Detection and Classification of Sunflower leaf Disease Using Convolutional Neural Network

(CNN)

By: Daniel Solomon

A Thesis Submitted to

The Department of Computer Science and Engineering,

School of Electrical and Computing engineering

Presented in partial fulfillment of Master’s degree in Computer Science and Engineering

Office of graduate studies

Adama Science and Technology University

August 2021

Adama Ethiopia

Detection and Classification of Sunflower leaf Disease Using Convolutional Neural Network

(CNN)

By: Daniel Solomon

Name of Advisor: Worku Jifara (Ph.D.)

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in partial fulfillment of Master's degree in Computer Science and Engineering

Office of graduate studies

Adama Science and Technology University

August 2021

Adama Ethiopia

APPROVAL SHEET

The advisor of the thesis entitled" **Detection and Classification of Sunflower leaf disease using Convolutional Neural Network (CNN)" and developed by Daniel Solomon Mola.** Hear by certifying that the recommendation and Suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis

Worku Jifara (Ph.D.)	________________	________________
Advisor	Signature	Date

We, the undersigned, members of the board of Examiners of the thesis by **Daniel Solomon Mola** Have read and evaluated the thesis entitled "**Detection and Classification of Sunflower leaf disease using Convolutional neural network (CNN)"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Masters of Science in Computer Science and Engineering.

________________	________________	________________
Chairperson	Signature	Date
________________	________________	________________
Internal Examiner	Signature	Date
Almebante mulu kumlign (PH.D.)	________________	________________
External Examiner	Signature	Date

Final approval and acceptance of the thesis are contingent upon submission of its Final copy to the office of Postgraduate Studies (**OPGS**) through the Department Graduate Council (**DGC**) and School Graduate Committee (**SGC**)

Biruk Yirga Nidaw (PH>D.)	________________	________________
Department Head	Signature	Date
Dereje Tekilu (PH.D.)	________________	________________
School Dean	Signature	Date
Legesse Lemeche Obsu (PH.D.)	________________	________________
Office of postgraduate studies. Dean	Signature	Date

DECLARATION

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been properly acknowledged.

Daniel Solomon	__________________	__________________
Name Student	Signature	Date

ACKNOWLEDGMENT

"The fear of the LORD is the beginning of Wisdom" so that I give my praise to the Almighty God, who made everything possible. Next, I would like to express my appreciation and great thanks to **Dr. Worku Jifara**, for his vital effort in advising, checking, and providing knowledgeable, constructive, and valuable comments by quality his professional experience Finally, I love you so much, my mother Thank you for being your child. May God Grant you Long life and Health. And I thank you, my dear wife, Tigist Brille, for giving me as much love and money as my mother, who never stopped helping her financially and mentally from the beginning to the end. And thank you, my brothers, Abebe and Yemane, for the enduring love you have given me. I would like to express my genuine thank to my friends Muluken Yohannes, Dr. Ziyn, Tasew Tekle for their supports.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

CPU	Central Processing Unit
DL	Deep Learning
FC	Fully Connected
GPU	Graphical Processing Unit
ML	Machine Learning
RAM	Random Access Memory
SVM	Support Vector Machine
YOLO	You Only Look Once
ANN	Artificial neural network
Adam	Adaptive learning rate optimization algorithm
CPU	Central processing unit
Conv	Convolution
CL	Convolution layer
ConvNets	Convolutional neural networks
FN	False negatives
FCL	Fully connected layer
GHz	Giga hertz
GB	Gigabytes
GPU	Graphical processing unit
LCD	Liquid-Crystal Display
PL	Pooling layer
FP	False positives
PPV	Positive predictive value
CNN	Convolutional Neural Network
P	Precision

ABSTRACT

Sunflower is a seed crop for essential oils that can be employed in agricultural and oil production systems. Sunflower crops are harmed by a variety of diseases, insects, and nematodes, resulting in a wide range of productivity losses. Disease detection can be done with the naked eye; however, this procedure is impractical on large farms. To address this problem, several scholars used a variety of methods and techniques to conduct research. Previous research has used particle swarm optimization, digital image processing, K-means ++ clustering, and other methods to solve this problem. Ethiopian researchers conduct numerous plant disease detection and classification investigations. Even with Ethiopia's large production, however, the detection and classification of sunflower diseases receive little attention.

The main objective is to develop a system for detecting and classifying sunflower leaf disease using deep learning technologies. *To achieve the main objective Convolution neural network methods are used in this research. CNN is used in image classification because of its accuracy, while RNN is commonly utilized with sequences such as text, sound, video, and more. The SunfNet model was created in this study utilizing these strategies. The constructed model has ten layers: nine convolution layers and one output layer. This model is developed using a 7850 (seven thousand eight hundred fifty) training data set augmented from original images. Back-propagation and gradient descent optimization algorithms are used to develop this model. The confusion matrix is used to assess model performance. SunfNet model is tested in different epochs and learning rates to generate an efficient model. Our result of the experiment is 99.88% training accuracy with 99% validation accuracy. This system's accuracy is about ~99.8.*

Keywords*: Convolutional Neural Network (CNN), Sunflower Leaf Disease, Classification.*

1 CHAPTER ONE: INTRODUCTION

1.1 Background of the study

Ethiopia has been one of Africa's and the world's developing countries in the last decade, and its economy is based on agriculture and industry, with agriculture serving as the backbone of the country. More than 85 percent of the population relies on agricultural products, including cattle, for a living, and agriculture products account for 45 percent of the country's Gross Domestic Product (GDP). Sunflower [5], a member of the Asteraceae family, is native to temperate North America, which is where these vital edible oil-producing species thrive. Governments in both wealthy and developing countries see smart agriculture as a way to improve food security and the economy. [7] The goal of this study is to identify and classify sunflower diseases.

After soybean, oil palm, and canola, it ranks fourth among vegetable oil seeds in the globe. Sunflower is a temperate zone crop that is currently cultivated on roughly 23 million hectares in 40 countries throughout the world, including some areas in humid tropical Africa, because of its rustic nature and ability to function well under a variety of climatic and soil conditions [5]. Sunflower oil is premium oil with a light hue that is extensively utilized in the diets of heart patients because it includes very low cholesterol and a high concentration (90%) of unsaturated fatty acids.

Sunflower is one of the most important crops for oil production, with an average of 25 million hectares of sown lands around the world, following soy, cotton, and rapeseed. Ethiopia is a country in Africa. Fungi, viruses, bacteria, and nematodes are all known to cause disease in sunflowers. Fungi are responsible for the majority of sunflower leaf diseases. If these diseases are not recognized and monitored early enough, they might hurt productivity. Smart agriculture is important for lowering agricultural production costs in terms of productivity, environmental effect, food security, and long-term sustainability. [9] One of the most important activities in agriculture is the detection of plant diseases.

In recent years, different technologies have been used to detect disease. To classify plant health state nowadays, an image processing technique is often utilized. [11]. Deep learning is a new, cutting-edge method for picture processing and data analysis. Deep learning has been used successfully in a variety of fields. The first is an agricultural domain [10]. Deep learning is a type

Of machine learning that adds more complexity to the model and transforms the data using different functions that allow data to be represented hierarchically. [12], [13] are two examples.

Deep learning has the advantage of automatically extracting features from raw data, with features from higher levels of the hierarchy created by the composition of lower-level features [12]. Because of the more complicated models used, deep learning is mostly used to tackle more complex issues well and quickly [14]. Disease detection might be possible through the naked eyes but the problem with this method is that when one must survey sizable sunflower farms it is very hard to detect and monitor. To overcome this difficulty, researchers used sunflower leaf photos to construct a detection and classification system for the disease. The goal of the study is to accurately classify sunflower leaf photos using CNN and hybrid algorithms.

1.2 The Motivation of the study

Sunflower is one of the most incredible plants that is cultivated in a short amount of time but its productivity can be affected easily by numerous diseases, tackling this problem using deep learning and other accurate algorithms can prevent this damage which is a massive gain for both farmers and industries.[14] We are trying to observe and analyze especially on Sunflower seeds that are rich in the B complex vitamins, which are essential for a healthy nervous system, and are a good source of phosphorus, magnesium, iron, calcium, potassium, protein, and vitamin E.

They also contain trace minerals, zinc, manganese, copper, chromium, and carotene as well as monounsaturated and polyunsaturated fatty acids - types of 'good' fat that may help to protect the arteries. Agriculture is the backbone of Ethiopia's economy, accounting for 41.4 percent of GDP, 83.9 percent of total exports, and 80 percent of all employment [16]. In comparison, Ethiopia's important agricultural industry has risen at a rate of roughly 10% per year over the last decade, far faster than the country's population growth. The service and industrial sectors are also important, accounting for 43 percent and 15.6 percent, respectively [17]. It is possible to detect sunflower leaf diseases by the naked eye, but the problem arises when one person must examine the plants when there is a large volume of space to cover, which takes a long time and requires low precision, resulting in low productivity.

1.3 Statement of the problem

Agriculture is the leading sector of the economy in Ethiopia, [16] the land is cultivated by old production techniques, which lead to different structural problems. In agriculture disease, detection and classification is a major challenge. Detection and classification of agricultural illness is a big concern. As a result, appropriate measures must be made to manage illnesses in agricultural goods while lowering the usage of chemicals to do so. The procedure for diagnosing various plant diseases relies on specialist knowledge. Picture processing these days has turned into the vital method for the conclusion of different components of the plant in the space of horticulture since it limits disarray and aiding the master in staying away from the maltreatments during analysis and control of plant sicknesses. Plant infections have caused huge financial misfortunes in every last one of the nations.

In the sunflower plant so many types of diseases. Many of the researchers are conducting their research around these areas. However, the accuracy of the method and algorithm they used is not Efficient. The gap that is trying to identify the plant disease incorrectly will lead to huge loss of both quantity and quality of the product and it also incurs in time and money. Hence, identifying the condition of the plant plays a major role in successful cultivation. Now a day's image processing technique is being employed as a focal technique for diagnosing the various features of the crop. The image processing techniques can be used for the identification of the plant disease and hence classify the plant disease. Generally, the symptoms of the disease are observed on leaves, stems, flowers, etc. Here, the plants of the affected plant are used for the detection and classification of the disease. In this projected research image processing and the different algorithm will be used to solve the sunflower disease recognition and arrangement problem. The experiment is shown on a collected data set to get an exact model. This research will attempt to solve these problems by answering the following.

1.4 Research questions

1. Is CNN able to correctly classify the disease using the extracted features?
2. How enhanced methods and algorithms are to be implemented to classify and detect the disease properly?
3. What is the best way to create a system that increases Sunflower productivity?

1.5 Objectives

1.5.1 General objectives

This thesis' major objective is to develop a system for detecting and classifying sunflower leaf disease using deep learning technologies.

1.5.2 Specific objectives

The following specific objectives are organized to be achieved through the research phases:

1. Identify and develop an appropriate method for solving the identified gap(s)
2. Conduct a comprehensive assessment of recent scientific studies on the detection and classification of sunflower leaf disease.
3. Create appropriate data sets for the model to detect and classify sunflower leaf disease.
4. Design the SunfNet model for the proposed system.
5. Conduct a precise and accurate experiment in the identification and classification of sunflower leaf disease.

1.6 Significance of the study

For many stakeholders and other researchers, the study has distinct benefits. It can be used in a variety of fields following the outcomes evaluations:

- The study benefits the government and private agriculture-related enterprises, as well as farmers and investors.
- This work could be beneficial to agricultural research institutes.

1.7 Scope of the study

This study proposes to look at four different types of sunflower leaf diseases. Alternaria, leaf spot, downy mildew, and rust are among them. The suggested approach concentrates only on the leaf of the sunflower plant, ignoring the flower, root, fruit, and other parts of the plant.

1.8 Limitation of the study

The disease phased another area of the sunflower is not the topic of this study. Instead, it concentrates just on the design and development of a Model for the detection and categorization of sunflower leaf diseases. There are a variety of diseases that can affect sunflowers, but this study focused on only four of them. Time and financial constraints are the causes for this. To incorporate many disease kinds in this study, proper dataset gathering is also a challenging activity.

1.9 Organization of the Thesis

Chapter one introduces the background concept and ideas about this research paper. Chapter two presents the review of literature that discusses ANN, CNN, and related works. Chapter three of this research paper deals with methodologies, methods, and tools used in conducting this research. Chapter four is about the implementation of the system as well as the results of the research experiment. Chapter five covers discussions, analysis, and evaluation of the result whereas chapter six generalize the research using conclusions and recommendations. References, interview questions, and programming scripts are also included as appendixes.

2 CHAPTER TWO: LITERATURE REVIEW AND RELATED WORK

2.1 Introduction

In this section, literature that represents researches related to the topic has been reviewed. The review covers concepts of ANN, CNN, RNN, GAN Deep Learning, Data Mining, Sunflower disease, and plant disease detection approaches

2.2 Artificial Neural Network

An artificial neural network (ANN) is a computational model based on the structure and functions of natural neural systems. These structures are influenced by the information flow through the networks. This model has three layers, which are interconnected with each other to make the structure. The first layer is known as the input layer or input neurons. These neurons send data into the second layer which is called the hidden layer. This layer in turn sends the output data into the third layer which is called the output layer [28], [29].

In neural networks, the relations are covered up within the networks and their parameters. In the application of neural networks, explicit knowledge is not required, like knowledge-based techniques. Neural systems can in this way serve as black-box models of nonlinear, multivariable inactive, and energetic frameworks and can be prepared by utilizing input-output information watched on the framework. Most common ANN's consist of basic processing components called neurons, weighted interconnection among them. The weight between these neurons represents the strength of their relationships.

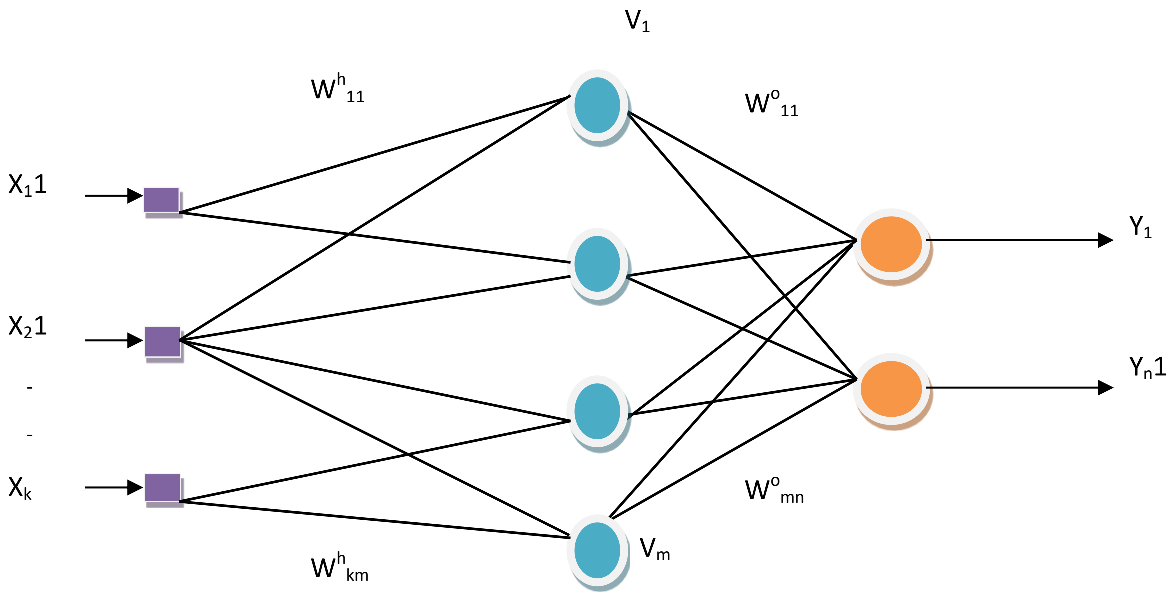


Figure 1: A feed-forward with one hidden layer [27].

2.3 Deep Learning vs. Machine Learning

2.3.1 Machine Learning

A branch of artificial intelligence that makes the machines intelligent to learn from learning algorithms and decide by applying what has been learned. Even if it is still a machine in some cases, it wants human intervention, because it works only for what it is designed for. It is less powerful in performance while deep learning is more powerful and flexible.

2.3.2 Deep Learning

Deep learning is a subset of machine learning but it has more power and flexibility in performance than traditional machine learning. The deep learning algorithms are trying to learn high-level features from data in an incremental manner while traditional machine learning algorithms extract features manually. When we have a large data size and good computational performances deep learning is the recommended approach. It is has been a good performance when it approaches complex problems like image classifications, natural language processing, and speech recognition.

When we have a large dataset and good computational performance, deep learning performs better and better. When it comes to very complicated issues like picture classifications, natural language processing, and speech recognition, it beats other approaches. In Deep Learning, we require two things. The first is a big amount of data to train our model, and the second is a good computational device because deep learning learns millions of parameters in a model, and our device should be able to quickly overcome those clusters of parameters.

2.3.2.1 Recurrent neural networks (RNN)

RNN is exceptionally valuable to handle time arrangement information and has been utilized in numerous rural zones, such as arrive cover classification, phenotype acknowledgment, edit abdicate estimation, leaf region record estimation, climate forecast, soil dampness estimation, creature inquire about, and occasion date estimation. Land cover classification (LCC) is considered a crucial and challenging errand in farming, and the key point is to recognize what courses an ordinary piece of arriving is in. Within the past, parcels of applications are based on mono-temporal perceptions and disregarding time-series impacts in a few issues. For occurrence, vegetation changes its spatial appearance intermittently, which can confound the mono-temporal approaches [59]. The difference between CNN and RNN is justified as follows:

- CNN's are commonly utilized in understanding issues related to spatial information, such as pictures. RNNs are way better suited to analyzing worldly, successive information, such as content or videos.
- A CNN encompasses a diverse design from an RNN. CNN's are "feed-forward neural systems" that utilize channels and pooling layers, though RNNs bolster comes about back into the organize (more on this point below).
- In CNN's, the estimate of the input and the coming about yield are settled. That's, a CNN gets images of settled measures and yields them to the fitting level, together with the certainty level of its forecast. In RNNs, the measure of the input and the coming about yield may vary.
- Use cases for CNNs incorporate facial acknowledgment, therapeutic examination, and classification. Utilize cases for RNNs incorporate content interpretation, normal dialect preparing, and estimation investigation, and discourse analysis.

2.3.2.2 Generative adversarial networks (GAN)

GAN may be a way that a set of commotion is utilized to memorize the dispersion of the genuine information and to generate unused information. The structure of this organize comprises of two models, an era show which is to capture the dispersion of the genuine information, and a separation show which is comparative to a parallel classifier [60].

2.4 Data Mining

Machine learning uses data mining techniques and different learning algorithms to build a model. Data mining is the process of extracting knowledge from large amounts of data, whereas machine learning is a way to invent a new algorithm the knowledge. In several fields, the advancement of information technology has resulted in the creation of vast databases and large amounts of data. Database and information technology research has resulted in a method for storing and manipulating this valuable data to make more informed decisions. The practice of extracting usable information and patterns from large amounts of data is known as data mining. It is a logical process for searching through vast amounts of data for relevant information. The purpose of this technique is to discover previously unknown patterns. Once these patterns have been discovered, they can be used to make specific business decisions. Exploration, pattern identification, and deployment are the three processes involved in data mining [57].

Data is cleansed and translated into a different form in the first step of data exploration, and then essential factors and the nature of data based on the problem are determined. The second phase is to identify patterns in the data after it has been studied, refined, and specified for the specific variables. Identify and select the patterns that make the most accurate predictions. The pattern is then deployed for the needed purpose. For knowledge discovery from databases, several algorithms, and techniques such as Classification, Clustering, Regression, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees, Genetic Algorithms, the Nearest Neighbor method, and others are employed [57].

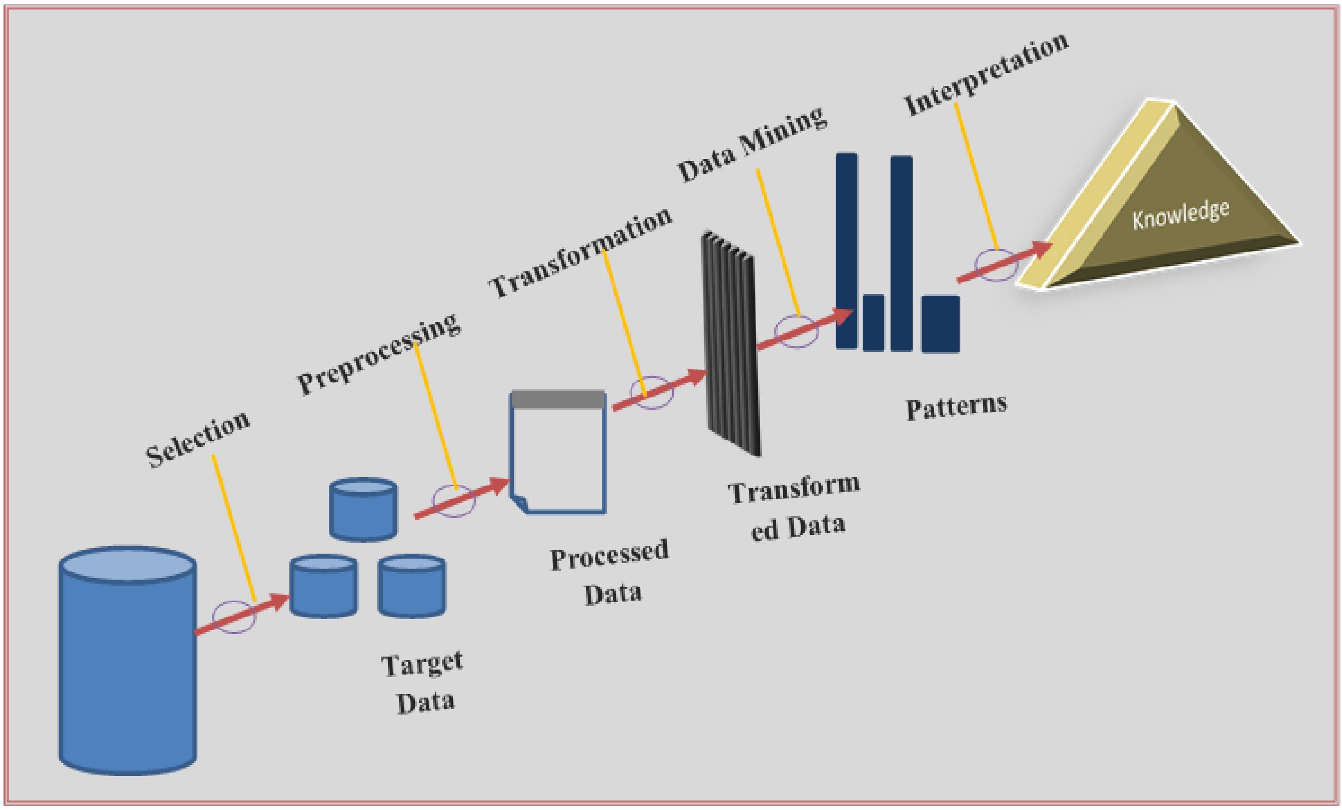


Figure 2: Knowledge discovery process in data mining [57].

The most widely used data mining technique is classification, which uses a set of pre-classified examples to create a model that can categorize a huge number of records. This form of analysis is especially well suited for fraud detection and credit risk applications. This method usually employs classification algorithms based on decision trees or neural networks. Learning and classification are both involved in the data classification process. The training data is examined by a classification algorithm in Learning. Data from classification tests are used to measure the accuracy of categorization rules. The rules can be applied to the new data tuples if the accuracy is appropriate [57].

Clustering is the process of identifying objects that belong to the same general category. We can detect overall distribution patterns and relationships among data attributes by utilizing clustering techniques to locate dense and sparse regions in object space. Classification can be used to differentiate groups or classes of objects, but it is time-consuming, therefore clustering can be used as a preprocessing method for attribute subset selection and classification. Prediction can be accomplished using regression techniques. One or more independent variables and dependent variables can be modeled using regression analysis. In data mining, independent variables are traits that are previously known, whereas responsive variables are the variables that we want to forecast. Sales volumes, stock prices, and product failure rates, for example, are all challenging to forecast due to intricate interactions among various predictor variables. To estimate future values, more complicated techniques (such as logistic regression, decision trees, or neural networks) may be required [57].

Data mining is a young technology that has yet to reach its full potential. Despite this, it is already being used by a variety of businesses regularly. Retail stores, hospitals, banks, and insurance companies are just a few of these businesses. Many of these businesses are merging data mining with other crucial technologies like statistics and pattern recognition. Data mining can be used to uncover patterns and connections that would be difficult to uncover otherwise [57].

2.5 Convolutional Neural Network

The field of AI has taken an emotional turn lately, with the ascent of the Fake Neural Organization (ANN). These naturally propelled computational models can far outperform the exhibition of past types of computerized reasoning in like manner AI undertakings. Quite possibly the most noteworthy type of ANN engineering is that of the Convolutional Neural Organization (CNN). CNN's are principally used to settle troublesome picture-driven example acknowledgment assignments and with their exact yet straightforward design. [34]

Convolutional neural organizations (CNNs) have accomplished cutting-edge results on genuine applications, for example, picture characterization and items. [35] CNN has been shown effectively on different PC vision undertakings. CNN applied picture characterization, location, and division. [30, 31, 32, 33] Profound learning alludes to the magnificent part of AI that depends on learning levels of representations. Convolutional Neural Organizations (CNN) is one sort of profound neural organization. Convolutional Neural Organizations (CNN) are the lone sort of feed-forward neural organization. CNN is an effective acknowledgment measure that is generally utilized in plan acknowledgment and picture handling. It has many components like a straightforward construction, less preparing boundaries, and versatility. [36]

The design of CNN incorporates two layers one is the element extraction layer, the contribution of every neuron is associated with the nearby responsive fields of the past layer and concentrates the neighborhood include. When the nearby elements are separated, the positional connection among it and different components are additionally not set in stone. The other is the element map layer; each registering layer of the organization is made out of a majority of component maps. Each element map is a plane, the heaviness of the neurons in the plane are equivalent. The construction of the component map utilizes the sigmoid capacity as the enactment capacity of the convolution organization, which makes the element map have shift-invariance. Additionally, since the neurons in a similar planning plane offer weight, the quantity of free boundaries of the organization is decreased. Every convolution layer in the convolution neural organization is trailed by a figuring layer which is utilized to ascertain the nearby normal and the subsequent concentrate, this one-of-a-kind two-component extraction structure lessens the goal. [36]

CNN is primarily used to recognize uprooting, zoom, and different types of mutilating in the change of two-dimensional illustrations. In this way, the element identification layer of CNN learns via preparing information. CNN evades express element extraction and verifiably gains from the preparation information. Additionally, the neurons in a similar element map plane have indistinguishable loads, so the organization can concentrate all the while. This is a significant benefit of the convolution network concerning the neural organization association. This design of CNN's nearby common loads causes it to enjoy an extraordinary benefit in picture handling. Its design is nearer to the real organic neural organization. A common weight decreases the intricacy of the organization. [36]

2.6 CNN Architecture

The CNN engineering incorporates a few structure blocks, for example, convolution layers, pooling layers, and complete associated layers. Regular engineering comprises reiterations of a pile of a few convolution layers and a pooling layer, trailed by design and the preparation cycle. A CNN is made out of stacking a few structure blocks: convolution layers, pooling layers (e.g., max-pooling), and completely associated (FC) layers. A model's exhibition under bits and loads is determined with a misfortune work through at least one completely associated layer. The progression where input information is changed into yield through these layers is called forward proliferation. [37]

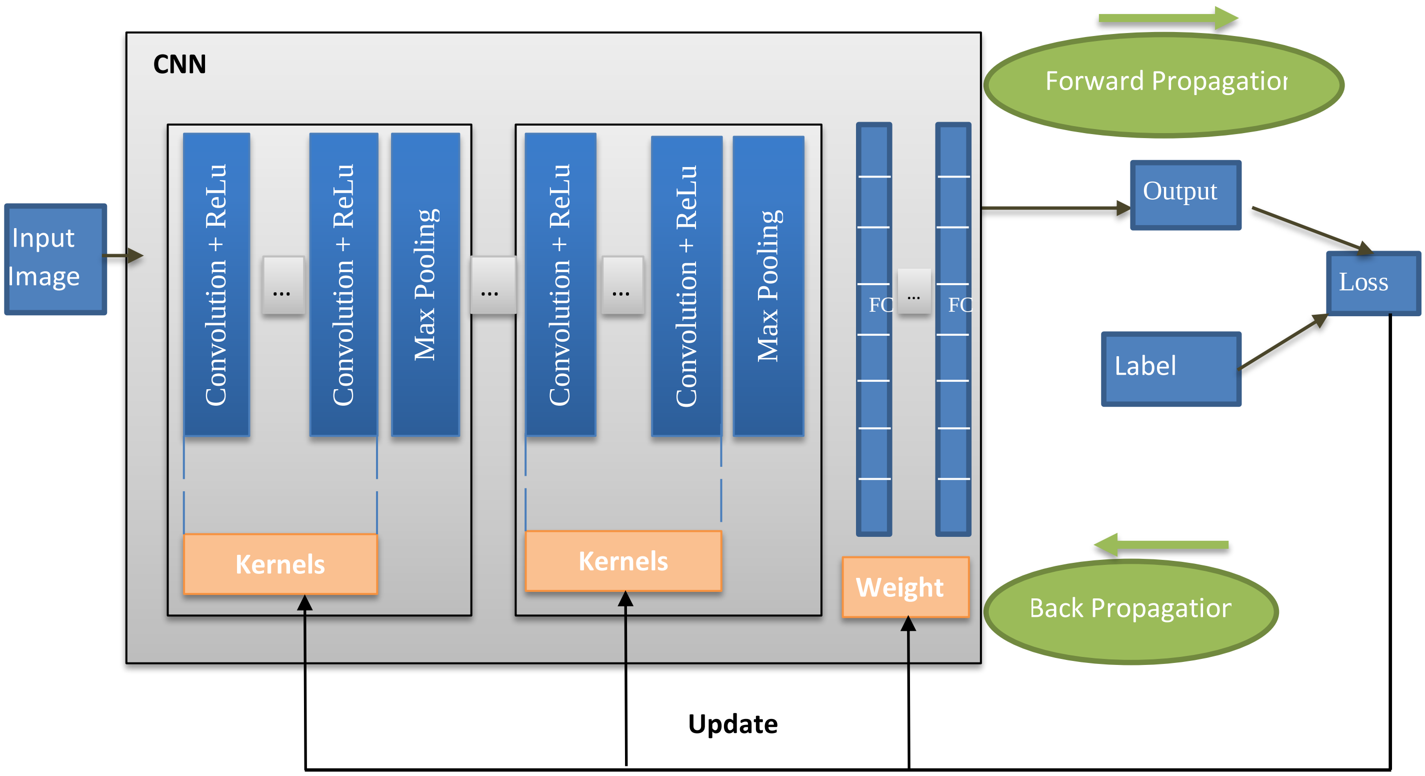


Figure 3: CNN Architecture

In CNN architecture there are various layers such as convolution layers, pooling layers, and fully connected (FC) layers.

I. Convolution Layer

One major component of CNN architecture is the convolution layer which mainly performs feature extraction. ReLU contains convolution operation and activation function which is a combination of linear and nonlinear operation. Images are read as a pixel and represented in the form of a matrix (hxwx3) in computer vision. This means height by width by depth respectively. The depth 3(three) represents that of the colored image which has three channels (RGB) [26][27]. Filtering is the main process of this layer in CNN architecture. A set of learnable filters (kernels) are made by this layer. Detecting the presence of specific features or patterns present in the input image is done by filters kernels).

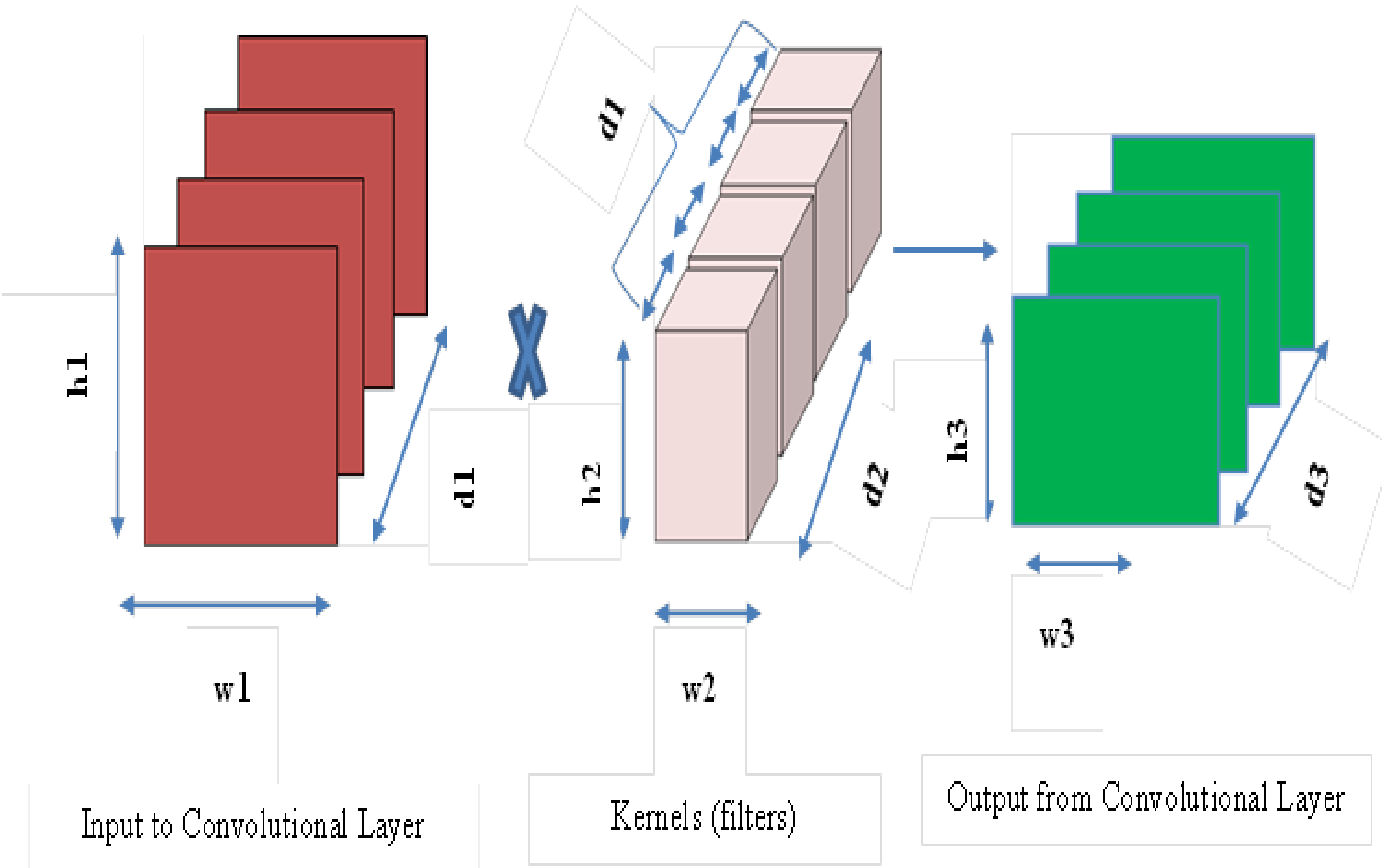


Figure 4: Convolutional Layer

II. Pooling Layer

The one important layers of CNN is the pooling layer which lowers the computational burden, by reducing the number of connections between convolutional layers [26] [28]. It is used to control overfitting by reducing the spatial size of the network. The most used pooling layer in a convolutional neural network is max pooling. As the name implies max-pooling takes only the maximum from the pool. The main purpose of this layer is to reduce the number of input tensor parameters thus support to reduce over-fitting, extracting representative features and reduce competition which enhances performance. In this layer, the input image is downsampled in its height and width but the width stays constant.

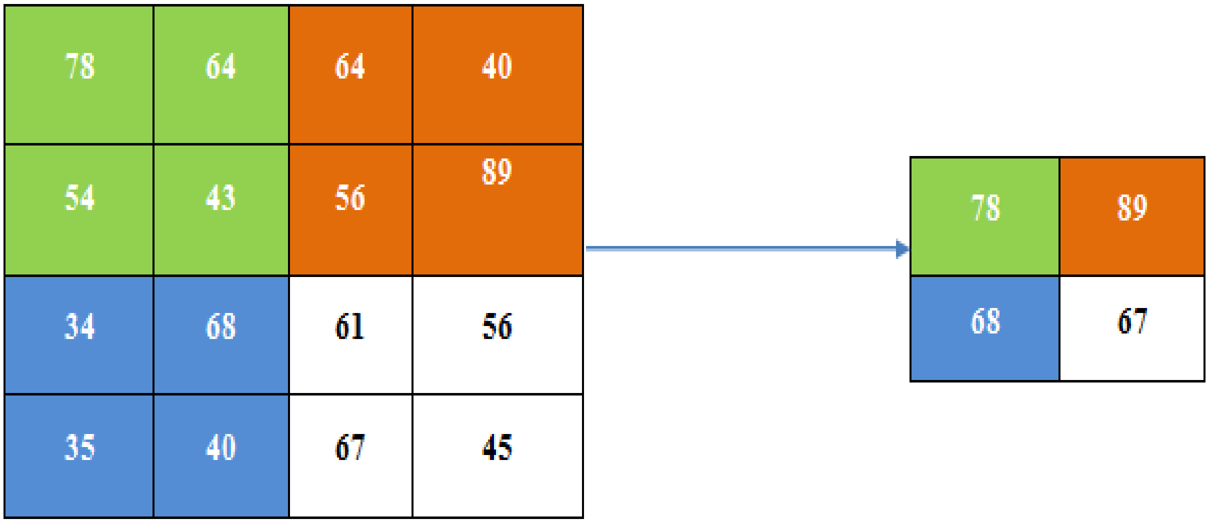


Figure 5: Max Pooling [46]

III. Fully Connected Layer (FCL)

Fully connected is the last layer in CNN architecture. It means that neurons of preceding layers are connected to every neuron in subsequent layers. Input to fully connected layer is the output from prior layers which is the pooling layer. The input to the fully connected layer is in flattened mode. Fully connected layers are used to calculate the model performance under entire kernels and weights with a loss function. The step in which input data is transformed into output through these layers is known as forwarding propagation.

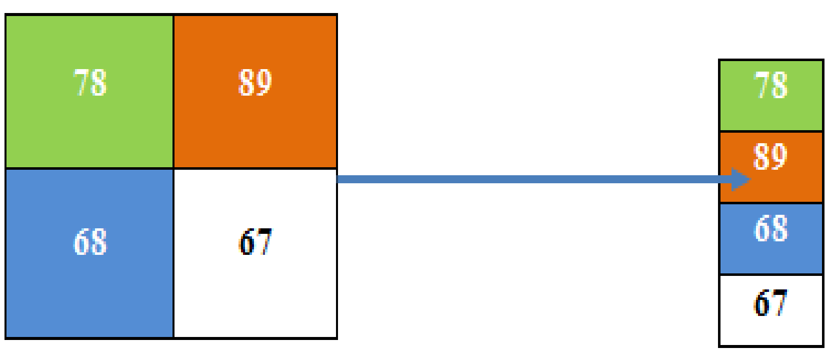


Figure 6: Flattening [46]

2.7 Activation Function

The flow of signals from one layer to the next is controlled by activation. In the design of neural networks, the activation functions are a critical part. Choice of activation function in deep learning is important because it defines the network model prediction. The activation functions used in hidden layers and output layers are typically different. There are three types of activation functions used in neural network models. Activation functions are functions that are employed in neural networks to compute the weighted total of input and biases, which are then used to determine whether a neuron can fire. It manipulates the provided data using gradient processing, most commonly gradient descent and then produces an output for the neural network that contains the data's parameters.

Activation functions are essential to convert linear input models into non-linear output models, which facilitate the learning of high-order polynomials beyond one degree for deeper networks. Non-linear activation functions have the unique virtue of being differentiable; otherwise, they will not work during deep neural network back-propagation. The deep neural network is a neural network with numerous hidden layers and an output layer. An understanding of the character of the multiple hidden layers and output layer is our consideration. A typical deep learning model block diagram is depicted in the diagram below. Figure 3 depicts the three layers that comprise a DL-based system, with a focus on the activation positions. in the respective blocks, which are represented by the dark shaded region.

Figure 7: Block diagram of Deep Learning

The input layer acknowledges information in many organizations for preparing the neural organization, including photographs, recordings, texts, voice, sounds, and numeric information. The convolutional and pooling layers make up the vast majority of the secret layers, with the convolutional layers recognizing neighborhood examples and provisions in information from earlier layers and introducing them in cluster-like structures for pictures. The pooling layer semantically combines comparative elements into one [52].

The yield layer shows network results that are oftentimes affected by AFs, like groupings of figures, along with related probabilities. The job of an AF in an organization structure figures out where it is situated in the construction; for instance, when the AF is put after the secret layers, it deciphers the learned straight mappings into non-direct structures for proliferation, and when it is set in the yield layer, it makes expectations. Profound designs are comprised of various handling layers, every one of which incorporates both direct and non-straight activities and is prepared altogether to do a job. More profound organizations have more prominent execution, however, they experience the ill effects of hardships like evaporating slopes and detonating angles as a result of the subordinate terms, which are frequently under 1.

As the subordinate terms are duplicated more occasions, the worth therapists and psychologists until it arrives at nothing, bringing about the slope. Therefore, if the qualities are multiple, successive augmentation will raise the qualities and angle. Subsequently, the AFs keep the inclinations' qualities inside specific limits. These are accomplished utilizing different numerical capacities, and Elliott, 1993 investigated a portion of the early proposition for actuation capacities utilized in neural organization registering while at the same time exploring the utilization of AFs in neural organizations [56].

I. ReLU activation Function

It is the main popular activation function used in hidden layers of neural networks. This is because of its simplest to implement and effective in performing than others activation functions. ReLU is less vulnerable by vanishing gradients which is unable to propagate useful gradient information from the output of the model back to the layers near the input of the model.

The mathematical form of the ReLU activation function is as follows.

$$\mathbf{F(x) = (0.0, x)}$$

Equation 1: ReLU Function

It means that if the input is negative value the result becomes zero. When the result is zero the neuron does not get activated. Subsequently, only a certain number of neurons are activated, this makes a ReLU function more computationally efficient when compared to the sigmoid and tanh functions.

II. Softmax Activation Function

One more type of enactment work used in neural processing is the Softmax work. It is utilized to make a likelihood appropriation out of a bunch of genuine numbers. The yield of the Softmax work is a scope of qualities somewhere in the range of 0 and 1, with the amount of the probabilities equivalent to 1. In multi-class models, the Softmax work returns probabilities for each class, with the objective class having the most elevated probability. Practically every one of the yield layers in profound learning frameworks utilizes the Softmax work [53], [55].

III. Sigmoid Activation Function

It is quite possibly the most utilized non-straight enactment capacities in customary neural organizations. It changes the qualities somewhere in the range of 0 and 1. The numerical articulation for the sigmoid capacity is as per the following. Typically, we utilize a sigmoid capacity when an issue is settled with the double arrangement, which implies that the outcome lies somewhere in the range of 0 and 1 [32]. In this enactment work, the neuron initiation becomes soaked at one of the other 0 or 1. Accordingly, the backpropagation neglects to alter whole boundaries.

$$\boldsymbol{f(x) = 1/(1 + e^{-x})}$$

Equation 2: Sigmoid Function

In certain publications, the Sigmoid AF is referred to as the logistic function or squashing function [54]. The results of the sigmoid function study have resulted in three sigmoid AF versions that are employed in DL applications. The Sigmoid is a nonlinear AF that is mostly utilized in feed-forward neural networks.

IV. Tanh Activation Function

It is one of the neural network activation functions which is much more like the sigmoid activation function. The difference between is only the symmetric around the origin. Tanh is transforming the value between -1 and 1 while sigmoid is between 0 and 1. The mathematical representation for tanh is as follows.

$$\mathbf{tanh(x) = 2\,sigmoid(2\boldsymbol{x}) - 1}$$

Equation 3: Tanh Function

2.8 Overview of Problem domain

Sunflower is one of the significant oilseed crops and conceivably fits in the horticultural framework. The sunflower is the fourth most generally developed oil crop on the planet after palm, soybean, and rapeseed. [5] rather than other oil crops, sunflower is ecologically amiable, since it develops under low sources of info (water, soil composts, and fungicides). It is developed on more than 22 million hectares around the world, with a creation of 26 million tons. [38] Sunflower sicknesses are one of the significant constraints in impacting the creation steadiness of this harvest around the world. Even though there are more than twelve microbes that might assault developed sunflower bringing about yield misfortunes, only a couple are of worry in a specific nation or district. [39]

A few sicknesses, creepy crawlies, and nematodes assault harm the sunflower crop brings about a wide scope of misfortunes underway and yield. Sunflower is defenseless to illnesses of different sorts. The incomparable genuine illnesses of sunflower are brought about by parasites. The primary sicknesses incorporate rust, fleece mold, charcoal decay, and leaf curse (Alternaria), and leaf spot. The effect of these infections on plant yields is serious. [38]

- **Rust**

Sunflower rust is the most economically significant foliar disease. Sunflower rust is caused by pathogens that are native to sunflowers and cannot be transmitted to other crops. As spores can survive on wild, volunteer, or commercial sunflower residue, the sunflower rust fungus can survive even the harshest winters. Frequent periods of leaf wetness (fog, dew) and temperatures between 55 and 85 F are suitable for rust infection [47].

- **Downy mildew**

This is one of the most popular sunflower diseases. The disease preys on plants early in their growth cycle, causing them to wither and die. On seedlings, the symptoms include a yellowing of the leaves and a cottony white fungus development on the lower leaf surfaces. In the four to eight leaf stages, there is also a secondary infection.

- **Charcoal rot**

This disease is caused by the organism Macrophomina phaseolina which offensives 400 plants, such as soybeans, eatable beans, corn, and sorghum. The illness is most regularly found in hot and dry locales. The indications are by and large not seen until after blooming and are silver-gray injuries on the stalk close the soil line. This infection has not been of financial concern but for confined regions.

- **Alternaria**

This disease has significant yield losses in hot and humid areas. The disease defoliates the leaves by creating dark spots. The stem is infected as well resulting in stem breakage.

2.9 Overview of Plant Diseases

Agriculture is a crucial component of the World's economy. Cultivation is the primary source of income for the majority of the world's people. Many crops are grown in Ethiopia, with sunflower being one of the most important food grains and oilseeds that the country grows and exports. As can be seen, sunflower is a significant aspect of their agricultural system and economy. As a result, maintaining the above-mentioned crop's consistent production is critical. The major goal of this study is to present a unique approach based on a deformation model for handling sunflower detection and classification. To identify disease in plant leaves, the existing system uses the segmentation approach simple linear iterative clustering. Visual aspects such as color, gradient, texture, and shape are also used to define the characteristics of leaves. The cropped image of a sunflower plant will be defined by image processing and feature extraction techniques in the proposed system. The RGB color space is converted so that the color information contained in the images can be used effectively to differentiate images and segmentation usage of the k-means algorithm has been suggested. [40]

This paper presents a survey on methods that use digital image processing techniques to detect, quantify and classify plant diseases from digital images in the visible spectrum. Although disease symptoms can manifest in any part of the plant, only methods that explore visible symptoms in leaves and stems were considered. This was done for two main reasons: to limit the length of the paper and because methods dealing with roots, seeds and fruits have some peculiarities that would warrant a specific survey. The selected proposals are divided into three classes according to their objective: detection, severity quantification, and classification. Each of those classes, in turn, is subdivided according to the main technical solution used in the algorithm. This paper is expected to be useful to researchers working both on vegetable pathology and pattern recognition, providing a comprehensive and accessible overview of this important field of research. [41]

The quality and amount of the rice crop are indeed diminished because of plant sickness. This paper proposes a rice impact infection discovery component utilizing a Machine learning calculation, to distinguish the illness in the beginning phase of yield development. The proposed strategy would discover the impact of illness and diminish the harvest misfortune and thus viably increment rice horticulture creation. The pictures of the paddy field are caught and eight elements are separated to recognize the sound and the infection-influenced leaves. The proposed AI-based arrangement procedure incorporates KNN and ANN. The presentation of these two characterization methods is thought about utilizing a suitable disarray lattice. The reenactment results show that KNN based characterization strategy gives an exactness of 85% to the impact-influenced leaf pictures and 86% for the typical leaf pictures. The exactness is improved to almost 100% and 100% separately for the ANN-based characterization instruments. [42]

The issue related to programmed plant infection recognizable proof utilizing noticeable reach pictures has gotten impressive consideration over the most recent twenty years; nonetheless, the methods proposed so far are normally restricted in their extension and ward on ideal catch conditions to work appropriately. This evident absence of huge progressions might be to some degree clarified by some troublesome difficulties presented by the subject: the presence of mind-boggling foundations that can't be effortlessly isolated from the locale of interest (generally leaf and stem), limits of the indications regularly are not obvious, uncontrolled catch conditions might introduce qualities that make the picture examination more troublesome, certain illnesses produce manifestations with a wide scope of attributes, the side effects created by various sicknesses might be the same, and they might be available at the same time. This paper gives an investigation of every last one of those difficulties, underscoring both the issues that they might cause and what they might have conceivably meant for the strategies proposed previously. Some potential arrangements fit for defeating to some degree a portion of those difficulties are proposed. [43]

2.10 Related Work

In the previous few years analysts have proposed various strategies in the order and recognition of plant illness some of them are;

In paper [13] creators proposed an apple leaf affliction affirmation using significant learning systems subject to thick net 121 significant CNN, proposing three methods for backsliding, multi mark portrayal, and focus Los work. The proposed procedure has an exactness of 93.51%, 93.31%, 93.71% separately. In the paper [14] creators proposed tomato leaf contamination acknowledgment using the CNN approach. In the proposed model they introduced 3 convolution and 3 MAX poling layers followed by 2 related layers similarly they proposed a pre-arranged model, for instance, VGG 16, beginning V3, and convenient net. They achieved a gathering precision of 76%-100% concerning classes and the ordinary accuracy of the proposed strategy is kept up 91.2%.

In paper [15] creators proposed a significant trade acquiring from the image for plant sickness ID using the significant convolutional neural association strategy, the redesigned VGG-Net with inception module alongside INC-VGGN organization. They presented the weighted pictures using pre-arranged association rather than getting ready without any planning by aimlessly instating the heaps using a colossal named enlightening record. The proposed approach has played out no under 91.83% endorsement accuracy on open educational assortment.

In the paper [16] creators proposed a significant adapting approach to manage recognize leaf disorders using pictures of plant leaves. They composed three basic strategies for finders, which are speedier R-CNN, R-FCN, and SSD. The proposed procedures feasibly perceived different sorts of leaf contaminations with bewildering circumstances.

In the paper [27] creators present a picture handling method for distinguishing the Malus Domestic leaves sickness. Power upsides of grayscale pictures are gotten by the histogram evening out strategy. In picture division, the Co-event network strategy calculation is utilized for surface examination, and the K-implies grouping calculation is utilized for shading investigation. The surface investigation is the portrayal of districts in a picture by surface substance. Shading investigation alludes to limiting the number of squares of the distance among items and class centroid or relating group. In the edge coordinating with measure, singular pixels esteem is contrasted and limit esteem, assuming the worth is more noteworthy than the edge, it is set apart as an article pixel. The surface and shading investigation pictures are contrasted and the past pictures for the discovery of plant infections. The creator will utilize Bayes and K-implies bunching later on.

In the paper [24] creators present a picture preparing the procedure for Orchid leaf sickness recognition. Dark leaf spot and Sun sear are two kinds of orchid leaf sicknesses for the most part found. The fundamental stage of picture preparing is picture obtaining for catching pictures and stores them in a PC for additional activity. Picture pre-preparing includes histogram evening out, power change, and separating for improving or adjusting the picture. Three morphological cycles are utilized in the line division method to eliminate the little item and safeguard the huge article in the picture. Thresholding in the division is utilized for the beginning and stop point of the line to follow edges. The creator added ROI (area of interest) in GUI. After the line division measure, the order is finished by figuring white pixels in the picture. This framework gives high exactness and a low level of blunder in the outcome.

In the paper [25] creators, present a picture handling strategy for Tomato leaves infections recognition. In the picture procurement stage, advanced pictures of tainted tomato leaves are gathered which incorporate two sorts of tomato sicknesses specifically: Early curse and Powdery buildup. In the pre-preparing stage, a few procedures are applied for picture upgrade, perfection; expulsion of commotion, picture resizing, picture confinement, and foundation evacuation. The creator presented Gabor wavelet change and Support vector machine for the distinguishing proof and order of tomato infections. In the component extraction stage with the assistance of the Gabor wavelet change highlight, vectors are gotten for the following arrangement stage. In the grouping stage, the help vector machine (SVM) is prepared for distinguishing the class of tomato illnesses. The contributions of SVM are highlight vectors and comparing classes, while the yields are the choice that identifies tomato's leaf infection. SVM is utilized utilizing Invmult Kernel, Cauchy Kernel, and Laplacian Kernel capacities. Network search and N-overlay cross-approval strategies are utilized for execution assessment.

Sunflower infections are of significant worry in the creation of this harvest around the world. This is because of the normal and regularly extreme assaults by various microbes. Accordingly, extensive yield misfortunes happen or the nature of the item reduces. However the quantity of microbes known to assault sunflower is somewhat high, simply a modest bunch to twelve is viewed as significant ones relying upon district and cultivar. In this audit, I am zeroing in especially on these critical microbes. The accentuation will be on discoveries and results acquired by specialists identified with microbe science, environment, hereditary qualities, have obstruction, and control. It was intriguing to take note of a significant change in the overall strength of infections throughout the most recent four years as reflecting in the number of distributions accessible. The researchers' endeavors have brought about a superior comprehension of individual sicknesses and underline their importance in the improvement of sunflower the board. [39]

China is the biggest country on the planet for planting sunflowers, and sunflower is a significant money crop in Chinese agribusiness. Sunflower illnesses are turning out to be progressively genuine, precise recognizable proof, and powerful control of sunflower sicknesses is vital for nearby monetary improvement in China. Generally, sunflower infection recognizable proof is predominantly founded on the eye acknowledgment technique, which has incredible restrictions and this way is hard to meet the improvement needs of current farming. As of late, the fast improvement of computerized picture preparing innovation and PC innovation gives another stage to the advancement of sunflower illness distinguishing proof frameworks. This paper proposes a sunflower leaf illness recognizable proof strategy dependent on SIFT focuses for the most part for sunflower leaf infections. An inside and out examination was led on sunflower dark spots, bacterial leaf spots, fine buildup, and wool mold. [44]

Sunflower [5] is one of the fundamental oilseed plants and is likely appropriate for the farming framework and oil fabricating space of India. Sunflower crop gets broken through the method of a method for the impact of assorted diseases, bugs, and nematodes following in a gigantic assortment of misfortune in assembling. Sickness discovery is feasible through uncovered eye perception; in any case, this methodology is fruitless while one should uncover the enormous ranches. As a choice to this issue, we progressed and gift a gadget for division and sort of Sunflower leaf pictures. This examinations paper gives overviews performed on explicit ailment type techniques that might be utilized for sunflower leaf infirmity recognition. Division of Sunflower leaf pictures, which is a fundamental component for infirmity type, is completed through the method of the method for the utilization of Particle swarm improvement set of rules. Acceptable outcomes had been given through the method of the method for the investigation completed on leaf pictures. The normal precision of the kind of proposed set of rules is 98.0% when contrasted with 97.6 and 92.7% referenced in present-day techniques. [45]

Tomatoes are the second most broadly developed vegetable plant on the planet, after potatoes. It started in the tropical district between Mexico and Peru. Tomatoes are developed all through the nation and are perhaps the main vegetable harvests, with creation expanding drastically. Tomato leaf sicknesses are utilized to recognize and order indications of plant infections; notwithstanding, identifying tomato leaf illnesses with the unaided eye is pointless, particularly because there are such large numbers of them. Accordingly, we need to make a framework for identifying and characterizing tomato leaf infections naturally. To recognize and group tomato leaf ailments using RGB, Grayscale, and improved information picture datasets, we fostered the TomdiseasesNet model design, which depends on commencement v3, VGG16, and versatile net engineering. Contaminated harvest leaves are accumulated and marked by sickness. Picture preparing is joined with the pixel-by-pixel methodology to work on the picture. Then, at that point, to distinguish tomato plant leaf diseases, include extraction and arrangement of gathered leaf designs are performed.

Tomato Bacterial Spot, Tomato Early Blight, Tomato Sectorial Leaf spot, Tomato Leaf Mold, Tomato Yellow Leaf, and Healthy Leaf are the six classifier names that are utilized. With 200 ages, 70/30 parting proportion, and 0.0001 learning rate, the recovered elements are fitted into a neural organization. Testing exactness was 99.68 percent, preparing precision was 99.97 percent, and approval precision was 99.78 percent, as indicated by the TomdiseasesNet model. Because of the general exactness of 99.68 percent, tomato plant leaf sicknesses brought about by Tomato Bacterial Spot, Tomato Early Blight, Tomato Sectorial Leaf Spot, Tomato Leaf Mold, and Tomato Yellow Leaf might be precisely identified and grouped [48].

Due to many harvest sicknesses, crop decline is at present the world's most genuine worry in ensuring food security and fostering a reasonable horticultural framework. Following the medical issue of harvests is a key undertaking that should be finished to forestall the spread of different agrarian sicknesses, and it ought to be finished utilizing innovation as opposed to work. Utilizing a Deep Learning technique, explicitly the Convolutional Neural Network (CNN), this examination fosters a model that analyzes three sorts of wheat rust illness utilizing an image of the wheat crop.

Wheat leaf rust, stem rust, and yellow rust in three wheat infections vary by the way they sway the harvest and how much mischief they cause. This examination proposes utilizing shading code division to remove just the important data from a wheat picture to recognize the sound and wiped outcrops. The investigation has at last acquired 99.76 percent precision in recognizing wheat rust from sound harvests in the wake of executing more than 200 hundred preliminaries utilizing various influencing boundaries like Learning rate, Dropout, and train test split proportion [49].

It incorporates results from overviews on a few infection classification frameworks that can be utilized to identify sunflower leaf illness. The Particle Swarm Optimization method is utilized to fragment Sunflower leaf pictures, which is a basic advance in illness order. Examinations utilizing leaf pictures have shown acceptable outcomes. The proposed calculation's normal order precision is 98.0 percent, contrasted with 97.6 and 92.7 percent in current methodologies [50].

The application of digital image processing and machine learning technologies to achieve automatic recognition of plant leaf diseases is critical for crop disease prevention and control. Powdery mildew, bacterial leaf spot, black spot, and downy mildew of sunflower leaves were investigated in this research. The image of sunflower leaf disease is segmented using the K-means + + clustering technique and the watershed algorithm. To detect the infected areas, 19 color and texture feature values are taken from the diseased areas, and a random forest algorithm is built. The overall recognition rate is estimated to be around 95% [51].

These papers show that there is a need for in-depth research on sunflower disease for more accurate detection and classification. Further, it has been observed while reviewing the literature of research on leaf image Segmentation that proposed methods work well only on leaf images that were captured in a controlled environment. Texture feature extraction is a mainly used technique used by researchers and there is a need for in-depth research on sunflower leaf diseases for more accurate detection and classification. The proposed methods and algorithms are used to solve the gaps of these finds.

3 CHAPTER THREE: RESEARCH METHODOLOGY

3.1 General Approach

To accomplish the targets of the examination and to respond to the exploration questions, the accompanying examination techniques are utilized. Via looking through changed related writing (Internet, books, diaries, and so forth) are surveyed to comprehend the idea of Deep learning CNN and related works. To accomplish this examination level headed, the latest and current writing is investigated. During the modification of these bits of writing, the strength and shortcomings of the past arrangements were recognized to utilize them as a base to think of the proposed arrangement. The hotspots for the sorts of writing going to survey will be incorporated Springer Link, IEEE Xplore, Science Direct, IJERCSE, Academia, and the preferences. The calculation composed underneath delineated the bit by bit approach for the proposed sunflower illnesses location and order.

3.2 Software Tools and Techniques

3.1.1 Python

Python is an interpreted high-level, general-purpose programming language. It is an easy and powerful programming language that is dynamic and easy to catch up with. As it is a lo built-in library for Artificial intelligence and Machine Learning, well documented, simple enough to catch up with, python is preferred in this study. Some of the libraries are Tensor Flow, Keras, and NumPy. In this study, to produce the result, python 3.7.7 is used and runs on Windows 10, 64- bit operating system, with 4 GB RAM and Intel(R) Core (TM) i5-7200u CPU @ 2.50GHz 2.70 GHz of Processor.

3.1.2 Tenser flow

Tensor flow is a python library for fast numerical computing created by Google. It is a foundation library that can be used to create deep learning models and it is an open-source library. Tenser flow was designed for use in research and production systems. It can run on single CPU systems, GPU as well as mobile devices, and large-scale distributed systems on hundreds of machines. Tensor flow is a brilliant tool with a lot of power and flexibility.

3.1.3 Keras

Keras is a high-level neural network tool. Keras is written in Python and capable of running on top of the tenser flow. It was developed with motivation on enabling fast experimentation. It is a model-level library providing high-level building blocks for developing deep learning architectures. It does not allow low-level operation like tenser manipulation and differentiation, in the sake of that, it uses a specialized, well-optimized tenser library, serves as a back-end engine for Keras. It is used for easy and fast prototyping in user-friendliness, modularity, and extensibility.

Figure 8: some open-source libraries and our thesis Environment

3.3 Data collection

It is the way or mechanism in which we gather data to develop the system. The data set is collected using a digital camera and using other online datasets. To collect the required data traveling to sunflower farms will be undertaken. The dataset contains images with several diseases and healthy in many sunflower leaf plants. For proposed research, all the required data will be collected for those sunflower plant leaves from different sources like images download from the Internet, or simply taking pictures using camera devices.

In deep learning, data is the major important thing that is used to develop the model. In deep learning huge number of datasets are needed for training. In this study, three types of datasets are used for implementing the model which are the training set, validation, and testing set. Data collection is undertaken through traveling to Hawassa agriculture research institute and Bishoftu research institute. In addition to this, some of the data is collected from the internet to mix with local data images. To generalize this data some of the data images were also taken from different sunflower farms during different times. Through these processes total collected data for the experiment is about 9806 augmented images. This data set is the collection of the healthy and unhealthy data images. This research experimented on four different common types of sunflower diseases.

3.3.1. Data pre-processing

3.3.2. Procedures

Image Acquisition and Pre-processing: Image acquisition involves the steps to obtain the plant leaf and capture the high-quality images to create the required database. The images used in this experiment came from a variety of places. Images were acquired using a camera and some were obtained from the internet to construct the datasets for this study. Image preprocessing involves the steps of image contrast enhancement. Here, the captured image is enhanced to remove the noise from the image. The collected images are preprocessed by correcting their brightness, contrast, rotates, and scaling up their size.

Image Augmentation: is a technique that allows us to generate large data sets from small ones. We may not be able to obtain enough data for our study for a variety of reasons. Deep Learning is a type of methodology that works effectively when we only have a large data collection. The original images collected during data collection are insufficient for developing an efficient model. In this study Image Data Generator class of Keras library is used. Zooming, inverting, and shifting original image features are used as augmenting parameters in this process.

Feature Extraction: The disease portion from the image is extracted. This leaf diseases area is treated as a region of interest for image processing. Then, further features are extracted based on the disease symptoms that are used to detect the disease types.

The collected data images are RGB images of any size. Because most neural network models require a square input image, it is resized to 150x150 pixels with a consistent aspect ratio. Low-frequency background noise is removed, and the intensity is normalized. All the photos were manually cropped during image processing. Cropping all the photos by hand, creating a square around the leaves as part of the image processing. The image is checked to see if it has all the necessary information for feature extraction.

3.4 Data Augmentation

Data Augmentation is a technique that allows us to generate large data sets from small ones. We may not be able to obtain enough data for our study for a variety of reasons. Deep Learning is a type of methodology that works effectively when we only have a large data collection. As a result, we will need to supplement the small quantity of data we originally acquired to build a good model. Adding effects such as enhancing contrasts, flipping, rotating at different angles, and shifting vertically and horizontally are some of the ways that can be used to accomplish this. This process of augmentation allows us to generate many alternative data samples from a single data set.

This process allows us to generate many images from a single image by allowing us to build many diverse data examples from a single image. It is the best way to reduce over-fitting. Boosting the size of the training set is used to generalize the model. In this study, there were a total of 400 original images and these images are augmented to 9806 images using 10 features. Ten characteristics, for example, are utilized to create ten separate images from a single image these are such as, rotation, width shift, height shift, zoom, horizontal flip, fill mode, data format, brightness, shear, and rescaling.

3.5 Performance Evaluation Metrics

After implementing the system evaluating the system whether it is met the proposed objective or not is the next step. Different measures were used to assess the model's success rate and efficiency. Here the following confusion metrics are used to evaluate implemented model.

A confusion matrix is an evaluation statistic for describing a classification task's performance. A confusion matrix can be used to calculate precision, recall, and accuracy.

I. Recall

The recall measures the fraction of examples that were correctly identified to be positive among all positive examples.

Recall=TP/TP+FN

II. Precision

The precision score measures how precisely the classifier identifies positive examples by avoiding the incorrect identification of negative examples as positive. Thus, the precision score is calculated as a fraction of examples that are correctly predicted to be positives that have been predicted positive (correctly or incorrectly).

Precision= TP/TP+FP

III. F1 Score

F1 score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy.

F1 Score=2*(Recall*Precision)/ (Recall Precision)

Table 1: Confusion Matrix

	Actual positive	**Actual negative**
Prediction positive	True Positive (TP)	False Positive (FP)
Prediction negative	False Negative (FN)	True Negative (TN)

3.6 Optimization Algorithm

In the SunfNet model gradient descent optimization algorithm is used. The gradient descent algorithm is continuously resampling the gradient of the model parameters. The back-propagation step of the gradient descent process is where the error vectors are computed backward, starting from the final layer. Depending on the activation function, the partial derivative of the function concerning w is used to determine how much change is required. The learning rate is multiplied by the change value. It takes this value and subtracts it from the previous output to get the new value as part of the output. It could continue to do so until it reaches convergence.

4 CHAPTER FOUR

4.1 Proposed System

This topic focuses on the design of the proposed work and its experimental parameters. Specifically, the design of the proposed model and descriptions, how features are extracted, and classification are performed in the proposed model.

4.2 Model Design

A SunfNet is a Deep Learning calculation that can accept a picture as information, relegate significance (learnable loads and predispositions) to unmistakable viewpoints/objects in the picture, and recognize one from the other. The design of a SunfNet is roused by the association of the Visual Cortex and is comparable to the available example of Neurons in the Human Brain. Singular neurons can just react to boosts in a little space of the visual field called the Receptive Field. A few comparative fields can be stacked on top of one another to traverse the full visual field. When contrasted with other grouping techniques, the measure of pre-handling needed by a SunfNet is altogether less.

The vital benefit of using the CNN calculation utilized in the SunfNet model for identifying and ordering sunflower leaf illnesses is that it is progressively robotized than customary AI calculations. The need to plan various calculations for various issues in the traditional AI calculation requires the use of more hand-tailored calculations, while on CNN of this model, we simply need to plan one calculation for all issues. SunfNet is simply the convolutional neural organization model named without help from anyone else. This model is made from scratch utilizing an aggregate of 9806 increased information pictures which are classified into five classes. These classes are containing mildew, Alternaria, rust, leaf spot, and solid which are normal kinds of sunflower infections. The all-out gathered dataset is separated into five preparing sets, approval, and test informational collection. Diverse division strategies are utilized to part the dataset into subsets. For instance, 70%, 15%, and 15 %, for preparing, approval, and test, separately.

Experimentation with this information index is done to track down the best-performing model. The misfortune capacity of the double cross-entropy is utilized with the sigmoid activation work for the yield layer, and the effective Adam inclination plummet procedure is used to become familiar with the loads utilizing the most regular and proficient learning paces of 0.001 and 0.0001. The primary layer is a convolution layer called a Conv2D. The layer has 4 element maps, every one of which has a size of 3x3 and a rectifier initiation work. This is the info layer, which expects pictures looking like (pixels, width, and height), which I gave picture sizes of 150X150 pixels. The accompanying layer involves a pooling layer named MaxPooling2D that takes the greatest worth. It is set up with a 2x2 foot-size pool.

A convolutional layer with 4 element guides of 3x3 sizes and a rectifier actuation work called ReLU, just as a pooling layer of 2x2. The subsequent layer incorporates a convolutional layer 8 component maps every one of which has a size of 3x3 and a rectifier enactment work, trailed by a maximum pooling layer with a 2x2 pool size. The third layer has 16 element maps, every one of which has a size 3x3 and a rectifier actuation work, trailed by a maximum pooling layer with a 2x2 pool size. The fourth layer has 32 element maps, every one of which has a size 3x3 and a rectifier initiation work, trailed by a maximum pooling layer with a 2x2 pool size. The fifth layer has 32 element maps as the fourth layer yet in this layer, each element map has a size 5x5 and a rectifier initiation work, trailed by a maximum pooling layer with a 2x2 pool size. The 6th layer has 64 component maps, every one of which has a size 3x3 and a rectifier enactment work, trailed by a maximum pooling layer with a 2x2 pool size. The seventh layer has 64 element maps as the fourth layer yet in this layer, each component map has a size 5x5 and a rectifier actuation work, trailed by a maximum pooling layer with a 2x2 pool size. The eighth layer has 150 component maps, every one of which has a size 3x3 and a rectifier initiation work, trailed by a maximum pooling layer with a 1x1 pool size. The 10th layer has 150 component maps, every one of which has a size 1x1 and a rectifier actuation work, trailed by a maximum pooling layer with a 1x1 pool size. The following layer changes the 2D network information over to a vector called to straighten it permits the yield to be handled by standard completely associated layers streamed by a rectifier actuation work. The following layer is a completely associated layer with 64 neurons with a rectifier initiation work. The yield layer has two neurons for every one of the two classes, just as a Softmax/sigmoid initiation work produces likelihood-like expectations for each class. The model is prepared with the ADAM angle drop procedure and diverse learning rates utilizing twofold cross-entropy as the misfortune work.

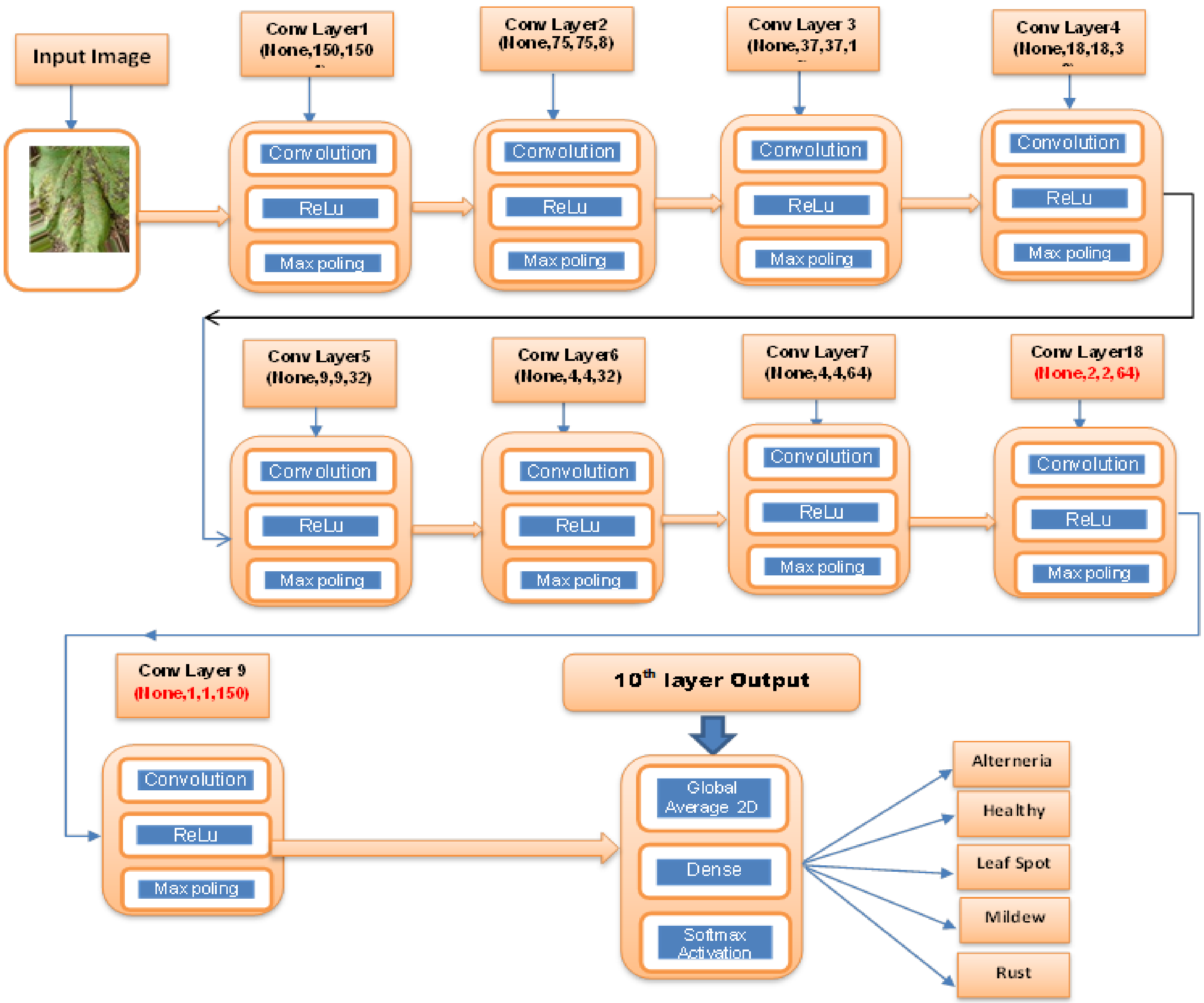


Figure 9: SunfNet model schematic diagram

1) Input Layer

The input layer of our SunfNet model accepts grayscale and RGB images of size 150*150*3 with five different classes (1, Alternaria. 2, leaf spot. 3, Healthy. 4, mildew. 5, rust.) This layer only passes the input to the first convolution layer without any computation. Therefore, there are no learnable features and the number of parameters in this layer is 0.

2) Convolutional Layer

The convolutional layer is the main and first layer of the CNN architecture. It is used to extract the characters from the input image. The convolution layer maintains the association connects pixels by learning image features from input data. The mathematical operation takes two input image matrices and 3x3 kernel filters. The 5x5 image pixel values and the convolution filter of 3x3. The CNN adds and divides the result by the total number of pixel and then create a map and put the value of the filter at that place. After that, it moves the feature to every other position of the image and obtains the output of the matrix, and repeats the steps for the other filters. This layer moves the filter to every possible position on the image.

3) Pooling Layer

This layer compresses the image into a smaller size by taking the maximum value from the filtered image and obtains a matrix out of it. It also controls overfitting. In the SunfNet model, there are three max-pooling layers.

4) Fully Connected (FC) Layer

In the SunfNet model, there are two fully connected layers including the output layer. The first fully connected layers have 64 neurons each and the final layer which is the output layer of the model has only one neuron. The first FC layer accepts the output of the sevens Convolution layer after converting the 3D volume of data into a vector value (Flattening). This layer computes the class score and the number of neurons in the layer predefined during the development of the model. It is the same as ordinary neural networks and as the name implies, each neuron in this layer is connected to all the numbers of layers.

5) Output Layer

The output layer is the last (the third FC layer) of the model and it has 1 neuron with a SoftMax activation function. Because the model is designed to classify five classes called categorical classification.

Table 2: Summary of SunfNet model parameters

Model: "sequential_2"
____
Layer (type) Output Shape Param #
==
conv2d_18 (Conv2D) (None, 150, 150, 4) 112
____
norm_1 (Batch Normalization) (None, 150, 150, 4) 16
____
max_pooling2d_18 (Max Pooling (None, 75, 75, 4) 0
____
conv2d_19 (Conv2D) (None, 75, 75, 8) 296
____
norm_2 (Batch Normalization) (None, 75, 75, 8) 32
____
max_pooling2d_19 (Max Pooling (None, 37, 37, 8) 0
____
conv2d_20 (Conv2D) (None, 37, 37, 16) 1168
____
norm_3 (Batch Normalization) (None, 37, 37, 16) 64
____
max_pooling2d_20 (Max Pooling (None, 18, 18, 16) 0
____
conv2d_21 (Conv2D) (None, 18, 18, 32) 4640
____
norm_4 (Batch Normalization) (None, 18, 18, 32) 128
____
max_pooling2d_21 (Max Pooling (None, 9, 9, 32) 0
____
conv2d_22 (Conv2D) (None, 9, 9, 32) 25632
____
norm_5 (Batch Normalization) (None, 9, 9, 32) 128
____
max_pooling2d_22 (Max Pooling (None, 4, 4, 32) 0
____

conv2d_23 (Conv2D) (None, 4, 4, 64) 18496
____
norm_6 (Batch Normalization) (None, 4, 4, 64) 256
____
max_pooling2d_23 (Max Pooling (None, 2, 2, 64) 0
____
conv2d_24 (Conv2D) (None, 2, 2, 64) 102464
____
norm_7 (Batch Normalization) (None, 2, 2, 64) 256
____
max_pooling2d_24 (Max Pooling (None, 1, 1, 64) 0
____
conv2d_25 (Conv2D) (None, 1, 1, 150) 86550
____
norm_8 (Batch Normalization) (None, 1, 1, 150) 600
____
max_pooling2d_25 (Max Pooling (None, 1, 1, 150) 0
____
conv2d_26 (Conv2D) (None, 1, 1, 150) 22650
____
norm_9 (Batch Normalization) (None, 1, 1, 150) 600
____
max_pooling2d_26 (Max Pooling (None, 1, 1, 150) 0
____
global_average_pooling2d_2 ((None, 150) 0
____
activation_4 (Activation) (None, 150) 0
____
dropout_2 (Dropout) (None, 150) 0
____
dense_2 (Dense) (None, 5) 755
____
activation_5 (Activation) (None, 5) 0
===
Total params: 264,843

Trainable params: 263,803
Non-trainable params: 1,040

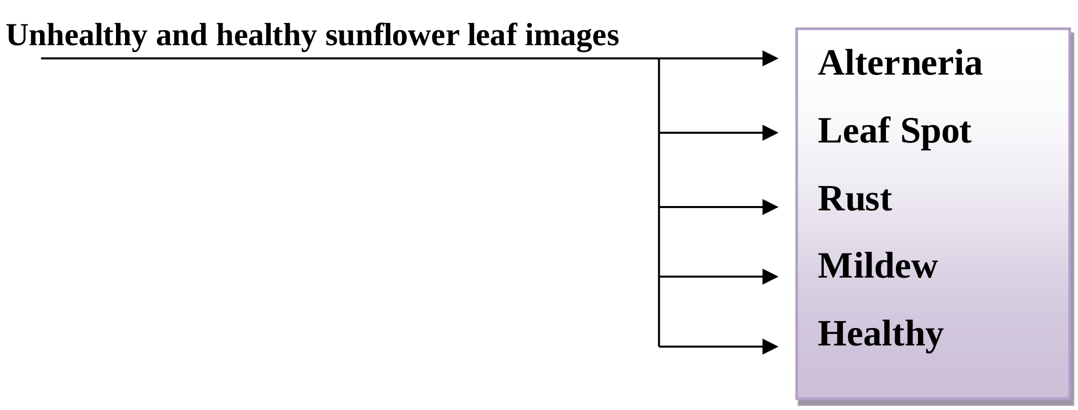
- **Feature Extraction**

Feature extraction is the main stage of image classification. The important features that are used to classify the image are extracted by the CNN algorithm. In the existing system using some features parameters like color feature, this feature is considered because their visual color difference identifies whether the crop is infected by the disease or not by a human vision in the previous system. In this proposed model it automatically extracts features from sunflower leaf. Deep learning enables machines to learn from experience and understand. Deep learning learns pathogens and classifiers naturally not like machine learning.

- **Dataset**

The data has been collected from a different place as discussed in a methodology. It contains the data which is affected by the different common diseases of sunflower. The dataset used in this study has five classes. These are the common types of sunflower diseases. Which are namely Alternaria, leaf spot, mildew, and rusts in addition to this, the healthy leaf images are also included as one class.

There is a total of 4 kinds of diseases and 1 health leaf that resend to categorize input images the data structure format is shown as follows:



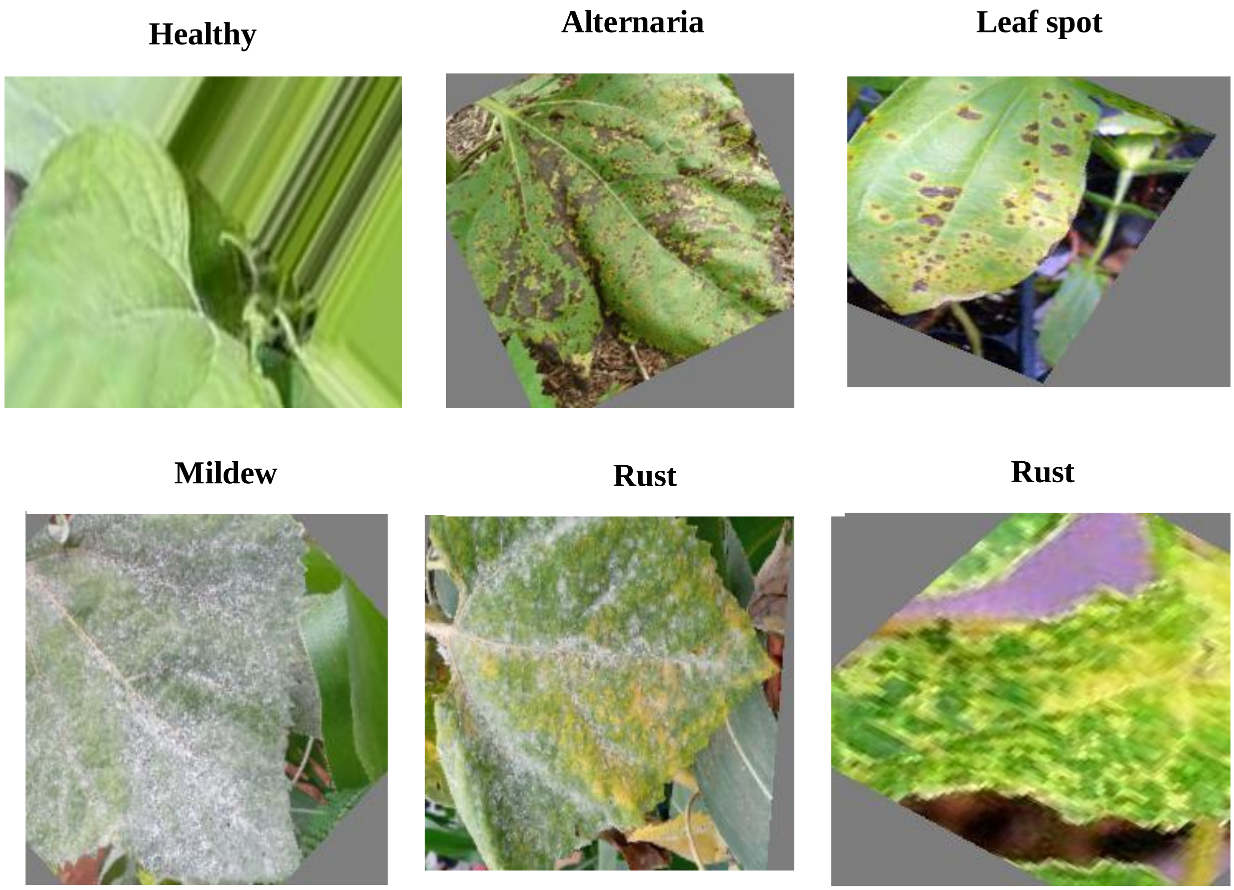


Fig 10: Sample image from the defined dataset

4.3 Optimization algorithms

SunfNet model is prepared to utilize a slope plunge advancement calculation to limit the mistake rate and the back-proliferation calculation is utilized to refresh the loads. Slope plunge is the most famous and the most generally utilized enhancement calculation in profound learning investigates. Simultaneously, every cutting-edge Deep Learning library contains executions of angle plummet enhancement calculations like Keras (utilized in this postulation). It refreshes the heaviness of the model and tunes boundaries, accordingly, limit the misfortune work. The Adaptive Moment Estimation (Adam) streamlining agent is utilized to advance the slope drop. The most well-known advancement calculation utilized today for preparing profound neural organizations is Adam to processes versatile learning rates. As it is clear to execute, computationally exceptionally productive, and is extremely powerful in managing huge information and boundaries. Every boundary and utilizations squared angles to scale the learning rate; it likewise utilizes the moving normal of the inclination.

4.4 Activation function

Experiments are conducted by using two different activation functions: Softmax and Sigmoid are used the SunfNet model. Especially, in the output layer of the model, the sigmoid activation functions.

1) Epochs

It is the number of times the whole training data is shown to the network while training. It is the number of iterations the entire training dataset passes forwarded and back warded through the model or the network. This model was trained by using different epochs starting from 10 to 50. This was done to get the best optimal epoch.

2) Batch size

It is the total number of training input passed into training. The single set is divided into small subsets called mini-batches to update the parameters. Though, to prevent any loss of accuracy and keep the generality of the model, the model is trained in the whole dataset.

3) Learning rate

Back-propagation was used to train the SunfNet model; the learning rate is used during weight updates. It controls the amount of weight to update during back-propagation. The challenging part was to choose the proper learning rate during the experiment. The learning rates with a value of too small take longer to train than a value of larger. But when given a smaller value the model is more optimal than a model with a learning rate of larger. The experiment was done by using a learning rate of 0.01, 0.001, and 0.0001. This was done to pick the best optimal learning rate.

Table 2: Summary of hyper-parameters used during model training

Parameter	***Value***
Activation Function	Softmax, Sigmoid
Optimization algorithm	Adam, MS
Loss Function	BCE, CCE
Epoch	10-50

Bach size	32-64
Learning rate	0.0001,0.001,0.01 and 0.1

4.5 Hyper-parameters for SunfNet Model

All the convolutional neural network models were employed using various parameters. These parameters make so many possible changes in the architecture. Training them with the huge dataset needed quite a long time. Performing hyperparameter optimization is used to makes efficient computing with available resources. Thus, we utilized some common hyperparameter values and looked further into techniques to achieve a better model evaluation. The following are hyper-parameters that are chosen for the SunfNet model;

- ✓ **Deep learning architecture:**
  SunfNet
- ✓ **Training mechanism**
  Training from Scratch
- ✓ **Training testing set distribution**
  Train: 80%, Test: 10%, valid: 10%

  Train: 70%, Test: 15%, valid 15%

  Train: 70% Test: 30%

  Train: 80% Test: 20%
- ✓ **Data set type**
  Grayscale image

  RGB Augmented image

5 CHAPTER FIVE: IMPLEMENTATION

5.1 INTRODUCTION

This topic describes how the entire implementation was done. In this chapter, all the process passed to get the research result is described in detail. It contains data preparation, data augmentation, data split, and image processing, and building the model.

5.2 Data preparation

Since the images had different proportions, their size was modified to a 150x150 pixel proportion. This resolution was the same satisfactory for both maintaining the image features, as well as keeping computational time to a minimum. All images were originally saved in an RGB 3-channel. To achieve the efficient model as required the first thing that must be done is sort outing the data set. Before train the model it is important to well a proper sufficient data set. Originally the collected images from the fields are not enough to use for deep learning. So, the first thing done to increase the data set size is using a Data Augmentation. This is used to enhance the size of the original data. For this study, the collected original data is a total of 400 images. This data is augmented to around 9806 images by applying 10 (ten) feature factors.

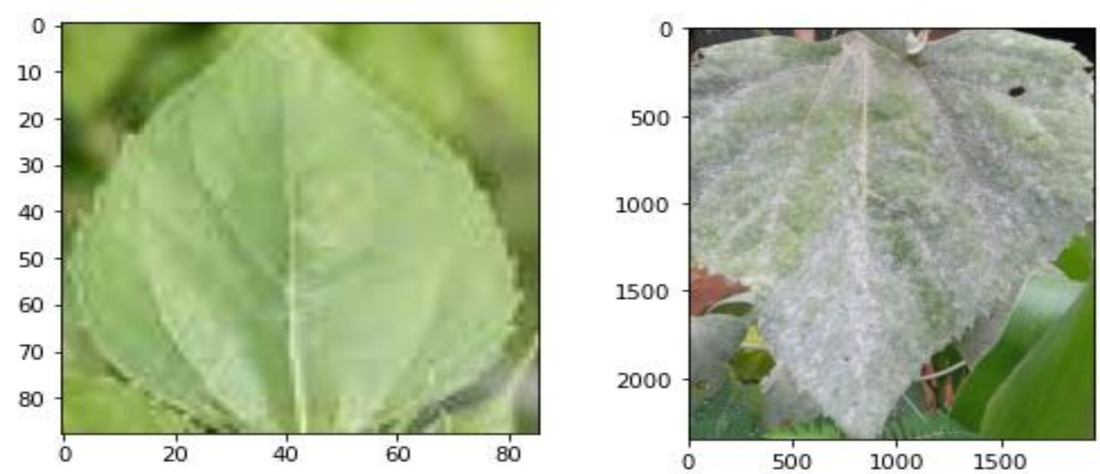


Fig 11: Sample Original Image before Augmentation

These original data images are multiplied using different features through augmentation processes. Sample of augmented images are the following:

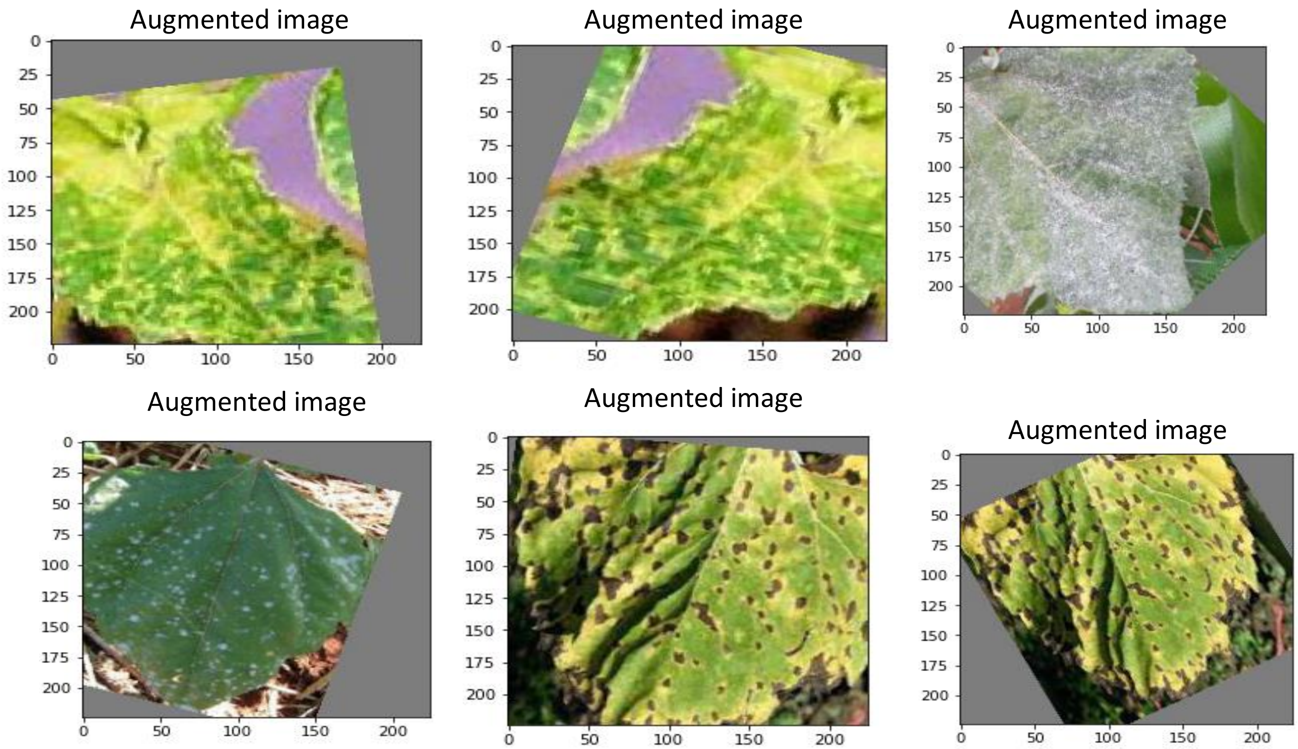


Fig 12: Sample Augmented images

- **Sample code of data augmentation**

```
datagen = ImageDataGenerator(
        rotation_range=40,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        rescale=1./255,
   data_format='channels_last',
    brightness_range=[0.5,0.5],
        fill_mode='nearest')
```

5.3 Image Preprocessing

All pictures in the preparation information, they have an inadequate nature with various measurement size, hence it needs to standardize to a similar picture measurement, which is making the preparation information normalized to a similar picture measurement size. Subsequently, I have utilized an element of 150X150 for all pictures in the preparation set. Pictures are stacked recursively dependent on their pathname. Contingent upon the prefix of their filename, a name worth '0' or '1' is affixed to each stacked picture. Shading change: Two capacities for shading transformation were carried out in this proposal: a transformation from the BGR to RGB shading space, where the request for the picture's shading space, where the pictures lose all the shading data and keep up with just the splendor and immersion for every pixel.

Information standardization is accomplished by discovering the distinction of the mean from each pixel, after that isolating the result by the standard deviation. The dataset values are standardized inside the (0, 1) territory to guarantee that the misfortune work, which is typically not curved, tracks down the worldwide least as effectively as could be expected. Diminishing the scope of contribution for preparing values additionally helps the union of the back-proliferation calculation. Dataset rearranging: The dataset is being rearranged by python's haphazardly Shuffle strategy, which is a request rearranging calculation dependent on an arbitrary number generator. After the application, the request for the pictures is initially was in sequential order, presently blended all through the dataset.

Dataset parting: the dataset is being parted in the accompanying design: First 15% of the all-out number of the dataset is held out and set as a testing dataset, second 15% of the dataset is held out and set the approval set, then, at that point, the leftover dataset is parting again the 80/20 style, where the 20% piece is utilized as the approval dataset. The rest is utilized for preparing the dataset calculation. The preparation and approval sets are utilized for the preparation of the calculation, while the testing dataset is utilized solely after the classifier is prepared, to make forecasts. Coming up next are test preprocessed images.

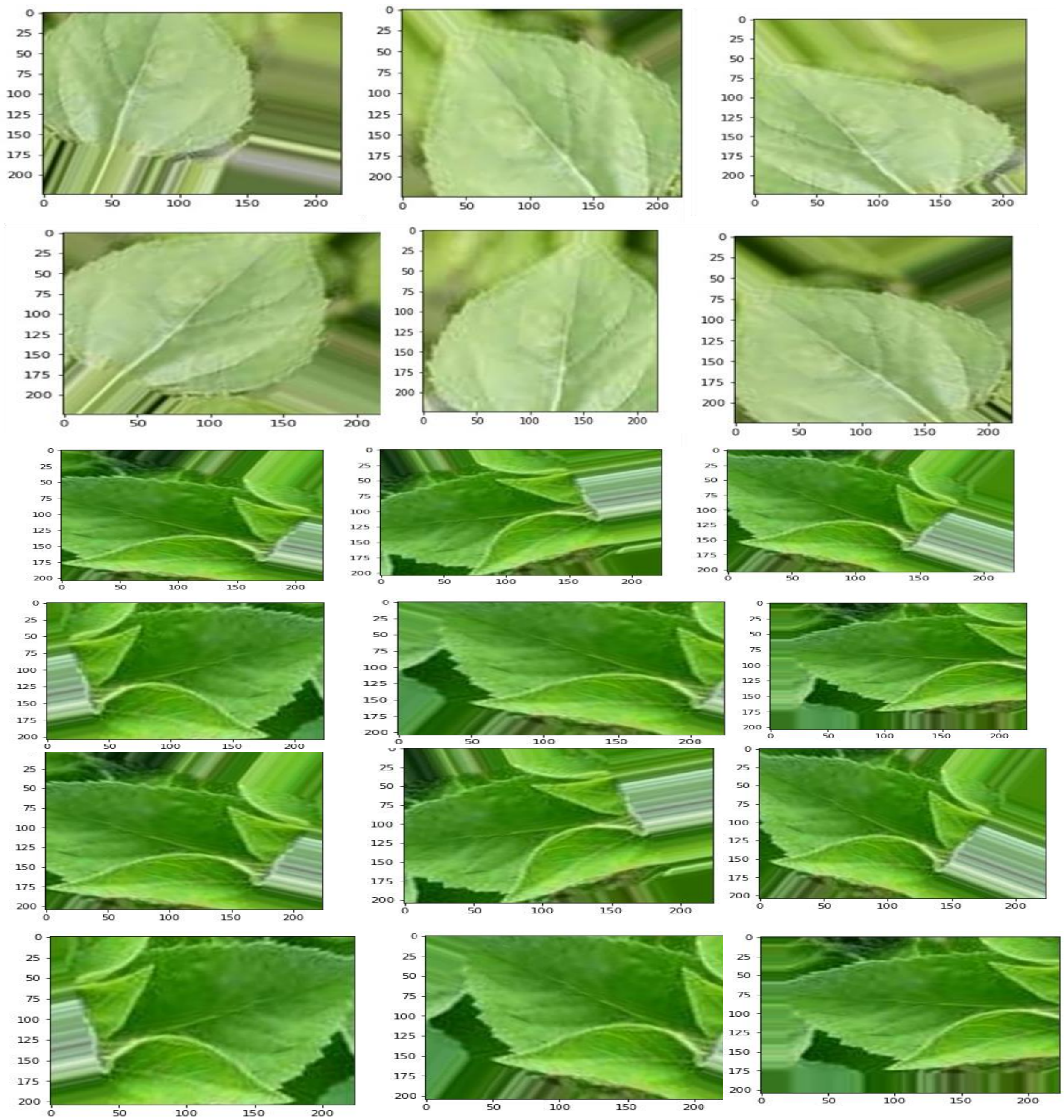


Fig 13: Sample preprocessed images

```
In [5]: import os, shutil

        # Path to the directory where the original dataset is
        original_dataset_dir ='C:/Users/Administrator/Desktop/Research/aug/'

        # list of all plant diseases classes in the original dataset
        classes_list = os.listdir(original_dataset_dir)

        #Directory where you'll store your dataset splits
        base_dir = './dataset_splits'
        os.mkdir(base_dir)

        #Directories for the training, validation, and test splits
        train_dir = os.path.join(base_dir, 'train')
        os.mkdir(train_dir)
        validation_dir = os.path.join(base_dir, 'validation')
        os.mkdir(validation_dir)
        test_dir = os.path.join(base_dir, 'test')
        os.mkdir(test_dir)


        for cls in classes_list:
            #Directory for training pictures
            os.mkdir(os.path.join(train_dir, cls))

            #Directory for validation pictures
            os.mkdir(os.path.join(validation_dir, cls))

            # Directory for test pictures
            os.mkdir(os.path.join(test_dir, cls))
```

Fig 14: Sample code of data splitting

5.4 Build SunfNet Model

5.4.1 Parameters for layers

Convolutional layer filter size filters the number of filters should depend on the complexity of the dataset and the depth of the neural network. A common setting to start with is [150, 64, 32, 16, 8, and 4] for nine layers. Kernel size = number of filters =a small window of pixels at a time (3x3) which will be moved until the enter image is scanned. If the image is smaller than 150x150, work with a smaller filter of 1x1; the Width and Height of images were already providing. 2D convolutional layers take a three-dimensional input, typically an image with three color channels such as RGB; Max pooling is used which further reduces the size of the convolved image and prevents overfitting. It is the application of a moving window across a 2D input space, where the maximum value within that window is the output: 2x2.

5.4.2 Parameters to fit the model

Epoch is the number of passes of the entire training data set when an entire dataset is passed forward and backward through the neural network. It describes the number of images (items) and the algorithm sees the entire data set and the algorithm has seen all samples in the training dataset. The number of epochs used is 50. This number of epochs is enough scope for the loss function to converge if it is possible. This also means that we are giving enough time to the architecture to learn. It takes around 2 minutes per epoch which is 2*50=100 minutes. A single number of epochs are too big to feed to the computer at once so divide it into several smaller batches. In the experiment, it tried in different batches to come up with the best learning.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 150, 150, 4)       112
_________________________________________________________________
norm_1 (BatchNormalization)  (None, 150, 150, 4)       16
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 75, 75, 4)         0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 75, 75, 8)         296
_________________________________________________________________
norm_2 (BatchNormalization)  (None, 75, 75, 8)         32
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 37, 37, 8)         0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 37, 37, 16)        1168
_________________________________________________________________
norm_3 (BatchNormalization)  (None, 37, 37, 16)        64
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 18, 18, 16)        0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 18, 18, 32)        4640
_________________________________________________________________
norm_4 (BatchNormalization)  (None, 18, 18, 32)        128
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 9, 9, 32)          0
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 9, 9, 32)          25632
_________________________________________________________________
norm_5 (BatchNormalization)  (None, 9, 9, 32)          128
_________________________________________________________________
max_pooling2d_22 (MaxPooling (None, 4, 4, 32)          0
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 4, 4, 64)          18496
_________________________________________________________________
norm_6 (BatchNormalization)  (None, 4, 4, 64)          256
_________________________________________________________________
max_pooling2d_23 (MaxPooling (None, 2, 2, 64)          0
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 2, 2, 64)          102464
_________________________________________________________________
norm_7 (BatchNormalization)  (None, 2, 2, 64)          256
_________________________________________________________________
max_pooling2d_24 (MaxPooling (None, 1, 1, 64)          0
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 1, 1, 150)         86550
_________________________________________________________________
norm_8 (BatchNormalization)  (None, 1, 1, 150)         600
_________________________________________________________________
max_pooling2d_25 (MaxPooling (None, 1, 1, 150)         0
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 1, 1, 150)         22650
_________________________________________________________________
norm_9 (BatchNormalization)  (None, 1, 1, 150)         600
_________________________________________________________________
max_pooling2d_26 (MaxPooling (None, 1, 1, 150)         0
_________________________________________________________________
global_average_pooling2d_2 ( (None, 150)               0
_________________________________________________________________
activation_4 (Activation)    (None, 150)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 150)               0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 755
_________________________________________________________________
activation_5 (Activation)    (None, 5)                 0
=================================================================
Total params: 264,843
Trainable params: 263,803
Non-trainable params: 1,040
```

Fig 15: Sample code of SunfNet model

SunfNet or CNN (Convolutional neural network or ConvNet) is a class of deep neural networks, commonly applied to analyzing visual imagery. Here is the SunfNet model's example of CNN with few layers using Conv2D -2D convolutional layer, Batch Normalization, and MaxPooling2D application of a moving window across 2D input space. Hyperparameters are set before training they represent the variables that determine the neural network structure and how it is trained. Conv2D with hyperparameters mentioned above: Conv2D (kernel size, (filters, and filters), input shape = (150, 150, and 3) with activation function for each layer as a ReLU.

- MaxPooling2D layer to reduce the spatial size of the incoming features;
- 2D input space: MaxPooling2D ((2, 2)) and ((2, 2))
- Do the same increasing the kernel size: 4-8-16-32-64-150
- Provide the last fully connected layer which specifies the number of classes of sunflower leaf diseases.
- Print the summary of the model.

6 CHAPTER SIX: RESULT AND DISCUSSION

6.1 Experiment Result

Afterward performing all the evaluation on the Grayscale and RGB segmented dataset on the SunfNet model, experiments with top results are discussed below. The SunfNet model is evaluated using different parameters that can affect the efficiency of the model. These parameters are several epochs; train test split ratio, learning rate, and dropout ratio which are explained concisely in previous chapters. Different parameter ratios are used for the evaluation purpose and the most successful ratios are discussed in this chapter.

In this thesis, three questions stated in the problem statement part are tried to be answered. To select the best model to solve the stated problem, various experiments are conducted using the CNN method. Using this method, the SunfNet model is build using 6868 (six thousand eight hundred sixty-eight) training samples from 9806 (Nine thousand eight hundred six) total data set. This is 70% of the total dataset. Model fit is implemented using 1470(one thousand for hundred seventy) validation (testing) dataset from 9806 (Nine thousand eight hundred six) total dataset. This is 15% of the total dataset. In another experiment, the data was divided to 80% for training samples from the total dataset, 20% for test and validation data samples.

To come up with the best model the third try of the experiment was dividing the dataset into 60% for training samples and 40 % testing and validation data samples. The lack of a globally acknowledged standard ratio for separating the training and test data sets is the reason behind this. The SunfNet model was built using RGB image training datasets and grayscale image datasets. Performances of both models are evaluated against validation datasets. The different scenarios are used in an experiment in addition to dividing datasets into training and testing sub-samples. The result obtained from the experiment done using RGB images is discussed as follows in different perspectives. The experiment was done in different data splits as mentioned above. These are 60-40, 70-30, and 80-20 the experiments are done accordingly. Using this different data partition, the result obtained is also different as listed in table 4.

Table 4: Experiment result using learning rate

No	*Training samples*	*Test and validation samples*	*Accuracy*	*Loss*	*Epochs*
1	60%	40%	99.04%	0.34	50
2	70%	30%	99.86%	0.011	50
3	80%	20%	99.88%	0.0013	50

6.2 Result of Experiment done on RGB data images

Multi-channel images are displayed using the colors in RGB images (24-bit with 8-bits for each red, green, and blue channel). Colors have been chosen to reflect natural hues (i.e., the green in an RGB image reflects the green color in the specimen). As stated previously, this thesis is based on RGB and greyscale data images. The experiment is carried out utilizing various circumstances to improve the SunfNet model's performance. The results of the experiments are presented in the tables below.

6.3 Experiment result using output Activation Function

The Sigmoid Activation Function is a mathematical function that has a distinct "S" curve. It is mostly used for logistic regression and the creation of simple neural networks. This function is used separately for each element's raw output in this thesis experiment. Softmax is a technique for mapping non-normalized data output to a probability distribution for output classes. In neural network-based classifiers, it is used in the final layers. They are trained under either the log-loss or cross-entropy regime. Softmax is often used in artificial and convolutional neural networks. These activation functions are used in the SunfNet model. The following table 5 shows the results of the model experiment using these functions:

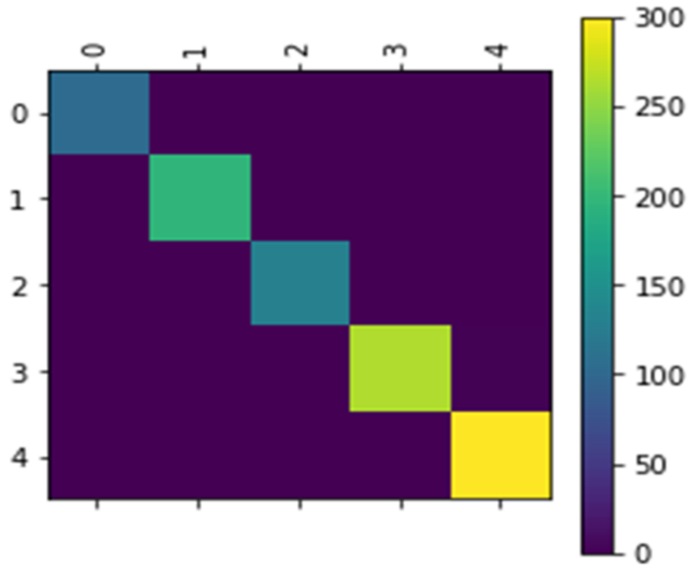
6.4 Experiment result using various Learning Rates

The learning rate is a hyper-parameter that governs how much the weights of our network are adjusted about the loss gradient. A value too little may result in a long training process that becomes stuck, whereas a value too large may result in learning a sub-optimal set of weights too quickly or an unstable training process. The stochastic gradient descent algorithm estimates the error gradient for the current state of the model using examples from the training dataset, then updates the weights of the model using the back-propagation of errors process, sometimes known as simply back-propagation. The step size, often known as the "learning rate," is the amount by which the weights are changed during training. The learning rate is an adjustable hyperparameter that has a modest positive value, usually between 0.0 and 1.0, and is used in the training of neural networks. This experiment was done by using a learning rate of 0.1, 0.01, 0.001, and 0.0001.

6.5 Experiment result using various Learning Rates and Softmax Activation on RGB images

Table 5: Experiment Result using Softmax on RGB color Image

No	*Activation function*	*Data split Ratio*	*Training accuracy*	*Validation Accuracy*	*Epochs*	*Optimizer*	*LR*
1	**Softmax**	8:2	99.80%	99.79%	50	ADAM	0.0001
2		8:2	99.51%	99.37%	50	ADAM	0.001
3		8:2	99.72%	99.79%	50	ADAM	0.01
4		8:2	99.85%	99.99%	50	ADAM	0.1




```
Confusion Matrix
[[105   0   0   0   0]
 [  0 196   0   0   0]
 [  0   0 131   0   0]
 [  0   0   0 264   2]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      1.00      1.00       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       131
      Mildew       1.00      0.99      1.00       266
        Rust       0.99      1.00      1.00       300

    accuracy                           1.00       998
   macro avg       1.00      1.00      1.00       998
weighted avg       1.00      1.00      1.00       998
```

Fig 16: Confusion Matrix for result 1 (Softmax).

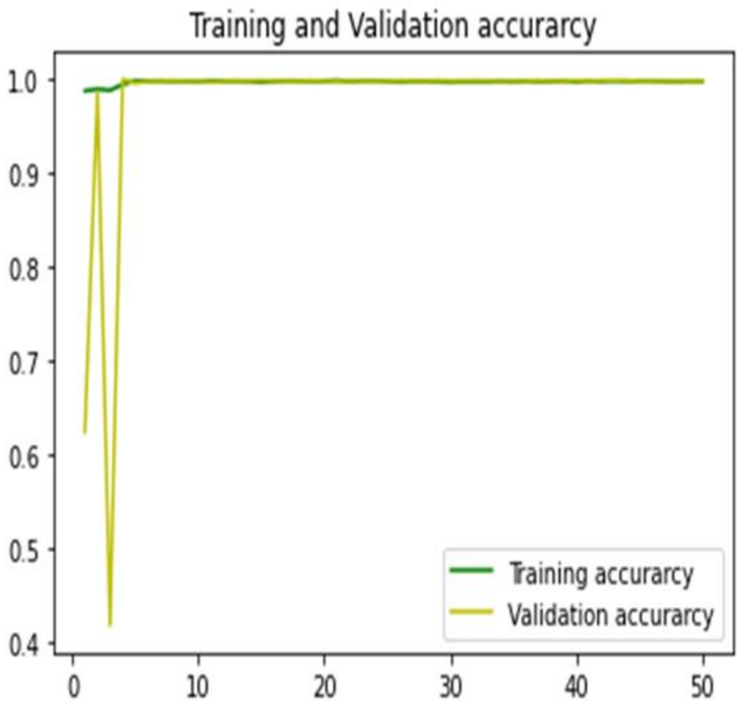
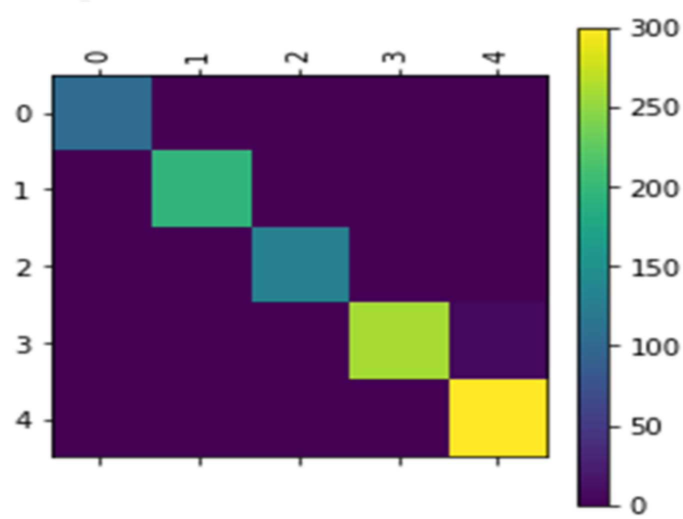


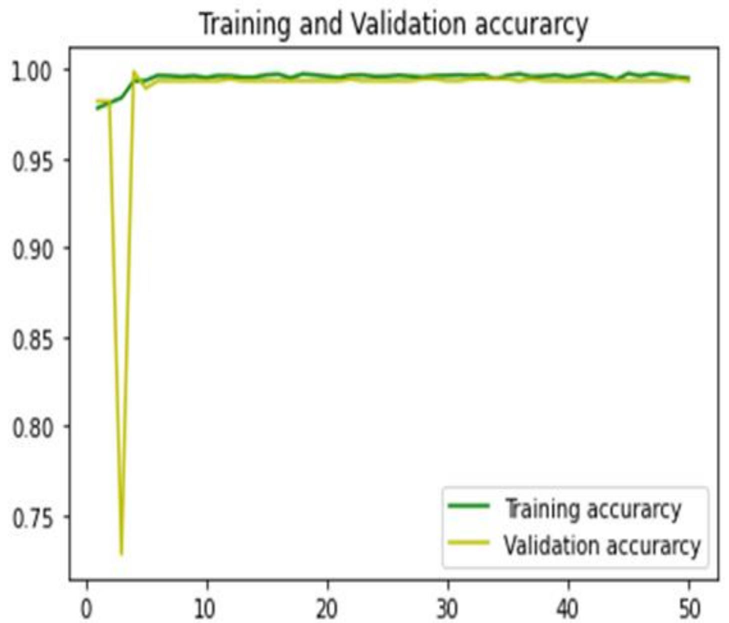
Fig 17: Training and Validation Accuracy Analysis

Fig 18: Training and Validation Loss Analysis




```
Confusion Matrix
[[105   0   0   0   0]
 [  0 196   0   0   0]
 [  0   0 131   0   0]
 [  0   0   0 260   6]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      1.00      1.00       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       131
      Mildew       1.00      0.98      0.99       266
        Rust       0.98      1.00      0.99       300

    accuracy                           0.99       998
   macro avg       1.00      1.00      1.00       998
weighted avg       0.99      0.99      0.99       998
```

Fig 19: Confusion Matrix for result 2(Softmax).



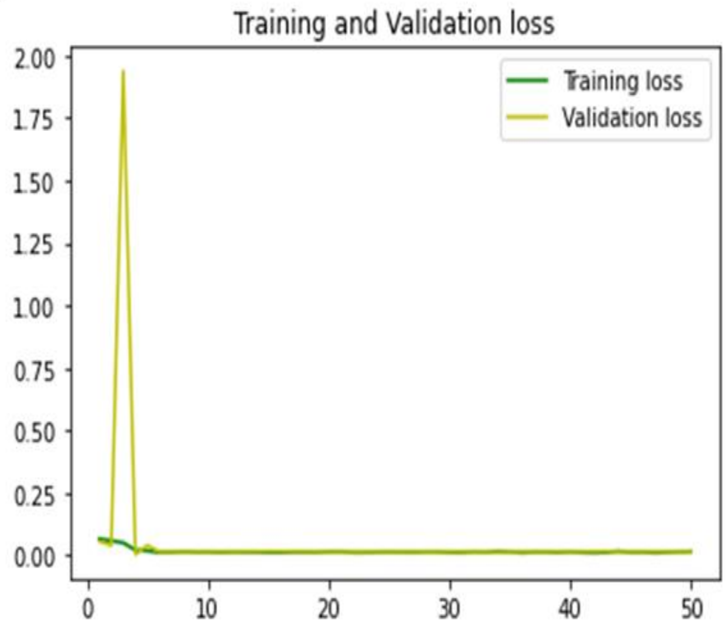
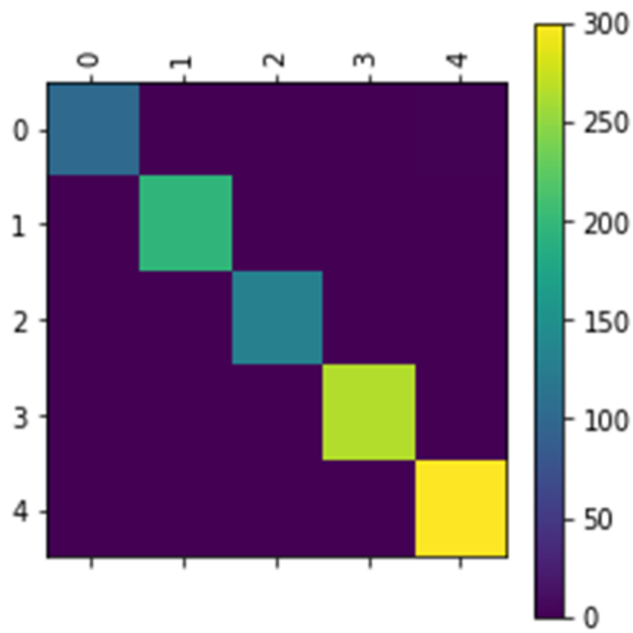


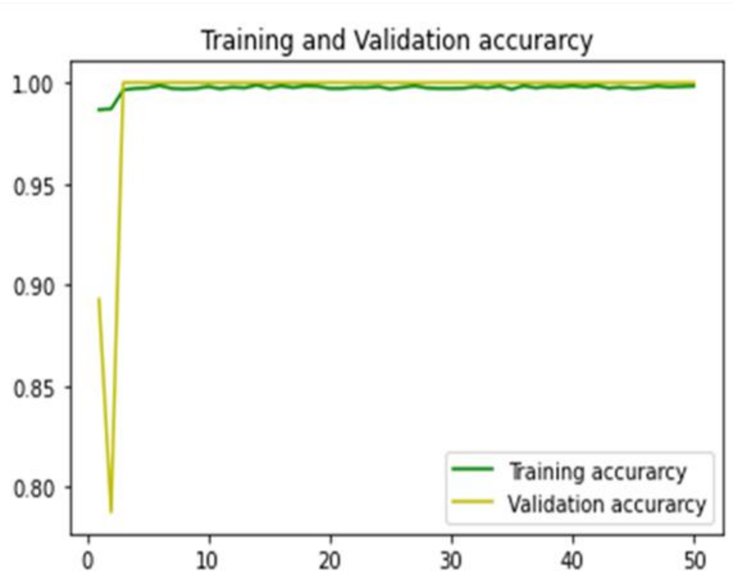
Fig 20: Training and Validation accuracy Analysis

Fig 21: Training and Validation Loss Analysis



```
Confusion Matrix
[[103   0   0   0   2]
 [  0 196   0   0   0]
 [  0   0 131   0   0]
 [  0   0   0 266   0]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      0.98      0.99       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       131
      Mildew       1.00      1.00      1.00       266
        Rust       0.99      1.00      1.00       300

    accuracy                           1.00       998
   macro avg       1.00      1.00      1.00       998
weighted avg       1.00      1.00      1.00       998
```

Fig 22: Confusion matrix for result 3 (Softmax).



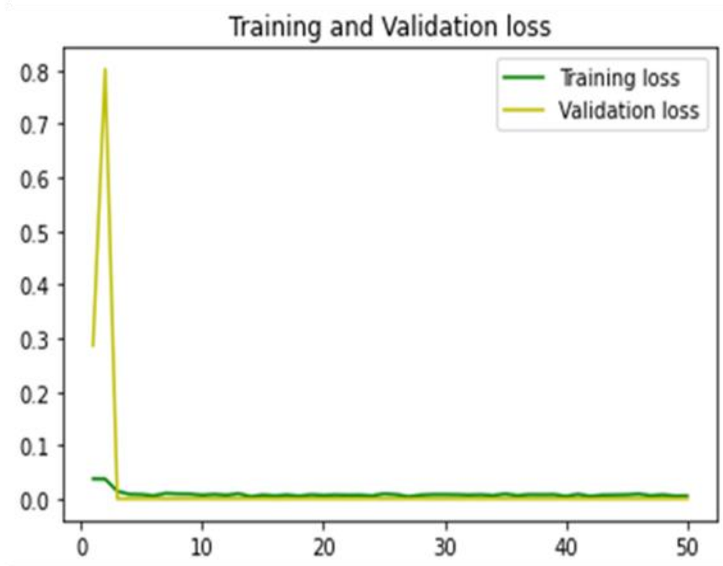
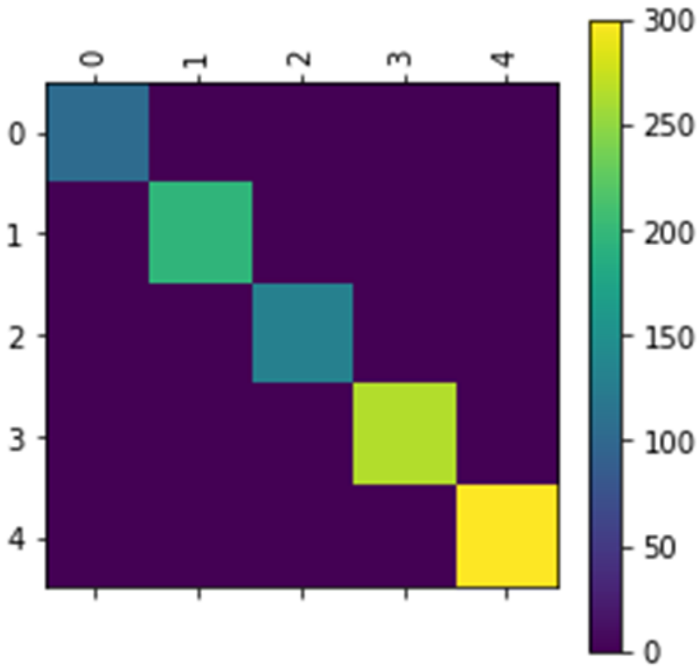
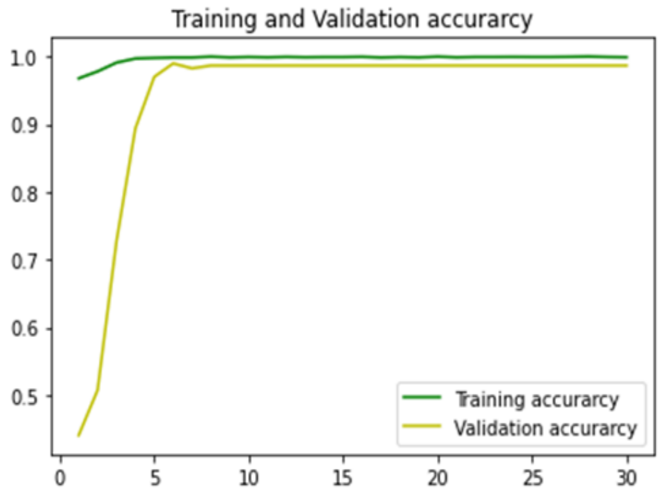


Fig 23: Training and Validation Accuracy Analysis Fig 24: Training and Validation Loss analysis




```
Confusion Matrix
[[105   0   0   0   0]
 [  0 196   0   0   0]
 [  0   0 131   0   0]
 [  0   0   0 266   0]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      1.00      1.00       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       131
      Mildew       1.00      1.00      1.00       266
        Rust       1.00      1.00      1.00       300

    accuracy                           1.00       998
   macro avg       1.00      1.00      1.00       998
weighted avg       1.00      1.00      1.00       998
```

Fig 25: Confusion Matrix for result 4 (Softmax).



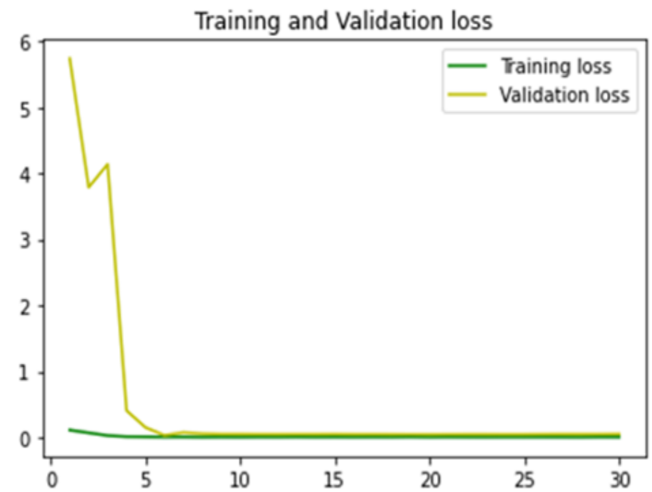
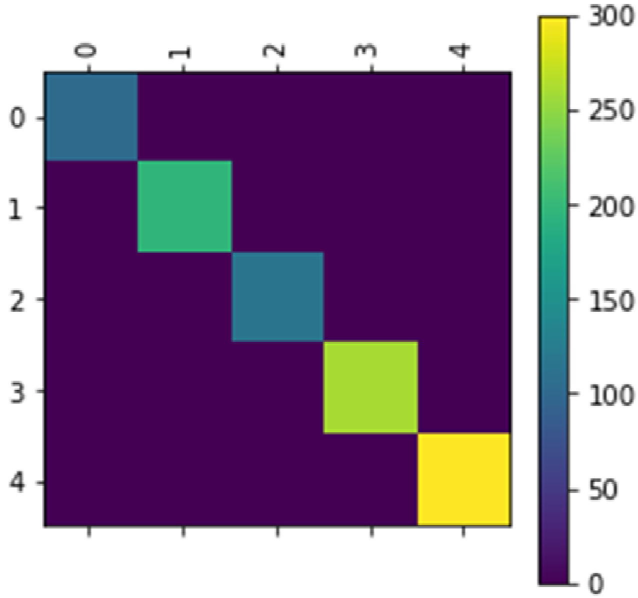


Fig 26: Training and validation accuracy Analysis **Fig 27: Training and validation loss Analysis**

6.6 Experiment result using various Learning Rates and Sigmoid Activation on RGB images

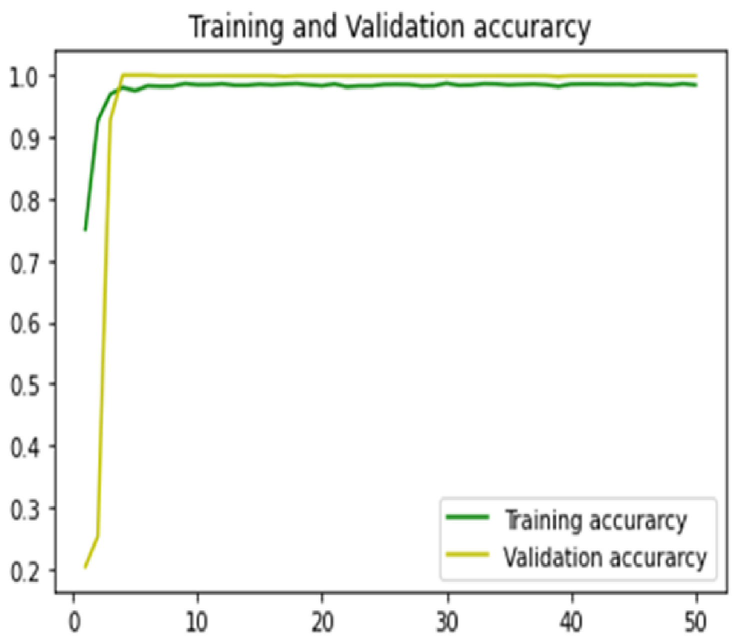
Table 6: Result obtain using sigmoid function on RGB color image

No	*Activation function*	*Data split Ratio*	*Training accuracy*	*Validation Accuracy*	*Epochs*	*Optimizer*	*LR*
1	**Sigmoid**	8:2	99.44%	99.64%	50	ADAM	0.0001
2		8:2	99.25%	99.85%	50	ADAM	0.001
3		8:2	98.79%	99.44%	50	ADAM	0.01
4		8:2	99.12%	99.22%	50	ADAM	0.1




```
Confusion Matrix
[[104   0   0   0   1]
 [  0 196   0   0   0]
 [  0   0 116   0   0]
 [  0   0   0 261   0]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      0.99      1.00       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       116
      Mildew       1.00      1.00      1.00       261
        Rust       1.00      1.00      1.00       300

    accuracy                           1.00       978
   macro avg       1.00      1.00      1.00       978
weighted avg       1.00      1.00      1.00       978
```

Fig 28: Confusion Matrix for result 1(sigmoid).



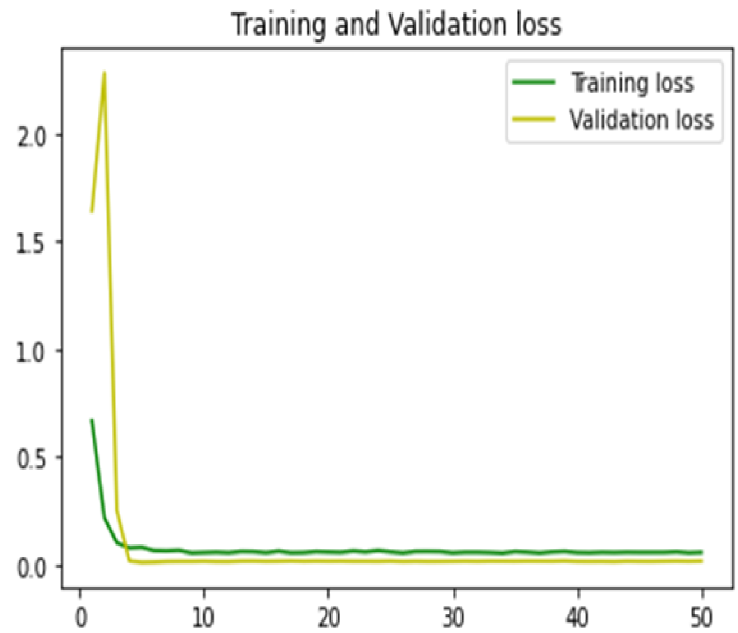
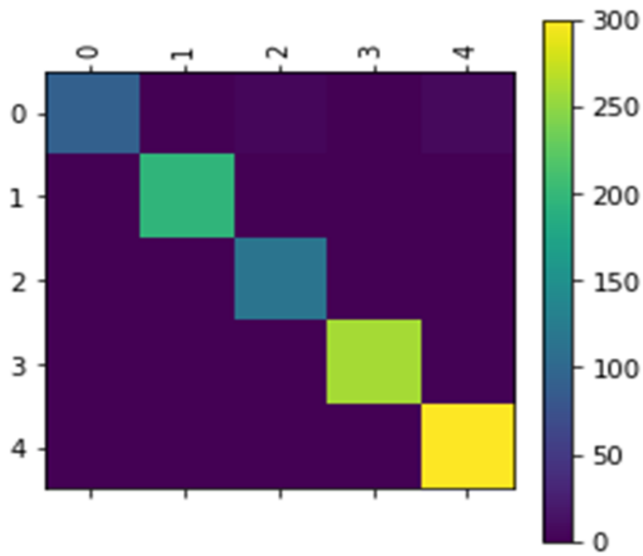
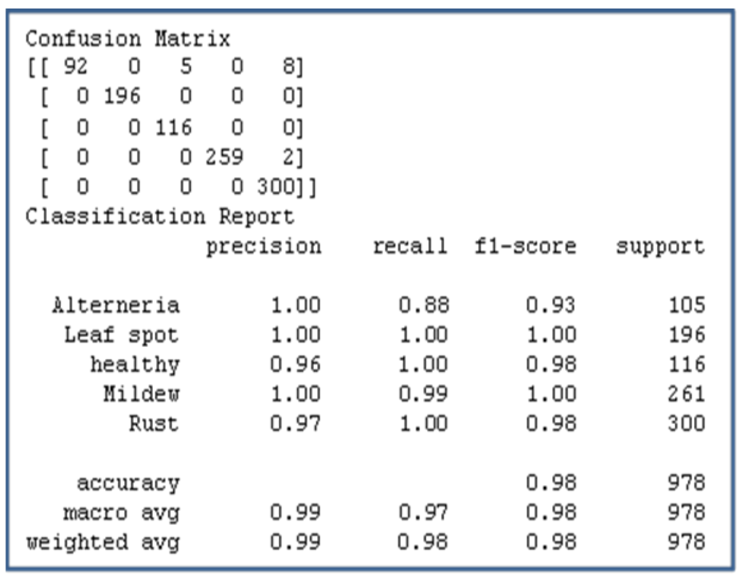


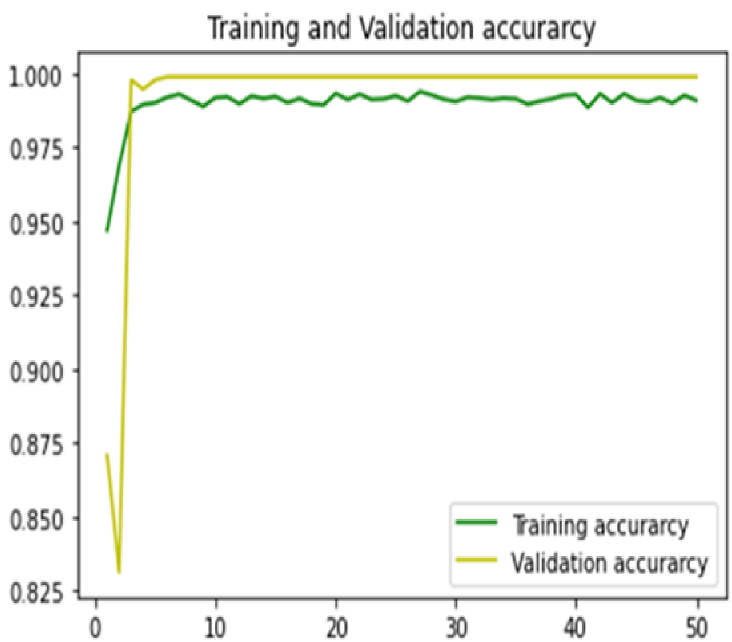
Fig 29: Training and validation accuracy analysis **Fig 30: Training and validation Loss analysis**

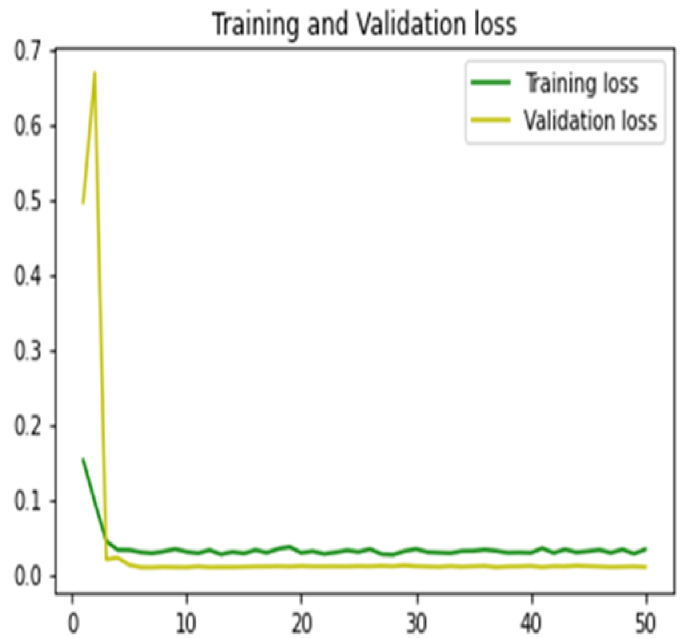




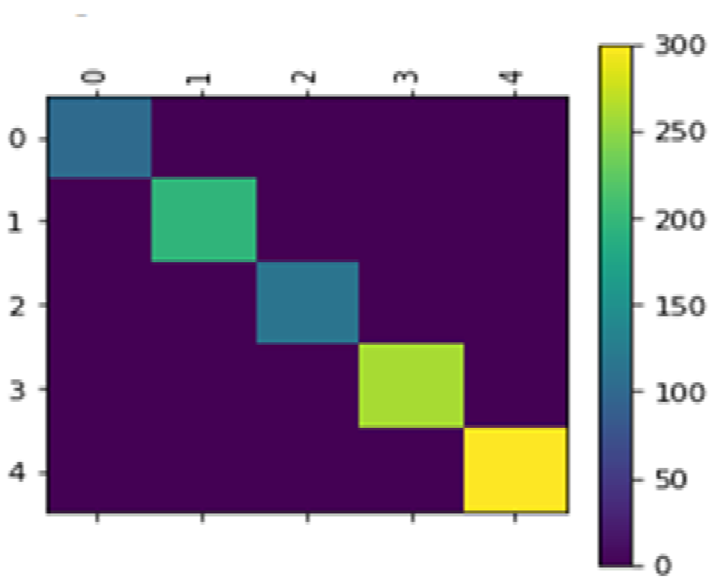
```
Confusion Matrix
[[ 92   0   5   0   8]
 [  0 196   0   0   0]
 [  0   0 116   0   0]
 [  0   0   0 259   2]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      0.88      0.93       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       0.96      1.00      0.98       116
      Mildew       1.00      0.99      1.00       261
        Rust       0.97      1.00      0.98       300

    accuracy                           0.98       978
   macro avg       0.99      0.97      0.98       978
weighted avg       0.99      0.98      0.98       978
```

Fig 31: Confusion Matrix for result 2 (Sigmoid).



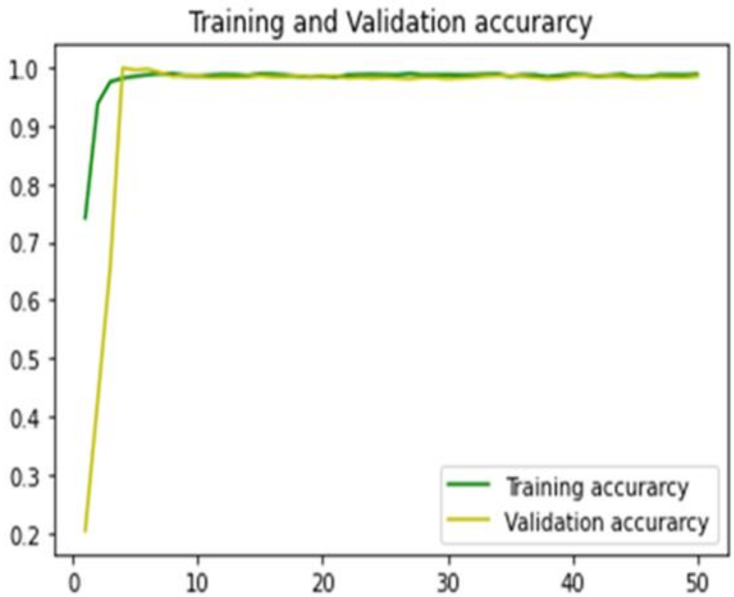


2.1. **Fig 32: Training and validation accuracy analysis** **Fig 33: Training and validation Loss analysis**



```
Confusion Matrix
[[104   0   0   0   1]
 [  0 196   0   0   0]
 [  0   0 116   0   0]
 [  0   0   0 261   0]
 [  0   0   0   0 300]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       1.00      0.99      1.00       105
   Leaf spot       1.00      1.00      1.00       196
     healthy       1.00      1.00      1.00       116
      Mildew       1.00      1.00      1.00       261
        Rust       1.00      1.00      1.00       300

    accuracy                           1.00       978
   macro avg       1.00      1.00      1.00       978
weighted avg       1.00      1.00      1.00       978
```

Fig 34: Confusion Matrix for result 3(sigmoid).



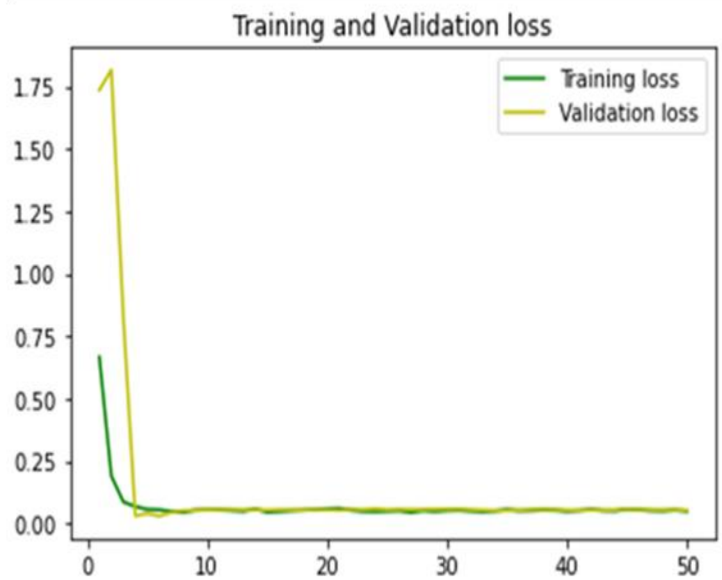


Fig 35: Training and validation accuracy analysis **Fig 36: Training and validation Loss analysis**

6.7 Result of Experiment done on Grey-scale data images

6.8 Experiment result using output Activation Function

The Sigmoid Activation Function is a mathematical function that has a distinct "S" curve. It is mostly used for logistic regression and the creation of simple neural networks. This function is used separately for each element's raw output in this thesis experiment. Softmax is a technique for mapping non-normalized data output to a probability distribution for output classes. In neural network-based classifiers, it is used in the final layers. They are trained under either the log-loss or cross-entropy regime. Softmax is often used in an artificial and convolutional neural network. These activation functions are used in the SunfNet model.

6.9 Experiment result using various Learning Rates and Softmax Activation on Grayscale images

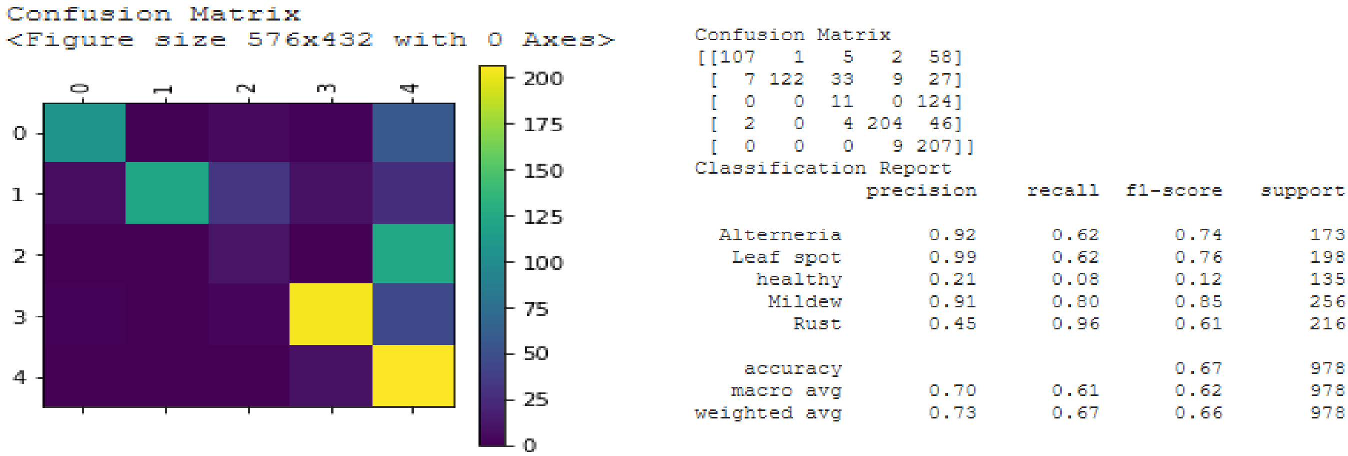
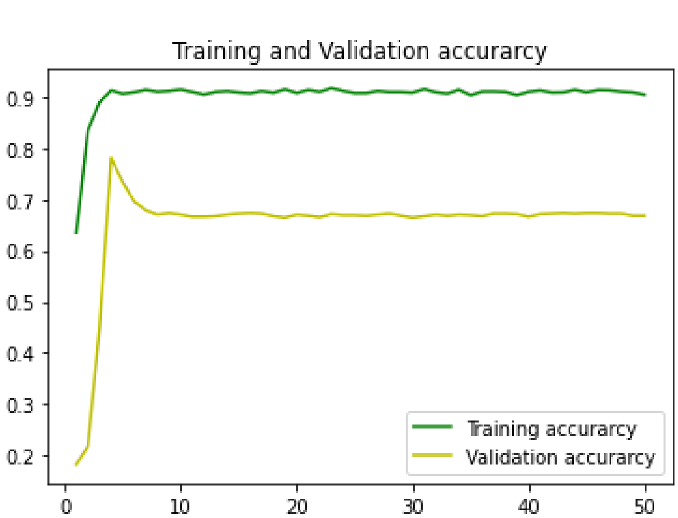


Fig 37: Confusion Matrix for result 1 (Softmax on the grayscale)



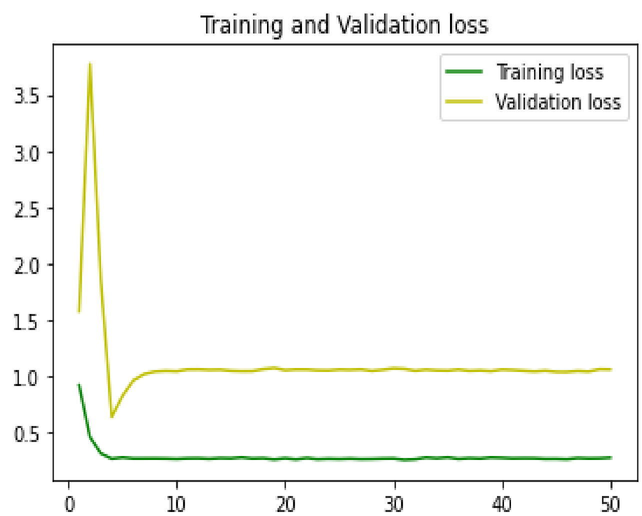
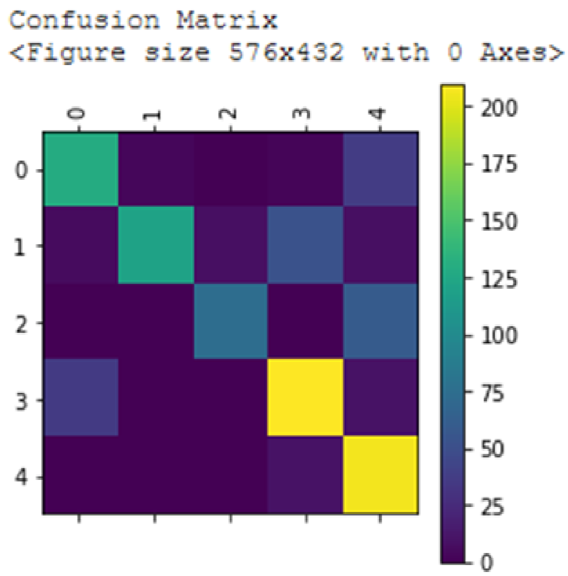


Fig 38: Training and validation accuracy Analysis **Fig 39: Training and validation loss Analysis**

Table 7: Experiment Result using Softmax on Grayscale color Image

No	*Activation function*	*Data split Ratio*	*Training accuracy*	*Validation Accuracy*	*Epochs*	*Optimizer*	*LR*
1	**Softmax**	8:2	90.51%	66.97%	50	ADAM	0.0001
2		8:2	96.97%	84.27%	50	ADAM	0.001
3		8:2	97.16%	76.35%	50	ADAM	0.01
4		8:2	98.34%	78.75%	50	ADAM	0.1




```
Confusion Matrix
[[129   4   0   3  37]
 [  6 121   9  53   9]
 [  0   0  75   0  60]
 [ 36   0   0 210  10]
 [  0   0   0  10 206]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       0.75      0.75      0.75       173
   Leaf spot       0.97      0.61      0.75       198
     healthy       0.89      0.56      0.68       135
      Mildew       0.76      0.82      0.79       256
        Rust       0.64      0.95      0.77       216
```

Fig 40: Confusion Matrix for result 2(Softmax on Grayscale)

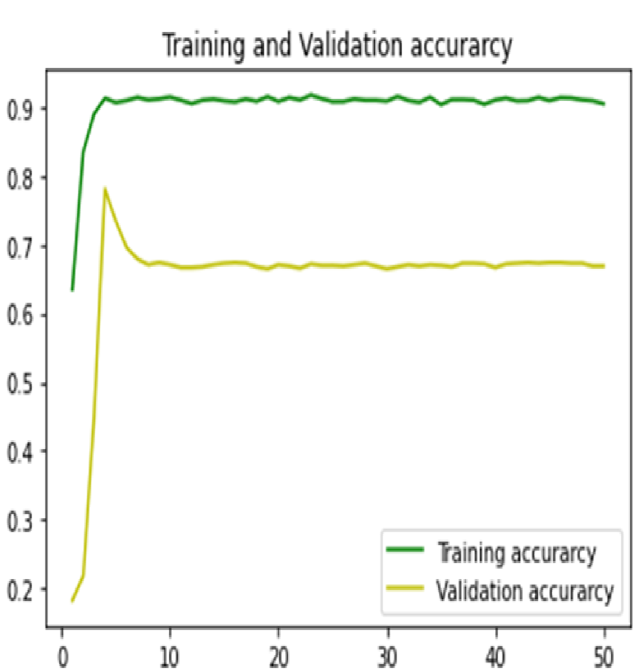


Fig 41: Training and validation accuracy Analysis

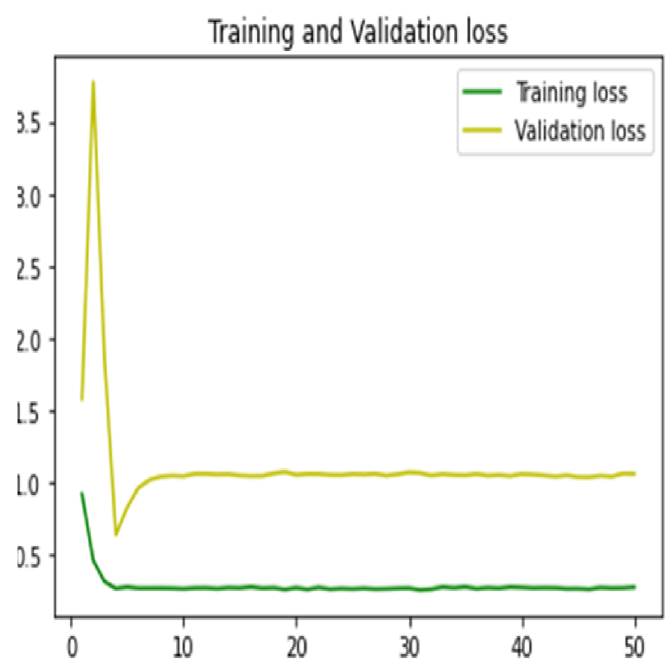
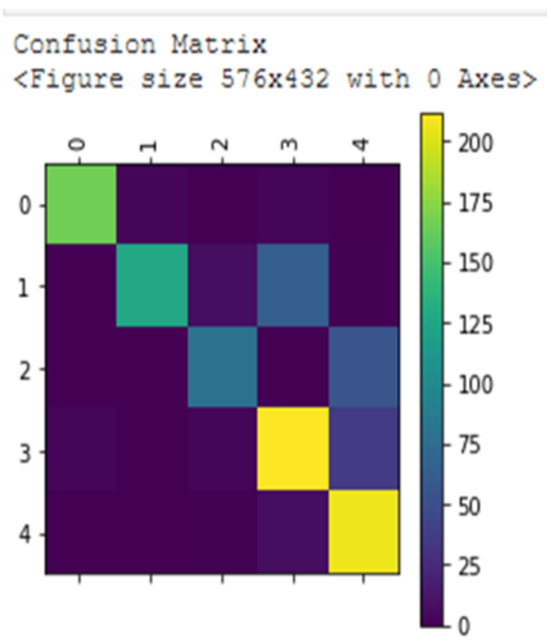


Fig 42: Training and validation loss Analysis

- **Experiment result using various Learning Rates and Sigmoid Activation on Gray scale images**

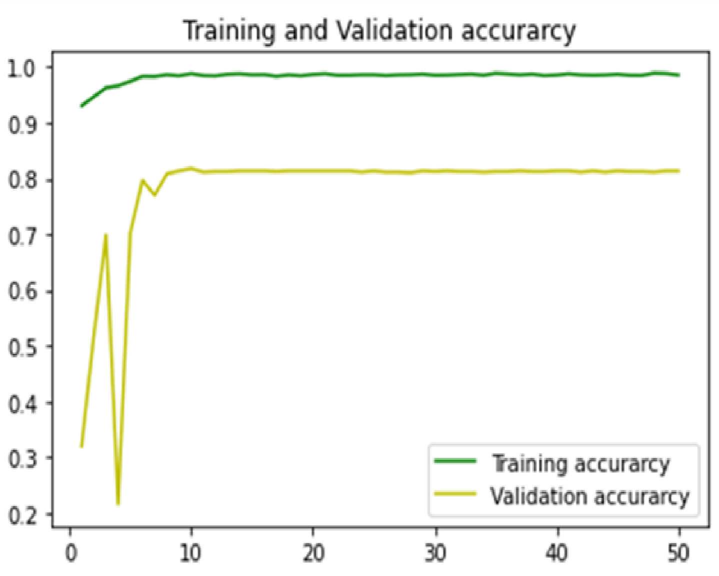
Table 8: Experiment Result using Sigmoid on Grayscale color Image

No	*Activation function*	*Data split Ratio*	*Training accuracy*	*Validation Accuracy*	*Epochs*	*Optimizer*	*LR*
1	**Sigmoid**	8:2	92.83%	75.10%	50	ADAM	0.0001
2		8:2	96.97%	84.27%	50	ADAM	0.001
3		8:2	98.55%	81.35%	50	ADAM	0.01
4		8:2	98.89%	83.23%	50	ADAM	0.1




```
Confusion Matrix
[[166   4   0   3   0]
 [  0 126   8  63   1]
 [  0   0  79   0  56]
 [  4   0   3 212  37]
 [  0   0   1   8 207]]
Classification Report
              precision    recall  f1-score   support

  Alterneria       0.98      0.96      0.97       173
   Leaf spot       0.97      0.64      0.77       198
     healthy       0.87      0.59      0.70       135
      Mildew       0.74      0.83      0.78       256
        Rust       0.69      0.96      0.80       216

    accuracy                           0.81       978
   macro avg       0.85      0.79      0.80       978
weighted avg       0.83      0.81      0.80       978
```

Fig 43: Confusion Matrix for result 1(sigmoid on Grayscale)



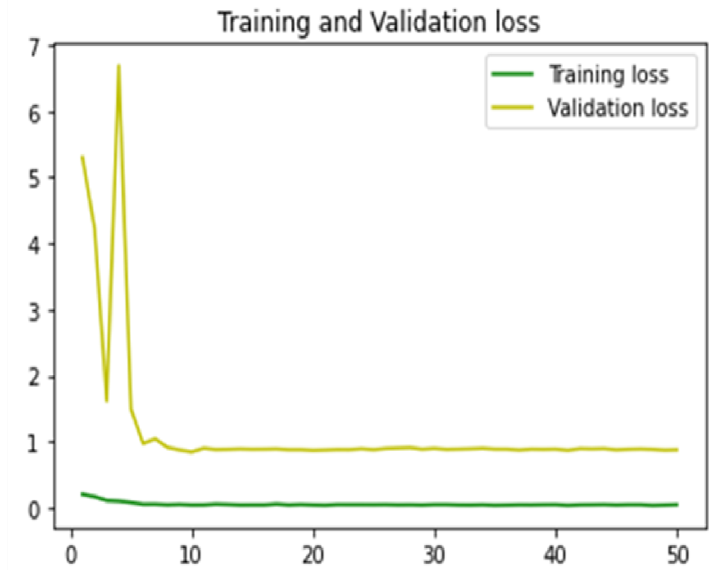


Fig 44: Training and validation accuracy Analysis

Fig 45: Training and validation loss Analysis

Table 9: Evaluation Result on Both RGB and Grayscale

6.10 Evaluation result

No	***Function***	***LR***	***Epoch***	***Data split ratio***	***Optimizer***	***Training accuracy***	***Validation Accuracy***	***Image Type***
1	Softmax	0.0001	50	8:2	ADAM	99.80%	99.79%	RGB
2	Softmax	0.001	50	8:2	ADAM	99.51%	99.37%	RGB
3	Softmax	0.01	50	8:2	ADAM	99.72%	99.79%	RGB
4	Softmax	0.1	50	8:2	ADAM	99.88%	99.99%	RGB
5	Sigmoid	0.0001	50	8:2	ADAM	98.64%	99.85%	RGB
6	Sigmoid	0.001	50	8:2	ADAM	99.15%	99.99%	RGB
7	Sigmoid	0.01	50	8:2	ADAM	98.79%	99.44%	RGB
8	Sigmoid	0.1	50	8:2	ADAM	98.12%	98.22%	RGB
9	Softmax	0.0001	50	8:2	ADAM	90.51%	66.97%	Grayscale
10	Softmax	0.001	50	8:2	ADAM	96.97%	84.27%	Grayscale

11	Softmax	0.01	50	8:2	ADAM	97.16%	76.35%	Grayscale
12	Softmax	0.1	50	8:2	ADAM	98.34%	78.75%	Grayscale
13	Sigmoid	0.0001	50	8:2	ADAM	92.83%	75.10%	Grayscale
14	Sigmoid	0.001	50	8:2	ADAM	96.97%	84.27%	Grayscale
15	Sigmoid	0.01	50	8:2	50	98.55%	81.35%	Grayscale
16	Sigmoid	0.1	50	8:2	50	98.89%	83.23%	Grayscale

In the above table, total evaluation results are listed. These evaluation results are including only the experiment result obtained from the selected model with its learning rate, activation function, image types, epochs, data splits ratio, evaluation metrics, and accuracy. According to this evaluation result, the SunfNet model with 99.88 training accuracy and 99.99 validation accuracy is the best model gained from these experiments. This model is assessed using new testing images as follows:

```python
from keras.preprocessing import image
import numpy as np
image_path = './dataset_splits1/test/healthy/aug_6_415892.jpg'
new_img = image.load_img(image_path, target_size=(150, 150))
img = image.img_to_array(new_img)
img = np.expand_dims(img, axis=0)
img = img/255
```

```python
print("Following is our prediction:")
prediction = model.predict(img)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
d = prediction.flatten()
j = d.max()
for index,item in enumerate(d):
    if item == j:
        class_name = li[index]


plt.figure(figsize = (4,4))
plt.imshow(new_img)
plt.axis('off')
plt.title(class_name)
plt.show()
```

```
Following is our prediction:
```

healthy

Fig 46: Sample truly classified healthy images

```python
from keras.preprocessing import image
import numpy as np
image_path = './dataset_splits1/test/alterneria/aug_0_4557.jpg'
new_img = image.load_img(image_path, target_size=(150, 150))
img = image.img_to_array(new_img)
img = np.expand_dims(img, axis=0)
img = img/255
```

```python
print("Following is our prediction:")
prediction = model.predict(img)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
d = prediction.flatten()
j = d.max()
for index,item in enumerate(d):
    if item == j:
        class_name = li[index]


plt.figure(figsize = (4,4))
plt.imshow(new_img)
plt.axis('off')
plt.title(class_name)
plt.show()
```

Fig 47: Sample truly classified Alternaria infected image

```python
from keras.preprocessing import image
import numpy as np
image_path = './dataset_splits1/test/leafspot/aug_1_1270889.jpg'
new_img = image.load_img(image_path, target_size=(150, 150))
img = image.img_to_array(new_img)
img = np.expand_dims(img, axis=0)
img = img/255
```

```python
print("Following is our prediction:")
prediction = model.predict(img)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
d = prediction.flatten()
j = d.max()
for index,item in enumerate(d):
    if item == j:
        class_name = li[index]


plt.figure(figsize = (4,4))
plt.imshow(new_img)
plt.axis('off')
plt.title(class_name)
plt.show()
```

Fig 48: Sample truly classified leaf spot infected image

6.11 Discussion

The objective of this study is to develop efficient sunflower disease detection and classification for target beneficiaries. These beneficiaries could be private organizations, government organizations, farmers, and others. As mentioned in the problem statement the study focuses on modeling the best-performed methods and algorithms to reduce the challenge of sunflower leaf disease detection and classifications. The selected method to overcome this challenge is the convolutional neural network and its optimizing algorithms. To find the result of this study the prerequisite undertaken is reviewing the related works. As mentioned in the introduction part of this research plant disease detection and classification is the major problem of farmer's societies of the world. Sunflower is the one which is an essential oilseed crop that may be used in both agricultural and oil production systems. Various diseases, insects, and nematodes harm sunflower plants, resulting in a wide range of productivity losses. Disease detection can be done with the naked eye, but when it comes to huge farms, this method is ineffective. To answer this challenge, numerous researchers used various approaches and algorithms to conduct the study.

Particle swarm optimization technique, digital image processing, K-means ++ clustering, and other methods have been utilized in previous research to handle this problem. Many plants disease detection and categorization studies are carried out in Ethiopia. However, despite Ethiopia's significant production, the identification and classification of sunflower diseases rootreceivede attention. As a response to this difficulty, this work developed and demonstrated a model for detecting and classifying Sunflower leaf images. This research is conducted using convolutional neural network methods and their algorithm. Convolutional neural network (CNN) as discussed in the literature part of this study it is the best method of deep learning. The research finding is the SunfNet model with nine (9) convolutional layers and one (1) output layer. Generally, the model created as the result of this study is SunfNet with ten (10) layers. Therefore, it is selected to conduct this research experiment. Different tools and libraries are used during implementation as discussed in chapter 4. The results of the experiments are mentioned in the result portion. The experiments were undertaken using different learning rates, different activation functions, different data split ratios, different data images, and different epochs. These all experiments are used to select the best-performed model which is the solution to the stated problem. The experimented results are evaluated in different evaluation methods as proposed. An evaluation of the model ADAM is used with different learning rates. By another hand, RMS is used as another evaluation method in experiments. Additionally, the Confusion matrix is another method used to evaluate the classification performance of the model. According to the model evaluations the best-fit model found is 99.88 training accuracy and 99.99 validation accuracy. This was obtained from the experiment done using Softmax activation function on RGB color Image with 0.1 learning rate. As discussed, there were plant disease detection and classification were conducted previously. But, the method they used, data they used, layers their model had, the model performance they find are different from this study. For example, the paper [13] used the apple leaf dataset, and performance the achieved is on average 93.5%. The paper [14] used a tomato leaf infection dataset, its model had four layers and the model performance result they obtain is 91.2%. The local paper which is conducted by Mosisa Desalegn [49] is on three wheat diseases. Its most model had three (3) convolutional layers and one (1) output layer and model performance result is 99.76%, but hear the weakness is the model is only detected whether the leaf is infected or not, while SunfNet is detected what disease infects the plant. The second local paper is conducted by Zewdu Tiumay [48] is on five tomato leaf diseases. Its model TomdiseasesNet had seven (7) convolution layers and one (1) output layer and the model performance result is 99.68%, this is increased to **~99.88%** model performance results in the SunfNet model. Generally, the contributions are enhancing research focuses on sunflower farms on step forward, develop Sunflower disease detection and classification as another plant, improving the disease classification accuracy, and solving the challenges of beneficiaries those works on sunflower farms more of in our country.

7 CHAPTER SEVEN: CONCLUSION AND RECOMMENDATION

Plant disease detection and classification is the major problem of farmer's societies of the world. Sunflower is an essential oilseed crop that may be used in both agricultural and oil production systems. Various diseases, insects, and nematodes harm sunflower crops, resulting in a wide range of productivity losses. Disease detection can be done with the naked eye, but when it comes to huge farms, this method is ineffective. To answer this challenge, numerous researchers used various approaches and algorithms to conduct the study. Particle swarm optimization technique, digital image processing, K-means ++ clustering, and other methods have been utilized in previous research to handle this problem. Many plants disease detection and categorization studies are carried out in Ethiopia. However, despite Ethiopia's significant production, the identification and classification of sunflowedisease rootreceivee attention. As a response to this difficulty, this work developed and demonstrated a model for detecting and classifying Sunflower leaf images. This research is conducted using convolutional neural network methods and their algorithm.This study's experiment is focused on four common sunflower diseases. Around four hundred (400) original sunflower photos were gathered from various locations to serve as data samples for these investigations. Nine thousand eight hundred six (9806) augmented images are created from these original images. The target SunfNet is modeled and trained using these images. The model is used to classify and detect sunflower images into their respective categories. The model categorizes the input image based on what it learned during training. The target classes are the diseases that were chosen for this study. Alternaria, leaf spot, mildew, rust, and healthy are the classes. This research finds the more efficient model with a training accuracy of 99.88%. The validation accuracy of this model is 99.34% and the testing accuracy is 99.56%. The research result is more efficient and solves the mentioned problem if properly deployed.

7.1 Recommendation

The research result is more efficient and solves the mentioned problem if properly deployed. The following points can be future work.

- Develop a real-life mobile application for sunflower disease detection
- Develop the model that fire the remedy on detected disease

REFERENCES

1. M.akila and P. Deepan (2018). Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm
2. Jun Dechen, InxiuChen et al. (2020) Using deep transfer learning for image-based plant disease identification
3. Yong Zhong, Ming Zhao (2020) Research on deep learning in apple leaf disease recognition
4. Mohit Agrawal, Abhishek Singh et al. (2020) TOLED: Tomato leaf disease detection using convolutional neural network.
5. Leather, G.R. (1983). Sunflower (Helianthus Annuus) is allelopathic to weeds. Weed Science, 37-42.
6. Wijnands, J. H. M., Gurmesa, N. D., Lute, J. C. M., & Van Loo, E. N. (2011). Ethiopian soya bean and sunflower value chains: Opportunities and challenges (No. 2011-016). LEI, part of Wageningen UR.
7. Hailu, M., & Kelemu, K. (2014). Trends in soybean trade in Ethiopia. Research Journal of Agriculture and Environmental Management, 3(9), 477-484.
8. Belayneh, H. (1990). Progress in sunflower research in Ethiopia. In Oil Crops: Sesame and Sunflower Subnetworks; proceedings of the joint second workshop, Cairo, Egypt, 9-12 Sept. 1989. IDRC, Ottawa, ON, CA.
9. Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. Science, 327(5967), 828-831.
10. Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. Biosystems Engineering, 144, 52-60.
11. Liu, J., Lv, F., & Liu, B. Identification Method of Sunflower Leaf Disease Based on SIFT Point.
12. Schmidt Huber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
13. LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

14. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature 521.
15. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311-318.
16. Matouš, P., Todo, Y., & Mojo, D. (2013). Roles of extension and ethnoreligious networks in acceptance of resource-conserving agriculture among Ethiopian farmers. International Journal of Agricultural Sustainability, 11(4), 301-316.
17. Asia, S., & Lanka, S. (2017). The world factbook. Washington DC: Central Intelligence Agency.
18. Zhong, Y., & Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. Computers and Electronics in Agriculture, 168, 105146.
19. Agrawal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. Procedia Computer Science, 167, 293-301.
20. Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105393.
21. Akila, M., &Deepan, P. (2018). Detection and classification of Plant Leaf diseases by using the Deep Learning Algorithm. Int. J. Eng. Res. Techno, 6, 2-7.
22. Singh, V. (2019). Sunflower leaf diseases detection using image segmentation based on particle swarm optimization. Artificial Intelligence in Agriculture, 3, 62-68.
23. Beyyala, A., & Beyyala, S. P. (2012). Application for diagnosis of diseases in crops using image processing. International Journal of Life Sciences Biotechnology and Pharma Research, 1(1), 172-175.
24. Al Bashish, D., Braik, M., &Bani-Ahmad, S. (2010, December). A framework for detection and classification of plant leaf and stem diseases. In 2010 international conference on signal and image processing (pp. 113-118). IEEE.
25. Wan MohdFadzil W.M.N, Shah Rizam M.S.B, and R. Jailani, Nooritawati M.T "Orchid Leaf Disease Detection using Border Segmentation Techniques" 2014 IEEE Conference on Systems, Process, and Control (ICSPC 2014), 12 - 14 December 2014, Kuala Lumpur, Malaysia, pp. 978-1-4799-6106-1/14.

26. Usama Mokhtar, Mona A. S. Alit, Aboul Ella Hassenian, HeshamHefny "Tomato leaves diseases detection approach based on support vector machines" 2015 IEEE pp. 978-1-5090-0275-7/15.
27. Sabah Bashir and Navdeep Sharma "Remote Area Plant Disease Detection Using Image Processing" IOSR Journal of Electronics and Communication Engineering (IOSRJECE) ISSN: 2278-2834 Volume 2, Issue 6 (Sep-Oct 2012), PP 31-34.
28. Jansson, P. A. (1991). Neural networks: An overview. Analytical Chemistry, 63(6), 357A-362A.
29. Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modeling. Hydrological Sciences Journal, 43(1), 47-66.
30. [27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Over feat: Integrated recognition, localization, and detection using convolutional networks. In International Conference on Learning Representations (ICLR 2014). CBLS, April 2014.
31. 5. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In ICML,2014.
32. 17. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Image Net classification with deep convolutional neural networks. In NIPS, pages 1106–1114, 2012.
33. [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich features hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
34. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
35. Lin, X., Zhao, C., & Pan, W. (2017). Towards accurate binary convolutional neural network. arXiv preprint arXiv:1711.11294.
36. Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195.
37. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging, 9(4), 611-629.
38. Mokhtar, I. (2009). Sunflower disease and insect pests in Pakistan: A review. African crop science journal, 17(2).

39. Virányi, F. (2008, June). Research progress in sunflower diseases and their management. In Proc 17th Intl. Sunflower Conf., Cordoba, Spain (pp. 8-12).
40. Kumar, K. V., & Jayasankar, T. (2019). An identification of crop disease using image segmentation. Int. J. Pharm. Sci. Res, 10(3), 1054-1064.
41. Barbedo, J. G. A. (2013). Digital image processing techniques for detecting, quantifying, and classifying plant diseases. Springer Plus, 2(1), 1-12.
42. Ramesh, S., & Vydeki, D. (2019). Application of machine learning in the detection of blast disease in south Indian rice crops. J. Phytol, 11(1), 31-37.
43. Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. Biosystems Engineering, 144, 52-60.
44. Liu, J., Lv, F., & Liu, B. Identification Method of Sunflower Leaf Disease Based on SIFT Point.
45. Singh, V. (2019). Sunflower leaf diseases detection using image segmentation based on particle swarm optimization. Artificial Intelligence in Agriculture, 3, 62-68.
46. https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05.
47. https://www.sunflowernsa.com/gallery/default.aspx?CID=28&SCID=21,2021-17-may
48. Zewdu T. (2020). Tomato leaf disease detection and classification using CNN.
49. Mosisa D. (2019). Applying a Deep Learning Approach for Wheat Rust Disease Detection
50. Vijai Singh, Sunflower leaf diseases detection using image segmentation based on particle swarm optimization, Artificial Intelligence in Agriculture, Volume 3,2019,
51. J. Liu, F. Lv and P. Di, "Identification of Sunflower Leaf Diseases Based on Random Forest Algorithm," *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, 2019, pp. 459-463.
52. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: HTTP: //doi.org/10.1038/nature14539
53. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Image Net classification with deep convolutional neural networks," 2012. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary
54. J. Turian, J. Bergstra, and Y. Bengio, "Quadratic features and deep architectures for chunking," in Proceedings of Human Language Technologies: The 2009 Annual

Conference of the North American Chapter of the Association for Computational Linguistics, vol. Companion Volume: 2009, pp. 245–248. [Online]. Available: https://dl.acm.org/citation.cfm

55. D. L. Elliott, “A Better Activation Function for Artificial Neural Networks,” 1993. [Online]. Available: https://drum.lib.umd.edu/handle/1903/5355
56. V. Badri Narayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” arXiv, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561
57. Bharati, M., & Ramageri, M. (2010). Data mining techniques and applications
58. Authors, “Vijai Singh GL Bajaj Institute of Technology and Management, Gr. Noida, U.P., Indi”
59. Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., ... & Guo, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, *11*(4), 32-44.
60. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR, 2016. Available at: https://arxiv.org/abs/1511.06434.

Appendix

Appendix I: Program to create SunfNet model

```
import numpy as np
import keras
from keras import backend as k
from keras.regularizers import l1
from keras.layers import Dense, Dropout, GlobalAveragePooling2D, Reshape, Activ
Conv2D, Input, MaxPooling2D, BatchNormalization, Flatten, Dense, Lambda, InputL
from keras.optimizers import Adam
from keras.metrics import categorical_crossentropy, categorical_accuracy
from keras.preprocessing.image import ImageDataGenerator
from keras import models, regularizers
from keras.applications import imagenet_utils
from sklearn.metrics import confusion_matrix
import itertools
import matplotlib.pyplot as plt
from keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoard
import tensorflow as tf
import os
from keras.models import Sequential
from keras.layers import Flatten
# import os
num_classes = 5  # set the classes as the number of folder the data is divided
img_rows, img_cols =150, 150    # set image height and width
batch_size = 64   # no. of data pass per process
from keras.preprocessing.image import ImageDataGenerator
train_data_dir = 'dataset_splits1/train'
validation_data_dir = 'dataset_splits1/validation'
test_data_dir = 'dataset_splits1/test'
# Let's use some data augmentaiton
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    rotation_range=30,
    width_shift_range=0.3,
    height_shift_range=0.3,
    horizontal_flip=True,
    fill_mode='nearest')
validation_datagen = ImageDataGenerator(rescale=1. / 255)
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_rows, img_cols),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True)
validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_rows, img_cols),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False)
Found 7850 images belonging to 5 classes.
Found 978 images belonging to 5 classes.
```

Appendix II: Data splits

```
import os, shutil
# Path to the directory where the original dataset is
original_dataset_dir = './augmented gray/'
# list of all plant diseases classes in the original dataset
classes_list = os.listdir(original_dataset_dir)
#Directory where you'll store your dataset splits
base_dir = './dataset_splitsgrayscale'
os.mkdir(base_dir)
#Directories for the training, validation, and test splits
train_dir = os.path.join(base_dir, 'train')
os.mkdir(train_dir)
validation_dir = os.path.join(base_dir, 'validation')
os.mkdir(validation_dir)
test_dir = os.path.join(base_dir, 'test')
os.mkdir(test_dir)
for cls in classes_list:
    #Directory for training pictures
    os.mkdir(os.path.join(train_dir, cls))
    #Directory for validation pictures
    os.mkdir(os.path.join(validation_dir, cls))
    # Directory for test pictures
    os.mkdir(os.path.join(test_dir, cls))

import math

for cls in classes_list:
    path = os.path.join(original_dataset_dir, cls)
    fnames = os.listdir(path)
    train_size = math.floor(len(fnames) * 0.8)
    validation_size = math.floor(len(fnames) * 0.1)
    test_size = math.floor(len(fnames) * 0.1)
    train_fnames = fnames[:train_size]
    print("Train size(",cls,"): ", len(train_fnames))
    for fname in train_fnames:
        src = os.path.join(path, fname)
        dst = os.path.join(os.path.join(train_dir, cls), fname)
        shutil.copyfile(src, dst)
    validation_fnames = fnames[train_size:(validation_size + train_size)]
    print("Validation size(",cls,"): ", len(validation_fnames))
    for fname in validation_fnames:
        src = os.path.join(path, fname)
        dst = os.path.join(os.path.join(validation_dir, cls), fname)
        shutil.copyfile(src, dst)
        test_fnames = fnames[(train_size+validation_size):(validation_size +
+test_size)]
    print("Test size(",cls,"): ", len(test_fnames))
    for fname in test_fnames:
        src = os.path.join(path, fname)
        dst = os.path.join(os.path.join(test_dir, cls), fname)
        shutil.copyfile(src, dst)
Train size( alterneria ):  845
Validation size( alterneria ):  105
Test size( alterneria ):  105
Train size( healthy ):  3865
Validation size( healthy ):  483
Test size( healthy ):
Train size( leafspot ):  1087
Validation size( leafspot ):  135
Test size( leafspot ):  135
Train size( mildew ):  2053
Validation size( mild
Test size( mildew ):  256
Train size( rust ):  2295
Validation size( rust ):  286
Test size( rust ):  286
```

Appendix III: Program code for Image Augmentation

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.applications import imagenet_utils
from sklearn.metrics import confusion_matrix
import itertools
import os
import shutil
import random
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Dense
from keras import applications
%matplotlib inline
from keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
        rotation_range=40,
        width_shift_range=0.2,
        height_shift_range=0.2,
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest')
i = 0
for batch in
datagen.flow_from_directory(directory='C:/Users/Administrator/Desktop/Research/dani2/rust/',
                                          batch_size=32,
                                          target_size=(150, 150),
                                          color_mode="grayscale",

save_to_dir='C:/Users/Administrator/Desktop/Research/aug/rust/',
                                          save_prefix='aug',
                                          save_format='jpg'):
```

Appendix IV: Model testing sample

```
from keras.preprocessing import image

import numpy as np

image_path = './dataset_splits1/test/alterneria/aug_0_4557.jpg'

new_img = image.load_img(image_path, target_size=(150, 150))

img = image.img_to_array(new_img)

img = np.expand_dims(img, axis=0)

img = img/255

print("Following is our prediction:")

prediction = model.predict(img)

# decode the results into a list of tuples (class, description, probability)

# (one such list for each sample in the batch)

d = prediction.flatten()

j = d.max()

for index,item in enumerate(d):

    if item == j:

        class_name = li[index]

plt.figure(figsize = (4,4))

plt.imshow(new_img)
```

Appendix V: Sample Visualized image