ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

White Blood Cell Classification Using Neural Network Approach

By
Kindalem Sirak

Addis Ababa, Ethiopia
October 2019

ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

White Blood Cell Classification Using Neural Network Approach

By

Kindalem Sirak

A thesis submitted to the School of Graduate Studies of Addis Ababa Institute of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Computer Engineering

Advisor

Dr. Eng. Getachew Alemu

Addis Ababa, Ethiopia

October 2019

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

White Blood Cell Classification Using Neural Network Approach

By

Kindalem Sirak

Approval by Board of Examiners

Dr. Yalemzewd Negash ________________________
Dean, School of Electrical and Signature

Computer Engineering

Dr. Eng. Getachew Alemu ________________________
Advisor Signature

________________________ ________________________
Internal Examiner Signature

________________________ ________________________
External Examiner Signature

Abstract

In medical diagnosis blood test is very essential. For this purpose identifying the white blood cell type and recognizing their number are important and useful measure, which indicates the health status of the body. For the analysis of blood cell, laboratory technicians use manual microscopic evaluation which is extremely time-consuming and tedious to segment and classify white blood cells and on the other side, the instruments which are being utilized by specialists for segmentation and classification of blood cells are not economical and affordable for every doctor or hospital. To overcome this, various computational techniques have been developed for segmentation and classification in recent years with improvements in outcomes.

In this respect, Artificial Neural Network (ANN) provides the ability and potentials to make classification. The aim of this research work is to design and implement for the classification of white blood cell types from microscopic images of blood samples. Therefore, this research focused on the tasks including the segmentation process, extract suitable features, design the classifier and classify them into five types using the designed ANN model.

The system was experimentally analyzed with microscopic images for the classification of the white blood cell types. To acquire region of interest all of microscopic images were segmented. Subsequently, various feature vectors were extracted from the segmented image. After the extraction of feature vectors the classification of each microscopic image for a particular category at the next step was performed using the designed ANN model.

The extracted features were used as an input to the neural network. Three feature sets were used to evaluate and compare the performance of the classifier. Accordingly, the segmentation results show that k-means clustering outperforms Otsu thresholding with an average segmentation accuracy of 91.6% and 88.2% respectively. The designed classifier model also yields a classification accuracy of 93.8% to 96.5% based on extracted features from segmented images. It is understood that this research provides the possibility of increasing the speed to find the results of medical analysis by using ANN especially as the number of blood samples increase.

Key words: classification, microscopic images, neural network, white blood cells

Acknowledgement

First of all, I would like to thank Almighty God and his mother Saint Marry for giving me strength, patience, and stability to go through this thesis.

In the second place, I would like to record my gratitude to my advisor Dr. Eng. Getachew Alemu for his supervision, unlimited support, direction, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. His involvement with his originality has triggered and nourished my intellectual maturity.

I also would like to thank my friends for their valuable support, help, encouragement, and providing me many suggestions during the course of this work.

I would like to express my deepest and everlasting gratitude to my parents for their love, support and encouragement throughout my life. Without them I would not be where I am right now and their contribution towards my thesis is immeasurable.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network
BP	Back Propagation
CGD	Conjugate Gradient Descent
CNN	Convolutional Neural Network
GB	Giga Byte
GHZ	Giga Hertz
GLCM	Gray Level Co-Occurrence Matrix
MLP	Multilayer Perceptron
RAM	Random Access Memory
RBC	Red Blood Cell
RGB	Red Green Blue
ROI	Region of Interest
RQ	Research Question
SLP	Single Layer Perceptron
WBC	White Blood Cell

Chapter One

Introduction

1.1 Background

In the fields of hematology and infectious diseases, classifying different kinds of blood cells can be used as a tool in diagnosis. By counting certain cells' relative frequencies and comparing to what is normal, conclusions can be made about possible blood diseases. The quantity of blood cells plays important role to ensure that whether the person is healthy or not [1].

There are three types of cells in normal human blood: red cells, leukocyte or white blood cells and platelets. Generally, red cells are simple and similar. White blood cells are clinically more important than red cells and many of blood disorders are related to them [1]. Thus, accurate classification of these cells is very important. Leukocytes count is used to determine the presence of an infection in the human body.

While white blood cells contain nucleus and cytoplasm and there are different types of them. White cells are categorized into five groups: neutrophil, eosinophil, basophil, monocyte and lymphocyte [2, 24]. The texture, colour, size and morphology of these cells make differences among these groups.

White blood cells (WBCs), also called leukocytes, are an integral part of the human immune system. These cells fight infections by attacking bacteria, viruses, and other foreign pathogens that invade the body. Having too many or too few white blood cells can indicate a blood disorder.

For performing Differential Blood Count (DBC) the counts of type of WBC is important. No doubt that manual microscopic evaluation is essential when suspicious abnormality found in blood smear [2]. But manual microscopic evaluation of blood sample requires expert lab technicians. Due to the different morphological features of the white blood cells, manual classification of such cells is a cumbersome process, which is time-consuming and susceptible to human error as it is mostly related to the hematologists' experience. This fact actually emphasize a crucial need for fast and automated method for identifying the different blood cells. Analyzers are available to perform differential count (DC) but they are costly and not affordable by remote area hospitals and it cannot detect irregularities in shapes and sizes of cell.

As a result, identifying the WBC type and recognizing their number are of important issue for medical diagnosis and useful measure, which indicates the health status of the body. For example, excess of lymphocytes (above 30000 cells per micro-liter) may indicate a Lymphocytic Leukemia [7].

Therefore, in this research work an automatic segmentation and classification of the five types of leukocytes with the implementation of neural networks has been proposed. Because of that automated methods provide classifying and counting of WBCs which are required to facilitate this process and to reduce the work load on lab technicians. One of the most significant benefits of the automation of visual sample inspection is to aid pathologists to recognize abnormalities in blood samples competently, accurately, and faster.

1.2 Statement of the Problem

Blood test is used as a health indicator for detecting various diseases. WBC differential diagnosis is vital in today's medical activity because it helps in correctly analyzing the conditions of healthy and unhealthy patients. The normality and abnormality conditions of WBC provide hematologists with a great amount of useful data and knowledge about a patient's condition.

In this regard, the identification/classification of the five types of white blood cells provides valuable information about the patient's condition. Instruments which are Complete Blood Count (CBC) and Differential Blood Count (DBC) machines can classify each type of white blood cells, but they are expensive, may not be affordable at every hospital and their maintenance cost is high. Hematologist uses microscope to classify each type of the five white blood cells and this microscopic evaluation can identify and recognize each WBC type in which class it belongs to. However, the main problem arises when large amounts of blood samples are required to be processed by the hematologist.

Even though, the manual microscopic evaluation of blood sample can classify blood cells that may be misclassified by an automated systems and machines, it is tedious and time consuming. That means as the number of the blood samples increase, the identification process for each type tends to human error which is the main disadvantage of manual classification. The time it takes to classify each types and skill required for the classification task also limits the speed and accuracy with which the blood sample can be processed.

The proposed system can improve the strategy to a better way in such a way it automatically segments and classifies the five classes of white blood cell using neural networks. Besides, this thesis work can increase the efficiency of the system by increasing speed to find the results especially when the sample images are more.

1.3 Research Question

The goal of this thesis work is classification of WBC types using artificial neural network by answering the following research questions. The scientific research questions (RQs) are:

RQ1: Can the segmentation result affects the performance of the classifier?

RQ2: Can the combination of shape, intensity and texture features provide a better classification result?

1.4 Objectives

The objectives of the research are as follows:

1.4.1 General Objective

- The main objective of this research is to design and implement a neural network based white blood cell classification system.

1.4.2 Specific Objectives

- To conduct literature review on white blood cell segmentation and classification
- To select appropriate image processing technique to process and segment microscopic images of blood sample

- To extract the features that feed as an input to the neural network
- To design a classifier model by using Artificial Neural Networks
- To classify the five types of white blood cells using the designed classifier model
- To evaluate the accuracy of segmentation techniques and the classifier in the classification of WBCs
- To compare the performance of the classifier using different extracted feature sets

1.5 Scope of the Study

From its superficial view, this work is bounded and focused on the classification of the five types of white blood cells namely neutrophil, eosinophil, basophil, monocyte and lymphocyte using ANN by performing the tasks, segmentation of an image, feature extraction and classification. Three feature sets like shape, intensity and texture features are extracted and used as an input for the classifier. For the classification purpose the use of artificial neural networks which is (MLP-BP) multilayer perceptron with back propagation algorithm was applied.

1.6 Significance of the Study

The importance of this research are:-

- It provides a vital information about a patient's health to the examiner.
- It improves wastage of time that is being done on traditional manual analysis which is the identification of the cells type using microscope.
- It makes easier the medical activity in such a way that it reduces the number of lab technicians as the number of blood samples of patients' increases.

1.7 Methodology

The research plan is performed in four major phases: literature review, image data acquisition, design and implementation, and discussion and result analysis.

I. **Literature Review:** A literature survey was conducted on the area of image processing for every stage of the system. Available books, journals, case studies, previous research works & guidelines were surveyed in order to have a clear understanding of the subject matter.

II. **Image Acquisition:** Images are downloaded from public and free available website that consists dataset of microscopic images of blood samples, specifically designed for the evaluation and the comparison of algorithms for image classification.

III. **Design and Implementation:** This phase deals with designing and conducting experiments for classification of images using MATLAB.

This phase encompasses three typical steps. The first step is segmentation of an image. The second step is the extraction of features of segmented image that are used as an input to the neural network were extracted. Finally, the features collected in step two were given to the classifier model to classify each types of white blood cells.

IV. **Discussion and Result Analysis:** The last phase is concerned with make discussion and results analysis in the classification of white blood cells and evaluating the performance of the classifier model in terms of accuracy based on the achieved results.

Generally, the core thesis work is made up of image data acquisition, segmentation of an image to acquire the region of interest, feature extraction from the segmented image, and finally, classifying the five classes of white blood cells from the inputs of extracted features.

Figure 1 Summary of methodology

1.8 Research Contribution

The ambition to replace human tasks by machines led to invention of different intelligent systems that are capable of processing, analyzing and manipulating information alone. So far, a number of researches have been conducted to classify white blood cell types from microscopic images. But the intention of this research is to fill the gap on the other researches as it is described from the literature review. This paper extends the field of a fully automated white blood cell classification system from microscopic images of blood samples for a better investigation of patients with the following contributions: (1) Selecting an appropriate segmentation technique and make the segmentation task to acquire region of interest (ROI). (2) Extract and identify the most suitable features that are used as an input to the classifier which allow to uniquely identify for each classes and identifying optimal number of features to achieve a good result. (3) Designing the classifier model and make the classification task. (4) Evaluate and compare the performance of the classifier with different feature sets in terms of accuracy.

1.9 Organization of the Thesis

The thesis report is organized in the following manner: In Chapter Two a review of literature for each and every step of the thesis work is presented. In the subsequent chapter, the discussion on the segmentation and feature extraction is presented. The classification and introduction to neural networks is provided in Chapter Four. An in depth discussion of the design and the logical flow of the methods implemented in this work is described in Chapter Five. Chapter Six analyzes the results and discusses the performance of the overall system, while Chapter Seven concludes and gives recommendations for future work.

Chapter Two

Literature Review

In this chapter literature survey of the detailed processes employed in the thesis work is presented. It includes basics of white blood cells, related works with this research and different image segmentation techniques has been reviewed. In recent year, various models have been proposed by many researchers and which are clearly explained below.

2.1 Basics of White Blood Cells

White blood cells serve as the principal actors of the immune system. The five classes of WBCs, or leukocytes, differ in appearance and function. These classes include neutrophils, monocytes, lymphocytes, eosinophils and basophils. WBCs function primarily to protect and defend the body against infectious invaders, including bacteria, viruses, fungi and parasites. A reduced number of white blood cells in the body, called leukopenia, can leave the body vulnerable to wide variety of infectious diseases [24].

The details of each of the classes are as follows: -

1) Neutrophils

Neutrophils are normally the most abundant of the white blood cells. The bone marrow typically produces approximately 100 billion neutrophils daily, and these WBCs normally account for roughly 50 to 70 percent of leukocytes in the bloodstream. Neutrophils play a crucial role in fighting infections, especially bacterial and fungal infections. Their role in viral infections remains an area of active research.

Neutrophils are phagocytes, meaning they surround and engulf invading germs, which are then killed and digested. They also produce microscopic webs, known as neutrophil extracellular traps that ensnare disease-causing germs. Neutrophils migrate through blood vessel walls into the tissues in response to chemical signals triggered when an infectious invasion occurs.

2) Lymphocytes

Lymphocytes are normally the second most abundant type of WBC in the bloodstream, account for roughly 20 to 40 percent. Lymphocytes circulate between the bloodstream and lymph nodes and other lymphatic tissue, such as the spleen. Although they look the same under the microscope, two types of lymphocytes T cells and B cells exist and perform different functions.

3) Monocytes

Monocytes are the largest type of WBC, and normally account for roughly 1 to 10 percent of leukocytes in the bloodstream. They possess receptors that detect chemical signals indicating infection or tissue inflammation, and other receptors that recognize invading organisms. Monocytes are highly versatile immune cells that undergo further maturation known as differentiation that enables them to perform specific functions.

Monocytes normally circulate in the bloodstream for several hours to a few days before migrating into the body tissues, where they differentiate into macrophages. These cells are capable of engulfing and killing infectious organisms, especially bacteria. They also secrete a variety of immune system signaling chemicals, which can either promote or suppress tissue inflammation. Macrophages also interact with T cells to help the immune system recognize and react to infections.

4) Eosinophils

Eosinophils normally account for roughly 1 to 4 percent of the leukocytes circulating in the bloodstream. Eosinophils are particularly important in fighting parasitic infections. Human parasites range in size from single-celled organisms, such as those that cause malaria, to tapeworms that can grow to 16 feet or longer. Eosinophils are also capable of phagocytosing bacteria, although less efficiently than neutrophils and macrophages. These WBCs play a role in certain types of allergic reactions as well. Markedly elevated levels of eosinophils can potentially cause organ and tissue damage, with the heart, lungs, spleen and skin most often affected.

5) Basophils

Basophils are the least common type of WBC, accounting for less than 1 to 2 percent of leukocytes in the bloodstream. These WBCs are less well understood than other types of leukocytes, but research has uncovered some of the functions of these cells. Basophils possess surface receptors that bind a type of antibody responsible for triggering allergic reactions.

Basophils and their tissue counterparts called mast cells release histamine when allergy-specific antibodies on the cell surface encounter a triggering substance. While the release of histamine occurs almost immediately, production and secretion of other immune system signaling chemicals from basophils is believed to perpetuate allergy-induced tissue inflammation and the symptoms associated with it [24].

2.2 Related Works

A. Gautam et al. [2] proposed a method for an automatic classification of leukocytes using morphological features and Naïve Bayes classifier. The technique used for the segmentation of leukocytes was Otsu thresholding. After the segmentation of leukocytes mathematical morphing is used to remove all components other than the WBC and then after the nucleus region was considered for feature extraction stage. The classifier used for the classification of leukocytes was Naïve Bayes classifier and results a classification rate of 88.8%.

Abdul Nasir et al. [3] proposed application of classification of acute leukaemia cells using multilayer perceptron and simplified fuzzy ARTMAP neural networks for classifying the individual WBC as lymphoblast, myeloblast and normal cell based on the extracted features from both acute lymphoblastic leukaemia (ALL) and acute myelogenous leukaemia (AML) blood samples. Two different training algorithms namely Levenberg-Marquardt and Bayesian Regulation algorithms have been employed to train the MLP network. There are a total of 42 input features that consist of the size, shape and colour based features, have been extracted from the segmented WBCs, and used as the neural network inputs for the classification process. The MLP network trained by Bayesian Regulation algorithm has produced the best classification performance with testing accuracy of above 90% for the overall proposed features.

W.Yu et al. [8] worked a research on an automatic classification of leukocytes using deep neural networks. They used different algorithms for comparison of leukocytes' classification. They achieved an overall accuracy of 82.9%, 84.68%, 85.39% 70.53%, 88.5% using the algorithms K-

Nearest Neighbors with k=3, Support Vector Machine, Logistic Regression, Decision Tree and CNN model respectively.

Zhao et al. [9] have proposed automatic detection and classification of white blood cells from microscopic images. They classified all of the five types of leukocytes namely: basophil, neutrophil, eosinophil, monocyte and lymphocyte. The classifier used for their work is convolutional neural network and achieved an accuracy of 92.8%.

In [13] to classify the white blood cell types both shape and texture features were extracted. From both shape and texture features a total of 19 features has been computed. The classifier used for their work was artificial neural network and achieved a classification accuracy of 86%.

Ongun et al. [18] proposed a fully automated classification of blood and bone marrow smears by using different classifiers including neural network based classifiers and Support Vector Machine (SVM), applied morphological pre-processing combined with fuzzy patch labelling to segment WBC from other blood elements. They extracted the morphological features such as shape, color and texture features. They achieved a classification result of 91.03% using SVM and they have applied the multilayer perceptron network trained using conjugate gradient descent (CGD), linear vector quantisation (LVQ) and k-nearest neighbour classifier which produced 89.74%, 83.33%and 80.76% of accuracy, respectively.

Bikhet et al. [19] applied segmentation, feature extraction and classification of the five types of WBC’s presented in peripheral blood. They used Gray image of blood smears. The use of hierarchical thresholdings helps allocate and segment WBC. Different features extracted from each WBC image. The classifier results 90% of classification accuracy from 71 cells sample image. In their work, they used only 71 images. However, in this research more number of images and three type of features extracted to compare the accuracy of the classifier with different feature sets.

Sabino et.al [20] classified five types of white blood cells using shape, color and texture features. Using Neives Baysian classifier they achieved 89.1% classification rate.

2.3 Summary of Literature Review

The following table summarizes the above mentioned papers.

Table 1 Summery of Literatures

Reference	Classification objective	Feature extraction	Classifier	Accuracy
Ref[2]	Classification of Leukocytes	A number of features extracted	Naïve Bayes	80.88%
Ref[3]	Classifying WBC as lymphoblast, myoblast and normal cell	42 input features are extracted	MLP network trained by Bayesian Regulation	96%
Ref[8]	Classification of Leukocytes	A number of features extracted	K-Nearest Neighbors (k=3), SVM, Logistic Regression, Decision Tree and CNN model	82.9%, 84.68%, 85.39% 70.53%, 88.5%
Ref[9]	Classification of Leukocytes	A number of features extracted	CNN	92.8%
Ref[13]	Five types of WBC	19	Neural network	86%
Ref [18]	Classifying WBC as lymphoblast, myoblast and normal cell	57 input features used as an input	SVM, MLP trained using (CGD), LVQ and K-NN	91.03% 89.74%, 83.33% 80.76%
Ref [19]	Five types of WBC	More number of features extracted	Neural network	90%
Ref [20]	Five types of WBC	More number of features extracted	Naïve Bayesian classifier	89.1%

As shown from the above table different type of classifies have been used and different classification rate achieved. From reference [2] the accuracy achieved by Naive Bayes classifier is poor. The results obtained from reference [3, 18] used more number of features for the purpose of classification with an accuracy of 96% and 91.03% respectively. According to reference [3], they classified WBC as lymphoblast, myoblast and normal cell using 42 input features for an accuracy of 96%. In paper [8] they have used five classifiers for the classification of leukocytes with a maximum accuracy of 88.5% using CNN model.

In reference [18] the researchers have used four classifiers using 57 input features to classify WBC as lymphoblast, myoblast and normal cell but the maximum accuracy is 91.03% using Support Vector Machine. However, for the other papers the exact number of extracted features are not disclosed.

In this research work the classification objective is to classify five types of white blood cells as neutrophils, basophils, eosinophils, lymphocytes and monocytes using a MLP-BP (multi-layer perceptron with back propagation) algorithm has been proposed.

In the proposed method of this thesis not only shape features but also intensity and texture based features are also considered to be extracted for better classification result. And an automatic segmentation followed by feature extraction and classification of white blood cell types using neural networks is also included. Meanwhile, the performance of the classifier is also compared with the three feature sets and significant features are identified and also none of the above researchers didn’t determine significant features for their classification.

Chapter Three

Segmentation and Feature Extraction

3.1 Introduction

The main goal of this research work is classify the white blood cell types which are neutrophil, basophil, eosinophil, lymphocyte and monocyte by the analysis of microscopic images, in order to provide a fully automatic procedure to support the medical activity.

Thus, in order to classify these cells the primary stage is that segmentation of an image. White blood cells also called as Leukocytes are easily identifiable from microscopic images, as their nuclei appear darker than the background. The subsequent stage after segmentation is feature extraction which allows to uniquely identify each classes of WBCs and fed in to the neural network as an input. The last stage is the classification stage which is the recognition part of the system.

3.2 Segmentation

Segmentation is the process of subdividing an image into constituent parts, or isolating certain aspects of an image such as finding lines, circles or particular shapes in an image. It refers to the operation of partitioning an image into component parts, or into separate objects. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. In general, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

Image segmentation is one of the most important steps in image partitioning and their analyses. It can be used for various applications in computer vision and digital image processing. Many of the applications require highly accurate and computationally faster image processing algorithms. The success of any application depends on reliability and accuracy of the image processing used.

Image segmentation involves separating an image into regions corresponding to objects; it involves selecting only the area of interest that is WBC in an image from the other image components. This is done by removing all the unwanted area of an image those are RBC, platelets and stains in an image and preserving only the required area.

In this work the two most common and popular segmentation techniques threshold based and clustering based segmentation are used.

3.2.1 Threshold Based Segmentation

Thresholding is one of the simplest approaches for image segmentation based on intensity levels. Threshold based technique works on the assumption that the pixels falling in certain range of intensity values represents one class and remaining pixels in the image represents the other class. For global thresholding brightness threshold value is to be selected to segment the image into object and background. It generates binary image from given input image. Thresholding creates binary images from gray-level ones by turning all pixels below some threshold value to zero and all pixels about that threshold value to one. That means the pixels satisfying threshold test are considered as object pixels with binary value ‘1’ and other pixels are treated as background pixels with binary value ‘0’.

Selection of threshold is very crucial in image segmentation process. Threshold value can be determined either by an interactive way or can be the outcome of automatic threshold selection method. N. Otsu method is optimal for thresholding large objects from the background [21]. Threshold based approaches are computationally inexpensive fast and can be used for real time applications. A single global threshold partitions image into objects and background, but objects may have different characteristic grey value. In such situations multiple threshold values are needed, for applying over different areas of the image.

As all images were in RGB format, green channel isolation simply involved taking the green layer of the matrix containing the image, returning a grayscale image based on the green intensity of the original image. The other planes such as red and blue are not considered because they contain less information about the image. And the nucleus region is darker from the green channel.

Otsu's Thresholding

Thresholding is used to extract an object from its background by assigning an intensity value T (threshold) for each pixel such that each pixel is either classified as an object point or a background point.

Image thresholding classifies pixels into two categories: The first one is those to which some property measured from the image falls below a threshold, and those at which the property equals or exceeds a threshold. The second one is thresholding creates a binary image which called binarization.

Otsu's thresholding chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels. Based on the idea that finds the threshold that *minimizes the weighted within-class variance.*

From here, it can be implemented in matlab using a built in function called graythresh. Otsu's method is one of the better threshold selection methods for general real world images with regard to uniformity and shape measures [40].

3.2.2 Clustering based segmentation

Clustering is a method to divide a set of data into a specific number of groups. And one of the popular method is k-means clustering. ***K*-Means clustering** algorithm is an unsupervised algorithm and it is used to segment the interest area from the background [41]. It clusters, or partitions the given data into K-clusters or parts based on the K-centroids. It classifies a given set of data into *k* number of disjoint cluster. *K*-means algorithm consists of two separate phases. In the first phase it calculates the *k* centroid and in the second phase it takes each point to the cluster which has nearest centroid from the respective data point. The distance of the nearest centroid can be calculated using Euclidean distance. Once the grouping is done it recalculate the new centroid of each cluster and based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster which have minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. So *K*-means is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, over all clusters.

Let us consider an image with resolution of $x \times y$ and the image has to be cluster into k number of cluster. Let $p(x, y)$ be an input pixels to be cluster and c_k be the cluster centers. The algorithm for k-means [41] clustering is following as:

1. Initialize number of cluster k and center.
2. For each pixel of an image, calculate the Euclidean distance d, between the center and each pixel of an image using the relation given below.

$$d = \|p(x, y) - c_k\| \tag{3.1}$$

3. Assign all the pixels to the nearest center based on distance d.
4. After all pixels have been assigned, recalculate new position of the center using the relation given below.

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y) \tag{3.2}$$

5. Repeat the process until it satisfies the tolerance or error value.

3.3 Feature Extraction

Feature extraction is a kind of dimensionality reduction to efficiently represent the major attributes of an image that are useful for effective classification of white blood cell types.

To recognize or classify an object in an image, one must first extract some features out of the image, and then use these features inside a pattern classifier to obtain the final class [16]. Feature extraction is the process of capturing essential characteristics. It is the process of transforming the input data into set of features/descriptors. Feature extraction reduces the amount of data by extracting the relevant information. The main challenge in feature extraction is to identify the characteristic features that can form a good representation of the object, so as to discriminate across the object category. The criteria for selecting a good feature are; features should be easily computable, preferably independent of translation and size, and chosen so that they do not replicate each other.

Feature extraction aims to locate significant feature regions on images depending on their intrinsic characteristics and applications. These regions can be defined in global or local neighborhood and distinguished by shapes, textures, sizes, intensities, statistical properties, and so on.

To classify each class of WBCs, features that represent their images have to be obtained. Those features have to be capable of distinguishing between WBCs and it is also desired to have as few features as possible. Although the classification accuracy is dependent on the classifier, the performance of the classifier is totally dependent on having adequate features.

Those extracted features provide a means to identify and classify each type of WBCs.

The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements.

These features can be grouped into shape features, intensity features, and texture features.

3.3.1 Shape Based Features

These shape based features are geometrical shapes descriptor and characterized by the following parameters.

Area: - Area of the object is scalar and is calculated by counting the actual number of pixels in the region of interest or the number of pixels on the interior of the cell.

$$Area = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij} \tag{3.3}$$

Where, b_{ij} is the value of binary image (0, 1) at the pixel coordinate (i, j) within an m × n image.

Perimeter: - It is an important feature of an object. Perimeter is the total distance between consecutive points of the border. If $x_1 \ldots x_n$ is a boundary coordinate list, then the object perimeter is given by:

$$Perimeter = \sum_{i=1}^{N} \left| x_i - x_{(i+1)} \right| \tag{3.4}$$

Eccentricity: - Eccentricity of the object is a scalar between (0 and 1), which describes the degree of similarity to a circle. If the object shape is very similar to a circle shape, then the object will have an eccentricity of 0. If the object shape is very similar to a line, it will have an eccentricity of 1.

3.3.2 Intensity Based Features

These features are based only on the absolute value of the intensity measurements in the image. A histogram describes the occurrence relative frequency of the intensity values of the pixels in an image. The intensity features that will be considered are the first four central moments of this histogram: mean, standard deviation, skewness, and kurtosis.

Mean: - For a grayscale image, the mean of the white blood cell image is equal to the average brightness or intensity and it is given by:

$$\bar{\mathrm{X}} = \frac{\mathrm{X1+X2+\ldots Xn}}{\mathrm{N}} = \sum_{i=1}^{N} \frac{Xi}{N} \tag{3.5}$$

Where, $\bar{\mathrm{X}}$ is the mean, N number of pixels, X1…Xn are the grayscale image data.

Variance: - The image variance, gives an estimate of the spread of pixel values around the image mean. And it is given by:

$$\sigma^2 i = \frac{1}{N}\sum_{j=1}^{N} (f_{ij} - \mu_i)^2 \tag{3.6}$$

Where, f_{ij} is the value of the ith colour components of the image pixel j, N is the number of features over all database, μ_i is the mean of the colour i.

Skewness: - The skewness measures the symmetry about the mean. It is calculated by:

$$skewness = \frac{1}{N}\left(\frac{\sum_{\mathrm{i=1}}^{\mathrm{N}} (Xi - \bar{X})^3}{\delta^3}\right) \tag{3.7}$$

Kurtosis: - The kurtosis is a measure of whether the data are peaked or flattened, relative to a normal distribution and can be computed as:

$$kurtosis = \frac{1}{N}\left(\frac{\sum_{\mathrm{i=1}}^{\mathrm{N}} (Xi - \overline{X})^4}{\delta^4}\right) - 3 \tag{3.8}$$

3.3.3 Texture Based Features

These features contain information about the spatial distribution of tonal variations within a band. The identification of specific textures in an image is achieved primarily by modelling texture as a two-dimensional gray level variation. This two dimensional array is called gray level co-occurrence matrix (GLCM). Co-occurrence matrix which is a tabulation of how often different combinations of pixel values occur in an image.

Contrast: - Contrast measures the difference in grayscale intensity between adjacent pixels over the entire image. The greater the difference in intensity values, the higher the value of contrast.

$$contrast = \sum_{i,j} |i - j|^2 \ p(i, j) \tag{3.9}$$

Where, *i* and *j* are the horizontal and vertical cell coordinates and *p* is the cell value.

Homogeneity: - Homogeneity measures the distances of GLCM elements from the GLCM diagonal. Homogeneity ranges from 0 to 1. If adjacent pixels always have very similar values of grayscale intensity, the homogeneity will be close to 1.

$$Homogeneity = \sum_{i,j} \frac{1}{1 - (i - j)^2} \ p(i, j) \tag{3.10}$$

Entropy: - Entropy is a measure of the randomness of grayscale intensity values of pixels. Entropy is based on the grayscale histogram of the image. The histogram can be created from the GLCM by summing across the rows to find the total number of pixels p (i) for each grayscale intensity value i.

$$Entrophy = -\sum_{i,j=0}^{Ng-1} p(i,j)\log\left(p(i,j)\right) \qquad (3.11)$$

Where, *P(i, j)* is the (*i j*)th entry of the normalized co-occurrence matrix, Ng is the number of the gray levels of the white blood cell image.

Correlation: - it is the statistical measure of how correlated a pixel is to its neighbor the whole image and calculates the correlation matrix for the linear model parameters and it can be given by the following equation.

$$correlation = \frac{\sum_{i=0}^{Ng-1}\sum_{j=0}^{Ng-1}(i,j)\ p(i,j) - \mu_x\mu_y}{\delta i\ \delta j} \qquad (3.12)$$

From the above equation,

$$\mu_i = \sum_i i \sum_{i,j} p(i,j) \qquad and \qquad \mu_j = \sum_j j \sum_{i,j} p(i,j)$$

$$\delta i = \sum_{i,j}(i-\mu_i)^2 \sum_{i,j} p(i,j) \qquad and \qquad \delta j = \sum_{i,j}(j-\mu_j)^2 \sum_{i,j} p(i,j)$$

where, μ_x, μ_y, δ_x, and δ_y are the means and standard deviations of the marginal probabilities Px(i) and Py(j) obtained by summing up the rows or the columns of matrix Pij (co-occurrence matrix), respectively.

Energy: - is the summation of squared elements in the GLCM and it ranges from 0 to 1. And it is given by the following equation as follow:

$$Energy = \sum_{i,j} p(i,j)^2 \qquad (3.13)$$

Where, *P (i, j)* is the (*i j*)th entry of the normalized co-occurrence matrix.

Chapter Four

Classification

Classification is the problem of identifying to which of a set of categories (sub-population) a new observation belongs, on the basis of a training set of data. It also refers to the process of assigning a given input data into one of a given number of categories or it is the process of identifying each white blood cell type and assigning it to the correct white blood cell class. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. Classification compares feature vectors to the various models and find the closest match.

In this work, neural network is used to classify the features extracted from the white blood cells. The network is trained using the backpropagation algorithm. The MLP uses a non-linear function, in the hidden layers, and appropriate weights between the layers so that a particular input leads to a specific target output. The number of neurons in the first layer is as same as the dimension of the input features. The number of neurons in the last layer depends on the number of classes. More detail description of the classifier is stated as follows.

4.1 Introduction to Artificial Neural Network

Artificial Neural Networks (ANN) are models that attempt to mimic some of the basic information processing methods found in the human brain. As our brains perform complex tasks, ANN modelled after the brain has also been found useful in solving complex problems [35]. Also an ANN is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use [36].

ANN consists of sets of input layer, hidden layer, output layer, each layer consists of a number of neurons and weighting functions, as shown in Figure 9. The artificial neurons are organized in layers with one or more intermediate hidden layers put in between the input layer and output layer, and send their signals “forward”. Each layer has a number of neurons connected with neurons in the adjacent layers through unidirectional connections. The information flow is only allowed in one direction during the training process that is from the input layer to the output layer through the hidden layers. There can be any number of hidden layers in the architecture. The hidden layer has a synaptic weighting matrix and the weights are associated with all the connections made from the input layer to the hidden layer.

4.2 From Biological Neuron to Artificial Neuron

Biological neuron: In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites* (Figure 2) [33]. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

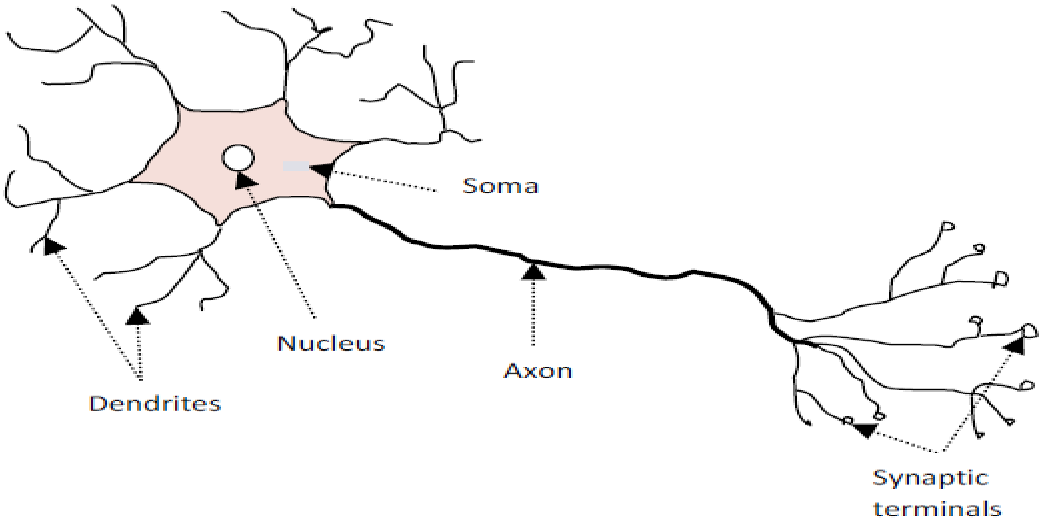


Figure 2 Components of a Biological Neuron [33]

Artificial neuron: An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the testing mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the testing mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not. Consider x, w represents inputs and weights respectively.

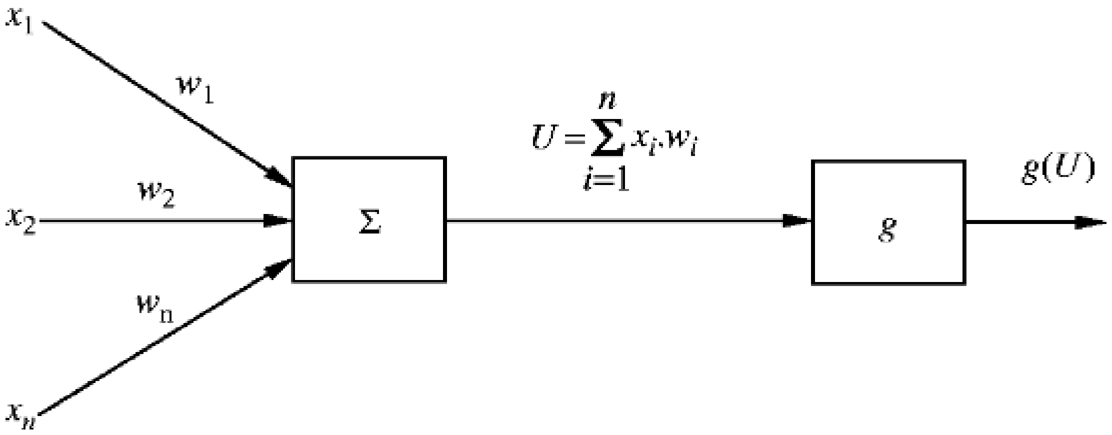


Figure 3 Artificial neuron [33]

4.3 Components of Neural Network

A neural network is made up of a large number of simple processing elements called neurons, units, cells, or nodes. Each neuron is linked to other neurons by the techniques of directed communication links, each with the associated weight. The weights illustrate information being used by the net to solve a problem. Each neuron has an internal state, called its activation function, which is a function of the inputs that it has received. Normally, a neuron sends its activation as a signal to several other neurons. It is important to note that a neuron can send only one signal at a time, although that signal is broadcasted to several other neurons [17].

4.3.1 Neuron

A neuron is an information processing unit that is fundamental to the operation of a neural network. The fundamental processing element of a neural network is a neuron. An artificial neuron is a mathematical function conceived as a simple model of a real (biological) neuron. Neurons are the basic computing unit that performs on the local data being processed inside a network. These neurons form massively parallel networks, whose function is determined by the network structure [35]. A neuron is often called a node or unit. It receives input from some other nodes by inputs, or perhaps from an external source. Each input has an associated weight (*w*). The node computes some function (*f*) of the weighted sum of its inputs. Its output, in turn, can serve as input to other units.

4.3.2 Weight

ANN consists of two or more layers, each layer contains a number of nodes (neurons), as shown as Figure 4. The neurons are joined by directed arcs-connections. The neurons and arcs constitute the network topology. Each arc has numerical weight that specifies the influence between two neurons. Positive weights indicate reinforcement; negative weights represent inhibition.

4.3.3 Activation Function

Over the years, researches have tried several functions to convert the input into an output [17]. The neural activation functions have some important characteristics that make them vital to neural information processing. They are nonlinear, continuous functions that remain within some upper and lower bounds. Nonlinear means that the output of the function varies nonlinearly with the input [35].

Activation function does the final mapping of the activation of the output neurons into the network outputs. But, the outputs from a single cycle of the operation of neural network might not be the final outputs, as the network has to comply with the convergence criterion [35]. The output of any neuron is thus a result of thresholding or its internal activation, if any. Activation function is important for multi-layer networks to preserve meaningful range areas of each layer's operations. From figure 4 $a_{1.....}a_n$ are inputs and w is weight.

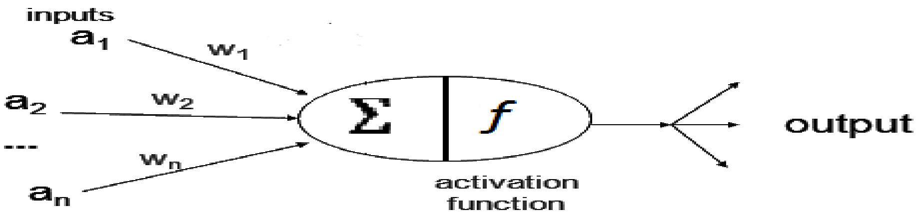


Figure 4 Activation function [36]

The activation function, denoted by *f* (Σ), as shown in Figure 4, defines the output of a neuron in terms of the induced local field Σ, w represents for weight and a represents inputs. There are three basic types of activation function:

4.3.3.1 Threshold Function

A threshold (hard-limiter) activation function can either a binary type or a bipolar type, as shown in Figure 5. Neuron with hard-limiter activation function is known as the McCulloch-Pitts model.

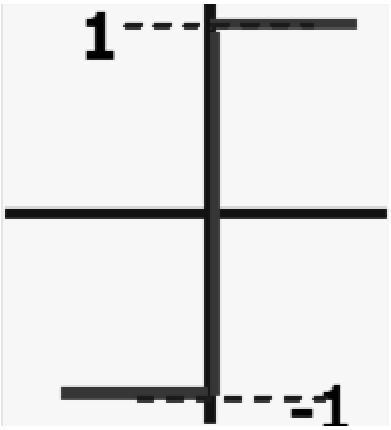


Figure 5 Threshold function [36]

4.3.3.2 Piecewise Linear Function

This activation function is also known as saturating linear function and can have either a binary or bipolar range for the saturation limits of the output. The mathematical model representing a symmetric saturation function is shown in Figure 6.

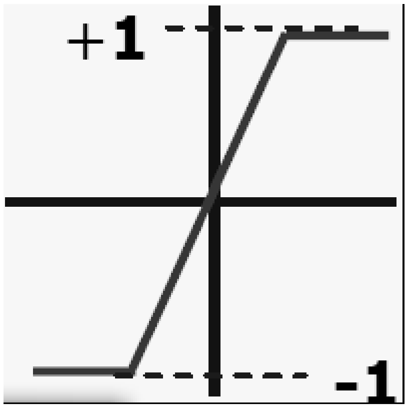


Figure 6 Piecewise Linear Function [36]

4.3.3.3 Sigmoidal Function (S-shape function)

The nonlinear curved S-shape function is commonly known as the sigmoid function, as shown in Figure 7. This is the most widely type of activation used to construct the neural networks. It performs mathematically well with a differentiable and a strictly increasing function.

The sigmoidal function is reached by using the exponential equation. It can be obtained by changing the different shapes of the function, which adjusts to the abruptness of the function as it changes between the two asymptotic values.

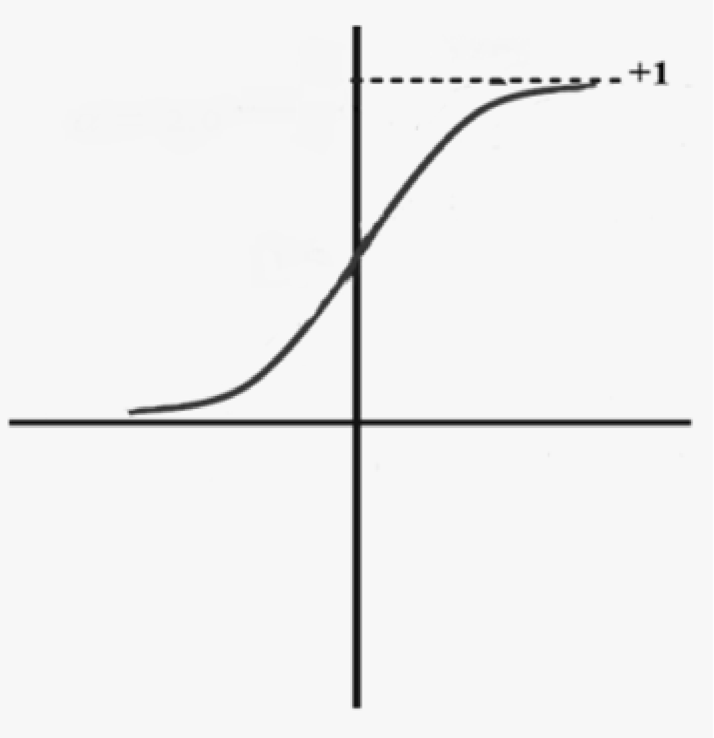


Figure 7 Sigmoidal Function [36]

4.4 Neural Network Architectures

There are three fundamental different classes of network architectures [27].

1) Single-layer Feed forward Networks:

In a layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network, there is an input layer of source nodes that projects onto an output layer of neurons, but not vice versa. This network is strictly a feed forward type. In single-layer network, there is only one input and one output layer. Input layer is not counted as a layer since no mathematical calculations take place at this layer. On other words a feed forward network is that data can flow only in one direction from the input layer through the hidden layers and finally to the output layer. There are no feedbacks connections presented in this type (in which connections are extended from the output layer into the input layer).

This type of networks can perform pattern classification only on linearly separable patterns, regardless of the form of nonlinearity used. Linear separation requires that the pattern is classified to be sufficiently separated from each other to ensure that the decision boundaries are hyper planes.

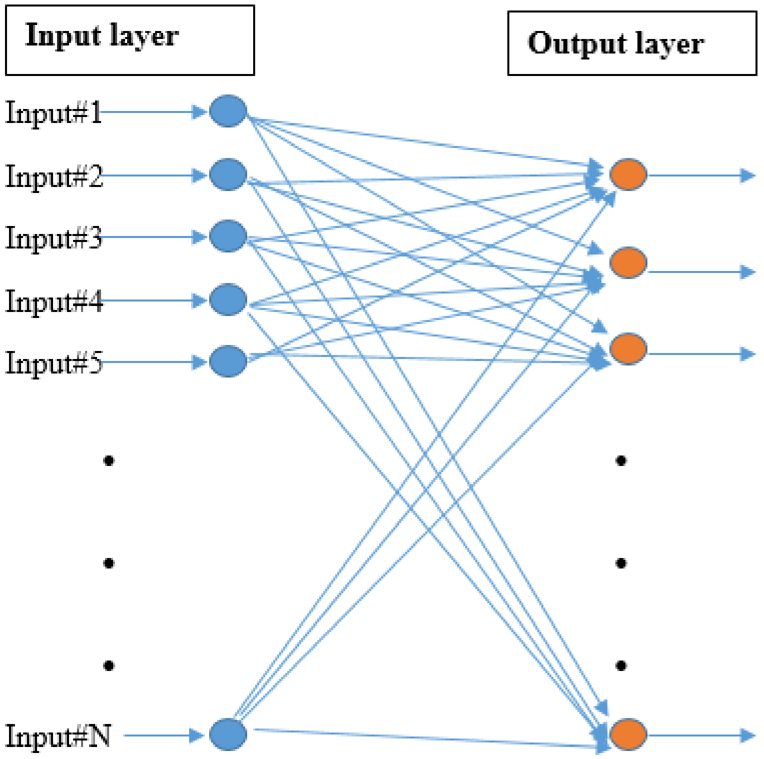


Figure 8 Single layer perceptron

2) Multilayer Feed forward Networks

The second type of a feed forward neural network is known by the presence of one or more hidden layers, whose computational nodes are correspondingly called hidden neurons. The prepuce of the hidden neuron is to add up more computation between the input layer and the output one of cores in some useful manner (e.g. improves the accuracy of the used network, enable the network to do more difficult operations and so on…). By having more hidden layers added, the network is enabled to comprehend more input data and extract higher order statistics.

The input signal is applied to the neurons in the second layer. The output signal of second layer is used as inputs to the third layer, and so on for the rest of the network.

MLP can solve more complicated problems than can SLP. However, training might be more difficult. Nevertheless, in some cases, training may be more successful, because it is possible to solve a problem that SLP cannot be trained to perform correctly at all [17].

The typical MLP architecture is shown from the figure below and it consists of an input layer, hidden layer and output layer.

Input Layer: A layer of neurons that receives information from external sources, and passes this information to the network for processing. These may be either sensory inputs or signals from other systems outside the one being modeled.

Hidden Layer: A layer of neurons that receives information from the input layer and processes them in a hidden way. It has no direct connections to the outside world (inputs or outputs). All connections from the hidden layer are to other layers within the system.

Output Layer: A layer of neurons that receives processed information and sends output signals out of the system.

Bias: Acts on a neuron like an offset. The function of the bias is to provide a threshold for the activation of neurons. The bias input is connected to each of the hidden and output neurons in a network.

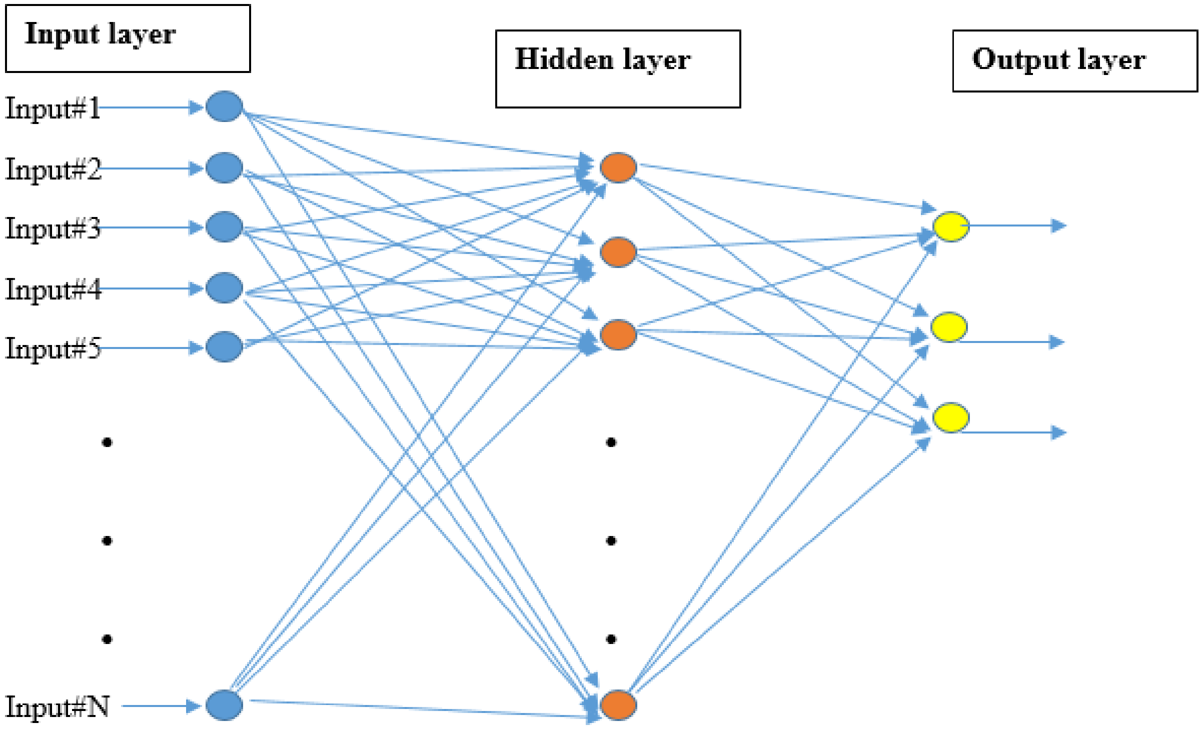


Figure 9 Multilayer perceptron

3) Recurrent networks

A recurrent neural network has at least one feedback loop. A recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons. Self-feedback refers to a situation where the output of a neuron is fed back into its own input. The presence of feedback loops has a profound impact on the learning capability of the network and on its performance. On other words it can be said that the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore.

4.5 Learning Processes

A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs.

Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a *priori* knowledge. Another way is to “train” the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. By learning rule we mean a procedure for modifying the weights and biases of a network. The purpose of learning rule is to train the network to perform some task (e.g. white blood cell classification). They fall into three broad categories:

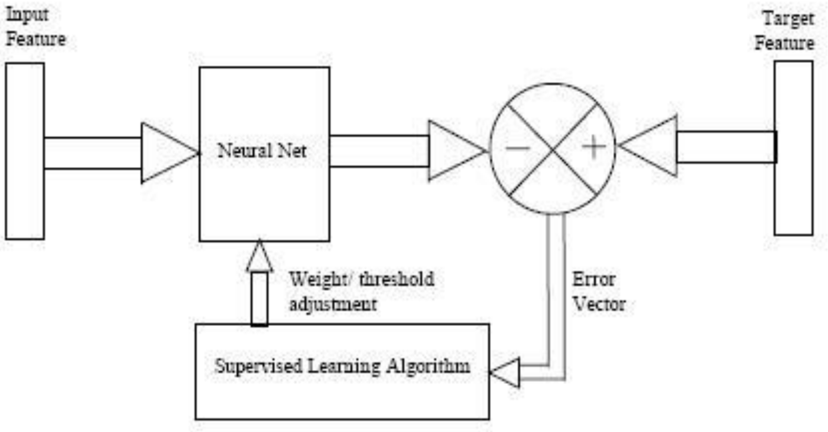


Figure 10 Learning process [34]

1. Supervised learning:

It is also called Associative learning the learning rule is provided with a set of training data of proper network behavior. As the inputs are applied to the network, the network outputs are compared to the targets. The inputs are applied either by an external teacher (the trainer) or by the system which contains the network (self-supervised). The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets.

2. Reinforcement learning:

It is similar to supervised learning, except that, instead of being provided with the correct output for each network input, the algorithm is only given a grade. The grade is a measure of the network performance over some sequence of inputs. Whenever the results (output) is getting a good grade (e.g. 1 means the output is the same as the measured value, 0 means the predicted output has nothing to do with the expected one) then the prediction is improves.

3. Unsupervised learning:

Also called self-organization in which the weights and biases are modified in response to network inputs only. There are no target outputs available. Most of these algorithms perform some kind of clustering operation. They learn to categorize the input patterns into a finite number of classes. In this type the network is trained in a way to respond to clusters of pattern within the input. In this case the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning type, there is no a *priori* set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

4.6 Back-propagation Neural Networks

The feed forward back propagation neural network, which is a very popular model in biological and biomedical applications, is used [13-15]. It sends a signal from the output of the network back down towards the input. This signal will encode how changing a weight will propagate to the output. The back-propagation algorithm is used in layered feed-forward ANNs [33]. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the *input layer*, and the output of the network is given by the neurons on an *output layer*. There may be one or more intermediate *hidden layers*.

This type of neural network configuration does not have feedback connections, but errors are propagated back during training using least mean squared error. The back propagation neural network is a multi-layer, feed-forward supervised learning, which requires pairs of input and target vectors. They often have hidden layers of sigmoid neurons followed by an output layer of linear neurons.

Back-propagation is commonly used in image processing applications because the problem is extremely complex (many features) but has a clear solution (certain number of classes) [14, 15]. Back-propagation works in small iterative steps. The input matrix is applied to the network, and the network produces an initial output based on the current state of its weights (before training the output will be random since these weights are random). This output is compared to the target matrix, and a mean-squared error signal is calculated.

The error value is propagated backwards through the network and small changes are made to the weights in each layer to reduce this error. The whole process is repeated for each training white blood cell image then back to the first case. This process is iterated until the error no longer changes.

The back propagation (BP) learning algorithm

The BP learning process can be described as follows:

(1) Forward propagation of operating signal: the input signal is propagated from the input layer, via the hide layer, to the output layer. During the forward propagation of operating signal, the weight value and offset value of the network are maintained constant and the status of each layer of neuron will only exert an effect on that of next layer of neuron. In case that the expected output cannot be achieved in the output layer, it can be switched into the back propagation of error signal.

(2) Back propagation of error signal: the difference between the real output and expect output of the network is defined as the error signal; in the back propagation of error signal, the error signal is propagated from the output end to the input layer in a layer-by-layer manner. During the back propagation of error signal, the weight value of network is regulated by the error feedback. The continuous modification of weight value and offset value is applied to make the real output of network closer to the expected one [13].

The backpropagation algorithm can be defined mathematically as follows. For a test set, propagate one test through the MLP in order to calculate the output using the formula [13].

$$h_i = f \Sigma x_i w_{ij} \tag{4.1}$$

$$yi = f \Sigma h_i w_{jk} \tag{4.2}$$

Where *h* is the hidden node

x is the input node

w is the weight

y is the output node

Then compute the error, which will be the difference of the expected value ***t*** and the actual value, and compute the error information term ***δ*** for both of output and hidden nodes.

$$\delta y_i = y_i\,(1 - y_i).\,(t - y_i) \tag{4.3}$$

$$\delta h_i = h_i\,(1 - h_i)\,.\,\delta y_i.\,w_{jk} \tag{4.4}$$

δj the information error of the nodes

Finally, back propagate this error through the network by adjusting all of the weights; starting from the weights to the output layer and ending at the weights to the input layer, as shown in Figure 11.

$$\Delta w_{jk} = \eta.\,\delta y_i.\,h_i \tag{4.5}$$

$$\Delta w_{ij} = \eta.\,\delta h_i.\,x_i \tag{4.6}$$

$$w_{new} = \Delta w + w_{old} \tag{4.7}$$

Where ***η*** is the learning rate

Backpropagation adjusts the weights in an amount proportional to the error for the given unit (hidden or output) multiplied by the weight and its input. The training process continues until some termination criterion, such as a predefined mean-squared error, or a maximum number of epochs.

The backpropagation algorithm is a typical supervised learning algorithm, where inputs are provided and propagated forward to generate one or more outputs. Given the output, the error is calculated using the expected output. The error is then used to adjust the weights. There are two types of error functions for backpropagation. The first error function is used for output cells, and the second is used on for hidden cells. The back-propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated.

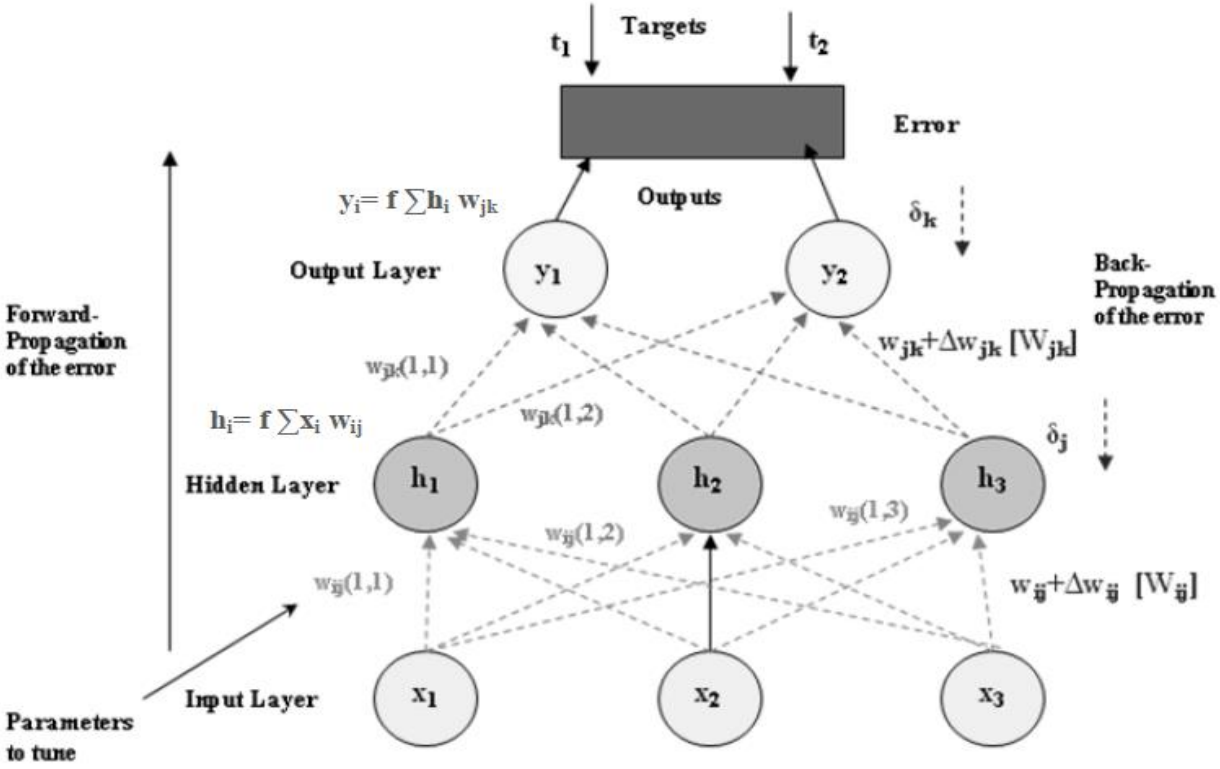


Figure 11 Backpropagation neural network [13]

The learning algorithm can be summarized as follows:

- Initialize the weights of networks.
- Choose a random training sample, and assign input vector to the input neurons.
- Propagate all neurons in the forward direction to obtain output at the output layer.
- Evaluate error values at each neuron in the output layer as the difference between obtained output and the desired output of the training sample chosen.
- Evaluate Mean Squared Error value. This value reflects the effectiveness of training done so far.
- Back propagate the errors, all the way up to the input layer.
- Calculate delta (weight update) for all synapses.
- Update the weights all synapses such that the sum-squared value of error is minimized.
- Now, choose another random sample and repeat the process. In this fashion, train all samples in some random order. A training epoch is a cycle through all the samples in the

training set. Typically, many training epochs are required to train a backpropagation network.

- Check if the stopping criterion has reached. If not, continue with the next training epoch.

The stopping criterion is usually a limit to acceptable mean squared error or a limit to the number of training cycles to use.

Chapter Five

System Design and Implementation

The proposed white blood cell classification system involves the following four stages: image acquisition, segmentation, feature extraction and classification. The high level view of the overall system is shown in figure 12. The classification process starts by an image acquisition step. To acquire an image, first the patient gives blood sample and microscope is needed coupled with camera. The camera then took a snap shot of the sample, sent it to the computer in order to be assessed. After having this microscopic image, the second step is segmentation of an image to acquire a region of interest. In particular, image segmentation methods, like thresholding, are frequently used to detect boundaries in images. After performing segmentation, the next step is extract various suitable features from segmented image those are fed to the neural network as an input. The final step is classification of white blood cell types using neural networks. The classification stage is the decision making part of a recognition system and it uses the features extracted in the previous stage.

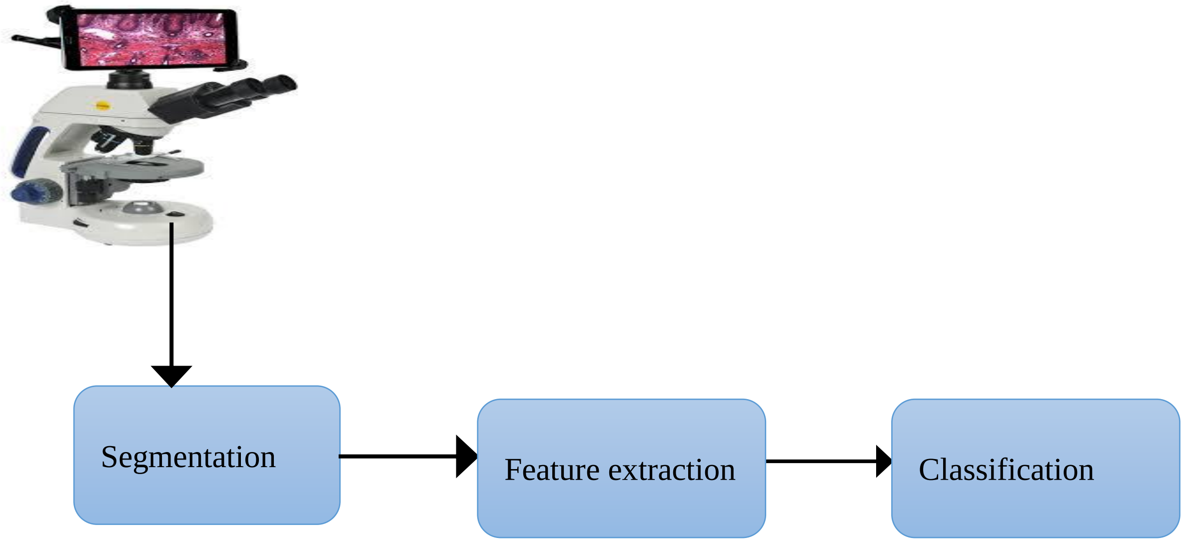


Figure 12 General Scheme of the Design

The details of the design and implementation of each major step is provided in detail in this chapter.

5.1 Image Acquisition Module

This step is one of the most important and crucial phase for obtaining a good result. A good and clear image eliminates the process of noise removal and also helps in avoiding errors in calculation.

In this research images are downloaded from public and free available website that consists dataset of microscopic images of blood samples, specifically designed for the evaluation and the comparison of algorithms for image classification. The images of the dataset have been captured with an optical laboratory microscope coupled with a Canon Power Shot G5 camera. All images are in JPG format with 24 bit color depth, resolution 2592 x 1944. This dataset is composed of 260 microscopic images.

The ‘imread’ Matlab function is the one used to load and obtain matrix representation of images for our work.

Image_wbc=imread(‘image_wbc.jpg);

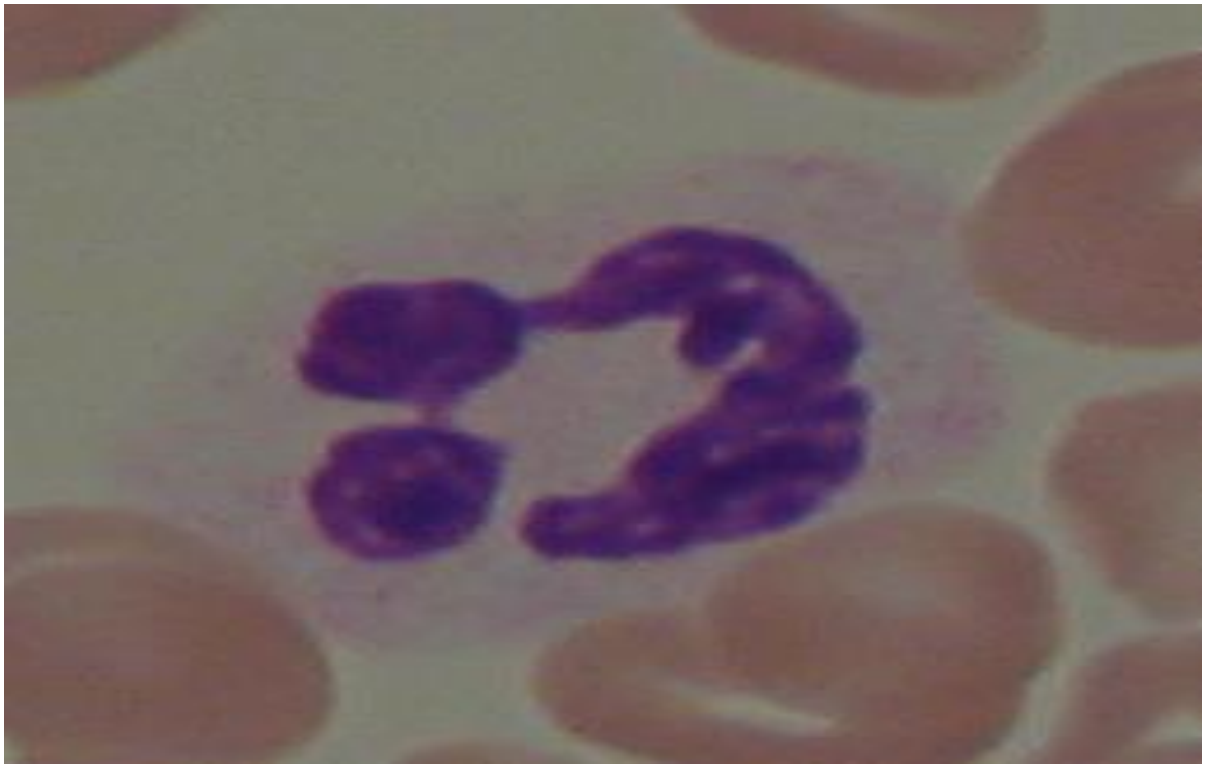


Figure 13 Image of white blood cell

5.2 Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Goal of segmentation of blood cells is to isolate the region of interest from the complicated background and to segment every cell into its components [2]. In other words it is a means to differentiate background region which is red blood cells and others, and the foreground region/ region of interest which is white blood cells.

5.2.1 Threshold Based Segmentation

In this research work, the first technique for segmentation of an image Otsu’s thresholding algorithm is used because of the following advantages as it is explained in detail from Chapter 3.

- It is fast and simple for implementation,
- It does not require prior information of the image,
- It is computationally inexpensive and
- It can be used in real time applications.

The diagram below shows that the steps towards segmentation.

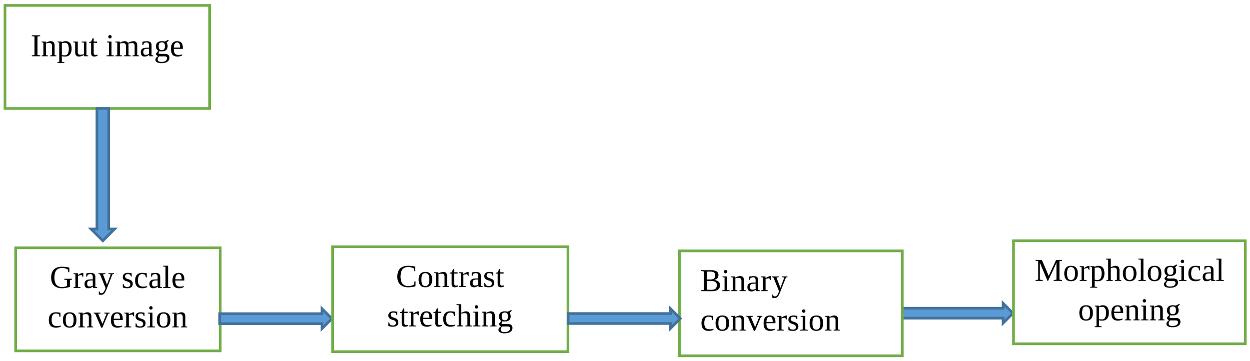


Figure 14 Block diagram for image segmentation

The image segmentation process is basically consists of the following steps:-

1. RGB to grayscale: to improve the quality of the image with only one plane.

2. Adaptive histogram equalization: to adjust image intensity level of grayscale image.

3. Binary conversion: to acquire binary image apply a global threshold Otsu's method on image.

4. Morphological opening: to remove small pixel groups and finally connect the neighboring pixels to form objects.

1) Gray scale conversion

The main aim of this stage is to improve the quality of the image for image segmentation and feature extraction process. Gray scale conversion is the process of converting the true color image (RGB) to the gray scale (intensity) image. In this phase, the Matlab function *rgb2gray( )* is used.

For example: - in order to read the image and for gray scale conversion it can be done as follows.

```
clear;clc;
for i=1:10
im=imread(['G:\DATASET\TESTINGDATASET\Im' num2str(i) '.jpg']);
im2=rgb2gray(im);
figure(i), imshow(im2);
end
```

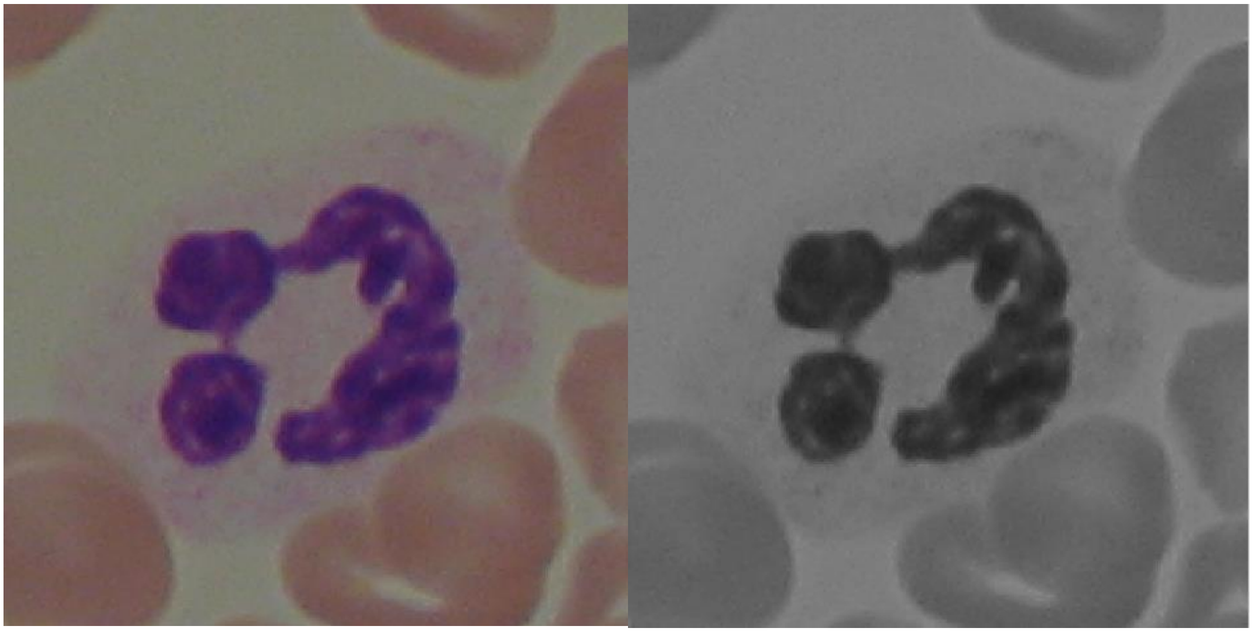


Figure 15 Gray scale image

Converting the RGB image into gray scale (intensity) image is preferred because:

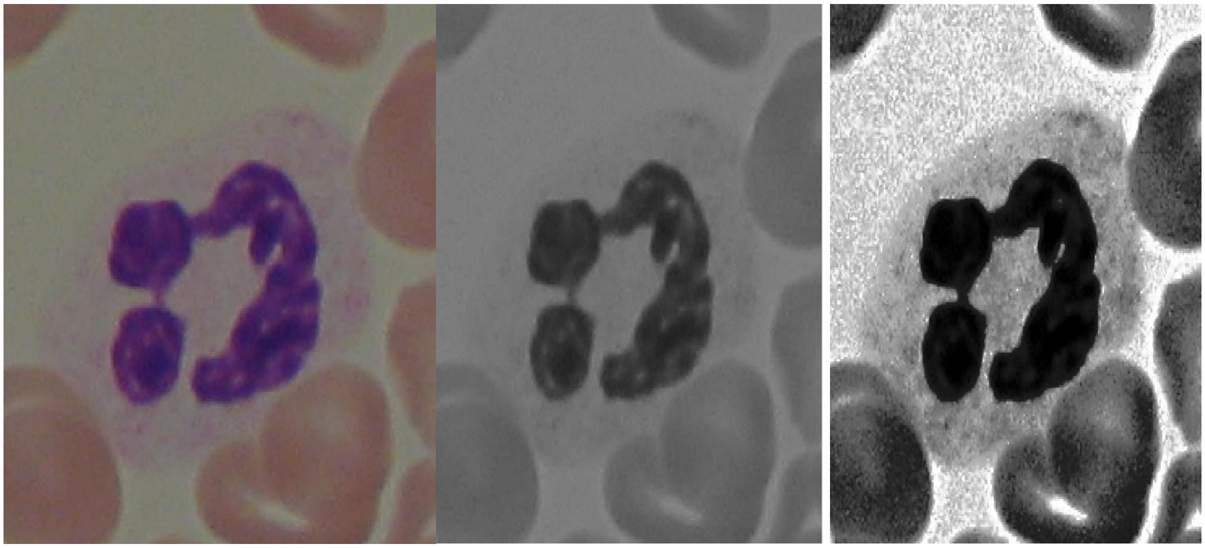
1. It is more efficient for manipulation and processing.

2. When working with RGB or other color coordinate images such as HSL/HSV, it is frequently necessary to access one plane at a time. But with gray scale images, we have only one plane, the intensity value (that ranges from 0-255). So due to this it reduces access time, which indirectly decreases the computational time.

2) Histogram equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast. It accomplishes this by effectively spreading out the most frequent intensity values. That means stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

```
J = histeq(I);
imshow(J);
figure, imhist(J,64);
```



(a) RGB image (b) Gray scale image (c) Image enhancement

Figure 16 Histogram equalization

As show from the above figure (a) represents the normal RGB image of the white blood cell neutrophil type. Figure (b) represents the gray scale conversion of RGB image and figure (c) represents image enhancement of the gray scale image to increase the contrast of the image for a better quality of the image for further processing.

3) Binary conversion

In image processing, segmentation is the process of partitioning an image into several segments or partitions. Segmentation is used for locating objects and boundaries in an image. It is also useful to change the representation of an image for easy analysis and manipulation. For example, we can convert an intensity image to binary image using segmentation technique.

There are different types of segmentation techniques. From these methods, Otsu thresholding segmentation and one of the main advantages of Otsu's method is that it is adaptive. That is for each image, a different threshold is calculated to compute the ultimate global image threshold.

Otsu thresholding segmentation involves the process of obtaining a binary image based on the brightness of the pixels, that is, pixels brighter than a specified brightness level are taken as 1 and the rest are left 0 according to the value assignment. This method is chosen because it is a simple technique and it provides an easy and convenient way to perform image segmentation based on different intensities or colors in the foreground and background regions of an image. The main challenge in this technique is selecting the threshold value that will give a good segmentation result.
Otsu's thresholding chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels, based on the idea: find the threshold that *minimizes the weighted within-class variance.*
Thresholding was performed on the grayscale images with MATLAB's graythresh function using Otsu's method.The general mathematical model for the Otsu's threshold technique algorithm is:
If $p(x, y) \geq T$, $p(x, y) = 1$ and this pixel belongs to white blood cells

Else $p(x, y) < T$, $p(x, y) = 0$ and this pixel belongs to unwanted-objects

function level = graythresh(I);

graythresh compute global threshold using Otsu's method. The final output of this phase is a binary image. For example the figure below shows the original RGB image and the image obtained after thresholding.

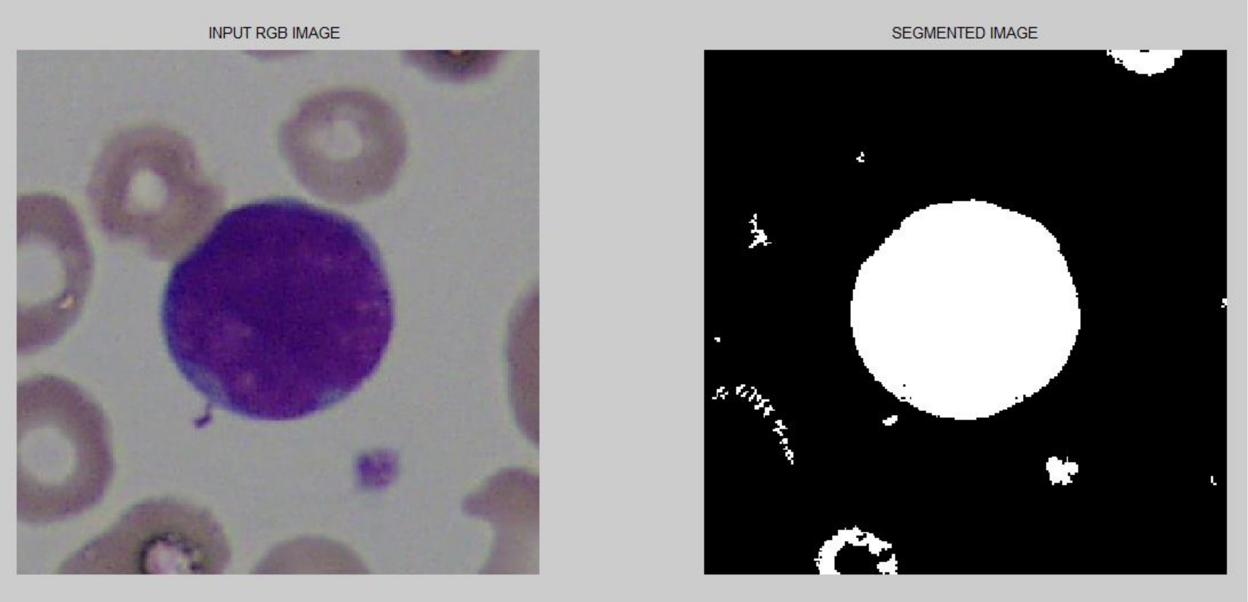
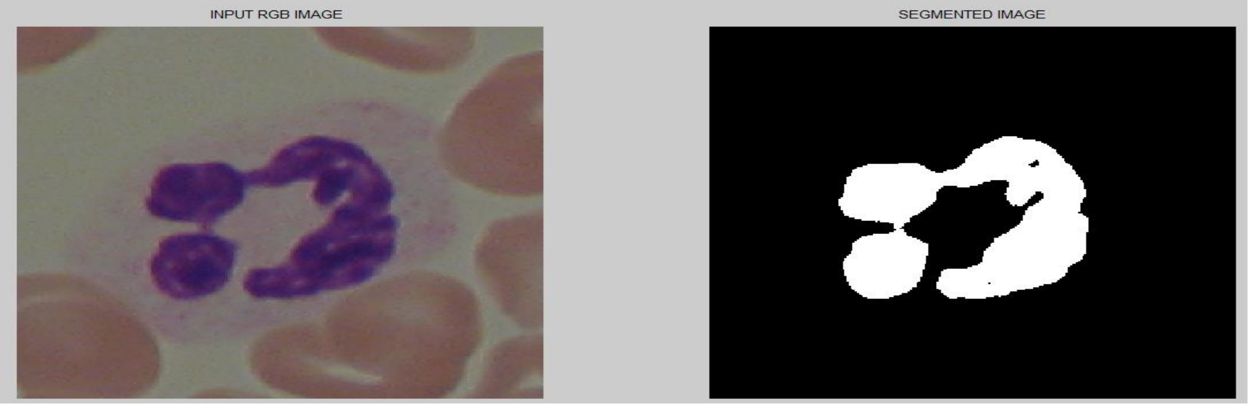


Figure 17 Segmented image before area opening

4) Morphological operation

Finally, the binary mask was applied to the grayscale image and in order to clean up the operation called area opening is used, that allows to delete all the objects with a size smaller than the structuring element.



(a) Original image

(b) image obtained after removing Small components and masking

Figure 18 Segmented image after area opening

5.2.2 Clustering based segmentation

One of most used clustering algorithm is *k*-means clustering. K-means is an automatic segmentation algorithm whose speed depends on the number of clusters K. According to this approach, similar intensities are clustered in the same cluster while different intensities are clustered to other clusters based on the value of K, which is selected manually.

K-means works based on the idea that grouping the pixels having similar properties and defines the cluster values according to their visible intensities.

Steps in K-Means algorithm:

1. Select the number of clusters K.
2. Randomly selecting K data points for the centroids.
3. Assign each data point to the closest centroid → that forms K clusters.
4. Keep iterating until there is no change to the centroids. i.e. assignment of data points to clusters is not changing.
    - ➢ Compute the sum of the squared distance between data points and all centroids
    - ➢ Assign each data point to the closest cluster (centroid).
    - ➢ Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

By applying the above steps the resultant segmented image is as follows:

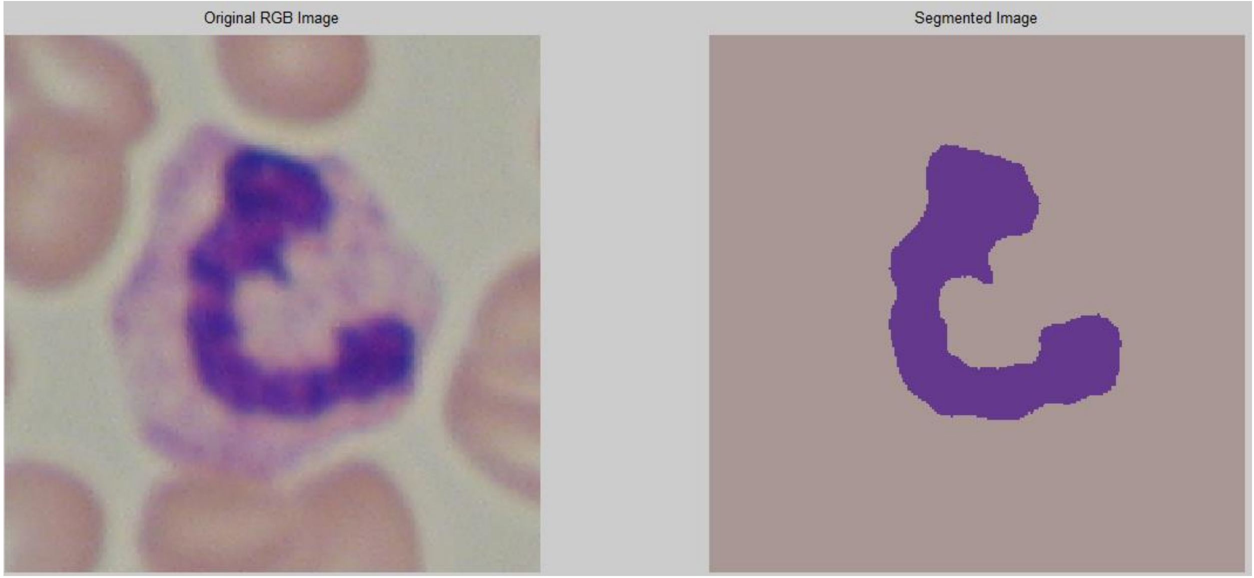


Figure 19 Applying K-means segmentation

5.3 Feature Extraction

Feature extraction refers to finding common features from the segmented image or in the Region of Interest (ROI) to classify it to different categories. It is the objective of this work to find the minimum features required to classify five WBC classes.

Feature extraction efficiently represent the major attributes of an image that are useful for effective classification of white blood cell types. The extracted features provide a means to identify and classify each type of WBCs.

The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements.

These features can be grouped into shape features, intensity features, and texture features which are clearly explained in Chapter 3.

Shape/Geometric based features

- Area: Area of the object is scalar and is calculated by counting the actual number of pixels in the region of interest.
- Perimeter: the total distance between consecutive points of the border.
- Eccentricity: which describes the degree of similarity to a circle. If the object shape is very similar to a circle shape, then the object will have an eccentricity of 0. If the object shape is very similar to a line, it will have an eccentricity of 1.

Intensity/Statistical based features

- Mean, variance, standard deviation, skewness and so on which are clearly stated in Chapter 3.

Texture based features

Here are some of texture based features: -

- Contrast: Contrast measures the difference in grayscale intensity between adjacent pixels over the entire image. The greater the difference in intensity values, the higher the value of contrast.

- Homogeneity: Homogeneity measures the distances of GLCM elements from the GLCM diagonal. Homogeneity ranges from 0 to 1. If adjacent pixels always have very similar values of grayscale intensity, the homogeneity will be close to 1.
- Entropy: Entropy is a measure of the randomness of grayscale intensity values of pixels and it is based on the grayscale histogram of the image.

Regionprops feature extraction

The first extraction method explored was regionprops. Regionprops is an image processing toolbox function available in Matlab that allows users to measure/extract a set of properties from an image. The image is then labeled using the values obtained from the region. The measured features of the bloodstain image that were used in this research were stored as matrix with constant values.

The syntax for regionprops is:

STATS = regionprops (BW, properties)

Where STATS measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array and it can have any dimension.

The features as matrix values of the image are like area, perimeter, eccentricity, circularity, major and minor axis length of the segmented region. These image properties are used widely for the technique for the conversion of an image to values in matrix format.

Graycoprops feature extraction

The second extraction method explored was Graycoprops. Graycoprops is also another image processing toolbox function available in Matlab that allows users to measure/extract a set of properties for an image.

The identification of specific textures in an image is achieved primarily by modelling texture as a two-dimensional gray level variation. This two dimensional array is called gray level co-occurrence matrix (GLCM). Co-occurrence matrix which is a tabulation of how often different combinations of pixel values occur in an image.

The syntax for graycoprops is:

STATS = graycoprops(glcm, properties)

Where *STATS* calculates the statistics specified in properties from the gray-level co-occurence matrix glcm. glcm is an m-by-n-by-p array of valid gray-level co-occurrence matrices. If glcm is an array of GLCMs, stats is an array of statistics for each glcm.

The features as matrix values of the image are like contrast, energy, correlation and so on.

5.4 Classification

Classification refers to the process of assigning a given input data into one of a given number of categories or it is the process of identifying each white blood cell type and assigning it to the correct white blood cell class. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. Classification compares feature vectors to the various models and find the closest match.

The classification stage is the decision making part of a recognition system and it uses the features extracted in the previous stage. In other words, the system can tell whether a given white blood cell belongs to a one of five normal classes or not. For the purpose of classification the use of artificial neural network is proposed. The features extracted from the segmented image are given as an input to the neural network for automatic classification of each type of white blood cells. ANN is chosen as a classifier because it is an effective classifier for many real time applications.

As referred from many literatures, MLP is used widely and it is quite enough for image classification tasks. Back propagation algorithm is by far the most widely used algorithm to train MLPs for pattern recognition and other purposes.

Why using neural network?

- Neural network enable solutions to be found to problems where algorithmic methods are too computationally intensive or do not exist.
- The problem of feature extraction and classification seems to be a suitable application for neural nets.
- They offer significant speed advantages over conventional techniques.

For the classification purpose multilayer perceptron with back propagation algorithm has been used. Due to learning capability, better performance in classification issues as efficient and decision making based on diagnoses features, MLP ANNs are the best system to do this task. The designed neural network model as shown from Figure 20 has the following specifications.

The designed ANN consists of three layers: input layer, hidden layer and the output layer.

- **Input Layer:** This layers comprises of input units which signifies the unprocessed information provided for the networks. Since the number of inputs should be equal to the number of input layer neurons, number of neurons on input layer is 17, which representing the 17 extracted features.
- **Hidden Layer:** This layer is characterized by hidden units which are inclined by the behavior of the input units and the weight that join these input and the hidden components. The designed classifier contains a single hidden layer with 20 nodes. The number of hidden layer neurons is determined through experimentation. During experimentation the following core issue is also put in to consideration. This number must not be too large to avoid waste of computations and slow convergence process.
- **Output Layer:** The output units' behavior is dependent on the particularity of the hidden units and the weights joining the hidden and output units. On determining the number of output layer neurons, it is the same as the classifier outputs. Thus, the output layer contains 5 neurons which represents the five WBC types.

In general the classifier has the specifications of a two layer feed forward network to be trained with back-propagation algorithm with 17 input features and 5 output categories each represents for Basophil, Eosinophil, Lymphocyte, Monocyte and Neutrophil. As it is indicated from Figure 35 the number of nodes used at the hidden layer is 20 which results a maximum accuracy to reach a minimum error 10^{-5}. The learning rate is 0.4 and number of iterations/epochs is settled to be 1000.

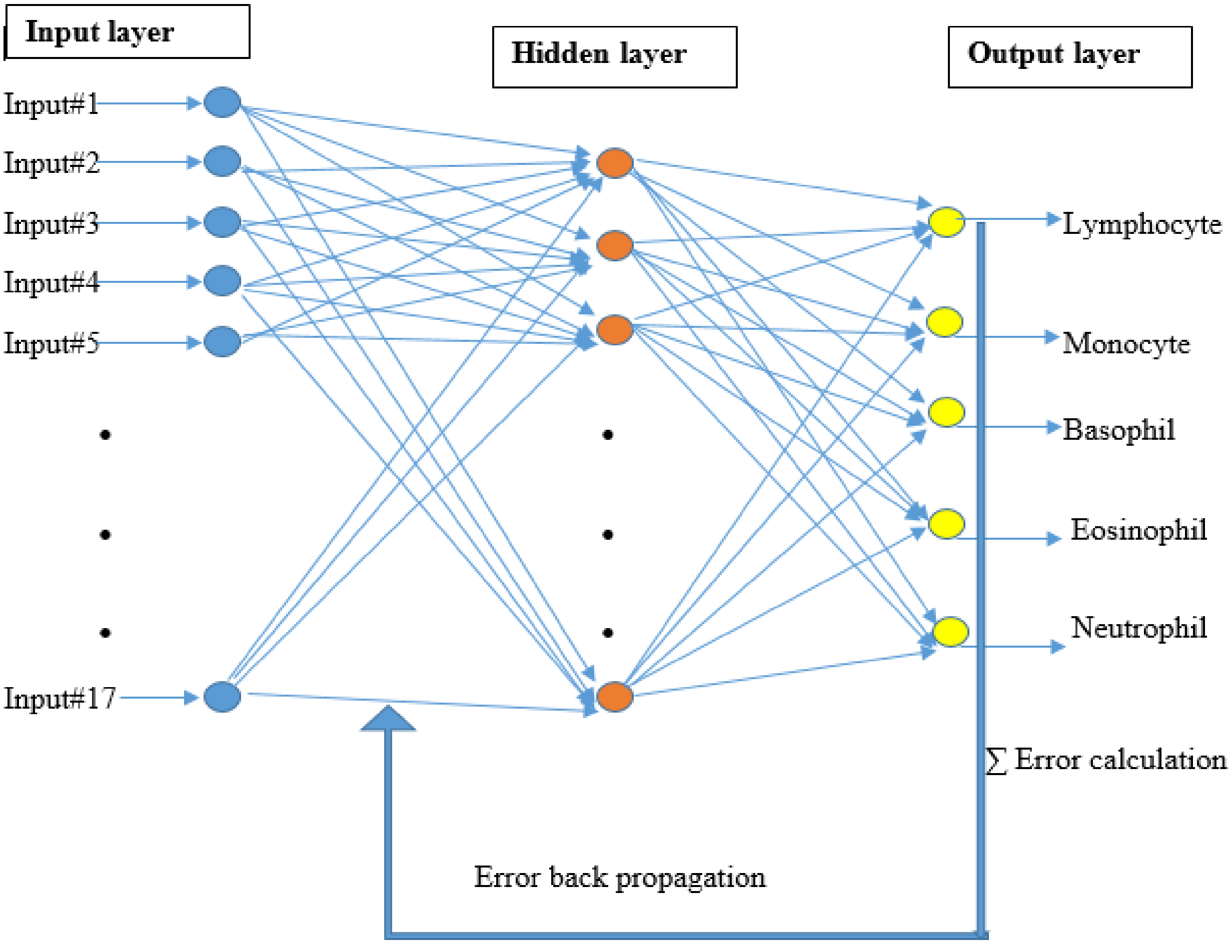


Figure 20 The designed classifier model

Chapter Six

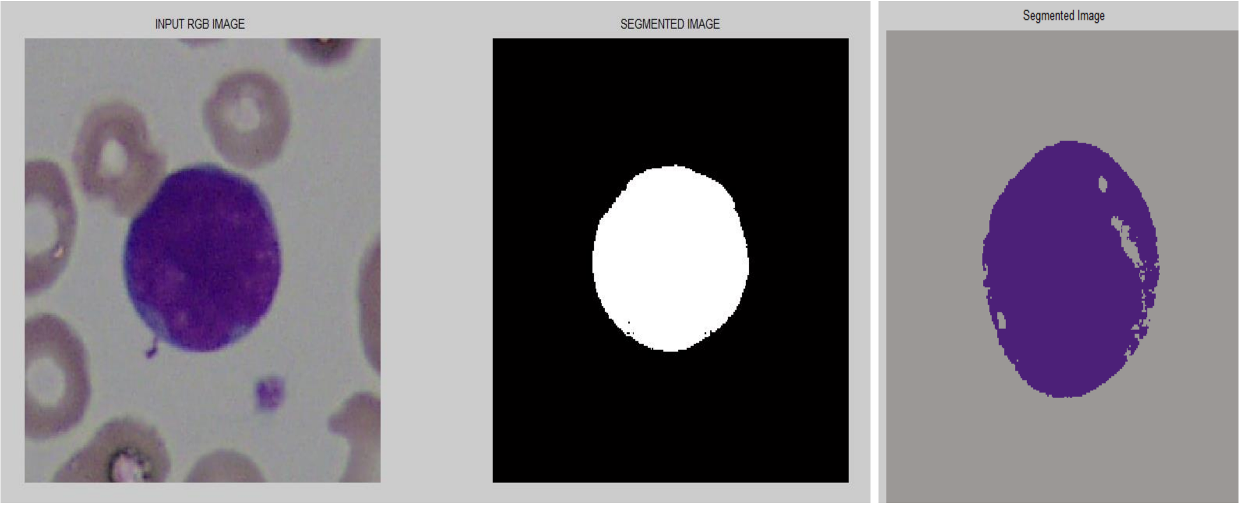
Results and Discussions

The design described in the previous chapter is implemented using Matlab software. For the purpose of experimentation, microscopic images of different types of leukocyte cells are used. The implementation was done on Intel(R) core-i3 processor, 4 GB RAM machine using Matlab R2014a. Therefore, the results and analysis of each step towards classification is presented in this chapter.

6.1 Segmentation Results

For the segmentation of the image both threshold based and clustering based algorithms has been used which is clearly explained from Section 5.2. The following figures show that the original image on the left side, an image segmented by thresholding at the middle and an image segmented by k-means clustering on the right side for each type of WBCs as its detail process is described in Chapter Five.

Segmented image for lymphocyte: Figure 21 shows that the original RGB image and the segmented image for lymphocyte type.



a) Original image b) segmented image by Otsu c) segmented image by K-means

Figure 21 Segmented image for lymphocyte

Segmented image for monocyte: Figure 22 shows that the original RGB image and the segmented image for monocyte type.

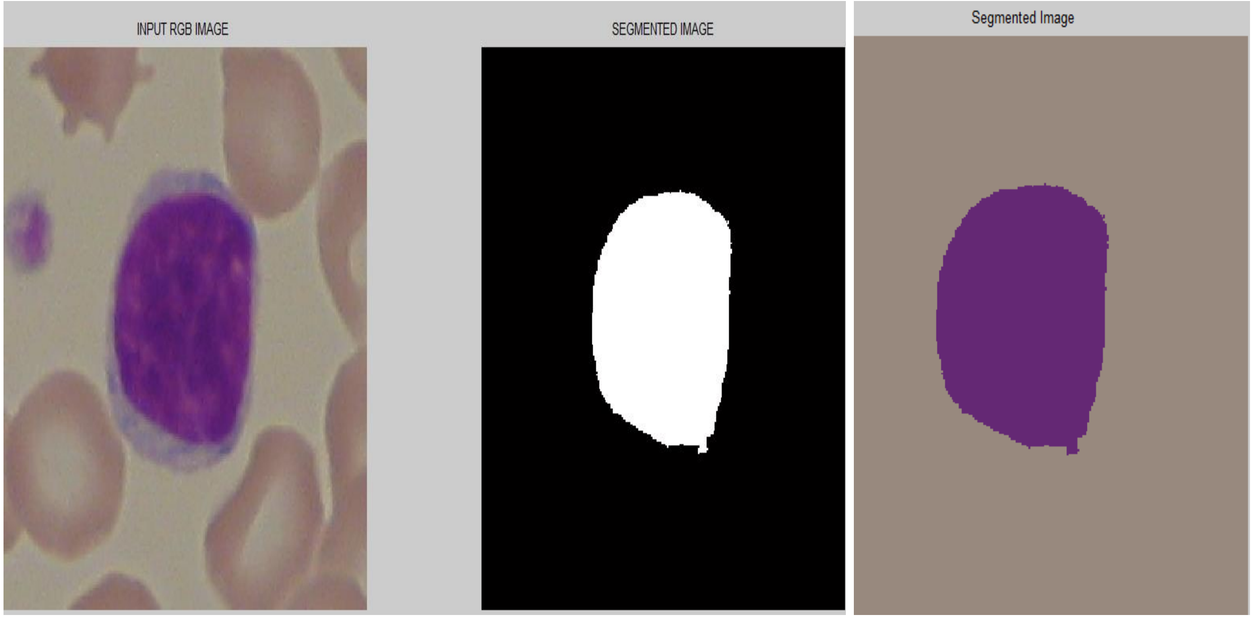


Figure 22 Segmented image for monocyte

Segmented image for basophil: Figure 23 shows that the original RGB image and the segmented image for basophil type.

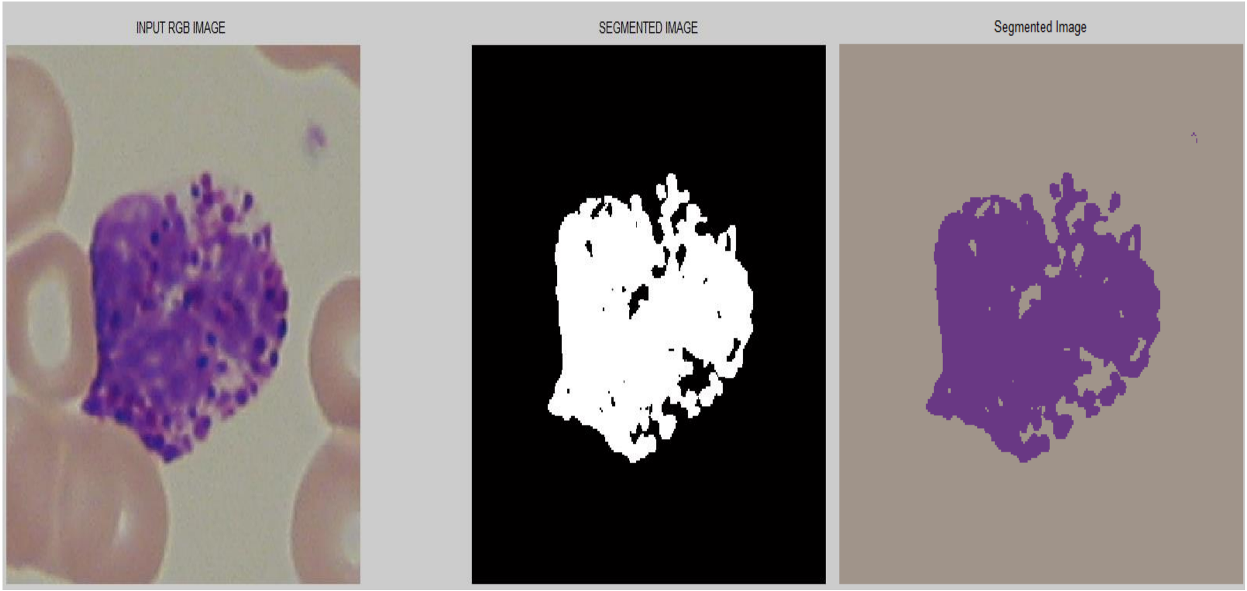


Figure 23 Segmented image for basophil

Segmented image for eosinophil: Figure 24 shows that the original RGB image and the segmented image for eosinophil type.

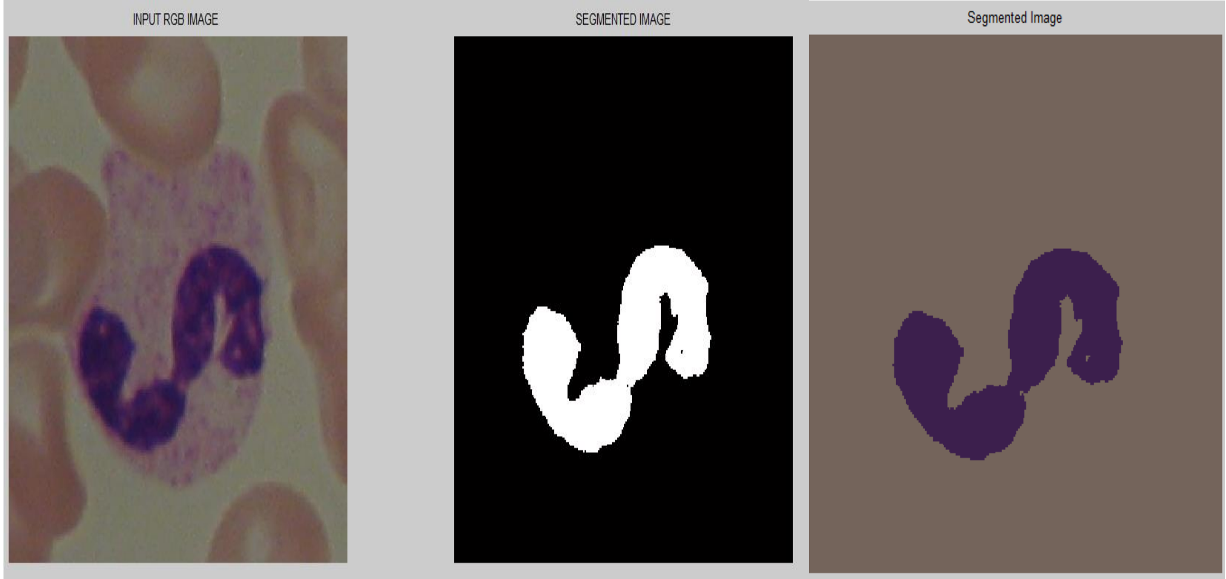


Figure 24 Segmented image for eosinophil

Segmented image for neutrophil: Figure 25 shows that the original RGB image and the segmented image for neutrophil type.

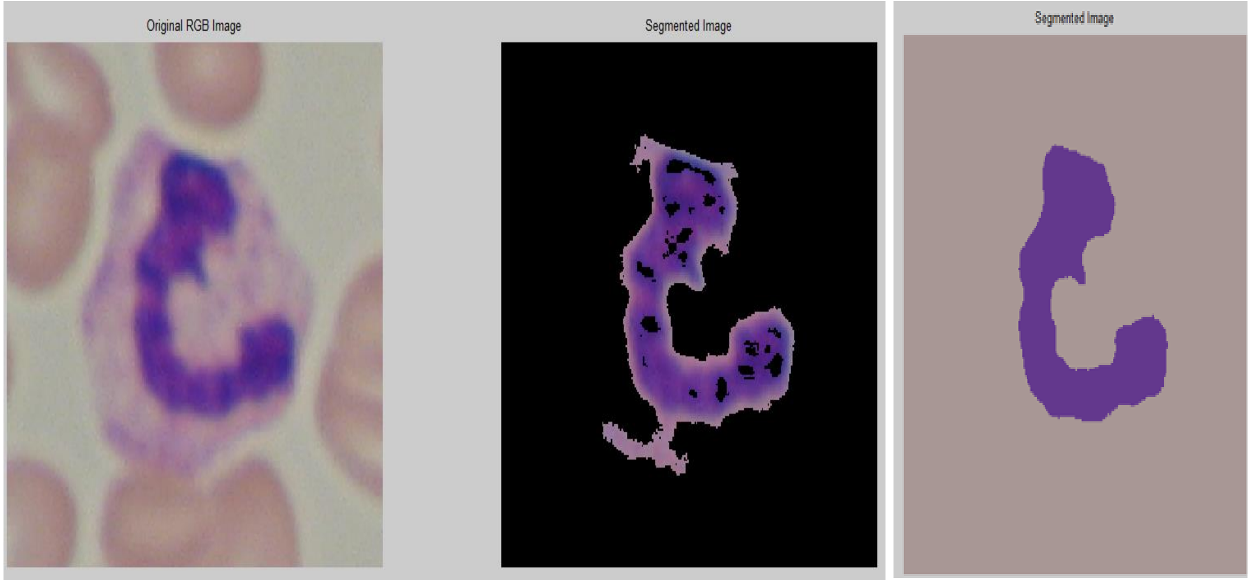


Figure 25 Segmented image for neutrophil

Evaluation of the segmentation

To evaluate the performance of the two segmentation techniques a similar approach [38] is used. The segmented images by these techniques compared with the ground truth that is manually segmented by experts to calculate the accuracy of the segmentation result.

To compute the accuracy the ground truth is necessary. By computing the difference between each segmented pixel value with the ground truth, the error and the accuracy can be calculated as follows.

For example, for the first image the total number of pixels segmented by Otsu thresholding is 11213 and total number of pixels segmented by experts is 10432. Here the difference is 781. Then error can be calculated by the difference of the two segment pixel values divided by the correctly segmented pixel value times by 100. i.e., (781/10423)*100=7.49. And the accuracy is 92.51.

In the same way for the first image the total number of pixels segmented by K-means clustering is 10814 and total number of pixels segmented by experts is 10432. Here the difference is 382. Then error can be calculated by the difference of the two segment pixel values divided by the correctly segmented pixel value times by 100. i.e., (382/10423)*100=3.66. And the accuracy is 96.34. For the evaluation of the rest of the images similar approach is used.

But, the above calculation works well if the two (ground truth and segmented images by this system) have an intersection area of interest. Otherwise, it affects the accuracy of segmentation. Therefore, to calculate the similarity or an intersection a built in Matlab function is used which is Intersection over Union the so called Jaccard index.

Similarity=jaccard (BW1, BW2);

Where, jaccard computes the intersection of binary images BW1 and BW2 divided by the union of BW1 and BW2.

BW1 and BW2 represents the first and the second binary image respectively.

Intersection over Union calculates an intersection of area of pixels divided by the union of area of pixels from the two regions. The Jaccard index which ranges from 0 to 100% where 0% is no intersection between the images and 100% is perfectly matched. The results showed that the value greater than zero for every segmented images. If there wasn't an intersection between the two images, the Jaccard index would be zero. Since as it is indicated from Figure 21 to Figure 25 all the images used in this thesis work are effectively segmented our region of interest which is the particular white blood cell, there is an overlap or intersection between the images segmented by experts and by the system.

After all images are segmented using thresholding and k-means which obtained an accuracy of 88.2% and 91.6% respectively.

Table 2 Accuracy of segmentation

Image no.	Total number of pixels segmented by Otsu thresholding	Total number of pixels segmented by k-means clustering	Total number of pixels segmented by experts/Ground truth	Accuracy in Otsu thresholding (%)	Accuracy in K-means clustering (%)
Image1	11213	10814	10432	92.51	96.34
Image2	10925	10664	9765	88.12	90.79
Image3	13126	12644	11234	83.16	87.45
Image4	9321	9884	10254	89.99	96.25
Image5	10113	8169	9216	90.27	87.18

6.2 Feature Extraction Results

To classify WBCs, features that represent their images have to be obtained. Those features have to be capable of distinguishing between WBC classes and it is desired to have as few features as possible. Although the classification accuracy is dependent on the classifier, the performance of the classifier is totally dependent on having adequate features.

Feature extraction is a superior form of dimensionality reduction and efficiently represent the major attributes of an image that are useful for effective classification of each classes. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called feature extraction. As it is clearly described in Chapter 3, in this work, various features of an image are extracted using the predefined formulas in Matlab.

Here, 17 input features are extracted which represent shape features (area, perimeter, major axis length, minor axis length, average axis length eccentricity and circularity), intensity features (mean, variance, standard deviation, skewness, kurtosis and smoothness), and textural features (energy, entropy, correlation, contrast, homogeneity) for the WBC classification. In order to extract features for our matrices 260 images are used. Thus the shape, intensity and texture features of 260 microscopic images of the blood samples are extracted and given as an input to the classifier (neural network).

The table below shows some of the recorded extracted features for the first 5 images.

Table 3 Extracted features

Features	**Image1**	**Image2**	**Image3**	**Image4**	**Image5**
Perimeter	565.3260	381.6750	813.2320	658.4120	679.1270
Major-axis length	139.3283	120.4543	150.4949	161.2241	145.3327
Minor-axis length	122.3277	102.0205	121.6180	140.7672	118.5222
Average-axis length	130.828	111.2374	136.0565	150.9956	131.9274
Area	10814	10664	12644	9884	8869
Eccentricity	0.3533	0.6274	0.5394	0.8994	0.5854
Contrast	0.0858	0.0700	0.1279	0.0476	0.0560
Energy	0.6380	0.6974	0.6325	0.6802	0.7534
Correlation	0.9097	0.9303	0.9243	0.9270	0.8832
Homogeneity	0.9828	0.9867	0.9763	0.9865	0.9887
Entropy	1.9817	1.5880	2.0385	1.6468	1.3305
RMS	0.0829	0.0765	0.1146	0.0843	0.0600
Mean	0.0496	0.0475	0.0683	0.0433	0.0308
Variance	0.0089	0.0088	0.0147	0.0077	0.0055
Smoothness	0.9997	0.9997	0.9998	0.9997	0.9995
Kurtosis	7.1389	6.7195	4.2512	4.5617	8.3170
Skewness	2.1873	2.1929	1.6800	1.8158	2.5284

6.3 Classification Results

Classification was performed using the designed ANN model described in Section 5.4. Thus, the detail description and performance analysis of the network is given as follows.

The feed forward back propagation neural network, which is a very popular model in biological and biomedical applications, was used. This type of neural network configuration does not have feedback connections, but errors are propagated back during training using least mean squared error.

The back propagation neural network requires pairs of input and target vectors which are specified below.

Input matrix: - [17x260], 17x260 matrix, representing the data: 260 images of 17 features.

Target matrix: - [5x260], a 5x260 matrix, representing the data: 260 images of 5 classes, where each column indicates a category with a one in either of Basophil, Eosinophil, lymphocyte, Monocyte and Neutrophil.

In order to classify the WBCs into five classes, the target value for the WBC neural network classifier is considered as shown below. The basic approach to arrange the target class is to classify by using static 1 or 0 such that 1 corresponding to target category and 0 for the rest of the others.

Table 4 Target values of WBC for the classifier

White blood cell types	**Neural network target**				
	Target1	Target2	Target3	Target4	Target5
Neutrophil	1	0	0	0	0
Eosinophil	0	1	0	0	0
Monocyte	0	0	1	0	0
Basophil	0	0	0	1	0
Lymphocyte	0	0	0	0	1

Specification of Neural Network

As the design of the neural network model is shown from figure 19 the specification of the neural network is described below.

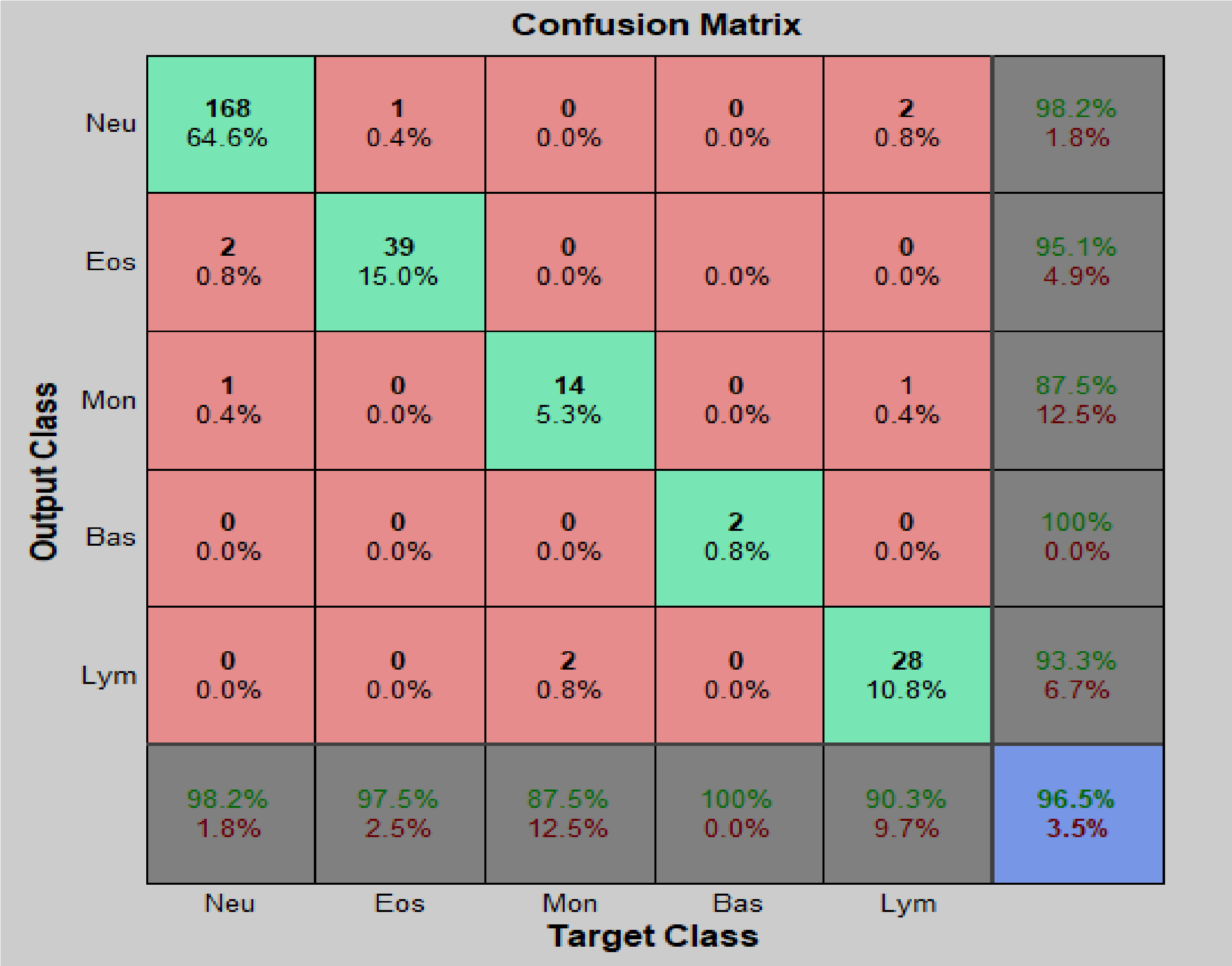
- Pattern of network....................... ………..Two layer feed forward.

- Number of hidden neurons ....20 (found experimentally).
- Training... Scaled Conjugate Gradient.
- Total number of samples .... 260.
- Number of inputs..17 input vectors.
- Number of output categories ....5 including (Basophil, Eosinophil, Lymphocyte, Monocyte and Neutrophil).
- Target data set is given as shown in table 4.

One of the measure of how well the designed neural network has fit the data is the confusion plot. The confusion matrix below which is generated using nprt (neural network pattern recognition tool) in matlab shows the percentages of correct and incorrect classifications. Correct classifications are the green squares on the matrices diagonal. Incorrect classifications are the red squares. The confusion matrix consists of an output class as a row and the target class as a column.

The output class and the target class encompass the symbols Neu, Eos, Mon, Bas and Lym which represents Neutrophil, Eosinophil, Monocyte, Basophil and Lymphocyte respectively. From the green squares each square consists of the number and the percentage of correctly classified elements from the total for each type. For example: the first green square row one column one holds a number 168 and 64.6%. This means that 168 elements for neutrophil type are correctly classified which is 64.6 percent from the whole elements.

After the extraction of input features by using regionprops and graycoprops from the segmented image using k-means clustering with a segmentation accuracy of 91.6 %, all 17 extracted input features are used for the classifier as an input to achieve the following result. Using these input features the classifier gives an accuracy of 96.5% and 3.5% which is percentage of correct classification and percentage of incorrect classification respectively.

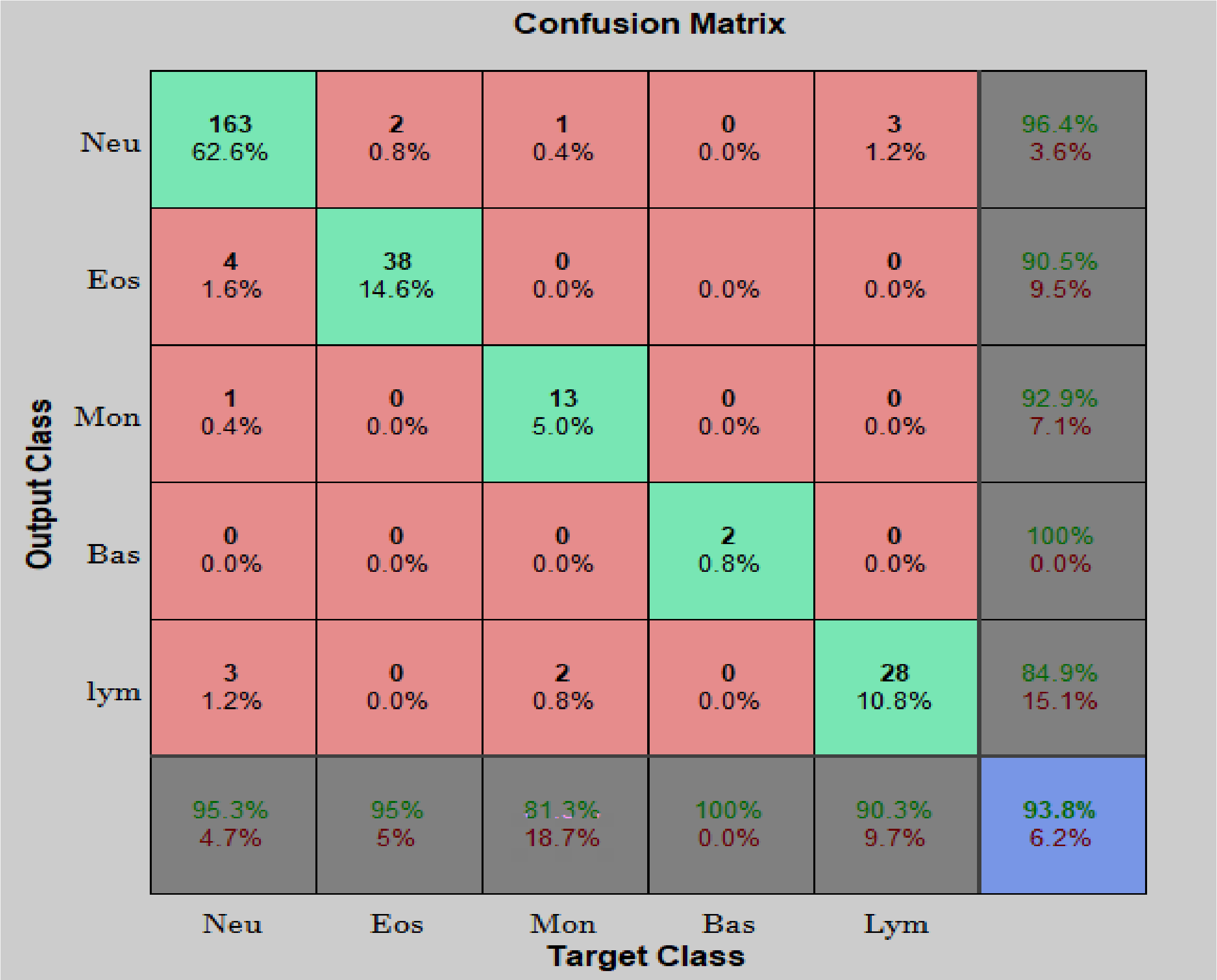


Confusion Matrix

Output Class \ Target Class	Neu	Eos	Mon	Bas	Lym	
Neu	168 64.6%	1 0.4%	0 0.0%	0 0.0%	2 0.8%	98.2% 1.8%
Eos	2 0.8%	39 15.0%	0 0.0%	0.0%	0 0.0%	95.1% 4.9%
Mon	1 0.4%	0 0.0%	14 5.3%	0 0.0%	1 0.4%	87.5% 12.5%
Bas	0 0.0%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	100% 0.0%
Lym	0 0.0%	0 0.0%	2 0.8%	0 0.0%	28 10.8%	93.3% 6.7%
	98.2% 1.8%	97.5% 2.5%	87.5% 12.5%	100% 0.0%	90.3% 9.7%	96.5% 3.5%

Figure 26 Confusion diagram for classification

Similarly, after the extraction of features using regionprops and graycoprops from the segmented image using Otsu thresholding with a segmentation accuracy of 88.2 similar number of 17 extracted input features are used for the classifier as an input to achieve the following result. Using these input features the classifier gives an accuracy of correct classification 93.8% and an incorrect classification of 6.2%.



Confusion Matrix

Output Class

	Neu	Eos	Mon	Bas	Lym	
Neu	163 62.6%	2 0.8%	1 0.4%	0 0.0%	3 1.2%	96.4% 3.6%
Eos	4 1.6%	38 14.6%	0 0.0%	0.0%	0 0.0%	90.5% 9.5%
Mon	1 0.4%	0 0.0%	13 5.0%	0 0.0%	0 0.0%	92.9% 7.1%
Bas	0 0.0%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	100% 0.0%
lym	3 1.2%	0 0.0%	2 0.8%	0 0.0%	28 10.8%	84.9% 15.1%
	95.3% 4.7%	95% 5%	81.3% 18.7%	100% 0.0%	90.3% 9.7%	93.8% 6.2%

Target Class

Figure 27 Confusion diagram for all features

Evaluation method: the performance of the designed classifier was evaluated using accuracy which is the percentage of correctly classified images of the total test input images.

Accuracy is the performance metric that projects the classification correctness of the classifier which can be calculated as follows.

$$\text{Accuracy} = \frac{\text{Number of correctly classified images}}{\text{Total number of images}} * 100 \qquad (6.1)$$

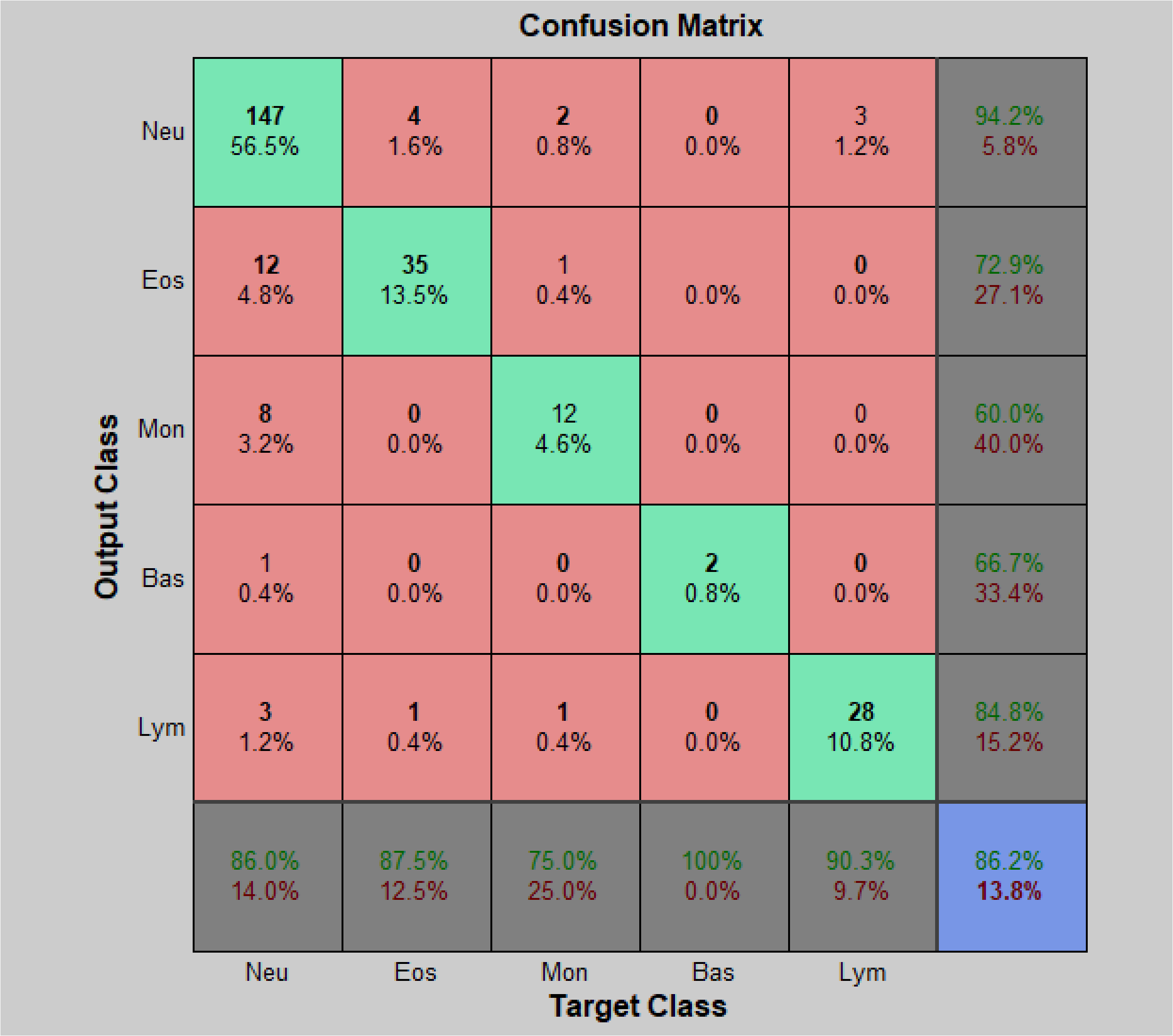
The table below illustrates that the result obtained by the designed classifier from confusion matrix diagram of figure 26 which consists of the names of the cell type, number of images for each type, number of correctly classified images, number of misclassified images, percentage of correctly classified images and overall accuracy. Using the equation 6.1 the overall accuracy is 96.5%.

Table 5 Classifier result using all features

White blood cell type	Number of images	Number of correctly classified images	Number of misclassified images	Percentage of correctly classified images	Overall accuracy
Lymphocyte	31	28	3	90.3	**96.5%**
Basophil	2	2	0	100	
Monocyte	16	14	2	81.3	
Eosinophil	40	39	1	97.5	
Neutrophil	171	168	3	98.2	
Total	260	251	9		

The confusion matrix diagram below shows the result for the classifier when the inputs are only shape/geometric features like: area, perimeter, major axis length, minor axis length, average axis length and eccentricity. The diagonal green matrix holds the number of elements and the percentage of correctly classified elements for each type. However, the red squares consist the number of elements and the percentage of misclassified elements from the whole input.

For example, row one column one shows 147 and 56.5% which means 147 images of neutrophil type are correctly classified and the percentage 56.5% indicates that the number 147 covers 56.5 percent of the whole input images. Similarly, row two column two shows 35 and 13.5% which is 35 images of eosinophil type are correctly classified. And this number covers 13.5 percent of the whole input images. The accuracy of the classifier using these input features is 86.2% as it is indicated from figure 28 at the bottom right position.

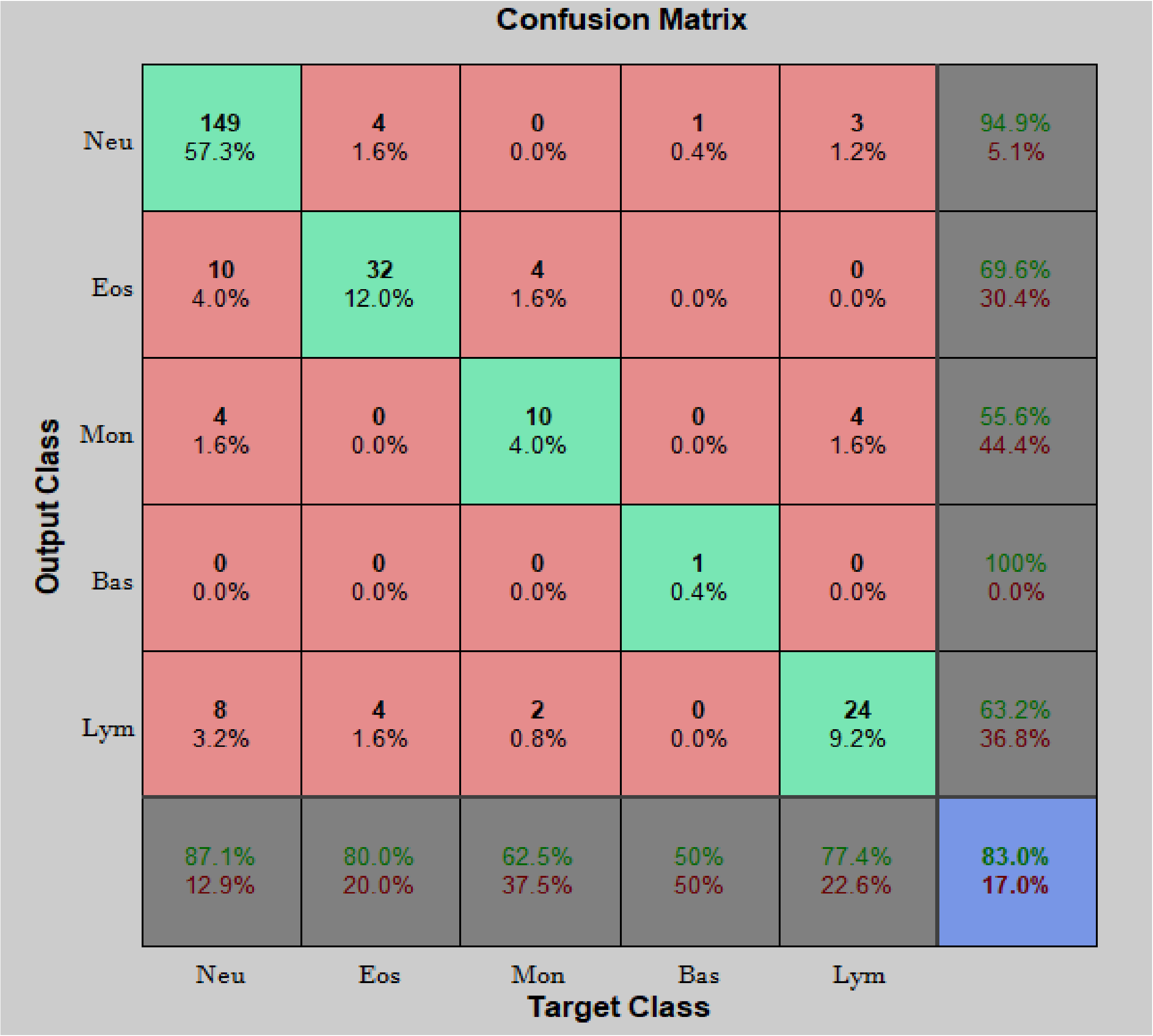


	Neu	Eos	Mon	Bas	Lym	
Neu	147 56.5%	4 1.6%	2 0.8%	0 0.0%	3 1.2%	94.2% 5.8%
Eos	12 4.8%	35 13.5%	1 0.4%	0.0%	0 0.0%	72.9% 27.1%
Mon	8 3.2%	0 0.0%	12 4.6%	0 0.0%	0 0.0%	60.0% 40.0%
Bas	1 0.4%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	66.7% 33.4%
Lym	3 1.2%	1 0.4%	1 0.4%	0 0.0%	28 10.8%	84.8% 15.2%
	86.0% 14.0%	87.5% 12.5%	75.0% 25.0%	100% 0.0%	90.3% 9.7%	86.2% 13.8%

Figure 28 Confusion diagram for shape features

The confusion diagram from figure 29 shows that the classification result for the classifier when only intensity/statistical features were applied. Intensity features like: mean, variance, standard deviation, skewness, kurtosis and smoothness were used as an input to the neural network classifier.

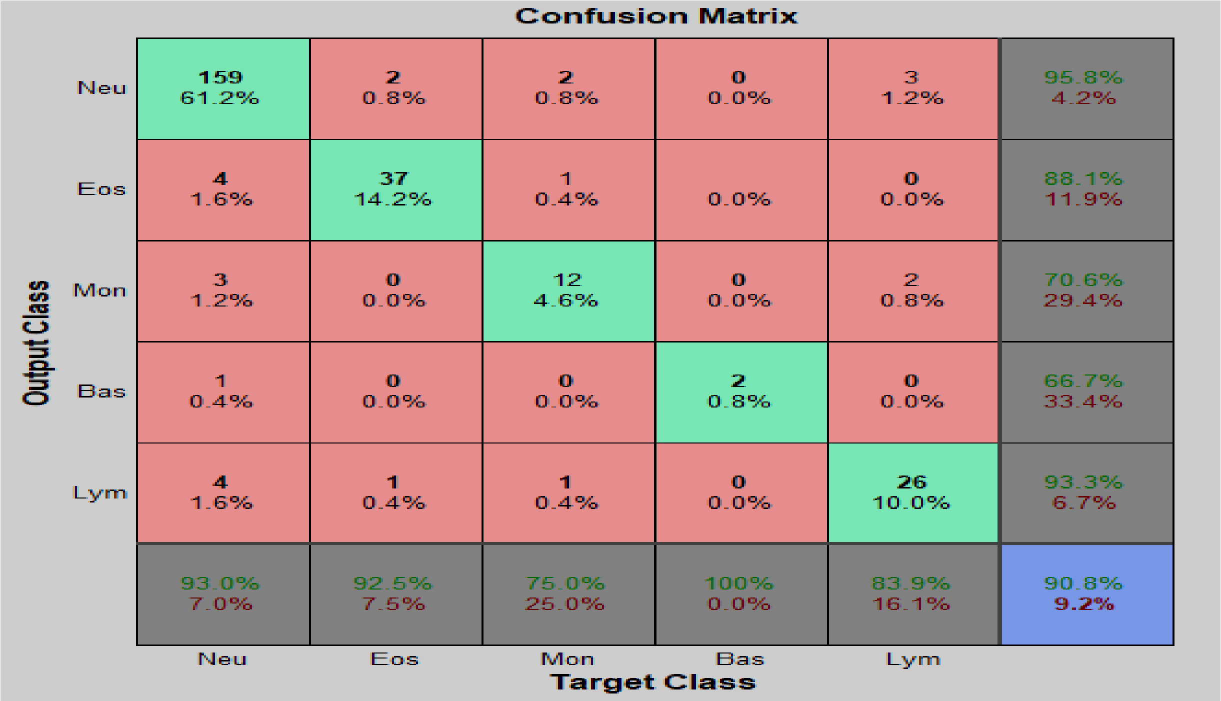
The diagonal green matrix holds the number of elements and the percentage of correctly classified elements for each type. In the same way, for example, row one column one shows 149 and 57.3% which means 149 images of neutrophil type are correctly classified and the percentage 57.3% indicates that the number 149 covers 57.3 percent of the whole test input images. Then the classifier yields only 83% of elements are correctly classified which is very small accuracy result compared to the other feature sets. And using these input features seventeen percent of test images are misclassified.



	Neu	Eos	Mon	Bas	Lym	
Neu	149 57.3%	4 1.6%	0 0.0%	1 0.4%	3 1.2%	94.9% 5.1%
Eos	10 4.0%	32 12.0%	4 1.6%	0.0%	0 0.0%	69.6% 30.4%
Mon	4 1.6%	0 0.0%	10 4.0%	0 0.0%	4 1.6%	55.6% 44.4%
Bas	0 0.0%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	100% 0.0%
Lym	8 3.2%	4 1.6%	2 0.8%	0 0.0%	24 9.2%	63.2% 36.8%
	87.1% 12.9%	80.0% 20.0%	62.5% 37.5%	50% 50%	77.4% 22.6%	**83.0%** **17.0%**

Figure 29 Confusion diagram for intensity features

The result gained using texture features is shown from figure 30. The highest accuracy achieved by the classifier is 90.8% compared to the three feature sets. Texture features like energy, entropy, correlation, contrast and homogeneity were used as an input to the classier.



Confusion Matrix

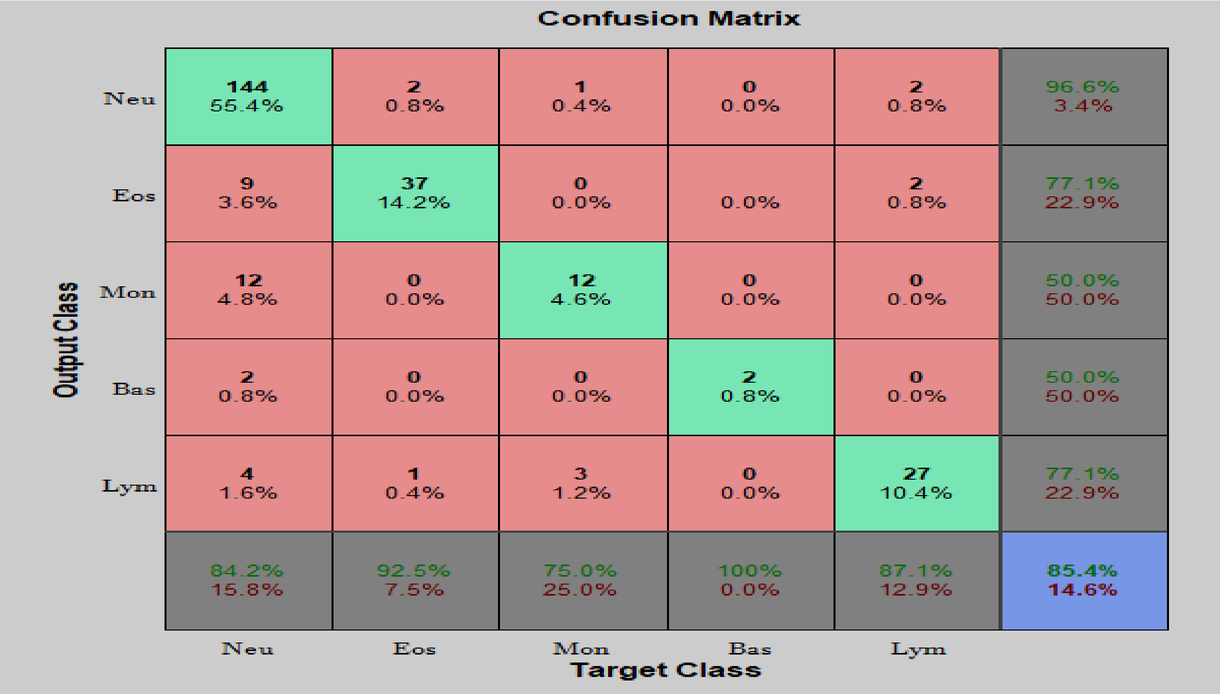
Output Class

	Neu	Eos	Mon	Bas	Lym	
Neu	159 61.2%	2 0.8%	2 0.8%	0 0.0%	3 1.2%	95.8% 4.2%
Eos	4 1.6%	37 14.2%	1 0.4%	0.0%	0 0.0%	88.1% 11.9%
Mon	3 1.2%	0 0.0%	12 4.6%	0 0.0%	2 0.8%	70.6% 29.4%
Bas	1 0.4%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	66.7% 33.4%
Lym	4 1.6%	1 0.4%	1 0.4%	0 0.0%	26 10.0%	93.3% 6.7%
	93.0% 7.0%	92.5% 7.5%	75.0% 25.0%	100% 0.0%	83.9% 16.1%	90.8% 9.2%

Target Class

Figure 30 Confusion diagram for texture features

After we made classification using each feature sets the next step was testing the classifier using the combination of shape with intensity features the following result is achieved.



Confusion Matrix

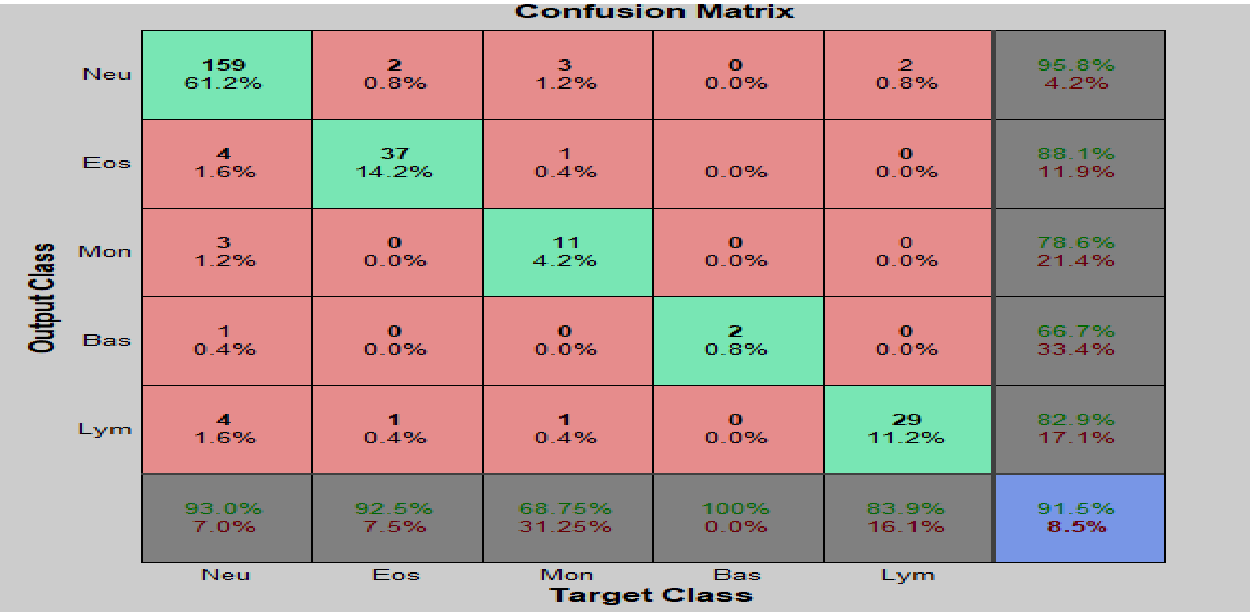
Output Class

	Neu	Eos	Mon	Bas	Lym	
Neu	144 55.4%	2 0.8%	1 0.4%	0 0.0%	2 0.8%	96.6% 3.4%
Eos	9 3.6%	37 14.2%	0 0.0%	0.0%	2 0.8%	77.1% 22.9%
Mon	12 4.8%	0 0.0%	12 4.6%	0 0.0%	0 0.0%	50.0% 50.0%
Bas	2 0.8%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	50.0% 50.0%
Lym	4 1.6%	1 0.4%	3 1.2%	0 0.0%	27 10.4%	77.1% 22.9%
	84.2% 15.8%	92.5% 7.5%	75.0% 25.0%	100% 0.0%	87.1% 12.9%	85.4% 14.6%

Target Class

Figure 31 Confusion diagram for shape with intensity features

In the same way by combining shape and texture features the classifier yields 91.5% accuracy as shown from figure 32.



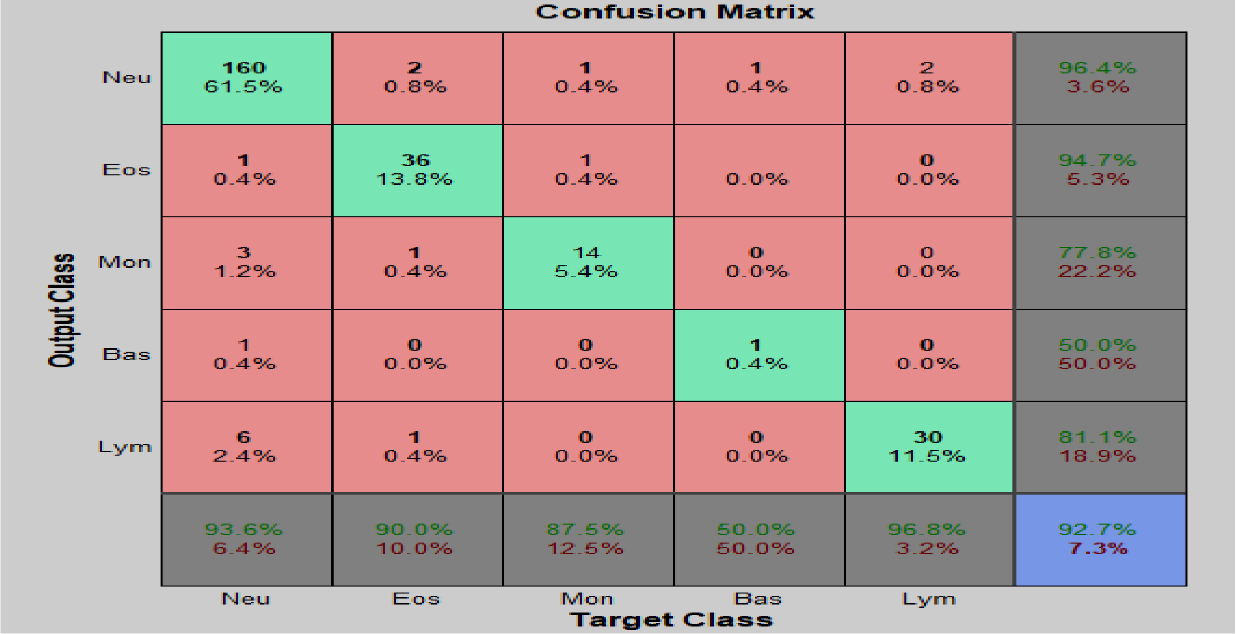
Confusion Matrix

Output Class	Neu	Eos	Mon	Bas	Lym	
Neu	159 61.2%	2 0.8%	3 1.2%	0 0.0%	2 0.8%	95.8% 4.2%
Eos	4 1.6%	37 14.2%	1 0.4%	0.0%	0 0.0%	88.1% 11.9%
Mon	3 1.2%	0 0.0%	11 4.2%	0 0.0%	0 0.0%	78.6% 21.4%
Bas	1 0.4%	0 0.0%	0 0.0%	2 0.8%	0 0.0%	66.7% 33.4%
Lym	4 1.6%	1 0.4%	1 0.4%	0 0.0%	29 11.2%	82.9% 17.1%
	93.0% 7.0%	92.5% 7.5%	68.75% 31.25%	100% 0.0%	83.9% 16.1%	91.5% 8.5%

Target Class

Figure 32 Confusion diagram for texture with shape features

The confusion diagram from figure 33 shows the accuracy achieved using for intensity with texture features.



Confusion Matrix

Output Class	Neu	Eos	Mon	Bas	Lym	
Neu	160 61.5%	2 0.8%	1 0.4%	1 0.4%	2 0.8%	96.4% 3.6%
Eos	1 0.4%	36 13.8%	1 0.4%	0.0%	0 0.0%	94.7% 5.3%
Mon	3 1.2%	1 0.4%	14 5.4%	0 0.0%	0 0.0%	77.8% 22.2%
Bas	1 0.4%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	50.0% 50.0%
Lym	6 2.4%	1 0.4%	0 0.0%	0 0.0%	30 11.5%	81.1% 18.9%
	93.6% 6.4%	90.0% 10.0%	87.5% 12.5%	50.0% 50.0%	96.8% 3.2%	92.7% 7.3%

Target Class

Figure 33 Confusion diagram for intensity with texture features

For the experimentation, different feature sets are used to compare the performance of the classifier with different feature sets. Namely: shape, intensity and texture features for classification and compare with the results obtained for each feature set. The performance comparison for each feature sets is summarized from the table 6.

Table 6 Comparison of the classifier with different feature sets

	Feature sets used						
	Shape	Intensity	Texture	Shape with intensity	Shape with texture	Intensity with texture	All features
Accuracy (%)	86.2	83	90.8	85.4	91.5	92.7	96.5

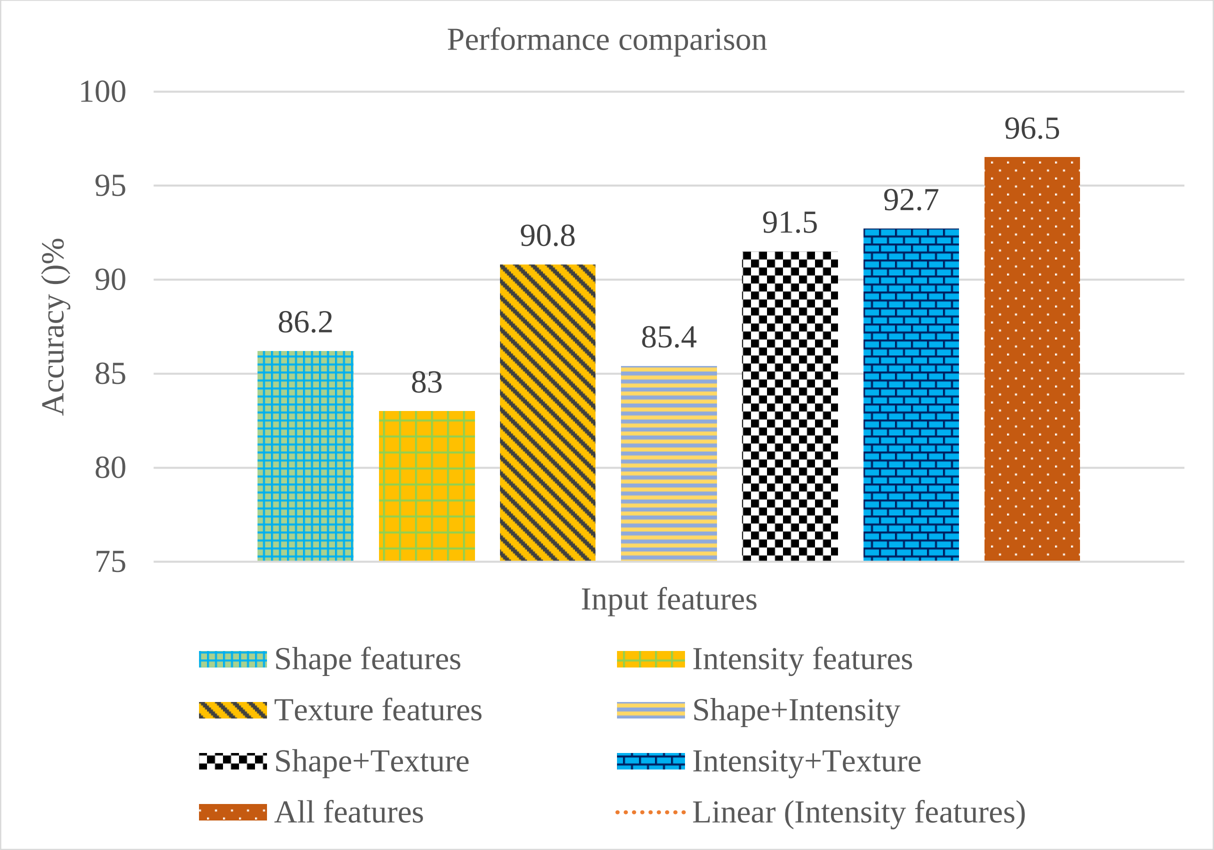


Figure 34 Performance comparison for all features

As the numbers of hidden nodes are achieved through trial and error method the graph below shows the achieved accuracy versus hidden node. The classifier yields a maximum accuracy of 96.5 % with an adequate single hidden layer and 20 nodes.

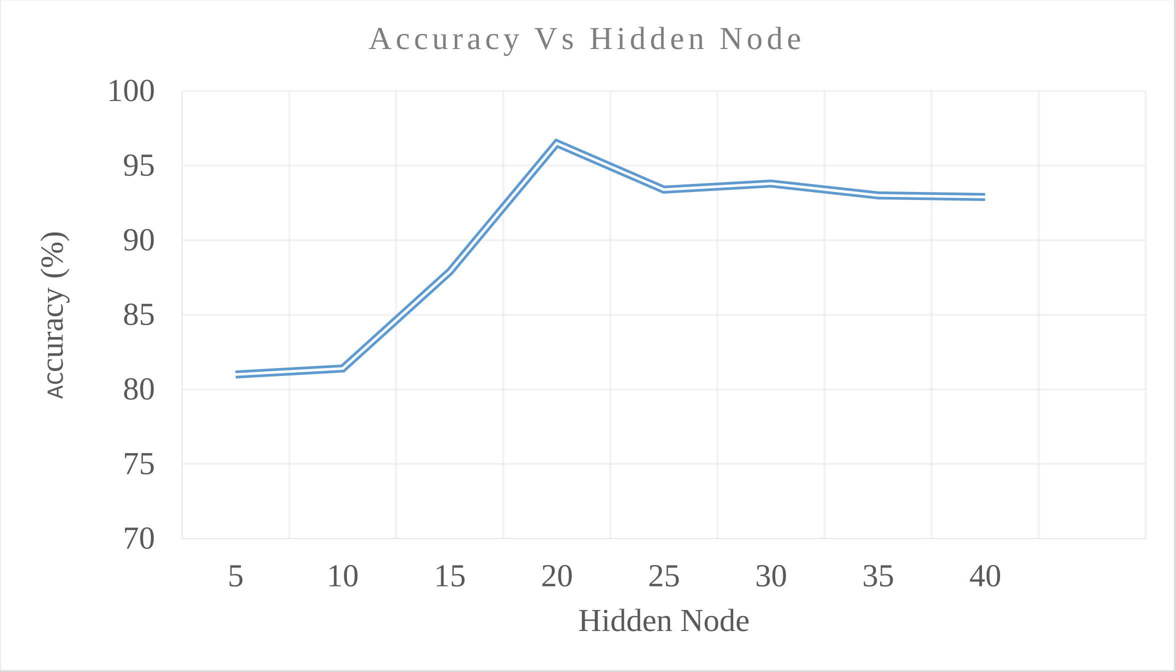


Figure 35 Accuracy vs hidden node

6.4 Answers for Research Questions

In this section, we briefly recall the research questions and provide adequate answers based on the research work.

RQ1: Can the segmentation result affects the performance of the classifier?

The whole work done from the previous sections describes in order to automate the classification problem which is defined from the statement of the problem. For segmentation purpose Otsu thresholding and k-means clustering are used and the accuracy is evaluated. Based on these different features are extracted from the two segmentation techniques. Then finally, the designed and trained neural network model classified the target classes based on the inputs/features provided which is shown using the confusion matrix from Section 6.3. As a result, the designed classifier yields 93.8% classification accuracy when the segmentation accuracy is 88.2%. And also the classifier yields 96.5% classification accuracy when the segmentation accuracy is 91.6%.

Therefore, it can be conclude that the better the segmentation result produces the better classification result.

RQ2: Can the combination of shape, intensity and texture features provide a better classification result?

As it is clearly shown from Section 6.3, the accuracy of the classifier for each feature sets is different. These features are categorized as shape features (area, perimeter, major axis length, minor axis length, eccentricity and circularity), intensity features (mean, variance, standard deviation, skewness, kurtosis and smoothness), and textural features (energy, entropy, correlation, contrast, homogeneity). The designed classifier model as shown from table 6 yields a classification accuracy of 86.2%, 83% and 90.8% for shape, intensity and texture features respectively. This implies that texture features are more significant features to recognize each types of WBCs. Moreover, the use of all features (shape, intensity and texture) combined together which fed to the neural network gives a better result which is 96.5% as it is shown from table 5 because of the neural network gets the valuable information from each feature sets that allows to uniquely recognize for each types of white blood cells those differ from each other in their shape and texture.

Chapter Seven

Conclusions and Recommendations

In this chapter, concluding remarks about the findings of the thesis and recommendations for future works based on the ideas discussed here or possible alternatives that could strengthen the issues in this thesis are presented.

7.1 Conclusions

This thesis addresses the recognition task for white blood cell types. The main processes in developing the system are segmentation of the image, feature extraction and classification.

The system was experimentally analyzed with 260 microscopic images for the classification of the white blood cell types. To acquire region of interest all of 260 microscopic images were segmented using two algorithms. Subsequently, various feature vectors were extracted from the segmented image. After the extraction of feature vectors the classification of each microscopic image for a particular category at the next step was performed using ANN model.

Since the choice of features and the type of classifier play a significant role in classification accuracy results, in this thesis, the MLP trained by BP algorithm was used to classify five types of WBCs. The extracted 17 features are used as an input to the neural network. These features are categorized as shape features, intensity features and textural features. All of these extracted features has been used as an input to the classifier.

The designed classifier model yields a classification accuracy of 96.5 % using all extracted features. Additionally the classier also yields an accuracy of 86.2%, 83% and 90.8% for shape, intensity and texture features respectively. Here, it implies that textures features are more significant than the other features to classify each types of white blood cells. Besides, it can be concluded that the use of all type of features (shape, intensity and texture) together yields better classification result than the individual feature sets. Additionally, since it is desired to have as few features as possible the designed neural network classifier yielded a nice classification accuracy with less number of features. In general, it is understood that this research automates the manual system to be easier for the segmentation and classification task, and increases the speed to find the results of medical analysis.

7.2 Recommendations

The following main points are recommended as a future work that can extend and improve the current work.

- This work is focused on the classification of white blood cell from microscopic images. But detail research work is required to classify white blood cell type and the other blood cell type like red blood cells and platelets. Thus, the segmentation approach will include all of the blood cell types.
- In this work, the main objective is to classify WBC from microscopic images. But it is likely to provide the condition of the cells as being healthy or diseased after classifying their categories.
- Artificial neural network with back propagation algorithm has been used for classification of the five classes of white blood cells from the inputs of the extracted features after the segmentation of microscopic image. Other classifiers like deep learning should be considered in order to see their impact on the performance of the system.

References

[1] P.S. Hiremath, P.Bannigidad and Sai Geeta,"Automated Identification and classification of White blood cell in digital microscopic Images", Recent Trends in Image Processing and Pattern Recognition, Vol.2,pp.59-63,2014.

[2] A. Gautam, P. Singh, B. Raman, H. Bhadauria, "Automatic Classification of Leukocytes Using Morphological Features and Naïve Bayes Classifier" IEEE Region 10 Conference (TENCON) 2016.

[3] A. N. Abdul, M.Y. Mashor and R. Hassan, " Classification of Acute Leukaemia Cells using Multilayer Perceptron and Simplified Fuzzy ARTMAP Neural Networks", The International Arab Journal of Information Technology, Vol. 10, No. 4, July 2013.

[4] M. Saravanan, C. Surya, R.Thirumurugan and P. Saravanakumar, "Automatic Classification of Leukocytes with Neural Network and Fuzzy Interference Systems", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Special Issue 1, March 2017.

[5] P. Tabrizi, S. Rezatofighi, M. Yazdanpanah, "Using PCA and LVQ neural network for automatic recognition of five types of white blood cells". Engineering in Medicine and Biology Society (EMBC), Annual international conference of the IEEE, Aug 31–Sept 4, 2010.

[6] F. Scotti and V. Piuri, "ALL-IDB: the Acute Lymphoblastic Leukemia Image Data Base for Image Processing", In Proceedings of the 18th IEEE ICIP International Conference on Image Processing, IEEE Publisher, pp. 2045-2048, Brussels, Belgium, September 11-14, 2011.

[7] M. Z. Othman. and Ali, Alaa, B.," Segmentation and Feature Extraction of Lymphocytes WBC using Microscopic Images", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 3 Issue 12, December-2014.

[8] W.Yu, C.Yang, L. Zhang, H. Shen et al "Automatic Classification of Leukocytes Using Deep Neural Network", IEEE 12th international conference on ASIC (ASICON) 2017.

[9] J. Zhao, M. Zhang, Z. Zhou, J. Chu and F. Cao, "Automatic Detection and Classification of Leukocytes Using Convolutional Neural Networks", Medical and Biological Engineering, Vol. 55 No.8, 2016.

[10] V. Magudeeswaran, P. Karthikeyan and P. Thirumurugan, "Feature Extraction and Classification of Blood Cells Using Artificial Neural Network", Department of Electronics and Communication Enginering, PSNA College of Engineering and Technology, Dindigul, Tamil nadu, India, 2012.

[11] J. Pradeep, E. Srinivasan, S. Himavathi, "Neural Network Based Recognition System Integrating Feature Extraction and Classification for English Handwritten", Vol. 25, No. 2, (May 2012) 99-106.

[12] A. Ibrahim, "An Application of Artificial Neural Network Classifier for Medical Diagnosis", Faculty of Computer Sciences and Information Technology University Tun Hussein Onn Malaysia, July 2013.

[13] S. Osowski, R.Siroic, and K. Siwek, "Application of Support Vector Machine and Genetic Algorithm for Improved Blood Cell Recognition", IEEE Transactions on Instrumentation and Measurement, 2009.

[14] M. Alsmadi, K. Omar and S. Noah "Back Propagation Algorithm: The Best Algorithm Among the Multi-layer Perceptron Algorithm", Department of System Science and Management, University Kebangsaan Malaysia, Kuala Lumpur, Malaysia, International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.

[15] Al-Allaf, "Fast BackPropagation Neural Network Algorithm for Reducing Convergence Time of BPNN Image Compression", International Conference on Information Technology and Multimedia (ICIM), pp. 1 – 6, 2011.

[16] Y.S. Frank, "Image Processing and Pattern Recognition: Fundamentals and Techniques".,#pp 269-293**.** A John Wiley and Sons, INC., Publications, IEEE Press, 2010.

[17] L. Fausett, "Fundamentals of neural networks: architectures, algorithms, and applications", Upper Saddle River: Prentice-Hall Inc.; 1994.

[18] G. Ongun, U. Halici, K. Leblebicioglu, V. Atalay, M. Beksac, and S. Beksac "Feature Extraction and Classification of Blood Cells for an Automated Differential Blood Count System," in Proceedings of the International Joint Conference on Neural Networks, USA, pp.2461-2466, 2001.

[19] S. F. Bikhet and A. M. Darwish and H. A. Tolba and S. I. Shaheen, "Segmentation and Classification of White Blood Cells," IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 2259-2261, 2000.

[20] D.M. Sabino, Costa, Rizzatti E.G, "Leukocyte recognition using Morphometry, texture and color", IEEE International Symposium on Biomedical Imaging, 2004.

[21] N. Otsu, "A thresholding selection method from gray-level histogram," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 8, pp. 62-66, 1978.

[22] Q. Liano and Y. Deng, "An Accurate Segmentation Method for White Blood Cell Images", IEEE Conf. Biomedical Imaging, 2002, pp. 245-248.

[23] F. Sadeghian, Z. Seman, A. R. Ramli et al., "A framework for white blood cell segmentation in microscopic blood images using digital image processing," Biological procedures online, vol. 11, no. 1, pp. 196, 2009.

[24] Five types of white blood cells, www.livestrong.com . Retrieved: February, 2018.

[25] S. Pogula and Sriram, "Developing Neural Network Applications Using LABVIEW", a thesis submitted to University of Missouri-Columbia, July 2005.

[26] G. Charu, "Implementation of Back Propagation Algorithm (of neural networks) in VHDL", thesis submitted to Thapar Institute of Engineering & Technology, June, 2006.

[27] A.F. James, M.S. David, "Neural Networks Algorithms, Applications, and Programming Techniques," Loral Space Information Systems and Adjunct Faculty, School of Natural and Applied Sciences University of Houston at Clear Lake, Addison-Wesley Publishing Company, 1991.

[28] P. Rajesh, Y. Jihoon and H. Vasant, "Constructive Neural-Network Learning Algorithms for Pattern Classification," IEEE Transactions on Neural Networks, Vol. 11, No. 2, March 2000.

[29] H. Hengen, O. Gabel, A. Hajra, T. Paulus, M. Ross and S. Spoor, "Development of a system for the analysis and classification of blood and bone marrow cell images to support morphological diagnosis of leukemia," Kaiserslautern, Germany, February 2004.

[30] D.C. Huang, K.D. Hung, and Y.K. Chan, "A computer assisted method for leukocyte nucleus segmentation and recognition in blood smear images," The Journal of Systems and Software, Elsevier Science Publication, Vol. 85, Issue 9, September, 2012.

[31] A. Trupti and Kulkarni-Joshi, “A Fast Segmentation Scheme for Acute Lymphoblastic Leukemia Detection”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 2, February 2014.

[32] "Blood image dataset", Mathworks File Exchangewebsite, [Online]. Available:http://www.mathworks.com/matlabcentral/fileexchange/36634. Retrieved, January 2018,

[33] A. Abraham, “Artificial Neural Network”, Oklahoma State Univ., Stillwater, USA.

[34] N. Michael, “Artificial Intelligence: A Guide to Intelligent Systems”, 2nd ed., 2005.

[35] Samarasighe, “Neural Network for Applied Science and Engineering: From Fundamentals to Complex Pattern Recognition”, New York: Auerbach Publication, 2006.

[36] W.S. McCulloch, and W. Pitts, “A Logical Calculus of the Ideas Immanent in Nervous Activity”, Bulletin of Mathematical Biophysics, 5, 115–137, 1943.

[37] P. Z. Guoqiang “Neural Networks for Classification: A Survey”. IEEE Transaction on Systems, Man and Cybernetics- Part C: Applications and Reviews, Vol. 30, No.4, 2000.

[38] H.T. Madhloom, S.A. Kareem, H. Ariffin, A.A. Zaidan, H.O. Alanazi and B.B. Zaidan, “An Automated White Blood Cell Nucleus Localization and Segmentation using Image Arithmetic and Automatic Threshold”, Journal of Applied Sciences, Volume 10 (11): 959-966, 2010.

[39] A. Gebejes, A. and Huertas, R.," Texture Characterization Based on Grey-Level Co-occurrence Matrix", Granada: Conference of Informatics and Management Sciences, 2013.

[40] J. Gong, L. Li, and W. Chen, “Fast recursive algorithms for two-dimensional thresholding,” Pattern Recognition, vol. 31, no. 3, pp. 295–300, 1998.

[41] S. K. Shehroz and A. Amir, “Cluster Centre Initialization Algorithm for *K*-means Cluster”, In Pattern Recognition Letters, pp. 1293–1302, (2004).

[42] P. Vincenzo, S. Fabio, “Morphological Classification of Blood Leukocytes by Microscope images”, IEEE International conference on computational Intelligence for Measurement Systems and Applications Boston. MD, USA, 14-16 July 2004.

[43] A. Khashman, “Face recognition using neural networks and pattern averaging”, In: Lecture notes on computer science. Berlin Heidelberg-New York, p. 98–103 (vol. 3972). 2006.

[44] J. de Villiers and E. Barnard, "Back propagation neural nets with one and two hidden layers," IEEE Trans. Neural Net., vol. 4, no. 1, pp. 136–141, Jan. 1993.

[45] Q. Zheng, B. K. Milthorpe and A. Jones, "Direct Neural Network Application for Automated Cell Recognition", Cytometry Part A, Vol. 57A, pp 1-9, 2004.

[46] S. Manisha, "Detection and Segmentation of WBC cells using Image Processing Technique", Department of Electronics and Telecommunication Engineering, Ponda, Goa, India, International Journal of Technology and Science, ISSN (Online) 2350-1111, (Print) 2350-1103 Volume IX, Issue 1, pp. 56-58, 2016.

[47] Q. Zheng, B. K. Milthorpe, and A. Jones, "Direct Neural Network Application for Automated Cell Recognition", Wiley-liss, Inc, Cytometry Part A, Vol. 57A, pp 1-9, 2004.

[48] R. Kaur, "Artificial Neural Network Learning Enhancement using Bacterial Foraging Optimization Algorithm," vol. 102, no. 10, pp. 27–33, 2014.

[49] L. Jing, C. Ji-hang, Jing-yuan Shi, and H. Fei, "Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement", Aviation University of Air Force, Changchun China, 2012.

[50] S. Rashmi and S. Mandar, "Textural Feature Based Image Classification Using Artificial Neural Network", Advances in Computing, Communication and Control. Communications in Computer and Information Science, vol 125. Springer, 2011.

Appendices

Appendix A: Sample images of the dataset

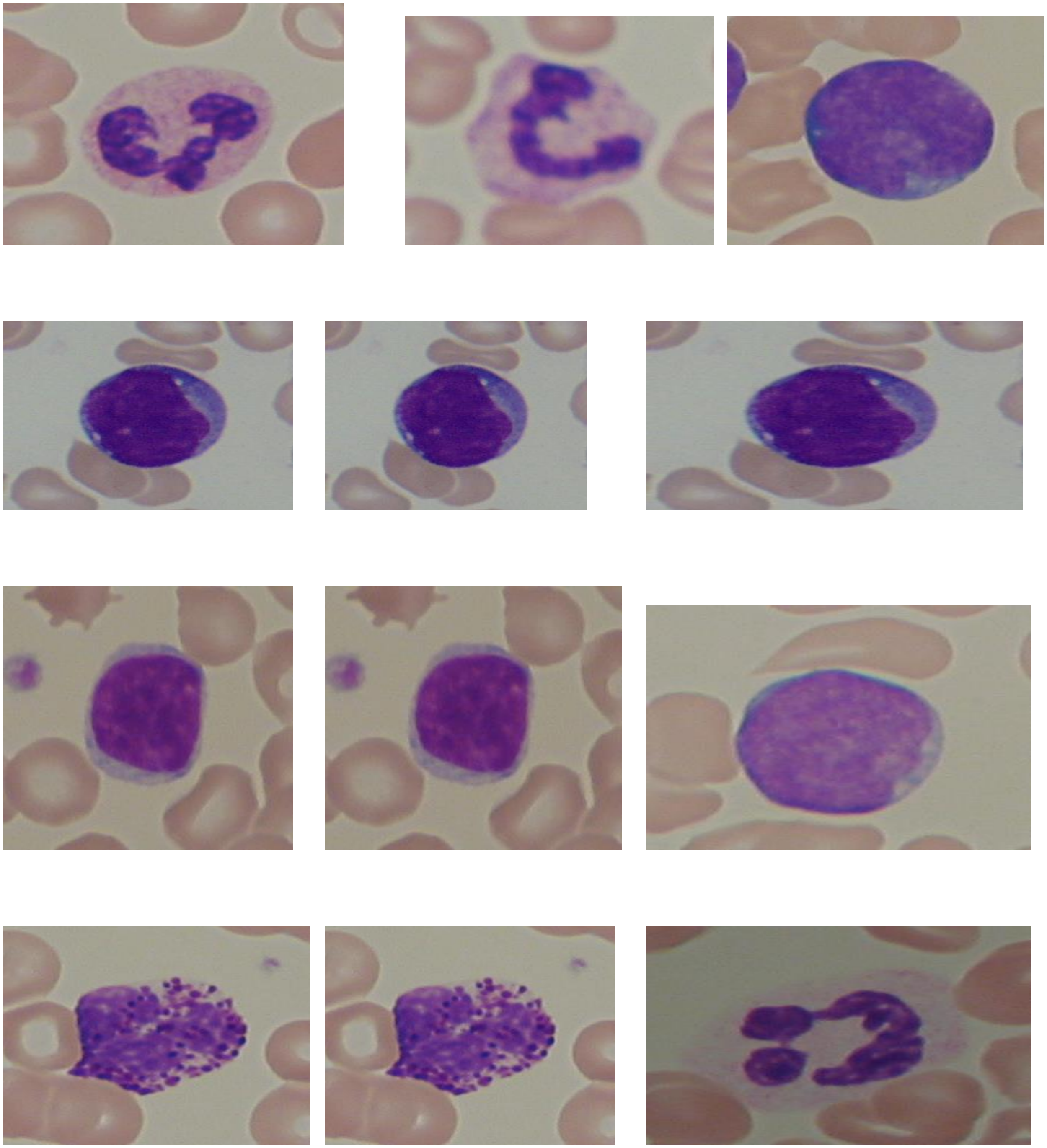
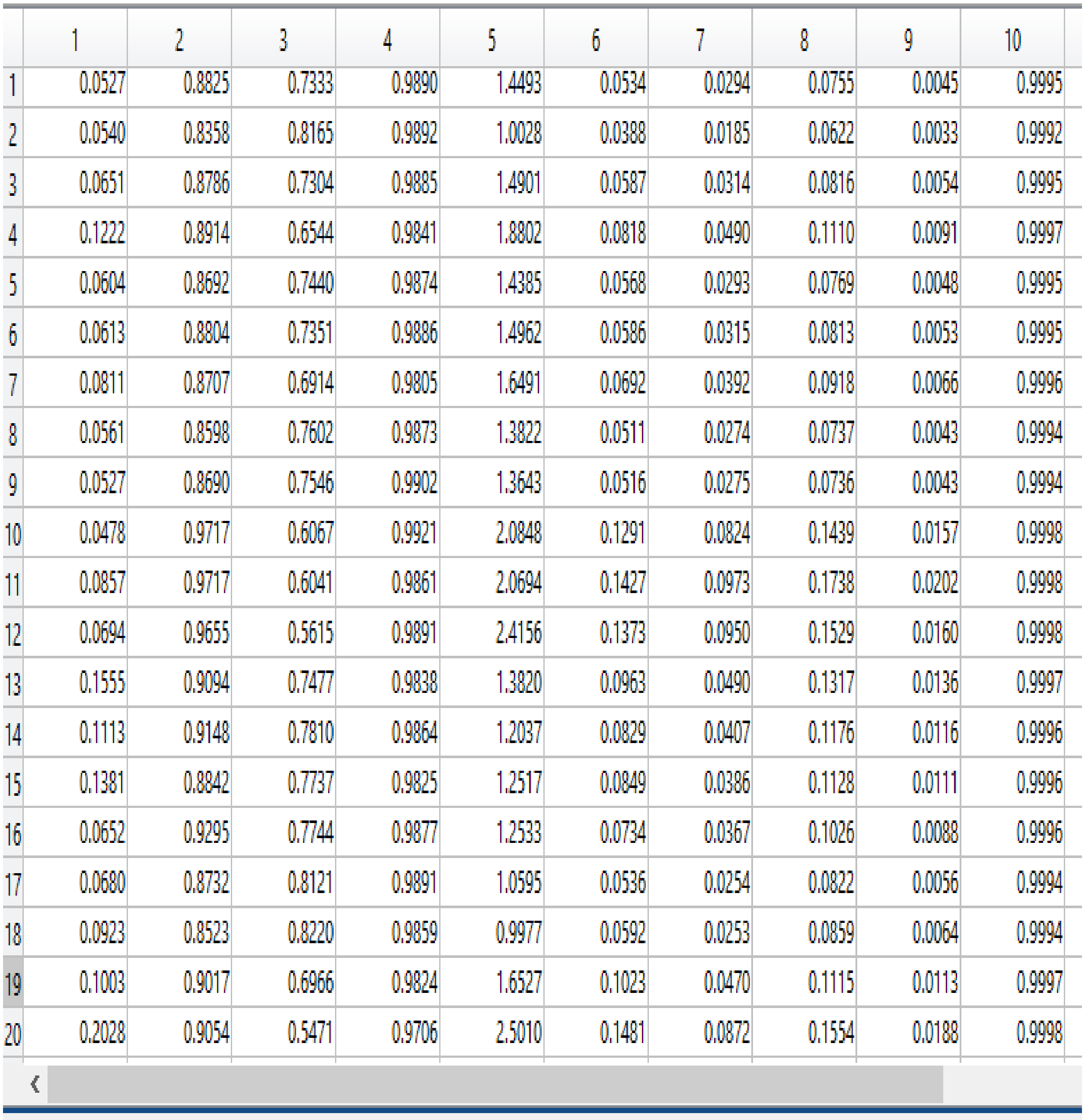


Figure 36 Sample images of the dataset

Appendix B: Sample extracted features

The table below shows some of the extracted features for the first 20 images and 10 features for each using GLCM.

Table 7 Sample extracted features using GLCM



	1	2	3	4	5	6	7	8	9	10
1	0.0527	0.8825	0.7333	0.9890	1.4493	0.0534	0.0294	0.0755	0.0045	0.9995
2	0.0540	0.8358	0.8165	0.9892	1.0028	0.0388	0.0185	0.0622	0.0033	0.9992
3	0.0651	0.8786	0.7304	0.9885	1.4901	0.0587	0.0314	0.0816	0.0054	0.9995
4	0.1222	0.8914	0.6544	0.9841	1.8802	0.0818	0.0490	0.1110	0.0091	0.9997
5	0.0604	0.8692	0.7440	0.9874	1.4385	0.0568	0.0293	0.0769	0.0048	0.9995
6	0.0613	0.8804	0.7351	0.9886	1.4962	0.0586	0.0315	0.0813	0.0053	0.9995
7	0.0811	0.8707	0.6914	0.9805	1.6491	0.0692	0.0392	0.0918	0.0066	0.9996
8	0.0561	0.8598	0.7602	0.9873	1.3822	0.0511	0.0274	0.0737	0.0043	0.9994
9	0.0527	0.8690	0.7546	0.9902	1.3643	0.0516	0.0275	0.0736	0.0043	0.9994
10	0.0478	0.9717	0.6067	0.9921	2.0848	0.1291	0.0824	0.1439	0.0157	0.9998
11	0.0857	0.9717	0.6041	0.9861	2.0694	0.1427	0.0973	0.1738	0.0202	0.9998
12	0.0694	0.9655	0.5615	0.9891	2.4156	0.1373	0.0950	0.1529	0.0160	0.9998
13	0.1555	0.9094	0.7477	0.9838	1.3820	0.0963	0.0490	0.1317	0.0136	0.9997
14	0.1113	0.9148	0.7810	0.9864	1.2037	0.0829	0.0407	0.1176	0.0116	0.9996
15	0.1381	0.8842	0.7737	0.9825	1.2517	0.0849	0.0386	0.1128	0.0111	0.9996
16	0.0652	0.9295	0.7744	0.9877	1.2533	0.0734	0.0367	0.1026	0.0088	0.9996
17	0.0680	0.8732	0.8121	0.9891	1.0595	0.0536	0.0254	0.0822	0.0056	0.9994
18	0.0923	0.8523	0.8220	0.9859	0.9977	0.0592	0.0253	0.0859	0.0064	0.9994
19	0.1003	0.9017	0.6966	0.9824	1.6527	0.1023	0.0470	0.1115	0.0113	0.9997
20	0.2028	0.9054	0.5471	0.9706	2.5010	0.1481	0.0872	0.1554	0.0188	0.9998

Declaration

I, the undersigned, hereby declare that this thesis is my original work performed under the supervision of Dr. Eng. Getachew Alemu, has not been presented as a thesis for a degree program in any other university and all sources of materials used for the thesis are duly acknowledged.

Kindalem Sirak	____________________________
Name	Signature