Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Automatic Malaria Detection Using Machine Learning Approaches

A Thesis Submitted to Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University

In Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Engineering

By

Brhane Gebremedhn Gezehegn

Advisor**:** Mr. Yonas Yehualaeshet

Addis Ababa, Ethiopia

December 2020

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

(Computer Stream)

By: Brhane Gebremedhn Gezehegn

Approved by Board of Examiners

Chair of Department	Signature
Advisor	Signature
External Examiner	Signature
Internal Examiner	Signature

DECLARATION

I, the witnesses, hereby declare that this thesis is my original work performed with the supervision of Mr. Yonas Yehualaeshet has not been presented as a thesis for MSc degree program in any other university and all sources of materials used for the thesis work has been fully approved.

Name: Brhane Gebremedhn Gezehegn

Signature: _____

Place: Addis Ababa University, Addis Ababa, Ethiopia

Date of submission: December 2020

This thesis has been submitted for examination with my approval as a university advisor.

Advisor's Name: Mr. Yonas Yehualaeshet

Signature: _____

Place: Addis Ababa University, Addis Ababa, Ethiopia

Date of submission: December 2020

ACKNOWLEDGMENT

This thesis would not have been possible without the upkeep of many people and some organizations. First of all, I would like to express my deepest gratitude to my advisor **Mr. Yonas Yehualaeshet** for his supervision, persistent advice, motivation, patience, time, and continued guidance right from the moment of becoming my advisor, topic selection, statement of the problem formulation to the completion of the work. I also want to thank him for his comprehensive lectures on Machine Learning which helped me solve the problem of the thesis.

I acknowledged the National malaria Institute and the Foundation for the US National Institutes of Health for their critical role in the creation of the free publicly available databases used in this thesis work. I would like to express my honest appreciation to all electrical and computer engineering staff who guided and extended their valued knowledge and guidance through the different phases of the seminars which assisted me in my research work. Last but not least, I would like to honestly thank all of my friends, colleagues, classmates for their assistance, help, and inspiration in this thesis work.

Above all, praise and thanks to God, the almighty, for His blessings throughout my research work.

ABSTRACT

Malaria parasites are one of the most common infectious diseases, causing widespread suffering and deaths in various parts of the world. To ease the process of detecting whether a person is infected or not, various studies have been conducted for a long time. However, most of the proposed techniques that have been used by different researchers for automating the detection process have limited detection accuracy. Besides, those proposed techniques are only focused on specific types of features rather than finding better feature types for automating the detection process. Thus, leading to models not generalizing very well. Furthermore, it is an active area of research demanding the development of automatic, efficient, reliable, and accurate detection systems. Due to this reason, this thesis aims to assess various features and classification techniques and selects the best possible method that yields the highest detection performance.

The approaches followed in this study to determine whether a patient's blood sample is infected with malaria or not are dataset collection, image preprocessing, feature extraction and classification. To conduct the experiments a total of 27,558 segmented cell images extracted from thin blood smear slide images were used from the US National Institute of Health (NIH) recorded data. These images are enhanced using various preprocessing techniques. Once the preprocessing phase is done, three types of features namely color histogram features, haralick texture features and the combination of the two features are extracted. Finally, different supervised machine learning techniques with different model parameters such as support vector machine, decision tree, K nearest neighbor, multi-layer perceptron, random forest, and naive Bayes were used for the classification purpose.

The proposed techniques were evaluated using a confusion matrix, and classification performance report to assess which has a higher classification potential. The random forest algorithm has achieved an average accuracy of 95%, average precision of 95.0%, 95.0% of average recall and an average F1 value of 95.0% over a test dataset of previously unseen 8266 images. From the analysis of the experimental results, the random forest algorithm gives better results than the other supervising machine learning classifiers. Thus, due to the fact that random forest aggregates more than two decision trees to avoid overfitting as well as error due to bias making it more accurate from the analyzed algorithms, and thereby showing the feasibility of its usage in real-time applications for determining whether a cell is infected with the malaria parasite.

Keywords: Malaria, Blood smear, Image processing, Supervising Machine learning, Feature extraction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Abbreviations	Definition
NN	Neural Network
ANN	Artificial Neural Network
RF	Random Forest
SVM	Support Vector Machine
KNN	K nearest Neighbor
DT	Decision Tree
NB	Naïve Bayes
RGB	Red, Green and Blue Color Mode
MLP	Multi-layer perceptron
ML	Machine Learning
US	United States
NIH	National Institute of Health
WHO	World Health Organization
CDC	Center of Disease Control
RBC	Red Blood Cell
HSV	Hue Saturation Value
CAD	Computer Aided Diagnosis
GLCM	Gray Level Co-Occurrence Matrix
PCA	Principal Component Analysis
SOMs	Self-Organizing Maps
RBF	Radial Basis Function
LDA	Linear Discrimination Analysis
SURF	Speed up Robust Features
TP	True Positive
TN	True Negative
FP	False Positive or False alarm
FN	False Negative
PNG	Portable Network Graphics

CHAPTER ONE

INTRODUCTION

1.1. Background

Malaria is a serious blood ailment produced as a result of parasites spread to humans through the mouthful of the anopheles mosquito, called malaria vectors. When infected mosquito bites a human and spreads the parasites, those parasites reproduce in the host's liver before infecting and rescinding red blood cells. It is a global health problem causing over a million human infections annually, according to the World Health Organization (WHO) Statistics. For instance, as in [1] stated, there were around 219 million cases of malaria in 87 countries affected by epidemic parasite. According the WHO report, in Africa most top ten of the death cases were caused by malaria. This is because of the favorable environmental conditions of some African regions which were helpful in nurturing mosquitoes. Moreover, the social and poor economic conditions of African people making it more difficult to receive suitable treatment and equipment in countering malaria producing a high rate of death from malaria disease[2].

According to [3] Ethiopia as one of the sub-Saharan countries in Africa is the victim of the malaria epidemic with far-reaching negative impacts. Malaria disease is a serious public health concern in Ethiopia, as 75% of the land and 60% of the population are exposed to the disease. The malaria disease has been consistently reported as one of the topmost leading causes of outpatient visits, admissions, and deaths among all age groups in Ethiopia. Hence, knowing the trends of malaria prevalence is essential to design appropriate interventions against the disease[3]. It becomes an individual's health problem affecting the economic development of the country as it prevents the infected people from participating in their day to day activities. The coverage of malaria varies markedly by location and season as its occurrence in most parts of the country is unstable mainly due to the country's topographical and climatic features several peoples are at risk through years being susceptible to malaria[4].

In most parts of Ethiopia, malaria is seasonal with a periodic transmission that lends to the outbreak of an epidemic. The transmission strength varies greatly due to the large diversity in altitude, rainfall, and population migration. Hence, the environment plays a great vital role for the spread of malaria disease as the vector requires favorable habitats in reproduction that includes the availability of ample

surface water and temperature. There are different methods to diagnose malaria, of which manual microscopy is considered to be the gold standard [5].This manual method of diagnosis is time-consuming and may lead to inconsistency. The proposed approach of an automated method for parasite detection based on a machine learning approach once digitized will reduce the time taken for screening the disease, and also improves consistency in diagnosis when compared with the manual [6].

Malaria parasite detection in this work is performed based on color histogram features, haralick texture features and the combination of haralick texture features with the color histogram features. A malaria detection system must be equipped with functions to perform are image Acquired, preprocessing, features extraction and classification tasks. In order to perform analysis on the blood smear, the system must be capable of differentiating between malarial infected artifacts and non-infected blood smear components. Different supervised machine learning techniques with different model parameters such as support vector machine, decision tree, K- nearest neighbor, multi-layer perceptron, random forest and naive Bayes were used for classification purposes.

1.2. Motivations for the study

Malaria is a mortal disease for humans, through the years. The major method used for detecting malaria parasites in the blood is to prepare a blood smear, stain it and look for the parasite under the conventional microscope. Different rapid diagnostic test kits have been developed but they still have their limitations[7].Manual assessment of blood smear films is time-consuming, error-prone. Furthermore, the correctness of the final analysis ultimately depends on the time spent studying each slide and the skills and experience of the technician.

A supervised machine learning technique creates a new uprising technique for medical image analysis. Machine learning methods in malaria detection systems gives better automate in medical image analysis. The implementation of automated malaria detection using images extracted from blood smear films would result in accurate measurement in malaria diagnosis and improves the delay in treating patients and also reduces the time the physicians spent on the diagnosis. Therefore, the motivation behind this thesis work is to implement an automatic malaria detection system that enables accurate diagnosis in detecting the malaria parasite using supervising machine learning approaches.

1.3. Problem Statement

Medical diagnosis of malaria is the procedure of detecting parasite disease by critical analysis of its symptoms and is often supported by a series of laboratory tests of varying complexity. Accurate medical diagnosis is important to afford the most effective treatment option for patient.

A Number of studies have aimed to assess the accuracy of the manual microscopic diagnosis of malaria. It was shown that the manual method itself may not be a reliable screening method when it

is performed by clinical examiners who have limited training especially in the rural areas where Malaria is prevalent. This means that the result of the manual malaria screening technique is subjective among different experts. Consequently, in this thesis, a system is developed that can automatically detect malaria parasites whereby the decision is made by a computer using image processing algorithms applied to digital images acquired from a digital microscope. There have been many advances since the development of computer-based malaria detection. However, there is still a growing demand for developing more efficient and accurate techniques. Therefore, this study aims to assess various feature extraction and classification techniques and select the best one that will improve the entire detection procedure.

1.3.1. Research Questions

To this end, this thesis was attempts to answer the following research questions:

- Which supervised machine learning model provides the highest performance?
- Which feature extraction technique provides the highest performance?
- Can the combination of feature sets of different feature extraction techniques outperform an individual feature set of single feature extraction techniques?

1.4. Objectives of the Study

1.4.1. General Objectives

The general objective of this research is to build a machine learning model that can automatically classify whether a cell is healthy or infected and assess the performance of the developed model.

1.4.2. Specific Objectives

In light of this general theme, the specific objectives of the thesis are the followings:

- To select and apply the proper preprocessing method
- To extract the features that feed as an input to the model
- To test the effect of feature extraction on detection system performance
- To develop a feature set combiner algorithm
- To compare the results and performances of the machine learning techniques used.

1.5. Significance of the Study

The majority of malaria finding techniques usually require human mediation to support in the interpretation of their results. An attempt conventional microscopy which is the gold standard method of malaria diagnosis has yielded little success as the degree of accuracy for parasite detection reported

remains low. In this research work, the development of malaria disease detection using machine learning techniques is carried out.

The importance of this research are:

- It provides a research output for malaria detection researchers in the development of malaria detection systems from microscopic images.
- The research plays a great role in understanding the steps and challenges of malaria detection from microscopic images through color histogram features, haralick texture features with supervising machine learning approaches.
- The study helps to initiates researchers to do malaria detections with different approaches such as SVM, KNN, Naïve Bayes, MLP, Decision trees and Random forest algorithms.

1.6. Thesis Contribution

Automating malaria disease detection has been an issue for a long time. Different researchers have attempted to automate it using various techniques. However, the aim was to develop a model using a particular supervising machine learning and a particular feature extraction technique that determines whether a person is affected by malaria or not. Besides, there are other features and models that could yield the highest performance. hence, after different literature reviews were conducted and to the best of my knowledge, this study select best features using different features extraction techniques to determine the best descriptor that can provide the highest accuracy for malaria detection using the state of the art supervised machine learning techniques. This is a really great contribution to the process of malaria disease detection as miss-detection will have a severe impact on the patient. The other contribution of this study is determining the best possible supervised machine learning classifier following the best malaria detection descriptor identification. All in all, the contribution of this study is the performance improvement in comparison with the other research work.

1.7. Scope and Limitation of the study

While conducting the study, it has been observed that some aspect of the study was not attempted. These parts of the study are not part of the scope of this thesis. The limitations include the following.

- The scope of the study is only limited to the determination of whether a blood smear is infected or not infected by malaria.
- The study focused only on the color histogram features, haralick texture features and combination of color histogram features with haralick texture features.
- This study didn't include the determination of infecting malaria species and the stage of the parasite.

- ❖ The scope of the study conventional ML approaches are used not deep learning approaches.
- ❖ The other limitation of the study is the unavailability of a local dataset. Since it was difficult to obtain a dataset locally a public dataset is used for testing purposes.

1.8. Research Methodology

The research methodology mentioned below would be used to select and implement appropriate methods and techniques: literature review, image processing, features extraction and disease classification techniques.

Literature Review: Before starting the actual work, a detailed study was conducted in the literature written on this area to have a clear picture of the work at hand. Research written on the malaria detection system will be reviewed to get an understanding of the various techniques and methods of an automatic malaria detection system using machine learning approaches. A literature survey was conducted on the area of image processing and machine learning for every stage of the system design. Available books, journals, case studies, previous research works & guidelines were surveyed in order to have a clear understanding of the subject matter.

Data collection and Preparation: Images are downloaded from the US National Institute of Health (NIH) website. Labeled blood smear image needed for experiments are prepared and collected.

Design and Implementation: this phase deals with designing and conducting experiments for classification. This phase encompasses the following steps. The first step is preprocessing of an image. The second step is the extraction of features of image that are used as an input to the supervising machine learning algorithm classifier. Final phase of the research work was targeted at classifying the disease into infected and non-infected. The architecture of the system determining the training approach of each component of the system, determining system parameters and their value. The implementation part is just how to implement the system from different perspectives like development language, development resource anaconda 2019 software has been used for implementing the proposed system.

Experimentations and Discussion: Discussed on the Results, give conclusions and the future Points. Generally, Methodology Followed to conduct the Research is as figure 1.1.

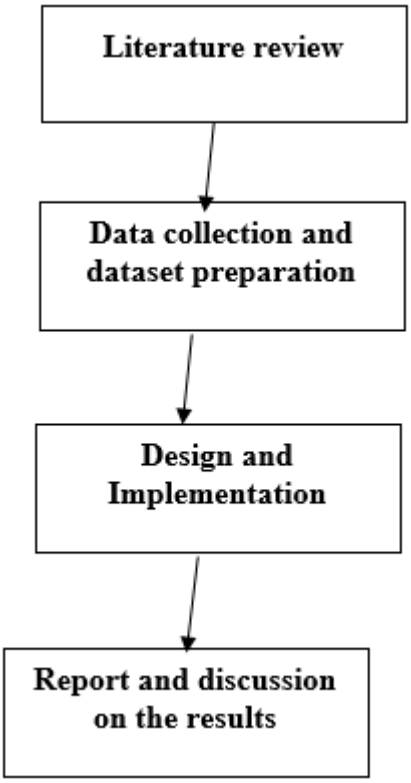


Figure 1. 1 Methodology Followed to conduct the Research

1.9. Organizations of the Thesis

This thesis is organized into five different chapters. The first chapter presents a preliminary introduction to this study. It offers the general structure included in this study. It provides enough background information to help the reader understand the reason behind the study and what the researcher plans to accomplish by carrying out the research. The second chapter studies an explanation of malaria which will be selected for demonstrating the proposed methodology, machine learning techniques that are used in this research, and present reviews of previous work related to the study topic with specific reference to the research objectives. It presents summaries from books, journals, and collected works that are helpful in accomplishing this work. Chapter three gives a detailed explanation about the proposed methodology that are used in this research, the hyper-parameters range for the machine learning algorithms, the features that are extracted and used in this thesis. Chapter four presents the research results and a detailed analysis obtained through the methodology presented in chapter three. The last chapter draws conclusions from the study, gives recommendations for users of the research, and provides the future work for this study.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1. Introduction

In this chapter, the theoretical related background of the system is discussed and a detailed literature the review is performed on different malaria detection approaches using machine learning techniques. A review of existing articles written by other scholars or authors which are relevant for this study would also be summarized focusing on automatically detecting malaria disease. Automatically detecting malaria using the machine learning approach problem undergoes the following sequence of steps. Image acquired, Image pre-processing, Feature extraction and Image classification. Image acquisition involves the capturing of a patient's blood smear image using an image-capturing device. Image preprocessing makes the image more suitable for subsequent processing stages. The next step is the extraction of suitable features from an image by the use of appropriate image processing and feature extraction techniques. Based on the extracted features, the classification of individual objects present in an image is undertaken using a classifier. The last step in the malaria detection system using the machine learning approaches process is image understanding or making sense of a blood smear infected or not infected by the parasite.

2.2. Malaria Diagnosis

Malaria diagnosis is in general the process of testing the existence of malaria parasites inside a patient's blood. There are two ways of diagnosing malaria in the form of automatic and manual approaches. The manual approach includes signs and symptoms[8] used in the microscopic examination. Microscopic examination is a method of diagnosing malaria by examining a drop of the patient's blood smear under the microscope, spread out as a blood smear on a microscope slide. Prior to the examination, the specimen is stained to give the parasites a distinctive appearance. The second approach is an automated version of diagnosing malaria. This on the other hand is a computerized approach that enables to detect malaria parasites. In contrast to manual-based malaria diagnosis approaches, the automated diagnosis is usually fast and accurate.

2.3. Computer Aided Diagnosis system

A computer-aided diagnosis system is a clinical decision support system, which assists doctors in the interpretation of medical images. Computer-based diagnosis used as a tool to provide additional information to clinicians, who will make the decision as to the diagnosis of a patient[9]. Computer-based diagnosis is becoming one of the major research areas in medical imaging and has been the

inspiration for significant advances in many areas including image processing, machine learning and clinical systems integration with the blood smear. Computer-aided diagnosis systems are getting more frequent in malaria detection systems using machine learning approaches. Based on the computerized diagnosis of malaria is a microscopy diagnosis technique used as an aid or a complete automated diagnosis technique, which replaces the manual microscopy examination. Its ability to replace an expert depends on the accuracy of its diagnosis performance[6].

A computerized diagnosis system can be used in various areas such as research in clinical diagnosis, evaluation of the treatments, blood screening or in any place where Plasmodium should be observed, counted, or linked to other clinical data. The requirements for computerized microscopy diagnosis would be similar to those for manual diagnosis. It could reduce the training needs for an examiner significantly, however, it would require a computer and imaging equipment, which can make it less accessible in rural areas. The advantage of this is making it accessible than an expert is one of the advantages of the system. The imaging and analysis of a specimen can be performed in constant time with computers. More importantly, the computerized diagnosis can provide more consistent and objective results compared to manual microscopy. However, the study in this thesis is only concerned with the problem from the computer vision point of view, which is seen as the most essential part. A basic computer vision system to perform automated diagnosis malaria using machine learning approach has to provide solutions for the sub-problems listed follow.

2.3.1. Blood Smear Image Acquisition

Acquisitions of data from the various imaging modalities for input to any system are an essential task. Image acquisition is the initial step in any image processing work, in which an original input image data is acquired from the initial source (where an image can be found). Image acquisition can be applied in different methods. Most of the time, this step is only considered as capturing an image from an actual environmental scene but it can also be browsing an already existing image file from any electronic source with any method of acquisition. The image that is acquired is unprocessed when seen from the corner of the intended image processing application. Unprocessed image is acquired from different sources according to the application or type of process, for instance, the image can be acquired using a digital camera, microscopic camera, cell phone camera or webcam for an application mostly related with capturing an environmental scene, like malaria disease detection and so many others. Possibly images can be found from medical imaging equipment like microscopic images.

2.3.2. Image Preprocessing

Image preprocessing is an important step used mainly to reduce the noise artifacts in the image and misrepresentations of the image [10]. Image processing is the way of manipulating images in

numerous methods in order to get easily visualized and detected images[10][12][13].The aim of image processing in this proposed system architecture is to improve the quality of images taken from Blood smear image files that helps in finding suspicious objects. It also reduces false positives from the input data. Besides, it makes the input data suitable for machine learning algorithms to train over the data. Generally, image preprocessing is to reduce or eliminate noise from the acquired image, resize the image and to enhance the image contrast for visual evaluation. Image enhancement is a vital procedure to improve the visual appearance of the image. The HSV color represents every color in three components namely Hue (H), Saturation (S), Value (V). It strongly represents colors in a way that is very similar to how the human eye senses color. The HSV is a very popular color space because it separates the pure color aspects from the brightness[14].

Histogram equalization is a way in image processing for contrast adjustment using the image's histogram .This method increases the contrast of the images, especially when the usable data of the image is represented by close contrast values [16]. Through this adjustment, the image intensities can be better distributed on the histogram. Since the input data is highly affected by intensity variability and uneven illumination, histogram equalization for better views and better detail in microscopic images. It is used for intensity transformations, that changes the given image distribution to a uniform distribution component. The key advantage of using this technique is that it is a fairly straightforward technique.

Median filtering is a nonlinear technique used to remove noise from images pixel, over the entire image. The median filter calculated by using sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle (median) pixel value. The Median Filter is a non-linear filtering method, frequently used to remove noise from an image. Noise reduction is a typical pre-processing part to increase the results of later processing. Median filter is used for removing noise while preserving useful features in an image. Median filtering run through the signal entry by entry, replacing each entry with the median of the neighboring entries[15].Before replacing the values with the median value the pixel values are sorted in ascending order. The advantage of using a median filter is to remove the noise without disturbing the edges. Median filtering for this research is proved to be important in removing the noise that happens from the images.

The median calculation [16] considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. If the neighborhood under consideration contains an even number of pixels, the average of the two middle

pixel values is used. The median filter preserves brightness differences resulting in minimal blurring of regional Boundaries. Preserves the positions of boundaries in an image, making this method useful for visual examination and measurement.

2.3.3. Feature Extraction

Feature extraction is the process of extracting certain characteristic features and generating a set of meaningful descriptors from an image. Feature extraction is a form of dimensionality reduction and efficiently represents the major attributes that are useful for the effective classification of each class. Transforming the input data blood smear image into a set of features is called feature extraction. The purpose of the feature extraction stage is to extract various features from a given blood smear image which best characterizes a given blood smear cell. Features are the information or list of numbers that are extracted from an image. These are real-valued numbers (integers, float or binary). There are a wider range of feature extraction algorithms in Computer Vision. When deciding about the features that could quantify malaria diseases, could possibly think of Color, Texture and Shape as the primary ones. This is an obvious choice to quantify and represent the blood smear image. The feature extraction is an important part of classifier because it affects working of classifier.

The aim of the feature extraction process is to identify and extract relevant information from the image. Feature extraction means method of capturing visual content of image for indexing and retrieval. Texture based features such as standard deviation, momentum and kurtosis of RBC are calculated. Gray Level Co-Occurrence Matrix (GLCM)[17] has proved to be a popular statistical method of extracting textural features from images. According to co-occurrence matrix, Haralick textures features defines fourteen textural features measured from the probability matrix to extract the characteristics of texture statistics of blood smear images. The Haralick texture features are functions of the normalized GLCM, where different aspects of the gray level distribution. Each texture feature is a function of the elements of the GLCM, and represents a specific relation between neighboring pixels. Haralick Texture features is used represent the healthy blood smear and infected blood smear have different textures. The Haralick texture feature to distinguish between the textures of healthy blood smear and infected blood smear. It is based on the adjacency matrix which stores the position of (x, y). Texture features are calculated based on the frequency of the pixel x occupying the position next to pixel y.

Color histograms are computed for each image so as to identify relative proportions of pixels within certain values[40]. Color histogram features emphasis on the proportion of the number of different types of colors, regardless of the spatial location of the colors. Color Histogram is one of the widely method for the color feature extraction. Color Histogram is a technique the color content of

the image, built by counting the numeral of pixel of each color. Color histogram features which is efficient, quick and enough robust. In the interest of this used some features of color histograms, and classified the images using these features. The advantage of this approach is the comparison of histogram features is much faster and more efficient than of other color features commonly used methods. A color histogram features represents the distribution of colors in an image, through a set of bins, where each color histogram bin corresponds to a color in the quantized color space.

In this work, the task is to distinguish whether or not a blood smear is infected by malaria or not. Thus, the features must offer information which will be used to carry out such a classification task. When extracting features, it is advantageous to apply expertise, a priori knowledge to a classification problem.

2.3.4. Classification

The classification phase of the diagnostic system is the one in charge of creating the inferences about the extracted information in the previous phases in order to be able to yield a diagnostic result of the input image. The data used to build the final model usually comes from datasets. In particular, three datasets are commonly used in different stages of the creation of the model. The training step consists of developing a classification model to be used based on the samples of the training set. Each sample of features extracted from a given image and its corresponding class value, which are applied as input data to the classifier for the learning process. The fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyper parameters. Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset. According to[18] The testing step consists of measuring the accuracy of the model learned by the training step over the test set.

Generally, the classification has two phases, a learning phase, and the evaluation phase. Through the learning phase, the classifier trains its model on a given dataset, and in the evaluation phase, it tests the classifier performance. Performance is evaluated based on various measures such as accuracy, f-score, precision, and recall.

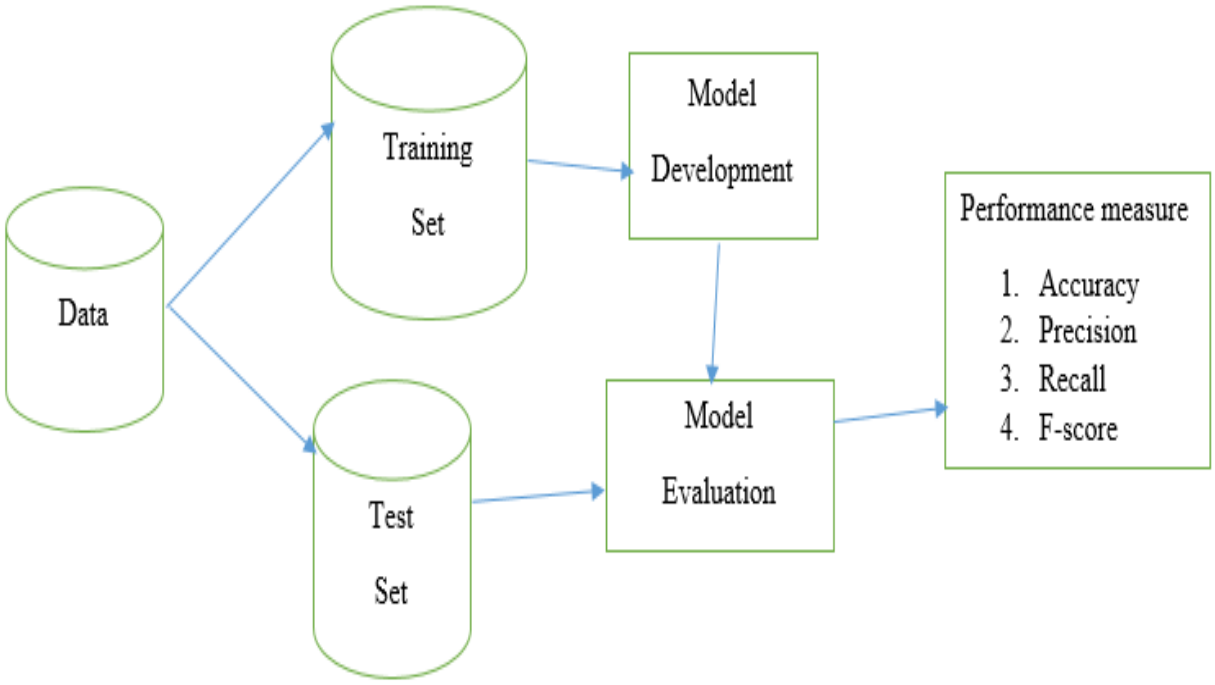


Figure 2. 1 Performance evolution model

2.4. Machine Learning Techniques

Machine learning deals with developing algorithms and techniques that allow machines to automatically learn and make accurate predictions based on past observations. Machine learning is to extract information from data automatically, by using computational and statistical ways. According to[19] machine learning is a subfield of computer science that explores the study and construction of algorithms that can learn from and make predictions based on data. Machine learning allows computers to find unseen insights without being explicitly programmed where and what to look for.

Machine learning algorithms can be classified into three groups. Supervised machine learning, unsupervised machine learning and semi-supervised machine learning. Supervised machine learning algorithms are trained using a set inputs along with corresponding accurate outputs, and the task of the algorithm is then to generate a general rule that maps inputs to outputs. Supervised machine learning algorithms include decision tree, Support Vector Machines, naïve Bayes, random forest algorithm, and multi-layer perceptron and K-Nearest Neighbors algorithms etc.

In unsupervised machine learning, the algorithm is only given input data without any corresponding output data and its task is then to explore the data and find structure within. Examples of unsupervised learning algorithms include K-means - clustering, Self-Organizing Maps (SOMs) and Principal Component Analysis (PCA). Semi-supervised machine learning is combination of supervised and

unsupervised machine learning methods. In semi-supervised machine learning, an algorithm learns from a dataset that includes both labeled and unlabeled data. Semi-supervised machine learning algorithm falls between unsupervised learning algorithm (without any labeled training data) and supervised learning algorithm (with completely labeled training data. Some of the Examples semi-supervised machine learning self-training, co-training and graph based methods.

The following Block diagram in figure 2.2 explains the working principle of Machine Learning algorithm

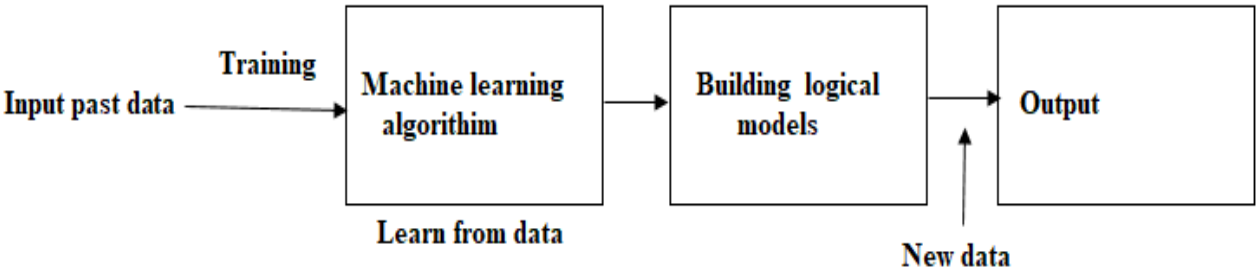


Figure 2. 2 working machine learning algorithm

This study will concentrate on supervised learning, since it will predict the probability of a certain class, and the fact that a labeled dataset is available.

2.4.1. Support Vector Machines

Support vector machine is a supervised learning approach. SVM plots the training data into another space higher than the original space and divides the instances belonging to different categories by separating these instances non-linearly and linearly. Support vector machine tries to keep the separation boundary between two different categories (classes) as wide as possible. The perpendicular bisector of the shortest line connecting the two classes is called a hyperplane. The training instances closest to the hyperplane are called support vector machines. The support vectors are very important because they determine the hyperplane. After drawing the hyperplane, the test instances are mapped into the same training space. A class value is determined for each test instance by SVM model[9].The SVM estimates a function for classifying data into two classes. SVM classification finds the hyper-plane where the margin between the support vectors is maximized. If all classifications contain two-class dependent variables with two predictors, then the points of each class could be easily divided by a straight line. Support vector machines are used as an algorithm for the classifications of both linear and nonlinear data. The SVM algorithm creates a line or a hyperplane which separates the data into classes.

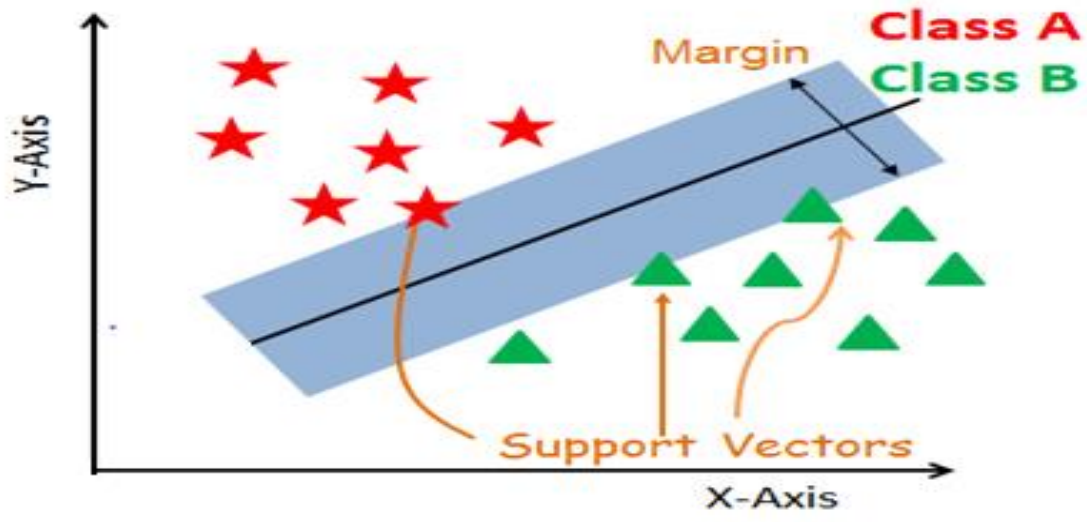


Figure 2. 3 SVM for linearly separable[20]

In the figure, the green triangle and Red Cross are data points belonging to two different classes. In this study, there are two classes which are infected blood smear and Non-infected blood smear. For each data point (x, y), x is the condition of the patient, y is either 1or-1 denoting the class to which point x belongs. The classes can be fully separated by the optimal hyperplane. The separation boundary hyperplane that leaves the maximum margin from the classes (infected and Non infected). The margin is the distance between the hyperplane and the closest data point (called support vectors) to the hyperplane. Margin of the classifier is the maximum width of the band that can be drawn separating the support vectors of the two classes and maximum-margin hyperplane decision surface that separates the two classes. To avoid misclassification between the two classes, SVM tries to maximize the margin. The support vector machine classification is done based on the hyperplane function. Those hyperplane functions are Support vectors, Hyperplane and Margin.

SVM Kernels: SVM kernels is function of the SVM algorithm is implemented in practice using a kernel. according[20] kernel transforms an input data space into the required form using kernel function. SVM uses a technique called the kernel trick. Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space. In other words, you can say that it converts non-separable problems to separable problems by adding more dimension to it. It is most useful in non-linear separation problems. Kernel trick helps to build a more accurate classifier. There are different kernel tricks, some of them are Linear Kernel, Polynomial Kernel and RBF were used in this study.

2.4.2. K-Nearest Neighbors

K-nearest neighbors (KNN) algorithm is a kind of supervised machine learning algorithm. KNN is a non-parametric learning algorithm where the classification is achieved by identifying the nearest neighbors to a query examples and then make use of those neighbors for determination of the class of the query. In K- nearest neighbors the classification to which class the given point is belongs is based on the calculation of the minimum distance between the given point and other point. It is frequently used as the prior picks for a classification study. The nearest neighbor algorithm is based on the principle that the instances within a dataset will generally exist in close proximity to other instances with similar properties[21] .This method only determines the category of the sample is subdivided according to the category of the nearest one or several samples. Prediction in the KNN algorithm is in the way that for any new sample it looks for K most similar samples based on some distant metrics[22] after that, it assigns a class label to the new sample by a majority voting as it is shown in figure 2.4. The triangle class label as shown in the figure 2.4 has been assigned to the new data sample according to the majority nearest neighbors. There are different mechanisms to compare the similarity between the data samples some of them are the following:

2.4.2.1.Euclidean distance

Euclidean distance is can be calculated using the square root of the sum of the squared differences between two points x, y and it is calculated as follows:

$$D(x, y) = \sqrt{(x1 - y1)^2 + (x2 - y2)^2 \dots\dots + (xn - yn)^2} \quad (2.1)$$

$$= \sqrt{\sum_{i=1}^{n} (Xi - Yi)^2}$$

2.4.2.2. Hamming Distance

Hamming Distance is the number of bits where two binary vectors differ and it is calculated as follows:

$$D(x, y) = \sum_{i=1}^{n} |(Xi - Yi)| \quad (2.2)$$

2.4.2.3. Manhattan Distance

Manhattan Distance is the sum of the absolute difference between two points and it is calculated as follows:

$$D(x, y) = \sum_{i=1}^{n} |(Xi - Yi)| \quad (2.3)$$

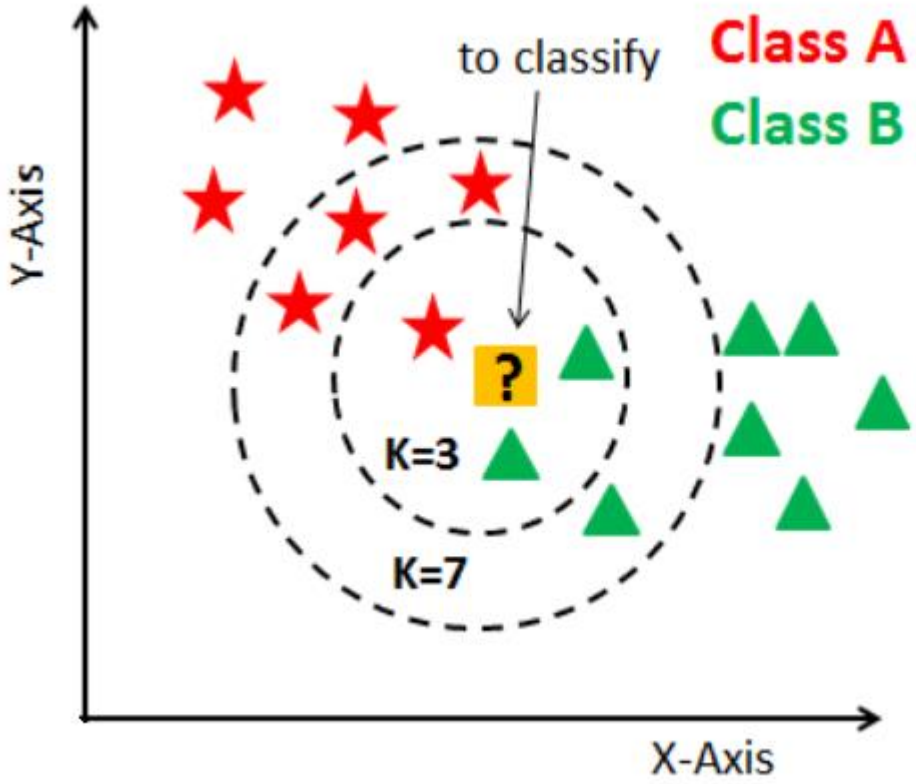


Figure 2. 4 k-nearest neighbor method[22]

KNN models help in predicting the goal class of the incoming test object. The major premise behind KNN is that similar things appear close to each other. Consequently, the class of an object by a majority vote of its neighbors and is assigned to the class most common among its k-nearest neighbors. K is a positive integer parameter passed to the KNN algorithm and can be tuned to achieve better prediction accuracy.

2.4.3. Decision Tree classifier

The decision tree is another supervised machine learning tool used in classification problems to predict the class of an instance. Decision tree is a tree-like structure where internal nodes of the decision tree test an attribute of the instance and each subtree indicates the outcome of the attribute split [23]. Leaf nodes represent the class of the instance based on the model of the decision tree. The uppermost node in a decision tree is known as the root node. It learns to partition based on the feature value. It partitions the tree with a recursive method called recursive partitioning. Decision trees offer an effective method of decision making by visibly laying out the problem so that all options can be challenge. In decision tree the main challenge is to identify of the features for the root node in each level. This procedure is known as attribute selection. In decision tree two popular attribute selection measures are information gain and gini index.

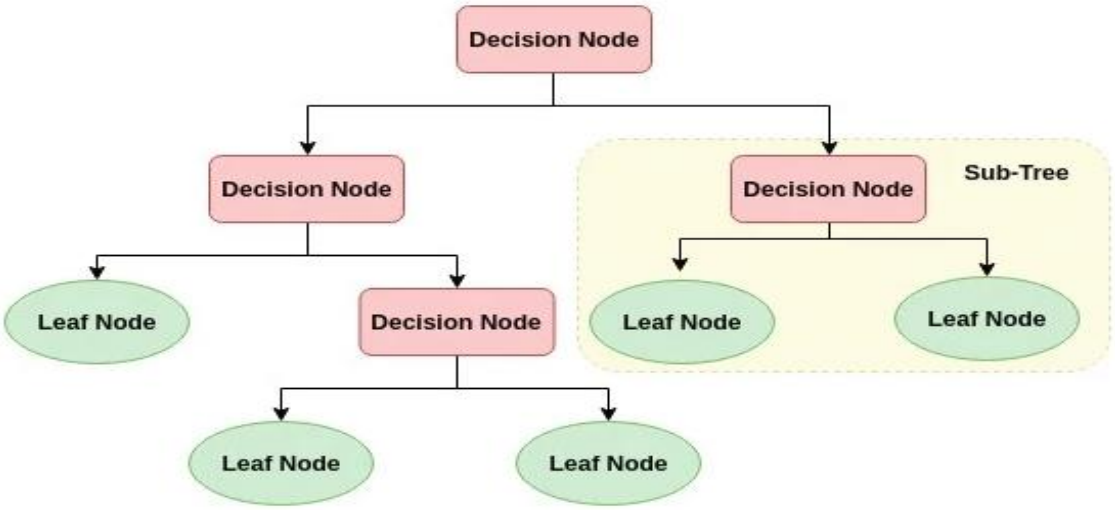


Figure 2. 5 visualization of the decision tree classifier algorithm[23]

How does the Decision Tree algorithm work?

The basic working principle behind any decision tree algorithm is shown in figure 2.6.

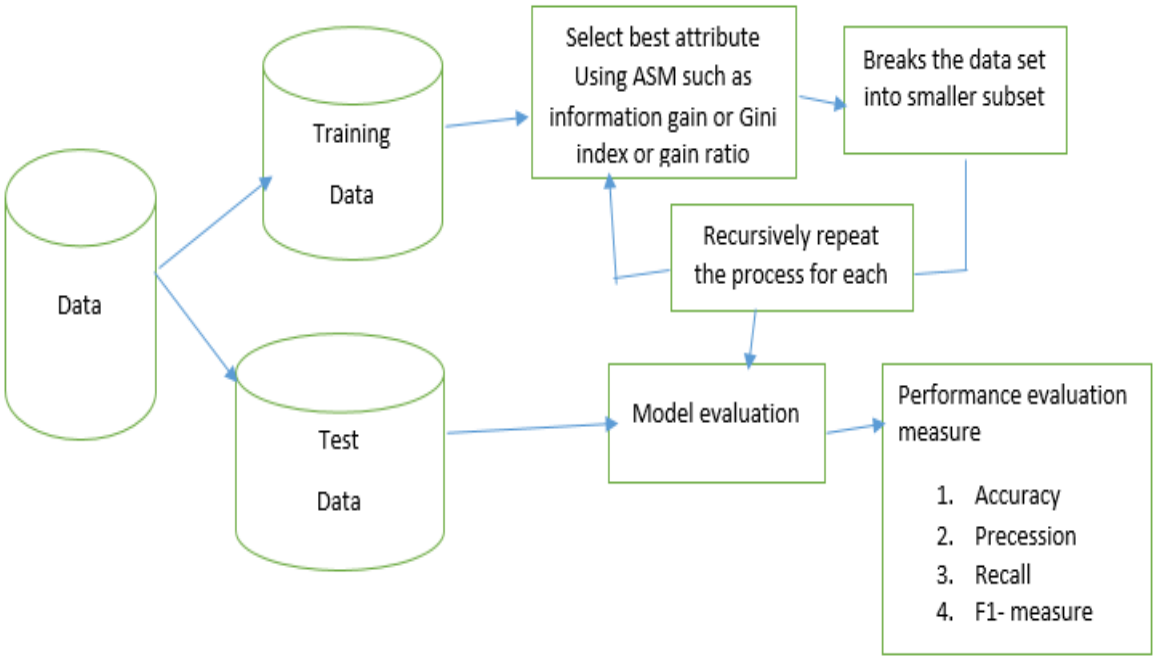


Figure 2. 6 Performance evolution model for decision tree

2.4.4. Naïve Bayes classifier

The Naïve Bayes algorithm represents a supervised machine learning method for classification .The Naïve Bayesian classifier is developed on Bayes conditional probability rule used for performing classification tasks, assuming features as statistically independent. It makes use of all the features contained in the data, and analyses them individually as though they are equally important and independent of each other .The Naïve Bayes classifier assumes that the presence or absence of a

particular feature of class is unrelated to the presence or absence of any other feature, meaning that it assumes the independency of variables given the class. In spite of its naive design and apparently over-simplified assumptions, Naïve Bayes can often outperform more sophisticated classification methods applied in many complex real world situations [24]. Naïve Bayes provides a way that we can calculate the probability of data belonging to a given class, given our prior knowledge. It predicts membership probabilities for each class such as the probability that given record data point belongs to a particular class. The class with the highest probability is considered as the most likely class[25]. Naive Bayes offer a way that we can calculate the probability of data belonging to a given class, given our prior knowledge. Bayes theorem is stated as.

$$P(\text{class}|\text{ data}) = (P(\text{data}|\text{ class}) * P(\text{class})) / P(\text{data}) \quad (2.4)$$

Where P (class| data) is the probability of class given the provided data.it is a supervised machine-learning algorithm that uses the Bayes' Theorem, which assumes that features are statistically independent Regardless of this assumption, it has proven itself to be a classifier. There are different models parameters in naïve Bayes theorem that are used for classification. Some of them are listed follow. Bernoulli Naive Bayes, Multinomial Naive Bayes and, Gaussian Naive Bayes. Naive Bayes is a classification algorithm for binary (two-class) the calculations of the probabilities for each class are simplified to make their calculations tractable[25]. It uses a Bayesian algorithm for the total probability procedure, the principle is according to the probability belongs to a category of prior probability, and the category of posterior probability.

2.4.5. Random Forest

Random forest is another supervised learning algorithm and consists of many individual decision trees. Each decision tree votes for the classification of a given data. The random forest algorithm then accepts the classification which got a maximum number of votes from an individual tree[26]. Random forest algorithms are a type of ensemble learning algorithm, which creates a number of decision trees from the training dataset to predict the outcomes of test data. When the training data (consisting of target and feature values) is given as input to the decision tree, it produces a set of rules. These rules are then used to predict the class of the new instances in test data. Random forests create a forest of decision tree models, where each tree is created over a random sample of data chosen from the training set. In contrast to decision trees that consider all features while building a model, these algorithms choose a random sample of features from the feature space to create each decision tree. Each tree votes for a specific target class and the test instance is then assigned to the class with a majority vote. Trees with high error values are weighted low and finally, a mode of the prediction class from trees with

higher weights is predicted as the final target class. After construction of such trees the same test data through each of these trees can be run and gather votes from them[27].it works in the following steps:

- Select random samples from a given dataset.
- Construct a decision tree for each sample and get a prediction result from each decision tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction

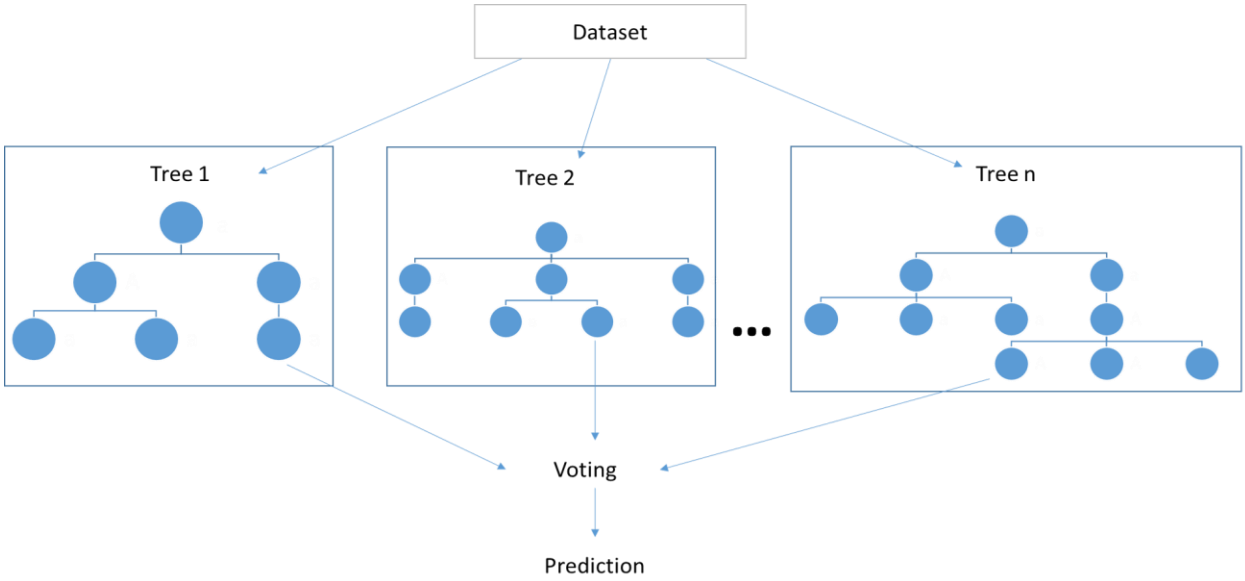


Figure 2. 7 the visualization of the random forest algorithm.

2.4.6. Multi-layer Perceptron (MLP)

Multilayer perceptron (MLP) are feed-forward neural networks that include multilayer nodes with at least one hidden layer. Each node is a neuron that has a non-linear activation function which describes its output given a set of inputs. A back-propagation learning approach is used by MLP to train the network to find the weight that plots an input to an output. A multilayer perceptron neural network can solve classification problems based on the activation function[9]. Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns function by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output. Given a set of features x=x1, x2, x3……..Xn and a target, it can learn a nonlinear function approximate classification. Figure 2.8 shows a one hidden layer MLP with output.

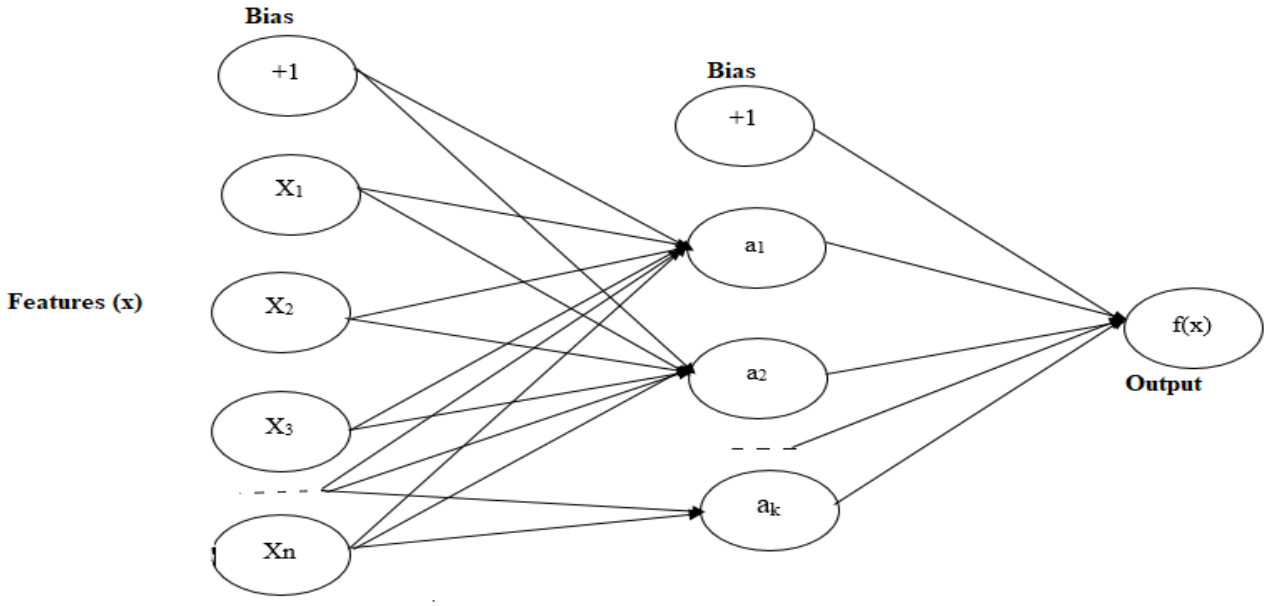


Figure 2. 8 One hidden layer MLP[28]

The left most layer, known as the input layer, consists of a set of neurons {Xi/X1, X2…Xm} representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation,$W1X1 + W2X2 \ldots WmXm$ followed by a non-linear activation function .The output layer receives the values from the last hidden layer and transforms them into output values. According [28] *v*ery briefly, the workings of the backpropagation network (the algorithmic discussion) consists in learning from a set of input-output pairs by means of the following process

1. First, an input pattern is applied as a stimulus for the first layer of neurons of the network
   - ❖ It continues propagating through all the adjacent layers until generating an output
   - ❖ the results obtained in the output neurons are compared with the desired output
   - ❖ An error value is calculated for each output neuron.
2. Next, these errors are transmitted backwards,
   - ❖ starting from the exit layer, toward all the neurons of the intermediate layer that contribute directly to the output
   - ❖ Receiving the percentage of error that corresponds to the participation of the intermediate neuron in the original output.
   - ❖ This process continues, layer by layer, until all the neurons of the network have received an error that describes their relative contribution to the total error.
3. Based on the value of the error received the weights of the connections between the neurons are readjusted.

4. Thus, the next time the same pattern occurs the output will be closer to the desired value and in this way the error decreases.

2.5. Literature Review on Related Works

Malaria detection system in computer networks has been the active research field for a long period of time. There are so many researches by different researchers which have been done for a long period of time until now. Depending on the applications, many systems have been proposed to solve or at least to reduce the problems, by making use of image processing and some automatic classification tools for detection the malaria. Most of the proposed approaches used for malaria detection follow the steps shown in figure 2.9.

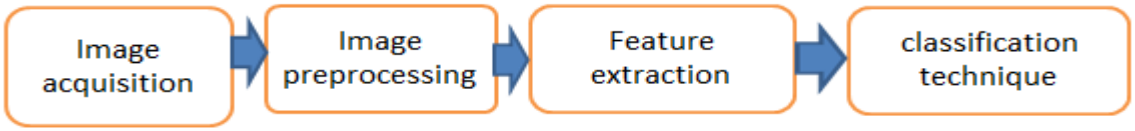


Figure 2. 9 Visualization of previously used approach

The main processes of the malaria diagnosis system image acquisition entails capturing of blood smear images using microscope fitted with a digital microscopic camera. After images are captured they are loaded to a computer where they are processed. Processing involves the following stages. Image resizing, noise reduction and feature extraction step comes features such as color and texture are extracted from the image. Finally, the classification step is performed. Different classification algorithms are used in the literature such as KNN and neural networks[29] ,support vector machines [8] and so on. Some systems in the area will be explained as follows.

A Work in[2] presented an approach that uses the artificial neural network (ANN) to detect the malaria parasite. In this study feature extraction and classification of malaria parasites are used to detect the parasite. Enhancement of image was developed before the feature extraction. Feature extraction based on histogram-based texture is used to extract feature parasite cells and a Multilayer perceptron algorithm is used to classify the features. The proposed method achieves an accuracy of 87.8%, sensitivity of 81.7% and specificity of 90.8%for detecting infected blood smear cells. As the development and improvement of this study, further studies are necessary to increase the classification with more feature extraction methods for detection of the malaria parasite. Another work in[30] proposed an automatic approach to detect the malaria parasite. Color and statistical features were extracted from blood smear image and SVM binary classifier was used for classification of normal and parasite-infected cells. The approach in this work achieves a sensitivity of 93.12%, and specificity of 93.17%. The aim of[31] is to develop a system for detect malaria parasite in blood samples stained with giemsa. The method consists of preprocessing, feature extraction based on the

histogram features of different color channels and malaria infected classification using support vector machines (SVM), nearest mean (NM), K nearest neighbors (KNN), 1-NN, and Fisher. The experimental analysis of all the classifiers with the features have been carried out on a clinical database. Based on the experimental results it concluded that K nearest neighbors provides the higher classification rate in comparison to other classifiers which provides an overall accuracy of 91%. The work in[32] used Giemsa-stained blood cell images. The infected red blood cells can be distinguished from non-infected ones since color distributions of these two kinds of cells are different; so, to differentiate non-infected and infected blood smear cells they used the color feature. The highest detecting accuracy belongs to k-NN, with 92% accuracy, and the lowest one belongs to the LDA classifier with 84% accuracy. They concluded by stating that their approach is better in detecting and classifying malaria disease based on their experimentation.

Another work on[33] is a comparison of different classification techniques using knowledge discovery to detect malaria infected red blood cells. The objective of this research paper is to present an analysis on the main machine-learning algorithms for the classification techniques used for malaria-infected red blood cells (RBCs) and determine the best techniques by comparing classification accuracy. The experimental results show that ANN is more accurate than SVM, having 94% accuracy compared with SVM's 92.3%. [34] Proposed malaria parasite detection using different machine learning classifiers. The speed up robust features (surf) to represent the image categories, then the number of features are reduced technique to acquire plasmodium infected and non-infected erythrocytes and relevant features were extracted. Six different machine learning techniques for classification are used in the experiments. The algorithms with their classification accuracy are linear SVM 85.1%, quadratic SVM 85.7%, fine Gaussian SVM 82.0%, cosine KNN 85.1%, and Boosted tree 82.6% and subspace KNN 86.3%. Among the six classifiers implemented in the classification tasks, subspace KNN has the best overall performance.

In [35] conventional image processing techniques are compared in order to detect and classify microscopic blood smear images of Malaria parasites. The Scale Invariant Features Transform (SIFT) were used for feature extraction and SVM was used for classification technique. SVM is used to classify the features which are extracted using SIFT. The overall performance measures of the experimentation are accuracy (78.89%), sensitivity (80%),and specificity (76.67%.[36] Presented a paper that uses Malaria Disease Detection Using Different Classifiers. This addresses how to detect malaria diseases using image processing by effectively analyzing various parameters of blood cell image by using GLCM as Energy and others like Skewness, Kurtosis, and Standard Deviation. From the experimental results indicate that the proposed approach is significantly supporting

the accurate detection of malaria diseases in with a little computational effort. Experimental results showed that the proposed approach has obtained a classification accuracy of 87.85 % using the SVM and the accuracy using NN is 75%.

The aim of the research in[37] is to propose a comparative study on classifying the imbalanced malaria disease data using Naive Bayesian classifiers in different environments. This presents, clinical descriptive study on 165 patients of different age group people collected at medical wards of Narasaraopet(Indian city) from 2014-17. Synthetic Minority Oversampling Technique (SMOTE) technique is used to balance the class distribution and then they performed a comparative study on the dataset using Naïve Bayesian algorithm in various platforms. Out of balance class distribution data, 70% of data was given to train the Naive Bayesian algorithm and the rest of the data was used for testing the model. Experimental results have indicated that classification of malaria disease data found the highest accuracy of 88.5% using naïve Bayes classifier.

The aim of the paper in[38] is to address the development of computer- assisted malaria parasite classification using a machine learning approach based on microscopic images of peripheral blood smears. In doing this, microscopic image acquisition from stained slides, illumination correction, and noise reduction, erythrocyte feature extraction and finally classification of infected and non-infected is performed. Features describing shape-size and texture of blood smear are extracted in respect to the parasite infected versus uninfected cells. Texture features such as mean, standard deviation, entropy, variance, smoothness, skewness, kurtosis, contrast, correlation, energy, and homogeneity from the diseased regions. Those features are then fed into the SVM and Bayesian approaches, which performs the final categorization. This proposed approach has obtained a classification accuracy of 83.5%, using SVM and 84% using Bayesian approaches.

From the detailed literature review, it was noticed that there is a room for improvement in automatic malaria detection systems by extracting a new set of features, combining them together and using additional unused machine learning approaches. Furthermore, this research tends to address the shortcoming of previous studies in detecting the parasite in the infected cells by applying the new approach. The proposed framework would be implemented on the microscopic image of malaria-infected blood smear samples parasite and with non-infected blood smear samples.

CHAPTER THREE

PROPOSED METHOD FOR MALARIA DETECTION

3.1. Introduction

In chapter 2 theoretical concepts on different types of supervised machine learning techniques, literature reviews on malaria infection detection, and classification have been discussed. In this chapter, the proposed method for the automatic malaria detection system using a machine learning approach along with from the proposed system architecture will be discussed. The methods and techniques being used while the proposed models and an overview of the image processing applied on the dataset prior to model training will be discussed in detail. The goal of this work is to develop a system for automatic malaria detection using supervising machine learning approaches from blood smear samples.

3.2. Proposed System Architecture

The proposed system block diagram as shown in figure 3.1 is starts by selecting dataset for training and testing from the US National Institute of Health (NIH) recorded dataset collection. Generally, the procedure followed in solving such a problem is as follows. First, an image is collect and pre-processed then the appropriate features are extracted. Next, an appropriate classifier is used to categorize the features into their classes. This proposed architecture tells the overall process followed to classify a particular input image in either of two classes (infected and non-infected). The design figure has two separate phases in the system. The training phase and the testing phase with a slight difference in approach. The training phase starts by importing a batch of input images, which means several images are started to be processed at a given time. In the testing phase, a single image is imported for the process. After the image is imported both phases pass through the same processes that are providing the same purpose. The preprocessing and feature extraction functionalities are the same for both phases. After feature extraction, both phases follow different paths. The training phase provides a feature vector with a label input for the model to train and the build trained model. The testing phase provides a feature vector to the model and expects a label return classifier that returns a label from the trained model previously. The processes are briefly described in the next sections.

Block diagram

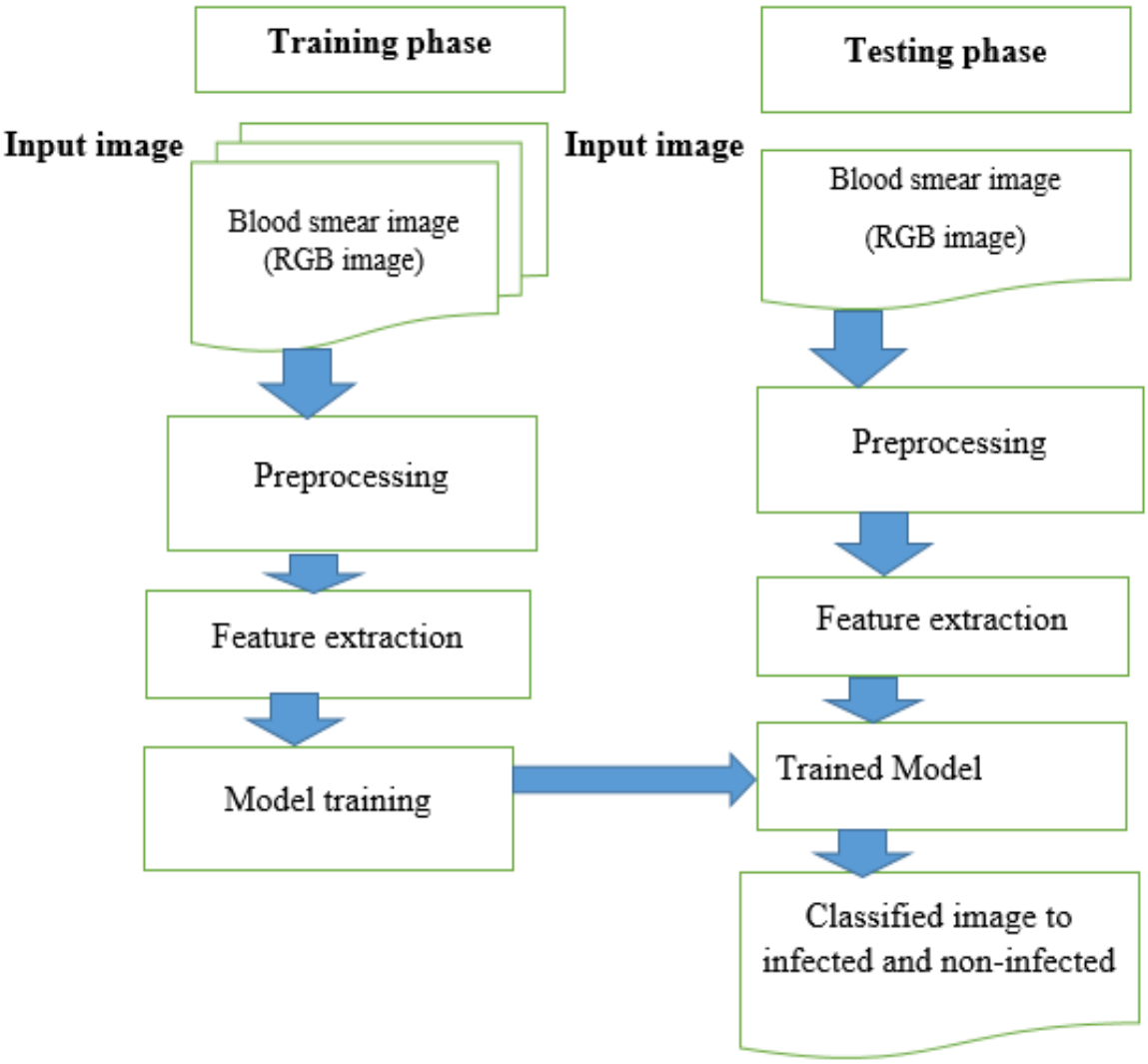


Figure 3. 1 Block diagram summarizing the proposed approach for malaria detection

3.3. Preprocessing

This is the initial phase of the proposed approach. The aim of preprocessing the datasets is to prepare it in a compatible format for machine learning and to improve the quality of the image by removing noises. Noise refers to variation in intensity or brightness in an image [36]. It might get added during the acquisition of the images, which is introduced by camera flash, change in illumination, noise background of the image. The purpose of image pre-processing is to remove noise from the image for enhancing the quality of the input image, getting a maximum accurate result, and good efficiency. A median filter is applied to reduce the different noises. Hence, the size of blood smear images in the database is resized to reduce image processing time. All images are in PNG format and the original size of the images was in different resolutions (100 × 100, 224 × 224, 227 × 227 and 299 × 299) is

reduced to 100 × 100 pixels. The image size was carefully chosen by testing different sizes not to compromise the performance of malaria disease classification of the automatic system.

Image resizing is also important to have uniform image sizes. For simplified computation, images can be resized to a light weight version of the original image by minimizing the number of pixels call this image resizing task. But this is only recommended when the resizing action does not affect the resultant image meaning or appearance of the original image. Image resizing is applied for ease of computation in some cases image files with large size can slow down the overall performance of a model. This resizing task is applied in way which cannot abuse the information that is found in the original image.

Image conversion is one part of preprocessing and can include the process of converting an image between different types of formats or color spaces for the ease of computation. For example, colorful images are converted into gray scale images or black and white images because images with less color ranges are easier to process than those with many color possibilities like RGB or other types. Histogram equalization is another sub-task under the pre-processing task, that is applied for contrast enhancement which enables to get detailed information from input image. Input image has RGB color space originally, but for additional purposes this color image can be converted to different color spaces. Color is a very important perceptual phenomenon related to human response to different wavelengths in the visible electromagnetic spectrum. The image is usually described by the distribution of three-color components R (red), G (green), B (blue). Color image is often also represented by three psychological qualities (H) hue, (S) saturation and (I) intensity or (V) value. These color features and many others can be calculated from R, G and B by either a linear or a non-linear transformation. Hue is an attribute of light that distinguishes one color from the other, for example a red color from green or yellow color. Saturation describes the amount of whiteness of a light source in a given image while Intensity or Value is a measure of the brightness of a given image. To grayscale conversion helps to work fine with many image processing algorithms grayscale images are easy to process than RGB images. Because of the number of possible pixel values in different color spaces varies, most algorithms treat input images differently. Additionally, HSV images are easy for the machine to identify the dominant color in a given Image than the RGB one. The median filter for each of the RGB components separately, this is not a good choice, because the components are correlated, and false colors may appear. Convert to HSV from RGB and then do the median filter on the hue, saturation and value, then convert back to RGB, this method is usually better (for most applications).

The Preprocessing pseudocode

```
Input: RGB image
Output: Noise removed and enhanced image
Start
    Read the RGB image
   Convert the RGB image to image in the HSV color model
   Median filter the components (HSV) of the converted image
   Combine the hue and saturation component with the filtered value
   Convert the image to RGB image
   Save the result
End
```

Figure 3. 2 infected blood smear

Figure 3.3 non-infected blood smear

In figure 3.2 of parasitized cells, one or more purple blobs can be seen. This is the malaria parasite stained with a contrasting agent. In contrast, figure 3.3 of uninfected cell images do not show staining.

3.3.1. Grayscale Conversion

All the input images are presented in an RGB format. These have to be first converted from RGB format to a grayscale format. A grayscale (or gray level) image is simply one in which the only colors are shades of gray. Grayscale conversion is the process of converting the true color image (RGB) to the grayscale image which is usually performed by matching the luminance of the color image. Converting the RGB image into a gray scale images are preferred over colored ones to simplify mathematics. It is relatively easier to deal with (in terms of mathematics) a single color channel (shades of white/black) than multiple color channels. The grayscale conversion was performed using OpenCV cvtColor module. An example image is shown in Figure 3.4 and 3.5 created using the OpenCV cvtColor module.

Figure 3. 4 grayscale infected

Figure 3. 5 grayscale non-infected

3.4. Feature Extraction

In feature extraction, the original vector space is transformed to form a new minimalistic feature vector space to distinguish between infected and non-infected red blood cells. After preprocessing the input images has been completed, the feature extraction step is carried out. It extracts information from the input image to serve as an input into the conventional supervising machine learning method. For an image, a feature can be defined as measures describing dataset properties and characteristics. This feature extraction plays a great fundamental role in classification. Features are necessary for differentiating one category from another. Features may have various categories necessary to extract (feature extraction) that can differentiate one class of objects from another. In this thesis haralick texture features and color histogram feature are selected as they have been extensively used in image processing and recognition by different researchers to extract features.

These haralick texture features are 13 in number which includes, angular second moment, contrast, and correlation, the sum of squares (variance), inverse difference moment, sum average, sum variance, sum entropy, entropy, and difference entropy, information measures of correlation, difference variance and maximal correlation coefficient. These features create a feature vector that would be served as an input for the training of a classification model with training labels. Other features used in this study also include color histogram features. The color histogram features are extracted features from the blood smear necessary for differentiating one category from infected and non-infected using the classifier model. These features create a feature vector that will be served as an input for the training of a classification model with training labels.

3.4.1. Haralick Textural Features

Haralick texture features are calculated from a Gray Level Co-occurrence Matrix, (GLCM) a matrix that counts the co-occurrence of neighboring gray levels in the image. To extract Haralick Texture features from the image were used from the Mahotas library. The function would be used by MahotasFeaturesHaralick (). Before doing that, the color image converts into a grayscale image as the haralick feature descriptor expects images to be grayscale.

Texture features using Haralick's from the co-occurrence matrix have been extracted for the images. These haralick texture features are 13 in number which includes, angular second moment, contrast, correlation, the sum of squares (variance), inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, information measures of correlation and maximal correlation coefficient.

A co-occurrence matrix or co-occurrence distribution is a matrix that is defined over an image to be the distribution of co-occurring pixel values (grayscale values, or colors) at a given offset:

- The offset, $(\Delta x, \Delta y)$, is a position operator that can be applied to any pixel in the image (ignoring edge effects): for instance, $(1, 2)$ could indicate "one down, two right".
- An image with p different pixel values will produce a $p\ x\ p$ co-occurrence matrix, for the given offset.
- The $(i, j)^{th}$ value of the co-occurrence matrix gives the number of times in the image that the $i^{th}$and j^{th} pixel values occur in the relation given by the offset.

$$C_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \begin{Bmatrix} 1, if\ I(x,y) = i\ and\ I(x+\Delta x, y+\Delta y) = j \\ 0, otherwise \end{Bmatrix}$$

Where: i and j are the pixel values; x and y are the spatial positions in the image I; the offsets $(\Delta x, \Delta y)$ define the spatial relation for which this matrix is calculated; and $I(\Delta x, \Delta y)$ indicates the pixel value at pixel(x, y).

Notations:

$p(i,j)$ (i,j)Th entry in the normalized gray level co-occurrence matrix

$p_x(i)$ iTh entry in the marginal probability matrix

N_g Number of distinct gray levels in the quantized image.

According to [40] the following list of haralick texture features are identified:

1) Angular Second Moment:

$$f_1 = \sum_i \quad \sum_j (p(i,j)^2) \tag{3.1}$$

Angular Second Moment measures the smoothness of the image. There are two cases,

If all pixels have same gray level I=k,

$p(k,k) = 1$ If $(i = j)$ and $p(i,j) = 0$ if otherwise.

$ASM = 1$

If all pixels have different gray level,

$$p(i,j) = {}^{1}/_{R} \qquad \& \qquad ASM = {}^{1}/_{R}$$

2) Contrast:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \quad \sum_{j=1}^{N_g} p(i,j) \right\} \tag{3.2}$$

Contrast measures the image contrast (locally gray level variations). The term n^2 is used to take off the largest contrast value.

3) Correlation:

$$f_3 = \frac{\sum_i \quad \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{3.3}$$

Correlation measures how the pixels are correlated with each other. Where μ_x, μ_y is the means and σ_x, σ_y are standard deviation of p_x, p_y

4) Sum of squares Variance: This feature puts relatively high weights on the elements that differ from the average value of P (i, j).

$$f_4 = \sum_i \sum_j (i - \mu)^2 \, p(i,j) \tag{3.4}$$

5) Inverse Difference Moment (Homogeneity): Homogeneity measures the distances of GLCM elements from the GLCM diagonal. Homogeneity ranges from 0 to 1. If adjacent pixels always have very similar Values of grayscale intensity, the homogeneity will be close to 1

$$f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j) \tag{3.5}$$

Inverse Difference Moment takes care of low contrast images. It takes care of low contrast images because of the inverse$(i-j)^2$.

6) Sum Average

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i) \tag{3.6}$$

7) Sum Variance

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i) \tag{3.7}$$

8) Sum Entropy

$$f_8 = -\sum_{i=2}^{2N_g} p_{x+y}(i) log\{p_{x+y}(i)\} \tag{3.8}$$

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

$$f_9 = -\sum_i \quad \sum_j p(i,j) log\{p(i,j)\} \tag{3.9}$$

Entropy takes low values for smooth images. It measures randomness.

10) Difference Variance

$$f_{10} = variance\ of\ p_{x-y} \tag{3.10}$$

11) Difference Entropy

$$f_{11} = -\sum_{i=0}^{N_g-1} p_{x-y}(i)\ log\{p_{x-y}(i)\} \tag{3.11}$$

12) Information Measure of Correction

$$f_{12} = \frac{HXY-HXY1}{max\{HX,HY\}} \tag{3.12}$$

$$f_{12} = (1 - exp[-2.0(HXY2 - HXY)])^{1/2}$$

$$HXY = -\sum_i \quad \sum_j p(i)log\{p(i)\}$$

Since some of the probabilities becomes zero and $log(0)$ is very high so arbitrary small positive constant is added to avoid the infinite number.

Where, HX and HY are entropies of p_x and p_y and

$$HXY1 = -\sum_i \quad \sum_j p(i,j)log\{p_x(i)p_y(j)\}$$

$$HXY2 = -\sum_i \quad \sum_j p_x(i)p_y(j)log\{p_x(i)p_y(j)\}$$

13) Maximal Correction Coefficient(Energy)

$$f_{13} = (second\ largest\ eigenvalue\ of\ Q)^{1/2} \tag{3.13}$$

Where,

$$Q(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$$

3.4.2. Features Based on Color Histogram

Color Histogram is one of the widely method for the color feature extraction. Convert the image to HSV color-space. Color Histogram based approach is based on the intensity value concentrations on all or part image represented as a histogram. A color histogram features for a given image is symbolized by a vector:

H={H[0],H[1],H[2],H[3],…….H[i],…….H[n]} Where **i** is the color bin in the color histogram and H[i] represents the number of pixels of color **i** in the image, and n is the total number of bins used in color histogram[40]. Histogram offers the explanation about the number of pixels available in the given color ranges. The larger the data set, the more likely want a large number of bins for improvement the performance. To extract Color Histogram features from the image, cv2.calcHist () function provided by OpenCV is used. The arguments it expects are the image,

channels, mask, histSize (bins) and ranges for each channel .Normalize the color histogram using normalize() function of OpenCV.

3.5. Training model

The training process is the main part of this study. As explained, two separate folders named infected and non-infected are created for training. These two folders hold blood smear image data (input data) that are classified by the guidance of systematic sampling. The training process uses different machine learning algorithms to build a model. The input for this process is a training containing images with their feature vector. After the training data is fed, the model building process starts to form a model. Features are represented in the form of vector numbers, therefore different supervising machine learning methods have been developed that can be applied to almost any data problem. Accuracy, number of parameters, and features are some of the points that have to be considered when choosing the best classifier for each specific use case. As explained in chapter two of the machine learning section six of the most commonly used algorithms are chosen. A brief overview of the selected machine learning methods would provide as follow in figure 3.6.

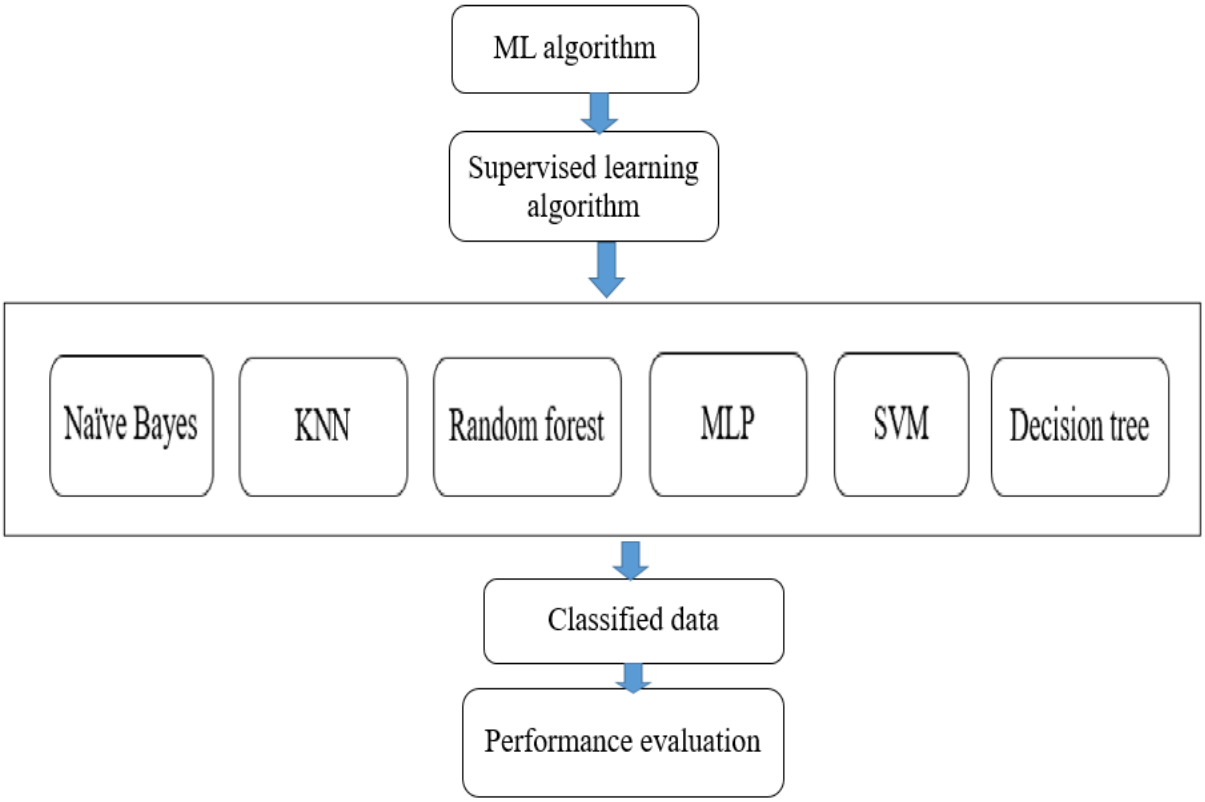


Figure 3. 6 ML Classification tasks

3.5.1. Hyper-parameter Ranges

The distribution of all hyper-parameters that were used in this experiment is described in Table 3.1

Table 3. 1 Hyper-parameters Ranges of the applied machine learning algorithms.

Machine learning Method	Parameters	Range
K-NN	K	[167]
Support vector machine	C Gamma Kernel Degree	[10] scale rbf 3
Random Forest	Estimators max_features min_samples_leaf max_depth min_samples_split criterion	[100] [auto] [1] [None] [2] ["gini index"]
Decision Tree	max_features min_samples_leaf max_depth min_samples_split criterion	[None] [1] [None] [2] ["entropy"]
MLP	Activation Alpha hidden_layer_sizes learning rate learning_rate_init max_iter momentum	Relu 0.0001 (100) constant 0.001 200 0.9
Naïve Bayes	class_prior alpha fit_prior	None 1.0 True

3.5.2. Decision Tree Classifier

The classifier is fitted with the Train Data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be mentioned in the result section. The learning algorithm recursively learns the tree for implementing the decision tree is given below. The decision tree algorithm pseudocode[41] is:

Step1: Create the root node of the tree *S*.

Step2: if all instances belong to the same class *C* then

Step3: *S*= leaf note labeled with class *C*

Step4: if attribute list (*A l*) is empty then.

Step5: *S*= leaf node labeled with the majority class.

Step6: Otherwise:

Step7: Select test attribute (*A t*) from *Al* with the greatest information gain.

Step8: Label node *S* as *A t*

Step9: For each possible value *v i* of *A t*

Step10: grow a branch from *S* where the test attribute *A t = v i*.

Step11: Let *S v* be the subset of *S* for each value of Attribute *A t = v i*.

Step12: if *S v* is empty then

Step13: label the node *S v* as a leaf with the most common class

Step14: Else below this branch add the subtree node

3.5.3. K-NN Classifier

The K-NN classifier algorithm is implemented in python sklearn library. The sklearn neighbors' library is used to import the K nearest Neighbors Classifier class. The object of the class is created and passed the arguments specified in the table 3.1.

The K-NN classifier is fitted with the Train Data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be explained in the result and discussion section.

The K nearest Neighbors algorithm pseudocode[42] is:

Step1: Load the training and test data

step2:Choose the value of K

step3 For each point in test data:

a, find the distance using Euclidean to all training data points

b, store the Euclidean distances in a list and sort it

c, choose the first k points

d, assign a class to the test point based on the majority of classes present in the chosen points

End

3.5.4. Multi-layer perceptron

The MLP classifier algorithm is implemented in python sklearn library. The object of the class is created and passed the arguments specified in the table 3.1. The MLP classifier is fitted with the Train Data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be explained in the result and discussion section. The Multi-layer perceptron algorithm pseudocode[43] is:

Step1: Initialize network weights
Step2: Present first input vector, from training data, to the network.
Step3: Propagate the input vector through the network to obtain an output
Step4: Calculate an error signal by comparing actual output to the desired (target) output.
Step5: Propagate error signal back through the network.
Step6: Adjust weights to minimize overall error
Step7: Repeat steps 2–7 with next input vector, until overall error is satisfactorily small

3.5.5. Random Forest Classifier

The random forest algorithm is combines decision trees, resulting in a forest of trees, hence the name Random Forest. The sklearn Ensemble library is used to import the Random Forest Classifier class. For the classification with random forest, an object of the class is created and passed the arguments mentioned in Table 3.1

The random forest classifier is fitted with the Train data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be explained in the result and discussion section. The random forest algorithm pseudocode [44]is:

Step1:Randomly select M features from total n features where M<< n
Step2:Among the M Features, calculate the node d using the best split point
Step3:Split the node into daughter nodes using the best split
Step4:Repeat the 1 to 3 steps until one number of nodes has been reached
Step5:Build forest by repeating steps 1 to 4 for K number times to create K number of trees

3.5.6. Naïve Bayes Classifier

Naïve Bayes is one of the supervised algorithms applied in this thesis. The naïve Bayes Classifier model is implemented in python sklearn library. The sklearn Tree library is used to import the naïve Bayes classifier class. The object of the class is created and passed the arguments specified in Table

3.1.The classifier is fitted with the Train data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be mentioned in the result section. The Naïve Bayes algorithm pseudocode[45] is:

Input:

Training dataset T,

F=(f1,f2,f3.....fn) //value of the predictor variable in testing dataset

Step1: Read the training data set T;

Step2: Calculate the mean and standard deviation of the predictor variable in each class;

Step3: Calculate the probability of fi using the Gaussian density equation in each class;

Until the probability of all predictors variables (f1, f2.....fn)

Step4: Calculate the likelihood for each class;

Step5: Get the greatest likelihood

end

3.5.7. Support vector machine

Support vector machine classifier algorithm is implemented in python sklearn library. The sklearn SVM library is used to import the SVM class. For the classification with SVM, an object of the class is created and passed the arguments mentioned in Table 3.1.

The SVM classifier is fitted with the Train data and Train test dataset. The confusion matrix is generated to test the accuracy of the model as it will be explained in the result and discussion section. The Support vector machine pseudocode is:

```
# Load data
filename =’path to training data’
Data=read Data (filename)

# Partition the data into training and testing splits

Print ([INFO] constructing the training/testing split...)

 (Train Data, test Data, train Labels, test Labels) = train_test_split (data, labels, test size=0.3, random

_state=42)

Model= SVM (kernel="RBF", C=10,degree=3, gamma=scale),

# make predictions on our data and show a classification report

Print ([INFO] evaluating...)

Predictions = model. Predict (test data)

Print (classification_ report (test labels, predictions, target_ names=le.classes_))
```

3.6. Testing Phase

As specified in the proposed architecture of the system in figure 3.1, the testing phase followed the same function calls as the training phase does. An input image passes through the same techniques and algorithms like preprocessing and feature extraction tasks. The only difference here is, the feature vectors in the testing phase are input for the model without any label. The classifier is expected to return the predicted label (infected or non- infected) based on the provided feature vector. For testing and comparing the different Machinelearning classifiers with their different parameters and feature extraction techniques a total of 4133 images of healthy and 4133 images of unhealthy blood smear were used.

CHAPTER FOUR

EXPERIMENTAL RESULT AND DISCUSSION

4.1. Dataset Collection

A public dataset of 27,558 images of unhealthy and healthy blood smear were downloaded from the US National Institutes of Health website. In this study the malaria dataset is a collected of a total of 27,558 segmented cell images extracted from blood smear images were used from US National Institutes of Health recorded data (website). The cell images are categorized into two folders, unhealthy and healthy, with 13,779 cell images in each, making this a balanced dataset. The recorded dataset is divided into training (50%), validation (20%) and testing (30%). The dataset contains a variety of different images including the image with various resolutions.

(a)**Training dataset**

Training data set is the general term for the samples used to create the model.

Class	Data size
Infefeted	9646
Non- infected	9646

(b) **Test dataset**

Test data set is used to qualify performance.

Class	Data size
Infefeted	4133
Non- infected	4133

4.2. Data Preparation

Data preparation is the most significant phase of the data analysis activity which involves the construct of the final data set (data that will be fed into the modeling tool) from the initial raw data. From the total datasets (27,558 images) mentioned above in the datasets section, 19292 (70%) of those were used for training but from 70% the 20% of the data were used for validation and the remaining 8266 (30%) were used for testing. The datasets in the testing set are different from the dataset that is used for the training set. To prepare those amounts a systematic random sampling techniques was used. In a systematic random sampling technique probability sampling technique which sample members from the population are selected based on a random starting point but with a fixed, periodic interval. Sampling interval, is calculated by dividing the population size by the desired sample,

elements of a sample are chosen at a regular interval of the population except for the first element. Steps in selecting a systematic random samples are:

- Firstly calculate the sampling interval the number of population size divided by the number of sub groups needed for the sample.
- Then select a random start between one and sampling interval
- Repeatedly add sampling interval to select subsequent subgroup

4.3. Dataset Split

In the training phase, before each image is fed into the training models the training dataset is divided into training and validation depending on the split ratio using train_test_split of the sklearn package. Validation is the part of training. Using the sklearn's train_test_split method training dataset was partitioned into training and validation set. The training data is used to make sure the machine classifies patterns in the data, and the testing data is used to see how well the machine learning can predict new answers based on its training model. The test set is not used for training models so that when testing occurs, that data is unseen by the model. The metrics achieved for each iteration are aggregated to form the final result.

4.4. Software tools and libraries

As a programming language Python 3.7 is used which an open-source, with variety is of free libraries, rich documentation, including contributor support. The supportive libraries and Software tools are listed next.

- Numpy : library for mathematical functionalities
- Matplotlib : plotting library
- OpenCV : image processing library and computer vision library
- Scikit-learn : machine learning library
- Mahotas : additional computer vision and image processing library

Jupiter notebook development IDE is used and the program is done with Python 3.7 language with OpenCV and Keras. OpenCV is an open-source library of image processing functions, whose goal is real-time computer vision. Keras is the system modular library written in Python capable of running on top of Tensor Flow. The TF (Tensor Flow) was selected as a backend that both TF and Keras were optimized to perform tasks. Both systems are implemented in Python which allows the user to work with them in a compact way without having to use multiple files as a programming language.

4.5. Setting up Development Environment

The implementation of this research work is done under a machine that has the following Specification details. Experiments and related analysis processes are done:

- Computer with Intel (R) Core (TM) i5- 4210U CPU
- Speed 2.4GHz
- 8.00 GB RAM
- 750 GB hard disk space
- Windows 10 (Pro) installed

4.6. Evaluation Metrics

A confusion matrix is an evaluation metrics that are mostly used to define the performance of a classification model on a set of test data for which the correct values are known. Confusion Matrix also is known as the contingency table provides a comprehensive overview by summarizing the classification results. It shows the individual results for each of the categories by tabulating the predicted and actual categories.

True Positive (TP):- is the amount of the specified class detected when it is actually that class Predicted values correctly predicted as actual positive or result where the model correctly predicts the positive class

True Negative (TN):- it is the sum of all true positives except the specified class true positive value or Predicted values correctly predicted as an actual negative . True negative the result where the model correctly predicts the negative class.

False Positive (FP):- It is the sum of the predicted class row except for the TP value of that class. It is the amount of data predicted as that class but actually, they are not a members of the predicted class or Predicted values incorrectly predicted as an actual positive. Negative values predicted as positive.

False Negative (FN):- It is the sum of the actual class column except the TP value of that class. It is the amount of actual class members but predicted as member of other class or Positive values predicted as negative.

In this research case, these representations can be interpreted as:

TP: number of infected blood smear classified as infected blood smear.

TN: number of healthy blood smear classified non- infected blood or no malaria in the blood.

FP: number of healthy blood smear classified as infected blood smear or have a parasite in the blood.

FN: number of infected blood smear classified as non-infected blood smear.

Total: total number of samples (blood smear image)

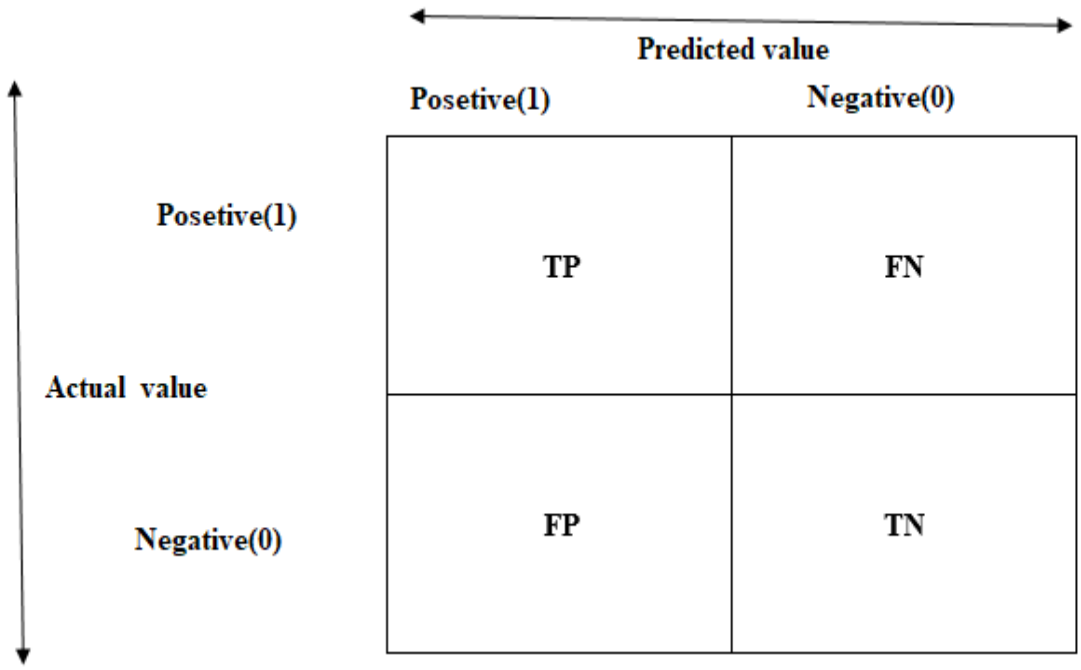


Figure 4. 1 Confusion Matrix for the Binary Classification

Figure 4.1 shows a confusion matrix for a two-class classification problem. The numbers along the diagonal from lower-right to upper-left symbolize the accurate decisions made, and the numbers outside this diagonal represent the mistakes. The equations of the most commonly used metrics can be calculated from the matrix. The total accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the overall number of samples. Other performance measures, such as recall (sensitivity), precision, and F-measure are also used for calculating other accumulated performance measures. Different metrics are used to measure the performance of each method.

Accuracy is the proportion of samples that are classified properly among the whole samples of the dataset. It is calculated using the formula as follow:

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FP+FN} \qquad (4.1)$$

Where true positive refers to TP, True negative refers to TN, false- positive refers to FP, and false negative refers to FN.

Precision:

Precision is a measure for the positive predictive value and is given calculated by the formula as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (4.2)$$

Recall:

Recall is a measure for the true positive rate and the formula how to calculate is as follows:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4.3)$$

High precision means a low false-positive rate and a high recall means a low false-negative rate. High precision and high recall mean that you have accurate results but if you have a high recall and low precision, then it means that most of the predicted values are false. If you have a low recall and high precision at the equal time, then it means that most of the predicted values are accurate. The best case for a model is when it has a high precision and a high recall. One way to summarize both metrics precision and recall is the F-score.

F-score

F-score is a harmonic mean for both recall and precision is calculating as in the following:

$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.4)$$

4.7. Result of the Study

4.7.1. Experiment 1: KNN Classification

Table 4. 1 Results from KNN and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
Knn +color histogram	Non- infected	.82	.84	.83
	infected	.84	.81	.82
	Weighted Avg	.83	.83	.83
Knn+color + haralick textures	Non- infected	.69	.78	.73
	Infected	.75	.65	.70
	Weighted Avg	.72	.72	.72
Knn + haralick textures	Non- infected	.67	.76	.71
	Infected	.72	.63	.67
	Weighted Avg	.70	.70	.69

As indicated in table 4.1 the classification performance metrics on average 83% precision, 83% recall, and 83% f1 score is obtained for the color histogram feature. A 72% precision, 72% recall, and 72% f1 score is obtained by combining the two features and 70% precision, 70% recall, and 69% f1 score is obtained for the haralick texture feature respectively. Looking at the F1-score, non-infected has 83%

and then infected with 82%, have the highest F-measure value based on the color histogram features than other attributes.

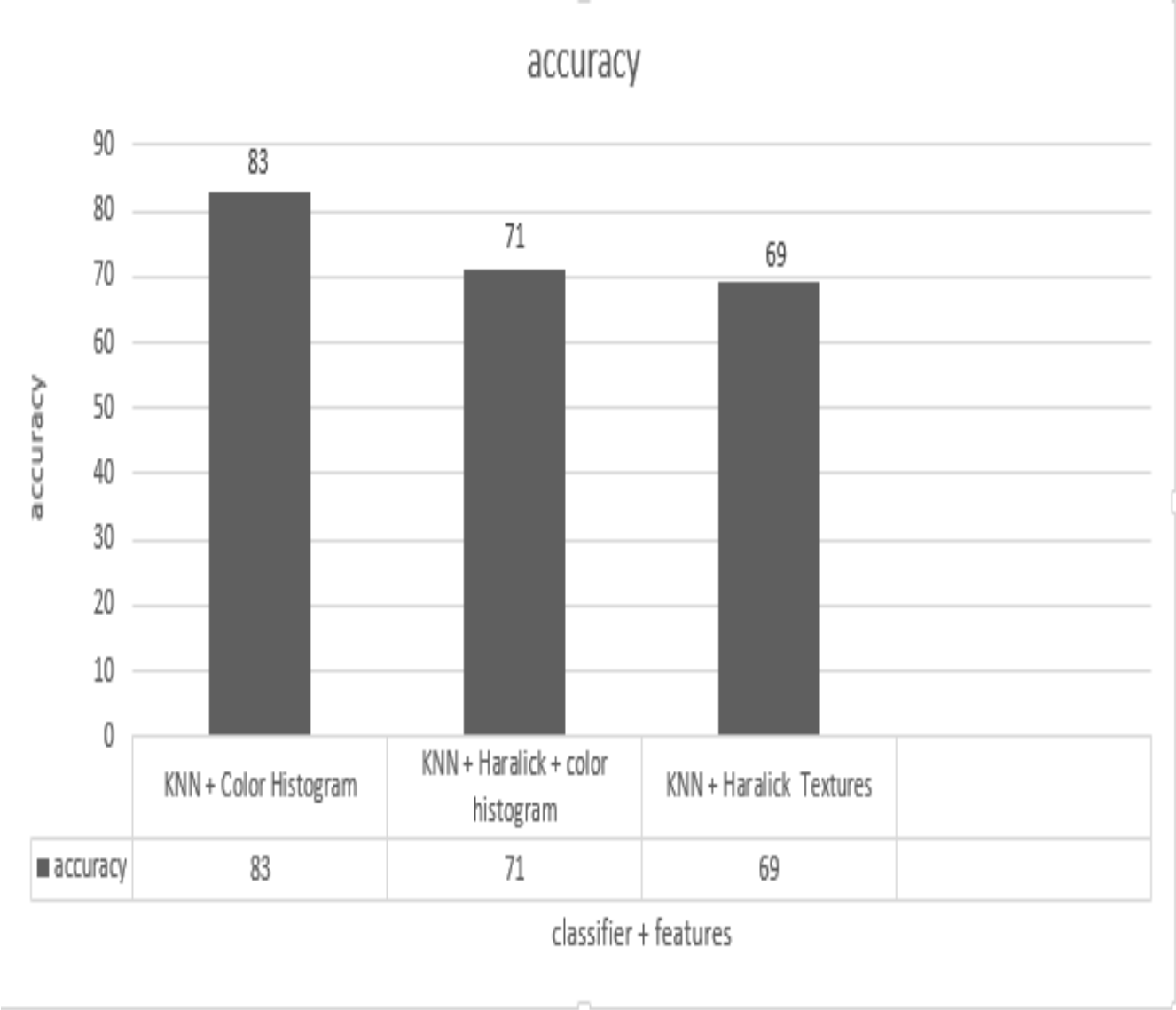


Figure 4. 2 Accuracy comparison of KNN model

The histogram chart above compares the accuracy of KNN models using color histogram, haralick texture and a combination of color histogram and haralick texture features. Color histogram outperforms by combining color histogram with haralick textures and haralick textures by 12% and 14% respectively. It is apparent from figure 4.2 that the accuracy of KNN models using color histograms are better than the others. Taken together, these results suggest that the color histogram feature for malaria disease detection using KNN is the best descriptor than the haralick texture and the combination of haralick with color histogram features for classification.

4.7.2. Experiment 2: Naïve Bayes Classification

Table 4. 2 Results from NB and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
NB +color histogram	Non- infected	.85	.91	.88
	Infected	.90	.84	.87
	Weighted Avg	.88	.88	.88
NB+ color + haralick textures	Non- infected	.71	.78	.74
	Infected	.76	.68	.72
	Weighted Avg	.74	.73	.73
NB+ haralick textures	Non- infected	.70	.71	.70
	Infected	.71	.70	.70
	Weighted Avg	.71	.71	.70

As shown in Table 4.2 based on their precision value the classes descending order on non- infected and infected based on the color histogram, by combining haralick with color histogram features and haralick textures .Infected class is the most precise than the other because of the color histogram. But the non-infected class is less precise than the other due to the haralick texture features. From the two class domains their recall value order is non-infected and infected. It means that in non-infected and infected class the model was able to predict truly 91% and 84% to their respective class out of the given true class data based on the color histogram. In the case of non-infected, infected class it is around 78% and 68% respectively based on the combined features and non-infected, infected class it is around 71% and 70% respectively based on the haralick feature. Looking at the F1-score, non-infected has 88% and then infected with 87%, have the highest F-measure value based on the color histogram features than other attributes.

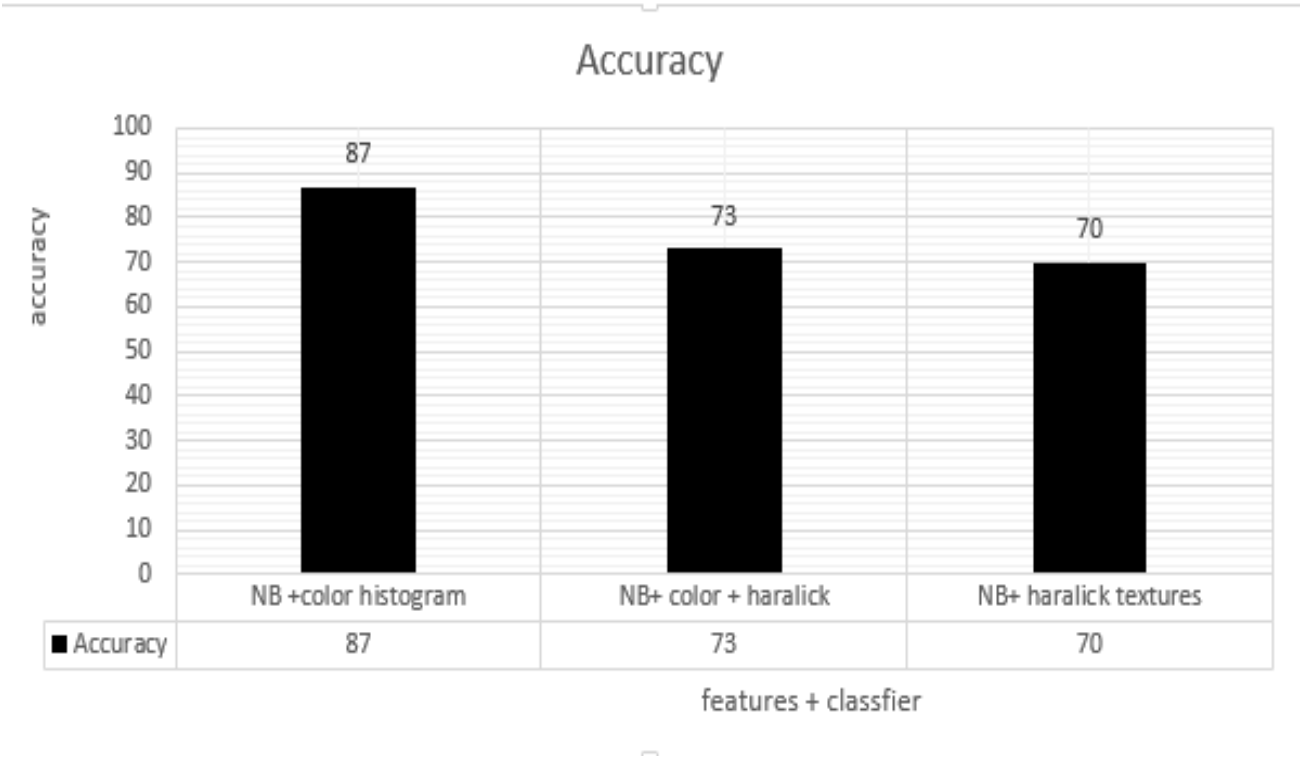


Figure 4. 3 Accuracy comparison of naïve Bayes

As can be seen from figure 4.3 the naïve Bayes algorithm has generated relatively comparable model accuracies with varied parameters. The chart histogram presents the accuracy summary statistics for naive Bayes classifiers using color histogram, haralick texture and combination (haralick with color) of them with Bernoulli parameters. From this data it can be observed that naive Bayes classifiers using haralick texture features with Bernoulli parameters resulted in the lowest accuracy value than other mechanisms. An experimental analysis shows that naive Bayes classifier using color histogram features with Bernoulli is the better model in detecting and classifying malaria disease.

4.7.3. Experiment 3: Support vector machine Classification

Table 4. 3 Results from SVM and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
SVM + color histogram	Non- infected	.89	.91	.90
	Infected	.91	.89	.90
	Weighted Avg	.90	.90	.90
SVM+ color + haralick	Non- infected	.77	,79	.78
	Infected	.78	.76	.77
	Weighted Avg	.78	.78	.78
SVM + haralick textures	Non- infected	.74	.73	.73
	Infected	.74	.75	.74

	Weighted Avg	.74	.74	.74

As indicated in the classification performance metrics from Table 4.3, on average 90% precision, 90% recall, and 90% f1 score is obtained from the color histogram features. The precision of this model at infected cases is 91% which indicates that the model is good at detecting infected cases. SVM classifiers using Haralick texture features were used for classification is not good performance when compared with others in the table 4.3. The experimental result showed on average 74% precision, 74% recall, and 74% f1 score using haralick textures. Overall, the SVM model for the haralick texture feature did not outperform.

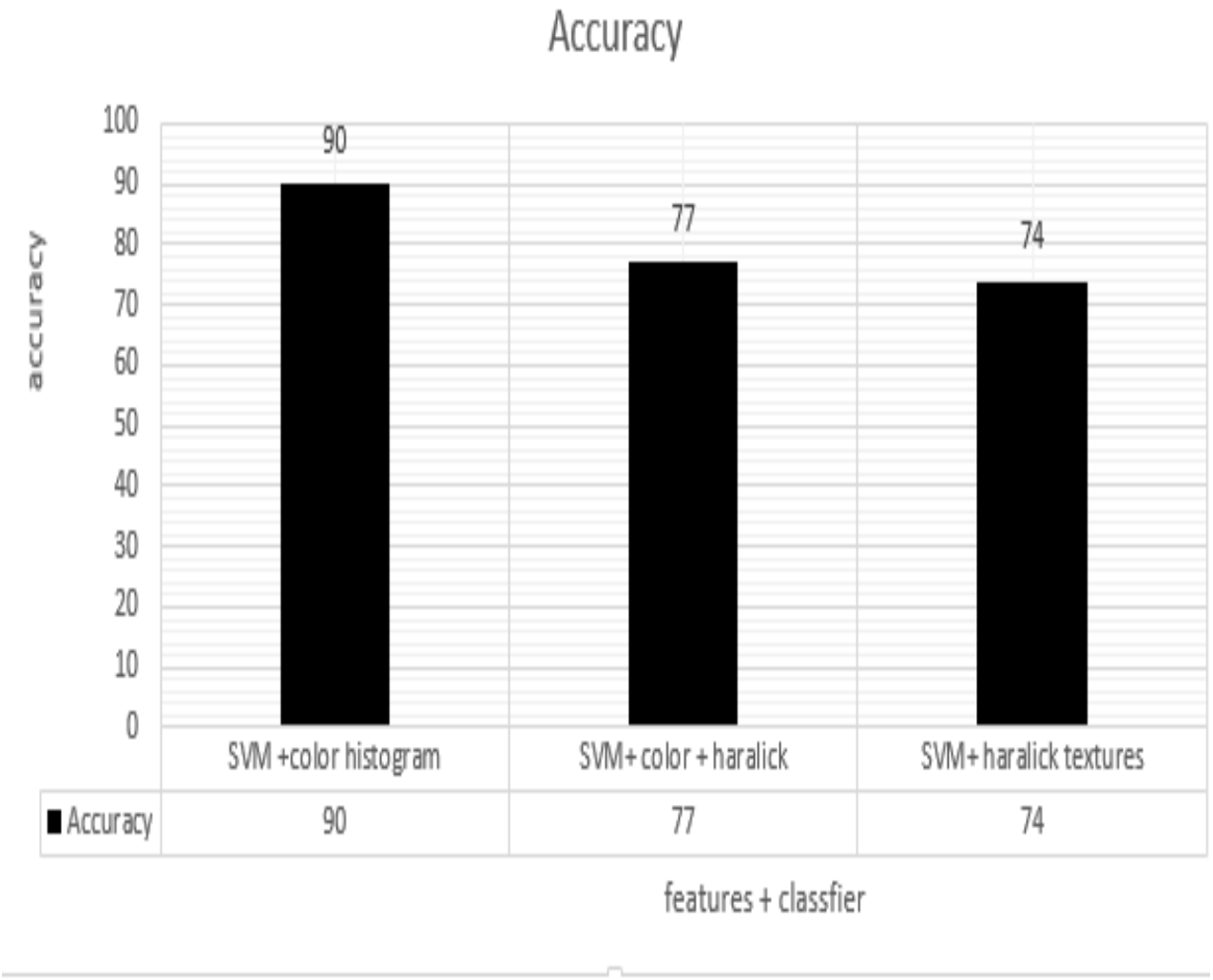


Figure 4. 4 Accuracy comparison of support vector machine

As can be seen from Figure 4.4 the SVM algorithm has generated relatively comparable model accuracies with varied parameters. The chart histogram provides the accuracy obtained from the SVM classifier using color histogram features, haralick texture and a combination of color and texture with

linear, polynomial and RBF parameters. From the chart, it can be seen that by far the greatest accuracy is achieved based on the color histogram with RBF parameters than other mechanisms. Experimental analysis indicates that the SVM classifier using color histogram features with RBF is the better model classifying malaria disease.

4.7.4. Experiment 4: MLP Classification

Table 4. 4 Results from MLP and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
MLP +color histogram	Non- infected	.84	.94	.89
	Infected	.93	.82	.87
	Weighted Avg	.89	.88	.88
MLP+ color + haralick	Non- infected	.77	.84	.80
	Infected	.83	.82	.82
	Weighted Avg	.80	.83	.81
MLP+ haralick textures	Non- infected	.73	.79	.76
	Infected	.77	.70	.73
	Weighted Avg	.75	.75	.75

As indicated in the classification performance metrics of Table 4.4, on average 89% precision, 88% recall, and 88% f1 score is obtained from the color histogram features. MLP classifiers using Haralick texture features achieved on average 75% precision, 75% recall, and 75% f1 score. MLP classifiers using a color histogram with Haralick texture features achieved on average 80% precision, 83% recall, and 81% f1 score. Overall, the MLP model for the haralick texture feature did not outperform when compared with color histogram and by combining the two features.

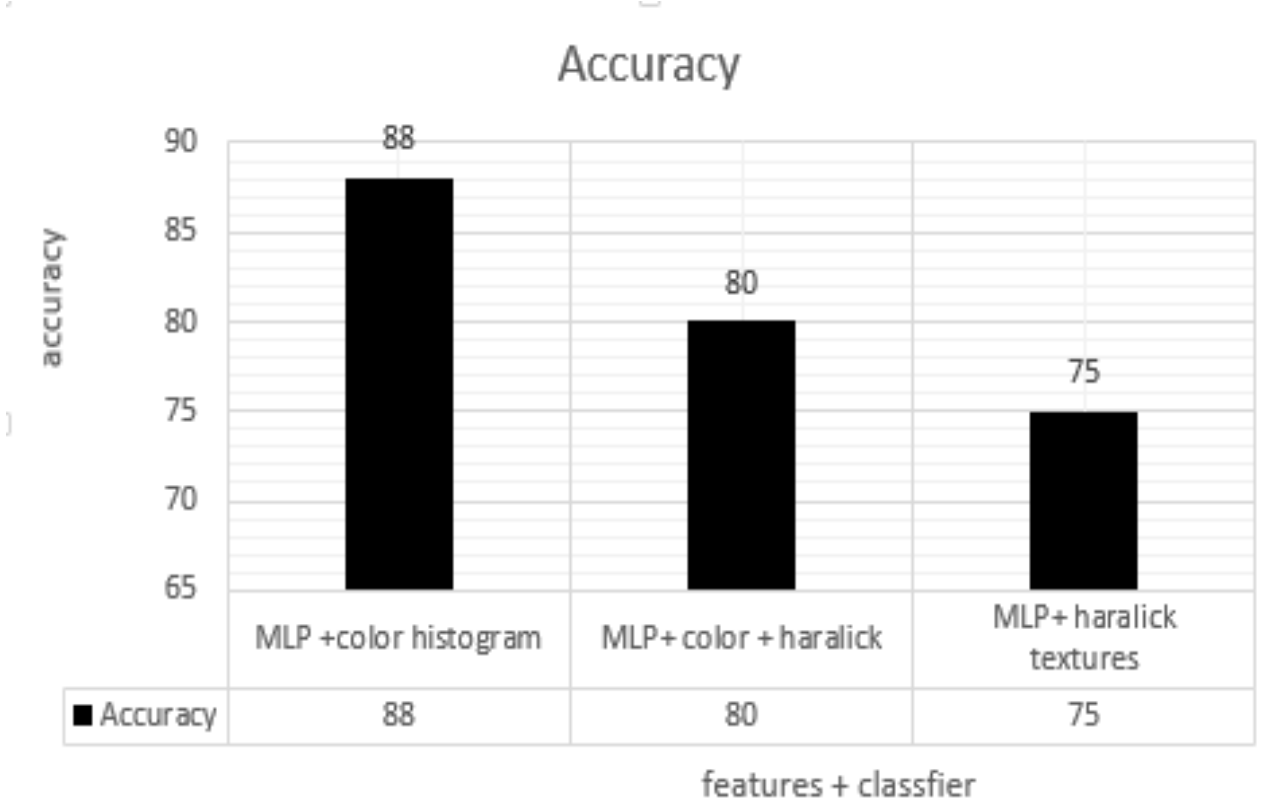


Figure 4. 5 Accuracy comparison of MLP

The chart histogram provides the accuracy obtained from the MLP classifier using color histogram features, haralick texture, and combination of color and texture. From the above chat experimental result, observed that the combination feature sets(haralick with color histogram), haralick texture, color histogram feature the color histogram feature relatively better results in MLP for malaria detection than other feature sets within an accuracy of 88%.

4.7.5. Experiment 5: decision tree classification

Table 4. 5 Results from DT and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
DT+ color histogram	Non- infected	.92	.91	.91
	Infected	.91	.92	.91
	Weighted Avg	.91	.91	.91
DT+ color + haralick textures	Non- infected	.79	.81	.80
	Infected	.81	.79	.80
	Weighted Avg	.80	.80	.80
DT + haralick textures	Non- infected	.75	.76	.75
	Infected	.76	.75	.75
	Weighted Avg	.76	.76	.75

As shown in Table 4.5, the classification average precision of each class is 91% for infected and non-infected using color histogram features. The precision is descending order based on color histogram features when compared combining the two features and haralick textures. When compared this result color histogram is more precise than other features. From the two class domains their recall value order is non-infected and infected. It means that in non-infected and infected class the model was able to predict truly above 91% and 92% to their respective class out of the given true class data based on the color histogram. In the case of non-infected, infected class it is around 81% and 79% respectively based on the combined features and non-infected, infected class it is around 76% and 75% respectively based on the haralick feature.

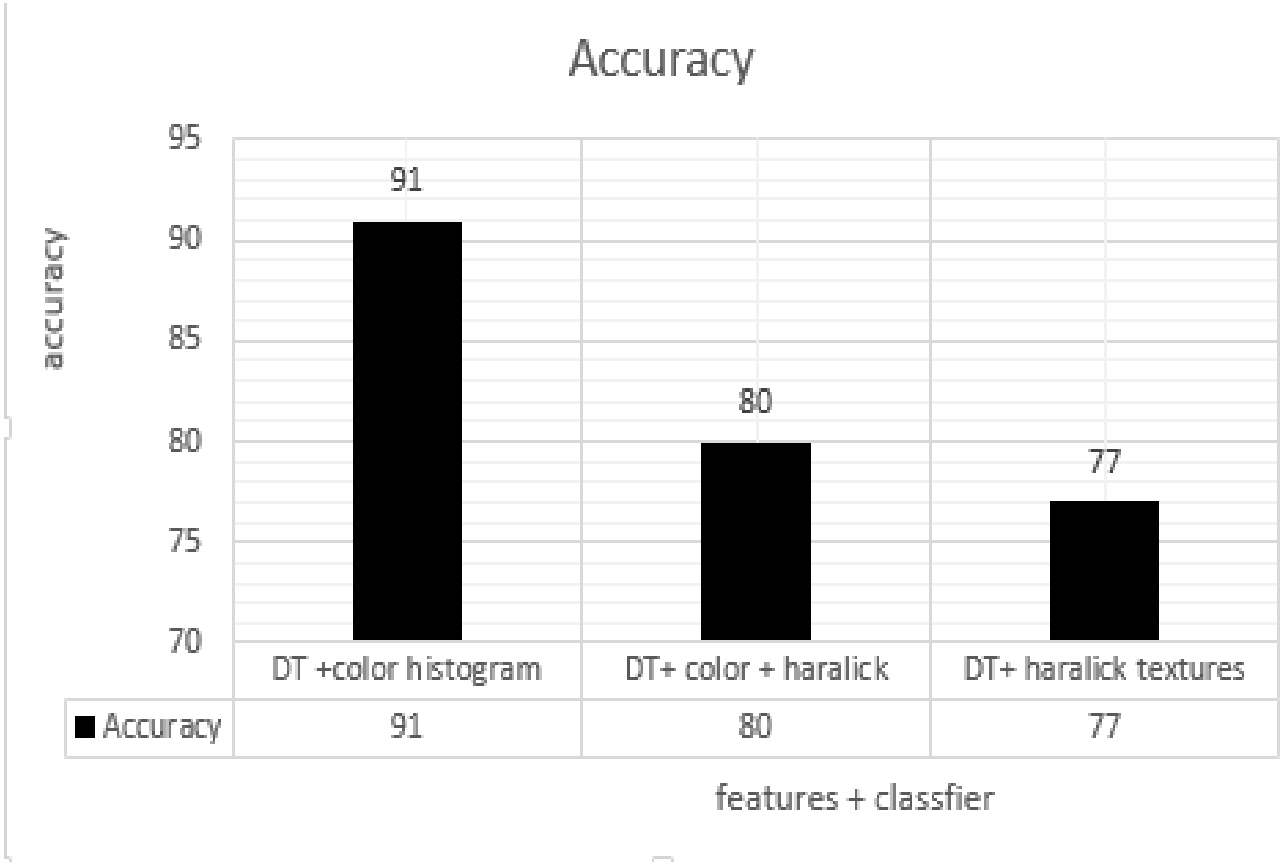


Figure 4. 6 accuracy comparison of decision tree

The above chart histogram presents the accuracy summary statistics for the decision tree classifier using color histogram feature, haralick texture, and combining the two features. From this data, it can be observed that the decision tree classifier using the haralick texture feature resulted in the lowest accuracy value than other mechanisms. An experiment shows that the decision tree classifier using color histogram features is the best possible in detecting and classifying malaria disease. Color histogram features are the better descriptor of malaria disease for decision tree classifier. Color histogram feature relatively better results in decision tree classifier for malaria detection than other feature sets within an accuracy of 91%.

4.7.6. Experiment 6: Random forest Classification

Table 4. 6 Results from RF and descriptor (scale 100%)

Classifier + descriptor	Class	Precision	Recall	F1-score
RF+ color histogram	Non- infected	.96	.94	.95
	Infected	.94	.96	.95
	Weighted Avg	.95	.95	.95
RF +color + haralick	Non- infected	.80	.88	.84
	Infected	.87	.78	.82
	Weighted Avg	.83	.83	.83
RF + haralick textures	Non- infected	.76	.83	.79
	Infected	.82	.74	.78
	Weighted Avg	.79	.79	.79

As shown in table 4.6 based on their precision value the classes descending order is non- infected and infected based on the features of color histogram, by combining color with haralick and haralick textures. Non infected class is the most precise than the other based on the color histogram. But non-infected class is less precise than the other because as based on the haralick texture features. From the two class domains their recall value order is non -infected and infected. It means that in non-infected and infected class the model was able to predict truly 94% and 96% to their respective class out of the given true class data based on the color histogram. In the case of non -infected, infected class it is around 88% and 78% respectively based on the combined features and non- infected, infected class it is around 83% and 74% respectively based on the haralick feature. Non-infected has 95% and then infected with 95%, have the highest F-measure value due the color histogram than other attributes.

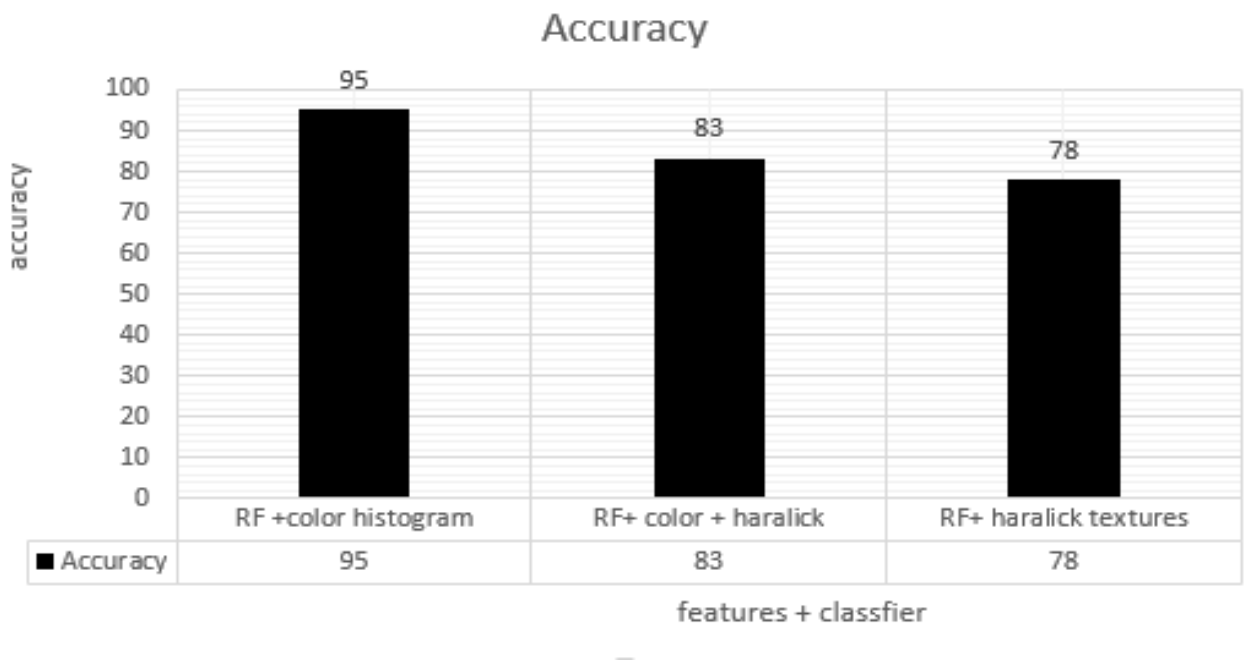


Figure 4. 7 accuracy comparison of random forest

The chart histogram presents the accuracy summary statistics for the Random Forest classifier using color histogram, haralick texture, and combining the two features. From this data, it can be observed Random forest classifiers using the haralick texture feature resulted in the lowest accuracy value than other mechanisms. The experiments show that the Random forest classifier using color histogram features is the best possible in detecting and classifying malaria disease. Color histogram features are the better descriptor of malaria disease detection using the Random forest. Generally, the accuracy comparison among the three features implemented by Random forest classifier using color histogram feature descriptor performs better in the classification.

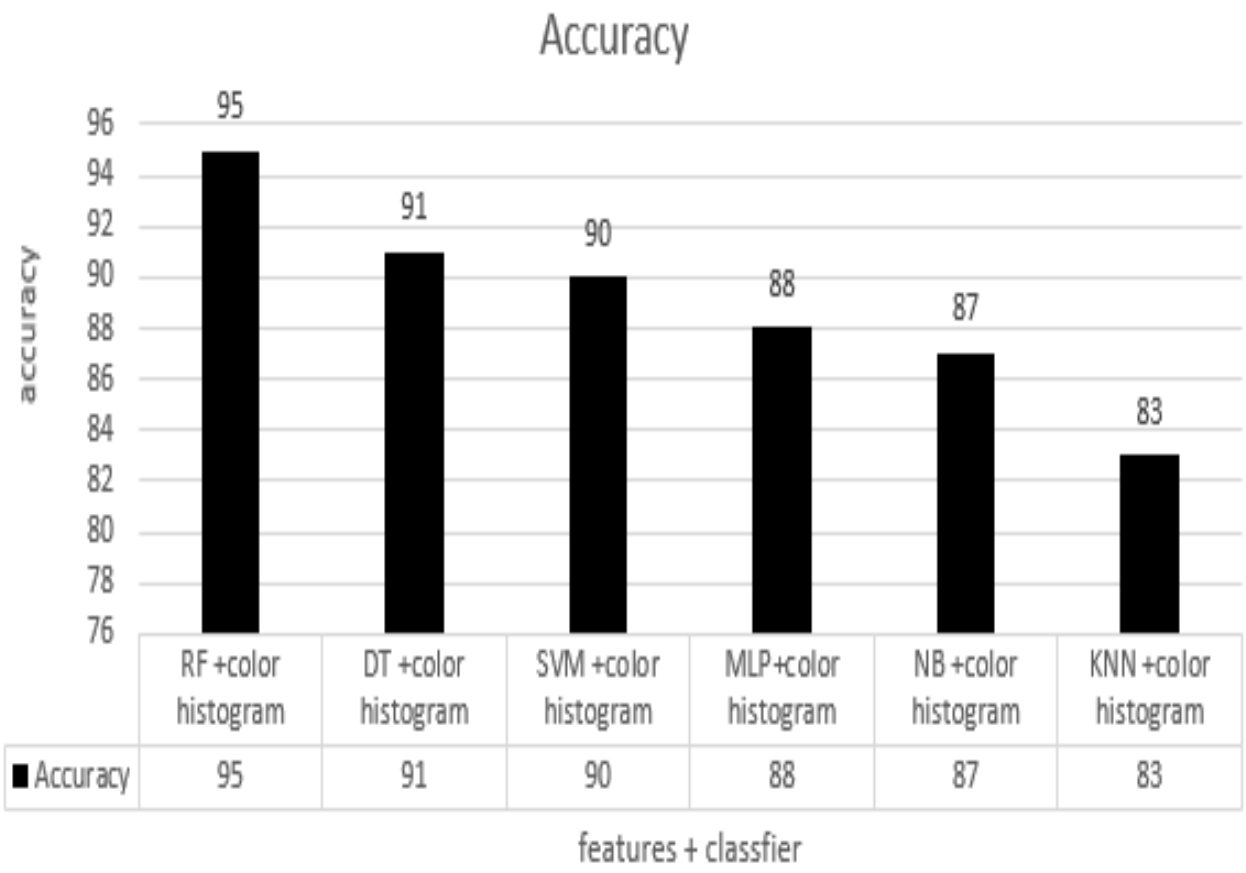


Figure 4. 8 accuracy comparison of the supervised models

Referring to the results obtained, six of the classifiers have shown high performance with average accuracy of 95%, 91%, 90%, 88%, 87%, and 83% for RF, DT, SVM, MLP, NB and KNN respectively. RF outperforms DT, SVM, MLP, NB and KNN by 4%, 5%, 7%, and 8% and 12% respectively. Compared to results the random forest algorithm gives better results than the other machine learning classifier. Thus, due to the fact that random forest aggregates more than two decision trees to avoid overfitting as well as error due to bias making it more accurate, and thereby showing the feasibility of its usage in real-time applications for determining whether a cell is infected with the malaria parasite.

4.8. Answers to the Research Questions

RQ #1: Which supervised machine learning model provides the highest performance?

From the analyzed and compared models, the random forest algorithm gave a better result. For instance, the random forest algorithm achieved an average accuracy of 95%, average precision of 95.0%, average recall of 95.0%, and an average F1 value of 95.0% over a test dataset of previously unseen 8266.

RQ #2: Which feature extraction technique provides the highest performance? Color histogram features

RQ #3: Can the combination of feature sets of different feature extraction techniques outperform an individual feature set of single feature extraction techniques? No

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

In this thesis, developing and implementing automatic malaria detection using machine learning approaches is proposed. To conduct the experiments a total of 27,558 segmented cell images extracted from thin blood smear slide images were used from the National Institute of Health (NIH) recorded data. Through the literature review, several researchers proposed different machine learning mechanisms to automate and evaluate a malaria disease. However, in most of the works, their system does not use enough dataset or they have not achieved sufficient or expected results. The main objective of this study was to build machine learning models that can automatically classify whether a cell is healthy or infected and assess the performance of the developed model. The acquired data are noisy, different image processing techniques such as median filter, histogram equalization and normalization were applied to enhance the quality of the image. After enhanced the quality of the acquired images, the original vector space of the acquired image is transformed to form a new minimalistic feature vector space to distinguish between infected and non-infected red blood cells using a color histogram, haralick texture, and combination of a color histogram and haralick texture (i.e. color histogram-haralick texture features) feature extraction techniques. Then, suitable supervised machine learning classifier techniques with different model parameters such as support vector machine, decision tree, K nearest neighbor, multi-layer perceptron, random forest, and naïve Bayes were used to categorize the features into their different classes. By applying dataset for every classification technique, their performance was measured based on precision, recall, f1 score, and accuracy; and showed the evaluation result by tabulating the predicted and actual categories using a confusion matrix. From the analyzed and compared models, the random forest algorithm gave a better result. For instance, the random forest algorithm achieved an average accuracy of 95%, average precision of 95.0%, average recall of 95.0%, and an average F1 value of 95.0% over a test dataset of previously unseen 8266 images. Furthermore, this classification proposed an experimental analysis using conventional supervising machine learning methods. This automated newly developed system would alleviate the problems faced before in the over-dependence of expert's skills and experience and delay in diagnosis. Finally, the results from this thesis are compared with previous related works that are based on the dataset. The proposed system was able to achieve better performance especially in the case of feature extraction and classification.

5.2. Recommendation and Future Works

This platform for the automatic detection of Malaria provides many useful and interesting directions in this area. There may still be a gap to be improved on the malaria detection system using machine learning approaches, since the problem is a fatal Public health issue as stated on the problem statement. The main goal of public health issue problem solving is to work for performance improvement to be done continuously until the highest accuracy level is reached. Therefore, the following are some notable future work recommendations observed while implementing this research work.

- Adding more features to represent the image will undoubtedly increase the performance of the classification.
- If more labeled data sets are found it will lead to better result
- Our proposed research focused to classify healthy and unhealthy further researches could consider to perform to build a model that can classify species of malaria.
- Our proposed research the time consumption of the algorithm proposed for training is approximately 10 to 15 min, this time can be improved by applying parallel Computing methods to the implementation.
- Our proposed research focused on only feature extraction further researches could consider feature selection on malaria detection system using machine learning approaches.

REFERENCES

[1] World Health Organization, *WHO | The World malaria report 2018*. 2018.

[2] H. A. Nugroho, S. A. Akbar, and E. E. H. Murhandarwati, "Feature extraction and classification for detection malaria parasites in thin blood smear," *ICITACEE 2015 - 2nd Int. Conf. Inf. Technol. Comput. Electr. Eng. Green Technol. Strength. Inf. Technol. Electr. Comput. Eng. Implementation, Proc.*, vol. 1, no. c, pp. 197–201, 2016.

[3] B. Berhe, F. Mardu, H. Legese, and H. Negash, "Seasonal distribution and seven year trend of malaria in North West Tigrai: 2012–2018, Ethiopia; 2019," *Trop. Dis. Travel Med. Vaccines*, vol. 5, no. 1, pp. 1–7, 2019.

[4] A. Ababa and A. B. Siragi, "Addis Ababa Institute of Technology School of Civil and Environmental Engineering Assessment of Stormwater Drainage System in Assosa Town Addis Ababa University Addis Ababa institute of Technology School of Civil and Environmental Engineering Assessment o," 2019.

[5] Y. Purwar, S. L. Shah, G. Clarke, A. Almugairi, and A. Muehlenbachs, "Automated and unsupervised detection of malarial parasites in microscopic images," *Malar. J.*, vol. 10, no. December, 2011.

[6] Z. Jan, A. Khan, M. Sajjad, K. Muhammad, S. Rho, and I. Mehmood, "A review on automated diagnosis of malaria parasite in microscopic blood smears images," *Multimed. Tools Appl.*, vol. 77, no. 8, pp. 9801–9826, 2018.

[7] J. Somasekar, A. Rama Mohan Reddy, and L. Sreenivasulu Reddy, "An efficient algorithm for automatic malaria detection in microscopic blood images," *Commun. Comput. Inf. Sci.*, vol. 270 CCIS, no. PART II, pp. 431–440, 2012.

[8] D. Usha, "Detection of Malaria Based on the Blood Smear Images Using Image Processing Techniques," vol. 5, no. 21, pp. 1–4, 2017.

[9] S. Chauhan, "Pattern Recognition System using MLP Neural Networks," *IOSR J. Eng.*, vol. 02, no. 05, pp. 990–993, 2012.

[10] "Comparison of Image Preprocessing Techniques for Textile Texture Images," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 12, pp. 7619–7625, 2010.

[11] Z.-H. Cho, Y.-D. Son, and Y.-B. Kim, *ebooksclub.org__Biomedical_Image_Processing.pdf*. 2011.

[12] H. Ackar, A. A. Almisreb, and M. A. Saleh, "A Review on Image Enhancement Techniques," *Southeast Eur. J. Soft Comput.*, vol. 8, no. 1, 2019.

[13] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, “Image analysis and machine learning for detecting malaria,” *Transl. Res.*, vol. 194, pp. 36–55, 2018.

[14] A. R. Smith, “Color gamut transform pairs,” *Proc. 5th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 1978*, no. August 1978, pp. 12–19, 1978.

[15] “sciencedirect-topic-median-filter.pdf.”2018 .

[16] D. Saibannavar and S. S. Bisalapur, “detection of marsh fever in blood images using neural network,” vol. 8354, no. 3, pp. 101–115, 2014.

[17] A. Haralick, Robert M., Shanmugam. K and I. Dinstein, “TexturalFeatures,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, No. pp. 610–621, 1973.

[18] O. Pentakalos, “Introduction to machine learning,” *C. Impact 2019*, 2019.

[19] Muhammad Jamil Moughal, “Which Machine Learning algorithm to use? – Muhammad Jamil Moughal – Medium,” 2018. .

[20] "datacamp.com/community/tutorials/svm-classification-scikit-learn-python” December 27th, 2019.

[21] M. K. Albert, D. W. Aha, and D. Kibler, “Instance-Based Learning Algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[22] "https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn”.2020

[23] "javatpoint.com/machine-learning-decision-tree-classification-algorithm"Feb 13th,2019.

[24] G. H. John and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers,” 2013.

[25] N. Bayes, “Naive Bayes classifier,” pp. 1–9, 2006.

[26] A. Cutler, D. R. Cutler, and J. R. Stevens, “Ensemble Machine Learning,” *Ensemble Mach. Learn.*, no. February 2014, 2012, doi: 10.1007/978-1-4419-9326-7.

[27] Abhay Padda, “Introduction to Random Forest.,” *March* 2018.

[28] "scikit-learn.org/stable/modules/neural_networks_supervised", July 2019.

[29] L. Anguage, U. Bhavsar, Y. Kajabe, and S. Patil, “R Eview S Ummarization With M Achine L Earning,” vol. 6, no. 5, pp. 454–457, 2014.

[30] S. S. Savkare, “Automatic Detection of Malaria Parasites for Estimating Parasitemia,” *Int. J. Comput. Sci. Secur.*, vol. 5, no. 3, pp. 310–315, 2011.

[31] L. Malihi, K. Ansari-Asl, and A. Behbahani, “Malaria parasite detection in giemsa-stained blood cell images,” *Iran. Conf. Mach. Vis. Image Process. MVIP*, no. September 2013, pp. 360–365, 2013

[32] L. Malihi, K. A. Asl, and A. Behbahani, “Improvement in Classification Accuracy Rate Using Multiple Classifier Fusion Towards Computer Vision Detection of Malaria Parasite (Plasmodium vivax),” *Jundishapur J. Heal. Sci.*, vol. 7, no. 3, 2015.

[33] M. R. B. Cells, J. A. Alkrimi, S. A. Tome, L. E. George, and J. A. Alkrimi, "Comparison of Different Classification Techniques Using Knowledge Discovery to Detect Comparison of Different Classification Techniques Using Knowledge Discovery to Detect Malaria- infected Red Blood Cells," 2019.

[34] A. Olugboja and Z. Wang, "Malaria parasite detection using different machine learning classifier," *Proc. 2017 Int. Conf. Mach. Learn. Cybern. ICMLC 2017*, vol. 1, pp. 246–250, 2017.

[35] H. M. Hussien and Y. N. Shiferaw, *Information and Communication Technology for Development for Africa*, vol. 244, no. Cdc. Springer International Publishing, 2018.

[36] A. M. Sutkar and M. N. V, "Malaria Disease Identification and Detection Using Different Classifiers," *IJCSN Int. J. Comput. Sci. Netw.*, vol. 4, no. 2, pp. 2277–5420, 2015.

[37] "Classification of Imbalanced Malaria Disease Using Naïve Bayesian Algorithm," vol. 7, pp. 786–790, 2018.

[38] 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, 2013.

[39] R. M. Haralick, I. Dinstein, and K. Shanmugam, "Textural Features for Image Classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[40] A. S. Gomashe and P. R. R Keole, "International Journal of Computer Science and Mobile Computing A Novel Approach of Color Histogram Based Image Search/Retrieval," *Int. J. Comput. Sci. Mob. Comput.*, vol. 4, no. 6, pp. 57–65, 2015.

[41] by Michael d. twa, od, MS, faao, Srinivasan parthasarathy, PhD, Cynthia Roberts, phd,ashraf m. Mahmoud, thomas w. raasch, od, PhD, faao, and mark a. bullimore,mcoptom, phd, faao "automated decision tree classification of corneal shap." Optima Vis Sci. 2005 December; 82(12): 1038–1046.

[42] "k-nearest-neighbours-introduction-to-machine-learning-algorithms". Rohith Gandhi
Jun 13, 2018

[43] m. w. Gardner*and s. r. Dorling "artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences," first received20february1997and in final form4september1997.published june1998.

[44] T. S. S, P. Karthikeyan, A. Vincent, V. Abinaya, G. Neeraja, and R. Deepika, "Random Forest Algorithm," pp. 7–12, 2016.

[45] M. F. A. Saputra, T. Widiyaningtyas, and A. P. Wibawa, "Illiteracy Classification Using K Means-Naïve Bayes Algorithm," *JOIV Int. J. Informatics Vis.*, vol. 2, no. 3, p. 153, 2018.

APPENDIX

Appendix A: Sample images of the dataset

The malaria dataset is composed of a total of 27,558 segmented cell images extracted from thin blood smear slide images. The cell images are organized into two folders, parasitized and uninfected, with 13,779 cell images in each, making this a balanced dataset. National Institute of Health (NIH) the malaria dataset is available for download from https://ceb.nlm.nih.gov/repositories/malaria-datasets. Parasitized are Implying that the region contains malaria and Uninfected there is no evidence of malaria in the region.

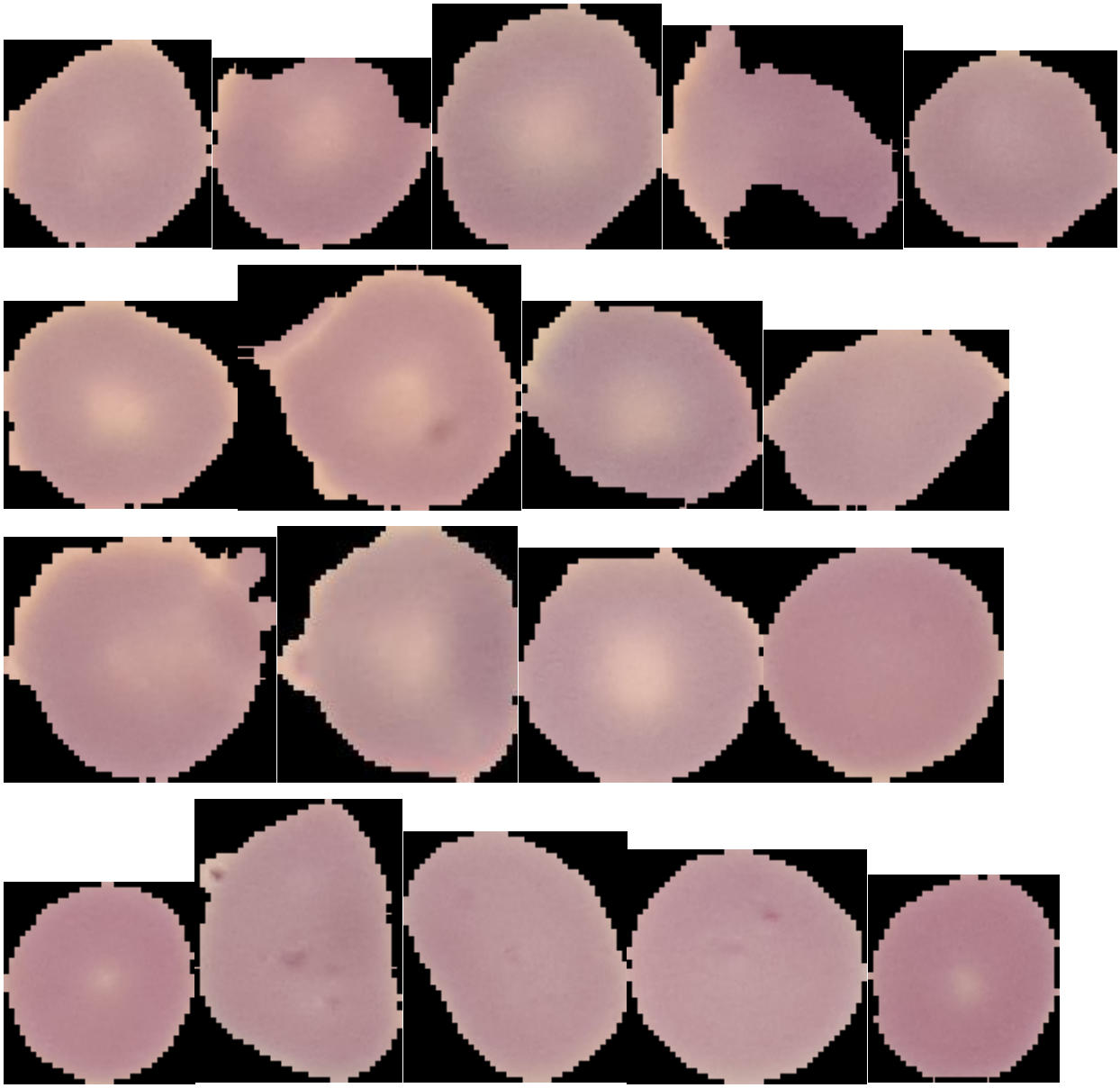


Figure Sample images of the dataset for non-infected blood smear

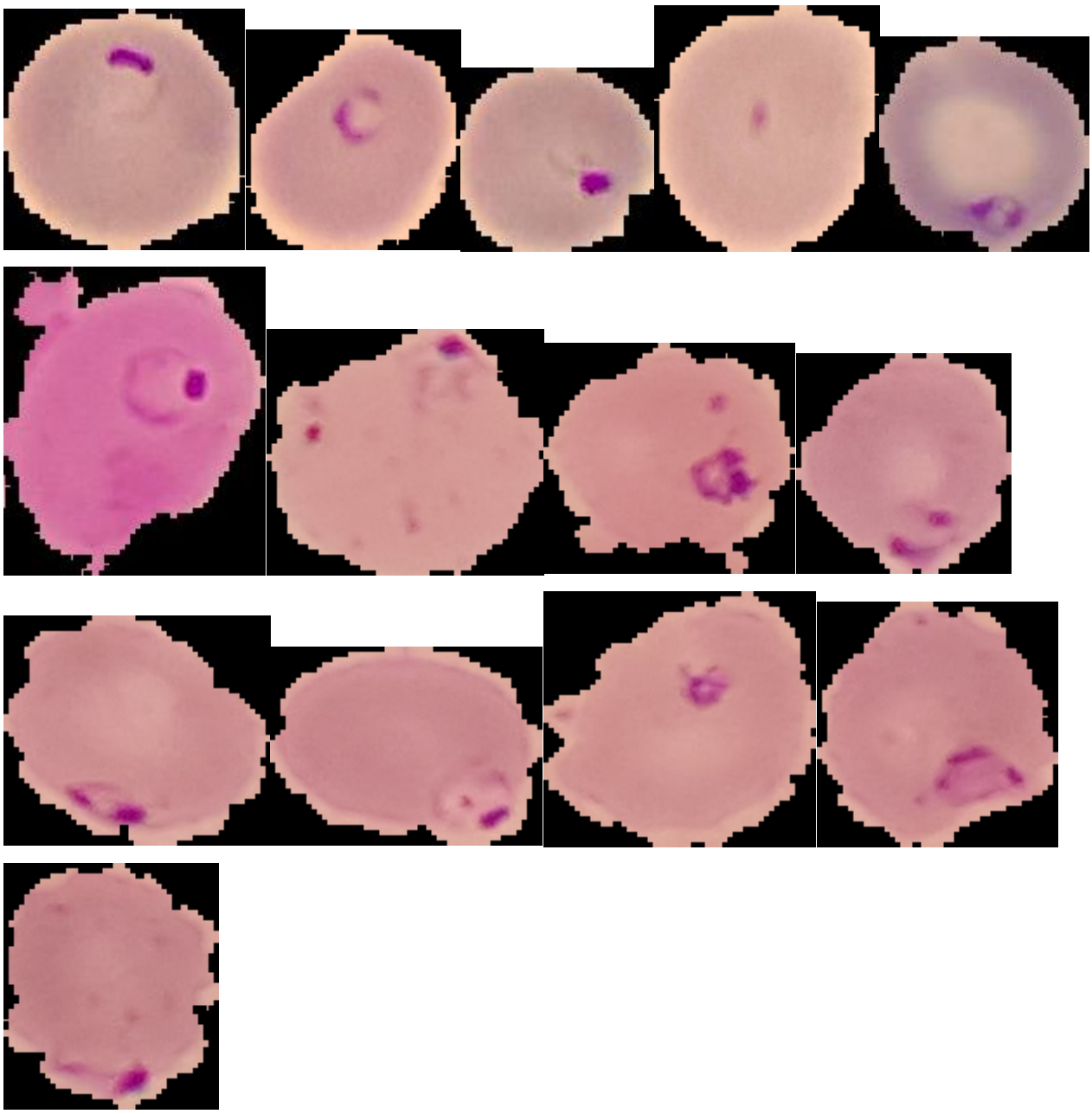
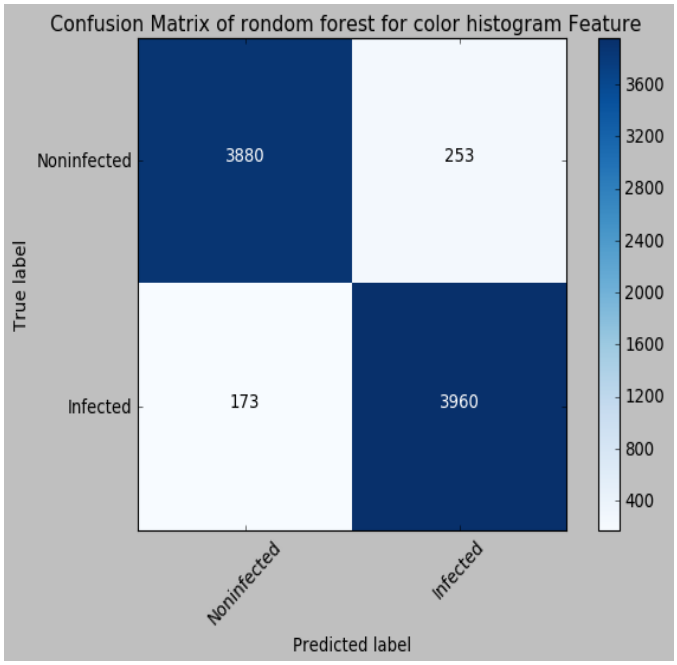


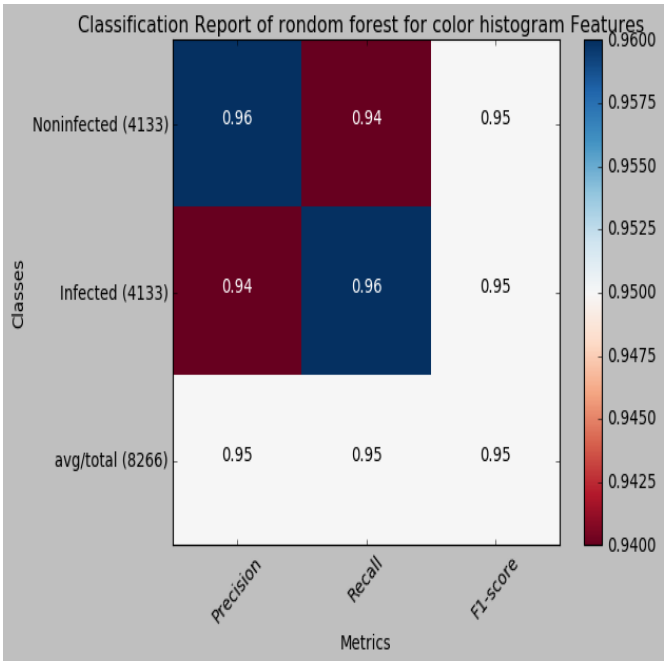
Figure Sample images of the dataset for infected blood smear

The table shows haralick features. Used which are then input to the feature Vector for some of the training images are shown below.

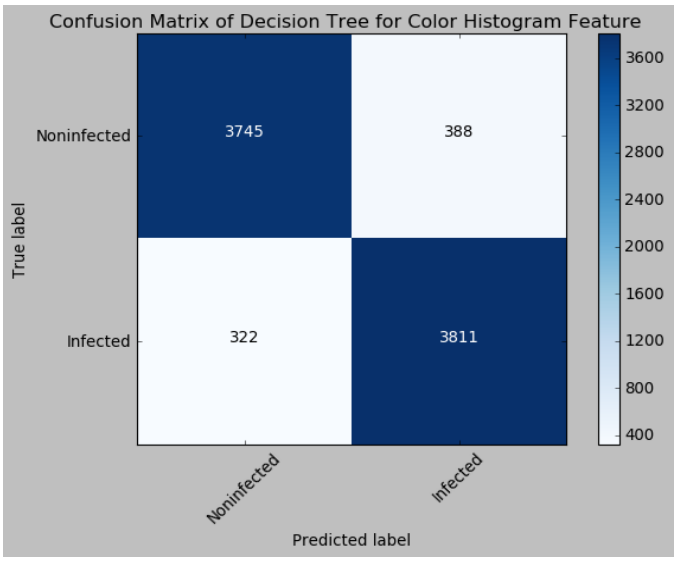
Features	Image1	Image2	Image3
Angular Second Moment	9.16002974e-02	6.87187916e-02	9.72369214e-02
Contrast	4.33527578e+02	4.29523602e+02	4.66080470e+02
Correlation	9.63831501e-01	9.63060081e-01	9.59486714e-01
Sum of Squares Variance	6.00016176e+03	5.82366459e+03	5.75889084e+03
Inverse Difference Moment (Homogeneity)	6.44959931e-01	6.19570778e-01	6.58152088e-01
Sum Average	2.29277734e+02	2.57784576e+02	2.21200813e+02
Sum Variance	2.35671194e+04	2.28651347e+04	2.25694829e+04
Sum Entropy	4.94431764e+01	4.84975800e+01	4.85721387e+01
Entropy	6.15810596e+02	6.16313948e+02	6.01500849e+02
Difference Variance	1.60517632e-03	1.46383440e-03	1.71029526e-03
Difference Entropy	2.28778557e+02	2.34389061e+02	2.36944906e+02
information Measure of Correction	-5.14769732e-01	-4.68124837e-01	-5.21360862e-01
Maximal Correction Coefficient(Energy)	9.92669893e-01	9.87745343e-01	9.92476403e-01

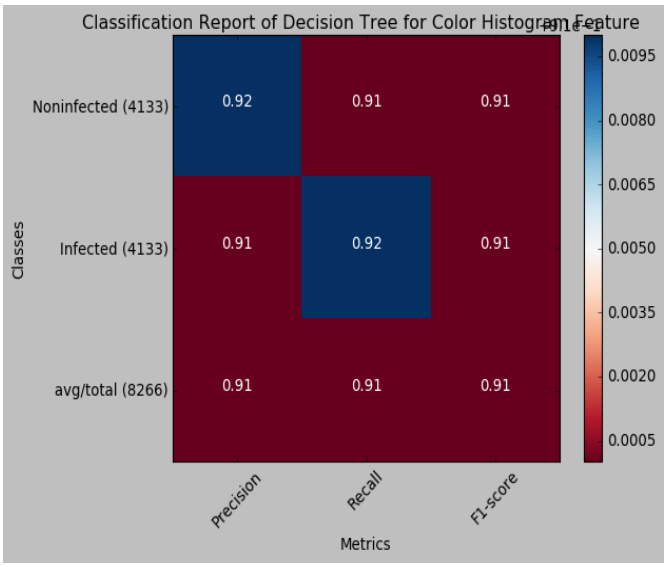
Shows the Confusion Matrix, and Classification Performance metrics value of random forest

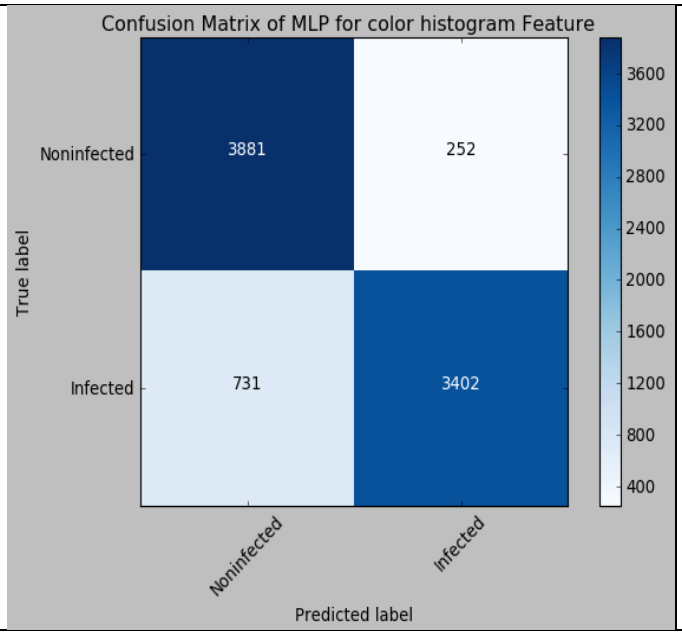


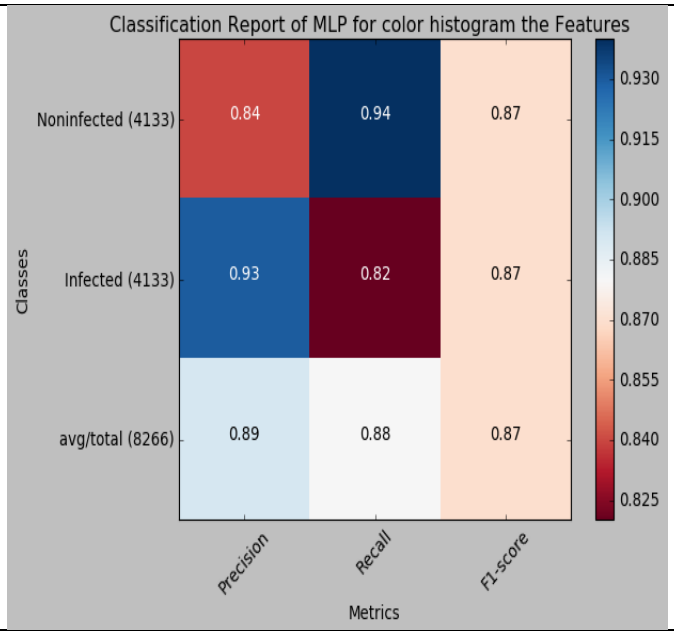


Shows the Confusion Matrix, and Classification Performance metrics value of Decision tree









Shows the Confusion Matrix, and Classification Performance metrics value of mlp