

**U-NET ARCHITECTURE FOR SEGMENTATION OF  
OVERLAPPING RED BLOOD CELLS IN MALARIA  
BLOOD SMEAR IMAGES**

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

**By  
FATİH VEYSEL NURÇİN**

**In Partial Fulfillment of the Requirements for  
the Degree of Doctor of Philosophy  
in  
Biomedical Engineering**

**NICOSIA, 2021**

**FATİH VEYSEL  
NURÇİN**

**U-NET ARCHITECTURE FOR SEGMENTATION OF OVERLAPPING  
RED BLOOD CELLS IN MALARIA BLOOD SMEAR IMAGES**

**NEU  
2021**

**U-NET ARCHITECTURE FOR SEGMENTATION OF  
OVERLAPPING RED BLOOD CELLS IN MALARIA  
BLOOD SMEAR IMAGES**

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

**By  
FATİH VEYSEL NURÇİN**

**In Partial Fulfillment of the Requirements for  
the Degree of Doctor of Philosophy  
in  
Biomedical Engineering**

**NICOSIA, 2021**

**Fatih Veysel NURÇİN: U-NET ARCHITECTURE FOR SEGMENTATION OF OVERLAPPING RED BLOOD CELLS IN MALARIA BLOOD SMEAR IMAGES**

**Approval of Director of Institute of Graduate Studies**

**Prof. Dr. K. Hüsnü Can BAŞER**

**We certify this thesis is satisfactory for the award of the degree of Doctor of Philosophy in Biomedical Engineering**

**Examining Committee in Charge:**



Assoc. Prof. Dr. Murat Akkaya

Committee Chairman, Department of Management Information Systems, Faculty of Business and Economics, GAU



Prof. Dr. Ali Haydar

Department of Computer Engineering, Faculty of Engineering, GAU



Prof. Dr. Terin Adalı

Department of Biomedical Engineering, Faculty of Engineering, NEU



Assoc. Prof. Dr. Boran Şekeroğlu

Department of Information Systems Engineering, Faculty of Engineering, NEU



Assist. Prof. Dr. Elbrus Imanov

Supervisor, Department of Computer Engineering, Faculty of Engineering, NEU

I hereby declare that the current PhD thesis is my own original work and I have fully cited and referenced all sources used that are not original to this work. Further, it has not been submitted to any institution for assessment purposes.

Name, Last Name: Fatih Veysel Nurçin

Signature:

Date: 2.2.2021

## **ACKNOWLEDGEMENTS**

I would like to express my gratitude to my supervisor Assist. Prof. Dr. Elbrus Imanov for the consistent support and guidance throughout this thesis. I am grateful for his tolerant and helpful attitude from beginning to end.

I would also like to thank Prof. Dr. Terin Adalı for her long-term support and guidance since the undergraduate times.

To everyone who accompanied me on this journey...

## ABSTRACT

In order to assess red blood cells for certain diseases, automated red blood cell segmentation is a widely applied task. The counting of malaria parasites involves individual red blood cell segmentation that allows to determine the severity of the infection. The correct segmentation of red blood cells is needed for such an evaluation. However, due to the presence of overlapping red blood cells, it is a hard task. Pre-processing steps are used in existing methodologies to segment red blood cells. To provide fully automated segmentation of red blood cells without any initial preprocessing, we propose a deep learning approach that has a U-Net architecture. Irrelevant objects, such as white blood cells, platelets and artifacts, were removed while red blood cells were segmented. Data augmentation techniques were employed to increase sample size. Total of 5600 images were used to train the network whereas 600 images were used for testing. The network achieved segmentation of red blood cells with 93.8% Jaccard similarity index.

**Keywords:** Overlapping red blood cells; segmentation; deep learning; U-net; malaria

## ÖZET

Otomatize kırmızı kan hücreleri segmentasyonu, kırmızı kan hücrelerinin belirli hastalıklar için birebir değerlendirilmesi için önemli bir adımdır. Parazitlerin sayımı için kırmızı kan hücrelerinin birebir segmentasyonu gereklidir ve böylece enfeksiyonun şiddeti belirlenebilir. Kırmızı kan hücrelerinin doğru segmentasyonu böyle bir değerlendirme için önemlidir. Fakat, üst üste binen kırmızı kan hücrelerin varlığı bu segmentasyonu zorlu kılar. Mevcut metodolojiler ön işleme adımlarını kullanarak kırmızı kan hücrelerinin segmentasyonunu sağlar. Bu tezde, kırmızı kan hücrelerinin herhangi bir ön işleme olmadan tam otomatik segmentasyonunu sağlamak için, U-Net mimarisine sahip bir derin öğrenme yaklaşımı öneriyoruz. Sonuç olarak kırmızı kan hücreleri segmente edilirken, beyaz kan hücreleri, trombositler ve gürültü gibi ilgisiz nesnelere çıkarılması sağlandı. Örnek boyutunu artırmak için veri artırma teknikleri kullanıldı ve ağı eğitmek için 5600, test etmek için ise 600 görüntüden faydalanıldı. Ağı, 93,8% Jaccard benzerlik indeksi ile kırmızı kan hücrelerinin segmentasyonunu sağladı.

**Anahtar Kelimeler:** Üst üste binen kırmızı kan hücreleri; segmentasyon; derin öğrenme; U-net; sıtma

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>ÖZET</b> .....	iv
<b>TABLE OF CONTENTS</b> .....	v
<b>LIST OF TABLES</b> .....	viii
<b>LIST OF FIGURES</b> .....	ix
<b>LIST OF ABBREVIATIONS</b> .....	xi
<b>CHAPTER 1: INTRODUCTION</b> .....	1
<b>CHAPTER 2: CLINICAL BACKGROUND</b>	
2.1 How to Diagnose Malaria.....	5
2.2 Malaria Incidence Rate.....	6
2.3 Burden of the Malaria.....	6
2.4 Malaria Health Consequences for Pregnant, Infant and Child.....	7
2.5 The Life Cycle of Malaria Parasite.....	7
2.6 Blood Films.....	8
2.7 Stages of malaria parasites.....	9
<b>CHAPTER 3: TECHNICAL BACKGROUND</b>	
3.1 Convolutional Neural Networks Introduction.....	11
3.1.1 Convolutional layer.....	11
3.1.2 Filters.....	13
3.1.3 Volume of feature maps.....	16
3.1.4 The rectified linear unit.....	18
3.1.5 Pooling.....	19
3.1.6 Gradient Descent.....	20
3.1.7 Learning rate.....	23
3.1.8 Back-propagation.....	24
3.1.9 Optimizers.....	25

3.1.10 Data augmentation.....	25
3.1.11 Autoencoders.....	26
3.2 Image Processing Background.....	28
3.2.1 Spatial filtering.....	28
3.2.2 Averaging filters.....	29
3.2.3 Filtering in frequency domain.....	29
3.2.4 Median filter.....	31
3.2.5 Sharpening filters.....	31
3.2.6 Laplacian filter.....	32
3.2.7 Morphological operators.....	32
3.2.8 Dilation and erosion.....	33
3.2.9 Connectivity.....	35
3.2.10 Threshold segmentation.....	36
3.2.11 Global thresholding.....	37
3.2.12 Local dynamic thresholding.....	37
3.2.13 Region growing.....	37
3.2.14 Clustering.....	38
3.2.15 Watershed segmentation.....	38

## **CHAPTER 4: MATERIAL AND METHOD**

4.1 Neural Networks Architecture.....	40
4.1.2 Data Augmentation.....	41
4.1.3 Optimizer.....	42
4.1.4 Evaluation Metric.....	43
4.2 Local Minimum Histogram Segmentation wit Selective Hole Filling Approach.....	43
4.2.1 Preprocessing.....	44
4.2.2 Colour Segmentation.....	47
4.2.3 White blood cell segmentation.....	47
4.2.4 Dust particles segmentation.....	48
4.2.5 Refilling of holes.....	48

4.2.6 Convex hull.....	49
4.2.7 Hough transform for circle detection.....	49
4.2.8 Watershed segmentation.....	50
4.2.9 Evaluation.....	51
4.3 Data Summary.....	52
4.3.1 U-net architecture.....	52
4.3.2 Selective hole filling.....	52
<b>CHAPTER 5: RESULTS</b>	
5.1 U-Net Architecture.....	53
5.2 Marker-controlled Watershed Segmentation.....	56
<b>CHAPTER 6: DISCUSSION.....</b>	<b>61</b>
<b>CHAPTER 7: CONCLUSION.....</b>	<b>63</b>
<b>REFERENCES.....</b>	<b>64</b>
<b>APPENDICES.....</b>	<b>70</b>
Appendix 1: Ethical Approval Document.....	71
Appendix 2: Signed Similarity Report.....	72
<b>CURRICULUM VITAE.....</b>	<b>73</b>

## LIST OF TABLES

<b>Table 5.1:</b> Results of U-net architecture on different data divisions.....	55
<b>Table 5.2:</b> Jaccard similarity coefficient (U-net vs others) .....	55
<b>Table 5.3:</b> Jaccard similarity coefficient results (Proposed watershed vs others).....	57
<b>Table 5.4:</b> Performance evaluation on high number of overlapping cells samples.....	57

## LIST OF FIGURES

<b>Figure 2.1:</b> The incident rate for countries .....	6
<b>Figure 2.2:</b> Malaria parasite life cycle .....	8
<b>Figure 2.3:</b> Trophozoites on top, schizonts at the middle and gametocytes at the bottom	10
<b>Figure 3.1:</b> Layers of CNNs and local rectangular receptive fields.....	11
<b>Figure 3.2:</b> Layer connections and zero padding .....	12
<b>Figure 3.3:</b> Stride of receptive fields with value of 2 .....	13
<b>Figure 3.4:</b> Horizontal and vertical line filters applied to the image .....	14
<b>Figure 3.5:</b> Example 3x3 CNN filter .....	15
<b>Figure 3.6:</b> In the CNN layer, representing filters and feature maps as neurons .....	16
<b>Figure 3.7:</b> CNN with multiple feature maps and input with RGB color format .....	17
<b>Figure 3.8:</b> ReLU output .....	19
<b>Figure 3.9:</b> Result of 2x2 pooling with stride = 2 applied to similar images .....	20
<b>Figure 3.10:</b> Dimension reduction as result of max-pooling with stride 1 and 2 .....	21
<b>Figure 3.11:</b> Minimization of error with respect to weights .....	22
<b>Figure 3.12:</b> Error surface presented with a set of contours .....	23
<b>Figure 3.13:</b> Difficulty of convergence with large learning rate.....	24
<b>Figure 3.14:</b> Data augmentation for generation of images from a single image .....	26
<b>Figure 3.15:</b> Structure of the autoencoders .....	27
<b>Figure 3.16:</b> Input-output relationship of autoencoders .....	28
<b>Figure 3.17:</b> A filter kernel with its coefficients applied to the corresponding pixel in the image.....	30
<b>Figure 3.18:</b> Filter coefficients of box filter kernel on left and Gaussian filter kernel on right.....	31
<b>Figure 3.19:</b> Low pass filter.....	32
<b>Figure 3.20:</b> An example of Laplacian filter.....	32
<b>Figure 3.21:</b> Output of structuring element interaction with binary image where erosion on top and dilation at bottom.....	33
<b>Figure 3.22:</b> The structuring element is visualized as a ball rolling in the image and places it touches are resulting output for morphological opening.....	34

<b>Figure 3.23:</b> The structuring element is visualized as a ball rolling at outer boundary of image and the contours it draws are resulting output for morphological closing.....	35
<b>Figure 3.24:</b> Direction of 4 and 8-connected pixels.....	36
<b>Figure 3.25:</b> Application of thresholding.....	37
<b>Figure 3.26:</b> Region growing steps.....	38
<b>Figure 3.27:</b> Watershed transformation steps.....	39
<b>Figure 4.1:</b> The U-net architecture .....	41
<b>Figure 4.2:</b> Random elastic formation applied to a image.....	42
<b>Figure 4.3:</b> Flowchart of selective hole filling approach.....	45
<b>Figure 4.4:</b> Local minima on smoothed histogram are marked with red stars. Smoothed histogram is illustrated with an orange sloping line on the histogram.....	46
<b>Figure 4.5:</b> Illustration of proposed local minimum histogram background segmentation steps.....	50
<b>Figure 5.1:</b> Original image and its corresponding label.....	53
<b>Figure 5.2:</b> Training and validation loss expressed as epoch on x axis and loss on y axis	54
<b>Figure 5.3:</b> Segmentation and separation of overlapping red blood cells.....	55
<b>Figure 5.4:</b> Segmentation of red blood cells with central pallor.....	56
<b>Figure 5.5:</b> Segmentation of red blood cells with presence of white blood cell.....	56
<b>Figure 5.6:</b> Comparison of two dataset with different number overlapping red blood cell distribution.....	58
<b>Figure 5.7:</b> Illustration of markers.....	59
<b>Figure 5.8:</b> Separation of overlapping red blood cells.....	59

## LIST OF ABBREVIATIONS

<b>WHO:</b>	World Health Organization
<b>SGD:</b>	Stochastic gradient descent
<b>RBC:</b>	Red blood cells
<b>WBC:</b>	White blood cells
<b>pRBC:</b>	Parasitized red blood cells
<b>RBCs:</b>	Red blood cells
<b>WBCs:</b>	White blood cells
<b>CNN:</b>	Convolutional neural network
<b>CNNs:</b>	Convolutional neural networks
<b>ReLU:</b>	Rectified Linear Unit
<b>HSV:</b>	Hue Saturation Value
<b>RGB:</b>	Red Green Blue

## **CHAPTER 1**

### **INTRODUCTION**

Malaria is transmitted by the bite of female *Anopheles* mosquitoes that infects red blood cells (RBCs) with protozoan parasites of the genus *Plasmodium*. Humans can be infected by 5 *Plasmodium* species that can cause malaria: *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium malariae*, *Plasmodium ovale*, and *Plasmodium knowlesi* (Poostchi et al., 2018b). In the World Malaria Report 2019, *Plasmodium falciparum* accounted for 99,7% of all the cases in WHO African region. World Malaria Report 2019 noted an estimated number of 228 million cases of malaria across the world. The highest concentration of cases with 93% is in WHO African region followed by 3.4% and 2.1% for the WHO South-East Asia Region and the WHO Eastern Mediterranean Region respectively (World Health Organization, 2020). Diagnosis of malaria relies on trained microscopists who manually examine blood films. The microscopist analyses red blood cells for the presence of malaria parasites by a conventional microscope with respect to WHO guideline (World Health Organization, 2010). In case of infection, manual counting of malaria parasites is followed through. In addition to detection, counting the number of malaria parasites is necessary to classify disease progression and test drug-resistance, effectiveness (Poostchi et al., 2018b; Pillay et al., 2019). On the other hand, this procedure relies on the skill and experience of the specialist. In addition to this, external conditions such as lack of resources to work in isolation can lower the performance of the specialist. This can result in false diagnosis that will lead to false treatment. In case of false-positives, the patient can be affected by possible side effects of anti-malarial drugs and in case of false-negatives, the disease can progress into a severe state accompanied by unnecessary use of antibiotics (Poostchi et al., 2018b; Pillay et al., 2019; Loddo et al., 2018). As the manual approach is prone to human errors, automated approaches that can provide a standardized diagnosis while reducing the workload of field workers becomes important.

The task of automated quantification of malaria parasites consists of several steps. The first step is preprocessing of images to correct uneven illumination and colour differences between samples due to staining conditions. The second step includes the segmentation of red blood cells followed by the separation of overlapped red blood cells in order to accurately count red blood cells. At the end of this step, usually, post-processing methods are acquired

to remove remaining wastes. The third step is the extraction of features from each red blood cell in order to classify each red blood cell as parasitized or healthy. The efficiency of the third step relies on the second step, where the separation of overlapped red blood cells in cell clumps takes place. The final step is classification, where extracted features are used by an algorithm to automatically assess red blood cells as healthy or parasitized.

The pre-processing step usually handles uneven illumination and noise. The grayscale conversion is commonly employed for illumination correction whereas, morphological operators remove noises that include objects smaller than red blood cells (Loddo et al., 2018). Many algorithms are utilized to segment red blood cells from the background in an attempt to remove irrelevant information on thick and thin blood smears. Global and adaptive thresholding, k-means clustering, morphological operators were utilized in red blood cell segmentation task in many studies. Savkare and Narrote (2011) used Otsu's threshold (Global thresholding) on the enhanced green channel for segmentation of red blood cells. In another study, an adaptive thresholding technique combined with mathematical morphological operators were employed in order to deal with noise variations (Dave and Upla, 2017). Mushabe et al. (2013) used Zach's thresholding to extract red blood cells on limited variety of images. Top-hat filtering was employed by Tek et al. (2010) which utilizes morphological area for segmentation.

Segmented cell clumps are to be separated to evaluate each red blood cell and to count infected red blood cells accurately.

Zafari et al. (2015) proposed a novel ellipse fitting approach for the separation of partially overlapping objects. However, this method didn't produce comparable results for varying shapes of RBCs. Morphological operators were utilized in the separation of overlapped red blood cells by Di Ruberto et al. (2002). Integrity and circularity of red blood cells were enhanced by hemispherical disk-shaped structuring element and red blood cells were separated by disc-shaped flat structuring element to improve the efficiency of watershed segmentation. Savkare and Narrote (2015) employed the k-means algorithm ( $k = 2$ ) to segment red blood cells from the background. Later, edge detection followed by watershed segmentation was applied to separate overlapping red blood cells. Marker-controlled watershed transformation is a commonly used algorithm in biomedical images and introduced to limit over-segmentation with conventional watershed segmentation. Marker-controlled watershed segmentation is also utilized in the separation of overlapped red blood

cells. The word “marker” here refers to the foreground and background markers. The connected blob of pixels inside each segmented object is foreground markers whereas background pixels are background markers. Foreground markers are linked to the specific watershed region. The location and number of foreground markers contribute to the accurate separation of overlapping objects. González-Betancourt et al. (2016) employed Radon transform in an attempt to improve the selection of markers. However, the only improvement achieved was in running time.

At the classification step, many algorithms were utilized to evaluate the presence of malaria parasites. In a study, the level set method was utilized to extract stained red blood cells and stained objects were used to train the neural network. However, healthy red blood cells were excluded as they were not stained (Bibin et al., 2017). In another study, the level set method was used to segment red blood cells and segmented red blood cells were used to train several pre-trained models. However, in this study white blood cells were filtered by manual ground truth annotation (Rajaraman et al., 2018). Moreover, Vijayalakshmi and Rajesh Kanna (2020) employed a pre-trained Visual Geometry Group (VGG) model to extract the mid-level representation of malaria images. Extracted features were used in the Support Vector Machine (SVM) classifier to evaluate the presence of malaria parasites rather than the evaluation of each red blood cell. In another study, a large number of pre-segmented images were used to train convolutional neural networks (Reddy and Juliet, 2019). However, extraction, separation of red blood cells, and removal of artifacts were not included. A recent work utilized a popular global thresholding algorithm, Otsu’s thresholding, and watershed segmentation to segment and separate red blood cells. Texture and color features were used to train SVM and Linear Discriminant Analysis (LDA) algorithms. However, the removal of white blood cells was not mentioned in the study which suggests they were ignored (Molina et al., 2020).

In this thesis, we propose U-net architecture for segmentation and separation of red blood cells while removing irrelevant objects such as white blood cells and platelets. The data is labeled in Matlab 2019a by using image labeller app. The Labelling based on using polygons to label white blood cells while excluding the background and touching points between overlapping red blood cells. The idea of marking touching points between overlapping red blood cells, allow network to learn separating overlapping red blood cells.

On the other hand, we also implemented marker-controlled watershed segmentation with local minimum histogram segmentation with novel selective hole filling approach along with pre-processing and post-processing steps. For this study, larger images that encompass higher of number red blood cells were used.

## **CHAPTER 2**

### **CLINICAL BACKGROUND**

In certain parts of the world, malaria is a significant public health problem. Disease attacks can be serious and, if untreated, can lead easily to death. There are also chronically sick residents of populations with high levels of malaria, which leads to absences from work and education. Repeated incidents not only contribute to high treatment spending, they can affect income, the amount of food that can be grown by the household, and the income that a family receives. For pregnant women and children, malaria holds a significant risk and is a frequent cause of miscarriage. Chronic diseases, poverty, lack of knowledge on malaria, united creates a vicious cycle (World Health Organization, 2010).

A microscopic living organism, called a parasite, causes malaria, which infects the red blood cells of a human. The transmission occurs via bite of female *Anopheles* mosquitoes. Before infection may take place, the parasite must go through a complicated cycle in both the mosquito and in human beings. Depending on a variety of factors, such as the malaria parasite species, the air temperature, and the relative humidity, the cycle continues for 1-3 weeks in the female *Anopheles* mosquito. Attempts have been made to control the disease with prompt diagnosis, application of treated nets and appropriate treatment. This contributed in some countries to substantial decreases in mortality and morbidity and in some places the disease is still the primary cause of deaths and illness. The symptoms (common) include headache, high fever, profuse sweating, rigor or severe chills and general body pains. However, these symptoms are seen in other diseases as well, more information required before diagnosis is made. In patients who have had a number of malaria attacks, the clinical presentation of malaria is much less evident, as they usually show no clear signs or symptoms (World Health Organization, 2010).

#### **2.1 How to Diagnose Malaria**

The malaria parasite is microscopically small and can be only visualized under microscope with appropriate magnification. However, there is a procedure to prepare the sample for the microscope. The blood film is dried and stained for further examination. In this process, parasites are stained and detected by the microscopist for diagnosis. The stages and species of malaria parasite and the density of the infection can be established by microscopists who



and India region where nearly half of the total malaria deaths occurred in Nigeria (World Health Organization, 2020).

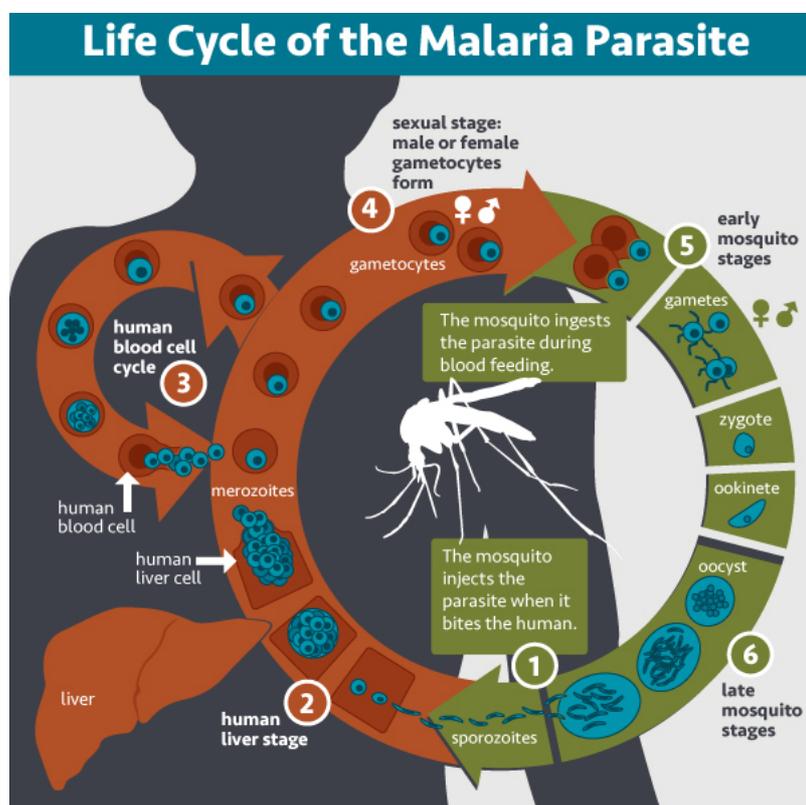
#### **2.4 Malaria Health Consequences for Pregnant, Infant and Child**

Malaria infected pregnant women are under high risk due to several complications that might occur. More specifically, the women, her fetus and her newborn child can be affected due to infection. The acquired immunity of individual and transmission intensity for corresponding geographical area plays role in symptoms and complications of the pregnant. *P. falciparum* infection which is common in Africa mostly causes anemia and maternal illness, low weight birth and preterm birth. Additionally, it is important factor in maternal deaths before and after childbirth. Moreover, preterm births and low weight children at birth as a result of malaria causes cognitive development and child growth problems along with being risk factor for infant, neonatal, perinatal mortality. In high acquired immunity levels, the disease can be asymptomatic during pregnancy while parasite might be present in placenta and can cause anemia regardless of absence of peripheral parasitemia. *P. vivax* infection contributes to chronic anemia, which reduces birth weight and raises the likelihood of neonatal mortality. The decline in birthweight attributed to *P. vivax* infection is nearly two-thirds of the decrease associated with *P. falciparum* (World Health Organization, 2020).

#### **2.5 The Life Cycle of Malaria Parasite**

There are 5 species of malaria plasmodium that cause malaria in human follows as *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium malariae*, *Plasmodium ovale*, and *Plasmodium knowlesi*. *P. falciparum* is most lethal form amongst them and *P. vivax* has parasites in dormant liver stage. Plasmodium has life cycle that is shared in both human and mosquito. The plasmodium sporozoites present in salivary glands are transmitted from infected *Anopheles* mosquito to human with biting that leads release of sporozoites into blood stream to eventually end up with the liver. The liver becomes infected by sporozoites and sporozoites are multiplied in here. Parasite forms called merozoites are subsequently released into the blood where they invade red blood cells (RBCs) in less than a minute and mature inside them. As they feed on hemoglobin, ring stages develop into trophozoites, and then divide their nuclei to form schizonts, in order to invade new RBCs from the next generation of merozoites, the so-called intraerythrocytic cycle is perpetuated. In each cycle,

a small number of parasitized RBCs (pRBCs) differentiate towards the formation of gametocytes, sexual forms of the parasite that sequester until they mature and return to circulation in the bone marrow. Gametocytes egress from the pRBC when removed with a mosquito blood meal and become gametes. Fertilization creates a zygote that becomes a motile ookinete, leaving the blood bolus to enter and cross the endothelium of the mosquito midgut, differentiating it into an oocyst, that grows to spawn sporozoites. They close the cycle by moving through the haemolymph to the salivary glands of the mosquito, where they are prepared for new inoculation into a human host (Borgheti-Cardoso et al., 2020). This cycle is illustrated in Figure 2.2.



**Figure 2.2:** Malaria parasite life cycle (National Institute of Allergy and Infectious Diseases., 2016)

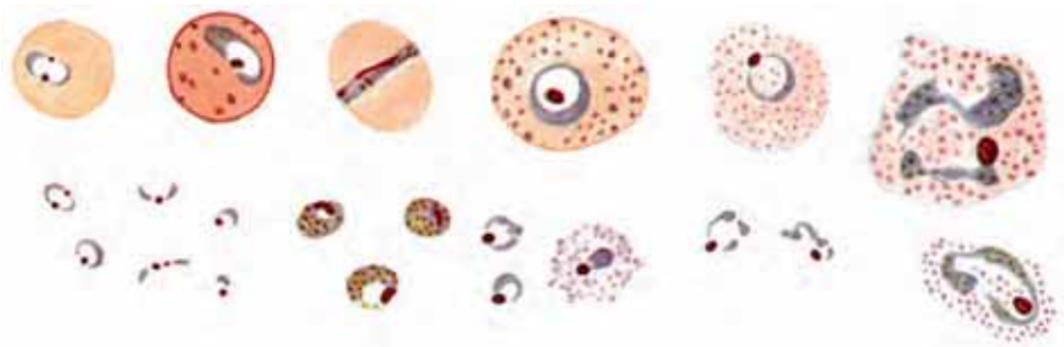
## 2.6 Blood Films

There are two type film that are prepared for microscopic investigation, thick and thin blood films. Thick films are used for parasite detection as it consists of many layers that includes white blood cell and red blood cells that are more concentrated than thin films. On the other

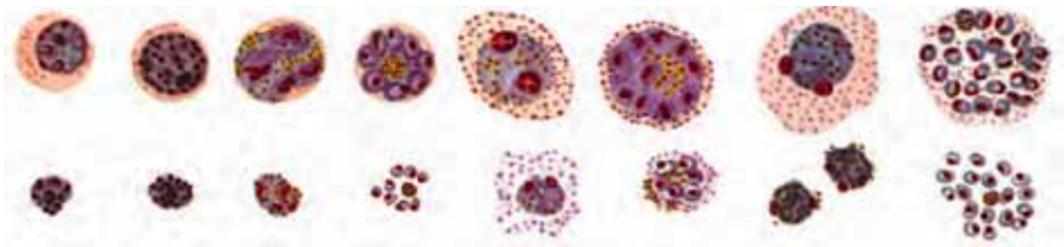
hand, thin films are investigated for malaria species detection (World Health Organization, 2010).

### **2.7 Stages of Malaria Parasites**

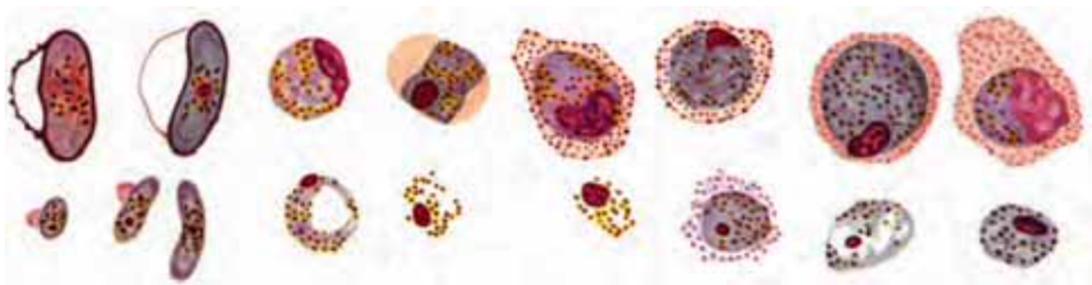
There are 3 stages of malaria parasites. First stage is trophozoite, second schizont and third gametocyte. The trophozoite stage is the most common one, where chromatin dot and ring like structure is observed. At the second stage, chromatin divides into two, parasites begin to asexually reproduce and chromatin division continues for several times. At the final stage, the parasite progress into a female or a male gametocyte where it prepares for the sexual phase in the female mosquito vector of Anopheles (World Health Organization, 2010). All three stages on thick and thin blood films are illustrated in Figure 2.3.



**Upper line: thin film parasites. Lower line: thick film parasites**



**Upper line: thin film schizonts. Lower line: thick film schizonts**



**Upper line: thin film male, female gametocytes. Lower line: thick film gametocytes**

**Figure 2.3:** Trophozoites on top, schizonts at the middle and gametocytes at the bottom (World Health Organization, 2010).

## CHAPTER 3

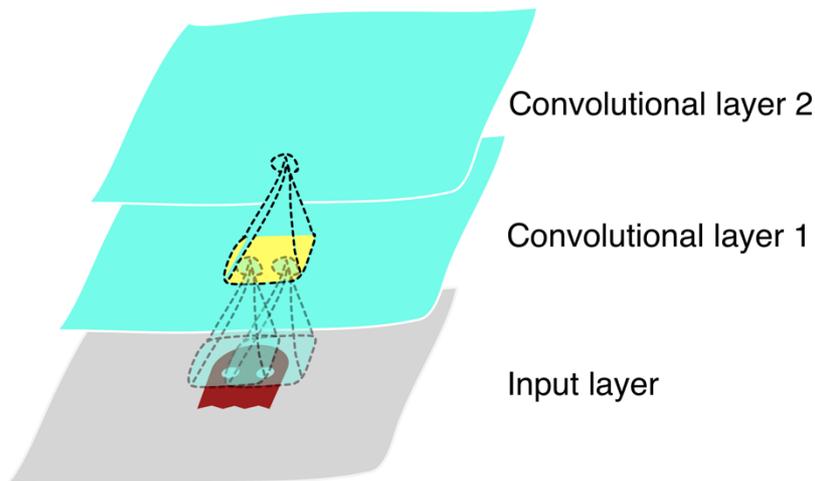
### TECHNICAL BACKGROUND

#### 3.1 Convolutional Neural Networks Introduction

Convolutional neural networks (CNNs) are a specific type of neural network for the processing of data with a known grid-like topology. Time series and images can be used as examples with 1-D grid samples with defined intervals and image with 2-D grid of pixels respectively. Practical applications of CNNs have showed impressive results. Convolutional neural network takes its name from the mathematical operation of convolution. So that, CNNs employ convolution which is specific type of liner operation to obtain general matrix multiplication (Goodfellow et al., 2016).

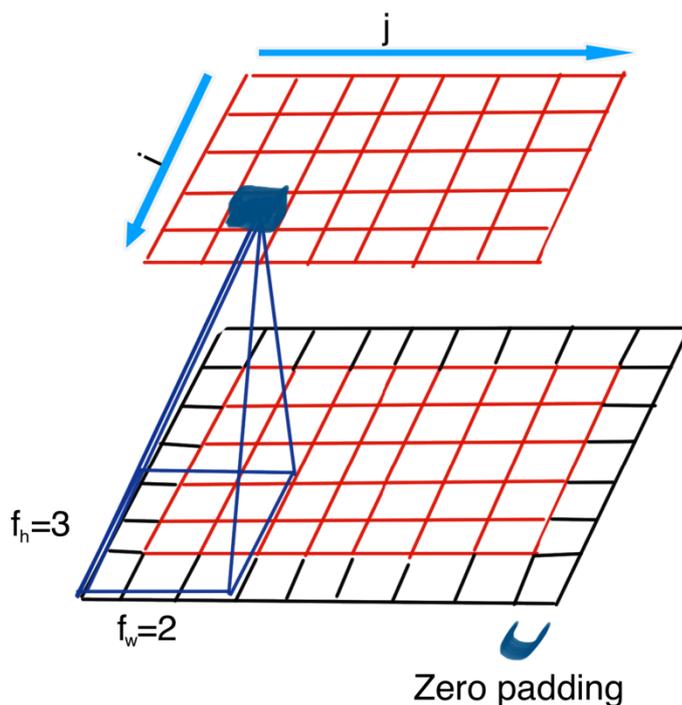
##### 3.1.1 Convolutional layer

The convolutional layer is the most crucial part of the CNNs. The neurons in the first convolutional layer are connected to the pixels of the input image that are in their receptive field. Similarly, neurons in the second convolutional layers are connected to pixels in their receptive field (small rectangle) in the first layer. The architecture allows small low level features to be concentrated in first hidden layer and brings together small low level features for larger higher level features in second hidden layer (Géron, 2019). This can go on for many layers. This hierarchical structure is illustrated in Figure 3.1.



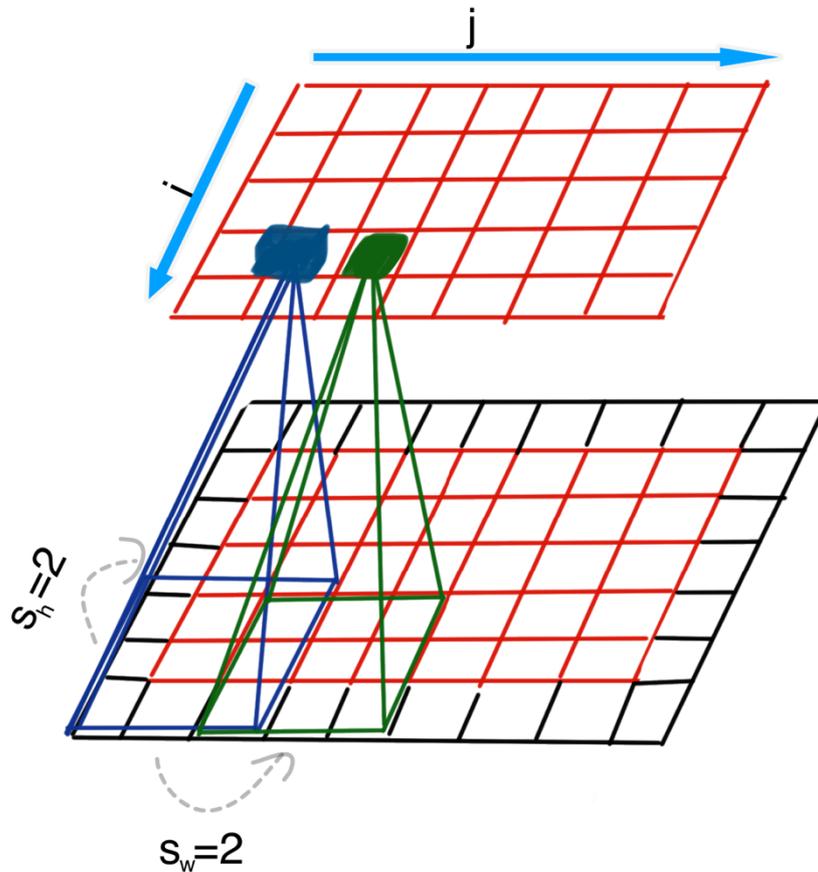
**Figure 3.1:** Layers of CNNs and local rectangular receptive fields (Géron, 2019)

For the given image in Figure 3.2, it is shown that a neuron at  $i, j$  (row, column) is connected to the output of neurons in rectangular area of the previous layer at  $i$  to  $i + f_h - 1$  and  $j$  to  $j + f_w - 1$  (row, column) as  $f_h$  and  $f_w$  denote for height and width of the receptive field respectively (Géron, 2019).



**Figure 3.2:** Layer connections and zero padding (Géron, 2019)

Zero padding is necessary in order to keep the size (height, width) of the next as the previous one. Common approach is adding zeros at the outer border of the image that is illustrated in Figure 3.2. On the other hand, receptive fields can be spaced out to connect large input to smaller next layer which can be seen in Figure 3.3. As receptive field travels through the image, it shifts by steps. The number of steps is defined as stride. In the Figure 3.3 larger input layer with  $5 \times 7$  size is connected to layer with  $3 \times 4$  size through  $3 \times 3$  receptive field and stride = 2. Note that,  $5 \times 7$  input image is zero padded and stride value is applied to both directions. In Figure 3.3 it is shown that, a neuron at  $i, j$  (row, column) in next layer is connected to the output of the previous layer in the receptive field that includes  $i \times s_h$  to  $i \times s_h + f_h - 1$ ,  $j \times s_w$  to  $j \times s_w + f_w - 1$  (row, column), as  $s_w$  is horizontal and  $s_h$  is vertical strides (Géron, 2019).

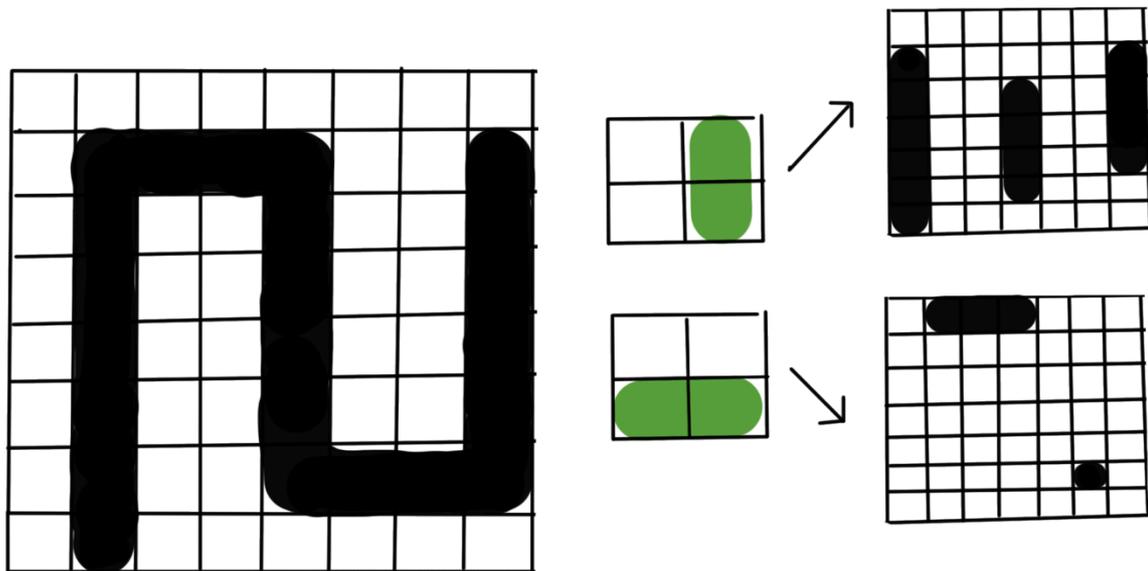


**Figure 3.3:** Stride of receptive fields with value of 2 (Géron, 2019)

### 3.1.2 Filters

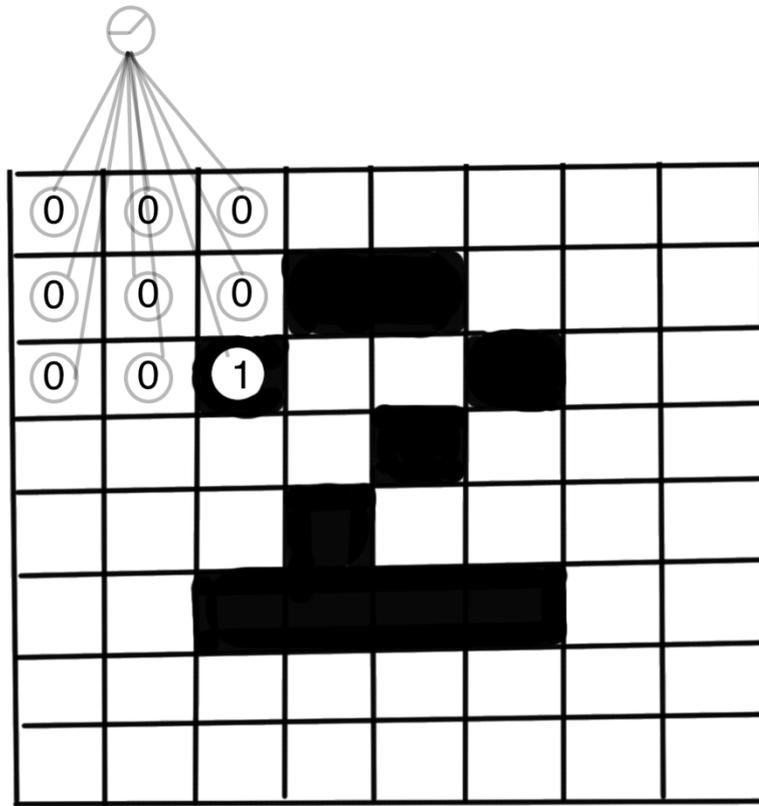
David Hubel and Torsten Wiesel discovered that some neurons in the visual cortex responded to horizontal, others to vertical lines and still others to lines with a particular angle in 1959. This was measured by placing electrodes into the brain of a cat while the cat was exposed to black and white patterns. Moreover, it was assessed that each layer depends on the previous layer, the features detected from the previous layer is used to build the next layer. So that, entire object depends on shapes, shapes to contours, contours to lines. Additionally, features detectors were replicated over the whole area in a layer of the visual cortex to capture features in entire region of an image which gave inspiration to design of CNNs (Buduma and Locascio, 2017).

The idea of a filter was introduced initially in which Viola and Jones had a close estimate. A filter can be considered as a feature detector which is illustrated in Figure 3.4.



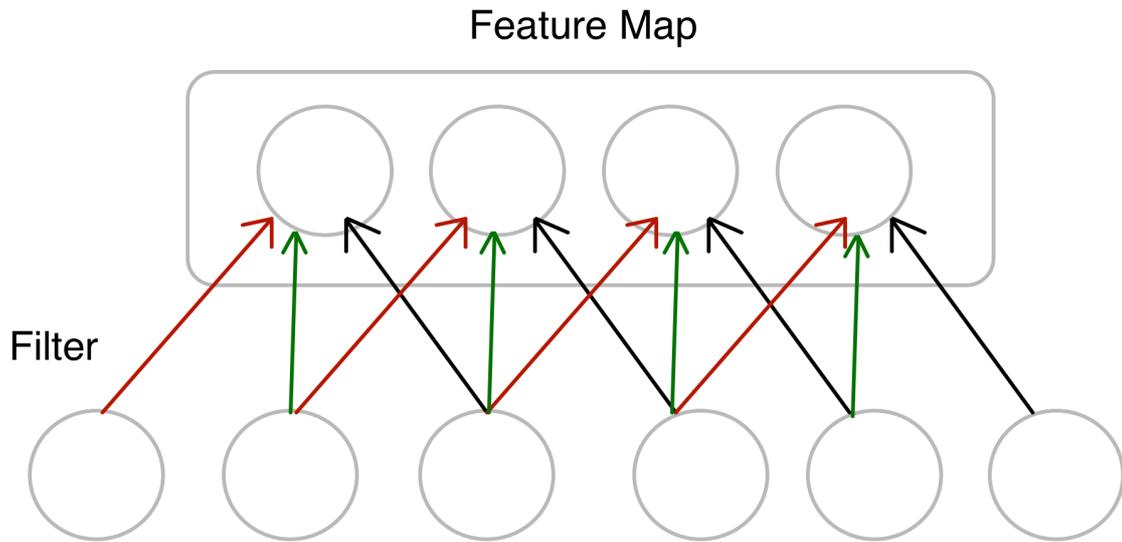
**Figure 3.4:** Horizontal and vertical line filters applied to the image (Buduma and Locascio, 2017).

The application of a feature detector and the resulted output can be understood by the example given in Figure 3.4. Here, the image consists of vertical and horizontal lines, and 2 feature detectors are acquired. The feature detector on top is slid over the entire image and each time it matches the image box it is on, a vertical line is detected. This process works similarly with the feature detector at the bottom (Buduma and Locascio, 2017). The corresponding output from these feature detectors is defined as a feature map on the right in Figure 3.4. The operation of filter sliding over the image is defined as convolution. This operation can be applied with different size of filters such in Figure 3.5 where 3x3 filter is utilized.



**Figure 3.5:** Example 3x3 CNN filter (Trask, 2019)

Combination of connections replicated for whole region of the image, connections with same color are limited to same weights that is illustrated in Figure 3.6. All connections in a group are initialized with same weights and average of updated weights in group is taken to be applied after each iteration of backpropagation. The resulted output is feature map. If a filter detects a feature at its position that is connected to a neuron in the feature map, that neuron is activated (Buduma and Locascio, 2017).



**Figure 3.6:** In the CNN layer, representing filters and feature maps as neurons (Buduma and Locascio, 2017)

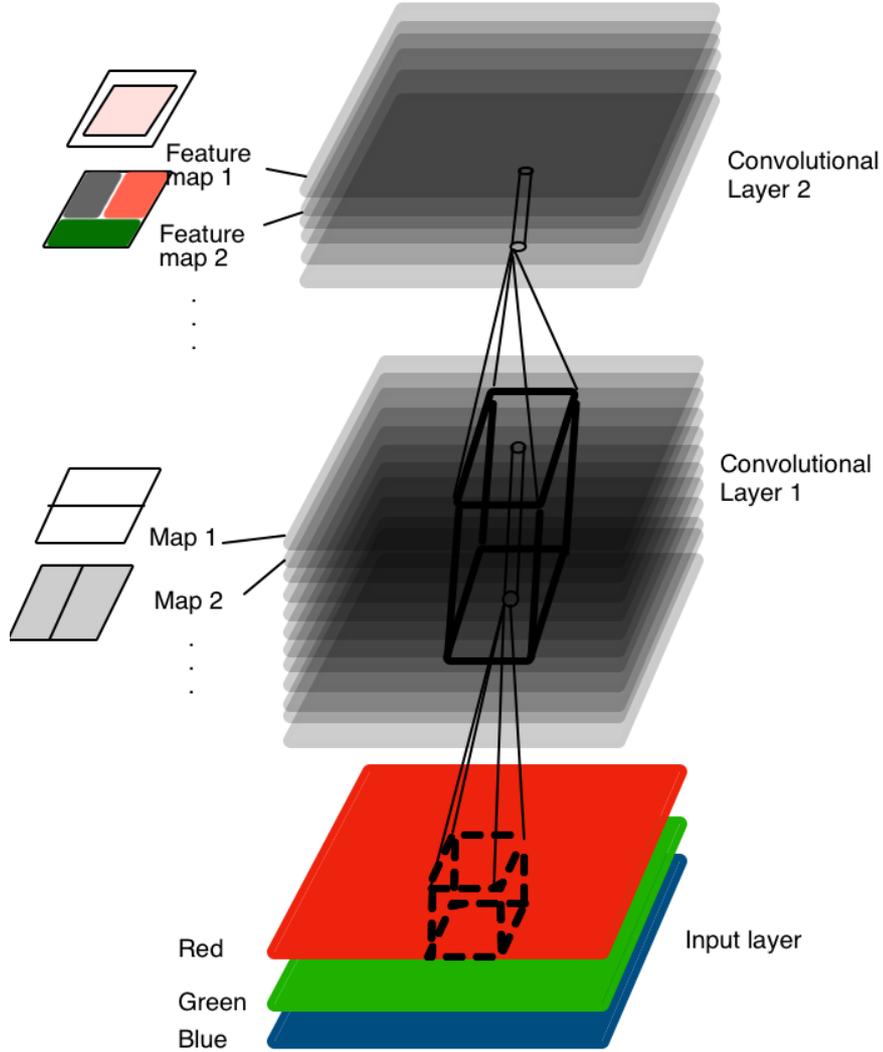
Feature map can be mathematically expressed as:

$$m_{ij}^k = f((w * x)_{ij} + b^k) \quad (3.1)$$

where  $m^k$  is  $k^{\text{th}}$  feature map in the layer  $m$  with its corresponding filter values  $w$  and  $b^k$  is bias for neurons in feature map. However, this is not complete description for filters as filters operate on volume of features maps instead of only one (Buduma and Locascio, 2017).

### 3.1.3 Volume of feature maps

In practical applications, there isn't a single but multiple feature maps that is illustrated in Figure 3.7. Each feature map corresponds to a filter. In each feature map of the convolutional layer has one neuron per pixel, and all neurons within a given feature map share the same weights and bias. These multiple trainable filters applied to input enables detection of multiple features. On top of that, input images can also be composed of multiple channels such red, green, blue color format abbreviated as RGB. The number of layers can depend on the nature of the image, it could be one layer as grayscale or could be much more such satellite images (Géron, 2019).



**Figure 3.7:** CNN with multiple feature maps and input with RGB color format (Géron, 2019)

For given convolutional layer  $l$ , a neuron in the feature map  $k$  located at  $i, j$  (row, column) that is connected to output of neurons in the previous layer  $l-1$ , at  $i \times s_h$  to  $i \times s_h + f_h - 1, j \times s_w$  to  $j \times s_w + f_w - 1$  (row, column).

Output of a neuron for a specific convolutional layer is mathematically expressed as:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{n'-1} x_{i',j',k'} \cdot w_{u,v,k',k} \text{ with } \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \quad (3.2)$$

where  $z_{i,j,k}$  refers to output of the neuron in the feature map  $k$  with  $i, j$  (row, column) coordinates for convolutional layer  $l$  and  $s_w, s_h$  are horizontal, vertical strides. The width and height of the receptive field are denoted by  $f_w$  and  $f_h$ . The number of feature maps is  $f_n$  in the layer before ( $l-1$ ). The bias term for feature map in layer  $l$  is  $b_k$  and  $x_{i,j,k}$  refers to output of the neuron in the layer before ( $l-1$ ) for corresponding row column and feature map as  $i, j, k$ . Finally,  $w_{u,v,k',k}$  is weight for a neuron in feature map  $k$  in the layer  $l$  for its corresponding input at  $u,v$  (row, column) with respect to receptive field of neuron and feature map  $k'$  (Géron, 2019).

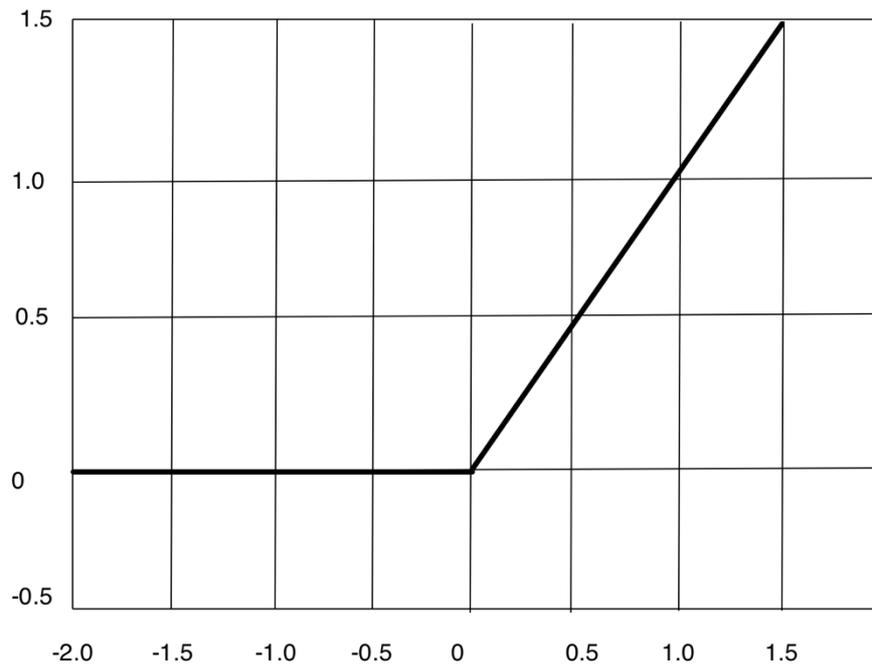
### 3.1.4 The rectified linear unit

The series of convolutional operations includes the pooling and Rectified Linear Unit (ReLU) operation between each subsequent convolution. The ReLU is an activation function that doesn't alter dimension of the layer as it maps activation values one to one. This layer is usually not illustrated in network architectures.

The introduction of ReLU activation function to neural networks is recent evolution. Saturating activation functions, such as sigmoid and tanh, were used in the earlier years. The replacement of these functions with ReLU has shown that, ReLU is advantageous with regard to accuracy and speed. Moreover, increased speed also has effect on accuracy as it enables deeper model to be trained for longer times (Aggarwal, 2018). The mathematical notation for ReLU is as follows:

$$ReLU(z) = \max(0, z) \quad (3.3)$$

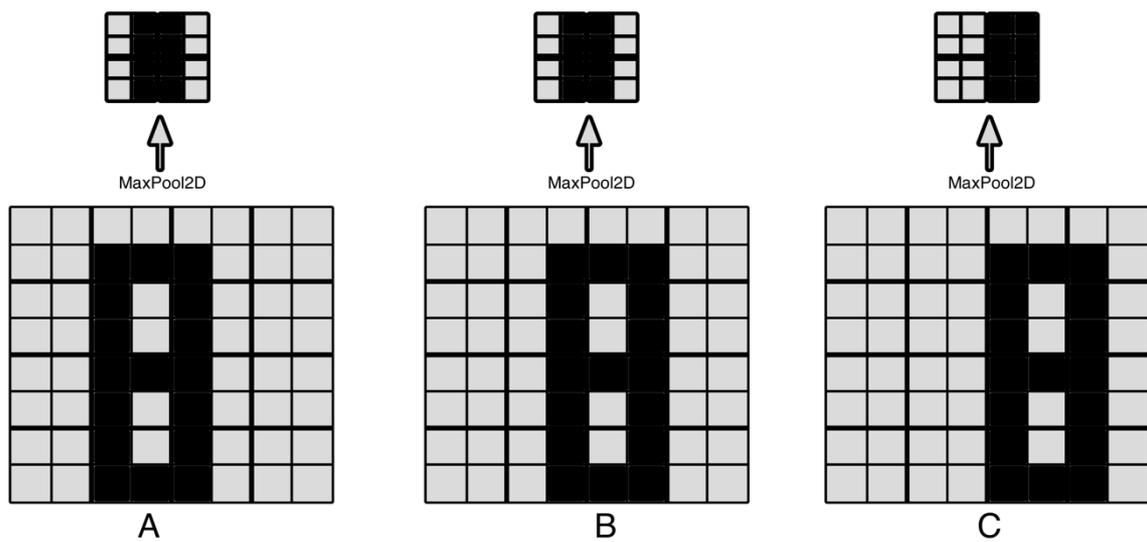
where  $z$  refers input to the activation function and output of ReLU function is illustrated in Figure 3.8.



**Figure 3.8:** ReLU output (Chollet, 2018)

### 3.1.5 Pooling

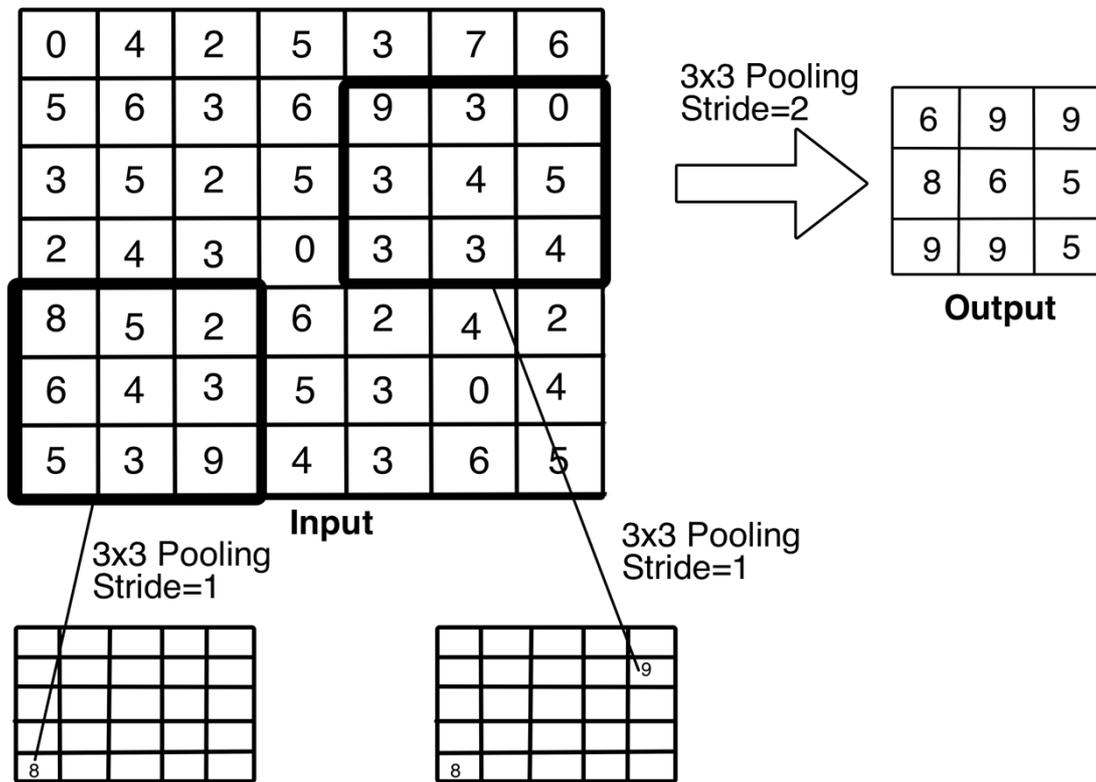
In each layer, the pooling process operates on small grid regions, and creates another layer with the same depth. The maximum of these values is returned for each square region in each of the activation map depths. This technique is known as max-pooling. If a stride of 1 is used a new layer of size will be created. However, the use of stride  $S > 1$  in pooling is more common. So that, pooling significantly decreases each activation map's spatial dimensions. An example of  $2 \times 2$  pooling size with 2 strides applied to 3 images as they are translated little compared to each other with their corresponding pooling result is illustrated in Figure 3.9. In Figure 3.10, examples of pooling with strides 1 and 2 are shown. Here, pooling over  $3 \times 3$  region is used. However, typical pooling size is  $2 \times 2$ . It is common to use stride of 2 where there would be no overlap of regions being pooled (Aggarwal, 2018).



**Figure 3.9:** Result of 2x2 pooling with stride = 2 applied to similar images (Géron, 2019)

### 3.1.6 Gradient descent

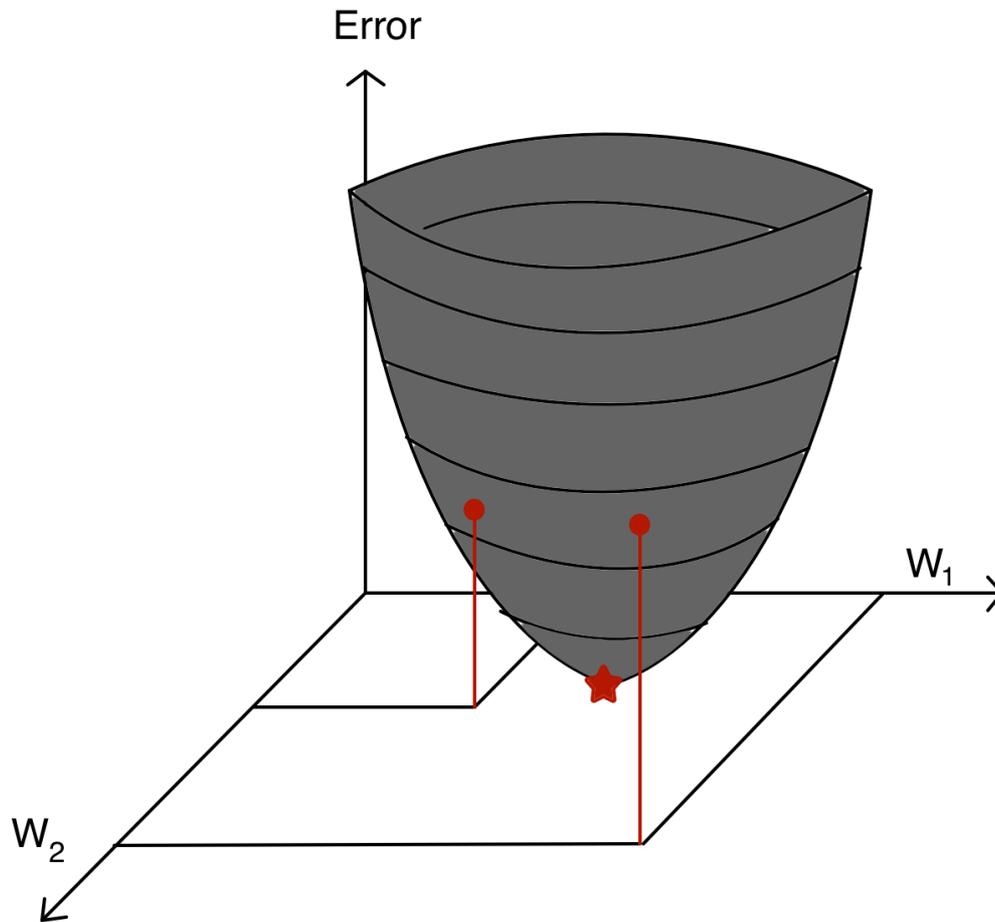
The ultimate purpose of machine learning algorithms is creating a model that represents certain beliefs that require a design of a cost function that can minimize the difference between reality and belief with the use of a training algorithm (Goodfellow et al., 2016).



**Figure 3.10:** Dimension reduction as result of max-pooling with stride 1 and 2 (Aggarwal, 2018)

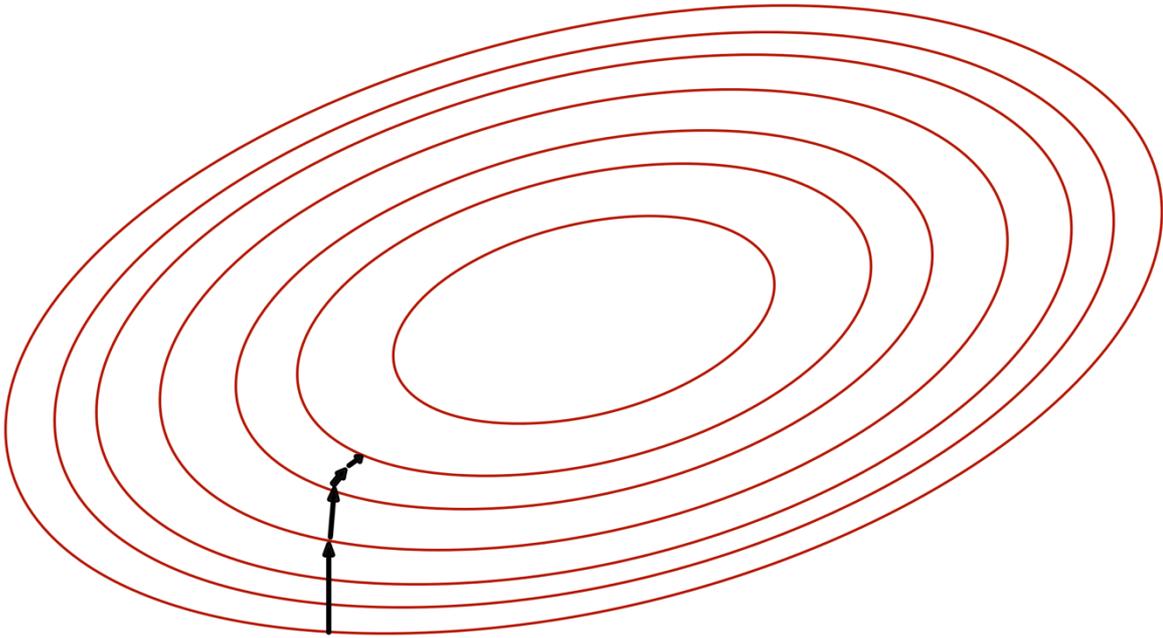
The value, the cost function outputs is the error. In stochastic gradient descent (SGD), not all data is used to compute gradient like traditional gradient descent, mini batches of samples are used.

In order to visualize minimization of the squared error over all of the training samples, a simplified illustration with a linear neuron with 2 inputs and two corresponding weights can be considered. This can be illustrated in a three-dimensional space as vertical dimension represents error and horizontal dimension represents the weights (Buduma and Locascio, 2017). The weights are corrected considering the error, surface of three-dimensional space is reached for given quadratic bowl illustrated in Figure 3.11.



**Figure 3.11:** Minimization of error with respect to weights (Buduma and Locascio, 2017)

Another visual presentation can be made with elliptical contours as illustrated in Figure 3.12. The center of the ellipses visualizes the minimum error. In this two-dimensional plane, dimensions correspond to weights ( $w_1, w_2$ ). Contours refer to  $w_1, w_2$  that measure the same error value. The steepness of slope increase depending on closeness of the contours and the direction of the steepest descent is always perpendicular to the contours. This vector expression is known as the gradient (Buduma and Locascio, 2017).



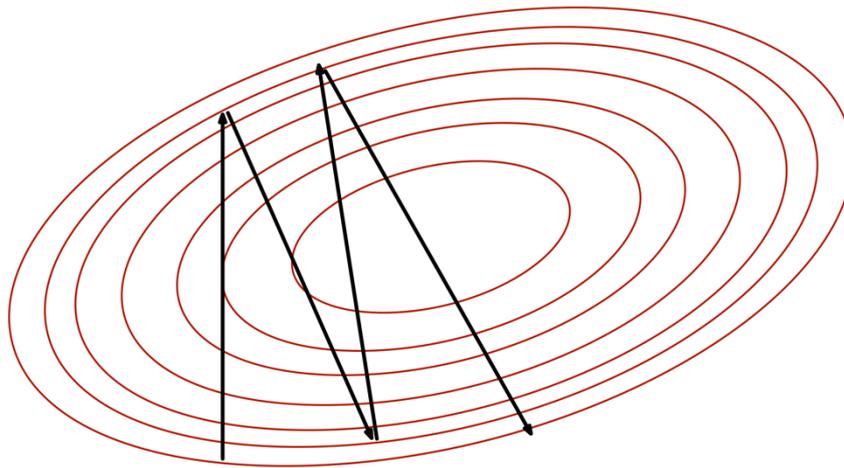
**Figure 3.12:** Error surface presented with a set of contours (Buduma and Locascio, 2017)

Let's say weights are randomly initialized and error-weight relation is considered on the horizontal plane. Evaluation of gradient for current position of weights in horizontal plane allows finding position of the steepest descent and a step is taken in that direction. As a result, new position will be closer to be minimum error than before. Now, direction of the steepest descent can be reevaluated for new position and the step can be taken in that direction. This strategy will eventually minimize the error as it is illustrated in Figure 3.12. This method is defined as gradient descent (Buduma and Locascio, 2017). However, computation of gradient can be very slow in case of large number of training samples (inputs). Therefore, stochastic gradient descent method which picks randomly chosen small number of samples (mini-batch) for training is introduced. The gradient is computed for those small number of samples each time, which provides a good estimate and quickens the learning (Nielsen, 2015).

### 3.1.7 Learning Rate

Once direction of steepest descent is calculated by the gradient, the step size in that direction needs be determined before recalculating the new direction. In that case, the step size needs to depend on the steepness of the surface. That could be explained by that as new location is

closer to the minimum, the surface is flatter and step size should be shorter. Therefore, the steepness can be used to determine the closeness to the minimum. Nevertheless, in case of a mellow error surface, the time training will take can be potentially large. So, the gradient is multiplied by the learning rate. The learning rate needs to be picked carefully. If the learning rate is picked as a too small number, there is a chance that it will take very long time to train. And if it is too big, there will be possible divergence from the minimum as it is illustrated in Figure 3.13 (Buduma and Locascio, 2017).



**Figure 3.13:** Difficulty of convergence with large learning rate (Buduma and Locascio, 2017)

### 3.1.8 Back-propagation

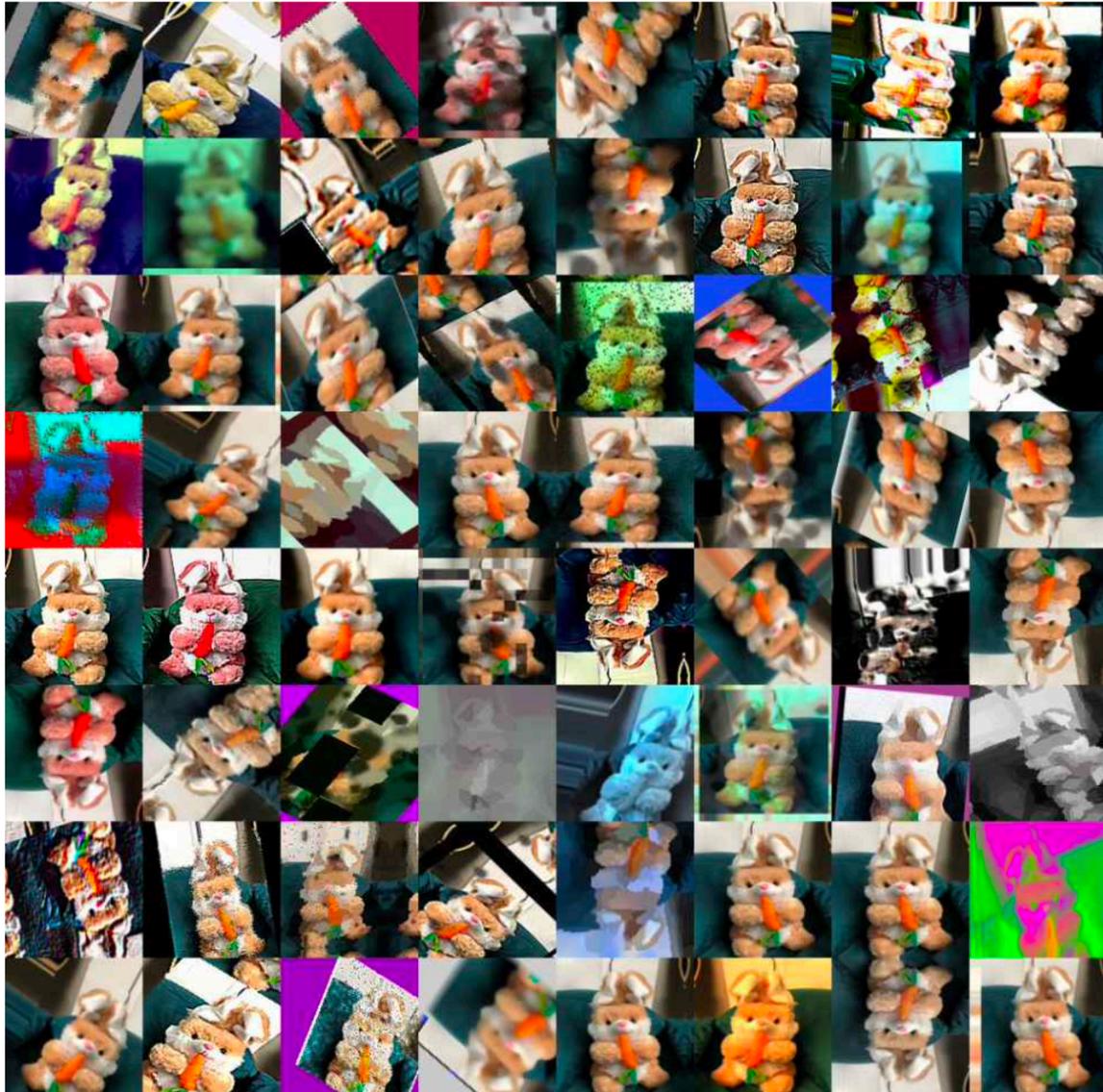
When the input  $x$  is accepted to the feed-forward network that generates output  $\hat{y}$ , information is propagated through the network. The output  $\hat{y}$ , is produced as input  $x$  delivers the initial information that flows through hidden units in each layer up to the output. This process is defined as forward propagation. Forward propagation will continue during training until a scalar cost is generated. In order to compute the gradient, the back-propagation algorithm enables the cost information to propagate backwards over the network (Goodfellow et al., 2016).

### **3.1.9 Optimizers**

Optimizers controls reevaluation of parameters depending on loss function. As a result, specific variant of SGD is applied (Chollet, 2018). The learning rate is recognized as one of most difficult hyperparameters to determine. This results from the fact that, it has important role in model efficiency. In parameter space, the cost is always extremely sensitive to certain directions and indifferent to others. This sensitivity can be reduced with the use of momentum algorithms, but with the expense of another hyperparameter introduced. In this case, a question is asked if it is possible to use a separate learning rate for each parameter and these learning rates are adapted throughout the process of learning (Goodfellow et al., 2016). There are several learning rates introduced such as AdaGrad, RMSProp and Adam.

### **3.1.10 Data Augmentation**

In case of limited data, data augmentation can be useful. The data at hand can be used to increase the sample size with help of data augmentation techniques. Multiple versions of each sample can be created with use of translation, cropping, rotation, scaling, noise addition or with change of color balance and brightness. The image label shouldn't change if these changes are small. And the training of the network with such augmented data will make the model robust (Russell and Norvig, 2020). The data augmentation is also known has having regularization effect on the model (Burkov, 2019). Possible augmentations of images created from a single image are illustrated in Figure 3.14.

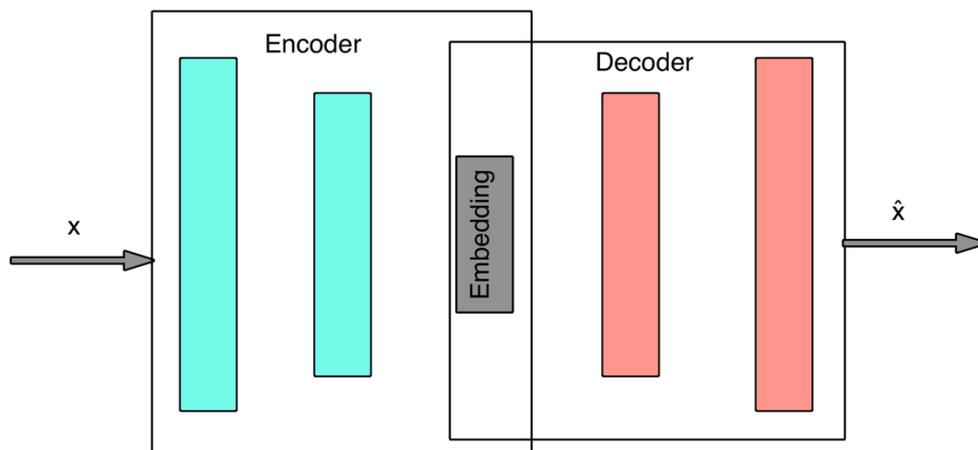


**Figure 3.14:** Data augmentation for generation of images from a single image (Koul et al, 2020)

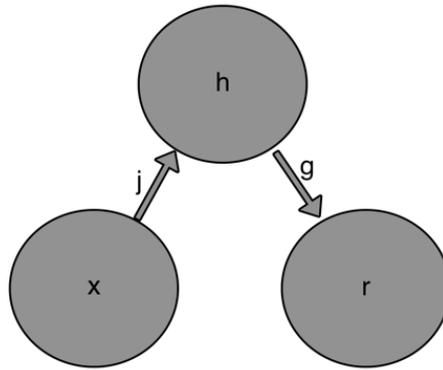
### 3.1.11 Autoencoders

A neural network that is equipped to attempt to copy its input to its output is an autoencoder. It has a hidden layer internally that represents a code used to represent the input. The network can be interpreted as consisting of two components: an encoder function and a reconstruction generating decoder. The intention behind the design of autoencoders is to restrict them in a way that they can't completely copy. The restriction only allows them to copy with approximation where it depends on input similar to training data. So, the model is required

to prioritize useful elements of the input to be learned (Goodfellow et al., 2016). A significant detail here is that the network of an autoencoder looks like a hourglass with a bottleneck layer in the middle that holds embedding of the D-dimensional input vector. Typically, the embedding layer has far fewer units than D. Here, the decoder's goal is to reconstruct the input feature vector using this embedding (Burkov, 2019). The general structure of auto encoder is illustrated in Figure 3.15. It is safe to say that, we can mathematically express encoder as  $h = f(x)$  and decoder as  $r = g(h)$  which is illustrated in Figure 3.16. The U-net architecture that is employed in this thesis is a convolutional autoencoder with skip connections between decoder and encoder (Karimov et al., 2019). The U-net architecture employed in this thesis is further explained in chapter 4.



**Figure 3.15:** Structure of the autoencoders (Burkov, 2019)



**Figure 3.16:** Input-output relationship of autoencoders as  $x$  input,  $h$  hidden layer,  $r$  output (Goodfellow et al., 2016).

### 3.2. Image Processing Background

Classic image processing algorithms utilize handcrafted filters and models to segment images unlike CNN architectures. In other words, the models that are created don't learn by themselves. The fundamental approaches such as filters, and segmentation models are discussed in the following parts.

#### 3.2.1 Spatial filtering

Spatial filtering requires a filter kernel which has a predefined size and coefficients. The size determines the operation of the neighborhood and coefficients defines the characteristics of the filter. A sum of products of an image with a filter kernel is performed by a linear spatial linear filter. An example of a kernel and its produced output is illustrated in Figure 3.17. If we are to choose a point  $(x,y)$  in the image, the output will be  $g(x,y)$  which will depend on sum of products between image pixels and their corresponding filter coefficients. That is mathematically denoted as:

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots (3.4) \\ + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

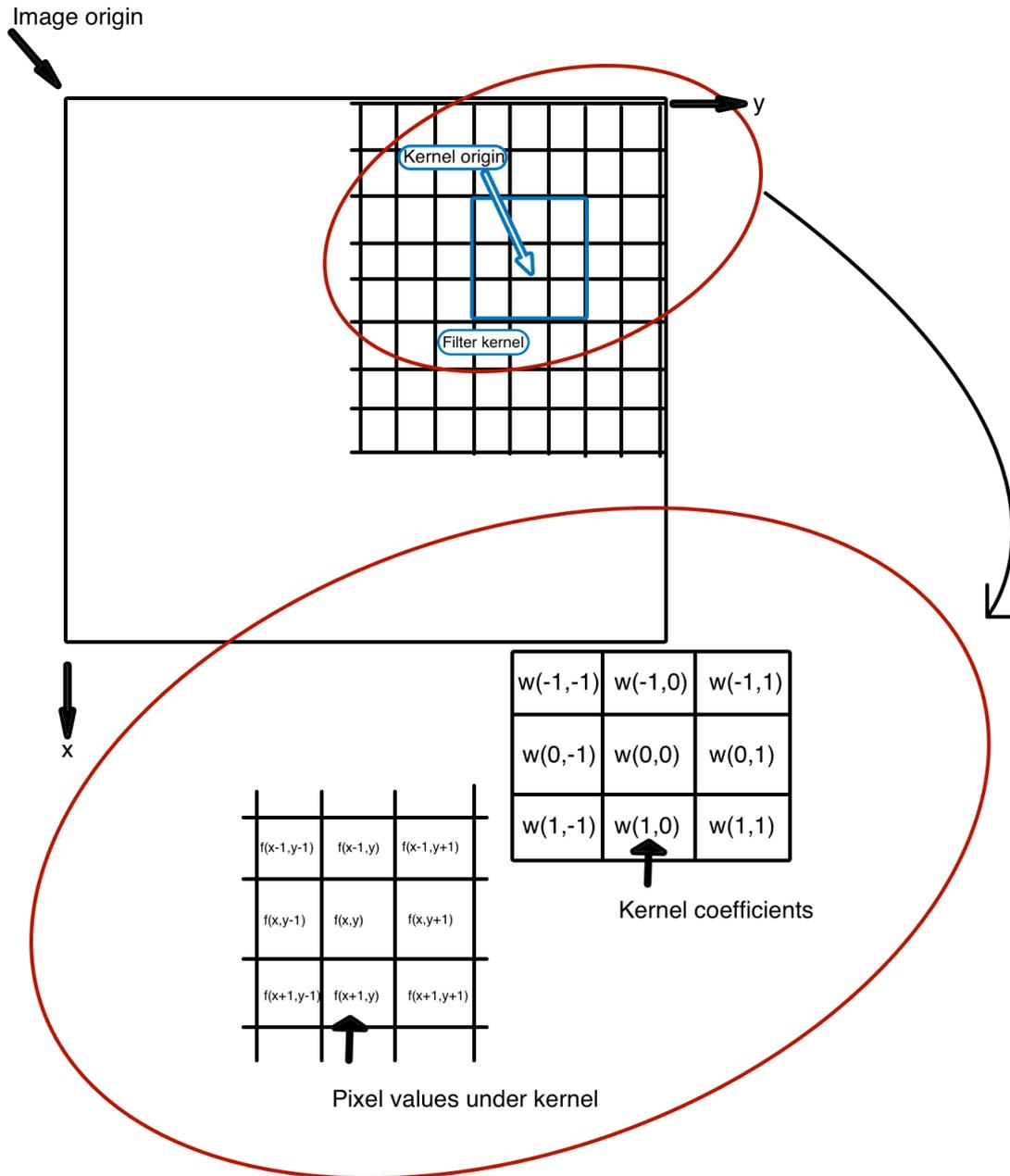
$w$  refers to mask coefficients and  $f$  refers to corresponding image pixels (Gonzalez and Woods, 2018).

### **3.2.2 Averaging filters**

To minimize sharp transitions in intensity, averaging (smoothing) spatial filters are used. Since random noise usually consists of sharp transitions in intensity, noise reduction is an apparent application of smoothing. The smoothing kernel is convolved over the image to blur the image where size of the kernel and coefficients determines the level of blurring. Gaussian and box filter kernels are two kernels that are used for smoothing. The box filter has same filter coefficient everywhere which is usually equals to “1” and Gaussian filter kernels are circularly symmetric (Gonzalez and Woods, 2018). These two filter kernels are illustrated in Figure 3.18.

### **3.2.3 Filtering in frequency domain**

The use of Fourier transform allows transition from spatial to frequency domain and inverse Fourier transform allows returning back to spatial domain. An image can be represented by sum sinusoids of different frequencies and amplitudes. In that way, the frequencies of image’s sinusoidal components determine the image’s appearance. Any change on these frequencies will result in change in the image (Gonzalez and Woods, 2018).



**Figure 3.17:** A filter kernel with its coefficients applied to the corresponding pixels in the image (Gonzalez and Woods, 2018).

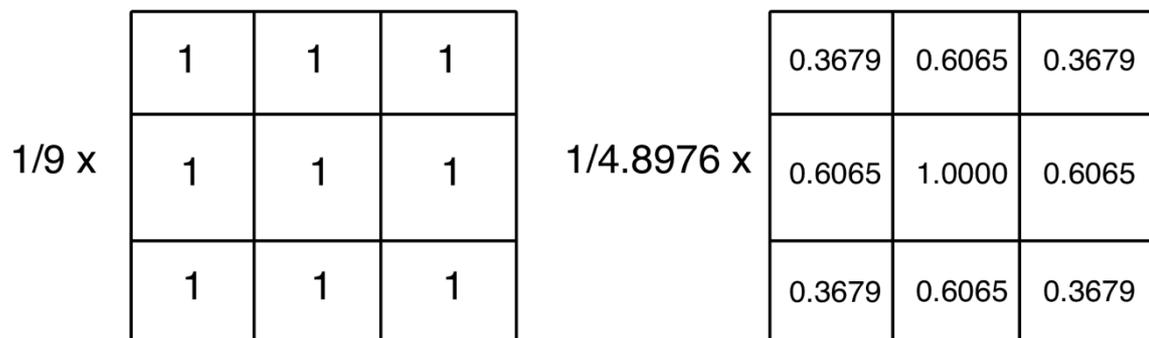
The sharp changes are considered as high frequencies whereas slow changes are low frequencies. Therefore, reducing an image's high-frequency components would appear to blur it. The low pass filter design is illustrated in Figure 3.19 where  $u_o$  determines which frequencies below certain level will be included in the image (Gonzalez and Woods, 2018).

### 3.2.4 Median filter

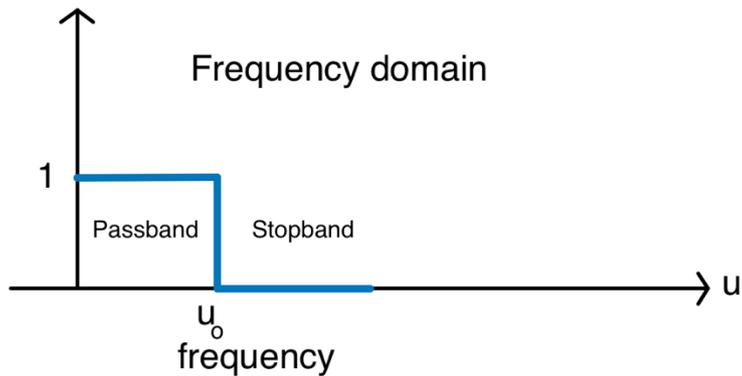
Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter. Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result. The best-known filter in this category is the median filter, which, as its name implies, replaces the value of the center pixel by the median of the intensity values in the neighborhood of that pixel (Gonzalez and Woods, 2018).

### 3.2.5 Sharpening filters

Spatial differentiation is used to perform sharpening. The power of a derivative operator's response is proportional to the magnitude of the discontinuity of the intensity at the point at which it is applied. Image differentiation thus improves edges and other discontinuities (such as noise) and de-emphasizes areas with intensities that vary slowly (Gonzalez and Woods, 2018).



**Figure 3.18:** Filter coefficients of box filter kernel on left and Gaussian filter kernel on right (Gonzalez and Woods, 2018)



**Figure 3.19:** Low pass filter (Gonzalez and Woods, 2018)

### 3.2.6 Laplacian filter

Second-order derivatives can be used for image sharpening. The discrete formulation of the second-order derivative is defined and a filter kernel is constructed by using that formulation (Gonzalez and Woods, 2018). An example of Laplacian filter is illustrated in Figure 3.20.

0	1	0
1	-4	1
0	1	0

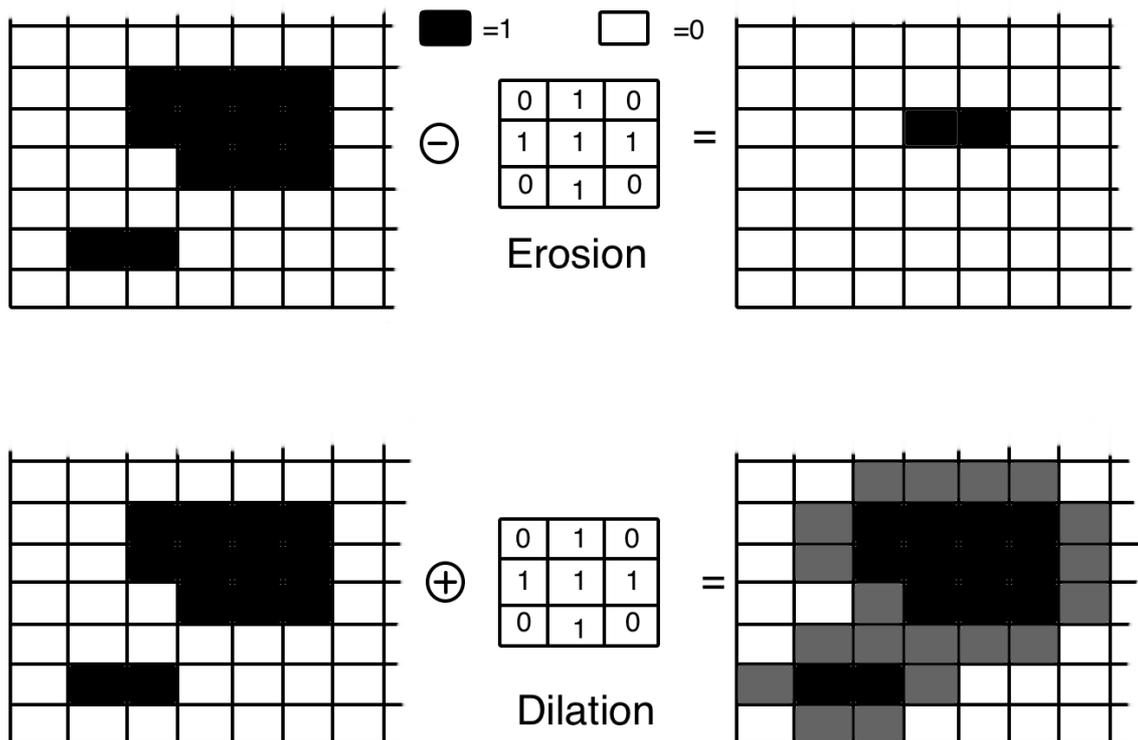
**Figure 3.20:** An example of Laplacian filter (Gonzalez and Woods, 2018)

### 3.2.7 Morphological operators

The morphological operators are primarily employed for processing binary images. The fundamental operators are dilation and erosion. The complicated morphological techniques can be explained with series of dilations and erosions (Solomon and Breckon, 2011).

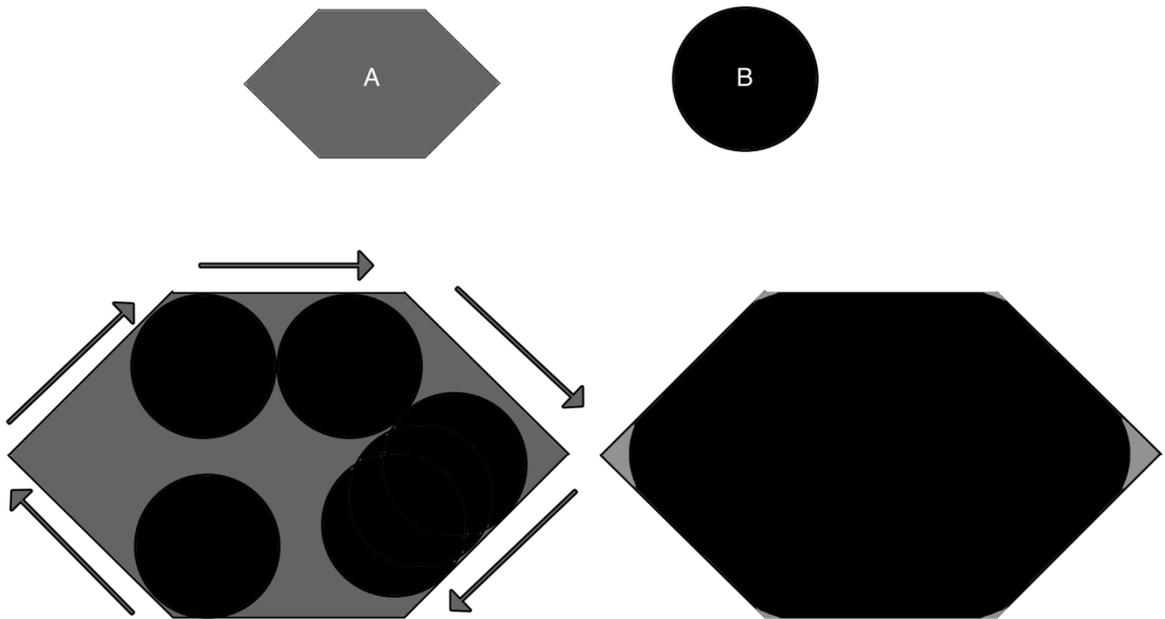
### 3.2.8 Dilation and erosion

The dilation and erosion mechanics work in a very similar manner to the convolution kernels used in spatial filtering. The structuring element moves over the image where its center pixel is placed successively on top of each front or background pixel. Let's refer to an image as  $A$  and structuring element as  $B$ . For erosion, center of structuring elements is placed on foreground pixel, and presence of any background pixel in the neighborhood would change the foreground pixel to background pixel. In case of dilation, structuring element is placed on background pixel, and presence of any foreground pixel in the neighborhood would change the background pixel to foreground pixel. Erosion and dilation are denoted by  $A \ominus B$  and  $A \oplus B$  respectively. The new value of each pixel of the image then relies on the neighbourhood pixel values identified by the structuring element (Solomon and Breckon, 2011). An example of dilation and erosion on binary image is illustrated in Figure 3.21.

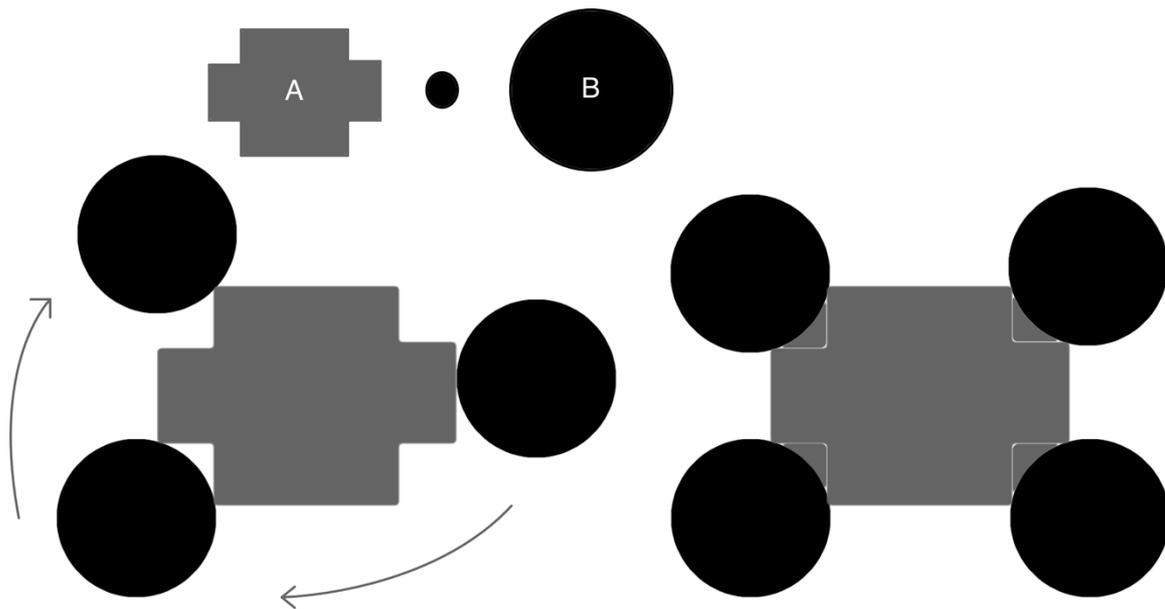


**Figure 3.21:** Output of structuring element interaction with binary image where erosion on top and dilation at bottom (Solomon and Breckon, 2011)

Morphological opening and closing can be expressed with erosions and dilations. The morphological opening is erosion followed by dilation whereas morphological closing is dilation followed by erosion by using same structuring element during the operation (Solomon and Breckon, 2011). An example of opening and closing is illustrated in Figure 3.22 and Figure 3.23 respectively.



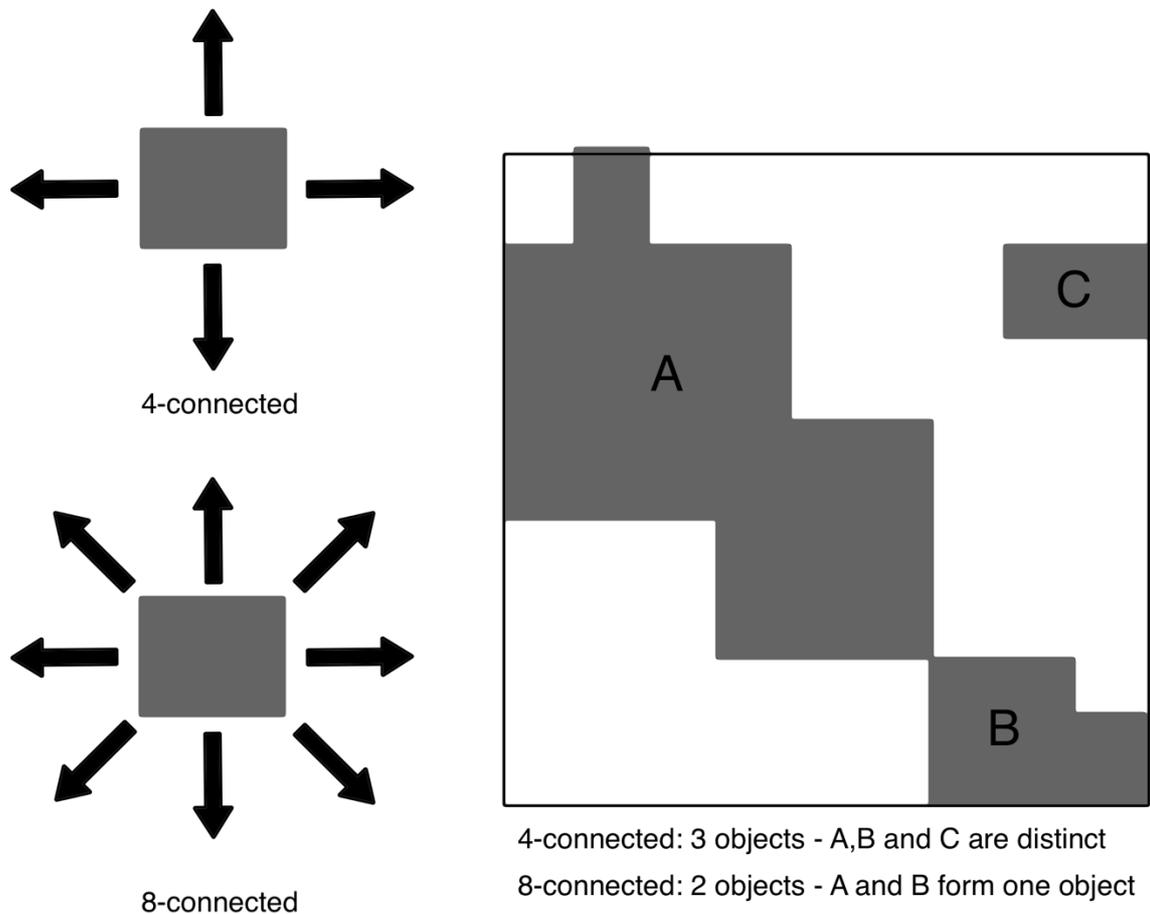
**Figure 3.22:** The structuring element is visualized as a ball rolling in the image and places it touches are resulting output for morphological opening (Solomon and Breckon, 2011)



**Figure 3.23:** The structuring element is visualized as a ball rolling at outer boundary of image and the contours it draws are resulting output for morphological closing (Solomon and Breckon, 2011)

### 3.2.9 Connectivity

The binary image consists of values of 1 and 0. The pixels with value of 1 are defined as foreground pixels and whereas 0 as background. An object in a binary image can be expressed with number of connected pixels. In case of foreground pixel having at least one neighbouring pixel in horizontal or vertical directions, it is said to be 4 connected. And if, there is a neighbouring pixel (foreground) in a diagonal direction, then it is sufficient to define it as 8-connected (Solomon and Breckon, 2011). The directions of connections for 4-connected and 8 connected is illustrated in Figure 3.24.



**Figure 3.24:** Direction of 4 and 8-connected pixels (Solomon and Breckon, 2011)

### 3.2.10 Threshold segmentation

A commonly used segmentation technique is image threshold segmentation. First, it compares the variations in the gray level between the target regions and the background regions and then chooses a more appropriate threshold for the object pixels and background pixels to be calculated. Finally, it produces a binary image corresponding to it (Gong et al., 2018).

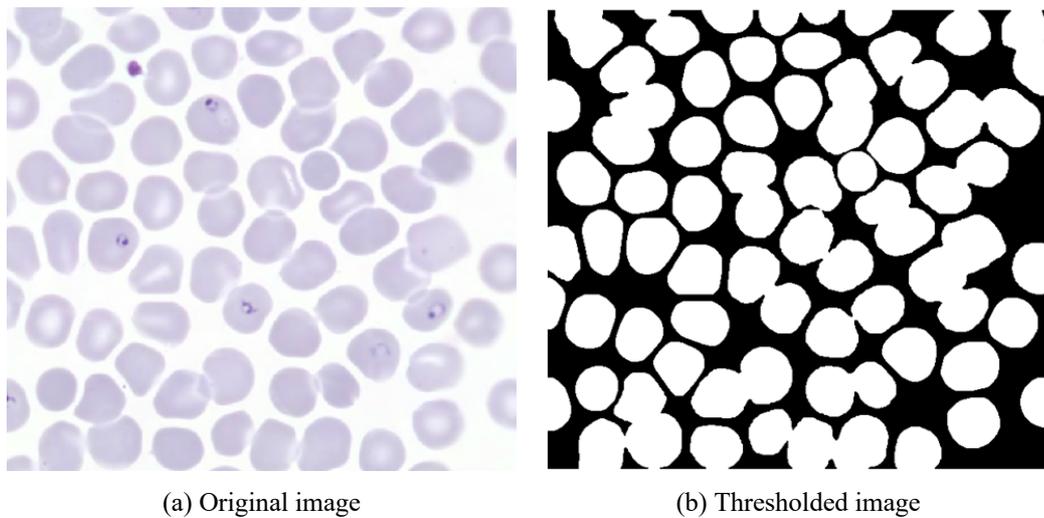
Let's say the image is  $f(x,y)$  and eigenvalue  $t$  is found with a certain criteria that classifies pixels into two group and black ( $b_0=0$ ) and white ( $b_1=1$ ). The mathematical formulation is defined as:

$$g(x,y) = \begin{cases} b_0 & f(x,y) < t \\ b_1 & f(x,y) \geq t \end{cases} \quad (3.5)$$

Where  $g(x,y)$  refers to output as result of particular threshold  $t$  (Gong et al., 2018).

### 3.2.11 Global thresholding

Global thresholding can produce satisfactory results in case of bimodal histogram distribution and relatively simple background. Histogram thresholding, maximum entropy thresholding, interclass difference thresholding are some of methods that are employed for global thresholding (Gong et al., 2018). An example of global thresholding is illustrated in Figure 3.25.



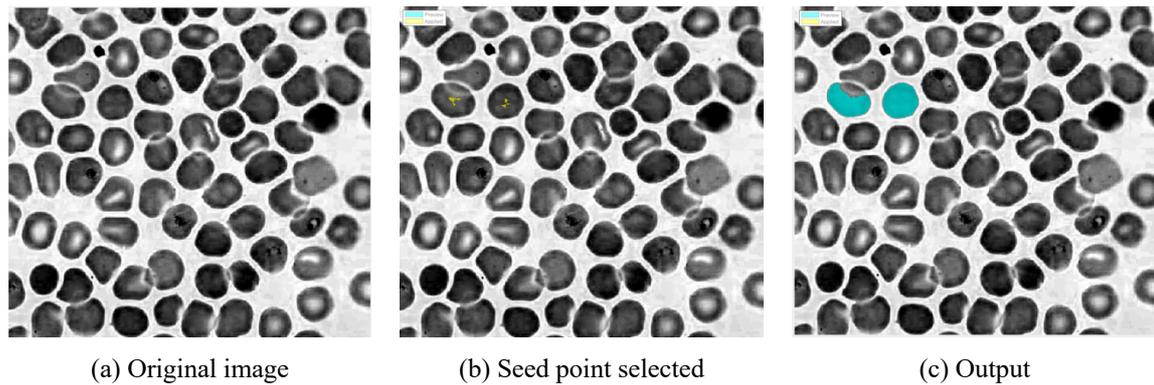
**Figure 3.25:** Application of thresholding

### 3.2.12 Local dynamic thresholding

When image brightness is non-uniform, local dynamic thresholding is used for normalization of the brightness. To perform this image is divided into sub-image and threshold is selected for that particular sub-image (Gong et al., 2018).

### 3.2.13 Region growing

Combining the pixels with similar properties to form a region is the fundamental idea of region growing. Initially, find a seed pixel for each growth area as the starting point. And later combine the pixels with the same or similar properties of the seed in the neighborhood. As the new pixels continue to seed to grow around them while the pixels meet the conditions, a region grows (Gong et al., 2018). An example of region growing is illustrated in Figure 3.26.



**Figure 3.26:** Region growing steps

### 3.2.14 Clustering

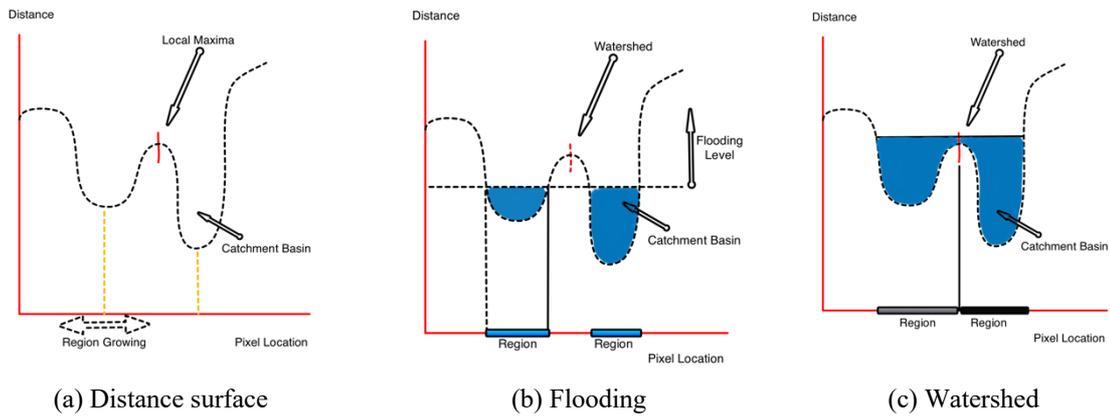
Clustering is the process in which things are distinguished and classified according to certain requirements and rules. Often, the number of clusters and the initial clustering centers need to be given. The clustering method of image segmentation is to represent the pixels in the image space with the corresponding feature space points. The segmented outcomes are obtained based on their clustering in the feature space to segment the feature space and then map them back to the original image space. K-means clustering is one of widely used clustering techniques. First, the K-means algorithm selects the initial mean class of K, and then each pixel selects the average of its nearest class and calculates the new mean class. Iterate the previous steps until there is less than a threshold between the average of the old and new classes (Gong et al., 2018).

### 3.2.15 Watershed transformation

The method of the watershed derives from morphological operations, and by growing regions from a set of initial seeds, it divides the picture into many parts. The main concept is to use labels in an iterative process to aggregate pixels. There is a different label on the pixels in each area. Pixels not aggregated into any area are classified according to the labels of their neighbors during each iteration. As a result, connected components grow and it is possible to delineate edges that restrict the regions (Nixon and Aguado, 2019).

Using flood analogy, the watershed method can be divided into three main steps. The first step is computation of a distance or property for each pixel where the property expresses the

organization of pixels and defines the corresponding topographic surface. The second step follows as, finding local minima for representation of initial region. Finally, at the third step the topographic surface is flooded by labelling pixels and delineating edges (Nixon and Aguado, 2019). These steps are illustrated in Figure 3.27.



**Figure 3.27:** Watershed transformation steps (Nixon and Aguado, 2019)

## **CHAPTER 4**

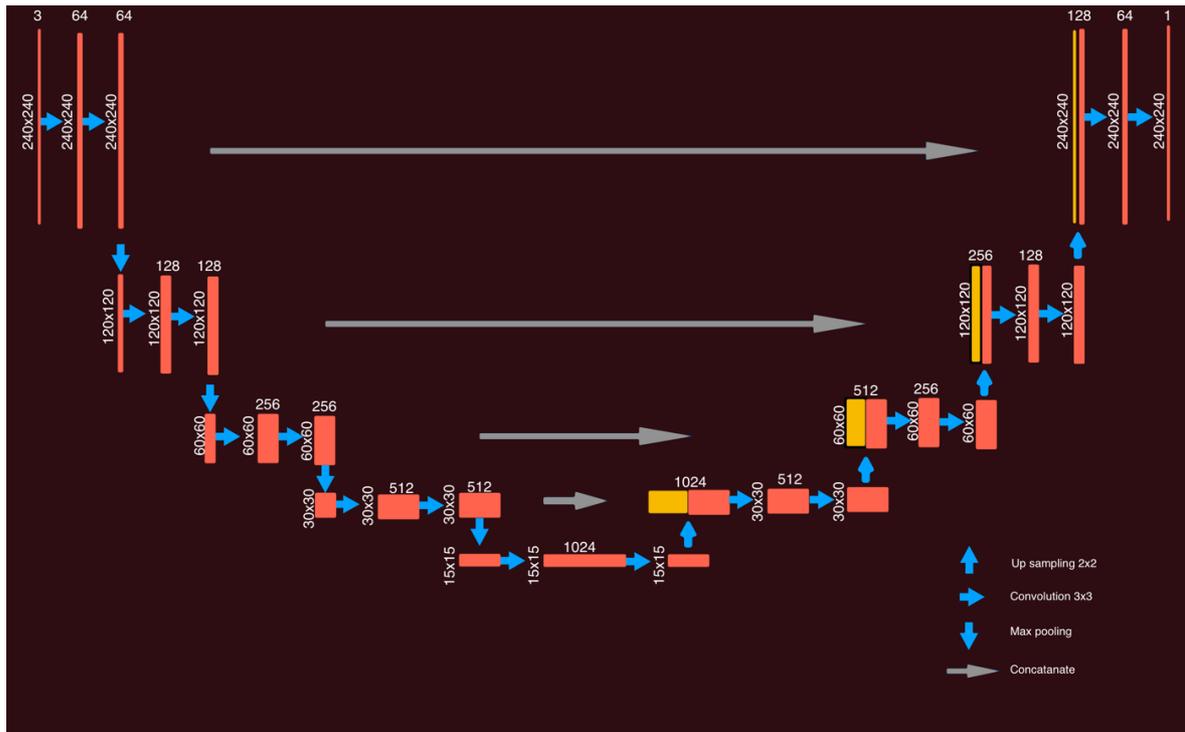
### **MATERIAL AND METHOD**

Dataset of malaria images was acquired from Mamic image database (Mamic image database, n.d). Randomly picked 100 images with 100x zoomed were divided into 4 pieces. Images will be used to train deep learning network architecture therefore, divided was preferred instead of resizing as the number of images will increase consequently. As a result, 400 images were gathered. The ground truth data for corresponding images were created using Matlab 2019a image labeller app and pixel by pixel labelling was applied for each image in the dataset. The U-net architecture was constructed in google colab environment and images were stored in google drive. The images were divided by 70%, 15%, 15% for training, validation, testing respectively. However, a large number of samples are required to train deep neural networks, and pixel by pixel labelling is required in the case. Therefore, labelling of a large number of images is not easily available which leads us to data augmentation techniques. Random elastic deformation was used to increase sample size mainly, as it was applied 10 times to the training validation and testing set. Vertical and horizontal flips were used for training and validation only. To summarize, 5600, 1200, 600 images were created for training, validation, and testing respectively. The purpose of the proposed network is to learn to segment red blood cells while separating overlapping red blood cells and removing irrelevant objects such as white blood cells and platelets. For that purpose, intersection points of overlapping red blood cells along with irrelevant objects were marked as background in order to train the network to learn segment and separate red blood cells.

#### **4.1 Network's Architecture**

The U-net architecture has 2 parts. The first part consists of consecutive convolutional layers with multi-channel feature maps followed by pooling operator and each convolutional layer is followed by a rectified linear unit (ReLU). After each max-pooling operation (downsampling) number of feature maps are doubled. The first part is defined as contracting path. At the second part, upsampling operators replaces pooling operators and after each upsampling number of feature maps are halved. The second part is defined as expanding path. As expanding path increases resolution gradually it takes high resolution features from

contracting path by concatenation (Ronneberger et al., 2015). The target (output) is a binary mask for corresponding input where foreground pixels have value of 1 and background pixels 0. The visual representation of the U-net architecture is shown in Figure 4.1. The images used in this study was in RGB format in an attempt to help network to discriminate red blood cells from white blood cells.



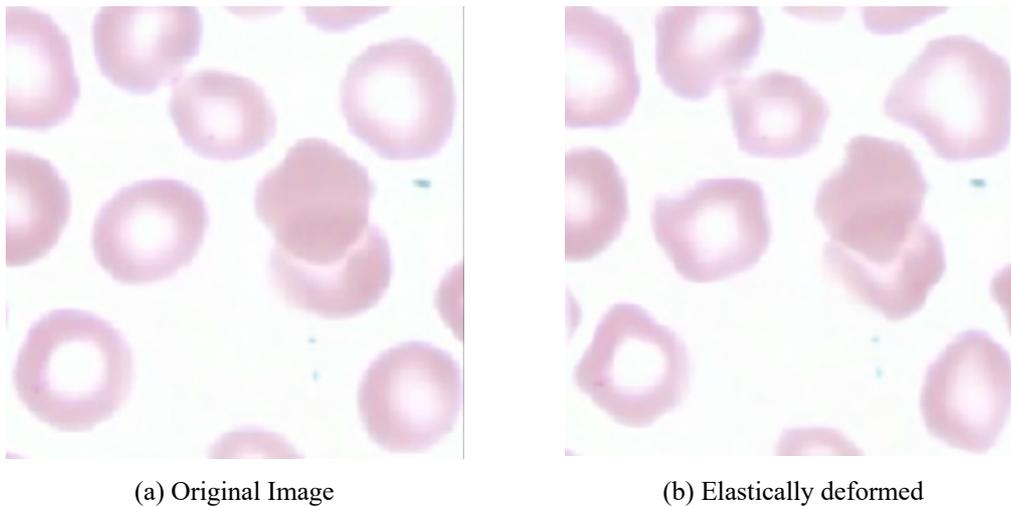
**Figure 4.1:** The U-net architecture

#### 4.1.2 Data augmentation

Data augmentation is an essential step to increase sample size and provide variation for data. In this study, random elastic deformation, horizontal and vertical flips were employed for data augmentation. Deformability is a significant characteristic of RBCs that allows them to pass through even the human body's smallest capillaries (Huisjes et al., 2018). Therefore, application of elastic deformation for data augmentation is suitable data augmentation technique for red blood cells.

Random displacement fields are used to create elastic deformation where these fields are convolved with Gaussian standard deviation  $\sigma$ . Value of  $\sigma$  determines deformities where large  $\sigma$  result in very small deformities and small  $\sigma$  result in too random deformities. The

small value of  $\sigma$  in case of deformation of red blood cells, produce pseudopods around cells that image no longer represents red blood cells. The intensity of deformation is controlled by a scaling factor  $\alpha$ . The scaling factor is multiplied with displacement fields to achieve this operation. These two parameters,  $\sigma$ , and  $\alpha$  were adjusted as 8 and 1200 respectively. Each time, displacement fields are generated, they are generated randomly. As a result, it is required to apply generated deformation to input and target images at once. A total of 400 images were divided as 280, 60, 60 for training testing and validation respectively. Each generated elastic deformation applied to training testing and validation set produces 280, 60, and 60 images respectively. This process was repeated 10 times and as result, dissimilar 2800 training, 600 validation, and 600 testing images were created. After that, random vertical and horizontal flips were used to double the sample size of the training and validation set. As a result, 5600, 1200, and 600 images for training, validation, and testing were created respectively.



**Figure 4.2:** Random elastic deformation applied to a image

### 4.1.3 Optimizer

Color images in RGB format with dimension of 240 x 240 pixels were used in this study. 5600 and 1200 images were used for training and validation respectively for 3200 epochs with Adam learning rate of 0,001 as an optimizer.

#### 4.1.4 Evaluation Metric

Jaccard similarity index was employed as evaluation metric for pixel classification. The predicted image  $A$  and the ground truth image  $B$  was compared for similarity. The Jaccard similarity index is mathematically defined as intersection over union, which is noted as:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \quad (4.1)$$

where  $A$  indicates the predicted image and  $B$  indicates the ground truth. The discrete representations of pixels can be derived from this equation,

$$J = \frac{1}{n} \sum_{i=1}^n \left( \frac{y_i \hat{y}_i}{y_i + \hat{y}_i - y_i \hat{y}_i} \right), \quad (4.2)$$

where  $y_i$  refers to corresponding binary label for pixel  $i$  and  $\hat{y}_i$  refers to predicted probability for corresponding pixel  $i$ . Segmentation of any object into foreground and background is a pixel classification task, so that common loss function for pixel classification was employed which is defined as:

$$H = -\frac{1}{n} \sum_{i=1}^n (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \quad (4.3)$$

These expressions can be generalized for the loss function as

$$L = H - \log J, \quad (4.4)$$

where minimization of loss indicates maximization of the probability for right pixels to be predicted and maximization of intersection between binary labels (masks) and predicted pixels (Igloukov and Shvets, 2018).

#### 4.2 Local Minimum Histogram Segmentation with Selective Hole Filling Approach

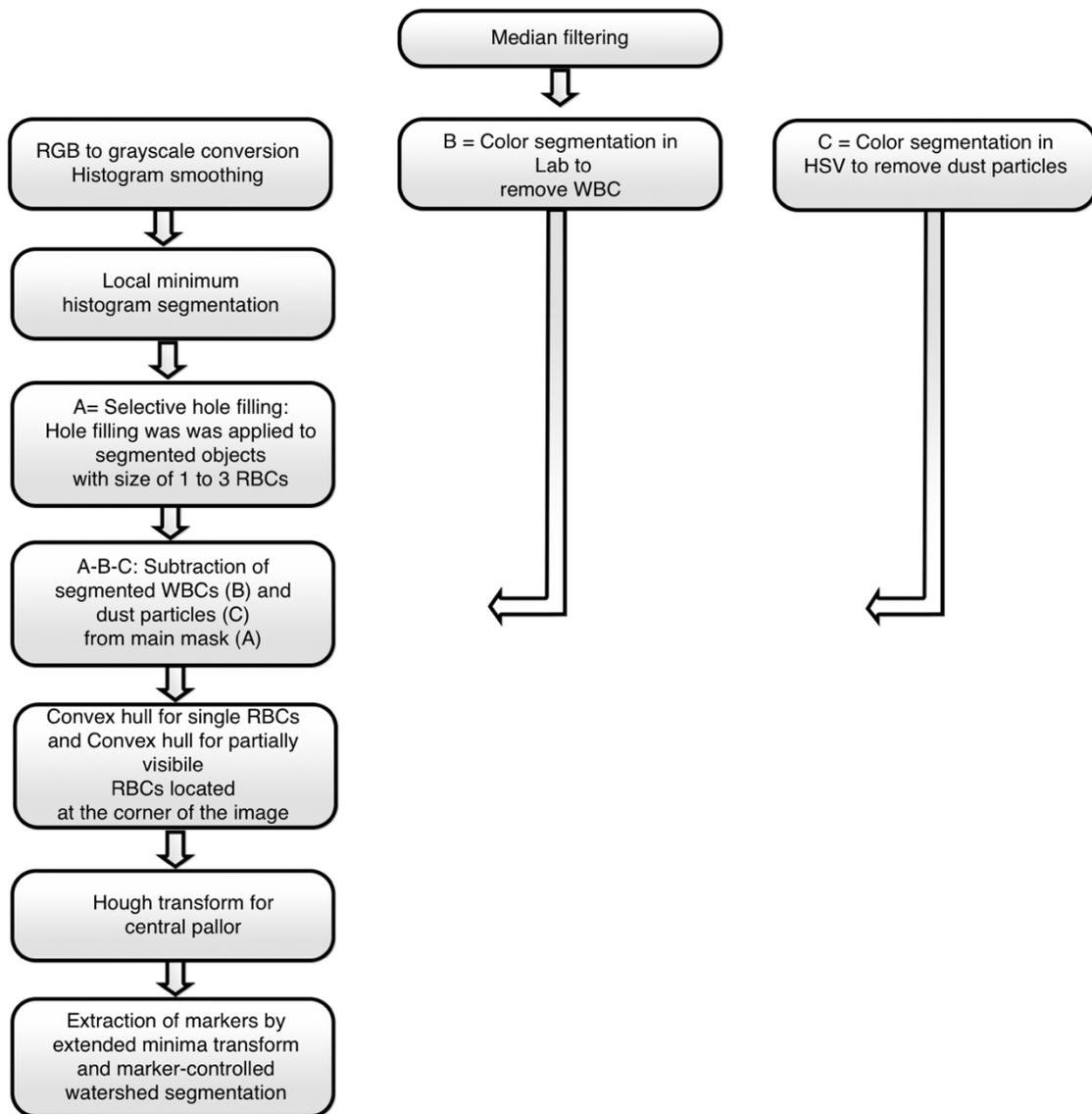
The samples involved 66 images with 100x magnification, where images consist of normal and infected red blood cells, white blood cells, platelets, dust particles, and artifacts. The images were randomly chosen among the dataset. The algorithm was modelled on these 66 images. However, a successful evaluation of the segmentation of overlapping red blood cells requires an abundance of overlapping samples. To this end, the algorithm was also tested on a different set of 15 images of overlapping red blood cell samples. Matlab 2019a platform

was used to perform the following tasks. Our algorithm consisted of ten main steps. The first three steps are the pre-processing steps including Median filtering, RGB to gray conversion, and histogram smoothing. This is followed by local minimum histogram background segmentation for RBC segmentation and HSV and Lab colour space for WBC and dust removal as the segmentation steps. The area filtering by connected components was used to limit hole filling which we named as selective hole filling. Morphological operators, selective refilling of holes was used in order to remove wastes. Convex hull was utilized for both single RBCs, and partially visible RBCs located at the corner of the image. Hough transform was used for circle detection to fill bright areas at the centre of some RBCs known as central pallor. As the final step, the extended minima transform was used to extract markers and marker-controlled watershed segmentation was carried out to count RBCs in an automated manner. However, all these steps could be simplified to fewer steps; including only RBCs segmentation, removal of WBCs, dust, and artifacts, separation of overlapping cells, and finally cell counting. The steps of the algorithm are illustrated in Figure 4.3.

#### **4.2.1 Preprocessing**

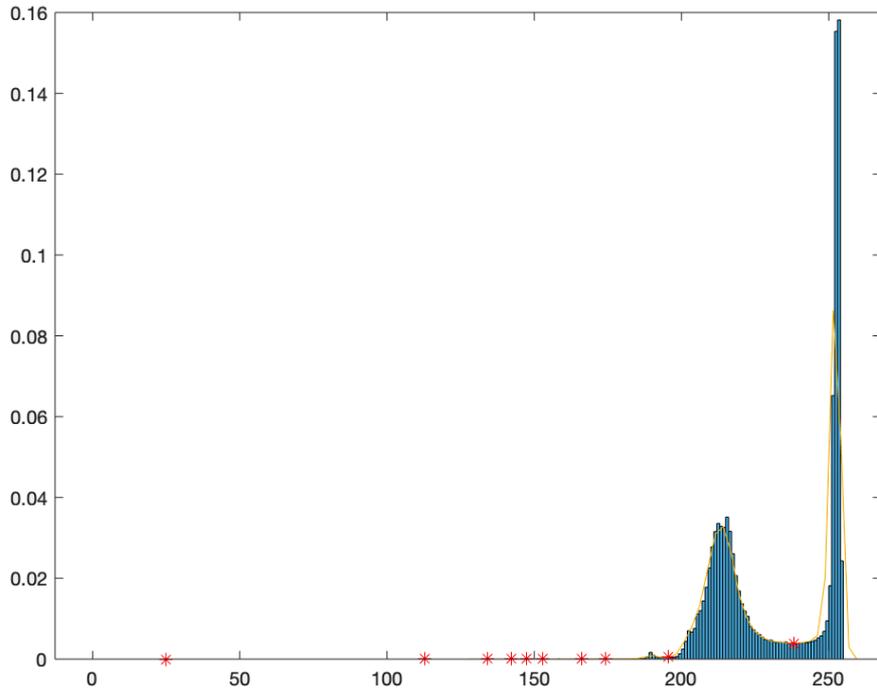
Median filtering was applied to red, green, and blue channels of input images to preserve the edges of red blood cells. There are lots of variations in colour due to different staining conditions. The grayscale transformation was employed to equalize this difference among the other images. Prior to local minimum histogram segmentation, histogram smoothing was required to avoid the selection of false peaks. Therefore, histogram smoothing was applied to the image. Pre-processing steps of white blood cells and dust particle segmentation only consisted of median filtering to RGB channels.

The first step of segmentation is the separation of background from the foreground. Segmentation of background is a crucial step in order to extract RBCs in an intact way and remove irrelevant data. In the histogram, two groups of pixels were observed where one group represents RBCs and the other group, the background. The algorithm was utilized to search for the local minimum between the highest two peaks in the smoothed histogram that is illustrated in Figure 4.4. The steps of finding the local minimum are as follows:



**Figure 4.3:** Flowchart of selective hole filling approach

- Find the highest two peaks in the histogram
- Set the first peak value (height of the peak)- 0.001 as the threshold for finding peaks in the next step
- Find all peaks with the threshold. This leads to the detection of only two peaks.
- Find local minimum (intensity value) after the first peak. In the case of many local minima, the first local minimum is chosen for the histogram thresholding point.



**Figure 4.4:** Local minima on smoothed histogram are marked with red stars. Smoothed histogram is illustrated with an orange sloping line on the histogram.

The mathematical formulation of local minima is defined as:

$$f(x) > f(x_1) \quad \forall x \in (x_1 - \delta, x_1 + \delta) \quad (4.5)$$

where  $x_1$  represents the local minimum on the x-axis whereas  $\delta$  denotes its neighbourhood. It is possible to have many local minima outside the range of the highest two peaks. However, the algorithm was designed to work in an explicit way to avoid false local minimum detection. Choice of first local minimum after the first of the highest two peaks allows intact RBCs segmentation. Figure 4.4. illustrates the presence of many local minima. The local minimum thresholded image resulted in a binary mask. Area filtering by connected components was applied to remove small irrelevant objects. The purpose of area filtering operations is to remove irrelevant objects and residual pixels that are not related to the segmented object. This method requires the analysis of objects in binary masks. It was found that anything less than 750 pixels is not a red blood cell. Therefore, this number was used to remove small irrelevant objects. The areas of objects were analysed through all images by Matlab image region analyser app.

Hole filling was selectively applied to segmented objects with areas having 1 to 3 RBCs, where marker-controlled watershed transformation is suitable for separating these RBCs clumps. This step is crucial for marker-controlled watershed transformation. However, for larger clumps it is almost impossible for watershed transformation to separate clumps without any clue of RBCs corners. These corners look dark among RBC clumps after local minimum histogram segmentation as they represent the corners of overlapping RBCs. Therefore, only 1 to 3 area sized RBCs were extracted by connected components, and hole filling was applied for those RBCs. The average area for triple overlaps was calculated as 9680 pixels. The up limit was determined by adding 10% to the average area and was approximated to 10750 pixels. The down limit was set by 55% of the smallest individual red blood cell area as 1000 pixels in order to include partially visible red blood cells as well. Matlab Image Region Analyzer application was used to analyse area of red blood cells. Overlapping cells that touch the corner of the image could be lost during the border cleaning operation. To this end, no border cleaning operation was applied. The selectively hole filled mask was combined with the post-processed local minimum histogram segmented mask via union set operation.

#### **4.2.2 Colour segmentation**

White blood cells and dust particles present differentiable colours compared to RBCs. Therefore, colour segmentation can be used to remove such irrelevant objects. White blood cells and dust particles present intense colour of purple and light brown to black respectively.

#### **4.2.3 White blood cell segmentation**

Literature shows the use of Hue Saturation Value (HSV) colour segmentation for the removal of WBCs (Li et al., 2016). The hue channel was manipulated in order segment WBCs. While the image consists of a white blood cell along with red blood cells, thresholding over HSV colour space can be useful. The image can be separated into white blood cells and background depending on the distribution of pixels.

However, in our case, where not all of the images exhibit white blood cells. Since the distribution of pixels can't be used to provide consistent segmentation over the dataset and Hue channel can't be used where the colour of white blood cells varies due to staining conditions over the dataset, colour dimensions were observed to present more

discrimination. Therefore, the image was converted into lab space to apply white blood cell segmentation, where a and b channels represent colour dimensions. Channel “a” refers to red/green and channel “b” refers to blue/yellow value. Post-processing of this segmentation includes the complement of the output image, followed by area filtering to remove residual objects in the WBC boundaries initially (less than 200 pixels). The complement of resulted output was taken for morphological closing with a disc-shaped structuring element that has a radius of 15 to smooth the shape of WBCs. Then, morphological dilation was applied with a disc-shaped structuring element with a radius of 7, which ensures the total segmentation of WBCs.

Area filtering was applied to leave only WBC sized elements and eliminate any objects other than white blood cells. White blood cells can be as small as 10  $\mu\text{m}$  in diameter where red blood cells are about 7  $\mu\text{m}$ . The largest red blood cells in our data are around 4000 pixels. Area filtering with a lower limit of 5000 pixels would be sufficient enough to filter other objects that share similar colour dimensions with white blood cells such as platelets and malaria parasites. Then, the resulted segmentation of WBCs was subtracted from the mask.

#### **4.2.4 Dust particles segmentation**

Blood smear images could be contaminated by dust particles. These dust particles might be present either in the blood film or on the camera or lens of the microscope. The colour of these dust particles varies from light brown to dark. The hue channel of HSV colour space can be used for their segmentation. The range of brown colour was set on the hue channel to allow the segmentation of dust particles. Morphological image closing with a disc-shaped structuring element with a radius of 15 was applied to smooth the edges of dust particles followed by area filtering for very small objects. Area filtering was applied to remove small-sized unrelated pixels (less than 200 pixels) which were taken as dust particles. Finally, resulted segmentation of dust particles was subtracted from the mask and colour segmentation was completed.

#### **4.2.5 Refilling of holes**

The selective hole filling operation was applied considering the same area parameters with union set operation, that was previously applied after local minimum histogram segmentation. Prior to convex hull step, image opening with disk-shaped structuring element

with a radius of 5 was used in order to prevent the fusion of neighbouring RBCs and platelets in the convex hull operation.

#### 4.2.6 Convex hull

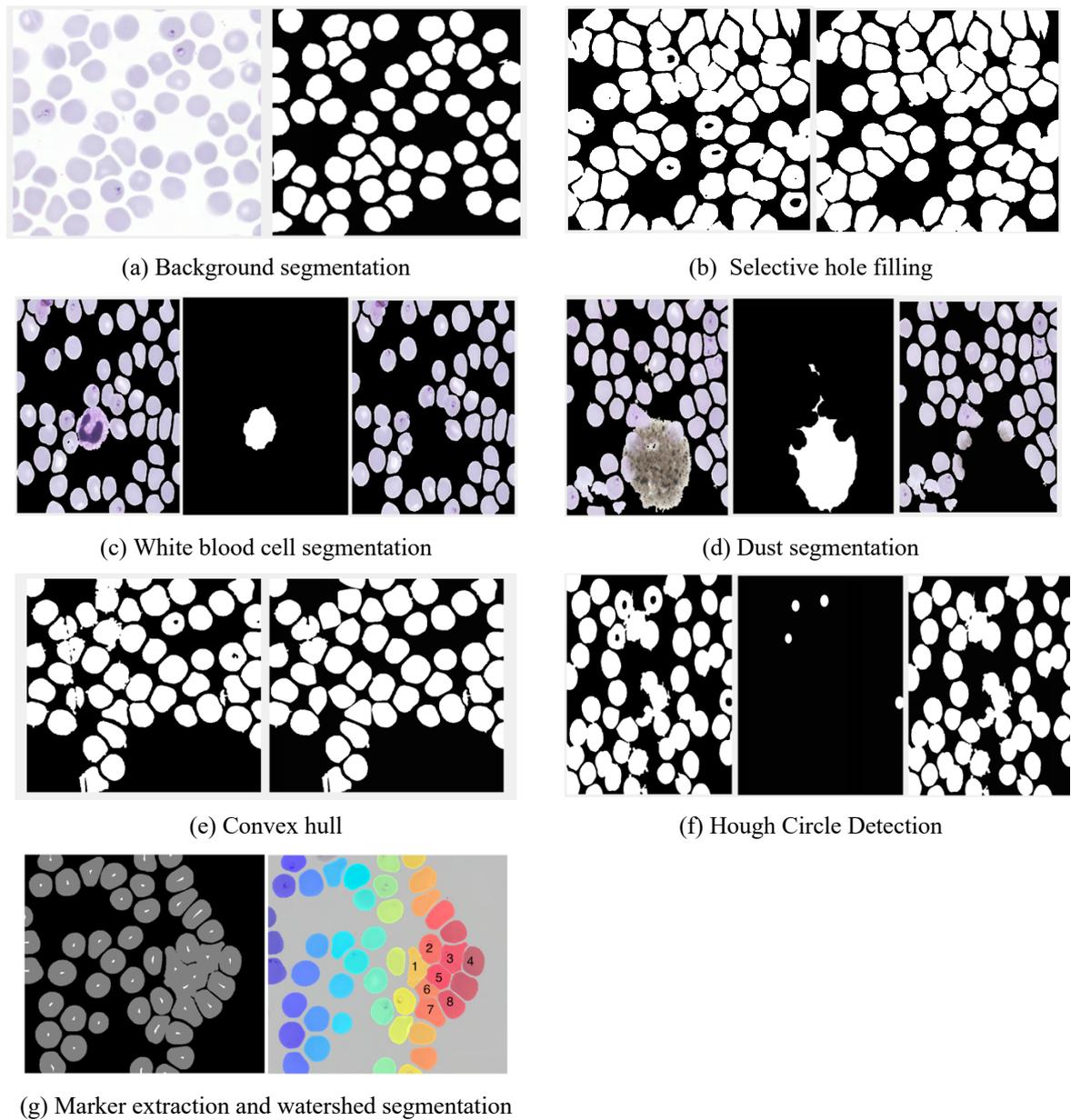
Some RBCs were disturbed and partially separated due to physical damage. Convex hull method was utilized to encapsulate these separated RBCs. However, application of this method was limited to single sized RBCs in this study. In other studies, this feature of the method was exploited for the detection of single or two overlapped cells (Romero-Rondón et al, 2016; Abbas et al., 2018). In our study, area filtering was utilized to extract partially separated single sized RBCs. Convex hull was then applied to encapsulate RBCs for intact segmentation. Convex hull operation was also employed for partially visible RBCs at the corner of the image. As RBCs at the corner are in small size due to partial visibility, small area RBCs were extracted. Then convex hull operation was performed. The result of two convex hull and selective hole refilling operation outputs were combined with union set operation. Final area filtering was applied to remove irrelevant objects other than red blood cells (less than 750 pixels). The convex hull is mathematically defined as:

$$\sum_{i=1}^{|x|} \alpha_i x_i |(\forall_u: \alpha_i \geq 0) \& \sum_{i=1}^{|x|} \alpha_i = 1 \quad (4.6)$$

where  $x_i$  is a point in a set of finite points  $|x|$ , and  $\alpha_i$  is the weight of  $x_i$  while the addition of all weights must equalize to 1 as mean normalized.

#### 4.2.7 Hough transform for circle detection

In our study, the hole filling method was applied to segmented objects which has an area between 1 to 3 RBCs size. However, some high numbers of overlapping RBCs present central pallor which is segmented as background and causes over-segmentation of related red blood cells by the watershed transformation. In order to solve this problem, Hough transform was applied to detect circles with a radius range from 5 to 25, with 88% sensitivity on Matlab 2019a. These circles were dilated by a disk-shaped structuring element with a radius of 5 and combined with segmented red blood cells. The final mask was created at the end of this operation.



**Figure 4.5:** Illustration of proposed local minimum histogram background segmentation steps

#### 4.2.8 Watershed segmentation

Prior to the watershed algorithm, distance transformation followed by extended minima transform was applied to extract foreground-markers. Then, the watershed algorithm was used to segment overlapping and single RBCs. Each segmentation was surrounded by a bounding box. The boxes with areas less than 500 pixels were not counted as RBCs. All the

steps are illustrated in Figure 4.3. with different image inputs in order to emphasize the purpose of algorithm steps. It could be visually seen in Figure 4.5. (a) that background segmentation was successfully achieved as all red blood cells were segmented and in Figure 4.5. (b) selective hole filling is applied. In Figure 4.5. (c), the white blood cell was removed from the image. Segmentation of red blood cells by the removal of dust particles is illustrated in Figure 4.5 (d), whereas Figure 4.5. (e) shows the encapsulation of disturbed RBCs. This allows the segmentation of disturbed RBCs as one piece instead of falsely over-segmented. Figure 4.5 (f) shows overlapping 5 red blood cells where hole filling operation was not applied to. Here, Hough transform was required in order to fill those 3 holes to prevent over-segmentation in the next step. Finally, in Figure 4.5. (g), marker extraction, and successful separation of overlapping 8 red blood cells is shown.

#### 4.2.8 Evaluation

Jaccard similarity index was employed as background segmentation metric and the relationship between manually counted and automatically counted red blood cells can be quantified by accuracy, precision, recall, and F1 score. The elements of confusion matrix were used to calculate these parameters as:

$$Accuracy = \frac{|TP+TN|}{|TP+FP+TN+FN|} \quad (4.7)$$

$$Precision = \frac{|TP|}{|TP+FP|} \quad (4.8)$$

$$Recall = \frac{|TP|}{|TP+FN|} \quad (4.9)$$

$$F1 = 2x \frac{Precision \times Recall}{Precision+Recall} \quad (4.10)$$

where TP is the number of correctly classified cells and TN is the background region, which is not considered in the context of cell detection. FN indicates the number of missed red blood cells and FP refers to false detections. In 4.7 - 4.10 performance parameters are illustrated where accuracy refers to the proportion of true results that is used to quantify the relation between automated and manually counted data. Precision refers to the proportion of truly counted cells among positive results. Recall is also named as sensitivity, which refers

to the proportion of positives for correctly counted cells. F-measure (F1) is harmonic mean of recall and precision.

### **4.3 Data Summary**

#### **4.3.1 U-net architecture**

Data augmentation was used on 400 images where 5600, 1200, and 600 for training, validation, and testing was acquired. The algorithm of Molina et al. (2020) and k-means clustering was evaluated on these 400 images without data augmentation applied whereas The U-net architecture was evaluated on the test data.

#### **4.3.1 Selective hole filling approach**

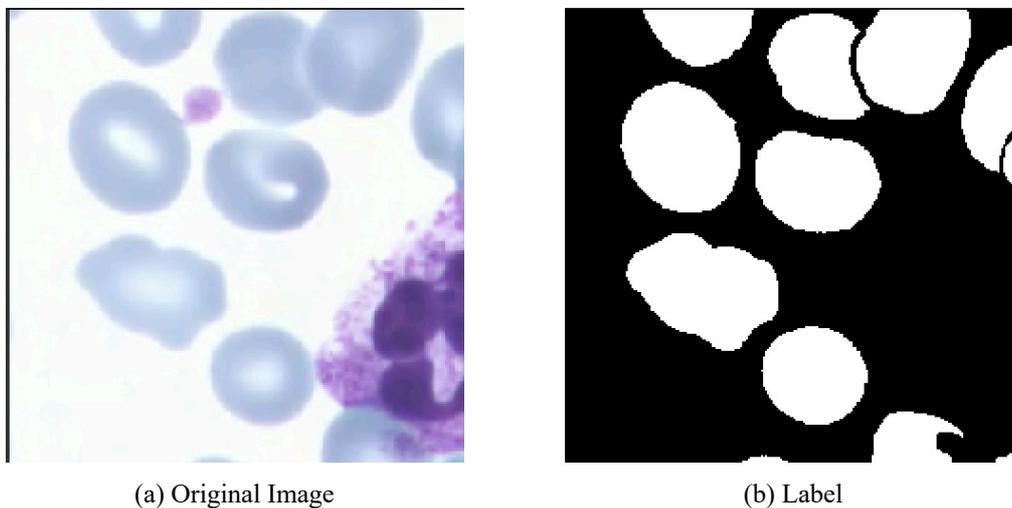
A different set of 66 images was used for red blood cell segmentation evaluation and another set of 15 images that consists of a high number of overlapped red blood cells was used for evaluation of segmentation and separation of red blood cells.

## CHAPTER 5

### RESULTS

#### 5.1 U-Net Architecture

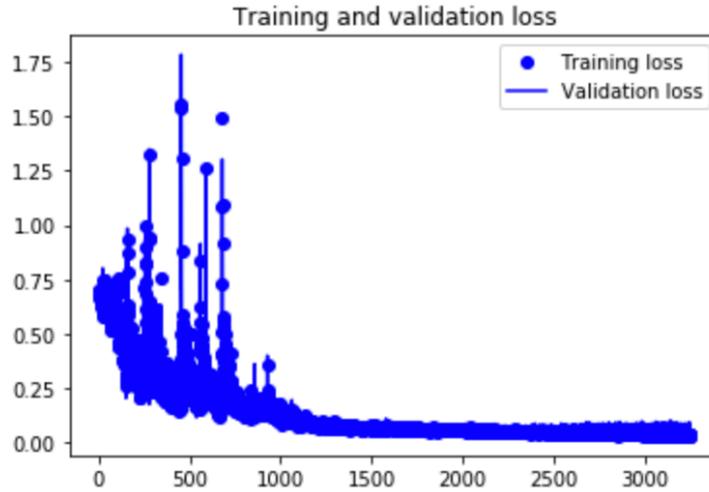
The ground truth data for corresponding images was used as target in the network. The ground truth data includes red blood cells labelled as bright and background as dark. The intersection points among cluster of red blood cells are also labelled as background which is illustrated in Figure 5.1. So that, network learns to separate overlapping and touching red blood cells through this intersection points.



**Figure 5.1:** Original image and its corresponding label

The other objects in the film that are not in area of interest such as white blood cells, dust particles, platelets and artifacts are also labelled as background.

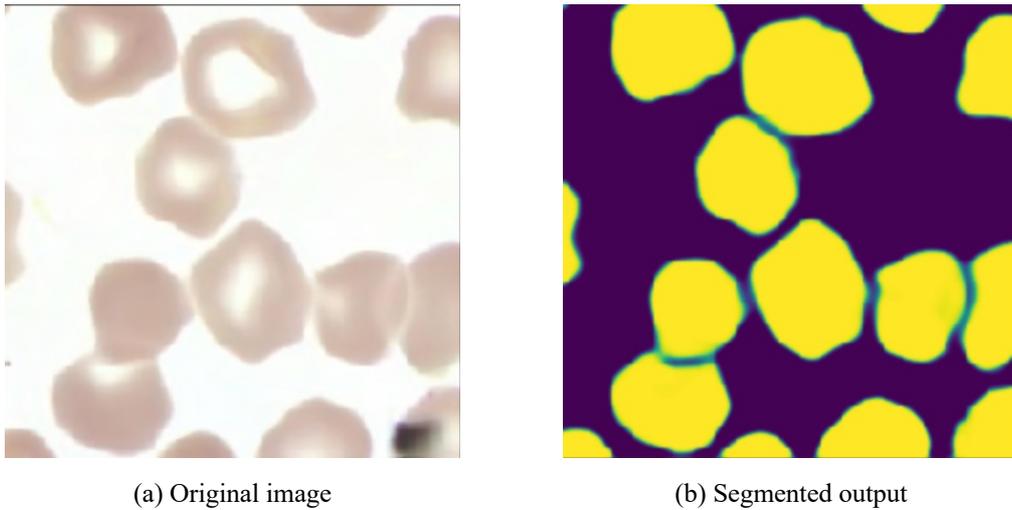
The network was trained for 3200 epochs where best training was achieved at the 1769th epoch with 0,04795 validation loss. The corresponding loss versus epoch curve is illustrated in Figure 5.2. The test data was used to evaluate the segmentation efficient of the network where 93,8% Jaccard similarity coefficient was achieved. To best of our knowledge, this exceeded previous results. The data was also divided as 80%/10%/10% and 60%/20%/20% for training, validation and testing and the corresponding results are illustrated in Table 5.1.



**Figure 5.2:** Training and validation loss expressed as epoch on x axis and loss on y axis

Recently, Poostchi et al. (2018a) had segmented red blood cells with 92.5% Jaccard similarity coefficient even after manual filtration of white blood cells. Our study was also compared with popular segmentation algorithms employed in this domain, Otsu's thresholding & k-means clustering. Recently in a study, Otsu's thresholding was employed with pre-processing and post-processing steps in order to segment red blood cells (Molina et al., 2020). The segmentation algorithm of the recent study was created with its preprocessing and postprocessing steps. However, border cleaning operation was excluded as images in our dataset exhibits important number of red blood cells at the borders. K-means clustering algorithm is another segmentation algorithm that is popular for blood cell WBC, RBC segmentations (Savkare and Narote, 2015; Zhang et al., 2014). K-means clustering was created for comparison to the proposed study by following steps. Median filtering and morphological reconstruction were used for edge preservation and removal of platelets respectively. A disc-shaped structuring element with radius of 3 was used for markers of morphological reconstruction. After that, segmentation of red blood cells was achieved by k-means clustering with  $k=2$  followed by morphological hole filling for central pallor. These two created algorithms, Otsu's thresholding and k-means clustering was applied to our data set that is consisted of 400 images without any data augmentation, and 88.26% and 87.89% Jaccard similarity index was acquired respectively. The results are illustrated in Table 5.2. The proposed network achieved segmentation and separation of overlapping red blood cells which is illustrated in Figure 5.3. On top of that, red blood cells with central pallor were

(white area at the center of the red blood cells) segmented as part of red blood cells that is shown in Figure 5.4. The color of the central pallor is very similar to the background. So, the network has learned to segment central pallor together with red blood cells providing intact RBC segmentation. Moreover, RBC segmentation with WBC presence is illustrated in Figure 5.5. It should be noted that, white blood cell was removed from the image. In other words, white blood cell was segmented as a part of the background.



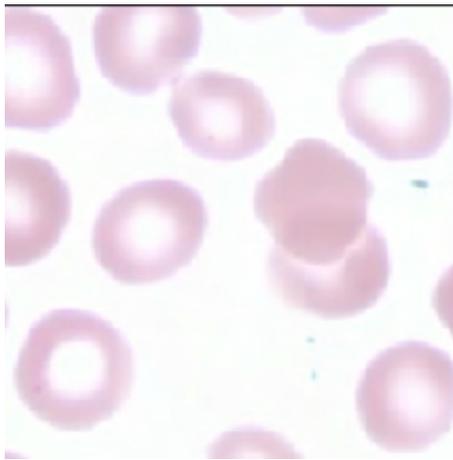
**Figure 5.3:** Segmentation and separation of overlapping red blood cells

**Table 5.1:** Results of U-net architecture on different data divisions

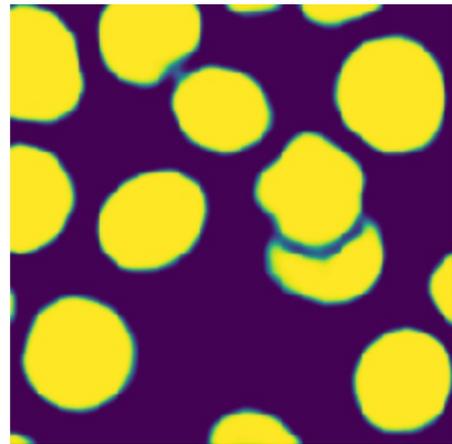
Training, Validation, Testing	Jaccard similarity coefficient
80%, 10%, 10%	93.3%
70%, 15%, 15%	93.8%
60%, 20%, 20%	92.9%

**Table 5.2:** Jaccard similarity coefficient (U-net vs others)

Study	Jaccard similarity coefficient
The proposed the U-net architecture	93.8%
Molina et al.	88.26%
K-means clustering	87.89%

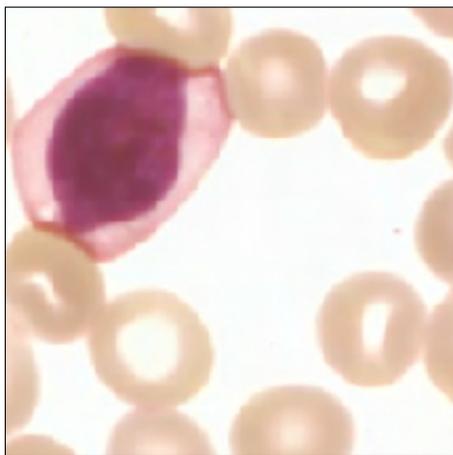


(a) Original image

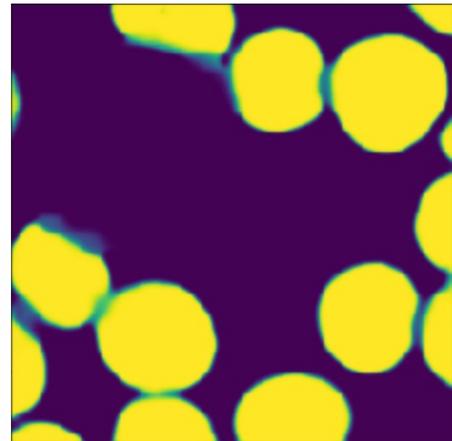


(b) Segmented output

**Figure 5.4:** Segmentation of red blood cells with central pallor.



(a) Original image



(b) Segmented output

**Figure 5.5:** Segmentation of red blood cells with presence of white blood cell.

## 5.2 Marker-controlled Watershed Segmentation

The proposed study was compared to another histogram segmentation algorithm, Zack's thresholding. Zach's thresholding is a popular histogram segmentation algorithm used to segment red blood cells (Damahe et al., 2011). Another comparison was made with the k-means clustering algorithm with  $k = 2$ . The results of corresponding algorithms are given in Table 5.3.

**Table 5.3:** Jaccard similarity coefficient results (Proposed watershed vs others)

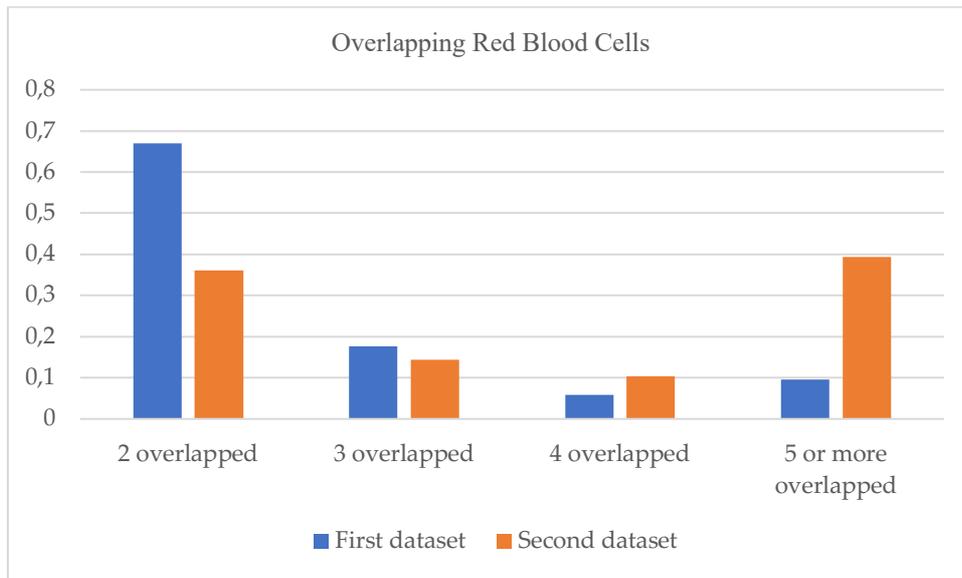
<b>Study</b>	<b>Jaccard similarity coefficient</b>
Proposed study	93.15%
Zack's thresholding	85.53%
K-means clustering	88.34%

The success of our algorithm depends on combinations of several steps that complement each other. However, evaluation of the selective hole filling could be determined by replacing selective hole filling with classic hole filling, and keeping the rest of the algorithm steps untouched. Another comparison was done by replacing the local minimum histogram thresholding by k-means clustering with  $k=2$ . Another set of 15 images with an abundance of overlapping red blood cells were used to evaluate the segmentation of overlapping RBCs. The comparison of results was illustrated in Table 5.4. The visual comparison of extracted markers and overlapping red blood cell segmentation is shown in Figure 5.7 and Figure 5.8 separately.

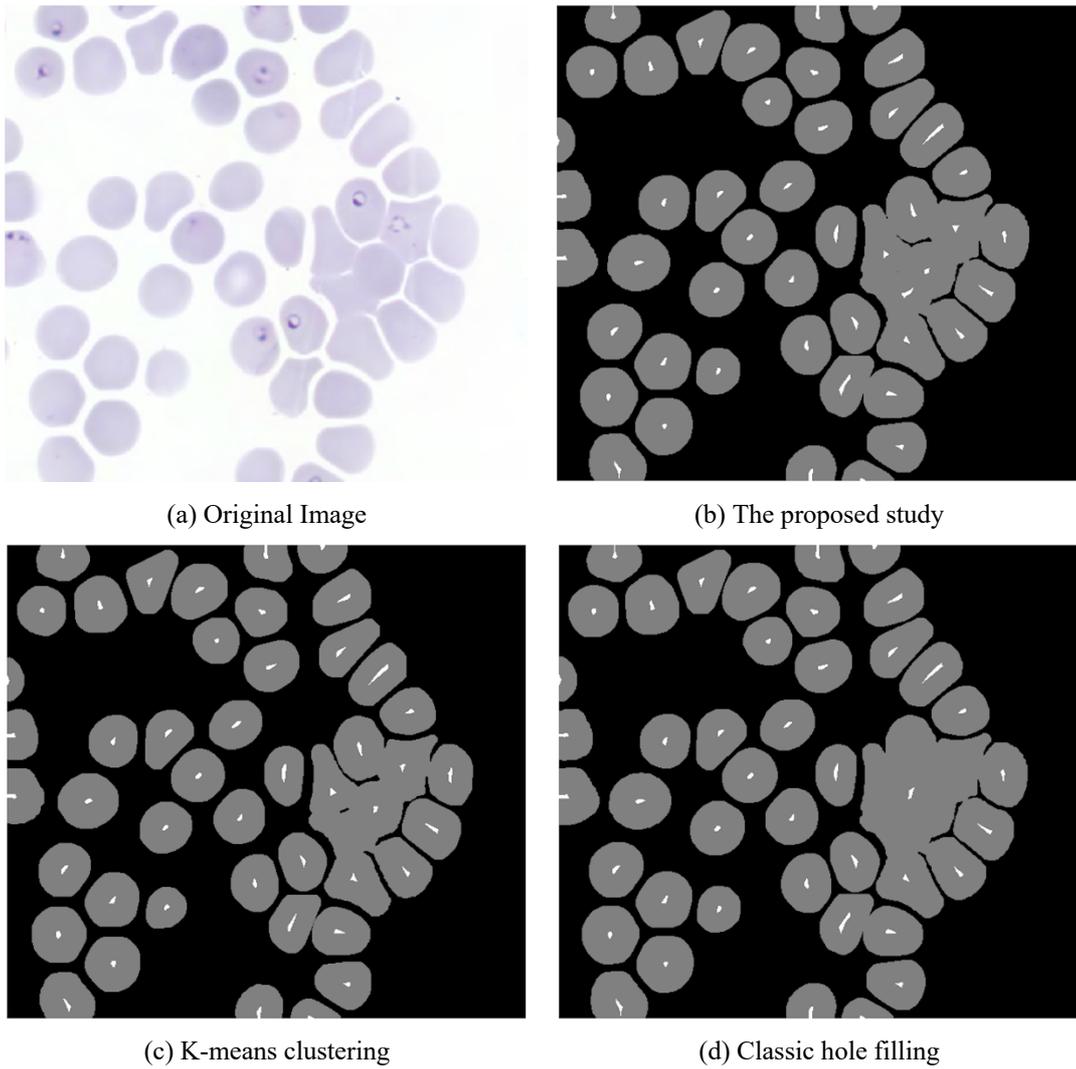
The overlapping red blood cells ratio versus whole red blood cells was analysed as 17% for the first dataset whereas 34% for the second. The first dataset included randomly chosen 66 images and the second data set included 15 images with an abundance of overlapping cells. The ratio of the different number of overlapping RBCs is given in Figure 5.6. The first dataset had an abundance of two overlapped cells whereas the second dataset showed an abundance of 5 or more RBCs.

**Table 5.4:** Performance evaluation on high number of overlapping cells samples.

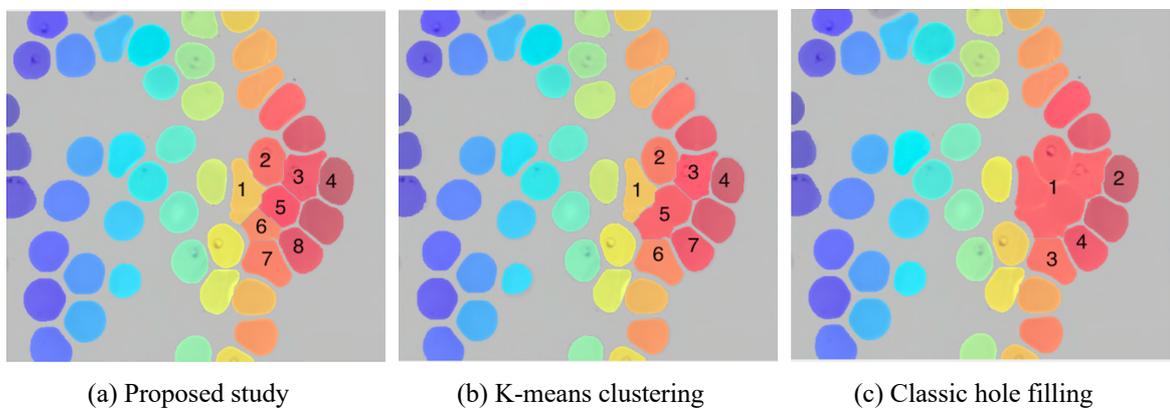
<b>Method</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
Proposed study	793	2	7	0,989	0,997	0,991	0,994
Classic hole filling replacement	728	2	28	0,960	0,997	0,962	0,979
K-means clustering replacement	733	15	64	0,902	0,979	0,919	0,948



**Figure 5.6:** Comparison of two dataset with different number overlapping red blood cell distribution



**Figure 5.7:** Illustration of markers



**Figure 5.8:** The separation of overlapping red blood cells

The proposed segmentation algorithm was able to compensate sensitivities of watershed transformation. As a result, segmentation and separation of multiple RBCs were improved. Images used in this study varied in terms of colours and artifacts. The local minimum histogram thresholding was employed for the initial segmentation of the background. The selective hole filling algorithm was used to keep the inner edges of overlapping RBCs, which aids watershed transformation to segment a high number of overlapping red blood cells. One of the problems with the selective hole filling method is the exhibition of central pallor which causes over-segmentation issues in case of high numbers of RBCs overlapping. Hough transform for circle detection was employed for this problem. However, not all images exhibit circular central pallor that can be detected by the Hough transform, therefore, this problem remained for some RBCs. Convex hull algorithm was employed for the intact segmentation of disturbed RBCs, which improved segmentation and quantification of RBCs in return.

## CHAPTER 6

### DISCUSSION

Even the segmentation of red blood cells is a challenge, the proposed thesis aims to remove white blood cells and other irrelevant objects besides. In the previous works, Poostchi et al. (2018a) utilized SEPACELL R-500 II to filter white blood cells manually beforehand and Rajaraman et al. (2018) ground truth filtered the white blood cells. So, the removal of white blood cell task was not included. However, removal of white blood cells is required to be not dependent on other applications especially manual ones. Therefore, manual removal of objects was not included in this study as it is intended to automate the segmentation task from end to end. On the other hand, RBC segmentation for the evaluation of malaria parasites is either neglected or oversimplified in the literature for deep learning algorithms for the detection of malaria parasites. RBCs were either segmented by classic image processing algorithms (Bibin and Punitha, 2017; Rajaraman et al., 2018; Molina et al., 2020; Poostchi et al., 2018a) or presegmented RBCs (Reddy and Juliet, 2019) were used to evaluate individual RBCs as discussed previously. Individual evaluation of red blood cells for malaria parasites requires segmentation of individual red blood cells. Improvements in the efficiency and reliability of segmentation favor quantitative and qualitative blood sample evaluation. The purpose of this research was to implement a deep learning method that could segment red blood cells without the need for preprocessing and manual filtering steps to be applied. The proposed approach enables individual evaluation of red blood cells in a fully automated way.

In this thesis, improvement of marker-controlled watershed segmentation was also studied besides U-net architecture. The extraction of markers plays important role in the success of watershed segmentation. The classic hole filling approach is not suitable for the extraction of markers for a high number of overlapping red blood cells. The selective hole filling approach was proposed to overcome this issue, where inner corner details contribute to the extraction of markers which, is illustrated in Figure 5.7. As morphological hole filling is not applied to the high number of overlapped red blood cells, the Hough circle detection was employed to compensate the central pallor problem. However, not all central pallors are circular. Therefore, central pallor problem remained for some red blood cells. Moreover, the

selective hole filling method requires an area estimation of triple overlapped red blood cells. Therefore, any other dataset should be evaluated for the area of triple overlapped red blood cells.

## **CHAPTER 7**

### **CONCLUSION**

In this thesis, data augmentation algorithms such as random elastic deformation, horizontal and vertical flips were employed to increase the sample size. The segmentation and separation of red blood cells were achieved by the U-net architecture successfully. This technique removed white blood cell manual filtering and allowed the segmentation algorithm to run automatically from end to end. Irrelevant artifacts have also been eliminated which poses another problem for classic image processing algorithms. Besides, the highest value of the Jaccard similarity index was obtained to the best of our knowledge. This study, however, aims only to conduct segmentation. This study can also be used prior to the classification of malaria blood smear images for the segmentation of red blood cells. In order to identify individual red blood cells as healthy and infected, classifier algorithms could be introduced. The degree of infection may in return, be assessed.

On the other hand, marker-controlled watershed segmentation with selective hole filling approach has also been studied in this thesis. However, this algorithm has been tested on different sized images with more red blood cells. Here, it has been noted that the selective hole filling method increases the success of segmentation of high numbers of overlapping red blood cells. Additionally, application of the local minimum histogram segmentation has increased the success of segmentation.

## REFERENCES

- Abbas, N., Saba, T., Mohamad, D., Rehman, A., Almazayad, A. S., & Al-Ghamdi, J. S. (2018). Machine aided malaria parasitemia detection in Giemsa-stained thin blood smears. *Neural Computing and Applications*, 29(3), 803-818.
- Aggarwal, C. C. (2018). *Neural networks and deep learning*. New York, NY: Springer.
- Bibin, D., Nair, M. S., & Punitha, P. (2017). Malaria parasite detection from peripheral blood smear images using deep belief networks. *IEEE Access*, 5, 9099-9108.
- Borgheti-Cardoso, L. N., San Anselmo, M., Lantero, E., Lancelot, A., Serrano, J. L., Hernández-Ainsa, S., ... & Sierra, T. (2020). Promising nanomaterials in the fight against malaria. *Journal of Materials Chemistry B*, 8(41), 9428-9448.
- Buduma, N., & Locascio, N. (2017). *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. Sebastopol, CA: O'Reilly Media, Inc.
- Burkov, A. (2019). *The hundred-page machine learning book (Vol. 1)*. Quebec City, Can.: Andriy Burkov.
- Chollet, F. (2018). *Deep learning with Python*. Shelter Island, NY: Manning Publications Co.
- Damahe, L. B., Krishna, R., Janwe, N., & Thakur, N. V. (2011). Segmentation based approach to detect parasites and RBCs in blood cell images. *Int J Comput Sci Appl*, 4(2), 71-81.
- Dave, I. R., & Upla, K. P. (2017). Computer aided diagnosis of malaria disease for thin and thick blood smear microscopic images. *In 2017 4th international conference on signal processing and integrated networks (SPIN)* (pp. 561-565). Noida: IEEE.

- Di Ruberto, C., Dempster, A., Khan, S., & Jarra, B. (2002). Analysis of infected blood cell images using morphological operators. *Image and vision computing*, 20(2), 133-146.
- Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
- Gong, S., Liu, C., Ji, Y., Zhong, B., Li, Y., & Dong, H. (2018). *Advanced Image and Video Processing Using MATLAB (Vol. 12)*. Cham: Springer.
- Gonzalez, R. C., and Woods, R. E. (2018). *Digital Image Processing (4th. ed.)*. New York, NY: Pearson
- González-Betancourt, A., Rodríguez-Ribalta, P., Meneses-Marcel, A., Sifontes-Rodríguez, S., Lorenzo-Ginori, J. V., & Orozco-Morales, R. (2016). Automated marker identification using the Radon transform for watershed segmentation. *IET Image Processing*, 11(3), 183-189.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.
- Huisjes, R., Bogdanova, A., van Solinge, W. W., Schiffelers, R. M., Kaestner, L., & Van Wijk, R. (2018). Squeezing for life—properties of red blood cell deformability. *Frontiers in physiology*, 9, 656.
- Iglovikov, V., & Shvets, A. (2018). Terausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*.
- Koul, A., Ganju, S., & Kasam, M. (2020). Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python. Keras & TensorFlow.

- Loddo, A., Di Ruberto, C., & Kocher, M. (2018). Recent advances of malaria parasites detection systems based on mathematical morphology. *Sensors*, 18(2), 513.
- Molina, A., Alférez, S., Boldú, L., Acevedo, A., Rodellar, J., & Merino, A. (2020). Sequential classification system for recognition of malaria infection using peripheral blood cell images. *Journal of Clinical Pathology*. 73(10), 1-5.
- Mushabe, M. C., Dendere, R., & Douglas, T. S. (2013, July). Automated detection of malaria in Giemsa-stained thin blood smears. *In 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3698-3701). Chicago: IEEE.
- National Institute of Allergy and Infectious Diseases. (2016). Malaria: Malaria Parasite, Mosquito, and Human Host. Retrieved December 21, 2020 from <https://www.niaid.nih.gov/diseases-conditions/malaria-parasite>
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 2018). San Francisco, CA: Determination press.
- Nixon, M., and Aguado, A. (2019). *Feature extraction and image processing for computer vision*. London: Academic press.
- Pillay, E., Khodaiji, S., Bezuidenhout, B. C., Litshie, M., & Coetzer, T. L. (2019). Evaluation of automated malaria diagnosis using the Sysmex XN-30 analyser in a clinical setting. *Malaria journal*, 18(1), 1-14.
- Poostchi, M., Ersoy, I., McMenamin, K., Gordon, E., Palaniappan, N., Pierce, S., ... & Jaeger, S. (2018). Malaria parasite detection and cell counting for human and mouse using thin blood smear microscopy. *Journal of Medical Imaging*, 5(4), 044506.
- Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S., & Thoma, G. (2018). Image analysis and machine learning for detecting malaria. *Translational Research*, 194, 36-55.

- Rajaraman, S., Antani, S. K., Poostchi, M., Silamut, K., Hossain, M. A., Maude, R. J., ... & Thoma, G. R. (2018). Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6, 1-17.
- Reddy, A. S. B., & Juliet, D. S. (2019, April). Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. In *2019 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0945-0949). Chicago: IEEE.
- Romero-Rondón, M. F., Sanabria-Rosas, L. M., Bautista-Rozo, L. X., & Mendoza-Castellanos, A. (2016). *Algorithm for detection of overlapped red blood cells in microscopic images of blood smears*. *Dyna*, 83(198), 187-194.
- Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Munich: Springer.
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4<sup>th</sup> edition). Hoboken: Pearson.
- Savkare, S. S., & Narote, S. P. (2011). Automatic detection of malaria parasites for estimating parasitemia. *International Journal of Computer Science and Security (IJCSS)*, 5(3), 310.
- Savkare, S. S., & Narote, S. P. (2015). Blood cell segmentation from microscopic blood images. In *2015 International Conference on Information Processing (ICIP)* (pp. 502-505). Pune: IEEE.
- Solomon, C., and Breckon, T. (2011). *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. Hoboken, NJ: John Wiley & Sons.

- Tek, F. B., Dempster, A. G., & Kale, I. (2010). Parasite detection and identification for automated thin blood film malaria diagnosis. *Computer vision and image understanding*, 114(1), 21-32.
- The MaMic Image Database. (n.d). Thin smear blood samples for identification of Plasmodium Falciparum. Retrieved August 14, 2019 from <http://fimm.webmicroscope.net/research/momic/mamic>
- Trask, A. (2019). Grokking deep learning. Shelter Island, NY: Manning Publications Co.
- Karimov, A., Razumov, A., Manbatchurina, R., Simonova, K., Donets, I., Vlasova, A., ... & Ushenin, K. (2019). Comparison of UNet, ENet, and BoxENet for Segmentation of Mast Cells in Scans of Histological Slices. *In 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)* (pp. 0544-0547). Novosibirsk: IEEE.
- Vijayalakshmi, A., Rajesh Kanna, B. (2020). Deep learning approach to detect malaria from microscopic images. *Multimedia Tools and Applications*, 79(21), 15297-15317.
- World Health Organization. (2010). *Basic malaria microscopy: Part I. Learner's guide (No. Ed. 2)*. World Health Organization.
- World Health Organization. (2020). World malaria report 2019. Geneva: World Health organization; 2019.
- Li, Y., Zhu, R., Mi, L., Cao, Y., & Yao, D. (2016). Segmentation of white blood cell from acute lymphoblastic leukemia images using dual-threshold method. *Computational and mathematical methods in medicine*, 2016.
- Zafari, S., Eerola, T., Sampo, J., Kälviäinen, H., & Haario, H. (2015). Segmentation of overlapping elliptical objects in silhouette images. *IEEE Transactions on Image Processing*, 24(12), 5942-5952.

Zhang, C., Xiao, X., Li, X., Chen, Y. J., Zhen, W., Chang, J., ... & Liu, Z. (2014). White blood cell segmentation by color-space-based k-means clustering. *Sensors*, 14(9), 16128-16147.

## **APPENDICES**

**APPENDIX 1**  
**ETHICAL APPROVAL DOCUMENT**



**Near East University**

**ETHICAL APPROVAL DOCUMENT**

Date: 21/12/2020

To the Graduate School of Applied Sciences,

For the thesis Project entitled as “U-NET ARCHITECTURE FOR SEGMENTATION OF OVERLAPPING RED BLOOD CELLS IN MALARIA BLOOD SMEAR IMAGES”, the researchers declare that they did not collect any data from human/animal or any other subjects. Therefore, this project does not need to go through the ethics committee evaluation.

Title: Assist. Prof. Dr.

Name Surname: Elbrus Imanov

Signature: 

Role in the Research Project: Supervisor

## APPENDIX 2

### SIGNED SIMILARTY REPORT

	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
<input type="checkbox"/>	Fatih Veysel Nurçin	Abstract	0% 	--	--		1495552553	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 5 - Results	0% 	--	--		1495556277	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 6 - Discussion	0% 	--	--		1495558334	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 7 - Conclusion	0% 	--	--		1495557166	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 4	5% 	--	--		1495555627	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 1	7% 	--	--		1495552909	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Entire Thesis	8% 	--	--		1495563131	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 2	11% 	--	--		1495553485	27-Jan-2021
<input type="checkbox"/>	Fatih Veysel Nurçin	Chapter 3	12% 	--	--		1495554751	27-Jan-2021

Assist.Prof.Dr. Elbrus Imanov



## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Nurçin, Fatih Veysel  
Nationality: Cyprus  
Date and Place of Birth: 17 March 1990, Girne  
Marital Status: Single



### EDUCATION

Degree	Institution	Year of Graduation
B.Sc.	Near East University, Biomedical Engineering	2013
M.Sc.	Near East University, Biomedical Engineering	2015

### WORK EXPERIENCE

Year	Place	Enrolment
2013-2015	Near East University	Research Assistant
2015-	Near East University	Lecturer

### FOREING LANGUAGES

English Language, fluently spoken and written

EPTT, Near East University Proficiency

### PUBLICATIONS

- Ozsahina, D. U., Uzuna, B., Musaa, M. S., Helwana, A., Wilsona, C. N., Veysel, F., ... & Ozsahina, I. (2017). Evaluating cancer treatment alternatives using Fuzzy PROMETHEE Method. *International Journal of Advanced Computer Science and Applications*, 8(10), 177-82.

## CONFERENCES & CONGRESSES

- Nurçin, F. V., Imanov, E., Işın, A., & Ozsahin, D. U. (2017). Lie detection on pupil size by back propagation neural network. *Procedia Computer Science, 120*, 417-421.
- Ozsahin, D. U., Uzun, B., Musa, M. S., Şentürk, N., Nurçin, F. V., & Ozsahin, I. (2017). Evaluating nuclear medicine imaging devices using fuzzy PROMETHEE method. *Procedia Computer Science, 120*, 699-705.
- Altıparmak, H., & Nurçin, F. V. (2019, February). Segmentation of Microscopic Breast Cancer Images for Cancer Detection. *In Proceedings of the 2019 8th International Conference on Software and Computer Applications* (pp. 268-271).

## COURSES GIVEN

### *Undergraduate:*

- Medical English (English and Turkish)
- Introduction to Biomedical Engineering (English and Turkish)
- Biomedical Sensors (English and Turkish)
- Biomedical Instrumentation (Turkish)
- Biomedical Electronics (English and Turkish)