

Jimma University

Jimma Institute of Technology

School of Biomedical Engineering

MSc. In Biomedical Engineering (Biomedical Imaging)

A Master's Thesis Report

On

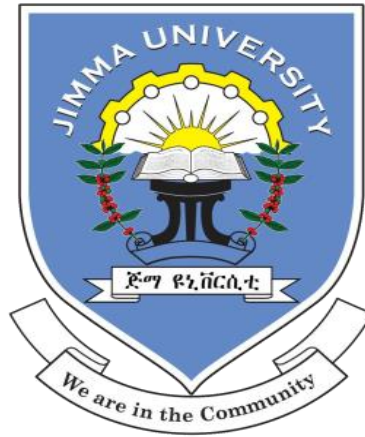
***Automatic Detection, Classification, Counting and Quantification of
Morphological Parameters of Blood Cells from Microscopic Images Using
YOLOv2***

A thesis report submitted to school of graduate studies of Jimma University in partial fulfilment for the degree of Master of Science in Biomedical Engineering (biomedical imaging)

By: Abel Worku Tessema

Jimma, Ethiopia

January, 2019



Jimma University

Jimma Institute of Technology

School of Biomedical Engineering

MSc. In Biomedical Engineering (Biomedical Imaging)

A Master's Thesis Report

On

***Automatic Detection, Classification, Counting and
Quantification of Morphological Parameters of Blood Cells
from Microscopic Images using YOLOv2***

A thesis proposal submitted to school of graduate studies of Jimma University in partial fulfilment for the degree of Master of Science in Biomedical Engineering (biomedical imaging)

By: Abel Worku Tessema

Advisors:

- 1. Timothy Kwa (PhD.)**
- 2. Mohammed Ali (MSc.)**

Jimma, Ethiopia

January, 2019

Declaration

This thesis entitled as “*Automatic detection, classification, counting and quantification of morphological parameters of blood cells from microscopic images using YOLOv2*” is my original work and has not been presented for a degree in any other university.

Name: Abel Worku Tessema

Signature

Date

We confirmed below that this research is approved as MSc. Thesis for the student, as university advisors.

1. Dr. Timothy Kwa (PhD.)

Signature

Date

2. Mohammed Aliy (MSc.)

Signature

Date

Chair of Biomedical Imaging

Hakkins Raj (Associate Professor)

Signature

Date

External examiner

Name:

Signature

Date

Internal examiner

Name:

Signature

Date

Abstract

It is well known that the current methods used for blood cell segmentation and counting are very tedious and are highly prone to different sources of errors especially if the density of cells is very high, and overlapping of cells are more common. In addition, those techniques did not provide full information related to blood cells like their shape and size, which play important roles in clinical investigation of serious blood related diseases such as leukemia and sickle cell disease.

Among currently emerging techniques for blood cell detection and counting, image processing and neural-network-based methods are mostly used to automate the tasks and provide accurate diagnostic results. This research used YOLOv2 model, which is currently the state of the art for real time object detection and classification. The model is faster and more accurate than any other currently available real time object detection networks such as CNN, RCNN, fast-RCNN due to its unique features consisting of global vision, optimization and others. It is used to automatically detect classify, count and measure the morphological parameters of blood cells.

The model was trained on 1558 images and 2733-labeled blood cells with network a resolution of 608x608, different batch sizes (16, 32, 64 and 128), a learning rate for more than 15000 number of steps in which the loss converged from 234.5 to 1.54. Non-maximum suppression was applied on the output of the network to remove duplicated detections of objects based on class label, score, overlapping area and non-maximum suppression threshold and finally it was tested on 26 images containing 1454 red blood cells, 159 platelets, 3 basophils, 10 eosinophils, 24 lymphocytes, 13 monocytes and 28 neutrophils. The network achieved detection and segmentation of blood cells with average accuracy of 80.6% and precision of 88.4%.

After classification and counting, measurement of area, diameter and aspect ratio of cells were calculated using image processing and the results were evaluated using a manual method (ImageJ), resulting in a mean accuracy of 92.96%, 91.96%, 88.736% and 92.7% in the measurement of area, aspect ratio, diameter and counting of cells respectively.

Key Words: classification, counting, YOLOv2, image processing, non-maximum suppression, morphological parameters, blood cells

Acknowledgement

First of all, I would like to give a heartfelt thanks to my Almighty God, for whom without his support it was unthinkable to deal with anything. Secondly, I would like to express my gratitude to Jimma University, Jimma Institute of Technology, School of Biomedical Engineering for providing me the opportunity to study master program.

I would also extend my special thanks to my advisors Dr. Timothy Kwan (PhD), and Mohammed Aliy (MSc.) for their encouragements and help to do my thesis on this area and for being my advisors. Then I need also to thank Ind Ing. Wasihun Alemayehu, Hakkins Raj (Associate professor), Dr. Bheem, Dr. Bira(from JUCAVM), Dr. Alemseged and Dr. Tilahun (microbiologist in JUMC) and staffs of hematology department of JUMC for their valuable guidance and helping me in all activities of the research.

Lastly but not least I would like to say thank you for all my classmates for sharing their idea, suggestions and experiences while doing my thesis proposal.

Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgement	iii
List of Figures and Tables.....	vi
List of Figures	vi
List of Tables	vii
Acronyms.....	viii
CHAPTER ONE	1
Introduction.....	1
1.1. Background	1
1.1.1. Blood Cells.....	2
1.1.2. Manual Blood Cell Counting Using Microscope	4
1.1.3. Image Processing for Blood Cell Counting and Measurement of Morphological Parameters	6
1.1.4. Artificial Neural Network	6
1.2. Statement of The Problem	9
1.3. Research Objectives.....	10
1.3.1. General Objective.....	10
1.3.2. Specific Objectives.....	10
1.4. Significance of The Study.....	11
1.5. The Scope of The Research	11
CHAPTER TWO	12
2. Related Works in Analysis of Microscopic Images of Blood Cells	12
2.1. Introduction.....	12
2.2. Major Gaps of Researches	15
CHAPTER THREE	16
3. Materials and Methodology.....	16
3.1. Research Methods.....	16
3.1.1. Image Data Collection.....	16
3.1.2. Pre-Processing	16
3.1.3. Comparing Current Methods of Object Detectors	17
3.1.4. YOLOv2 Real Time Object Detection.....	19
3.1.4.1. Background of YOLOv2	19
3.1.4.2. Loss Functions of The Network	22

3.1.4.3.	Non-Maximum Suppression.....	23
3.1.4.4.	Hyper Parameters of The Network.....	24
3.2.	Overall Block Diagram of Methodology.....	27
3.3.	Materials.....	27
CHAPTER FOUR.....		29
4.	Experiment, result and discussion.....	29
4.1.	Experiment and Results.....	29
4.1.1.	Image Collection, Preparation and Pre-processing.....	29
4.1.2.	Image Annotation.....	29
4.1.3.	Network Configuration.....	31
4.1.4.	Training The Network.....	31
4.1.4.1.	Performance Analysis of Training Using Error Convergence.....	32
4.1.5.	Detection, Segmentation, Counting and Quantification of Morphological Parameters.....	32
4.1.5.1.	Detection and Classification of Blood Cells.....	32
4.1.5.2.	Counting of Blood Cells.....	38
4.1.5.3.	Quantification of Morphological Parameters.....	39
CHAPTER FIVE.....		42
5.	Conclusion and Recommendation.....	42
5.1.	Conclusion.....	42
5.2.	Recommendation.....	43
List of References.....		44
APPENDICES.....		48
Appendix A: Sample Training Data.....		48
Appendix B: Real Number of Cells in 26 Test Images and Detected Cells.....		49
Appendix C: Source Code.....		51
Appendix D: Sample Images Showing Output of the Network.....		55
Appendix E: Comparison of a System Output With Manual Methods.....		56

List of Figures and Tables

List of Figures

Figure 1: manual methods of blood cell counting	1
Figure 2: Red blood cells	2
Figure 3: different types of white blood cells	3
Figure 4: platelets, RBCs and Neutrophil (WBC)	4
Figure 5: neural network architecture	7
Figure 6: image pre-processing on image dataset of the experiment	17
Figure 7: yolo object detection method, from image grids to feature map[34].....	21
Figure 8: yolo network with stride of 32, class of 7 and resolution of 608x608.....	22
Figure 9: IOU for selecting best responsible bounding box where BB (is area covered by bounding box) and (GT) is area covered by ground truth	22
Figure 10: Non Maximum Suppression used to remove multiple detection per object	24
Figure 11: over all block diagram of the research	27
Figure 12: image annotation tool and produced xml file	30
Figure 13: error convergence of training	32
Figure 14: detections and counting of a network at step number of 3,000	33
Figure 15: detection, segmentation and counting of network at step number of 14,500.....	33
Figure 16: output of the network at step number of 16, 998.....	34
Figure 17: accuracy performance of the system	36
Figure 18: precession performance	36
Figure 19: recall of the network	37
Figure 20: image of stage micrometer seen under microscope of 100 magnification	39
Figure 21: performance of system in measurement of area	40
Figure 22: The performance comparison of the system in measuring aspect ratio of blood cells	41
Figure 23: The performance comparison of the system in measuring average diameter of blood cells in μm	41
Figure 24: partial view of xml files containing annotations	48
Figure 25: sample image used for training (training dataset)	48
Figure 26: diameter measurement	55
Figure 27: aspect ratio measurement.....	55

List of Tables

Table 1: comparison of neural networks for real time object detection 18

Table 2: hardware and software materials used for the research28

Table 3: Total number of images in different classes/groups and their sources29

Table 4: total number of labeled objects used for the training from different classes.30

Table 5: confusion matrix showing the performance of the network output34

Table 6: overall network performance35

Table 7: performance of the network in counting classes of blood cells for 26 images:
.....38

Table 8: image calibration/scaling39

Table 9: list of images containing different blood cells used for testing.50

Table 10: Total number of cells in each class detected by the network per image.....51

Table 11: The performance comparison of the system in measuring area of blood cells
in μm^2 56

Table 12: The performance comparison of the system in measuring aspect ratio of
blood cells56

Table 13: The performance comparison of the system in measuring average diameter
of blood cells in μm 57

Acronyms

ANN → Artificial Neural Network

CBC → Complete Blood Count

CNN → Convolutional Neural Network

GB → Giga Byte

IEEE → Institute of Electrical and Electronics Engineers

JUMC → Jimma University Medical Center

MP → Mega Pixel

NN → Neural Network

RAM → Random Access Memory

RBC → Red Blood Cell

R-CNN → Region based convolutional Neural Network

ReLU → Rectified Linear Unit

SVM → Support Vector Machine

TB → Tera Byte

VGG → Visual Geometry Group

YOLO → You Only Looks Once

CHAPTER ONE

Introduction

1.1. Background

It has been 2000 years since people started to see small things using optical lenses and the invention of microscope vastly improved the way we understand the world. After its invention, different improvements were made for different applications and increasing the quality of result. However, nowadays, the much attention has been drawn to a new microscope applications and enhancing the diagnostic utility through techniques including image processing and artificial neural networks. For example, cell counting, detection and segmentation are new areas in which microscopes are enhanced through software while using the same hardware design [1].

It is known that the current methods used for blood cell segmentation and counting are very tedious and are prone to different sources of errors like measurement error. Most of the techniques use averaging to suppress the effect of variation, e.g. performing cell counting using the grids of a hemocytometer, as shown by the figure 1, and taking the mean population density to estimate the total number of cells [2]. Manual methods are not accurate enough to make appropriate clinical decisions [3]. First, since they are manual methods, the tasks are more laborious, and are exposed to measurement error. Secondly, the techniques are not more appropriate if the number density is very high, and the overlapping of cells are more common [4][5]. In addition those techniques do not provide complete information in regard to blood cells i.e. their shape and size which are more important in clinical investigation of blood related diseases.

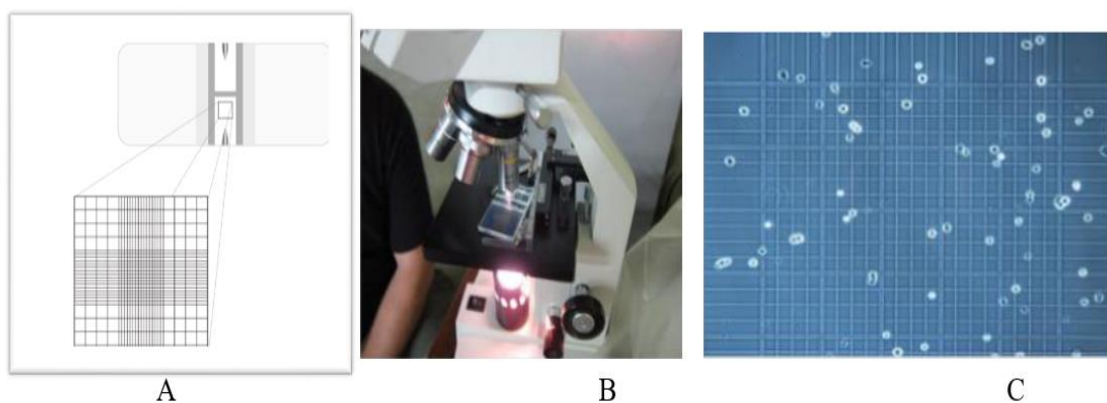


Figure 1: manual methods of blood cell counting

A, Hemocytometer B, microscope C, image of cells seen using microscope
Before going in-depth about cell segmentation, counting and quantification of morphological parameters, it is better first to see blood cell types and their properties.

1.1.1. Blood Cells

Hematologists generally classify three types of blood cells in our body, which are red blood cells (Erythrocyte), white blood cells (leukocytes) and platelets (thrombocytes)

Erythrocyte

Red blood cells are types of cells that circulate through the body to provide oxygen and are most abundant cells in blood accounting for 40-45% of blood volume. They are biconcave disc-shaped thinning in the middle and with average size of around 7-8 micrometer as shown by the figure 2 [6]. The normal count in male adult human being is around 4.7-6.1 million cells per microliter and in female it ranges from 4.2-5.4 million cells per microliter. Any change from the normal count as well as shape and size provides information to diagnose blood-related diseases including anemia, leukemia and others [4], [7], [8].



Figure 2: Red blood cells

This an image of blood cells viewed under a microscope taken from JUMC. The red blood cells are biconcave discs that are present in large concentrations in the body.

Leukocytes

White blood cells or leukocytes are cells that are main actors of the immune system, protecting our body from external invaders. The normal white blood cell count in an

adult human ranges from 5000-7000 cells per mm^3 . Accounting for only 1% of total blood volume. They are categorized in to five, varying in terms of function, shape, as well as number density, which are Neutrophil, Eosinophil, Basophil, Monocyte, and Lymphocyte. Neutrophil is the most common type of WBC, accounting around 55-70% of total WBC count, having an average diameter of 12-14micrometer and having an average lifespan of less than one day [7], [9]. Any deviation from the normal range indicates the presence of infection, tissue damage or other diseases including leukemia. Therefore, it is not only the total number of WBC that provides clinically relevant information, but also the differential count of the individual types of WBC provides specific information related to the patient; because each types of white blood cells are specific for a given disease [10]. In addition, the morphology of the white blood cells indicates their state like reactive or normal depending on the patient status [11]. The following figure (figure 3) indicates five types of white blood cells seen under compound light microscope taken from JUMC.

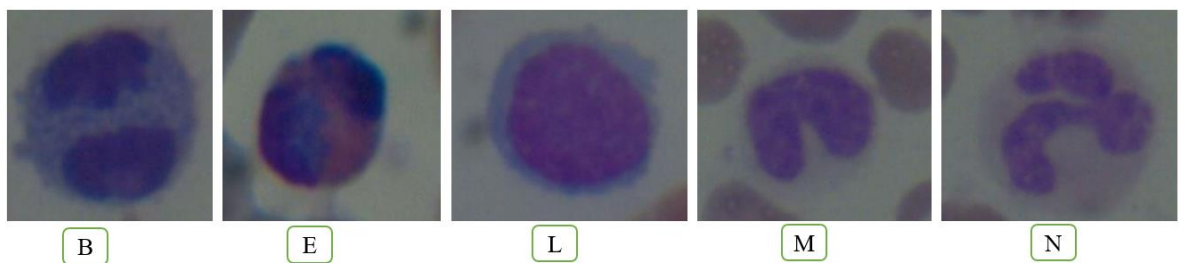


Figure 3: different types of white blood cells

B: Basophils, E: Eosinophils L: Lymphocyte, M: Monocyte and N: neutrophil

Thrombocytes

Platelets or thrombocytes are fragments of cells used for blood clotting. They are produced when megakaryotes break into pieces to produce from 2000-3000 platelets and they are like disc shaped and around $2-4 \mu\text{m}$ in diameter as shown by the figure 4 [12]. Low platelet counts can cause the patient to bleed without clotting while high counts may cause spontaneous blood clotting and thrombosis. Figure 4 shows the images of platelets and other blood cells seen under microscope taken from JUMC.

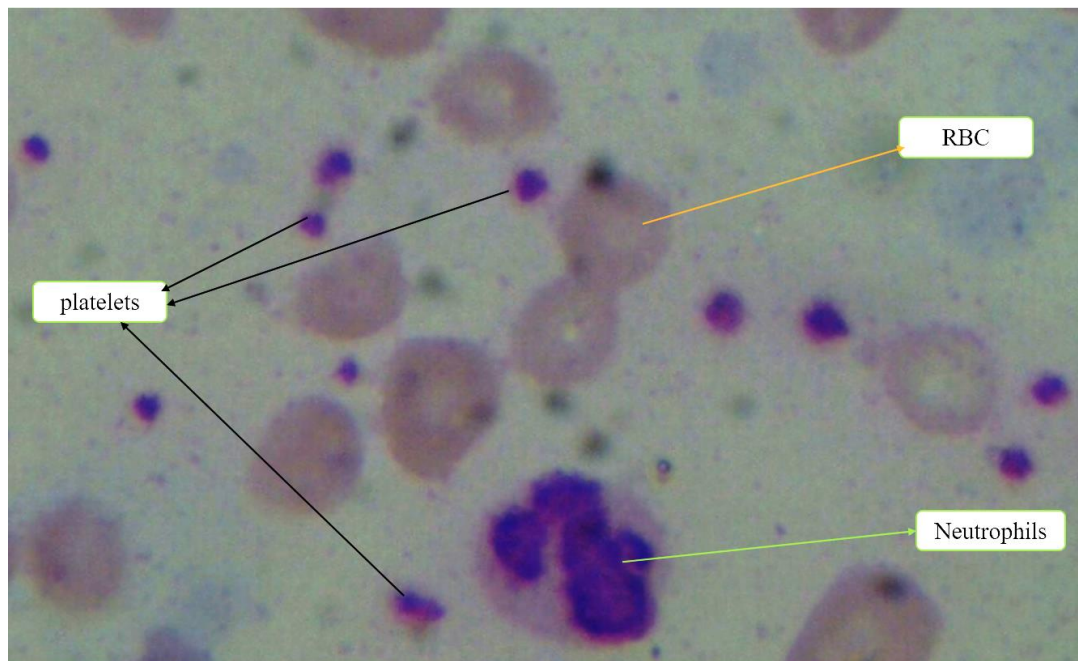


Figure 4: platelets, RBCs and Neutrophil (WBC)

1.1.2. Manual Blood Cell Counting Using Microscope

Professionals prepare the slide of hemocytometer by mixing blood with methyl blue solution of known quantity to estimate the dilution factor. Then the slide is mounted on the stage of microscope for visualization and counting of cells typically under 100x magnification [1].

Counting of white blood cells

To count all types of white blood cells, first the blood is mixed with a methyl blue solution of known quantity that completely destroys red blood cells by bursting through osmotic lysis and stains the nucleus of white blood cells only. Then the counting chamber is filled with the cells and observed under microscope determined by the following formula where NC is number concentration[1].

$$\text{NC of WBCs} = \frac{\text{number of WBCs counted} \times \text{dilution factor}}{\text{number of squares} \times \text{area of the one square}} \dots\dots (1)$$

Counting Red Blood Cells

10 microliter of blood is mixed with 2000 microliter of methyl blue solution to create a dilution factor of 1:20. Then the solution is added to the counting chamber and the slide is mounted on the stage of microscope after 20 minutes of staining. Then the concentration of red blood cells is determined by the following formula [1].

$$\text{Concentration of RBCs} = \frac{\text{number of RBCs counted} \times \text{dilution factor}}{\text{number of squares} \times \text{area of the one square}} \dots (2)$$

Counting Platelets

The method of counting platelets is almost similar with counting of red blood cells, but in counting of platelets, red blood cells are first destroyed by a solution composed of 3.8 gm of Trisodium Citrate, 0.1gm of brilliant Cresyl Blue, 40% formalin and 100ml distilled water, and the mixture is kept at room temperature so that the platelets are settled and concentration of platelets is determined by the following formula [1] [13].

$$\text{NC of platlets} = \frac{\text{number of platlets counted} \times \text{dilution factor}}{\text{number of squares} \times \text{area of the one square}} \dots (3)$$

1.1.3. Image Processing for Blood Cell Counting and Measurement of Morphological Parameters

Recently, image-processing techniques are playing greatest roles in solving the problem of manual counting techniques like improving the accuracy of detection and counting results. Currently, most procedures related to measurements are only counting of blood cells with the absence of other parameters that can enable us to narrow down the clinical decisions.

Image processing helps us to improve the quality of the image that will in turn increase the accuracy of image analysis [14]. It is very wide area with applications in robotics, medical imaging, radar, sonar, and others [15]. It plays a greatest role in segmentation and classification of objects in an image. The types as well as the sequence of image processing depends on area of application like blood cell segmentation[16].

Until now, many researches are done to count the number of each blood cells and rarely to measure the morphological parameters such as area, circularity, diameter and others using image processing. Those techniques are preceded by a sequence of pre-processing techniques depending on the property or type of cells increasing the overall performance of detection, segmentation and counting of blood cells. Pre-processing may include: background subtraction, histogram processing, Color extraction and conversion, contrast stretching, averaging, adaptive filters and others. Following those pre-processing techniques, other advanced image processing used for segmentation of cells are used which may also vary depending on the property of the cells under question.

- ✓ Otsu and/or adaptive thresholding
- ✓ Edge detection (canny)
- ✓ Snake and watershed algorithms
- ✓ Active contour
- ✓ Dilation, Erosion, Closing, Opening and others

1.1.4. Artificial Neural Network

Artificial neural network or ANN is newly emerged field of computer science that enables a computer to learn, predict and adapt like human intelligence. It is mostly

used for detection and segmentation of a variety of data types into “classes” using different methods.

A neural network is composed of three main parts, consisting of an input layer, hidden layer and output layer. Depending on the type of network, how deep we need to train and the properties of our data; the number of neurons denoted by circle in figure 5 can be varied. The circle represents a neuron which is connected with other neurons by synapses like axons in a natural nervous system. The following figure shows the typical graphical representation of a neural network.

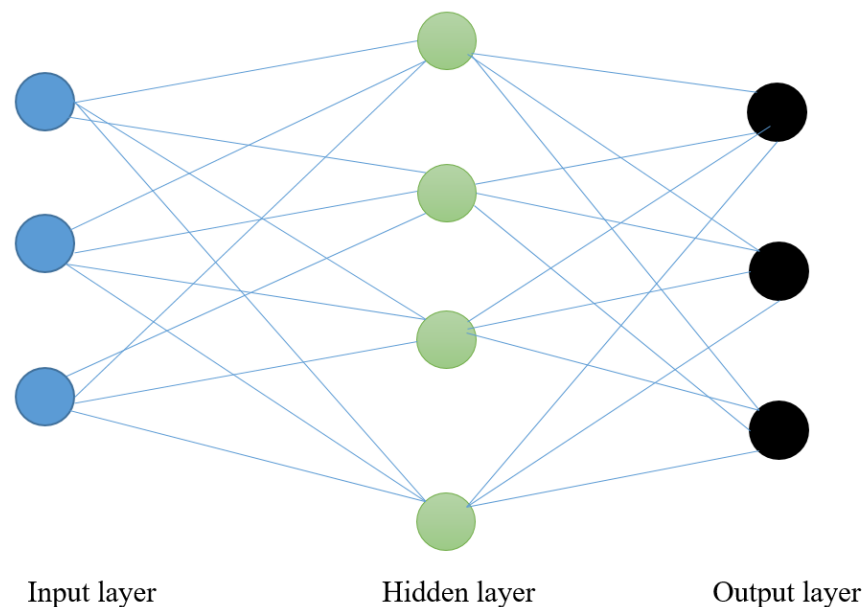


Figure 5: neural network architecture

The first input neurons accept input data and apply basic operations e.g. multiply by a random weight, the addition of a bias and activation or propagation. The neurons at the second layer take the output of the previous layer or input layer and multiply first with a weight value representing the synapse, adding a second bias term and activate the whole sum of product. After activation, the number is changed into a probability and the next neurons apply the same procedure in the forward direction until the output layer is reached, providing a final value representing the probability that the initial input belongs to a particular class. According to the error, backward operations are also done termed backpropagation to decrease the loss or error of the network.

Depending on the type of our data and application, different activation functions can be applied like sigmoid, tanh, relu, Softmax leaky and others. They have their own pros

and cons related to cost of computation, vanishing gradient (the gradient of the function becomes zero at + and – infinity) and performance time. In general the process of multiplying the output of previous neuron by a weight, adding a bias and applying an activation operation are common to all types of neural network designs.

In this research, YOLOv2, which is a special type of convolutional neural network is used to achieve the main goal of the research. Convolutional neural network or CNN is the most popular types of neural network due to its powerful property for both qualitative and quantitative analysis of scientific data.

1.2. Statement of The Problem

The quality of health service and clinical decisions are largely dependent on the available health technologies that provide relevant and more accurate results [1]. Automatic analysis of medical data like microscopic image is one of the highest factor, affecting the accuracy of medical result. Currently hemocytometers and complete blood cell count (CBC) machines are mostly used for counting blood cells in hospitals, clinics and research institutes but they are not efficient enough in provision of complete information related to the subject [8], [17].

In addition, currently available counting devices provide only the number of each classes of blood cells in a sample, without additional information about the morphological properties of cells like shape, size and others important parameters which can assist in identification of specific disease states of the patient. Sickle cell anemia and leukemia are the most serious disease, which requires additional information on morphological features of cells in addition to cells count.

Specifically sickle cell disease or anemia and leukemia affect several millions of people living all over the world with varying degrees of severity [18][19]. The case is more severe especially in developing countries due to the absence of early treatment and absence of advanced technologies that can automatically analyze blood cells and provide relevant diagnostic information [4], [20]. Generally, due to the absence of automatic analysis and quantification of morphological parameters of blood cells, clinical professionals and researchers cannot acquire the necessary information pertaining to blood cells, for the diagnosis, interpretation, ultimately yielding accurate diagnostic results [11].

1.3. Research Objectives

1.3.1. General Objective

- ✓ The main objective of this research is to detect, classify, count and measure morphological parameters of blood cells observed under a clinical microscope using YOLOv2.

1.3.2. Specific Objectives

The specific objectives of the research are to:

- ✓ Train YOLOv2 Neural Network
- ✓ Detect and segment blood cells in to seven classes
- ✓ Count all blood cells types with average accuracy of greater than or equal to 90% using the trained network.
- ✓ Provide differential count of white blood cells
- ✓ Measure size or area, diameter and aspect ratio of blood cells.
- ✓ Validate measured results of the network with manual method using ImageJ to evaluate the system performance.

1.4. Significance of The Study

The blood cell count provides very valuable clinical information about the patient. Any deviation in the number of each blood cells (increasing or decreasing) implies the presence of infection, inflammation, disorders, edema, bleeding, and other blood related issues [3]. In addition to the cell count, their morphology or anatomical structure provides important clinical information that can be used to narrow down the diagnosis and increase the accuracy of the result. Morphology or anatomical structure of cells indicates whether the cell is active or dead, affected by pathogen or not and are more useful especially to diagnose blood related cases like leukemia and sickle cell anemia.

1.5. The Scope of The Research

This research focused on counting and quantification of morphological parameters of blood cells seen under compound light microscope using YOLOv2 implemented in python. It cannot be applied for other types of cells, tissues or microorganisms; which can be seen under microscope. It is only be used to segment, count and measure the properties of seven blood cell types (red blood cells, platelets and 5 white blood cells). In addition, it cannot be used for images from other imaging modalities and microscope types like scanning electron microscope, fluorescence microscope and others, but it can be used for clinical, research and academic institutions.

CHAPTER TWO

2. Related Works in Analysis of Microscopic Images of Blood Cells

2.1. Introduction

Starting from the invention of microscope before 2000 years ago [21], researchers focused on how to improve the performance of optical microscope. But currently the improvements based on the hardware design has saturated due to the physics of light, and researcher are now turning their faces in searching new applications using computer vision, image processing, artificial intelligence and other [8], [21].

Among the new applications, segmentation, detection and counting of blood cells, especially towards the measurement of morphological parameters like surface area/size and shape which plays crucial role in clinical as well as research areas. Most of the research focuses on using morphological image processing techniques to segment and count cells. The following papers are some of researches, which are done in mostly counting of blood cells including the quantification of specific parameters using image processing and artificial neural networks (for classification and detection) as well as support vector machines.

Gautam, et al., proposed the method for extraction of nucleus of white blood cells from microscope –acquired images to provide information for doctors to quickly identify most diseases involving abnormalities, counts and activities of white blood cells [22]. They used mathematical operations like thresholding for extraction of features from segmented image. Using the size difference between white blood cells and other cells, they used simple mathematical methods to extract the nucleus of white blood cells, highlighting them for facile visualization. In addition, they used multiple thresholding to segment different cells e.g. Otsu thresholding techniques. After completing different pre-processing steps, they extracted important features of white blood cells such as eccentricity, circularity, area and perimeter. Then they use maximum and minimum values of the features from different white blood cells, and in comparison to the testing images, if the features lies between certain ranges of feature values, the label of blood cell is determined. Using 20 training images of five WBC types and by calculating the min and max values of each features, they achieved an accuracy ranging from 50 –

100% depending on the type of cell with the average accuracy of 73%. Here the accuracy varies depending on the type of WBC and few features were considered [23].

The other research done by Santi, et al., focused on how to learn to count objects using the features of the object, based on machine learning approach [24]. They used a convolutional neural network first to segment and count hand written digits then apply the concept with little modification to count pedestrians from frames acquired from surveillance camera. For learning to count things, they used one million images of size 100 pixels by 100 pixels and a four-layer architecture of CNN and 300,000 iterations to achieve an accuracy of 93.8% in segmentation and counting of handwritten digits. For pedestrian counting, they used 200,000 images with 50,000 iterations to achieve a mean absolute error of 0.74 and a mean square error of 1.12 [24].

Automatic detection and quantification of blood cells can also be done using morphological operations like watershed, Hough transform circle detection and others as indicated by the research article done by Yazan, et al [20]. The article mentioned that counting and quantification of microscope-acquired data is crucial and fundamental work in clinical as well as in research areas. But despite the importance, the techniques that are currently available including CBC machines are tedious and time consuming. The article also mentioned weakness and strengths of current methods.

Segmentation, detection and counting of WBC and RBC were done using two major steps from blood smear by Yazan, et al. Classification was done using morphological operations and counting was done using circle detection algorithm taking advantage of the shape of blood cells. The procedure were performed in two lines for the detection and counting of WBCs and RBCs with slight modifications in pre-processing based on properties of blood cell types. The performance of system were evaluated using precision, recall and f-measurement for two blood cell types and for segmentation and counting. They achieved different performance metrics for different measurements with average precision of 85% and 90% for WBC and RBC respectively. Lastly, the paper also compared the results obtained using different methods [20].

Other research done by Weidi, et al., tried to automate the counting of cells from microscopic images using fully convolutional regression networks [25]. The article used a simple ground truth that is each dot in the image represents one cell. The CNN used contains convolution ReLU pooling, which refers to max pooling or down

sampling, and results in shrinking of features map. Backpropagation and stochastic gradient descent methods were used for optimization of the network parameters. 500 random small patches of size 100pixels by 100 pixels from 500 pixels by 500 pixels input image were used where each small patch were normalized by subtracting their mean intensity and dividing by the standard deviation. In general even if the paper demonstrates the methods for counting cells using CNN, the total performance of the network/accuracy by comparing with currently applicable methods were not mentioned[26].

Othman, et al., have demonstrated the technique to classify five different types of white blood cells using a neural network [27]. The paper mentioned that due to shape differences of white blood cells, the manual method of classification is cumbersome. To avoid this, they used different features like shape, texture and intensity that contains valuable information about the spatial distribution of tonal variations. Using the above features, classification was performed using the relationship b/n the feature set and class label of input data. They used a neural network with 16 input layers representing 16 input extracted features, 5 output layers representing 5 different types of white blood cells and applied on 40 microscope-acquired images to generate 100 sub-images in which half of them were used for training and the other half used for testing. As a result they achieved 96% correct classification and 4% total false classification [27].

The recently published (2018) article by Carlos, et al., focused on segmentation and counting within an image acquired under a microscope [28]. The paper demonstrated a two-step analysis which are segmentation or identification of cells and counting using CNN. At the time of publication, there was no state of the art to use CNN for counting cells. They used a feature pyramid network to segment or detect cells from the background designed to extract feature maps and a 3x3 convolution layer with 128 filters. For counting, a Visual Geometry Group (VGG-11) network with 11 layers of a 2 dimensional convolution with filter size of 3x3 pixels was used, followed by a further convolution using a batch normalization layer. Using 600 images and the L^2 loss function, they achieved a Mean Square Error on 100 validation images that converged to a value less than 0.1 after 50 epochs. Generally more than 80% ground truth counts were found within the models predicted 95% confidence interval on the 500-test image set. The paper also mentioned the cases where the model fails counting with in a

confidence interval of the model due to high cell overlapping, irregular shapes and bad focal planes [28].

2.2. Major Gaps of Researches

There are many gaps that exist in the existing methods or works, which are listed below.

Firstly, almost all methods categories three types of blood cells which limits the amount of clinically relevant data regarding white blood cell structure and functions. In addition in image processing-based analysis, a single frame or image taken from a video is used to analyze, segment and count cells. Even if different methods are used to count overlapping cells, it is not more accurate in detection and counting of all available cells without considering additional frames, and to confound the issues, cells are often detected and counted without a tag number;- Thus, if they became hidden in subsequent frames, they will be omitted. However, if the label/tag number is given for all detected cells, and if they are overlapped by previously hidden cells, they will be considered as existing.

Secondly, almost all cell-counting methods focus only on segmentation and counting of blood cells without additional information on morphological parameters/features of cells, which are critical for clinical and research decisions; cell count alone cannot provide complete information about the subject [7]. Any abnormality in morphology can also provide additional information like in case of sickle cell anemia and leukemia.

CHAPTER THREE

3. Materials and Methodology

3.1. Research Methods

For automatic detection, classification, counting and quantification of blood cells, the research followed different methodologies at different steps starting from acquisition of a real time microscopic image for training of the network as well as for testing and evaluating the performance are discussed below.

3.1.1. Image Data Collection

The first step of this research was acquiring proper sample images (1000 images) from Jimma University Medical Center (JUMC) specifically from clinical microscope in hematology department of the hospital. In addition, to the above sources around 708 images were downloaded with known label/name from shutterstock online image database for researches which is a global leading company offering scientific data like images, videos and whose founder is Jon Oringer in USA. This increases the possibility of getting variety of images as well as to consider many parameters/features that different professionals require in both the clinical setting and research cases as well as teaching the network to perform on the images coming from different sources.

3.1.2. Pre-Processing

In order to increase the overall performance of the network, first different successive pre-processing was performed on the input images. This enables the network to extract features that can best represent the subject. Background subtraction was done on the images that were captured from the microscope in order to increase the content of visual information. Then histogram processing (histogram equalization), contrast stretching, and image averaging were done especially to reduce the presence of random noises during capturing of images. The overall preprocessing done are summarized by the figure 6 [20].

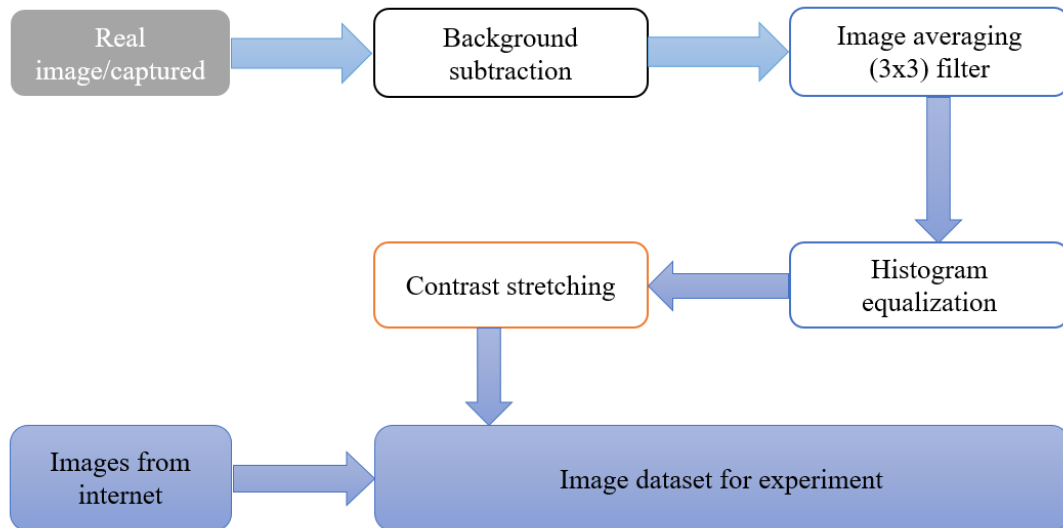


Figure 6: image pre-processing on image dataset of the experiment

3.1.3. Comparing Current Methods of Object Detectors

Before directly choose YOLOv2 comparative studies were performed on different neural networks used in real time object detection as shown by the table 1.

There are different methods to develop artificial network for real time object detection. Convolutional neural network is mostly used in almost all networks with different hyper parameters in real time object detection algorithms. Among those, RCNN, fast-RCNN, faster-RCNN, SSD and YOLO types are mostly used [29]. The following table (table 1) shows the comparison of those networks using different metrics like mean average precession, speed or frames per second, size of network and their specific features.

Networks	Principle	Speed(fps)	mAP
RCNN	Region proposal and selective search (2000) Grouping of pixels Three networks(steps) <ul style="list-style-type: none"> ✓ Feature extractor ✓ SVM (classification) ✓ Regression (optimization) Around 2000 passes per image Training three networks separately [29]–[31] Bottleneck →speed	0.0068	<70%

Fast-RCNN	ROI pooling One pass for all regions (2000 → 1) Or Jointly training feature extractor, classifier and regression networks to reduce time.	0.5	70%
Faster-RCNN	Remove bottleneck of above (region proposer using selective search) One CNN become region proposer and classification ✓ region proposal network	5-10	76.4%
Mask-RCNN	Pixel level segmentation instead of bounding boxes Checking all pixels (belongs to object or background) using binary mask (faster-RCNN) [29]. Misalignment of ROI pooling leads wrong detection (ROIAlign) ✓ Image size and feature map are not equal ✓ Fail if objects are overlapped	≤20	<86.1%
YOLOV2 (228X228)	All the above networks see the image locally. Yolo looks the image globally. Single neural network is used to predict presence of object, class probabilities and confidence using grid cells. it can't detect more than 69 objects per image.	≤150	69.0%
YOLOV2 (416X416)	The same as above but the network resolution is 416x416 and can detect maximum of 169 objects.	67	76.8%
YOLOV2 (608X608)	The same as its predecessors but no of objects that can be detected per image is 361 with stride of 32	20	88.6%

Table 1: comparison of neural networks for real time object detection

In general, all neural networks used in real time object detection have their own tradeoffs.

Speed Vs Accuracy Vs Resolution

The research used the latest version of convolutional neural network which is the *state-of-the-art* known as “YOLO” which stands for **You Only Look Once**, chosen for its properties like speed, real time detection, and boasting the highest performance in almost all metrics compared to other convolutional neural networks like CNN, RCNN, fast-RCNN and faster-RCNN. It shares some properties of other networks like using region proposal but it is unique in visualizing the images once and divide the input image into equally spaced grids with different sizes.

3.1.4. YOLOv2 Real Time Object Detection

3.1.4.1. Background of YOLOv2

The main goal of any object detection algorithms using neural networks is to recognize the presence of any object which is a member of a predefined set of classes and describe the object in terms of its location in the image defined by the top left coordinate of bounding box, size or area using the rectangles defined by width and height or bottom right corner.

YOLO (you only looks once) is currently the fastest for state-of- the-art real time object detection. It is composed of many convolutional layers for feature extraction and detection. It has also exhibits greater mAP (mean average precision) as well as faster FPS (frames per second) processing compared to other object detection algorithms [31], [32]. Due to these features plenty of researchers are involved in improving, customizing and using it for different applications. There are 3 versions of YOLO which are YOLOv1, YOLOv2 and YOLOv3. Starting from the first YOLO model, different improvements has been done to increase evaluating metrics like speed, mAP, FPS, resolution number of layers and others [33]. It can detect and classify around 9000 different classes of data. It is 1000 times faster than RCNN and 100 times faster than Faster-RCNN [29], [34]

This research used YOLOv2 model on windows machine with pre-trained weight from a coco dataset and configuration file that describes the network architecture provided by the official website of YOLO. After installation, pertinent configuration of the

network hyper parameters according to the class, data and performance needed were established.

YOLO uses convolutional layer with skip connection and up sampling layer. Stride is used to down-sample the image feature extracting low-level features that are used to segment and detect smaller objects like platelets. The network is invariant to the size of input image so different sized images from different acquisition systems were used for training, ultimately removing systemic noise from the network. The network generates feature maps from the given image determined by $S \times S(5+C)$. The size of the feature map is determined by the resolution of the network which shrunk from 416 pixels by 416 pixels to 13 grids by 13 grid cells. The number 5 in a bracket indicates the number of variables used to describe the detected object inside that grid cell which are center coordinates (X_c , and Y_c), width, height normalized using the size of the image and objectness score (P_{obj}).

Objectness score is a probability of the presence of any object contained within that grid cell. $P_{obj} = 1$ represents the presence of object whose center lies inside the grid cell and 0 represents that there is no object detected in that grid space. The grids containing the local maxima are selected and become responsible for detection of object different bounding boxes.

Class confidence C_i represents to which class that detected object belongs and it is (C_0 , C_1 , C_2 , C_3 , C_4 , C_5 , and C_6) for seven classes of objects. If the value of objectness score is zero then the values of class confidences are ignored, because empty grids containing no objects cannot be simplified. The Softmax function were used to convert class confidence in to probability because all classes are mutually exclusive (no object which can belong to two classes at the same time). The following figure (figure 7) shows the feature map for grid cells [32].

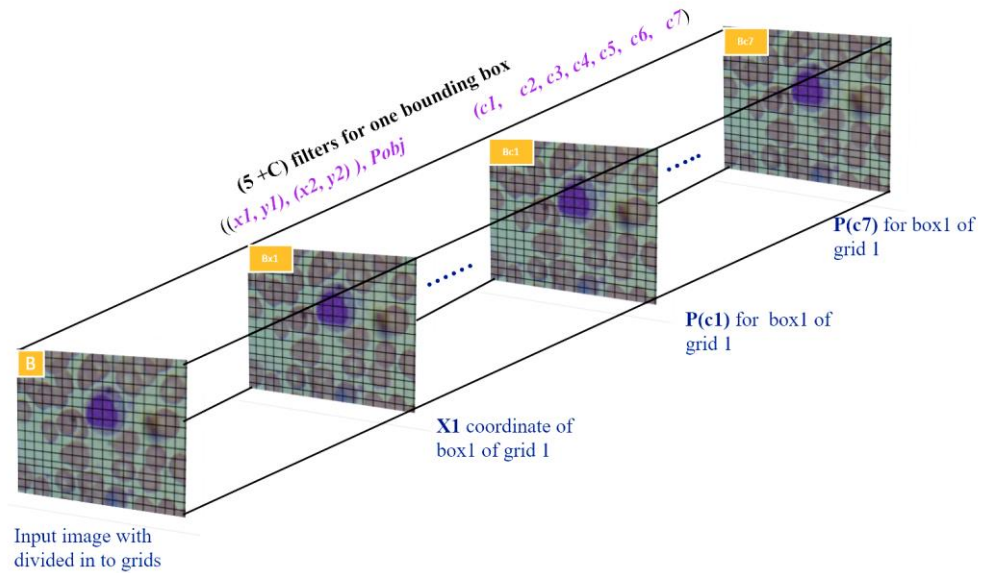


Figure 7: yolo object detection method, from image grids to feature map[35].

However, an image may contain multiple objects that can be detected and one cell may be responsible in detection of multiple classes with score difference. Under this case, YOLOv2 implements multiple bounding boxes per a grid cell responsible for detecting multiple objects which is defined by (B). Assuming a feature map of a one cell which is $(5+C)$, if multiple box is implemented, then a single grid cell will have $(5XC)XB$ number of features. Generally if one image is segmented in to $(S \times S)$ grid cells, e.g. YOLOv2 with network resolution of 418 pixels by 418 pixels, it is going to be segmented in to 13×13 cells and the output tensor of image become $13 \times 13 (5+C) \times B$. Therefore, the maximum number of objects that can be detected by the network of 418 pixels by 418 pixels resolution and stride of 32 will be $13 \times 13 = 169$ objects. But depending on the size of objects we need to detect, we can customize the network resolution so that more objects per image can be detected.

The general goal of YOLO model is to build an optimized convolutional neural network to create a $S \times S (5XB) + C$ tensor which contains all information of the detected objects, i.e. object classes, location, classification confidence and size [36]. The following figure (figure 8) shows the yolo network with resolution of 608 pixels by 608 pixels, stride of 32 and seven classes.

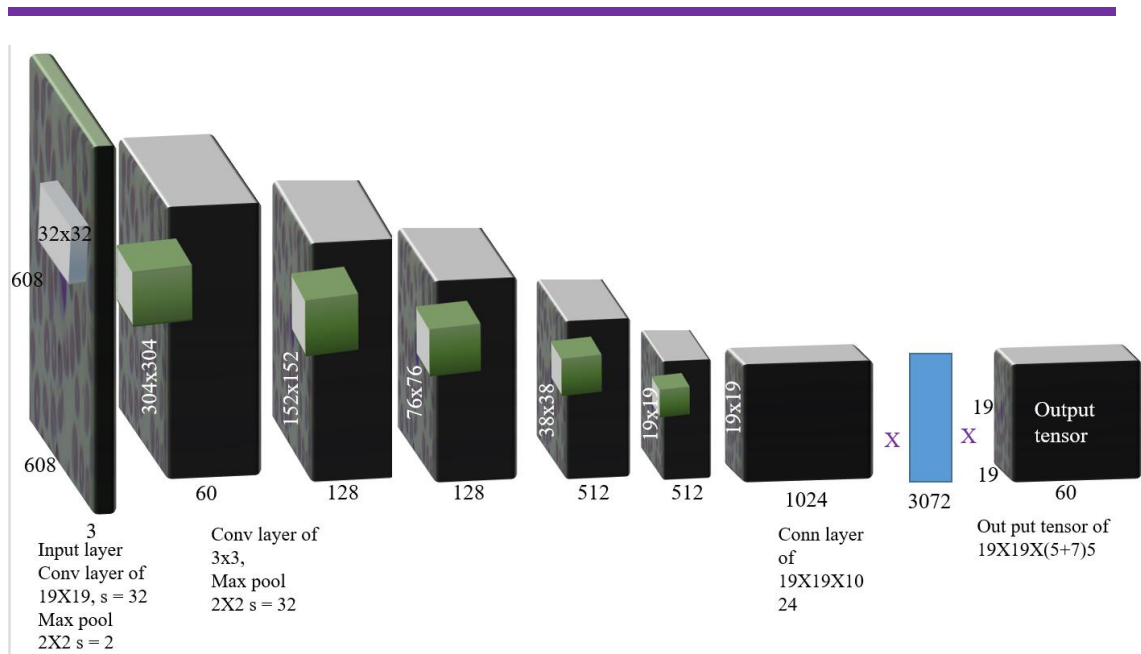


Figure 8: yolo network with stride of 32, class of 7 and resolution of 608x608

3.1.4.2. Loss Functions of The Network

YOLOv2 predicts multiple bounding boxes per grid cell of image. To calculate the loss of predictions the detection of object must be centered on the most appropriate pixels of bounding box obtained, using the principle of Intersection Over Union (IOU) so that the bounding box with high overlapping with ground truth will have highest IOU and selected as responsible bounding box than others. Generally, the higher IOU is required for better prediction as shown by the Figure 9.

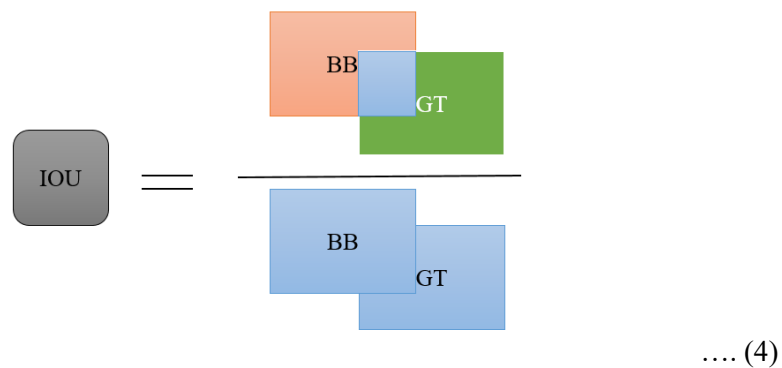


Figure 9: IOU for selecting best responsible bounding box where BB (is area covered by bounding box) and (GT) is area covered by ground truth

YOLOv2 uses the sum of the squared error between predictions and the ground truth to calculate the total loss which is composed of three loss functions.

Classification loss

If an object is detected in a grid cell, then classification loss is squared error of the class conditional probabilities for each class in a dataset, which can be described by the equation.

$$\text{Classification loss} = \sum_{i=0}^{s^2} 1_{ij}^{obj} \sum_{j=0}^B (P_i(c) - P_i^*(c))^2 \dots\dots\dots (5)$$

Where 1_{ij}^{obj} is 1 if there is detection, 0 otherwise and $p_i(C_i^*)$ denote conditional class probability of class C on the cell i.

Localization loss

Localization loss is related to the location and size of bounding box in relative to the ground truth.

Confidence loss

Confidence loss is related to objectness score or confidence of that detected object to the class it belongs which can be described by the equation 2.

$$\text{confidence loss} = \mu_{no-obj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - C_i^*)^2 \dots\dots (6)$$

Where C_i^* is box confidence score of detected object, 1_{ij}^{obj} is 1 if there is detection, other wise 0 and μ_{no-obj} is a factor to minimize the loss when there is no detection; It is 1 if the object is detected.

3.1.4.3. Non-Maximum Suppression

Non-maximum suppression or NMS were used for removing multiple detections for a single object. YOLOv2 can perform multiple detections for a single object by grid cells that are closer to the center of the ground truth. Even if the bounding boxes are derived from different locations, they may not have the same confidence or score. The bounding box coming from the most center part of ground truth have the highest IOU and confidence value than the others. Here we can suppress other detections for a single object using confidence threshold so that the boxes with confidence less than threshold

value can be suppressed. However, in some cases, decreasing the threshold may remove other predictions for other objects [37].

To remove duplications, OpenCV which is open library for computer vision applications as well as YOLOv2 provides non-max suppression employing class score, class ID and a variable overlapping threshold. First, it checks if the boxes deduce the same class ID, then it analyzes confidence difference, and lastly it assesses the degree of overlapping. If the boxes have the same class label and overlapping greater than the overlapping threshold, then the box with higher confidence is chosen and the others will be removed as shown by the Figure 9. Ordinarily, NMS cannot remove boxes that are overlapped but posses different class labels [37].

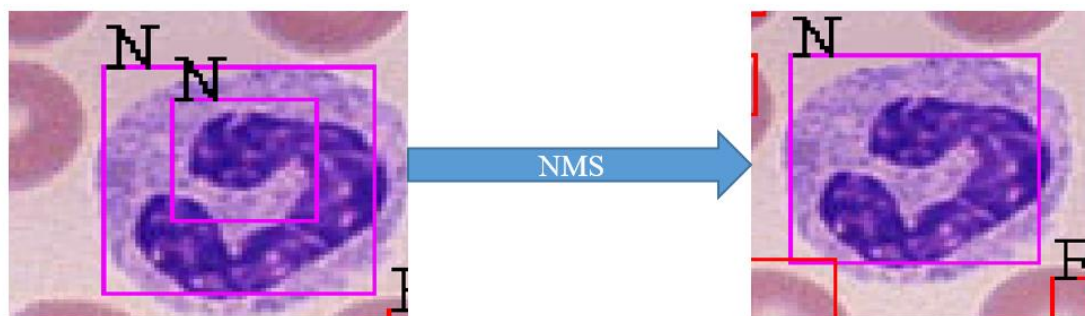


Figure 10: Non Maximum Suppression used to remove multiple detection per object

3.1.4.4. Hyper Parameters of The Network

The following hyper-parameters of the network were considered and modified during training according to the output metrics used to evaluate the performance.

Batch number

Batch number represents the number of images that can be processed in one full iteration/step. Depending on the GPU resource, it is possible to limit the batch size from 1 to any number of images. The batch size has an effect on the convergence of losses due to the number of images processed in one iteration. The typical strategy is to start from smaller batch sizes, increasing gradually until the sum of losses described above converges to acceptable values. In the case of this research the batch was gradually increased from 1 to 128.

Subdivision

Subdivision represents the number of images that can be processed at the same time from the batch selected. Once again, depending on the availability of resources, the subdivision value can range from one to the total batch number. In other words, this represents the sub-batches from batch of images selected. For example if we select batch of 64 and subdivision of 8, then from the 64 images selected to be processed in one iteration, 8 images are going to be processed at one time. So 64 images are subdivided in to eight groups and one-step process 8 sub-groups, which is equivalent to one batch size. If sub-division is increased, an “out of memory error” may occur depending on the resources available. In case of this research, subdivisions were set from 1 to 16 due to the memory limitation of graphics processor.

Epoch

The epoch of a network represents one complete dataset to be learned by the machine from the whole dataset used in the training. It is a single training cycle, which contains many steps depending on the data and configuration. After all training samples are finished or seen by the network, one epoch has elapsed, and then, the second set will be started at second epoch. In short, one epoch represents one full cycle of the network analyzing all training samples. In this study, epoch was set to 500, meaning each image was processes 500 times by the network.

Filter size

The filter size of YOLOv2 is dependent on the number of classes which is defined by $(5 + \text{number of classes}) \times 5$. The first five in bracket corresponds to the properties of bounding box and objectness score and the latter 5 represents the number of bounding boxes. In this research, the filter size was $(5+7) \times 5 = 60$ as seven classes of blood cells are utilized.

Resolution

It is the main parameter of the network, directly affecting the speed, size of small boxes and more importantly the total number of objects that can be detected per image. In this case, the resolution of the network was changed from 416 pixels by 416 pixels to 608pixels by 608 pixels as that the number of grid cells were changed from $13 \times 13 = 169$ to $19 \times 19 = 361$ with a stride of 32 pixels.

Stride

The stride of the network is the ratio of input image size and the number of grid cells. For example if stride of 32 is used then the input image of size 960x960 is segmented in to 30x30 grid cells.

Learning Rate

It represents how much weight of the network is adjusted with respect to the loss gradient. If it is small, then the speed of convergence will be small, but the loss will be more stable. If the value is larger, then the speed along the downward direction is higher but the training loss will be not stable, so electing proper learning rate according to the situation is very important. It is recommended to start by higher learning rate, then decreasing gradually to make the loss stable and converge to the ideal value. In this case, learning rate from 0.01, 0.001, 0.0001 and 0.00001 were used.

3.2. Overall Block Diagram of Methodology

The following block diagram shows the overall methods of this research

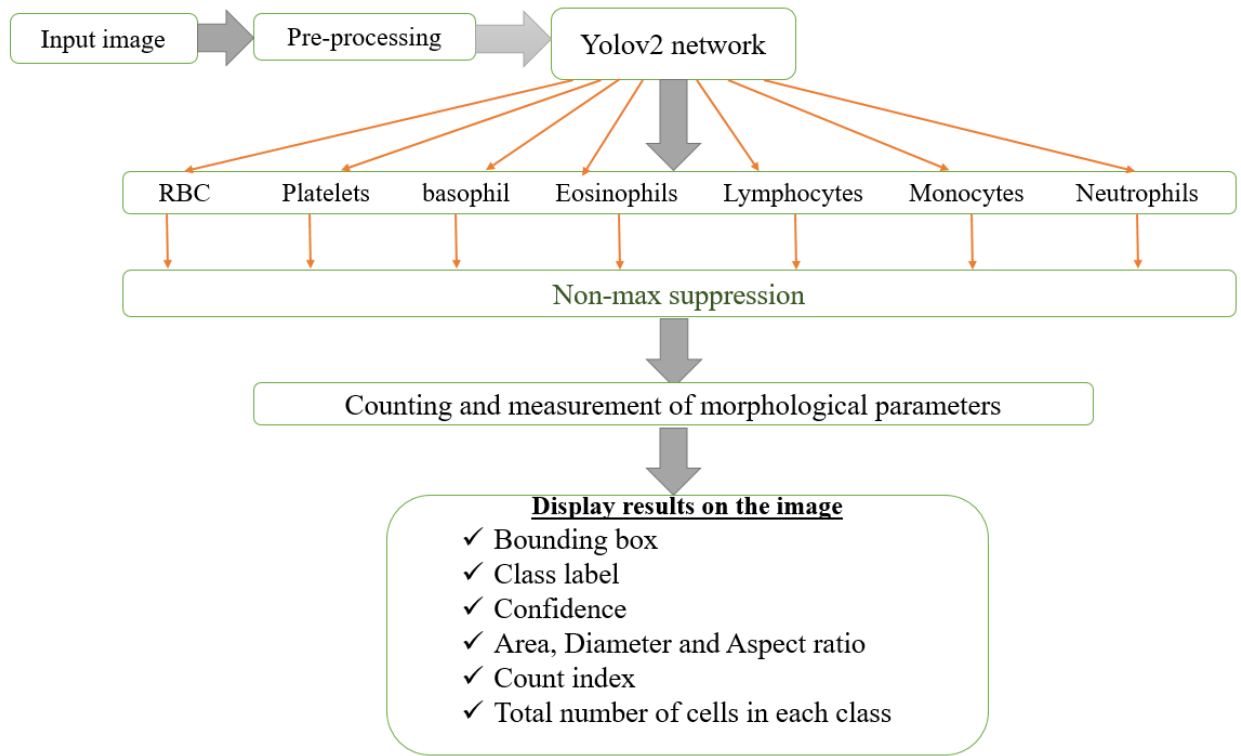


Figure 11: over all block diagram of the research

3.3. Materials

Python is a programming language suited for program developments that involve technical computing for scientific researchers. It is a recently developed platform which is popular, easy-to-use and consumes less memory to run scripts compared to traditional, bulky software such as MATLAB. It is specifically useful for image processing and machine learning due to the abundance of different packages/modules/libraries for specific applications like numpy, openCv, scipy and PyQt4. For this reason, python has become extremely useful for machine learning researchers for designing of neural networks and support vector machines. Therefore, for this thesis work, Python 3.6.5 along with different site packages or modules and along with darknet-53 for installation of yolov2 were used to design the convolution neural network specifically YOLOV2. In addition to the software, Olympus compound light microscope with 14Mp camera attached on the eyepiece and connected with computer were used to capture varieties of images from Jimma University Medical

Center. The table below summarize hardware as well as software materials used for this research.

Hardware	Software
<ol style="list-style-type: none"> 1. 14MP USB camera 2. Laptop: <ul style="list-style-type: none"> ✓ Core-i5 ✓ 12GB RAM ✓ 128GB SSD and 1TB HDD ✓ Nvidia GEFORCE 950m Graphics processor((Nvidia, Santa Clara, CA USA) 3. 32GB USB and 1TB hard disc 4. Compound light microscope 	<ol style="list-style-type: none"> 1. Python 3.6.5 with <ul style="list-style-type: none"> ✓ OpenCv ✓ Keras ✓ Tensorflow GPU 1.12 ✓ Numpy ✓ And others installed with anaconda 3 2. Visual studio 2017 community edition(build tools 2015) 3. ImageJ/Fiji (National Institutes of Health, Bethesda, MD USA)

Table 2: hardware and software materials used for the research

CHAPTER FOUR

4. Experiment, result and discussion

4.1. Experiment and Results

4.1.1. Image Collection, Preparation and Pre-processing

Firstly more than 1000 real images of blood cells were captured by a digital compound light microscope with 14Mp camera attached to the eyepiece and connected to a computer from Jimma university medical center in which 788 were selected for training. In addition, another 770 images were gathered from the internet to collect a total of 1,559 images that were classified in to 7 classes depending on the type of blood cells they mainly contain as shown by the table below. All images were used in the training of network.

Type of blood cells	No of images from the internet	No of images from the research site	Total no of images
Basophils	50	69	119
Eosinophils	144	68	212
Lymphocytes	81	95	176
Monocytes	61	66	127
Neutrophils	167	204	371
Platelets	79	82	161
Red Blood Cells	188	204	392
Total number of images	770	788	<u>1,558</u>

Table 3: Total number of images in different classes/groups and their sources

After the collection of images, image pre-processing according to the methodology was performed, followed by the annotation of images using an annotation tool written in python producing an xml file describing the object being proposed as ground truth.

4.1.2. Image Annotation.

After sorting images in to seven folders corresponding to seven classes of blood cells, image annotation was done. YOLOv2 requires the correct labels for training the algorithm, so during annotation, different number of cells present in images were proposed using rectangular form describing the location and size of object. So for each

image, multiple objects/cells were proposed and all details were saved in the form of xml file describing the location of the folder, image name, class of the object being proposed, height, width and top left corner of the object. Both image annotation and xml file created per image are shown by the figure 12.



Figure 12: image annotation tool and produced xml file

To generate additional labeled objects to serve as the ground truth, multiple objects were labeled per image depending on the frequency/availability of that object in that image. Thus a total of 2,733 labeled or indicated objects were used from 1,558 images as shown by the table 4.

Type of blood cells	Total no of images per group	Total number of labeled cells/group
Basophils	119	124
Eosinophils	212	260
Lymphocytes	176	227
Monocytes	127	151
Neutrophils	371	595
Platelets	161	560
Red Blood Cells	392	816
Total	<u>1,558</u>	<u>2,733</u>

Table 4: total number of labeled objects used for the training from different classes

4.1.3. Network Configuration

After preparing the data set, the network was configured according to the data size, class number, and nature and performance of our system. During training, the network was configured to have a stride of 32 and resolution of 416 pixels by 416 pixels so that any image were segmented or divided in to $13 \times 13 = 169$ grid cells. However, the performance was poor as the objects/cells that were selected were small. Thus the resolution was set to 608 pixels by 608 pixels so that any image would be segmented in to $19 \times 19 = 361$ grid boxes; so that the total number of objects that can be detected was greater than the maximum number of cells present in one field of view or image.

4.1.4. Training The Network

When training was started for the first time, pre-trained weight file was used instead of starting from random number. This helps to reduce training time and to converge the loss in fewer steps. Then the network was over-fitted to small number of images consisting of 8 images per class to make sure than the network can converge. Upon overfitting, the following hyper-parameters were selected for the first phase training from the first step to 7500 steps. The loss converged from 234 to 2.5 using the following hyper-parameters being selected.

- ✓ Batch = 16
- ✓ Learning rate = 0.001
- ✓ Subdivision = 8
- ✓ Epoch = 500

Then depending on the status of the training or convergence of the error, hyper parameters were changed based on their effect on the training, especially the batch number and learning rate. As the number of steps increased, the learning rate was decreased and batch number was increased to make the loss more stable and converge to the smallest point possible. Even if the loss is not a major evaluation criterion for YOLOv2 model, the smaller is the better. But the loss is mainly dependent on the number of data, the number of objects proposed and annotated in training image as well as the number of classes following the rule of thumb, the number of steps was 14,000, corresponding to 2000 times the number of classes. The first 6,125 steps, took 11.3days, corresponding to an average of 130 seconds per step. The next 8,375 steps took around 15 days with average time of 150 second per step with a learning rate

changed from 0.001 to 0.0001, 0.00001 and 0.000001 and batch changed from 16 to 32, 64 and 128 sequentially to converge the training loss to 1.5. The final 2,498 took 12 days resulting in a total training time of 38 days. In addition to changing the hyper-parameters, new images for all classes were added to increase the performance and reduce the effect of overfitting.

4.1.4.1. Performance Analysis of Training Using Error

Convergence.

The following figure (figure 13) shows the convergence of the training loss. The graph shows the loss curve from the first step to 6,121 in which the loss converged from 234.534 to 2.567 which is the first phase training. At this point, the hyper parameters were changed and new data were added so that the loss converged to 1.5 and after step number of 16,998 the loss shows no change except slight oscillations, which is an indication of convergence.

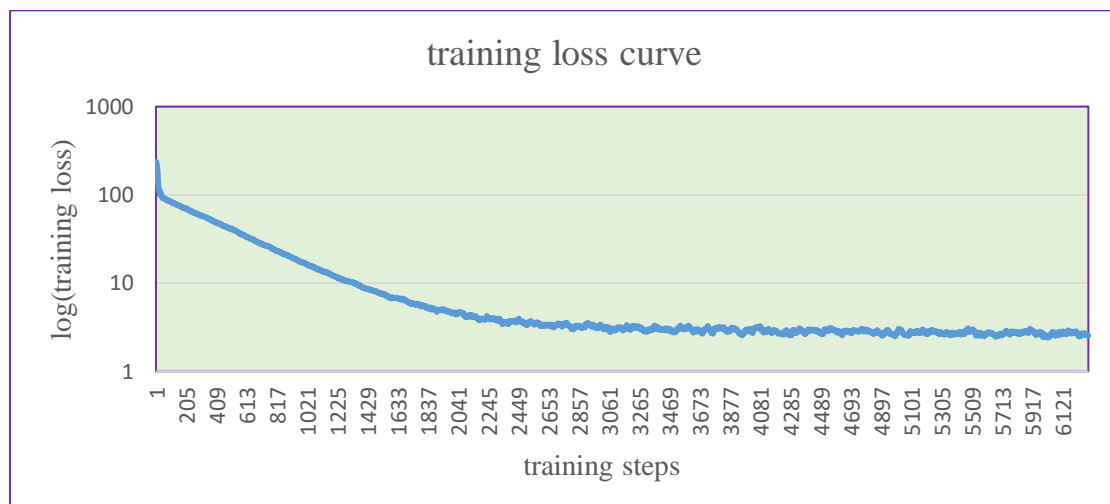


Figure 13: error convergence of training

4.1.5. Detection, Segmentation, Counting and Quantification of Morphological Parameters

4.1.5.1. Detection and Classification of Blood Cells

The network was evaluated at different steps using checkpoint files. Hence, the network provided the updated weight file at each 125 steps, testing on a-new images was done sequentially and it started detection at a step number of 2,125 and increased

performance as the step number increased. Qualitatively, the following three images shows the output of the network at three different steps in detecting, classification and counting of blood cells.

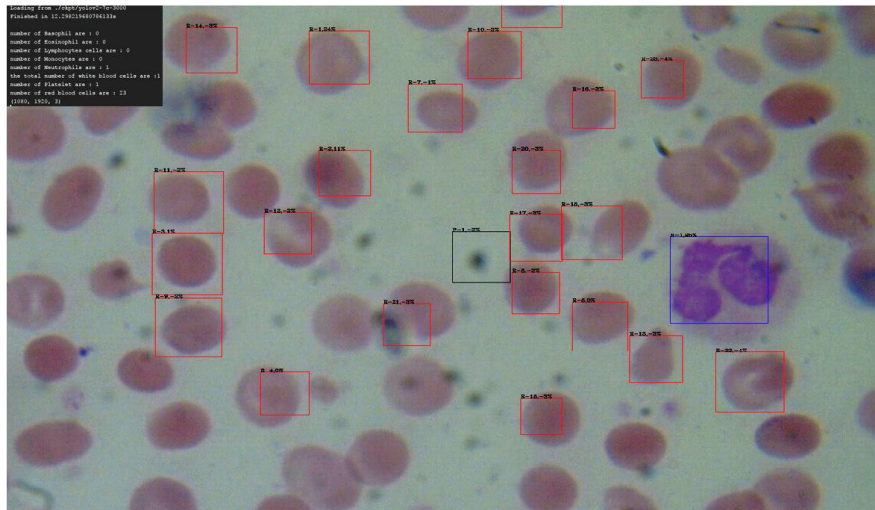


Figure 14: detections and counting of a network at step number of 3,000

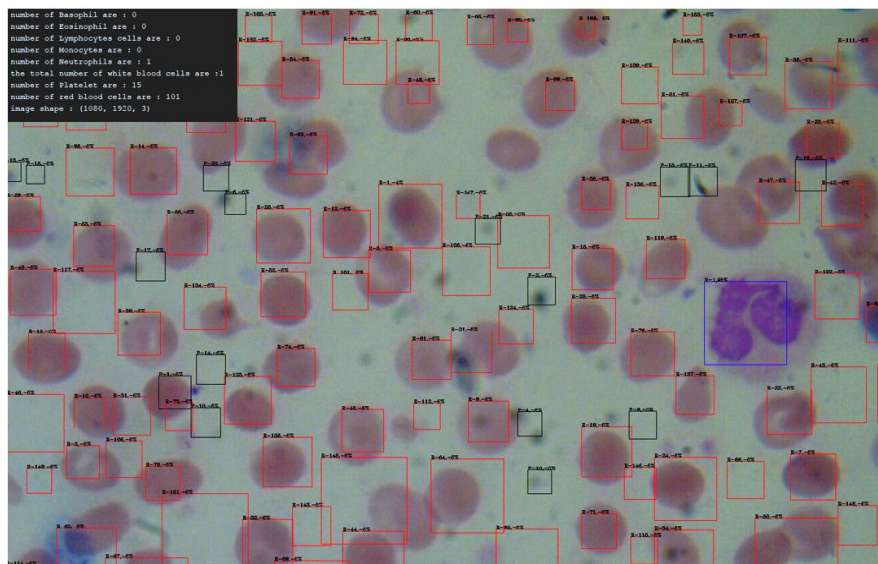


Figure 15: detection, segmentation and counting of network at step number of 14,500

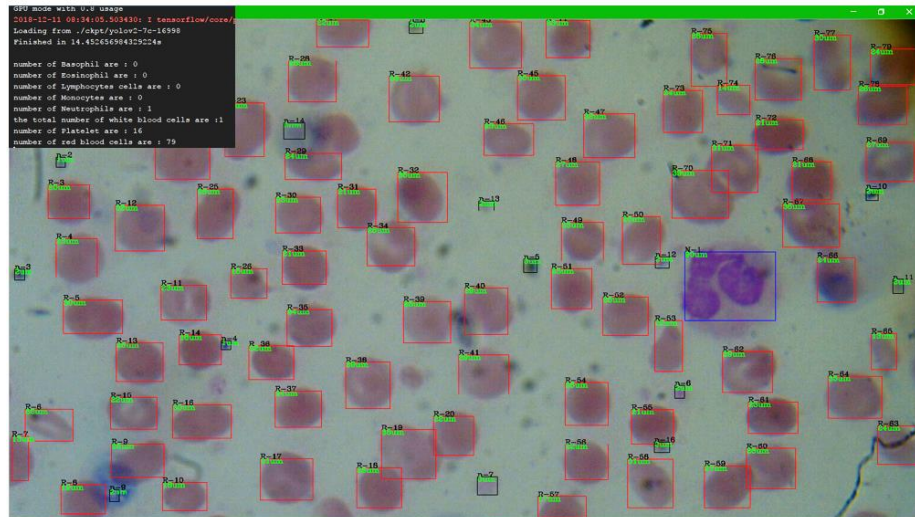


Figure 16: output of the network at step number of 16, 998

After the network has been trained for detection, segmentation and counting of blood cells, performance evaluation was assessed using the confusion matrix displayed in table 5. As true label, the total number of labeled cells in 26 test images were used and as a prediction label, the total number of cells in each class detected by the network were used.

		Prediction								
		<i>RBC</i>	<i>P</i>	<i>B</i>	<i>E</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>undetected</i>	<i>FP from noise</i>
Ground truth	<i>RBC</i> (1,454)	1,304	17	0	0	0	0	0	133	91
	<i>P</i> (159)	14	129	0	0	0	0	0	11	16
	<i>B</i> (3)	0	0	3	0	0	0	0	0	0
	<i>E</i> (12)	0	0	1	10	0	0	1	0	0
	<i>L</i> (24)	0	0	0	0	21	1	1	1	0
	<i>M</i> (13)	0	0	0	0	1	12	0	0	0
	<i>N</i> (28)	0	0	0	1	0	0	27	0	0

Table 5: confusion matrix showing the performance of the network output

So using the above matrix, true positives, false positives and false negatives are determined as follows. Here true positive detection indicates the output of the network, which correctly detect and segment correctly to the class label. False positive results

are the sum of detection of cells belongs to other class and detection of background noise as object/cell of specific class. From the above matrix, there were 1454 red blood cells from 26 test images, among them 1,304 blood cells were accurately detected and classified as red blood cells. Out of 1,454 red blood cells, 17 cells were detected but classified as platelets, which were false negative for classification of red blood cells and false positives for platelet classification. The remaining 133 red blood cells were not detected by the network accounting for the bulk of false positives for red blood cells class. In addition to 1,304 true red blood cells, the network has also detected 91-background noise objects as red blood cells accounting for the other source of false positive for this specific class.

Using the above method, accuracy, recall and precision of the network were calculated as follows.

Blood cells(classes)	True positive	False negative	False positives
RBC (1,454)	1,304	150	105
Platelet (159)	129	25	33
Basophils (3)	3	0	1
Eosinophils (12)	10	2	1
Lymphocytes(24)	21	3	1
Monocytes (13)	12	1	1
Neutrophils (28)	27	1	2

Table 6: overall network performance

Where:

TP is the number of cells correctly identified in their class

FP is the number of cells detected but not belonging to that specific class or noise

FN is the number of cells, which are not detected or have been detected but classified under a different class.

Accuracy

Accuracy describes the closeness of the number of detected cells to the actual number of cells in a given image and calculated as follows for all classes of blood cells to provide average an overall accuracy of 80.6%

$$accuracy = \frac{\text{true positives}}{\text{true positive}+\text{false positive}+\text{false negative}} \dots (7)$$

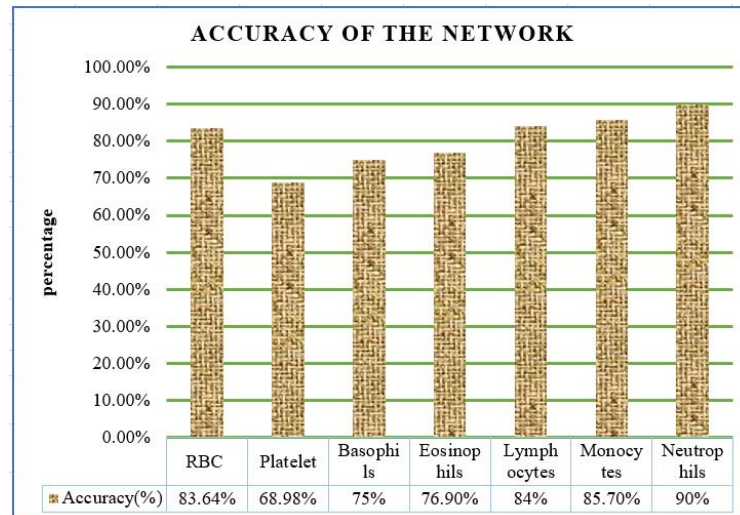


Figure 17: accuracy performance of the system

Precision

Precision or positive predictive value describes the proportion of positive or correct classification of blood cells (true positives) from all cases as predicted positive as follows. Precision of the network for all class of blood cells was calculated and the average precision was found to be 88.4%.

$$precision = \frac{\text{true positives}}{\text{true positive} + \text{false positive}} \quad \dots (8)$$

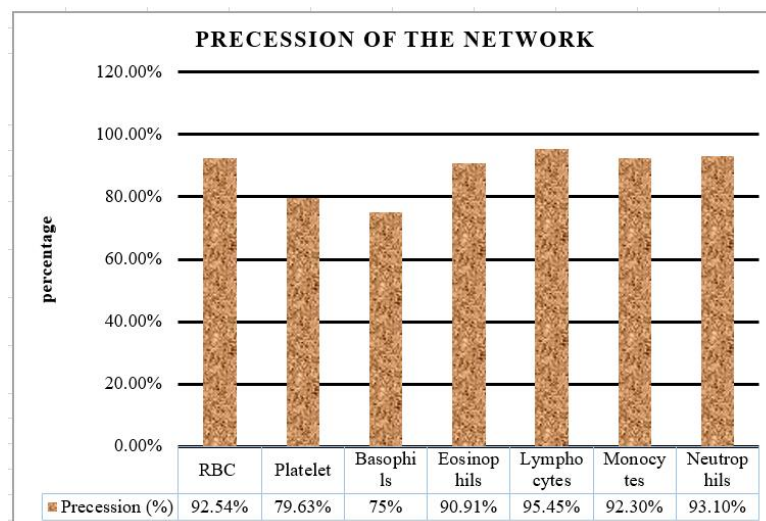


Figure 18: precision performance

Recall

Recall also known as sensitivity describes the fraction of true detection over the total number of cells present in images. Recall for all cell classes was calculated and the average recall of the network was 89.6%.

$$recall = \frac{\text{true positives}}{\text{true positive} + \text{false negatives}} \dots (9)$$

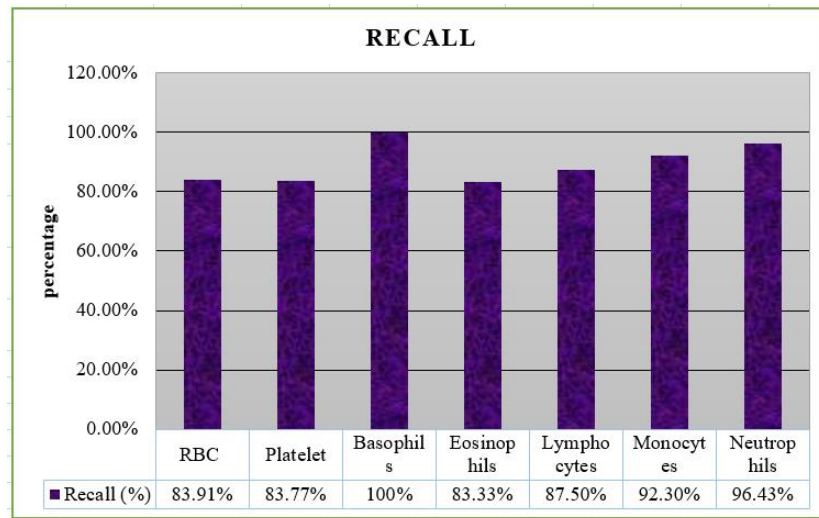


Figure 19: recall of the network

4.1.5.2. Counting of Blood Cells

After proper detection segmentation and labeling of all cells in-to seven classes, counting was done automatically using the index of each cell's bounding box and class label. Counting was performed on 26 images containing multiple blood cells of multiple classes shown by the table 9 (at appendix). The number of cells counted by the system was slightly higher than the true positives output due to the presence of false positives detected either from background or cells from other class. The output of network for 26 images containing multiple objects is shown by the table 10, in which the number is the sum of true positive and false positives of each class. Improving the network further can reduce false positives and false negatives and increase true positives. The following table shows the performance of the network in counting of blood cells with a mean counting accuracy of 92.702% and mean error of 7.298%.

	Number of cells in each class						
	RBC	P	B	E	L	M	N
The real number of cells in all 26 images	1,454	159	3	12	24	13	28
The total number of cells detected and counted (TP+FP)	1,409	162	4	11	22	13	29
Error (%)	3.1%	2.0%	25%	9.09%	8.33%	0%	3.57%

Table 7: performance of the network in counting classes of blood cells for 26 images:

Considering the tables 09 and 10, the accuracy of the network was calculated by using the total number of cells present in all images (true value) and the total number of cells detected by the network as shown by the table 7.

4.1.5.3. Quantification of Morphological Parameters

Measurement of morphological parameters that can best describe the status of cells were done using detected cell information including location, height, and width. First, scale bars were obtained and calculated by imaging a microscope stage micrometer slide. The calibration/scaling displayed in table 8 was obtained by facile analysis using ImageJ enables us to convert pixel dimension into micrometers. The calibration and image of stage micrometer are shown by the table 8 and figure 20 respectively.

Magnification power including eyepiece	Scale(pixels per micrometer)
40X (4X objective)	0.628
100X (10X objective)	1.57
400X (40X objective)	6.28
1000X (100X objective)	15.7

Table 8: image calibration/scaling

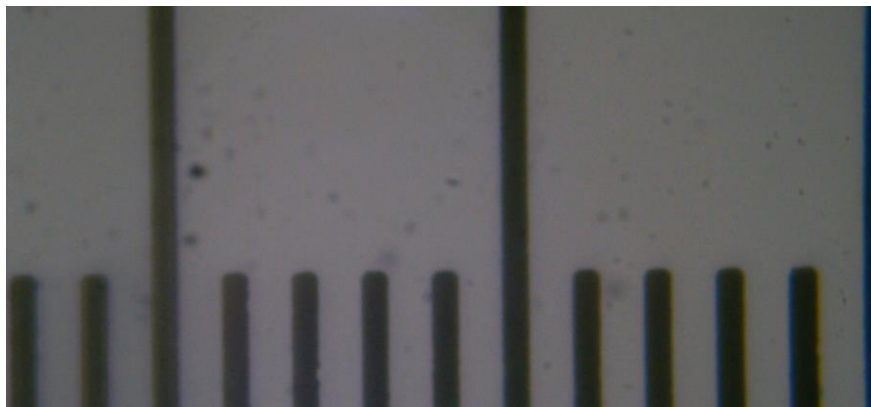


Figure 20: image of stage micrometer seen under microscope of 100 magnification

After image calibration, size or area, average diameter and aspect ratio were measured for each detected object directly from the bounding box information [38] [39].

Average Diameter (D)

Average diameter was calculated using average of width and height

Area (A)

Area or size of the cell was calculated using scaled height (H) and scaled width (W). but all blood cells show circularity, so, area was measured using average diameter or average radius of the detected cell as follows [40].

$$\text{Average radius (Ar)} = (H + W)/4$$

$$\text{Area} = \pi(\text{Ar})^2$$

Aspect ratio(AR)

Aspect ratio/circularity was calculated by first comparing the height and width of the box and if height is less or equal to width then circularity can be calculated as follows [40].

$$\text{AR} = H/W \quad \text{else: } \text{AR} = W/H$$

The area shows the size of the cell under consideration. Since all cells shows circularity property, we can use diameter and circularity to investigate full status of the cells. Those measurements were compared with available tools like ImageJ and shows similar measurement value as shown by the following three tables (table 11, 12 and 13) where B is Basophil, E is Eosinophils, L is Lymphocyte, M is Monocyte and N is Neutrophil. In addition, all measurements were done on images with scale bar and by making measurements using manual method (ImageJ) as true value

Area measurement

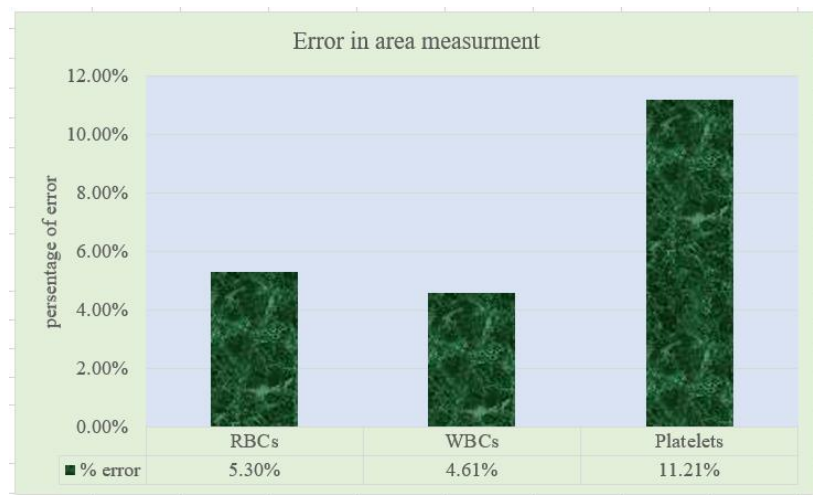


Figure 21: performance of system in measurement of area

Aspect ratio measurement

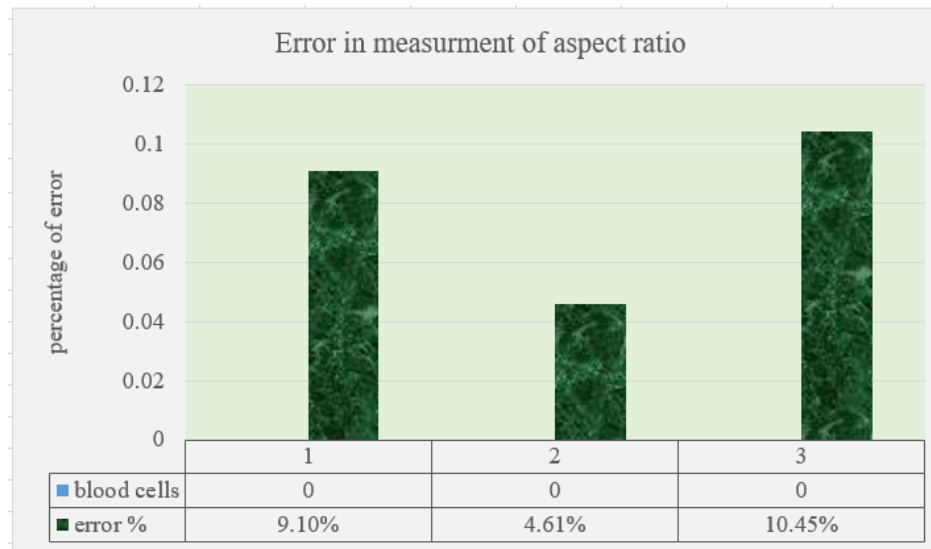


Figure 22: The performance comparison of the system in measuring aspect ratio of blood cells

Diameter measurement

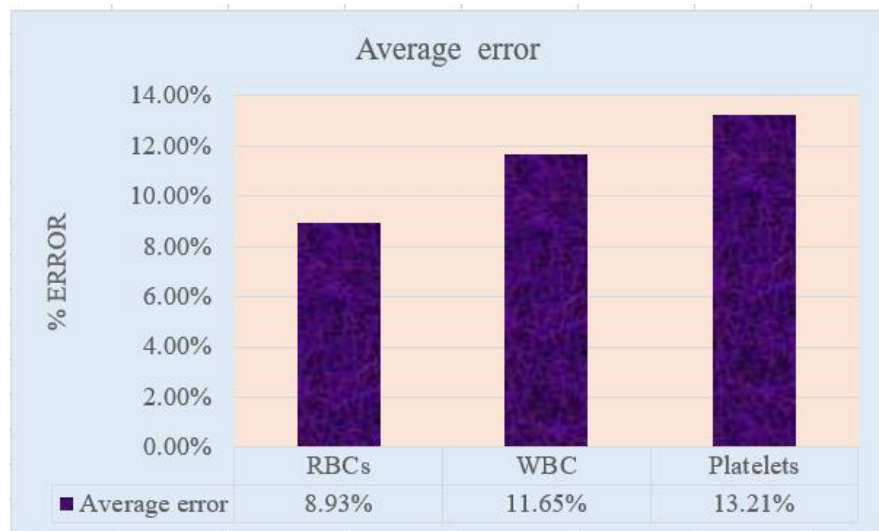


Figure 23: The performance comparison of the system in measuring average diameter of blood cells in μm

From the above figures, it is clearly seen that the measurements done using the two methods on three different categories shows average accuracy of 92.96%, 91.95% and 88.74% in measurement of area, aspect ratio and average diameter of cells respectively.

CHAPTER FIVE

5. Conclusion and Recommendation

5.1. Conclusion

The current available methods used for blood cell segmentation and counting are very tedious and are highly prone to different sources of errors. In addition, the existing techniques do not provide full information related to blood cells e.g. their shape and size, which are of increasing importance in clinical investigation of serious blood-related diseases like leukemia and sickle cell disease.

This research employed a YOLOv2 convolutional neural network, which is currently the state-of-the-art in real-time object detection and classification performing faster and more accurate than other object detection networks including faster R-CNN and its predecessors.

The model was trained on more than 1558 images containing 2733-labeled blood cells with network resolution of 608x608, processed using various batch sizes and learning rates over 16,998 steps. It was tested on 26 test images containing 1454 red blood cells, 159 platelets, 3 basophils, 10 eosinophils, 24 lymphocytes, 13 monocytes and 28 neutrophils. The network performs detection and segmentation of blood cells with an average accuracy of 80.6% and precision of 88.4%.

After classification and counting, measurement of area, diameter and aspect ratio of cells were computed and the results were evaluated using manual method (ImageJ). It achieved a mean accuracy of 92.956%, 91.96%, 88.736% and 92.7% in measurement of area, aspect ratio, diameter and counting of cells respectively.

5.2. Recommendation

As a researcher I recommend different researchers at different level of education to see the opportunity of recent methods in real time object detection like yolov2 and others. They can be applied in different areas of research, which focuses on object detection and classification.

I also recommend researchers interested in analysis of microscopic image of blood cells to focus on measurement of more morphological parameters of blood cells like roundness, volume, curvature, elongation, eccentricity, convexity, sphericity, aspect ratio and others which can improve the diagnostic of blood related diseases. In addition, it is more valuable to detect blood cells with detector of the same size like blood cell instead of using rectangular detector so that analysis of the detector can be correlated to the subject cell. It is also possible to use the features of detected blood cells to automatically predict different blood related diseases using second artificial neural network.

Lastly I would like also to recommend Ethiopian ministry of health and all concerned authorities to focus on how to implement different researches and innovations done from academic and research institutes on our country so that the quality of health service can be improved.

List of References

- [1] A. Khan, “An Accurate and Cost Effective Approach to Blood Cell Count An Accurate and Cost Effective Approach to Blood Cell Count,” no. January, pp. 17–24, 2017.
- [2] S. Khan, A. Khan, F. Saleh Khattak, and A. Naseem, “An Accurate and Cost Effective Approach to Blood Cell Count,” *Int. J. Comput. Appl.*, vol. 50, no. 1, pp. 18–24, Jul. 2012.
- [3] F. Yi, I. Moon, and B. Javidi, “Automated red blood cells extraction from holographic images using fully convolutional neural networks,” *Biomed. Opt. Express*, vol. 8, no. 10, p. 4466, Oct. 2017.
- [4] M. Thejashwini and M. C. Padma, “Counting of RBC’s and WBC’s Using Image Processing Technique,” *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 5, pp. 2948–2953, 2015.
- [5] C. Automated and C. Counter, “Comparison of image-based cell counting methods ;,” no. C, pp. 1–4.
- [6] “Red Blood Cells: Function and Structure.” [Online]. Available: <https://www.thoughtco.com/red-blood-cells-373487>. [Accessed: 31-May-2018].
- [7] S. Azwai and O. Abdousslam, “Morphological characteristics of blood cells in clinically normal adult llamas (Lama glama),” *Vet. ...*, vol. 77, no. 1, pp. 69–79, 2007.
- [8] U. P. Moravapalle, A. Deshpande, A. Kapoor, R. Ramjee, and P. Ravi, “Blood Count on a Smartphone Microscope,” *Work. Mob. Comput. Syst. Appl.*, pp. 37–42, 2017.
- [9] A. I. Shahin, Y. Guo, K. M. Amin, and A. A. Sharawi, “White blood cells identification system based on convolutional deep neural learning networks,” *Comput. Methods Programs Biomed.*, Nov. 2017.
- [10] S. Paxton, M. Peckham, K. Adele, S. Paxton, K. Adele, and M. Peckham, “The Leeds Histology Guide.” 2003.
- [11] N. Ramesh, B. Dangott, M. E. Salama, and T. Tasdizen, “Isolation and two-step classification of normal white blood cells in peripheral blood smears,” *J. Pathol. Inform.*, vol. 3, no. 1, p. 13, 2012.
- [12] “Components of blood (article) | Khan Academy.” [Online]. Available: <https://www.khanacademy.org/science/biology/human-biology/circulatory-pulmonary/a/components-of-the-blood>. [Accessed: 31-May-2018].
- [13] S. BATRA, “Total Platelet Count Using Hemocytometer | Hematology Practicals.” [Online]. Available: <https://paramedicsworld.com/hematology->

practicals/total-platelet-count-using-hemocytometer-neubauers-chamber-micro-dilution-macro-dilution-method/medical-paramedical-studynotes#.XCz952hKhEY. [Accessed: 02-Jan-2019].

- [14] Q. Wang, L. Chang, M. Zhou, Q. Li, H. Liu, and F. Guo, “A spectral and morphologic method for white blood cell classification,” *Opt. Laser Technol.*, vol. 84, pp. 144–148, Oct. 2016.
- [15] M. Rahnemoonfar and C. Sheppard, “Deep count: Fruit counting based on deep simulated learning,” *Sensors (Switzerland)*, vol. 17, no. 4, pp. 1–12, 2017.
- [16] H. Tulsani, S. Saxena, and N. Yadav, “Segmentation using Morphological Watershed Transformation for Counting Blood Cells,” *Int. J. Comput. Appl. Inf. Technol.*, vol. 2, pp. 2278–7720.
- [17] K. Adi, S. Pujiyanto, R. Gernowo, A. Pamungkas, and A. B. Putranto, “Identifying the Developmental Phase of Plasmodium Falciparum in Malaria-Infected Red Blood Cells Using Adaptive Color Segmentation And Back Propagation Neural Network,” *Int. J. Appl. Eng. Res. ISSN*, 2016.
- [18] centers for disease control and prevention, “Data and Statistics | Sickle Cell Disease | NCBDDD | CDC,” *Sickle Cell Disease Homepage*. 2016.
- [19] T. Palmerini *et al.*, “Impact of leukocyte count on mortality and bleeding in patients with myocardial infarction undergoing primary percutaneous coronary interventions: Analysis from the harmonizing outcome with revascularization and stent in acute myocardial infarction trial,” *Circulation*, vol. 123, no. 24, pp. 2829–2837, 2011.
- [20] Y. M. Alomari, S. N. H. Sheikh Abdullah, R. Zaharatul Azma, and K. Omar, “Automatic detection and quantification of WBCs and RBCs using iterative structured circle detection algorithm,” *Comput. Math. Methods Med.*, vol. 2014, p. 979302, 2014.
- [21] X. Chen, B. Zheng, and H. Liu, “Optical and digital microscopic imaging techniques and applications in pathology,” *Anal. Cell. Pathol.*, vol. 34, no. 1–2, pp. 5–18, 2011.
- [22] A. Gautam and H. Bhadauria, “Classification of white blood cells based on morphological features,” in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 2363–2368.
- [23] A. Gautam and H. Bhadauria, “Classification of white blood cells based on morphological features,” *2014 Int. Conf. Adv. Comput. Commun. Informatics*, no. May, pp. 2363–2368, 2014.
- [24] S. Segui, O. Pujol, and J. Vitria, “Learning to count with deep object features,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2015–Octob, pp. 90–96, 2015.
- [25] W. Xie, J. A. Noble, and A. Zisserman, “Microscopy cell counting and detection with fully convolutional regression networks,” *Comput. Methods*

- Biomech. Biomed. Eng. Imaging Vis.*, vol. 6, no. 3, pp. 283–292, May 2018.
- [26] W. Xie, J. A. Noble, and A. Zisserman, “Microscopy Cell Counting with Fully Convolutional Regression Networks.”
- [27] M. Z. Othman, T. S. Mohammed, and A. B. Ali, “Neural Network Classification of White Blood Cell using Microscopic Images,” *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 5, 2017.
- [28] C. X. Hernández, M. M. Sultan, and V. S. Pande, “Using Deep Learning for Segmentation and Counting within Microscopy Data,” 2018.
- [29] D. Parthasarathy, “A Brief History of CNNs in Image Segmentation : From R-CNN to Mask R-CNN,” *Medium*, 2017. [Online]. Available: <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4>. [Accessed: 05-Dec-2018].
- [30] “Review of Deep Learning Algorithms for Object Detection.” [Online]. Available: <https://medium.com/comet-app/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>. [Accessed: 05-Dec-2018].
- [31] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms.” [Online]. Available: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. [Accessed: 05-Dec-2018].
- [32] J. Hui, “Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3,” *Medium*, 2018. [Online]. Available: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088.
- [33] J. Redmon and A. Farhadi, “YOLO: Real-Time Object Detection,” 2018. [Online]. Available: <https://pjreddie.com/darknet/yolo/>. [Accessed: 05-Dec-2018].
- [34] J. Hui, “Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and....)” [Online]. Available: https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359. [Accessed: 11-Dec-2018].
- [35] R. Adrian, “(Faster) Non-Maximum Suppression in Python,” *Machine Learning Tutorials*, 2015. [Online]. Available: <https://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/>.
- [36] A. M. Herrera-Navarro, H. Jiménez Hernández, H. Peregrina-Barreto, F. Manríquez-Guerrero, and I. R. Terol-Villalobos, “A New Measure of Circularity Based on Distribution of the Radius,” vol. 17, no. 4, pp. 515–526, 2013.
- [37] S. Khan, A. Khan, F. Saleh Khattak, and A. Naseem, “An Accurate and Cost

Effective Approach to Blood Cell Count,” *Int. J. Comput. Appl.*, vol. 50, no. 1, pp. 18–24, Jul. 2012.

- [38] A. M. Herrera-Navarro, H. Jiménez Hernández, H. Peregrina-Barreto, F. Manríquez-Guerrero, and I. R. Terol-Villalobos, “A New Measure of Circularity Based on Distribution of the Radius,” vol. 17, no. 4, pp. 515–526, 2013.

APPENDICES

Appendix A: Sample Training Data

The following two images shows sample data (xml files and images) used for training of the network.

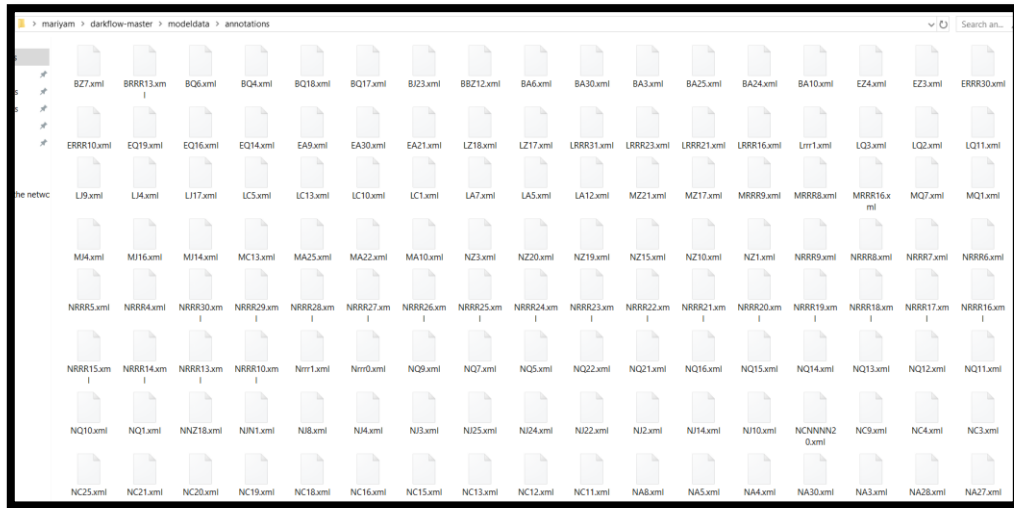


Figure 24: partial view of xml files containing annotations



Figure 25: sample image used for training (training dataset)

Appendix B: Real Number of Cells in 26 Test Images and Detected Cells

Number of cells in 26 test images are shown by the following table.

Input images	Number of cells in each cell class						
	RBC	P	B	E	L	M	N
T1	63	1	1	1	1	1	2
T2	45	3	0	0	1	0	0
T3	64	5	0	0	5	0	0
T4	54	2	0	1	0	0	0
T5	57	5	0	0	1	0	0
T6	87	3	0	0	4	4	0
T7	17	1	1	0	0	0	0
T8	110	15	0	1	0	0	12
T9	83	4	0	7	0	0	0
T10	20	0	0	0	2	0	0
T11	64	3	0	0	1	1	0
T12	62	8	0	0	1	4	3
T13	54	4	0	1	0	1	4
T14	81	18	0	0	0	0	1
T15	44	5	0	0	1	0	1
T16	73	2	0	0	0	1	1
T17	31	3	1	0	0	0	0
T18	29	5	0	1	0	0	0
T19	27	1	0	0	1	0	0
T20	28	2	0	0	1	0	0
T21	34	3	0	0	0	1	0
T22	21	4	0	0	0	0	1
T23	67	6	0	0	5	0	0
T24	33	22	0	0	0	0	1
T25	116	21	0	0	0	0	1
T26	90	15	0	0	0	0	1

Total	1,454	159	3	12	24	13	28
-------	-------	-----	---	----	----	----	----

Table 9: list of images containing different blood cells used for testing.

The following table shows the number of cells detected truly (only true positives) for all test images.

Input images	Number of cells in each class detected by the network considering only true positives						
	RBC	P	B	E	L	M	N
T1	61	1	1	1	1	1	2
T2	43	2	0	0	1	0	0
T3	62	3	0	0	4	0	0
T4	49	2	0	1	0	0	0
T5	52	5	0	0	1	0	0
T6	76	1	0	0	3	4	0
T7	15	1	1	0	0	0	0
T8	89	12	0	1	0	0	11
T9	79	4	0	6	0	0	0
T10	18	0	0	0	2	0	0
T11	45	3	0	0	1	1	0
T12	60	6	0	0	1	4	3
T13	49	4	0	1	0	1	4
T14	79	15	0	0	0	0	1
T15	42	1	0	0	1	0	1
T16	69	1	0	0	0	1	1
T17	25	2	1	0	0	0	0
T18	23	2	0	1	0	0	0
T19	25	1	0	0	1	0	0
T20	24	2	0	0	1	0	0
T21	29	1	0	0	0	1	0
T22	18	2	0	0	0	0	1
T23	61	4	0	0	4	0	0
T24	28	20	0	0	0	0	1
T25	103	19	0	0	0	0	1
T26	84	15	0	0	0	0	1

Total	1,308	129	3	11	21	13	27
-------	-------	-----	---	----	----	----	----

Table 10: Total number of cells in each class detected by the network per image

Appendix C: Source Code

The following is part of the python script written for detection, segmentation, counting and measurement of blood cells.

```
# ##### done by Abel Worku From Jimma University #####
##### School of Biomedical Engineering #####
# let's first import, numpy, installed darknet and opencv
import numpy as np
import tensorflow as tf
from darkflow.net.build import TFNet
import cv2
import math
# creating list of boxes, labels and scores
boxes = []
labels = []
scores = []
conf_threshold = 0.1
nms_threshold = 0.03
#####COUNTING PARAMETERS#####
NR = 0
NB = 0
NP = 0
NN = 0
NM = 0
NE = 0
NL = 0
#####
# creating dictionary of options that define the network
options = {
    'model': 'cfg/yolov2-7c.cfg',
    'load': 15125,
    'threshold': 0.1,
    'gpu' : 0.8
}
# creating object tfnet
mymodel = TFNet(options)

##### input image options #####

img = cv2.imread('testingimages/t1.JPEG')
path = 'C:/Users/Abel/Desktop/mariyam/darkflow-master/output of the network'
#####
# pre-processing to the image under investigation
# 1. APPLY 3X3 SMOOTHENNG FILTER TO SUPPRESS RANDOM NOISE
kernel = np.ones((3, 3), np.float32)/9
img = cv2.filter2D(img, -1, kernel)
# applying histogram equalization and brightness by extracting color channels
```

```

ht = img.shape[0]
wd = img.shape[1]
br = 25
R, G, B = cv2.split(img)
PR = cv2.equalizeHist(R)
for i in np.arange(ht):
    for j in np.arange(wd):
        original = PR.item(i, j)
        processed = original + br
        if processed > 255:
            processed = 255
        PR.itemset((i, j), processed)
PG = cv2.equalizeHist(G)
for i in np.arange(ht):
    for j in np.arange(wd):
        original = PG.item(i, j)
        processed = original + br

        if processed > 255:
            processed = 255
        PG.itemset((i, j), processed)
PB = cv2.equalizeHist(B)
for i in np.arange(ht):
    for j in np.arange(wd):
        original = PB.item(i, j)
        processed = original + br

        if processed > 255:
            processed = 255
        PB.itemset((i, j), processed)
img = cv2.merge((PR, PG, PB))

#####

# feeding the processed image to the network and call list of detection
detections = mymodel.return_predict(img)
# looping through the detections to extract coordinates, scores and class labels of the
object being detected from the image
for detection in detections:
    x1 = detection['topleft']['x']
    y1 = detection['topleft']['y']
    x2 = detection['bottomright']['x']
    y2 = detection['bottomright']['y']
    label = detection['label']
    score = detection['confidence']
    w = x2 - x1
    h = y2 - y1

    boxes.append([x1, y1, w, h])
    labels.append(label)
    scores.append(float(score))

```

```
# after sorting all detections to boxes, lables and scores, let's apply non max
suppression to reduce duplication
indices = cv2.dnn.NMSBoxes(boxes, scores, conf_threshold, nms_threshold)
#output = 'C:/Users/Abel/Desktop/mariyam/darkflow-master/output of the network'

# itrating through all selected indices of all boxes detected to only extract boxes
selected
for i in indices:
    i = i[0]
    box = boxes[i]
    xi = box[0]
    yi = box[1]
    wi = box[2]
    hi = box[3]
# applying scaling factor for the image (how many pixels corrsponds to 1
micrometer length) assumenig 15.8 pixels = 1 micrometer

    wsi = wi/15.8
    hsi = hi/15.8
    adi = (wsi + hsi)/2
    ari = adi/2
# average diameter of the cell
areai = math.pi*ari**2 # area of detected cell with index i
if wsi >= hsi: # comparing height and width
    Ci = hsi/wsi # circularity
else:
    Ci = wsi/hsi
label = labels[i]
score = scores[i]
##### lloping through labels to display the name of detected cell and to count
them based on their catagory

if label == 'Basophils':
    label = 'B'
    color = (255, 100, 100)
    NB += 1
    index = label + str(NB) + ',' + str(areai)
elif label == 'Eosinophils':
    label = 'E'
    color = (200, 50, 150)
    NE += 1
    index = label + str(NE) + ',' + str(areai)
elif label == 'Lymphocytes':
    label = 'L'
    color = (0, 255, 24)
    NL += 1
    index = label + str(NL) + ',' + str(areai)
elif label == 'Monocytes':
    label = 'M'
    color = (110, 50, 200)
```

```

    NM += 1
    index = label + str(NM) + ',' + str(areai)
elif label == 'Neutrophils':
    label = 'N'
    color = (255, 0, 24)
    NN += 1
    index = label + str(NN) + ',' + str(areai)
elif label == 'Platelets':
    label = 'P'
    color = (10, 10, 2)
    NP += 1
    index = label + str(NP) + ',' + str(areai)
elif label == 'Red-Blood-Cells':
    label = 'R'
    color = (10, 0, 255)
    NR += 1
    index = label + str(NR) + ',' + str(areai)
print(index)
cv2.rectangle(img, (xi, yi), (xi+wi, yi+wi), color, 1)
cv2.putText(img, label, (xi, yi), cv2.FONT_HERSHEY_COMPLEX, 0.3, (0, 0, 0),
1)
cv2.putText(img, str(score), (xi+20, yi+20), cv2.FONT_HERSHEY_COMPLEX,
0.5, (0, 255, 0), 1)
NWBC = NB + NE + NL + NM + NN    # total number of white blood cells in
image
cv2.namedWindow("output", cv2.WINDOW_NORMAL)
cv2.resizeWindow("output", 400, 300)
cv2.imshow("output", img)
cv2.imwrite(path, 'O1.JPEG', img)
print('number of Basophil are : ' + str(NB))
print('number of Eosinophil are : ' + str(NE))
print('number of Lymphocytes cells are : ' + str(NL))
print('number of Monocytes are : ' + str(NM))
print('number of Neutrophils are : ' + str(NN))
print('the total number of white blood cells are : ' + str(NWBC))
print('number of Platelet are : ' + str(NP))
print('number of red blood cells are : ' + str(NR))
print(img.shape)
cv2.waitKey(0)
cv2.destroyAllWindows()

```


Appendix D: Sample Images Showing Output of the Network

The following images shows output of the network in detection, classification, counting and measurement of morphological parameters.

1. Diameter measurement

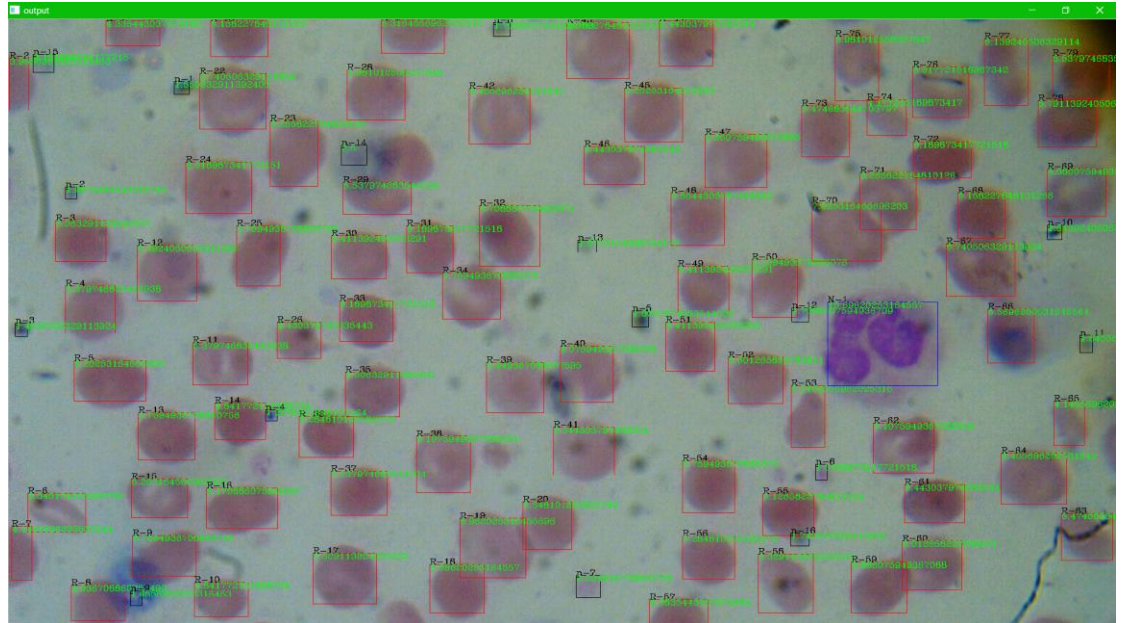


Figure 26: diameter measurement

2. Aspect ratio measurement

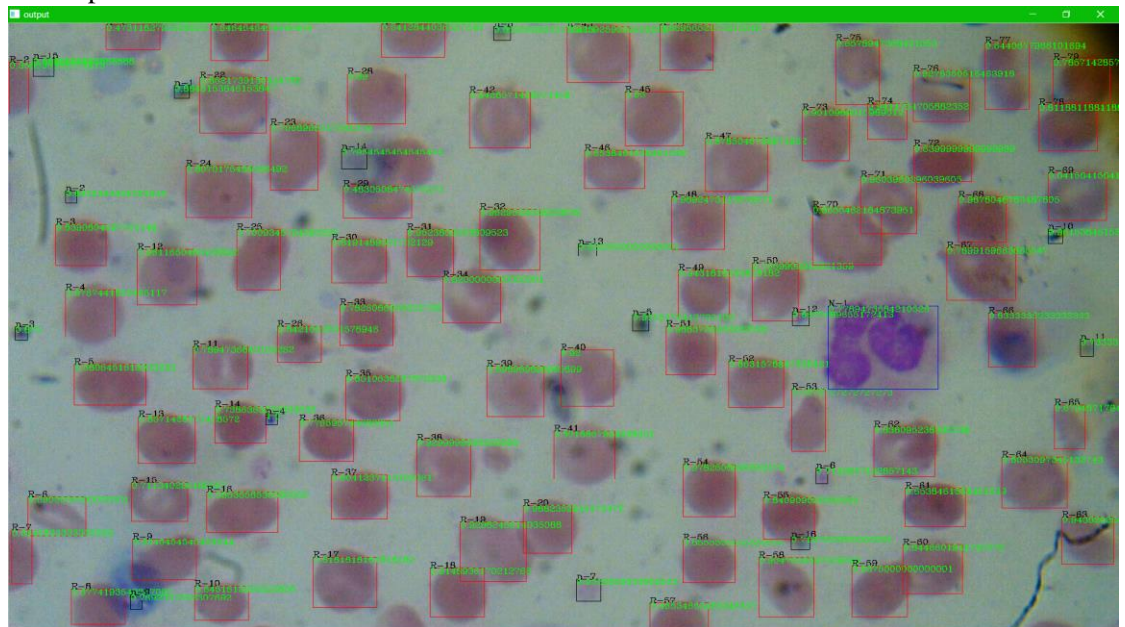


Figure 27: aspect ratio measurement

Appendix E: Comparison of a System Output With Manual

Methods

Area measurement

Blood cells		Manual (ImageJ)	Using Yolov2	Error%	Average error
RBCs	1	26.654	26.52	0.5%	5.3%
	2	24.981	22.63	9.4%	
	3	23.988	22.53	6.08%	
WBCs	B	73.59	69.86	5.068%	4.606%
	E	96.83	97.89	1.094%	
	L	35.494	34.100	3.9%	
	M	122.113	113.59	6.97%	
	N	89.043	90.231	1.33%	
Platelets	1	3.961	4.194	5.882%	11.214%
	2	4.491	4.03	10.26%	
	3	4.327	3.57	17.5%	

Table 11: The performance comparison of the system in measuring area of blood cells in μm^2

Aspect ratio measurement

Blood cells		Manual (ImageJ)	Using Yolov2	Error	Average error
RBCs	1	0.93	0.871	6.344%	9.095%
	2	0.83	0.91	9.64%	
	3	0.79	0.88	11.3%	
WBCs	B	0.817	0.81	0.9%	4.612%
	E	0.901	0.851	5.55%	
	L	0.87	0.91	4.60%	
	M	0.965	0.867	10.16%	
	N	0.92	0.903	1.85%	
Platelets	1	0.827	0.69	16.57%	10.45%
	2	0.827	0.78	5.68%	
	3	0.88	0.96	9.091%	

Table 12: The performance comparison of the system in measuring aspect ratio of blood cells

Diameter measurement

Blood cells		Manual (ImageJ)	Using Yolov2	Error%	Average error
RBCs	1	6.1	5.83	4.426%	8.929%
	2	5.43	6.13	12.89%	
	3	5.49	6.01	9.47%	
WBCs	B	14.57	12.91	11.39%	11.65%
	E	13.21	14.87	12.57%	
	L	16.20	18.13	11.91%	
	M	15.32	13.27	13.38%	
	N	12.06	10.97	9.01%	
Platelets	1	2.423	2.601	7.34%	13.213%
	2	3.12	3.052	2.20%	
	3	1.86	2.42	30.1%	

Table 13: The performance comparison of the system in measuring average diameter of blood cells in μm