ADDIS ABABA UNVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

COFFEE DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK: AN IMAGE PROCESSING APPROACH

A Thesis submitted to Addis Ababa University School of information science and system program in Partial Fulfillment of the Master of Information Science Degree Requirements

BY

HAYMANOT TADDESE WORKU

ADVISOR

MILLION MESHESHA (PhD)

November, 2021

Addis Ababa, Ethiopia

ADDIS ABABA UNVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

COFFEE DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK: AN IMAGE PROCESSING APPROACH

A Thesis submitted to Addis Ababa University School of information science and system program in Partial Fulfillment of the Master of Information Science Degree Requirements

BY

HAYMANOT TADDESE WORKU

Members of the Examining Board's Names, Date and Signatures

Advisor Name	Date	Signature
MILLION MESHESHA (PhD)	_________________	_________
Examiners Name	**Date**	**Signature**
1. Martha Yifiru (PhD)	_________________	_________
2. Melkamu Beyene (PhD)	_________________	_________

Declaration

I declare that this thesis is my original work and has not been submitted as part of any university's degree requirements.

As a university advisor, I approved the submission of this thesis for examination.

Name of Advisor: MILLION MESHESHA (PhD)

Advisor Signature: _____________________

Date: _____________________

Acknowledgment

First and foremost, I would like to express my deepest gratitude to the Almighty God, for his blessing of successfully completion the thesis. After that, I would like to thank my advisor Dr. Million Meshesha, for his advice and constructive guidance throughout this thesis. The discussions with him always made me think that things are possible.

I'm also grateful to Mr. Awole Beshir, a plant science expert and researcher at the Southern Agricultural Research Institute's Bonga Agricultural Research Center, for providing expert advice on the disease that affects coffee leaves and beans, as well as assisting in obtaining images of coffee leaves and beans affected by the disease in the production area. I'm also grateful to Mr. Mohammed Kedir, who works at the Agaro Agricultural Research Sub Center as a plant science expert. My heartfelt gratitude goes to Abay Mulisa for his unlimited encouragement on this thesis. Finally, I would like to thank my family, who always supported me throughout this thesis.

Table of Contents

List of Figures

List of Tables

List of Acronym

Acronym	Acronym Definition
AARC	Agaro Agricultural Research Sub Center
ANN	Artificial Neural Network
BARC	Bonga Agricultural Research Center
BPNN	Back Propagation Artificial Neural Network
CBD	Coffee Berry Disease
CLR	Coffee Leaf Rust
CLS	Cercospora leaf spot
CWD	Coffee Wilt Disease
CNN	Conventional Neural Network
CPD	Coffee Phoma Disease
DIP	Digital Image Processing
DL	Deep Learning
DLDR	Deep Learning Disease Recognition
ECX	Ethiopian Commodity Exchange
FCL	Fully Connected Layer
FCN	Fully Convolutional Network
GLCM	Gray Level Co-occurrence Matrix
HPCCDD	Homogeneous Pixel Counting Technique for Cotton Diseases Detection
JARC	Jimma Agricultural Research Center
TBDR	Texture Based Disease Recognition
RBF	Radial Basis Function
ReLu	Rectify Linear Unit
KNN	K-Nearest Neighbors
SOM	Self-organizing Map

Abstract

Coffee is one of the most important products in Ethiopia. Coffee has a great contribution in Ethiopia economy since it increases foreign currency of the country; and is the source of daily income earning for farmers. Therefore, controlling coffee diseases and ensuring quality of coffee product is the major issue for the country. Currently disease identified manually by experts and they identify by eye, so this is makes challenged and expert's not available in everywhere in production area and other researchers don't see Cercospora leaf spot and coffee berry disease. The aim of this research is therefore detecting common coffee diseases using digital image processing and deep learning technique.

In this study, we consider the most common coffee diseases such as Cercospora leaf spot, coffee phoma disease and coffee berry disease. Convolutional Neural Network has showed its efficiency and accuracy on image processing in representing images and creating patterns to identify coffee diseases. This research proposed Convolutional Neural Network technique to detect coffee leaf and coffee beans diseases. This study follows experimental research methodology. 552 coffee leaf and coffee beans images dataset captured by HD camera and Motorola Phone from popular coffee production areas of Ethiopia, such as Jimma (agaro) and Bonga (kefa) zone farm and 5334 coffee images collected from Jimma Agricultural Research Center (JARC) and Bonga Agricultural Research Center (BARC) database. We have used four-classes for classification; namely, Cercospora leaf spot, coffee phoma disease, coffee berry disease and Healthy coffee. The total number of data sets used for experimentation is 5886. From the total data sets, 80% is used for training and the remaining 20% for testing purpose.

Experimental result shows that the proposed model detects the disease with 96.1 % accuracy. This is a promising result towards designing a model that can be used for automatic coffee disease detection. As a future work, we would like to recomendede the model to recognize other coffee parts stems and roots with large amount of images.

Keywords: *Conventional Neural Network, coffee Leaf and coffee Beans Disease, Image processing, Augmentation.*

CHAPTER ONE
INTRODUCTION

1.1. Background of the study

Ethiopia is the world's fifth largest coffee producer and Africa's top producer, with estimated coffee production of more than 450,000 tons and marketable supply of 334,000 metric tons in farm year 2012/13 [1]. Ethiopians consume half of the tons, and the country leads the continent in domestic consumption. The major markets for Ethiopian coffee are the EU (about half of exports), East Asia (about a quarter) and North America. The total area used for coffee cultivation is estimate to be about 4,000 km^2. The exact size is unknown due to the fragmented nature of coffee farms. The way of production has not changed much, with nearly all work, cultivating and drying, still done by hand. The revenues from coffee exports account for 10% of the annual government revenue, because of the large share, the industry is giving very high priority, but there are conscious efforts by the government to reduce coffee industry's share of the GDP by increasing the manufacturing sector [2].

In Ethiopia coffee is one of the major export crops; Coffee export contributes to the incomes of more than 5 million smallholder farmers. However, these smallholders find themselves at the bottom of a long value chain that includes collectors, traders, processors and exporters. In 2008, a marketplace, the Ethiopia Commodity Exchange (ECX), was launched with the aim of improving agricultural marketing and shoring up producer prices. In the ECX, registered buyers and sellers meet to agree on a price for agricultural commodity lots. The ECX trading floor is a physical space, located in Addis Ababa, but commodity lots are inspected, graded and stored in ECX-controlled warehouses across Ethiopia's main agricultural production areas. Suppliers store the lots against receipts, which they then auction electronically [3]. Ethiopian Federal Government Tea and coffee Authority, handles anything related to coffee and tea, such as fixing the price at which the washing stations buy coffee from the farmers [4].

As shown in Table 1-1, as shown in Table 1-1, Ethiopia is the largest producer and consumer of coffee in the region. Ethiopian coffee production has increased substantially over the last three years and is expected to reach 7.62 million bags (457,200 MT) in 2021/22 if growing conditions are favorable. Domestic consumption accounts for 50-55 % of Ethiopia's output. In 2020/21, local consumption is expected to climb to 3.55 million bags. Coffee production is expected to reach 7.62 million bags (457,200 MT) in 2022/23, up 20,000 million bags from the present crop projection for 2020/21, assuming favorable weather and appropriate rain. In comparison to our forecasts, the crop estimate for 2020/21 has been revised upwards to 7.6 million bags [5]. The following table shows increasing Productivity of coffee each year.

	2017/18	2018/19	2019/20	2020/21	2021/22 *
Production (tons)	423.6	423.6	448.5	456	457.2
Production area (1000 ha)	532	532	538	540	541
Productivity (ton/ha)	0.8	0.8	0.83	0.84	0.85

* This indicates Forecasted amount of coffee production.

Table 1-1. Ethiopia's Coffee Production estimate with respect to area [5]

As shown in Figure 1-1 below, coffee Arabica is mostly grown in the forest areas of the southwestern highlands of the Jimma, Kaffa and Buno districts. Around 400,000 hectares area of Ethiopia is under coffee cultivation.

Figure 1-1. Common Coffee growing areas in Ethiopian [6]

The country produces almost 200,000 metric tons of coffee every year. 95% of the coffee is produced in the forest area and is claimed to be organic. A major part of the Ethiopian coffee is exported in green coffee beans form, to the Rest of the World. Therefore, this factor gives a key advantage to Ethiopian coffee in the international market [7] [8]. Figure 1-2 shows coffee production with metric ton in each year in Ethiopia.

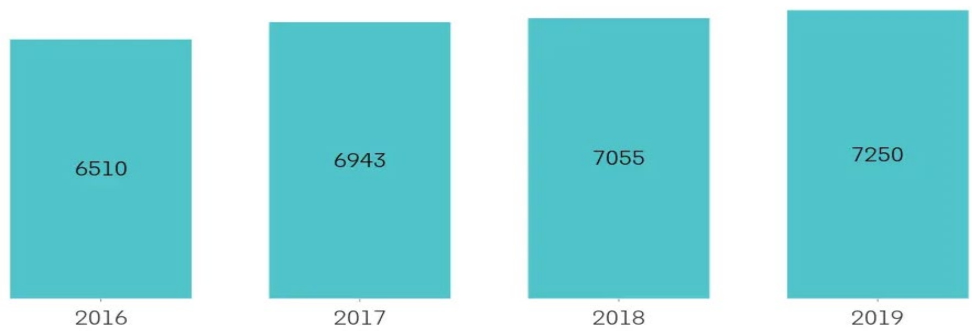


Figure 1-2. Graph of coffee production, Ethiopia [8]

Convenience is the foremost factor driving the instant coffee market, as it is easily prepared compared to fresh coffee. The haste lifestyle of the consumers is aiding the market for convenient food. The majority of the retail channels equips the distribution of the product. The instant coffee market is highly fragmented due to the presence of global players in the market. Companies are competing with other companies through joint ventures, partnerships, and product launches in order to stay in the market. Example, Nescafé Gold sachets were introduced in June 2019. Instant coffee brands popular in the region includes the Nescafe, Lavazza, Moccona, and Robert Timms. So, this indicates major coffee production [8]. Instant coffee is made out of 100 % coffee beans that have been roasted, ground, and boiled with water into a liquid before being dehydrated, giving it the same health advantages as ground coffee beans. Despite the differences in flavor, the antioxidant content of both drinks is nearly same. Figure 1-3 shows that which available coffee product type in Ethiopian market.

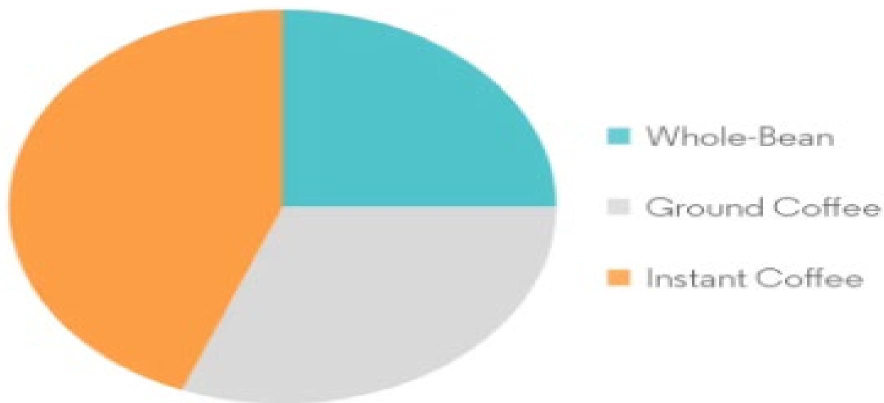


Figure 1-3. Coffee market, market share, by product type in Ethiopia, 2019 [8]

Cooper's Cask Coffee Company, Forest Coffee, Ethiopian Yirgacheffe Coffee, and Square One Coffee are just a few of the prominent participants in the Ethiopian coffee market. Starbucks, for example, has mostly concentrated on partnerships and cooperation with other emerging market participants, as well as new product introductions. [8].

Arabica coffee (Coffea Arabica) and Robusta coffee (Coffea Caneophora) are the two most common commercial coffee species, followed by Liberica coffee (Coffea Liberica) and Excelsa coffee (Coffea Excelsa) [9].The Arabica coffee is originated in the highland forests of southwestern Ethiopia whereas that of Robusta coffee in the wild forest of humid hot lowland areas of Middle East, Central, and West Africa. In southwestern part of Ethiopia, about 400,000 ha of an ancient forest where coffee occurs as understory shrubs still remain [10]. Moreover, there are also high genetic diversities of coffee in the region that are used as source of plant stock for the selection of disease resistance, high yields and top quality in terms of aroma as well as flavor [11].

Even though agriculture is Ethiopia's most important economic sector, the land is under-utilized by conventional production systems, which leads to a variety of structural issues. Image-based disease recognition is a major difficulty in the agriculture sector when it comes to extracting and analyzing crop and plant quality, especially in coffee plants. Is challenged to identify disease from image because of images have more elements and is have more pixels to recognize those pixels is challenged [12].

1.1.1. Digital Image Processing

Digital image is composed of a finite number of elements, each of which has a particular location and value. The digital elements are picture elements, image elements and pixels. Pixel is the term used most widely to denote the elements of digital image. An image is a two-dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene. An image is a projection of a 3-D scene into a 2D projection plane. An image may be defined as a two-dimensional function f (x,y), where x and y are spatial (plane) coordinates, and the amplitude of *f* at any pair of coordinates (x,y) is called the intensity of the image at that point [13]. Figure 1-4 shows that how images are preprocess.

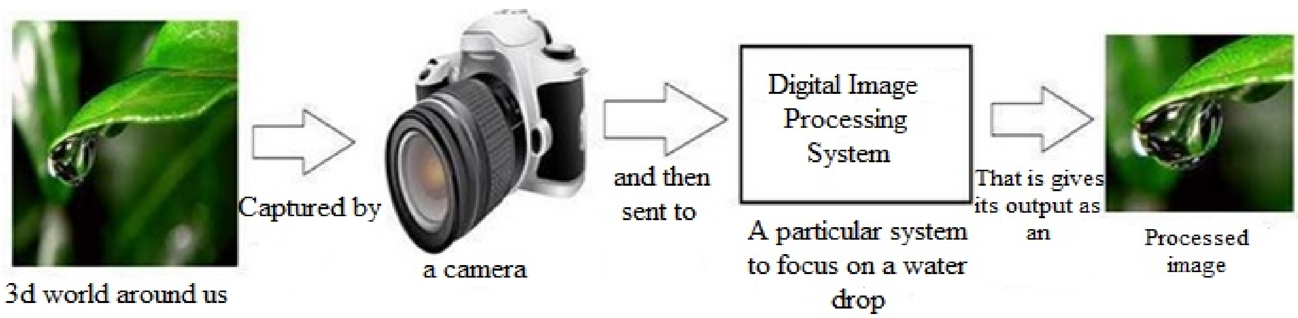


Figure 1-4. How Digital Image processing works [13]

The term gray level is used often to refer to the intensity of monochrome images. Color images are formed by a combination of individual 2-D images. For Example, in the RGB color system, a color image consists of three (Red, Green and Blue) individual component images. For this reason, many of the techniques developed for monochrome images can be extended to color images by processing the three components of images (Red, Green and Blue) individually. An image may be continuous with respect to the x- and y- coordinates and in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized [13].

1.1.2. Steps in Digital Image Processing

Digital image processing passes through the following steps [14].The first step is image acquisition which is the principal central advance in image processing which gives an idea regards to the root of digital images. In addition, this stage includes the pre-processing undertaking, for example, image scaling. Acquisition of image is very testing challenge. Because to detect identify images it needs quality images, so, it requires proper gadget like scanner or good resolution camera.

Image acquisition is followed by pre-processing. This step is a procedure to change an input image onto digital frame and performed a couple of process, with the true objective to get a redesigned picture or to separate some significant data from it. It is a sort of signal dispensation which input is image, like video edge or photo and result may be picture or characteristics related with that image. The image pre-processing includes filtering, color conversion and detail enhancement of image [14]. Image pre-processing is the major procedures to improve image quality in order to extract appropriate features from image for the purpose of object recognition. The main goal of pre-processing is noise suppression (usually the origin of the noise is digitizing and transmission), removal of distortion given by the scanning device, eventually suppress or highlight other attribute, which are important for the following tasks, such as segmentation, edge detection and feature extraction [15].

Image segmentation meant a way towards portioning image into its tremendous sections or objects. It is a standout amongst the most troublesome errands in computerized image processing. It is used to find desired objects. Segmentation implies segment of picture into assorted piece of same skin tone or having some resemblance analysis implies distributing image into various piece of same components or having some similarity. The segmentation ought to be conceivable using distinctive Algorithms like Otsu' methodology, k-means clustering, conversion of RGB image into HIS mode interest, image segmentation is applied, and pattern recognition based on multi-Level thresholding approach [14].

The feature extraction technique plays an important role in image classification. The features are the main parameter that are involved for classification of image. Texture extraction is determined as the example of information or course of action of the structure with random interval. Accordingly, texture attributes and appearance of object size, shape, thickness, characterization, extent of its basic properties. A fundamental stage to accumulate such features through texture extraction called as texture component extraction. Sequentially, the importance of texture data, texture component extraction is a core limit in various image-processing applications like remote detecting, biomedical imaging and object-based image [14].

Color extraction is an important factor of distinctive classes. When image detected needs image processing, one of image processing steps is color extraction and images display different color disease symptoms. So, to use RGB color extraction is important to identify image colors. An image pixel typically addressed in the RGB space, in which the color space at each pixel addressed as a combination of RGB. Other color spaces like the HIS and CIE color space model mostly used in various other segmentation procedures where their benefits and constraints analyzed announced and examined. It is understood that Euclidean separation of the distinct color are proportional to the variation that human visual impact over CIE Lab color space [14].

Shape extraction is a general descriptor, for example, object count, region of the shape, image dimension, and zone of picture are essential object to depict the shape of an image. Those qualities are utilized to remove include the sore and level of the injury. Mass Analysis utilized in this examination to compute insights for marked regions in a de-noised data, for example, quantity of the protest, region, and edge [14].

Edges in image are the portions with solid boundaries and object with one pixel then onto the following can make genuine assortment in the picture quality. Edge detection is an image processing strategy for finding the limits inside the corresponding image. It works by distinguishing discontinuities in brightness. Edge recognition utilized for image segmentation and information extraction in regions, for example, picture processing, computer vision, and machine vision [14].

1.2. Motivation of the study

Coffee is one of the most important commodities in the international agricultural trade, representing a significant source of income to several countries of Africa, Asia and Latin America [16]. Ethiopia is a leading Arabica coffee producer in Africa, ranking the fifth largest Arabica coffee producer and tenth in coffee export worldwide. Its total coffee production and export respectively increased by 107% and 226% for the crop year 2009/10 and 2010/11 [17]. These are primarily credited to various problems, including inadequate access to improved production and processing technologies, together with deficient services, poor market access and lack of incentives.

In addition, coffee leaf and coffee beans infected by different disease, that is the problem and to identify coffee diseases from infected coffee images we apply digital image processing. So this study is motivated to apply image processing and machine learning for detecting coffee diseases from the images of coffee seed, stem and/or leaf in order to come up with the solution that can enhance the quality of coffee production.

1.3. Statement of the problem

Coffee diseases cause considerable losses when not properly treated and 57% yield loss was observed by the infection of disease-causing organisms on coffee crop [18]. The most economically important pathogenic coffee diseases are coffee berry disease (CBD), coffee wilt disease (CWD) and coffee leaf rust (CLR), and physiological disorder like coffee branch die back is caused by pseudomonas syringe and non-pathogenic agents. Similarly, CBD and branch dieback were causing high yield loss of coffee production. In the same way, insect pests such as Anthestia bug and coffee blotch miner are the major ones causing considerable damage [18].

Nowadays, experts identify coffee diseases with the naked eye. According to the interview made with experts: First, when they plant the coffee, they saw the seeds that are resistance to coffee disease. Second, when coffee grows it may be affected by wilt disease. If experts and farmers observe such Wilt disease, they cut down the entire tree and burn it. Also, whenever they are in doubt of the diseases occurred, they take sample from existing disease and test it to identify through laboratory. The challenge in the current disease identification mechanism is that, they iterate until they are sure about the type of diseases tested in the laboratory. The process usually takes time, prone to error and in efficient, as a result of which the disease greatly affects coffee production. This in turn decreases efficiency and effectiveness of coffee production in the country and a major threat on the foreign currency earnings of the country.

According to coffee expert interview,the following diseases commonly happen on Kefa and Agaro production area: Coffee Berry Disease (CBD), Coffee Leaf Rust (CLR) and coffee wilt Disease (CWD) and insects. CBD affects coffee beans, CLR affects the part coffee leaf , CWD is commonly affect root of the coffee and coffee leaf infected by insects. Insects are that affect coffee leaf parts.

Local and foreign scholars conducted to explore the problem on coffee diseases. Abrham Debasu et.al. [12], evaluated four types of classifiers (ANN, KNN, Naïve and combination of RBF and SOM) for recognition of Ethiopian coffee plant diseases such as Coffee Leaf Rust (CLR), Coffee Berry Disease (CBD), and Coffee Wilt Disease (CWD) from coffee leaf. Based on experimental result, they recommend for further research and improvements on identification of Ethiopian Coffee diseases type by exploring more features not only on their leaf but also on their stem and roots. One of the drawbacks of the research is the attempt they made to identify CWD from coffee leaf, while it affects coffee stem, according to experts comment and CBD is coffee beans disease but as the researchers write it is Coffee leaf disease.

It is therefore the aim of this study to investigate and construct a model that detects and identifies coffee diseases from input coffee leaf and coffee beans images using image processing and Deep learning technologies. Other researchers implement image processing to identify the images whether infected or healthy on coffee leaf. But this study build model that categorized the disease from coffee leaf and coffee beans image using image processing and deep learning technique.

To this end, in this research an attempt is made to explore and answer the following research questions.

- ✓ What method could be used to detect Cercospora leaf spot (CLS), coffee phoma disease (CPD) and coffee berry disease (CBD) of coffee leaf and coffee beans?
- ✓ To what extent the model performs in classify coffee disease?

1.4. Objective of the study

1.4.1. General objective

The main objective of this research is to investigate the design and development of an automatic Cercospora leaf spot, coffee phoma disease and coffee berry disease identification model by using deep learning approaches.

1.4.2. Specific objectives

To achieve the general objective of this research, the following specific objectives are formulated.

- ✓ Review literature to understand concepts, principles and technologies of digital image processing, convolution neural Network and recurrent neural network.
- ✓ Collect healthy and infected coffee leaf and coffee beans images directly from coffee production area.
- ✓ Identify and design suitable Convolutional neural network model to detect the disease.
- ✓ Evaluate the performance of the model by using unseen data.

1.5. Scope and limitation of study

The Arabica coffee is originated in the highland forests of southwestern Ethiopia whereas that of Robusta coffee in the wild forest of humid hot lowland areas of Middle East, Central, and West Africa [19]. Therefore, the current research is targeted towards identifying diseases in Arabica coffee type and mainly concentrated on coffee Leaf and coffee beans diseases that are common in our country, such as Cercospora leaf spot, coffee phoma disease, and coffee berry disease. Because of coffee Arabica is most type of product in Ethiopia and the disease Cercospora leaf spot and coffee berry disease are not seen in the previse researches. Accordingly, we collect images of coffee leaf and coffee beans from Ethiopian popular coffee growing regions, such as Jimma (Agaro zone) and Bonga (Kefa zone) regions.

This study did not consider those diseases affecting coffee root and Coffee stem and hence collecting coffee root and coffee stem images is out of the scope of the study. As an input, both infected and healthy coffee leaf and bean images are used for training and constructing the model. The study uses CNN architecture for learning and coffee disease detection.

This study design model to detect coffee leaf (Cercospora leaf spot (CLS) and Coffee Phoma Disease (CPD) and coffee beans coffee berry disease (CBD). so, that it makes different from other researches. When we collect coffee image data was not seasonally that disease display on coffee parts, but we can collect available data from farm and databases so, that this is limitation of this study.

1.6. Significance of the study

We all agree with the fact that technology has become to stay and there is no way we can avoid its use. Therefore, we must use technology to improve our living conditions. Deep learning and Machine learning is one of the current technologies used for the detection of coffee diseases from coffee leaf, stem and seed image. This research is enabling agriculture experts to appreciate the importance of image processing in the field of agriculture. Therefore, experts can easily identify coffee disease using image processing. If the disease differs before the disease spread mean that if early detected the disease helps them to take action with experts before a severe damage happens. In addition, farmers can get quality coffee production. Also, as the quality coffee production increases, the country will get high foreign currency and will become a priority area for coffee export. The study design and develop model that detected the disease automatically, so it has benefits in terms of produce quality coffee, consuming time.

The study has benefits for other researchers to be motivated to do researches in applying image processing and deep learning.

1.7. Methodology of the Study

Methodology is the step-by-step procedure followed in conducting a given research. Selecting the right procedures is critical for the success of the study under investigation. Hereunder we provide methods and techniques used to conduct the current study.

1.7.1. Research Design

For conducting Digital Image Processing (DIP) research, this study follows experimental research. Experimental research is a study that strictly follows a scientific research design. It includes variables that can be measured, calculated and compared. Most importantly, experimental research is complete in a controlled environment. The researcher collects data and conduct experiments to select more effective algorithms for constructing an optimal model.

Experimental research seeks to determine a relationship between two variables: The field of digital image processing is broad, spanning digital signal processing techniques as well as picture-specific approaches [20]. A picture can be thought of as a function of two continuous variables, x and y, called f (x, y) f as color, brightness of image. So, this study uses this variable on coffee image. After completing an experimental research study towards conducting experimental research, there is a need to prepare data set, select implementation tools and algorithms and evaluate the proposed prototype. When we conduct a study and measure the dependent variable, like calculating model accuracy.

1.7.2. Research Process

To achieve the objective of this thesis, the following research process depicted in figure 1-4 is followed. As shown in the figure, this study is conducted with three main phases. The first process is concerned with identifying the domain of the problem; that means understanding the problem by reviewing different kinds of literature. The second process is about data collection and preparation. 552 images Data was capture from Jimma and Bonga farm places, the most popular coffee production areas and was the disease labeled by experts. Also, 5334 images were collected from Jimma Agricultural Research Center (JARC) and Bonga Agricultural Research Center (BARC) database. The collected images were labeled by experts and that captured images it added in each classis. Then, to increase our model training image data we use Image_data_generator augmentation using python code like rotation range, width shift range, height shift range... The third process is about implementation of the coffee diseases detector, where CNN model has more Architecture, from those AlexNet Architecture is constructed by creating a sequential model and adding Convolutional, Normalization, Pooling, Dropout and Activation layers at the appropriate positions. Conv2D and relu, MaxPooling2D are used for learn the feature Flatten fully connected Layer and softmax are used for classification. After that, evaluate the model by comparing the accuracy and loss by plotting the graph for training and validation. And we test the model using unseen image data.

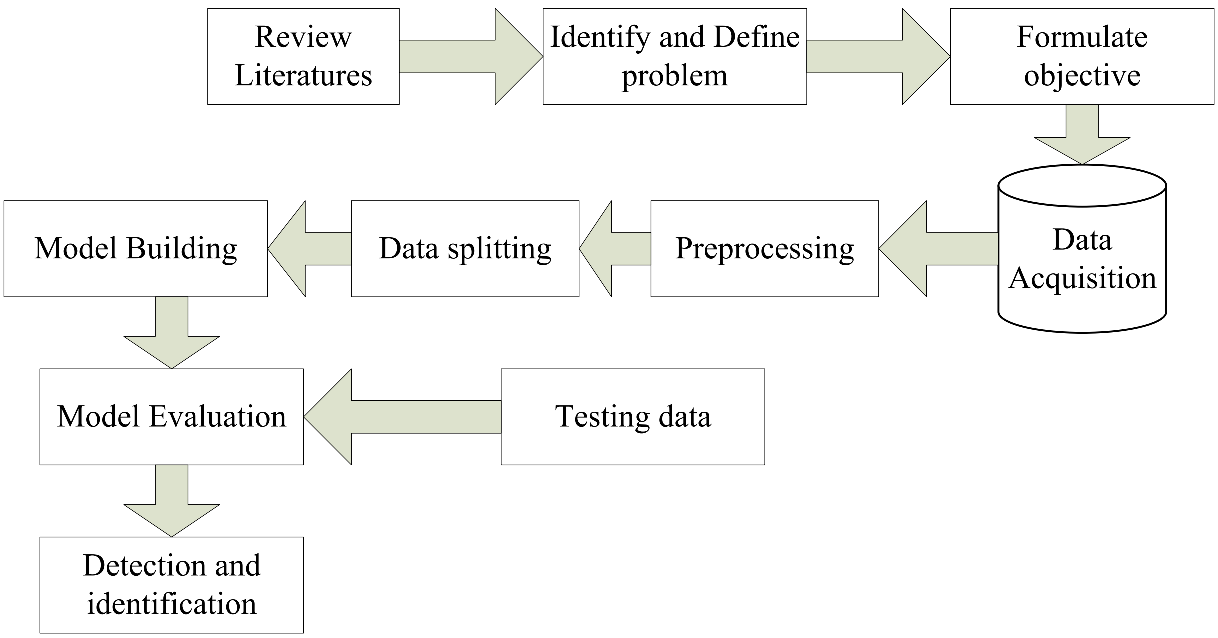


Figure 1-5. Research process

1.7.3. Data Preparation

1.7.3.1. Source of Data

Data set is collected from well-known coffee production area of Ethiopian region such as Jimma (Agaro zone) and Bonga (Kefa zone). 5334 labeled Images data collected from Jimma Agricultural Research Center (JARC), Bonga Agricultural Research Center (BARC) and Agaro agricultural research sub center (AARC) database. And the images were labeled by experts. 552 labeled Images are captured from coffee farm using HD 360 x 360 pixels camera and 48 MP Motorola phone.

Also captured images labeled by experts. Totally, 5886 both infected and healthy data sets collected and we categorized in to the following classes (normal, CBD, CLS, CPD). From the total of data sets 4743 we used for training purpose and 1143 for testing purpose. 166 images data used for validation from training data. When images were taken, the camera was fixed on a stand, which reduces the movement of hand and capturing uniform images of coffee leaf and coffee beans image. After capturing image, we standardize image resize to 256 x 256 dimensions using python (keras) tools. The required image data was collected at the end of July, 2021 G.C.

1.8.Implementation Tools

Python programming language and Anaconda Jupyter notebook are used for this research. Because nowadays python suitable for image processing tool and it is the state-of-the-art programming language means that it is recent stage in development and newest technology tool rather than Weka and MATLAB tools. It is a powerful language for image preprocessing and analysis. We need common libraries like OpenCV packages and tensororflow.

1.8.1. Software Tools

For image processing and illness recognition in coffee Python 3.9.5 and Anaconda3 2020 were chosen on the Windows platform since Python is appropriate tool for image processing and deep learning. As a result, the python and anaconda3 tools were used to display, edit, process, analyze, and recognize coffee illnesses. In order to identify the best tool for implementing the CNN algorithm for coffee disease identification and classification, a review of various software tools and their libraries is carried out. During our inquiry, we discovered that there exist tools that are both general and tailored to deep learning algorithms. Before choosing the tools, we considered a few characteristics that would aid in the selection of the suitable software tools and libraries. The choice of programming language to implement the algorithm is the most important criterion

The tools must also be utilized in machines with limited resources. Other requirements include selecting tools with sufficient learning materials, such as literature, free video tutorials, and books, and the tools must be used in computers with low resources (like CPU only). Python as a programming language with Tensor-Flow and Keras libraries on an Anaconda environment were utilized to create the CNN algorithm. These tools meet all of the criteria for consideration and are written in Python, which we are familiar with.

- ✓ **Anaconda3:** is used for the implementation of the model and it is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment. It contains different IDE which are used to write the coding part such as Jupyter Notebook and Spyder. We have used Jupyter notebook to implement the coding part. It is easy and runs in a web browser.

- ✓ **Keras** is a model-level library, providing high-level building blocks for developing deep learning models. It does not handle itself low-level operations such as tensor manipulation and differentiation. Instead, it relies on a specialized, well-optimized tensor library to do so, serving as the "backend engine" of Keras. Rather than picking one single tensor library and making the implementation of Keras tied to that library, Keras handles the problem in a modular way, and several different backend engines can be plugged seamlessly into Keras. Currently, the three existing backend implementations are the TensorFlow backend, the Theano backend, and the CNTK backend. In the future, it is likely that Keras will be extended to work with even more deep learning execution engines [21].

- ✓ **TensorFlow**, CNTK, and Theano are some of the main platforms for deep learning today. Theano is developed by the MILA lab at University de Montréal, while TensorFlow is developed by Google, and CNTK is developed by Microsoft. Used to create Deep Learning models [21].

- ✓ **Adobe Photoshop CS5:** Adobe Photoshop CS5 is a popular image editing program that works in a similar way to Adobe Illustrator, Adobe in Design, Adobe Photoshop, and other Adobe Creative Suite programs. This study uses Adobe Photoshop CS5 for resized captured images data.
- ✓ **Microsoft Visio:** This tool is used for designing the research process and architecture of the proposed system.

1.8.2. Hardware Tools

We used canon EOS 5d3 with 3.2-inch LCD with 1,040,000 dots, HD 360 x 360 pixels digital camera used to capture the sample images from the field. To implement the CNN algorithm with the selected software with CPU Intel(R) Core (TM) i5-5437U CPU @ 2.40GHz - 2.70GHz processor, RAM 8 GB, 1.5 TB hard disk is used.

1.9.Evaluation Methods

One of the performance evaluation methods is Accuracy. One of the most commonly used metrics while performing classification is accuracy. The accuracy of a model (through a confusion matrix) shows an overall performance of the system proposed by the study [22]. Because of this study have the different number of samples in each class, accuracy is a useful statistic.

Accuracy can be misleading if used with imbalanced datasets, and therefore there are other metrics based on confusion matrix which can be useful for evaluating performance. There are several evaluation metrics for classifiers such as precision, recall, and the F-1 score [23]. Precision is about determining how often predictions is it correct as per the defined classes. Recall is on the other hand concerned with the truth-ness of the prediction as a base. F-Score is a weighted average of the true positive rate (recall) and precision.

1.10. Organization of the Thesis

This thesis is organized as follows. In chapter one includes introduces statement of the problem, objective, scope, significance and methodology of the study. Chapter two presents, reviews different international journals, thesis reports, and all issues related to history of coffee, coffee leaf diseases, digital image processing. Also related works is reviewed to show the extent to which the problem is solved. In chapter three we discuss about the proposed architecture by the study, in addition, algorithms and methods are discussed. Chapter four discusses experimental results of the proposed model in detecting coffee disease with analysis and interpretation. Finally, concludes the thesis by conclusion and recommending some future work. It also shows some research directions that can use in the future to improve coffee leaf diseases recognition system.

CHAPTER TWO
LITERATURE REVIEW

This chapter focuses on coffee plant, diseases that affected coffee as well as digital image processing techniques and detection approaches and algorithms. The Chapter also provides more detail on the diseases affecting coffee leaf, coffee beans and symptoms of the diseases. This chapter focuses on image processing techniques that aid to identification of the diseases and reviewed the existing related literatures.

2.1. Ethiopian Coffee

Plant is which grows in all over the world mostly in Ethiopia. In Ethiopia, agricultural sector plays a central role in the economic and social life of the nation. Around 80 to 85 % of people in Ethiopia are dependent on agriculture; among 80 to 85 %, about 40% of the sector contributes from cultivation of coffee [24].

Ethiopia is the birthplace of coffee, which was discovered earlier in the world than the rest of the world. When David Beatty was a goatherd in Ethiopia's southwestern highlands, he discovered the Ethiopian area where coffee originated more than 1,000 years ago. Coffee is referred to as kaffa. Nobody knows when coffee was first found as a beverage plant, but it is thought to have been cultivated and used in Ethiopia as early as the 9th century. Around AD 575, it began cultivating Yemen. While it originated in Ethiopia, it went to Yemen some 600 years ago and continued its trek throughout the world from there. Among the many legends, Kaldi, an Abyssinian goatherd, who lived around AD 850 found the origin of coffee. Ethiopians rely on it for their cultural and socioeconomic well-being; it generates 25% - 30% of the country's foreign exchange, half of GDP, 90% of exports, 85% of total employment, and is an integral element of the country's culture; around 50% of the coffee produced is eaten internally. The review's overall goal is to better understand Ethiopia's coffee production and marketing value chain system, with a specific goal of revising the value chain, amount of production, domestic and international marketing, and consumption level [25].

Ethiopia produces some of the world's best highland coffees, and coffee is the country's main economic crop. Its coffee is almost entirely *arabica*, which is a variety of coffee native to Ethiopia and widely grown in Latin America. Other parts of Africa, on the other hand, grow *robusta coffee*, which bears both flowers and fruits throughout most of the year, whereas *arabica coffee* has a distinct and brief harvesting season. Coffee thrives at elevations of 1000 to 2000 meters, and it may be found growing wild in various places of Ethiopia, while the majority of Ethiopian coffee is grown in the Sidamo, Ilubabor, Gamo Gofa, Welega, Harerge, and the southern and western regions of Kefa. Coffee is grown on around half a million hectares, with peasants producing 98 % of the crop on smallholdings of less than a hectare and commercial (public and private) farms producing the remaining 2%. The southern and eastern regions of the country have bimodal rainfall distributions, while the western part has a monomodal distribution. This distribution system allows the country to harvest coffee at various periods throughout the year, ensuring a steady supply of fresh coffee all year. External inputs like as fertilizers, herbicides, and fungicides are used extensively in coffee production in the country [26].

Coffee *(Coffea spp.)* is found across the tropics and comes in over 70 different species, many of which are native to Africa. The Arabica (*Coffea arabica,* 64 % of world production) and Robusta (*Coffea canephora, var. Robusta,* 35 %) varieties are the most economically important now, with 10.3 million hectares farmed worldwide. The most important producers are Brazil, Vietnam, Colombia, Indonesia, and Ethiopia. More than 60 % produce and export the crop, making it one of the most valuable cash crops in the developing world [27].

2.2.Coffee Leaf Disease

2.2.1. Cercospora Leaf Spot (Brown Eye Spot)

Cercospora leaf spot is a fungus that occurs on leaves when plants are under stress. The fungus can develop both in seedbeds and after plants have transplanted into bags. It is the most common nursery disease and a sign of poor management. The symptoms are brown spots on leaves gradually expanding with reddish brown margin and Spots on both sides of the leaf (see figure 2-1). Brown Eye Spot causes brown rounded necrotic lesions, with dark center, surrounded by a yellow halo. The affected leaves fall rapidly and the branches dry, causing a reduction in plant productivity and lowering the fruit’s quality. This disease also causes intense leaf fall and poor plant growth [28] [29].

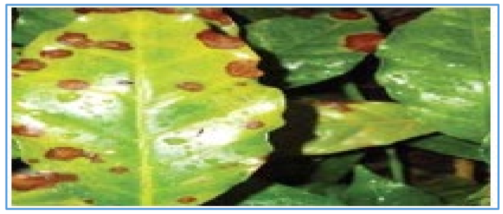


Figure 2-1, Nursery plants affected with Cercospora [29]

2.2.1. Phoma Leaf Spot

Phoma leaf spot, caused by the fungus *Phoma tarda* (Stewart) Boerema & Bollen, is one of the most important coffee diseases in Brazil and has caused significant loss of quality and productivity of coffee crops. Symptoms observed are tip and branch dieback, necrosis of rosettes, mummified berries and leaf spots. Among the control methods, it is recommended the planting in areas less subject to cold winds, installation of windbreaks, balanced fertilization with nitrogen (N), calcium (Ca) and micronutrients, as well as chemical control when necessary [30].

Figure 2-2. Phoma leaf spot disease [29]

2.3. Coffee Beans Disease

2.3.1. Coffee Beery Disease (CBD)

Coffee Berry disease is caused by a mutant pathogenic strain of Colletotrichum coffeanum Noack which has spread throughout tropical Africa from its point of origin in Kenya. Fruit is attacked in all stages, and total destruction of the crop can be caused. The disease was confined for many years to the higher altitudes, because suitable conditions of temperature and humidity were infrequent in the lower ones. Following the inception of a wetter and cooler climatic phase in East Africa in 1961, coffee in the lower altitudes is now attacked [31]. This disease has the following Symptoms; Dark sunken lesions on green berries; berries dropping from plant; mummified berries and the Cause is Fungus.

Figure 2-3. Coffee Berry Disease [29]

2.4. Lists of Common Coffee Pests and Disease

The following table describe common coffee pest and disease category, symptom of the disease, cause and how we manage the diseases.

No	Disease Category	Symptom	Cause	What we suggest	How to manage
1.	Bacterial blight; Pseudomonas syringe	Water-soaked areas on leaves that dry out and turn brown and necrotic with yellow halos; necrosis of shoot tips that spreads quickly along branches; leaves that turn black and die but remain attached to the tree.	Bacterium	Disease can be transferred over large distances by sick seedlings moving throughout the field, or within the field by water splash; bacteria can enter the plant through wounds.	Copper-based protective sprays should be applied to the plants immediately before the rainy season begins and should be continued through the short rains.
2.	Fungal; Cercospora leaf spot (Brown eyespot, Berry blotch) Cercospora co_eicola.	Brown patches on leaves that expand and develop a gray white core with a red-brown rim; lesions may be encircled by a yellow halo or appear burned if there are a lot of them; Infected leaves may drop prematurely	Fungus	Wind, water splash, and human activity through fields, especially when plants are moist, can all transmit disease.	Ensure that the crop is properly fertilized, as disease-prone plants are nutrient-deficient. To avoid inoculum buildup, remove all crop debris from the field after pruning. Plant spacing and pruning to open up

No	Disease Category	Symptom	Cause	What we suggest	How to manage
		from the plant; lesions on green berries are brown and sunken, with a reddish rim; lesions on infected red berries are big black sunken regions.			the canopy promotes optimal air circulation and disease resistance. If disease does develop, copper fungicides can be used to control it if they are accessible.
3.	Coffee berry disease (CBD) Colletotrichum kahawae.	Green berries with dark deep blemishes; fruit falling from the shrub; mummified berries	Fungus	Diseases that can wipe out up to 80 % of a crop's yield.	Spraying copper-containing fungicides on the plants can help control the illness; any sick berries should be removed from the plants; and resistant types are available and should be planted in affected regions.
4.	Coffee leaf rust Hemileia vastatrix.	Small, pale yellow spots on upper leaf surfaces are followed by powdery orange-yellow lesions on the undersides of leaves; symptoms often begin on the lower leaves of the plant and spread; infected leaves drop from the plant, and twigs and branches become defoliated.	Fungus	When the spores reach a leaf, they use the rough side of their spines to cling to the surface. The presence of liquid water on the leaves is required for the spores to germinate. A temperature of 17 to 25°C (62.6 to 77°F) is also recommended,	Varieties that are resistant commercially grown coffee has lost much of the genetic variety of its wild predecessors due to monoculture practices. Unfortunately, outside of Ethiopia's evolutionary center, wild coffee has lost much of its genetic variety owing to the

No	Disease Category	Symptom	Cause	What we suggest	How to manage
				with 22°C (71.6°F) being ideal. Rainstorms can wash the spores from the leaves, preventing infection. When the conditions are right, the spores form long tubes called germ tubes that migrate across the leaf looking for stomata.	consequences of deforestation
5.	Pests Category: Insects Black twig borer Xylosandrus compact us.	Wilting and yellowing of foliage, especially at the ends of twigs and branches (called "flagging"); a pin-sized hole can often be found on the underside of flagging stems or twigs where the insect has entered the plant; twigs and stems are hollowed out and can be seen by cutting open the affected tissue; twigs and stems are hollowed out and can be seen by cutting open the affected tissue; Adult beetles are little and	Insect	Beetle damage encourages bacteria and other fungi to infest the plant, and adult beetles overwinter in the plant.	Infested twigs and stems should be trimmed out and destroyed; sagging branches should be clipped back a few inches from the start of symptomatic areas; appropriate fertilizer and moisture can help plants recover quickly from pruning injury.

No	Disease Category	Symptom	Cause	What we suggest	How to manage
		black, measuring around 2 mm in length, and are infrequently seen; eggs and pupae are creamy white.			

Table 2-1. Common coffee pests and disease [29]

2.5. Deep learning

Deep learning techniques are a type of machine learning that excels at dealing with unstructured data. Current machine learning techniques are outperformed by deep learning techniques. It allows computational models to learn features from data at various levels progressively [32]. Because of the increased availability of high-performance computing, deep learning approaches based on deep neural networks have become increasingly popular. When dealing with unstructured data, deep learning achieves greater strength and flexibility due to its capacity to analyses a huge number of features. Deep learning algorithm passes the data through several layers; each layer is capable of extracting features progressively and passes it to the next layer. Initial layers extract low-level features, and succeeding layers combines features to form a complete representation [32].

Since 1989, various advancements in CNN architecture have been made. Parameter optimization, regularization, structural reformulation, and other improvements are examples of these advancements. The main drive in CNN performance improvement, however, appears to have come from the rearrangement of processing units and the design of new blocks. The majority of CNN architectural advancements have been in the area of depth and spatial exploitation. CNNs can be divided into seven categories based on the sort of architectural alterations used, including spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention based CNNs. The taxonomy of CNN architectures is depicted in the following figure 2-4 [33].

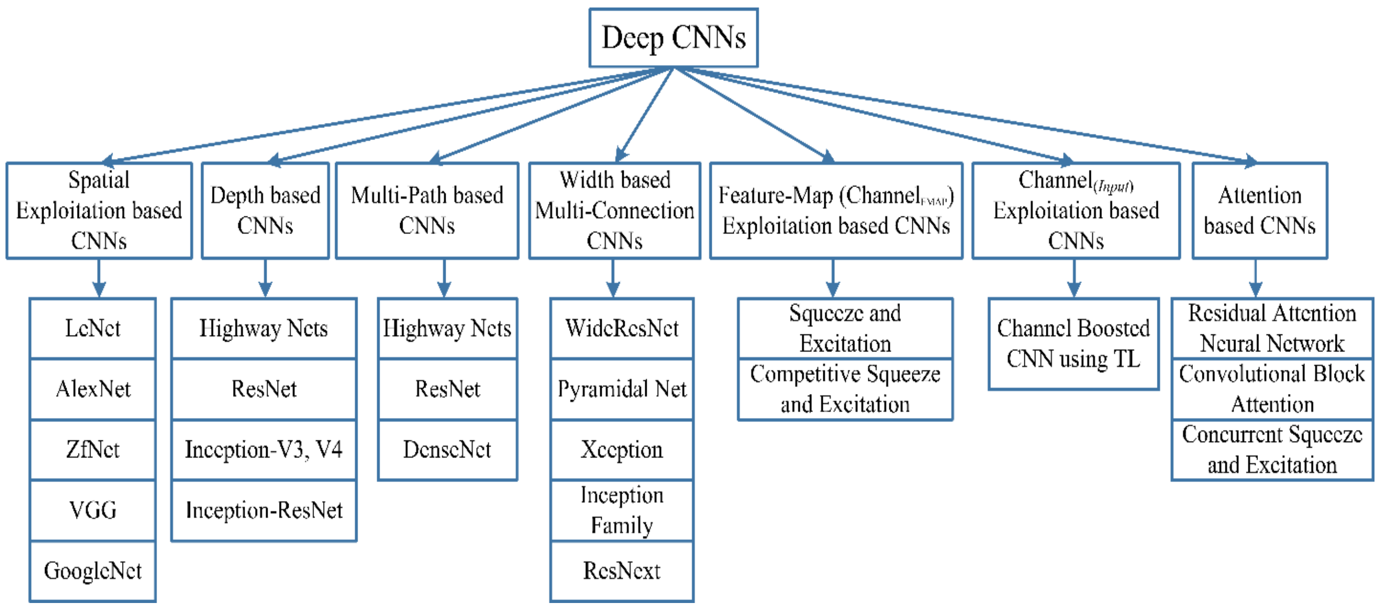


Figure 2-4. Deep CNN architectures classified into seven main groups in this taxonomy [33]

2.6. Digital Image Processing

An image is defined as a two dimensional function, f(x,y), where x and y are spatial (plane) coordinates, and the amplitude of 'f' at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. A digital image is a 2D representation of a scene as a finite set of digital values, called picture elements or pixels. The field of digital image processing refers to processing digital image by means of a digital computer [34].

$$f(x, y) = \begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & 2 & & N-1 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ \\ M-1 \end{array} & \begin{bmatrix} f(0, 0) & f(0, 1) & f(0, 2) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & f(1, 2) & \cdots & f(1, N-1) \\ \vdots & & & & \vdots \\ f(M-1, 0) & & \cdots & & f(M-1, N-1) \end{bmatrix} \end{array}$$

Figure 2-5. Matrix Dimension [34]

Digital image processing is the use of computer algorithms to perform image processing on digital images. The objective of image analysis is to create accurate image of an area viewed by satellite sensors. Digital image processing encompasses four major areas of computer operation [35].

1. Image restoration or preprocessing; computer routines to correct a degraded digital image to its intended form, usually a precursor to the steps that follow.
2. Image enhancement; to improve the detectability of objects or patterns in a digital image for visual interpretation.
3. Image classification; quantitative decision rules classify or identify objects or patterns on the basis of their multispectral radiance values (as such, the output is analogous to an image map requiring little or no visual interpretation).
4. Dataset merging; computer routines integrate multiple sets of data from same location such that congruent measurements can be made.

2.7. Artificial Neural Network (ANN)

Artificial Neural Networks are based on the functioning of the nervous system in the living beings. These networks are similar to the biological neural networks. The following figure shows the modelled neuron used for artificial neural network [36].

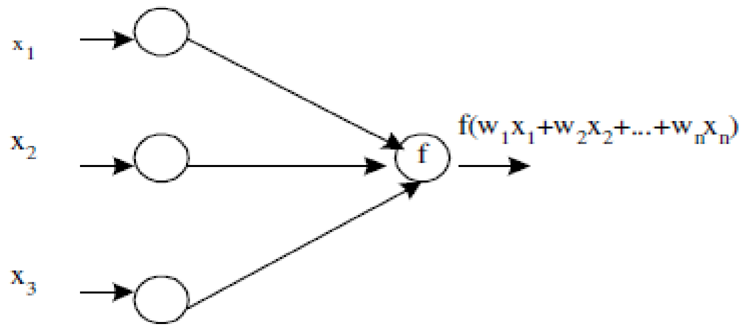


Figure 2-6. Artificially modeled Neuron [36]

Artificial Neural Networks are inspired from the human brain and the network of neurons present in the brain. The information is processed and passed on from one neuron to another through neuron synaptic junctions. Similarly, in artificial neural networks there are different layers of cells arranged and connected to each other. The output/information from the inner layers of the neural network are passed on to the next layers and finally to the outermost layer which gives the output. The input to the outer layer is provided non-linearity to inner layers' output so that it can be further processed. In an Artificial Neural Network, activation functions are very important as they help in learning and making sense of non-linear and complicated mappings between the inputs and corresponding outputs [37].

2.7.1. Neural Network

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another [38].

A neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns [39].

The goal of an artificial neuron is to emulate the behavior of a biological neuron, which is to accept a variety of signals, xi, from many distinct nearby neurons and process them in a per-defined straightforward manner. The neuron j decides whether or not to fire an output signal yj based on the results of this processing. If it is activated, the output signal can be either 0 or 1, or any real value between 0 and 1. From a historical standpoint, the function f(x), which calculates the output from the m-dimensional input vector x, is thought to be made up of two pieces. The first part evaluates the so called 'net input', net, while the second one 'transfers' the net input Net in a non-linear manner to the output value y [40].

2.7.2. Activation Function

In artificial neural networks, activation functions are employed to convert an input signal to an output signal, which is then sent as input to the next layer in the stack. We calculate the sum of products of inputs and their corresponding weights in an artificial neural network, then apply an activation function to it to produce the output of that layer, which we then feed as the input to the next layer. The number of layers and, more crucially, the type of activation function used in a Neural Network determine its prediction accuracy. There is no manual that specifies the minimum or maximum number of layers that should be used to improve the accuracy and outcomes of neural networks, but a thumb rule suggests that at least two layers should be employed. There is also no indication of the type of activation function to be employed in the literature. Studies and research have shown that utilizing a single/multiple hidden layers in a neural network minimizes prediction error [37].

2.8. Convolutional Neural Network (CNN or ConvNet)

Convolutional Neural Networks (CNNs) have proven to be adept at finding structure in raw image data. Historically, the image modeling field has been dominated by excessive amounts of preprocessing techniques to get the input images aligned and transformed into a form that modeling techniques could better handle. Slight variations in rotation or scale made image processing a hard task. CNNs have made it possible to let the network handle the raw image data and let the practitioner focus on adjusting the network architecture.

The goal of a CNN is to learn higher-order features in the data via convolutions. They are well suited to object recognition with images and consistently top image classification competitions. They can identify faces, individuals, street signs, platypuses, and many other aspects of visual data. CNNs overlap with text analysis via optical character recognition, but they are also useful when analyzing words as discrete textual units. They're also good at analyzing sound.

The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning [41].

Convolutional neural network works based on basic neural networks, which is describe above. So what does the CNNs change? There are several variations on CNNs layers architecture: Convolutional Layer, Pooling Layer and Fully-Connected Layer. Fully-Connected Layer is just acting as neural network, which we have already covered in, previous. CNN algorithm has two main processes: convolution and sampling, which will happen on convolutional layers and max pooling layers [42].

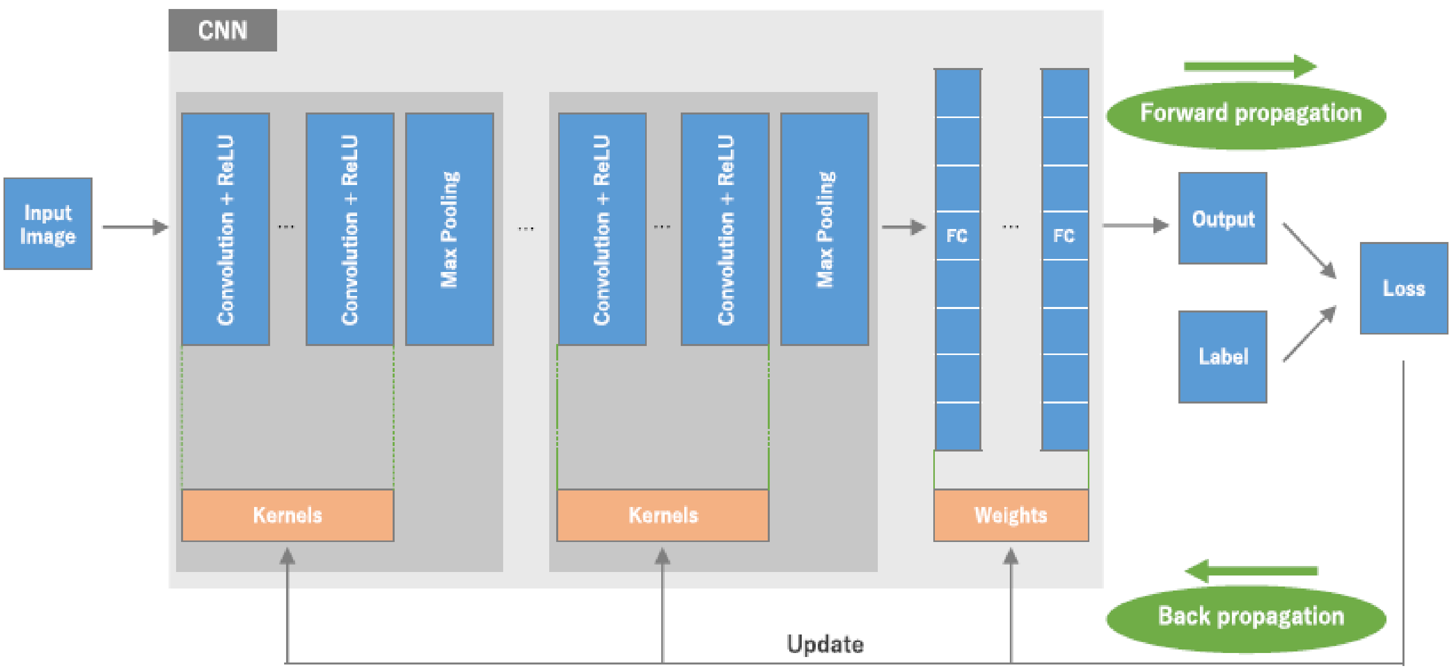


Figure 2-7. An overview of CNN architecture and the training process [42]

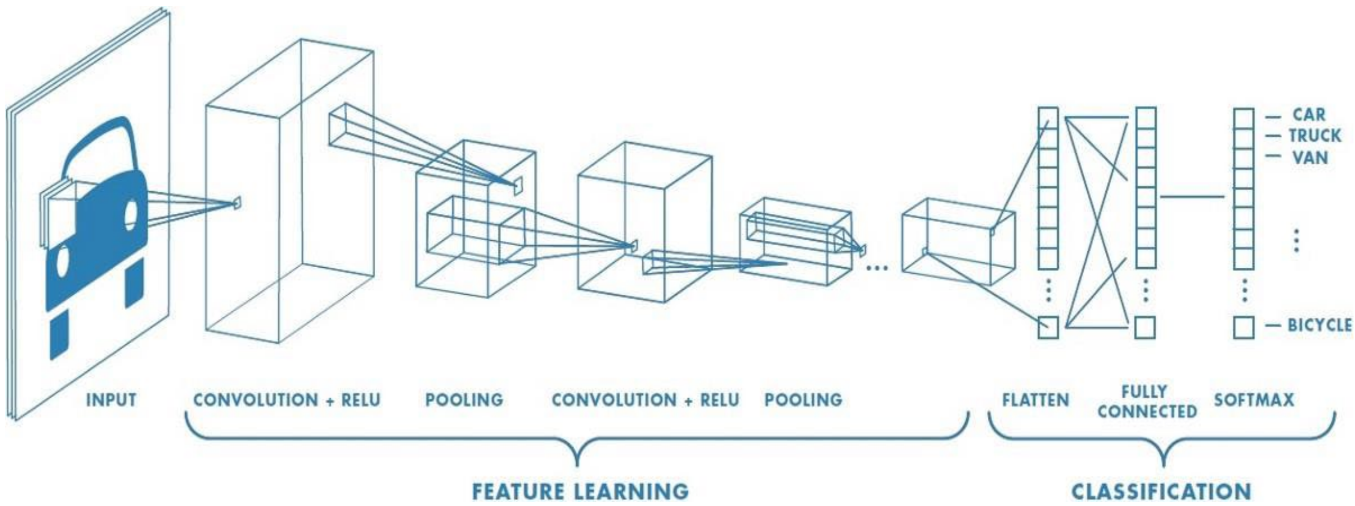


Figure 2-8. CNN Architecture [42]

This thesis, use CNN model to detect and classify the disease from the coffee images by input healthy and infected images. For the process of classification, CNN is used which is composed of various sequential layers and every layer of the algorithm transforms one volume of activation to another using different functions.

Convolutional neural network is one of the most popular ANN. It is widely used in the fields of image and video recognition. Convolution and a mathematical concept is almost similar to multi-layer perceptron except it contains series of convolution layer and pooling layer before the fully connected hidden neuron layer.

There are three layers to it [43].

- ✓ Convolution layer: It is the primary building block and perform computation computational tasks based on convolution function.
- ✓ Pooling layer: it is the arranged next to convolution layer and used to reduce the size of inputs by removing unnecessary information so computation can be perform faster.
- ✓ Fully connected layer: It is the arranged to next to series of convolution and pooling layer and classify input into various categories.

A. Convolution Layer

Convolutional Layer is used to Extract features from a given input images. Images represented by matrix of pixels. The images have RGB (Red, Green and Blue) channels. The RGB image represented as three 2D-matrices staked over each other color and each image having a pixel value of a range of 0 to 255. Convolution layer is formed from a combination of a set of convolutional filters (aka kernels or feature detectors) which are small matrix values with size like 3×3, 5×5, and so on [44].

The main objective of the convolution layer is to extract useful features from the input image. Every image is represented as a matrix of pixel values in a computer. The image captured by quality and standard digital camera; images have three channels: Red, Green, and Blue (RGB). This type of image is represented as three 2D-matrices staked over each other (one for each color) and each having a pixel value of a range of 0 to 255. Convolution layer formed from a combination of a set of convolutional filters [45] [46].

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Input

Filter

Figure 2-9. Filter and input volume example [43]

B. Pooling Layer

To reduce the number of parameters, to extract dominant features in some spatial location, to progressively reduce the spatial size of the convolved feature, and to control the problem of over fitting in the network we need to add pooling layer (also called subsampling or down sampling) in between some successive convolution layers in CNN [45] [46]. This layer helps to reduce the computation power that is required to train the network. The pooling operation is performing by sliding the filter on the convolved feature

.

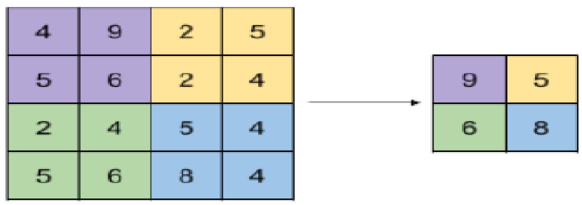


Figure 2-10. MAX Pooling Layer Example [43]

There are three types of polling: Max pooling, Average polling, and less commonly used type, which is Sum, pooling. Max polling is the most commonly used polling operation and its output is the maximum value from the portion of the image covered by the filter. The average pooling returns the average of all the values from the image covered by the filter and finally, the sum pooling returns the sum of all the values from the portion of the image covered by the filter. It performs de-noising along with dimensionality reduction but average polling only used for dimensionality reduction. Therefore, max polling is better than average pooling. The pooling operation applied in all of the depth slices of the image after the convolution operation, commonly used filter is 8×8, and stride 2 but we can change accordingly. For example, if we take the commonly used 8×8 filter, for the max pooling, it returns the maximum value from the four values [45] [46].

C. Fully Connected Layer (Dense layer)

In a fully connected layer, every neuron in the previous layer connected to every neuron in the next layer. This layer accepts the output of the convolution or pooling layer, which is high-level, features of the input volume. These high-level features are in the form of a 3D matrix but the fully connected layer accepts a 1D vector of numbers. Therefore, we need to convert the 3D volume of data into a 1D vector called flattening and that becomes the input to the fully connected layer. The flatten vector is given to the fully connected layer and it performs mathematical computation like any ANN. Activation functions such as ReLU in the hidden layers are used to apply non-linearity in these layers. By using sigmoid activation function the output layer of the fully connected layer, perform classification based on the training data. In this thesis, the image classification will have two classes: Diseased and Healthy [47] [46].

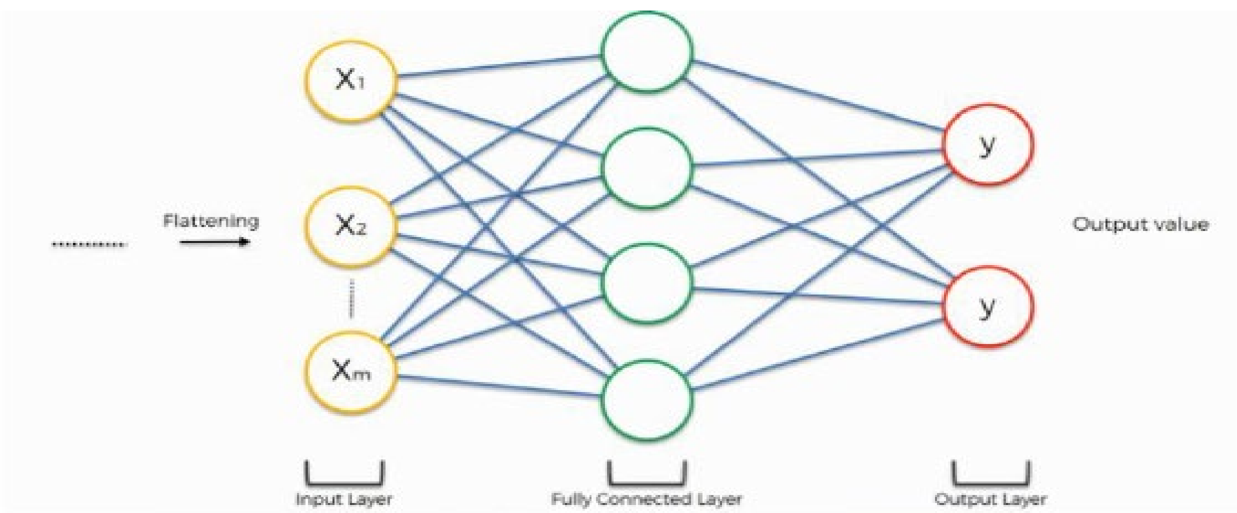


Figure 2-11. Fully Connected Layer Example [43]

2.9. Related Works

An Automatic Coffee Plant Disease Recognition is presented in [48]. Brown Eye spot and coffee leaf rust disease were the focus of the research. Cercospora, Rust, and healthy leaf are the three experimental output classes. Text Based TBDR (Texture Based Disease Recognition) and Deep Learning Disease Recognition (DLDR) are the two methodologies they use. Statistical attribute extraction algorithms are based on image Gray Level Co-Occurrence Matrix (GLCM), while TBDR uses texture attribute vectors as input to a neural network classifier combining statistical and local binary characteristics. GLCM is a matrix that counts the number of surrounding pixels with the same gray level that occur when a reference pixel with its own gray level is taken into account. Each element of GLCM is defined by a specific pair of gray level values seen in the reference pixel and its neighboring pixel in the original image. Local binary pattern has eight neighbors and uses a small 3x3 portion on the central pixel. An LBP pattern is made up of a series of binary values, and a uniform pattern is defined as a collection of LBP patterns with only one transition in the binary values sequence. DLDR applies deep learning directly on the sample photos using a convolutional neural network. For each experiment, the Kappa coefficient and sensitivity rates were computed to give data for comparisons. Using DLDR, the researcher selects the strategy with the best 98 % accuracy out of all of them [48].

A machine learning technique presented in [49] to recognize Coffee Leaf Rust (CLR), Coffee Berry Disease (CBD), and Coffee Wilt Disease (CWD) coffee plant disease. Totally the study used 9100 coffee leaf data sets, 6370 data sets used for model training and 2730 data sets used for testing purpose. The data sets are collected from Ethiopian Southern Nations, Nationalities, Jimma and Zegie zons. The study used eleven features from coffee leaf, Five GLCM and six color features. In this research work they evaluate four types of classifiers namely; ANN, KNN, Naïve and combination of RBF and SOM for recognize Ethiopian coffee plant diseases from coffee leaf. In line with this, in combine of RBF and SOM. RBF computes and the output is given to SOM. And RBF and SOM has a better performance than the other classifiers. After testing the four different classifiers, extremely Randomized 90 % of accuracy from RBF and SOM algorithms [49].

A paper presented in [50] for Automatic Identification of Plant Diseases using Convolutional Neural Network with Limited Data. And they compare Three CNN architectures (ResNet18, ResNet34 and ResNet50) are compares in this study. The study used two datasets for training and testing, namely the Plant Village dataset and the coffee leaf dataset. Plant Village Dataset has 54,305 leaf images of 14 crop species and 26 diseases distributed among 38 crop-disease pairs. This dataset contains clear images of plant leaves and each image contains only one leaf. The study develops four approaches Transfer Learning (Baseline and Baseline++), Metric Learning Using Triplet Network and Deep Adversarial Metric Learning. The study achieved a very high accuracy of 99% for new classes when the source and target domain data are captured under the same condition and a reasonable accuracy of 81% for novel dataset that is captured under different conditions [50].

No	Author and title	Statement of the problem	Methodology (Approaches)	Result	Research gap
1.	Lucas ximenes Boa sorte et.al. 2019 [48].	Coffee Leaf Disease Recognition Based on Deep Learning and Texture Attributes	Texture Based Disease Recognition (TBDR) and Deep Learning Disease Recognition (DLDR) approach	The researcher selects best result with DLDR approach 98 % of accuracy	✓ The study focusses only two namely; Cercospora, Rust with two approaches. ✓ Not including other part of coffee plants. Like coffee beans and coffee stem.
2.	Abrham Debasu et.al. 2016 [49].	Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques	Imaging and machine learning techniques. ANN, KNN, Naïve and a hybrid of Self-organizing Map SOM and Radial basis function (RBF) are used	Achieved 90 % using SOM and RBF with texture color features	✓ The study focuses on three major type of coffee disease which occurs on the leaf part of a coffee plant, (CLR, CBD and CWD). One of the drawbacks of the research is the attempt they made to identify CWD from coffee leaf, while it affects coffee stem. ✓ The study doesn’t mention testing class from which class tested the model. Means there is no healthy classes.
3.	Ahmed Afifi, Abdulaziz Alhumam, Amira Abdelwahab (2020) [50].	Convolutional Neural Network for Identification of Plant Diseases with Limited Data	Employed four approaches: transfer learning, triplet networks, & deep adversarial metric learning	Approaches for classifying plant diseases that learn from little data and achieved high accuracy of 99%	✓ They only compare only ResNet18, ResNet34, and ResNet50 Architecture.

Table 2-2. Summary of related works

2.10. Summary

As mentioned in related works, local and international scholars show that automatically coffee disease detection has been widely used in the field of agriculture especially for coffee plant disease identification with best model performance evaluation results. More specifically machine learning and deep learning algorithms such as ANN and CNN are implemented. Based on such scholars Coffee berry disease, phoma disease and coffee rust disease are commonly affected coffee plants. Computer vision techniques are also applied for the detection and classification of different diseases in the coffee plant including Coffee berry disease, phoma disease and coffee rust disease, but there is still a need to develop a more accurate and efficient model with different part of coffee plant.

As we see in the Section 2.9 all previously conducted papers have some problems which we need to overcome in this thesis. For instance, most of the papers used their datasets from internet searches or publicly available databases such as in Plant Village, Kaggle. Most of scholars focus on coffee leaf rather than coffee fruits and stem. Most importantly the methods used by previously conducted research works are not state of the art for instance, other previse scholars model implement on MATLAB tools. Most researchers are don't identified the disease from coffee plant parts like the disease are must categorized with coffee leaf part, coffee fruit parts.

And the other main point of this thesis is there is build model to detect the disease from coffee leaf and coffee beans using deep learning techniques to detect Cercospora leaf spot (CLS) and coffee phoma disease (CPD) from coffee leaf and coffee phoma disease (CBD) from coffee beans by comparing VGG19 and modified AlexNext model.

CHAPTER THREE

PROPOSED ARCHITECTURE AND METHODS

This chapter focuses on the description of approaches and methods of data collection that are used in order to accomplish this thesis including methods to implement the model, data collection, data preparation, what we use software and hardware configuration of the system used. Finally, the model evaluated. Different experiments carried out by using different dataset ratio, different input layer. In addition, many experiments are conducted by using different activation function.

3.1. Proposed Architecture

The proposed Architecture is depicted in figure 3-1. It has two phases; the first phase is concerned with training parts. The first step is image data collection, then we make preprocess the image and also as the same time we split image data in to training and test data. After that we augment from training data, then we implement feature extraction. Feature extraction is a step in the dimensionality reduction process, which divides and reduces a large set of raw data into smaller groupings. In a digital image or video, features like shape, edges, and motion are important. The second step is testing phase. On this phase first we need testing data from unseen then preprocess the data, after that model detect coffee leaf or coffee beans. Finally, the model classifies the image is healthy, CBD, CLS and CPD. The overall model architecture depicted in Figure 3-1.

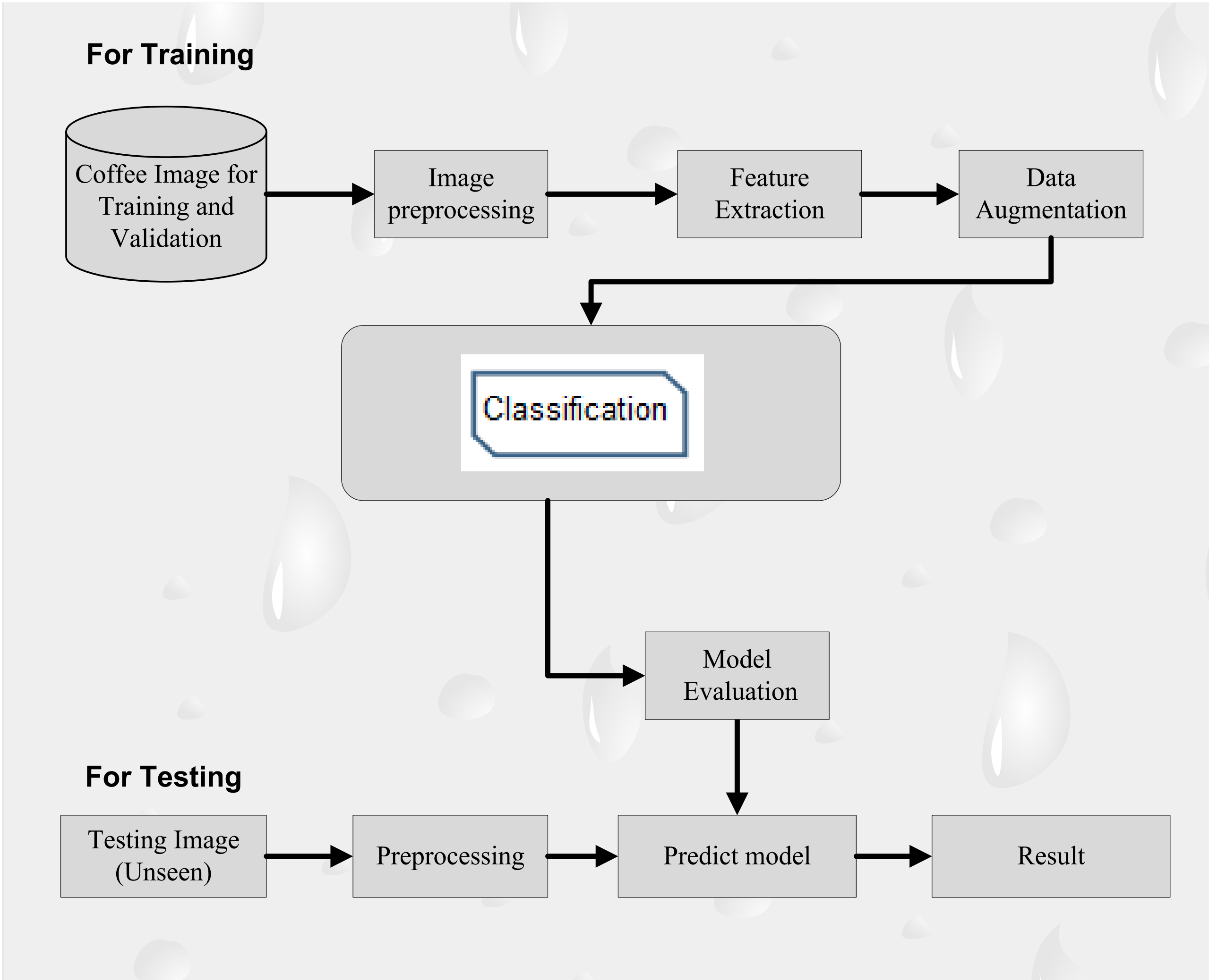


Figure 3-1. Proposed model Architecture

3.2. Image Preprocessing

Image preprocessing are the steps taken to format images before we used by model training and inference. This includes, resizing, positioning, and color corrections. The performance of two preprocessing procedures, namely Contrast adjustment and Banalization operation, is compared in this study. In addition, image data is transformed before being fed into a neural network or deep learning algorithm. Cropping, scaling, and color conversion are all examples of pre-processing. The dimensions of the acquired photographs differ. As a result, we have 445 * 345 and must resize the photos to a uniform 256 By 256 dimension in order to process them efficiently. The study uses 256 x 256 default image size on pre-processing part. To resize an image to the size DEFAULT IMAGE SIZE we defined earlier, we utilized the method convert image to array. Label binarizer is also used for class label pictures. Because noises increase inaccuracy in identifying coffee plant illnesses, pre-processing of images is widely employed to remove low frequency background noise, normalize the intensity of individual particles on a given image, remove reflection, and mask a segment of image. On photos of coffee plants, median filtering was applied to reduce noise.

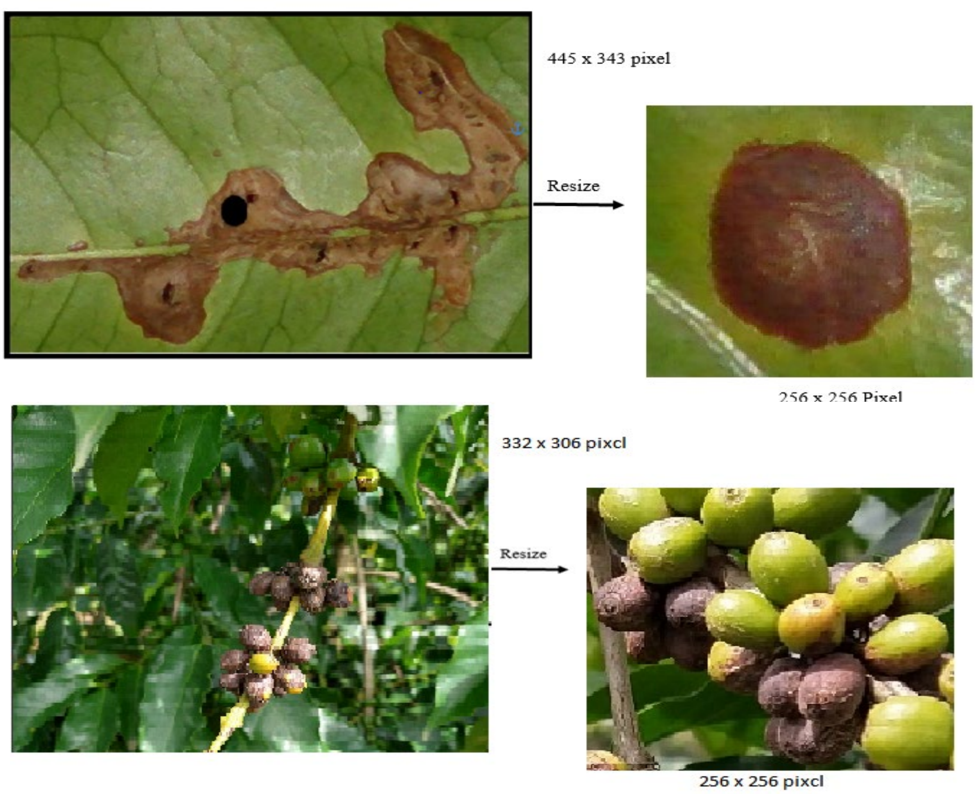


Figure 3-2. Image resize and cropping

3.3. Data Augmentation

We need a large data collection for deep learning. As a result, we employ data augmentation. Data augmentation is the practice of creating new data from an existing training sample in order to increase the number of training data points in a dataset. It's critical to expand the amount of data points. It's also used to extract complicated features from data while avoiding the problem of over-fitting. The solution to the problem of limited data is data augmentation, which is a data-space solution. Data Augmentation is a term that refers to a range of strategies for increasing the size and quality of training datasets so that stronger Deep Learning models may be generated using them. Various data augmentation techniques were used on the original photos in this thesis to obtain additional images for our data set. Those are rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip and fill_mode are among the image augmentation technologies explored in this survey. Data augmentation is useful to improve performance and outcomes of deep learning models by forming new and different examples to train datasets.

Data augmentation can be done both before and after the data is fed into our model (Image-Data-Generator augmentation) and during the training process. Using Keras libraries, data augmentation is conducted during network training in this thesis. The original image generates each image that was delivered into the network during the training. This can be accomplished in Keras by defining a set of random transformations to be applied to the images read by the Image-Data-Generator instance.

As Data augmentation technique we use decrease the size of images by using decreasing image size (scaling), and we cropping the images, Rotation and we implement padding for image augmentation technique.

3.3.1. Data Augmentation for Proposed Model

To increase number of images the study implement augmentation. We implement image data generator to supplement data. And the images that preprocessed data are augmented.

Image_Data_Generator Augmentation techniques	values
Rotation_range	25
Width_shift_range	0.1
Height_shift_range	0.1
Shear_range	0.2
Zoom_range	0.2
Horizontal_flip	True
Fill_mode	nearest

Table 3-1. Augmentation techniques that have been used on proposed model

3.4. Data splitting after augmented images for Proposed Model

The study uses Image_Data_Generator Augmentation techniques. These techniques used to increase our dataset, after that augmented dataset divided in to two parts; training and test data sets also there is validation data sets, this validation data is from training datasets. So, the model train from training data and validation data also used to whether the model valid or not. Finally, we test performance of the model by testing data set this data sets also must be unseen datasets.

Epochs	Steps	Learning Rate	Batch Size	Width, Height, Depth	Activation Function	Loss Function	Optimization Algorithm
10/5	100	le-3	64/32	256,256,3	Softmax	Categorical_cross_entropy	Adam optimizers

Table 3-2. Hyper-parameters that have been used on proposed model

Hyper-parameters are the variables which determines the network structure. For example, Number of Hidden Units and the variables which determine how the network is trained (Learning Rate), batch size, etc. Hyper parameters are set before training.

3.5. Image Feature Extraction for Proposed Model

Feature extraction is the process of assessing and extracting attributes such as picture color, edges, interest points, texture, and image resize from coffee leaves and beans. GLCM color feature extraction from photos is used in this thesis. To define the areas of a coffee leaf and coffee beans, we use RGB color characteristics. Color feature extraction reduces the original data set by measuring qualities, or features, that differentiate between the four forms of coffee plant illnesses. Why did we choose GLCM? Different color variations of each coffee leaf and coffee bean image can be seen in Ethiopian coffee plant diseases.

3.5.1. Gray Level Co-occurrence Matrix (GLCM)

A Gray Level Co-occurrence Matrix (GLCM) is a histogram of co-occurring grayscale values over a picture at a specified offset. Image analysis approaches include the GLCM and its supplemental texture feature computations. An image with pixels and intensity for a specific GLCM gray level, as represented by a table of different gray level combinations that occur in an image. The texture feature calculated by using the GLCM's detailed information to produce intensity variation at the most critical pixel. A gray-scale photograph was used to construct the GLCM [51] .

The following Four mathematical processes are used in the GLCM method. The RGB photos of the coffee leaf and coffee beans are first converted into (hue, saturation, and luminance) HSI color space representation. Each pixel map is utilized to build a color co-occurrence matrix after this procedure is completed, resulting in three color co-occurrence matrices, one for each of H, S, and I. Local homogeneity, contrast, cluster shade, Energy, and cluster prominence are texture features that are generated for the H picture using the following equation;

- ✓ Contrast: The sharpness of images and the depth of textural grooves are reflected by contrast.
- ✓ Energy: The grayscale distribution uniformity of images and texture crudeness are reflected by the square sum of each matrix member is called Energy.
- ✓ Local Homogeneity: Homogeneity reflects picture texture homogeneity and scales local changes in image texture.
- ✓ Entropy: The non-uniformity and complexity of visual texture are reflected in entropy.

$$contrast = \sum_{i,=0}^{N-1} (i,j)^2 \, c(i,j) \qquad \textit{Equation (1)}$$

$$Energy = \sum_{i,j=0}^{N-1} C(i,j)^2 \qquad \textit{Equation (2)}$$

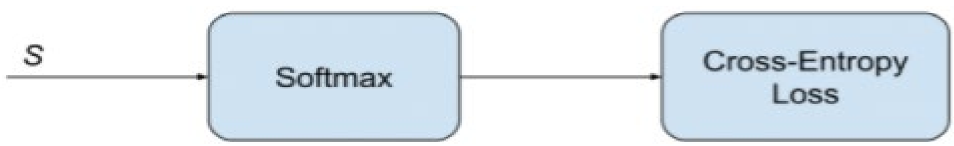
$$Local\ Homogeneity = \sum_{i,=0}^{N-1} C(i,j)/(1+(i-j)^2) \qquad \textit{Equation (3)}$$

$$Entropy = -\sum_{i,=0}^{N-1} C(i,j)\, log\, C\,(i,j) \qquad \textit{Equation (4)}$$

3.6. Loss and Metrics for the Model

3.6.1. Categorical Cross-Entropy loss

We employ Categorical Cross-Entropy as a model for evaluation. Softmax Loss is another name for it. It's a combination of a Softmax activation and a Cross-Entropy loss. We will train a CNN to output a probability over the C classes for each image if we utilize this loss. It is employed in the classification of many classes [52].



$$f(s)_i = \frac{e^{8_i}}{\sum_j^c e^{8_j}} \quad CE = -\sum_{i}^{C} t_i \, log(f(s)_i)$$

Figure 3-3. Loss of Categorical Cross-Entropy

In the specific case of multi-Class classification, the labels are one hot, so only the positive class C_p keeps its term in the loss. There is only one element of the Target vector t which is not zero $t_i = t_p$. So, discarding the elements of the summation, which are zero due to target labels, we can write:

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)$$

Where S_p is the CNN score for the positive class.

After defining the loss, we must compute its gradient in relation to the CNN's output neurons in order to back propagate it through the net and optimize the defined loss function by modifying the net parameters. As a result, we must compute the Cross Entropy Loss gradient for each CNN class score in s. The loss terms from the negative classes are all equal to zero. However, because the Softmax of the positive class is equally dependent on the scores of the negative classes, the loss gradient with regard to those negative classes is not cancelled.

The gradient expression will be the same for all *C* except for the ground truth class C_p, because the score of *Cp* (Sp) is in the nominator.

After some calculus, the derivative respect to the positive class is:

$$\frac{\partial}{\partial_{s_p}}\left(-\,log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)\right)=\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}-1\right)$$

In addition, the derivative respect to the other (negative) classes is:

$$\frac{\partial}{\partial_{s_n}}\left(-\,log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)\right)=\left(\frac{e^{s_n}}{\sum_j^c e^{s_j}}\right)$$

Where s_n is the score of any negative class in *C* different from *Cp*.

When Softmax loss is used is a multi-label scenario, the gradients get a bit more complex, since the loss contains an element for each positive class. Consider *M* are the positive classes of a sample. The CE Loss with Softmax activations would be:

$$CE=\frac{1}{M}\sum_{P}^{M}-log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)$$

Where each s_p in M is the CNN score for each positive class. A scaling factor *1/M* to make the loss invariant to the number of positive classes, which may be different per sample. The gradient has different expressions for positive and negative classes. For positive classes:

$$\frac{}{\partial_{s_{p_i}}}\left(\frac{1}{M}\sum_{P}^{M} -log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)\right) = \frac{1}{M}\left(\left(\frac{e^{s_{pi}}}{\sum_j^c e^{s_j}} - 1\right) + (M-1)\left(\frac{e^{s_{pi}}}{\sum_j^c e^{s_j}}\right)\right)$$

Where $s_p i$ is the score of any positive class.

For negative classes:

$$\frac{\partial}{\partial_{s_n}}\left(\frac{1}{M}\sum_{P}^{M} -log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)\right) = \frac{e^{s_n}}{\sum_j^c e^{s_j}}$$

3.6.2. Binary Cross-Entropy Loss

Loss of Sigmoid Cross-Entropy is also known as Sigmoid Cross-Entropy loss. It's a sigmoid activation with a loss of Cross-Entropy. It is independent for each vector component (class), unlike Softmax loss, which means that the loss computed for each CNN output vector component is unaffected by the values of other component components. That's why it's employed in multi-label classification, when the knowledge of an element belonging to one class shouldn't influence the categorization of another. Because it creates a binary classification problem between C′=2 classes for each class in C, it's called Binary Cross-Entropy Loss [52].

So when using this Loss, the formulation of Cross Entropy Loss for binary problems is often used 0 and 1

$$CE = -\sum_{i=1}^{C\prime=2} t_i\, log\big(f(s_i)\big) = -t_1 log\big(f(s_1)\big) - (1-t_1) log(1-f(s_1))$$

$$CE = -t_1 log\big(f(s_1)\big) - (1-t_1) log(1-f(s_1))$$

$$f(s_i) = \frac{1}{1+e^{-s_i}}$$

Figure 3-4. Cross-Entry Loss in Binary

This would be the pipeline for each one of the C classes. We set C independent binary classification problems (C′=2). Then we sum up the loss over the different binary problems: We sum up the gradients of every binary problem back-propagate, and the losses to monitor the global loss. s1 and t1 are the score and the ground truth label for the class C1, which is also the class Ci in C. s2=1 −s1 and t2=1−t1 are the score and the ground truth label of the class C2, which is not a "class" in our original problem with C classes, but a class we create to set up the binary problem with $C1=Ci$. We can understand it as a background class.

The loss expressed: $$CE = \begin{Bmatrix} -log\big(f(s_1)\big) & if \quad t1 = 1 \\ -log\big(1 - f(s_1)\big) & if \quad t1 = 0 \end{Bmatrix}$$

Where t1=1 means that the class C1=Ci is positive for this sample.

In this case, the activation function does not depend in scores of other classes in C more than C1=Ci. Therefore, the gradient respect to each score s_i in s will only depend on the loss given by its binary problem.

The gradient respect to the score *si =s1*: Written as:

$$\frac{\partial}{\partial_{s_i}}(CE(f(s_i)) = t1(f(s_1) - 1) + (1 - t1)f(s_1)$$

Where *f ()* is the sigmoid function. Written as:

$$\frac{\partial}{\partial_{s_i}}(CE(f(s_i)) = \begin{Bmatrix} f(s_i) - 1 & if \quad ti = 1 \\ f(s_i) & if \quad ti = 0 \end{Bmatrix}$$

The proposed model uses Categorical Cross-Entropy for model accuracy performance evaluation, which is based on the two evaluation techniques mentioned above.

```
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

3.7. Evaluation Methods

After our model training, we must be testing how much the model accurate. When we test the model with unseen data, we say the model correctly classify or not. Therefore, as a result, model evaluation is the process of estimating the model generalization accuracy using unknown data. When we test the model, it is not recommended to use training data for evaluating a model, because the model remembers all data samples which are fed during training, i.e., it predicts correctly for all the data points in the training but not for data which hasn't seen during the training. In this thesis, classification accuracy metrics are used which is recommended technique for classification. It is not suggested to evaluate a model using training data since the model remembers all data samples fed during training. It not predicts properly so testing data must be from unseen data sets. The classification accuracy measures are employed in this thesis, which is a preferred classification technique.

In this technique, the dataset is separated into three sections: training, validation, and testing. We can input the validation split to the model during training to acquire performance measures. The model returns training data accuracy and loss, as well as validation data accuracy and loss, which are training accuracy, validation accuracy, training loss, and validation loss, respectively. Using these measures, we may plot a loss and accuracy graph with respect to epochs. Finally, the testing data means images have not been used from training or validation data sets is given to the trained model to test the performance of the model, then the model returns accuracy and loss of the testing data.

3.7.1. Proposed Model Performance Evaluation

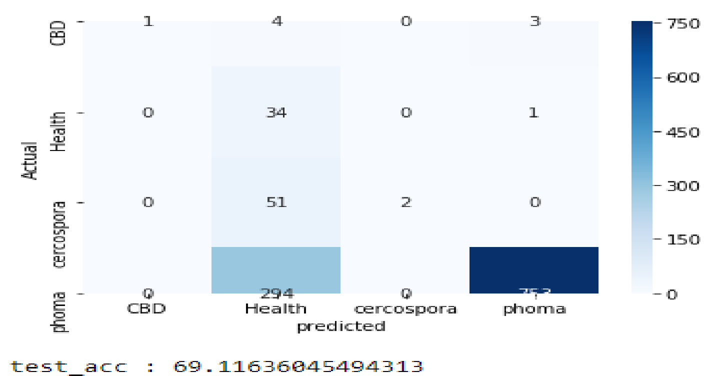


Figure 3-5. Proposed model Confusion matrix

Figure 3-5 shows model performance evaluation to testing data. The figure shows that diagonally is true positive class. When we take example, phoma disease vs others disease are the same on color on images, so it has same characteristics because of that the classes shift to other classes.

CHAPTER FOUR

EXPERIMENT AND DISCUSSION

In this chapter, an attempt is made to construct a model for coffee leaf and beans diseases detection and classification process. The experiment, shows the process of disease detection and classification by using deep learning.

4.1. Model Selection

This thesis select Deep learning algorithm called CNN model for coffee disease detection. These days CNN model is suitable for image processing. This algorithm extract feature by itself means CNN have convolution layer, Relu so this layer used to extract feature from input images also on GLCM we use entropy as feature extraction. The model takes image as an input, preprocess the image data and categorized the image data in to healthy, CBD, CLS, CPD.

A convolutional network ingests such images as three separate strata of color stacked one on top of the other. A normal color image is seen as a rectangular box whose width and height are measured by the number of pixels from those dimensions. The depth layers in the three layers of colors (RGB) interpreted by CNNs are referred to as channels [53].

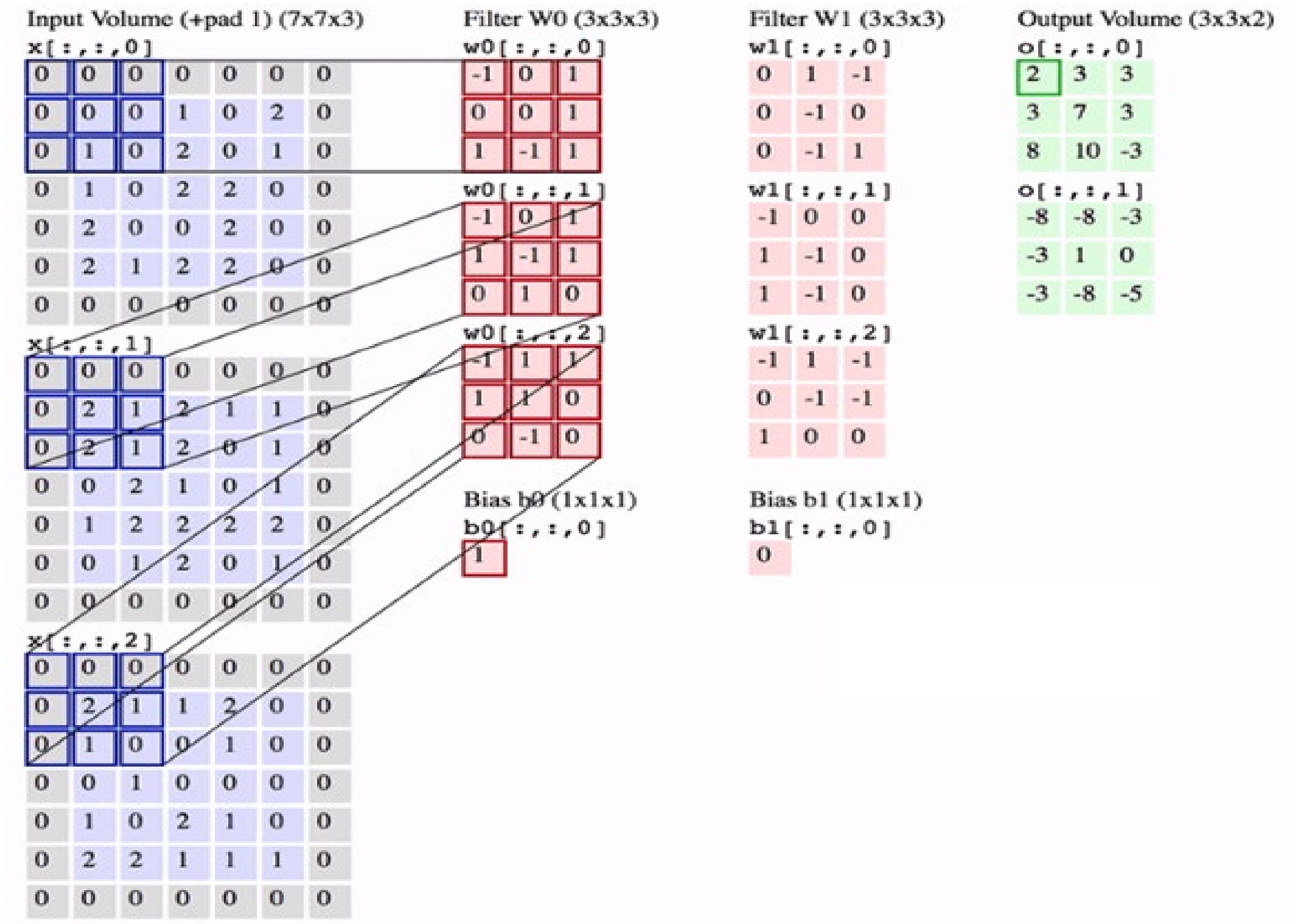


Figure 4-1. How do CNN Works?

The Convolutional Layer is the first layer in a CNN network, and it is the main building element that does the majority of the computational effort. Filters or kernels are used to convolve data or images. Filters are small units that we use in a sliding window to apply across the data. The image's depth is the same as the input; for example, if the RGB value of depth is four, a filter with a depth of four would be applied to the image.

This procedure entails taking the element-wise product of the image's filters and then adding those specific values for each sliding action. A 2d matrix would be the output of a convolution with a 3d color filter [53].

4.2. Comparison of Pre-trained CNN Model

This study compares vgg19, Inceptionv3, ResNet50 and AlexNet pre-trained architectures. From those architectures the study compares vgg19 and alexNet architectures by minimizing number of parameters, number of layers. Why we minimize those parameters and layers, if we implement all number of parameters and layers, they have millions of parameters this are needs GPU or TPU processor for analyses the image data so we minimize and we use CPU process and we get best model performance after training the model. We see comparison of architectures from the following.

VGG-19

VGG is a convolutional neural network which has a depth of 19 layers. It was built and trained by Karen Simonyan and Andrew Zisserman at the University of Oxford in 2014 and you can access all the information from their paper, Very Deep Convolutional Networks for Large-Scale Image Recognition, which was published in 2015. Vgg-19 needs More than 1 million image datasets to train the model. You can import the model with Image-Net training weights, of course. This network has been pre-trained to classify up to 1000 items. The network was trained on colored images with a resolution of 224x224 pixels [54]. Here's a quick rundown of its dimensions and capabilities.

Number of Memory Size: 549 MB
Top-1: Accuracy: 71.3%
Top-5: Accuracy: 90.0%
Number of Parameters: 20,124,740
Depth: 3

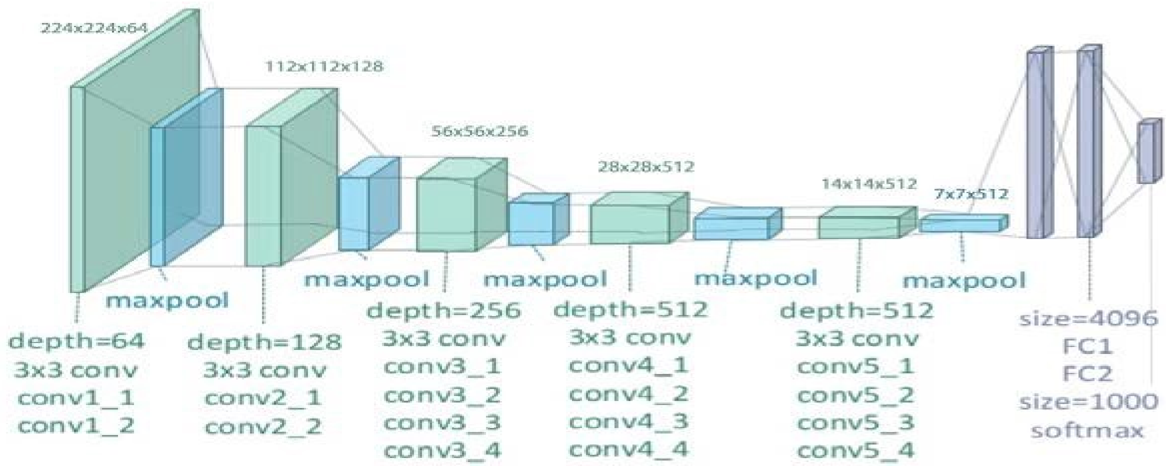


Figure 4-2. Architecture for Vgg19

Inceptionv3 (GoogLeNet)

Inceptionv3 is a convolutional neural network, which has a depth of 50 layers. It was built and trained by Google and you can access all the information on the paper, titled "Going deeper with convolutions". The pre-trained version of Inceptionv3 with the ImageNet weights can classify up to 1000 objects. The image input size of this network was 299x299 pixels, which is larger than the VGG19 network. While VGG19 was the runner up in 2014's ImageNet competition, Inception was the winner [54]. The brief summary of Inceptionv3 features is as follows.

- Number of Memory Size: 92 MB
- Top-1: Accuracy: 77.9%
- Top-5: Accuracy: 93.7%
- Number of Parameters: 23,851,784
- Depth: 159

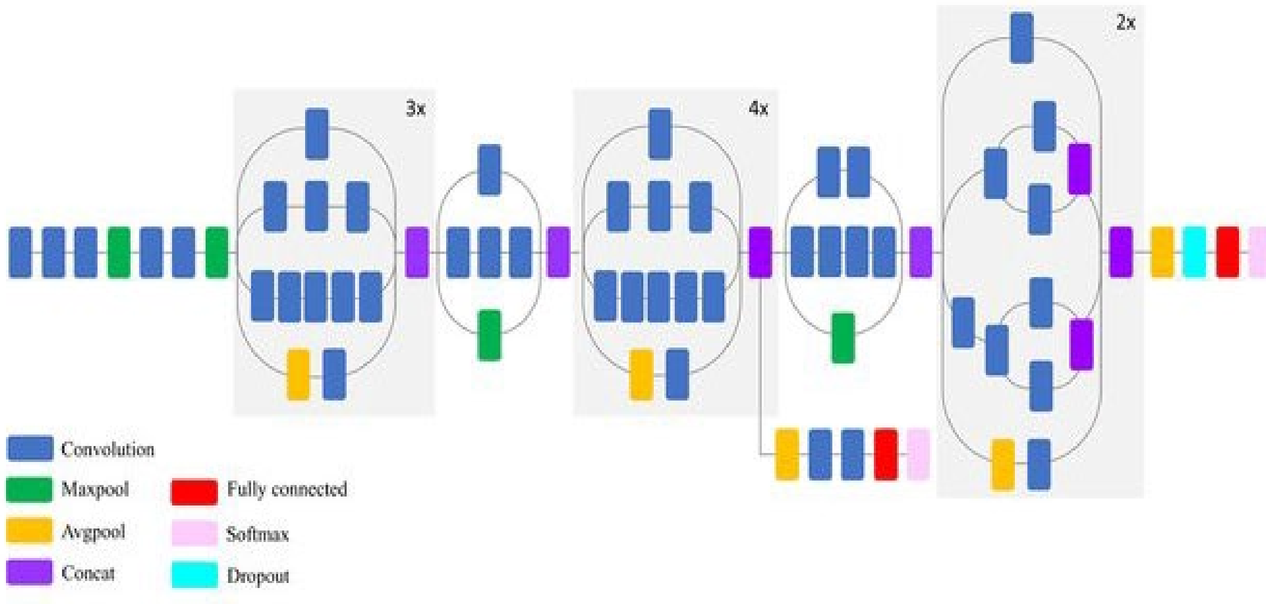


Figure 4-3. Architecture for Inceptionv3

ResNet50 (Residual Network)

ResNet50 is a convolutional neural network which has a depth of 50 layers. It was built and trained by Microsoft in 2015 and you can access the model performance results on their paper, titled Deep Residual Learning for Image Recognition. This model is also trained on more than 1 million images from the Image-Net database. Just like VGG-19, it can classify up to 1000 objects and the network was trained on 224x224 pixels colored images [54]. Here is brief info about its size and performance:

- Number of Memory Size: 98 MB
- Top-1: Accuracy: 74.9%
- Top-5: Accuracy: 92.1%
- Number of Parameters: 25,636,712

If you compare ResNet50 to VGG19, you will see that ResNet50 actually outperforms VGG19 even though it has lower complexity. ResNet50 was improved several times and you also have access to newer versions such as ResNet101, ResNet152, ResNet50V2, ResNet101V2, ResNet152V2.

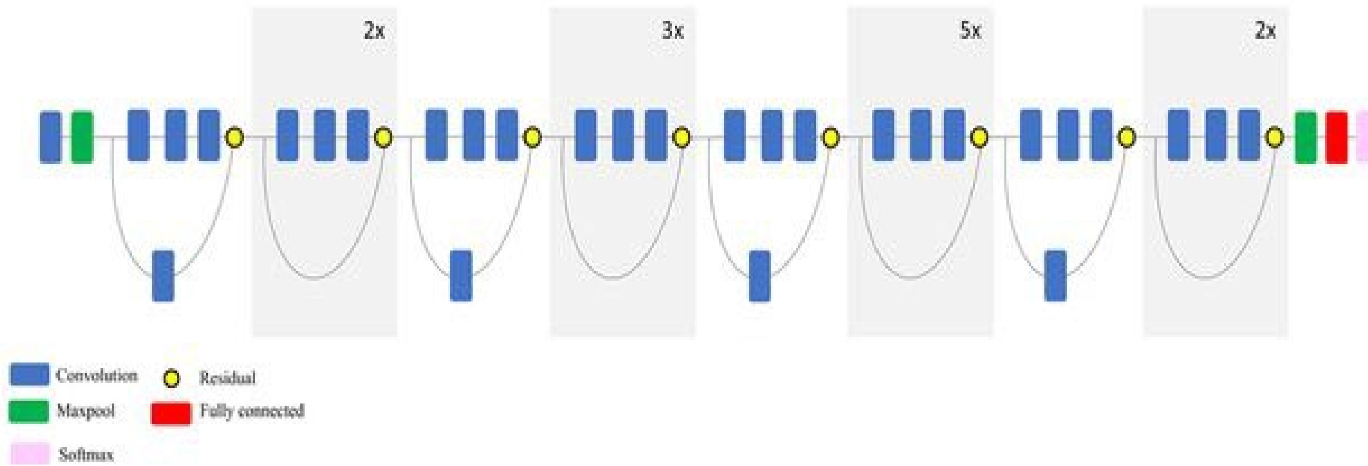


Figure 4-4. Architecture for ResNet50

AlexNet

The architecture consists of eight layers: five convolutional layers and three fully-connected layers. But this isn't what makes AlexNet special; these are some of the features used that are new approaches to convolutional neural networks [55].

- ✓ ReLU Nonlinearity: AlexNet uses Rectified Linear Units (ReLU) instead of the tanh function, which was standard at the time. ReLU's advantage is in training time; a CNN using ReLU was able to reach a 25% error on the data set six times faster than a CNN using tanh.

- ✓ Multiple GPUs: Back in the day, GPUs were still rolling around with 3 gigabytes of memory (nowadays those kinds of memory would be rookie numbers). This was especially bad because the training set had 1.2 million images. AlexNet allows for multi-GPU training by putting half of the model's neurons on one GPU and the other half on another GPU. Not only does this mean that a bigger model can be trained, but it also cuts down on the training time.

- ✓ Overlapping Pooling: CNNs traditionally "pool" outputs of neighboring groups of neurons with no overlapping. However, when the authors introduced overlap, they saw a reduction in error by about 0.5% and found that models with overlapping pooling generally find it harder to over fit.

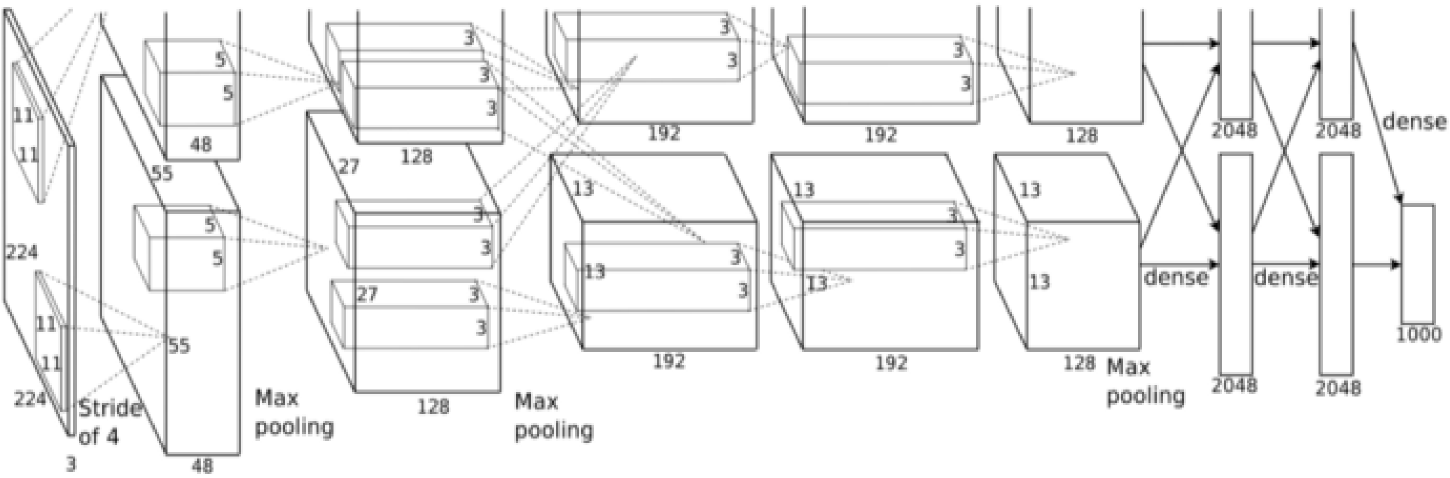


Figure 4-5. Architecture for Alex Net

Finally, based on the preceding CNN model architecture, this study implements AlexNet architecture by decreasing the layers and parameters and model that uses 256 x 256 pixels and minimizes layers by 10 and total parameters 693,733, depth = 3 with 93.5 accuracy. The model has the best accuracy based on this. Why use AlexNet architecture in this study? AlexNet design enables for multi-GPU training by putting half of the model's neurons on one GPU and the other half on a different GPU. This not only allows for the training of a larger model, but it also reduces the training time and Overlapping Pooling.

.

4.3. Process of Training on Proposed Model

The enhanced data is used for training, and the original validation data is used to calculate performance measures. The CNN Conv2D, MaxPooling2D, Flatten, and Dense layers gather important information from each image and categorize the disease via model training. Using the validation dataset, which is intended to test the model's performance, we may access the model's performance during the training process. After accessing the model's performance, save the model that performed the best using the.h5 and pickle format types. The testing phase is then carried out by feeding the predicted model unseen photos during training. Finally, the model gives a class prediction, which is the likelihood that an image belongs to one of the specified classes during training (infected disease name and healthy).

We don't utilize a predefined model since predefined models require millions of data sets and more parameters, but we implement by minimizing parameters and layers on our CPU processor. Predefined models, on the other hand, require GPU and TPU. The proposed model includes five convolution layers, three fully connected layers, a ReLU activation function in the hidden layers to provide nonlinearity during network training, and dropout after the first two fully connected levels to avoid overfitting. We reduced the number of neurons, parameters, and filters in CNN models due to a decrease in trained classes, hardware resources, and the number of images.

- ✓ **Padding:** the padding parameter of the Keras Conv2D class can take one of two values: 'valid' or 'same'. Setting the value to “valid” parameter means that the input volume is not zero-padded and the spatial dimensions allowed to reduce via the natural application of convolution.
- ✓ **Activations** In machine learning and deep learning, learning, is a special function used to find whether a specific neuron is activated or not. Activation function does a nonlinear transformation of the input data and thus enable the neurons to learn better. Output of a neuron depends on the activation function. As you recall the concept of single perception, perception, the output of a perceptron (neuron) (neuron) is simply the result of the activation function, which accepts the summation of all input multiplied with its corresponding weight plus overall bias, if any available.

4.4. Proposed Model Description

- ✓ **Input layer**: the input layer of our CNN model accepts RGB images of size 256×256×3 with four different classes (Cercospora leaf spot, Coffee Phoma Disease, Coffee Berry Disease and Healthy). This layer only passes the input to the first convolution layer without any computation. Therefore, there are no learnable features and the number of parameters in this layer is zero.
- ✓ **Convolutional layer**: In the proposed model, there are five convolutional layers. The first convolutional layer of the model filters the 256×256×3 input image by using 32 kernels with a size of 3×3 with padding='same'. Totally, we have 693,476 parameters on the model.
- ✓ **Pooling layer**: There are two max-pooling layers layer three and layer six convolutional layers of the proposed model. The first max-pooling layer reduces the output of the first convolutional layer with a filter of size 5×5 with padding = "same". The second max-pooling layer takes as an input the output of the second convolutional layer and pools by using 5×5. On both max-pooling layers, numbers of parameters are 0.
- ✓ **Fully Connected (FC) layer**: in the proposed model there are two fully connected layers including the output layer. The first fully connected layers have 256 input layers and have 663808 parameters.

- ✓ **Output layer**: proposed AlextNet architecture have four layers namely convolutional layer, max_pooling layer, flatten and dense layers and it has forth layer with a softmax is for activation function. In addition, categorical_cross_entropy with rmsprop optimizer used for output probability over the classes.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 254, 254, 32)      9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 50, 50, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 50, 50, 32)        9248
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 48, 48, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 9, 32)          0
_________________________________________________________________
activation_1 (Activation)    (None, 9, 9, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2592)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               663808
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 1028
=================================================================
Total params: 693,476
Trainable params: 693,476
Non-trainable params: 0
_________________________________________________________________
```

Figure 4-6. Total number of parameters on proposed model

The model uses a limited number of parameters (693, 476) when compared to other deep learning architectures such as Alex Net, which has 60 million parameters, VGG, which has 138 million parameters, and GoogLeNet, which has 4 million parameters. They require a GPU or TPU processor, as well as a vast volume of data. As a result, we minimized the layers of CNN model. When we minimize AlexNet architecture layers and parameters we consume run time of the processes for image detection and classification time with available CPU.

4.5. Experimental Result

This content describes the implementation of the classification process of, cercospora, Lesions on coffee berries, phoma on coffee leaf and coffee beans by using the CNN algorithm, which was specified in detail in the previous model description content. In this section all the experimentation details such as results of each experiment, and discussion of these results are presented briefly. The results of the experiments are shown in different graphs and tables.

4.5.1. Feature Extraction

Feature extraction apply before model training. Is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups? So, when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process them. So, Feature extraction helps to get the best feature from those huge number of data sets by select and combine variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with the accuracy and originality. Feature extraction helps to reduce the amount of redundant data from the data set.

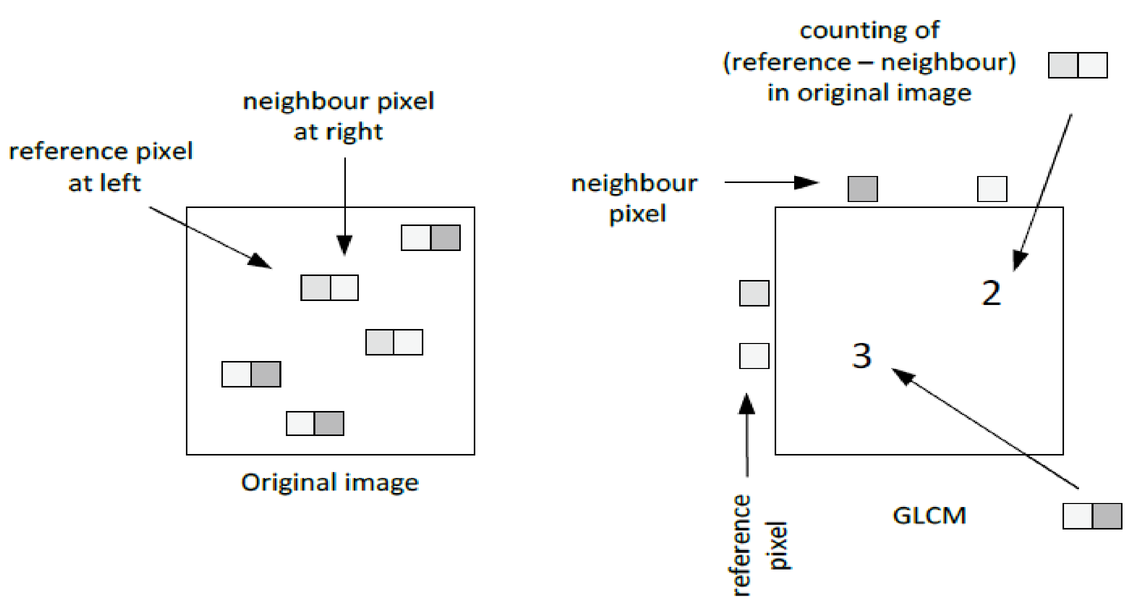


Figure 4-7. How to extract feature from original image in to GLCM

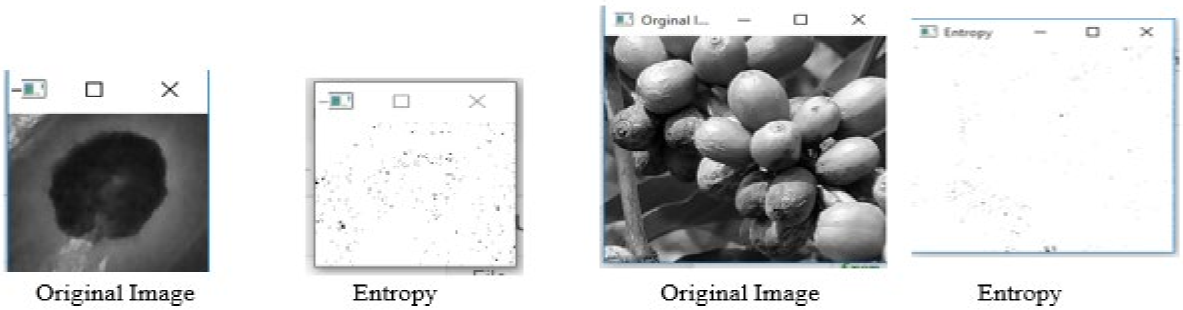


Figure 4-8. Simple output Feature Extraction

4.5.2. Result Analysis of Proposed CNN model

The following two plots show the classification accuracy and loss with respect to epochs using classification accuracy metrics such as training and validation accuracy, training loss, and validation loss of a 256 x 256 x 3 AlextNet model that we have experimented with by making some changes to the original pre-trained model in order to be able to classify well in our datasets.

When we see the training accuracy in epoch it is around 90 % and it increases permits 96% at epoch 5. In between epochs 3-5, the training accuracy of the model gets high accuracy, which means from 94% up to 96%. As we can see from the following two, high accuracy gets with small epochs. This is because of data sets and no need more time to process the epochs. Finally, as we can see from the following two plots the validation accuracy line is almost in sync with the training accuracy line and at the same time, the validation loss line is also in sync with the training loss.

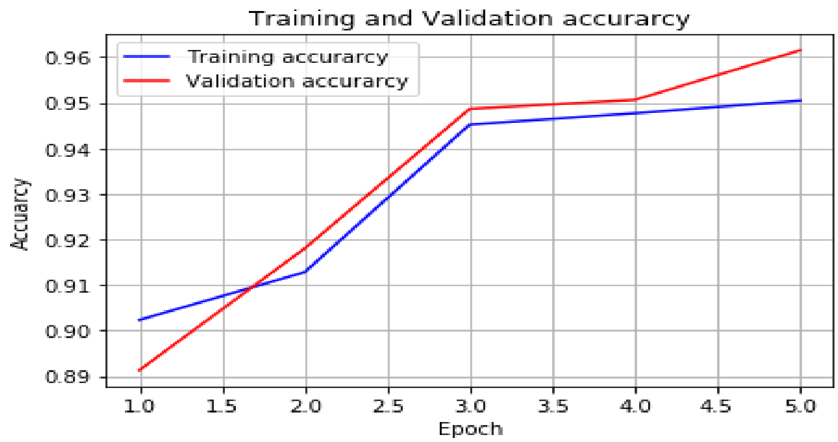


Figure 4-9. Accuracy of training and validation proposed model

Figure 4-10. Loss of train and validation proposed model

4.5.3. CNN Architectures Result Analysis

CNN Architectures	Number of Memory Size	Number of Parameters	Number of epochs	Depth	Accuracy	Processor Types	Remark
VGG19 (224x224 pixels)	549 MB	20,124,740	19/10	3	71.3% -90.0%	GPU or TBU	
AlexNet (256x256 pixels)	37 MB	693,476	10/5	3	93.5 – 96.14 %	CPU	Proposed Architect ure

Table 4-1. Result of the study by comparing CNN Architectures

4.6. Discussion of the Result

We have designed experimental scenarios to test the recognition performance by taking the extracted features of Ethiopian coffee leaf and coffee beans image. We have the following Feature extraction process and attributes such as picture color, texture, and size from coffee leaves and beans. GLCM color feature extraction from photos is used in this thesis. To define the areas of a coffee leaf and coffee beans, we use RGB color characteristics. The performances of detection and classification were tested by CNN (Convolutional neural Network) algorithm and according to table 4-1 we compare the following architecture Vgg19 (224 x 224 pixels) and AlexNet (256x256 pixels) architecture. Based on comparison we select best model in terms of model evaluation.

As Data sets Vgg19 architectures needs large image data sets, they have more parameters, layers because of that they need Graphics processing unit (GPU) and Tensor Processing Unit (TPU) processor. But our proposed model implemented by minimum data sets best accuracy on image processing. And the study uses image processing and deep learning approaches. In addition, the study implements Convolutional neural Network (CNN) algorithm because CNN is more suitable for image data processing. Based on our data sets we have four classes namely; Cercospora leaf spot, coffee phoma disease, coffee berry disease and Healthy coffee. Finally, the proposed model keeps time efficiency and effectiveness on coffee disease detection and classification with 96.1 % accuracy in training model and 69.1% on testing.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

This day's coffee production is affected by different disease namely; Cercospora leaf spot (CLS), Coffee Phoma Disease (CPD) and Coffee Berry Disease (CBD), which reduces the production and quality of coffee yield. Also, the shortage of diagnostics tools in developing countries like Ethiopia has impact on their development of country. Therefore, there is an urgent need to detect the disease at an early stage with affordable and easy to use technological solutions. In order to make early identification of the diseases we have proposed and implemented a deep learning approach by using CNN algorithm. We have presented a CNN model to identify and classify CLS, CPD and CBD by using coffee leaf and coffee beans images of the yield as an input. The proposed CNN model can be used as a tool to identify CLS, CPD and CBD disease of coffee.

The first contribution of this research is to design and develop a CNN model correctly detect and classify the well-known coffee leaf and coffee beans diseases (CBD, CLS, and CPD). Which is disease by using images that are taken in the real scene and under challenging condition such as complex background, dynamic image resolution, different illumination and orientation. The second main contribution of this thesis was well organized and managed dataset of coffee leaf and coffee beans. To accomplish these, we have conducted two experiments by using modifying pre-trained models and proposed model. During the experiment we have used images that are directly collected from coffee farm with the help of agriculture exerts and from existing data base is used. We have trained the two modified pre-trained models namely the VGG19 and AlexNet. After two experiments, the models were able to find a good classification result. The VGG19 model gives training accuracy of 90.0% and testing accuracy of 71.3%, the AlexNet modified pre-trained model gives training accuracy of 96.1% and testing accuracy of 69.1%, the proposed model gives 96.1% for both training and testing.

Finally, the proposed modified AlexNet model can significantly support accurate detection and classification of the three coffee diseases with little computational effort and insufficient images, which is far less than expected for deep learning algorithms because most deep learning algorithms are trained with millions of images and high computational resources. To that aim, the experiment's findings have encouraged us, and we propose to continue working and testing our model with additional coffee Disease.

5.2. Recommendations

Because Ethiopia's economy is based on agriculture and agricultural products, the primary goal of agriculture sectors should be to preserve crops from disease. As a result, image analysis techniques are critical for identifying and classifying early crop diseases. In Ethiopia, there is research has been conducted on some parts of coffee disease like coffee leaf, so far needs to detect the disease from coffee beans with difference types of disease using image analysis technique. Hence this thesis may initiate researchers to work more in the area. Like stem and roots of the coffee parts.

For the future, the study recommended the following two recommendation

- ✓ To train and test the model to detect the disease from coffee stem and root parts and it is also better to train the dataset with other pre-trained deep learning models such as ResNet which are not conducted in our experiment due to computational resources such as GPU.

References

[1] Assfa, Alemseged; , Getaneh Arega;, "Agricultural economics, research extension and farmers' linkage coordination research directory," *Ethiopian Coffee Exporters Association ECEA,* June 2013.

[2] ALIYI, RUKIYA, "A DISSERTATION SUBMITTED TO ADDIS ABABA UNIVERSITY COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE DEPARTMENT OF PHYSICS, IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN NUCLEAR PHYSICS," June 2020.

[3] Aparisi, Alban Mas, "Ethiopian coffee marketing reforms and smallholder coffee producers," International Institute for Environment and Development (UK), London, 2021.

[4] Keyzer, Michiel; , Merbis Max; , Overbosch; , Geert ;, "Types of coffee grown in ethiopia," 2000.

[5] Abu Tefera, "Coffee Annual Report," United States Department of Agriculture Foreign Agricultural Service, Ethiopia, May 14, 2021.

[6] F. Staff, "ASSESSMENTS OF COMMODITY AND TRADE ISSUES MADE BY USDA STAFF AND NOT NECESSARLY STATMENTS OF OFFICIAL U.S. GOVERNMENT," USDA Foreign Agricultural Service, Ethiopia, 2019.

[7] Ethiopian, USDA, "Ethiopia Coffee Market - Growth,Trends and Forecasts (2020 - 2025)," egion: Ethiopia, 2020.

[8] Ethiopia-coffee-market, "Mordor Intellegency," [Online]. Available: https://www.mordorintelligence.com/industry-reports/ethiopia-coffee-market.

[9] S.K.Mangal, "Coffee Planting, Production and Processing," *Gene-Tech Books,* 2007.

[10] Ervine, F.R, "West African Agriculture: West African Crops 3rd Edition.," vol. Volume 2, 1969.

[11] Melkamu Alemayehu, "Ethiopian Highlands: Home for Arabica Coffee (Coffea arabica L.)," 30 May 2017.

[12] Mengistu, Abrham Debasu; Alemayehu, Dagnachew Melesew ; Mengistu, Seffi Gebeyehu ;, "Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques," *International Journal of Database Theory and Application,* vol. Vol.9, 2016.

[13] Dr. P. LAKSHMI DEVI, Professor; , Dr. N. SUBASH, Assistant Professor;, DIGITAL IMAGE PROCESSING, India: AICTE, 2019-20.

[14] Sethya, Prabira Kumar; Barpanda, Nalini Kanta ; Rathb, Amiya Kumar ; behera, santi kumari;, "Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey," *Published by Elsevier B.V,* 2020.

[15] Stastny, Jiri; , Martin Minarik;, "A BRIEF INTRODUCTION TO IMAGE PRE-PROCESSING FOR OBJECT RECOGNITION," *Researchgate,* 18 April 2016.

[16] Kufa, Taye; , Ashenafi Ayano; , Alemseged Yilma; , Teshome Kumela; , Wondiyfraw Tefera;, "The contribution of coffee research for coffee seed development in Ethiopia," *E3 Journal of Agricultural Research and Development,* vol. Vol. 1, pp. 1-9, November .

[17] I. C. Organization, "ICO trade statistics, Exports by exporting countries to all destinations," 2011.

[18] Tadesse, Tesfaye; , Bizuayehu Tesfaye; , Girma Abera;, "Coffee production constraints and opportunities at major growing districts of southern Ethiopia," *Cogent Food & Agriculture,* pp. 1-36, 20 April 2020.

[19] M. Alemayehu, "Ethiopian Highlands: Home for Arabica Coffee (Coffea arabica L.)," *ResearchGate,* 30 May 2017.

[20] Darci J. Harland, An Introduction to Experimental Research.

[21] François Chollet, Deep Learning with Python, Manning Publications, 2017.

[22] A. Kulkarni and F. A. B. , "Foundations of data imbalance and solutions for a data democracy," 2020.

[23] Al-jabery, Khalid K.; , Donald C Wunsch II;, "Data analysis and machine learning tools in MATLAB and Python," 2020.

[24] Atsbaha GEBRE-SELASSIE; , T Bekele;, "Areview of Ethiopian agriculture: Roles, Policy and Small scale farming System," 2011.

[25] Alemayehu Asfaw Amamo, "Coffee Production and Marketing in Ethiopia," *European Journal of Business and Management,* vol. Vol.6, 2014.

[26] Gorfu, Dereje; , Eshetu Ahmed;, "Crops and Agro-ecological Zones of Ethiopia".

[27] Pohlan, Hermann A. Jürgen; , Marc J. J Janssens;, "GROWTH AND PRODUCTION OF COFFEE," vol. Vo.III.

[28] Bittenbender, H. C.; Smith, Virginia Easton ;, "Growing Coffee in Hawaii," *College of Tropical Agriculture and Human Resources (CTAHR),* 2008.

[29] "Common Pests and Diseases," [Online]. Available: https://plantvillage.psu.edu/topics/coffee/infos.

[30] Catarino, Aricléia de Moraes; Pozza, Edson Ampélio ; Pozza, Adélia Aziz Alexandre; Santos, Leone Stabile dias; Vasco, Gabriel Brandão ; Souza, Paulo Estevão de;, "Calcium and potassium contents in nutrient solution on Phoma leaf spot intensity in coffee seedlings," *ResearchGate,* 25 March 2017.

[31] Nutman, F. J., "Coffee Berry Disease," 06 July 2009.

[32] Mathew, Amitha; , P. Amudha; , S. Sivakumari;, "Deep Learning Techniques: An Overview," *researchgate.net/publication/341652370,* 02 August 2020.

[33] Khan, Asifullah; , Anabia Sohail; , Umme Zahoora; Qureshi, Aqsa Saeed ;, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *Artificial Intelligence Review,* pp. 1-71, 15 May 2020.

[34] Digital Image Processing.

[35] R. S. Dwivedi, Digital Image Processing, Germany , 2017.

[36] Ambekar, Shubhangi S.; , Madhuri A. Chaudhari;, "ARTIFICIAL NEURAL NETWORK CONTROLLER FOR PERFORMANCE OPTIMIZATION OF SINGLE PHASE INVERTER," *International Journal of Artificial Intelligence & Applications (IJAIA),* vol. Vol.3, January 2012.

[37] Sharma, Siddharth; , Simone Sharma; , Anidhya Athaiya;, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology,* vol. Vol. 4, no. Issue 12, ISSN No. 2455-2143, pp. Pages 310-316, 2020.

[38] I. C. Education, "Neural Networks," 17 August 2020.

[39] K. Gurney, An introduction to neural networks, London and New York, 2004.

[40] Jure Zupan, "Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them," September 1994.

[41] Patterson, Josh; , Adam Gibson;, "Deep Learning," in *Deep Learning*, O'Reilly Media, Inc., 1005 Gravenstein Highway North,Sebastopol, CA 95472, 2017.

[42] Yamashita, Rikiya; , Mizuho Nishio; Gian Do, Richard Kinh ; , Kaori Togashi;, "Convolutional neural networks: an overview and application in radiology," 22 June 2018.

[43] F. Li, J. Justin and Y. Serena, "Convolutional Neural Networks for Visual Recognition," *http://cs231n.stanford.edu/index.html,* 20 january 2019.

[44] F. Li, J. Justin; , Y. Serena;, "Convolutional Neural Networks for Visual Recognition," 20 January 2019.

[45] F. Li, J. Justin; , Y. Serena;, "Convolutional Neural Networks for Visual Recognition," *http://cs231n.stanford.edu/index.html,* 20 january 2019.

[46] D. A. Khan, "Introduction to Deep Convolutional Neural Networks," *ResearchGate,* 08 November 2016.

[47] YIDNEKACHEW KIBRU AFEWORK, "DEVELOPING BACTERIAL WILT DETECTION MODEL ON ENSET CROP USING A DEEP LEARNING APPROACH," 2019.

[48] Lucas ximenes Boa sorte; Feeeaz, Carolina Toledo ; , Francisco Fambrini ; , Roseli dos Reis Goulartc; Saito , Jose Hiroki ;, "Coffee Leaf Disease Recognition Based on Deep Learning and Recognition Based Texture Attributes," *Elsevier B.V,* p. 135–144, 2019.

[49] Mengistu, Abrham Debasu; Dagnachew, Melesew Alemayehu; Seffi , Gebeyehu Mengistu;, "Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques," *International Journal of Database Theory and Application,* 2016.

[50] Afifi, Ahmed; , Abdulaziz Alhumam; , Amira Abdelwahab;, "Convolutional Neural Network for Automatic Identification of Plant Diseases with Limited Data," *MDPI,* 24 December 2020.

[51] Castillo, Alvin M.; , Raymond D. Aradanas; , Edwin R. Arboleda; , Andy A. Dizon; , Rhowel M. Dellosa;, "Coffee Type Classification Using Gray Level Co-Occurrence Matrix

Feature Extraction And The Artificial Neural Network Classifier," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* Vols. VOLUME 8,, 10 OCTOBER 2019.

[52] Raúl Gómez Bruballa, "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names," 23 May 2018. [Online].

[53] Simran Bansari, "Introduction to how CNNs Work," 13 Feb 2019.

[54] Yalçın, Orhan G, "Using State-of-the-Art Pre-trained Neural Network Models to Tackle Computer Vision Problems with Transfer Learning," 23 Sep 2020.

[55] Jerry Wei, "AlexNet: The Architecture that Challenged," 3 Jul 2019.

[56] Joo, Javier Montenegro, "Digital Image Processing," *Research Gate,* 13 September 2015.

[57] Krizhevsky, Alex; , Ilya Sutskever; , Geoffrey E. Hinton;, "ImageNet Classification with Deep Convolutional Neural Networks," *International Conference on Neural Information Processing Systems,* vol. Volume 1, December 2012.

[58] Massimo Buscema, "Back Propagation Neural Networks," *Semeion Research Center of Sciences of Communication,* 09 May 2015.

[59] "https://plantvillage.psu.edu/topics/coffee/infos/diseases_and_pests_description_uses_propagation".

[60] Rutherford, Mike A.; , Noah Phiri;, "Pests and Diseases of Coffee in Eastern Africa: A Technical and Advisory Manual," *Crop Protection Product,* 2006.

[61] Mengistu, Abrham Debasu; Mengistu, Seffi Gebeyehu ; , Dagnachew Melesew;, "An Automatic Coffee Plant Diseases Identification Using Hybrid Approaches of Image Processing and Decision Tree," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. Vol. 9, March 2018.

[62] Shorten, Connor; , Taghi M. Khoshgoftaar;, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data,* 2019.

[63] Visa, Sofia; , Brian Ramsay; , Anca Ralescu; , Esther van der Knaap;, "Confusion Matrix-based Feature Selection".

[64] "Adobe Photoshop CS5 Tutorial".

[65] Cerda, Rolando; , Jacques Avelino; , Christian Gary; , Philippe Tixier; , Esther Lechevallier; , CleÂmentine Allinne;, "Primary and Secondary Yield Losses Caused by Pests and Diseases: Assessment and Modeling in Coffee," *PLOS ONE,* pp. 1-17, 3 January 2017.

[66] Vipin Tyagi, "Understanding Digital Image Processing," *ResearchGate,* September, 2018.

Appendix

Coffee leafe and coffee beens disease Detection.

Import Libraries

Importing necessary libraries and modules required to build the coffee leaf and coffee been classification model.

In [4]:
```
import tensorflow as tf
import numpy as np
import pickle
import os
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
```

preprocess image data for train,val and test

In [16]:
```
import splitfolders
```

In [18]:
```
input_folder ="C:/Users/Admin/Desktop/coffee dataset"
output= "C:/Users/Admin/Desktop/processed_data"
splitfolders.ratio(input_folder, output, seed=42, ratio=(.8, .1, .1))
```

```
Copying files: 5720 files [01:09, 82.55 files/s]
```

Load Dataset

Initializing a few parameters required for the image dataset preprocessing.

In [5]:
```
# Dimension of resized image
DEFAULT_IMAGE_SIZE = tuple((256, 256))

# Number of images used to train the model
N_IMAGES=5720

# Path to the dataset folder
root_dir = 'C:/Users/Admin/Desktop/processed_data'

train_dir = os.path.join(root_dir,'C:/Users/Admin/Desktop/processed_data/train')
validation_dir = os.path.join(root_dir,'C:/Users/Admin/Desktop/processed_data/val')
test_data_dir = os.path.join(root_dir,'C:/Users/Admin/Desktop/processed_data/test')
```

Augmentation for training

In [27]:
```
img_height, img_width = (256,256)
batch_size = 32

train_datagen = ImageDataGenerator(
                                rotation_range=25,
                                width_shift_range=0.1,
                                height_shift_range=0.1,
                                shear_range=0.2,
                                zoom_range=0.2,
                                horizontal_flip=True,
                                validation_split=0.2,
                                fill_mode="nearest")

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training') #set as training data

validation_generator = train_datagen.flow_from_directory(
    validation_dir,# same directory as training data
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation') #set as validation data
```

```
Found 3662 images belonging to 4 classes.
Found 112 images belonging to 4 classes.
```

for testing

In [28]:
```
test_generator = train_datagen.flow_from_directory(
    test_data_dir,
    target_size=(img_height, img_width),
    batch_size=1,
    class_mode='categorical',
    subset='validation') #set as test data
```

```
Found 113 images belonging to 4 classes.
```

In [29]:
```
x,y =test_generator.next()
x.shape
```

Out[29]:
```
(1, 256, 256, 3)
```

Here, we load the training data images by traversing through all the folders and labels into separate lists respectively.

In [7]:

```
image_list, label_list = [], []

try:
    print("[INFO] Loading images ...")
    processed_data_folder_list = listdir(train_dir)

    for processed_data_folder in processed_data_folder_list:
        print(f"[INFO] Processing {processed_data_folder} ...")
        processed_data_image_list = listdir(f"{train_dir}/{processed_data_folder}/")

        for image in processed_data_image_list[:N_IMAGES]:
            image_directory = f"{train_dir}/{processed_data_folder}/{image}"
            if image_directory.endswith(".jpg")==True or image_directory.endswith(".JPG")==True:
                image_list.append(convert_image_to_array(image_directory))
                label_list.append(processed_data_folder)

    print("[INFO] Image loading completed")
except Exception as e:
    print(f"Error : {e}")

# Transform the loaded training image data into numpy array
np_image_list = np.array(image_list, dtype=np.float16) / 225.0   #normalization
print()

# Check the number of images loaded for training
image_len = len(image_list)
print(f"Total number of images: {image_len}")
```

```
[INFO] Loading images ...
[INFO] Processing CBD ...
[INFO] Processing cercospora ...
[INFO] Processing Health ...
[INFO] Processing phoma ...
[INFO] Image loading completed

Total number of images: 5057
```

Split Dataset

In [13]:

```
print("[INFO] Splitting data to train and test...")
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_size=0.2, random_state = 42)
```

```
[INFO] Splitting data to train and test...
```

Build Model

In [ ]:

```
#Defining the hyperparameters of the coffee disease classification model.
```

In [14]:

```
EPOCHS = 5 #no of iteration
STEPS = 100
LR = 1e-3 #epsilon
BATCH_SIZE = 32     # 64 to adgest converjency
WIDTH = 256
HEIGHT = 256
DEPTH = 3
```

In [15]:

```
model = Sequential()   # Sequential model is basically a linear composition of Keras Layers
inputShape = (HEIGHT, WIDTH, DEPTH)
chanDim = -1 #noofchanel

if K.image_data_format() == "channels_first":
    inputShape = (DEPTH, HEIGHT, WIDTH)
    chanDim = 1 #no of chanel rgb

model = Sequential()
model.add(Conv2D(32,(3,3),padding='same',input_shape=(256,256,3),activation='relu'))# 3 RGB with is HWD
model.add(Conv2D(32,(3,3),activation='relu'))
# The padding parameter of the Keras Conv2D class can take one of two values: 'valid' or 'same'.
# "valid" parameter means that the input volume is not zero-padded and
# the spatial dimensions are allowed to reduce via the natural application of convolution.

model.add(MaxPooling2D(pool_size=(5,5)))

model.add(Conv2D(32,(3,3),padding='same',activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu')) #activation function

model.add(MaxPooling2D(pool_size=(5,5)))

model.add(Activation('relu')) #if the input is negative (x<0), it converts it to zero and the neuron is not activa

model.add(Flatten())
model.add(Dense(256,activation='relu'))
model.add(Dense(4,activation='softmax')) #  signma called binary , If we use this loss, we will train a CNN to out

model.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy']) # categorical_crossentropy

model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 254, 254, 32)      9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 50, 50, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 50, 50, 32)        9248
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 48, 48, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 9, 32)          0
_________________________________________________________________
activation_1 (Activation)    (None, 9, 9, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2592)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               663808
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 1028
=================================================================
Total params: 693,476
Trainable params: 693,476
Non-trainable params: 0
```

Train Model

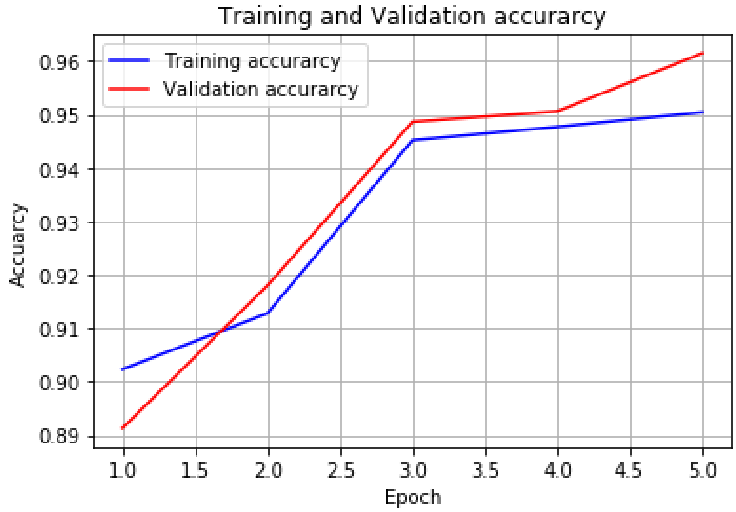
In [ ]:
```
#We initialize Adam optimizer with learning rate and decay parameters.

#Also, we choose the type of loss and metrics for the model and compile it for training.
```

In [13]:
```
# Initialize optimizer
opt = Adam(lr=LR, decay=LR / EPOCHS)

# Compile model
#model.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])

# Train model
print("[INFO] Training network...")
history = model.fit_generator(train_datagen.flow(x_train, y_train, batch_size=BATCH_SIZE),
                              validation_data=(x_test, y_test),
                              steps_per_epoch=len(x_train) // BATCH_SIZE,
                              epochs=EPOCHS,
                              verbose=1)
```

```
[INFO] Training network...
WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_
variables is deprecated. Please use tf.compat.v1.global_variables instead.

Epoch 1/5
126/126 [==============================] - 640s 5s/step - loss: 0.3265 - accuracy: 0.9023 - val_loss: 0.2627 - val_accuracy: 0.
8913
Epoch 2/5
126/126 [==============================] - 572s 5s/step - loss: 0.2008 - accuracy: 0.9128 - val_loss: 0.1601 - val_accuracy: 0.
9180
Epoch 3/5
126/126 [==============================] - 618s 5s/step - loss: 0.1254 - accuracy: 0.9452 - val_loss: 0.1026 - val_accuracy: 0.
9486
Epoch 4/5
126/126 [==============================] - 634s 5s/step - loss: 0.1183 - accuracy: 0.9477 - val_loss: 0.1098 - val_accuracy: 0.
9506
Epoch 5/5
126/126 [==============================] - 651s 5s/step - loss: 0.1089 - accuracy: 0.9504 - val_loss: 0.0995 - val_accuracy: 0.
9615
```

Comparing the accuracy and loss by plotting the graph for training and validation.

In [54]:
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

# Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.ylabel('Accuarcy')
plt.xlabel('Epoch')
plt.grid(True)
plt.legend()

plt.figure()

# Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.ylabel('loss')
plt.xlabel('Epoch')
plt.grid(True)
plt.legend()
plt.show()
```



In [76]:
```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
915/915 [==============================] - 32s 35ms/sample - loss: 0.1248 - acc: 0.9421
Test Accuracy: 94.20765042304993
```

save model

In [15]:
```
model.save('C:/Users/Admin/Desktop/my_model9504.h5')
```

In [27]:
```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
1012/1012 [==============================] - 32s 32ms/sample - loss: 0.0995 - acc: 0.9615
Test Accuracy: 96.14624381065369
```

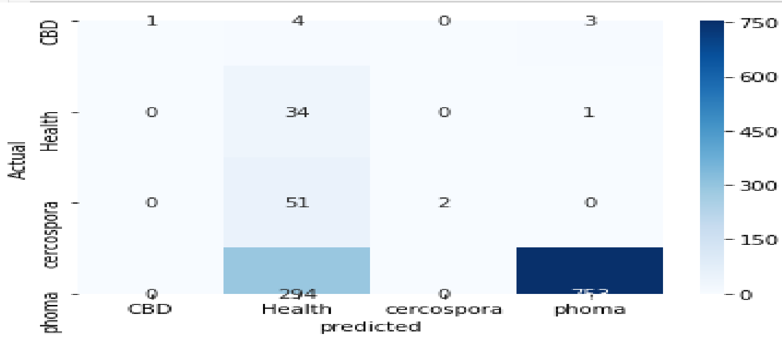
Test accuracy

In [28]:
```
test_loss, test_acc = model.evaluate(test_generator, verbose=2)
print('\nTest accuracy:', test_acc)
```

```
1143/1143 - 77s - loss: 107.4570 - acc: 0.6885

Test accuracy: 0.6885389
```

In [30]:

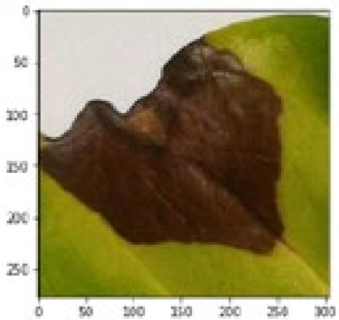
```python
import pandas as pd
import seaborn as sn
import tensorflow as tf
model = tf.keras.models.load_model('C:/Users/Admin/Desktop/my_model9504.h5')
filenames = test_generator.filenames
nb_samples = len(test_generator)
predicted_class=[]
actual_class=[]
test_generator.reset()
for _ in range(nb_samples):
    x_test, y_test = test_generator.next()
    predicted_class.append(model.predict(x_test))
    actual_class.append(y_test)


predicted_class = [list(train_generator.class_indices.keys())[i.argmax()]for i in predicted_class]
actual_class = [list(train_generator.class_indices.keys())[i.argmax()]for i in actual_class]

out_df = pd.DataFrame(np.vstack([predicted_class,actual_class]).T,columns=['predicted_class','actual_class'])
confusion_matrix = pd.crosstab(out_df['actual_class'],out_df['predicted_class'], rownames=['Actual'], colnames=['predicted']

sn.heatmap(confusion_matrix,cmap='Blues' , annot=True,fmt='d')
plt.show()
print('test_acc : {}'.format((np.diagonal(confusion_matrix).sum()/confusion_matrix.sum().sum()*100)))

```




```
test_acc : 69.11636045494313
```

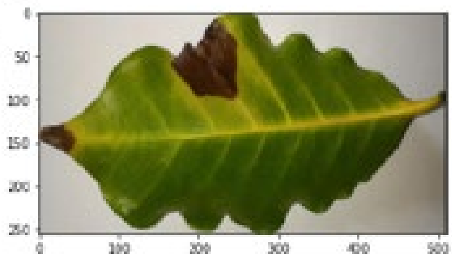
```
predict_disease('C:/Users/Haymi/Desktop/AAU/AAUCLDD/Dataset/test2.JPG')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\sequential.py:450: UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01.
Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
phoma
```



```
predict_disease('C:/Users/Haymi/Desktop/AAU/AAUCLDD/Dataset/test.JPG')
```

```
rust
```



```
predict_disease('C:/Users/Admin/Desktop/coffee dataset/test10.JPG')
```

```
CBD
```