücudun aşırı miktarda beyaz kan hücresi ürettiği bir kanser türüdür. Yapay sinir ağları tıbbi görüntüleme ve tıbbi görüntüleme klasifikasyonunda kullanılmaktadır. Bu akıllı modeller bu alanda üstün etkinlik gösterse de, bazı alanlarda, sinir ağları veri belirsizliğinden dolayı iyi bir performans sergileyememiştir. Akut lösemili kan hücreleri kanserli hücrelerin yanı sıra farklı hücreler içeren mikroskobik görüntülerdir. Bu da, kanserli hücreleri kansersiz hücrelerden ayırt etmeyi zorlaştırmaktadır. Bu tezde, sonraki aşamada kan hücrelerinin normal veya Akut Miyeloid Lösemi şeklinde sınıflandırılması için otomatik bir sistem olarak kullanılacak geri yayılımlı uyarlanır öğrenme algoritmalı bir beslemeli sinir ağı tasarlanmış ve üzerinde çalışılmıştır. Deneysel olarak, sistem, öğrenim sürecini kolaylaştırabilecek bazı görüntü işlem tekniklerini ağ beslenmeden önce kan hücrelerinin kalitesini artırmak için kullanmaktadır. Bunun yanı sıra, görüntülerdeki hücre örüntüleri ayırt saptama teknikleri kullanılarak çıkartılmıştır: Canny, Sobel ve Prewitt. Tezde, farklı ayırt saptama algoritmaları kullanan ağ modelleri arasında kıyaslama yapılmıştır. Sonuç olarak, Canny'nin mikroskobik kan hücreleri görüntüleri arasında en gerçekçi kanserli hücre örüntülerini çıkardığı saptanmıştır. Bu, Canny ayırt detektörünün ağda en yüksek doğruluk payını göstermesiyle bulunmuştur.

***Anahtar kelimeler**:* Sinir ağı; geri yayılım; Akut Miyeloid Lösemi; Canny; Sobel; Prewitt

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN: Artificial Neural Network

BPNN: Back Propagation neural network

MSE: Mean Square Error

SEC: Second

SVM: Support Vector Machine

CHAPTER 1
INTRODUCTION

1.1 Introduction

Leukemia is a kind of cancer in which the body creates extensive quantities of strange cells regularly white blood cells (WBC). There are two noteworthy sorts of leukemia: constant and intense (Lim, 2002). The word intense means the ailments develop and advances quickly (Lim, 2002).

The bone marrow is invaded with over 20% of impact cells and when myeloid cells are influenced, the sickness is called intense myelogenous leukemia(AML) while the cells influences lymphoid cells, it is called intense lymphocytic leukemia (ALL) (Panovska-Stavridis et.al, 2008). The early and fast classification of the intense leukemia sicknesses, significantly helps in giving the fitting and successful treatment for that specific kind (Khasman et.al, 2010). This is critical, as the common history and response to treatment shifts as per the sort of shoot engaged with the leukemia procedure (Panovska-Stavridis et.al, 2008).

The primary classification contrive proposed by the French-American-British (FAB) Cooperative Group confines AML into 8 subtypes (M0 to M7) and ALL into 3 subtypes (L1 to L3). The FAB classifications of ALL (L1 to L3) which are isolated in perspective of morphology, including cell measure, unmistakable nature of nucleoli, and the whole and appearance of cytoplasm (Bennett et.al, 1976 and Panovska-Stavridis et.al, 2008).

As per French-American-British (FAB) classification additionally, the depiction of cells is little and uniform for ALL-L1. In the interim, cells of AML-M1 are extensive and standard (Bennett et.al, 1976 and Panovska-Stavridis et.al, 2008). Right now, routine conclusion and classification of morphological highlights for intense leukemia blood tests is utilizing magnifying instrument assessment (Sabino et al., 2003). The procedure is a thorough, difficult and tedious work performed by hematologists, technologists or restorative skill (Sabino et al., 2003).

To enhance the unwavering quality of the determination and diminishing the reliance on human specialists, a few past investigations created robotized and semi computerized conclusion and classification utilizing counterfeit neural system for therapeutic pictures.

Manmade brainpower in view of neural system applications greatly affects the translation of medicinal pictures (Chenn et.al, 2004).

The works in neural system usage for intense leukemia determination have been finished by a few analysts. By utilizing AML/ALL informational collections Toure et al. (2001) proposed Multilayer Perceptron Network (MLP) and the most elevated execution rate 58% was accomplished, Ryu et al. (2002) tried different things with Modular Neural Networks as the classifier and the best execution was accomplished 75% lastly, Xu et al.(2002) proposed ellipsoid ARTMAP neural system and the best outcome was 97.1%.

In 2000, Mashor has presented the Hybrid Multilayered Perceptron Network (HMLP) that has been demonstrated to altogether enhance the execution of Multilayered Perceptron Network (MLP). The HMLP arrange has been tried on other informational collections, for example, on cervical cancer and bosom sores informational indexes and effectively confirmed each sort of informational indexes accurately with high rate in both preparing and testing stage (Mat-Isa et al., 2004 and Mashor et al., 2007). Other than that, Mashor et al. (2004) had utilized the HMLP system to ordered 3D question utilizing 2D minute and best acknowledgment precision of up to 100%.

In this work, a neural network is employed for the purpose of the identification of acute myeloid leukemia. The network is simulated and tested using normal and acute myeloid leukemia images obtained from a public database. The work includes an image preparation stage where the images are first processed in order to be ready to be fed into network. Image enhancement techniques such as histogram equalization and edge detection techniques for the segmentation of cancer blood cells, are used. This thesis includes a comparison of the performance of network when different types of edge detection techniques are used. At first, the Canny operator will be applied on the histogram equalized images and the network will be trained and tested so that its performance can be evaluated. Secondly, same images will be used but here canny is replaced by the Sobel and Prewitt edge detectors and similarly the network performance is checked and evaluated. This aims to investigate the network performance and accuracy on the different types of edge detectors.

1.2 Aims of Thesis

Neural networks have shown great accuracies when used in medical fields such as in disease classification and diagnosis. Classifying diseases has been and still a hot topic for artificial intelligence. However, the difficulties of such classification tasks lie under the type of medical images used as inputs of the network. Blood cell images are complex structures that involve different parts including the cancer cells, and this is what makes this type of images a complex task that needs to be processed before feeding into neural network. Thus, in this thesis images are processed first in order to extract the regions of interest and removing other useless parts, which may help the network in learning the differences between normal and diseased images. The aim of this work is the acute myeloid leukemia identification of blood cell images using a backpropagation neural network trained using microscopic images of healthy and acute myeloid leukemia blood cells.

Moreover, in this thesis, we aim to use three types of edge detectors and check the performance of network of each one.

1.3 Thesis Structure

The following thesis is presented as follows:

- Chapter one is a general introduction of the thesis in addition to defining its main aims.
- Chapter two presents an overview of the image processing tools used in medical field and medical imaging applications. Moreover, this chapter discusses the artificial neural network and its working principles in addition to the backpropagation learning algorithm modeling.
- Chapter three discusses the methodology of the presented study. In this chapter, the database used for training and testing the system is described. Moreover this chapter shows the training phase of the system where the performance of the networks is shown and discussed.
- Chapter four discusses the testing performances of the networks in which the accuracy of each network is calculated and discussed. This chapter also includes the results discussion of the networks used.
- Chapter five is a conclusion of the thesis in addition to the future recommendations.

CHAPTER 2
OVERVIEW OF IMAGE PROCESSING AND NEURAL NETWORKS

2.1 Segmentation

Segmentation is a partitioning of an image so that a particular region is extracted or segmented. However, this cannot be easily achieved, as it depends on some properties of the image or the region that should be detected such as edges, shapes, textures, intensities etc..

Over the past decades, different and many algorithm were developed for segmentation purposes in medical images (Fu and Mui, 1981) (Pal and Pal, 1993) (Koshana, 1994) (Lucchese and Mitra, 2001). Those approaches are all based on different properties of images. Those properties can be the points, regions edges, objects or regions etc..

- ***Algorithms based on the points properties***.

This algorithm is based on detecting a point in a homogeneous part of the image. This is achieved by analyzing some properties of the point such as colour, brightness, intensity and other characteristics. The drawback of this algorithm is the difficulties in selecting the important and useful features in images that have many homogenous segments of similar point characteristics. Many researches have used these approaches for segmenting medical images (Sharma et al., 2010)(Withey and Koles, 2007)(Zhang and Wang, 2000).

- ***Algorithms based on the edge detection.***

This algorithm is very popular for segmentation, in particularly, in medical field where a certain region segment in the image needs to be extracted (Aroquiaraj and Thangavel, 2013) (Wu et al., 2015) (Sahakyan and Sarukhanyan, 2015). Edges in an image are the changes and discontinuities in intensities of the image pixels. Hence, this approach works mainly on the images which have brightness or intensity changes on its region edges. Thus, detecting these intensity changes can lead to segmentation of the region edges which for an object in an image.

Researchers have used various algorithms for segmenting the breast tumorous cells in histological images. The authors in (Erezsky et al., 2015) reviewed different segmentation

algorithms such as K-means, Watershed, and texture segmentation. These 3 techniques were applied to breast cell images and the signal to ration for each technique was calculated. Moreover, the authors proposed their own technique for breast cells segmentation which is based on detecting the properties of point connections. Moreover, the authors claimed that their proposed method yielded better segmentation results and lesser signal to noise ration compared to other discussed techniques.

Another breast cancer cell segmentation and contouring algorithm is proposed in (Mouelhi et al., 2011). In their work, an algorithm for segmenting the breast cancer cells is based on watershed and concave vertex graph as a next stage since the segmentation here occurs on many stages. At first, the malignant cells are detected using the geodesic active contour. Then high concavity points are taken from the cell contours to be then used for selecting the clustered cell regions only. Secondly, the touching cells regions are first segmented using watershed technique and then a concave vertex graph is constructed. This shows the inner edges and concave points which helps in separating cells regions. Finally, the authors of this work showed that their algorithm is very accurate in breast cancer cells segmentation without losing geometrical features.

An algorithm for the tumor cells detection breast cells microscopic images is proposed in (Phukpattaranont and Boonyaphiphat, 2006). The algorithm is comprised of two processing stages. The first one is the segmentation of breast cells using watershed mathematical process. Second, the breast cells are extracted or described using Fourier transform descriptors and the principal components analysis is performed to classify cells into normal or cancerous cells.

Moreover, authors in (Vahadane and Sethi, 2013) improved the watershed segmentation algorithm to detect breast cancer cells in histological images using nuclear segmentation. Their algorithm is based on many image processing techniques such as image enhancements and Ostu's tresholding in addition to the fast radial symmetry transform (FRST) for the nuclei extraction and foreground seeds generation.

Gaussian smoothing is first used to remove the high frequency noise and the blurred nuclei segmentation. Then, background markers are used based on the image information to reduce the over-segmentation. FRST is also used to extract nuclei and to form foreground seeds. Finally,

post-processing takes place by using erosion and dilation which results in segmenting the cell nuclei.

2.2 Neural Networks

Artificial neural systems are structure that originated from the cerebrum of the human brain that is used for reasoning. The structure has been used to deal with troublesome issues in science. The vast majority of the structures of neural systems are like the organic mind in the requirement for preparing before having the capacity to complete a required assignment (Zurada, 1992). Like the standard of the human neuron, neural system processes the aggregate of every one of its data sources.

On the off chance that that aggregate is more than a decided level, the journalist yield would then be able to be enacted. Something else, the yield isn't go to the actuation work. Figure 1 illustrates the principal assembly of the neural system where the source of the weight and info on summation of work is shown. The quantity of neuron that is find in a structure can is referred to as the yield work.

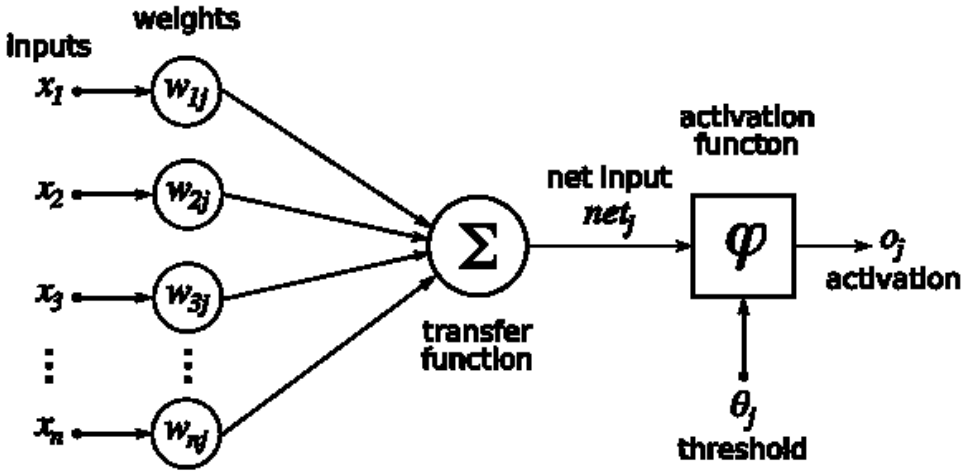

Figure 2.1: Artificial Neural Network's Basic Structure (Zurada, 1992)

2.3 Structure of ANN

The ANNs structure contains three layers despite the learning technique. These angles are the layers, weights, and initiation capacities. Every last one of these three sections plays an

imperative lead in the ANN limit. The three sections or segment works collectively to ensure proper working of the system (Sathya and Abraham, 2013).

2.3.1 ANN Layers

The mutual relationship that occurs between the layers of ANN is the major derivative to its creations. The layers interact by sending information between each other using the synaptic weight. The ANN structure can be subdivided into three layers that is listed in the subsequently section below.

1. Input layer: This is the first layers that are found in the neural system of ANN. This layers is major that send information or data to other layers in the neural system. It can be regarded as a sensor because it doesn't process later but only pass information processed by other layers.

2. Hidden layers: this can be regard as the central bit of the neural system. It involves no less than one of the layers which is the input layer and the neural layer. This layer transmits the data to the output layers. The Hidden layer can be regards as the intermediate layers or as a principal layer because the synaptic weight found in it is reliable (Sathya and Abraham, 2013).

3. Output layer: This layer is regard as the output layers because its last contact where the results of the neural system are gotten, the output layer got its information that is processed from the Hidden layer.

The figure below shows the neural system and the interactions that occur between its three layers. The first layer which is the input layers is the source of the data that is passed to the hidden layer and later to the output layer. The yield or result of the neural system is gotten from the output layer.

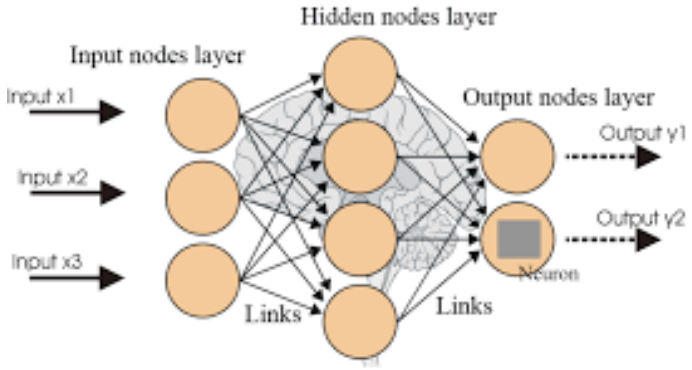

Figure 2.1: The ANN structure showing the three layers (Sathya and Abraham, 2013)

2.3.2 Weights

The ANN weights stands for the network memory in which all information is provided. The weights estimations are invigorated reliably in the midst of the planning of the system until the point when the yield is met. The weights or memory are then secured to be used as a piece of future. The estimations of the weight of ANNs can be regards as the network memory (Rojas, 1996).

- **The activation capacities**

Once the data are enacted from the source and passed across the layers through the synaptic weight, the yield or output is known or gotten by using a trade work. Also, on the other hand in some actuation capacities, the capacity is utilized to decide how much the handled information will partake in developing the aggregate yield of the system.

The neural system is very reliable in determining whether the neuron can adequate transmit its self to the associated layers or not and this made the initiation capacity to be very critical (Rojas, 1996):

- **Linear initiation functions or slope**

In this sort of the work, the yield is fluctuating straightly when the input is close to nothing. When the input value is massive, the preeminent yield is limited by 1 as showed up in Figure 3.

- **Threshold function (Hard activation function)**

The limit yield is zero if the summed input isn't as much as certain estimation of edge, and 1 if the summed input is more significant than edge. The yield can be located between zero and one. The limits yield can be enacted and be deactivating as found.

- **Sigmoid function**

This function can run in the vicinity of 0 and 1, however sometimes it’s better to run its within -1 and 1. The most perceived sigmoid limits are the logarithmic sigmoid and hyperbolic digression. The above listed functions are the most utilized as a part of the back proliferation since they are differentiable. The recipes of these two capacities notwithstanding the bends are displayed in Figure 3. The incline of the bends can be changed in light of the purpose is to be utilized for (Michael, 2005).
In the process of calculating the back-induction, the log-sig and tan-sig capacities are utilized. This two function listed above can also be used separately.

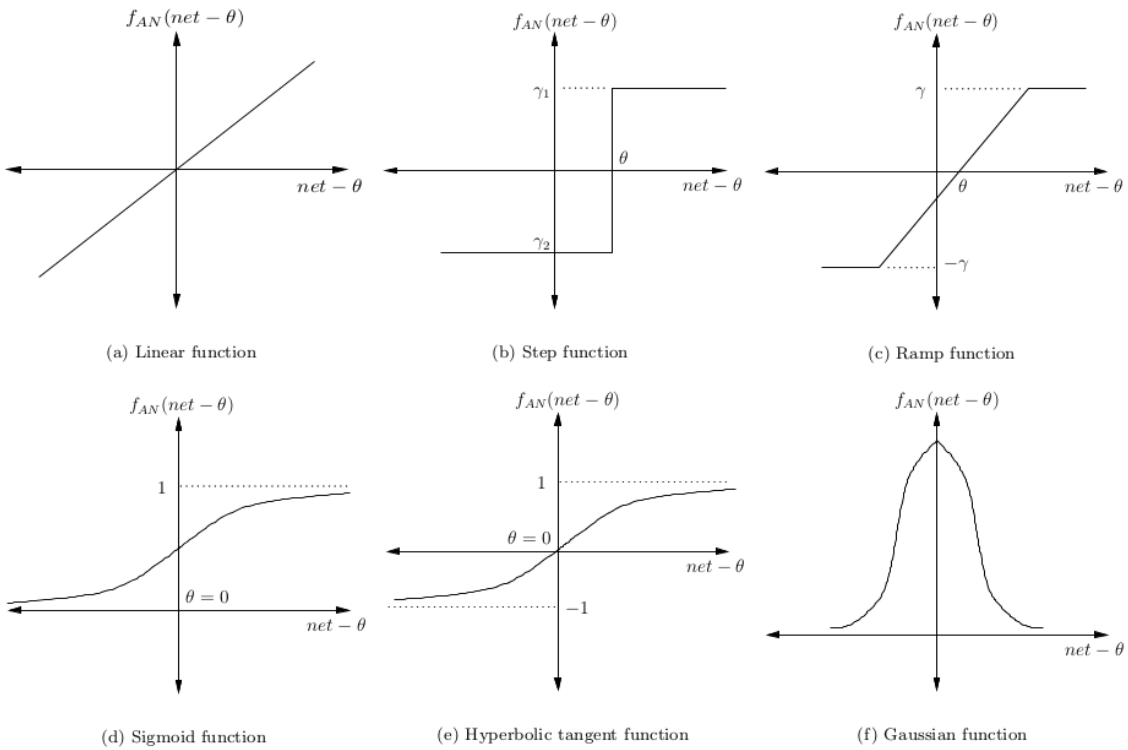

Figure 2.2: Activations functions of neural networks (Michael, 2005)

2.4 Classification of ANNs

ANNs are sometimes described using different approach such as; information, limits and preparatory system. The transmission of data in the ANNs system started from the input later to the hidden layer and later to the yield or output layer. On the aspect of functions, neural system can dedicated to varieties of assignment and can be accomplished with it. This functions can subdivided into four major classes:

- Classification: This is when enquiry is passed out but done using a known arrangement.
- Association: This is creating interaction or relationship between articles to achieve a more outline program.
- Optimization: This is when the action is to establish a better response to an issue or case.
- Organization: The ANNs attributes is needed to factor out using the preparation method.

2.4.1 ANNs Training Methods

The main purpose of preparing a system is in order to achieve a wanted result or yields (Krenker et al., 2011). The two fundamental learning method which comprises of the; coordinated and the unsupervised learning method are utilized in order to enlighten the systems.

- **Supervised learning method**

The ANNs values are gotten from the input information. The system at that point refreshes its weights as indicated by a characterized calculation govern until the point that it unites to a base mistake or achieves a most extreme number of emphases. An imperative case of the directed learning technique is the mistake back engendering strategy.

- **Unsupervised learning**

In this technique, the input information is given to the system which thusly alters its weights as per characterized conditions.

2.5 Backpropagation Learning Algorithm

The Back propagation neural network algorithm is executed utilizing a feed forward network, back spread updating process, and lastly supervised learning topology. This method of neural network algorithm was developed in the late 1980s. Back propagation is a multipurpose in the field of recognition algorithm. Even though this algorithm is a very efficient and accurate model it has a major constrained which time is consuming. Back propagation network when given a certain amount of elements to simulate can produce a certain level of correctness (Krenker et al., 2011).

Back propagation since its creation has a simple attribute thus making it a legacy algorithm by maintaining it initial attributes till today. The first layer is the input layer in which the initial weights are being inputted, next is the activation function layer in which the weights are processed before the last layer which is the output layer for the weights to be presented. Lastly is the error layer in which the weights are update in the input layer before the network is run again for another iteration until a minimal level of error is achieved which can be neglected

The said process above is repeated until a certain level of error achieved which is to a bare minimal then the network can be said as learned network. The Figure 4 shows the artificial neural network layer with error back propagation

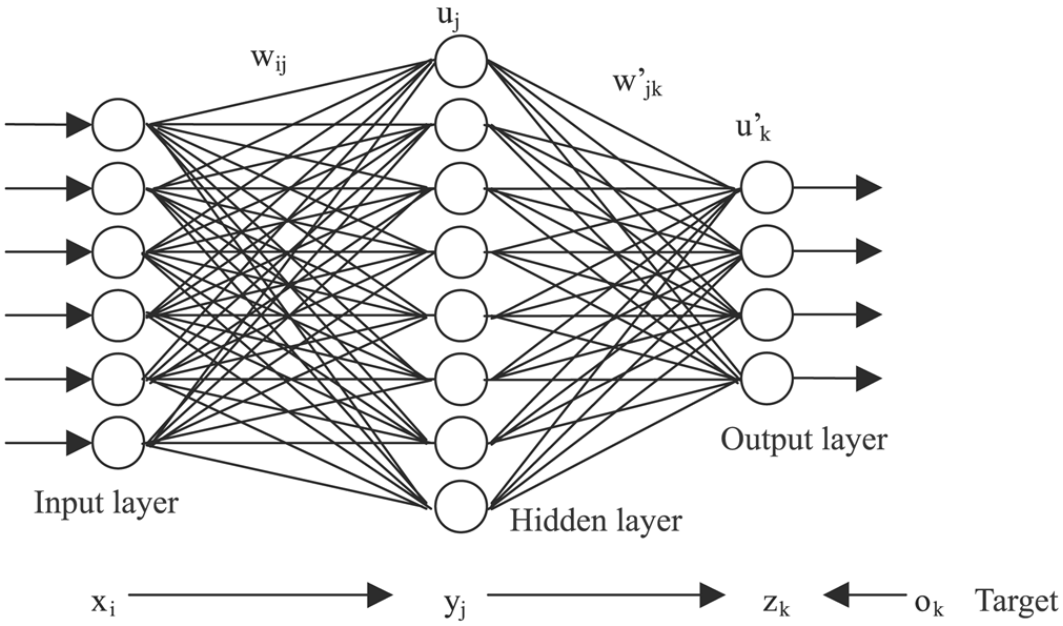

Figure 2.3: The artificial neural network structure with error backpropagation (Krenker et al., 2011)

There exist two basic protocols in the process of back propagation which are learning rate and momentum factor, the first which is the learning rate determines after a test of the network if the network weights shall be updated or not, thus for every iteration the learning rate determine if there should be an update of the weights or not, eventually learning rate should be set to the minimal because a network with a higher learning rate makes the network to memorize instead of learning the updates, and lastly is the factor of momentum utilized in organizing the update intensity that the system can do.

2.5.1 Backpropagation Algorithms Modeling

As an algorithm, back propagation utilizes the error minimization theory coupled with gradient descent to figure and point out errors that are least squared, doing so ensures every iteration done will have gradient error calculated which results to a hindered delivery of functions.

In most of cases, the tangent or logarithmic sigmoid functions are used. The sigmoid function is defined by (Jaleel et al., 2012):

$$o(x) = \frac{1}{1+e^{-ax}} \tag{2.1}$$

The above equation is the constant, which control the slant, consequently the derived sigmoid is:

$$o^{\backslash}(x) = f(x)(1-f(x)) \tag{2.2}$$

Training of neural network can be categorized into sub divisions as equated below:

1- Feed forward training: used in training as well as testing the network.

2- Error back propagation: categorically used to train the network.

In a feed forward network, output and can be denoted as

$$TP = \sum x_n \omega_n + b_n \tag{2.3}$$

The x_n is the (input data), the w_n is the (weight matrix), while the b_n is the (double values). The total values of each layers is the Tp. The starting functions exist as a straight or as a non-direct function. A typical straight capacity that is broadly conveyed in neural networks is the sigmoid capacity which is characterized in the capacity, another case which is the digression of the sigmoid that is specified by:

$$o(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.4}$$

Another fact about this function is that it can be continuous and derived also, the derived function can be equated by: T

$$o^{\backslash}(x) = 1-\frac{(e^{x}-e^{-x})^{2}}{(e^{x}+e^{-x})^{2}} \tag{2.5}$$

the consequence of the past activation function is the right aftereffect of the NN, the output is define as the objective which is utilized to deliver the amount of error, the error rate is expressed

by the equation underneath, the reason for preparing NN is to decrease the amount of error in the Neural network.

$$E = \sum (T - o)^2 \tag{2.6}$$

T is the objective output, while E is value of the error functions which:

$$\Delta_j = (T_j - o_j) o_j (1 - o_j) \tag{2.7}$$

The values gotten from the network is defined in the equations and the network becomes updated, the weights are updated using:

$$\omega_{jhnew} = \omega_{jhold} + \eta \Delta_j o_h + \alpha(\delta \omega_{jhold}) \tag{2.8}$$

The hidden layers weights are updated using error update which is given as:

$$\Delta_h = o_h (1 - o_h) \sum \omega_{jh} \Delta_j \tag{2.9}$$

The new weights would be given by:

$$\omega_{hinew} = \omega_{hiold} + \mu \Delta_h o_i + \alpha(\delta \omega_{hiold}) \tag{2.10}$$

the momentum factor values is known as the α which is used in reducing the number of updates and η as the learning rate which is used in updating the weights, after successful network run, a new iteration is done until a desired until it arrives to an acceptable error value (Jaleel et al., 2012).

CHAPTER 3
SYSTEM METHODOLOGY AND DESIGN

In this chapter, the need and significance of the blood cells cancer identification system is presented. Moreover, the materials and methods are shown.

3.1 Why Blood cancer

Image processing has been used in medical applications for various purposes. Image processing techniques play an important role in medical image diagnosis, classification, segmentation, analysis etc.. Analytically, comparative results between cancers show that, blood cancer is the highest killer followed by blood cancer. As demonstrated in (Tripathi et al., 2014), blood cancer is mostly found in woman.

Lesion as classified into two categories; malignant and benign is actually the cause of this dangerous illness. Among these two categories, benign is removable and unlikely to reoccur and hence termed; harmless lesion. While malignant in other hand is termed cancerous cell having high potential to grow and spread to other parts of the body (Dudea et al., 2013). High number of patients diagnosed of blood cancer does not notice its presence and probably died before getting proper medication. Therefore, to reduce the number of death resulting from breast cancer, early detection is necessary for proper treatment.

Blood cells cancer segmentation in microscopic images is a tedious job, in particularly when it comes to detecting whether the cells are with tumor or not. This is usually due to the similarity and between both types of tumorous and healthy cells. Doctor's diagnosis decisions, like segmentation, are usually based on some visual inspections in which they check or depend on the size, colour, and texture of the cells. Those decisions may be affected by fatigue, stress, and less experience that humans may have. Thus, there is a need for an automated and computerized diagnosis system that helps in segmenting and detecting the cells with tumors in a breast microscopic image. These kinds of systems are lesser time consuming, not costly and may be more accurate since they are not affected by the aforementioned human factors.

3.2 Materials and Methods

Image processing has been extensively used in medicine. Segmentation is always the most common process needed in this field. A blood cell image contains millions of cells. This makes it very tough for doctors to find the tumorous cell. Image processing can be a useful tool in this case as it helps in detecting and segmenting the tumorous breast cells since they differ from other cells in terms of brightness and intensities.

Thus, in this study, a system for the identification of Myeloid acute leukemia is developed. The system attempts to first detect and segment the blood cancer cells found in images using image processing tools. Those tools are used in order to find the tumorous cells in blood images and segment them. This is done by margining the parts of image that have no tumor and spotting on the cells that are cancerous by extracting their features using different techniques. Figure 3.1 shows an image of blood cells with Myeloid acute Leukemia. Figure 3.2 shows a normal blood cell image

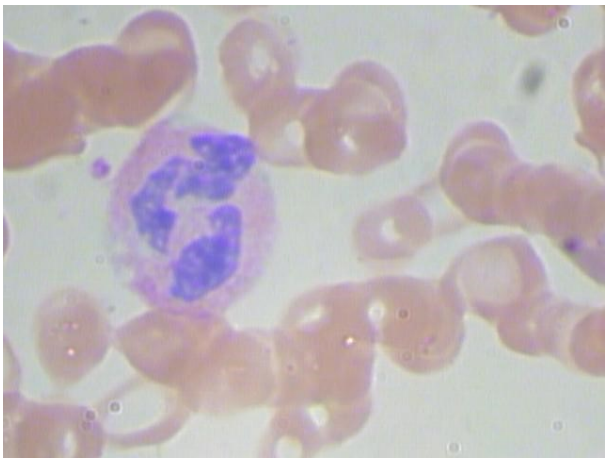

Figure 3.1: Blood cell with Myeloid acute Leukemia

The point is to detect the infected cells i.e. myeloid leukemia cells if there is, and regard the other cells. This is seen as a feature extraction phase where the region of interest is only spotted on. Then those resulted image which contain the segmented tumors are fed into network which then gets trained to classify them into normal or cancer blood cell images.

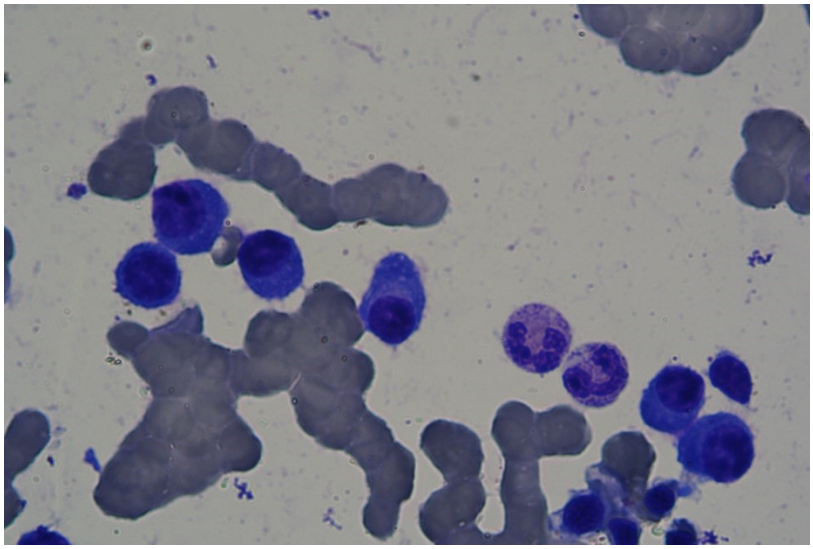

Figure 3.2: Normal blood cell

3.3 Image Analysis

Image are first analyzed using some processing algorithms which aim to enhance and extract the features of images which would help in reducing time and increasing the accuracy in the next stage where the network is used as a classifier of both types of images.

Firstly, the images are enhanced using image sharpening which is an image processing technique that is usually used to enhance the image by sharpening its features such as edges, corners, and small objects; as some of them might have noise artifacts which should be removed to not affect the segmentation process in later stages. The result of image sharpening is shown in figure 8(c) below. Note that image sharpening is done by converting the RGB image into L*a*b colorspace. Then, the L channel is sharpened only, and the L*a*b is converted to RGB again which results in a sharpened image of the original one where the strong features presented in the first RGB image are enhanced, like edges.

The blood cell images undergo histogram equalization in order to enhance the contrast of images, in particularly, the sharp edges which represent the tumors in the cancer blood cell images. Histogram equalization is an image processing technique meant to enhance the contrast of the image by mapping or transforming the values of the intensities of pixels in image into

different ranges so that the histogram becomes flat. This results in better quality images where the pixels are sharper and brighter. The result of histogram equalization is shown in Figure 3.3.

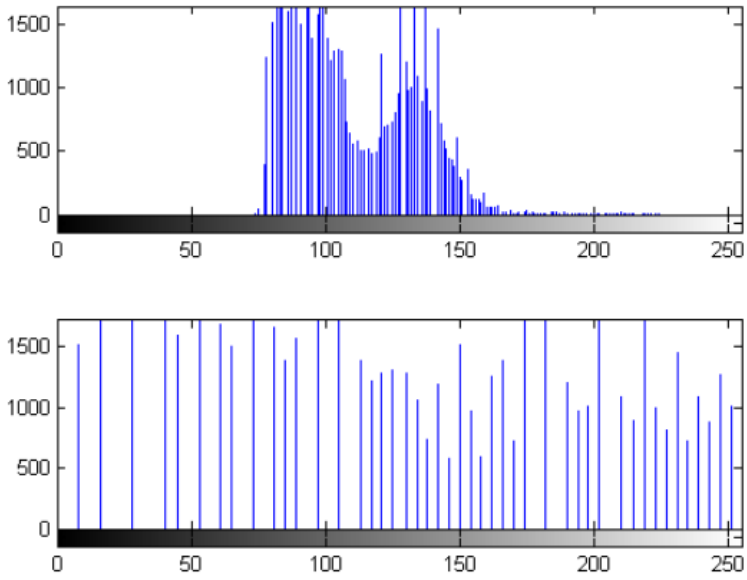

Figure 3.3: Histogram equalization

Figure 3.3 shows a histogram of an image before and after applying the histogram equalization.

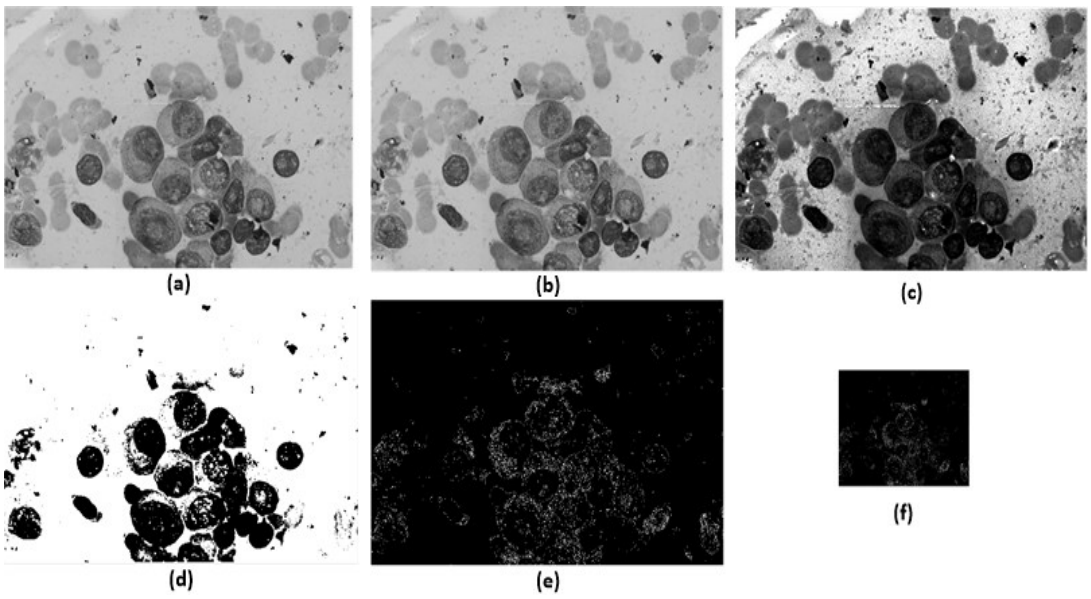

Figure 3.4: Blood cell image undergoes Canny edge detection

To highlight the images more and more, the intensities of pixels are increased by mapping them into other values. This ended up with brighter images where the cells are clearer; including the cancerous cells (Figure 3.4(d)).

After enhancing, the patterns extraction and edge detection take place. In here, edge detection techniques are used. Three different edge detection operators are used on the same images in order to investigate the one that end up with a better classification results in the classification phase. The one that results in a better classification rate is consequently the detection operator that extracts the better features that distinguish the blood cancer cells images from the normal images which have no cancer cells.

Canny, Sobel and Prewitt operators re both investigated here as edge detectors. Edge detection can be defined as an image processing technique for finding the boundaries of an object in an image. Basically, this technique works by detecting the discontinuities in the intensities of pixels. This allows detecting the edges which are the discontinuities in intensities between two pixels. Edge detection can be used for segmentation of objects in images and also for data extraction from images. Many algorithms were proposed for edge detection; each is used based on the application. However, the most common used is called “Canny”, ”Sobel”, and “Prewitt” edge detection in which image are filtered and then edges are detected.

In this work, “Canny”, “Sobel”, and “Prewitt” detectors are used for detecting the edges of the tumors found in the blood cell images in order to investigate the operator that results in better performance.

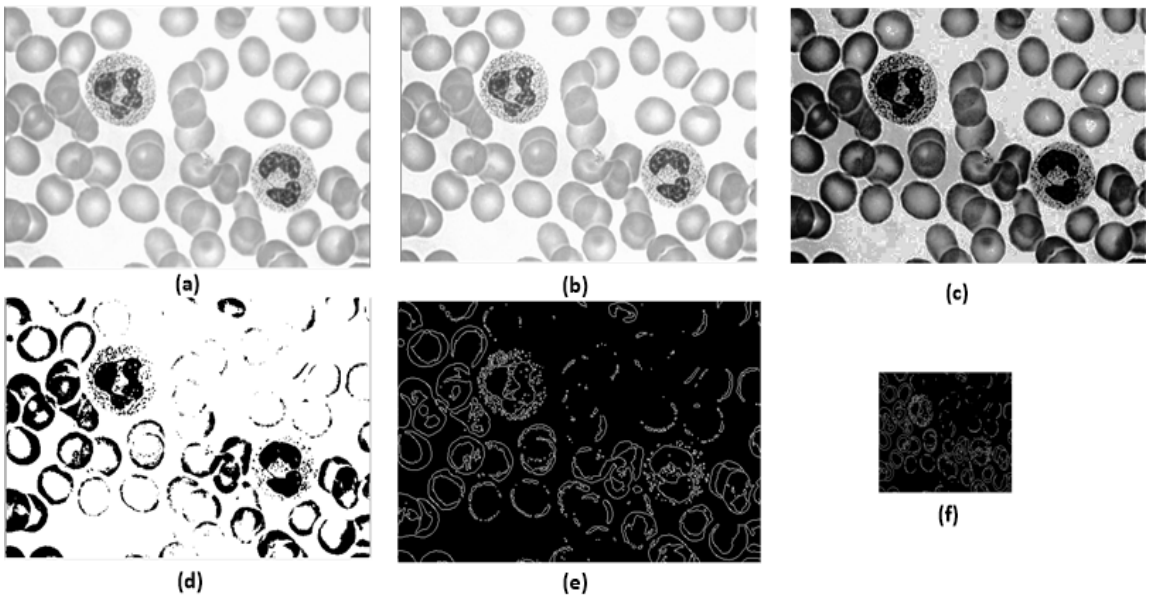

Figure 3.5: Blood cell image undergoes Sobel edge detection

Note that Figures 3.5, 3.6, and 3.7 show some examples of blood cell images which have myeloid acute leukemia, which are converted to grayscale to be then sharpened using image sharpening and enhanced using histogram equalization. The images are also segmented using Canny, Sobel and Prewitt edge detection algorithms, and finally they are reduced to 64*64 pixels.

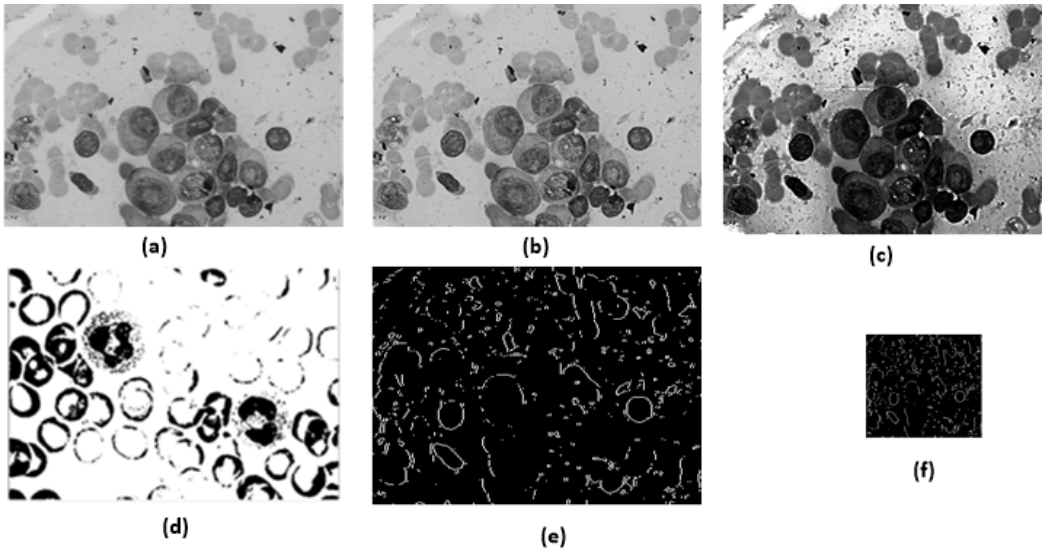

Figure 3.6: Blood cell image undergoes Prewitt edge detection

3.4 Database

A neural network is a data-hungry system as data is its source of knowledge and memory associations. The more data the neural network gets the smarter it becomes. Therefore, the first step is starting a neural network that meant to classify medical images is to find a good source of data which are used to feed the hungry network.

Therefore, in order to saturate our proposed system or the classification of blood cell images we used a public database that contains blood cell images of normal and myeloid acute leukemia cases (Acute Myeloid Leukemia Genomics, 2017). The database consists of 320 images. Among them, 66 are healthy (have no cancer), and 254 are cancerous (myeloid acute leukemia). The figure 3.7 below shows a sample of the normal and the cancerous blood cell images where the first row shows the cancerous images while the second row show the healthy cells.

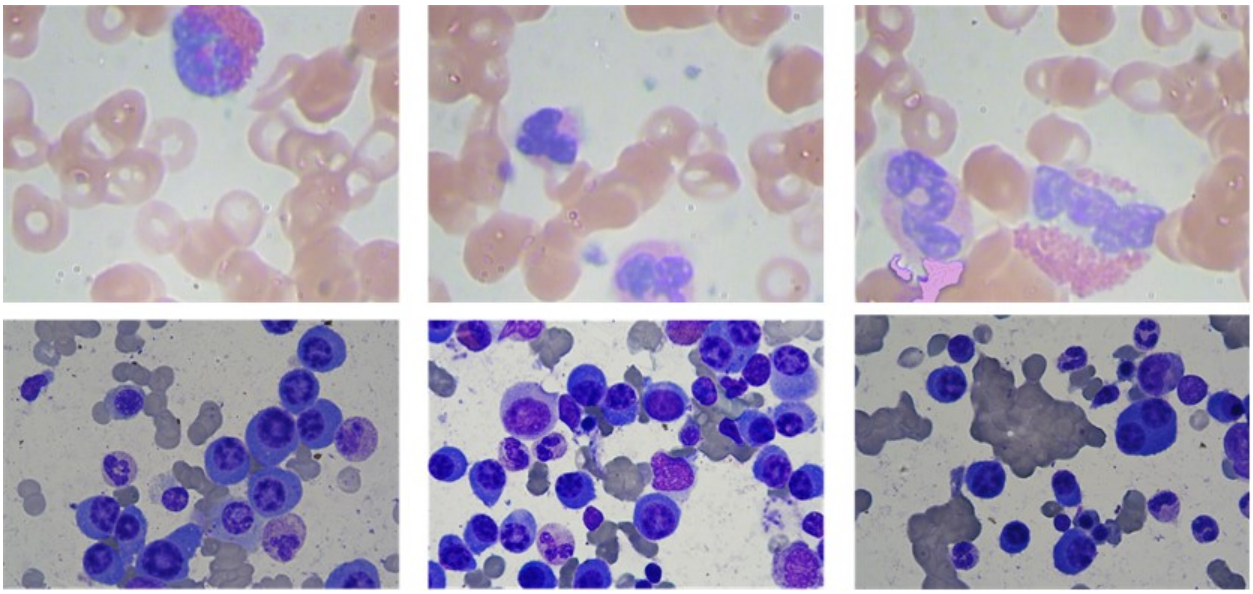

Figure 3.7: Sample of blood cell images

The table below shows the number of images in the database. It also shows the number of images used for training and testing the neural network.

Table 1: Database images

	Healthy blood cell images	**Myeloid acute leukemia images**
Training	50	200
Testing	16	54
Total	66	254

3.5 Proposed Neural System

The proposed methodology is a myeloid acute leukemia identification intelligent system based on some image analysis techniques and a learning system named as backpropagation neural network.

The aim of this dissertation is to determine the model approach of back propagation in the field of neural network to determine various blood cell images which contain healthy and cancerous images, and evaluate the obtained results with that in the existing literature. The thesis includes two parts which can be categorized as the processing and classification stages. In digital image processing, image enhancement technique, and segmentation using edge detection algorithms are presented. These processes are done to acquire high resolution image, convert them to grayscale and spot on the cancerous areas if there is in the images, so that they can be fed to the neural network. After all these processes, the images are inputted into the neural network, after which they are categorized into either healthy blood cell or myeloid acute leukemia images.

In this work, we attempt to show and compare the performance of network when different types of edge detection techniques are used.at first, the Canny operator will be applied on the histogram equalized images and the network will be trained and tested so that its performance can be evaluated. Secondly, same images will be used but here canny is replaced by the Sobel and Prewitt edge detectors and similarly the network performance is checked and evaluated. This aims to investigate the network performance and accuracy on the different types of edge detectors.

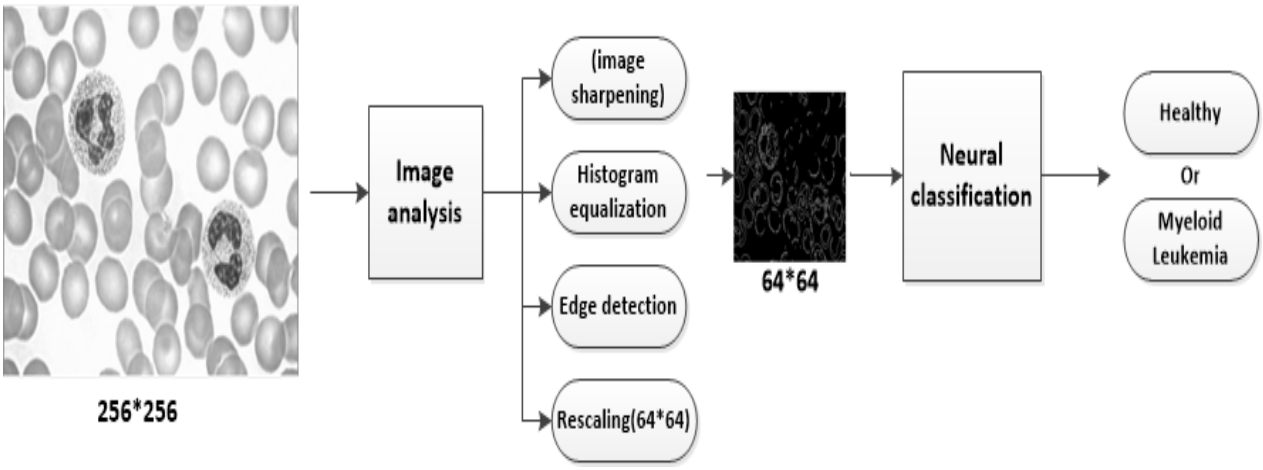

Figure 3.8: Proposed Myeloid leukemia Classification

Furthermore, for more comparison results we investigated the performance of network with image sharpening and without. In other words, images are first processed using image sharpening, histogram equalization, and rescaling to 64*64 pixels; and then fed into network. In another experiment image sharpening is not used and the network is tested and evaluated.

The use of these different learning schemes is to investigate the network performance at different edge detection algorithms; so that the scheme that results in better performance and lower error is selected. Moreover, the aim is to make a beneficial comparison of the network accuracy, and effectiveness at different edge detection techniques and with or without image sharpening.

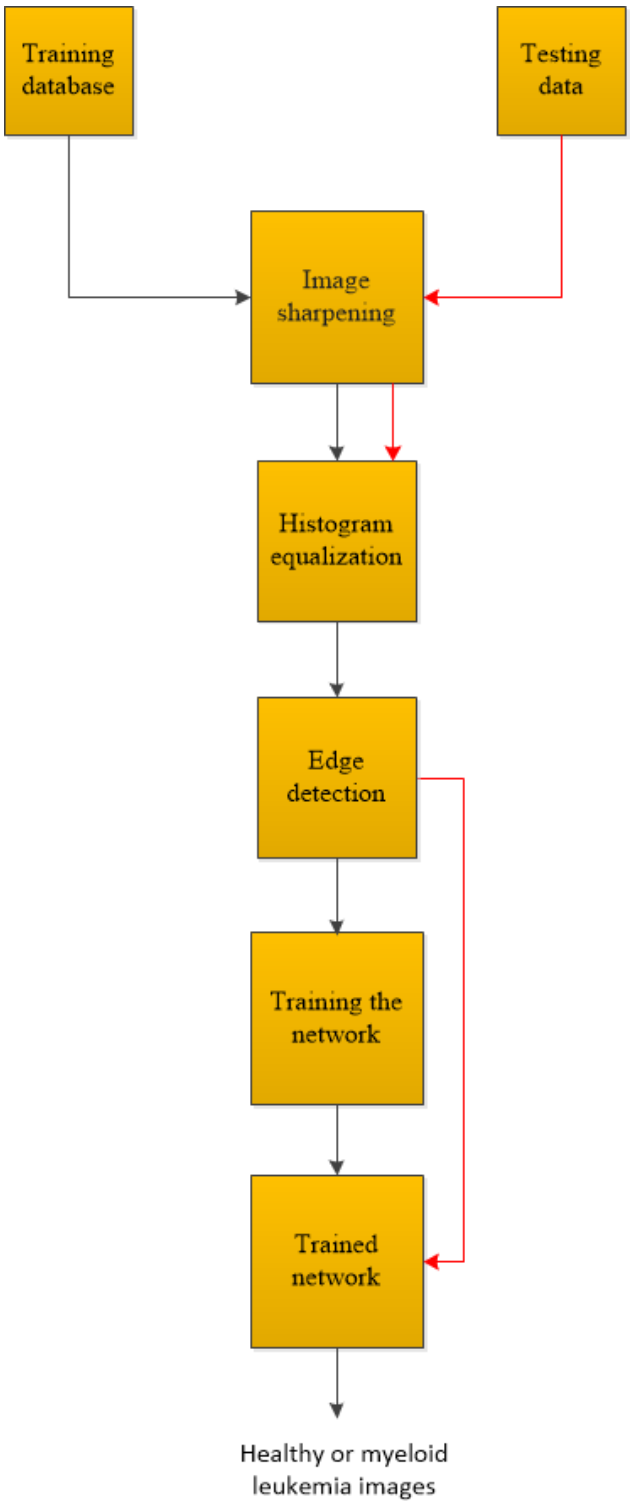

Figure 3.9: Flowchart of the image analysis and classification process of the system

As seen in Figures 3.8 and 3.9, the system is basically based on some analysis of the image which helps in extracting the features on both images types. Then, those images are reduced to 64*64 and fed into network to be then used for training and also testing the network. Network is later on evaluated in terms of training recognition rate, testing recognition rate, training time, and error achieved.

3.6 Networks Training

Neural network gain its power from knowledge and experience it gets during its training. Training is done by sowing the network examples of the different classes that is meant to classify. Thus, more training examples results in a better generalization capability of the network when it is tested.

In this work, a network is to classify the blood cell images in to two classes: normal blood cell images and images with Myeloid leukemia. Thus, the network is trained on both types of images which have different illuminations. The images are collected from one database which contains normal and cancerous blood cell images. The database contains 66 normal images and 254 myeloid leukemia images. As seen in table 1 above, the network is trained on 50 normal images and 200 leukemia images. Note that the network is trained and simulated on Matlab environment.

The network is trained using gradient descent algorithm with adaptive learning and momentum rate. Note that the learning rate and momentum were adjusted to obtain the best convergence of network. Moreover, different values of learning rate, momentum, and hidden neurons, and iterations were experienced until the optimum values of them are selected.

As discussed before, the network is trained first with using ‘Canny’ edge detection and called as BPNN1 and second with ‘Sobel’, and called BPNN2, and third with ‘Prewitt” called BPNN5.

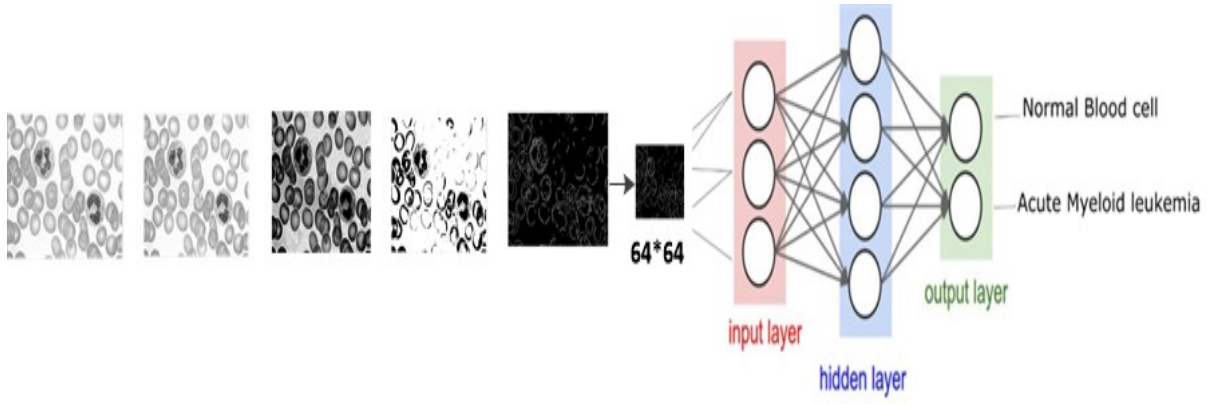

Figure 3.10: Proposed network and image analysis system

Figure 3.10 shows the image analysis stages taken before the image is fed into the network which consists of three layers: input, hidden, and output layer of two neurons, since we have only two classes.

3.6.1 Training the BPNN1

The table 2 shows the network input parameters values during the training. Note that this table is for the network that uses 'Canny' operator in images processing phase.

Table 2: Training parameters of BPNN1

Input parameters	Values
Edge detector	Canny
Hidden neurons number	200
Learning rate	0.12
Momentum rate	0.5
Activation function	"Sigmoid"
Maximum iterations number	2000
Reached iterations	370
Training images number	250

As seen in table 2 the network was trained on 250 images, which were processed using “Canny” edge detector, of both normal and cancerous images with 0.12 and 0.5 as learning rate and momentum, respectively.

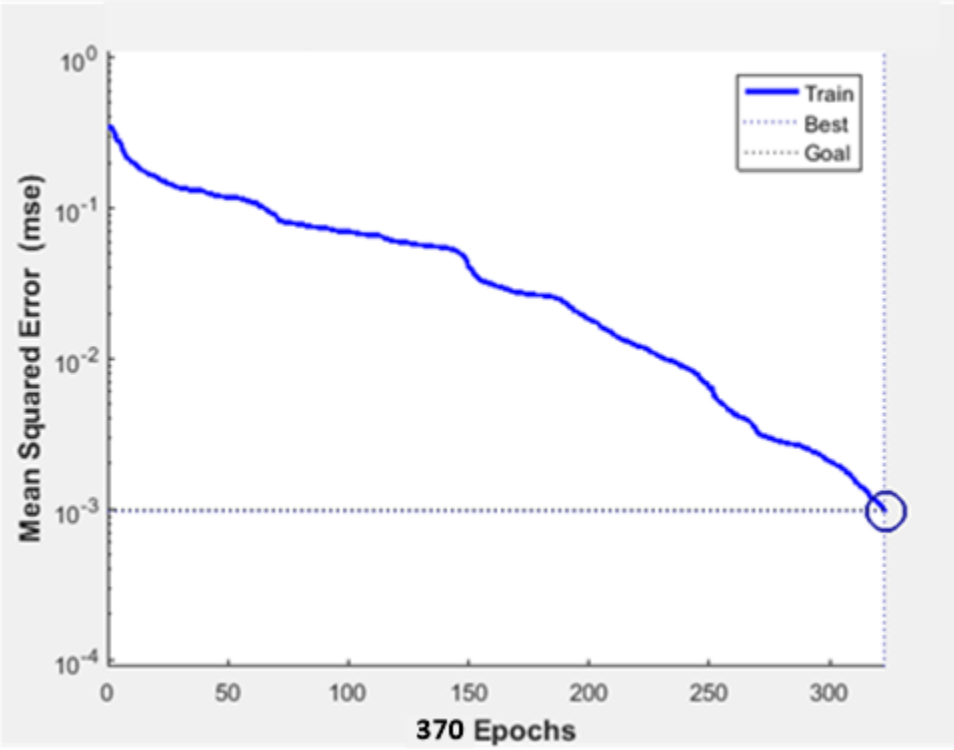

Figure 3.11: Learning curve of BPNN1

Figure 3.11 shows the variations of error with respect to the increase of iterations number. It is seen that the error was decreasing exponentially until the network reaches its minimum error at epoch 370.

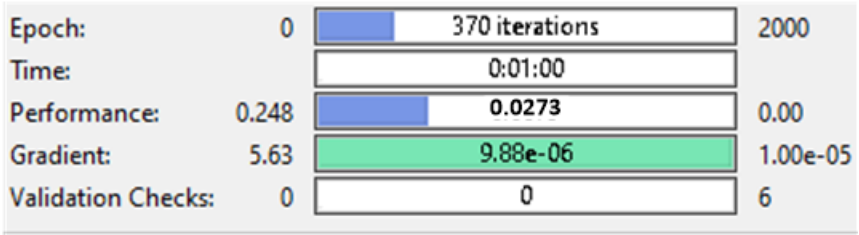

Figure 3.12: Training time and error reached

Figure 3.12 shows the time taken and the error reached by the network when trained with 2000 iterations.

Table 3: Training results of BPNN1

Training Results	**BPNN1**
Training Recognition rate	95.8%
Mean square error reached	0.0273
Training time	60 secs
Iterations required	370

Table 3 shows the training outcomes of the BPNN1 which was trained on 200 images that were processed using "Canny" edge operators. It is seen that the network achieved 95.8% recognition rate in 60 seconds. Moreover, this recognition rate was achieved with only 370 iterations and with a mean square error of 0.0273.

3.6.2 Training of BPNN2

BPNN2 is the network that uses the images processed using "Sobel" operator and similarly it is trained in the same number of images and also same input parameters values as seen in Table 4.

Table 4: Training parameters of BPNN2

Input parameters	**Values**
Edge detector	Sobel
Hidden neurons number	200
Learning rate	0.12
Momentum rate	0.5
Activation function	"Sigmoid"
Maximum iterations number	1000
Reached iterations	112
Training images number	250

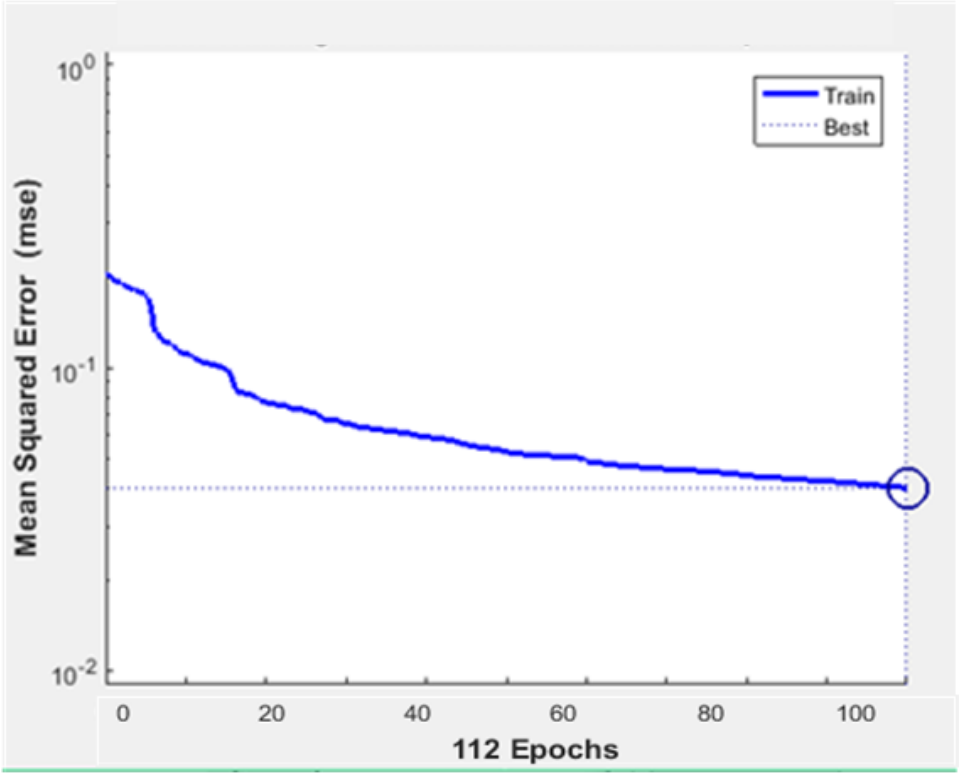

Figure 3.13: Learning curve of BPNN2

Figure 3.13 shows the variations of error with the iteration number. It is seen that the error is increasing when the iteration number is increasing, until it reaches a specific mean square error which was achieved at iteration 112.

Epoch:	0	112 iterations	1000
Time:		0:00:01	
Performance:	0.451	0.0434	0.00
Gradient:	0.108	7.24e-06	1.00e-05
Validation Checks:	0	0	6

Figure 3.14: Training time and error reached of BPNN2

Figure 3.14 shows that that BPNN2 that uses the Sobel detector was able to achieve only 0.0434 error with only 1 sec and with 112 iterations.

Table 5: Training results of BPNN2

Training Results	**BPNN2**
Training Recognition rate	92.3%
Mean square error reached	0.0434
Training time	1 secs
Iterations required	112

Table 5 shows the recognition rate and error achieved of the BPNN2 training. It can be seen that the network couldn't achieved a good recognition rate in training as compared to BPNN1. Moreover, it is seen that the network achieved 92.3% in a very short time (1 sec) and with low number of iterations (112) compared to that of BPNN1.

3.7 Training Without Image Enhancement

For more experimental work, we removed the image sharpening and histogram equalization techniques and processed the images using edge detection only. Images are also processed using "Canny", "Sobel", and "Prewitt" edge detectors in this experiment. The aim of this experiment is to show whether the enhancement plays an effective role in improving the learning of the network or not.

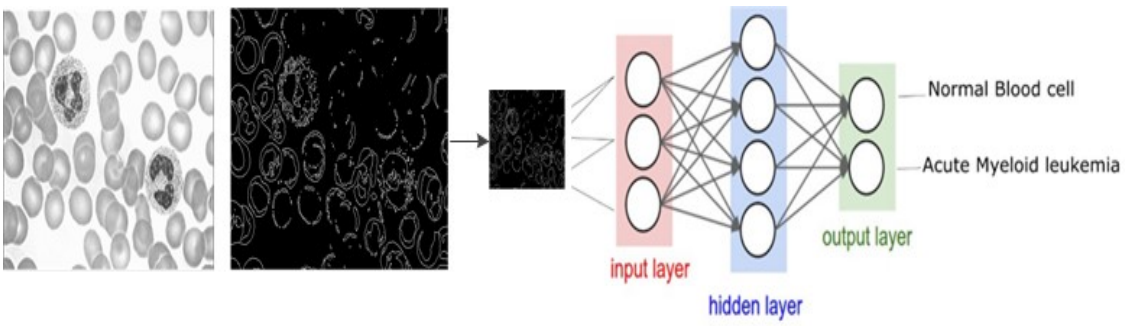

Figure 3.15: Training without image enhancement

Figure 3.15 shows the proposed training of the network in which the images are segmented using edge detection algorithms and rescaled to 64*64, and then fed into network.

Note that "Canny" edge detector is used first and the network is so called BPNN3, and also "Sobel" edge detector is used and the network is so called BPNN4.

Similarly, networks are trained in same images without image enhancement and the results discussed. Note that same learning parameters are used here for both networks.

Table 6: Training parameters of BPNN3 (without image enhancement)

Input parameters	**Values**
Edge detector	Canny
Hidden neurons number	200
Learning rate	0.12
Momentum rate	0.5
Activation function	"Sigmoid"
Maximum iterations number	2000
Reached iterations	30
Training images number	250

The learning curve of the BPNN3 is shown in Figure 3.16. It is seen that the network trained faster that BPNN1 however, it couldn't get a low error compared to that obtained by BPNN1.

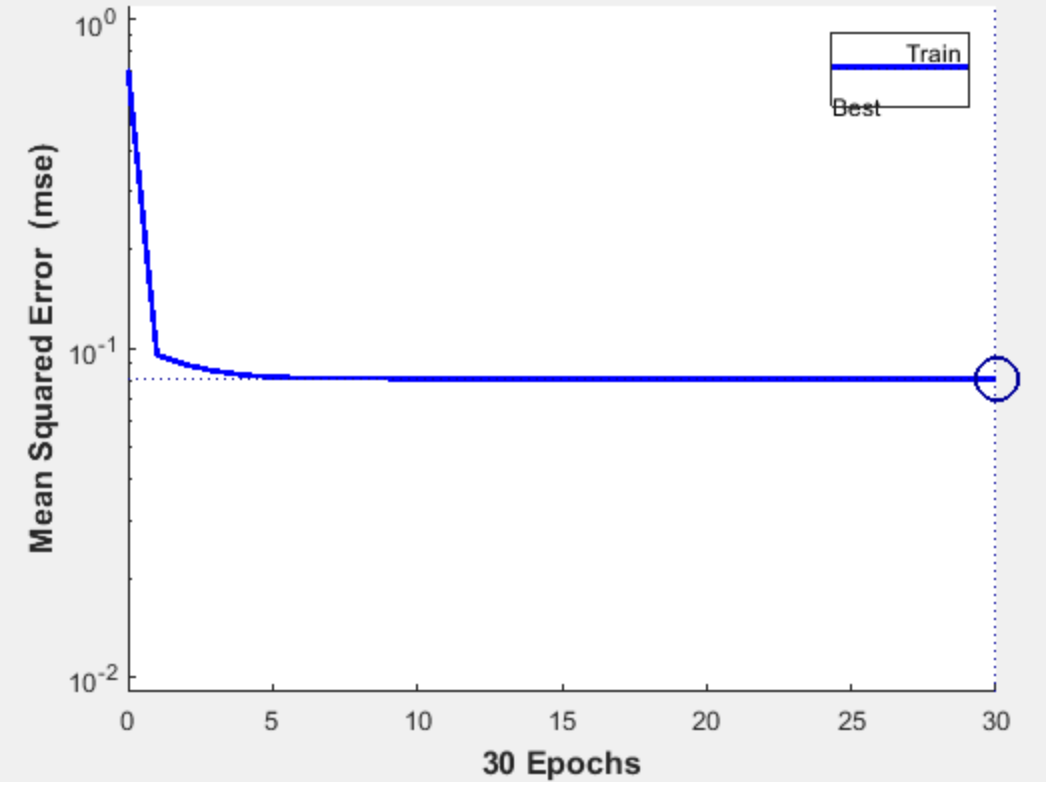

Figure 3.16: Learning curve of BPNN3

The table 7 shows the training results of the BPNN3 in which images were processed without image enhancement. It is seen that the network achieved a lower recognition rate (91%) than that obtained by BPNN1 with a higher mean square error but with lower number of iterations and shorter time

Table 7: Training results of BPNN3 (without image enhancement)

Training Results	**BPNN3**
Training Recognition rate	91%
Mean square error reached	0.0510
Training time	7 secs
Iterations required	30

Similarly, an experiment was conducted by removing the image enhancement techniques but here the "Sobel" edge detection algorithm is used. The results of this experiment are discussed below in which this network is called BPNN4.

Table 8: Training parameters of BPNN4

Input parameters	Values
Edge detector	Sobel
Hidden neurons number	200
Learning rate	0.12
Momentum rate	0.5
Activation function	"Sigmoid"
Maximum iterations number	1000
Reached iterations	119
Training images number	250

Figure 3.17 shows the learning process of the BPNN4 which seems to be as same as the BPNN2. This means removing the image enhancement techniques didn't have much effect on the learning phase of the network when "Sobel" edge detector is used.

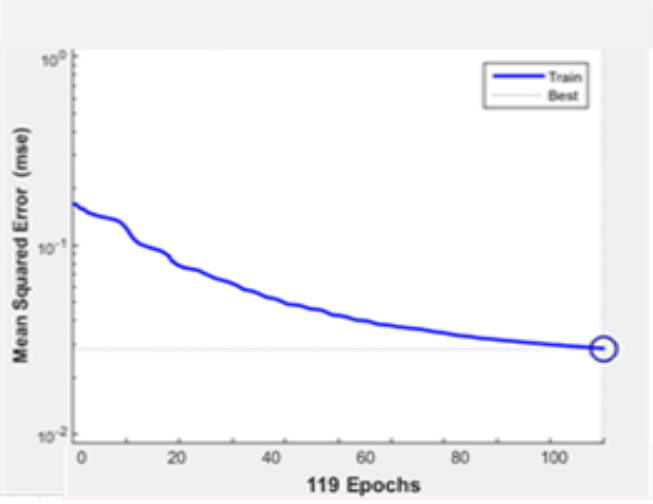

Figure 3.17: Learning curve of BPNN4

As seen in Table 9 the network performance has decreased when image enhancement techniques are removed. This is in terms of training recognition rate, however, in terms of achieved mean error and iteration number, the results remains the same as the error didn't change but the network needed more iterations (119) than those needed for BPNN2.

Table 9: Training results of BPNN4

Training Results	**BPNN4**
Training Recognition rate	90.3%
Mean square error reached	0.0720
Training time	1 secs
Iterations required	119

3.6.2 Training of BPNN5

BPNN5 is the network that uses the images processed using "Prewitt" operator and similarly it is trained in the same number of images and also same input parameters values as seen in Table 10.

Table 10: Training parameters of BPNN5

Input parameters	**Values**
Edge detector	Prewitt
Hidden neurons number	200
Learning rate	0.12
Momentum rate	0.5
Activation function	"Sigmoid"
Maximum iterations number	1000
Reached iterations	11
Training images number	250

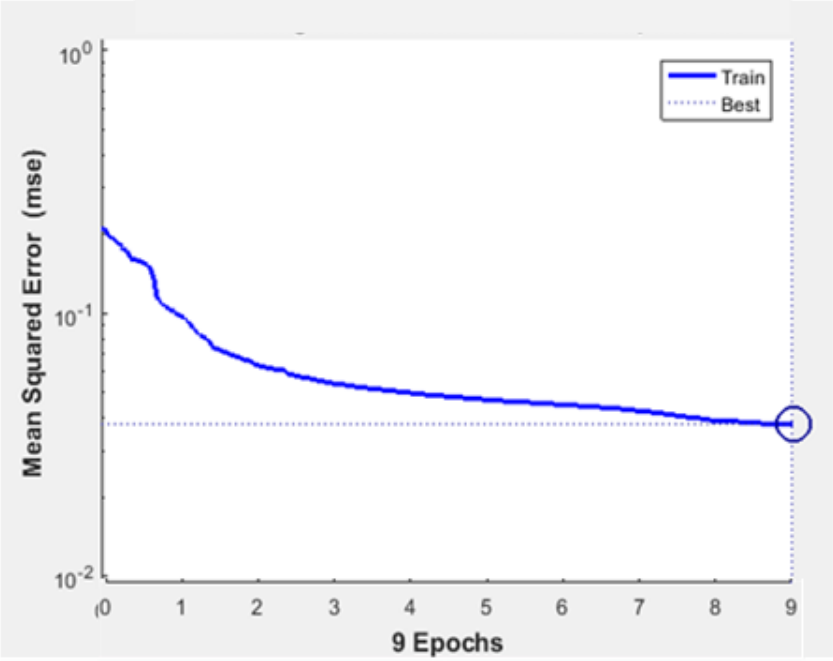

Figure 3.18: Learning curve of BPNN5

Figure 3.18 shows the learning curve of the network that uses Prewitt edge detector. It is seen that the network couldn't learn well since the error is not decreasing.

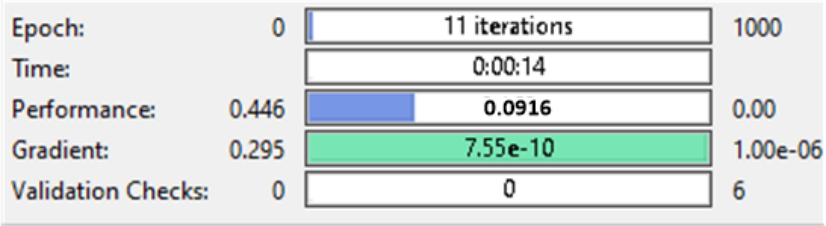

Figure 3.19: Training time and error reached of BPNN5

Figure 3.19 shows that that BPNN5 that uses the Prewitt detector was able to achieve only 0.0916 errors with only 14 sec and with 11 iterations.

Table 11: Training results of BPNN5

Training Results	**BPNN5**
Training Recognition rate	90.13%
Mean square error reached	0.0916
Training time	14 secs
Iterations required	11

Table 11 shows the recognition rate and error achieved of the BPNN5 training. It can be seen that the network couldn't achieved a good recognition rate in training as compared to BPNN1 & 2. Moreover, it is seen that the network achieved 90.13% in a very short time (14 sec) and with low number of iterations (11) compared to that of BPNN1.

This network was also trained without image enhancement techniques and the results were also the same which means that the Prewitt detectors are not detecting the real patterns that dhow the blood cancer cells, which make it difficult for the network to learn. Hence the learning results of the network are low and not that good compared to other networks.

CHAPTER 4
NETWORKS PERFORMANCE EVALUATION

4.1 Testing the Models

All networks are tested on 70 blood cell images. Among them 16 are healthy blood cells and 54 are myeloid leukemia images. Figure 4.1 shows a sample of the images that are used for testing the networks in which the row 1 shows the normal images and row 2 shows the blood cell images containing Myeloid leukemia.

Table 12: Recognition rates of networks during testing

Network models	Training images (250)	Testing images (70)
BPNN1	**95.8%**	**94.2%**
BPNN2	92.3%	91.2%
BPNN3	91%	90.4%
BPNN4	90.3%	90.21%
BPNN5	90.13%	90%
BPNN6	90.13%	90%

Table 12 shows the recognition rates achieved by all trained models when tested on 70 images of healthy and cancerous cases. It can be seen from the table 12 that though all the back propagation neural networks trained (BPNNs) have motivating performance on both the training and test databases, BPNN1 achieved the highest recognition rate on both the training and test data compared to the other networks. Note that BPNN1 uses “Canny” edge detectors which means that this kind of edge detectors was the best in extracting the real features that distinguish the two types of images, which helps in getting the best performance.

Moreover, it is seen that removing the image enhancement techniques from the image analysis part yielded to worse recognition rate; because as seen in table, BPNN3 and BPNN4 achieved the lowest training and testing recognition rates compared to those where the enhancement techniques were used.

Furthermore, it is seen that the Prewitt detector is not effective at all in this medical application. This is due to the training and testing results obtained when it is used. This can prove that this kind of detectors is not efficient in extracting the patterns of the cancer cells in a blood image, which can make the network unable of learning the distinct features that distinguish the healthy and cancerous cells.

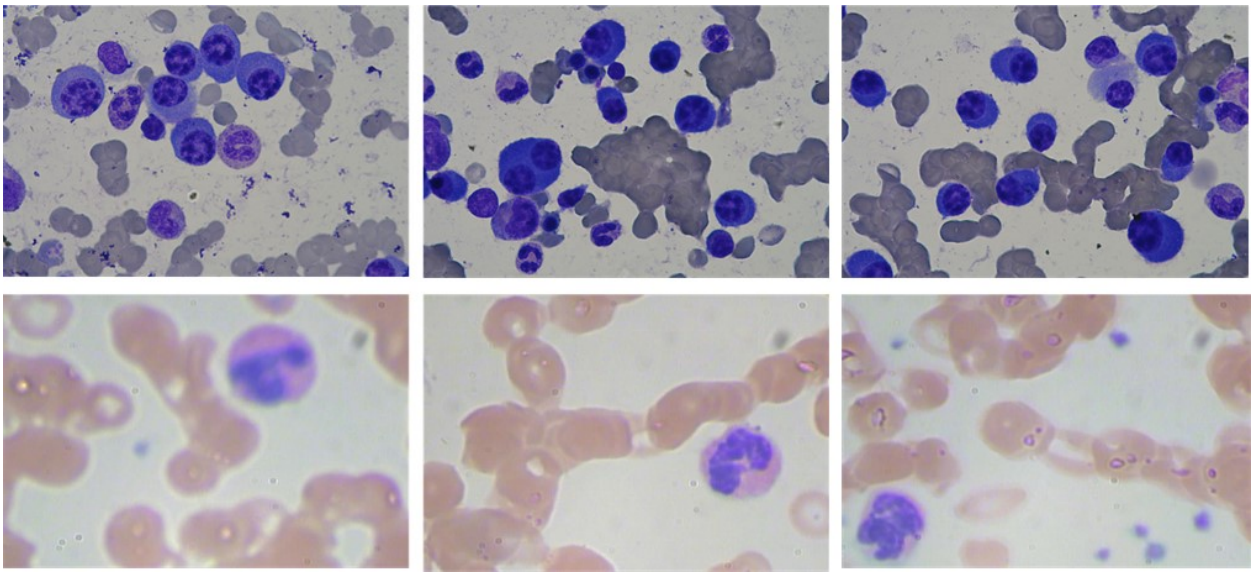

Figure 4.1: Sample of testing images

Figure 4.2 shows some images which were incorrectly classified during the testing of the networks. The network considered these two images as cancerous images however, they are normal blood cells.

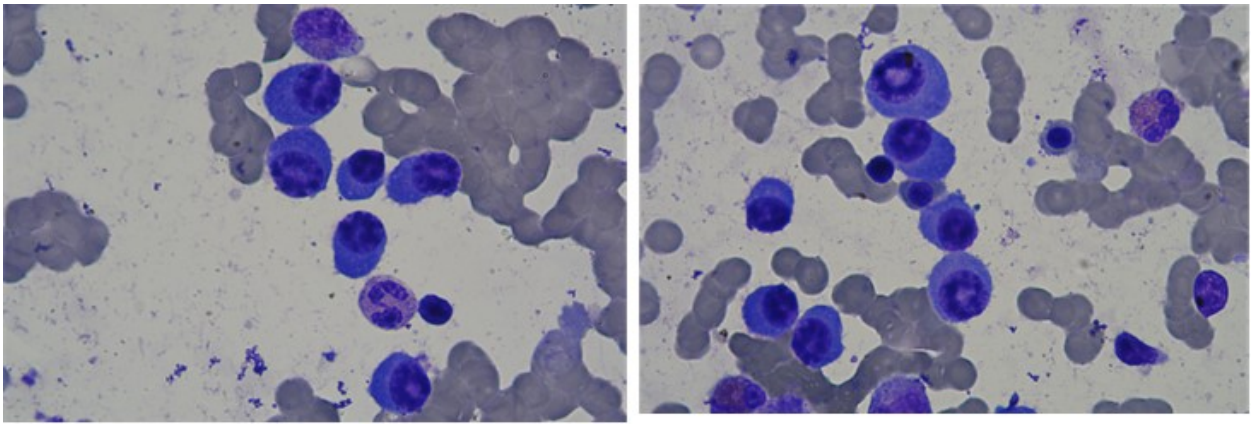

Figure 4.2: Incorrectly classified images

4.2 Results Discussion

In this thesis, myeloid acute leukemia in blood cell images is diagnosed using a neural network that is trained to classify normal and cancerous blood cell images to be capable of diagnosing whether a blood cell image is normal or it has myeloid acute leukemia.

A backpropagation neural network is selected to be used as the intelligent classifier that would perform this classification task. BPNN is a network that uses a supervised learning algorithm where output should be labeled, and error is calculated using the gradient descent.

In the first stage of the work where the images are analyzed, different image enhancement and segmentation are used so that the good features that distinguish normal and cancerous cells are extracted. In this stage image sharpening and histogram equalization are explored for enhancing the quality of images and spotting on the cancer cells in leukemia images. Moreover, edge detection algorithms were also used for segmenting the cells inside images. And here, two algorithms were used for comparison purposes as the one that extracts the best features that distinguish the two classes of images is the one that would help the learning of the neural network and end up with the higher recognition rate.

For more comparison purposes, and to investigate the benefits of using image enhancement techniques we simulated the proposed network on images with and without those techniques.

Table 13 shows the training and testing results of all used models. It shows the mean square error achieved and the iterations needed for the networks to achieve their highest training and testing recognition rates.

Table 13: Performance comparison of all models

Network models	Error	Iterations	Training images (250)	Testing images (70)
BPNN1	**0.0273**	**370**	**95.8%**	**94.2%**
BPNN2	0.0434	112	92.3%	91.2%
BPNN3	0.0510	30	91 %	90.4%
BPNN4	0.0721	119	90.3%	90.21%
BPNN5	0.0916	11	90.13%	90%
BPNN6	0.0916	11	90.13%	90%

As we interpret the Table 13, it is seen that the network that uses "Canny" edge detector in its image processing stage has achieved a better training and testing recognition rates (95.8% & 94.2%) than the one uses "Sobel" operators (92.3% & 91.2%). Note that the BPNN1 has reached a lower error compared to that of BPNN2 but with larger number of iterations.

It also shows the Prewitt detector is not good to be used in this applications since it couldn't extract the good features from both healthy and cancerous images and therefore, the network couldn't learn and the results were very low in both cases: with and without enhancements.

Moreover, it is noted that BPNN3 and BPNN4 achieved a lower recognition rates of 91% and 90.4%, respectively. Note that these recognition rates are lower than those of BPNN1 and BPNN2. This is obviously because of the image enhancement techniques which were removed in those two networks.

Finally it is to be noted that the network that achieved the highest recognition rates during training and testing is BPNN1 which is the one that uses "Canny" edge detector with images sharpening and histogram equalization.

CHAPTER 5
RESULTS DISCUSSION AND CONCLUSION

5.1 Conclusion

Myeloid acute leukemia can be an exceptional example of some medical image diagnosis problems in medicine which involves extraordinary cells and objects. A big number of cells may be generic. Moreover, a myeloid acute leukemia can be derived from a picture, or using neural system strategies. Thus, there is a need for an intelligent system that is trained on large number of different blood cell images till it gains the capability of generalizing the type of any leukemia regardless of its shape, illumination, and translational constraints.

The aim of this thesis is to design a system that is able to read images and train the images to determine the identity of the blood cell inputted from the image. It is pertinent to note that myeloid acute leukemia classification is not an easy task in the field of computer vision due to some factors that deter it such as illumination, rotation, noise and different cells that maybe found in the image. Our system starts by reading the pictures to exclude noise, and then the pictures were segmented using edge detection algorithms. An efficient feature extraction was used to extract only the crucial features before being fed into the neural network for back propagation.

Note that different image analysis schemes were used in order to investigate the efficiency of network for each one. In the first image analysis scheme the input blood cell images are processed using image enhancement and Canny edge detection algorithm and then reduced to 64*64 and this shows a better training and also testing recognition rate in addition to the lesser time and error the network achieved. In contrast, the other analysis scheme which uses Sobel and Prewitt edge detectors and images of size 64*64 has shown a lower recognition rates and longer training time.

In fact, our created acute myeloid blood cells framework is simple when contrasted with other existing frameworks. In brief, the outcomes acquired, after effective tests demonstrates that it is more exact than other existing works. This is because of powerful strategy of fragmenting the

images of blood cells and furthermore the extraction of highlights of the picture which encourages the learning phase of the neural network.

Overall, it is found that a “Canny” edge detector has a great efficiency in distinguishing the normal blood cell images from those which have myeloid leukemia. This was shown to help the network in learning and achieving he highest recognition rates during training and testing compared to that when “Sobel” and “Prewitt” detector was used.

5.2 Recommendations

Conclusively, future recommendation works that are open for exploitation is the support vector machine (SVM) in recognition. It is believed that SVM can improve in eliminating the classifier re-training as against back propagation neural network; this is due to the fact that SVM is a most extreme margin classifier that focalizes to a similar local minima. It shall be noted also that computing with SVM on data with exceptional dimension requires a great amount of time; therefore future reviews should be based on comparisons between back propagation and SVM.

REFERENCES

Acute Myeloid Leukemia (AML) Genomics, (2017). Retrieved June 19, 2009 from http://genome.wustl.edu/projects/detail/acute-myeloid-leukemia-aml-genomics//

Aroquiaraj, I. L., & Thangavel, K. (2013). Mammogram Edge Detection Using Hybrid Soft Computing Methods. *arXiv preprint arXiv:1307.4516*.

Berezsky, O., Batko, Y., Melnyk, G., Verbovyy, S., & Haida, L. (2015, September). Segmentation of cytological and histological images of breast cancer cells. *In Proceeding of the 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (*pp. 287-292). France: Paris.

Dudea, S. M. (2013). Differentiating benign from malignant superficial lymph nodes with sonoelastography. *Medical Ultrasonics*, *15*(2), 132-139.

Huang, C. J., & Liao, W. C. (2004). Application of probabilistic neural networks to the class prediction of leukemia and embryonal tumor of central nervous system. *Neural Processing Letters*, *19*(3), 211-226.

Jaleel, J. A., Salim, S., & Aswin, R. (2012). Artificial neural network based skin cancer detection. *International Journal of Advanced Research in Electrical*, *1*(6), 200-5.

Khashman, A. (2011). Credit risk evaluation using neural networks: Emotional versus conventional models. *Applied Soft Computing, 11*(8), 5477-5484.

Khasman,A. & Al-Zgoul, E. (2010). Image Segmentation of Blood Cells in Leukemia Patients. *Recent Advances in Computer Engineering and Applications*, 2(1), 104-109.

Krenker, A., Kos, A. and Bešter, J., 2011. *Introduction to the artificial neural networks*. New York, NY: INTECH Open Access Publisher.

Mashor M.Y. (2000). Hybrid Multilayered Perceptron Networks. *International Journal of System and Science*, *6*(2), 771-185.

Mashor, M.Y., Osman, M. K., & Arshad, M. R. (2004, July). 3D object recognition using 2D moments and HMLP network. *In Proceeding of the International Conference on Computer Graphics, Imaging and Visualization* (pp. 126-130). Malaysia: Penang.

Mat-Isa, N. A., Mashor, M. Y., & Othman, N. H. (2003). Classification of cervical cancer cells using HMLP network with confidence percentage and confidence level analysis. *International Journal of the Computer*, *11*(1), 17-29.

Mouelhi, A., Sayadi, M., & Fnaiech, F. (2011, March). Automatic segmentation of clustered breast cancer cells using watershed and concave vertex graph. *In Proceeding of the International Conference on Communications, Computing and Control Applications* (pp. 1-6). USA: Chicago.

Panovska-Stavridis, I., Cevreska, L., Trajkova, S., Hadzi-Pecova, L., Trajkov, D., Petlichkovski, A., & Spiroski, M. (2008). Preliminary results of introducing the method multiparameter flow cytometry in patients with acute leukemia in the Republic of Macedonia. *Macedonian Journal of Medical Sciences*, *1*(2), 36-43.

Rojas, R. (1996). *Neural Networks: a systematic introduction*. Springer-Verlag.Davis.

Sahakyan, A., & Sarukhanyan, H. (2012). Segmentation of the breast region in digital mammograms and detection of masses. *International Journal of Advanced Computer Science And Applications*, *3*(2), 102-105.

Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advance Research in Artificial Intelligence, 2*(2), 34-38.

Sharma, N., & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of Medical Physics / Association of Medical Physicists of India*, *35*(1), 3–14.

Vahadane, A., & Sethi, A. (2013, November). Towards generalized nuclear segmentation in histological images. *In Proceeding of the 8th International Conference on Bioinformatics and Bioengineering* (pp. 1-4). USA: Florida.

Withey, D. J., & Koles, Z. J. (2007, October). Medical image segmentation: Methods and software. *In Proceeding of International Conference on Functional Biomedical Imaging* (pp. 140-143). Torronto: Canada.

Wu, Y., Zhao, X., Xie, G., Liang, Y., Wang, W., & Li, Y. (2015, November). Multi-scale Medical Image Segmentation Based on Salient Region Detection. In Proceedings of the *Chinese Conference on Biometric Recognition* (pp. 624-632). China: Tianjin.

Zhang, C., & Wang, P. (2000). A new method of color image segmentation based on intensity and hue clustering. *Pattern Recognition, 2*(7), 5-12.

Zurada, J.M. (1992). *Introduction to artificial neural systems (Vol. 8).* St. Paul: West publishing company.

APPENDICES

APPENDIX 1
BPNN1

```
%

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName)

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');
```

```
% a=a1(:,:, 2);   % only the Green componant

b = imsharpen(f);

% figure, imshow(b),title('Sharpened Image');

%% Enhance contrast using histogram equalization

H = histeq(b);

% figure, imshow(H), title('histogram Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(H, 'sobel');

%%%%%%%%%%%%%%%%%%%%%%%%%%%

  T=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T=T(:);

% concatenate reshaped images horizontally

h=[h T];

end

t1=ones(1,254);

t2=zeros(1,254);

targets1=[t1;t2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%normal

h2=[];
```

```
for k = 1:66

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\normal plasma cell (NPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');

% a=a1(:,:, 2);   % only the Green componant

b = imsharpen(f);

% figure, imshow(b), title('Sharpened Image');

%% Enhance contrast using histogram equalization
```

```
H = histeq(b);

% figure, imshow(H), title('histogram Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(H, 'sobel');

%%%%%%%%%%%%%%%%%%%%%%%%%%%

  T1=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T1=T1(:);

h2=[h2 T1];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

t1=zeros(1,66);

t2=ones(1,66);

targets2=[t1;t2];

inputs=[h(:,1:200),h2(:,1:50)];

targets=[targets1(:,1:200), targets2(:, 1:50)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks
creation

%Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

%
```

```
% This script assumes these variables are defined:
%
% CREATING AND INITIATING THE NETWORK
net = newff(minmax(inputs),[20 2],{'logsig','logsig'},'traingdx');
% TRAINING THE NETWORK
net.trainParam.lr = 0.022;  % Learning Rate.
net.trainParam.show = 300; % Frequency of progress displays (in epochs).
net.trainParam.epochs =1000;% Maximum number of epochs to train.
net.trainParam.mc = 0.7 % Momentum Factor.
[net,tr] = train(net,inputs,targets);
%RECOGNITION RATE OF TRAIN DATA
%target max indices
[M,I_t]=max(targets);% row vector
%dimensions of target matrix
[u,v]=size(targets);
%actual output matrix sim_net
sim_net=sim(net,inputs);
[N,I_sim_net]=max(sim_net);% row vector
%comparison of target and actual outputs
result = I_t==I_sim_net;% row vector
%sum of all elements,1s, to know how many corrects
```

```
corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w


fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)


%

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);
```

```
  baseFileName = jpegFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName)
  fprintf(1, 'Now reading %s\n', fullFileName);
I  = double(imread(fullFileName));
f=rgb2gray(I);
f=imresize(f, [256 256]);
%  figure;imshow(f);title('Grayscale image');
% a=a1(:,:, 2);   % only the Green componant
b = imsharpen(f);
% figure, imshow(b),title('Sharpened Image');
%% Enhance contrast using histogram equalization
H = histeq(b);
% figure, imshow(H), title('histogram Image');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S = edge(H, 'canny', 0.45);
%%%%%%%%%%%%%%%%%%%%%%%%
   T=imresize(S, [64 64]);
%  figure, imshow(T),title('Patterns extracted')
T=T(:);
% concatenate reshaped images horizontally
h=[h T];
```

```
end

t1=ones(1,254);

t2=zeros(1,254);

targets1=[t1;t2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%normal

h2=[];

for k = 1:66

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\normal plasma cell (NPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);
```

```
f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');

% a=a1(:,:, 2);   % only the Green componant

b = imsharpen(f);

% figure, imshow(b), title('Sharpened Image');

%% Enhance contrast using histogram equalization

H = histeq(b);

% figure, imshow(H), title('histogram Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(H, 'canny', 0.15);

%%%%%%%%%%%%%%%%%%%%%%%%%

   T1=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T1=T1(:);

h2=[h2 T1];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

t1=zeros(1,66);

t2=ones(1,66);

targets2=[t1;t2];
```

```
inputs=[h(:,1:200),h2(:,1:50)];

targets=[targets1(:,1:200), targets2(:, 1:50)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks creation

%Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

%

% This script assumes these variables are defined:

%
```

APPENDIX 2

BPNN2

```
% CREATING AND INITIATING THE NETWORK

net = newff(minmax(inputs),[150 2],{'logsig','logsig'},'traingdx');

% TRAINING THE NETWORK

net.trainParam.lr = 0.35;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =2000;% Maximum number of epochs to train.

net.trainParam.mc = 0.75 % Momentum Factor.

[net,tr] = train(net,inputs,targets);

%RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I_t]=max(targets);% row vector

%dimensions of target matrix

[u,v]=size(targets);

%actual output matrix sim_net

sim_net=sim(net,inputs);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects
```

```
corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)

%%%%%%%Canny 91% (train), 90.2%,
```

```
%

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName)

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(f, 'sobel',0.45);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   T=imresize(S, [64 64]);
%  figure, imshow(T),title('Patterns extracted')
T=T(:);
% concatenate reshaped images horizontally
h=[h T];
end
t1=ones(1,254);
t2=zeros(1,254);
targets1=[t1;t2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%normal
h2=[];
for k = 1:66
myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\normal plasma cell (NPC)';
if ~isdir(myFolder)
  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);
  uiwait(warndlg(errorMessage));
  return;
end
filePattern = fullfile(myFolder, '*.jpg');
```

```
jpegFiles = dir(filePattern);
  baseFileName = jpegFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName);
  fprintf(1, 'Now reading %s\n', fullFileName);
I  = double(imread(fullFileName));
f=rgb2gray(I);
f=imresize(f, [256 256]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S = edge(f, 'sobel',0.45);
%%%%%%%%%%%%%%%%%%%%%%%%%
   T1=imresize(S, [64 64]);
%  figure, imshow(T),title('Patterns extracted')
T1=T1(:);
h2=[h2 T1];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
t1=zeros(1,66);
t2=ones(1,66);
targets2=[t1;t2];
inputs=[h(:,1:200),h2(:,1:50)];
```

```
targets=[targets1(:,1:200), targets2(:, 1:50)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks creation

%Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

%

% This script assumes these variables are defined:

%

% CREATING AND INITIATING THE NETWORK

net = newff(minmax(inputs),[20 2],{'logsig','logsig'},'traingdx');

% TRAINING THE NETWORK

net.trainParam.lr = 0.022;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =1000;% Maximum number of epochs to train.

net.trainParam.mc = 0.7 % Momentum Factor.

[net,tr] = train(net,inputs,targets);

%RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I_t]=max(targets);% row vector

%dimensions of target matrix

[u,v]=size(targets);
```

```
%actual output matrix sim_net

sim_net=sim(net,inputs);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)

%

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));
```

```
  return;
end
filePattern = fullfile(myFolder, '*.jpg');
jpegFiles = dir(filePattern);
  baseFileName = jpegFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName)
  fprintf(1, 'Now reading %s\n', fullFileName);
I  = double(imread(fullFileName));
f=rgb2gray(I);
f=imresize(f, [256 256]);
%  figure;imshow(f);title('Grayscale image');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S = edge(f, 'canny', 0.35);
%%%%%%%%%%%%%%%%%%%%%%%%%
   T=imresize(S, [64 64]);
%  figure, imshow(T),title('Patterns extracted')
T=T(:);
% concatenate reshaped images horizontally
h=[h T];
end
t1=ones(1,254);
```

```
t2=zeros(1,254);

targets1=[t1;t2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%normal

h2=[];

for k = 1:66

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\normal plasma cell (NPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
S = edge(f, 'canny', 0.35);

%%%%%%%%%%%%%%%%%%%%%%%%%%%

  T1=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T1=T1(:);

h2=[h2 T1];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

t1=zeros(1,66);

t2=ones(1,66);

targets2=[t1;t2];

inputs=[h(:,1:200),h2(:,1:50)];

targets=[targets1(:,1:200), targets2(:, 1:50)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks
creation

%Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

%

% This script assumes these variables are defined:

%

% CREATING AND INITIATING THE NETWORK
```

```
net = newff(minmax(inputs),[200 2],{'logsig','logsig'},'traingdx');
% TRAINING THE NETWORK
net.trainParam.lr = 0.22;  % Learning Rate.
net.trainParam.show = 300; % Frequency of progress displays (in epochs).
net.trainParam.epochs =2000;% Maximum number of epochs to train.
net.trainParam.mc = 0.53 % Momentum Factor.
[net,tr] = train(net,inputs,targets);
%RECOGNITION RATE OF TRAIN DATA
%target max indices
[M,I_t]=max(targets);% row vector
%dimensions of target matrix
[u,v]=size(targets);
%actual output matrix sim_net
sim_net=sim(net,inputs);
[N,I_sim_net]=max(sim_net);% row vector
%comparison of target and actual outputs
result = I_t==I_sim_net;% row vector
%sum of all elements,1s, to know how many corrects
corrects=sum(result);
%recognition rate,
w=double(corrects*100/v); %let recognition rate be w
```

```
fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%BPNN5

%

clear

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName)

  fprintf(1, 'Now reading %s\n', fullFileName);
```

```
I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');

b = imsharpen(f);

% figure, imshow(b),title('Sharpened Image');

%% Enhance contrast using histogram equalization

H = histeq(b);

% figure, imshow(H), title('histogram Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(H, 'prewitt', 0.73);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   T=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T=T(:);

% concatenate reshaped images horizontally

h=[h T];


end

t1=ones(1,254);
```

```
t2=zeros(1,254);

targets1=[t1;t2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%normal

h2=[];

for k = 1:66

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad jom3a\myeloma\normal plasma cell (NPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');
```

```
% a=a1(:,:, 2);   % only the Green componant

b = imsharpen(f);

% figure, imshow(b), title('Sharpened Image');

%% Enhance contrast using histogram equalization

H = histeq(b);

% figure, imshow(H), title('histogram Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(H, 'prewitt',0.73);

%%%%%%%%%%%%%%%%%%%%%%%%%%%

  T1=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T1=T1(:);

h2=[h2 T1];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

t1=zeros(1,66);

t2=ones(1,66);

targets2=[t1;t2];

inputs=[h(:,1:200),h2(:,1:50)];

targets=[targets1(:,1:200), targets2(:, 1:50)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks creation
%Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
%
% This script assumes these variables are defined:
%
% CREATING AND INITIATING THE NETWORK
 net = newff(minmax(inputs),[200 2],{'logsig','logsig'},'trainscg');
% net = feedforwardnet(100, 'trainlm');
% TRAINING THE NETWORK
net.trainParam.lr = 0.12;  % Learning Rate.
net.trainParam.show = 300; % Frequency of progress displays (in epochs).
net.trainParam.epochs =1000;% Maximum number of epochs to train.
net.trainParam.mc = 0.5 % Momentum Factor.
 [net,tr] = train(net,inputs,targets);
%RECOGNITION RATE OF TRAIN DATA
%target max indices
[M,I_t]=max(targets);% row vector
%dimensions of target matrix
[u,v]=size(targets);
```

```
%actual output matrix sim_net

sim_net=sim(net,inputs);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%BPNN6

%

h=[];

for k = 1:254

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad
jom3a\myeloma\blast plasma cell (BPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);
```

```
  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName)

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);

%  figure;imshow(f);title('Grayscale image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(f, 'prewitt',0.45);

%%%%%%%%%%%%%%%%%%%%%%%%%

   T=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T=T(:);

% concatenate reshaped images horizontally

h=[h T];

end
```

```
t1=ones(1,254);

t2=zeros(1,254);

targets1=[t1;t2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%normal

h2=[];

for k = 1:66

myFolder = 'C:\Users\TOSHIBA\imp. documents\students projects\Mohamad jom3a\Mohamad
jom3a\myeloma\normal plasma cell (NPC)';

if ~isdir(myFolder)

  errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);

  uiwait(warndlg(errorMessage));

  return;

end

filePattern = fullfile(myFolder, '*.jpg');

jpegFiles = dir(filePattern);

  baseFileName = jpegFiles(k).name;

  fullFileName = fullfile(myFolder, baseFileName);

  fprintf(1, 'Now reading %s\n', fullFileName);

I  = double(imread(fullFileName));

f=rgb2gray(I);

f=imresize(f, [256 256]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S = edge(f, 'prewitt',0.45);

%%%%%%%%%%%%%%%%%%%%%%%%%

  T1=imresize(S, [64 64]);

%  figure, imshow(T),title('Patterns extracted')

T1=T1(:);

h2=[h2 T1];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

t1=zeros(1,66);

t2=ones(1,66);

targets2=[t1;t2];

inputs=[h(:,1:200),h2(:,1:50)];

targets=[targets1(:,1:200), targets2(:, 1:50)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Neural networks
creation

%Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

%

% This script assumes these variables are defined:

%
```

APPENDIX 3

BPNN3

```
% CREATING AND INITIATING THE NETWORK

net = newff(minmax(inputs),[200 2],{'logsig','logsig'},'traingdx');

% TRAINING THE NETWORK

net.trainParam.lr = 0.122;  % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =1000;% Maximum number of epochs to train.

net.trainParam.mc = 0.4 % Momentum Factor.

[net,tr] = train(net,inputs,targets);

%RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I_t]=max(targets);% row vector

%dimensions of target matrix

[u,v]=size(targets);

%actual output matrix sim_net

sim_net=sim(net,inputs);

[N,I_sim_net]=max(sim_net);% row vector

%comparison of target and actual outputs

result = I_t==I_sim_net;% row vector

%sum of all elements,1s, to know how many corrects
```

```
corrects=sum(result);

%recognition rate,

w=double(corrects*100/v); %let recognition rate be w

fprintf('train recognition rate is %d\n',w);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%Testing code

Testinputs=[h(:,201:254),h2(:,51:66)];

ActualOut=sim(net,Testinputs)
```