

中图分类号:  
UDC:

学校代码: 10055  
密级: 公开

南开大学  
Nankai University  
硕士学位论文  
Master's Dissertation

论文题目: 基于机器学习模型的疟疾图像分类

Title: Malaria Image Classification using  
Machine Learning Models

论文作者 弗朗西斯科·恩维拉 Author Francisco Ngwira

指导教师 张玉志 教授 Supervisor Prof. Yuzhi Zhang

申请学位 软件工程硕士 Degree Master of Software Engineering

培养单位 软件学院 Institute College of Software

学科专业 软件工程 Discipline Software Engineering

研究方向 机器学习 Research Area Malaria Image Classification

答辩委员会主席 张波 Chairman of Committee Zhang Bo

评阅人 高铁杠 许静 Reviewers Gao Tiegang, Xu Jing

南开大学研究生院

Graduate School of Nankai University

2020 年 5 月

## The Authorization of Dissertation in Nankai University

I fully understand the management regulations of Nankai University on the retention and utilization of graduate student's dissertations and agree to submit my electronic form and corresponding paper version of my dissertation to Nankai University.

I understand that Nankai University has the right to utilize the dissertation within “The Law of Copyright” of the People's Republic of China, and I agree to authorize the university in the following aspects. That is:

1. The university will adopt the dissertation into "The Full-text Database of Nankai University's Dissertations " and can provide the public dissertations to teachers and students as the reading in the library and free information services including catalog search, abstracts and the first 16-pages browsing on the campus network;

2. The university can adopt photocopying, micro-printing or other means to save a copy of the dissertation. According to the provisions of the Ministry of Education, the university should submit public dissertations to the institution designated by the Ministry of Education;

3. This authorization applies to confidential dissertations after decryption.

I promise: the dissertation is completed by myself during studying at Nankai University and has been passed the defend; the electronic version of the dissertation is consistent with the contents of this paper; if these contents are different, I undertake my responsibility.

I have signed one copy of this authorization (that is one page of the dissertation), which shall be submitted to the library for retention

Author's signature: \_\_\_\_\_ Date: 2020.04.14

### The Author's Information

|                           |  |            |            |                          |                      |
|---------------------------|--|------------|------------|--------------------------|----------------------|
| Title                     | Malaria Image Classification using Machine Learning Models   |            |            |                          |                      |
| Name                      | Francisco Ngwira   | Student ID | 2120186030 | The Date of Oral Defense | 2020.06.01           |
| The Type of Dissertation: | Master   |            |            |                          |                      |
| Institution/Department    | College of Software  |            |            | Major:                   | Software Engineering |
| Email Address:            | <a href="mailto:fngwira14@gmail.com">fngwira14@gmail.com</a> or <a href="mailto:ngwirafrancisco@gmail.com">ngwirafrancisco@gmail.com</a> |            |            |                          |                      |

Note: The authorization applies to all students' dissertations. The author fills in the duplicate authorization and hands in the library. The non-public dissertation should be with “The Approval Form of non-public dissertation at Nankai University

## Declaration

I solemnly declare: The presented dissertation titled; Malaria Image Classification using Machine Learning Models, is my original work under the guidance of my supervisor. All references have been stated and the dissertation doesn't contain any other works created, published or not published. Other individuals and the collectives that contribute to the dissertation have been stated in the dissertation. I will be responsible for all the legal liability of the statement.

Author's signature: \_\_\_\_\_ Date: \_\_\_\_\_.

IJSER

## **Acknowledgment**

This work would not have been possible had it been for the innumerable amount of help I got from different people and institutions.

First and foremost, I am grateful to the administration and staff of the Ministry of Commerce of the People's Republic of China and Software Engineering Department, Nankai University, who organized the bursary for me to pursue the master's program.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every department of my life.

I would also like to express special thanks to my supervisor Dr. Shenglin Zhang for his help throughout my thesis and also for the Algorithm Design course which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to Ms. Wei Pan, Professor Wang Binhui, Professor Guifei Jiang, and Professor Zhang Bo for their tremendous support and cooperation. Each time we meet, they were always providing me with the the help that have contributed to my research and been able to complete my thesis. I appreciate their patience and guidance throughout the whole thesis.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

## **Plagiarism Certificate (CNKI Report)**

This thesis has been checked for Plagiarism. CNKI report endorsed by Supervisor is attached.

Signature of Student

IJSER

Signature of Supervisor

### **Copyright Statement**

- Copyright in content of this proposition rests with the understudy creator. Duplicates (by any process) either in full or of extricates, may be made as it were in agreement with instructions given by the creator and held up within the Library of College of Program, Nankai College (NKCS). Points of interest may be gotten by the Librarian. This page must frame a portion of any such duplicates made. Assist duplicates (by any process) may not be made without the consent (in composing) of the creator who happens to be the author.
- The possession of any property rights which may be portrayed in this proposal is vested in College of Computer program, Nankai College, subject to any earlier assention to the opposite, and may not be made accessible for utilize by third parties without the composed consent of the NKCS, which can endorse the terms and conditions of any such understanding.
- More data on the conditions beneath which revelations and abuse may take put is accessible from the Library of College of Program, Nankai College.

## **Dedication**

I dedicate this thesis to my parents, Smith and Annie Ngwira, for their prayers and encouragement. Second, my brothers and sisters for fueling the fighting spirit through words of encouragement. Third, dedication goes to my friends, Nolan, Odel, Quincy, Quinton and Nina for cheering me, not forgetting late Dumison we all miss you. Lastly, special thanks to my supervisor, Dr. Shenglin Zhang for his guidance and support.

IJSER

## 中文摘要

疟疾是由按蚊引起的，通过蚊子叮咬传播。检测疟疾的过程如下：扎破手指收集血液样本并涂在平板玻璃上干燥，之后使用显微镜扫描，生成疟疾细胞图像用以检测细胞病毒。然而，人工检测疟疾细胞图像容易出错并且人力消耗大。因此，本文应用了一种深度学习模型——卷积神经网络——对疟疾细胞图像进行分类以自动识别疟疾病毒。

卷积神经网络包括特征提取、边缘检测以及对图像进行池化，以学习正常细胞和疟疾细胞的模式。在卷积神经网络中，每个像素都被赋予一个加权值用于图像分类。为了提高卷积神经网络的性能，将训练的迭代次数逐步提升到了 30 次。此外，本文的卷积神经网络使用一个基于 Visual Geometry Group 19 (VGG19) 的 sigmoid 激活函数作为最终激活函数，使用 Adam 优化器来解决噪声问题，还使用了二元交叉熵来计算训练过程中的损失。

为了验证卷积神经网络的性能，本文在由 28000 张疟疾细胞图像组成的真实数据集上进行了实验。其中 14000 张为感染了疟疾的细胞图像，剩余 14000 张为未感染疟疾的细胞图像。本文将卷积神经网络与支持向量机和随机森林进行了比较。在这三种方法中，卷积神经网络的 F1 分数最高。具体来讲，卷积神经网络的 F1 分数为 99%，显著高于支持向量机（86%）和随机森林（96%）。

关键词：卷积神经网络，支持向量机，随机森林分类器



## Abstract

Malaria disease, caused by anopheles mosquito, is transmitted through mosquito bite. The process of detecting malaria involves finger pricking to collect blood samples and smearing on the plate glass for drying, which later is scanned using a microscope to generate malaria cell images for malaria virus identification. The manual inspection of malaria cell images, however, is error prone and labor intensive. Consequently, in this paper, we apply a deep learning model, ConvNets, to automatically identify malaria virus by classifying malaria cell images.

ConvNets involves feature extraction, edge detection, and pooling of layers from images, to learn the patterns of normal and malaria cells. In ConvNets, each pixel is given a weighted value for image classification. To improve the performance of ConvNets, we escalate the EPOCHS number to 30. In addition, ConvNets applies a sigmoid activation function as the final activation function with Visual Geometry Group 19 (VGG19), and utilizes the Adam optimizer to address noisy problems and a binary cross-entropy to measure the loss during training procedure.

To demonstrate the performance of ConvNets, we experiment it on real-world dataset, which is composed of 28,000 malaria cell images, 14,000 of which are infected and the remaining 14,000 are uninfected. We compare ConvNets with SVM and Random Forest. ConvNets achieves the best F1-score among the three methods. More specifically, its F1-score is 99%, significantly higher than those of SVM (86%) and Random Forest (96%).

**Key Words:** *Convolutional Neural Network, Support Vector Machine, Random Forest Classifier*

## Table of Contents

|   |              |
|---|--------------|
| <b>The Authorization of Dissertation in Nankai University .....</b> | <b>i</b>     |
| <b>Declaration .....</b>  | <b>ii</b>    |
| <b>Acknowledgement .....</b>  | <b>iii</b>   |
| <b>Copyright Statement.....</b>                                     | <b>v</b>     |
| <b>Dedication.....</b>  | <b>vi</b>    |
| <b>中文摘要 .....</b>   | <b>vii</b>   |
| <b>Abstract .....</b>   | <b>viii</b>  |
| <b>1. Chapter: Introduction .....</b>                               | <b>- 1 -</b> |
| 1.1 Motivation .....  | - 3 -        |
| 1.2 Objectives.....   | - 3 -        |
| 1.3 Structure of Document .....                                     | - 5 -        |
| 1.4 Related Work.....   | - 5 -        |
| 1.5 The diagnosis process .....                                     | - 6 -        |
| 1.6 Diagnosis methods.....  | - 7 -        |
| <b>2 Chapter: Related Technologies .....</b>                        | <b>- 8 -</b> |
| 2.1 Support Vector Machines.....                                    | - 8 -        |
| 2.1.1 Support Vector Classifier .....                               | - 8 -        |
| 2.1.2 Support Vector Machine.....                                   | - 11 -       |
| 2.2 Random Forest.....  | - 13 -       |
| 2.2.1 Decision Trees .....  | - 14 -       |
| 2.2.2 Classification .....  | - 14 -       |
| 2.2.3 Training of decision trees .....                              | - 15 -       |
| 2.2.4 Splitting function.....                                       | - 16 -       |
| 2.2.5 Variations .....  | - 16 -       |
| 2.2.6 Random Forest.....  | - 17 -       |
| 2.2.7 Regression Forest.....  | - 17 -       |
| 2.2.8 Impurity Measure.....   | - 19 -       |
| 2.3 Convolution Neural Networks.....                                | - 21 -       |
| 2.3.1 ConvNet Architecture and Layers.....                          | - 21 -       |

|          |  |               |
|----------|--|---------------|
| 2.3.2    | Convolutional layer .....                            | - 22 -        |
| 2.3.3    | Activation features .....                            | - 25 -        |
| 2.3.4    | Pooling layer .....                                  | - 26 -        |
| 2.3.5    | Fully connected layer .....                          | - 27 -        |
| 2.3.6    | ConvNet Training .....                               | - 27 -        |
| 2.3.7    | Backpropagation and update parameter .....           | - 28 -        |
| 2.3.8    | Regularization .....                                 | - 29 -        |
| 2.3.9    | Batch normalization .....                            | - 31 -        |
| <b>3</b> | <b>Chapter: Related Works .....</b>                  | <b>- 32 -</b> |
| 3.1      | Automatic Diagnosis .....                            | - 32 -        |
| 3.2      | Image Retrieval using Machine Learning .....         | - 35 -        |
| 3.3      | Machine Learning Malaria Diagnosis .....             | - 37 -        |
| 3.4      | Deep learning and convolutional neural network ..... | - 38 -        |
| 3.5      | Comparing ConvNet with other Methods .....           | - 38 -        |
| 3.6      | Challenges to CNN applications .....                 | - 39 -        |
| <b>4</b> | <b>Chapter: Proposed Method .....</b>                | <b>- 41 -</b> |
| 4.1      | Proposed Method .....                                | - 41 -        |
| 4.1.1    | Convolutional Layer .....                            | - 43 -        |
| 4.1.2    | Pooling layer .....                                  | - 44 -        |
| 4.1.3    | Activation Layer .....                               | - 44 -        |
| 4.1.4    | Output layers .....                                  | - 46 -        |
| 4.1.5    | Model Optimization and fine-tuning .....             | - 47 -        |
| 4.1.6    | Convolutional Neural Network Architecture .....      | - 53 -        |
| <b>5</b> | <b>Chapter: Evaluation .....</b>                     | <b>- 54 -</b> |
| 5.1      | Our Training Dataset .....                           | - 54 -        |
| 5.2      | Experimental Environment .....                       | - 54 -        |
| 5.3      | Support Vector Machine .....                         | - 54 -        |
| 5.3.1    | Methodology .....                                    | - 54 -        |
| 5.3.2    | Visualization of histogram .....                     | - 56 -        |
| 5.3.3    | Model Evaluation .....                               | - 56 -        |
| 5.3.4    | Support Vector Machine Tuning Hyperparameter .....   | - 57 -        |
| 5.3.5    | Model Performance .....                              | - 58 -        |

|          |  |               |
|----------|--|---------------|
| 5.3.6    | Support Vector Machine Algorithm Performance .....               | - 58 -        |
| 5.4      | Random Forest.....   | - 61 -        |
| 5.4.1    | Random Forest Performance.....                                   | - 62 -        |
| 5.4.2    | Entropy .....  | - 63 -        |
| 5.5      | Convolutional Neural Network.....                                | - 66 -        |
| 5.5.1    | The Training model configurations .....                          | - 69 -        |
| 5.5.2    | CNN Algorithm Performance .....                                  | - 72 -        |
| <b>6</b> | <b>Chapter: Conclusion.....</b>                                  | <b>- 74 -</b> |
| <b>7</b> | <b>Bibliography .....</b>  | <b>- 76 -</b> |
| <b>8</b> | <b>Appendix.....</b>   | <b>- 81 -</b> |
| 8.1      | SVM turning color images to grey scale .....                     | - 81 -        |
| 8.2      | CNN configurations: Adam, Sigmoid and binary cross entropy ..... | - 81 -        |
| 8.3      | 10 – fold cross validation .....                                 | - 82 -        |

## List of figures

|  |               |
|--|---------------|
| <i>Figure 1 Support Vector Machine Hyperplane.....</i>         | <i>- 9 -</i>  |
| <i>Figure 2 Hyperplane and support vectors.....</i>            | <i>- 10 -</i> |
| <i>Figure 3 Linear Classification using SVM.....</i>           | <i>- 11 -</i> |
| <i>Figure 4 SVM Margins.....</i>                               | <i>- 12 -</i> |
| <i>Figure 5 Geometric Shapes Classes .....</i>                 | <i>- 14 -</i> |
| <i>Figure 6 Decision Based on Geometric shapes .....</i>       | <i>- 15 -</i> |
| <i>Figure 7 Activation map.....</i>                            | <i>- 23 -</i> |
| <i>Figure 8 Convolution Layers.....</i>                        | <i>- 24 -</i> |
| <i>Figure 9 Convolution Filters .....</i>                      | <i>- 25 -</i> |
| <i>Figure 10 Filter with 2x2 max filter.....</i>               | <i>- 27 -</i> |
| <i>Figure 11 Neural Network generalization.....</i>            | <i>- 31 -</i> |
| <i>Figure 12 Neuron.....</i>                                   | <i>- 42 -</i> |
| <i>Figure 13 Back Propagation using Gradient Descent .....</i> | <i>- 42 -</i> |
| <i>Figure 14 Feedforward and Backpropagation .....</i>         | <i>- 42 -</i> |
| <i>Figure 15 Convolution layers with activation maps .....</i> | <i>- 44 -</i> |
| <i>Figure 16 Downsampling .....</i>                            | <i>- 44 -</i> |
| <i>Figure 17 Three Activation functions.....</i>               | <i>- 46 -</i> |
| <i>Figure 18 Softmax classifier .....</i>                      | <i>- 47 -</i> |
| <i>Figure 19 Overfitting and Underfitting Graph .....</i>      | <i>- 48 -</i> |
| <i>Figure 20 Different SGD Performance .....</i>               | <i>- 53 -</i> |
| <i>Figure 21 CNN Architecture .....</i>                        | <i>- 53 -</i> |

|   |               |
|---|---------------|
| <i>Figure 22 Blood smear cell images.....</i>                   | <i>- 54 -</i> |
| <i>Figure 23 SVM Process.....</i>                               | <i>- 55 -</i> |
| <i>Figure 24 Greyscale images .....</i>                         | <i>- 55 -</i> |
| <i>Figure 25 Data Visualization .....</i>                       | <i>- 56 -</i> |
| <i>Figure 26 Visualization .....</i>                            | <i>- 56 -</i> |
| <i>Figure 27 SVM Graphical Representation .....</i>             | <i>- 57 -</i> |
| <i>Figure 28 Variance .....</i>                                 | <i>- 57 -</i> |
| <i>Figure 29 RF and SVM Performance Comparison.....</i>         | <i>- 58 -</i> |
| <i>Figure 30 Algorithm Performance .....</i>                    | <i>- 58 -</i> |
| <i>Figure 31 Accuracy and Performance .....</i>                 | <i>- 59 -</i> |
| <i>Figure 32 Gamma.....</i>                                     | <i>- 59 -</i> |
| <i>Figure 33 Accuracy heatmap .....</i>                         | <i>- 59 -</i> |
| <i>Figure 34 Support vectors and hyperplane .....</i>           | <i>- 60 -</i> |
| <i>Figure 35 Random forest implementation .....</i>             | <i>- 62 -</i> |
| <i>Figure 36 RF 10-fold.....</i>                                | <i>- 62 -</i> |
| <i>Figure 37 Representation of the splitting mechanism.....</i> | <i>- 65 -</i> |
| <i>Figure 38 Entropy against training samples .....</i>         | <i>- 66 -</i> |
| <i>Figure 39 Validation curve.....</i>                          | <i>- 66 -</i> |
| <i>Figure 40 Training Dataset.....</i>                          | <i>- 67 -</i> |
| <i>Figure 41 Malaria Cell images.....</i>                       | <i>- 69 -</i> |
| <i>Figure 42 CNN 30 Epochs.....</i>                             | <i>- 71 -</i> |
| <i>Figure 43 CNN Performance Graphs .....</i>                   | <i>- 72 -</i> |
| <i>Figure 44 CNN Algorithm Performance .....</i>                | <i>- 73 -</i> |
| <i>Figure 45 CNN Heatmap.....</i>                               | <i>- 73 -</i> |

## List of Tables

|   |               |
|---|---------------|
| <i>Table 1 Blood smear images.....</i>                    | <i>- 6 -</i>  |
| <i>Table 2 SVM and RF Accuracy Results.....</i>           | <i>- 58 -</i> |
| <i>Table 3 Confusion matrix .....</i>                     | <i>- 62 -</i> |
| <i>Table 4 Recall, Precision, F1-score 10 K-fold.....</i> | <i>- 63 -</i> |
| <i>Table 5 CNN Model configuration.....</i>               | <i>- 70 -</i> |

## 1. Chapter: Introduction

Malaria disease is caused by the mosquito bite and it is commonly found in warm regions of Africa and other countries like Asia and the United States of America. The female anopheles mosquito carries the parasites that are transmitted to the human circulatory system through the bites from these infected mosquitoes. Once the parasites enter the circulatory, they go to the human liver where they multiply and spread through the whole circulatory system causing fever and headaches. The research that was conducted in 2015 by WHO revealed, an estimated 214 million malaria cases were reported globally with 438,000 deaths caused by malaria[1], [2].

In this paper, our case study focuses on Malawi where malaria is also one of the major health concerns with an estimate of 4 million cases being reported yearly. The vulnerable citizens are mostly children who are below five years of birth and pregnant women. The Ministry of Health in collaboration with its partners came up with the Malawi Health Sector Strategic Plan that is being implemented from 2017 - 2022. The strategic plan articulates the importance of developing the health sector and one of the priorities being the eradication of malaria, "*malaria-free Malawi*". The current efforts like village clinic services, environmental management and the use of insecticides are part of the early interventions being implemented[3]–[5].

Malaria microscopy is the current method used to discover plasmodium using blood films. The microscopic malaria detection method is the approved platform by WHO worldwide. The method involves collecting blood through finger pricking and spreading it on the clean, grease-free plate glass and leave it to be air-dried. Later the dried blood smears have to be put on the microscope to check for parasites presence, to identify the species type in the thin blood smears can be utilized but where there is a low presence of parasites it is recommended to use the thick smear whilst the thick smears are used for checking the presence of the parasites. The quality and accuracy of this work depend on the skills and experience of the microscopists. The process requires someone who is highly skilled to

bring out accurate results and it is time -consuming to quantify the parasites present in the blood smears[6]. This research paper, suggests the use of one of the machine learning techniques that is utilized in performing image classification to improve accuracy in the process of malaria detection, Convolutional Neural Networks (CNN).

The current microscopy method for malaria detection is done manually hence time-consuming to complete the diagnosis process. Based on the WHO report the period to train technicians takes about 4-5 weeks. The process of preparing the blood smears takes about thirty minutes which involves the extension and the staining process. Based on the density of the parasitemia, it takes two to five minutes to search and find the malaria plasmodium parasite. Furthermore, it is also difficult to conduct periodic training for technicians based in rural areas due to poor road networks and poor infrastructures[7], [8].

Based on the above overview, it is essential to execute a technique that will be able to automatically detect malaria using the unsupervised method to overcome the current technical malaria diagnosis challenges. The conducted previous research mainly in the area of image processing has facilitated the implementation of this research paper. The image processing techniques will be leveraged to analyze the blood smear films for the malaria detection technique.

This research paper focuses on exploring the algorithm that can be utilized for images processing in the area of machine learning used for deep learning, Convolution Neural Network have been used in many research fields and has ended up to effective when dealing with image classification. Furthermore, this paper also looked at other algorithms that have been utilized for image classification for the results comparison sake in terms of performance and accuracy. The main aim of the research is to check the performance of CNN that has been implemented in other different areas that can be broadened to solve the malaria detection challenges.

African Bible College hospital has been very helpful in the provision of information regarding the malaria diagnosis process and the National Library of Medicine for providing the well-segmented malaria cells image dataset that has been utilized for the implementation of the project. In particular, we also recognize the effort and support that

was provided by some of the medical practitioners who played a vital in providing necessary information during our research.

### **1.1 Motivation**

The project research was developed as a requirement of the final years' academic thesis of Master in Software Engineering. The researcher proposed the idea and received support and guidance from the supervising professor for directions on how to implement a malaria detection project based on cell images. Some initiatives currently are being implemented by the government which resides under the early intervention methods in the battle against malaria, such methods include distribution of insect sides treated nets, environmental care, and rural health care centers. The interventions led to a decrease in malaria prevalence from 33% to 24% as reported by the Malawi malaria indicator survey 2017.

Based on the World Health report 2019, Malawi is the first country in Africa to implement the malaria vaccine pilot project and this was announced during the World malaria day 2019. The vaccine pilot project aims to immunize children in contributing to the battle towards malaria. In support, the battle to eliminate malaria an automated technique for the diagnosis process is ideal to contribute to the idea of zero malaria Malawi[9]. The interest in the project is one way of expanding knowledge in machine learning and the opportunity to learn more in the medical field.

### **1.2 Objectives**

The main objective of implementing this project is to come up with an automatic system that can differentiate the malaria-infected and uninfected cell images using the machine learning algorithms; Convolutional neural networks, random forest, and support vector machine. Bearing in mind the success of the thesis we knew that there is a need to conduct in-depth research to achieve our goal. We researched the type of data that we will need for implementation and information about malaria and methods used for diagnosis. Lastly, we had to find methods on how we are going to implement our project and this led to the need for expanding our understanding of machine learning algorithms. Therefore, the below-listed goals for our project will facilitate our implementation;



- Learn different techniques about image processing that have been discovered for detecting malaria and find the appropriate technique to be used in our project.
- Research deep learning and focus on Convolutional Neural Networks which will be leveraged in our implementation.
- Evaluate the results of each technique based on performance and results.

The project implementation is mainly based on the malaria detection process using images. Some different approaches and methods are utilized for the detection of malaria and much information has been made available on the internet by different researchers. In our project, we used the thin blood smear images because they found out that they were appropriate for our implementation based on the previous medical researchers. In our experiment, we will be dealing with image datasets composed of 28,000 blood smear images. During the training process, CNN gave the best accuracy compared to other algorithms because of its capability when dealing with large image datasets.

Different research articles have been published based on the automatic detection process but our research differs items of dataset size. We also noticed that other methods mainly focused on the tools used for processing like the support vector machine segmentation technique. In our research, we explored all the other techniques that have been implemented before and evaluated their performance-based accuracy. Our work shares some similarities with some of the implemented techniques but our methods are outstanding based on accuracy. Even though CNN took more time to understand the concepts and technique but it has proved to be the best with image processing. Our implementation used libraries and there is not much information regarding how we achieved the significance in terms of our results.

Finally, our research project focused on malaria detection process using the thin blood smear images. We wanted to come up with a technique that can be able to analyze, identify, and categorize them as healthy (infected) or unhealthy (uninfected). Further work on this research can be done on identifying parasite species.

### 1.3 Structure of Document

This research paper has been structured into six chapters, the first chapter gives an overview of malaria disease, the preparation process of the blood smears, and machine learning algorithms overview. In the second chapter, we will discuss some of the related machine learning algorithms that have been used in this thesis, SVM, RF, and CNN. The third chapter looks at the background and scope of the project, the fourth chapter discusses the implementation of the project, the fifth chapter explains the results obtained from the implementation and finally, the sixth chapter looks and the evaluation and conclusion.

### 1.4 Related Work

In this chapter, we will briefly have an overview of malaria and the current methods being used for diagnosis. The process of preparing blood samples to be diagnosed will also be discussed. Further, we will conclude with the microscopy blood samples scanning methods.

Malaria is a disease that transferred by the anopheles mosquito which carries the malaria parasites called plasmodium and this is transferred to humans through a mosquito bite that injects the protists into the human blood circulatory system. Thereafter parasite resides in the human liver where it multiplies and spread to different parts of the body. The common signs of infection are; headache, fever, and abdominal pain, delay in medical attention result in coma or loss of life[10].

There are five different types of malaria named based on the parasite species;



- Plasmodium falciparum – the common parasite in Africa mainly in the sub-Sahara. Some cases are being reported in other regions where the parasite was eradicated.
- Plasmodium vivax - This is a mild disease, the parasites can reside in the liver for years without causing health problems. The P.vivax is the widest spread parasite and commonly found in India which causes 60% of the infections.
- Plasmodium malariae - The P.malariae is also mild because the parasite can stay in the body for a longer period without causing any health problems, but if the infection stays untreated it may lead to health problems.
- Plasmodium ovale - milder can stay in the body for decades but if it remains untreated may cause health problems.

- Plasmodium knowlesi - P.knowlesi can cause infection in humans and macaques.

### 1.5 The diagnosis process

There different methods that are currently used for malaria diagnosis, we will have a brief overview of the three methods which include; clinical diagnosis methods, microscopy diagnosis, and rapid diagnostic tests but based on the WHO report (2009) only two diagnosis methods are recommended. But before we dive into the diagnosis methods let have an overview of how blood samples are collected and processed[11], [12].

- Blood smear preparation process - the blood smear preparation process has standardized by the Special Programme for Research and Training in Tropical Diseases (TDR) which is an informal consultative. The process of preparing blood smear starts with collecting blood samples from the patient by finger pricking or vein puncture (venepuncture). The blood samples should then be smeared on a clean, grease-free glass slide, labeling of the sliding glass should include; date, identification code, and objective of the collected samples. The thick blood smears that are well prepared contain white blood cells of about 10 - 15 on the average range which is referred to us a high power field (HPF). The thin blood smear should process the same way using the unchipped edge of a slide to have a monolayer of the blood cells. Then later the collected blood samples should be left to dry, ensure that the drying environment free from dust any other objects that can fall on the samples as this may affect the test results. The image below illustrates the thin and thick smear blood sample slide. The thick blood smear s are used to determine parasites' presence and thin smears used to identify species of plasmodium[13].

|                          |  |
|--------------------------|--|
| <i>Thin blood smear</i>  |   |
| <i>Thick blood smear</i> |  |

*Table 1 Blood smear images*

## 1.6 Diagnosis methods

- Clinical diagnosis methods - the method is to diagnose patients through physical examination based on the following symptoms; vomiting, fever, headache, abdominal pains, and dizziness. This method is cheap and does not require a lot of medical utensils but the challenging part is the overlapping of symptoms and requires very skilled medical personnel. However, the symptoms used for malaria diagnosis are also common in other life-threatening diseases and may lead to wrong medical prescriptions[6], [13].
- Rapid diagnosis test - approved by WHO, RDT is a malaria diagnosis method which easy and cheap. This method is effective in areas where there are electricity challenges and there no requirement of any other medical utensils. The challenge with this method is that some of the products are designed targeting specific species of malaria parasites and the commonly used the one used to diagnose p.falciparum. RDT method is suitable for remote areas (rural areas) where electricity is a major problem and requires general knowledge to use the product. Despite the benefits, the methods also need ways to confirm the obtained results[7], [13].
- Microscopy diagnosis - microscopy diagnosis is the common and widely used method to diagnose malaria. The method involves collecting blood samples from a finger prick and blood drops are put on the glass slide where they are left to dry for approximately 20 minutes. The glass slide is then later put under the microscope to check for the presence of parasites if it is the thick smear slide glass and identify the parasite species if it is the thin smear slide glass. The microscopic method has proved to be the most effective method to diagnose malaria. However, the process is time-consuming and requires skilled medical personnel to identify the species. The method is not suitable for rural areas simply because of the lack of reliable electricity supply[13].

## 2 Chapter: Related Technologies

In this chapter, we will explain the technologies that have been used in this project and their characteristics. The technologies explained below are all related to the field of machine learning, mainly with classification. Firstly, we will have an overlook at support vector machines, this is mainly used for classification of dataset composed of two classes. Secondly, we will have an overview of random forest, this is an ensemble learning technique used for classification and regression. The algorithm works by constructing multiple decision trees based on the output class of classes. Finally, we will discuss the convolutional neural network, this is a deep learning technique and has shown great performance in the field of computer vision hence a good opportunity to be taken into consideration to improve malaria diagnosis performance in the medical field.

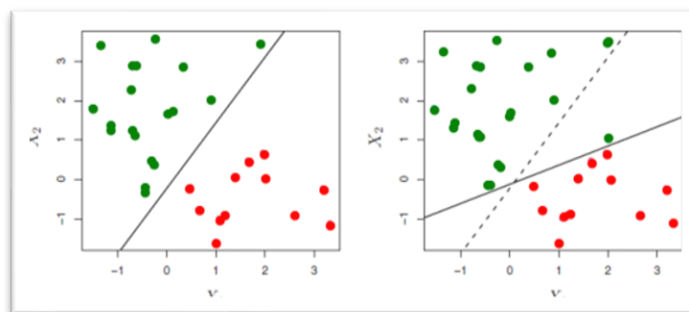
### 2.1 Support Vector Machines

This is one of the machine learning algorithm mainly used for classification, the algorithms learn by categorizing data into two categories. Assuming there is a dataset that has two classes, the support vector machine will learn from the data and categorize it into one class or the other. Data in support vector machine is categorized into n-dimensional space or data is represented in the form of a 2D map. We will discuss the support vector machine classifier and then the actual concept of SVM[14].

#### 2.1.1 Support Vector Classifier

The concept of support vector machine mainly deals with a linear classification process which involves the use of the hyperplane to create a decision boundary. However, there are some conditions where the hyperplane may not be able to separate the two classes. There are instances where a classifier based on the hyperplane might not be needed. To perfectly classify all the observations from the training, it is recommended to use a classifier that is based on the hyperplane which leads to observation sensitivity. The figure below shows observations based on two classes, green and red. The figure on the left side indicates two classes that have been separated by the maximal margin hyperplane. In the

next figure on the right side, a green observation has been added leading to the drastic change in the maximal marginal hyperplane which is represented in a solid line and the dashed line represents the first marginal hyperplane before the adding the green observation[15], [16].

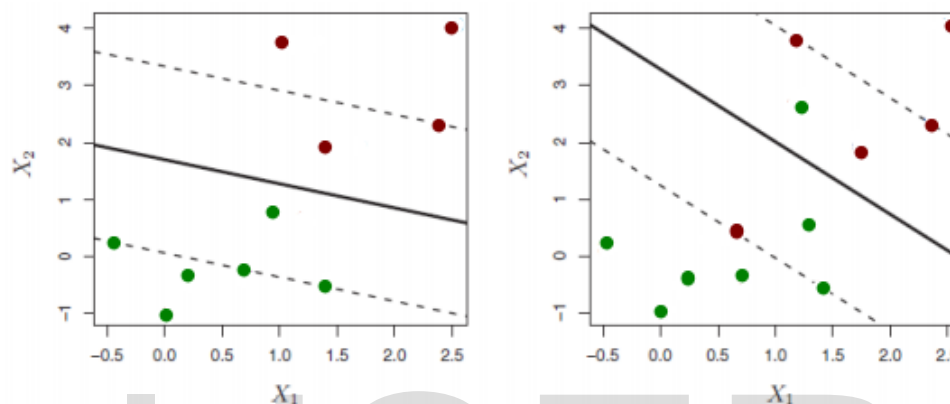


*Figure 1 Support Vector Machine Hyperplane*

The maximal margin hyperplane above has a small margin based on the resulting hence not satisfactory. This is can be supported with a statement that distance from the observation is a measure of confidence that the observation has been classified correctly. In addition to that, the hyperplane is sensitive meaning that any change in the current observation will lead to the training observation being overfitting. This will take us to put into consideration a hyperplane that is not separating the classes, this will assist us to obtain a good classification from our training observation as we gain great robustness. This is to say that to achieve better classification results it is important to allow some misclassification for a few training samples during the classification process[15], [17].

To achieve this task we suggest using the support vector classifier which is also called the soft margin classifier. The classifier can achieve this by defining a tube that will contain the hyperplane which is used to the edge that separates the classes and the 2 margins. This distance from the margin hyperplane and the two margins are used to give a precise amount of observations. This is called the soft margin because it ignores some of the observations during training. Instead of allowing the observations to fit correctly on one region of the hyperplane, alternatively, we accept that some of the observations to be incorrectly classified by leaving them in the wrong region of the hyperplane[16], [17].

The figure below indicates the support vector machine illustration, the hyperplane is represented as in the form of a solid line and the margins dashed lines. The illustration of the left side figure shows that all the observations are on the correct side of the margin hyperplane and only two observations have crossed the soft margin but have been classified correctly. The right margin indicates that some observations have been wrongly classified because they are on the wrong side of the hyperplane.



*Figure 2 Hyperplane and support vectors*

The technique of discovering the tube together with the hyperplane and the margins are the alternative to reducing the problem that can well be expressed using the Lagrange multipliers to include the desired limitation. The processing of this kind of optimization involves the key parameters, the tube width, and the budget[17].

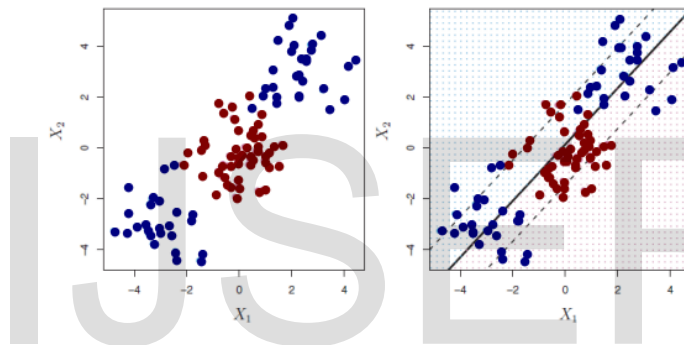
The budget simply means the amount of violated margin that can be made on the observations and this looks at the width. The increase in the budget results in isolation tolerance and widening the margin. Conversely, the decrease in the budget results in the reduction in margin violation making it narrower. Practically this is considered to be the tuning parameter for cross-validation.

This leads us to the theory of optimization which has the following property: the observation that has violated the margin affect the classifier or observations that are on the correct side of the classifier do not affect the support vector classifier even if they move to a different position so long as they remain on the same side. The observations that are on the wrong side or within the margin are called the support vectors and these affect the

support vector classifier. The only thing that distinguishes support vector classifiers from other classifiers is that it based on the rule of a small subset of observations from the training resulting in robustness in its behavior[17].

### 2.1.2 Support Vector Machine

The process of classifying two classes can be implemented better using the support vector classifier taking into consideration that the boundary of the two categories is linearly separated. However, there are some situations when the observations have non-linear class boundaries. For example, the left figure below illustrates the data that any linear classifier will not perform better and the right classifier shows the support vector classifier implementation whose performance is poor.



*Figure 3 Linear Classification using SVM*

The outcome of the result illustrates that the linear classification did not perform better for the nonlinear observations. The best alternative is to put into consideration predictors' functions which will assist in addressing the non-linearity from our observations. This task can be implemented better using the support vector machine which is part of the support vector classifier. The support vector machine enlarges the space between the features with the support of the kernels[15], [17].

Now, let us have a look at how the support vector operates assuming that we have a situation where our observations are linearly separable. Our data  $x_1 \dots x_n$  has the following targets that are corresponding,  $t_1 \dots t_n$  with  $t_1 \in \{-1, 1\}$ . Now let's consider the classification rule to be of this form:

$$h(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



$$= \text{sign}(y(x))$$

$$y(\mathbf{x}) = \mathbf{w} \mathbf{x} + b$$

Now let us re-scale  $(\mathbf{w}, b)$ , will do this without any changes to the decision boundary. Let's select  $(\mathbf{w}, b)$  to ensure training points that are close to the boundary should satisfy the following,  $y(x) = \pm 1$ . See the figure below on the left-hand side. Let's say,  $x_1$  the close point with class  $t_1 = -1$  that  $\mathbf{w} x_1 + b = -1$  and  $x_2$  the close point with class  $t_2 = 1$  that  $\mathbf{w} x_2 + b = 1$

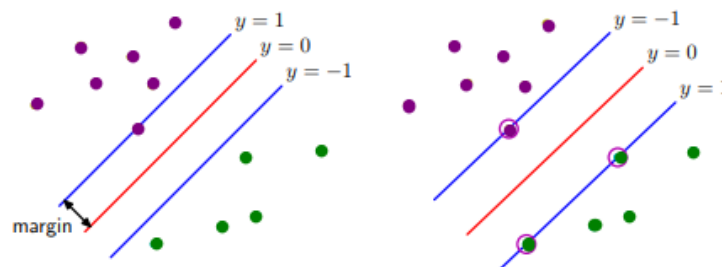


Figure 4 SVM Margins

The distance from the red line (hyperplane) and the blue line is called the margin. This distance from the decision boundary is determined by the closest observations as indicated in the figure on the left side. Likewise, widening the margin will also lead to a change in the decision boundary as indicated by the left side figure. Now let's have a look at the geometrical calculation used to maximize the margin.

Assuming we have point  $\mathbf{x}$  as our perpendicular distance based on the hyperplane;  $\mathbf{w} \mathbf{x} + b = 0$ , which is obtained from  $|\mathbf{w} \mathbf{x} + b|/\|\mathbf{w}\|$ , the distance from the all close points of each class can be classified as  $1/\|\mathbf{w}\|$ . Support vector machine mainly search for the maximum margin classifier that can be used to separate the data and this can also be expressed using statistical theory where margin increase,  $1/\|\mathbf{w}\|$  is also the same as reducing  $f(\mathbf{w}) := \frac{1}{2} \mathbf{w}^T \mathbf{w}$ . This will lead us to a situation where we will be able to obtain the primal problem which is a quadratic program.

$\min_{\mathbf{w}, b} f(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$  and this is subjected to:  $t_i (\mathbf{w} x_i + b) \geq 1, i = 1, \dots, n$  this is to ensure that all the observations have been classified correctly.

Therefore, to overgeneralize our discussion we have observed that the classification process of the support vector classifier is based on the actual inner observations. However, if there happens to be a product appearing in the inner observation, we take this product and place it in the inner products that have already been generalized. This will be in the form of  $K(x_i, x_i)$ ,  $K$  represents the kernel function. This process will enable us to increase the feature space and it is known to be the trick of the kernel.

The kernel is responsible for the quantification of the observations based on their similarity. The fact that the support vector classifier is linear therefore its' kernel is also called a linear kernel. There are some of the options which are viable for the function of the kernel  $K$ . An instance that is within the inner product can be substituted by polynomial based on the certain degree, also called the polynomial kernel of degree which aims at fitting the support vector classifier in a higher dimensional area that includes degree,  $d$  polynomials. The other common choice is the use of Gaussian function mainly used for computing the space between the two observations, this is also called the radial basic-function kernel.

The use of Kernels offer benefits in reducing the computational procedures because it only performs computation on values of  $K(x_i, x_i)$  for dissimilar two observations. This task can be achieved without explicitly increasing the feature area. This offers good benefits because support vector machine applications enlargement feature space is huge resulting in computational intractability. As for the radial kernel computational will be not possible because of feature area which is implicit and infinite-dimensional. When implementing the RBF kernel the width size of Gaussian should be selected using cross-validation.

## 2.2 Random Forest

The random forest can also be referred to as a random decision forest, this is a machine learning technique that is mainly used for classification and regression. This ensemble learning technique works by developing several decision trees during the training process and the output results will be the classes of a particular prediction based on different trees. Before going further into more details about the random decision forest, let

us have a brief overview of decision trees as this will assist us to understand better how random forest operates[18].

### 2.2.1 Decision Trees

This is a method that is widely used to perform the classification and regression process. The methods operate by splitting the data samples at every node basing on the feature values. The splitting process works with the sample to make sure that the end leaf samples have some similarities. During the testing period, the samples go down the tree following the specified splitting rules[19]. At the final leaf node, classification is performed on the samples that were trained. Let us move further explaining more details about the classification process.

### 2.2.2 Classification

The classification is the process that involves identifying patterns from non-class data samples and group them to their respective class. Based on the specified values, the training sample goes down either on the left side or right side of the sub-nodes and every leaf node has a specified class. This process continues until it reaches the final leaf nodes[20]. Let us have a look at a simple class example using shapes.

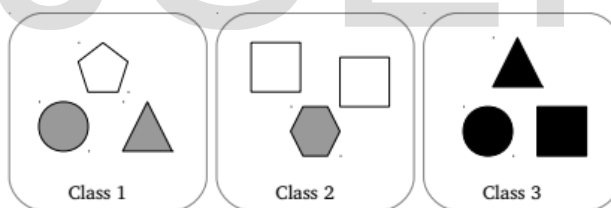


Figure 5 Geometric Shapes Classes

The figure above shows three classes of different geometric shapes which we will use for exemplary as our training data. The testing parameters used are the number of corners which is zero to six and the color brightness where black is equal 0, grey is equal to 0.5 and white equal to 1. The triangle vector from the first class can be represented in the following manner, [3, 0, 5]. The assignment process involves the splitting of nodes whilst descending from the root node to the child nodes where the training dataset is classified into a particular.

### 2.2.3 Training of decision trees

The process of splitting the node is done by the machine, but the main thought is the splitting of the training dataset at every node. The splitting of nodes will result in classifying the node to a specific class and should be pure. Suppose we have  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$  represents sample data that is to be trained. Every sample is composed of  $d$ - dimensional of vector  $x = (\varphi_1, \varphi_d)$ ,  $\varphi_1 \in \mathbb{R}$  with the distinct class with the label  $y \in C$ ,  $C$  represents each the class labels. However, to simplify the details, each  $S$  is used to classify  $S^x$  and  $S^y$  in the following way:

$$S^x = \{x | (x, y) \in S\}$$

$$S^y = \{y | (x, y) \in S\}$$

Where the split of  $x = (i, \tau)$  with the following element index  $i \in (1, \dots, d)$  with the threshold  $\tau \in \mathbb{R}$ . If the  $x$  is applied to the test set  $S$  this lead to the division of  $(S_l, S_r)$  of  $S$  which lead to

$$S_l = \{(x, y) | (x, y) \in S, x = (\varphi_1, \dots, \varphi_d), \varphi_i < \tau\}$$

$$S_r = \{(x, y) | (x, y) \in S, x = (\varphi_1, \dots, \varphi_d), \varphi_i \geq \tau\}$$

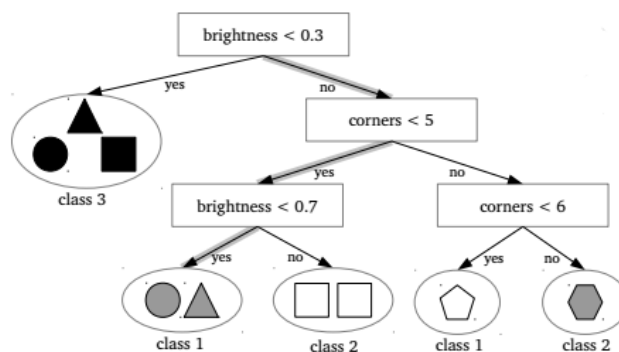


Figure 6 Decision Based on Geometric shapes

Therefore,  $s_l \cup s_r = s$  with  $s_l \cap s_r = \emptyset$ . They represent every sample with  $\varphi_i < \tau$  move the left-hand side and the rest to the right-hand side. Beginning from the tree root node a unique figure of randomly selected split  $x$  is then tested to  $S$  and then a scoring function is applied to give a score  $f_{score}(S, x)$ . Then  $X$  represents the tested splits. Therefore,  $x^*$  represents the maximizing split  $f_{score} x^* = \arg\max_{x \in X} f_{score}(S, x)$ .

The value  $x^*$  splits  $S$  into  $S_l$  and  $S_r$ . The node stores the value and  $S_l$  and  $S_r$  are moved to the left child node and right child node, then the process of splitting recursively goes on. The number of splits and how they are selected all rely on the actual implementation. The splitting process continues until a given criterion is met then that final node becomes a leaf node because there is no further splitting of the node.

#### 2.2.4 Splitting function

The split scoring function is used to quantify the data samples by checking their similarities before splitting the node into subsets. The commonly used splitting function is information gain. The information gain based on node splitting is referred to as a decrease in ambiguity attained through the splitting of training data at a node into various sub-nodes[20]. The equation below illustrates the binary split:

$$I(S, S_l, S_r) = H(S) - \sum_{i=l,r} \frac{S_i}{S} H(S_i)$$

In the above equation  $H(S)$  represents the entropy set  $S$ . The entropy is defined based on the task. When performing the classification task it is recommended to use the Shannon entropy:

$$H(S) = - \sum_{y \in c} p(y) \log(p(y))$$

#### 2.2.5 Variations

There different ways on how a decision tree can be expanded and below are some of the methods that can be used to expand the decision tree based on the simple example illustrated above.

- i. Including extra features in the splitting process and use the hyperplane to implement the splitting of the samples.
- ii. Using the leaf to keep all the distribution of all leaf samples therefore including the uncertainty into consideration.
- iii. Replacement of the information-gain score when implementing the splitting process with another scoring function that will suit the challenged being solved.

### **2.2.6 Random Forest**

The implementation of decision trees posed some challenges as the model goes deeper which results in overfitting. The merging of all the trees forming a forest has proved to be the best generalization technique compared to one tree. This true based on the argument that accuracy improves when the number of trees is added to the forest. When there is some modification during the implementation process either predictions the outcomes are straight forward[18], [21].

- i. Training - Based on the above explanation, ensemble training involves the training of trees individually. During the training process, the bagging technique is used replacing the process of training every tree. This helps to minimize the correlating among the trees and has generalized better. The placement technique makes a sample appear several instances resulting in obtaining higher weight. This technique can be explained as giving random weights to the samples during training.
- ii. Prediction - to obtain the prediction based on the data position  $x$ , every tree is applied to  $x$  separately. Assuming the forest has  $T$  trees, this leads to  $T$  separate prediction to  $x$ . The final classification of the forest is based on the class which has the highest votes from the trees. If the final output is a probability distribution, computation is performed on output distribution by multiplication or summing. The computational process can be speedup through parallel training of the trees.

### **2.2.7 Regression Forest**

We have discussed random forest which based on classification and now let us have an overview of the regression forest[22]. To implement the regression forest we just need to make little modification so that our implementation method should be capable to implement the regression challenges. The classification and

regression forest differ in their target prediction values, from discrete class  $y \in \mathcal{C}$  to a value that is continuous  $\in \mathbb{R}$ . The change from classification to regression does not change the structure of the tree but the splitting function and the leaf node data change.

- i. Prediction - to obtain the prediction based on the new data-position  $x$ , this position moved across the trees within the forest in the same way when implementing classification. The variation is shown at the leaf node, in place of a separate class label, the leaf node keeps the likelihood of mass dispense over continuous variable  $y$ . The Gaussians could be a simple example even though difficult distribution is also feasible. The results of the  $t$ -th tree that are used for the input value  $x$  is equivalent to the distribution  $p_t(y|x)$ , the final results of the forest are gained through averaging results from individual trees. The forest absolute output is the average of  $T$  trees:

$$p(y|x) = \frac{1}{T} \sum_{t=1}^T p_t(y|x)$$

- ii. Training - The technique to implement regression to gain the best split differs from classification mainly on the tree nodes. The main idea to maximize the information gain and the entropy must be adjusted, for example, if we have the Gaussian allocation of data, entropy can be explained in the following way;

$$H(S) = \frac{1}{|S|} \sum_{x \in S^x} \frac{1}{2} ((2\pi e)^2 \sigma_y^2(x))$$

The conditional variance is represented with  $\sigma_y^2(x)$  this used to calculate linear probability. Now adding this equation to the binary fit equation expanding to multiple dimensional inputs the regression for the information gain looks like this: note that  $\Lambda_y(x)$  represents the conditional variance and  $H(S)$  variance of data

in  $S^y$ . However, when implementing multiple dimensional regression the dimension uses the average variance.

$$I(S, S_l, S_r) = \sum_{x \in S^x} \log(|\Lambda_y(x)|) - \sum_{i=l,r} \left( \sum_{x \in S_i^x} \log(|\Lambda_y(x)|) \right)$$

### 2.2.8 Impurity Measure

- i. Gain Ratio - this is a modern edition of the Information Gain and it was initiated Interactive Dichotomiser 3 abbreviated ID3. The term Information Gain refers to the process of measuring how much gain is in entropy. Information theory states, when there is high data distraction the entropy is high, entropy is simply the measure of information. Based on our research paper, we have 28,000 observations which are composed of unhealthy and healthy cell images. In the process of performing our classification using a random forest classifier, before splitting the data into different classes there was a higher diversion which resulted in higher entropy. The main focus is to reduce the diversion which will result in lowering the entropy and obtaining purer classes, this can be achieved by splitting the  $D$  - distribution. Through the splitting process, the class that has the lowest diversion has attained the highest gain in lowering the entropy which is referred to as Information Gain[23], [24]. The entropy for data division ( $D$ ) can be computed as follows before splitting.

$$Entropy(D) = - \sum_{j=1}^{|C|} p(c_j, D) \times \log_2(p(c_j, D))$$

Now, using the equation above we can explain how the entropy of data segmentation is computed. Let us illustrate this using 30 observations in  $D$  ( $|D| = 30$ ). assuming within our observation we have two classes we have two observations  $g_1$  and  $g_2$  in  $G$  ( $|G| = 2$ ), based on this example we 14 samples belong to the class  $g_1$  and 16 samples belong to the class  $g_2$ . Therefore,  $j = 1, p(g_1, D)$  computes the probability of  $g_1$  in dimensional  $D$ ,  $\frac{14}{30} = 0.47$  and for the second  $g_2, j = 2, p(g_2, D)$  which will result in  $\frac{16}{30}$



= 0.53 and the entropy is equal to 0.9974. The splitting can further be performed on the data D to S partitions non-attribute class  $A_i$  resulting in bringing together entropy from all partitions and the equation will like this;

$$Entropy(A_i, D) = \sum_{i=1}^s \frac{|D_i|}{|D|} \times Entropy(D_i)$$

Therefore, we can compute the attribute information gain in the following way.

$$InformationGain(A_i, D) = Entropy(D) - Entropy(A_i, D)$$

The disadvantage of Information Gain is that it treats unfairly attributes containing domains with higher values which may lead to increased fragmentation level causing overfitting. The Information Gain is corrected through implementing Gain Ratio which operates by splitting information and below is the equation used to calculate Gain Ratio.

$$GainRatio(A_i, D) = \frac{InformationGain(A_i, D)}{SplitInformation(A_i, D)}$$

Therefore, Split for  $A_i$  can be computed using the following equation below;

$$SplitInformation(A_i, D) = \sum_{i=1}^s \frac{|D_i|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right)$$

The equations propose that if a large number of divisions are produced, splitting information is also large thereby reducing Gain Ratio.

- ii. Gini Index - this is used to perform the computation of Gini Gain for feature  $A_i$ , the equation is more related to how Information Gain is computed;

$$GiniGain(A_i, D) = Gini(D) - Gini(A_i, D)$$

$$Gini(D) = 1 - \sum_{j=1}^{|c|} (p(c_j, D))^2$$

## 2.3 Convolution Neural Networks

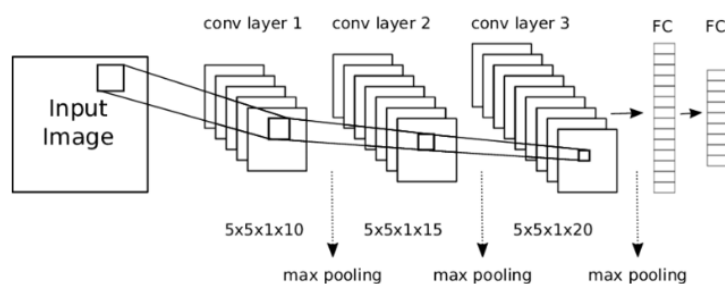
This abbreviated as CNN or ConvNet is an algorithm used in neural networks that learn is widely used for image analysis and works through analysis of the patterns. ConvNet is composed of multiple layers that consist of a collection of neurons used for processing portions from an input image which is referred to as receptive fields. This is one of the techniques that make the ConvNet ideal when dealing with images because they can manage input images by dealing with portions from the original image. Convolutional Neural Networks are mainly applied in image recognition applications which include, object detection, image classification, and action recognition. However, ConvNet is also being used in other different fields apart from image recognition and these include natural language processing, video processing, and speech recognition[25].

The ConvNet consist of a huge amount of parameters available in convolutional layers makes the architecture to be very strong. This can also contribute to the overfitting situation which is also a disadvantage. CNN has also another advantage which location invariance, it has the capability of detecting objects from anywhere on an image. CNN is capable of providing accurate results because of its compositionality. The filter in CNN is composed of local paths for the features at a lower level to representation at a higher level. Through the implementation of CNN, we can experience how pixels build edges, how shapes are built from edges and objects that are complex from shapes. The process of the convolutional neural network involves learning from the filter values using the given specific parameters to make an estimation. Based on the given requirements the network will carry out some action from the learned values during training. The training process could either be the image classification process for a certain class[25]–[27].

### 2.3.1 ConvNet Architecture and Layers

The process of the Convolutional Neural Network involves dealing with consecutive convolutional layers. The input and output neurons are connected during the feed-forward neural network training and this is referred to as a fully connected layer which is important in the final layer for the reverse process. Convolutional neural network deals with convolutions as the name states, this is used to computational operations on an input image to produce the output. The convolutional neural network is presented using the

actual convolutional layers which clearly shows the inputs and outputs at different stages coupled with dimensions as shown in the image below.



### CNN Implementation

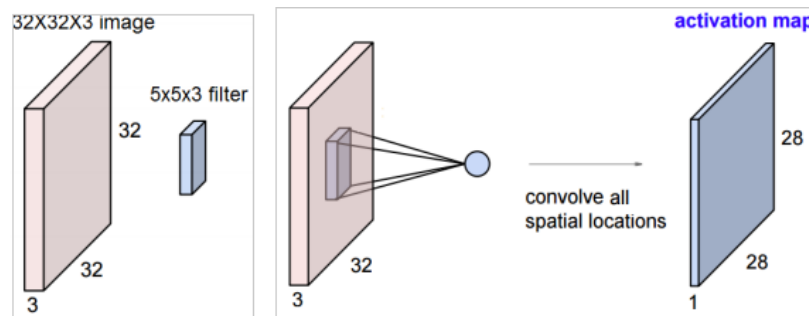
When implementing ConvNet there can be different configurations that can make on the architecture but the basic structure will remain the same. The process of CNN involves bringing together a convolutional layer before applying the non-linear activation function on the outcome of every layer. The pooling function is applied to each convolutional layers. The linked structure can be seen and is composed of the following: convolutions, pooling, and activation function which is also taken as a component of the convolution. At the end of the structure, there is a fully connected layer but it is before the final layer which is the final classification out of classes. The following paragraphs will discuss different layers that exist in ConvNet architecture[26], [28].

#### 2.3.2 Convolutional layer

In ConvNet, the convolutional layer is the most essential feature, also take note that the fully connected layer's inputs are represented in the form of the vertical line of neurons. In ConvNets the inputs are ordered in the form of square neurons and their values match with pixel intensity from the input image. The images are composed of two dimensions which are width and height and the also contain depth. The greyscale images their depth is one whilst the RGB images their depth is three. The 3 dimensions have different sizes as they are processed through the network resulting in the application of dissimilar layers[25].

The filters in convolutional layers have equal width and height. The input volume depth should also be the same as the depth of the filter. At the same time, the width and height of the filter should be smaller to ease the process of shifting during convolution.

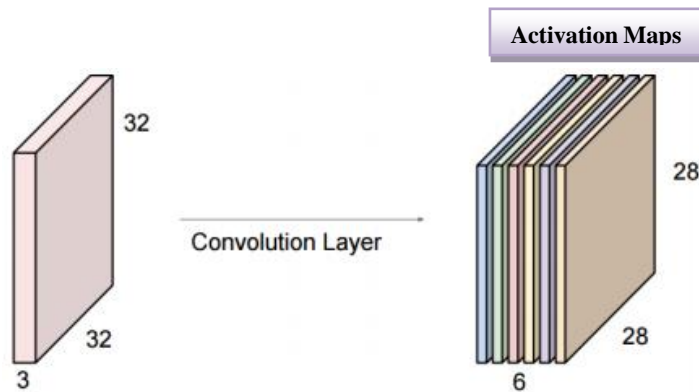
This can be explained using the diagram below where we have elements of the convolutional layer.



*Figure 7 Activation map*

The computational process involves applying filters on the small number of pixels at every stage. The produced neurons are linked with the input neurons in a small region referred to as the receptive field. The process of forwarding pass involves convolving of every filter around height and width of the particular input volume, performing computational operations on the dot product involving filter entry and input, and finally resulting in the production of the two-dimension activation map of that particular filter. Therefore, there is an on-going process that involves neurons learning the weight and bias is also learned in every layer. Also, take note that the width and height from the output are smaller compared to the input and this is because of the convolutions[25], [29].

The process of bringing together all the activation maps for the filters with the depth dimension results in the complete output volume of the convolution layer. Every filter is composed of a single activation map and the output depth is determined by filter numbers from the layers or input depth is used for determining the depth of every filter. The figure below indicated the input and the output of the convolutional layer, there are six activation maps which also means that there are six filters that have been used on the input. The filter depth should be three so it should suit the input dimensions. Assuming there is an addition of another convolutional on the output, regardless of how many filters are added the actual depth should be equal to 6.

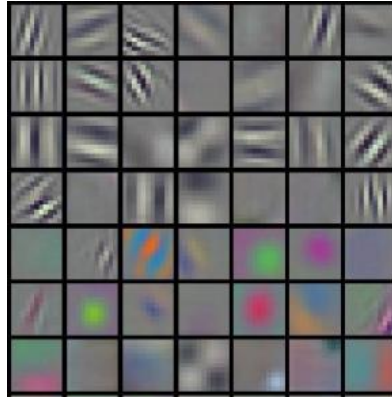


*Figure 8 Convolution Layers*

The output of every neuron is linked with the small part of the output and this is referred to as local receptive fields theory. The process of computing the activation map the links used are local in space coupled with the height and width whilst on the other side extending the depth of input volume. This kind of structure guarantees that the learned filters construct well-built results to a local distribution input pattern. This result in the activation of the network learns filter after discovering features of the spatial position from the input.

Once we have the activation map arrangement, the reinterpretation of the significance of the output can be made easily. Every depth has it a value that has a unique activation map whose computation is based on a filter that can recognize particular patterns. There are some points of width and height whose number of values and depth corresponds to a similar input section. The values correspond to the application of individual filters.

The figure below illustrates the first convolutional layers of filters from one of the know CNN, AlexNet. If we take a close look at the figure there are different shapes with their colors from each targeted filter processing feature detection[30], [31]. There are some filters with red, green, blue blobs, but as we go deep into the layers the interpretation becomes difficult because filters are less shaped.



*Figure 9 Convolution Filters*

There are also some other two hyperparameters which are important in convolutional layers and they are also found in other layers:

- i. Stride parameter - this used to identify the number of rows and columns a filter can be moved in the convolution process. The stride parameter can accept any number of values and below is the example of an expression. The number of input sizes is represented by letter N and filter size by F. However, there are situations where if you have input and filter it is impossible to have stride values. The mismatch between the filter edge and the input edge is a result of a certain condition not satisfied during filter shifting up to the last stage.

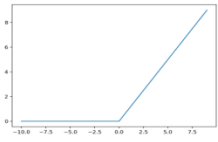
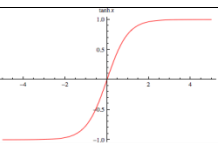
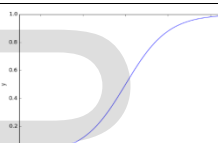
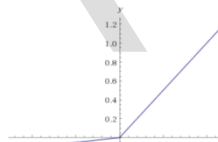
$$\frac{N+F}{STRIDE} + 1$$

- ii. Zero-padding parameter - this technique operates by adding the zeros to the input so that it can control the spatial amount of output volumes. It is also important to keep the spatial size from the input volume, this will allow as to apply a specific stride value. However, if the above situation does not satisfy present values, zero-padding which can be the alternative because it will assist to increase the N value.

### 2.3.3 Activation features

The activation feature or function is the additional function to every output of the convolutional layer following every fully connected layer. The purpose is to grant non-linear activation of convolution outputs to ConvNet. This tells that activation functions are too non-linear[29], [30], [32]. To allow the gradient optimization technique, the activation function should be composed of continuously differentiable properties. The methods show

good performance in different environments and they both have their advantages. The implementation of the ReLu function in neural network training is much faster with no considerable differentiation to accuracy. The process of choosing the activation function is very complex which is also not in the scope of this research paper. Below is the list of the common activation functions together with their equations, ReLu is the commonly used function.

|                       |  |   |
|-----------------------|--|---|
| Rectified Linear Unit | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ $f(x) = \max(0, x)$ |    |
| Hyperbolic tangent    | $f(x) = \tanh(x)$  |    |
| Sigmoid               | $f(x) = \frac{1}{1 + e^{-x}}$  |   |
| Leaky ReLu            | $f(x) = \max(ax, x)$   |  |

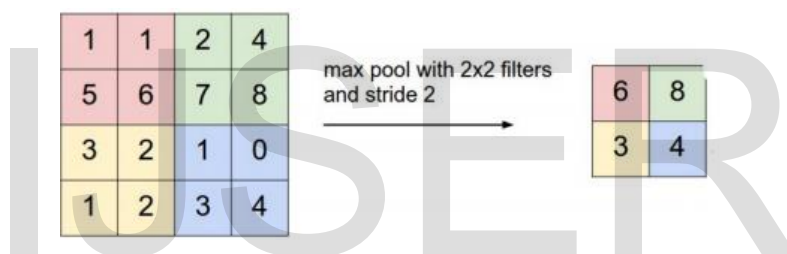
Activation functions list

### 2.3.4 Pooling layer

Pooling is a non - linear down sampling mechanism that is also available in ConvNets. Apart from several non-linear techniques that can be used to implementation of pooling max is commonly used. The max-pooling operates by partitioning input image into rectangles that do not lie on top of each other and for each associated region outputs the upper limit or maximum. The perception is that the identified features location is not as significant as its rough position comparative to extra features. The pooling layer function is aimed at downsizing the spatial size from the depiction through down-sampling of the results within the layers. This is also aimed at controlling the training through the reduction

of parameters amount in training network at the same time managing the overfitting problem. The pooling layers can be inserted during successive convolutional layers within the ConvNet architecture[29].

The pooling layers work in isolation at each slice depth of input and rearranges it spatially. The 2x2 filter size is the most widely used and it is applied to the stride of two. This reduces 75% from the inputs because 1 out of 4 elements is used. The pooling is performed on the activation map separately and this does not affect the depth dimensions. Other pooling functions can be applied apart from the max pooling and these are; L2-norm pooling and average pooling. The image below illustrates the implementation of max pooling which was done on a single activation map using the 2x2 filter with stride 2. The different colors represent the filter position from the input activation map. The output is the maximum location value.



*Figure 10 Filter with 2x2 max filter*

### 2.3.5 Fully connected layer

The implementation of ConvNets involves linking activations to neurons in fully connected layers. In ConvNet structures, the fully connected layers are at the final section of the training network and this is after several pooling and convolutions have been performed. These fully connected layers can also be referred to as an associate cluster of convolutional layers where the filter size matches the input volume. Here the filter can be referred to coefficients set and is multiplied by input elements followed with a bias counteract. In our situation, 1x1 is our output size and its depth is the same as filter number and this can be characterized to a column[33].

### 2.3.6 ConvNet Training

There are similarities in convolutional neural network training and the conventional neural network. This we will discuss the main features that are common in ConvNet.



Different researches are being performed that is aimed at advancing the training like the regularization methods, parameter revision and new techniques to amplify the performance. The forward and backward pass are the two phases involved in the training procedure. The process of forward-pass involves performing network computation on outputs through the application of pooling, activation, and convolutions until the end of the training where the final output is obtained. This level acts as the new image inference level and predictions even though there might be some alterations. After the forward pass, the backward pass follows aiming at correcting values from all layers through the identification of the loss function by comparing values produced during forward-pass with the estimated results[29].

The training implementation procedure is recurring together with the epochs which are the training bound. During the training period, the dataset elements are utilized in the ConvNet weights updating at every epoch stage. The higher the epoch number mean deeper the training and perfect output or accuracy. Several parameters can be utilized during the implementation with different customization to suit our problem. In the following sections, we will discuss some of the important issues that can be utilized during ConvNet training.

### **2.3.7 Backpropagation and update parameter**

This process is also known as backward propagation of errors commonly implemented in neural networks together with the optimization mechanism like gradient descent. The mechanism works in a way of circulating errors to the entire network to bring up to date weight on layers. These are the error that was produced by loss function which evaluates the produced results with the predicted value. The main reason for implementing backpropagation is to amend weights on actual layers and reduce the loss function. To successfully implement backpropagation so that we can calculate loss function, we need to have known expected output from every input value. This also sometimes referred to us supervised learning mechanism even though it is also implemented in the unsupervised network[30], [32].

When implementing the optimization method, gradient descent is the preferred mechanism. The mechanism is used to perform compute loss function from the gradient in connection with network weights. To control or reduce the loss function, the gradient is

incorporated into an optimization mechanism. The chain of rules is applied to perform parameter modernization and repetitively calculate the gradient of every layer. Therefore, the process of backpropagation needs activation functions in ConvNet to be differentiated to compute the gradient.

It is less assured that the combination of backward pass and gradient descent can be used to obtain a global minimum of error function but it can be used to obtain a local minimum. Based on the gradient surface, the union algorithm can be time-consuming. However, when working with little datasets and the limited networks the problems may not be experienced but as the training dimensions start to grow it can worsen the condition. In this situation there some other algorithms that can be utilized in updating the parameters. Updating the momentum will assist to maintain the velocity whilst the update process is in progress. If there are some amendments to the gradient values the impact on the bearing of the actual updates will be minimal. Adam update, AdaGrad, and RMSProp are some of the methodologies that are aimed at overcoming the challenges with the convergence of backward pass[34], [35].

Despite the selected methodology used when implementing backward pass also ensure that parameters have been configured properly. The learning rate is responsible for ensuring the weights are updated correctly following the gradient direction. If the learning rate is higher it will lead to a quick movement to preferred directions but also forms a basis of jittering when close minimums. The reverse of the above statement will result in slowing down of the convergence and trapped in local minimums. The best alternative is to use a higher learning rate in the early epochs and reduce the values so that convergence is easy.

### **2.3.8 Regularization**

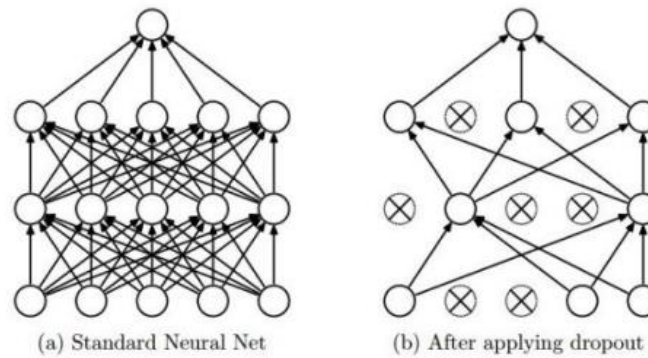
When the training neural networks there are some of the problems that can be experienced, overfitting is one of the problems which is a result of the model being extremely complex this is due to many parameters concerning observation numbers. Overfitting affects the model causing poor performance when processing predictions because it reacts excessively to slight oscillations during data training. The overfitting problem can be resolved by raising the quantity of data to be trained or decrease the network size. Also, put into consideration that larger networks provide better performance

compared to small networks. The regularization technique is used to solve the overfitting problem even if the network and the training data are fixed[36].

The main initiative of having several regularization mechanisms is to append the regularization term to the loss function. The technique is also referred to as weight decay and the L1 and L2 are the famous regularizations. The implementation of L1 regularization involves adding up the sum of all total values of weights whilst L2 regularization involves adding up the computation of squares of weights in the entire network. Regularization can be implemented differently based on the technique nevertheless it must be multiplied with the coefficient.

The implementation of regularization is to allow the network to learn even small weights. The involvement of large weights will be done only if they have a positive impact on the loss function. Regularization is a method of finding the middle ground involving small weights and cost function reduction. Take note that the importance of the two values depends on the coefficient.

Dropout is the other technique that is also used to control overfitting and it can be implemented together with regularization. During the training implementation, the dropout is used to store neurons that have the probability, and others are neutralized. The training makes use of the trimmed network using the available data. Later nodes that were taken away are incorporated into the network plus their usual weights and when testing begins the dropout is not functional. The implementation of the dropout has been illustrated in the image below, the first image shows the fully connected neural network training and the second image shows the network after the dropout has been implemented, here we see that most neurons have been deactivated. The training of a few nodes controls overfitting in the neural network at the same time improving training speed. Complexity and compactness of nodes are controlled resulting in improving the learning process and good data generalization[36].



*Figure 11 Neural Network generalization*

### 2.3.9 Batch normalization

Batch processing is the mechanism that is used in the training of data, the data is separated into clusters which are referred to as batches. Then independent batch processing is performed during the training. The process of allocating values to batch size before the training is important simply because of variations in performance within the process[37].

It has also been observed that variation within the allocation of activations due to parameter updates reduces the ConvNet learning and this is referred to as internal covariate shift. The internal covariate shift can be controlled through the implementation of batch normalization. Whilst performing the training every activation of the batch is set to zero-mean and unit variance. However, variance together with mean is evaluated throughout the entire batch separately at every activation[29], [37]. The implementation of the about mechanism will assist in speeding the learning process and obtain high accuracy.

## **3 Chapter: Related Works**

### **3.1 Automatic Diagnosis**

The introduction of image recognition using the machine learning technique was in the 1980s, this is when Kersten et. al. suggested a neural network model can be utilized in healthcare to perform the image recognition using 2-D images. But due to the lack of proper statistical models and software, the suggestion was overly controlled[38]. Marr thinks the processing of images using a computer is to imitate humans' process on visual information cognition[39]. However, Taylor et. al. also suggests that computers can not only be used for processing images but during processing, they can also discover some patterns[40].

In the 1990s they discovered that challenges in performing face and medical image recognition was due to lack of proper algorithm that could be used to perform image recognition task. The system of automatic diagnosis in the field of machine learning was introduced in the 1970s, but there have been advancements in information technology that has also positively impacted the health sector. The Computerized Decision Aid for Stroke is a predictive model for stroke patients using the available health information to diagnose patient whether they should receive thrombolysis treatment. The Electronic Medical Record is another system that is used to mine important information from the database for planning purposes within health care.

The machine learning technique is used in different fields which include number and text processing. There have been advancements in technology to the extent that it is also used for image processing in health care for diagnosis and judgment. The health care algorithms well assessed especially the one used for image processing and an evaluation was made that the technique is vital for computer-aided detection to sense the clinical important patterns from massive health care medical images, resulting in performing computer-aided diagnosis. However, De Bruijne also suggests that automation using a machine learning algorithm for image diagnosis has proved to be a successful approach. Furthermore, addressing the challenges suggests that there need for further research to

improve the models' architecture to have more reliable high tech systems. The machine learning technique used in health care for medical image diagnosis has two groups. The technique can be utilized to perform the detection and classification tasks on grayscale radiography images for pattern recognition produced by computerized X-ray imaging (computed tomography - CT), nuclear medicine function image system(positron emission tomography - PET)[41].

The applications can also be segmented based on their specific purposes into 6 groups, which include: pattern recognition for medical images, diagnosis with the support of computer (computer-aided diagnosis - CAD), the neurological diagnosis, image retrieval system based on 3d(CBIR), registration systems and natural language processing. Wang et. al. statements of a comprehensive sensing model (dual dictionary) to rebuild by computerized X-ray imaging and magnetic resonance imaging to achieve better classification results during training[42]. However, Nouretdinov et. al. suggest of a classification model based on transductive active learning (transductive conformal predictor - TCP) to be used for character recognition and prediction in Magnetic Resonance Segmentation on medical images as a biomarker which includes; diagnostic and prognostic in disorder[43].

To overcome the classification process in brain tumor (benign and malignant) which contains the oval-shaped tiny spot, Cheng et. al. launched the deep learning structure on computed tomography(CT). Therefore, based on all these different research from different researchers it is evidenced that the machine learning technique can be utilized in the medical field for diagnosis and detection. But due to some of the challenges like inadequate training, and the capabilities of the model that can be used in the health care field[44].

Therefore, the machine learning technique is utilized in the health care field of histopathological for classification and detecting patterns from color images, whilst for radiological we use the grayscale images. In this regard, the color images are portrayed by the three-dimension matrices. These 3-dimensional images need more computation to perform the analysis task using the machine learning technique and model fine-tuning

which is in contrast with the radiology methods which use greyscale representing one channel.

The introduction of computers in the medical field to carry out the diagnosis task (Computer-assisted diagnosis) started in the 1990s, this was when the computer was used to analyzing histopathological images. Gurcan et. al. suggests that the histopathological analysis process automation technology can be split into two: perform pathological detection through pattern recognition to check for the abnormalities which will result in some degree of disease sequence[45]. Lessmann et. al. suggests a model called query by image based on numerical and functional analysis to deal with the missing information among visual representation and the actual semantics. The techniques were implemented to recognize the tissue images of meningioma with 79% accuracy, computational challenges was a major problem coupled with features generally[46]. However, Vanderbeck et. al. suggest another model that could be utilized for histological classification, the supervised machine learning model. The training was performed on non-alcoholic fatty liver infection where the classification accuracy based on the algorithm was 89%, also take note that the results obtained from other tissues were different[47].

Svensson et al. implement the detection of tumors in the blood cells using the Bayesian method. The implementation evaluates the Bayesian methodology with other classification methods like the SVM, visual classification, and GMM. An accuracy of 99% was achieved using the region of interest and unprocessed images 75% accuracy[48]. The computer diagnosis system used for the detection of thyroid tumor was built by Gopinath and Shanthi, the implementation utilized different algorithms like; DT, k-nearest neighbor algorithm, SVM and recurrent neural network and achieved 90% accuracy[49]. However, Gertych et. al. implemented the SVM and RF using the machine learning algorithm model for detecting pattern disorder prostate tissue medical images. The training revealed that multifunctional methodologies achieve outstanding performance but there is also a need for advancements in the robustness of the actual methodology. Evaluation of microscopic examination of tissue image analysis reveals that the computer-aided detection process is a supportive alternative disease recognition, diagnosis, and forecast of a medical condition and its cause (pathology). Furthermore, it foresees that advancements in computer-aided

diagnosis will be composed of different models working together on multiple data sources computerized image analysis, microscopic examination of the surgical specimens, gene expression, and registration[50].

### **3.2 Image Retrieval using Machine Learning**

The automated image analysis using computerized methods can be transferred to the period of artificial intelligence for computer-based image retrieval. The computer-based image retrieval is vigorous in study areas of retrieving knowledge based on the information and can be traced back in the 1990s where the major interest was applications that can handle multiple tasks like indexing and searching. Tan et. al. made improvements on document ranking using the Bayes model together with stochastic sampling[51]. To improve the health care image retrieval Ayadi et. al. established a decision network model built on word finder to connect semantic meaning to the focal images for further improvement of healthcare image retrieval through scoring based on medical features like modality and dimension. The state of art results was achieved in contrast with the Boolean model of information retrieval[52].

The famous implementation of content-based image retrieval comprises of Query by Image Content and the Virage model and video retrieval process acquired by ConvNet. The execution of image IR involves pulling out important regions from an image using pixels like colors and edge detection and reconstruct the contents with semantic meaning. The content-based image retrieval structure is assembled upon retrieval system assisted seven feature algorithms, interval and resemblance algorithms, the right repository and permission methods, and adaptable interface.

There are two ways in how the visual features can be grouped: using the basic features like image dimension, color intensity and borders, and logical attributes which encompasses identification of objects mixture and precise environment. The basic feature images are recommended by image IR structures through the utilization of segmentation and feature abstraction to obtain the expected knowledge domain even though it is not easy to give an analysis of the sematic. Other researchers suggest captioning and annotation of text be used together with images to cover the gap in sematic, however, the current



methodologies can only manage to render all-purpose information with questionable accuracy.

The use of Image IR is being widely used in the biomedical field this is due to its accuracy in the diagnosis process and decision making which surpass human capability. The familiar non-invasive imaging techniques include automated radiological and microscopic tissue examination. Through the use of applicable classifiers, the information retrieval system is capable to select important visual information as well as annotated words. Therefore, the extraction of optical patterns from a picture coupled with syntactic knowledge will facilitate automated examination using health care images. For instance, partially strained ConvNet is utilized for nuclei cancer cell examination from the entire colon cancer tissue cell images.

Besides, different research was conducted using image annotation alongside numerous keywords, the classification process was done through the merging model with center symmetric using wavelet and visual descriptor. In the repossession section, every keyword is given a confidence score depending on the outcome results of the intelligent retrieval model to minimize semantic difference separating the user and information retrieval system. The implementation of this methodology minimizes the errors with an approximation of ten percent in comparison with modified support vector machines. Morioka et. al proved that contextual detail examination can be done using ultrasonic radiology classification with a context-based learning algorithm. However there different information retrieval technologies that can be utilized to improve the IR healthcare information system like rule-based machine learning, learning probability model, and the statistical model. All the algorithms produce different results when incorporated within the medical information management system, however, no information retrieval algorithm is greater over other methodologies. Their different experiments are conducted to find methods on enhancing semantic linkage separating pictures and words for instance captions and annotations or use of artificial intelligence for automatic image tagging to upgrade content-based image retrieval performance. For this reason, the deep learning algorithm can be utilized to perform image analysis and classification[53].

### 3.3 Machine Learning Malaria Diagnosis

The introduction of the automated system to perform image classification using the blood smear began in the period of the 2000s through the use of Malaria count to check malaria parasites using Giemsa blood smear images. Nevertheless, the researchers analyzed the correlation between manual and machine parasite recognition but did not perform the classification process between unhealthy and healthy cells.

The classification of blood smear using the support vector machine was done in 2009, Diaz et. al. implemented the method to examine the infected erythrocytes. The algorithm achieved better results in terms of precision and sensitivity but this training was performed on four hundred fifty malaria image cells. However, when the enforcement is done only one side of the unhealthy blood smears the model is negatively affected in terms of performance[54]. Verma et. al. implemented the classification process on image blood cells to identify proteins deposited by genus plasmodium into the red blood cells. The training model obtained 86% accuracy and 88% by separating amino acid and dipeptides formation[55]. The implementation of the support vector machine with the profile kernel model to check the presence of degradomes was done by Kuang et. al. this technique is used to foretell the anti-malaria ability to fight back. In 2011, there was another implementation that was based on turning the color images into greyscale and this technique proved to be better for the support vector machine classification process whose image dataset was 266 and gave a precise score of 0.97 with an average number of 0.8 false-positive discoveries for every picture[56].

Despite malaria detection using a support vector machine, there are some of the machine learning techniques that are utilized for the automation process. The implementation of the Bayesian model in the discovery of malaria pigment prohibiting compounds in the exploration of anti-malaria medicine. The Bayesian Structure Learning Algorithm which with a tree structure for classification was suggested by Yin et. al. , the model is capable of process large amount s of data simultaneously and was implemented on transcriptomic information in genomic research for malaria, the Bayesian technique performs better in the analysis of genomic information and can be utilized for future malaria research studies[57]. Scotti et. al. introduced another model that was to discover

anti-malaria medicine and this the artificial neural network technique[58]. Das et. al. utilized the two machine learning model, support vector machine and the Bayesian model, these models were used to discover malaria from unhealthy microscope blood smear cell images, 84% accuracy with cross-validation was achieved after the training. Nevertheless, the implementation of the model has difficult information processing phases which result in escalation workload because modifications need to be done manually. The discussed methodologies proper preprocessing methods and small data were used for model evaluation[59]. Zinszer et. al. contributed that machine learning techniques can be utilized in healthcare to enhance malaria prediction process. Despite better achieving better results some other areas need to be considered like data representation and algorithm stability[60].

### **3.4 Deep learning and convolutional neural network**

Deep learning is a branch of machine learning in the field of artificial neural network and it is an unsupervised learning technique. The use of this technology in medical research has proved to be beneficial because of being its capability of being applied in different medical fields. The implementation of ConvNet involves dealing with deep learning structures especially computer vision where it is applied for image recognition. Convolutional Neural Network implementation is composed of the following key elements: a network connection, pooling, weights, and convolutions. Convolutional Neural Network was introduced in the early 1990s and was used for voice recognition and character recognition. The use of ConvNet was prolonged to handwritten text recognition followed with natural image analysis. In 2012, Alex Krizhevsky proposed ImageNet (AlexNet) which was used to enhance the classification process of natural images in ConvNet. The introduction of AlexNet in the field of ConvNet was a step forward because of its capability in performing multiple category classification processes. In 2012 contest for ImageNet Large Scale Visual Recognition Competition, ImageNet with 7 Conv. layers performed classification process of 8.9 million images successfully[61].

### **3.5 Comparing ConvNet with other Methods**

The use of ConvNet to implement pattern recognition and identification is based on the complicated topological architecture which implements multiple-layer backpropagation. There is an application of convolutional layers above the ConvNet structure that is used to study the complicated high dimensional space, non-linear averaging

through the process called back-propagating, and utilizes the early minor layers of convolutions as a feature extractor for fully linked network layers as the classifier. In contrast with other ConvNet/CNN methods and machine learning techniques like KNN, and statistical learning model that leans on self-reliant feature abstraction algorithms to extract characters, ConvNet accepts raw pictures as input with little preprocessing like centering and normalization of the pixels. However, the major drawback of an unstructured neural network when performing image identification is the scarcity of inbuilt invariance beneficial to translation[62].

The ConvNet operates by applying a topology architecture to emphasize localized feature abstraction through prohibiting the responsive fields of unseen units to be localized. Hubel et. al reported that this mechanism is borrowed from physiological research on cats' visual discovery neural system introduced in 1959. Therefore the visual objects are easily discovered through recalling their local sensitivity, position selective neurons from the neural system. The neural network implementation involves the process of extracting basic features like edges, corners, and pixels which are later put together at higher layers of neural network architecture. The application of sequential convolutional kernels to identify localized visual characters and display them inform of a feature map. Later the feature maps are put together as a convolutional layer and send to the adjacent layer. In short, the ConvNet utilizes three techniques to improve its learning and deal with the barriers: convolution, activation functions, and max-pooling of the layers to enhance computing performance. Hence the implementation of ConvNet gives a direct explanation for visual pattern recognition and classification. The implementation of a convolutional neural network depends on high computational power because it must perform calculate different vector weights coupled with some matrix operations. The low storage cost and advantage of cloud computing resources offer a better implementation environment which leads to making ConvNet the leading-edge artificial intelligence for image diagnosis[63].

### **3.6 Challenges to CNN applications**

Different factors can alter the performance of ConvNet. The ConvNet implementation structure first operates by introducing small filters coupled with a deep learning structure to obtain detectable image features. However, the implementation of

complicated ConvNet structure requires more computational power. But the introduction of the Graphics Processing Unit has enhanced the training process but are not able to provide analysis performance. The training data size also has an impact on the performance of ConvNet in terms of analysis accuracy. Therefore the amount or size of the training data has a positive impact on the model accuracy, the previously discussed models used small data and are not suitable for deep learning models[64].

The introduction of transferred learning was aimed at solving the inadequate training data issues, here the pre-trained is utilized for feature abstraction and the abstracted features will be utilized for fine-tuning the classifier for the results. The implementation of transferred learning can also be seen as an alternative route for deep learning which is cost-effective and time-saving but still achieves good performance. This technique can be implemented in situations where there no large data to be used for training. In this research paper, we will utilize ConvNet to perform the classification process for malaria blood smears.

IJSER

## 4 Chapter: Proposed Method

### 4.1 Proposed Method

In this research paper, we will use deep learning to perform the image classification process, the structure of ConvNet will be utilized for performance evaluation. The essential feature of deep learning is the capability of applying a multiple-layer network to change the input field to convert it by the invisible nodes. Through a sequence of transformations, the training model determines the input data patterns using the technique called back-propagation. Through this method, part of the derivative from the input specification is calculated from the part derivative based on the output using the chain rule. This tells that modifications on one layer are calculated through measuring modifications of connected layers[29].

The process of training a ConvNet involves the following computational techniques: feedforward and backpropagation. During the training period, the feed-forward is mainly used to calculate the output of each layer from the entire units from that particular layer where the non-linear function which is represented by  $f()$  is tested to the weighted z sum of low layers. The applied either be ReLU, Tanh and logistic function. However, the implementation of backpropagation involved fine-tuning the deep learning model through calculating parameters of every layer in reverse form. For example, if we have a loss function from the first unit of our output layer and this will appear like this:  $0.5 \times (y_l - t_l)$ ,  $t_l$  represents output, error derivative  $y_l - t_l$  from the output can be changed to error derivative the sum through multiplication with a partial derivative from  $f(z)$ .

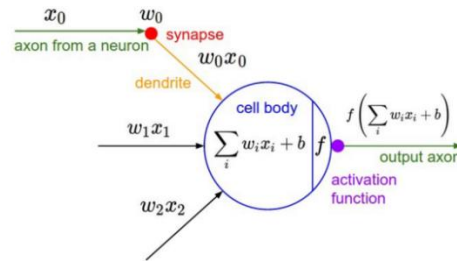


Figure 12 Neuron

$$\begin{aligned} \Delta z &= \frac{\partial z}{\partial y} \Delta y \\ \Delta y &= \frac{\partial y}{\partial x} \Delta x \end{aligned} \quad \longrightarrow \quad \Delta z = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x} \Delta x \quad \longrightarrow \quad \frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$$

$x \longleftarrow y \longrightarrow z$

Figure 13 Back Propagation using Gradient Descent

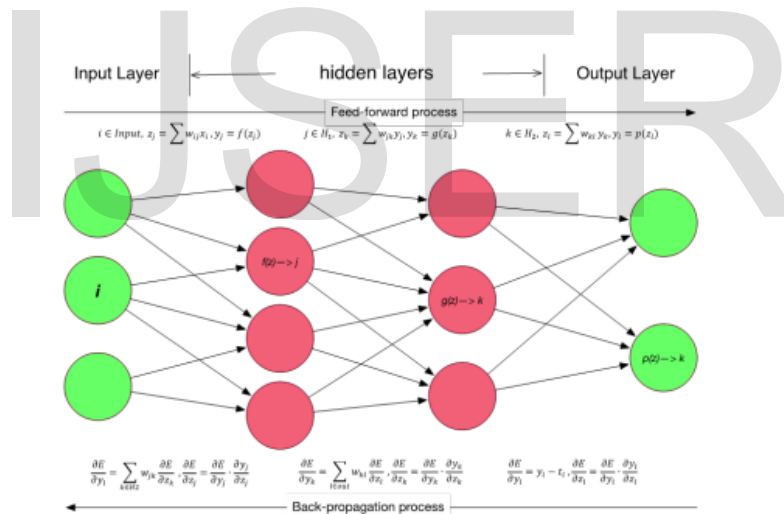


Figure 14 Feedforward and Backpropagation

Based on the above ConvNet process structures it clearly states that the process of learning is initiated when raw data has been feed into the network. The feed information pass through the entire network of the CNN model. The ConvNet model structure is composed of the five important layers which are applied at different stages and these are: convolutional, pooling, fully connected, and Softmax.

#### 4.1.1 Convolutional Layer

This section will explore one of the most essential components of the ConvNet architecture, the convolutional layer. The convolutional layer is composed of different filters that are used as feature maps to be linked to local patches within the feature maps from the preceding layer with some weight[28].

The convolution can be explained mathematically using the 2 vectors,  $e$  and  $t$  mean the degree of overlaps filter vector  $t$  moves across  $e$ . Then letter  $p$  be a measure of length for vector  $e$  and  $g$  represent length for filter vector  $t$ , the vector  $s$  is for the convolution  $e$  and  $t$  which can be express as  $s = e \otimes t$ , where  $s(b)$  the product from the convolution for  $b$ th entry in vector  $e$  and can be calculated using the formula below.

$$s(b) = \sum_q e(q)t(b - q + 1)$$

During the implementation process of the convolutional neural network, the representation of an image is a 3-dimensional matrix with the beginning 2-dimensions are for the pixel value and last for color channeling which represents: Red, Green, and Blue. The algebraic notation of the image Conv. operation can be written as the one below:

$$j(i) = f(i) \otimes m(i) = \int_{-\infty}^{+\infty} f(\tau)m(i - \tau)k\tau$$

Here  $f(i)$  stands for an image that will be processed and  $m(i)$  represents the filter function. The output of the convolution is the element of the dot output from the input image and filter whose filtered product is nonzero.

The local value from every unit weights is forward to non-linearity like sigmoid or rectified linear unit function. The feature map units have an equal proportion set of weights, whilst other feature maps operate other filter banks for element depiction. The logic of this architecture is established on two deliberations. The initial stage starts with the input of images which are fed into the network and are represented in the form of an array that represents details of the local pixels. these local pixels are more correlated resulting in an establishment of special local patterns that are easily identified. Furthermore, the stability of the image signals at a location enforces that any changes in the actual position do not redirect the element of the indistinguishable pattern in further position. During the process of performing calculations, the layer executes filtering activities through the element map in a distinct convolution. The figure below shows a 32 by 32 three-channel image which has been scanned with 6 filters with an output set of 28 x 28 x 6 maps for activation.



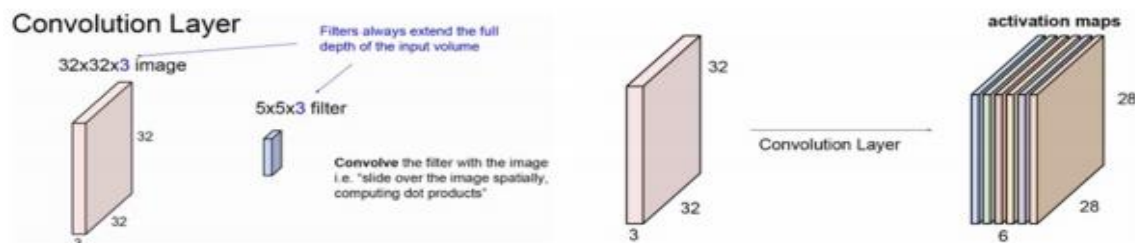


Figure 15 Convolution layers with activation maps

#### 4.1.2 Pooling layer

The pooling of layers during CNN training is different from the convolution layer which operates by detecting the local features, pooling operates through merging to minimize the feature map size. The process of merging the local semantically identical elements into further short illustration. The patterns that are established by parallel points of elements can differ, it is the responsibility of the pooling function to coarse-grain the location of every element with the pattern by a specific function. The regular functions comprise calculating the highest of units from the local patches in the corresponding element map. This results in the magnitude of the depiction of the elements are minimized and succeeded by an invariance established by the little changes and misrepresentation of pooling action[65].

The merging of pooling and convolutional layers establishes an elemental architecture to minimize the dimension of the element maps that final point to the aimed dimension for analysis using the Softmax function.

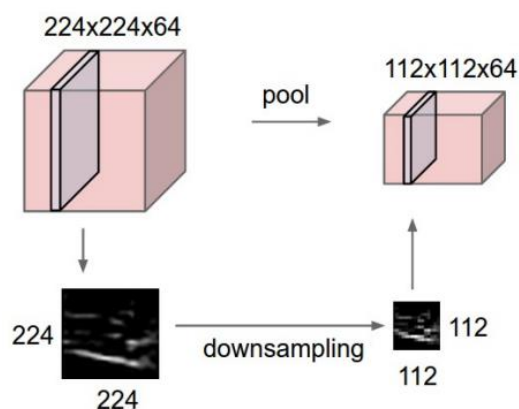


Figure 16 Downsampling

#### 4.1.3 Activation Layer

This is a non-linear function that is used to bring about the learning knowledge from the convolutional neural network structure. The ConvNet architecture is established by a pile of 2 or 3 steps of convolution, pooling, non-linearity and succeeded by further

convolutional and fully established layers. The studied arrangements are developed when the network is performing backpropagation gradients which mean rolling back through the already trained ConvNet model to perform the recalculation of the weights from the banks of the filters. The rectified linear unit, cross-channel normalization, and sigmoid are the three activation functions in ConvNet which can be utilized to perform the image classification process[66].

The rectified linear unit layer can extremely stimulate the merging of the stochastic gradient with minimal operation cost where all the stored values are positives and all the values that are reset to zero are the negatives. But when the rectified linear unit has been overused it will simply silence all the small non-zero values to zero and they will not be restored. However, the sigmoid layer flattens actual valued figure to amid zero and one which results in a rise in density depiction. On the other hand, the sigmoid layer will overfill and finally erase the gradients when gradient values change to tiny as infinite values of input raise.

$$f(y) = \begin{cases} y & y \geq 0 \\ 0 & y < 0 \end{cases}$$

$$f(y, d, s) = \frac{1}{1 + e^{-d(y-s)}}$$

The commonly used activation function is the rectified linear unit layers primarily used for developing learning. The convolutional neural network model is comprised of 10 to 20 rectified linear unit layers that are used to perform the computations of millions of weights among the billions of connected neurons which are time-consuming and need regular hardware updates e.g the GPU, Software e.g CUDA and the algorithms. The cross-channel normalization is an alternative activation layer that restores local feedback normalization from every input feature.

$$y^l = \frac{y}{k + \frac{a \times ss}{windowchannelsize}}$$

Here the hyperparameters are represented with  $k$ ,  $\alpha$ , and  $\beta$  and  $ss$  refers to the sum of squares of the features from the normalization window. The process of elements generalization can be enhanced by scheme normalization through the neuron action  $a_{y,t}^i$  which calculated by kernel  $i$  and location  $(y, t)$  and then uses the rectifier linear unit nonlinearity. The return normalized action  $b_{y,t}^i$  can be expressed using the formula below.

$$b_{y,t}^i = \frac{a_{y,t}^i}{(k + \alpha \sum_{\max(0, i-\frac{n}{2})}^{\min(N-1, i+\frac{n}{2})} (a_{y,t}^j)^2)^\beta}$$

The computation of  $n$  neighboring kernel connections at the equal spatial location is calculated and  $N$  is the overall number of kernels from the layer. To enhance the learning process the normalization layer executes local vary normalization pattern. The training process of our CNN model we implemented global contrast normalization as part of data refining hence the normalization layer has not been implemented in our ConvNet architecture. The table below illustrates the three activation functions.

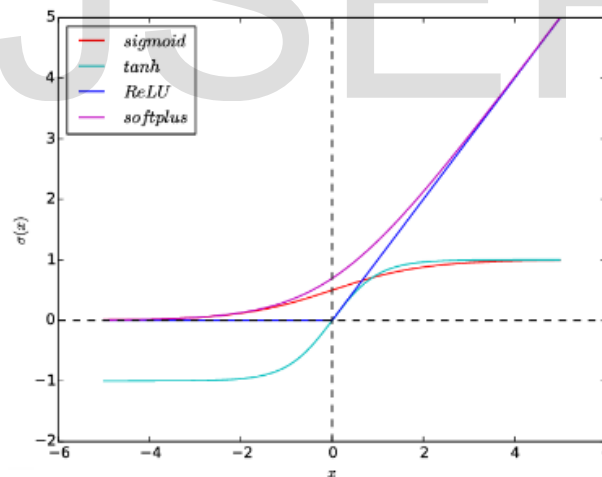


Figure 17 Three Activation functions

#### 4.1.4 Output layers

The final output of the convolutional neural network architecture is the fully connected layers which are linked to the Softmax activation function. The layers which are fully connected comprise of neurons that link to the whole input volume like other neural networks. We used a rectified linear unit within the fully connected layers to develop more

learning. The Soft-argmax function layer brings back the conditional probability of a certain class, is also called the normalized exponential which can also be utilized for generalization of multiple classes for logistic sigmoid function.

$$P(C_r|x) = \frac{P(x|C_r)P(C_r)}{\sum_{j=1}^k P(x|C_j)P(C_j)} = \frac{\exp(a_r)}{\sum_{j=1}^k \exp(a_j)}$$

$$(0 \leq P(C_r|x) \leq 1 \text{ and } \sum_{j=1}^k P(c_j|x) = 1)$$

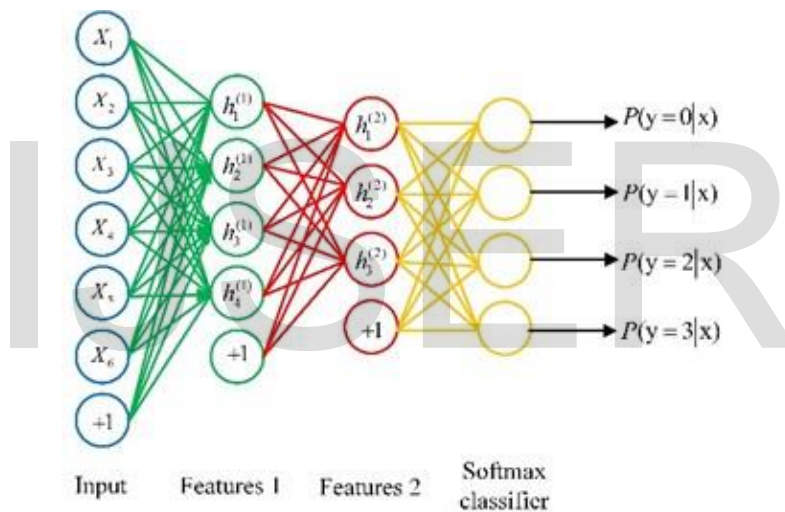


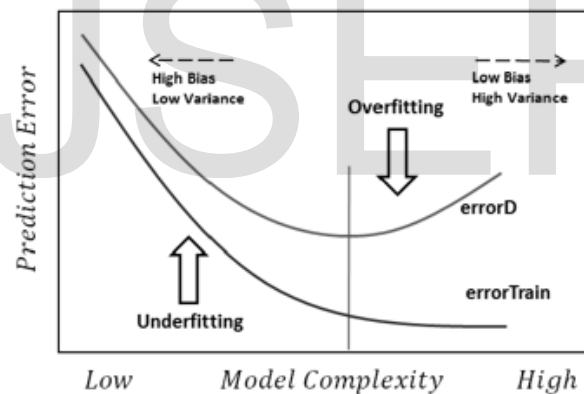
Figure 18 Softmax classifier

#### 4.1.5 Model Optimization and fine-tuning

The implementation of machine learning techniques is also mainly based on fine-tuning and optimization which supports the performance of the training algorithm. The main purpose of implementing the machine learning model is to estimate a depiction function to outline our input variables (Xs) which correspond to the output (Y) variable. During the processing of training a model the parameters (neurons weights) are fine-tuned with the input data motif. If we have a good depiction of our data sample, the training model achieves the volume of generalization to resolve the hidden case using the obtained intelligence, therefore if a machine learning model is fine-tuned well it will be able to

perform the learning and generalization process of the natural world. But assuming that we reach a point where our training model is failing to perform classification based on unseen issues, this can be categorized into underfitting and overfitting[67].

Overfitting refers to a situation where the model has been trained skillfully with the data sample to resolve hidden analogous issues. This is due to the capability of the model to study all the patterns from the training data and the produced noise by random variation. Therefore the overfitting function curve is near the sample training data to generalize thereby losing the capacity to estimate the usual patterns of the analogous data. However, the underfitting refers to the incapability of the model to resolve the training data or perform the generalization for the hidden data. The under fitting situation can be reached when the wrong training model has been selected or the application of unsuitable architecture. However, practically it is not difficult to discover and resolve the underfitting problems through alternative modeling.



*Figure 19 Overfitting and Underfitting Graph*

The main problem when implementing model optimization using deep learning is overfitting. However, several methodologies can be utilized for deep learning model optimization. Most methodologies are selected to minimize overfitting to grant the trained model with adequate generalized cognition scope. The frequent approach consists of alternating activation functions, fine-tune the learning rate depending on the modification of results of loss function at every epoch within the training, incorporation of random drop out to abandon the over tuned neurons and use the suitable backpropagation algorithm to speed up the knowledge gaining outcome at better low cost. The SGD (Stochastic gradient

descent) is an optimization approach for backpropagation by calculating the limited derivative of the sequence of tiny subsets of the entire dataset with the correct learning rate in various training epochs[68].

The SGD is a sequence of prevailing optimization mechanisms used for deep learning. However, if we compare stochastic gradient descent and batch gradient descent which calculates the gradient of the entire training dataset, stochastic gradient descent also called the incremental gradient descent, it is the stochastic similarity of the gradient descent optimization approach to reduce the aimed function represented as the sum of distinguishable functions by searching the lowest of the loss function in every loop of calculating the deep neural network. Mathematically the discrepancy between the real ground value and estimated value can be given as:

$$Q_{train}(w) = \frac{1}{2m} \sum_{i=1}^m (h_w(x^{(i)} - y^{(i)})x_j^{(i)})^2$$

The letter m represents the total figure of training data points. The calculated gradient can be conveyed in  $j$ th epoch using the equation below:

$$\frac{\partial}{\partial w_j} Q_{train}(w)$$

The function to minimize can be expressed in summation form below:

$$Q(w) = \frac{1}{n} \sum_{i=1}^n Q_i(w)$$

Here  $Q_i$  represents the  $i$ th examination of the training set. And where  $Q_i(w)$  represents the loss function value for the  $i$ th dataset coordinate and  $Q(w)$  is for the experimental risk. To reduce the loss amount we can utilize the standard batch gradient descent approach using the following repetitions or iterations:

$$w := w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^n \frac{\nabla Q_i(w)}{n}$$

The character  $\eta$  represents the training rate to proximate the loss. The BCG algorithm needs to calculate the gradient of the entire training dataset and this results in high computation cost when the training dataset very big.

However, SGD does not calculate the real gradient for  $Q(w)$  hence the gradient is aggregate proximate for the gradient of tiny subset randomly chosen from the entire batch. The  $i$ th discovered data coordinate is refreshed by:

$$\text{cost}(w, (x^{(i)}, y^{(i)})) = \frac{1}{2} (h_w(x^{(i)}, y^{(i)}))^2$$

The randomly selected subset gradient as illustrated below:

$$Q_{\text{train}}(w) = \frac{1}{n} \sum_{i=1}^n \text{cost}(w, (x^{(i)}, y^{(i)}))$$

Alternatively, the formula above can be shortened as:

$$w := w - \eta \nabla Q_i(w)$$

Here the letter  $i$  stands for the  $i$ th randomly chosen subset from the epoch of backpropagation. The calculating epoch in the SGD algorithm entails that the algorithm will go through the entire training set. The process of iterating signifies that when the random subset is randomly selected the SGD algorithm undergoes a limited subset to calculate the local aggregate of the gradient. When the training algorithm undergoes the entire training dataset, it upgrades the value of  $w$  by every training illustration up till algorithm assembles. The steps below illustrate procedural steps how SGD algorithm executed:

1. Select the basic criterion vector  $w$  and study rate  $\eta$
2. Reiterate till proximate smallest is obtained
3. Randomly organize the data illustration in the training set
4. For  $i = 1, 2, \dots, n$  do:

$$5. w := w - n_i(w)$$

To accurately reduce the inequality among the real gradient and the gradient calculated from one detected data illustration, the gradient can be calculated exceeding one learning illustration at every step. This technique is referred to as mini-batch gradient descent:  $w := w - \eta \nabla Q_j(w_{i:i+n})$ , here the letter  $j$  refers to  $j$ th batch of the entire training set and letter  $n$  refers to batch size. In our training model, we implemented an adaptive gradient algorithm (AdaGrad)

The AdaGrad is an improved stochastic gradient descent algorithm with each parameter learning rate. When training the dataset with scattered parameters, the AdaGrad is a useful SGD algorithm[35]. The elemental AdaGrad learning rate  $\eta$  is then multiplied with the vector  $G_{i,j}$  representing the crosswise of the exterior product matrix.

$$G = \sum_{\tau=1}^t g_{\tau} g_{\tau}^T$$

The letter  $g_{\tau} = \nabla Q_i(w)$  represents the gradient at the  $\tau$ th repetition, below is the diagonal:

$$G_{i,j} = \sum_{\tau=1}^t g_{\tau,j}^2$$

Therefore the  $G_{i,j}$  is upgraded at every iteration using the formula below:

$$w := w - \eta \text{diag}(G) - \frac{1}{2 \circ g}$$

Here the character  $\circ$  represents notation of element-wise product and the upgrade of every parameter is calculated by:

$$w_j := w_j - \frac{\eta}{\sqrt{G_{i,j}}} g_j$$



Considering that the divisor of the portion  $\sqrt{G_i} = \sqrt{\sum_{r=1}^t g_r^2}$  to be the  $\ell_2$  the measure of the previous derivatives, intense parameter restore will be silenced while tiny restores will be offered greater learning rate  $\eta$ . The Adam Gradient methodology is currently considered to be outstanding when implementing non-convex optimization.

However, Zeiler also suggests a different SGD optimization approach, the AdaDelta approach is an improved AdaGrad algorithm. In comparison with the AdaGrad approach, the AdaDelta approach enforced a confined window to acquire the previous gradients to join the calculation for the new gradient. Hence this provides a low destructive optimizing approach that could be utilized to fine-tune the learning rate vigorously rather than reducing the learning rate monotonically[62].

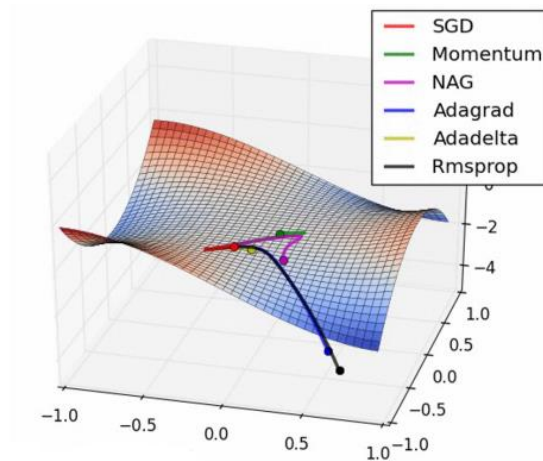
Tieleman and Hinton also came up with another approach that can also be utilized when performing optimization, the Root Mean Square Propagation - RMSProp. The RMSProp approach was introduced to fine-tune the learning rate for every parameter by splitting the learning rate for the standard deteriorating squared gradients[34]. The average can be calculated as follows:

$$v(w, t) := \gamma v(w, t - 1) + (1 - \gamma)(\nabla Q_i(w))^2$$

The character  $\gamma$  represents forgetting factor and Hinton also proposed that  $\gamma$  is to be set at 0.9 then the restoration parameters are:

$$w := w - \frac{\eta}{\sqrt{v(w, t)}} \nabla Q_i(w)$$

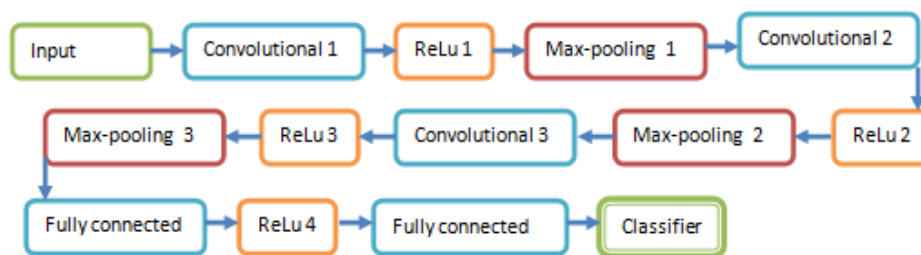
As proposed by Hinton et al. , the RMSProp approach has an outstanding capacity to adjust the learning rate for other applications. The improved optimization algorithm, Adaptive Moment Estimation was recommended in 2014 simply because it combines gradients and second moments of gradients. Adam is currently the most malleable and common SGD optimizer, hence used in this research paper[34]. The image below indicated the distinct computing ability of the above-discussed optimization approaches.



*Figure 20 Different SGD Performance*

#### 4.1.6 Convolutional Neural Network Architecture

The model below is composed of three convolutional layers and 2 fully connected layers. Our input is the segmented malaria image cells of  $125 \times 125 \times 3$  pixel resolution. The  $3 \times 3$  filters and 2 pixels strides have been used for the convolutional layers. The first convolutional layer is composed of 32 filters, 64 filters for the second, and the third 124 filters. The rectified linear units have been utilized to improve the learning rate during training. The maximum pooling of  $2 \times 2$  using the 2 pixels strides is performed after convolutional layers for outlining the outputs of bordering neuronal groups from the feature maps. The architecture can be illustrated as shown below:

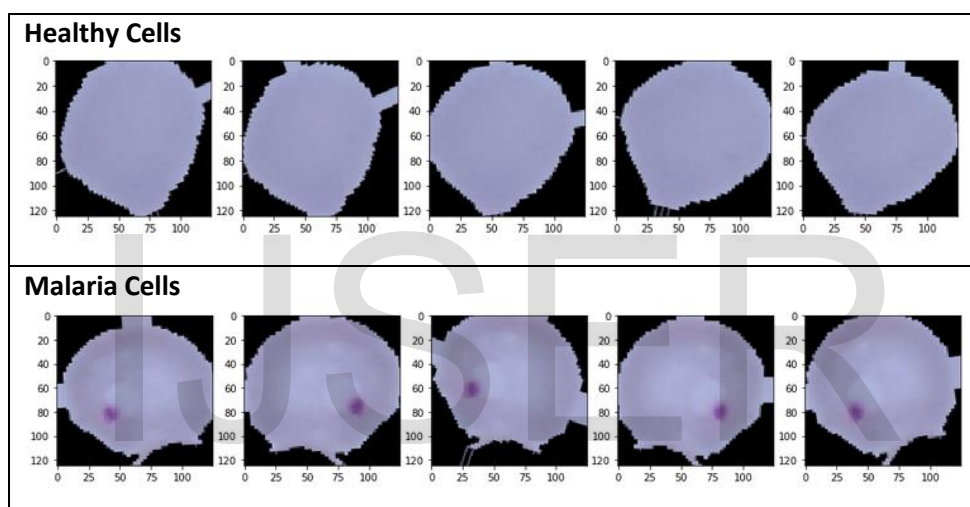


*Figure 21 CNN Architecture*

## 5 Chapter: Evaluation

### 5.1 Our Training Dataset

Our training data which has been used for this research experiment has been provided from the African Bible College health care center, and National Library for Medicine. The training dataset is composed of blood smear images where 14,000 are healthy cells and the other 14,000 cell images are infected. Below is the visual segmentation of the healthy (normal) cells and Infected (Malaria) cells.



*Figure 22 Blood smear cell images*

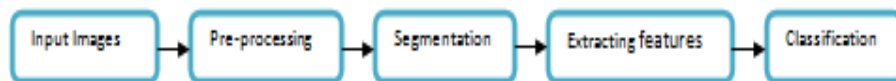
### 5.2 Experimental Environment

The training of all the models (Convolutional Neural Network, Support Vector Machine, and Random Forest) was carried out on an HP Pavilion CORE i7, with the following specification: Windows 8.1 pro, 64 bit, 8GB RAM and 2.60 GHz. The software used for the implementation is Jupiter notebook 6.0.2 from anaconda.

### 5.3 Support Vector Machine

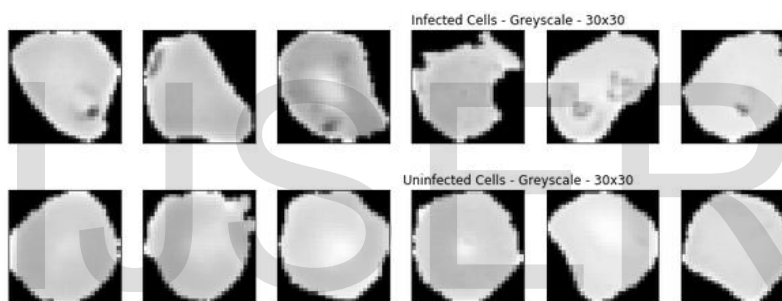
#### 5.3.1 Methodology

The diagram below illustrates how the support vector machine operates at every level of the process.



**Figure 23 SVM Process**

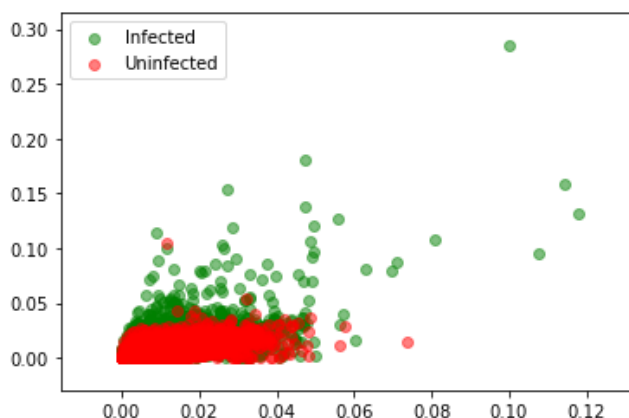
We implemented Support Vector Machine using our dataset composed of 28,000 blood smear images, the training was performed and different features were collected using the following techniques. In the first place, all the images were in color with different shapes, we had to turn all the images into grey color because classify whether the image is healthy or infected the process is entirely based on color differentiation through pixel examination. Later we after turning all the images into a greyscale we were able to visually spot the difference between the images then we had to resize all the images to 30×30 greyscale as illustrated below.



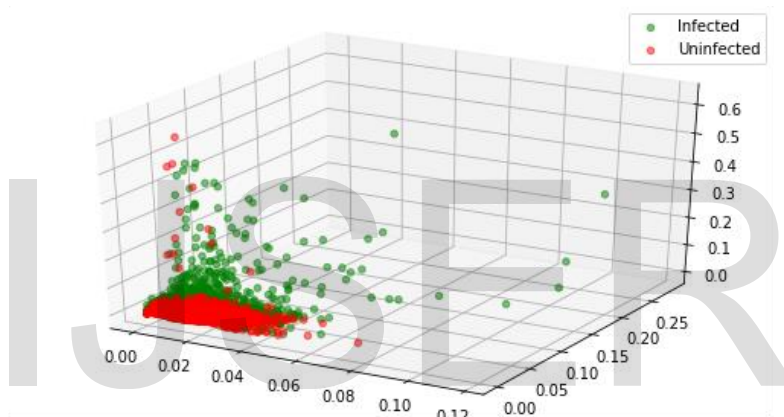
**Figure 24 Greyscale images**

After performing the above two tasks we had to start the feature extraction process using the color distribution from the images. Now with the understanding that our blood smear images (infected and uninfected cells) are different in color because of the stain in all infected cells. Then we move further to bin the colors 0 representing black and 1 representing white color, all this was done by using Numpy function for a histogram. Thereafter we divided count in every bin by total amount of counts to normalize every bin. To get rid of the background noise (the black pixels) we had to give a specification range of 0.1 to 1 in the function of the histogram.

### 5.3.2 Visualization of histogram



*Figure 25 Data Visualization*

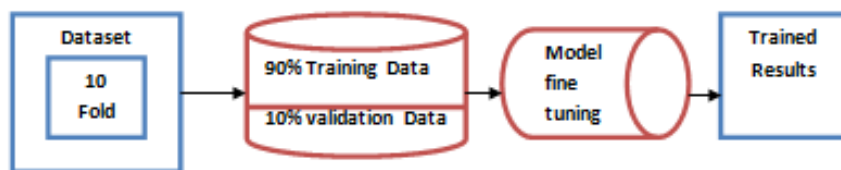


*Figure 26 Visualization*

Through implementing visualization we can notice that the observation cannot be linearly separate using a line or the plane. The alternative is to utilize the non-linear classifier like Support Vector Machine with the Radial BF kernel so that it will be able to single out among the healthy and infected blood smear images.

### 5.3.3 Model Evaluation

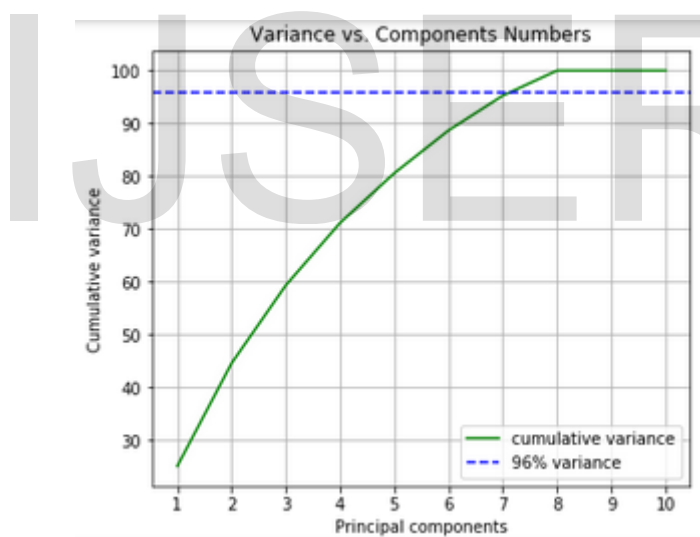
The 10-fold cross-validation method was implemented for the entire training dataset. We trained 90% of our dataset and the remaining 10% was used for testing purposes. The architecture below explains the process of 10-fold cross-validation implemented in Support Vector Machine.



*Figure 27 SVM Graphical Representation*

### 5.3.4 Support Vector Machine Tuning Hyperparameter

The grid search has been performed to ensure that it is in shape with multiple SVM models on our dataset to establish the right hyperparameter for cost and gamma for our kernel. We will make use of stratified cross-validation to make sure that there is dataset equality in every fold. However, we had to perform a preprocessing task to ensure that our dataset has been standardized and transformed by utilizing the principle components. We implemented the grid search to decide the number of principal components required to enclose 98% of the variance.



*Figure 28 Variance*

Based on the chart results we have seven principal components that will be utilized to clarify 96% variance. Later we develop a pipeline to be used for data standardization to every fold change it by utilizing PCA which is restricted to 7 and used to describe 96% variance.

### 5.3.5 Model Performance

We performed the classification process using two algorithms, Random forest classifier and the Support Vector Machine, and below are the results from the two models.

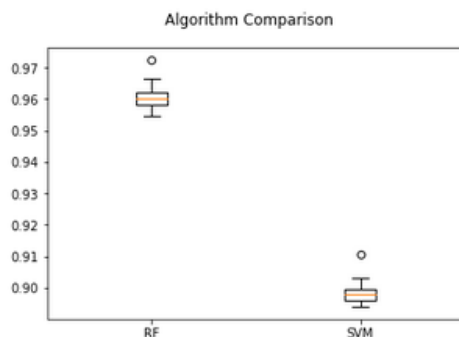


Figure 29 RF and SVM Performance Comparison

| Model                  | Accuracy             |
|------------------------|----------------------|
| Random Forest          | 0.9655990510083037   |
| Support Vector Machine | 0.8308936 (0.004666) |

Table 2 SVM and RF Accuracy Results

Then we evaluated our two models using precision, recall, and the f1 score.

### 5.3.6 Support Vector Machine Algorithm Performance

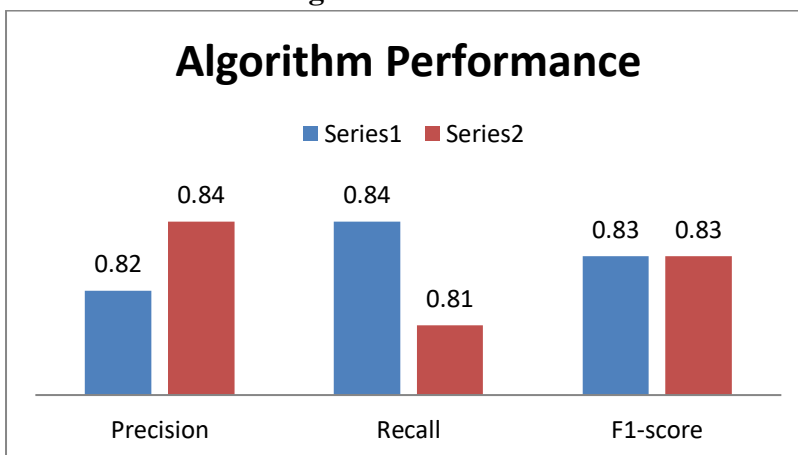
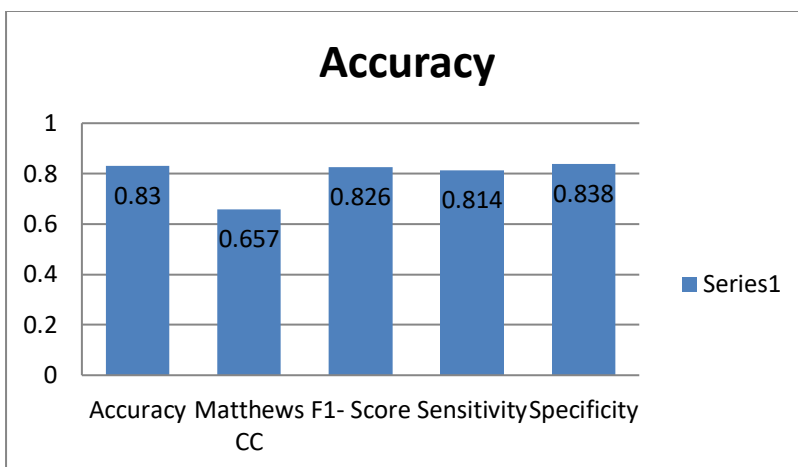
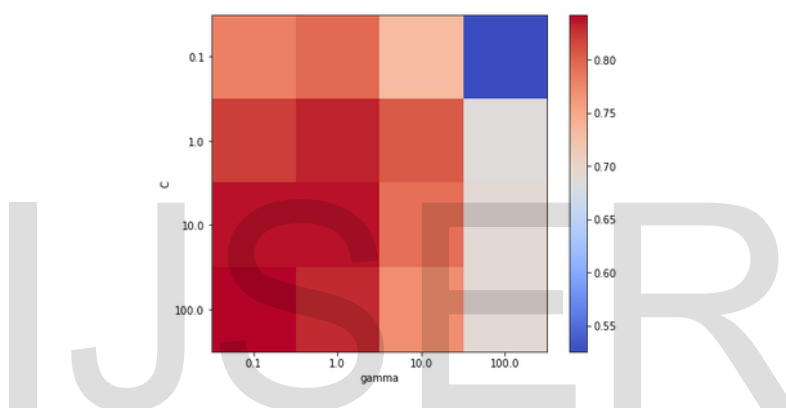


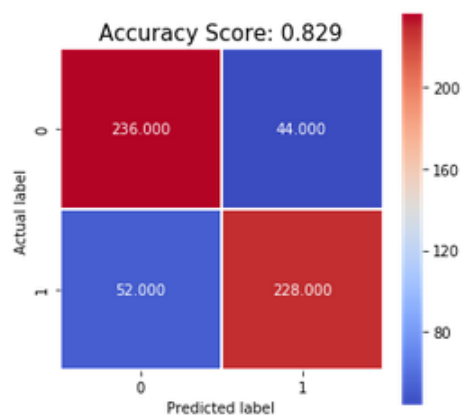
Figure 30 Algorithm Performance



*Figure 31 Accuracy and Performance*



**Figure 32 Gamma**

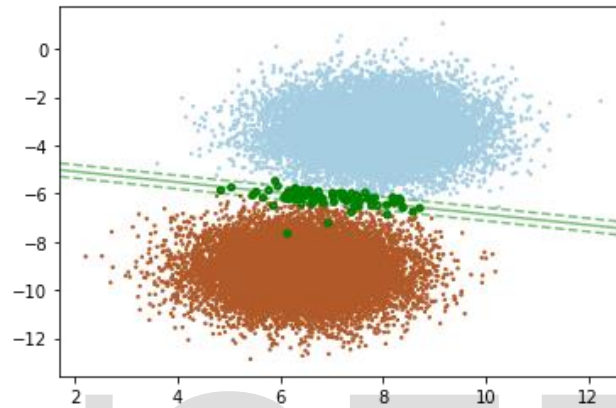


**Figure 33 Accuracy heatmap**

The heatmap above illustrates the gamma range and C together utilized to perform the grid search which resulted in the maximum mean accuracy of 82% and this was obtained by



utilizing  $SVC\_C = 100.0$  and  $SVC\_gamma=0.1$ . The model sensitivity 0.814 and specificity of 0.838. The left-hand side is the confusion matrix which is composed of true positive, false positive, false negative, and true negative. Based on the confusion matrix, we can see that there are false negatives and also the false positives with 52 and 22 in that order. This could be as a result of mislabeled cells or feature similarities. And due to this problem, we had to plot a linear classifier to explain our results.



*Figure 34 Support vectors and hyperplane*

The diagram above illustrates the Support Vector Machine with a linear classification of our dataset. The left side has been colored brown representing healthy cell images and the right-hand side is colored blue representing the unhealthy cell images. The green spots observed in between the infected and uninfected cells are representing the misclassified observations. If we take a close look at the left side colored brown, we will notice that there are some two green spots of misclassified cell images and this could be due to the feature classification problem. In between our observation we have three lines, two are dashed lines which represent support vectors and between them, we have one solid line representing optimal hyperplane. The hyperplane is used as the separating line between our two classes (healthy and unhealthy cell images). The hyperplane is calculated as  $n - 1$  dimensional subspace so for a 2-dimensional space it will be 1 dimension and for a 3 dimension, it will be 2-dimensional space. Mathematically the hyperplane can be written as below:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \dots + \beta_n * x_n = 0$$

Now let us have a look at the two-dimensional space and how it can be expressed mathematically. The hyperplane will look like this:  $\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 = 0$ . The spots on top of the line are represented using  $x_1$  and  $x_2$  with the following formula,  $\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 > 0$  and the ones below can be expressed in the same logic  $\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 < 0$ . Now based on the above expression we can write our mathematical as follows:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 > 0 \text{ if } k = 1 \text{ the blue}$$

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 < 0 \text{ if } k = -1 \text{ the brown}$$

Then the entire formula can be expressed as  $k * (\beta_0 + \beta_1 * x_1 + \beta_2 * x_2) > 0$ .

#### 5.4 Random Forest

The process of classifying disorganized data in the medical field is playing a very important role. The researchers have suggested different methods on how data imbalance can be managed like kernel function, sampling, and cost. But due to different challenges like overfitting and error rate have resulted in a rise in the misclassification during the learning process. In this research paper, we proposed ensemble methodology for classification process this is due to the ability of the method to perform the classification process which reduces the imbalance from our dataset through grouping which results in improving inaccuracy in performance resulting in reducing the overfitting problems. The method is capable of performing the classification process through the voting classifier, boosting, and bootstrap aggregation. The boosting technique operates through creating separate models that are used to correct faults that have been created by other models and AdaBoost is one of the boosting algorithms. Bootstrap aggregation collects multiple data samples and train each sample separately and the result prediction is then averaged across predictions the entire sub-models. We could site an example of the random forest where features are selected randomly for every split. Finally, the voting classifier operates through bringing together dissimilar classifiers and make use of the popular vote which is referred to us the soft vote.

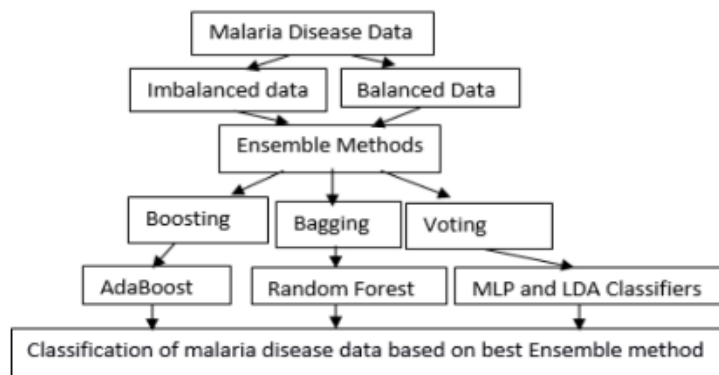


Figure 35 Random forest implementation

#### 5.4.1 Random Forest Performance

To check the performance of our algorithm we use the F1-score, Accuracy and the recall and the table below illustrates the 10-fold results based on accuracy.

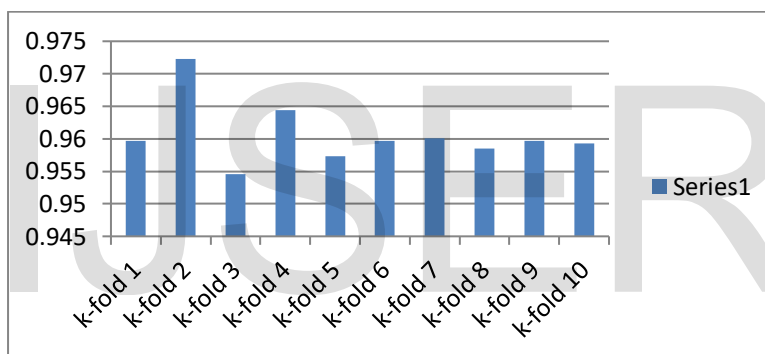


Figure 36 RF 10-fold

The bar chart above illustrates the accuracy percentage based on the 10-fold implementation of the malaria cell image dataset.

#### 6.1.2 Confusion matrix

|        |            | Forecasted/Predicted |                |
|--------|------------|----------------------|----------------|
|        |            | Infected             | Uninfected     |
| Actual | Infected   | True Positive        | False Negative |
|        | Uninfected | False Positive       | True Negative  |

Table 3 Confusion matrix

Therefore based on the table above we can compute the accuracy, precision, recall, and the f1-score in the following manner.

- Accuracy -  $[TP + TN] / [TP + FN + FP + TN]$
- Precision -  $[TP] / [TP + FP]$
- Recall -  $[TP] / [TP + FN]$
- F1-score -  $[2 \times \text{Precision} \times \text{Recall}] / [\text{Precision} + \text{Recall}]$

Initially, we used the ensemble method named random forest to perform the classification task. The table below illustrates the 10-fold performance which is based on the measure of precision, recall, and f1-score.

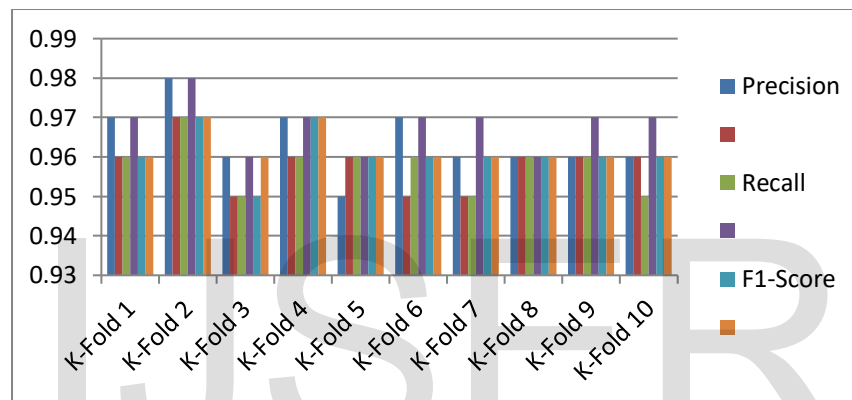


Table 4 Recall, Precision, F1-score 10 K-fold

The process of classification using random forest we managed to achieve overall accuracy of 96% using the malaria cell images and also the average score of 96% based on precision, f1-score, and recall. This indicates how the ensemble method called random forest is also capable of handling the overfitting problems and misclassification errors faced by the imbalanced dataset. This now leads us to the concept of thermodynamics called entropy.

#### 5.4.2 Entropy

We have used the Shannon Entropy as the score measure for our experiment to evaluate the splitting process. Therefore we will use letter  $t$  to represent test results from tree node and letter  $c$  denote a class we are predicting, hence these two ( $t$  and  $c$ ) different randomly scattered values ( $p(t_1), \dots, p(t_k)$ ) this representing  $t$  variable and

$(p(c_1), \dots, p(c_m))$  for  $c$  variable. Therefore the use of entropy is the measure the impurity at every node of the decision tree, now let us have a look at how this is performed.

$$H_c(\text{node}) \triangleq - \sum_{i=1}^m p(c_i) \log_2 p(c_i)$$

$$\triangleq -p_{\text{case}} \log_2 p_{\text{case}} - p_{\text{control}} \log_2 p_{\text{control}}$$

Here the proportion cases that arrive at the node is represented by  $p_{\text{case}}$  and  $p_{\text{control}}$ . Therefore the test entropy can be illustrated below.

$$H_T(\text{node}) \triangleq - \sum_{j=1}^k p(t_j) \log_2 p(t_j)$$

$$\triangleq -p_{\text{left}} \log_2 p_{\text{left}} - p_{\text{right}} \log_2 p_{\text{right}}$$

Here we have a proportion of two objects, the objects propagating to the right are represented using  $p_{\text{right}}$  and the objects propagating on the left side are denoted by  $p_{\text{left}}$ . The average or standard conditional entropy of the given class illustrated using the equation below.

$$H_{C|T}(\text{node}) \triangleq - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 p(c_i | t_j)$$

We can illustrate how the scoring measure is classified using the steps below.

$$\text{Score}(\text{node}) \triangleq H_C(\text{node}) - P_{\text{left}} H_C(\text{node}_{\text{left}}) - P_{\text{right}} H_C(\text{node}_{\text{right}})$$

$$\triangleq H_C(\text{node}) - H_{C|T}(\text{node})$$

$$\triangleq \mathbf{I}_C^T(\text{node})$$

This correlates to inequality of the present node and weighted impurity based on the child nodes produced. This takes as back to the purpose of tree induction whose target is impurity minimization performed at every test node. The purpose of the split is to reduce the class entropy at every node together with the child nodes. This is referred to us mutual

information  $I_C^T$  which is used to quantify the decrease of the unpredictability of the value  $c$  from the given value  $t$ . The given data size is upper constrained with the entropy  $H_C(node)$ , that part is responsive to the number of distributions of different classes which makes interpretation difficult. The implementation decision tree induction has witnessed that is fall in favor of the test at every node with a bigger number of the results. Other normalizations have been introduced like the gain ratio and symmetrical score measure.

In this research paper, we conducted our training using the randomly selected of 22,762 with an entropy of 1.0 from the root node. This indicates there is higher entropy on the root node which means there is a disorder in our training dataset. Now using the random forest classifier we utilized node splitting to reduce the entropy from our observations. The image below has been taken from the random forest that we generated to illustrate how entropy can be managed using a random forest splitting mechanism.

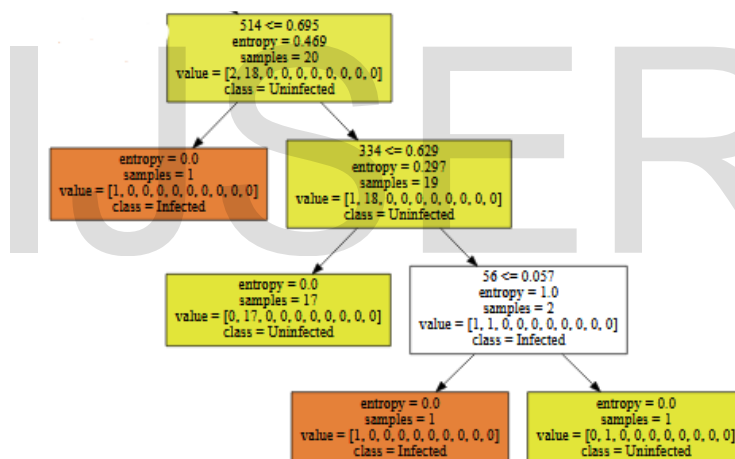


Figure 37 Representation of the splitting mechanism

Below is the graphical representation of entropy against training samples, based on this graphical representation entropy was high before the splitting process started. However, when we started the splitting process using 22,762 observations, 15,466 observations well classified to be infected and at this point, we can witness the decrease in our entropy from 1.0 going down to 0.896. The splitting process went further up to level 12 where the final 22 observations were classified to be infected with an entropy of 0.0.

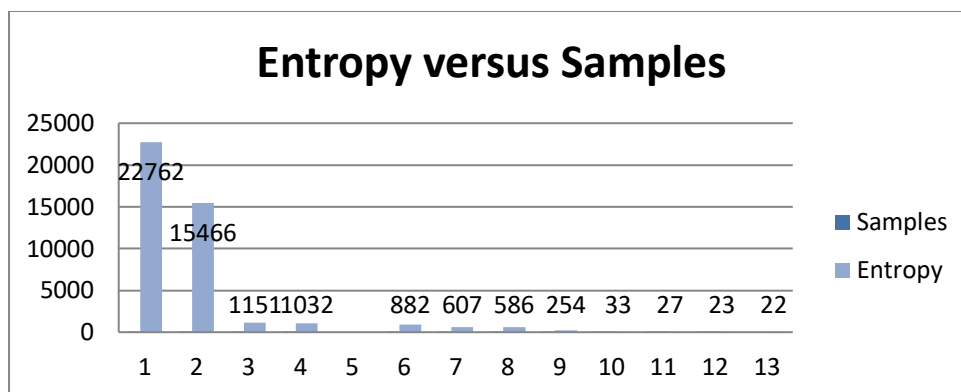


Figure 38 Entropy against training samples

Later we had to plot a validation curve based on the random forest to check the accuracy score of our model and cross-validation score. Now, based on the graphical illustration below we can see that the number of trees increases through node splitting, the training score increases too which also results in to decrease in entropy. The node splitting operates by randomly selecting the features and perform feature investigation.

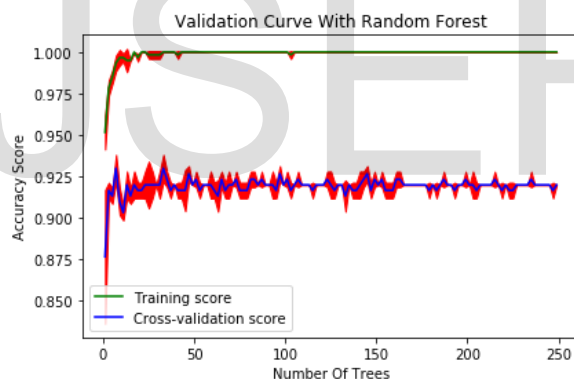


Figure 39 Validation curve

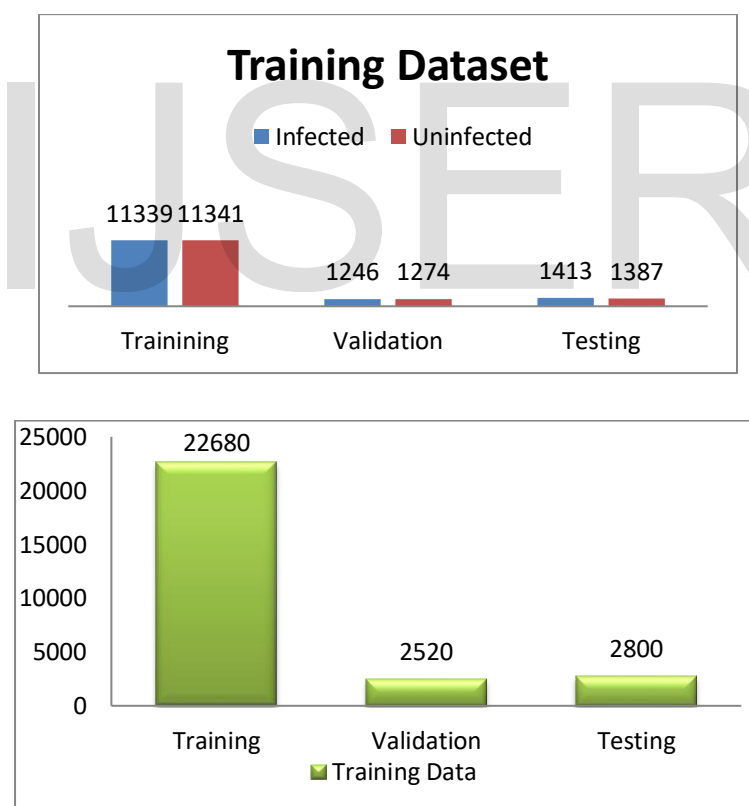
## 5.5 Convolutional Neural Network

In this section of this research paper, we are going to discuss the training results based on the state of the art Convolutional Neural Network. ConvNet is widely used for image processing due to its capability to perform image classification through edge detection and pattern recognition. The experiments were conducted using Jupyter notebook with the following libraries: Tensorflow for classification and discovery, matplotlib plotting library, pandas for data manipulation and analysis, cv2 used to perform computer

vision tasks, sklearn used for mining and analysis of data and NumPy for scientific operations.

The first step of implementing the Convolutional Neural Network using the blood smear images was to import the libraries listed above, then followed by loading the training dataset. Before loading our dataset we had to make sure that it is well labeled and balanced. Our dataset is composed of 28,000 blood smear images, where 14,000 are infected and the remaining 14,000 contains uninfected image cells. Thereafter we constructed a data frame that we utilized to make datasets for our training using pandas and NumPy libraries.

After building our model we used 70% for training, 10% for validation, and 20% as out testing data. During the entire training process we took advantage of the train together with validation we observed the model performance.



*Figure 40 Training Dataset*



Thereafter we moved to perform some statistics on our training dataset taking into consideration that the blood smear images might have different orientations, below is the statistics summary.

*Minimum dimensions* - [ 46 46 3]

*Average dimensions* - [132.97460317 132.45343915 3]

*Median dimensions* - [130. 130. 3.]

*Maximum dimensions* - [385 394 3]

After performing some statistical operations on our dataset, we further moved on to implement image resizing. We utilized parallel processing implementation just to accelerate the training process and performed 112x112 pixels image resize. Below are the obtained results:

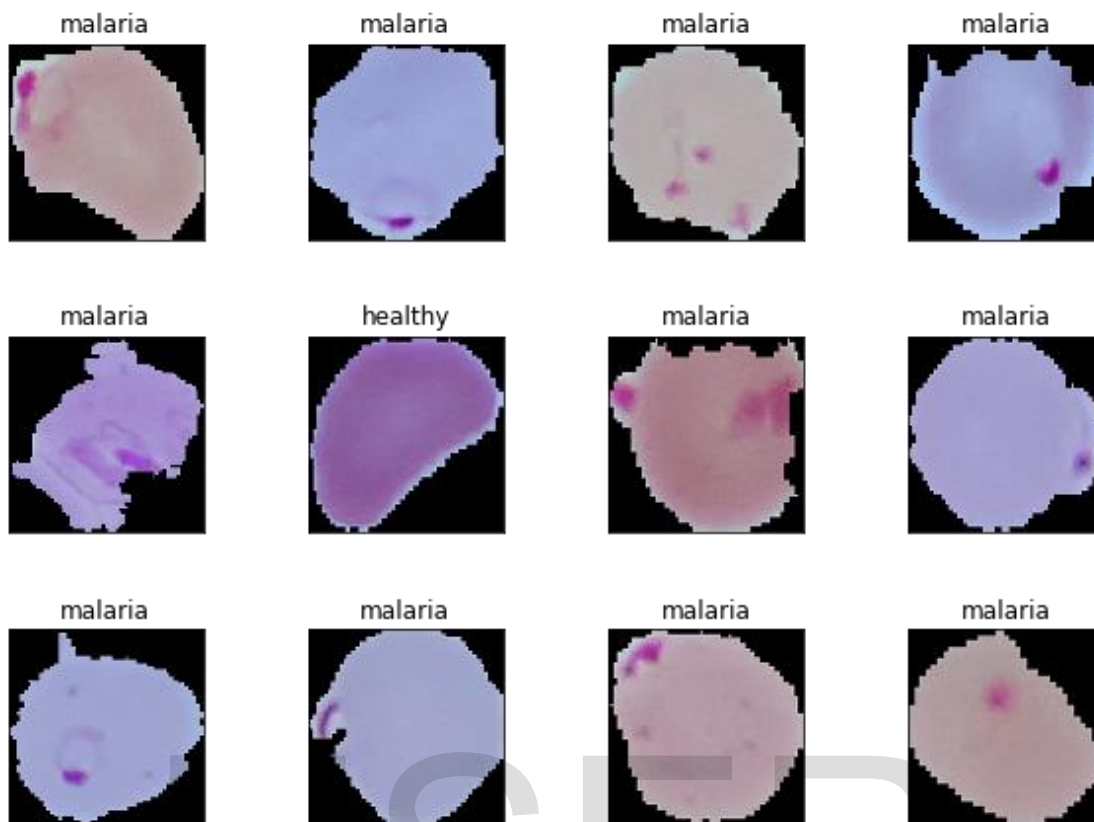
*Training images* - (22680, 112, 112, 3)

*Validation images* - (2520, 112, 112, 3)

*Testing Images* - (2800, 112, 112, 3)

#### 7.1.1 Data Visualization

Thereafter we had to visualize the images to see how our dataset looks like with the specified dimensions.



*Figure 41 Malaria Cell images*

The image above clearly illustrates that the difference involving the infected and uninfected cell images. Now we constructed a deep learning model that has been utilized to perform the task of pattern recognition using the 112x112 pixels images, 30 epochs and class number of 2 with TensorFlow version 211.0. We also encoded our dataset classes where zero represent uninfected cells and 1 for infected cells.

#### **5.5.1 The Training model configurations**

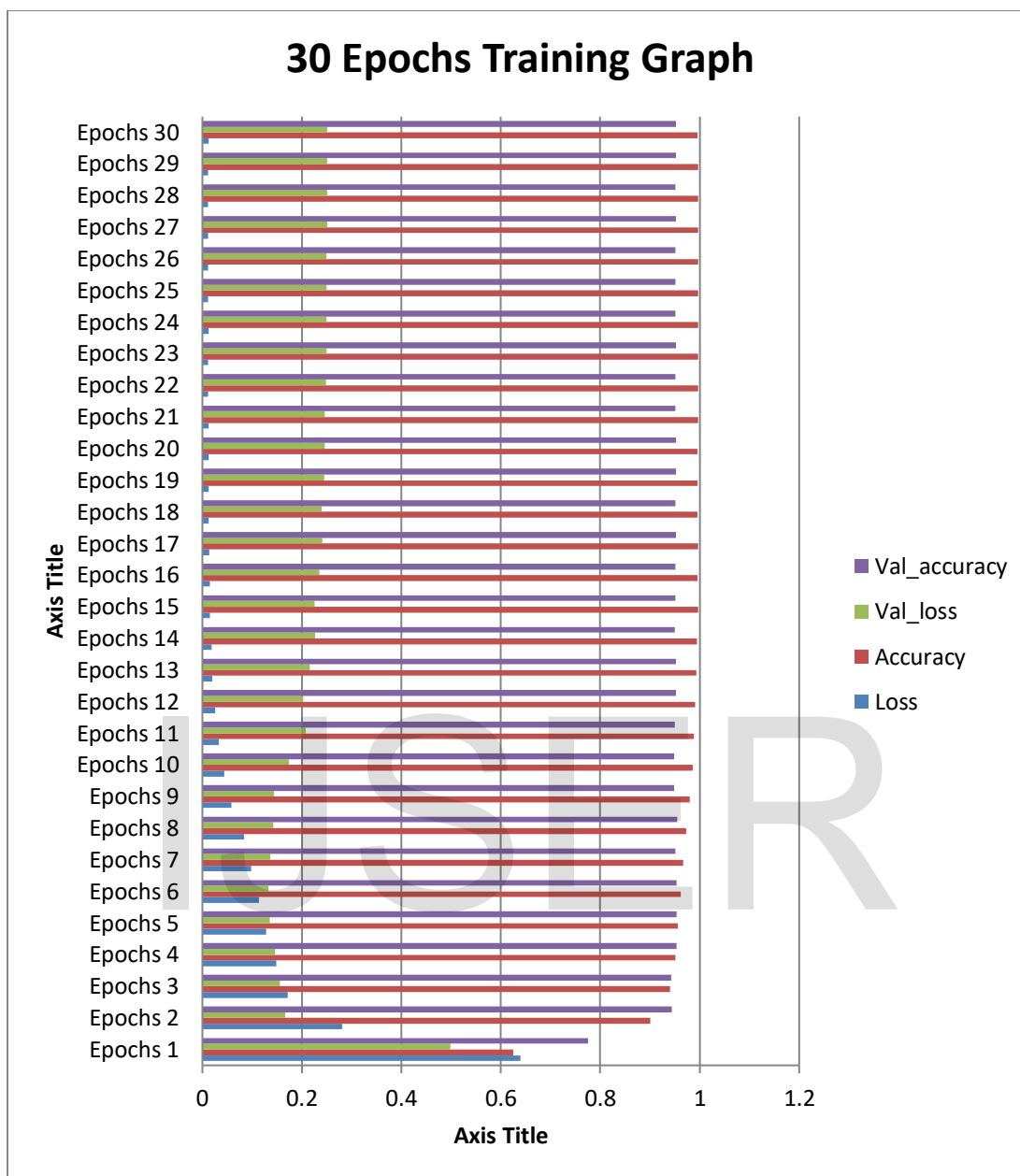
Here we will explore our Convolutional Neural Network model focusing on the architectural configurations. The table below illustrates the basic configurations that have been utilized within our training model.

|              | <b>Filters</b>      | <b>Kernel size</b> | <b>Activation</b> | <b>Padding</b>      | <b>Pool size</b> |
|--------------|---------------------|--------------------|-------------------|---------------------|------------------|
| Conv. 1      | 32                  | 3,3                | ReLu              | Same                | 2,2 max pool     |
| Conv. 2      | 64                  | 3,3                | ReLu              | Same                | 2,2 max pool     |
| Conv. 3      | 128                 | 3,3                | ReLu              | Same                | 2,2 max pool     |
|              |                     |                    |                   |                     |                  |
| <b>Dense</b> | <b>Dropout rate</b> | <b>Activation</b>  | <b>Optimizer</b>  | <b>Loss</b>         | <b>Metrics</b>   |
| 256          | 0.4                 | Sigmoid            | Adam              | Binary_crossentropy | Accuracy         |

*Table 5 CNN Model configuration*

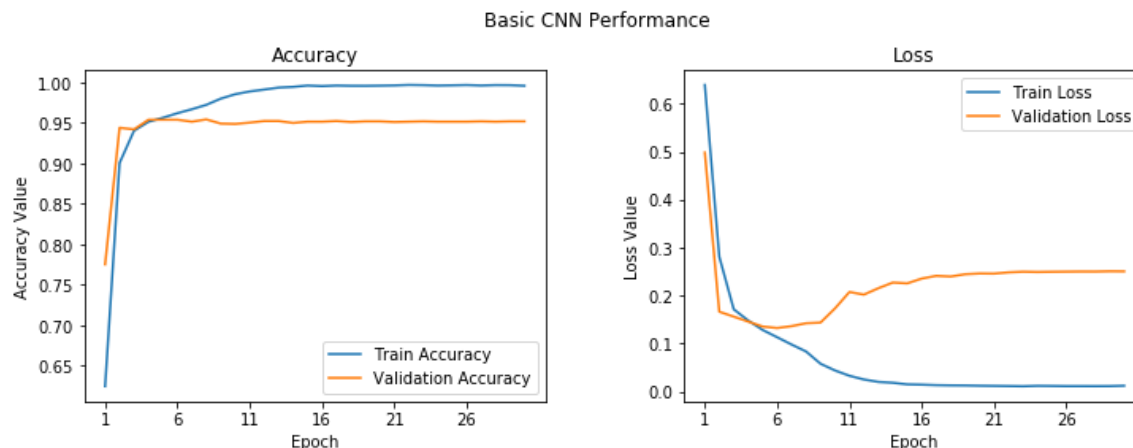
In our first convolution we have used the 32 filters, 64 filters for second convolution, and 128 for the third convolution with the following configurations, the kernel size of 3,3, Relu activation, and maximum pooling of 2,2. We have also used the dense of 256, the dropout rate of 0.4, sigmoid activation, Adam optimizer and for the loss, we have utilized the binary cross-entropy.

During our training, we have implemented 30 epochs and the bar graph below illustrates the results of the 30 epochs. The bar graph below illustrates 30 epochs implementation based on loss, accuracy, validation loss, and validation accuracy. If we use the bar graph above to explain the results, we can notice that on the first epochs the loss was high and our training model accuracy was low. As the number of epochs continued to increase the training accuracy also increases too and the loss starts to go down. However, upon reaching epochs number 12 our model accuracy was maintained at 99% until the final epochs.



*Figure 42 CNN 30 Epochs*

Therefore using the information above we managed to plot the accuracy and loss graphs to illustrate the relationship between the two during our model training.

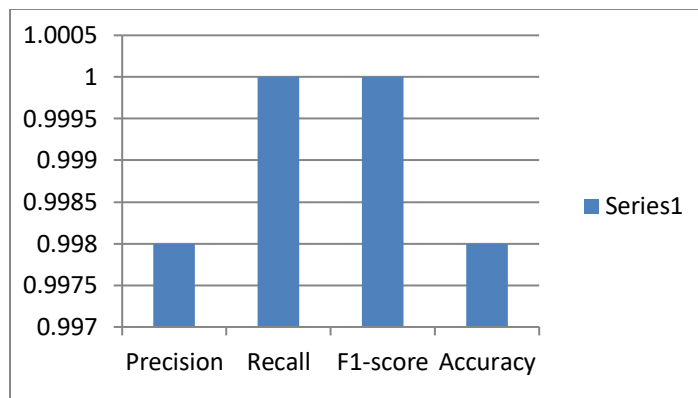


*Figure 43 CNN Performance Graphs*

The blue line on the left side represents training accuracy and the orange line is for validation accuracy. Therefore as the number of epochs increases, accuracy and validation accuracy improves too. This is referred to as the number of our training circles or the number of algorithm iteration and for our algorithm, we have implemented 30 epochs. The right-hand side is a plot of the loss graph where the blue line represents the training loss which decreases as the number of epochs increases and the validation loss too. We plot the training accuracy together with the training loss to illustrate what is happening when the number of epochs increases. The blue dotted curved line represents training accuracy whilst the bottom dotted line represents training loss.

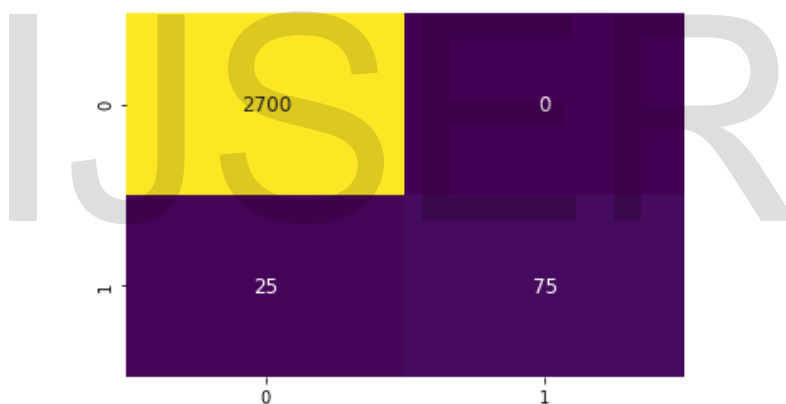
### 5.5.2 CNN Algorithm Performance

Thereafter we had to generate the classification report for the training model and we utilized the precision, recall, f1-score, and accuracy.



*Figure 44 CNN Algorithm Performance*

The graph above illustrates the summary results of our training model based on precision, accuracy, recall, and f1-score. The recall and f1-score achieved 100% and the accuracy and precision achieved 99%. Then we moved on to plot the heatmap for the confusion matrix which based on finding the true positive, true negative, false positive, and false negative.



*Figure 45 CNN Heatmap*

The map shows that 2700 have been classified as true positive, 25 false negatives, 0 false positives, and 75 true negatives. we can further display the results in a table format.

## 6 Chapter: Conclusion

In conclusion, the research paper aims at implementing an early intervention system that can be utilized in the process of malaria detection in the health care sector. There are currently other methods that are in place to reduce to help in the fight against malaria, environmental care, rural health care centers, and spraying of the insect sides. Recently, the World Health Organization have announced that it will be carrying the first pilot malaria vaccine in Malawi in trying to combat malaria.

As we can see, all these other methods fall under two categories, prevention and cure hence there is a need to put in place a method that can be utilized for early warning through the use of machine learning algorithms. We have implemented three algorithms; Random Forest which operates by constructing multiple decision trees through the spilling of the root node into two child nodes with 96% accuracy, support vector machine operates through separating the observations using the hyperplane and group them according to their classes with 83% accuracy, and finally, we implemented convolutional neural networks which operate through edge detection and pattern recognition when performing the classification process with an outstanding accuracy of 99%.

In our introduction, we discussed the impact of malaria in African countries mainly in the sub-Sahara regions where there are reports of high figures. We further look at the current methods in place used for malaria diagnosis. Microscopy diagnosis is the currently adopted method for malaria diagnosis approved by the World Health Organization. The process involves collecting the blood samples from the patients which are smeared on plate glass. The plate glass is put under the microscope to scan for the presence of malaria parasites. As we can see the process involves human expertise to perform this task successfully and sometimes their chances of providing wrong diagnosis results plus it is time-consuming.

This is where our motivation comes in to incorporate the machine learning algorithms so that they can be utilized in the medical field. To implement this research we had to make

an overview of the image processing techniques and how they can be used for the benefit of our research paper. An understanding of deep learning was important to have an insight into how we are going to process our datasets which are composed of 28,000 cell images. Then after performing the classification process, we measured the accuracy of our training models basing precision, recall, and f1-score.

Our random forest model achieved the f1-score of 96%, precision 97%, and the recall of 96% followed by support vector machine which achieved an f1-score of 83%, precision 82%, and 84% recall. We utilized the 10-fold cross-validation when performing the classification process and as we can see random forest performed better the classification process compared to support vector machines. Finally, we implemented the deep learning model using convolution neural networks which have proved to perform better the image analysis tasks. Through using ConvNets we were able to achieve better performance with f1-score of 100%, precision 99% and recall 100%. The results have proved that ConvNet performs better when it comes to image processing task is we compare it to the other two models in this report. This is true because ConvNets learns through edge detection and pattern recognition when performing the classification process. Pattern recognition involves dealing with image pixels by performing pooling which results in the reduction of spatial size. We also utilized the binary\_crossentropy to perform the predictions and classification of our output results.

Therefore, the results have proved that the machine learning algorithms can be incorporated within the health care system to automate the process of malaria detection through deep learning model. The model has successfully performed the malaria image dataset classification with 99% accuracy results. The authors recommend deep learning models should be utilized within the health care sector to speed up the diagnosis process and also act as the early warning system. This also has a positive impact on the economy of the nation as it will assist in preventing disease outbreaks which result in spending resources that could have been used in other areas of development. Future work will focus on developing a portable mobile application that can be installed on mobile devices to reduce the number of patients visiting health care for diagnosis.



## 7 Bibliography

- [1] World Health Organization, *World Malaria Report 2015*. World Health Organization, 2016.
- [2] "World malaria report 2019." <https://www.who.int/news-room/feature-stories/detail/world-malaria-report-2019> (accessed Mar. 27, 2020).
- [3] J. C. Reed, *Malawi Malaria Indicator Survey 2017*. 2018.
- [4] "Malawi Launches Health Sector Strategic Plan II." <http://www.healthpolicyplus.com/MalawiHSSP.cfm> (accessed Mar. 27, 2020).
- [5] "malawi health sector strategic plan 2017 - Google Search." [https://www.google.com/search?sxsrf=ALeKk01yIgsBD0vNPQfEkCDV8ULAYdxQbw%3A1585238462650&ei=vtF8XtSjJ4yasAeP3piQAQ&q=malaria+health+sector+strategic+plan+2017&oq=malaria+health+sector+&gs\\_l=psy-ab.1.1.0I3j0i22i30I5.7223.12845..18203...0.1..0.899.6792.2-1j10j1j2j2....2..0....1..gws-wiz.....0i71j35i39j0i273i395j0i67i395j0i7i30i395j0i67j0i395.INRLGaf5EDI](https://www.google.com/search?sxsrf=ALeKk01yIgsBD0vNPQfEkCDV8ULAYdxQbw%3A1585238462650&ei=vtF8XtSjJ4yasAeP3piQAQ&q=malaria+health+sector+strategic+plan+2017&oq=malaria+health+sector+&gs_l=psy-ab.1.1.0I3j0i22i30I5.7223.12845..18203...0.1..0.899.6792.2-1j10j1j2j2....2..0....1..gws-wiz.....0i71j35i39j0i273i395j0i67i395j0i7i30i395j0i67j0i395.INRLGaf5EDI) (accessed Mar. 27, 2020).
- [6] "Malaria Diagnosis: A Brief Review." <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2688806/> (accessed Mar. 27, 2020).
- [7] "TDR | Malaria Rapid Diagnostic Test Performance - results of WHO product testing of malaria RDTs: Round 2 (2009)," *WHO*. [https://www.who.int/tdr/publications/tdr-research-publications/rdt\\_round2/en/](https://www.who.int/tdr/publications/tdr-research-publications/rdt_round2/en/) (accessed Mar. 27, 2020).
- [8] "malaria clinical diagnostic methods pdf - Google Search." [https://www.google.com/search?sxsrf=ALeKk00cdQCmQac-Rnu5ff6sJ6kzMxHRwQ%3A1585245044010&source=hp&ei=c-t8XrrtOoeUkwW6u5OQBw&q=malaria+clinical+diagnostic+methods+pdf&oq=malaria+clinical+diagnosis+methods&gs\\_l=psy-ab.1.1.33i22i29i30I2.3077.12058..15862...1.0..0.1178.13601.2-1j27j0j3j1j2....2..0....1..gws-wiz.....35i39j0i67j0i273j0i395j0i10j0i22i30j33i10.zZEAnstkyoo](https://www.google.com/search?sxsrf=ALeKk00cdQCmQac-Rnu5ff6sJ6kzMxHRwQ%3A1585245044010&source=hp&ei=c-t8XrrtOoeUkwW6u5OQBw&q=malaria+clinical+diagnostic+methods+pdf&oq=malaria+clinical+diagnosis+methods&gs_l=psy-ab.1.1.33i22i29i30I2.3077.12058..15862...1.0..0.1178.13601.2-1j27j0j3j1j2....2..0....1..gws-wiz.....35i39j0i67j0i273j0i395j0i10j0i22i30j33i10.zZEAnstkyoo) (accessed Mar. 27, 2020).
- [9] "world health report 2019 malawi malaria - Google Search." [https://www.google.com/search?sxsrf=ALeKk02pG1t2-tn8NZ5wrMrhetqmvfxSWA%3A1585243702238&source=hp&ei=NuZ8XuXPC4T1kwWBsqv oBA&q=world+health+report+2019+malawi+malaria&oq=world+health+report+2019+malawi+malaria&gs\\_l=psy-ab.3...2789.24067..24662...1.0..0.1467.16351.2-7j18j3j5j2j2....2..0....1..gws-wiz.....35i39j0i273j0i67j0i273i395j0i67i395j0i395j0i67i70i257j0i203i395j0i22i30i395j0i22i30j33i160j33i22i29i30.zYN0V4Y8Afw&ved=0ahUKEwjIwoH-1LjoAhWE-qQKHQHZCk0Q4dUDCAg&uact=5](https://www.google.com/search?sxsrf=ALeKk02pG1t2-tn8NZ5wrMrhetqmvfxSWA%3A1585243702238&source=hp&ei=NuZ8XuXPC4T1kwWBsqv oBA&q=world+health+report+2019+malawi+malaria&oq=world+health+report+2019+malawi+malaria&gs_l=psy-ab.3...2789.24067..24662...1.0..0.1467.16351.2-7j18j3j5j2j2....2..0....1..gws-wiz.....35i39j0i273j0i67j0i273i395j0i67i395j0i395j0i67i70i257j0i203i395j0i22i30i395j0i22i30j33i160j33i22i29i30.zYN0V4Y8Afw&ved=0ahUKEwjIwoH-1LjoAhWE-qQKHQHZCk0Q4dUDCAg&uact=5) (accessed Mar. 27, 2020).
- [10] I. of M. (US) C. on the E. of A. Drugs, K. J. Arrow, C. Panosian, and H. Gelband, *A Brief History of Malaria*. National Academies Press (US), 2004.
- [11] "WHO | Microscopy," *WHO*. <http://www.who.int/malaria/areas/diagnosis/microscopy/en/> (accessed Mar. 27, 2020).
- [12] "WHO | Basic malaria microscopy – Part I: Learner's guide. Second edition," *WHO*. <https://www.who.int/malaria/publications/atoz/9241547820/en/> (accessed Mar. 27, 2020).
- [13] "Thick and Thin Blood Smear for Malaria diagnosis - Learn Microbiology Online." <http://microbeonline.com/microscopic-diagnosis-of-malaria/> (accessed Mar. 27, 2020).

- [14] "Understanding Support Vector Machines(SVM) algorithm (along with code)."  
<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/> (accessed Mar. 27, 2020).
- [15] "Support Vector Machine — Simply Explained - Towards Data Science."  
<https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496> (accessed Mar. 27, 2020).
- [16] R. Pupale, "Support Vector Machines(SVM) — An Overview," *Medium*, 11-Feb-2019.  
<https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989> (accessed Mar. 27, 2020).
- [17] "sklearn.svm.LinearSVC — scikit-learn 0.22.2 documentation." <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html> (accessed Mar. 27, 2020).
- [18] W. Koehrsen, "Random Forest Simple Explanation," *Medium*, 27-Dec-2017.  
<https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d> (accessed Mar. 27, 2020).
- [19] "1.10. Decision Trees — scikit-learn 0.22.2 documentation." <https://scikit-learn.org/stable/modules/tree.html> (accessed Mar. 27, 2020).
- [20] "How is Splitting Decided for Decision Trees?," *Displayr*, 02-Aug-2018.  
<https://www.displayr.com/how-is-splitting-decided-for-decision-trees/> (accessed Mar. 27, 2020).
- [21] Z.-H. Zhou, "Ensemble Learning.," *Encyclopedia of biometrics*, vol. 1, pp. 270–273, 2009.
- [22] "3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.22.2 documentation." <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (accessed Mar. 27, 2020).
- [23] "Random Forest Regression | Turi Machine Learning Platform User Guide."  
[https://turi.com/learn/userguide/supervised-learning/random\\_forest\\_regression.html](https://turi.com/learn/userguide/supervised-learning/random_forest_regression.html) (accessed Mar. 27, 2020).
- [24] "Random Forests Classifiers in Python," *DataCamp Community*, 16-May-2018.  
<https://www.datacamp.com/community/tutorials/random-forests-classifier-python> (accessed Mar. 27, 2020).
- [25] "A Beginner's Guide to Convolutional Neural Networks (CNNs) | Pathmind."  
<https://pathmind.com/wiki/convolutional-network> (accessed Mar. 27, 2020).
- [26] "An intuitive guide to Convolutional Neural Networks," *freeCodeCamp.org*, 24-Apr-2018.  
<https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/> (accessed Mar. 27, 2020).
- [27] "neuralnetworksanddeeplearning.pdf." Accessed: 25-Mar-2020. [Online]. Available: <http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf>.
- [28] "Convolutional neural network," *Wikipedia*. 10-Mar-2020, Accessed: 25-Mar-2020. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Convolutional\\_neural\\_network&oldid=944917484](https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=944917484).
- [29] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," *Medium*, 17-Dec-2018. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (accessed Mar. 27, 2020).
- [30] M. A. Nielsen, "Neural Networks and Deep Learning," 2015, Accessed: 25-Mar-2020. [Online]. Available: <http://neuralnetworksanddeeplearning.com>.

- [31] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1\_53.
- [32] "An intuitive guide to Convolutional Neural Networks."  
<https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/> (accessed Mar. 27, 2020).
- [33] "3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.22.2 documentation." <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (accessed Mar. 27, 2020).
- [34] "Understanding RMSprop — faster neural network learning."  
<https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a> (accessed Mar. 27, 2020).
- [35] "[1212.5701] ADADELTA: An Adaptive Learning Rate Method."  
<https://arxiv.org/abs/1212.5701> (accessed Mar. 26, 2020).
- [36] "An Overview of Regularization Techniques in Deep Learning (with Python code)," *Analytics Vidhya*, 19-Apr-2018.  
<https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/> (accessed Mar. 27, 2020).
- [37] J. Brownlee, "A Gentle Introduction to Batch Normalization for Deep Neural Networks," *Machine Learning Mastery*, 15-Jan-2019. <https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/> (accessed Mar. 27, 2020).
- [38] "(PDF) Associative learning of scene parameters from images."  
[https://www.researchgate.net/publication/44649471\\_Associative\\_learning\\_of\\_scene\\_parameters\\_from\\_images](https://www.researchgate.net/publication/44649471_Associative_learning_of_scene_parameters_from_images) (accessed Mar. 26, 2020).
- [39] D. Marr, "Visual Information Processing: The Structure and Creation of Visual Representations," *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 290, pp. 199–218, Aug. 1980, doi: 10.1098/rstb.1980.0091.
- [40] C. J. Taylor, T. F. Cootes, A. Lanitis, G. Edwards, P. Smyth, and A. C. Kotcheff, "Model-based interpretation of complex and variable images.," *Philos Trans R Soc Lond B Biol Sci*, vol. 352, no. 1358, pp. 1267–1274, Aug. 1997.
- [41] M. de Bruijne, "Machine learning approaches in medical image analysis: From detection to diagnosis," *Med Image Anal*, vol. 33, pp. 94–97, 2016, doi: 10.1016/j.media.2016.06.032.
- [42] S. Wang and R. M. Summers, "Machine learning and radiology," *Med Image Anal*, vol. 16, no. 5, pp. 933–951, Jul. 2012, doi: 10.1016/j.media.2012.02.005.
- [43] I. Nourtdinov *et al.*, "Machine learning classification with confidence: Application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression," *NeuroImage*, vol. 56, no. 2, pp. 809–813, May 2011, doi: 10.1016/j.neuroimage.2010.05.023.
- [44] J.-Z. Cheng *et al.*, "Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans," *Scientific Reports*, vol. 6, no. 1, pp. 1–13, Apr. 2016, doi: 10.1038/srep24454.
- [45] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, "Histopathological Image Analysis: A Review," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 147–171, 2009, doi: 10.1109/RBME.2009.2034865.
- [46] B. Lessmann, T. W. Nattkemper, V. H. Hans, and A. Degenhard, "A method for linking computed image features to histological semantics in neuropathology," *Journal of*

- Biomedical Informatics*, vol. 40, no. 6, pp. 631–641, Dec. 2007, doi: 10.1016/j.jbi.2007.06.007.
- [47] S. Vanderbeck, J. Bockhorst, R. Komorowski, D. E. Kleiner, and S. Gawrieh, "Automatic classification of white regions in liver biopsies by supervised machine learning," *Human Pathology*, vol. 45, no. 4, pp. 785–792, Apr. 2014, doi: 10.1016/j.humpath.2013.11.011.
- [48] "Automated detection of circulating tumor cells with naive Bayesian classifiers - Svensson - 2014 - Cytometry Part A - Wiley Online Library." <https://onlinelibrary.wiley.com/doi/full/10.1002/cyto.a.22471> (accessed Mar. 26, 2020).
- [49] B. Gopinath and N. Shanthi, "Development of an Automated Medical Diagnosis System for Classifying Thyroid Tumor Cells using Multiple Classifier Fusion," *Technol Cancer Res Treat*, vol. 14, no. 5, pp. 653–662, Oct. 2015, doi: 10.7785/tcrt.2012.500430.
- [50] A. Gertych *et al.*, "Machine learning approaches to analyze histological images of tissues from radical prostatectomies," *Computerized Medical Imaging and Graphics*, vol. 46, pp. 197–208, Dec. 2015, doi: 10.1016/j.compmedimag.2015.08.002.
- [51] X. Tan, J. X. Huang, and A. An, "Ranking Documents Through Stochastic Sampling on Bayesian Network-based Models: A Pilot Study," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, 2016, pp. 961–964, doi: 10.1145/2911451.2914750.
- [52] H. Ayadi, M. T. Khemakhem, J. X. Huang, M. Daoud, and M. B. Jemaa, "Learning to Re-rank Medical Images Using a Bayesian Network-Based Thesaurus," in *Advances in Information Retrieval*, Cham, 2017, pp. 160–172, doi: 10.1007/978-3-319-56608-5\_13.
- [53] C. Morioka *et al.*, "Automatic Classification of Ultrasound Screening Examinations of the Abdominal Aorta," *J Digit Imaging*, vol. 29, no. 6, pp. 742–748, 2016, doi: 10.1007/s10278-016-9889-6.
- [54] "(PDF) Automatic System for Classification of Erythrocytes Infected with Malaria and Identification of Parasite's Life Stage." [https://www.researchgate.net/publication/257743971\\_Automatic\\_System\\_for\\_Classification\\_of\\_Erythrocytes\\_Infected\\_with\\_Malaria\\_and\\_Identification\\_of\\_Parasite's\\_Life\\_Stage](https://www.researchgate.net/publication/257743971_Automatic_System_for_Classification_of_Erythrocytes_Infected_with_Malaria_and_Identification_of_Parasite's_Life_Stage) (accessed Mar. 27, 2020).
- [55] R. Verma, A. Tiwari, S. Kaur, G. C. Varshney, and G. P. Raghava, "Identification of Proteins Secreted by Malaria Parasite into Erythrocyte using SVM and PSSM profiles," *BMC Bioinformatics*, vol. 9, no. 1, p. 201, Apr. 2008, doi: 10.1186/1471-2105-9-201.
- [56] R. Kuang, J. Gu, H. Cai, and Y. Wang, "Improved prediction of malaria degradomes by supervised learning with SVM and profile kernel," *Genetica*, vol. 136, no. 1, pp. 189–209, May 2009, doi: 10.1007/s10709-008-9336-9.
- [57] W. Yin, S. Garimalla, A. Moreno, M. Galinski, and M. Styczynski, "A tree-like Bayesian structure learning algorithm for small-sample datasets from complex biological model systems," *BMC systems biology*, vol. 9, p. 49, Aug. 2015, doi: 10.1186/s12918-015-0194-7.
- [58] L. Scotti, H. Ishiki, F. J. B. Mendonca Junior, M. S. da Silva, and M. T. Scotti, "Artificial Neural Network Methods Applied to Drug Discovery for Neglected Diseases," 2015. <https://www.ingentaconnect.com/content/ben/cchts/2015/00000018/00000008/art00012> (accessed Mar. 26, 2020).
- [59] D. K. Das, M. Ghosh, M. Pal, A. K. Maiti, and C. Chakraborty, "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Micron*, vol. 45, pp. 97–106, Feb. 2013, doi: 10.1016/j.micron.2012.11.002.
- [60] K. Zinszer *et al.*, "A scoping review of malaria forecasting: Past work and future directions," *BMJ open*, vol. 2, Oct. 2012, doi: 10.1136/bmjopen-2012-001992.

- [61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [62] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1\_53.
- [63] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959, doi: 10.1113/jphysiol.1959.sp006308.
- [64] H. Sharma, N. Zerbe, I. Piwonski, O. Hellwich, and P. Hufnagl, "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology," *Computerized Medical Imaging and Graphics*, vol. 61, pp. 2–13, Jun. 2017, doi: 10.1016/j.compmedimag.2017.06.001.
- [65] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [66] R. Basili and D. Croce, "Neural Network Applications: CNNs."
- [67] "Improving optimization of convolutional neural networks through parameter fine-tuning | SpringerLink." <https://link.springer.com/article/10.1007/s00521-017-3285-0> (accessed Mar. 27, 2020).
- [68] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning Local Feature Descriptors Using Convex Optimisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1573–1585, Aug. 2014, doi: 10.1109/TPAMI.2014.2301163.

## 8 Appendix

### 8.1 SVM turning color images to grey scale

```
def show_sample_cells(cell_type, num_cells=8, greyscale=False, resize_image=False):
    fig, axes = plt.subplots(1, num_cells, figsize=(18,2))
    fig.suptitle('{0} Cells:{1}{2}'.format(cell_type,
                                         ' - Greyscale' if greyscale else '',
                                         ' - 30x30' if resize_image else ''))
    for idx, f in enumerate(listdir(path.join(malaria_dataset_path, cell_type))[:8]):
        cell_image = io.imread(path.join(malaria_dataset_path, cell_type, f))
        if resize_image:
            cell_image = resize(cell_image, (30, 30, cell_image.shape[2]), mode='reflect')
        if greyscale:
            cell_image = color.rgb2grey(cell_image)
            axes[idx].imshow(cell_image, cmap='gray')
        else:
            axes[idx].imshow(cell_image)
        axes[idx].set_xticks([])
        axes[idx].set_yticks([])
    show_sample_cells(Infected_dir)
```

Figure 46 Turning images to grey scale

### 8.2 CNN configurations: Adam, Sigmoid and binary cross entropy

```
inp = tf.keras.layers.Input(shape=INPUT_SHAPE)

conv1 = tf.keras.layers.Conv2D(32, kernel_size=(3, 3),
                                activation='relu', padding='same')(inp)
pool1 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv1)
conv2 = tf.keras.layers.Conv2D(64, kernel_size=(3, 3),
                                activation='relu', padding='same')(pool1)
pool2 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv2)
conv3 = tf.keras.layers.Conv2D(128, kernel_size=(3, 3),
                                activation='relu', padding='same')(pool2)
pool3 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv3)

flat = tf.keras.layers.Flatten()(pool3)
hidden1 = tf.keras.layers.Dense(256, activation='relu')(flat)
drop1 = tf.keras.layers.Dropout(rate=0.4)(hidden1)
hidden2 = tf.keras.layers.Dense(256, activation='relu')(drop1)
drop2 = tf.keras.layers.Dropout(rate=0.4)(hidden2)

out = tf.keras.layers.Dense(1, activation='sigmoid')(drop2)

model = tf.keras.Model(inputs=inp, outputs=out)
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.summary()
```

Figure 47 CNN configurations

### 8.3 10 – fold cross validation

```
# 10-fold cross validation
for name, model in models:
    kfold = KFold(n_splits=10, random_state=seed)
    cv_results = cross_val_score(model, trainDataGlobal, trainLabelsGlobal, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

*Figure 48 10 - fold cross validation*

IJSER