Addis Ababa University
College of Natural Sciences

Automatic Coffee Disease and Pest Damage Identification

Abnet Kifle Yimeru

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia
March, 2020

Addis Ababa University
College of Natural Sciences

Abnet Kifle Yimeru

Advisor: Dida Midekso (PhD)

This is to certify that the thesis prepared by *Abnet Kifle Yimeru*, titled: *Automatic coffee disease and pest damage identification* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor:	_____	
Examiner:	_____	
Examiner:	_____	

ABSTRACT

Coffee is a major commodity consumed as a beverage by one third of the world's population, and Ethiopia covers 7-10 % of its production. Around 15 million Ethiopian's income directly or indirectly depend on coffee production. Coffee plays a major role in the country's economic growth and stability, yet diseases (like coffee berry disease, coffee leaf rust and brown eye spot) and pests (like coffee leaf miner and green scales) affect its production in quality and yield. An early identification of these diseases and pests is very essential to control the damage caused by diseases and pest infestation. Also, developing an automatic disease and pest identification system that extracts important features from different parts of coffee plant is very crucial and yet lacking. In this research work, we proposed an automatic identification of coffee diseases and pest damages from leaf and berry parts of the plant using image processing and machine learning techniques.

We proposed a segmentation algorithm that separates healthy regions of coffee leaf/berry from damaged ones using L*a*b* color space, YCbCr color space and texture filter. We identified a total of 28 features (i.e. 22 color and 6 texture) to model the classes of coffee diseases and pest damages. Feedforward artificial neural network with backpropagation learning algorithm was used to classify coffee disease and pest types. The network is designed with 28 input and 5 output nodes.

The model is trained using 2,400 sample images of coffee leaves and berries collected from Jimma Agricultural Research Center, Tepi Agricultural Research Center, Mechara Agricultural Research Center, and Sheko and Limmu coffee farms. The training data is randomly split into 80% training and 20% testing. The classifier result was compared with results obtained from SVM and KNN classifiers. The trained classifier model reached an overall classification accuracy of 91.9%. Classification accuracy for coffee berry disease, coffee leaf rust, coffee leaf miner, brown eye spot disease and green scales is 89.6%, 91.5%, 93.3%, 96.6% and 89.0%, respectively.

Keywords: *Digital image processing, L*a*b* color space, YCbCr color space, local range filter, artificial neural network, coffee disease.*

ACKNOWLEDGMENTS

This thesis becomes a reality with the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

Foremost, I want to express my sincere gratitude to my advisor Dr. Dida Midekso for the continuous support of my research paper, for his patience, enthusiasm and motivation. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's degree study.

I would also like to thank the experts who were involved in the validation survey for this research paper:

- Mr. Eyobe Ashenafi Plant pathologist at Mizan-Tepi University
- Mr. Minalu Misganaw. laboratory technician at Tepi National Spices Research Center
- Mr. Sisay Teshome plant pathologist at Jimma Agricultural Research Center

Without their passionate participation and input, the data collection and validation survey could not have been successfully conducted.

TABLE OF CONTENTS

LIST OF TABLES ..... **iv**

LIST OF FIGURES ..... **v**

LIST OF ALGORITHMS ..... **vii**

LIST OF ACRONYMS AND ABBREVIATIONS ..... **viii**

CHAPTER ONE: INTRODUCTION 1

1.1 Background 1

1.2 Motivation 2

1.3 Statement of the Problem 3

1.4 Objectives 4

1.5 Methods 5

1.6 Scope and Limitations 6

1.7 Application of Results 6

1.8. Organization of the Thesis 6

CHAPTER TWO: LITERATURE REVIEW 7

2.1 Introduction 7

2.2 Coffee Plant Pests and Diseases 7

2.2.1 Coffee Plant Pests 8

2.2.2 Coffee Plant Diseases 11

2.3 Digital Image Processing 15

2.4 Fundamental Steps of Digital Image Processing 16

2.4.1 Image Acquisition 16

2.4.2 Image Preprocessing 16

2.4.3 Image Segmentation 17
2.4.4 Feature Extraction 20
2.4.5 Classification 21
CHAPTER THREE: RELATED WORK 23
3.1 Introduction 23
3.2 Plant leaf disease identification 23
3.3 Summary 26
CHAPTER FOUR: DESIGN OF COFFEE DISEASE AND PEST DAMAGE IDENTIFICATION 27
4.1 Introduction 27
4.2 System Architecture 27
4.3 Preprocessing 29
4.4 Segmentation 30
4.4.1 Edge Segmentation 31
4.4.2 Disease and Pest Damage Segmentation 33
4.4.3 Merging and Masking 36
4.5 Feature Extraction 39
4.5.1 Color Feature Extraction 39
4.5.2 Texture Feature Extraction 41
4.6 Identification 42
4.6.1 Supervised Learning 42
4.6.2 Prediction 43
4.7 Summary 45

CHAPTER FIVE: EXPERIMENT AND RESULTS 46

5.1 Introduction 46

5.2 Data Collection and Dataset Preparation 46

5.3 Implementation 47

5.4 Test Results 49

5.4.1 Support Vector Machine (SVM) Classifier Test Result 49

5.4.2 K-nearest Neighbor (KNN) Classifier Test Result 50

5.4.3 Artificial Neural Network (ANN) Classifier Test Result 51

5.5 Discussion 55

5.6 Summary 57

CHAPTER SIX: CONCLUSION AND FUTURE WORKS 58

6.1 Conclusion 58

6.2 Contribution to Knowledge 59

6.3 Future Work 59

REFERENCES 61

ANNEXES 66

ANNEX A: System code for main components 66

LIST OF TABLES

Table 5.1: Dataset description..... 48

Table 5.2: Confusion matrix of SVM classifier accuracy..... 49

Table 5.3: Confusion matrix of KNN classifier accuracy..... 50

Table 5.4: Confusion matrix of the overall classification accuracy of ANN classifier 53

Table 5.5: Confusion matrix for classification accuracy obtained while training with color features 54

Table 5.6: Confusion matrix for classification accuracy obtained while training model with texture features..... 55

LIST OF FIGURES

Figure 1.1: Sample coffee plant affected by pest and diseases 3

Figure 2.1: Plant Disease Pyramid 7

Figure 2.2: Sample image of a healthy coffee leaf 8

Figure 2.3: Sample image of coffee leaf infested with coffee leaf miner pest 9

Figure 2.4: Sample image of coffee berry infested with green scales 10

Figure 2.5: Sample image of coffee leaf infected with coffee leaf rust disease 11

Figure 2.6: Sample image of coffee berry infected with coffee berry disease 13

Figure 2.7: Sample image of coffee leaf infected with brown eye spot disease 14

Figure 4.1: Proposed System Architecture 28

Figure 4.2: (a) Scaled RGB image (b) Y Channel (c) Cb Channel (d) Cr Channel (e) Mask (f) Masked RGB image 30

Figure 4.3: (a) Masked RGB image (b) Binary Image (c) holes filled binary image 31

Figure 4.4: (a) Masked RGB image (b) Binary Image (c) holes filled binary image 31

Figure 4.5: (a) local range filtered image (b) dilated binary image (c) Small regions removed image 32

Figure 4.6: (a) (b) Masked RGB images (c) (d) Intensity adjusted images 33

Figure 4.7: (a) green pixels masked image leaf (b) green pixels masked image berry 34

Figure 4.8: (a) output of histogram thresholding on YCbCr color space leaf (b) output of histogram thresholding on YCbCr color space berry 35

Figure 4.9: (a) Preprocessed RGB image (b) disease and pest damage segmentation result 36

Figure 4.10: (a) binary image of segmented disease and pest damage (b) segmented edge binary image (c) result of binary subtracted image 37

Figure 4.11: (a) smoothed binary image (b) the final segmented masked RGB Image 38

Figure 4.12: Design of ANN used for the classfication of coffee plant disease and pest damage.45

Figure 5.1 Screenshot of the developed prototype for preprocessing, segmentation feature extraction and classification..... 48

Figure 5.2: Performance of the trained ANN classifier..... 51

Figure 5.3: ROC curve for training phase, validation phase and Test phase..... 52

Figure 5.4: Bar chart of discriminative power of color and texture feature 56

Figure 5.5: Bar chart for Performance analysis of classifiers..... 57

LIST OF ALGORITHMS

Algorithm 4.1: Input image preprocessing 29

Algorithm 4.2: Preprocessed image Segmentation 31

Algorithm 4.3: Edge Segmentation 32

Algorithm 4.4: Disease and Pest damage segmentation 35

Algorithm 4.5: Merging and Masking 38

Algorithm 4.6: Color feature extraction 40

Algorithm 4.7: Texture feature extraction 41

LIST OF ACRONYMS AND ABBREVIATIONS

ANN: Artificial Neural Network

BES: Brown Eye Spot

CBD: Coffee Berry Disease

CLM: Coffee Leaf Miner

CLR: Coffee Leaf Rust

GS: Green Scales

KNN: K-Nearest Neighbor

RGB: Red, Green and Blue

SVM: Support Vector Machine

CHAPTER ONE: INTRODUCTION

1.1 Background

Coffee is one of the most important traded commodities in the world [1]. It is currently consumed by more than one third of the world's population as a beverage [2]. Coffee is a major commodity on the global market and provides a source of revenue for many millions of people concerned with cultivation, marketing, and processing of the crop. More than 50 developing countries are involved in coffee production but the consumers are at distant mostly industrialized countries [2].

Ethiopia is the largest producer of coffee in Sub-Saharan Africa and is the fifth largest coffee producer in the world next to Brazil, Vietnam, Colombia, and Indonesia, contributing about 7-10% of total world coffee production [1]. Around 15 million people directly or indirectly deriving their livelihoods from a coffee system in Ethiopia [2]. Being the origin of coffee Arabica, Ethiopia has immense potential to offer the world market a wide variety of flavors of organically produced, washed and sun-dried coffee. More than 90% of the nation's harvested coffee is organically produced. The inherently superior quality of Ethiopian highland Arabica is unmatched, particularly in flavor and aroma [3].

The coffee crop has been contributing a lion's share to the Ethiopian economy. Despite its economic significance, it suffers from many production constraints, of which pests and diseases influence both the yield and quality of coffee [4].

Image processing has been proved to be an effective tool for analysis in various fields and applications. In the agriculture sector, parameters like quantity and quality of product are important measures from the farmers' point of view. Correct and timely identification of diseases and pests in crops is the basis for the integrated management of a farm. Generally, disease features of any plants are subjectively extracted by manual inspection. Automatic plant disease identification by machine inspection can be of great benefit to those users who have little or no information about the crop they are growing. Such users include farmers, gardeners, home makers who cannot afford the services of an expert agronomist [5]. Thus, in this thesis work are going to develop an automatic coffee disease and pest damage identification system using image processing.

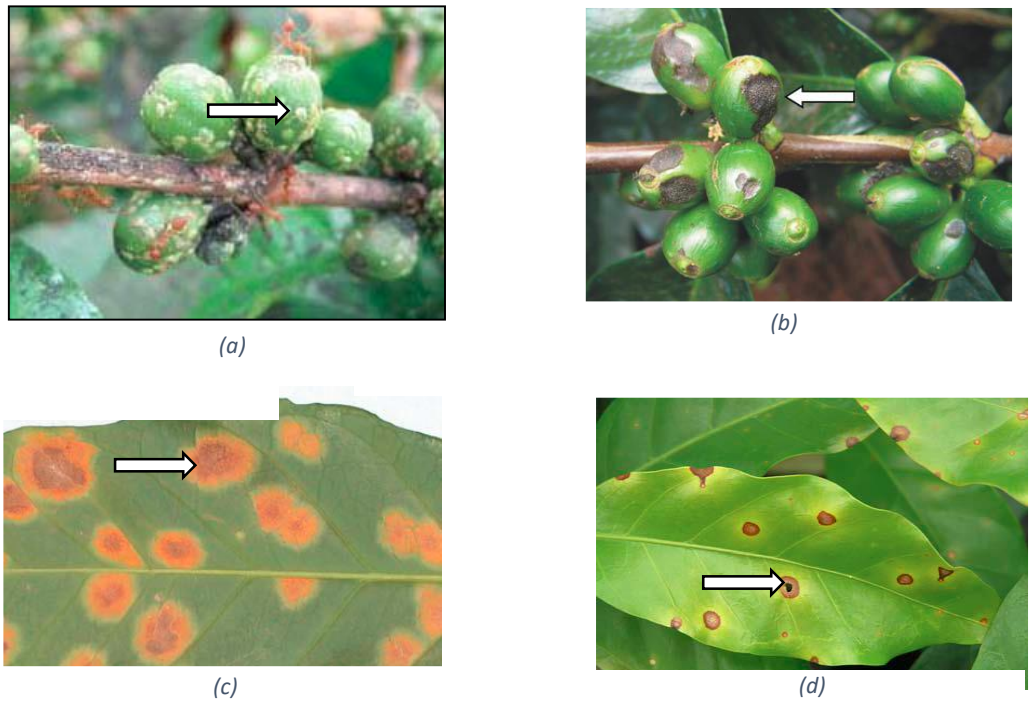
1.2 Motivation

Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security remains threatened by a number of factors including climate change [6], the decline in pollinators [7], plant diseases [8], and others. Plant diseases are not only a threat to food security on a global scale but can also have disastrous consequences for smallholder farmers whose livelihoods depend on healthy crops. In the developing world, more than 80 percent of the agricultural production is generated by smallholder farmers [9], and reports of yield loss of more than 50% due to pests and diseases are common [10].

Introduction to information and communication technologies (ICT) has an important role in the day to day life of farmers [11]. The agriculture sector contributes to the daily needs of the Ethiopian population and it is also the backbone of the country's economy. Farmers are the main elements in the agriculture sector, but they are facing complications occurring due to plant pests and diseases. Most farmers in Ethiopia do not have access to information about pests and disease types and also their identification mechanism [12]. They have to depend on plant pathologists to resolve these problems. Examining the plant affected by pests and disease through a plant pathologist manually is an expensive and time consuming process. Plants affected by a disease if not diagnosed within time will result in poor quality and low quantity of production. Thus, we are motivated to develop a system that uses a computer vision to identify coffee plant diseases and pest damages that can help coffee growing farmers in Ethiopia to use disease and pest control mechanisms effectively.

1.3 Statement of the Problem

Coffee production is constrained by many factors, including losses due to damage caused by pests and diseases, poor management practices, soil infertility, and poor pricing. The poor price of coffee is also associated with the damage caused by pests, particularly insects that are proliferated by climate change [13]. More than 45 insect pests were reported in coffee fields from Ethiopia [13]. Given the perennial nature of coffee, some pests and diseases are able to survive and multiply throughout the cropping season and are always present on the coffee crop, although their populations and hence their effect on the crop may vary throughout the year. Others may only visit and attack coffee during periods when conditions are favorable. Either way, the damage they cause and their impact on crop yield and quality can be considerable. White coffee stem borer, coffee wilt disease, parasitic nematodes, and root mealy bug, for example, can and often do kill a coffee plant. Coffee berry borer, green scales, leaf rust, and brown eye spot, however, will have more of a debilitating effect on plant growth, by causing defoliation for example, and can seriously affect berry quality [14]. Figure 1.1 depicts a sample of coffee plants affected by pests and diseases.



(a) *(b)* *(c)* *(d)*

Figure 1.1: (a) Green Scales (b) Coffee berry disease (c) Coffee leaf rust (d) Brown eye spot disease

There are previous studies on plant disease recognition and identification techniques using image processing, like the work done by Gethaun Tigistu [15] on flower disease identification and Abrham Debasu *et al.* [16] on coffee disease identification. The previous work on coffee plants focuses on the identification of coffee wilt, coffee berry, and coffee leaf rust diseases. However, the proposed method identifies the disease only from the leaf of the plant, and also the method works properly when leaves of the coffee plant are at a certain stage of growth. The other issue is the operating environment for capturing images of leaves is strict. In order for the method to work properly, images of leaves need to be captured from a specific distance and also needs a light bulb. In this research work, we develop an automatic coffee disease and pest damage identification method that considers the leaf and berry parts of the coffee plant by using image processing techniques.

1.4 Objectives

General Objective

The general objective of this research is to develop an automatic coffee disease and pest damage identification system using image processing.

Specific Objectives

In order to achieve the general objective, the following specific objectives are set:

- Review works that are related to plant disease and pest damage identification mechanisms.
- Study and explore types of coffee diseases, pests and the damage they cause at the different development stages of the plant.
- Select an appropriate methodology to analyze the images of the coffee plant.
- Design the architecture of the proposed system.
- Design algorithms that are best suited for image segmentation and feature extraction processes.
- Develop a prototype and implement it on smartphones.
- Evaluate the performance of the developed system.

1.5 Methods

Under this section, different methods of literature review, data collection, prototype development and evaluation are discussed. These techniques are used to identify the concepts that facilitate solutions to coffee pests and diseases identification, and then evaluate the solutions through prototype development.

Literature Review

Different kinds of literature related to coffee plant diseases, pests, image processing, and machine learning are reviewed to better understand the subject matter. This helps us to summarize related works that are relevant to this research.

We conduct an interview with experts in the field of plant science (i.e. plant pathologist specialized in coffee pest and disease) to clearly understand the types of diseases and pests that affect coffee plants in Ethiopia.

Data Collection

We collect sample coffee plant leaves and berries from different coffee growing regions of the country. Then the collected sample leaves and berries were captured using a digital camera.

Prototype Development

In order to experiment the proposed solution, development of a prototype for machine learning and digital image processing are conducted to solve the underlining problems. MATLAB, android and JAVA programing are used to develop the prototype.

Evaluation

The proposed research work is evaluated using relevant metrics and analysis. After the prototype is developed, we inserted sample images of coffee leaf and berry to evaluate the classification accuracy using confusion matrix measurements.

1.6 Scope and Limitations

The scope of this research is to develop automatic coffee plant disease and pest damage identification using image processing for pests and diseases that are common in Ethiopia, like coffee berry disease, green scales, coffee leaf rust, coffee leaf miner and brown eye spot.

Most pests and diseases of plants show signs and symptoms on the surface of the leaf or berry that can be captured using image processing techniques; but in some cases, pests and diseases can show symptoms that cannot be identified only by visual aid. Thus, this research is limited to identifying pests and diseases of a coffee plant with signs and symptoms that can be extracted by image processing techniques. Other limitation of this research work is, the background subtraction algorithm we used may present a poor result for images that are captured in natural background. The reason for this is, while collecting sample images we used a solid background to help us clearly segment leaf and berry features.

1.7 Application of Results

The development of such a system has a great application in the early stage of coffee plant disease and pest damage identification. Farmers can take a precaution to control pests and diseases since the proposed system uses a real time approach. It can also minimize farmer's cost for experts in areas of coffee disease and pest identification. It can help farmers in remote areas who have no access to expert knowledge.

1.8. Organization of the Thesis

The rest of the thesis is organized into five chapters. In Chapter Two, the literature is reviewed. In Chapter Three, image processing works related to plant disease identification are reviewed. Chapter Four discusses the design of the proposed solution. In Chapter Five, the experiments used to evaluate the performance of the proposed solution are discussed. In Chapter Six conclusions, future work and the contributions of this research work are presented.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter focuses on the literatures of coffee plant diseases, pests, image processing, machine learning and its approaches for real world applications. In digital image processing, each and every system of machine vision starts from image acquisition. After the images are captured there are a number of processes that we follow to reach the desired goal of a machine vision system. In this chapter, we explain the science of a machine vision system.

2.2 Coffee Plant Pests and Diseases

A disease is typically defined as abnormal growth and/or dysfunction of a plant. Diseases are the results of some disturbance within the normal life process of the plant. Diseases could also be the results of living and/or non-living causes. Biotic diseases are caused by living organisms (e.g., fungi, bacteria, and viruses). Abiotic diseases are caused by non-living environmental conditions, (e.g., soil compaction, wind, frost, soil salt damage, and girdling roots) [17] [18].

Specific conditions must be present for the biotic disease to develop. There must be a susceptible host plant, the pathogen (fungi, bacteria, viruses, etc.), and environmental conditions conducive to disease development; these must come together at a given point in time. These conditions make up what is called the ***Plant Disease Pyramid*** as shown in Figure 2.1 [19]. The biotic disease cannot occur if one of these pieces is missing.

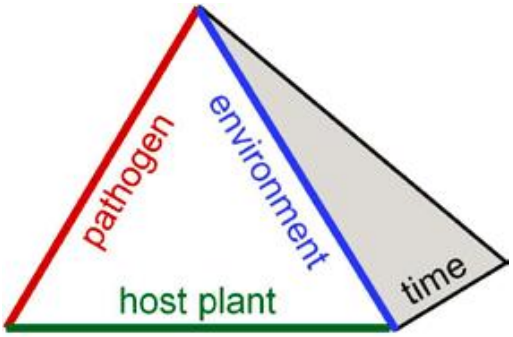


Figure 2.1: Plant Disease Pyramid

Coffee growing farmers in Ethiopia are continuously threatened by a range of problems caused by pests and diseases. Many of the pests and diseases are minor in terms of the damage they cause and their effect on yield and quality. However, diseases such as coffee berry disease, coffee leaf rust and coffee wilt disease (tracheomycosis), can be very serious indeed and can have a major impact not only on individual farmers but on the economy of countries or regions heavily dependent on coffee for foreign exchange earnings [20]. Figure 2.2 shows a healthy coffee leaf.

Figure 2.2: Sample image of a healthy coffee leaf

2.2.1 Coffee Plant Pests

Some of the major pests that affect the coffee plant in Ethiopia are coffee berry borer (CBB), green scales (GS) and coffee leaf miner (CLM) [14].

Coffee leaf miner

Coffee leaf-miner, Leucoptera caffeina Washbourn (Lepidoptera: Lyonetiidae) may be a major pest of arabica coffee especially in eastern Africa from Ethiopia to South Africa. Commonly occurring in shaded coffee, the larvae (caterpillars) produce brown irregular, blotch-like brown lines on the side of fully hardened leaves. Heavy infestation of leaf-miner was observed in several coffee growing areas of the country, causing heavy fall of leaves during a severe infestation [14]. Studies conducted at Agro Sub-Center, southwestern Ethiopia

indicated that percentage leaf damage due by this insect ranged from 2.2-55, with an average infestation of 13%. Moreover, it has also been observed to damage coffee seedlings. Recent studies showed average leaf infestation of 4.4%; 20 and 7.4 numbers of eggs and larvae, respectively per artificially infested coffee seedlings [18]. Figure 2.3 depicts a coffee leaf infested with coffee leaf miner.

Figure 2.3: Sample image of coffee leaf infested with coffee leaf miner pest

Green scales

Green scales are small insects that prey on plants by sucking the sap. In doing so, they deplete the plant of important nutrients required for growth and should cause it to become weak. While feeding, scales also inject saliva that's toxic to the plant and should cause discoloration and deformation of leaves and other plant parts. While variety of scale insects are hooked in to coffee as a source of nourishment, only a couple of those will cause noticeable damage [21] [14]. Figure 2.4 depicts a coffee berry infested with green scales.

Figure 2.4: Sample image of coffee berry infested with green scales

The nymph and adult, winged females are the stages of the scale insect's life cycle that are usually observed on a plant and are capable of causing damage. They excrete sap on the surface of the plant as clear, sugary and sticky 'honeydew'. A black, sooty fungal mold will often grow on the surface of the honeydew. Scales are also often associated with ants that feed on the excreted honeydew and, in return, protect the scales from predators. The presence of a black, soot-like coating on coffee leaves and other parts, and/or ants running about on plant surfaces, are therefore clear indications of infestation by scales [14, 21].

2.2.2 Coffee Plant Diseases

Some of the major diseases that affect the coffee plant in Ethiopia are coffee leaf rust, coffee berry disease, and brown eye spot disease [21].

Coffee leaf rust

Coffee leaf rust (CLR), caused by the fungus and affects C. arabica (arabica coffee), C. canephora (robusta coffee) and C. liberica (Liberian coffee tree). It is the foremost important disease of arabica coffee, the foremost susceptible of the three coffee species, and affects both yield quality and quantity. CLR first appeared in Sri Lanka within the 1860s but has since spread to most coffee growing areas of the planet and is endemic to the main producing countries. It is present in every coffee growing country in Africa. [14, 21].

Crop losses are due to premature shedding of leaves (defoliation) that occur following infection by the fungus. Defoliation, by reducing the growth potential of the plant, restricts the growth of new stems on which the following season's crop is to be produced. As such a disease outbreaks in one year directly affects cropping the following year, and the disease has a gradual (and often unnoticeable) weakening effect on the plant over successive seasons. High disease levels can also affect the current season’s crop, resulting in defoliation, premature ripening of berries and production of poor-quality, 'light' coffee beans [14, 21]. Figure 2.5 depicts a coffee leaf infected by coffee leaf rust.

Figure 2.5: Sample image of coffee leaf infected with coffee leaf rust disease

CLR is recognized by the development of yellow or blotchy, orange powdery pustules, or lesions, on the underside of coffee leaves, with corresponding chlorotic patches on the upper side. Initially, these are only 2-3 mm in diameter but, as the disease develops, they enlarge and can eventually become several centimeters in diameter. The pustules are made up of numerous small, yellow, powdery spores of the fungus called urediniospores. Young lesions may appear as small chlorotic spots before the spores are produced. As they become older their center becomes necrotic (dead) and the spores are confined to the outer areas. On older leaves, several lesions may also merge to produce irregularly shaped lesions that cover much of the leaf surface. However, diseased leaves are usually shed before this stage is reached [14, 21].

Coffee berry disease

Coffee berry disease (CBD), which affects Coffee arabica, is caused by a fungus. It is endemic to Africa and was first recorded in western Kenya in 1922. Since then the disease has spread to most arabica growing countries on the continent including Ethiopia. CBD has not been observed outside Africa [14, 21].

CBD can cause considerable yield losses of up to 75% when not adequately controlled. Even though the application of fungicides to control the disease can result in yields being doubled, losses of up to 30% can still occur when the attack is severe. Yield losses occur as a result of shedding and/or destruction of infected berries, which become mummified (dry, wrinkled and decayed, with hard skin). The pathogen can infect berries at the pinhead, expanding, mature green and ripe stages of development, as well as flowers. However, most losses are due to infection of the green expanding berries, between four and six weeks after flowering, when they are most susceptible, and also under wet conditions. Crop losses may, therefore, vary from year to year depending on weather conditions [21] [14]. Figure 2.6 depicts a coffee berry infected by coffee berry disease.

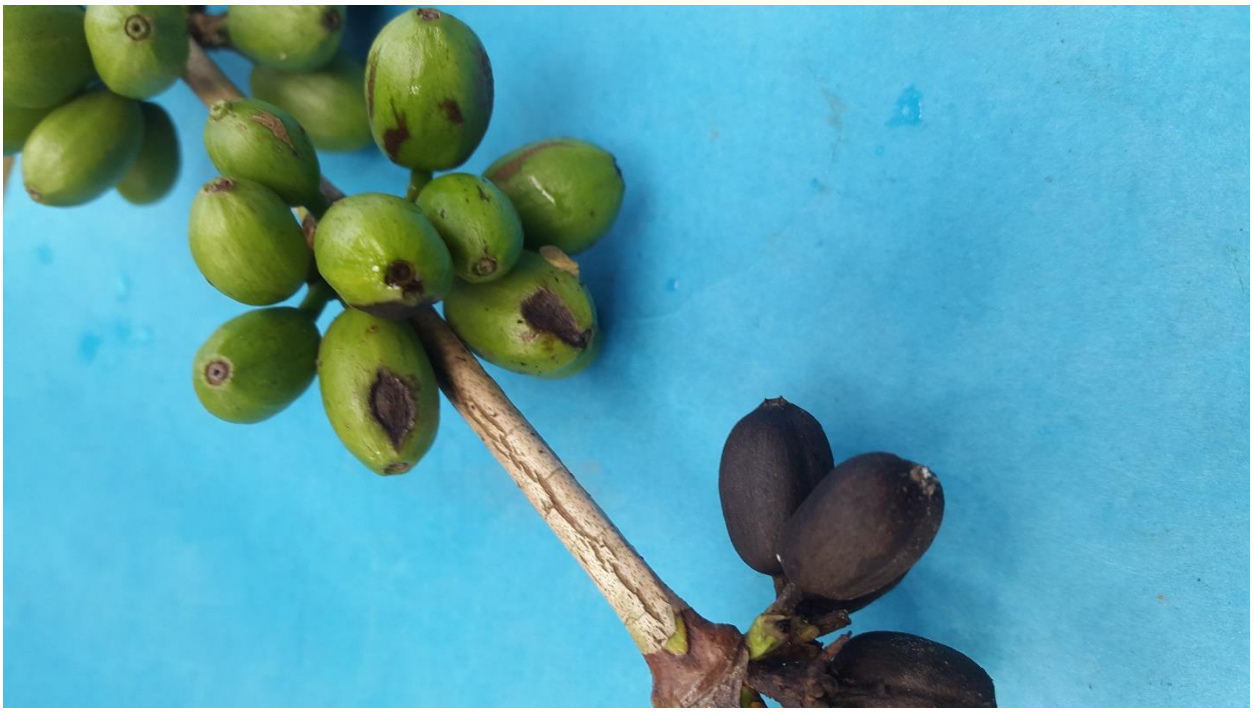


Figure 2.6: Sample image of coffee berry infected with coffee berry disease

The characteristic symptom of CBD is the development of small, water-soaked lesions on young, expanding berries that rapidly become dark brown or black and slightly sunken. They enlarge to cover the whole berry within about a week, which eventually rots. Under humid conditions, masses of pale pink spores become visible on the surface of lesions. These symptoms, collectively known as 'anthracnose', are typical of CBD. Another characteristic of the disease is the shedding of berries from branches at an early stage of disease development. Lesions may also occur on young berry stalks, causing them to be shed before lesions appear on the berry itself. CBD can also cause a 'brown blight' on ripening berries, denoted by dark, sunken lesions typical of the anthracnose enveloping the red berry. Under very wet weather conditions CBD may also cause brown lesions to develop on flower petals [21] [14].

Brown eye spot disease

Brown eye spot (BES), also mentioned as coffee leaf spot, coffee eye spot, brown eye spot, berry blotch or berry spot disease of coffee, is caused by a fungus. It flourishes in warm, humid conditions and occurs in all of the coffee growing areas of the planet. Although it's the foremost widespread disease of coffea arabica, brown eye spot rarely causes yield losses on this coffee species but can cause defoliation in, and reduce the vigor of, seedlings and young plants [21].

The infection of coffee by *C. coffeicola* can lead to a number of conditions. On *C. arabica,* it is first recognized by the appearance of small chlorotic spots on the leaves. These are usually more visible on the upper side of the leaf and gradually enlarge to become reddish-brown or brown with a greyish center. The spots are circular or perhaps angular, may also develop a yellow margin (‘halo’) and are delimited by the main veins of the leaf. As the fungus begins to sporulate (produce spores) small olive green patches may develop in the grey center of the spots. Small red spots may also develop on both green and ripening berries. The spots on the berries increase in size and coalesce to form dead blisters [14]. Figure 2.7 depicts a coffee leaf infected by brown eye spot.

Figure 2.7: Sample image of coffee leaf infected with brown eye spot disease

2.3 Digital Image Processing

In imaging science, image processing is processing of images using mathematical operations by using any sort of signal processing that the input is a picture, a series of images or a video, like a photograph or video frame; the output of image processing could also be either a picture or a group of characteristics or parameters associated with the image. Most image-processing techniques involve isolating the individual color planes of a picture and treating them as a two-dimensional signal and applying standard signal-processing techniques to them. Images also are processed as three-dimensional signals with the dimension being time or the z-axis [22] [23].

Image processing usually refers to digital image processing, but optical and analog image processing are also possible. For an extended time, employing a computer to control a digital image (i.e., digital image processing) was something performed by only a comparatively small group of specialists who had access to expensive equipment. Now, however, the mixture of a strong computer on every desktop and therefore the incontrovertible fact that nearly everyone has some sort of device for digital image acquisition, be it their telephone camera, camera , or scanner, has resulted during a plethora of digital images and, with that, for several digital image processing has become as common as data processing . [24].

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers [25]. The goal of this manipulation can be divided into three categories:

- Image processing: *image in → image out*
- Image analysis: *image in → measurements out*
- Image understanding: *image in → high-level description out*

Image processing has multiple purposes. To improve the quality of an image in a subjective way, usually by increasing its contrast. This is called **image enhancement**. To use as few bits as possible to represent the image, with minimum deterioration in its quality. This is called **image compression**. To improve an image in an objective way, for example by reducing blurring. This is called **image restoration**. To make explicit certain characteristics of the image which can be used to identify the contents of the image. This is called **feature extraction**.

Image processing has expanded and is further rapidly expanding from a few specialized applications into a standard scientific tool. Image processing techniques are now applied to virtually all the natural sciences and technical disciplines [26].

2.4 Fundamental Steps of Digital Image Processing

Image processing is not a one-step process [23]. We are able to distinguish between several steps which must be performed one after the other until we can extract the data of interest from the observed scene

2.4.1 Image Acquisition

Image acquisition is the action of retrieving an image from some source, usually a hardware-based source, so it can be passed through whatever process needs to occur afterward. The image that is acquired is completely unprocessed and is the result of whatever hardware was used to generate it, which can be very important in some fields to have a consistent baseline from which to work. One of the ultimate goals of this process is to have a source of input that operates with in such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly reproduced under the same conditions so anomalous factors are easier to locate and eliminate [27].

2.4.2 Image Preprocessing

Image preprocessing refers to the various initial image enhancing techniques of the captured raw images. It is one among the low low-level processes in image processing. During image acquisition, we might find yourself with noisy images. There are many sources of noise in images. The images captured should first be transferred onto a computer and be converted to a digital image. Digital images, though displayed on the screen as pictures, are digits readable by the computer and are converted to tiny dots or picture elements representing the real objects [28].

The aim of preprocessing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. A series of image preprocessing techniques precede the analysis and processing activities, to enhance image quality and avoid distortion, for valuable information retrieval [29]. Some of the preprocessing techniques that can be applied to raw images are background subtraction, contrast adjustment, intensity adjustment and scaling [39].

Background subtraction is a process of separating the foreground of an image region from its background. Background subtractions can be performed using color thresholding or other edge based region segmentation techniques [31]. Intensity adjustment is an image enhancement technique that maps the image intensity values to a new range. The low-contrast image with its histogram and all the values gathered in the center of the intensity range [31].

The scale operator performs a geometric transformation that can be used to shrink or zoom the size of an image (or part of an image). Image reduction, commonly known as *subsampling*, is performed by replacement (of a group of pixel values by one arbitrarily chosen pixel value from within this group) or by *interpolating* between pixel values in local neighborhoods. Image zooming is achieved by *pixel replication* or by interpolation. Scaling is used to change the visual appearance of an image, to alter the quantity of information stored in a scene representation, or as a low-level preprocessor in a multi-stage image processing chain that operates on features of a particular scale. Scaling is a special case of affine transformation [32].

2.4.3 Image Segmentation

Image segmentation is the process of dividing an image into regions or objects. It is a mid-level step in image processing. It is the first step in the task of image analysis. Image segmentation is an essential component of an image analysis technique that determines the quality of the final outputs [33]. Image segmentation enables one to discriminate objects from the background, into non-overlapping sets. It also subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being viewed. Segmentation involves the partitioning of an image into a set of homogeneous and meaningful regions, such that the pixels in each partitioned region possess an identical set of properties or attributes. These sets of properties of the image may include gray levels, contrast, spectral values, or textural properties [24].

Several different segmentation techniques are available, but there is not a single method that can be considered good for different images. All techniques are not equally good for a particular type of image. Hence, algorithm development for one class of images may not always be applied to other classes of images. As a result, there are many challenging issues like the development of a unified approach to image segmentation which can be applied to all types of images. Often, the selection of an appropriate technique for a specific type of image is a difficult problem. Thus, in spite of several decades of research, there is no universally accepted method for image

segmentation. Therefore, this problem remains a challenging issue in the field of image processing and computer vision [33].

The choice of one segmentation technique over another and the level of segmentation are decided by the particular type of image and characteristics of the problem being considered. Basically, there are two categories of segmentation techniques, namely, edge-based and region based segmentation. They are based on the detection of discontinuities and detection of similarities respectively. The segmentation technique based on the detection of discontinuities is the process of partitioning an image based on abrupt changes in intensity. Examples of such algorithms include all edge detection algorithms [34].

Edge detection techniques are generally used for finding discontinuities in gray level images. Edges are local changes in the image intensity and they typically occur on the boundary between two regions. Spatial masks can be used to detect all types of discontinuities in an image. These spatial masks are commonly referred to as edge detection operators. These spatial masks are grouped into two classes. The first group is called the first order derivative group and includes the Prewitt, Sobel, Canny and Test Operators. Canny is the most promising one but takes more time as compared to the Sobel operator [34].

The second group is known as the second order derivative group. This group includes Laplacian and Zero-crossings operators. Edge-based segmentation is based on the fact that the position of an edge is given by an extreme of the first -order derivative or a zero crossing in the second-order derivative.

On the contrary, region based methods are based on continuities. These techniques divide the entire image into sub regions depending on some similarity rules (criteria). Examples of algorithms that work based on the detection of similarity include thresholding and region growing. Whether the segmentation technique used is discontinuity or region based, the end result of any segmentation process is a binary image [23]. Some of the well-established segmentation techniques are thresholding and edge detection.

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in the application of image segmentation. In this method, the pixels are partitioned depending on their intensity value. By selecting an adequate threshold value T, the gray level

image can be converted to a binary image. The binary image should contain all of the essential information about the position and shape of the objects of interest [15].

Segmentation methods based on discontinuity search for abrupt changes in the intensity value are called edge or boundary based method [22]. The edges identified by edge detection are often disconnected. To segment an object from an image, one needs closed region boundaries. The desired edges are the boundaries between such objects. These segmentation methods detect discontinuities and produce a binary image contained edges and their background as the output of the image. It is also assumed that since it is the boundary of a region then it is closed and that the number of interesting objects is equal to the number of boundaries in an image. There is an exceptionally large number of edge detection methods available, each of which is designed to be sensitive to certain edge types because of some variables like edge orientation, noise environment and edge structure [15].

Recently, there are a large number of color image segmentation techniques based on different color space (i.e. RGB, YCbCr, L*a*b*). Different kinds of segmentation techniques are introduced and several results are discovered under color images. The RGB color pattern encodes colors as combinations of the three essential colors: red (R), green (G), and blue (B). This conspires is broadly utilized for transmission, representation, and capacity of color pictures on both analog gadgets such as TV sets and digital gadgets such as computers, advanced cameras, and scanners [35].

For this reason, numerous image-processing and design programs utilize the RGB patterns their inside representation for color pictures, and most dialect libraries, counting Java's imaging APIs, utilize it as their standard picture representation. The RGB color space can be visualized as a 3D unit cube in which the three primary colors form the coordinate axis. The RGB values are positive and lie in the range [0, 255] for most digital images [35].

YCbCr is another color space like RGB. But a bit of different spaces are taken. Here Y is the luminance component of the color. The luminance component is the brightness of the color, which means the light intensity of the color. The human eye is more sensitive to this component. Cb and Cr are the blue components and red component related to the Chroma component. That means "Cb is the blue component relative to the green component. Cr is the red component relative to the green component." These components are less sensitive to human eyes. Since the Y component is

more sensitive to the human eye, it needs to be more correct and Cb and Cr are less sensitive to the human eye [33].

Lab (also called CIELAB or L*a*b) is a three dimensional color space designed to approximate human vision, as opposed to the RGB color space which models the output of physical devices rather than human visual perception. It keeps the tone (also called lightness or value) separate from the color so that you can adjust one without changing the other. The dimensions in this color space are the luminosity $L*$ and the two color components $a*$, $b*$, which specify the color hue and saturation along with the *green red* and *blue-yellow* axes, respectively [35].

2.4.4 Feature Extraction

After segmentation is done, feature extraction is the next major activity in digital image processing. It is the task of describing objects to reduce them to a form suitable for computer processing. This form is called an image feature. Image feature is a distinguishing primitive characteristic or attribute of an image [36].

Image features have major importance in image classification. There are several types of image features that have been proposed for image classification. Morphology, color, and texture are some of the basic image features [36]. These image features are used as inputs to a classifier that discriminates objects into different categories [32].

Morphological features are the geometric property of an image like shape and size. They are physical dimensional measures that characterize the appearance of an object. For instance, area and perimeter are some of the most commonly measured size features and similarly circularity measures the shape of image compactness [36].

Color is one of the most widely used features for image classification. In an image, each pixel records a numeric value that is often the brightness of the corresponding point in the image. Several such values can be combined to represent color information. The most typical range of brightness values is from 0 to 255 (8 bit range), but depending on the type of camera, scanner or another acquisition device, a large range of 10 or more bits, up to perhaps 16 (0 to 65,535) may be encountered. However, in most cases, these images are still stored with a set of discrete integer grey values because it is easier to manipulate such arrays and convert them to display [36]. In line with this, the statistical values of color features like mean, mode, standard deviation, etc. are used for image classification [33].

The texture is a repeated pattern of information or arrangement of the structure with regular intervals. In a general sense, texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, and the proportion of its elementary parts. A basic stage to collect such features through texture analysis process is called as texture feature extraction. Due to the signification of texture information, texture feature extraction is a key function in various image processing applications [22].

2.4.5 Classification

Recognition of the characteristics of objects in an image from a specific set of measured values of features of the object facilitates the stratification of an image into various classes with similar features. This is the core business of the design of classifiers, which utilizes specified features of an object as its inputs, thereby generating a classification label or value depicting the correct class allotment of the object. A set of objects with predefined labels or values- the learning set-serve as the immediate basis for this mechanism [23].

A number of pattern classification techniques have been used for the recognition of patterns. Classification methods are mainly based on two types. They are supervised learning and unsupervised learning [36]. In supervised classification, the classifier is trained with a large set of labeled training pattern samples. The term labeled pattern samples mean that the set of patterns whose class memberships are known in advance [36]. Support vector machine, k-nearest neighbor and artificial neural network are the common classifiers.

The k-nearest neighbors (KNN) algorithm is a simple machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand but has a major drawback of becoming significantly slows as the size of that data in use grows. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression) [38].

Support vector machines belong to the area of supervised learning methods and therefore need labeling, known data to classify new unseen data. The basic approach to classify the data starts by trying to create a function that splits the data points into the corresponding labels with (a) the least possible amount of errors or (b) with the largest possible margin. This is due to the fact that larger empty areas next to the splitting function result in fewer errors because the labels are better

distinguished from one another. Formally, support vector machines create one or multiple hyperplanes in an n-dimensional space. The first attempt in the process of splitting the data is always, to try to linearly separate the data into the corresponding labels [36] [39].

Artificial neural networks (ANN) are highly distributed interconnections of adaptive nonlinear processing elements. In other words, they are a large set of interconnected neurons, which execute in parallel to perform the task of learning. Hence, ANN resembles the human brain in two respects. The first property is that knowledge is acquired by the network through a learning process. The other is interneuron connection strengths known as weights are used to store the knowledge, i.e., the weights on the connections encode the knowledge of a network. The neurons are modeled after the biological neurons and hence they are termed as neural networks [15] [40].

CHAPTER THREE: RELATED WORK

3.1 Introduction

Several research efforts have been conducted and proposed for automatic plant disease identification. However, each work is designed to address different plants or parts of a plant. So, in this Chapter, related works of different researchers in area of automatic plant disease identification using machine vision system are reviewed. Research works related to plant leaf, flower and root disease identification are also reviewed.

3.2 Plant leaf disease identification

A. Meunkaewjinda *et al.* [43] designed a system for identifying grape leaf diseases from color imagery using a hybrid intelligent system. The system diagnoses plant leaf diseases without maintaining any expertise once the system is trained. The proposed system consists of grape leaf color segmentation, grape leaf disease segmentation, analysis and classification of diseases. The grape leaf color segmentation is a preprocessing module that segments out any irrelevant background information. A self-organizing feature map together with a back-propagation neural network is deployed to recognize the colors of a grape leaf. This information was used to segment grape leaf pixels with the image. Then the grape leaf disease segmentation was performed using a modified self-organizing feature map with genetic algorithms for optimization and support vector machines. Finally, the resulting segmented image was filtered by the Gabor wavelet which allows the system to analyze leaf disease color features more efficiently. The system has some limitations on extracting ambiguous color pixels from the background of the image.

K. Wang *et al.* [42] designed and developed a method for identifying vegetable disease and insect pest recognition based on computer vision and image processing methods. The investigation of the vegetable disease and insect pest state used images captured using smart phones for judgment. A new extraction and classification algorithm was used to recognize leaves from images. Then a region-labeling algorithm was applied to calculate the insect number and diseased areas in the segmented images. To deal with the areas of adhesion, a mathematical morphology algorithm was used for separating the objects. The proposed method was implemented on mobile smart devices and tested with field experiments. The experimental result shows 91.96% accuracy. The method

has limitations in accuracy in some complex cases, such as the occlusion of leaves, the color similarity of the object and the leaves, the abnormal illumination, and the bad image quality.

B. Klatt *et al.* [41] designed a system for plant disease detection via smartphones. The proposed system can identify plant diseases by analyzing digital images using local servers and smartphones. The system first down-sample the input image to increase sharpness. Then a binary image will be computed to put the reddish pixels in the foreground and the greenish/bluish pixels in the background. To compute a region image, median filtering, component analysis, hole filling, filtering of regions adjacent to the image borders, as well as filtering of regions where the width/height ratio of their bounding boxes is not in the interval [1/2, 2] is applied. The last step of the preprocessing procedure is to convert the input image to a grey scale image and suppress all pixels in the grey scale image that are background pixels in the region image; then un-sharp masking is applied. The resulting image contains all the information for the subsequent analysis on a local server. To extract the features of each region, the local binary patterns (LBPs) of the gradient magnitudes are computed and binned into a histogram to use its entropy as a descriptor. The classification of the leaf spots is accomplished by thresholding entropy measures of the texture descriptors using naive Bayes-classifiers. The classifier attaches the regions to a certain class e.g., *Cercospora beticola*, *Ramularia betae*, *Phoma betae* or a *reject* class (objects that cannot be classified). The experimental result shows 97% accuracy. The segmentation algorithm used does not differentiate leaf parts from background environment.

Abrham Debasu *et al.* [16] designed a system for Ethiopian coffee plant disease identification. The proposed system can identify coffee leaf rust, coffee wilt and coffee berry diseases from the leaf of the coffee plant. The authors used a digital camera and florescent light to acquire an image of the leaf. They used an artificial neural network, k-nearest neighbors, radial basis function, Naïve and hybrid of a self-organizing map for classification purposes. Also, for segmentation purposes, the authors used Otsu, FCM, K-means, Gaussian distribution and the combinations of K-means and Gaussian distribution. The experimental result shows 92.10% accuracy. The classifier shows a lower classification accuracy for disease features that are extracted from the berry of the palnt.

Getahun Tigistu [15] designed an image processing based system for automatic identification of flower disease. The proposed system consists of two main phases. In the first phase, normal and diseased flower images are used to create a knowledge base. During the creation of the knowledge base, images are pre-processed and segmented to identify the region of interest. Then, seven

different texture features of images are extracted using Gabor texture feature extraction. Finally, an artificial neural network is trained using seven input features extracted from the individual image and eight output vectors that represent eight different classes of disease to represent the knowledge base. In the second phase, the knowledge base is used to identify the disease of a flower. In order to create the knowledge base and to test the effectiveness of the developed system, the author used 40 flower images for each of the eight different classes of flower disease and they have a total of 320 flower images. From those images, 85% of the dataset is used for training and 15% of the data set is used for testing. The experimental result demonstrates that the proposed technique is effective for the identification of flower disease. The developed system can successfully identify the examined flower with an accuracy of 83.3%.

A. Deokar *et al.* [11] designed a system for plant disease identification using content based image retrieval techniques based on android system. The proposed system can diagnose plant diseases through content based image retrieval (CBIR) techniques and k-means algorithm. Describing the proposed system, all the images of crops affected by the disease and those crops not affected by diseases are stored in a database. CBIR uses three steps to extract the properties from the image on the basis of color, texture, shape, and properties or features that are stored in the database. Average RGB is calculated for color comparison, co-occurrence matrix is used for texture comparison. When the user uses his android phone and gets a snap of the image, the features of a query image are first extracted using CBIR. The images in the database are first grouped in clusters using k-means. Clusters are with respect to the features of images like color, texture, shape. Features that are extracted from the query image are compared with the clusters and a set of images produced. By calculating the Euclidean distance, the minimum distance is considered, a cluster containing image matching nearest to the query image are generated. The proposed method makes use of a k-means segmentation algorithm which resulted in a slow segmentation process and also does not perform well when input images include dusts and glare.

Marcelo .v *et al.* [37] designed automatic detection of nutritional deficiencies in coffee tree leaves based on shape and texture descriptors. The authors focused on identification of nutritional deficiencies of Boron (B), Calcium (Ca), Iron (Fe) and Potassium (K). For the purpose of this study the authors used 269 sample images of coffee tree leafs. In this work, Otsu method was used for segmentation purpose. Blurred shape model (BSM) and gray-level co-occurrence matrix (GLCM) are used to extract shape and texture features of segmented images. The authors used the

extracted features to train the model using KNN, naïve Bayes and neural network classifiers. After this, the authors compared classification approaches of Naïve Bayes and Neural Network classifiers on each classification parameter of shape, texture and the combination of the two. To evaluate the classification accuracy, from the total of 269 data sets 80% were used for training and the remaining 20% was used for testing. Finally, it was found that the classification performance of the neural networks classifier was better than the Naïve Bayes classifier. And the overall classification accuracy was 74%.

3.3 Summary

Even though automatic plant disease identification is a challenging task, to some extent studies are made to address the issue. The studies we reviewed so far, to some extent tried to address coffee diseases and nutrition deficiencies identification focusing on the leaf part of coffee trees. To the best of our knowledge, there are no studies done to identify coffee plant disease and pest damages that show signs and symptoms on berry parts of the plant. Thus in this study, we will address the gaps mentioned above in coffee disease and pest damage identification through image processing.

CHAPTER FOUR: DESIGN OF COFFEE DISEASE AND PEST DAMAGE IDENTIFICATION

4.1 Introduction

In this chapter, we describe the overall structure for an automatic coffee disease and pest damage identification system. The sections in this chapter present details of the system architecture components. How input images of coffee plant leaf and berry are preprocessed and segmented, feature extraction tasks, extracted features and also, disease and pest damage identification process steps are presented.

4.2 System Architecture

The proposed system architecture for coffee plant disease and pest damage identification consists of four components, namely: preprocessing, segmentation, feature extraction, and identification. The system architecture for the proposed model is diagrammatically depicted in Figure 4.1. The first component of the proposed model is preprocessing. This component performs preprocessing tasks including image enhancement and background subtraction. In addition to this, the component does the initial task of making the input image ready for the segmentation component.

The second component of our proposed architecture carries out the task of separating the damaged part of the coffee plant leaf/berry from the healthy one. This segmentation component includes three sub-components, namely: edge segmentation, disease, and pest damage segmentation and finally merging and masking. Edge segmentation component extracts only the edge of coffee leaf/berry. This can help us identify the boundary of the leaf/berry. The damaged part segmentation component extracts only diseased and pest damaged regions of input coffee leaf/berry image. The third component of the segmentation algorithm merges the extracted edge with the segmented diseased/pest damaged region of the plant. The merged image is then mapped onto the original RGB image, resulting in a segmented masked RGB image. The segmented image contains all the necessary features needed for analysis and feature extraction.

The third component of the proposed architecture is feature extraction. This component applies color feature extraction and texture feature extraction techniques on the segmented image. The last component of the proposed architecture is identification. This component includes, feature set

preparation, supervised learning, detection, and prediction of the type of disease and pest damage in a given input coffee plant image.

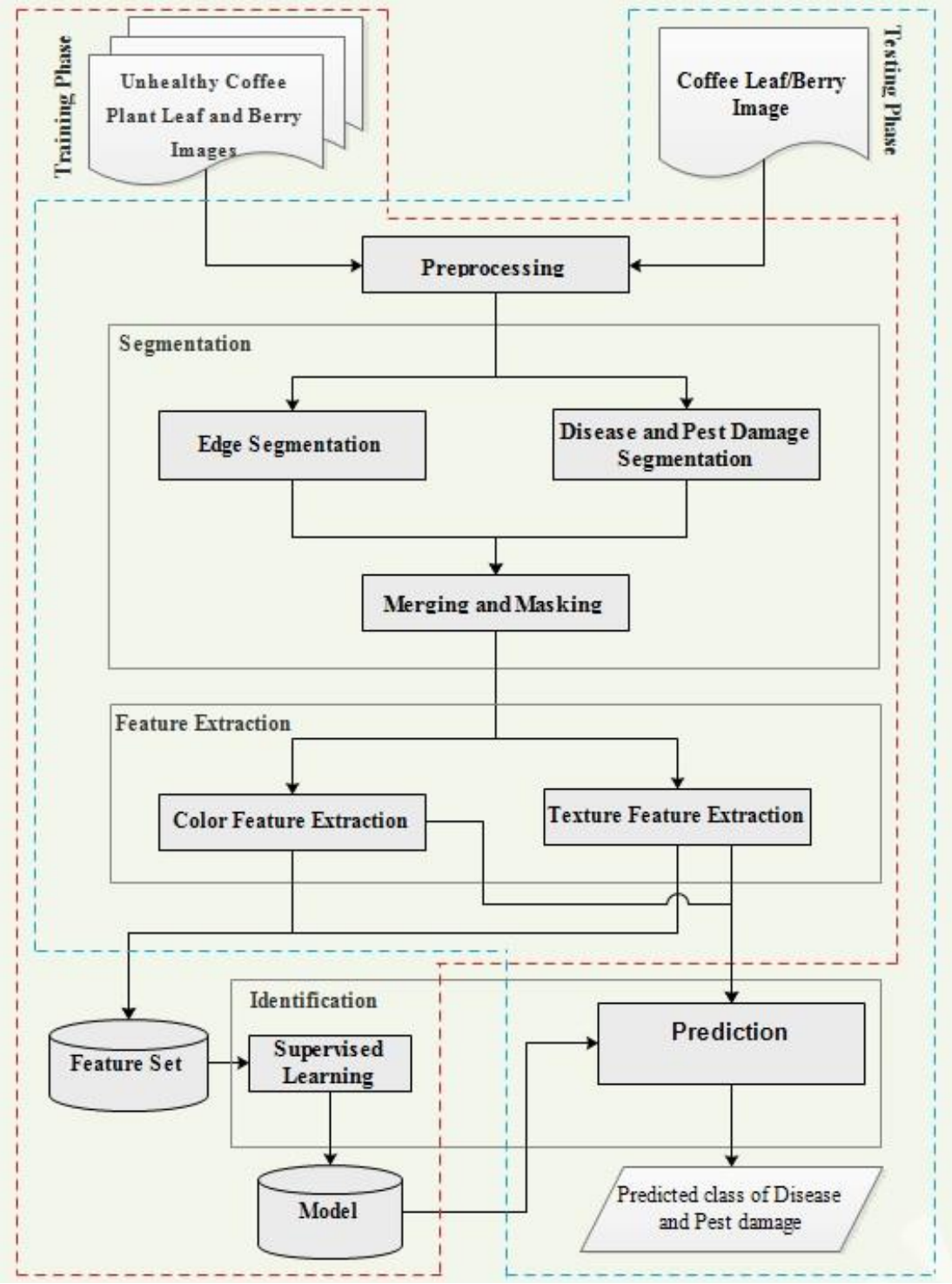


Figure 4.1: Proposed System Architecture

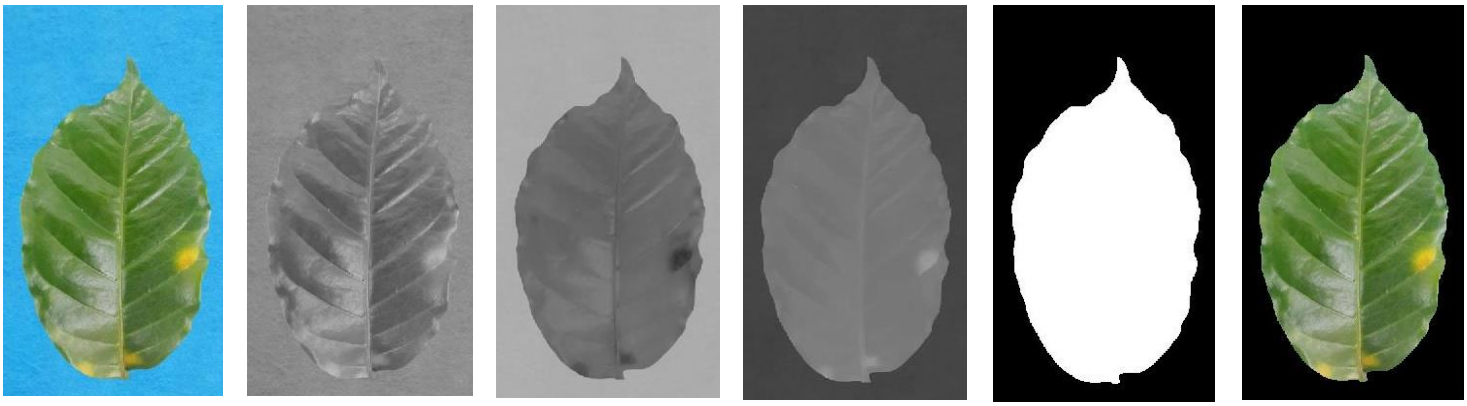
4.3 Preprocessing

Image preprocessing, is the task of enhancing some features of an input image for further processing. Thus, the preprocessing component of this architecture applies scaling and background subtraction on input images of coffee plant leaf/berry. We used *bicubic-interpolation* and *anti-aliasing* to resize the resolution of input images based on a resize factor of 0.25. After scaling the input images, the preprocessing component separates the leaf/berry from the background we used using color histogram thresholding. In order to achieve that, first the scaled RGB image is converted to YCbCr color space.

We defined a threshold value for each Y, Cb, and Cr channels based on their histogram setting. Using the chosen histogram threshold, we created a mask that later on used to initialize a masked image. Finally, we converted the masked YCbCr image back to RGB color space. Algorithm 4.1 shows the steps taken in the preprocessing component. Sample images of resized and background removed coffee plant leaf and berry are depicted in Figure 4.2 and Figure 4.3 with their corresponding Y channel, Cb channel, Cr channel, mask, and masked RGB images respectively.

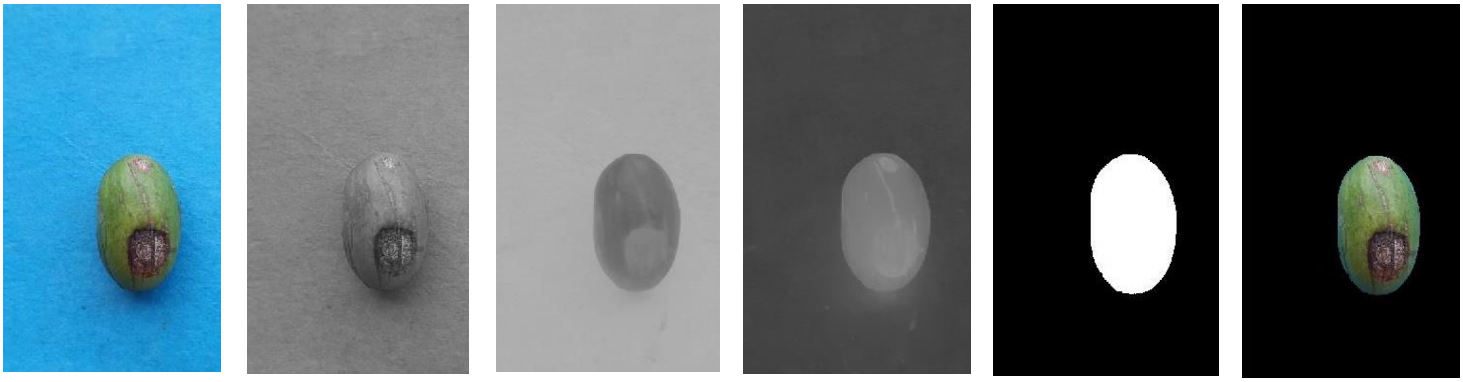
```
Input: im: RGB image of coffee leaf/berry

Output: Masked_RGB_Image: Rescaled and background removed image

                Resize the image

                Convert the resized image to YCbCr image

                Calculate threshold values for each Y, Cb and Cr channels

                Create a YCbCr image using the thresholds

                Convert the YCbCr color image to RGB color space
```

Algorithm 4.1: Input image preprocessing



(a) (b) (c) (d) (e) (f)

Figure 4.2: (a) Scaled RGB image (b) Y Channel (c) Cb Channel (d) Cr Channel (e) Mask (f) Masked RGB image



(a) (b) (c) (d) (e) (f)

Figure 4.3: (a) Scaled RGB image (b) Y Channel (c) Cb Channel (d) Cr Channel (e) Mask (f) Masked RGB image

4.4 Segmentation

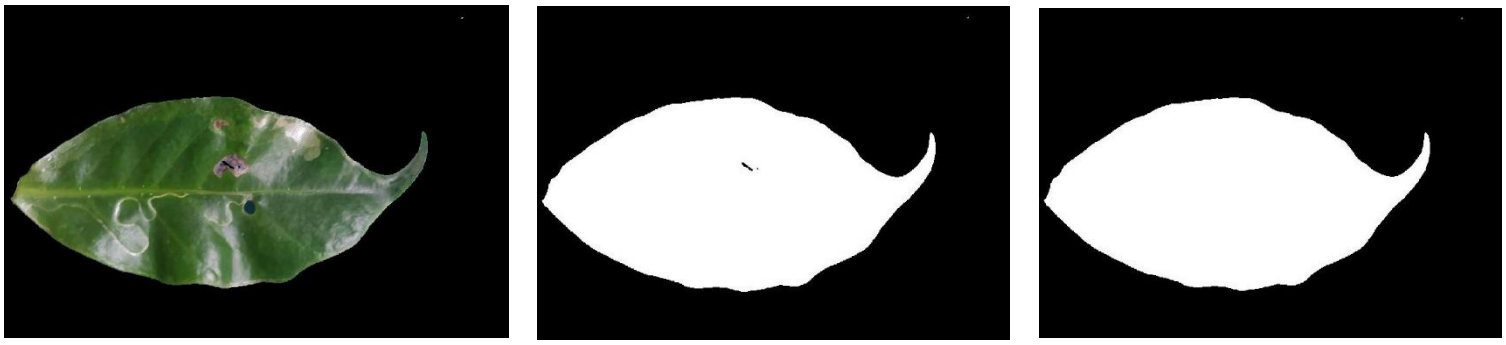
The segmentation component continues after receiving the output of the preprocessing algorithm (i.e. the *mask* and *masked RGB image*). The segmentation component of the proposed architecture applies edge segmentation, disease and pest damaged segmentation and finally merging and masking. The edge segmentation sub-component extracts only the edges of leaf/berry parts. While the disease and pest damaged segmentation sub-component analyzes and segments regions of interest from the surface of the leaf/berry part. The merge and masking sub-component, merges and masks the end results of the two sub-components and passes the output to the feature extraction component. The general steps followed to achieve the segmentation process is shown in Algorithm 4.2.

```
Input: Masked_RGB_Image: Preprocessed image

Output: Segmented Image: the result of the final segmentation step

                Load Masked_RGB_Image

                Segment edge of Masked_RGB_Image using Algorithm 4.3

                Segment damaged regions of leaf/berry using Algorithm 4.4

                Concatenate segmented edge and damaged regions using
                Algorithm 4.5
```

Algorithm 4.2: Preprocessed image Segmentation

4.4.1 Edge Segmentation

Extracting the edge of leaf/berry would help us identify the exact boundaries of the leaf/berry. In order to extract the edges, first, we convert the preprocessed image to a binary image and filled regions of holes using a morphological operation. Then, we filter the local range of the binary filled image. Figure 4.4 depicts masked RGB image, binary image and regions of holes filled images respectively.

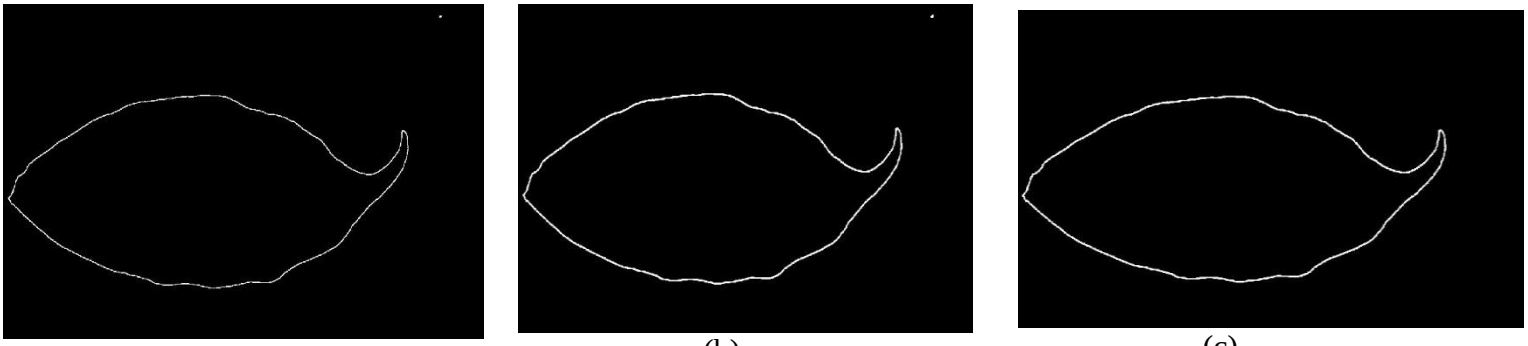


(a) (b) (c)

Figure 4.4: (a) Masked RGB image (b) Binary Image (c) holes filled binary image

Local range filter, in some cases, may result in weak edges. So, to resolve this issue, we applied image dilation on the resulting binary image using a structuring element object. The result of image dilation, as shown in Figure 4.5 b, also contains unwanted features that are not part of the leaf/berry edge. We call these objects small regions. In order to remove these regions, we experimented on the foreground connected components/objects pixel size. Based on our observation, small regions

are those regions with a pixel size of 400 or less. Thus, any connected components less than 400 pixels are removed using morphological opening. Figure 4.5 shows local range filtered, dilated and small regions removed binary images respectively. Algorithm 4.2 shows the steps taken for leaf and berry edge segmentation.



(a) (b) (c)

Figure 4.5: (a) local range filtered image (b) dilated binary image (c) Small regions removed the image

```
Input: Masked_RGB_Image: Preprocessed image

Output: Only_Edge: Leaf/Berry edge extracted binary image


            Load Masked_RGB_Image

            Convert Masked_RGB_Image to a binary image

            Apply morphological filling on the binary image

            Calculate the local range of the binary filled image

            Apply morphological erosion on the local range filled image

            Apply morphological opening on the eroded image to remove small
objects
```

Algorithm 4.3: Edge Segmentation

4.4.2 Disease and Pest Damage Segmentation

The second sub-component of the segmentation algorithm, disease, and pest damage segmentation analyzes and segments regions of diseased and pest damaged surfaces of coffee plant leaf/berry. In order to proceed with the segmentation process, first, the preprocessed masked RGB image is loaded and its' contrast and brightness are adjusted. We achieved this, by mapping the intensity values in the masked RGB image to new values in image intensity adjusted image. The masked RGB image and its' corresponding intensity adjusted images of sample coffee plant leaf and berry are shown in Figure 4.6.

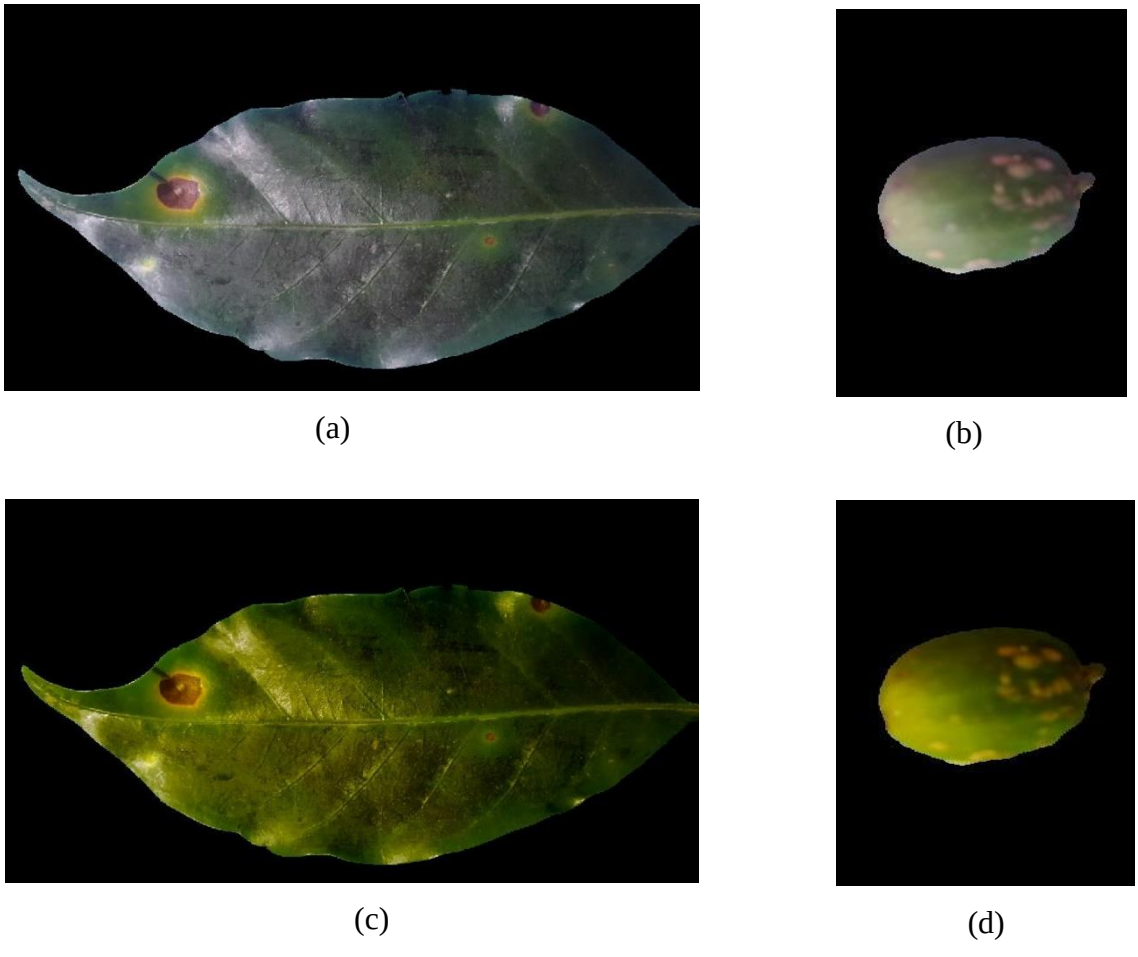
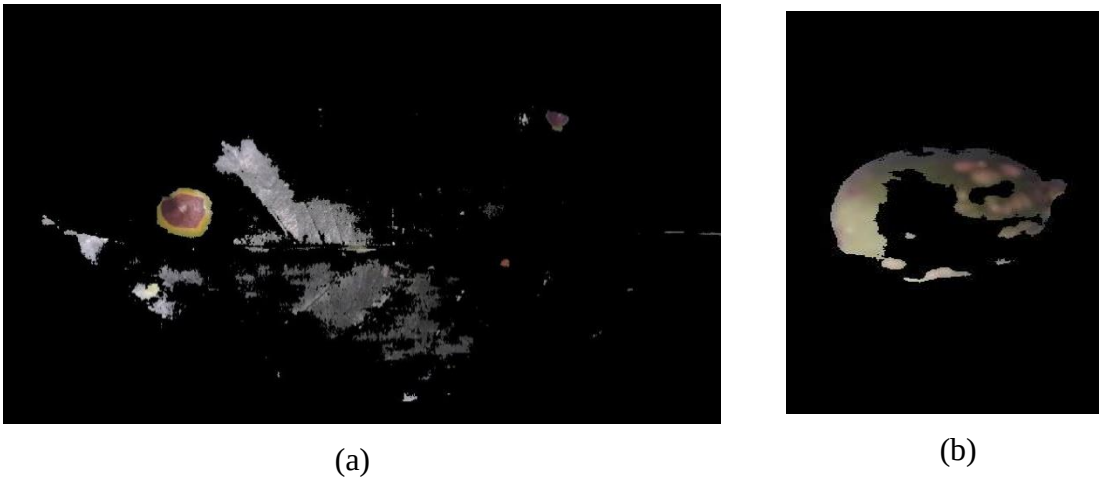


Figure 4.6: (a) (b) Masked RGB images (c) (d) Intensity adjusted images

The intensity adjusted image helped us better analyze the color features of the leaf and berry images. Based on our observation and analysis of coffee plant leaf and berry colors, the green component is more dominate than the red and blue components at the healthy regions of the

leaf/berry. This means the pixel value of the green component is greater than the pixel value of red and blue components. We used this to our advantage and subtract the healthy parts of the leaf/berry by creating a mask that checks the pixel values of each red, green and blue components at each pixel and masks the region, where the pixel value of green is greater than the pixel value of the red and the blue components.

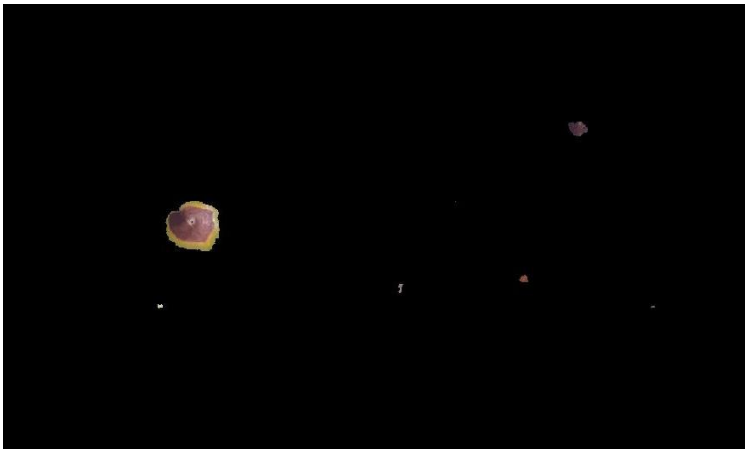
We converted the mask to the same datatype and sparsity as the preprocessed RGB masked image, then performed element-by-element binary multiplication between the two images, resulting in a healthy region masked RGB image. Figure 4.7 shows the green pixels masked RGB image.



(a) (b)

Figure 4.7: (a) green pixels masked image leaf (b) green pixels masked image berry

Pixel values of the green component, in some instances, can have a smaller pixel value than the blue component, even though that particular pixel point is considered to be a healthy region. This occurs mostly when there is a presence of glare in the input image caused by sunlight reflection or shadow. So, the healthy region masked image can include not only the disease/pest damage but also healthy regions of the leaf/berry as shown in Figure 4.7. To overcome this, we further analyzed the masked image and applied color thresholding by converting the RGB image to the YCbCr color space. In YCbCr color space, we set a threshold value for the Y, Cb and Cr channels based on the histogram settings. Using the chosen histogram threshold, we created a mask that later on used to initialize a masked image and converted the masked YCbCr image back to RGB color space. Figure 4.8 shows the end result after applying histogram thresholding on the YCbCr color space. Algorithm 4.4 shows the overall steps of disease and pest damaged segmentation.



(a)

(b)

Figure 4.8: (a) output of histogram thresholding on YCbCr color space leaf (b) output of histogram thresholding on YCbCr color space berry

Input: Masked_RGB_Image: Preprocessed image

Output: Disease_region: RGB image of Segmented disease/pest damage regions

Load Masked_RGB_Image

Adjust the intensity of the Masked_RGB_Image

Extract the Red, Green and Blue components of the intensity adjusted image

Create a Mask for Green pixels

Map mask by applying element- by- element multiplication of Mask and Masked_RGB_Image

Convert mapped RGB image to YCbCr color space

Threshold each Y, Cb and Cr channels based on a Threshold set and concatenate the channels

Convert the YCbCr image back to RGB color space

Algorithm 4.4: Disease and Pest damage segmentation

4.4.3 Merging and Masking

The final step in the segmentation process is merging segmented edges of leaf/berry with segmented regions of disease/pest damage and finally masking. The disease and pest damage segmentation algorithm does a fine job segmenting regions of interest, but rarely it also includes unnecessary edge features. This edge features often appear due to the presence of intensity difference caused as a result of color combinations between the background and foreground objects during image capturing. As a result, the background subtraction processes thresholds some pixel values as a foreground object.

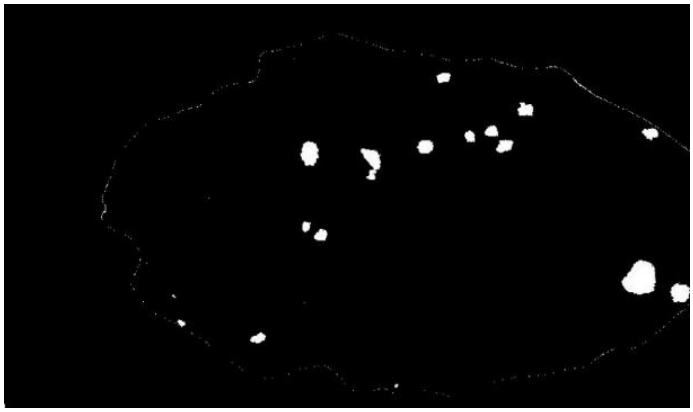
This effect in some cases can be visualized as a doted blue edge on the preprocessed RGB image as shown in Figure 4.9 (a) and (b). These edge features can affect the accuracy of the classifier, thus, merging sub-component removes them by applying image arithmetic on the edge segmentation result and disease and pest damage segmentation result.

(a) (b)

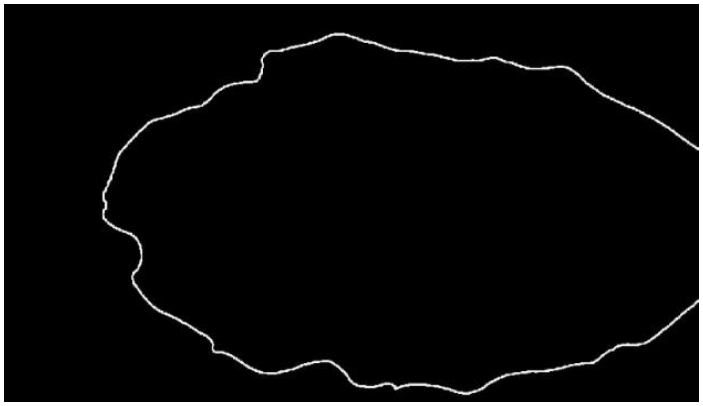
Figure 4.9: (a) Preprocessed RGB image (b) disease and pest damage segmentation result

The merging sub-component loads the end result of edge segmentation, binary image and the end result of disease and pest damage segmentation, RGB image. Disease and pest damage segmented RGB image is first converted into its equivalent binary image. We then subtracted the segmented edge binary image from segmented regions of disease and pest damage binary image, resulting in only diseased/pest damaged regions binary image. Figure 4.10 shows a binary image of segmented disease and pest damage, segmented edge binary image and the result of applying image

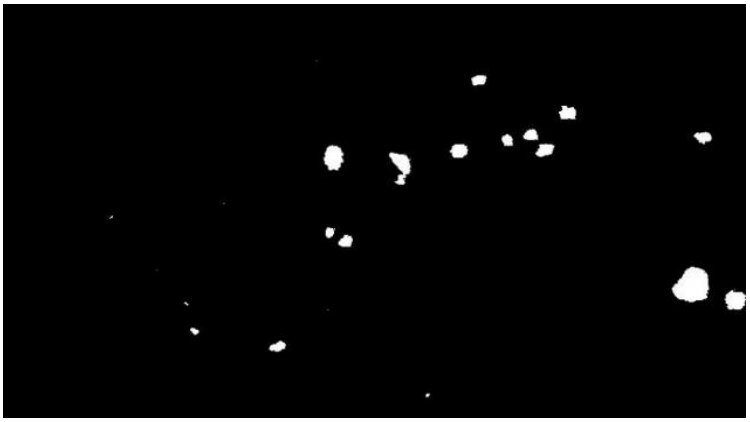
subtraction on segmented edge binary image and segmented disease and pest damage binary image.



(a)



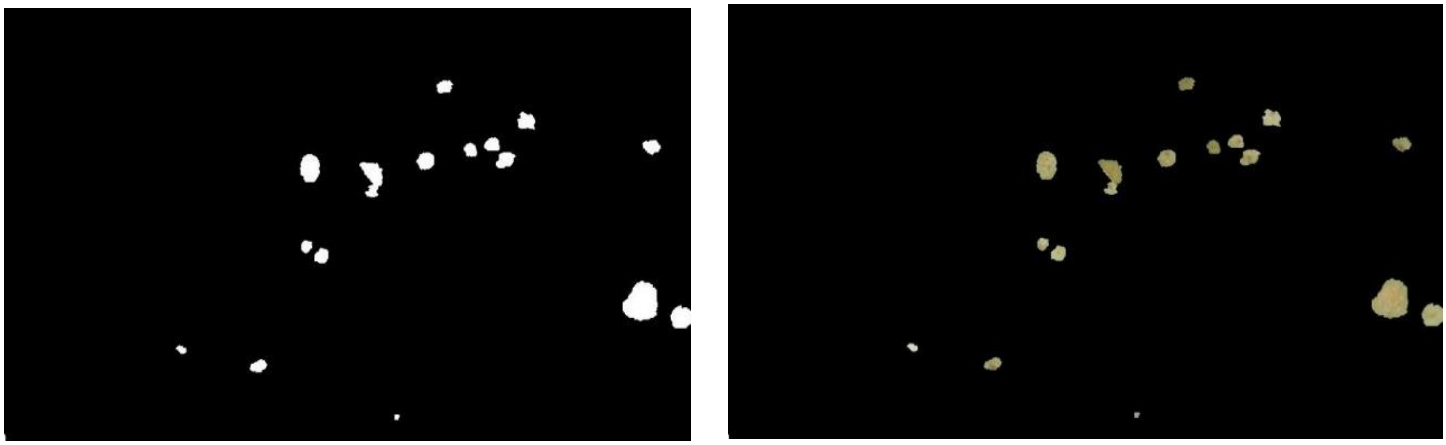
(b)



(c)

Figure 4.10: (a) binary image of segmented disease and pest damage (b) segmented edge binary image (c) result of binary subtracted image

Finally, we applied morphological dilation followed by morphological opening to smooth the resulting binary image. We then used the mask to map on the original image to get the final segmented image. The result of this process is a masked RGB image that contains diseased/pest damaged regions of coffee plant leaf/berry. The masked image is then passed to feature extraction component. Figure 4.11 depicts the smoothed binary image and the final segmented masked RGB Image. Algorithm 4.5 shows all the steps taken in merging and masking sub-component.



(a) (b)

Figure 4.11: (a) smoothed binary image (b) the final segmented masked RGB Image

Input: *Disease_region*: RGB image of Segmented regions of disease/pest damage

Only_Edge: Leaf/Berry edge extracted binary image

Output: Segmented_Image: Final segmented masked RGB image

Load Disease_region RGB image and Only_Edge binary image

Convert the Disease_region RGB image to binary image

Apply morphological subtraction of Only-Edge and Disease_region binary images

Perform morphological dilation on subtracted binary Image

Apply area opening on the dilated binary image

Map the resulting binary image on to Disease_region RGB image

Algorithm 4.5: Merging and Masking

4.5 Feature Extraction

The feature extraction component focuses on analyzing and extracting features like color, texture, size, and shape of segmented image regions. For this study, we have chosen color and texture features to best represent regions of interest. We have selected 22 color and 6 texture features to describe regions of coffee plant leaf/ berry, for classification of class CBD, CLR, CLM, BES and GS.

4.5.1 Color Feature Extraction

Color is one of the most important expressive features of plant leaf/ berry disease. For automatic coffee disease and pest damage identification, we have extracted 22 color features to represent disease and pest damaged regions of coffee plant leaf and berry. These color features are extracted from RGB, YCbCr and L*a*b* color models. Algorithm 4.6 shows the color feature extraction algorithm.

The extracted color features are mean value of red component, mean value of green component, mean value of blue component, mean value of RGB component, mean value of L* channel, mean value of a* channel, mean value of b* channel, median value of L* channel, median value of a* channel, median value of b* channel, standard deviation of L* channel, standard deviation of a* channel, standard deviation of b* channel, mean value of Y channel, mean value of Cb channel, mean value of Cr channel, median value of Y channel, median value of Cb channel, median value of Cr channel, standard deviation of Y channel, standard deviation of Cb channel and standard deviation of Cr channel.

Input: Labeled Segmented_Image

Output value: 22 color features

Begin:

For each Labeled_Region r in segmented_Image

Extract red, green, blue layers of r

Calculate

Mean red, green, blue, RGB value

Extract L, a*, b* layers of r from Segmented_Image*

Calculate

Mean L, a*, b**

Median L, a*, b**

Standard deviation L, a*, b**

Extract Y, Cb, Cr layers of r from Segmented_Image

Calculate

Mean Y, Cb, Cr

Median Y, Cb, Cr

Standard deviation Y, Cb, Cr

Value= {(mean(r,g,b,RGB)), (mean(L, a*, b*)), (median(L*, a*, b*)), (standard deviation(L*, a*, b*)), (mean(Y, Cb, Cr)), (median(Y, Cb, Cr)), (standard deviation(Y, Cb, Cr))}*

Return *Value*

END

Algorithm 4.6: Color feature extraction

4.5.2 Texture Feature Extraction

Texture analysis refers to the characterization of regions in an image by their texture content, like surface smoothness, roughness and so on. In this sub-component, we used texture features to help us separate images among certain classes of diseases and pest damages. For this study, we extracted six texture features namely: average gray level, average contrast, smoothness, skewness, uniformity, and entropy based on the intensity histogram of a region as depicted in Algorithm 4.7.

```
Input: Labeled Segmented_Image

Output: Value: six texture features

Begin:

      For each Labeled_Region r in segmented_Image

           convert each RGB r to grayscale

              T = Call statxture () file with parameter. r

              Assign texture feature values as:

                       Average gray level = T(1)

                       Average contrast = T(2)

                       Smoothness = T(3)

                       Skewness = T(4)

                       Uniformity = T(5)

                       Entropy = T(6)

             Value= {Average gray level, Average contrast, Smoothness,
Skewness, Uniformity, Entropy }

Return Value

END
```

Algorithm 4.7: Texture feature extraction

4.6 Identification

The identification component performs the classification of sample images of coffee plant leaf/berry into the types of disease and pest damage categories based on the extracted features of the image. In order to achieve this, the component contains feature set preparation, supervised training, detection, and prediction. Based on the features extracted using feature extraction components, we first created a feature set or knowledge base. Using supervised learning, we then trained a classifier model. The trained classifier is then used to predict the type of disease or pest damage of the coffee plant.

The knowledge base in digital image processing is created using features extracted from an image, which is stored in a database. For this study, we created a database that contains a feature table to store all the 22 extracted color and 6 texture features with their corresponding class category.

4.6.1 Supervised Learning

Although there are other methods like mathematical functions, rule-based algorithm or statistical methods available for classification, we chose ANNs over others. There are several reasons for choosing neural networks over other methods for the purpose of this research work. The classification of disease and pest damage cannot be easy using unique mathematical functions. This is due to the variation in morphology, color and texture of the classes under consideration. ANNs have the potential of solving problems in which some inputs and corresponding output values are known, but the relationship between the inputs and outputs is difficult to translate into a mathematical function.

When compared to other methods, ANNs can tolerate noise better and exhibit low classification error rates. Moreover, compared to statistical methods, ANNs using the BP network could be easily modified to accommodate more features. To add empirical experience to the above claims, we trained naïve Bayesian classifiers and ANN classifiers on the same training data set. We compared their performance based on classification accuracy and we found out that ANN performs better than naïve Bayesian classifier. The neural network architecture in this work is a 3 layered F-F network with sigmoid hidden and softmax output neurons. Such a network can classify vectors arbitrarily well, given enough neurons in its hidden layer. The input layer contains 28 neurons corresponding to every 28 inputs and the output layer consists of 5 neurons corresponding to each 5 output classes.

The network is designed to have 30 hidden layers. This number of neurons in the hidden layer is selected *empirically* based on the performance it exhibited over a smaller and larger number of neurons. The network is designed to use B-P algorithm training. To measure the performance of the network during the training phase, we preferred to use the cross-entropy error function over mean square error (MSE). Compared to MSE, cross entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance. The architectural design of this ANN is depicted in Figure 4.12.

4.6.2 Prediction

The final task of the identification component is predicting the type of disease and pest damage. This sub- component extracts 22 color features and 6 texture features for each input image and then, using the trained classifier it predicts the type of disease and pest damage.

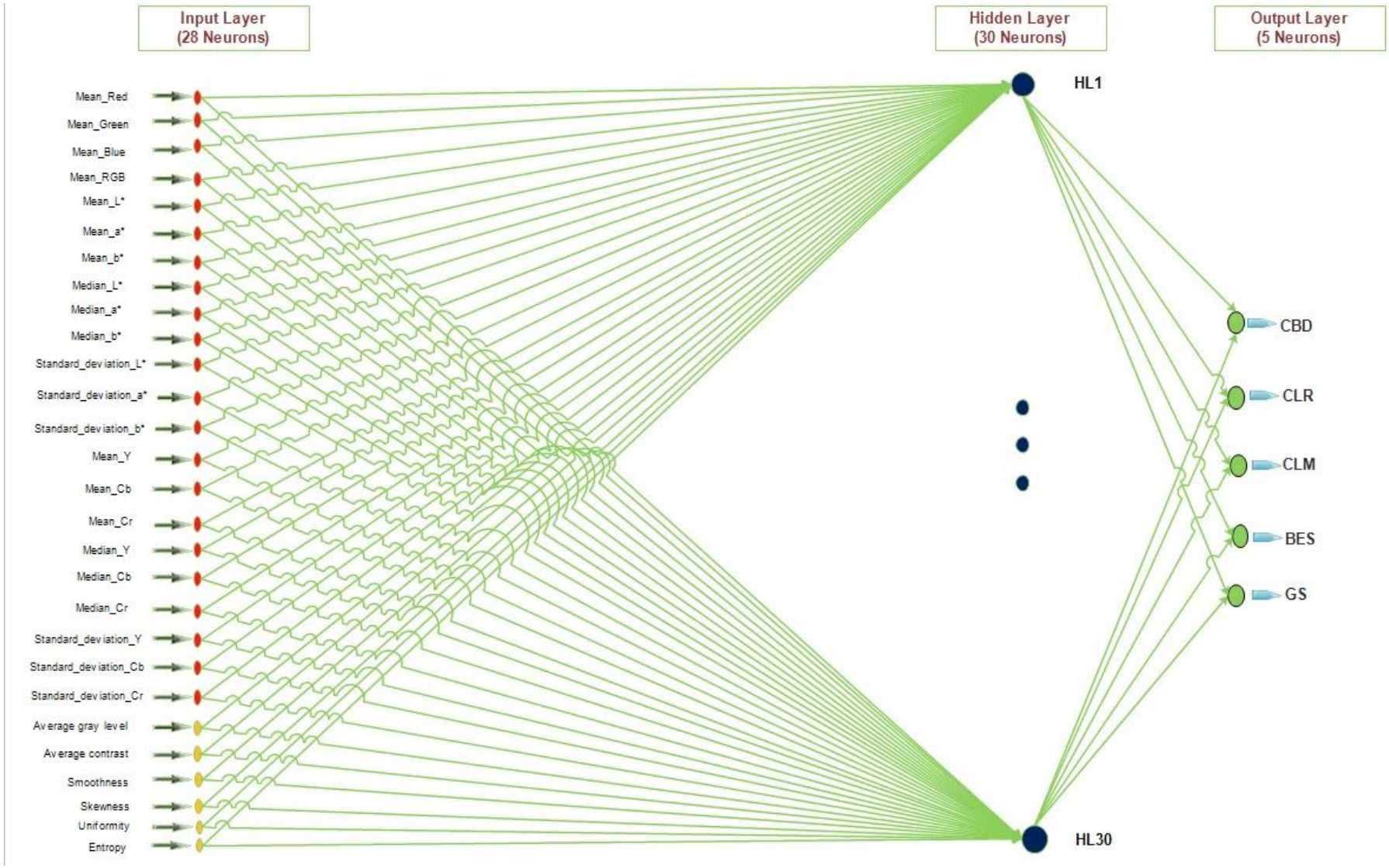


Figure 4.12: Design of ANN used for the classification of coffee plant disease and pest damage

4.7 Summary

We proposed a system architecture that consists of four components, namely, preprocessing, segmentation, feature extraction, and identification. The preprocessing component resizes the input image and subtracts the background. The output of the preprocessing component is then passed to the segmentation component so that the healthy region of the input image is separated from the unhealthy one. To accomplish this, we used edge segmentation and histogram thresholding. The final result of this component is then fed to the feature extraction component. The feature extraction component analyzed and extracted 22 colors and 6 texture features from the segmented image. These features are then stored in a database as a feature set to train a classifier model. We trained the classifier using ANN. The trained classifier is then finally used to predict the type of disease and pest damage.

CHAPTER FIVE: EXPERIMENT AND RESULTS

5.1 Introduction

In this chapter, we discussed the experiments conducted and the results obtained while testing the proposed system architecture. Thus, the dataset used, the classifiers used with accuracy and the discriminative power of color features and texture features are presented.

5.2 Data Collection and Dataset Preparation

Sample images of coffee arabica leaves and berries were collected from coffee growing areas in Ethiopia. We captured a total of 2,400 image samples from the Jimma Agricultural Research Center, Tepi National Spices Research Center, Metchara Agricultural Research Center, Sheko, and Limmu coffee farms. Image acquisition is done using a digital camera. In order to capture a good quality images, we used a camera with a 16 pixel resolution. Sample coffee leaves and berries are placed under the camera and captured vertically. While capturing each sample images we used a background object so that it can minimize unwanted features that affect the segmentation algorithm.

We used the extracted feature set to train the artificial neural network, supported vector machine, and K-nearest neighbor classifiers. In order to train the model, the dataset was split into the training set, validation set, and test set. In the case of ANN, 80% of the dataset was used for training, 10% for validation and the other 10% was used for testing. The training dataset was used to fit the model and the validation dataset provides an unbiased evaluation of the model fit on the training dataset while tuning model hyper parameters. Once the model was completely trained using the training set and validation set, the test set then provided a standard used to evaluate the model performance.

Similarly, while training the model using SVM and KNN classifiers, we used 80% of the dataset for training and the rest 20% for validation. We used 5-fold cross validation to help us better use our data and to get more information about our algorithm performance.

The learning algorithm we used is a supervised one (i.e. the training data must comprise examples of the input vectors along with their corresponding target vectors). Thus, in training data, each target vector was labeled. The labels are presented to the neural network as a one-hot code. Table 5.1 describes the number of sample images used for training purposes and the class labels for CBD, CLR, CLM, BES, and GS.

Table 5.1: Dataset description

No.	Disease and Pest damage class	Class Label	Number of images
1	CBD	00001	500
2	CLR	00010	500
3	CLM	00100	500
4	BES	01000	500
5	GS	10000	400
		Total	2,400

5.3 Implementation

Experimentation was carried out through a prototype developed using MATLAB version 2017(a), MySQL, JAVA NetBeans version 8.2, and android studio. To develop the prototype we used windows 10 operating system with Intel Core i5, 8GB RAM and 2.5GHz processor. MATLAB provides a large database of built-in algorithms for image processing and machine learning applications. It also allows testing algorithms immediately without recompilation. We used MySQL to store extracted features of diseased and pest damaged regions.

We used the android studio to develop a user interface for capturing images of coffee leaf/berry and receiving the predicted class of disease and pest in real time. We used java servlets as a gateway between MATLAB and android application. They receive input images from the android application and send that image to MATLAB for further processing. Once the classifier predicts the type of disease/pest damage, the result will be sent to the android application via java servlets. Figure 5.1 shows a screenshot of the developed prototype for preprocessing, segmentation feature extraction and classification.

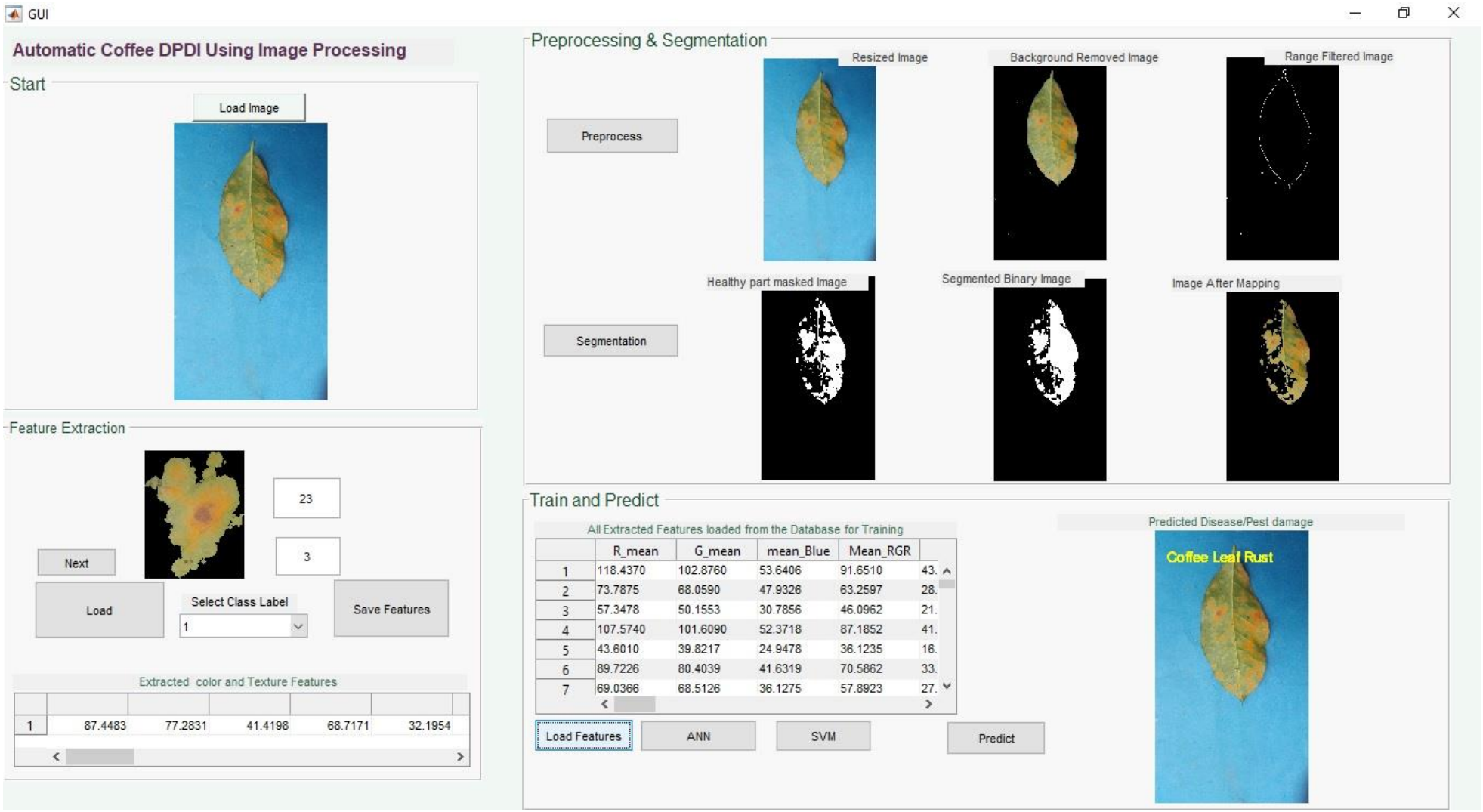


Figure 5.1 Screenshot of the developed prototype for preprocessing, segmentation feature extraction and classification

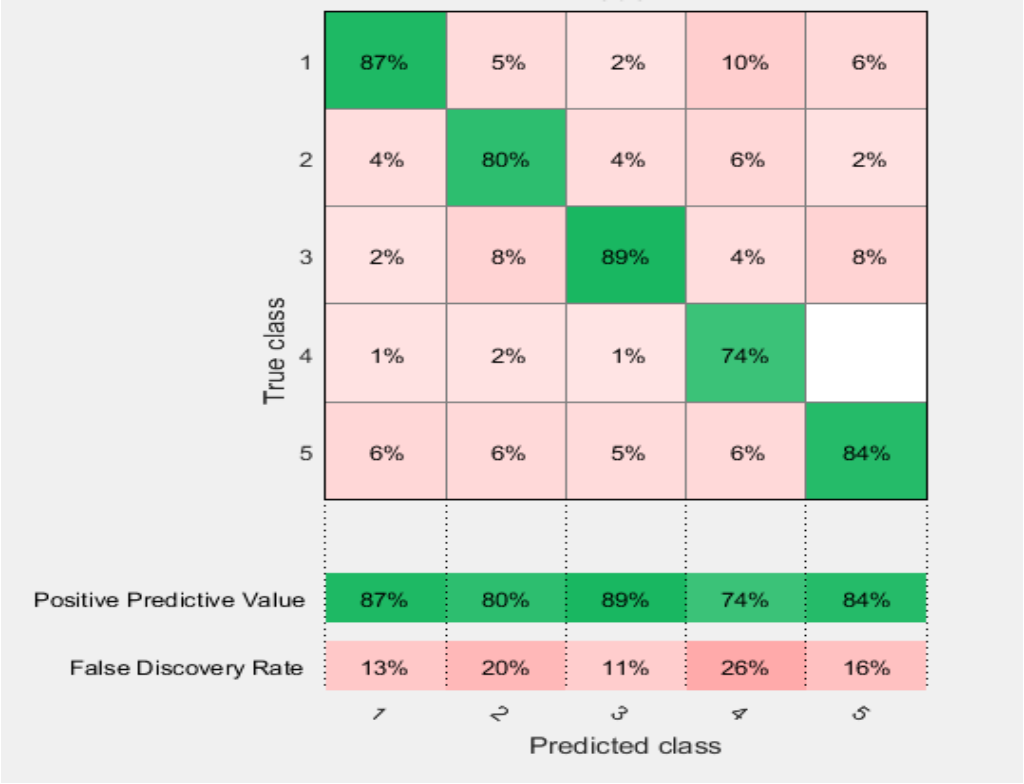
5.4 Test Results

Tests are conducted on SVM, KNN and ANN classifiers to determine the best performing classifier based on the classification accuracy.

5.4.1 Support Vector Machine (SVM) Classifier Test Result

The performance of the SVM classifier was tested with 480 (20% of the training data) dataset. The test confusion matrix of the trained SVM classifier is depicted in Table 5.2. The diagonal elements show the positive predictive values and the rest elements show a false discovery rate. The classification accuracy of CBD, CLR, CLM, BES and GS are 87%, 80%, 89%, 74% and 84% respectively.

Table 5.2: Confusion matrix of SVM classifier accuracy

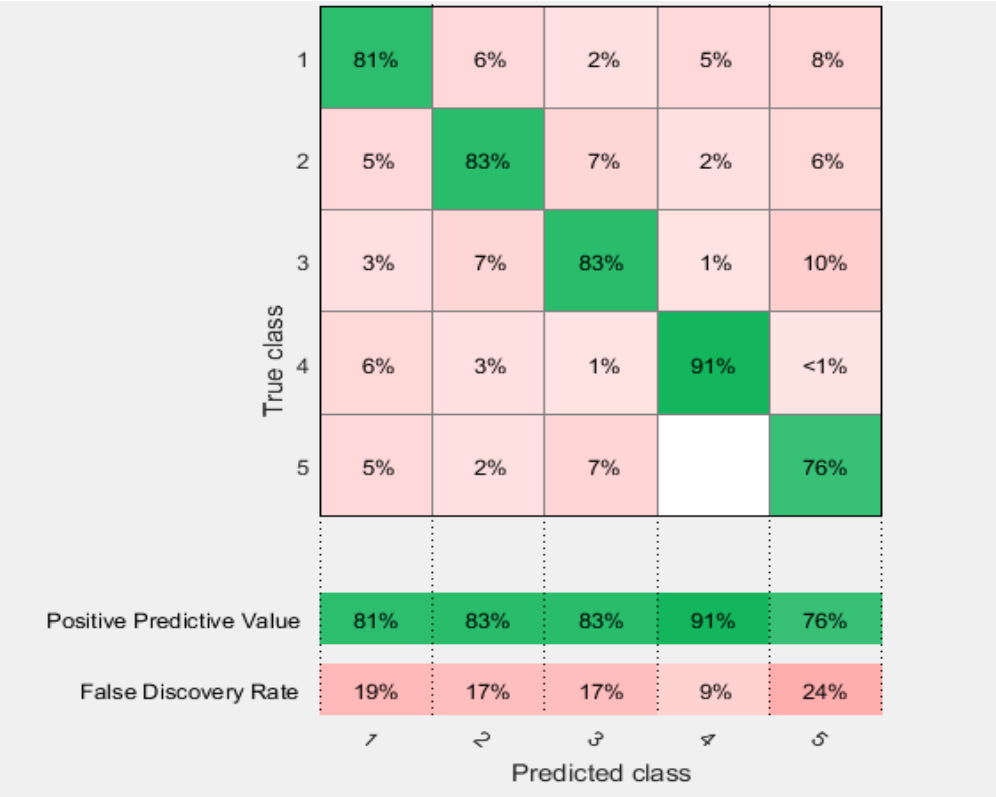


The overall classification accuracy of the SVM classifier is 82.1%. The classification accuracy for classes CLR, BES and GS is below 85%, whereas the classification accuracy for class CBD and CLM is above 85%. The false discovery rate is higher for CLR, BES and GS classes compared to CBD and CLM classes.

5.4.2 K-nearest Neighbor (KNN) Classifier Test Result

The performance of the KNN classifier was also tested with 480 (20% of the training data) image dataset. The test confusion matrix of the trained KNN classifier is depicted in Table 5.3. The diagonal elements show positive predictive values while the rest elements show a false discovery rate. The classification accuracy of CBD, CLR, CLM, BES and GS are 81%, 83%, 83%, 91% and 76% respectively.

Table 5.3: Confusion matrix of KNN classifier accuracy



The overall classification accuracy obtained using the KNN classifier is 82.8%. The classification accuracy for classes CBD, CLR, CLM and GS is below 85%, whereas the classification accuracy for class BES is above 90%. The false discovery rate is higher for classes CBD, CLR, CLM and GS compared to class BES. This has affected the overall performance of the classifier.

5.4.3 Artificial Neural Network (ANN) Classifier Test Result

During ANN training, cross-entropy was used as the error function. The network training process reaches the best validation performance at 87th iteration at which the validation error started to rise while the training error drops. Figure 5.2 shows the performance of the trained neural network classifier.

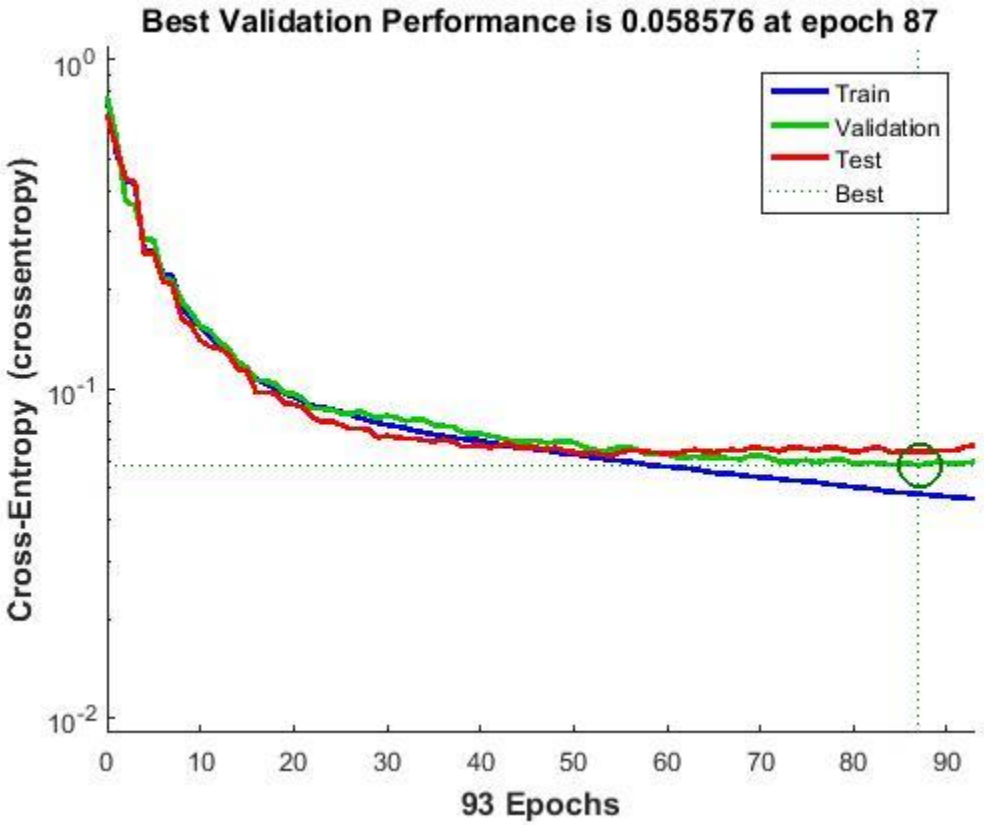


Figure 5.2: Performance of the trained ANN classifier

The Receiver Operating Characteristic (ROC) curve plots the true positive rate versus the false positive rate of the trained ANN classifier. Figure 5.3 shows the ROC curve obtained during training, validation and test phases of the classifier training.

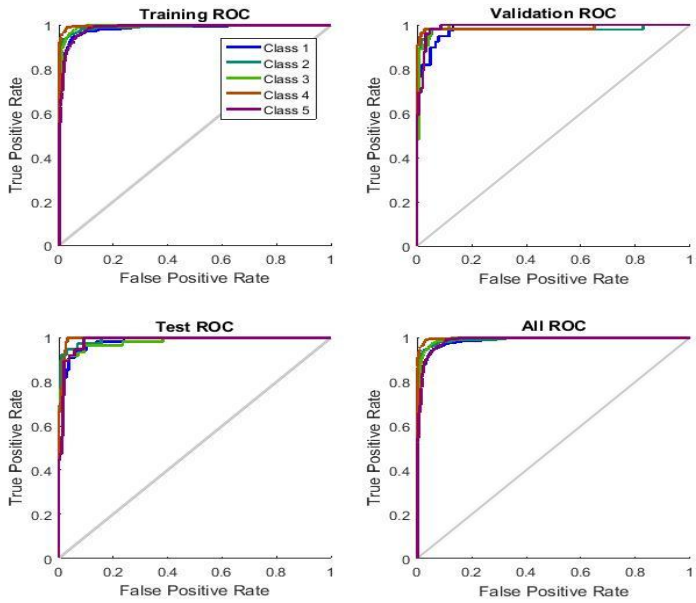
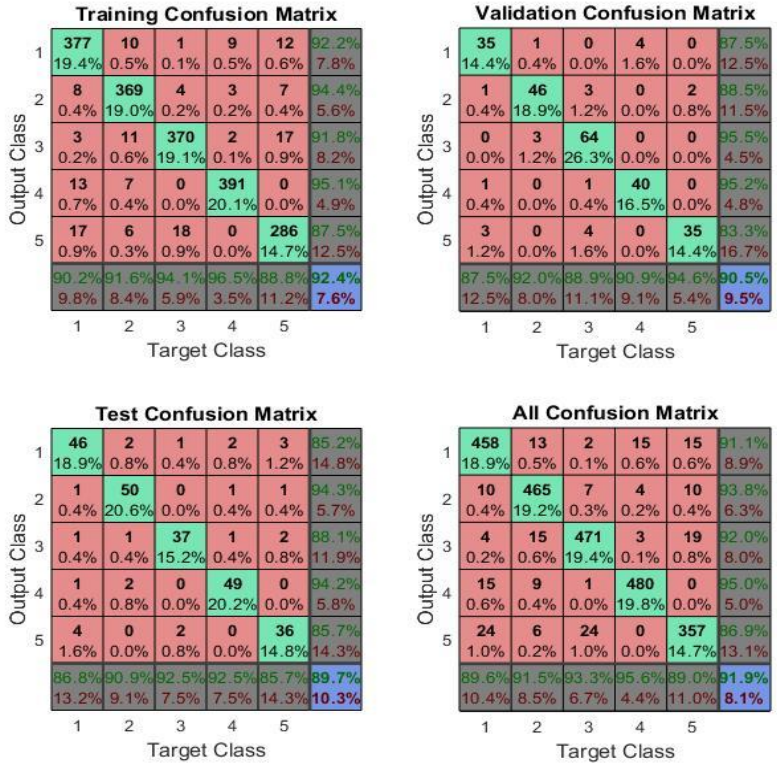


Figure 5.3: ROC curve for training phase, validation phase and Test phase

Classification accuracy obtained for training, validation and testing phases is 92.4%, 90.5%, and 89.7% respectively. The overall classification accuracy is 91.9%. Table 5.4 shows the confusion matrix for the training phase, validation phase, testing phase and also the overall classification accuracy.

Table 5.4: Confusion matrix of the overall classification accuracy of ANN classifier

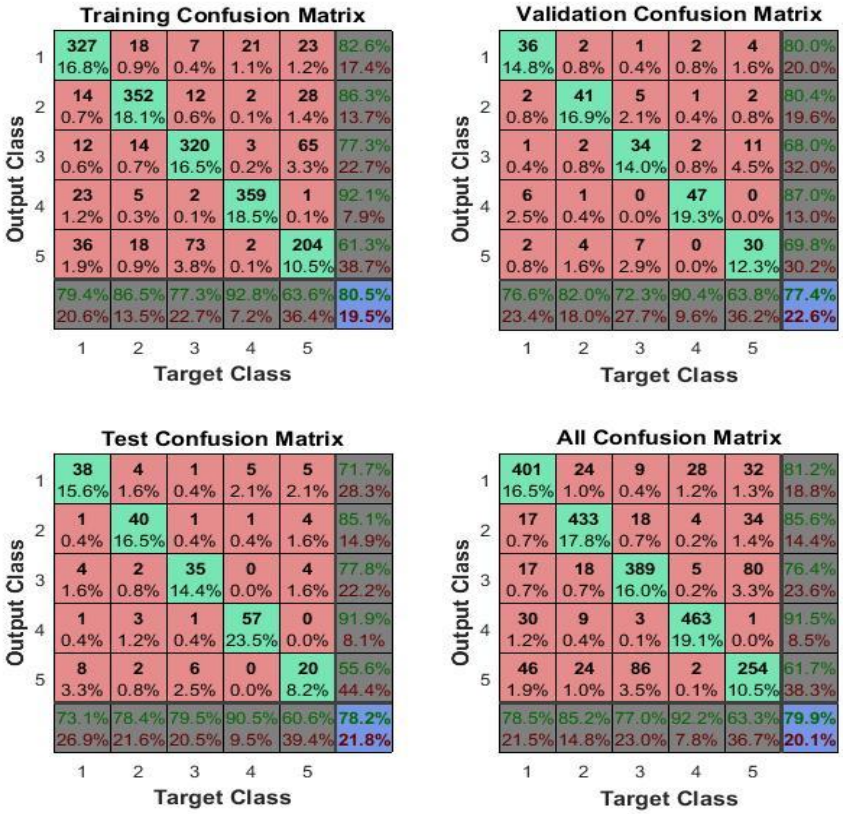


Identifying the impact of the feature types on the overall classification accuracy is vital. To compare the discriminative power of features we prepared two scenarios and trained the model under each scenario.

Scenario 1- Using color features

When we train the model using only color features, the overall classification accuracy scooped down to 79.9%. We obtained a classification accuracy of 76.6%, 82.0%, 72.3%, 90.4% and 63.8% for classes CBD, CLR, CLM, BES and GS respectively. Table 5.5 shows the confusion matrix plot for the training phase, validation phase, test phase and also overall classification accuracy while training the model using the 22 color features.

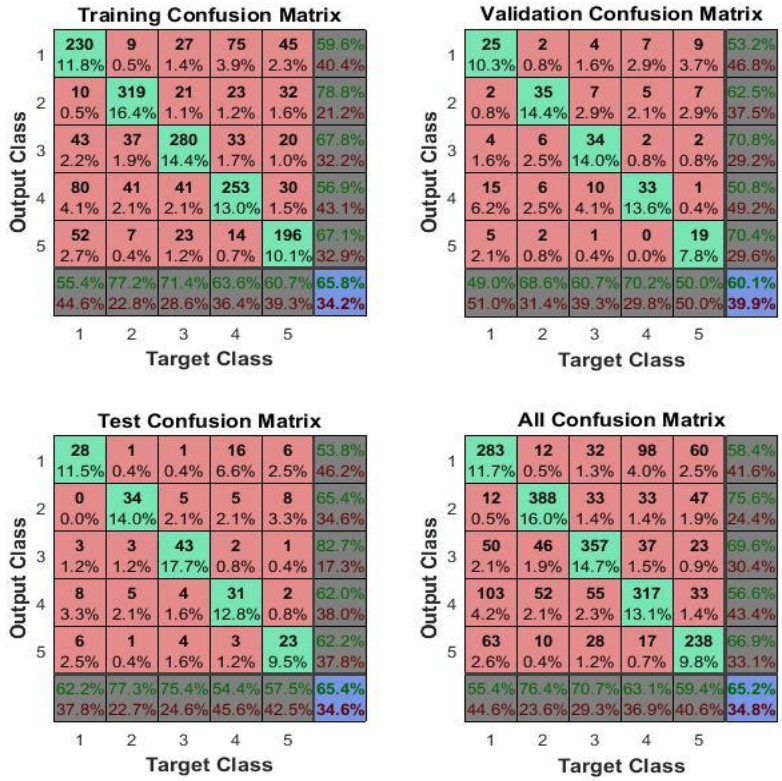
Table 5.5: Confusion matrix for classification accuracy obtained while training with color features



Scenario 2- Using texture features

Using only texture feature we trained the model just to obtain an overall classification accuracy of 65.2%. For classes CBD, CLR, CLM, BES and GS, classification accuracy was 55.4%, 76.4%, 70.7%, 63.1% and 59.4% respectively. Table 5.6 shows the confusion matrix plot for the training phase, validation phase, test phase and also overall classification accuracy while training the model using the 6 texture features.

Table 5.6: Confusion matrix for classification accuracy obtained while training model with texture features



5.5 Discussion

The discriminative power of color features shows that color features presented a higher performance to the overall classification accuracy of the model. As depicted in Table 5.5 for classes BES and CLR the classification accuracy was good. Classes CBD, CLM and GS show a high misclassification rate, particularly in the case of CLM and GS misclassification rate was above 3.3%. The reason for that is both CLM and GS present similar color features. Class CBD also exhibited a higher misclassification rate within classes BES and GS. The reason for that is CBD, BES and GS share some color features.

The result of training the ANN model with only 6 texture features shows a classification accuracy of 65.2% as depicted in Table 5.6. The classification accuracy for all classes dropped down significantly, especially classes CBD, BES and GS score below 70%. Some samples of class CBD were misclassified as BES and vice versa. Based on our observation, the reason for these misclassifications is CBD and BES in some cases present similar texture features. The same

misclassification was observed with CBD and GS, hence highly affecting the classification accuracy.

Based on the experimental results obtained from training the model separately with color features and texture features, it clearly shows that color features played a higher role in classification accuracy. Figure 5.4 shows the comparison of the discriminative power of color features and texture features using a bar chart.

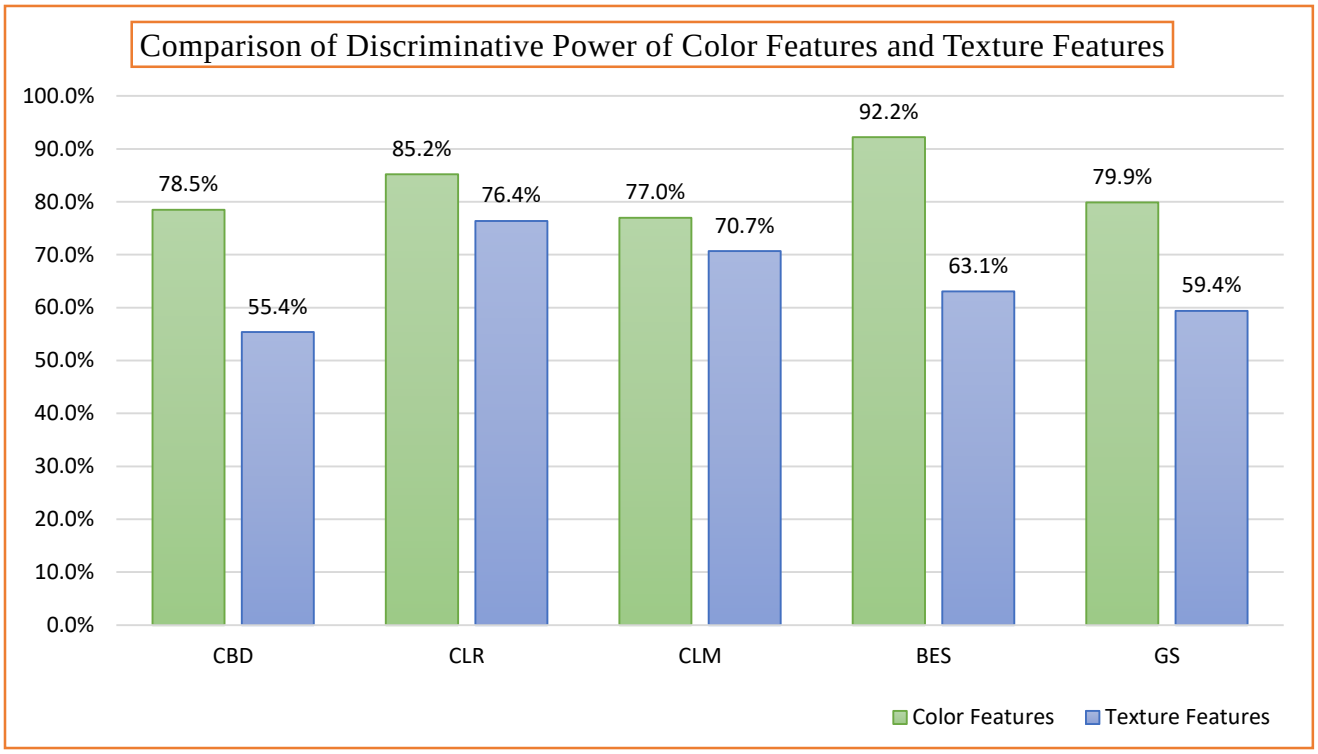


Figure 5.4: Bar chart of discriminative power of color and texture feature

The performance of SVM classifier shows a higher classification accuracy than obtained from KNN classifier for classes CBD, CLM and GS. But, when it comes to classes CLR and BES the performance of KNN classifier shows a better classification accuracy than SVM classifier. Even though KNN and SVM classifiers show a promising classification accuracy for a few classes, when compared with ANN classifier, the model shows improved classification accuracy for all classes. Figure 5.5 shows the overall performance analysis of ANN, SVM and KNN classifiers for this particular research using a bar chart.

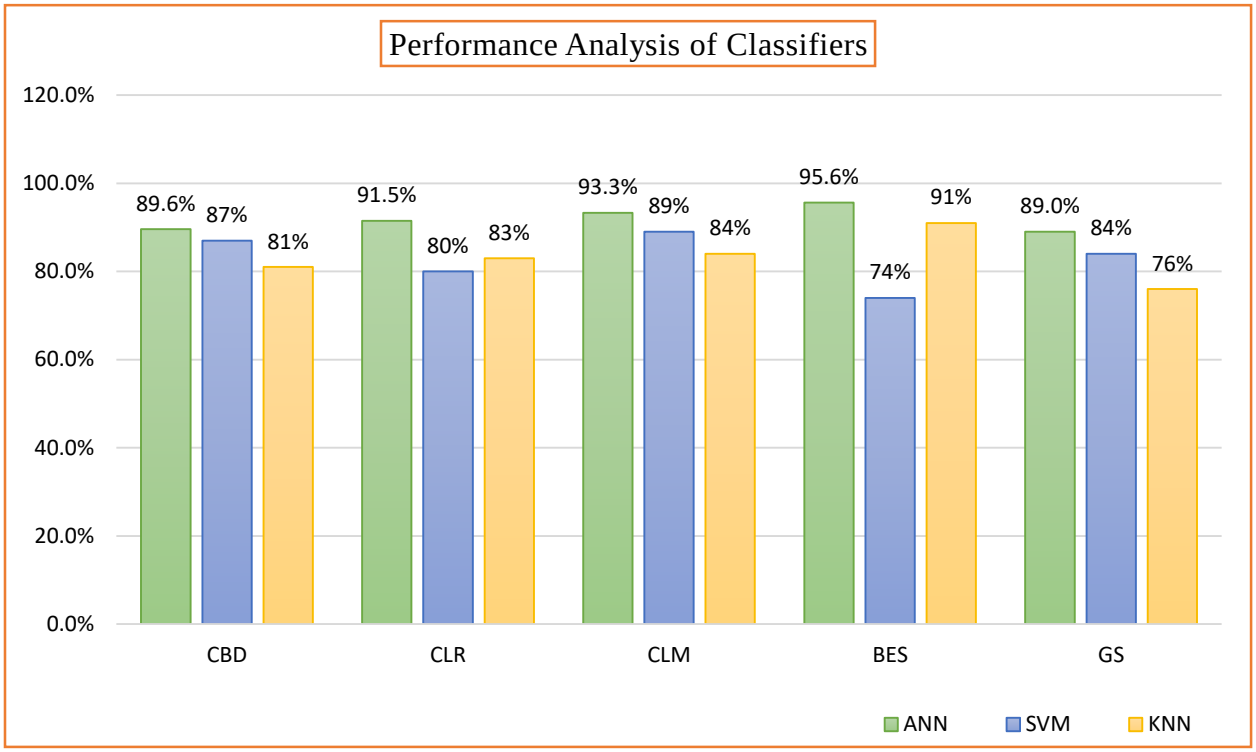


Figure 5.5: Bar chart for Performance analysis of classifiers

5.6 Summary

In this chapter, we have seen the experimental results of this research work from dataset collection, dataset preparation, and prototype design to classifiers selection, model training and discriminative power of features. Relatively high classification accuracy was achieved while training the model using ANN classifier. And also the discriminative power of color features is slightly greater than texture features.

CHAPTER SIX: CONCLUSION AND FUTURE WORKS

6.1 Conclusion

Coffee is a major commodity consumed as a beverage by one third of the world's population. Being the origin of coffee, Ethiopia covers 7-10 % of global coffee production. Around 15 million Ethiopians' income directly or indirectly depend on coffee production. Thus, the coffee crop plays a major role in the country's economic growth and stability. To increase the revenue and also to keep benefiting from coffee, the major treats that hinder its production (i.e. diseases and pests) should be controlled.

Pests and diseases are one of the major treats that affect the quality and yield of coffee production in Ethiopia. Some of the common diseases and pests that cause harm to coffee production are coffee berry disease, coffee leaf rust, coffee leaf miner, brown eye spot and green scale. In order to control the damage caused by these diseases and pests, a correct and timely diagnosis and identification of the diseases/pests type is crucial.

The main purpose of this research work is to develop an automated classification of coffee pests and diseases through image processing and machine learning techniques. For this reason, we presented a system architecture that composes preprocessing, segmentation, features extraction, and identification components.

The preprocessing component reduces the size of input images of coffee leaves/berries and also removes background features. In the segmentation component, we developed a segmentation algorithm that best outlined the damaged regions of coffee leaves and berries. This component includes edge feature segmentation sub-component and damaged region segmentation sub-components.

The third component of the system architecture is feature extraction. A total of 28 features are selected to represent the regions of interest. We categorized the 28 features in to 22 color features and 6 texture features. These features were extracted and stored in a database as a feature set.

The identification component includes classification and prediction. For classification, a feedforward artificial neural network with 28 input nodes and 5 output nodes is designed. We collected a total of 2,400 sample images for training and testing purposes. The neural network performance is compared with SVM and KNN classifiers. The result shows that the overall classification accuracy of ANN for coffee pests and diseases is 91.9%. The true positive rate for classes coffee berry disease, coffee leaf rust, coffee leaf miner, brown eye spot and green scale are 89.6%, 91.5%, 93.3%, 95.6%, and 89.0% respectively.

6.2 Contribution to Knowledge

This research work has contributed the following to the area of digital image processing and machine learning in relation to automatic coffee disease and pest damage identification.

- We proposed a segmentation algorithm capable of segmenting regions of diseased and pest damaged coffee plant leaf and berry.
- We proposed system architecture for automatic identification of coffee berry disease, coffee leaf rust, coffee leaf miner, brown eye spot and green scales.
- We built a classifier model using an artificial neural network that can successfully label coffee diseases and pest damage elements.

6.3 Future Work

This research work shows the identification of coffee berry disease, coffee leaf rust, coffee leaf miner, brown eye spot and green scales using image processing and machine learning. Yet, there are still a few works that need further investigation. In this section, we present possible future works.

- In this study, features are extracted from the leaf and berry parts of the coffee plant. But, the stem part of the plant also shows signs and symptoms for other hazards pests and diseases of the coffee plant. Extracting features from the stem of the coffee plant presents a challenge, thus a potential future work.

- As a result of an insufficient dataset, this research work didn’t include coffee berry borer and die back disease. Therefore, future studies can extend this work to include coffee berry borer and die-back disease and others as additional classes.

REFERENCES

[1] Minten. Bart, Seneshaw Tamru, Tadesse Kuma and N. Yaw, "Structure and performance of Ethiopia's coffee export sector," International Food Policy Research Institute, Washington DC, 2014.

[2] Tewodros Ayalew, "Characterization of organic coffee production, certification and marketing systems: Ethiopia as a main indicator: a review," *Asian journal of agriculture research,* vol. 8, no. 4, pp. 170-180, 2014.

[3] Alemayehu Asfaw, "Coffee Production and Marketing in Ethiopia," *European Journal of Business and Management,* vol. 6, no. 37, pp. 109-121, 2014.

[4] Fekadu Gemechu, Melese Maryo and Girmaye Benti, "The prevalence and impact of coffee arthropod pests in the gedeo indigenous agro forestry systems, Southern Ethiopia," *International Journal of Biodiversity and conservation,* vol. 8, no. 10, pp. 233-243, October 2016.

[5] Khadabadi Gouri, Rajpurohit Vijay, Kumar Arun and B.Nargund, "Disease Detection in Vegetables Using Image Processing Techniques: A Review," *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE),* vol. 14 , no. 2, pp. 954-960, APRIL 2015..

[6] T. Amos, M. V. Martin and C. L. Heald, "Threat to future global food security from climate change and ozone air pollution," *Nature Climate Change,* vol. 4, no. 9, pp. 817-821, 2014.

[7] Food and Agriculture Organization of the United Nations, "Pollinators vital to our food supply under threat," FAO, 2016.

[8] S. Peter and S. Richard, "ISPP and the Challenge of Food Security," in *Springer Science+Business Media*, Edinburgh, 2010.

[9] International Fund for Agricultural Development (IFAD) and UNEP, "Smallholders, food security, and the environment," International Fund for Agricultural Development (IFAD), rome, 2013.

[10] Celia. A. Harvey, Zo L. Rakotobe, N. S. Rao, R. Dave, H. Razafimahatratra, R. H. Rabarijohn, H. Rajaofara and J. L. MacKinnon, "Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar," *Royal Society publishing* , vol. B, no. 369, pp. 1-11, 2014.

[11] A. Deokar, A. Pophale, S. Patil, P. Nazarkar and S. Mungase, "Plant Disease Identification using Content Based Image Retrieval Techniques Based on Android System," *International Advanced Research Journal in Science, Engineering and Technology,* vol. 3, no. 2, pp. 44-46, February 2016.

[12] Bewketu Haile, Tsegaye Babege and Awalom Hailu, "Diseases and Insect Pests of Onion (Allium cepa L.) in Masha District of Sheka Zone, South-West Ethiopia," *Academia Journal of Agricultural Research,* vol. 4, no. 10, pp. 629-632,, 2016..

[13] Million Assefa, "Significance of Arthropod pests of coffee in Ethiopia," in *In "Proceedings of the workshop on the control of Coffeee Berry Disease/CBD in Ethiopia*, Addis Ababa,, 13-15 Aug. 1999.

[14] Mike A. Rutherford and Noha Phiri, "Pests and Diseases of Coffee in Eastern Africa," CAB International, Wallingford, 2006.

[15] Getahun Tigistu, "Automatic flower disease identification using image processing" Department of Computer Science, Addis Ababa University, 2015.

[16] Abrham Debasu, Seffi Gebeyehu and Dagnachew Melesew, "Image Analysis for Ethiopian Coffee Plant Diseases Identification," *International Journal of Biometrics and Bioinformatics*, vol. 10, no. 1, pp. 1-11, 2016.

[17] Jain, Anil K, "Fundamentals of digital image processing," *Englewood Cliffs*, vol. 3.

[18] Tadesse Gebremdhin, COFFEE PESTS IN ETHIOPIA, Addis Ababa: INSTITUTE OF AGRICULTURAL RESEARCH, September, 1984.

[19] Marc. J. Hermann A. Jürgen Pohlan, "Growth and production of coffee," *SOILS, PLANT GROWTH AND CROP PRODUCTION*, vol. 3.

[20] Anwar Abasanbia Abadiga, "Assessment Of Coffee Quality And Its Related Problems," Unpublished , Jimma, 2010.

[21] R.Muller, D. Berry, J. Avelino and D. Bieysse, "Coffee Diseases," in *Coffee: Growing, processing, sustainablr Production*, Viernheim, hagedorn, 2009, pp. 495-549.

[22] Maria Petrou, Costas Petrou, Image Processing: The Fundamentals , Second Edition, United Kingdom: A John Wiley and Sons, Ltd., 2010.

[23] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing Third Edition, United States of America: Pearson Education, Inc., 2008.

[24] Wilhelm Burger, Mark J. Burge, Principles of Digital Image processing: Fundamental Techniques, London : Springer-Verlag, 2009.

[25] Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, "Fundamentals of Image Processing," Delft University of Technology, The Netherlands, 2007.

[26] B. Jähne, Digital Image Processing 5th edition, Berlin Heidelberg : Springer-Verlag, 2002.

[27] Rivazhagan S, Newlin Shebiah, S. Ananthi, Vishnu Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," *Agricultural Engineering International*, vol. 15, no. 1, pp. 211-217, 2013.

[28] Narendra and K. V.G. and Hareesh, "Prospects of Computer Vision Automated Grading and Sorting Systems in Agricultural and Food Products for Quality Evaluation," *International Journal of Computer Applications (*, vol. 1, no. 4, p. 0975 – 8887, 2010.

[29] Meftah.S., Abdul Rashid, "Recent Methods and Techniques of External Grading Systems for Agricultural Crops Quality Inspection – Review," *International Journal of Food Engineering,*, vol. 7, no. 3, 2011.

[30] V. Kumar, S. Sood and S. Mishra, "Introducing Effects in an Image: A MATLAB Approach," HAL, 2008.

[31] M. Sonka, V. Hlavac and R. Boyle, "Image pre-processing," in *Image Processing, Analysis and Machine Vision*, Boston, MA, Springer, 1993, pp. 56-111.

[32] R. Gonzalez and P. Wintz, "Digital Image Processing," in *Digital Image Processing, 2nd edition*, Addison-Wesley Publishing Company, 1987, pp. 22 - 26.

[33] Pratt, william k, Digital Image Processing 4th edition, California: A John Wiley & Sons, Inc., 2007.

[34] S. Saini, K. Arora, "A Study Analysis on the Different Image Segmentation Techniques," *International Journal of Information & Computation Technology.*, Vols. 4, 14, no. ISSN 0974-2239 Volume 4, Number 14 (2014), pp. , pp. 1445-1452, 2014.

[35] H. Luijten, "Basics of color based computer vision implemented in Matlab," *Journal of Business Ethics J BUS ETHICS*, no. 2005/01/01, 2005.

[36] Daniel Hailemichael Lemessa, "Development of Automatic Maize Quality Assessment System Using Image Processing Techniques," Addis Abeba , 2015.

[37] Marcelo Vassallo-Barco, Luis Vives-Garnique, Victor Tuesta-Monteza, Heber I. Mejía-Cabrera, " automatic detection of nutritional deficiencies in coffee tree leavesThrough Shape

And Texture Descriptors" *Journal of Digital Information Management.,* Volume 15, Number 1 (2017)

[38] G. Guo, H. Wang, D. Bell and Y. Bi, "KNN Model-Based Approach in Classification," no. 2004/08/08, 2004.

[39] J. Han, M. Kamber and J. Pei, Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, 2012.

[40] Coolen, Anthony CC, Reimer Kühn, and Peter Sollich, Theory of neural information processing systems, London: Oxford University Press, 2005.

[41] B. Klatt, B. Kleinhenz, C. Kuhn, C. Bauckhage, M. Neumann, K. Kersting, E.C. Oerke, L. Hallau, A.K. Mahlein, U. Steiner Stenzel and M. Röhrig , "SmartDDS-Plant Disease Detection via Smartphone," *EFITA-WCCA-CIGR* in *Sustainable Agriculture through ICT Innovation*, Turin, 24-27 June 2013.

[42] K. Wang, S. Zhang, Z. Wang, Z. Liu and F. Yang, "Mobile smart device-based vegetable disease and insect pest recognition method," *National Engineering Research Center for Information Technology in Agriculture*, vol. 19, no. 3, pp. 263-273, 2013.

[43] A.Meunkaewjinda, P.Kumsawat, K.Attakitmongcol and A.Srikaew, "Grape leaf diseases detection from color imagery using hybrid intelligent system," in *in Proceedings of ECTI-CON 2008 978-1-4244-2101-5/08/$25.00 c2008 IEEE*, 2008.

ANNEXES

ANNEX A: System code for main components

```
function varargout = GUI(varargin)

% GUI MATLAB code for GUI.fig

% Last Modified by GUIDE v2.5 24-Dec-2019 17:02:07

 % Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton',  gui_Singleton, ...

                   'gui_OpeningFcn', @GUI_OpeningFcn, ...

                   'gui_OutputFcn',  @GUI_OutputFcn, ...

                   'gui_LayoutFcn',  [] , ...

                   'gui_Callback',   []);

if nargin && ischar(varargin{1})

   gui_State.gui_Callback = str2func(varargin{1});

end

 if nargout

   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

else

   gui_mainfcn(gui_State, varargin{:});

end

function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
%%%%%%%   Load Image   %%%%%%

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
 [filename, path]= uigetfile({'*.jpg'}, 'pick image');
Im=imread([path, filename]);
      handles.Or_image = Im; %creating a global Variable


%%%%%%% Preprocessing  %%%%%%

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
Im_Resize =  handles.Or_image;%calling global var
I=imresize(Im_Resize, 0.25);
 [Binary_Image, W]= blue2 (I);


%%%%%%% Segmentation %%%%%%%

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
I =  handles.Im_Resized;%calling global var
S=size(rgb2gray(I));
 % Remove Background By calling blue2 function
[Binary_Image, W]= blue2(I);
Binary_Image= imfill(Binary_Image, 'holes');
% CAll Edge Segmentation Function
rangeF= rangefilt(Binary_Image);
```

```
se=strel('Diamond', 1);

dilatedR= imdilate(rangeF, se);

RemoveFalseRegions= bwareaopen(dilatedR, 400);

% Dameged Part Segmentation function // call Disease_Segmentation function

DiseasePart= Disease_Segmentation(I);


%%Concat EDGE and Disease Part

ConcatEDGEdisease=imsubtract(DiseasePart,RemoveFalseRegions);

ConcatEDGEdisease= im2bw(ConcatEDGEdisease);

 X=imresize(ConcatEDGEdisease,S);

SEE= strel('Diamond', 2);

X= imfill(X, 'holes');

X=imdilate(X, SEE);

Binary_Segmented = bwareaopen(X, 80);

MapOnOrginalImage=bsxfun(@times, W, cast(Binary_Segmented, 'like', W));


                %%% Prediction Using ANN %

function pushbutton6_Callback(hObject, eventdata, handles)

 seg_im_bw =  handles.final_seg_im_bw;%calling global var

im_seg_masked_rgb =  handles.final_seg_im_rgb;%calling global var

Image= handles.img; % calling global variable img

R= im_seg_masked_rgb(:,:,1);

G=im_seg_masked_rgb(:,:,2);

B=im_seg_masked_rgb(:,:,3);
```

```
Rm = mean2(R);

Gm = mean2(G);

Bm = mean2(B);


if (Gm >= Rm) && (Gm>= Bm)

   figure; imshow(Image); title('healthy');

else

[LL,nn]=bwlabel(seg_im_bw);


LLL=bwlabel(seg_im_bw);


%// Apply regionprops to the binary mask

s = regionprops(seg_im_bw, 'BoundingBox');


%// Create a cell array for the objects

objects = cell(numel(s), 1);


%// For each object...

for idx = 1 : numel(s)


   %// Get the bounding box property

   bb = floor(s(idx).BoundingBox);
```

```
   %// Extract out the object from the segmented image and place in cell array
   objects{idx} = im_seg_masked_rgb(bb(2):bb(2)+bb(4), bb(1):bb(1)+bb(3),:);
end
%// Create a cell array for the objects
bw_obj = cell(numel(s), 1);
lab_idx=1;
for each_lb=1:nn


      temp_label=(LLL==each_lb);
           %// Apply regionprops to the binary mask
           s = regionprops(temp_label, 'BoundingBox');


           %// For each object...
           for idx_bw = 1 : numel(s)
              %// Get the bounding box property
              bb_bw = floor(s(idx_bw).BoundingBox);
              %// Extract out the object from the segmented image and place in cell array
              bw_obj{lab_idx} = temp_label(bb_bw(2):bb_bw(2)+bb_bw(4),
bb_bw(1):bb_bw(1)+bb_bw(3),:);
           end
           lab_idx=lab_idx+1;
end
Neural_network ={};
charset = {'CBD','CLR','CLM','BES', 'GS'} ;
```

```
%###### case I using Nural network
load('Input.mat');
load('Target.mat');
size(Input)
size(Target)
net = patternnet(40, 'trainscg', 'crossentropy');
%net = newff(Input,Target ,[35], {'logsig'});

% Set up Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;
[net,tr,Y,E] = train(net,Input,Target);
%masking and Extracting Features for Prediction
single_cell_RGB_cell=cell(nn,1);%declaring cell to hold each segmented RGB cells
for cell_idx=1:nn
    single_cell_RGB_cell{cell_idx}= bsxfun(@times, (objects{cell_idx}), uint8(bw_obj{cell_idx}));
    %single_cell_RGB_cell(cell_idx)= bsxfun(@times, (objects(cell_idx)), uint8(bw_obj(cell_idx)));
end
%==
%nn=handles.nn;
 x=handles.x;
cell_index=x;
```

```
for cell_index=1:nn
    %axes(handles.axes1);
    %imshow(single_cell_RGB_cell{cell_index});
     Red_cell_i =single_cell_RGB_cell{cell_index}(:,:,1);
    Green_cell_i =single_cell_RGB_cell{cell_index}(:,:,2);
    Blue_cell_i =single_cell_RGB_cell{cell_index}(:,:,3);
     mean_R_cell_i=mean2(Red_cell_i);
    mean_G_cell_i=mean2(Green_cell_i);
    mean_B_cell_i=mean2(Blue_cell_i);
     mean_RGR_cell_i=mean2(single_cell_RGB_cell{cell_index});
%%%%%%%% l*a*b*
    LAB= rgb2lab(single_cell_RGB_cell{cell_index});
    L= LAB(:,:,1);
    A= LAB(:,:,2);
    B= LAB(:,:,3);
%Mean, Median and STD
    mean_l=mean2(L);
    mean_a=mean2(A);
    mean_b=mean2(B);
    median_l=median((median(L))');
    median_a=median((median(A))');
    median_b=median((median(B))');
    std_l=std((std(L,0,1))',0,1);
    std_a=std((std(A,0,1))',0,1);
```

```
        std_b=std((std(B,0,1))',0,1);
%%%%%%%% YCbCr
        LAB= rgb2ycbcr(single_cell_RGB_cell{cell_index});
        Y=  LAB(:,:,1);
        CB= LAB(:,:,2);
        CR= LAB(:,:,3);
        %Mean, Median and STD
        mean_y=mean2(Y);
        mean_cb=mean2(CB);
        mean_cr=mean2(CR);
        median_y=median((median(Y))');
        median_cb=median((median(CB))');
        median_cr=median((median(CR))');
        std_y=std2(Y);
        std_cb=std2(CB);
        std_cr=std2(CR);
        %--//// Texture feature extraction///-----///----////---
        cell_i_rgb=single_cell_RGB_cell{cell_index};
        cell_i_gray=rgb2gray(cell_i_rgb);
        T=statxture(cell_i_gray);
        %=====================================Texture feature extraction END
Val={double(mean_R_cell_i),double(mean_G_cell_i),double(mean_B_cell_i),double(mean_RGR_cell
_i),double(mean_l),double(mean_a),double(mean_b),double(median_l),double(median_a),double(
median_b),double(std_l),double(std_a),double(std_b),double(mean_y),double(mean_cb),double(m
```

```
ean_cr),double(median_y),double(median_cb),double(median_cr),double(std_y),double(std_cb),do
uble(std_cr),double(T(1)),double(T(2)),double(T(3)),double(T(4)),double(T(5)),double(T(6))};
  Test=Val;
  Test=cell2mat(Test);

  %%% Predict using a new data val
  load('coffeeNet.mat'); %%Load The Traind Neural Network
  a=sim(coffeeNet,Test')
  %b= net(test)
  classes = vec2ind(a)
  Neural_network{cell_index}=classes;
% Class counters
CBD()=0;
CLR()=0;
CLM()=0;
BES()=0;
GS()=0;
for i=1:nn
 if(classes==1)
        CBD= CBD +1;
  %text(ss(i).Centroid(1),ss(i).Centroid(2),'Coffee Berry Disease','Color', [1 1 0]);
    elseif(classes==2)
        CLR=CLR+1;
  %text(ss(i).Centroid(1),ss(i).Centroid(2),'Coffee Leaf Rust','Color', [1 1 0]);
```

```
   elseif(classes==3)
       CLM=CLM+1;
%text(ss(i).Centroid(1),ss(i).Centroid(2),'Coffee Leaf Mainer','Color', [1 1 0]);
   elseif(classes==4)
       BES=BES+1
%text(ss(i).Centroid(1),ss(i).Centroid(2),'Brown-eye spot','Color', [1 1 0];
   elseif(classes==5)
 % text(ss(i).Centroid(1),ss(i).Centroid(2),'Green Scale','Color', [1 1 0]);
```

Signed Declaration Sheet

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by Advisor:

Name: _____

Signature: _____

Date: _____