Hawassa University
ሀዋሳ ዩኒቨርሲቲ

DEEP LEARNING BASED FABA BEANS LEAF DISEASES DETECTION AND CLASSIFICATION

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

BY: MARTHA MEZGEBU HAILU

HAWASSA UNIVERSITY, HAWASSA, ETHIOPIA

NOVEMBER, 2022

DEEP LEARNING BASED FABA BEANS LEAF DISEASES DETECTION AND CLASSIFICATION

BY: MARTHA MEZGEBU HAILU

PRINCIPAL ADVISOR: TILAHUN MELAK SITOTE (PhD)

CO-ADVISOR: DEGIF T (PhD)

A THESIS SUBMITTED TO THE

DEPARTMENT OF COMPUTER SCIENCE,

HAWASSA INSTITUTE OF TECHNOLOGY, SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

HAWASSA, ETHIOPIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

NOVEMBER, 2022

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

ADVISORS' APPROVAL SHEET

This is to certify that the thesis entitled "Deep Learning Based Faba Beans Leaf Diseases Detection and Classification" submitted in partial fulfillment of the requirements for the Degree of Masters with specialization in Computer Science the Graduate Program of the Faculty of Informatics, Department of Computer Science, Institute of the Technology and has been carried out by Martha Mezgebu under my/our supervision. Therefore I/we recommend that the student has fulfill the requirement and hence hereby can submit thesis to the department.

Tilahun Melak Sitote (PhD)		7/11/2022
Name of Principal Advisor	Signature	Date
Degif T. (PhD)	_____	_____
Name of Co-advisor	Signature	Date

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

EXAMINERS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by Martha Mezgebu Hailu have read and evaluated his thesis entitled "Deep Learning Based Faba Beans Leaf Diseases Detection and Classification" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree of master's Science in Computer Science.

Wogene Alemayehu Shenkute (MSc)	_____	_____
Name of the Chairperson	Signature	Date
Tilahun Melak Sitote (PhD)	_____	_____
Name of Principal Advisor	Signature	Date
Degif T. (PhD)	_____	_____
Name of Co- Advisor	Signature	Date
Anitha .V (_ PhD)	_____	_____
Name of Internal Examiner	Signature	Date
Michael Melese Woldeyohannis (PhD)	_____	_____
Name of External examiner	Signature	Date
_____	_____	_____
SGS Approval	Signature	Date

Declaration

I hereby declare this thesis work entitled “Deep Learning Based Faba Beans Leaf Diseases Detection and Classification” is my original work. This work is not present in any university for degree and all possible sources of material used in this thesis have been duly acknowledged.

Martha Mezgebu _____ _____

Name Signature Date

ACKNOWLEDGMENT

First of all, I would like to thanks to Almighty God, who gives me internal power, knowledge, idea, health, peace, and internal motivation to accomplish this study. Nothing is done without him.

Secondly, while there are several individuals who have made a difference me in one way or another to attain the completion of this study, it would have not been conceivable without the guidance, back and mastery of my research advisors, my principal advisor Dr. Tilahun Melak Sitote and my Co-advisor Dr. Degif T. So, I would like to begin by saying thanks to Dr. Tilahun Melak Sitote and my Co-advisor Dr. Degif T for their useful comments and exceptional offer assistance with this work, for permitting me the total flexibility to seek to pursue this study, to work on my claim activity and for making me to utilize the potential that I have with confidence on my ability. Thirdly, I am thankful to the experts of Faba bean plant for their willingness to help me by giving all possible information about Faba bean disease by devoting his time to explain disease symptom and categorize the disease class, also I might like to thank the private owner of Faba bean plantation which gave sample image of Faba bean leaf.

Lastly, I might like to thanks to my parent who putted a cornerstone in my life and my husband Abinet Feyisa and all my friend who help me through my thesis work.

Table of Contents

LIST OF ACRONYMS

DL	Deep Learning
CNN	Convolutional neural network
GDP	Gross Domestic Product
GPU	Global Position Unit
GPS	Global Position System
GIS	Geographical Information System
SVM	Support Vector Machine
ResNet	Residual Network
VGG Net	Visual Geometry Group Network
RELU	Rectified Linear Unit
TF	True positives
TN	True negatives
FN	False negatives
FP	False positives
AAERC	Asela Agricultural Engineering Research Center
FC	Fully Connected
Adam	Adaptive learning rate optimization algorithm
CL	Convolutional Layer
Conv	Convolution
PL	Pooling Layer
Train_accuracy	Training accuracy
Train_loss	Training loss
Val_accuracy	Validation accuracy
Val_loss	Validation loss
IDE	Integrated development environment

List of Tables

List of Figures

List of Equations

Abstract

Faba bean (Vicia Faba L.) is believed to be originated from the Near East and now days spread throughout the world. It's one of the most domesticly legume in the world next to chickpea and pea. Ethiopia is the second leading producer of Faba beans next to China in the world. It shares 6.96% of world production and 40.5% with of Africa. Faba bean is grown primarily for its edible seeds that are used for human consumption. It also used for keeping human healthy and sustaining the productivity of the farming system through the fixation of nitrogen. However, most of the time it is affected by different diseases that result in reduction of quality and quantity of the Faba bean production. Those diseases are caused by fungus, virus, and bacteria. Usually Faba bean diseases appear on the leaf, flower, pods, seed, and stem a step by step and makes the crop out of usage. Mainly, leaf of Faba bean is more affected by diseases than other parts. It attacks both inside and outside of the leaves. Leaf plays an important role during the growing period of Faba bean. Without leaf there is no flower, without flower there is no pod, without pod there is no seed. Traditionally, farmers and experts detect and identify plant diseases by naked eyes. This method is inaccurate and expensive, because there are numerous diseases. Detection by using image processing techniques has been more accurate and fast. Therefore, we need to develop automatic deep learning based Faba bean leaf diseases detection and classification model. We designed Faba bean leaf disease model architecture using convolutional neural network for Faba bean leaf diseases detection and classification. CNN become accurate and precise method for the detection and classification of plant diseases. The study can be conducted in the plantation area of Faba bean in Oromia region, Arsi zone, D/Xijo Woreda, from the farmer plantation land particular reference to Bucho Silase kebele, Ethiopia, where the dataset has been collected. Leaves of healthy and infected crops are collected and labeled. Processing of image has been performed with pixel-wise operations to enhance the image. It is followed with feature extraction the classification of patterns of captured leaves in order to identify Faba bean plant leaf diseases. Four classifier labels are used as ascochyta blight, chocolate spot- botrytis, rust, and Healthy leaf. The features extracted are fit into the neural network with the dataset was spilt into training set, validation set and testing set, 80%, 10%, and 10% respectively, with the batch size 32 and using Adam optimizer. Faba bean leaf diseases detection and classification model achieved the overall accuracy 99.58%.

Keywords: Deep Learning, Convolutional Neural Networks, Faba bean Leaf diseases Classification.

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Faba bean (Vicia Faba L.) is believed to be originated from the Near East and now days spread throughout the world [1]. It's one of the most domesticated grown legume in the world after chickpea and pea. Faba bean is the most widely grown food legume in Ethiopia [2]. Which covered 520,519 ha of the cultivated land with annual production and productivity of 6, 88,667 tons and 1323 kg ha-1 respectively [3]. Ethiopia is the second leading producer of Faba beans next to China in world, and it shares 6.96% of world production and 40.5% with in Africa [4]. The crop is among the major crop grown in different area in our country and its production is the rain fed a system, or it is an annual crop grown by subsistence farmers, during the cool main rain season from June to September.

In Ethiopia, Faba bean known as locally ''Bakela''. It's grown primarily for its edible seeds that are used for a human consumption [5]. Faba beans can be used either green or dried, fresh or canned [5]. Faba beans are one of the most promising crop. It is popularity used for fast food, business, and home use purpose. It's tightly coupled with every aspect of Ethiopian life. It is mainly used as to prepare a flour called shiro, which is used to make a shiro wot [5]. A stew used widely in Ethiopian dishes, an injera with a shiro wot is the most popular food in Ethiopia and most of the people eat this food [5]. During the fasting period in the Ethiopian Orthodox Church two uncooked spicy vegetable dishes are made using Faba beans. Those are Hilibet and Siljo. Both are served with other stews and an injera during lunch and dinner. In addition, Bakela nifro (boiled Faba beans), Gulban, Boq'ullit (boiled salted Faba bean embryo) and a Ripe is made using Faba beans. In industrialized countries cultivated Faba bean is used as an animal feed, mainly for pigs, horses, poultry and pigeons [5].

Faba beans have various importance in terms of keeping human healthy. Since, it is a good source of potassium: potassium helps to lower high blood pressure levels, helps in a kidney function, and it prevents heart diseases, it is a good source of vitamins like a vitaminsB1, a vitaminsB6, a vitaminsB9, a vitaminsB12, and a vitaminsC [5]. It is a good for pregnant women as it is packed with nutrients and a good source of a floate. Besides, it is a valuable source of a cheap protein for

the poor that cannot afford to buy an animal protein [5]. It also plays an important role in sustaining the productivity of the farming system through the fixations of nitrogen [5].

In spite of its enormous benefit, the productivity of Faba bean in Ethiopia remains far below the crop potential. Faba bean productivity is limited by various factors [4]. The major factors that limit the yield of Faba bean and cause an instability in yield productions are both biotic and abiotic factors [3] [4]. Diseases are among the most significant biotic factors and causing Faba bean yield reduction [4]. Due to these factors the total productivity of Faba beans is reduced. Faba beans productivity does reduced both in terms of a quality and a quantity due to the disease. The disease attack the different parts of Faba beans like leaf, stem, flower, pod and seed gradually, or a step by step and makes the crop out of usage. Faba beans plant diseases caused by bacterial, viral and fungal [6]. The most significant Faba beans yield limiting diseases are chocolate spot- Botrytis, Ascochyta blight, Rust, black rot, and a Faba bean necrotic yellow virus [4]. These are the common diseases that attack the Faba beans plants, and they hamper the Faba beans crop production [4]. Among those mentioned diseases like ascochyta blight, chocolate spot- botrytis, and Rust has high burdens in terms of effected the Faba bean in Ethiopia.

Those mentioned diseases attack the different parts of Faba beans. Mainly, from the parts of Faba beans it attacks the leaf than the other parts. Leaf plays an important role during the growing period of Faba beans, since without a leaf there is no flower, without a flower there is no pod, without a pod there is no seed. The Faba bean leaf diseases attack the inner side and outside leaves of Faba bean and an effect the crop from giving enough product. Due to their enormous effects, the researcher interested to study on the leaf parts than the other parts. The leaf of Faba bean is effected by different diseases [4]. Chocolate spot-Botrytis: is caused by Botrytis fabae, it is among the major diseases of Faba bean leaf which threaten its productivity [4]. A Chocolate spot-Botrytis occurs in all areas where Faba beans are grown, and it causes losses were ranging from minor to complete a crop failure depending on the severity of an infection [7]. The pathogen Botrytis fabae induces lesions or symptoms equally on the upper and lower leaf surfaces of Faba bean with a large number of spots. Affected plants usually have fewer pods. A Chocolate spot-Botrytis can cause the highest relative yield losses on Faba beans in our country [8] [4]. This disease reduces yields of Faba bean in Ethiopia by 94.6% [8].

Secondly, Ascochyta blight (Ascochyta fabae): is fungal diseases that attack all pulse crops, it caused by Ascochyta fabae. It is also the potential threat for Faba bean Production. Ascochyta blights can affect all growth stages of the plant and can cause lesions on all parts of the plant on leaves, stems, pods, and developing seeds. The disease has spread to all Faba bean growing areas with an average prevalence of 63.6% [1]. And thirdly, Rust (Uromyces viciae-fabae): is among the biotic constraints of fungal diseases. It is the main factors affecting the production and productivity as well as the quality of Faba bean in different parts of Ethiopia. Rust caused by Uromyces viciae-fabae. Normally, rust epidemics begin late in the season, when pod filling has started and crop losses usually range from 5 to 20% [1]. However, when the infection starts early in the season, severe epidemics can arise and yield losses can be as high as 60%. Faba bean rust is widely distributed in Ethiopia and causes yield losses of 27% [1].

Traditionally, identifying and detecting a symptom of those diseases by using a naked eye. Which required experience that is depending on the farmer's knowledge. Identifying the correct diseases symptom and understanding when to control those diseases are difficult for the farmers. Diseases detection by a visual way is a more laborious task and at the same time less accurate and can be done on a limited area. To identify a symptom and types of diseases the farmers ask a guidance from the professional. The professional uses naked eye for a detection and an identification of plant diseases. For doing so, a large number of a term of an expert as well as a continuous monitoring of plants required when they do with large farms. At the same time, some countries farmers do not have proper facilities, and even idea that they contact to an expert. Due to which a consulting expert cost is very high as well as time-consuming [9].

In agriculture, automatic plant diseases detection is essential one in monitoring large fields of crops, and this automatically detects a symptom of diseases as soon as appear on a plant leaf. Plant diseases automation increases the efficacy and accuracy of detecting diseases in plants. Nowadays, there is a method that has been used to detect and identify a plant disease discussed in this research that is using deep learning method in an image processing techniques. It has been applied in many application of smart agriculture such as water, soil management, crop cultivation, crop diseases detection, weeds removal, crop distribution and yield production. Deep learning became the most exact and a price platform for a detection of plant diseases. It is a multilayer perceptron. It is useful and promising in determine the severity of diseases in a plant and used to classify diseases and

avoid the late detection of diseases. Detection of diseases by using Deep learning is the easiest process and it is less time consuming. It understands features incrementally, doesn't require a domain expert to identify most applied features. It is also helping the expert to analyze the plant diseases and know their source in time.

In this study, an automatic model that detects and classifies Faba beans leaf diseases by using a deep learning has been developed. The proposed model detect and classify the leaf diseases of Faba bean from the symptoms that appear on the plant leaves. The detection and classification of leaf diseases have been done based on a deep learning approach. There are different kinds of Deep Learning architecture like deep neural networks, Belief networks, recurrent neural networks and convolutional neural networks [10].

An Image processing is used in this study for measuring an affected area of the plant parts and determines the difference by using the groups of features such as color, shape, and texture of the affected area. Detection in this study is used to refer if Faba bean leaf is infected with diseases or healthy. The study considers four class which contains three class of Faba beans leaf diseases, and one healthy leaf class of the plant. Those diseases are chocolate spot-Botrytis, Ascochyta blight, and Rust. Generally, the researcher has been developed suitable and accurate an automatic deep learning based Faba beans leafs diseases detection and classification model.

1.2 Motivation

Automatic plant disease detection is advantageous because it decreases the amount of labour required to monitor large crop farms and can identify disease symptoms as soon as they occur on a plant's leaf. In our country farmers and investors detect the disease by naked eye observation of experts in the traditional approach adopted in practice for diseases. So, the researcher is motivated to study this area. Because of losing cost, time, and manpower.

Agriculture is extremely important in a human beings' life [11]. In this world there is no person that live without eating food. So, agriculture is the warrant of a human beings' life. However, the crop production get easily impacted by different plant diseases. The estimated annual crop loss due to plant diseases at worldwide is $60 billion. The result of crop loss lead to the global food insecurity. Specially, in our country it has negative impact on farmers whose live hood is dependent on safe cultivation as well as it reduces economy of the country [11]. Faba bean is one

part of agricultural sector that contribute to the economy of country as well as worldwide. Like, other agricultural crop Faba bean crop also easily affected by various plant diseases. Hence, the researcher motivated to develop a model for the automatic detection and classification of Faba bean leaf diseases from the images by using deep learning (CNN).

1.3 Statement of the Problem

Productions based on a farming get easily affected by various plant diseases. Diseases have turned into a dilemma as it can cause a significant reduction in both the quality and a quantity of Faba bean products. In Ethiopia, the disease causes 50-100% Faba beans the crop yield loss [12].

Agriculture is that the leading sector of the Ethiopian economy. The majority of Ethiopians is farmers, but they have not yet secured food at large due to various factors such as plant diseases. These diseases cost as an ecological, social and economic loss to farmers as well as an economy of the country. So, it becomes important to analyze plant diseases accurately within specific time [11].

Identifying various diseases in a plants are very difficult for farmers [11]. Most of the time farmers monitors their cultivation crop after sow. But, their monitoring was not successful due to more of farmers in our country is not educated. So, diseases were not easily being identified by the farmers as they lack of knowledge about the crop diseases. In order to overcome this problem, farmers ask helps from the expert to get a guidance about their symptom and types of a disease in plants [9]. Experts use a naked eye in practice for a detection and an identification of plant diseases. However, this requires a continuous intensive care of experts which might be prohibitively expensive in large farms. Further, in developing countries, farmers may have to go a long distance to contact experts, this makes consulting professionals are too expensive and time-consuming. Nowadays, an image processing is the key technique for the diagnosis of various diseases of the crop in the areas of agriculture. It can automatically detect the affected area of diseases by measuring their feature like a color, a shape, and a texture. As well as it minimizes a confusion and helping the expert. So, developing crop diseases detection and a classification system [13] [11].

Faba beans are one of the multipurpose crop which is used for human consumption, it also used for keeping human healthy and sustaining the productivity of the farming system through the fixation of nitrogen as well as it is an important cash crop. Beyond to its various benefits the

productivity of Faba beans crops infects by different diseases. Most of the time diseases of Faba bean infect the leaf of the plant first and then hampers the other parts of the plants from growing. Faba beans leaf disease detection is very essential as it leads to huge damage to the whole Faba beans crop. Some research has been done on diseases detection and a classification of other plant leaf. But, further there is no work done on Faba beans leaves diseases detection and a classification [14],[11] ,[9]. Thus, developing Faba bean leaf diseases detection and classification model is mandatory as well as the objective of this study.

The researcher intends to develop a suitable and accurate model that has high accuracy and performance to detect and classify Faba beans leafs diseases. The researcher develops that, the automatic model that using deep learning to develop a model that allows as to detect different diseases of Faba beans leafs. Then after a detection, it classifies them in to their classes.

1.4 Research Question

To this end, this study tries to answer the following research questions to achieve the objectives of the research.

1. What are the hyper parameters to design efficient deep learning model for Faba bean leaf diseases?
2. Which deep learning algorithm is more appropriate to detect and classify Faba beans leaf diseases?

1.5 Objectives

1.5.1 General Objective

The general objective of this study is to design and develop an efficient and more accurate deep learning (CNN) based Faba beans leaf diseases detection and classification model.

1.5.2 Specific Objectives

In order to achieve the main goal of this research, the following specific objectives are carried out:

- To collect the image of affected and healthy Faba beans leafs, then organize them for the dataset.
- To identifies the hyper parameters to design the model.

- To design deep learning based Faba beans leaf diseases detection and classification model.
- To trains the designed deep learning based Faba beans leaf diseases detection and classification model.
- To test the designed model
- To predict the designed deep learning based Faba beans leaf diseases detection and classification model.
- To evaluate the performance of the designed model using accuracy, precision, and Recall.

1.6 Scope and Limitation of the Study

The study was focused on designing and implementing an automatic model that can detect if Faba beans leaf was infected by different types of Faba beans leaf diseases and healthy. Then after a detection, it classifies them in to their classes. The study was delimited on a detection and a classification of diseases on the leaf only, but it doesn't focus on the other parts such as: flower, seed, and stem. Also, the research focus on detecting and classifying diseases, but doesn't find out the medicine or treatments for the diseases.

1.7 Significance of the Study

The target of this study was to know how deep learning algorithm is efficient and accurate in detecting and classifying Faba beans leaf plant diseases. This study will come out with different uses to different parts of the societies. Among the beneficiaries of this study are an agricultural sectors, government, farmers, investors, experts, and researchers.

- **Agricultural sector: -** One of the major challenge in the agriculture sectors is plant diseases which limit the yield as well as damage the whole crop. The result of this study will be more significant for the agricultural sector. It will be tools to detect and classifies the diseases accurately and easily.
- **Farmer and investors: -** farmers and investors beneficiaries from this study. Because of saving cost, time, and manpower.
- **Experts: -** It helps to improve skills, knowledge, minimizes confusion and reduce the load of their work for the professional who works on Faba beans plant.

- **Researchers: -** The study will be open the door for the researchers who have interest to work on agro technology.
- This study it will be used as guide for future researcher.

1.8 Organization of the Thesis

The thesis is organized in to six chapters. The first chapter discussed in brief about the background of the study, statement of the problem, general objective, specific objectives, scope and limitation of study, and significance of the study. The Second chapter on reviewing existing literature and related work. In this chapter, the review is literature on the domain of Faba bean plant, Faba bean plant diseases, machine learning, and deep learning. Finally, studies that are directly related to this thesis were discussed and summarized. The third chapter is about the methodology of the study. In this chapter, data collection, study methodology, materials, and tools which have been used is discussed. Chapter four deals with the proposed model and experimental parameters on Faba bean plant leaf disease detection and classification remedy using infected and healthy leaf images of Faba beans. The architectural design of the proposed model, and evaluation techniques were elaborated in depth. Chapter five presented about different experimental results with detailed discussions. In the last chapter, chapter six conclusion, recommendation, contribution and insight for future researchers were putted on.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

Any scientific investigation needs evidence, unless a scientific investigation without evidence is fiction. To make the study scientific, different literatures' were reviewed during the period of the study. Related journal, articles and a prior thesis were searched and reviewed to the extent possible related to the topic. Mainly, this chapter gives emphasis on the diseases that attack Faba beans plant leaf, image processing, machine learning, deep learning, and its approaches for real world applications. In a digital image processing, each machine vision system starts from an image capturing. After images are captured there are a number of procedures that we follow to reach the desired goal of a machine vision system. In this chapter, we describe the science of an image based deep learning.

2.2 Faba beans plant

Faba bean is commonly known as Vicia faba.L. Faba beans are a pulse crop that's a part of the legume family, the word "pulse" derived from the Latin word puls, which it means potage or thick soup [15]. Pulses are the dry edible seeds of pod plants and are high in protein and fibre, and low in fat [15]. Due to this, Faba beans are belonged to the botanical family of leguminasea also known as fabaceae [16]. Faba beans are stiffly an erect, self-fertile, annual plant that grows 2- 6 feet tall [17]. It is a cool-season crop that grows best in temperatures ranging from 60° to 65 °F, however it can grow in conditions as low as 40 °F and as high as 75 °F. It requires 80 to 100 days to reach harvest. Furthermore, it has different parts during the growing stage, staring from the first stage of snowing to the end of harvesting stage. The first part of Faba bean is stem: Faba beans have two to four stems that are square in cross-section, it is relatively strong and grow upright, the stem supports a height of one to two meters tall with one or more strong, hollow stems emanating from the base, with green foliage, and the stem turns dark at maturity [18].

The second part of Faba bean is leaf: leaves are much larger than other pulses such as chickpeas, lentils, and peas, the leaves are alternate, up to 8 cm long, they are pinnately compound with two to six leaflets, the first true leaf is produced at the third node position. The first two leaves be made

up of two leaflets each, and the remaining leaves consist of three or more leaflets. Faba bean plants are shorn of tendrils, or with only rudimentary tendrils [19]. The bottom pods change to black and dry from the bottom to the top of the plant as the crop matures, while the lower leaves age and drop [17]. The third parts of Faba beans is flowers: flowers in Faba beans are known as reproductive nodes, it occurs in 45 to 60 days, and it matured in 110 to 130 days, they are white with purple or black marking and are both self-pollinated and cross-pollinated, each cluster may produce one to six pods, clusters of one to five flowers are borne on short pedicels, axillary racemes, usually between the 5th to 10th node stages, and all colored flower types have seed coats that contain tannins [19]. The fourth parts of Faba beans is pod: Pods are long and green, it grows up to 10 cm long, 1 to 2 cm wide, and contain two to eight seeds per pod, it turns brown or black as they mature and seeds turn tan to brown to grey, they're not formed from every flower on the plant, and Only one quarter of the flowers will produce plants [17]. And the last and the fifth parts of Faba beans is seed: seeds are oval and flat, with a ridged, dimpled seed coat, its color varies between varieties from white to light tan or green, brown, and even purple or black, the number of seed vary from one to eight per pod. Seeds are serves for human consumption and animal feed [19].

2.2.1 Faba Beans Plant Diseases

Diseases are the state of local or systemic abnormal physiological functioning of a plant, resulting from the endless, prolonged 'irritation' caused by phyto-pathogenic organisms. The Plant becomes diseased when it's continuously disturbed by some causal agent that results in an abnormal physiological process that disrupts the plant's normal structure, growth, function, color or other activities. Diseases have a detrimental effect on plants and animals, effect on market access, and agricultural production. Among this, Faba bean plants is also a one that affected by a disease. Faba beans are subject to a number of diseases that can reduce a yield and a quality of the crop. Yield loss caused by diseases is estimated as high as about 100% [20]. While certain Faba bean infections may not result in much damage, others may be far more serious and require careful control. It is important to note that many pathogen infections collectively caused more damage and yield loss to the Faba bean plant than did an infection by a single disease. The effect of disease on grain quality in pulses can be far greater than yield loss: this must be accounted for in thresholds because in pulses, visual quality has a major impact on market price. The diseases can be initially come

from the seed, the soil, and self-sown seedlings, or in some cases, other plant species, and once the disease is existing, the source is then from within the crop itself [21]. More than seventeen pathogens have been reported so far on Faba bean from different parts of the country [22]. Those diseases cause serious damage to the plants and consequently decrease in the seed yield and quality. Diseases are affected different parts of Faba bean includes stem, leaf, pod, and seed. Diseases in different growth stages affect the development of vegetative and generative part of the plants, which is directly related with Faba bean yields. The collaboration of host, pathogen and environment are all critical points in disease development. Most of the time Faba beans plants diseases appear on the leaves parts. Leaf holds the other parts of the Faba beans such flower, pod and seed. Once the leaves attacks by the diseases, the other parts also in high risk to attack by the diseases. In other way it has high probability to infect by the diseases. Most of Faba beans plant disease express their symptom on the leaf of Faba beans. In this study, the researcher consider only leaf disease of Faba beans plant.

2.2.2 Faba Beans Leaf Diseases

Faba beans leaf diseases caused by different factor. Those factor can be mostly classified according to the nature of their primary causal agent, either infectious (biotic) or non-infectious (abiotic). Non-infectious plant diseases are usually referred to as disorders. Non-infectious plant diseases are affected by conditions external to the plant, not living agents. They cannot be transmitted from a diseased plant to a healthy one. Those diseases are caused by unfavorable growing situations, including excesses of temperature, inconvenient relationships between moisture and oxygen, harm full substances in the soil or atmosphere, an excess or deficiency of an essential mineral and too little or too much water cause plants to grow poorly [23]. Infectious plant diseases are caused by a pathogenic or living microorganisms that attack and obtain their nutrition from the plant they infect. These pathogens are capable of reproducing within or on its host, they can be spread from an infected plant to a healthy plant and may infect all types of plant tissue such as leaves, shoots, stems, crowns, roots, tubers, fruit, seeds and vascular tissues [21]. Infectious plant diseases have become an increasingly significant factor affecting Faba bean yield and quality efficiency.

There are various pathogenic organisms that causes Infectious diseases in Faba beans plant. Among the common Faba beans leaf disease caused by pathogenic organisms are fungi, bacteria and virus. Fungi is a living microorganism agent that causes infectious diseases in leaves of Faba

beans plant, the majority of Faba beans leaf diseases are caused by fungi, they're small, usually microscopic organisms that contains of a mass of filaments or threadlike strands called hyphae. They are primarily reproduce and spread by spores. They produce more than one type of spore during their life span, and this often influences how diseases are spread [24]. The primary way that they infect plants is through direct penetration of tissues, although they infect through natural openings includes stomates, hydathodes, and lenticels or through wounds [23]. The common fungal diseases that attack Faba bean leaf is chocolate spot (Botrytis fabae), ascochyta blight (Ascochyta fabae), and cercospora leaf spot (Cercospora zonata) [4]. Secondly, bacteria: bacteria is also another pathogenic microorganism that causes infectious diseases in Faba beans plant, it is single-celled microscopic organisms with cell walls that reproduce by binary fission that do not have an organized nucleus, most of the bacteria diseases are in Faba beans have rod-shaped, it can occur in a short period of time, and their initial growth within a plant is usually not visible, it can spread from plant to plant by splashing water, by insects, and through a variety of cultural practices, and they can also be transmitted by seed from infected plant [23]. Thirdly, Viruses: Viruses are very small organisms, it cannot be seen with an ordinary microscope, it causes diseases that can reduce plant vigour or kill them, it is unique plant pathogens since they consist of nucleic acid and a protein coat and have no cellular structures, it replicates or reproduce them self with the aid of the component of the plant host cell, and the primary means of spread is through the feeding activities of a number of insects, predominantly aphids, white flies, and leaf hoppers [23]. In this research, the following Faba beans leaf disease was considered.

2.2.2.1 Chocolate spot-Botrytis

Chocolate spot-Botrytis caused mainly by the fungus Botrytis fabae and to some extent by Botrytis cinerea. It appears everywhere Faba beans are cultivated and it's causing losses ranging from minor to complete failure of crop [21]. It requires high leaf moisture or humidity greater than 70% within the crop canopy and optimal temperature of 15-25%: under such condition this dis-causative diseases can spread rapidly with in crop and offenly causes considerable yield loss. Firstly, lesion occurs on the leaves of the plants. Leaves are the core parts of Faba beans plants affected by chocolate spot diseases. But, under favorable condition for the diseases it also spreads to stem, flower, and pods. When, the Faba bean plants impacted by these diseases it shows different symptoms on the parts of leaves. The first symptom are small grey brown spots visible only on

leaves. Then, it rapidly develops in to large, irregular-shaped lesions on leaves. These symptoms are varied, and range from small spots on the leaves to complete blackening of the entire plant. The diseases have two stages, such as a non-aggressive phase and aggressive phase. Initially, there is a non-aggressive phase during which small reddish-brown spots are "peppered" over the leaves. Then, there is an aggressive phase during which the spots darken and combine to form bigger grey-brown target areas that may finally cover the entire plant.

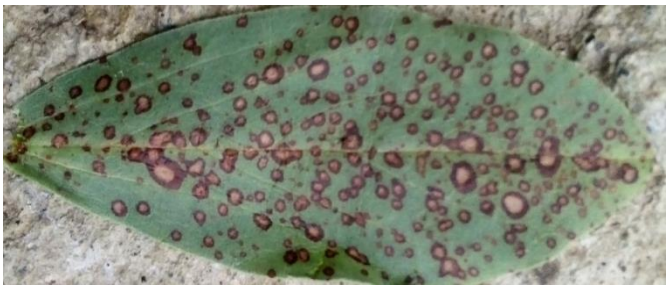


Figure 2-1: Chocolate spot-Botrytis Faba bean leaf diseases.

2.2.2.2 Ascochyta Blight

Ascochyta blight (Aschoyta fabae) also referred to as leaf blight. It is make happen by the fungus aschoyta fabae, the initial symptoms of aschoyta blight are lesion on the leaves of young plant [21]. They appear as small dark brown circular spots. As the diseases develops, lesions enlarge and turn light to dark grey in color which are visible on the both side of the leaves. Small black dots visible with in the Centre of the lesion. The lesion can also spread to stem and pod. Stem lesion can cause stem to break. It can cause both significant yield loss and down grading quality of Faba bean as a result of the disease discoloring the grain [21].

Figure 2-2: Aschoyta blight (aschoyta fabae) Faba bean leaf diseases.

2.2.2.3 Rust

Rust initiated by the pathogen Uromyces viciae-fabae [21]. It is most prevalent in the warmer Faba bean-growing areas. It is a serious disease of Faba beans. It doesn’t infect cereals or other break

crops. They usually occurs late in the growing season of Faba beans during podding. The diseases appear first on leaves of Faba beans plant as numerous small, orange-brown pustules; it surrounded by alight yellow halo, and as diseases develops, severely infected leaves wither and may fall from the plant [21]. In other ways, it's resulting in leaf damage and premature leaf drops. In favorable condition the diseases also spreads to other parts of Faba beans such us stem and pods. Rust epidemics can significantly reduce Faba beans yields. It has causes losses of yields up to 30 percent; while in combination with chocolate spot it causes yield reduction up to 50 percent.

Figure 2-3: Rust (Uromyces viciae-fabae) of Faba bean leaf diseases.

Fungal and Bacterial Diseases	Characteristic symptoms	Affected parts
Chocolate spot (Botrytis fabae and Botrytis cinerea)	- Discrete reddish-brown spot are 'peppered' over the leaves and stems. -Next an aggressive phase occurs when spots darken in color and coalesce to form larger grey-brown target spots that may eventually cover the entire plant.	Leaves are the main part of the plant affected, but under favorable conditions for the disease, it also spreads to stems, flowers and pods.
Ascochyta Blight (aschoyta fabae)	A grey spots which show through on both sides of the leaves. They develop grey centers which contain many black specks.	The whole part of the plant except stems.
Rust (Uromyces viciae-fabae)	A leaves will have numerous small, orange/brown pustules. Pustules are surrounded by a light yellow halo.	Affect the leaves and pod only.

Table 2-1: Faba bean leaves diseases symptoms and the affected parts.

2.3 Image Processing

Image processing is a technique to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a kind of signal dispensation in which input is an image, like video frame or photograph, and output may be image or characteristics associated with that image. It involves processing images using established signal processing techniques as two-dimensional signals. It has some basic steps that every image processing applications follow those are image capturing, image pre-processing, segmentation, feature extraction, image recognition and interpretation.

2.3.1 Image Acquisition

Image capturing is the action of retrieving an image from a source. The image can be retrieved usually by using hardware systems like cameras, sensors, etc. It is the initial and the most important step in the workflow sequence since, without an image, no actual processing is possible by the system [25]. It often used to create a digital representation of data from surveys and experiments. The image that is acquired by the system is totally unprocessed. Its aim is to transform an optical image into an array of numerical data which could be later manipulated on a computer. One of the ultimate objectives of this process is to have a source of input that operates within such controlled and measured guidelines that the similar image can, if necessary, be nearly perfectly reproduced under the same conditions, so anomalous factors are easier to locate and eliminate [24].

2.3.2 Image Pre-Processing

Pre-processing is a common name for tasks with images at the lowest level of abstraction. Whose target is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing and analysis task, improve the quality of the image in order to investigate it in a better way and convert the raw data into a clean data set. It is required to clean image data for model input. It may also decrease model training time and increase model inference speed. It decreases image information content. It uses the considerable redundancy in images and small neighborhood of a pixel in an input image to get a new brightness value in the output image. Neighboring pixels corresponding to one real object have the similar brightness value. The distorted pixel can be often bring back as an average value of neighboring pixels. There are various pre-processing techniques applied to the image based on the improvement

needed such as: removing noises, normalize the intensity or brightness of the individual particles, removing reflections and masking the portion of image, and adding edge detectors. The primary goal of the preprocessing stage is to gather relevant information from the rules and create efficient data structures that reflect the interdependence of the rules. Major tasks performed in image preprocessing are:

Image Resampling

It is a technique used to create a new version of the image with a different width and height in pixels by changing the pixel dimension of an image. It can be done by changing the image's file size by adding or deleting pixels within the image. Down sampling decreases the number of pixels in the image, while up sampling increases the number of pixels in the image.

Image Enhancement

Image enhancement is a technique for increasing the raw data's quality and information content before processing. It's the process of highlighting certain information of an image, as well as weakening or removing any unessential information according to specific needs. It brings out detail that is obscured and highlight certain features of interest in an image. It can be adjusted digital images so that the results are more suitable for display or further image analysis. It's a core area of image processing. It used to make it easier for visual interpretation and understanding of imagery. The main goal of image enhancement is to enhance the quality and visual appearance of an image, or to afford a better transform representation for future automated image processing and often to smooth images in more uniform regions but to preserve edges. Some of the most basic types of image enhancement tools are: change the contrast or brightness of an image, manipulate the grayscale or the red-green-blue color patterns of an image, eliminating noise, revealing blurred details, spatial filtering, density slicing and adjusting levels to highlight features of an image or making it easier to identify key features. Enhancement is based on human subjective preferences regarding what constitutes a "good" enhancement result.

Image Restoration

It deals with refining the appearance of an image in some predefined sense. It is an objective approach, in the sense that restoration methods tend to be based on mathematical or probabilistic

models of image processing. An image that has been degraded, reconstruct by using a priori knowledge of the degradation phenomenon. Image restoration refers to a class of methods that aim to remove or reduce the degradation that have occurred while the digital image was being gained. All natural images when displayed have gone through some sort of degradation: during display mode, acquisition mode, processing mode, sensor noise, blur due to camera miss focus, relative object-camera motion and random atmospheric turbulence.

2.3.3 Image Segmentation

Partitioning a picture into its various constituent portions or areas is a widely used approach in digital image processing and analysis. It often depends on the characteristics of the pixels in the image. It helps in the detection of objects and boundary line of the image, and it could involve in separating foreground from background, or clustering regions of pixels depend on similarities in color or shape [26]. Segmentation is a significant stage of the image recognition system, as it extracts the objects of our interest, for further processing includes description or recognition [26].

2.3.4 Feature Extraction

Feature extraction is a process of dimensionality reduction by which the first set of raw data is divided and reduced to more manageable groups for processing. Large data set characterized by a large number of variable that entail a lot of computing resource to process. It is useful when you have a large number of data set and need to reduce the number of resources needed for processing without losing important or relevant information. It can also decrease the amount of redundant data for a given analysis. The reduction of the data helps to build the model with less machine's efforts and also increase the speed of learning and generalization steps in the machine learning process. So, feature extraction used to get the best feature from those big data sets by select and combine variables into features, thus, effectively decreasing the amount of data that must be processed. It still accurately and completely describing the original data. Good feature extraction should have the following property such as uniqueness or in formativeness, repeatability, vicinity, accurateness, effectiveness, magnitude, robustness and invariance. There are different types of feature extraction techniques includes color, texture and shape [27].

Color Feature

Color feature is a commonly used feature for image representation. It is one of the most important and suitable feature which can easily recognize and retrieve image. It tends to be more domain independent when compared to other features. It takes a little semantic meaning and its compact representation. It is the most expressive of all the visual feature. The most essential benefits of color feature are power of representing visual content of images, easy extracting color information of images and high proficiency, relatively power in separating images from each other, and it is easy to extract, analyze and represent an object [27]. It has been widely used to show to its characteristics of robustness to background complication and independence over image size and orientation. It is represented by using color moment, fuzzy color moment and color histogram. Color feature extraction method can be classified into two categories, such as global method and local method. Global method consider a complete image, whereas local method consider a portion of the image.

Texture Feature

Texture is one of the essential features in image analysis for many applications. It is a repeated pattern of arrangement of the structure with regular intervals. It states the surface characteristics and appearance of an object set by the size, shape, density, arrangement, and proportion of its elementary parts [28]. It is used to partition images into regions of interest and to classify those regions. It affords information in the spatial arrangement of colors or intensities in an image. It is described by the spatial distribution of intensity levels in a neighborhood. Texture feature extracted using several methods Such as structural based method, statistical based method, model based method, and transform based methods [28].

Shape Feature

Shape is one of the most significant features in feature extraction. It is an important issue in image analysis for object recognition and classification. It is a very powerful feature. They are usually defined when the image has been segmented into different regions or objects. It can be categorized into either region based or boundary based. It is represented by a set of number called shape descriptor. A good and efficient shape representation feature for an object should be invariant to

translation, rotation, scaling, translation, affine invariance, noise resistance, oscillation invariance, statistically independent and reliability.

2.3.5 Image Detection

Detection is the process of detecting or noticing something in real world. It has the ability to detect if there is something verses nothing. So, detection in this study is used to refer to detecting if a Faba bean leave is infected with diseased or healthy.

2.3.6 Image Classification

Image classification is the method of sorting pixels into limited number of individual class of data based on their data files values [29]. If, a pixel satisfies a certain set of criteria, the pixels is assigned to the class that correspond to that criteria. It is used to reduce an image to information classes. Classification of pixels may be performed by using either supervised classification or unsupervised classification. Supervised classification is more thoroughly controlled by a human being than unsupervised classification. Supervised classification: it is used to extract quantitative information from an image, it selects pixels that represent patterns you recognize with help from other sources, knowledge of the data, the classes wanted and the algorithm to be used is required before you begin selecting training samples, and by identifying patterns in the image you can "train" the computer system to identify pixels with similar characteristics by setting a priority to the classes you supervise [29]. Unsupervised classification is use iso-data clustering. In this process, clusters of pixels based on their similarities in spectral information are automatically classified into classes of interest. The image is segmented into spectral classes automatically based on natural groupings of the data. In our research, we had used supervised approach to classify the three disease of Faba bean leave and healthy leaves.

2.4 Machine Learning

Machine learning is a subset of Artificial intelligence, which is a way of making machines intelligent to learn from a group of learning algorithm and apply what they have learned to make magnificent decision like human being. It deals a computer with the ability to learn and perform a task without explicitly being programmed, and being able to think and act with less human intervention. The workflow of machine learning starts with relevant features being manually extracted from images. It needs less computing power. It requires a domain expert to identify most

applied features. In other way, they require human intervention, since they work only for what they are designed only. That's why we require deep learning, which gives a bit better performance.

2.5 Deep Learning

Deep learning is a subclass of machine learning. It is representational learning method, which learn a good feature automatically from a raw data. It is a way of allowing computational model that are composed of multiple processing layer to learn representations of data with multiple levels of abstraction. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification, this avoiding a lot of time-consuming engineering, better chip processing capacities, considerable advances in the machine learning procedures, and affordable cost of computing hardware are primarily crucial reasons for the booming of deep learning [30]. A model learn a feature and a task, and extract a pattern directly from the data itself by using neural network, the data maybe an image, text, and sound, and it trained by using a large set of labeled data. Neural network is the fundamental building block of deep learning with several layers of nodes between input and output, which is designed to imitate how human think and learn or imitate the way human gain certain types of knowledge as well as these series of layers between input & output do feature identification and processing in a series of stages, just as our brains seem to. Each layer uses the output of the preceding layer as input. The learning process can be supervised, unsupervised or semi-supervised learning [31].

In deep learning: an image, emanates in the form of an array of pixel values, and the features learned in the initial layer of representation typically signify, the presence or absence of edges at particular orientations and locations in the image, the second layer, typically identifies motifs by spotting particular arrangements of edges, regardless of minor variations in the edge positions, the third layer, may collect motifs into bigger combinations that relate to parts of familiar objects, and subsequent layers would identify objects as combinations of these parts. One of the main importance of deep learning is that these layers of features are not designed by human engineers, while they are learned from data using a general-purpose learning procedure. In addition, deep learning can achieve state-of-the-art accuracy, sometimes exceeding human-level capacity and data scientists extremely beneficial, who are tasked with gathering, analyzing, and interpreting large amounts of data; it makes this process faster and easier. Furthermore, the appearance of GPUs

and the availability of large datasets were key enablers of deep learning, and they were greatly improved by the development of open source, flexible software platforms with spontaneous differentiation and this made it easy to train complicated deep networks and to reuse the latest models and their building blocks [31]. It has aided in image classification, image detection, language translation, speech recognition, and etc. There are various types of deep learning model. Among the different types of deep learning model the researcher select convolutional neural networks for Faba beans leaf diseases detection and classification.

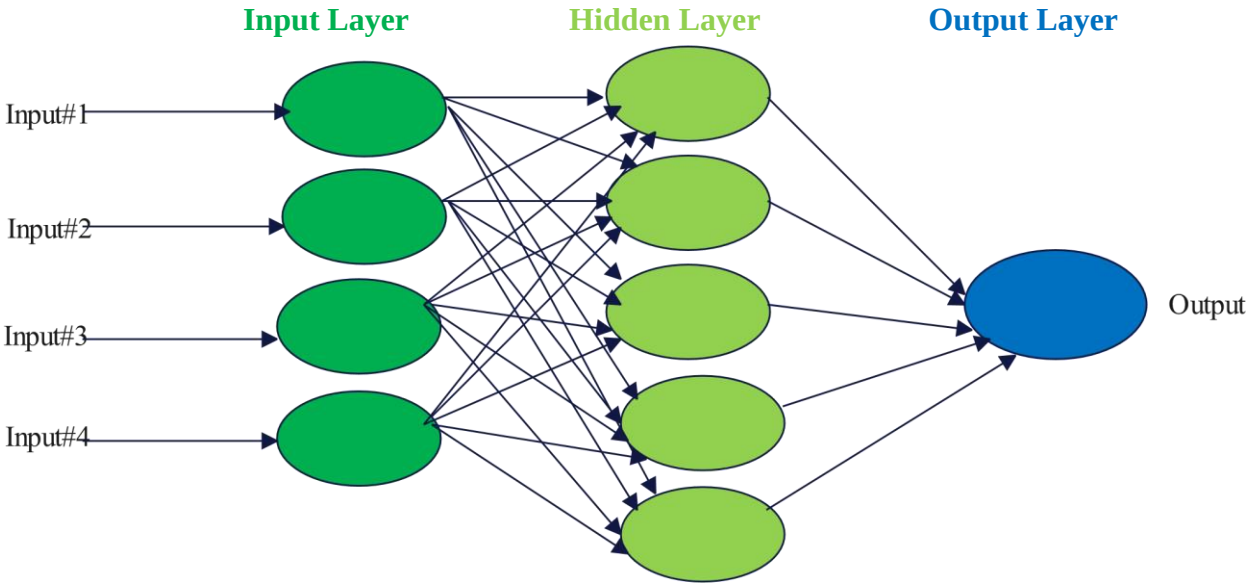


Figure 2-4: Neural Network Layer.

Why the researcher select convolutional neural network?

Convolutional neural network is useful in image processing rather than in other data. Why the researcher use CNN model was it is a very powerful and an efficient models which perform automatic feature extraction to achieve super human accuracy. Convolutional neural network can be more successful since they can model by filtering the entire image on a pixel basis and combine this information with deep networks. While, performing this process, it is passed out in a structure consisting of multiple layers. It does not require hand-crafted feature extraction because, it performs both feature extraction and classification by itself, and also it's a high ability to extract robust and high-level features due to its multiple convolution and pooling layers. It’s a one that

attains the benefit of providing maximum performance and computationally efficiency in image detection and classification. Convolutional Neural Network System predicts the disease by analyzing the first appearing symptoms. These models run on any device, and they are universally attractive, and also has high ability to develop an internal representation of a two-dimensional image.

2.5.1 Convolutional neural network

It is also known as "ConvNet". It is one kind of deep learning model that used in image recognition and processing that is specifically designed to process pixel data. It is the most representative supervised deep learning model. It consists of several layers of neurons, each performing a nonlinear task on the outputs of the layer before it in a linear transformation. It is used to analyze visual image by processing data with grid like topology. It uses CNN to either recognize the image or to classify the image. It uses a two-dimensional convolutional layer to process the data of a two-dimensional. CNN, which can easily learn a feature from the input data and extract a feature directly from the image. In this process, the relevant feature are learned while the network train on collection of images. It learns to detect different feature of an image using many hidden layers, these hidden layer increase the complexity of learned feature. It can automatically detect the important feature without any human supervision. Due to this, there is no manual feature extraction and no need of identification of feature used to classify image in Convolutional Neural Network.

It can easily identify and classify objects with minimal amount of pre-processing. It is a more scalable approach to image classification, object identification, and computer vision operation. Convolutional neural network comprises different layers includes convolutional layer, pooling layer, and fully connected layer [32].

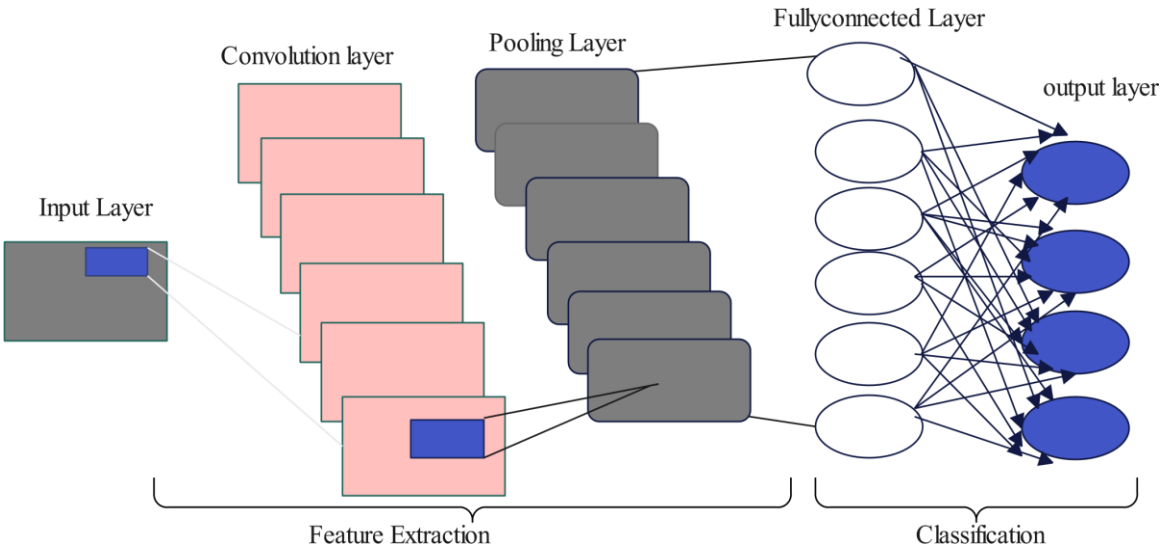


Figure 2-5: A general convolution neural network architecture.

2.5.1.1 Convolution Layer

Convolution layer is the first and one of the core building block of the convolutional neural network. This layer hold the raw pixel values of the training image as input, which is used to extract the various features from the input images. Convolution layer is to perform an operation called a convolution. Convolution is a simple mathematical operation that performed to produce a convolutional map: it is performed between the input image and a filter of a particular size MxM, it contains a set of filters whose parameters need to be well-read, the height and weight of the filters are smaller than those of the input volume, every filter is convolved with the input volume to compute an activation map made of neurons, by sliding the filter over the input image the dot product is taken between the filter and the parts of the input image with respect to the size of the filter MxM [32]. The output is named as the Feature map which gives us information about the image such as the corners and edges. The convolution layer extracts low-level features like edges, lines, and corners. Later, this feature map is fed to other layers to find out several other features of the input image. The convolution layer has far fewer weights than in a fully connected/dense layer, thus followed by a non-linear activation function. The size of the feature map is controlled by depth, stride and padding.

Parameters which helps in adjusting Convolutional neural networks performances are:

Stride: - is the most key advantage to building an efficient convolutional neural networks. Stride is the amount of pixels by which the filter matrix over the input matrix. It shows how many steps we are moving in each steps in convolution. By default, stride value is 1, but the value ranges from 1 to 3 based up on the amount of loss we can be accommodated during convolution and depending up on the input image size. Stride can decrease the resolution of the output.

Padding: - is the process of adding zero around the border of the input matrix symmetrically. The amount of pixels added to an image when it is being processed by the kernel of a convolutional neural network. Adding of padding have many purposes such as firstly, it is used to create space around an element content inside any defined border, space for a kernel is used to cover the image. Secondly, it allows for more accurate analysis of image. Thirdly, it can increase the height and width of the output and, fourthly, it is used to preserve the dimension of the image after convolution and, it allows the feature maps produced by the filter kernels is the same size as the original image.

Filters: - is one of the key assets used in the development of the convolutional neural networks. It is used to detect the key unique features from the images and then these features are multiplied with random weights specified. It increases the depth of the output generated after convolution. The value of the filters are not fixed, if we are using 3 filters then the depth of the output will be 3.

2.5.1.2 Pooling Layer

The layer, which is used in convolutional neural network for combining the feature learned by the convolutional layer feature map. They work by sliding a two-dimensional filter across the three-dimensional feature map and summarizes the features that come in the way of filters. The operation is processed on every slice of the representation individually. It used to decrease the dimensionality of the network and substitute the output of the network at definite location by deriving a summary statistic of the nearby output, this helps in reducing the spatial size of the representation, which reduces the required amount of computation and weights. The primary goal of a pooling layer is to reduce the number of parameters of the input tensor by down-sampling the representation and thus, helps in the reduction of over fitting by the time of training of the model by generalizing the

features in the feature map. Pooling layer perform their operation by either max pooling, which takes the maximum value from the portion of the image covered by the kernel, or average pooling which takes the average of all the values from the portion of the image covered by the kernel. The below figure displays maximum pooling which has input 4*4 , filter 2*2, stride of 2, and 2*2 output.

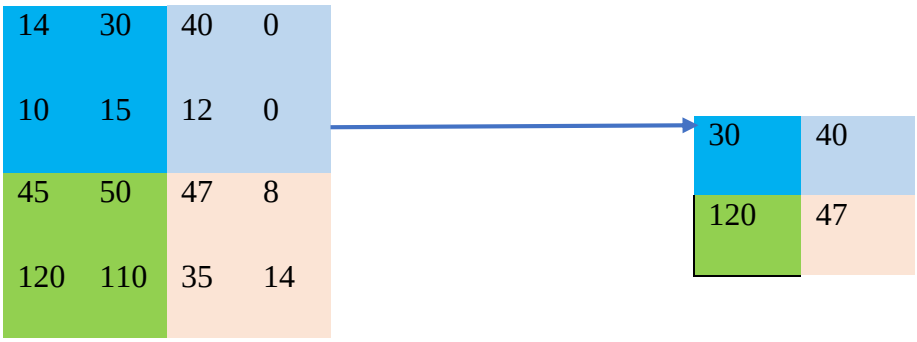


Figure 2-6: Maximum Pooling Operation.

2.5.1.3 Activation function Layer

Activation function layer is one of the most important parameters of the convolutional neural networks model. It is a node that is set either at the end of or in between Neural Networks. By generating a weighted total and applying bias to it, they assist in determining whether or not a neuron should be stimulated. Activations function brings non-linearity to the neural network, which can makes the network capable to learn almost any function representing a question, perform more complex task and provide accurate prediction. This output fed into the next layer as input. The choice of activation function in the hidden layer will manage how well the network model learns the training dataset, and in the output layer will define the type of predictions the model can make. There are different commonly used activation functions such as:

ReLU (Rectified Linear Unit) Activation Function

The Rectified Linear Unit (ReLU) layers, is an activation layer linked after a convolutional layer to generate non-linearity in the network. It is the most commonly used activation function in almost all convolutional neural network of deep learning model. ReLU activation function should only be used in the hidden layers. It helps the network to learn harder decision functions and reduce the over fitting. ReLU converts a negative number to zero and carry forward positive number, in which the feature is not useful the network make it zero, else carries it forward as such. It is easy to

compute and does not saturate, does not cause the Vanishing Gradient Problem, and it is simple and efficient. ReLU has some major problem such as : firstly, it has the issue is that all the negative values become zero directly, which reduces the ability of the model to fit or train from the data properly, if the output is zero for all negative input, then it causes some nodes to completely die and not learn anything. Secondly, ReLU is of exploding the activations since it is higher limit, this sometimes leads to unusable nodes.

Softmax Activation Function Layer

Softmax activation function is the last layer in CNN. They give the probability distribution of classes, in which the class with the highest probability will be selected as the predicted class. It is a more generalized logistic activation function which is used for multi-class classification. The Softmax activation function can be used in a classifier only when the classes are mutually exclusive. It is used to converts the scores to a normalized probability distribution, which can be displayed to a user or helps as input to other systems. The output of the Softmax provides us the likelihood of a particular image belonging to a certain class.

Batch Normalization

Batch normalization is a normalization method done among the layers of neural network instead of the raw data, and it allows every layer of network to learning independently. It is done along mini-batch, instead of the full dataset. It used to make artificial neural networks faster and more stable through normalization of the input layer by re-centering and re-scaling. It can be normalized, to the input by subtracting the mini-batch mean and dividing it by the mini-batch standard deviation. It is used to address the internal covariance shift with the feature map. It unifies the distribution of feature map value by setting them zero mean and unit variance. Batch normalization serves to speed up training and using higher learning rate and making learning rate easier.

2.5.1.4 Dropout Layer

Dropout has seen increasing use in deep learning for CNN, which is known to work well in fully-connected layers. It introduces regularization within the network. This ultimately improves generalization by randomly skipping some units or connections with a certain probability. It randomly drops out some unit in every iteration and produce several thinned network architecture.

It finally selected one symbolic network with small weight. Dropout layer increases the performance of the network.

2.5.1.5 Fully connected layer

It is mostly used at the end of the network for classification. It is a Multilayer Perceptron, composed by three types of layers: input, hidden, and output layers: the initial layer receives the features generated by the CNN, the hidden layer is a sequence of neurons with weights that will be learned in the training step [32]. Increasing the number of hidden units in the layer can increase the learning ability of the network, but there is a rises of a saturation accuracy of the network. The fully connected layer consists of the weights and biases along with the neurons. It is used to connect the neurons between two different layers, in which the layers are usually placed before the output layer and form the last few layers of a convolutional neural network architecture. In this layer, the output of the previous layer is fed into the fully connected layer, and the dot product of weight vector and input vector is computed in order to obtain the final output. The fully connected layer in the architecture contains the same amount of output neurons as the number of classes to be recognized. Each fully connected layer is followed by a nonlinear function.

2.6 Model Evaluation metrics

It is used to measure the generalization performance accuracy of a model on the feature of unseen or out of sample data. There are various evaluation metrics that are commonly used for evaluating the performance of this model, those were accuracy, precision, recall, F1score, and confusion matrix.

2.7 Related Works

Here, the researcher focused on the related work done on plant leaf diseases detection and classification using different techniques. Some of them discussed below:

The study [33] had developed Ethiopian coffee leaf diseases identification using deep learning features. This study, identify four major types of coffee leaf diseases. The researchers, collects, 1280 images of the coffee leaf diseases from the different regions of Ethiopia. They use 320 images per classes. They compared different filtering techniques to avoid noises from coffee leaf images includes Gaussian, median filtering, and the hybrid of the two, they have got better result from the hybrid of the two filtering techniques than individuals. They have used K-Means clustering for segmentation and convolutional neural network for feature extraction. They split augmented coffee leaf image dataset into 80% train dataset and 20% test dataset. The researchers again compared among the two classifier: CNN-Softmax classifier and CNN-SVM classifier. Among the two classifiers, in terms of performance and computational time, support vector machine classifier performs better than Softmax classifier. The proposed model achieved an overall classification accuracy of 96.5% with support vector machine classifier. They trained the model for 20 epochs, batch size of 32, and an initial learning rate of 0.001. They used small size dataset set, but deep learning need large number of dataset to get high accuracy.

The study [14] had built a deep convolutional neural network to conduct illness recognition and classification for soybean leaf spot using affected areas of disease spots. The researcher's collects a dataset for this work were downloaded from databases on the internet, the dataset includes 13243 leaf images of soybean crop with 5 classes of soybean leaf spot disease such as Alternaria Leaf Spot, Phyllosticta Leaf Spot, Target Leaf Spot, Frogeye Leaf Spot, and Bacterial Blight. They segment the affected area of diseases spot from the leaves images using the unsupervised fuzzy clustering algorithm and unsupervised optimal clustering algorithm. They split the dataset in two sets, 80% for the training and 20% for testing. They used data augmentation and drop out to enlarge the dataset and overcome over-fitting, respectively. They evaluate the performance of the model using the confusion matrix and three evaluation metrics for confusion matrix reports such as F1-score, Precision, and Recall on the test dataset. The researcher concluded that deep convolutional neural network has difficulty to recognize the leaf spot disease and predicting the correct class using the spot diseases segmented. In the proposed model, the researchers considers only the

spoted part of a soybean plant not the whole parts of the images of the leaf, plus it is not considered the healthy leaf of the soybean plant. More parts of the dataset images are not classified clearly to their classes.

The study [34] had developed a deep learning based approach to detect the leaf diseases of the plant, that identifies whether leaves are healthy or infected. The researchers take the images of leaves from different plants. They consider three detectors for their study, those are Faster Region-Based Convolutional Neural Network, Region-based Fully Convolutional Networks, and Single Shot Multibox Detector. They use the image with size 256*256 to minimize the time of training. The researchers are not simply detect the types of diseases only, also identifies the infection status of the diseases and tries to give suitable organic fertilizer for those diseases. In this study, the authors are not identified the name of diseases that they detect, the number of classes that recognized and the number of dataset they used for their study is not clearly mentioned.

The study [35] had develops a deep learning based model that detect the leaf of cotton disease and pests. The researchers used convolutional neural network method among a deep learning techniques for their study. They identify one healthy and three common types of cotton leaf diseases and pests such as bacterial blight, spider mite, and leaf miner. They collect, totally 2400 images for both healthy and infected of cotton leaf, uses 600 images per-classes. They have been used K-fold cross validation strategy to dataset splitting and boosted generalization of the convolutional neural network model. They have been used 80% for training and 20% for testing. The models implemented by using python and trained the model on the deep learning package called keras, tensorflow backed and jupyter. They calculate the performance of the model using the confusion matrix and four evaluation metrics for confusion matrix reports includes F1-score, Precision, Recall, and Accuracy on the test dataset. The model succeeded accuracy of 96.4% for classifying classes of leaf disease and pests in cotton plants.

The study [36] had presents a convolutional neural network for tea plant diseases detection, that identifies whether the leaf is healthy or infected. They implemented by using concatenated convolutional neural network such as GoogleNet, Xception, and Inception-ResNet-v2. They used 4727 images of tea leaves, and identify three types of diseases. They used 80% of the data for training, 10% for data validation, and 10% for testing data. The proposed system has achieved an accuracy of 89.64%. In the proposed system, the researchers have been not use batch normalization

this result for a long time training, in addition, the parameter cannot be adjusted well, and the methodology they used is not explained in detail.

In paper [37] the authors discussed a deep learning based approach were developed to perform banana diseases detection and classification using leaves images of healthy and infected plant. They use convolutional neural network from deep learning approach and use LeNet architecture for their study to classify image datasets. They identify one healthy leaf and two types of banana leaf disease, totally identifies three class of banana plant leaf. They use 3700 images for all classes. They determined a parameter empirically according to a series of experiments carried on the whole dataset that give the best results of classification.

In paper [38] the authors discussed the classification of maize leaf diseases by using a convolutional neural network that identifies whether the leaf is healthy or infected. The researchers used 200 images and identifies four class includes one healthy leaf and three diseased leaves. They had uses 50 images per class. The researchers had been tested seven architecture of convolution neural network such as AlexNet, virtually geometry group 16, VGG19, GoogleNet, Inception-V3, residual network 50 and ResNet101. Among this seven architectures based on the testing result, AlexNet has high accuracy from the others for this model. They also used machine learning method for classification including k-nearest neighbor, decision tree and support vector machine, among this machine learning method support vector machine has high accuracy based on their testing result. The proposed model follow two main steps, those are feature extraction and classification. In the proposed system, the researchers uses very low dataset. Since deep learning method uses large amount of data set to get high accuracy.

In papers [39] the authors presents an automated method that detects the mango leaf healthy or infected by using convolutional neural network from deep learning approach. They used 1200 images and identifies six classes. They use image with size 256*256. They trained a model with 100 images per classes and the remaining 600 images constituting of 100 images per class was tested. The proposed convolutional neural network model achieve accuracy of 96.6%.

In papers [40] the authors presents about the classification of apple leaf diseases using convolutional neural network. They identify four classes of apple leaf image includes one healthy and three diseased leaves classes. They were used total, 3642 leaf images. They had been carry out

a noise removal in the images, identification of the relevant area and histogram equalization on the YUV color space. The number of dataset used for all classes are not equal, due to that data augmentation was applied for the minority classes with the SMOTE method. After pre-processing step, they perform data partitions, they use 90% for training dataset was used as training data and 10% as validation data. The researcher's uses different feature extraction method includes DenseNet121, DenseNet201, InceptionResNetV2, InceptionV3, and ResNet50V2. Among this feature extraction network models, ResNet50V2 gave the highest accuracy value for classification of the proposed method than the others models. All pre-trained network has the same adjusted epoch value, while the pre-trained networks is the need for advanced computers. The training period does not take long since the data set is small. The extracted feature were classified with convolutional neural network with accuracy of 99%.

Authors	Years	Title	Methodology used	Accuracy	Gaps
M. Akila and P. Deepan	2018	Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm	Compare Faster Region-Based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Networks (R-FCN) and Single Shot Multi-box Detector (SSD).	--------	- They cannot stated the Methodology used clearly. - They did not mention the classification accuracy achieved. - They cannot classify different classes of diseases. -They cannot mentioned the number of dataset used for the study.

Linigerew Mengstie shita and Mogalla Shashi	2021	Ethiopian Coffee Leaf Diseases Identification Using Deep Learning Features	- Gaussian filtering, median filtering and the hybrid of the two. - K-means clustering for segmentation. - Convolutional neural network for feature extraction. - Support vector machine classifier.	96.5%	-They were used small dataset, while deep learning require large data set.
Jihen Amara, Bassem Bou-aziz, and Alsayed Algergawy	2017	A deep Learning based Approach for Banana Leaf Diseases Classification	-LeNet architecture. -Convolutional neural network for feature extraction. -Stochastic gradient descent (SGD).		- They did not mention the classification accuracy achieved. -The performance of the model is not measured. - Dataset size is too small.
Jiangsheng Gui and Mor Mbaye	2019	Identification of Soybean Leaf Spot Diseases using Deep CNN.	-Unsupervised fuzzy clustering algorithm and unsupervised optimal cluste-ring algorithm.	93.45 %	-They considers only the spoted part of a soybean plant not the whole parts of the images of the leaf. -The healthy leaf of the soybean plant isn't included in the study.

			- Support vector machine (SVM). -VGG16 architecture.		The dataset images is not classified clearly to their classes. -gathers the images from the internet. - Over-fitting in case of noisy data.
S.Arivazhagn, And S.Vineth Ligi	2018	Mango Leaf Diseases Identification Using Convolutional Neural Network	-Convolutional neural Network for feature extraction.	96.6%	- Data set size is too small. -The methodology used for their study is not mentioned.

Table 2-2: Summary table of related work.

Summary

The researchers used different techniques for diseases detection and classification, but each of the techniques has some limitation. More of the work done by gathering data from the internet or public dataset, in addition the number of dataset they use is small, since deep learning algorithm require large number of dataset to get high accuracy. Those researchers detect few classes such as healthy and unhealthy, and some of them are not clearly classified to their class. Some researchers did not apply image processing techniques for plant diseases detection and classification.

To the best of our understanding, there are no studies done to detect and classify Faba bean leaf diseases by using deep learning algorithm. In this study, the system developed by capturing the image. Also, it is found that using technology-based data preprocessing and enough dataset is used to remove over-fitting and has good results and better performance. This study is only capable of detecting three types of disease classes and one healthy Faba bean leaves.

CHAPTER THREE

3. RESEARCH METHODOLOGY

To conduct this study and to answer the specified research question, different methodology are used. In this chapter, the dataset collection method, pre-processing of images, feature extraction, detection, classification, model evaluation techniques, and hardware and software tools that are used to achieve the research objective is briefly discussed.

3.1 Data Collection

For this study, datasets are collected from different Faba beans plant farm. Farm is the primary source to get the image of Faba beans reliably. To recognize clearly, whether the diseases are effected the plant or not, if the diseases affect the plant, which parts are more essential part to be effected by the diseases, and what are the different types of diseases affect the parts of the plant, by present physically with the assistance of experts. Not used secondary data source collect from farm. The study can be conducted in the plantation area of Faba bean in Oromiya region, Arsi Zone, D/Xijo woreda, from the farmer plantation land particular reference to Bucho Silase kebele, Ethiopia, where the dataset can be collected. In this area, different varieties of crops can be grown, among the different varieties, plus crops are the famous widely grown plants due to appropriate weather condition. The zone has a big research center what we call them Asela Agricultural Engineering Research Center (AAERC). This research center performs different investigation includes exploring of new and modified seed, identifying, and classifying different diseases of the crop by using laboratory. The research center has different experts done on the plants. Some of the expert have a long year experience in working on the agricultural investigation, those are simply identifies the healthy and diseased leaf of the plant by eye and to make sure use the laboratory testing. AAERC plays a great role in terms of identifying the various types of Faba bean leaf diseases by the professionals with the help of laboratory and eye. For this study we had collect 5,404 images for all classes.

During data collection, we had used different methods of data collection. The following are the different techniques used for dataset collection of Faba bean leaf diseases detection and classification model.

3.1.1 Observation

It's one of the most essential and common methods of data collection. In this study, the researcher, makes observation before starting its works go the plant farm of the Faba bean and observe the plants.

3.1.2 Interview

After, makes observation of all things in that area regarding with the Faba bean plants, the researcher prepare and analysis different question to ask the expert regarding the Faba beans plant. For this study, three experts are participate. Since, the researcher as before have not a detail knowledge about the plants, we had raised different question for the expert. Among those question few of them are: during observation period more of the land of the farmer at that zone more than 90% is snowing Faba bean plant, in the researcher mind one question is raised, that is, is it Faba bean is not effected by diseases?, the expert answer no: the plant is effected by different types of diseases, sometimes the diseases destroyed all the snowing of Faba bean farm especially rust, but, most of the farmer snowing this plant due to the use in day to day life. And also this plant is very helpful for repairing the lands. The second question, after snowing Faba beans plant at which period of time the diseases attack this plant, the experts says, after three weeks to one month and above. The third question, does the diseases attack all parts or some parts of the plant, starting from single parts and gradually attack all parts. Fourthly, which parts of the plant highly effected by the diseases? The leaves part of the plant highly attacked by the diseases, then stems, flower and pods. After the researcher, ask these and other main question, then directly the researcher observe the plant farm with the expert in detail. Thus, the researcher recognize the leaf part is mainly affected by the diseases.

3.2 Building Dataset

To create a deep learning model for Faba beans leaf diseases detection and classification, collecting and building a dataset for training is necessary. It is necessary to gather and build a new dataset for the accomplishment of this study. To build dataset, we had applied the following procedures.

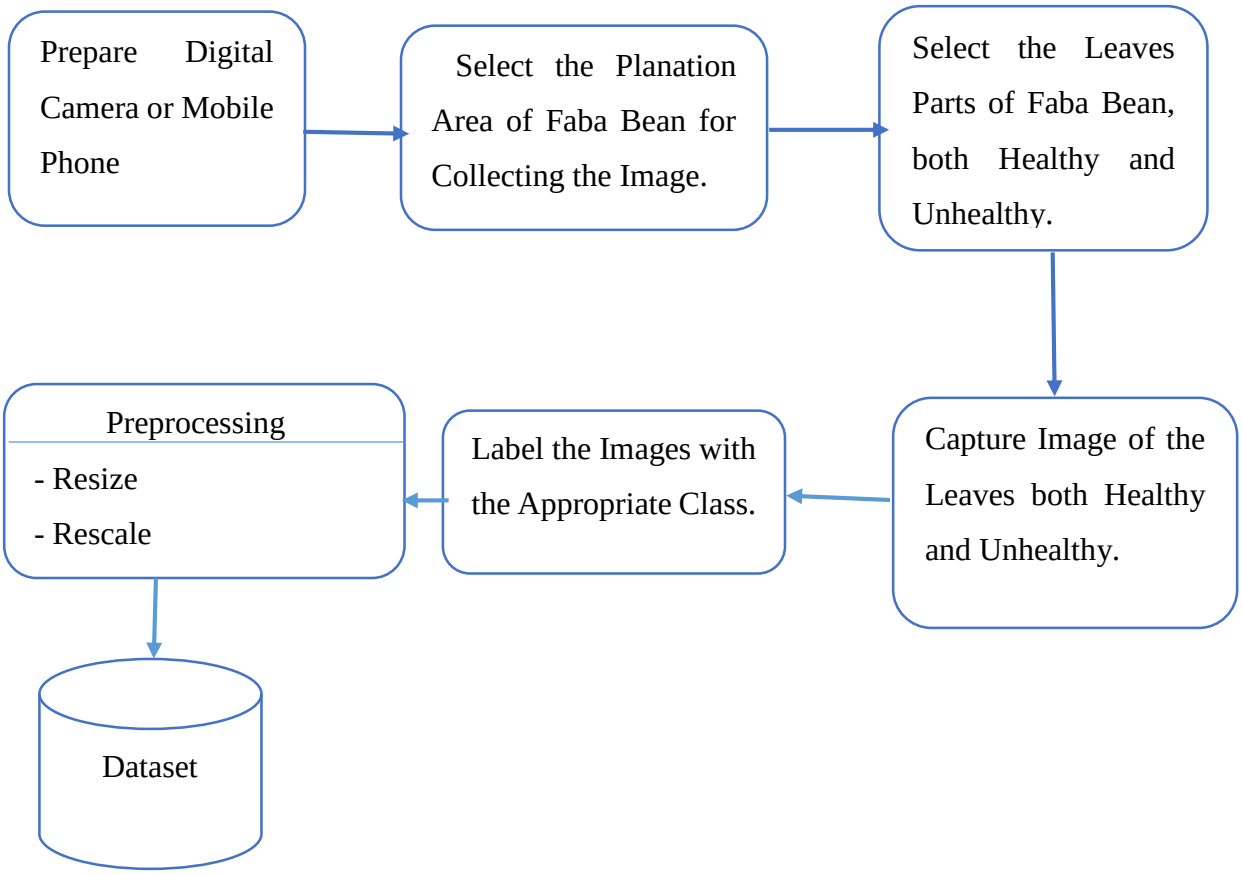


Figure 3-1: Dataset building procedure.

3.3 Faba bean leaf diseases detection and classification modeling

In order to accomplish the deep learning based Faba beans leaf diseases detection and classification model, the following methods are applied.

3.3.1 Image Preprocessing

After collecting the leaf images of Faba bean, the next stage is preprocessing of images. This method of image processing allows the collected images from a plant farm to be resized into unified dimension and scaled to have the same width and height before fed to the learning algorithm.

3.3.2 Feature Extraction

It's used to get the best feature from big data sets by select and combine variables into features, thus, accurately and completely describing the original data and effectively decreasing the amount of data that must be processed. Faba beans leaf diseases has some characteristics that differentiate it from the others diseases. So, identifying and knowing this feature is very important to distinguish one Faba bean leaf diseases from other the other Faba bean leaf diseases. For this study, features are extracted by using convolutional neural network layer called convolution layer. By using different parameters of convolution layer such filter size, kernels and padding. This can automatically extract the feature of image. This can happen due to the power of Convolutional neural network.

3.3.3 Detection

In this stage, after extraction of learnable features and the development of the model, then detection of the images is performed. This method is applied to identify or detect whether the Faba bean leaves are diseased or healthy.

3.3.4 Classification

After extraction of learnable features, development of the model, and detection of the image, the next stage is classification of Faba beans leaves images in to their corresponding class. For this study used softmax activation function to classify the images in to their classes.

3.4 Evaluation

Lastly evaluate the performance of the model by using different performance evaluation metrics such us confusion matrix, precision, recall, F1 score, and accuracy. Model evaluation metrics is used to measure the generalization performance accuracy of a model on the feature of unseen or out of sample data.

3.4.1 Confusion Matrix

It's techniques of performance measurement mechanism that's used to explain the performance of the classification model. It visualizing the performance of the prediction model in tabular ways. Confusion matrix is compares the actual target value with those predicated by the machine learning

model. This method is very useful since, the give direct comparison of values like true positive, false positive, true negative and true positive.

		Actual class (True class)	
		Positives	Negatives
Predicated class	Positives	True positive (TP)	False positives (FP)
	Negatives	False negatives(FN)	True negatives (TN)

Table 3-1: Confusion matrix.

3.4.2 Precision

It's the ability to identify only the relevant data point. It's the ratio of the result which is relevant. It's calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes. It is a good evaluation metrics to use when the cost of a false positive is very high and the cost of a false negative is low.

$$\text{Precision} = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False positives (FP)}} \quad \ldots\ldots\ldots\ldots\ldots \quad \text{Equation 2. 1}$$

3.4.3 Recall

It's the number of correct positives prediction made out of all positives predication that could have been made. It measures the models ability to detect positive samples. It helps when the cost of false negative is high. It's calculated as the sum of true positives across all classes divided by the sum of true positives plus false negatives across all classes.

$$\text{Recall} = \frac{\text{True positives (TP}}{\text{True positives (TP)} + \text{False negatives (FN)}} \quad \ldots\ldots\ldots\ldots\ldots \quad \text{Equation 2. 2}$$

3.4.4 F1 Score

It's the harmonic mean of precision and recall. This metric uses both false positives and false negatives under consideration**.** It performs well on an imbalanced dataset. It's the overall measure of a model accuracy that combines precision and recall. Fl score implies a high precision also as high recall. It's considered as prefect when F1-score is one, while the model is fail when it's zero. It's very useful especially for odd class distribution.

$$\mathrm{F1-Score} = \frac{2 * (\mathrm{Recall} * \mathrm{Precision})}{\mathrm{Recall} + \mathrm{precision}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{Equation 2. 3}$$

3.4.5 Accuracy

It's the ratio of correct prediction for the test data. It can be measured by dividing the number of correct prediction by the number of total predictions.

$$\mathrm{Accuracy} = \frac{\mathrm{True\ positive\ (TP)} + \mathrm{True\ negative\ (TN)}}{\mathrm{True\ positive} + \mathrm{True\ negative} + \mathrm{False\ positive} + \mathrm{False\ negative}} \ldots\ldots \quad \text{Equation 2.4}$$

3.5 Materials and Tools

3.5.1 Software Tools

There are different software tools, which are used for the implementation of Faba beans leaf diseases detection and classification model. We had conduct investigation of available software tools to select the appropriate tools for implementation, testing and developing a prototype of the CNN algorithm for the proposed model. During the study, we have seen that there are tools that are common for both deep learning and machine learning algorithms and specific only for either deep learning or machine learning algorithms. Before selecting the tools, we have considered some criteria which are helpful to select the appropriate software tools with their corresponding libraries. The other criteria are to select tools with enough learning materials such as free video tutorials, existing experience, and different written programming books. This study used python as a programming language with Tensorflow and Keras libraries and Google Colaboratory as a tool for implementation, testing, and developing a prototype.

Google Colaboratory

It's used to write and execute python code through their browser. It allows to use and share jupyter notebook with others without having to download and install anything. It has a number of pre-installed packages such us Numpy, Scipy, pandas, Tensorflow, pytorch and the images is saved on the cloud. The colab provides free resource like GPU.

Notebook

It's an open source web application that is used to create, edit, and share document that display the input and output of a python. It's easy to use and run in a web browser. We have used the notebook for this study to implement the coding part.

Python

It's an interpreted high-level and a dynamically-typed garbage-collected programming language for general-purpose programming. It can be implemented and used to do anything from webpages to scientific research. We had used python programming language for this study, since it contains several packages for image processing, and the familiarity of the researcher with python programming language is considered. Python contains several packages that help for preprocessing, feature extraction, classification, and evaluation of the model.

Tensorflow

It is free and an open source library that's used to design, build, and train deep learning models. It was created by Google. It is useful in executing various applications of mainly deep and machine learning algorithm. Tenserflow are often accessed and controlled by different programming languages, primarily Python. All the Computation in Tensorflow involves tensors that represent all kinds of data. Tensorflow also uses a graph framework for graphical representation of the series of computations during the training.

Keras

It is free and an open source software library. It is a high level neural network library that run on the top of Tensorflow framework and written in python. It's employed as an interface for the tensor flow library. It is most user-friendly, since it is built in python programming language. We had used for creating and training deep learning models, as well as it's a powerful and easy-to-use

free open source python library for developing and evaluating deep learning models. It contains pre-trained convolutional neural network models such as VGG16 and Inception that we during the experiment.

3.5.2 Hardware Tools

In order to accomplish the proposed model, we had used a hardware tools. The hardware tools that we had used in this study is smartphone, which plays an essential role during the period of data collection, by capturing images of Faba beans leaves from the plant farm. Since, without the presence of dataset, there is no the implementation of the proposed model.

CHAPTER FOUR

4. ARCHITECTURE OF FABA BEAN LEAFS DISEASES DETECTION AND CLASSIFICATION MODEL.

This chapter clarifies the proposed Faba bean leaf diseases detection and classification method. First, the general overview of Faba bean leaf diseases detection and classification architecture is discussed. Second, the model pass through a serious of step starting from image acquisition, image preprocessing, feature extraction, detection and classification, each and every steps of the model is described in detail. Thirdly, modeling, train the model, evaluate the model and evaluate the performance of the model using confusion matrix, accuracy, recall, precision and F1 score.

4.1 Architecture of the model

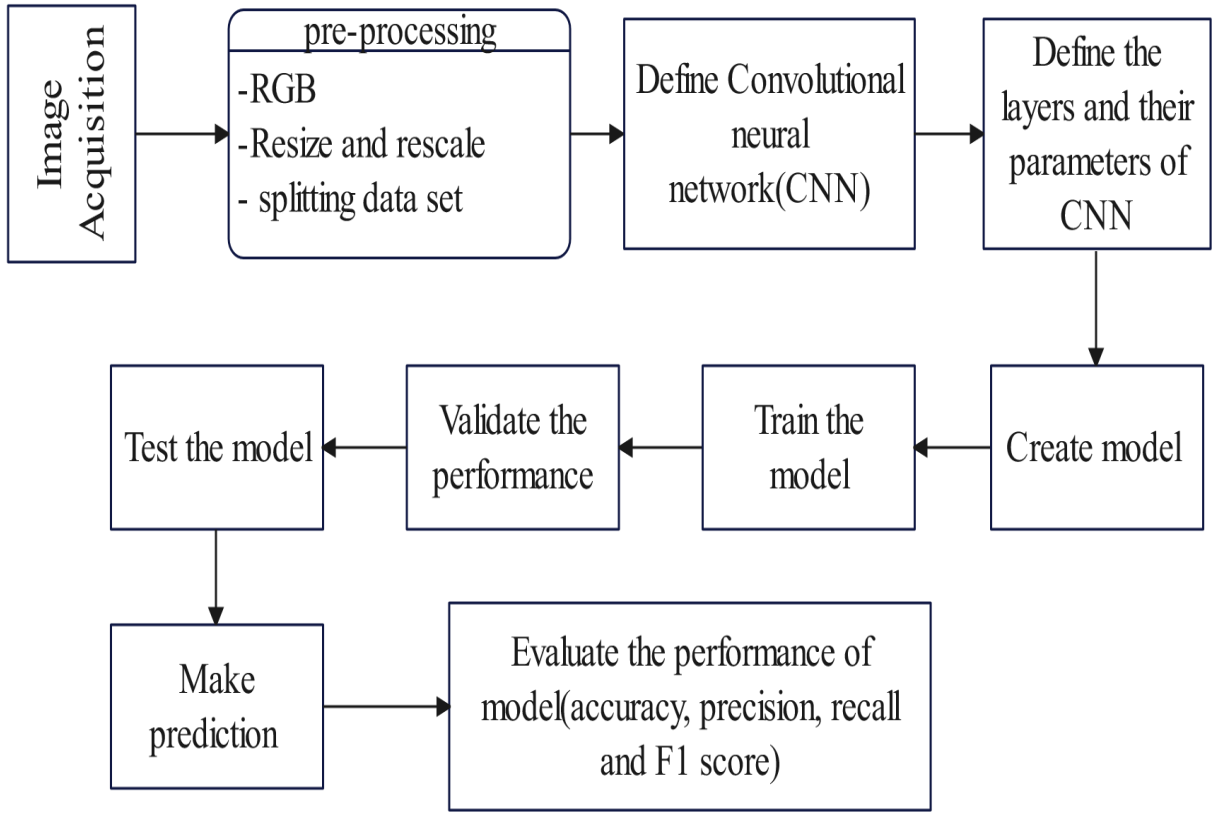


Figure 4-1: Architecture of Faba bean leafs diseases detection and classification.

4.1.1 Image Acquisition

It is the first step in any image processing system. In this method, we had performed capturing of the images for preparing the dataset. For this study, the dataset had been created by capturing image by using smartphone from Arsi Zone, D/Xijo Woreda, of Bucho Silase kebele of different plant farm of Faba beans from different farmer land during four different time's period (the end of June, June-July, July-August, and August-September) in order to reduce the loss of diseases feature due to various phenomenal. The captured images consist of four class, these are three types of diseased class such as chocolate spot, ascochyta blight, and Faba beans rust, and one healthy Faba beans leaf class. The captured image were in the JPG format. The image were captured by using Galaxy A51 with android version 11 which has resolution of 48M pixel (1080x2400 pixel). The totally images collected were 5,404 for all classes. Among this, 4800 images were used for this study. At the first stage, the expert identifies the different types of diseases with their symptom it's shown, this process makes the work simply to group to their classes. In the figure below, sample of Faba bean leaf images both diseased and healthy were shown. A. Ascochyta blight, B. Chocolate spot- Botrytis, C. Healthy leaf, and D. Rust.

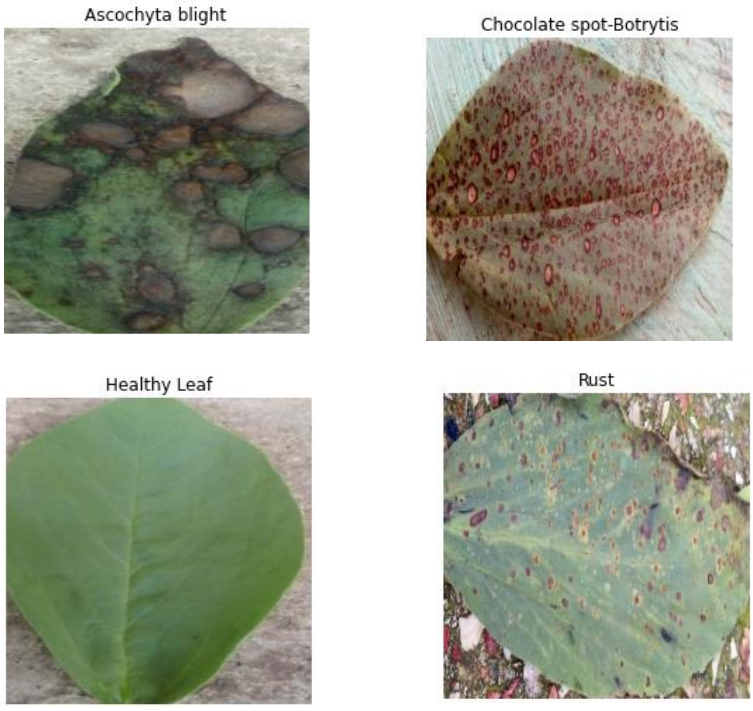


Figure 4-2: Sample images of Faba bean.

4.1.2 Image Preprocessing

This method of image processing allows the collected images from a plant farm to be cleared of unnecessary features. In order to have a cleaned dataset before they are used by model training and inferences, and to make it convenient for feature extraction, preprocessing of images is performed. Image preprocessing techniques are used for removing background noise or sharpen, adjusting the images into the similar size, identical width and height, and adjusting the brightness. This study had used image preprocessing for resizing and rescaling images. All images was resized into 256*256.

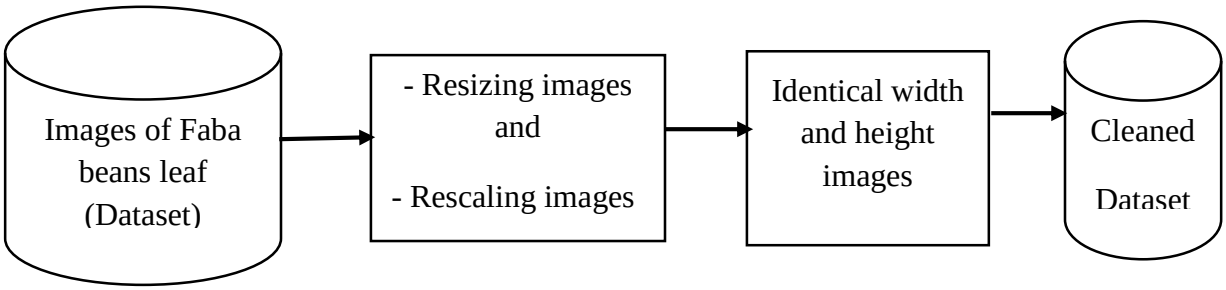


Figure 4-3: Image preprocessing step.

4.1.3 Dataset splitting

It is a method of partitioning the dataset in to multiple subset in order to evaluate how the model performs and in order to remove bias to the training data. In this study, the total dataset was splitting into training, validation and testing dataset.

Training dataset

It's the set of data that's used to train and make the model learn feature in the data.

Validation dataset

It's used to validate our model performance during training. This validation process, it's like a circuit tells whether the training is moving in the right direction or not. Model is trained on the training set and similarly the model evaluation is performed on the validation set after each epoch. Validation data set is used to prevent our model from over fitting.

Testing dataset

It is a separate set of data, which is used to test and predict the model after completing training. Using test data supports our model to be more robust and effective because the model evaluates the result of unseen data.

4.1.4 Model Selection

The researcher were used a deep learning algorithm which is convolutional neural network for this study. CNN is selected on different research and literature conducted in image processing mainly in image recognition and image classification. It used various algorithm for feature extraction, classification of image, training, testing, validating as well as for calculating the accuracy of the model. One of the best quality of convolutional neural network is, it extract features automatically from images without human interference as well as by using a single framework it performs feature extraction and classification naturally. During classification task CNN provided high generalizability and good performance.

Convolutional neural network has two main blocks. The first block of CNN which is perform feature extractor. To do this, it performs template matching by applying convolutional filtering operation. The second blocks of CNN performs classification. It performs this function at last layer of this block by using activation function.

4.1.5 Model Description

Layers Description

Convolutional neural network has different layer. Each layer has various parameters to build up the model. In this study we had used different kinds of parameters for each layers to build Faba bean leaf diseases detection and classification model. We had used each parameters by checking to the model that means by testing and compare the result among the parameters. Then, select the better performed parameters during build up the model.

4.1.5.1 Input Layer

The input layer fetches the first data into the system. The input layer of our model accept the RGB images of size 256*256*3 with four different classes (Ascochyta blight, Chocolate spot- Botrytis,

Healthy leaf and Rust). This layer only fetches the RGB images to the convolutional layer without any computation.

4.1.5.2 Convolution layer

It's the first layer of convolutional neural network. It's used to detect the presence of set of features from the input images. To do this, it performs convolution filtering. It drags a window representing the feature on the image to calculate convolution product between the feature and each portion of the scanned image. Then, a feature is seen as a filter. Feature has not particular formula rather learned by the network during the training phase. This layers contain two hyper parameters those were number of features and size of the feature. The study had use filter size 3*3 with the same padding. In the Faba bean leaf diseases detection and classification model there are seven convolution layers:

- The first convolution layer of the model filters 256*256*3 input image with 32 kernels.
- The second convolution layer takes the input of the first layers and filter it by using 64 kernels of size 85*85*64.
- The fourth convolution layer has128 kernels of size 28*28*128 and
- The fifth convolutional layer also has 256 kernels of size 9*9*256. Wholly convolution layers of Faba beans leaf diseases detection and classification model was used softmax activation function.

4.1.5.3 Pooling Layer

The layer found between two layers of convolution. This layer receives feature maps from convolution layer and performs pooling operation to each of them. Its function is to progressively decrease the size of the image and keep their important characteristics as well as it decreases the number parameters and calculation in the network. In this research we had used four maxpooling of (3*3) size. Poling layer has two hyper parameters those were window size and window stride. In the Faba bean leaf diseases detection and classification model there are four max-pooling layers:

- The first max-pooling layers decreases the output of the first convolution layer with filter of size 85*85*32.
- The second max-pooling layers has a filter of size 28*28 with stride 64.
- The third max-pooling layers has a filter of size 9*9 with stride 128.

- The fourth max-pooling layers has a filter of size 3*3 with stride 256.

Flatten Layer

Flatten Layer is between the convolution layer and the fully connected layer. The layer which is taking the pooled feature map that is generated in the pooling step and translate a high-dimensional input into one-dimensional vector. It's used to reduce the number of parameters and improve the computational efficiency of the network. This layer is done as input to the dense layer and it can changes the shape of data into the correct format for dense layer to interpret.

Dense Layer

Dense layer is the other name of fully connected layer. The layer that mostly used for the output layer.

Dropout Layer

This layer is a mask that nullifies the contribution of some neuron towards the next layers and leaves unmodified all others. It randomly set input unit to zero with a frequency rate at each step during training. This layer is used only when training is set true. It is used to prevent the over-fitting issue. The dropout values ranges from 0 to 1. We had used 0.25 and 0.5 dropout values for model training.

Batch Size

It is another method to regularize a Convolutional neural network. It allows every layer of the network to do learning more independently and normalize the output of the previous layers. It's used to decreases training time and result in better performance. We had used batch size 32 for this study.

Epoch

Epoch is one undergo all of the rows in the training dataset. The training process will run a fixed number of iterations through the dataset called epochs. The model was trained by using different epochs starting from 5 to 50.

4.1.5.4 Output layer

The final layer of convolutional neural network that produces given output for the model. This layer has a neuron called activation function with softmax that's used for multi-class classification. Due to the model is designed to classify four classes. It was defined sequential model called categorical classification, since our model is multi-classes.

```
[ ]  model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      896

 activation (Activation)     (None, 256, 256, 32)      0

 batch_normalization (BatchN  (None, 256, 256, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 85, 85, 32)       0
 )

 dropout (Dropout)           (None, 85, 85, 32)        0


 conv2d_1 (Conv2D)           (None, 85, 85, 64)        18496

 activation_1 (Activation)   (None, 85, 85, 64)        0

 batch_normalization_1 (Batc  (None, 85, 85, 64)       256
 hNormalization)


 conv2d_2 (Conv2D)           (None, 85, 85, 64)        36928

 activation_2 (Activation)   (None, 85, 85, 64)        0

 batch_normalization_2 (Batc  (None, 85, 85, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 28, 28, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 28, 28, 64)        0
```

conv2d_3 (Conv2D)	(None, 28, 28, 128)	73856
activation_3 (Activation)	(None, 28, 28, 128)	0
batch_normalization_3 (BatchNormalization)	(None, 28, 28, 128)	512
conv2d_4 (Conv2D)	(None, 28, 28, 128)	147584
activation_4 (Activation)	(None, 28, 28, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 28, 28, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 9, 9, 128)	0
dropout_2 (Dropout)	(None, 9, 9, 128)	0
conv2d_5 (Conv2D)	(None, 9, 9, 256)	295168
activation_5 (Activation)	(None, 9, 9, 256)	0
batch_normalization_5 (BatchNormalization)	(None, 9, 9, 256)	1024
conv2d_6 (Conv2D)	(None, 9, 9, 256)	590080
activation_6 (Activation)	(None, 9, 9, 256)	0
batch_normalization_6 (BatchNormalization)	(None, 9, 9, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 256)	0
dropout_3 (Dropout)	(None, 3, 3, 256)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 1024)	2360320

```
activation_7 (Activation)    (None, 1024)              0

batch_normalization_7 (Batc  (None, 1024)              4096
hNormalization)

dropout_4 (Dropout)          (None, 1024)              0

dense_1 (Dense)              (None, 4)                 4100

activation_8 (Activation)    (None, 4)                 0

=================================================================
Total params: 3,535,236
Trainable params: 3,531,332
Non-trainable params: 3,904
_________________________________________________________________
```

Figure 4-4: Summary of Convolutional neural network model.

4.1.6 Feature extraction

Convolutional layer is the first layer in Convolutional neural network that is used to extract the different feature from the RGB images. Convolutional layer can extract feature from input images by performing template matching by applying convolutional filtering operation, this layer filters the images with several convolutional kernels and return feature map. Which are then normalized with activation function and resized. To get a feature map the input image is convolved by the kernel filter to extract important features in a given region. During convolution the researcher considered important parameters such as padding, stride and filter size. In this research the researcher apply kernel size of (3*3) and stride of 3*3 and padding “same”. The researcher had used 32, 64, 64, 128, 128, 256, 256 number of filters (number of neurons) for convolution operation in the input convolution layer different filter was applied to extract more features from the input image. After the extraction of important feature all the important information is pass to fully connected layer or dense layer.

4.1.7 Detection

After extraction of learnable features and development of the model the next step is detection and classification of Faba bean leaf disease using four classes. During data collection process the experts detect each disease type and explained about their symptoms and their effects on the Faba bean leafs.

4.1.8 Classification

After extraction of learnable features, development of the model, and detection of the image, the next stage is classification of Faba beans leaves images in to their corresponding classes. The last layers of Convolution neural network is fully connected layer that performs classification. This layer usually utilize an activation function called softmax. Softmax is an activation function that used in this study to classify images into multi-classes. In our model softmax activation function classify Faba bean leaf images into four classes (Chocolate spot-Botrytis, Ascochyta blight, Rust, and Healthy Leaf).

4.1.9 Evaluation

The performance of a model depends on the features of the dataset to be classified. To measure the performance of convolutional neural network model, various empirical tests have been performed. On measuring the efficiency and effectiveness of the proposed Faba bean leaf diseases detection and classification model, using different performance evaluation metrics like Confusion matrix, accuracy, precision, Recall, and F1 score.

CHAPTER FIVE

5. IMPLEMENTATION AND EVALUATION

This chapter discuss how the model is implemented and the achieved result of the experiment by using methodology in chapter three and the proposed model presented in chapter four. Generally, the presented sub-section in this chapter are dataset description, implementation environment, preprocessing, dataset splitting, modeling, train, validate, predict and evaluate the model.

5.1 Dataset Description

The dataset had been created by capturing image from different plant farm of Faba beans from different farmer land during four different time's period. The totally images collected were 5,404 for all classes. Among this, 4800 images were used for this study. Since, the performance of deep learning model depends on the number of data they required. Using the small number of dataset resulting in over fitting of the learned model, which will perform poorly in cases of validation. The captured images consist of four classes (Chocolate spot, Ascochyta blight, Rust, and Healthy leaf). Each class was used 1200 images in order to keep data balancing. The figure below shows the number of images for each classes with corresponding diseases names.

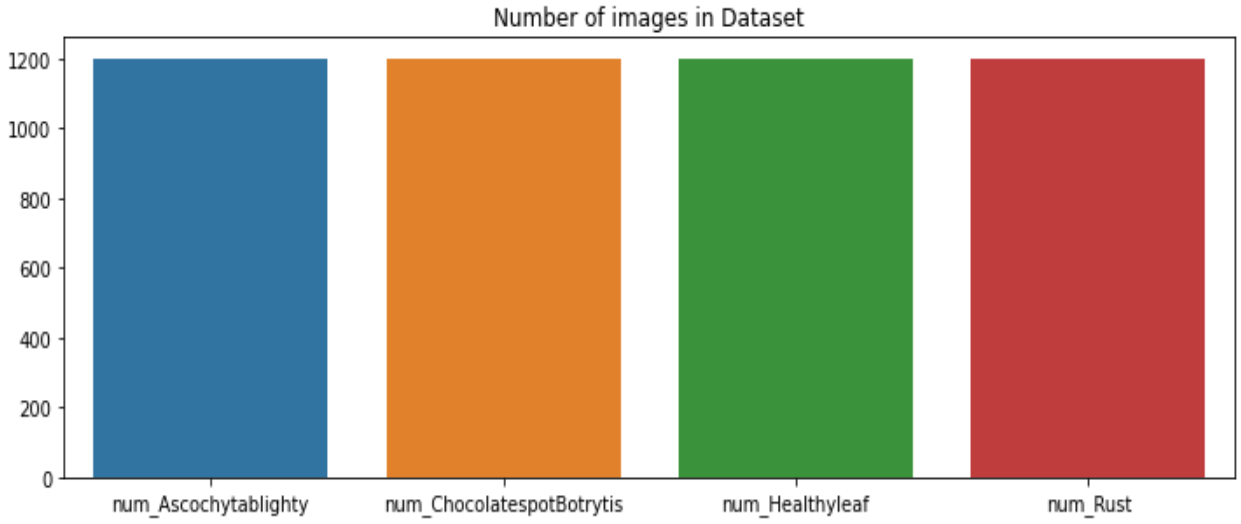


Figure 5-1: The diseases name and number of corresponding images in the dataset.

5.2 Implementation Environment

This study used python as a programming language and using notebook as the main integrated development environment (IDE). Python is chosen because it contains several packages for image classification using Tensorflow with Keras. Python contains several packages that help for preprocessing, feature extraction, classification, and evaluation of the model. The study was firstly taking place on a local windows system but was eventually moved to the Google Colaboratory due to a lack of resources. Google Colab is a superb tool for deep learning tasks. It stores files in Google drive and it allows to figure with bigger dataset and execute complex function. It allows users to write down and execute a python code in their web browsers. It's based on the Jupyter open source and allows you to create and share computation files without having to install any things. In Google Colaboratory, a Deep Learning images instance was created on the cloud, which is optimized for deep learning and has all the specified packages pre-installed, like Pandas, NumPy, Matplotlib, PyTorch, Keras, Tensorflow, and also provide free GPU.

5.3. Implementation

This study follows experimental research methodology so that it is necessary to show and implement a model for the specified problem. This section shows the steps and procedures of the implementation of the proposed Faba bean leaf diseases detection and classification model.

5.3.1 Importing Library and Declare Constant Variables

Library is an external package that is created by a python developers and it's made widely available for our use. The main advantage of importing different libraries is that they are just kind of pre-written section of code and we can actually import them and then we have all of this pre-written code available to us once we import and we can just pretty much accesses anything with in this library and use it in our code and this simplifies things a lot. The following is the different library importing for this study.

```
[ ]  #Importing required libraries

     import numpy as np

[2]  import pickle

[3]  import pandas as pd

[4]  import cv2

[5]  import PIL

[6]  import keras

[7]  import tensorflow as tf
```

```
[ ]  # Importing libraries to access files and directories in the specified directory

     import os

[9]  from os import listdir
```

```
[ ]  #Importing libraries to visualize data

[10] import matplotlib.pyplot as plt

[11] from matplotlib.pyplot import imshow

[12] from keras.preprocessing.image import img_to_array
```

```
[ ]  #Importing libraries to create data spilting

[13] from sklearn .model_selection import train_test_split
```

```
from sklearn .model_selection import train_test_split

[ ] # Importing libraries to create sequential model and layers

[14] from tensorflow. keras import models, layers

[15] from keras.models import Sequential

[16] from tensorflow. keras. layers import BatchNormalization
     from keras. layers.convolutional import Conv2D
     from keras. layers.convolutional import MaxPooling2D

     from keras. layers.core import Activation
     from keras. layers.core import Flatten
     from keras. layers.core import Dropout
     from keras. layers.core import Dense

[17] from tensorflow.keras.optimizers import Adam

[18] from keras import backend as K

[ ] # Importing libraries to create confusion matrix

[19] from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

[20] import seaborn as sns
```

Figure 5-2: Importing essential library for the model.

Declare constant variable

The researcher had declare constant variables in order to refer constant letters. The following hyper parameters was used for the Convolutional neural network model.

```
[ ]  # create constant variables

[21] IMAGE_SIZE = 256
     BATCH_SIZE = 32
     CHANNELS = 3
     EPOCHS =10
     HEIGHT = 256
     WIDTH = 256
     INIT_LR = 1e-3
     n_classes = 4
```

Figure 5-3: Sample code of creating constant variable.

5.3.2 Assessing the directory

After importing the essential library and declaring constant variable, the researcher had generates the dataset. The dataset of Faba bean leaf images is found in Google drive in folder of Faba bean leaf images folders, inside these folder there is another four folder. By writing the python code dataset, which contains images in the folders, are filtered and categorized in a folder.

5.3.3 Defining the Classes

For classification and detection, it is necessary to know the class types, meaning, and differences. During data collection process the experts detect each disease type and explained about their symptoms, their effects on the Faba bean leaf and they categorized into individual folder. The total number of dataset is 4800 belongs to four classes.

Those images has three channels that is RGB with image size of 256*256. Every images in the dataset belongs to Bach size of 32. Those classes has index (0, 1, 2 and 3), these index represents four classes.

```
# visualize the image shape, RGB and batch size
```

```
for image_batch, label_batch in Dataset.take(1):
  print(image_batch.shape)
  print(label_batch.numpy())
```

```
(32, 256, 256, 3)
[0 1 2 1 2 0 3 1 2 3 2 2 3 2 1 0 3 0 1 2 0 3 2 0 3 3 0 3 3 3 1 3]
```

Figure 5-4: Sample code of visualize image channels, batch size and image shape.

5.3.4 Visualize Images

In the dataset, there is many images among those images, the researcher try to visualize three dimensional array of single and multiple images with their class names by using Matplotlip as plt.imshow().

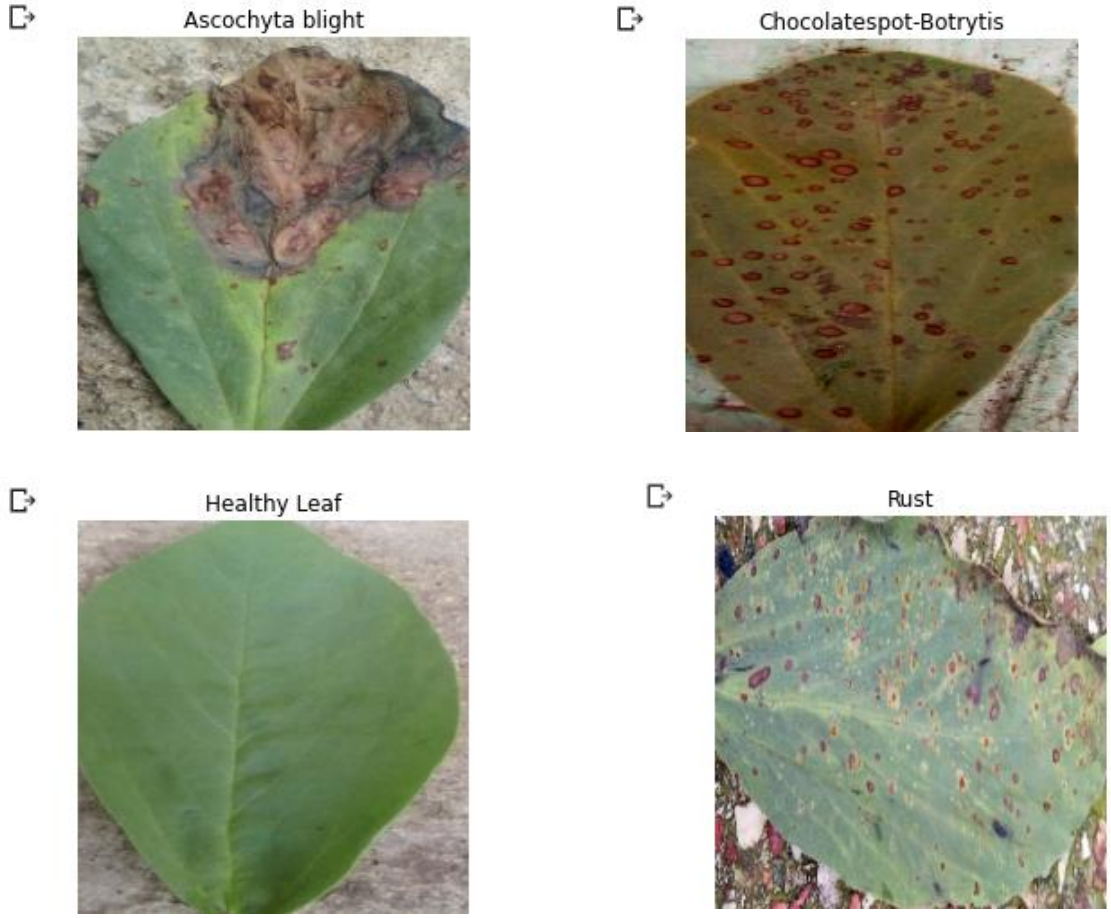


Figure 5-5: Sample output of Visualize a single Images with their corresponding names.

```
[ ]  # visualize multiple image of Faba beans
```

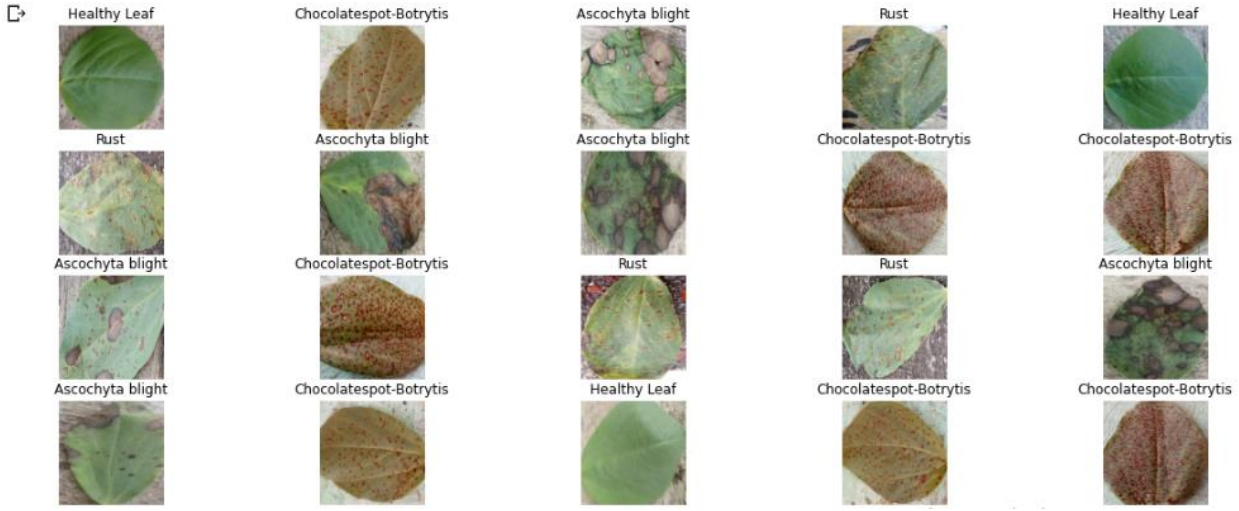


Figure 5-6: Sample output of Visualize multiple Images with their corresponding names.

5.3.5 Dataset Splitting

The dataset spilt into train set, validation set and test set. The length of dataset is 150*32. Because, each elements is 32 batch that means 150*32 equal to 4800. The study will keep eighty percent data as training data and the remaining twenty percent will do two split, one ten percent split will do validation and remaining ten percent will do testing. Training set is used during training the model. Validation set is used during the training process, when we run each epoch, after each epoch we will do validation on this set and testing set is used to measure the performance of the model. Before selecting these ratio that means 80%, 10% and10% training, validation and testing set respectively the researcher had test another two ratio 70%, 15% and 15% and 60%, 20% and 20% based on number epochs, accuracy and loss. Among those, 80%, 10% and 10% perform good accuracy and less loss on few epoch than the others.

```
[ ]  # length of dataset

[29] len(Dataset)

     150

[30] 150*32

     4800
```

Figure 5-7: Sample code of length of dataset.

The total dataset is split in training, validation and test set. The number of dataset was used for each set is shown in below by using the following code.

Number of training dataset

```
[33] len(train_ds)

     120
```

Figure 5-8: Sample code of length of train data.

The total number of training dataset used for train the model is 3,840. Because 120*32 equal to 3,840.

Number of validation dataset

```
[34] len(val_ds)

     15
```

Figure 5-9: Sample code of length of validation data.

The total number of validation dataset that used to validate the model is 480. Since, 15*32 equal to four hundred eighty.

Number of testing dataset

```
[35] len(test_ds)

     15
```

Figure 5-10: Sample code of length of testing data.

The number of testing dataset that is used to evaluate the performance of the model is 480. Since 15*32 is equal to four hundred eighty.

5.3.6 Preprocessing

In this study, during image preprocessing performs two tasks. Those are rescaling and resizing. Scaling is an essential part of preprocessing in image processing. So, the image the numpy array that we saw previously was between (0 and 255) is an RGB image scale. 1. 0 there by 255 this will just scale the images to 255. The second one is resizing. It will resize every images to 256 by 256. Resizing and rescaling are used in the model. One of the advantage of resizing is, when we have a training model and when it's starting predicting, during predicting if you are supplying any images which is not 256*256, this will take care of resizing.

5.3.7 Modeling

```
# Creating Model
```

```
[39] input_shape = (BATCH_SIZE, IMAGE_SIZE, CHANNELS, n_classes)
     model = Sequential()

     model.add(Conv2D(32, (3,3), padding = "same", input_shape=(256,256,3)))
     model.add(Activation("relu"))
     model.add(BatchNormalization())
     model.add(MaxPooling2D(pool_size=(3,3)))
     model.add(Dropout(0.25))

     model.add(Conv2D(64, (3,3), padding = "same", input_shape=(256,256,3)))
     model.add(Activation("relu"))
     model.add(BatchNormalization())
```

```
model.add(Conv2D(64, (3,3), padding = "same", input_shape=(256,256,3)))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3,3), padding = "same", input_shape=(256,256,3)))
model.add(Activation("relu"))
model.add(BatchNormalization())

model.add(Conv2D(128, (3,3), padding = "same", input_shape=(256,256,3)))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.25))

model.add(Conv2D(256, (3,3), padding = "same", input_shape=(256,256,3)))
model.add(Activation("relu"))
model.add(BatchNormalization())
```

Figure 5-11: Sample code of building model.

- The model type that had used is sequential, that's used to create a model layer by layer.
- Hyper parameters that are set before is used to determine the neural network structure and how it is trained.
- The Faba bean leafs diseases detection and classification model has seven layers. During building model we had use add () function to add a layers to the models:
- It has seven convolutional layers ((kernel size, (filters, and filters), input shape = (256, 256, 3). It used a kernel size of 3*3 filter matrix with padding same.
- Four max pooling layers that's used to decreases spatial size of the incoming features.
- Seven batch normalization layers, five dropout layers, and eight activation layers.
- This study had used flatten layer between convolutional and dense layer and
- It had used Softmax activation function to make a prediction or image classification.
- Print the model summary.

5.3.8 Compile the Model

In Deep learning, always the first task is define the neural network architecture. After define the architecture the next step is compiling the model by using optimizer, loss and matrix. The model compiled by using a famous optimizer called 'Adam'. Adam is good optimizer that is uses for many cases. It adjust the learning rate throughout training. The second parameter that is used during compiling is loss. The study had used Sparse Categorical Cross entropy. It is common for image classification task, and the last parameter is matrix. Accuracy is the matrix that is used to kind of track the training during the training process.

5.3.9 Train the Model

After building the model and compile the model the next step is training the model. Model. Fit () function is used to train the model by using different parameters. Such as batch-size, verbose, epoch and validation data used during each epoch. During training the model, the researcher had used different number of epochs like 5, 10, 15, 30 and 50 by using different ratio of data splitting. During training the model a number of epochs and a number of data splitting ratio can performed. After a number of trying, we get the best training and validation accuracy on data splitting ratio 80% for training, 10% validation and 10 % testing set and on epoch 15.

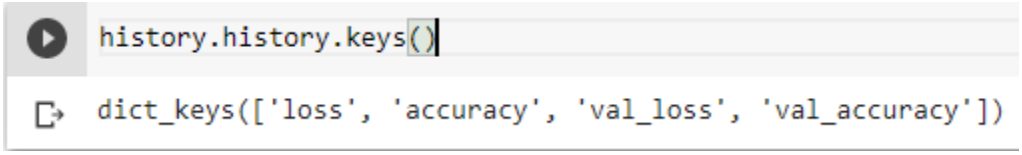
```
[ ]  history = model.fit(
          train_ds,
          epochs=EPOCHS,
          batch_size=BATCH_SIZE,
          verbose=1,
          validation_data=val_ds)
```

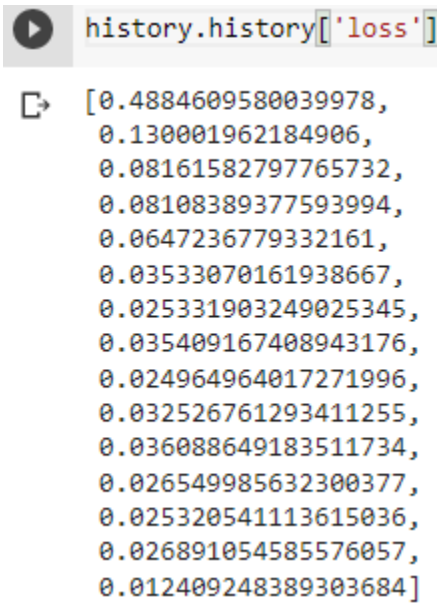
```
Epoch 1/15
120/120 [==============================] - 180s 778ms/step - loss: 0.4885 - accuracy: 0.8529 - val_loss: 1.9050 - val_accuracy: 0.5083
Epoch 2/15
120/120 [==============================] - 11s 90ms/step - loss: 0.1300 - accuracy: 0.9607 - val_loss: 0.1996 - val_accuracy: 0.9354
Epoch 3/15
120/120 [==============================] - 10s 87ms/step - loss: 0.0816 - accuracy: 0.9729 - val_loss: 1.8465 - val_accuracy: 0.6917
Epoch 4/15
120/120 [==============================] - 11s 87ms/step - loss: 0.0811 - accuracy: 0.9753 - val_loss: 0.5198 - val_accuracy: 0.8229
Epoch 5/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0647 - accuracy: 0.9831 - val_loss: 0.2545 - val_accuracy: 0.9312
Epoch 6/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0353 - accuracy: 0.9857 - val_loss: 0.1952 - val_accuracy: 0.9604
Epoch 7/15
120/120 [==============================] - 10s 87ms/step - loss: 0.0253 - accuracy: 0.9922 - val_loss: 1.2244 - val_accuracy: 0.7708
Epoch 8/15
120/120 [==============================] - 11s 88ms/step - loss: 0.0354 - accuracy: 0.9883 - val_loss: 0.6885 - val_accuracy: 0.8646
Epoch 9/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0250 - accuracy: 0.9927 - val_loss: 0.7331 - val_accuracy: 0.8917
Epoch 10/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0325 - accuracy: 0.9891 - val_loss: 1.2424 - val_accuracy: 0.8396
```

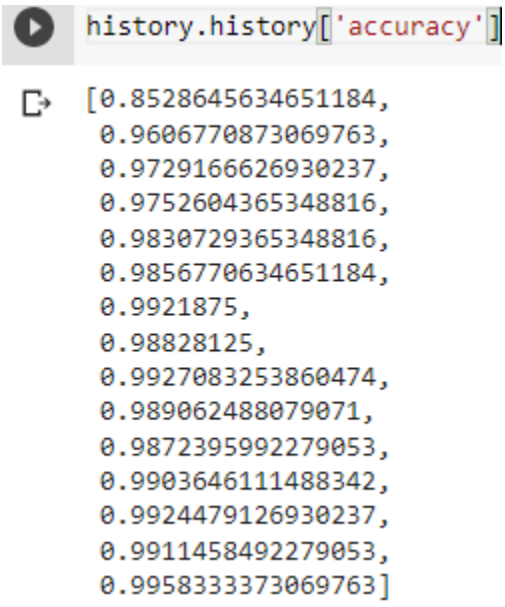
```
Epoch 11/15
120/120 [==============================] - 10s 87ms/step - loss: 0.0361 - accuracy: 0.9872 - val_loss: 2.2821 - val_accuracy: 0.6812
Epoch 12/15
120/120 [==============================] - 11s 88ms/step - loss: 0.0265 - accuracy: 0.9904 - val_loss: 0.0452 - val_accuracy: 0.9792
Epoch 13/15
120/120 [==============================] - 10s 87ms/step - loss: 0.0253 - accuracy: 0.9924 - val_loss: 0.1448 - val_accuracy: 0.9438
Epoch 14/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0269 - accuracy: 0.9911 - val_loss: 0.0799 - val_accuracy: 0.9750
Epoch 15/15
120/120 [==============================] - 10s 86ms/step - loss: 0.0124 - accuracy: 0.9958 - val_loss: 0.0137 - val_accuracy: 0.9937
```

Figure 5-12: Sample code of train the model.

History is keras.callbacks. The callbacks is automatically used to every keras model. The history objects gets returned by the fit method of the models. This history has different parameters and keys. This parameters and keys is shown below.







```
history.history['val_loss']
```

```
[1.905007243156433,
 0.1996029168367386,
 1.846498727798462,
 0.5197908282279968,
 0.25450894236564636,
 0.19523243606090546,
 1.224385380744934,
 0.6885157823562622,
 0.7330871224403381,
 1.2424018383026123,
 2.282053232192993,
 0.04516434296965599,
 0.14475947618484497,
 0.07989130914211273,
 0.013743468560278416]
```

```
history.history['val_accuracy']
```

```
[0.5083333253860474,
 0.9354166388511658,
 0.6916666626930237,
 0.8229166865348816,
 0.9312499761581421,
 0.9604166746139526,
 0.7708333134651184,
 0.8645833134651184,
 0.8916666507720947,
 0.8395833373069763,
 0.6812499761581421,
 0.9791666865348816,
 0.9437500238418579,
 0.9750000238418579,
 0.9937499761581421]
```

Figure 5-13: Sample code of parameters list of history of model. Fit.

5.3.10 Accuracy of the Model

To get a good accuracy for this study we had passed a number ways. During train a model a number of parameters was used. Those parameters has its own contribution to the model. Among them, epoch is the parameters used to train the model. As the number of epoch is increasing and decreasing the model become over fitting and under fitting. To make it the model normal we had used different number of epoch such as 5 to 50. As we had change the number of epochs, we had got a number of result based up on their result, we had choice a best result with their corresponding to data splitting ratio. The other things that play a great role during train a model is dataset splitting. We had used different ratio of data splitting such us 80-20%, 70-30% and 60-40% with different number of epoch. During training by using those parameters we had got different result such as we had got different number of accuracy and loss. Based up on those result we had select the epoch and data splitting ratio that perform good that means high accuracy with less number of loss. For this study we had select epoch at 15 and dataset splitting ratio at 80%, 10%, and 10%. We had summarized the result in the table below.

Data splitting ratio	Epochs	Test ratio	Accuracy	Loss
60-40	5	20%	0.6562	4.074
	10	20%	0.7906	1.0947
	15	20%	0.7354	1.6631
	30	20%	0.9833	0.0495
	50	20%	0.9899	0.0063
70-30	5	15%	0.7798	0.83
	10	15%	0.898	0.1225
	15	15%	0.9769	0.961
	30	15%	0.9402	0.2358
	50	15%	0.993	0.0148
80-20	5	10%	0.9812	0.0763
	10	10%	0.9937	0.0338
	15	10%	0.9958	0.0089
	30	10%	0.994	0.0063
	50	10%	1.0	3.22

Table 5-1: Result of an experiment by using different epochs, training, validation and testing dataset ratio.

As we can see in the below plot graph (fig 5-21), at the starting of the training the value of the training accuracy line becomes 80% and the values of the validation accuracy line is becomes 50% then becomes decreases to 40% then increases to 90% after that both training and validation accuracy increases very slowly. Also, when we can see the training and validation loss curves in

(fig 5-22) validation loss decreasing from five to two. Then, both training and validation loss decreases linearly from epoch four up to the end.

Lastly, we can see that the validation accuracy is in synchronized with the training accuracy and validation loss is in synchronized with training loss. Both curves are nearly linear. The curves are showing that there is no over fitting in the proposed model since us the value of validation accuracy increases the value of validation loss decreasing. There is no much gap between training accuracy and validation accuracy also between training and validation loss. This result is happen at epoch ten with data splitting ratio 80%, 10% and 10% with batch size 32 and with Adam optimizer.

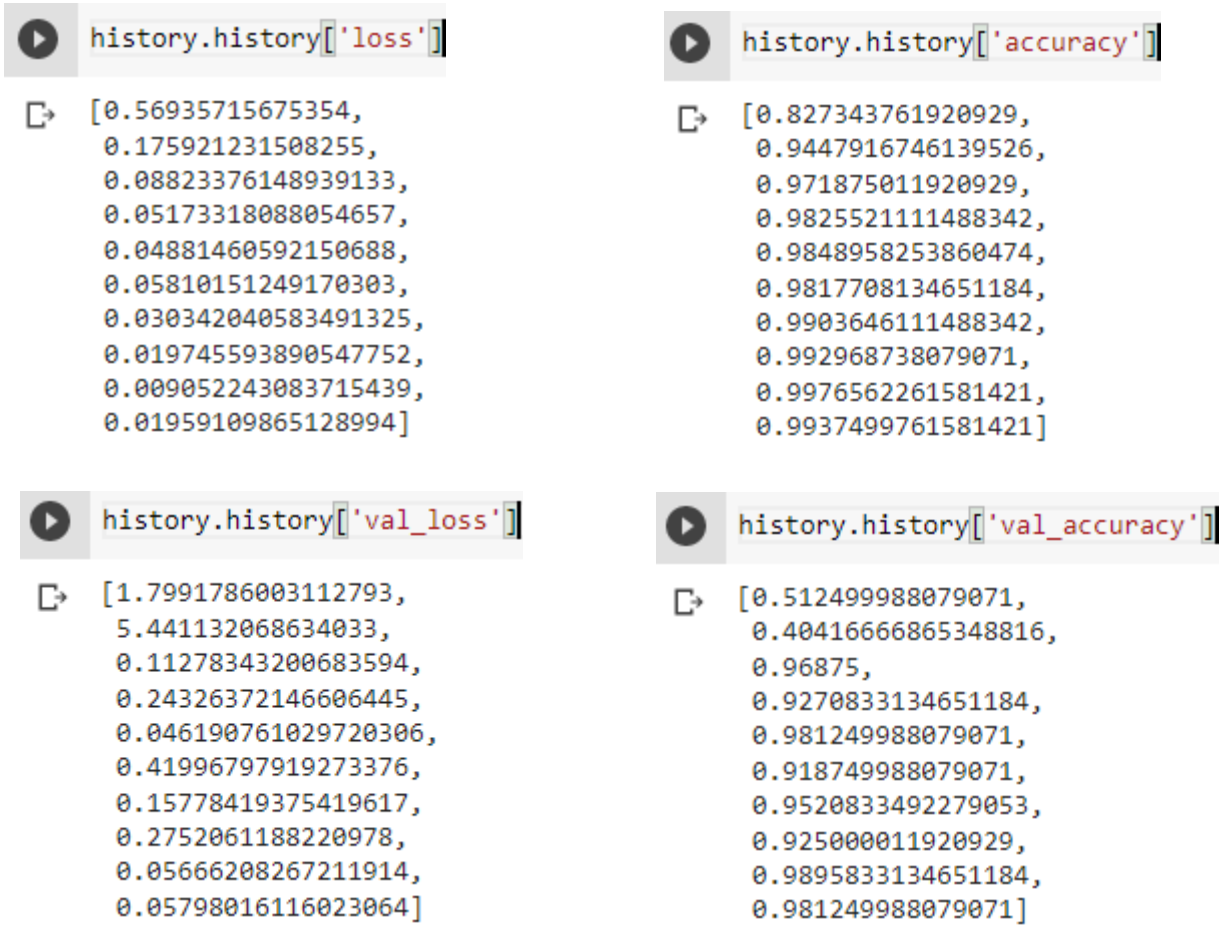


Figure 5-14: Training, validation accuracy and Training, validation loss.

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

```
[INFO] Calculating model accuracy
15/15 [==============================] - 0s 30ms/step - loss: 0.0338 - accuracy: 0.9937
Test Accuracy: 99.37499761581421
```

Figure 5-15: Sample code of calculating model accuracy at epoch ten.

Plotting the graph of training and validation accuracy

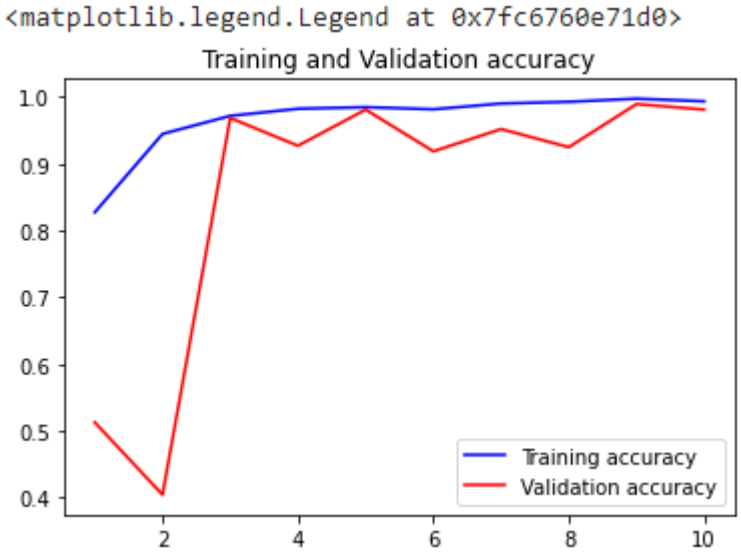


Figure 5-16: Training and validation accuracy of the Faba bean leaf diseases detection and classification model at epoch ten.

Plotting the graph of training and validation loss

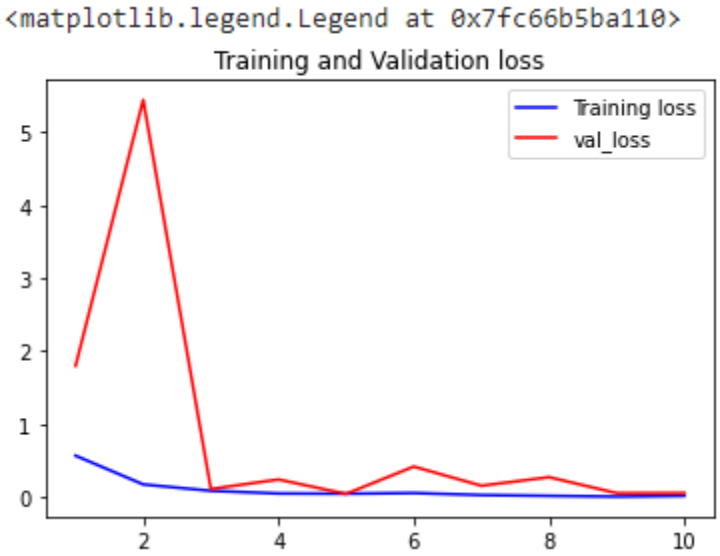


Figure 5-17: Training and validation loss of the Faba bean leaf diseases detection and classification model at epoch ten.

As the number of epoch increases the accuracy also increases. The result we get in epoch ten is modified in epoch 15, As we can see in the below plot graph (fig 5-23) at the starting of the training the value of the training accuracy line becomes 87% and the values of the validation accuracy line is becomes 30% then becomes increases to 90% after that both training and validation accuracy increases very slowly linearly. Also, when we can see the training and validation loss curves in (fig 5-24) the validation loss decreasing from 2.0 to 0.3, then both decreasing linearly from epoch 10 up to the end linearly.

Lastly, we can see that the validation accuracy is in synchronized with the training accuracy and validation loss is in synchronized with training loss. Both curves are nearly linear. The curves are showing that there is no over fitting in the proposed model since us the value of validation accuracy increases the value of validation loss decreasing. There is no much gap between training accuracy and validation accuracy also between training and validation loss. We get this result at epoch 15 with data splitting ratio 80%, 10% and 10% with batch size 32 and with Adam optimizer.

Plotting the graph of training and validation accuracy

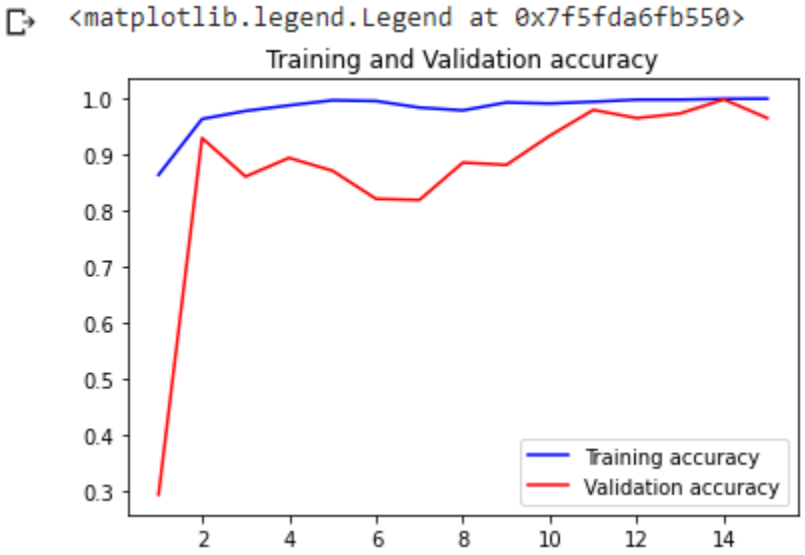


Figure 5-18: Training and validation accuracy of the Faba bean leaf diseases detection and classification model at epoch 15.

Plotting the graph of training and validation loss

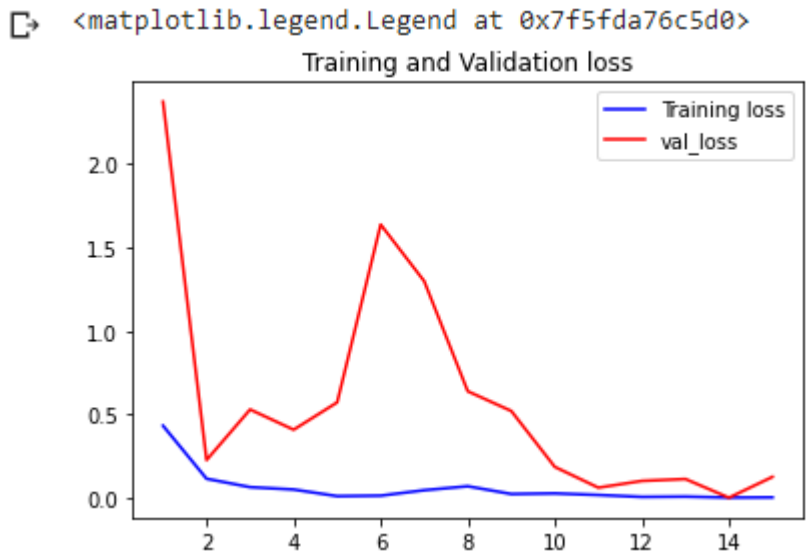


Figure 5-19: Training and validation loss of the Faba bean leaf diseases detection and classification model at epoch 15.

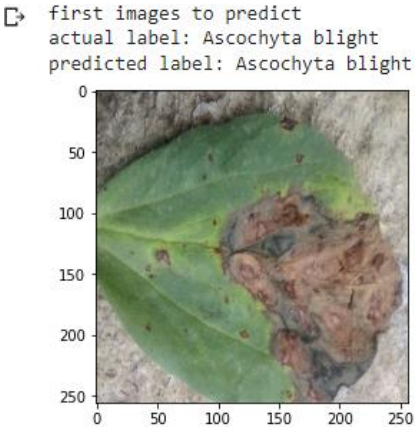
```
[ ]  print("[INFO] Calculating model accuracy")
     scores = model.evaluate(test_ds)
     print(f"Test Accuracy: {scores[1]*100}")
```

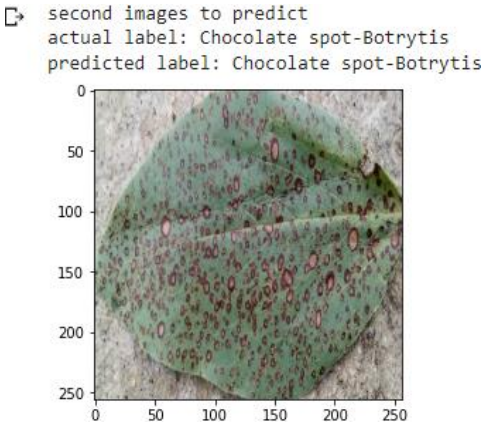
```
     [INFO] Calculating model accuracy
15/15 [==============================] - 88s 32ms/step - loss: 0.0089 - accuracy: 0.9958
```

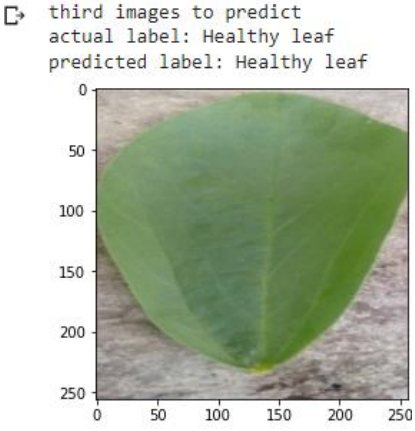
Figure 5-20: Sample code of calculating model accuracy.

5.3.11 Predicating

The model give the prediction based on the test data. After, train the model the next phase is predicating. In this study, the model first predict the size of image, and their index. Then display the actual and predicated images of a single and multiple image.







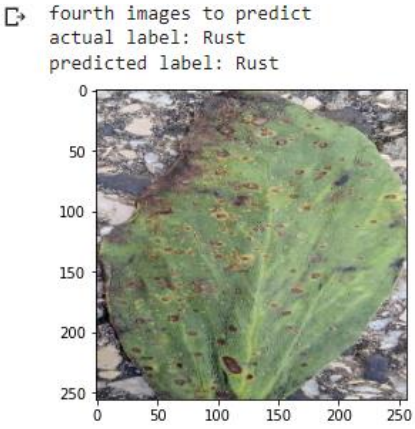


Figure 5-21: Sample output of predicting image size, index and displaying actual and predicted of single images.

```
[ ]  # model predict multiple class
```

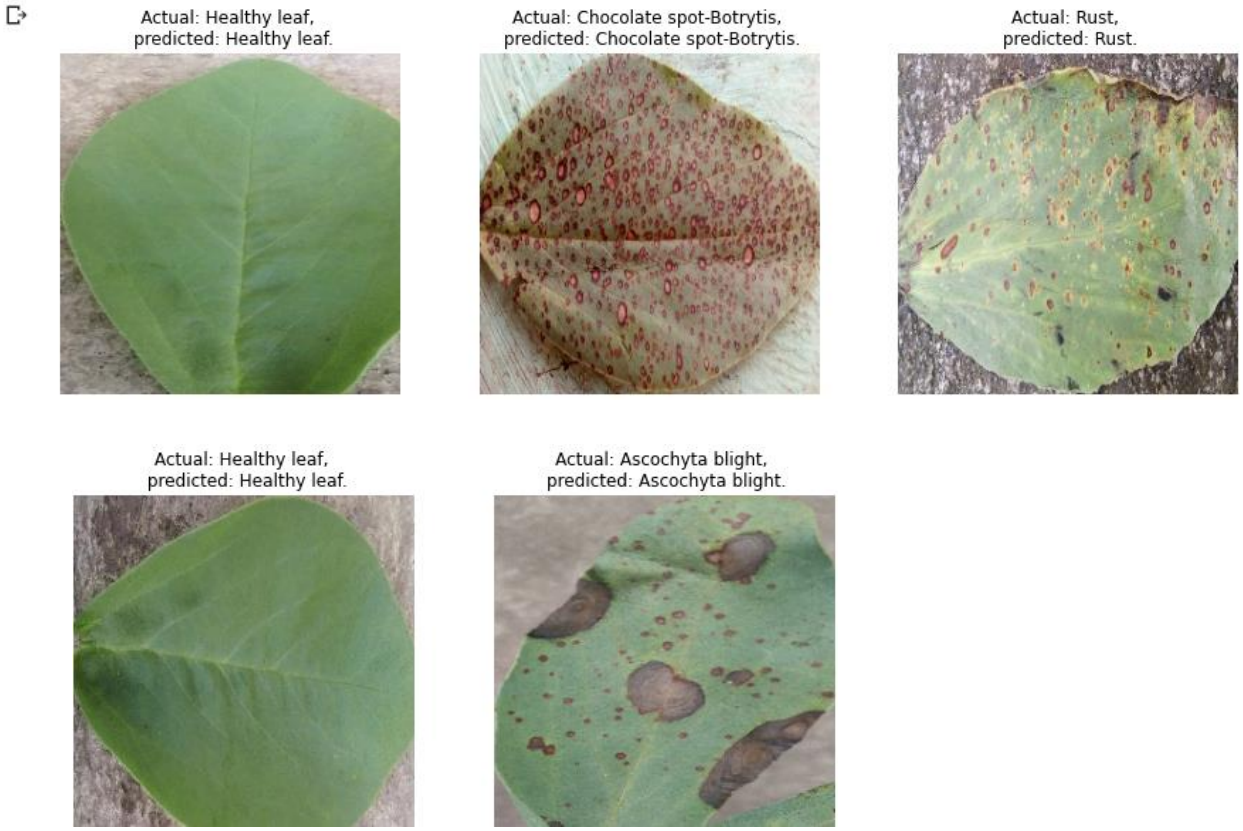


Figure 5-22: Sample output of predicating actual and predicted of multiple images.

The model was predicted the image size, index, the actual and predicted images of single and multiple images. Now, we had displace the actual class, predicted class and their confidence of the predicted images.

```
[ ]  # visualize the actual class, predicted class and confidence of the model
```

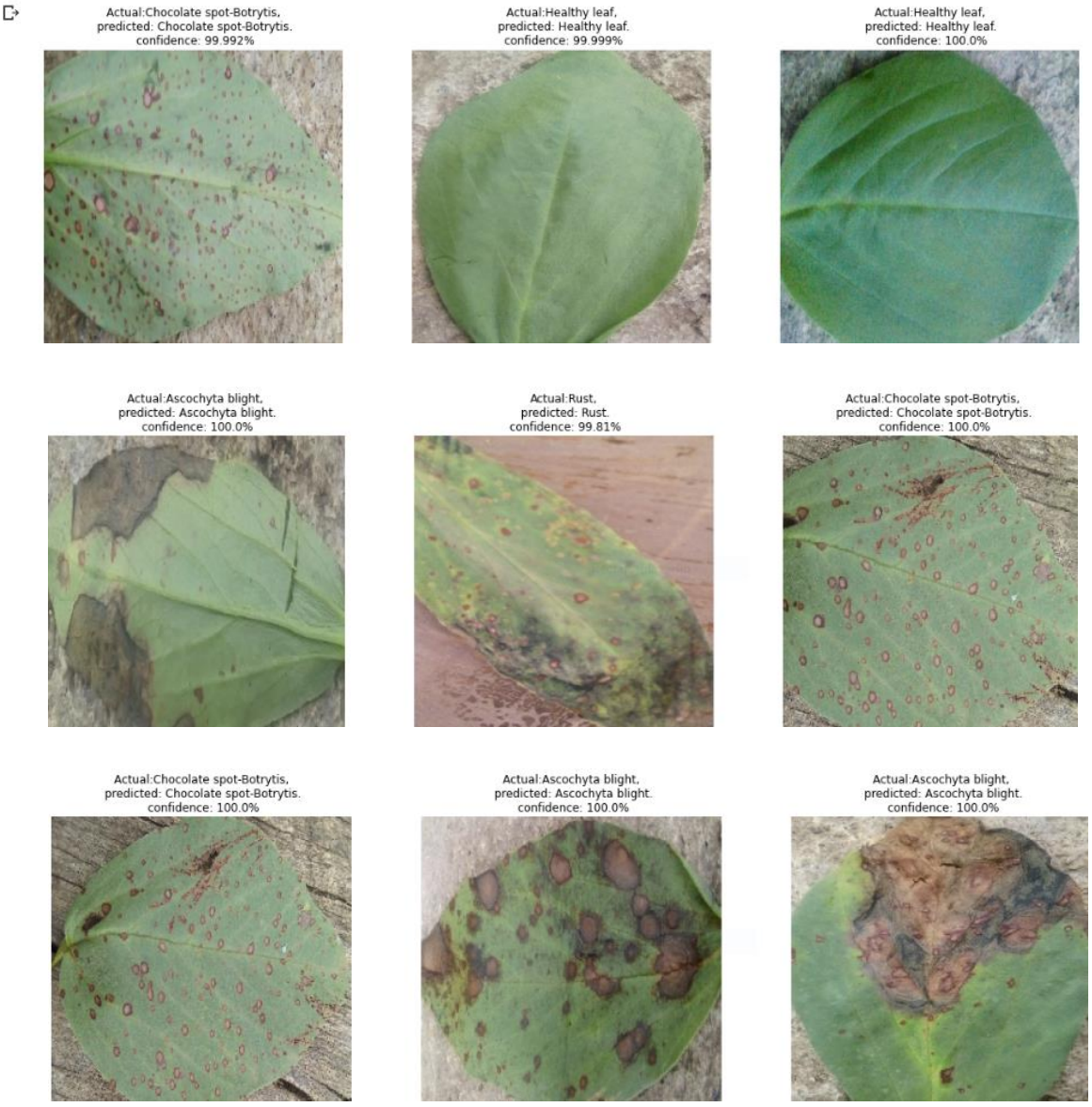


Figure 5-23: Sample output of predicting actual class, predicted class and their confidence.

5.4 Model Evaluation Result

After building, compiling, running and predicting a model is performed, performance of the Faba bean leaf diseases detection and classification model is evaluated using the following metrics.

5.4.1 Confusion matrix

Confusion matrix used in this study to lay out how many predicted classes were correctly predicted and how many were not. It evaluate the result of a predicted model with a class out come to see

number of classes that were correctly predicted as their true class. The classification report was shown in table below.

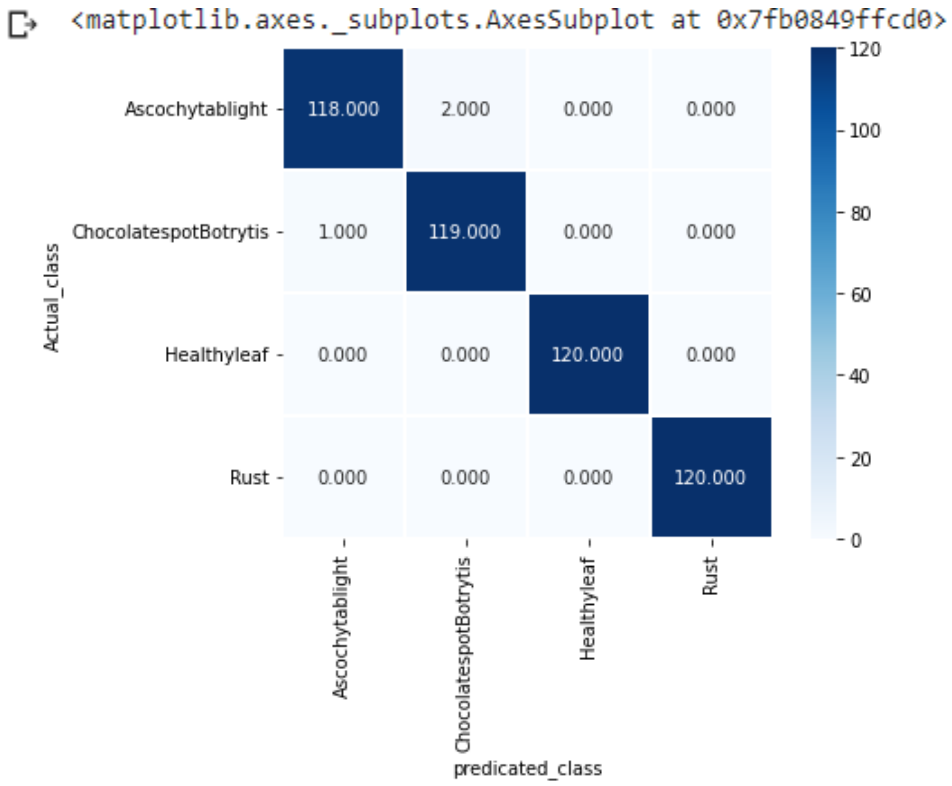


Figure 5-24: The result of Confusion Matrix.

The confusion matrix is calculated using testing data set which has total of 480 images. the model Truly predict of 118 of ascochyta blight out of 120, 119 true prediction of chocolate spot-Botrytis out of 120, 120 true prediction of health leaf out of 120 and 120 true prediction of rust out of 120. It has some misclassifying images on the test data which are three misclassifying images from the total testing data. Those are two images of ascochyta blight and one images of chocolate spot.

5.4.2 Precision

It's used to calculate the model ability to classify positive values correctly. True positive divided by the total true prediction which means true positive divided by true positive and false positive. When the model predicted true class, how often was it right. In this study, for ascochyta blight the precision is 0.9915 or 99.15% that means the total number of ascochyta blight correctly classified

as ascochyta blight but one chocolate spot images were classified as ascochyta blight and the result become 99.15%.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive+False positive}}$$

So that, the precision for ascochyta blight is calculated by using the above formula. True positive is equal to 118 and false positive is equal to one, $\frac{118}{118+1}$ = 0.991597. Similarly, for chocolate spot also calculated the precision as, True positive is equal to 119 and false positive is equal to two, $\frac{119}{119+2}$ = 0.983471. In the (table 5-2) the precision is showed below for all classes.

5.4.3 Recall

It's used to calculate the model ability to predict positive values. True positive divided by true positive and false negative. When the class was actually true how often the classifier got it right. Recall for Ascochyta blight is 0.983333 or 98.3333% which means 118 Ascochyta blight is correctly predicted and two Ascochyta blight images are wrongly classified to another classes.

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive+False negative}}$$

So that, the recall for ascochyta blight is calculated by using the above formula. True positive is equal to 118 and false negative is equal to two, $\frac{118}{118+2}$ = 0.983333. Similarly, for chocolate spot also calculated the precision as, True positive is equal to 119 and false negative is equal to one, $\frac{119}{119+1}$ = 0.991667. In the (table 5-2) the precision is showed below for all classes like this. In the (table 5-2) the recall is showed below.

5.4.4 F1 score

It is the harmonic mean between precision and recall. The score essentially used to how good the quality of the prediction are and how completely we have predicated the labels from the dataset.

$$\text{F1 score} = \frac{2*(\text{precision}*\text{recall})}{\text{precision+recall}}$$

F1 score for ascochyta blight is 0.987448 which mean precision equal to 0.991597 and recall equal to 0.983333 then F1 score equal to $\frac{2*(0.991597\ *0.983333\ )}{0.991597\ *0.983333}$ = 0.987448. F1 score for the left classes also done like this and the result is for classes shown in the (table 5-2).

5.4.5 Accuracy

Accuracy is used to find the portion of correctly classified values. It tells us how often our classifier is right. It's the number of true positive and true negative over the entire prediction. In the (table 5-2) the accuracy of the model is showed below.

$$\text{Accuracy} = \frac{\text{True positive+True negative}}{\text{Total}}$$

	precision	recall	f1-score	support
Ascochytablight	0.991597	0.983333	0.987448	120.00000
ChocolatespotBotrytis	0.983471	0.991667	0.987552	120.00000
Healthyleaf	1.000000	1.000000	1.000000	120.00000
Rust	1.000000	1.000000	1.000000	120.00000
accuracy	0.993750	0.993750	0.993750	0.99375
macro avg	0.993767	0.993750	0.993750	480.00000
weighted avg	0.993767	0.993750	0.993750	480.00000

Table 5-2: Performance evaluation using precision, recall, F1-score, and accuracy.

CHAPTER SIX

6. Conclusion, Recommendation, Future work, and Contribution

This chapter presents the conclusion derived from the conduct of this study, recommendations, and feature works are suggested for those who are studying or have a plan to work in this research area.

6.1 Conclusion

Currently the Faba bean production is suffered by a number of problems, different diseases like ascochyta blight, chocolate spot, and rust which decreases the yields sometimes 50-100% on the unimproved local cultivar and quality of Faba bean production. Therefore, the detection of Faba bean diseases is very essential to overcome the above problem.

To detect and classify the Faba bean leaf diseases we propose and implement a deep learning approach by using Convolutional neural network and image processing algorithm and using Faba bean leaf images. We have presented a model to detect and classify various Faba bean leaf diseased such as ascochyta blight, chocolate spot, rust, and healthy using leaf image as an input. The Faba bean leaf diseases detection and classification model can be used as a tool to detect and classify the Faba bean leaf diseases. Model was performed by collecting the Faba bean leaf images both diseased and healthy, perform image preprocessing, perform data splitting, building model, extracted important feature, classification of diseases was performed, finally predicting and evaluating the performance of model were performed.

In this study Convolutional neural network algorithm is used which is the most popular Deep learning algorithm. It was used to extract feature automatically without human interference. The algorithm was designed in a way that was appropriate for the detection and classification of Faba bean leaf diseases. The network was created by using four input which means three diseases classes and one healthy class and has seven layers. This study was used three dataset types and three option of data splitting ratio to conduct the experiment for the Faba bean leaf diseases detection and classification model. Data set type was RGB which means three channels images was used. The

Faba bean leaf diseases detection and classification model has achieved an accuracy of 99.58 % with 15 epochs, 80%, 10% and 10% dataset splitting ratio, batch size 32 and with RGB images.

Finally, one of the most important factors that have impacts on the efficiency and accuracy of the model is a number of training epochs, as a number of epochs increases and decreases the model become over or under fit. Also, the dataset splitting ratio has its own effect. This study was used different data splitting ratio and several number of epochs like 5-50. But it has achieved higher accuracy with 15 epochs on 80%, 10%, and 10% dataset splitting ratio for the Faba bean leaf diseases detection and classification model.

6.2 Recommendation

As we know agriculture is the base of human being life. Since one of the basic need of human being is food. This food is comes from the production of farming system. In our country this industry is used by most of the people. However, this industry is not sufficient in terms of production and product due to different factors such us, most of the farmers used traditional way of farming system, lack of knowledge on the way of farming and on the snow, identifying the various types of diseases that attack the crop, and lack of guidance by the experts. So, it has recommend different researchers and experts to grow this industry by contributing the following. Educating the farmers and makes the agricultural industry modernize, the professionals and experts guides the farmers to make the industry effective, and now a day technology is expand in all sectors, so I will recommend different researchers to do different application that easily detect and identify different kinds of diseases and that simplifies the work of farmers and experts.

6.3 Future work

From the experience gained during this study and from the observed gap during reviewing different related works, for those who are working in this area, for those who have the interest to work on this area for the future and as this study progresses in the future, the work to be added in the future is suggested as below.

- In this study, one healthy and three diseased leaf of Faba bean is detected and classified, but there is another leaf diseases that affect the plant. Detect and classify the other leaf diseases of Faba bean must be a feature work for a researchers.

- Detecting the diseases has further modified to other parts of Faba bean like seed and flower.
- Faba bean is a plus crop which is used daily in most of the people in this country and this plus crop yields is exported to foreign countries. So that develop a system for Faba bean grading system will be included for the future.

6.4 Contribution

After reviewing different related literature and identifying the gaps, the study contribute the following works.

- Prepare a dataset for Faba bean leaf diseases detection and classification model.
- By discussing with plant experts identify the most parts of Faba bean that affected by the diseases.
- Building a model that efficiently classified Faba bean leaf diseases by comparing with other model.
- Achieved a good accuracy for leaf diseases detection and classification of Faba bean.
- Extract essential feature by using Convolutional neural network layer which is convolution layer.

REFERENCES

[1] E. H. Gezahegne Getaneh and T. S. Negussie Tadesse, “Faba Bean Gall; a New Threat for Faba Bean (Vicia faba) Production in Ethiopia,” *Adv. Crop Sci. Technol.*, vol. 02, no. 04, 2014, doi: 10.4172/2329-8863.1000144.

[2] A. Abraham *et al.*, “Survey of faba bean (Vicia faba L.) virus diseases in Ethiopia,” *Phytopathol. Mediterr.*, vol. 39, no. 2, pp. 277–282, 2000, doi: 10.1400/57859.

[3] E. Fekadu, K. Kibret, A. Melese, and B. Bedadi, “Yield of faba bean (Vicia faba L.) as affected by lime, mineral P, farmyard manure, compost and rhizobium in acid soil of Lay Gayint District, northwestern highlands of Ethiopia,” *Agric. Food Secur.*, vol. 7, no. 1, pp. 1–11, 2018, doi: 10.1186/s40066-018-0168-2.

[4] M. Wolde, “Evaluation of Faba Bean (Vacia faba L.) Varieties for Chocolate Spot (Botrytis fabae L.) Disease Resistance at Bale Zone, Southeastern Ethiopia,” *Agric. Res. Technol. Open Access J.*, vol. 18, no. 4, 2018, doi: 10.19080/artoaj.2018.18.556067.

[5] B. Merga, M. C. Egigu, and M. Wakgari, “Reconsidering the economic and nutritional importance of faba bean in Ethiopian context,” *Cogent Food Agric.*, vol. 5, no. 1, p. 1683938, 2019, doi: 10.1080/23311932.2019.1683938.

[6] T. Degu, W. Yaregal, and T. Gudisa, “Status of Common Bean (Phaseolus vulgaris L.) Diseases in Metekel Zone, North West Ethiopia,” *J. Plant Pathol. Microbiol.*, vol. 11, no. 5, pp. 1–8, 2020, doi: 10.35248/2157-7471.20.11.494.

[7] S. Deneke, “Review on Epidemiology and Management of Faba Bean (Vicia fabae) Chocolate Spot (Botrytis fabae), Root Rot (Fusarium solani) and Rust (Uromyces vicia fabae) in Ethiopia,” *Int. J. Sci. Res. Publ.*, vol. 8, no. 5, pp. 105–111, 2018, doi: 10.29322/ijsrp.8.5.2018.p7717.

[8] S. Sahile, C. Fininsa, P. K. Sakhuja, and S. Ahmed, “Yield loss of faba bean (Vicia faba) due to chocolate spot (Botrytis fabae) in sole and mixed cropping systems in Ethiopia,” *Arch. Phytopathol. Plant Prot.*, vol. 43, no. 12, pp. 1144–1159, 2010, doi: 10.1080/03235400802343791.

[9] P. Devaraj, M. P. Arakeri, and B. P. V. Kumar, “Early detection of leaf diseases in Beans crop using Image Processing and Mobile Computing techniques,” vol. 10, no. 10, pp. 2927–2945, 2017.

[10] E. Journal and T. Ejt, “Vol 7, Number 2, 2017 European Journal of Technique EJT,” vol. 7, no. 2, pp. 207–218, 2017.

[11] A. Bapat, S. Sabut, and K. Vizhi, “Plant leaf disease detection using deep learning,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6, pp. 3599–3605, 2020.

[12] G. Yigezu Wendimu, “The challenges and prospects of Ethiopian agriculture,” *Cogent Food Agric.*, vol. 7, no. 1, 2021, doi: 10.1080/23311932.2021.1923619.

[13] R. U. Khan, K. Khan, W. Albattah, and A. M. Qamar, “Image-Based Detection of Plant Diseases: From Classical Machine Learning to Deep Learning Journey,” *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/5541859.

[14] Jiangsheng Gui, “Identification of Soybean Leaf Spot Diseases using Deep Convolutional Neural Networks,” *Int. J. Eng. Res.*, vol. V8, no. 10, pp. 289–294, 2019, doi: 10.17577/ijertv8is100130.

[15] V. Sardana, P. Sharma, and P. Sheoran, “GROWTH AND PRODUCTION OF PULSES.”

[16] A. Karkanis *et al.*, “Faba Bean Cultivation – Revealing Novel Managing Practices for More Sustainable and Competitive European Cropping Systems,” vol. 9, no. August, pp. 1–14, 2018, doi: 10.3389/fpls.2018.01115.

[17] M. Mitiku, “Integrated Management of Chocolate Spot (Botrytis Fabae) Disease of Faba Bean (Vicia Faba L.) in Ethiopia: a Review,” *Int. J. Res. -GRANTHAALAYAH*, vol. 5, no. 9, pp. 195–205, 2017, doi: 10.29121/granthaalayah.v5.i9.2017.2231.

[18] Y. Degago, “FABA BEAN (Vicia faba) IN ETHIOPIA,” p. 43, 2000.

[19] GRDC, “Faba bean: Plant growth and physiology,” *Grains Res. Dev. Corp. Grownotes-West.*, no. November, p. 19, 2017.

[20] B. K. Meresa and H. M. Gebremedhin, “Faba Bean Gall (Olpidium viciae K .) as a Priority

Biosecurity Threat for Producing Faba Bean in Ethiopia : Current Status and," vol. 2020, 2020.

[21] F. Disease *et al.*, "Faba bean," no. November, 2017.

[22] A. Anteneh, E. Yohannes, G. Mesganaw, G. Solomon, and T. Getachew, "Survey of faba bean (Vicia faba L.) diseases in major faba bean growing districts of North Gondar," *African J. Plant Sci.*, vol. 12, no. 2, pp. 32–36, 2018, doi: 10.5897/ajps2016.1615.

[23] P. A. Nazarov, D. N. Baleev, M. I. Ivanova, L. M. Sokolova, and M. V. Karakozova, "Infectious Plant Diseases: Etiology, Current Status, Problems and Prospects in Plant Protection," *Acta Naturae*, vol. 12, no. 3, pp. 46–59, 2020, doi: 10.32607/actanaturae.11026.

[24] G. B. Lucas, C. L. Campbell, and L. T. Lucas, "Causes of Plant Diseases," *Introd. to Plant Dis.*, pp. 9–14, 1992, doi: 10.1007/978-1-4615-7294-7_2.

[25] L. Zhu, P. Spachos, E. Pensini, and K. N. Plataniotis, "Deep learning and machine vision for food processing: A survey," *Curr. Res. Food Sci.*, vol. 4, no. March, pp. 233–249, 2021, doi: 10.1016/j.crfs.2021.03.009.

[26] H. J. He, C. Zheng, and D. W. Sun, "Image Segmentation Techniques," *Comput. Vis. Technol. Food Qual. Eval. Second Ed.*, no. December, pp. 45–63, 2016, doi: 10.1016/B978-0-12-802232-0.00002-5.

[27] T. Balaji, "An Automatic Color Feature Vector Classification Based on Clustering Method," vol. IV, no. Iv, pp. 13–18, 2017.

[28] R. Bala, "Survey on Texture Feature Extraction Methods," *Int. J. Eng. Sci. Comput.*, vol. 7, no. 4, pp. 10375–10377, 2017.

[29] G. Lotufo and K. Gustavson, "Introduction to Image Sensor," *Training*, pp. 1–63, 2009.

[30] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[31] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, no.

7, pp. 58–65, 2021, doi: 10.1145/3448250.

[32] T. Bezdan and N. Bačanin Džakula, “Convolutional Neural Network Layers and Architectures,” no. January, pp. 445–451, 2019, doi: 10.15308/sinteza-2019-445-451.

[33] L. M. Shita, “ETHIOPIAN COFFEE LEAF DISEASES IDENTIFICATION USING DEEP LEARNING,” vol. 8, no. 10, pp. 498–508, 2021.

[34] M. Akila and P. Deepan, “Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm,” *Iarjset*, vol. 6, no. 7, pp. 72–75, 2018.

[35] A. M., M. Zekiwos, and A. Bruck, “Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis,” *J. Electr. Comput. Eng.*, vol. 2021, pp. 1–10, 2021, doi: 10.1155/2021/9981437.

[36] D. Krisnandi *et al.*, “Diseases Classification for Tea Plant Using Concatenated Convolution Neural Network,” *CommIT (Communication Inf. Technol. J.*, vol. 13, no. 2, pp. 67–77, 2019, doi: 10.21512/commit.v13i2.5886.

[37] J. Amara, B. Bouaziz, and A. Algergawy, “A deep learning-based approach for banana leaf diseases classification,” *Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform.*, vol. 266, pp. 79–88, 2017.

[38] M. Syarief and W. Setiawan, “Convolutional neural network for maize leaf disease image classification,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1376–1381, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14840.

[39] S. Arivazhagan and S. V. Ligi, “Mango Leaf Diseases Identification Using Convolutional Neural Network,” *Int. J. Pure Appl. Math.*, vol. 120, no. 6, pp. 11067–11079, 2018.

[40] H. ÇETİNER, “Classification of Apple Leaf Diseases Using the Convolution Neural Network Approach,” *Mühendislik Bilim. ve Tasarım Derg.*, vol. 9, no. 4, pp. 1130–1140, 2021, doi: 10.21923/jesd.980629.