ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

PLANT DISEASE DETECTION AND CLASSIFICATION USING ARTIFICIAL NEURAL NETWORK

By

Endrias Haile

A Thesis submitted to Addis Ababa University, Addis Ababa Institute of Technology, School of Graduate Studies, School of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Computer Engineering

Advisor

Dr.Eng. Getachew Alemu

July, 2019
Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

(Computer Stream)

By Endrias Haile Gebrehiwot

Approved by Board of Examiners

Dean	**Signature**
School of Electrical and Computer Engineering	
Advisor	**Signature**
External Examiner	**Signature**
Internal Examiner	**Signature**

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work performed under the supervision of Dr.Eng. Getachew Alemu, has not been presented as a thesis for a degree program in any other university and all sources of materials used for the thesis work have been fully acknowledged.

Name:	Endrias Haile Gebrehiwot
Signature:	
Place:	Addis Ababa University, Addis Ababa, Ethiopia
Date of submission:	July 2019

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

Advisor's Name:	Dr.Eng. Getachew Alemu
Signature:	
Place:	Addis Ababa University, Addis Ababa, Ethiopia
Date of submission:	July 2019

DEDICATION

To my beloved family, your strong values, belief, and love make this thesis possible!!!

ACKNOWLEDGMENT

This thesis would not have been possible without the support of many people and some organization.

First of all I would like to express my deepest gratitude to my advisor Dr.Eng. Getachew Alemu for his supervision, persistent advice, inspiration, patience, time and continued guidance right from the moment of becoming my advisor, topic selection, statement of problem formulation to the completion of the work. I also want to thank him for his comprehensive lectures on Machine Learning which helped me solve the problem of the thesis.

I would like to sincerely thank my wife Selam Amaha and my daughter Hyab Endrias for their never ending love and support despite the fact that I abandoned them focusing on my work. Their contribution towards my thesis is immeasurable.

I would like to express my sincere appreciation to all electrical and computer engineering staff who guided and extended their valuable knowledge and advice through the different phases of the seminars which helped me in my research work.

I would like to forward my special thanks to Ethiopian Agricultural Research institute staff members of the organization for their kind cooperation and for giving me all the necessary information during my work. I would like also to acknowledge the plant village org website for the availability of their plant disease dataset which is used for this thesis.

Last but not least, I would like to sincerely thank all of my friends, colleagues, classmates for their assistance, encouragement, and inspiration in this thesis work.

Above all I would like to thank God for giving me courage, unconditional support not only in the good situations but also in the bad situations of my life, and strength to complete this work.

Table of Contents

List of Figures

List of Tables

List of Acronyms

CNN: Convolutional Neural Network

NN: Neural Network

ANN: Artificial Neural Network

RF: Random Forest

SVM: Support Vector Machine

KNN: K Nearest Neighbor

DT: Decision Tree

RGB: Red, Green and Blue Color Mode

CPU: Central Processing Unit

ML: Machine Learning

MSE: Mean Squared Error

RAM: Random Access Memory

SGD: Stochastic Gradient Descent

ABSTRACT

Continuous advancement in image processing and machine learning techniques have made it possible for computers to see and learn. What is seen by the eyes of human beings could be divided into pixels and given to a computer so that the computer will be able to see and learn based on the provided values. Based on the input values fed in computers could learn to identify various things based on the things they have been learnt from them. There are many possible areas in which computers can be applied to see and learn in order to make the life of human beings much easier.

In this thesis an approach has been proposed which is capable of automatically detecting and classifying plant disease from an image based on artificial neural network. Now a days, plants have become much more important than they used to some years ago where they have been only used to feed mankind as well as animals. Plant diseases are currently detected and classified using methods that requires a lot of manual work with experts, agricultural extension worker and farmers which is both time consuming and error prone. To automate the process of plant disease detection and classification different researchers have studied many techniques using both machine learning and image processing. However, these proposed techniques still have limitation.

The steps followed in this research for detecting and classifying the plant disease are: dataset collection, image pre-processing, masking, and removing the green part, feature extraction and selection, classification, and disease management techniques. For comparing and demonstrating the conventional machine learning techniques and proposed approach respectively two different types of plant have been selected namely, maize, and potato from the plantvillage.org website. Since the conventional machine learning techniques do not have the potential to extract and select features from a given raw data, texture features using Haralick's from color co-occurrence matrix have been extracted and selected using subset feature selection technique.

The proposed approach and the selected conventional machine learning techniques were evaluated using confusion matrix, classification performance report, and t-test to asses which has the higher classification potential. The proposed approach achieved an average accuracy of 97.6%, average precision of 97.0%, 97.0% of average recall ,and average F1 value of 97.0% over a test dataset of previously unseen 1201 images. From the analysis of the experimental results the proposed approach gives best result than the conventional machine learning classifier. This due to the fact that convolutional neural network extract high level features from the input raw data, making it more efficient and accurate, and avoid errors due to a subjective manual feature extraction thereby showing the feasibility of its usage in real time applications for the classification of healthy and non-healthy plants.

Keywords: plant disease, image processing, masking and removing feature extraction and selection, machine learning methods, disease management techniques.

CHAPTER ONE

1.1 Background

Nowadays, plants have become much more important than they used to some years ago where they have been only used to feed human beings as well as animals. This is due to the fact that as stated in [1] plants are now used to prevent soil erosion, reduce wind, and one-fourth of the drugs that are prescribed is derivatives of or come directly from plant to improve upon the living conditions of human beings. In addition, plants bring oxygen which is a by-product of photosynthesis and store carbon and they help in keeping a lot of the produced carbon dioxide from burning fossil fuels out into the earth's atmosphere.

However, plants also get sick. According to [2] the agents that cause disease in plants are the same or very similar to those causing disease in humans and animals. Plant disease is mainly caused may be due to pathogenic microorganisms, such as viruses, bacteria, fungi, protozoa, and nematodes, and unfavorable environmental conditions, such as lack or excess of nutrients, moisture, and light, and the presence of toxic chemicals in air or soil. Most of the diseases that affect plants are caused by fungi which appear as spots on plant leaves. These spots make it very difficult for such plants to prepare their food by means of photosynthesis since they affect the green pigments in the leaf, hence to a large extent affects the growth and the yield of such plants. In circumstances where the fungi infection becomes severe, the spots cover the entire surface area of the leaf.

As explained in [3] the diseases in plants do not only reduce the yield but can also deteriorate the variety of such plants and its withdrawal from cultivation. There are so many diseases that affect plants that cause great economic and productions losses. It can even lead to great ecological losses. Plant diseases especially leaf diseases are usually curbed using insecticides, fungicides, and pesticides. However, excessive application of these chemicals for the treatment of plant diseases can result in poisoning their produce as well as causing other harms to humans and animals. The danger of toxic residue on crops due to the application of pesticides on plants that have been affected by various forms of diseases has been identified as a major contributor to groundwater pollution and contamination. Again, too much application of pesticides by farmers increase the cost of production which can lead to a greater loss.

Since plants play a great role in the ecosystem, thus effective disease management mechanism using a nondestructive technique is required at an early stage so as to prevent the quality and quantity loss accurately and timely.

Plant diseases can be detected and classified through several means including manual and computer-based systems. As stated in[4] in most of the cases, plant diseases are seen as spots on the leaves which are more visible to human eye. The traditional approaches used for the disease detection and classification of the diseased plants are based on field scouting which is time consuming, tedious and may result in several errors and inaccurate results.

On the other hand, there are some diseases that do not appear on the leaves and others appear in later stages after they have already caused severe effect on the plants. In such instances, [5] recommended to use computerized systems that would be the only way to detect the situation timely and accurately using some kind of complex algorithms and analytical tools, preferably through the use of powerful microscopes and other machines. In some other instances, the signs can only be detected through the electromagnetic means which produces more images that are not visible to the human eye. [6] provides another means of investigating plant disease through a technique known as Remote Sensing Technique (RST) that examines and diagnoses using multi hyperspectral image captures. All the methods that use the RST approach usually fall on digital image processing tools to achieve their desired results. According to these referred papers the application of image processing techniques in conducting research in the Agric sector has helped in diverse ways to improve upon the development of the agricultural sector.

Plant disease detection and classification using image processing and machine learning techniques have been an issue for a long time. To improve the accuracy rate of plant disease detection and classification, researchers have studied many techniques using machine learning and image processing. Some of these techniques include machine learning methods such as Convolutional Neural Network [7], Artificial Neural Network [8], Back Propagation Neural Network [9], and Support Vector Machine [10]. However, none of these methods that have adopted by the various researchers have been outstanding. This research seeks to detect and classify plant disease on leaves using artificial neural networks.

1.2 Statement of the Problem

For increasing the crop production and productivity farmers contact experts to ask for their advice concerning the action towards the occurrence of the diseases to their crops and recommendation for control. Some times they have to go long distances to approach the experts, even though they go such long distances the expert whom the farmer wants to contact may not be in a position for advising the farmer. In these cases, looking for expert advice is very expensive and time-consuming.

In addition, plant disease or deficiency detection and classification are usually carried out by farmers, agricultural extension workers, and pathologies by continues monitoring of the plants. In

small scale farm, early detection and classification of disease are very much simple and the disease could be easily controlled by organic pesticides or by the use of a minimum amount of chemical pesticides. In large scale farm, continuous monitoring and early detection and classification of disease are very difficult and it results in a severe outbreak of the disease and pest growth could not be controlled by organic means. In this situation, farmers are urged to use chemicals pesticides so as to control the disease and to retain the crop yield.

These problems could be solved by automating the monitoring process with the help of image processing and machine learning techniques. Different researchers have attempted the problem of plant disease detection and classification for long time. Most of the machine learning techniques that have been used by those researchers for the classification purpose requires manual training and have limited accuracy. An additional limitation of the typical proposed techniques is that the set of features for the classification has to be generated by the developer to work with a specific configuration of the model. Those models do not generalize very well because the generated features are not high-level features that describe the disease.

Therefore rapid, accurate detection and classification of plant disease using deep learning neural network is very important to cure and control the spreading of diseases. High prediction accuracy and speed of plant disease detection and classification is vital for increasing agricultural production and productivity. The latest improvements in deep learning have increased its capability to solve complex visual recognition task. Therefore, using deep learning to improve the accuracy and speed of solving plant disease detection and classification task is reasonably promising.

1.3 Research Questions

The research questions that will be addressed in this study are:

Research question 1 Does convolutional neural networks out-perform other machine learning techniques for plant disease detection and classification?

Research question 2 Does convolutional neural networks adapt for detection and classification of the same diseases that threaten other plants than those which will be used for building the model?

1.4 Hypothesis

In this research the following hypothesis is formulated:

H1: Disease appeared in a leaf of maize and potato plants can be detected and classified using deep learning neural networks with better accuracy.

1.5 Objective

1.5.1 General Objective

The general objective of the study presented in this thesis is to research and develop a model that can automatically detect and classify plant diseases on leaves based on deep learning and to evaluate its performance.

1.5.2 Specific Objective

The specific objectives of this research work are:

- Explore the different types of maize and potato leave diseases.
- To select commonly used image processing algorithms used frequently in literature and apply them for plant disease detection and classification.
- To develop an algorithm for masking and removing the green part of the images.
- To design a deep neural network for plant disease detection and classification.
- To train the specified deep neural network architecture
- To verify the reliability and accuracy of the developed system using the datasets collected.
- Analyze and compare results with different machine learning methods
- Based on the obtained results, draw conclusions and propose a set of future improvements.

1.6 Scope and Limitation

During the study, it has been realized that certain aspects of the study were beyond his capacity. In such cases, nothing has been done about such aspects. Some of these limitations include the following:

The scope of the research was only limited to the detection and classification of two types of food plants namely, maize and potato which are the third and fourth commonly used crops worldwide. Due to the unavailability of datasets, the researcher didn't try to detect and classify all the diseases associated with maize and potatoes rather he chose the most common diseases appeared on the leaves of the selected crop.

It was difficult to get local datasets from the Ethiopian Agricultural Research Institute who had knowledge of the issues that should have been used for testing the model.

Well correlated features selection is computationally infeasible and lengthy process, so in this study it is not possible to check all possible subset of features. Only some random features based on the previous best features have been checked.

1.7 Significance of the study

It is a viable fact that artificial intelligence has come to make the life of human being easier and there is no way to eliminate its use. It is therefore left to the users to utilize it appropriately to improve their living conditions. In the first place, this study will help pathologist understand the importance of artificial intelligence in general and machine learning in particular in their area. In addition, it will help achieve high quality and quantity production because it lets detail disease management techniques in curing the disease appeared on the plant for pathologist, users, and agricultural extension workers.

In addition, the study will help to minimize excessive toxic waste to water bodies as well as plants that harm human life. The study will help farmers reduce the cost of production that brings huge loss due to excessive use of fungicides on their plants.

Once again, the outcome of the study can provide important suggestions and feedback to the corresponding authorities in order to take appropriate measures as far as agricultural production is concerned, in a case where there is an outbreak of crop diseases.

Finally, this study would serve as a reference resource for other researchers who would seek to conduct further studies into the problems related to plant disease detection and classification.

Generally, the result obtained by the proposed methodology was found very promising with respect to the environment, productivity, labor saving, and profitability and would be highly demanded by farmers, agricultural extension workers, and pathologist.

1.8 Methodology

The research plan was performed in four major phases: literature review, image data collection, image processing, disease classification and management techniques.

i. Literature Review: A literature survey was conducted on the area of image processing for every stage of the system. Available books, journals, case studies, previous research

works & guidelines were surveyed in order to have a clear understanding of the subject matter.

ii. Data Collection: The necessary image of maize and potato leaves diseases were collected from plant village .org website.

iii. Image processing: This phase consists of three typical steps. First, the images were preprocessed to create noise-free images. Then, masking and removing the green part of the image was performed. Finally, texture features extraction and selection for the consecutive green removed images were preformed. The features selected in step three were then fed to the conventional machine learning algorithms to classify the input image in to the corresponding category, whereas the green removed images were given to the proposed approach to extract and select high level features from the given raw input data and then to classify the given input image to the corresponding category.

iv. Classification and management: This final phase of the research work was targeted at classifying the disease into the corresponding categories. The features extracted in step three were given to the classification model to classify the disease into the corresponding. Finally, once the type of the disease is identified the management technique suggest the right procedure for curing it.

v. Generally, the core thesis work is made up of data collection, noise removal, masking and removing the green part of the image, reducing the dimension of each and every image, and finally, the classification and management of the identified disease are provided.

1.9 Research Contribution

So far, a number of researches have been conducted to detect and classify plant disease for early diagnosis of these diseases. The focuses were to detect and classify diseases that threaten particular plant using different image processing and machine learning techniques. In addition, it is likely that different plant could be threatening by the same disease. Thus, after a thorough literature review, and to the best of the researcher knowledge, this work is the first to perform comparison on the most commonly used machine learning techniques with the proposed approach for plant disease classification. This is particularly important in the activity of plant disease classification as a misclassification could have a severe impact on the plant. The second contribution of this thesis is that the developed model could detect similar disease that affects other plants which have not been involved in training the model. This contribution should help to develop a general framework instead of developing a separate model for each plant and to avoid training and retraining a model when necessary.

Finally, after what type of disease is occurred on the plant is detected and classified the disease management technique would help farmers, agricultural extension worker, and pathologies what kind of precaution should they take on that plant for curing and controlling the disease before it incur sever losses.

1.10 Organization of the Thesis

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

CHAPTER TWO

Literature Review and related works

The purpose of this study is to develop a model that is capable of automatically detecting and classifying plant diseases on leaves using neural networks. In this chapter diseases affecting maize leaves, and potato leaves along with their symptom and management techniques will be discussed. It will then focus on the definition of machine learning, most commonly machine learning techniques for plant disease classification will also be discussed.

Review of existing articles written by other scholars or authors which are relevant for the study will also be summarized. Finally, by reviewing various techniques, the merit and demerit of different method and their characteristic for plant disease detection and classification will be presented.

2.1 Plant Diseases

A plant become diseased when it is continuously disturbed by some causal agent that results in an abnormal physiological process that disrupts the plant's normal structure, growth, function, or other activities. Plant disease can be broadly classified according to the nature of their primary causal agent, either infectious or noninfectious [12]. Infectious plant diseases are caused by a pathogenic organism such as a fungus, bacterium, mycoplasma, virus, viroid, nematode, or parasitic flowering plant. An infectious agent is capable of producing within or on its host and spreading from one susceptible host to another.

In nature, plants may be affected by more than one disease-causing agent at a time. A plant that must contained with a nutrient deficiency or an imbalance between soil moisture and oxygen is often more susceptible to infection by a pathogen. A plant infected by one pathogen is often prone to invasion by secondary pathogens.

Maize and potatoes are some of the plant species that are attacked by different type of disease. Maize and potato diseases are caused by fungi, bacteria and pests. Some of the disease that attack the maize and potato species will be explained in the next section.

2.2 Corn (maize)

Corn, Zea mays, is an annual grass in the family of Poaceae and is a staple food crop grown all over the world. According to [13] Maize is Ethiopia's leading cereal crop in terms of production with 6.2 million tons produced in 2013 by 9.3 million farmers across 2 million hectares of land.

Ethiopian farmers grow maize, primarily for subsistence with 75% of all maize output consumed by farming households, making it a key crop for overall food security and for economic development in the country.

In Ethiopia, maize disease was first reported in 2013/2014 cropping seasons which cause various levels of damage ranging from low infection rate to total crop failure [14]. In some maize fields in Koka area in East Shewa Zones of Oromia Region and Duguna Fango area of Wolayta zone in SNNPR, the disease has caused a total crop failure forcing farmers to replace maize with other crops. There are more reports in recent months indicating the occurrence and damage of the disease in many districts of Jinka area of SNNPR as well as in Amhara region in the north. This entails the disease is expanding in higher rate in Eastern regions of Africa within relatively shorter period of time. The contents which will be used to describe diseases affecting maize and potato leaves, their symptoms and management techniques are taken from [1].

2.2.1 Diseases Affecting Maize leave

Corn (maize) is affected by different diseases such as Gray leaf spot, Northern Leaf Blight, Common rust, Maize lethal necrosis, maize streak etc. In this research three leaf disease that commonly affect maize will be discussed.

Gray leaf spot (Cercospora leaf spot) Cercospora zeae-maydis

Gray leaf spot (Cercospora leaf spot) Cercospora zeae-maydis is favored in areas where a corn crop is followed by more corn with no rotation. Severity and incidence of disease is likely to be continuous with minimum tillage. Prolonged periods of foggy or cloudy weather can cause severe Cercopora epidemics.

Symptoms of Gray leaf spot

Small necrotic spots with chlorotic halos on leaves which expand to rectangular lesions 1-6 cm in length and 2-4 mm wide. As the lesions mature they turn tan in color and finally gray. Lesions have sharp, parallel edges and are opaque. The disease can develop quickly causing complete blighting of leaves and plant death.

Management of Gray leaf spot

Plant corn hybrids with resistance to the disease. Crop rotation and plowing into soil may reduce levels of inoculum in the soil but may not provide control in areas where the disease is prevalent. Foliar fungicides may be economically viable for some high yielding susceptible hybrids.

Northern Leaf Blight Exserohilum turcicum

Northern corn leaf blight (NCLB) is a foliar disease of corn (maize) caused by Exserohilum turcicum, the anamorph of the ascomycete Setosphaeria turcica. With its characteristic cigar-shaped lesions, this disease can cause significant yield loss in susceptible corn hybrids [18].

Symptoms of Northern Leaf Blight

In the beginning elliptical gray-green lesions on leaves are noticed. As the disease process this lesions become pale gray to tan color. Later stage the lesions looks dirty due to dark gray spores particularly under lower leaf surface. The disease can be easily identified in the field due to its long, narrow lesions which are unrestricted by veins.

Management of Northern Leaf Blight

Follow proper tillage to reduce fungus inoculum. Apply crop rotation with non host crop. Grow available resistant varieties. In severe case of disease incidence apply suitable fungicide.

Common rust Puccinia sorghi

Common rust is caused by the fungus Puccinia sorghi. Late occurring infections have limited impact on yield. The fungus overwinters on plants in southern states and airborne spores are wind-blown to northern states during the growing season. Disease development is favored by cool, moist weather (60 – 70◦ F).

Symptom of Common rust

Oval or elongated cinnamon brown pustules on upper and lower surfaces of the leaves are noticed. If infection is severe, pustules may appear on tassels and ears and leaves may begin to yellow. In partially resistant corn hybrids, symptoms appear as chlorotic or necrotic flecks on the leaves which release little or no spore.

Management of Common rust

The most effective method of controlling the disease is to plant resistant hybrids. The application of appropriate fungicides may provide some degree on control and reduce disease severity. Fungicides are most effective when the amount of secondary inoculum is still low, generally when plants only have a few rust pustules per leaf.

2.3 Potato

Potato, Solanum tuberosum, is an herbaceous perennial plant in the family of Solanaceae which is grown for its edible tubers. The potato plant has a branched stem and alternately arranged leaves consisting of leaflets which are both of unequal size and shape. Among African countries, Ethiopia has possibly the greatest potential for potato production: 70 percent of its arable land - mainly in highland areas above 1 500 m - is believed suitable for potato. Since the highlands are also home to almost 90 percent of Ethiopia's population, the potato could play a key role in ensuring national food security. Currently, potatoes are still commonly used as a secondary crop, and annual per capita consumption is estimated at just 5 kg. However, There are many factors that reduce potato production among which are diseases like early blight, and late blight potato important role in reduction of the yield. According to [15] in Ethiopia potato yield loss is estimated from 6.5 to 61.7% due to disease especially by late blight. This research will discuss on the detection and classification of two leaf diseases that commonly affect potato.

2.3.1 Diseases Affecting Potato leaves

Early Blight

As explained in [16] Early blight is a very common disease of potato that is found in most potato-growing areas. Although it occurs annually to some degree in most production areas, the timing of its appearance and the rate of disease progress help determine the impact on the potato crop. The disease occurs over a wide range of climatic conditions and depends in large part on the frequency of foliage wetting from rainfall, fog, dew or irrigation, on the nutritional status of foliage and on cultivar susceptibility. Though losses rarely exceed 20 percent, if left uncontrolled, the disease can be very destructive.

Symptoms of Early Blight Potato Disease

Dark lesions with yellow border which may form concentric rings of raised and sunken tissue on the leaves and stems. Lesions initially circular but become angular. The leaves become necrotic but remain attached to plant. Dark, dry lesions on tubers with leathery or corky texture and watery yellow green margins

Management of Early blight

Application of appropriate protective fungicide can reduce severity of foliar symptoms. Reduce stress to plants by fertilizing and watering adequately. Plant late varieties which are less susceptible to disease and store tubers in cool environment.

Late blight Phytophthora infestans

Potato late blight is caused by Phytophthora infestans which is not a fungus but a water mould also known as "oomycete". It produces mycelium which can be of two different type of strains: A1 and A2, that are of opposite sexual compatibility. The mating of the two sexually compatible strains, on a potato plant, may lead to the formation of oospores. The latter are resistant organs which can survive in the soil for several years. More frequently, the cycle of Phytophthora infestans is asexual. In winter, it normally survives in the form of asexual mycelium in tubers left in the field (ground keepers), in waste piles and in storage. In the spring, this surviving mycelium produces sporangia which are spread by wind and rain, infecting new healthy plants and crops. Under conditions of high humidity, one cycle normally takes 4 to 6 days at an average temperature of 15°C. After the primary infection, subsequent infection cycles can occur and be responsible for a very rapid and destructive epidemic on the foliage.

Later in the growing cycle of the potato, newly formed tubers are directly infected in the field: sporangia formed on the aerial parts of the plants are carried by rainwater into the soil and infect the tubers.

Symptoms of Late blight

Irregularly shaped spreading brown lesions on leaves with distinctive white fluy sporulation at lesion margins on the underside of the leaf in wet conditions. In dry condition the lesions dry up and go dark brown with collapsed tissue, and water-soaked dark green to brown lesions on stems also with characteristic white sporulation. Later in infection leaves and petioles completely rotted

and severely aected plants may have an slightly sweet distinctive odor; red-brown firm lesions on tubers extending several centimeters into tissue. Lesions may be slightly sunken in appearance and lead to secondary bacterial rots.

Management of Late blight

Control depends on a multifaceted approach with importance of certain practices changing based on geographic location: destroy infected tubers; destroy any volunteer plants; application of appropriate fungicide to potato hills at emergence; time watering to reduce periods of leaf wetness e.g. water early to allow plant to dry during the day; plant resistant varieties; apply appropriate protective fungicide if disease is forecast in area

2.4 Machine learning techniques

To solve a problem on a computer, an algorithm, a sequence of repeatable instructions that should be carried out to transform input to output is needed. In conventional computing, this logic is often explicitly coded by a programmer after carefully learning the problem and determining logical steps to solve it. However, not all computer problems can be explicitly coded. According to [17] machine learning is a subfield of computer science that explores the study and construction of algorithms that can learn from and make predictions based on data. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where and what to look for. In crop diseases detection and classification applications, machine learning algorithms are often used to perform classification.

Machine learning algorithms can be categorized into two general groups: supervised learning and unsupervised learning. Supervised learning algorithms are trained using a set or sets of inputs along with corresponding correct outputs, and the algorithms' task is then to generate a general rule that maps inputs to outputs. The generated rule can then be used in similar applications to classify additional unlabeled data. Examples of supervised learning algorithms include Support Vector Machines and K-Nearest Neighbors algorithms. Supervised learning is commonly used in applications where historical data predicts likely future events and where there are well known outputs to given inputs.

In unsupervised learning, the algorithm is only given input data without any corresponding output data and its task is then to explore the data and find some structure within. Examples of

unsupervised learning algorithms include K-means - clustering, Self Organizing Maps (SOMs) and Principal Component Analysis (PCA). Other examples include Self-Organizing Maps and Independent Component Analysis algorithms. Unsupervised learning can be used to discover image features and to determine their classes as well.

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

2.4.1 Artificial Neural Network

As stated in [18], Artificial Neural Networks are inspired by biological neural networks. An ANN consists on a structured organization of multiple artificial neurons. These neurons receive one or multiple inputs and emit an output based on the weighted sum of the inputs and a function. The function applied to the weighted sum is called the activation function. A weight is a numeric value that determines the strength of the connection.

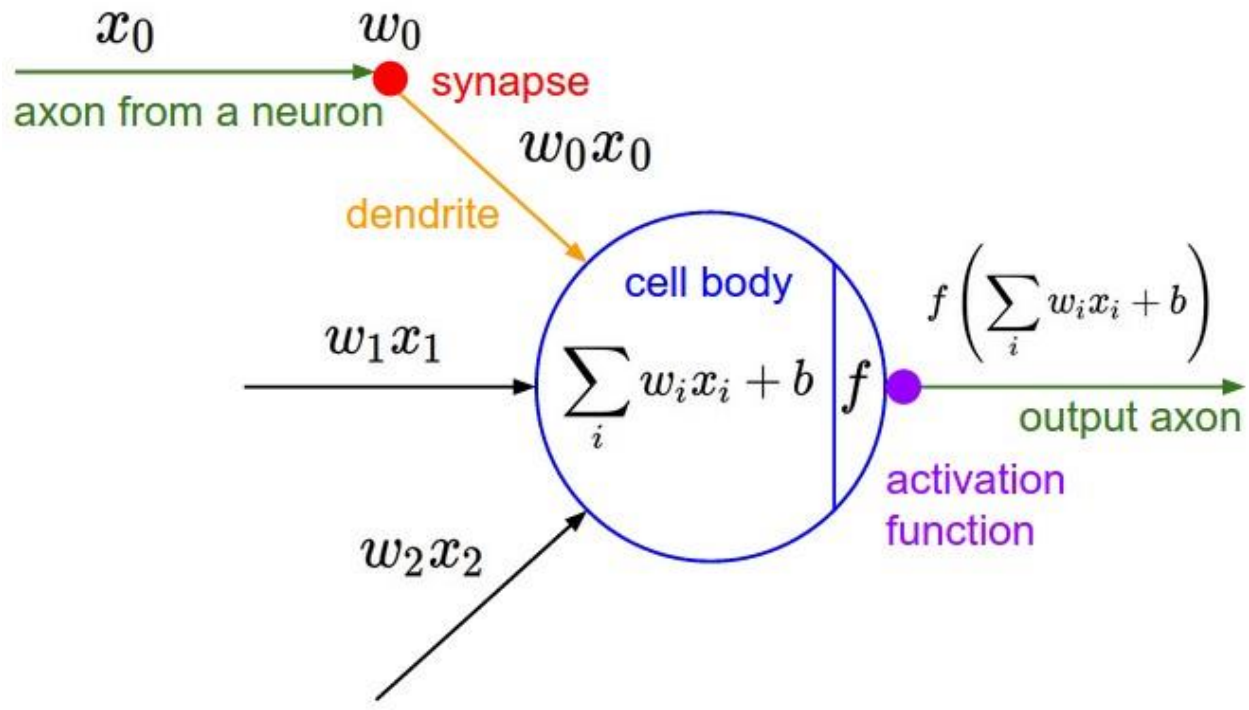


Figure 2.1: Representation of an artificial neuron compared to a biological neuron.

These neurons are grouped for creating layers, by stacking these layers the networks are created. The first layer of the network is known as the input layer, the intermediate are known as hidden layers and the last one is the output layer. The number of hidden layers will denote the depth of the network. The more layers the network has, the deeper the network will be. A neural network

consists of a number of layers of increasingly meaningful data representations, which in turn are a collection of one or more nodes as indicated in figure 2.2. Each layer is learned by exposure to examples, i.e. giving the model input-output pairs for training. Layers between the input and output layers are often referenced as hidden layers, as they are not visible as a network output.

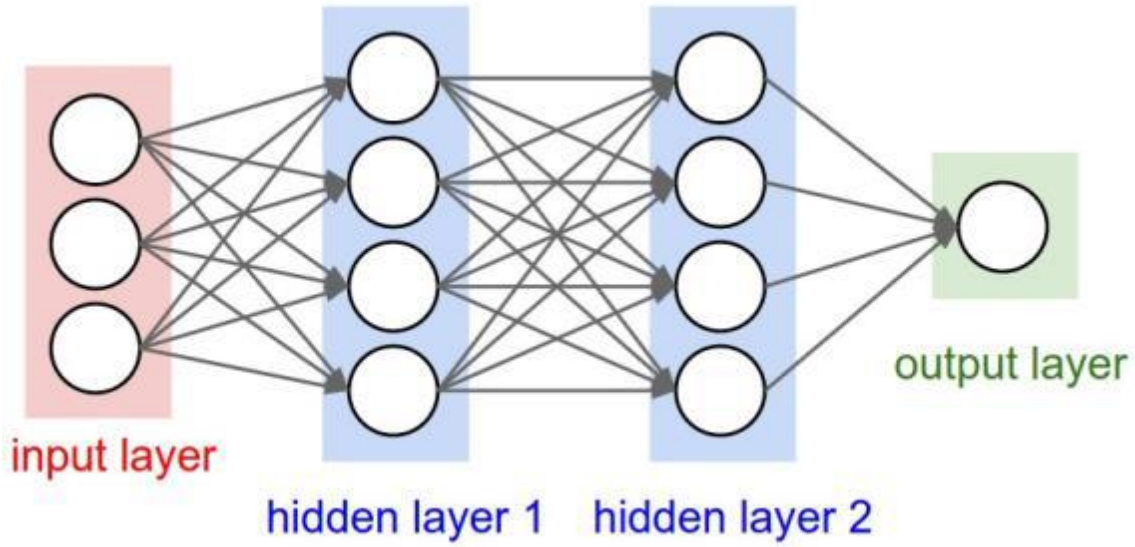


Figure 2.2:Example of a neural network with an input layer, two hidden layers, and an output layer. Every input to the neural networks passes through the hidden layers.

Activation

In order to make sure the network can represent a nonlinear function, an activation function is applied to each layer. Commonly, an activation function is a logistic function, also known as the sigmoid function, calculated with the following formula:

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

Another typical used activation function is the rectified linear unit, known as the ReLU function The ReLU takes all negative values and turns them into zero

$$f(x) = max(0, x)$$

Figure 2.3 shows the graphical representation of the sigmoid and ReLU functions. By applying non-linearity in the network with these functions, classification of data that is not linearly separable is possible.

Backpropagation Algorithm

Backpropagation is a fundamental algorithm in neural networks along with a loss function and an optimizer it monitors the network by propagating a signal from the output to previous layers.

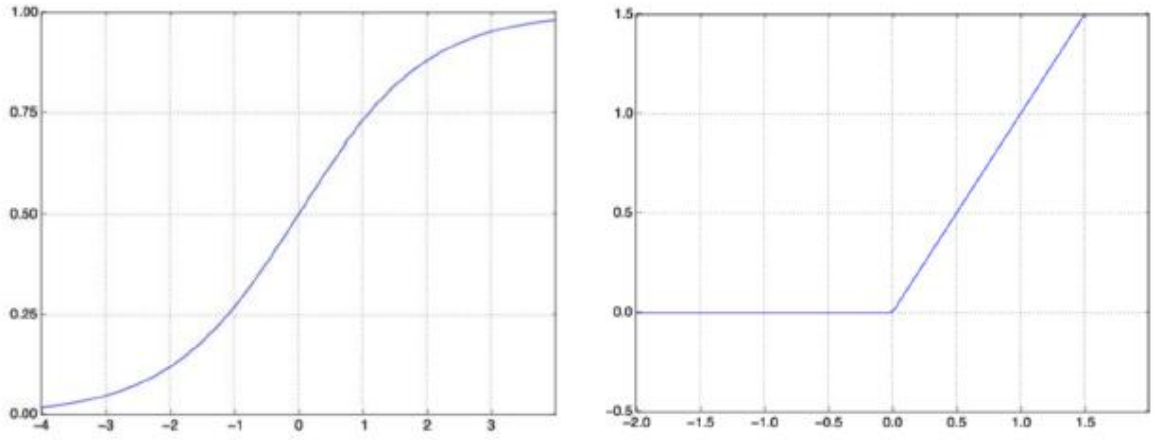


Figure 2.3:Graphical representation of sigmoid and ReLU functions

The loss function is used to control the performance of a model. It evaluates the error value between the expected outcome and target value. The result will be used as feedback for updating the values of the weights in a way to minimize the value of the loss function. The update of the weights is done by the optimizer, which uses the error values to calculate the gradient of the loss function. The error of the hidden layers is calculated as follows:

$$HiddenLayerError_i = \sum \left(Out\ putLayerError_i\ .w_{ij}\right).F'(HiddenlayerOut\ put_i)$$

Where $OutputLayerError_i$, $w_{i\ j}$,$HiddenlayerOutput_i$ are the error of next layer and the corresponding weights, and activation function respectively.

The variation of weights is calculated as a product between the hidden errors and the output of the input node:

$$\Delta w_{ij} = Output_i.HiddenlayerError_j$$

After the variation of weights for all layers are calculated, they are accumulated and the weights are updated. The update of weights is computed as follows:

$$w_{ij} = w_{ij} + \left(\Delta w_{ij}.LearningRate\right)$$

Where w_{ij} are the current weights, Δw_{ij} the accumulated weights and LearningRate is a constant that determine how fast the model will converge to a result.

2.4.2 Convolutional neural networks

Convolutional neural networks are a special type of neural networks especially designed for images recognition that fall into the category of deep learning. The content of this section is based on [19],[20],[21] and[22] . A simple and intuitive way to define a neural network f is to see it as a composition of several functions f_i. In the deep learning world, these successive functions f_i are called layers. A model with L layers takes the form

$$y = f(X; W_1, \dots, W_L)$$

$$= f_L\left(f_{L-1}\left(\dots\left(f_1(X, W1);\dots\right);W_{L-1}\right);W_L\right)$$

$$= \left(f_L(.;W_L) \bigcirc f_{L-1}(.;W_{L-1}) \bigcirc \dots \bigcirc f_1(.;W_1)\right)(X)$$

where the operator $\bigcirc$ denotes the composition of function, i.e. $(g \bigcirc f)(x) = g(f(x))$and w_l is the weight and bias vector of the layer is l. The weight and bias vectors $w_1; \dots ; w_L$ have to be numerically optimized in the learning phase using the learning set LS of N objects, by solving the optimization problem

$$\arg \min_{W_1,\dots,W_L} \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, f_{LS}(X_I; W_1, \dots, W_L)\right) + \frac{1}{2}\lambda \sum_{l=1}^{L-1} W_l^2$$

where the last term of the minimization is a regularization term to penalize too large weights. λ is the weight decay and is an hyperparameter that has to be tuned. It allows to avoid overfitting by controlling complexity since more non linearity is introduced with large weights.

where the last term of the minimization is a regularization term to penalize too large weights. λ is the weight decay and is an hyperparameter that has to be tuned. It allows to avoid overfitting by controlling complexity since more non linearity is introduced with large weights.

Without entering too much into details, the optimization problem is solved through stochastic gradient descent and backpropagation of derivatives. It calculates the gradients of the error with respect to all weights in the network and uses gradient descent to adjust all weights (depending on their contribution to the total error) to minimize the total error:

$$W_l \leftarrow W_l - \eta\left(\frac{\partial \ell\left(y_i, f(x_i; w_1, \ldots, w_L)\right)}{\partial w_l} + \lambda w_l\right)$$

for a sample (xi; yi) ϵ LS chosen randomly and in a cyclic way. The hyperparameter η is the learning rate. This parameter is often adjusted over time during training phase.

In the step decay approach, η is reduced by some factor every few epochs. An epoch means that every samples of the learning set has been seen once. The number of epochs to perform is an other hyperparameter. There is also the batch size b, which is the number of training samples that are used for one gradient update.

Convolution

The goal of convolution layers (CONV) is to extract features from the input image. The convolution is a linear operation that keeps the spatial relationship between pixels by learning image features using small squares of input data. A kernel or filter is a K X K matrix smaller than input images. In general, a convolution layer is always followed by a spatial pooling and a non-linear activation.

The activation map or feature map is the matrix obtained by sliding the kernel over all the image and calculating the dot product between the input and the kernel. The filters thus behaves like feature detectors from the input image. For an input image of size W X H, the size of the feature map is W' x H' X D with

$$W' = \frac{W-K+2P}{S} + 1 \; ; \; H' = \frac{H-K+2P}{S} + 1$$

and where

D (depth) is the number of filters used for convolution, producing thus 2D feature maps.

S (stride) is the number of pixels by which the filter matrix is slide over the input matrix.

P (padding) is the number of pixels by which the input is extended on each border in order to apply the filters to slide elements of the input image matrix.

Figure 2.4. illustrates a convolutional layer with a 5 X 5 filter on a 32 X 32 X 3 image. The layer has a depth D = 5, a stride S = 1 and no padding. The output activation map thus has a size 28 X 28 X 5 since $W' = H' = (32 - 5)/1 + 1 = 28$.

Spatial pooling

The pooling operation (POOL) reduces the quantity of information to keep only the most important one. As a connet, it reduces the amount of parameters and thus the computation time but also limits overfitting. Small square blocks are merged together according to an aggregation function such max or sum. For an input map W X H X D, the pooling operation leads to an output map W' X H' X D with

$$W' = \frac{W-F}{S} + 1 \; ; \; H' = \frac{H-F}{S} + 1$$

And where

F is the spatial extent of pooling square blocks. Typically, F = 2.

S is the stride. Typically, S = 2.

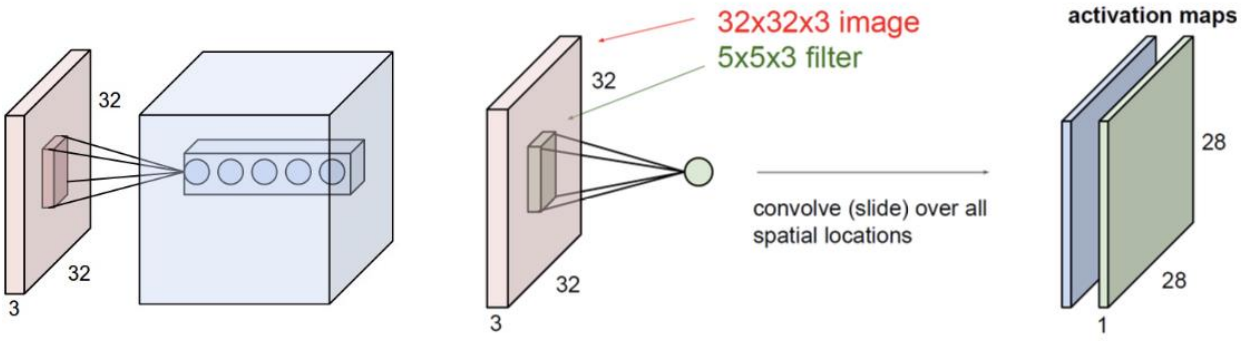


Figure 2.4:A convolutional layer

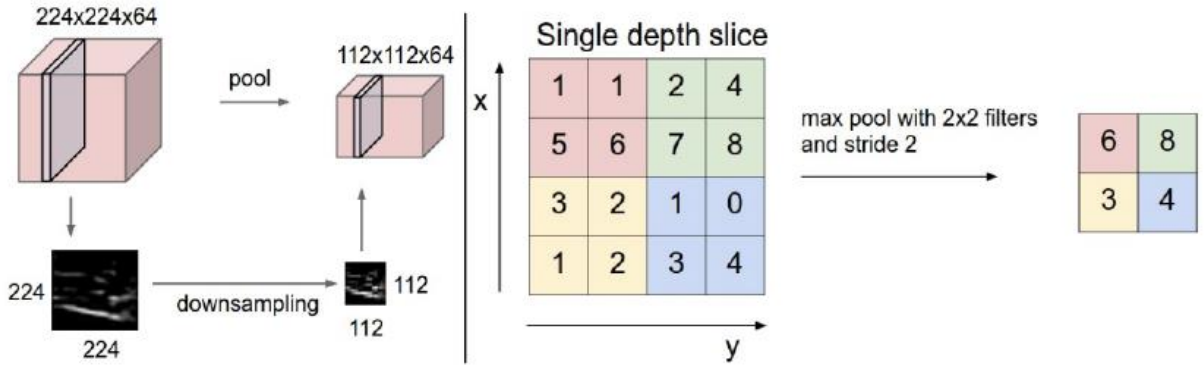


Figure 2.5:Example of max pooling layer

Figure 2.5 shows a max pooling layer with the typical values, F = 2 and S = 2. The output map has the same depth as in input but with and height are divided by2.

Non linearity

Most of real-world input-output relationships are nonlinear. However, the convolution operation is a linear operation. To introduce non linearity in the convolutional neural network, an activation function is employed. In the ReLU (Rectier Linear Unit) one, each element x of the input matrix is substituted by the non linear function f(x) = max(0; x). A smoothed alternative is the SoftPlus activation function f(x) = ln(1 + exp(x)) or the so-called sigmoid function

$$\frac{1}{1+e^{-x}}$$

Fully connected layers

The fully connected layer (FC) is a traditional Multi Layer Perceptron (MLP) where every neuron in the previous layer is connected to every neuron on the next layer. Their activations can then be computed with a matrix multiplication.

It is worth to notice that a fully connected layer can be re-interpreted as a convolution layer. Indeed, for an input of size WX HX D, a fully connected layer of s neurons can be seen as a convolution layer with a kernel of size W X H, S = 1 and P = 0 and an output feature map of size 1 X 1 X s where s is the depth. In fact, the only difference between fully connected and convolutional layers is that the neurons in the convolutional layer are connected only to a local region in the input, and

that many of the neurons in a convolutional layer share parameter. However, the neurons in both layers still compute dot products, so their functional form is identical.

2.4.3 Support Vector Machines

Support Vector Machine (SVM) is linear classifier, powerful and commonly used machine learning method. Support vector machine is supervised learning method that can be used for both object classification and function approximation problems, commonly it is used for classification purpose. There usually exists an infinite number of possible linear classifiers for classification problems, some of them better than the others.

SVM is based on hyperplanes that define specific boundaries. A hyperplane is a separation line between set of objects which belongs to different classes. The optimization objective is to maximize the margin which is explained as the distance between the separating hyperplane and the training data points that are closest to this hyperplane[23]. Those training samples are called support vectors (Fig 2.6).

When defining SVM, a data point is considered as a p-dimensional vector, with p equal to the number of classes. Such points have to be separated with (p-1)-dimensional hyperplane. There are many hyperplanes that might classify the data. The most reasonable choice for finding the best hyperplane is the one that represents the largest separation between the two classes. When maximizing the margin, the decision boundaries tend to have lower generalization error. On the other hand, models with small margins are easier to be overfitted[24].

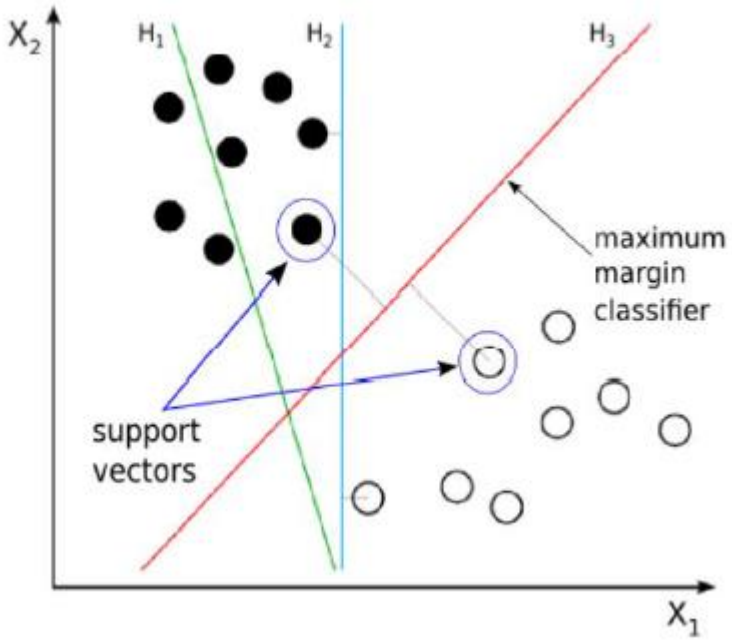


Figure 2.6:Support Vector Machine

In the case of non-linearly separable data, kernel methods could be used. The idea behind these is to transform the current data points to a new space where these instances are linearly separable. Instead of actually transforming the space, a kernel function that has enough of the same properties as the transformation can be used.

2.4.4 Random Forest

Random forest is a part of ensemble learning wherein a set of t classifier for t subsets of training data are computed. The testing phase involves a voting by each of these classifiers to determine the class of the training data and depending on the majority the class of the test data is chosen. Each time for creating a classifier the bootstrap sampling with replacement is used. Thus the data used for creating a classifier each time many contain duplicates. A hyperparameter t is chosen and many classifiers or trees are constructed where each time, random samples are chosen from the training data with replacement for construction of the tree. Thus each tree will be different in some way from the other as each of them have been built using different data. Another important hyperparameter in this case is m which needs to be set at a predefined value which has to quite. m should be chosen to be the square root of total no of features in the data. During construction of the trees, at each node m random features are chosen and find the best split among these m random

features. A continuous features can be chosen multiple times however a categorical features can be chosen only once along that path starting from root to the current node.

After construction of t such trees the same test data through each of these t trees can be run and gather votes from them[25]. An example of random forest is shown in figure 2.7.

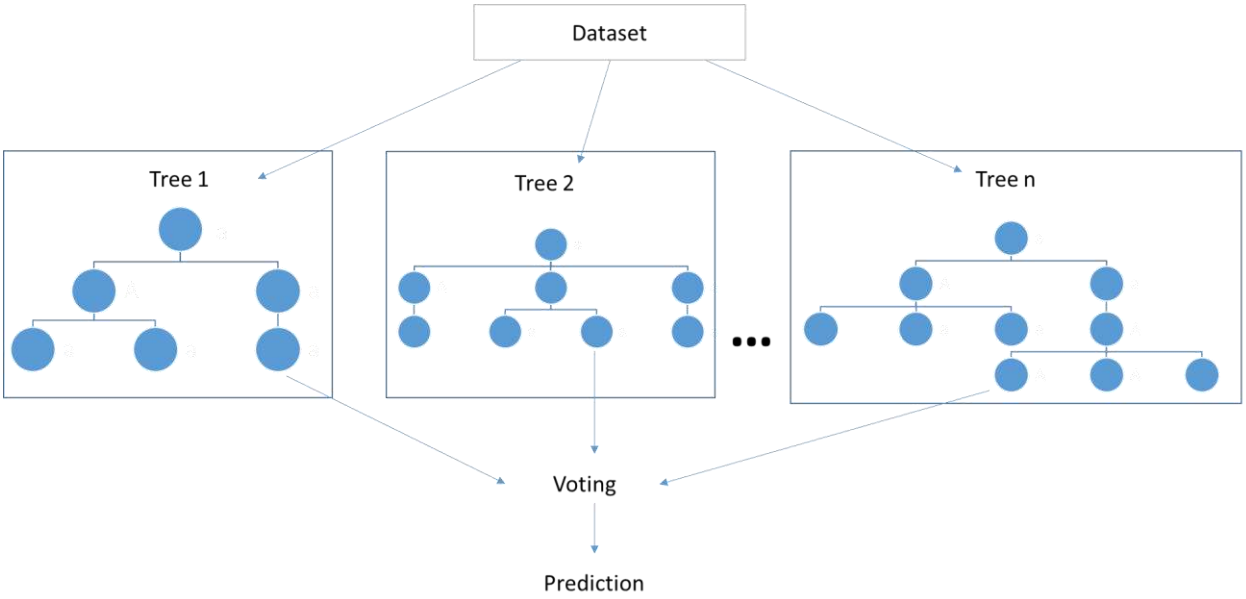


Figure 2.7:The visualization of the random forest algorithm.

2.4.5 K-Nearest Neighbors

K-Nearest Neighbor is a supervised machine learning method that can be used for classification and regression problems. It compares the new samples of the dataset to the existing ones that were kept during the prediction process. In fact, it doesn't make actual learning. That's why it is called a memory-based algorithm. Prediction in K-NN algorithm is easy in the way that for any new sample it looks for K most similar samples based on some distant metrics[26] after that, it assigns a class label to the new sample by a majority voting as it shown in the following figure 2.8. The triangle class label as shown in the figure below has been assigned to the new data sample(?) according to majority nearest neighbors. There are different mechanisms to compare the similarity between the data samples some of them are the following:

Euclidean distance

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

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

$$= \sqrt{\sum_{i=1}^{n} (xi - yi)2}$$

Hamming Distance

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

$$D(x,y) = \sum_{i=1}^{n} xi \; xor \; yi$$

Manhattan Distance

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

$$D(x,y) = \sum_{i=1}^{n} |(xi - yi)|$$

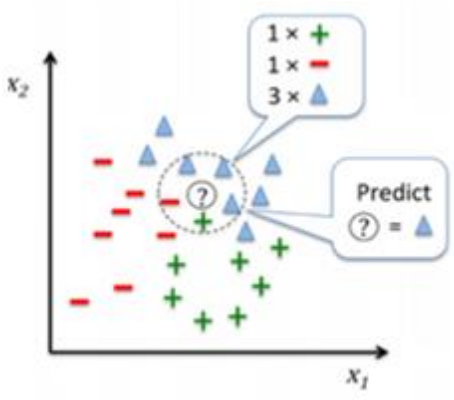


Figure 2.8: K-Nearest Neighbor method

2.4.6 Decision Tree classifier

Decision tree is another supervised machine learning tool used in classification problems to predict the class of an instance. It is a tree-like structure where internal nodes of the decision tree test an attribute of the instance and each subtree indicates the outcome of the attribute split. Leaf nodes represent the class of the instance based on the model of the decision tree. In supervised learning, the model is trained using correctly labeled instances. With decision trees, the training data set and threshold values based on the information gain metric are used to build the attribute tests. The information gain metric chooses an attribute and threshold that maximizes information learned,

which is calculated based on how well the attributes test splits the training data into two subsets each having all the same classification [27].

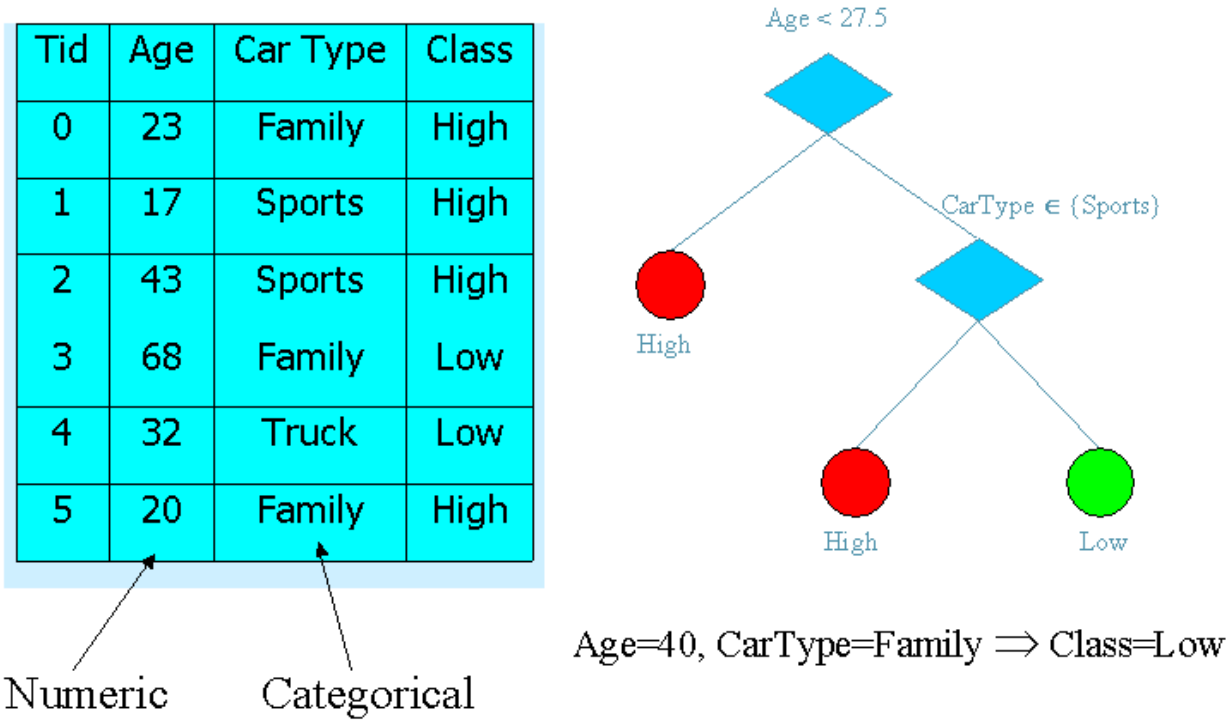


Figure 2.8:Example of a decision tree

2.5 Related Works

The problem of plant disease detection and classification has been an issue for a long time and is of great concern in the agriculture sector for quality management of the crop. Depending on the applications, many systems have been proposed to solve or at least to reduce the problems, by making use of image processing, pattern recognition and some automatic classification tools. Most of the proposed approaches used for plant disease detection and classification follow the steps shown in figure below.

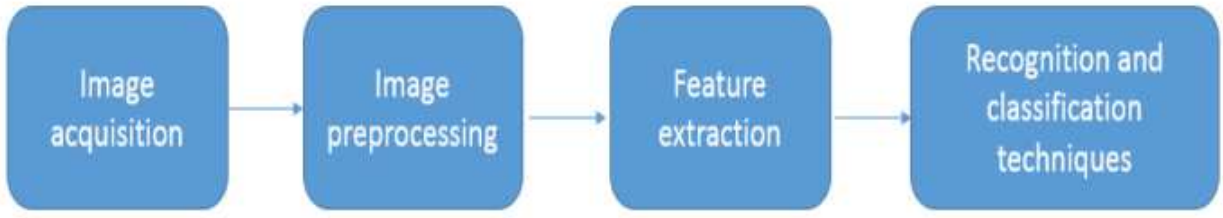


Figure 2.9:Visualization of previously used approach

As shown in the flow chart above the detection and classification process starts by an image acquisition step where different digital devices are used to capture healthy and infected plant images. Then, further analysis is applied to edit the image and prepare it for later treatment, such as image enhancement, segmentation, color space conversion and filtering. In particular, image segmentation methods, like thresholding, are frequently used to detect boundaries in images.

Next the feature extraction step comes, features such as color, shape and texture are extracted from the image. Finally, the classification step is performed. Different classification algorithms are used in the literatures such as neural network[28], support vector machine[8] and so on. Some already proposed systems in the area will be explained as follow.

[29]Presented a paper that uses the SVM to detect pomegranate leaf diseases. Images were captured from the leaves of the pomegranate plant for the experiment. They used the k-means clustering method to segment the healthy and diseased regions of the leaves before presenting the image into feature extraction phase. They then extracted texture features such as mean, standard deviation, entropy, IDM, RMS, variance, smoothness, skewness, kurtosis, contrast, correlation, energy and homogeneity from the diseased regions. Those features then are fed into the SVM, which performs the final categorization. Experimental results showed that the proposed approach has obtained a classification accuracy of 82.3%, using SVM. [30]Proposed a system to detect and classify the diseases of Cercospora leaf spot, common rust, leaf blight in the leaves of maize crop. They downloaded the images from the PlantVillage database and pre-processed the images to enhance the quality for further processing. They then extracted features like bag, statistical histogram and gray level co-occurrence based textural features. The features were individually submitted to the multiclass support vector machine under various configuration for classification. The results show an average accuracy of 83.7% is obtained using the bag of features and an average accuracy of 81.3% is obtained when the features are combined. Finally, they presented the reason for increasing or decreasing in accuracy of classification the specific disease type and healthy leaf.

[28]Proposed to detect soya beans ripeness by using image processing techniques and artificial neural network. They captured images of soya beans leave of various categories by using a digital camera of Sony with 20.1 megapixel and 5x optical zoom ranging from ripe leaf (but not ready for harvest, usually yellow in color), unripe (purely green), unhealthy leaf (that is diseased leaf usually in mix color of green, brown and grey) to fully ripe leaf (purely brown in purely brown in color).

They captured the images under uncontrolled environment, to maintain their natural color without introducing any variation in intensity or brightness to the images. They first converted the image from RGB to HIS. This is followed by a set of structural operations which aims to de-noising but maintaining important and useful features without eliminating them. They then extracted features they had already developed earlier which was described in this paper under feature extraction. Those features are then used to feed the neural network for classification. The results obtained shows an accuracy of 95.7% in classifying different types of soya bean. They concluded by stating that, their approach is better in detecting and classifying soya beans ripeness based on their experimentation.

[31] Aims to develop an image processing based system that could automatically identify flower disease from a given diseased flower mage. The research was done in two phase. In the first phase by using the normal and diseased flower images the author created a knowledge base. During the creation of the knowledge base the author perform image pre-processing and segmentation to identify the region of interest. In this paper, two basic image pre-processing tasks were done. The first task was noise removal and the other was background subtraction. At the first stage the author remove noises that are caused by illumination effects like blurs, light distortion and other noises using image enhancement. Thus, using the image enhancement techniques the author generate either normal or diseased flower image that contains a minimum noise compared with the original image and becomes an input to the next component of the training phase that is image segmentation. Following the image preprocessing the author apply image segmentation so that the image to be represented in to something that is more meaningful and easier to analyze. To perform the image segmentation the author have selected the global Otsu image segmentation technique using Otsu's method. Using the Otsu's method, the author exhaustively search the threshold value to minimize the intra-class variance (the variance within the class) defined as a weighted sum of variance of the two classes. Following the image segmentation the feature extraction is conducted. In this paper, the author uses texture features to represent an image for the identification of the normal and diseased flower images. In order to extract the texture feature of the image the author have used a set of Gabor filters with five spatial frequencies and eight distinct orientations. Since it was difficult to use the response of the Gabor filter as it the author converted the cell matrix in to a new usual matrix by concatenating its entire separate matrix to minimize the computational and storage time. To cut and reduce the dimensionality of the feature vector, the author uses seven

different measures of central tendency and dispersion of the extracted Gabor texture features have been calculated. Finally, the author trained the artificial neural network using the seven input vector features extracted from the individual image and uses eight output vectors that represent eight different classes of disease to represent the knowledge base. The author use 320 datasets of healthy and infected flower images for both training and testing stages. To validate the effectiveness of the proposed methodology, the author uses 40 flower images for each of the eight different classes of flower disease. 85% of the total flower images have been used for training and 15% of the total flower images is used for testing. Experimental results showed that the proposed identification approach has obtained an identification accuracy of 83.3%, using feed forward neural networks. But the symptoms of the diseased flower vary from the beginning to the later time. Here only hand-crafted texture feature extraction is used and instead of hand-crafted texture feature other algorithms could be developed to extract high level feature more accurately

[32]Aims to develop a prototype system to correctly detect and classify the Ethiopian coffee plant diseases with Coffee Leaf Rust (CLR), Coffee Berry Disease (CBD), and Coffee Wilt Disease (CWD) using image processing technique as an alternative or supplemental to the traditional manual method. The methodology that the authors used for identifying Ethiopian coffee plant diseases comprises of 4 phases 1. Pre-processing, 2. Segmentation, 3. Feature Extraction, 4. Identification and Recognition. This process involves several tasks, such as image acquisition and collection, image pre-processing, and image segmentation using Otsu, FCM, K-means, Gaussian distribution and the combinations of K-means and Gaussian distribution., GLCM (Grayscale Color Co-occurrence Matrix) feature extraction and color feature extraction, and Ethiopian coffee plant diseases identification using artificial neural network (ANN), k-Nearest Neighbors (KNN), Naïve and a hybrid of self organizing map (SOM) and Radial basis function (RBF). During the image acquisition the authors captured an RGB color images of coffee leaf using a canon EOS 600d camera, with pixel resolution 1632x1224. Then the authors cropped the images into a smaller image with dimension of 360 X 360 pixels for removing the extra areas. About 910 data samples have been collected. After the image acquisition the author perform an image pre-processing to reduce low frequency background noise, normalize the intensity of the individual particle image, remove reflection and masking portion of image by using median filtering. Following the image pre-processing the authors have used different types of image segmentation technique to find out the best image segmentation like Otsu, FCM, K-means, Gaussian distribution and the

combinations of K-means and Gaussian distribution segmentation. Finally the authors concluded that using combination of K-means and Gaussian distribution segmentation are better than other. After selecting the best image segmentation the authors extract features GLCM and color feature extraction techniques for the purpose of improving the identification and misclassification rate. The feature sets that were extracted from Ethiopian coffee leave image produces very big matrices, in order to cut and reduce the size of matrices they used PCA (principal component analysis) is applied finally GA(Genetic Algorithm) is used for feature selection.

The experimental results indicate that the combined segmentation technique can significantly support accurate identification of Ethiopian coffee leaf diseases. The authors conclude that the combined classifiers of RBF (Radial basis function) and SOM (Self organizing map) together with a combination of k-means and Gaussian distribution perform well and could successfully detect and classify the tested diseases with a precision of around 92.10%. Here Hand-crafted color feature extraction is employed and the classifier which is not that efficient in detecting and classifying the plant disease because the color of a plant disease changes with season, so using color as a feature is not an effective method.

[33]Proposed a leaf disease detection model based on deep Convolutional Neural Network for Soybean Plant. This work was able to classify 4 classes consisting of 3 disease and 1 healthy images using a dataset of 12673 images form Plant Village dataset. Firstly original image is downloaded and then that image is being used for processing for quality enhancement. They then converted the image size into 256X256 size to have uniform vector size and converted into gray color image. They then have used a convolutional neural network that consists of three convolutional layer, kernels of size 3x3, followed by fully connected perceptron and they gave the output of the last layer to the softmax function. Finally, the model is trained using adaptive moment estimation (Adam) with batch size of 100 for 1000 epochs. The experimental result shows that an average accuracy of 93.21% is achieved.

[34] proposed an algorithm that is tested on main five diseases on the plants; they are: Early Scorch, Cottony mold, Ashen Mold, Late scorch, tiny whiteness. Initially the RGB image is acquired then a color transformation structure for the acquired RGB leaf image is created. After that color values in RGB converted to the space specified in the color transformation structure. In the next step, the segmentation is done by using K-means clustering technique. After that the mostly green pixels

are masked. Further the pixels with zero green, red and blue values and the pixels on the boundaries of the infected object were completely removed. Then the infected cluster was converted into HIS format from RGB format. In the next step, for each pixel map of the image for only HIS images the SGDM matrices were generated. Finally, the extracted texture, color, and shape were recognized through neural network. The results show that the proposed system can successfully detect and classify the diseases with a precision between 83% and 94%. However, when the background contains other leaves or plants, the Otsu segmentation method might not be appropriate.

Even though different methods have achieved good classification results in identifying and recognizing some of the diseases, they suffer from some limitations. The implementation still lacks in accuracy of result in some cases. More optimization is needed. Speed of detection needs to be faster. In addition, very few diseases have been covered and needs to be extended to cover more diseases. Most of the methods will fail to effectively extract the best region of interest from its background which will lead to unreliable results. Also, some disease symptoms do not have well represented edges and they could gradually fade into healthy tissue. This may disturb solutions like color based methods and thresholding. Furthermore, a number of the methods rely on hand-crafted features such as color histograms, texture features, and shape features that requires expensive work and depends on expert knowledge. However, these methods do not generalize well and they are not effective when dealing with a large amount of data that could contain significant varieties. Hence, in this research work the stated limitation can be removed using the application of convolution neural network (CNN) as hand-crafted features are avoided. Also, the presence of a large amount of dataset and powerful feature of CNN is a suitable candidate for the current application.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

In this chapter the model design and implementation of the proposed approach and conventional machine learning methods for the automatic plant disease detection and classification along with the dataset, and with some findings from the design process will be discussed. A discussion to the methods, techniques, or approaches being used while designing and implementing the proposed approach and an overview of the image processing being applied on the dataset prior to model training will be given.

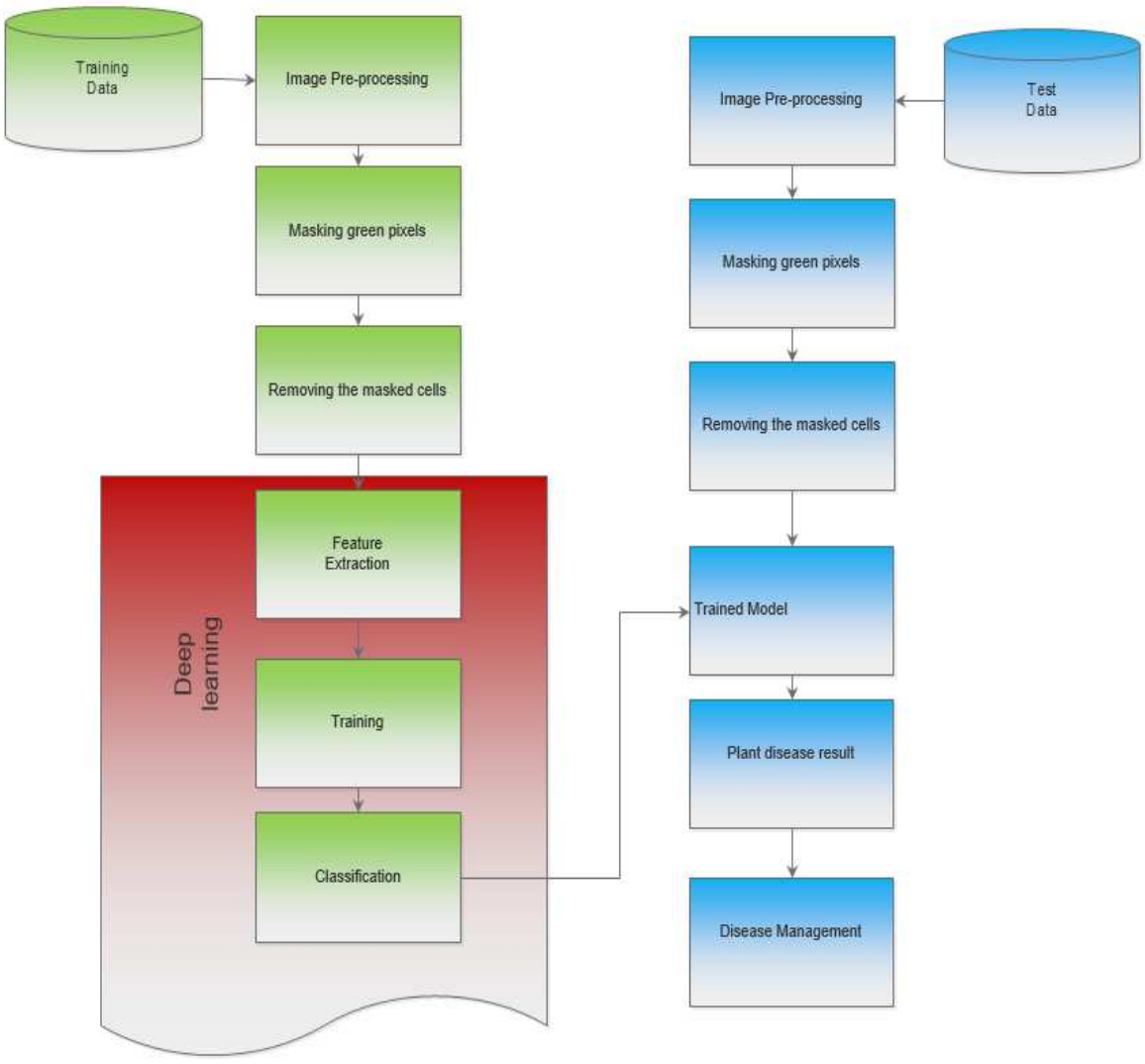


Figure 3.5: Block diagram summarizing the proposed approach for plant disease detection and classification

3.1 Dataset

A public dataset of 6004 images of diseased and healthy plants of the selected plants were downloaded from the plant village.org website to apply the proposed methodology and conventional machine learning models. The plant village org website contains thousands of images of healthy and diseased crop plants that are open and available on the web for more information refer [35]. The dataset consists of 152 images of healthy potato, 1000 images of early blight, 1000 images of late blight, 1162 images of healthy maize, 513 images of gray leaf spot, 1192 images of common rust, and 985 images northern leaf blight. Table 1. shows the distribution of the original dataset that have been used for training, validating, and testing the methodology. Furthermore, the datasets have been partitioned in training, validating, and testing. The 80% data, 4801 datasets have been used for training, and 20%, which consist of 959, 1202 have been used for validation and testing respectively.

The dataset contains a variety of different image including image with various resolutions, sample at early medium, and last infection status. Object surrounding the plant have been also downloaded for providing a good case for training and testing the proposed methodology. Figure 3.1 gives an example of raw image from the dataset downloaded.

3.2 Data augmentation

It is important to note that as the dataset of 4801 images, in general, is too small for properly training the machine learning techniques especially for CNN. Data augmentation technique have been used to generate a larger dataset and more varied for training. However, during validation with testing data, no data augmentation is applied on the test data to provide fair results. A random combination of 5 augmented images is created of each category image. Data augmentation makes the 6004-image dataset into a 48010 images dataset, which means there are over 1000 images for each classes. ImageGeneraor function is the python library which was used for experimenting the data augmentation technique, that could be configured to perform the random transformations and the normalization of input images as needed. In order to augment the input image the following parameters are applied:

a) Horizontally flipping image with 50% probability;
b) Randomly cropping image;
c) Strengthening or weakening the contrast in image with 50:50 probability;

d) Rotating image with 50% probability

Disease Category	#of Initial images	# of images from the initial dataset used for training	#of images from the initial dataset used for validation	# of images from the initial dataset used for testing
Healthy potato	152	121	24	31
Early blight	1000	800	160	200
Late blight	1000	800	160	200
Healthy maize	1162	930	186	232
Gray leaf spot	513	410	82	103
Common rust	1192	952	190	239
Northern Leaf Blight	985	788	157	197
Total	6004	4801	959	1202

Table 3.1: Dataset distribution of each disease and healthy category

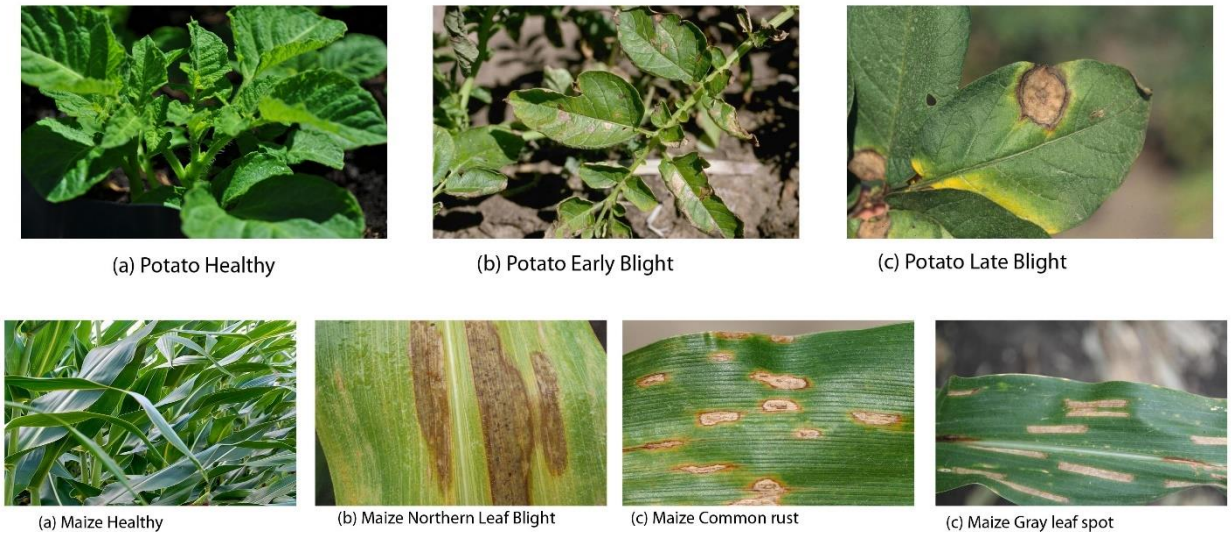


Figure 3.2: Samples of healthy and diseased potato and maize images from the dataset

Disease Category	#of Initial images	# of images from the initial and augmented dataset used for training	# of images from the initial and augmented dataset used for validation	# of images from the initial and augmented dataset used for testing
Healthy potato	152	1210	242	31
Early blight	1000	8000	1600	200
Late blight	1000	8000	1600	200
Healthy maize	1162	9300	1860	232
Gray leaf spot	513	4100	820	103
Common rust	1192	9520	1904	239
Northern Leaf Blight	985	7880	1576	197
Total	6004	48010	9602	1202

Table 3.2: Dataset distribution of each disease and healthy category after augmentation

3.3 Preprocessing

This is the initial phase of this proposed approach. Noise refers to variation in intensity or brightness in an image[36]. It might get added during the acquisition of the images, which is introduced by camera flash, change in illumination, noise background and presence of vein in the plant leaf. The purpose of image pre-processing is to remove noise from the image for enhancing the quality of the input image, getting maximum accurate result and good efficiency. Therefore, to reduce the computational burden and improve the storage efficiency in the later processing, the original size of the images (approx. 800x600 pixels on average) is reduced to 128 by 128 pixels. The optimal image size was carefully selected by testing different sizes not to compromise the performance of plant disease detection and classification of the automatic system. Image resizing is also important to have uniform image sizes. The downloaded images in their red, green, blue (RGB) color model are converted to hue, saturation, value (HSI) color model because according to [37], HSI model is an ideal tool used in image processing and when humans view a color, they describe it in its color form, thus, the model becomes an intuitive form of description for humans.. A median filter is applied to reduce the different noises. The algorithm used is described below:

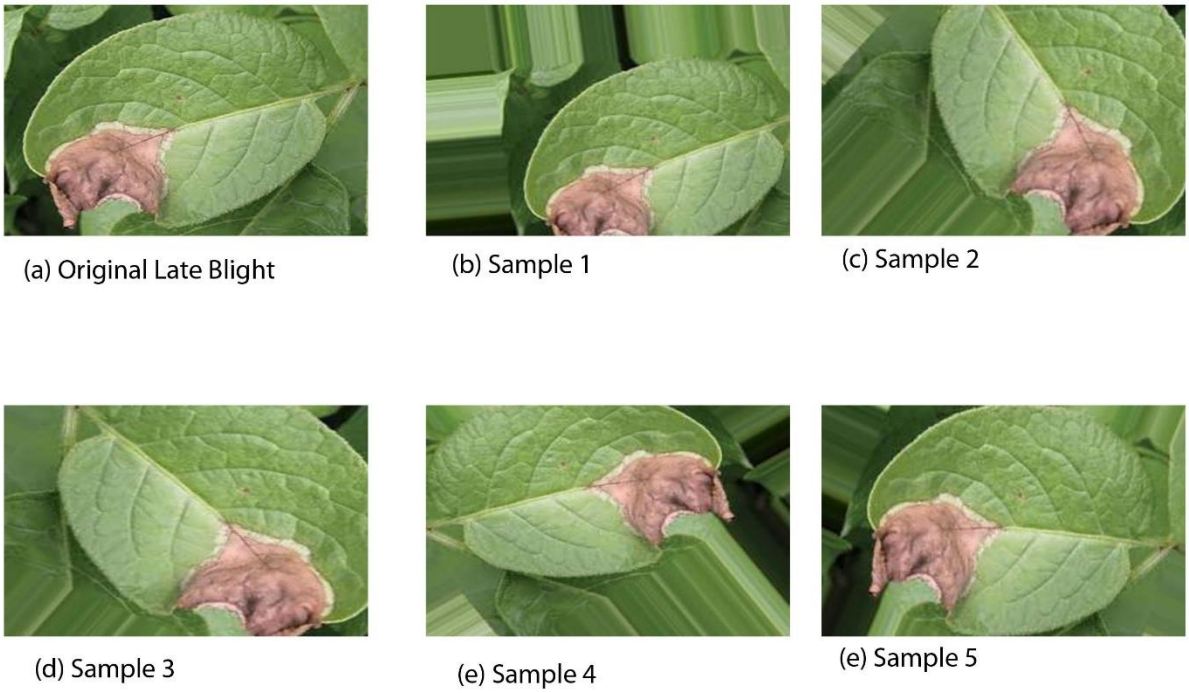


Figure 3.3 Single image of potato infected by late blight being used to generate augmented data for each machine learning to learn upon.

Algorithm: Preprocessing

Input: RGB image

Output: Noise removed and enhanced image

Start

Read the RGB image

Convert the RGB image to image in the HSI color model

Median filter the components (HSI) of the converted image

Combine the hue and saturation component with the filtered value

Convert the image to RGB image

Save the result

End

3.4 Masking the green part

This the second step of this proposed approach. In this step, the mostly green part of the input image was identified using the masking algorithm shown below to locate the green region and distinguish the diseased part from the healthy part. After that, based on the identified region that

is computed for these pixels, the mostly green pixels are masked as follows: if the green component of the pixel intensity is less than the identified region, the red, green, and blue components of this pixel is assigned to a value of zero. This is done in sense that the green colored pixels mostly represent the healthy areas of the leaf and they do not add any valuable weight to disease identification and furthermore this significantly reduces the processing time. In this thesis the masking operation is done by opencv. Figure 3.3 shows the same image after passing the masking the green steps. The detail algorithm for masking the green part is explained as follow:

Algorithm: Masking

Input: RGB image
Output: Green Masked image
Start

Check to ensure that the user has installed opencv in python.

Load the image into working space (with full file location)

Get the dimensions of the image

Convert the image to HSV color space

Compute the green pixel so that pixels in range (36,25,25) to (60.255,255) are set to white and those outside the range are set to black.

Mask the H, S, and V images

End

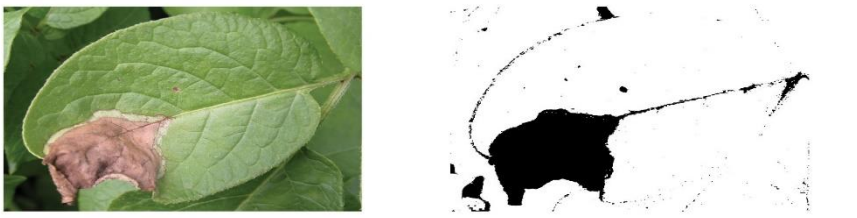


Figure3.4: Images showing masking the green part

3.5 Removing the green part

Removing the green part of the image was carried out after the masking operation. The pixels with zeros red, green, blue components as well as pixels on the boundaries of infected region have been completely removed. This is helpful as it gives more accurate disease detection and classification and significantly reduces the processing time. And then the diseased part is converted from RGB to gray color format. The result of this step is shown below in figure 3.4. The algorithm that was used to remove the green part of the input image is defined as follow:

Algorithm: Green Removal

Input: RGB and Masked image

Output: Green removed image

Start

Check to ensure that the user has installed opencv in python.

Load the original input image and its corresponding masked version into working space

Get the dimensions of each image

Set a loop up to the row of the masked image

Set a inner loop up to the column of the masked image

Check the pixel value of the current pixel is less than or equal to

If smaller than or equal to 0 then iterate to the next pixel

If greater then assign the current pixel value by zero

Save the green removed image after the execution of the loop

End

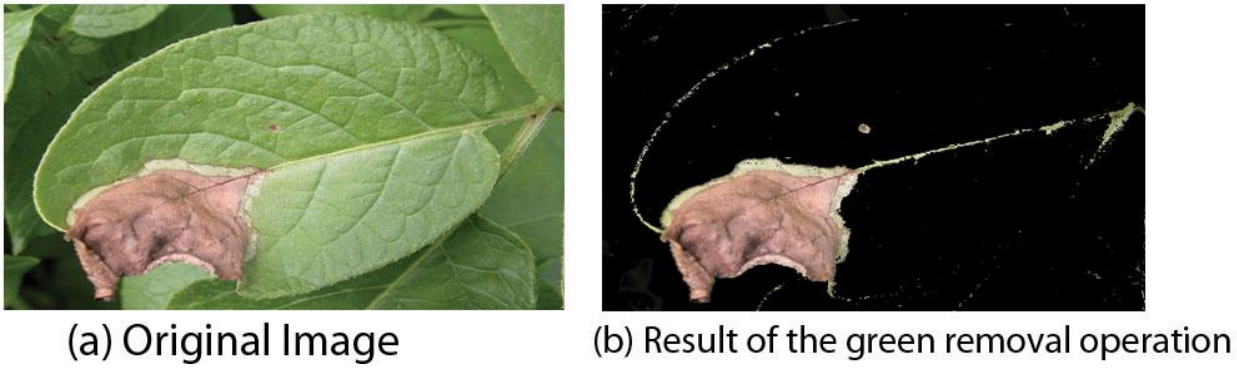
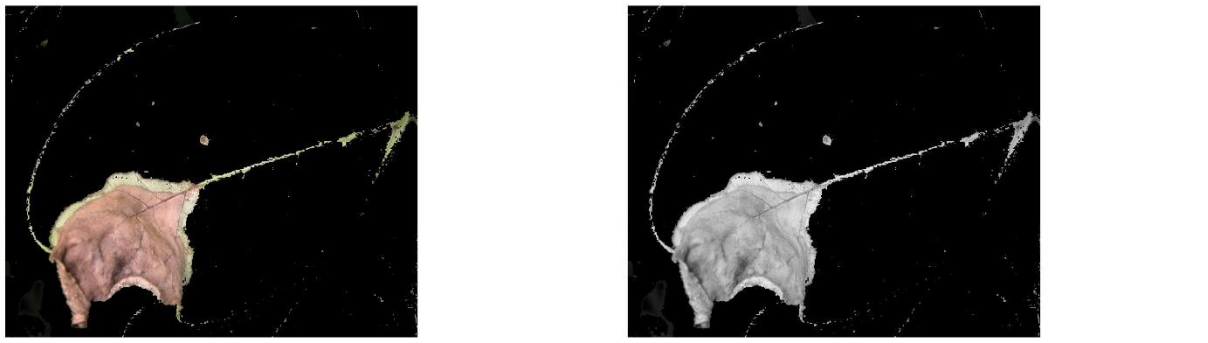


Figure 3.4: Images showing removing the green part of the image

3.6 Grey Scale Conversion

After the removal of the green part all the images are presented in an RGB format, these have to be first converted from RGB format to a grey scale format, which is usually performed by matching the luminance of the color image. A greyscale image is an image that only encodes intensity information. The grey scale conversion was performed using opencv cvtColor module. An example image is shown in Figure 3.6, created using the opencv cvtColor module.



3.7 Feature extraction

After masking and removing the green part of the input images have been completed, the feature extraction step is carried out. It extracts information from the input image to serve as an input into the conventional machine learning method. According to[38] feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input image will be transformed into a reduced representation set of features (also named features

vector). If the features extracted are carefully chosen, it is expected that the features set will perform the desired task using the reduced representation instead of the full-size input.

For an image, a feature can be defined as measures describing dataset properties and characteristics. These features play a fundamental role in classification. The features are necessary for differentiating one category from another. The method has to be used for describing the objects so that features of interest are highlighted. Hence, in this thesis texture features are selected as they have been extensively used in image processing and pattern recognition by different researchers to extract features.

3.7.1 Color Co-occurrence Matrix

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

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

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

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

The offset value $(\Delta x, \Delta y)$ is calculated by the spatial direction. If the direction is

$$0°, \text{ then } \Delta x = 0,\ \Delta y = 1$$

$$45°, \text{ then } \Delta x = 1,\ \Delta y = 1$$

$$90°, \text{ then } \Delta x = 1,\ \Delta y = 0$$

$$135°, \text{ then } \Delta x = -1,\ \Delta y = 1$$

Let's take an example of an $5x5$ image and the GLCM matrix is calculated by following procedure:

$$\begin{bmatrix} 32 & 230 & 190 & 121 & 85 \\ 35 & 250 & 102 & 190 & 128 \\ 55 & 120 & 220 & 240 & 181 \\ 133 & 50 & 160 & 15 & 189 \\ 190 & 100 & 80 & 25 & 255 \end{bmatrix}$$

The matrix size is $5x5$. So, the segment size will be $^{255}/_{5} = 51$ and all values of the above matrix will be replaced by below:

$$0 - 51 \rightarrow 0$$
$$52 - 103 \rightarrow 1$$
$$104 - 154 \rightarrow 2$$
$$155 - 205 \rightarrow 3$$
$$206 - 255 \rightarrow 4$$

By applying above mapping the resultant matrix is given below:

$$\begin{bmatrix} 0 & 4 & 3 & 2 & 1 \\ 0 & 4 & 1 & 3 & 2 \\ 1 & 2 & 4 & 4 & 3 \\ 2 & 0 & 2 & 0 & 3 \\ 3 & 1 & 1 & 0 & 4 \end{bmatrix}$$

When the direction is 0° and the offset value is 1, then the matrix which will be found is given below:

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 3 \\ 1 & 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 \end{bmatrix}$$

The transpose matrix of the above matrix is given below:

$$\begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 & 2 \\ 3 & 0 & 1 & 0 & 1 \end{bmatrix}$$

After Adding the above two matrices, the following matrix is obtained:

$$\begin{bmatrix} 0 & 1 & 3 & 1 & 3 \\ 1 & 2 & 2 & 2 & 1 \\ 3 & 2 & 0 & 2 & 1 \\ 1 & 2 & 2 & 0 & 2 \\ 3 & 1 & 1 & 2 & 2 \end{bmatrix}$$

Now, the determinant of this matrix = 40. By normalizing the matrix, the following resultant matrix is obtained:

$$\begin{bmatrix} 0 & 0.025 & 0.075 & 0.025 & 0.075 \\ 0.025 & 0.050 & 0.050 & 0.050 & 0.025 \\ 0.075 & 0.050 & 0 & 0.050 & 0.025 \\ 0.025 & 0.050 & 0.050 & 0 & 0.050 \\ 0.075 & 0.025 & 0.025 & 0.050 & 0.050 \end{bmatrix}$$

This is the resultant GLCM matrix when the offset value is 1 and rotation is 0°.

3.7.2 Textural Features

Notations:

$p(i,j)$	(i,j)th entry in the normalized gray level co-occurrence matrix
$p_x(i)$	ith entry in the marginal probability matrix obtained by summing the rows of $p(i,j)$
N_g	number of distinct gray levels in the quantized image.

According to [39] the following list of texture features are identified:

1) Angular Second Moment:

$$f_1 = \sum_i \sum_j (p(i,j)^2)$$

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

If all pixels have same gray level I=k,

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

$ASM = 1$

If all pixels have different gray level,

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

2) Contrast:

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

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

3) Correlation:

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

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

4) Sum of squares: Variance

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

5) Inverse Difference Moment(Homogeneity)

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

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

6) Sum Average

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

7) Sum Variance

$$f_6 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$$

8) Sum Entropy

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

9) Entropy

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

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

10) Difference Variance

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

11) Difference Entropy

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

12) Information Measure of Correction

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

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

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

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

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

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

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

13) Maximal Correction Coefficient(Energy)

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

Where,

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

3.7.3 Feature Selection

The main goal of feature selection is to find those features which are not well correlated. Not all features are necessary for classifying the different diseases. In this study from all 13 features only five of them have been found unique. For selecting those features which are not well correlated a

method called subset choosing method is applied. As stated in [43] this method enables to find the best model as can be shown in the algorithm below.

The algorithm is:

Algorithm: Feature Selection

Input: Total feature
Output: Selected features
Start

Let M_0 denote the null model which contains no predictors, this model simply predicts the sample mean of each observation

For m=1,2,..n.

Fit all $\binom{n}{m}$models that contain exactly m features

Pick the best among these $\binom{n}{m}$ models, and call it M_n. Here the best is defined as having the smallest RSS, or an equivalent measure

Select the single best model among M_0,M_1,...,M_n using cross validated prediction error, Cp, BIC, adjusted R2 or any other method.

End

There are 2^{14} possible subsets among them the subset which is the possible best subset. Following are the texture features which will be considered for the classification.

1) Homogeneity
2) Angular Second Moment(ASM)
3) Energy
4) Information Measure of Correlation 1
5) Information Measure of Correlation 2

During the training the input is normalized by subtracting the total mean and dividing by the total standard deviation estimated on a few hundred samples before training. After this process was completed, the images were available for training and testing on the various different models that

were highlighted as options to complete this task, broken up between the proposed approach and conventional machine learning models.

Dataset Split

In training phase, before each image is fed in to the training models the training dataset are divided in to training and validation depending on the split ratio using train_test_split of the sklearn package.

Classification Techniques

Numerous different machine learning methods have been developed that can be applied to almost any data problem. Accuracy, training time, number of parameters and features are some of the points that have to be considered when choosing the best classifier for each specific use case. Even the simplest methods can on some occasions outperform a more complex models. As explained in chapter two of the machine learning section six of the most commonly used algorithms are chosen. In the next section, brief implementation overview of the selected machine learning methods will be provided.

Hyper-parameter Ranges

The distribution of all hyper-parameters that were used in this experiment are described in the following table 3.3.

Machine learning Method	Parameters	Range
K-NN	k	[8]
Support vector machine	C Gamma Kernel Degree	[10^{-3}] [10^{-3}] poly 2
Random Forest	n_estimators max_features min_samples_leaf max_depth min_samples_split bootstrap, criterion	[1000] [50] [20] [32] [16] [True] ["entropy"]
Decision Tree	max_features min_samples_leaf max_depth	[5] [21] [51]

	min_samples_split criterion	[100] ["entropy"]

Figure 3.3: Hyper-parameters Ranges of the applied machine learning algorithms.

Decision Tree Classifier

The Decision Tree Classifier model is implemented in python sklearn library. The sklearn.tree library is used to import the DecisionTreeclassifer class. The object of the class is created and passed the arguments specified in the above table.

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

K-NN Classifier

The K-NN classifier algorithm is implemented in python sklearn library. The sklearn.neighbors library is used to import the KneighborsClassifier class. The object of the class is created and passed the arguments specified in the table above.

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

Random Forest Classifier

The random forest classifier algorithm is implemented in python sklearn libray. The sklearn.ensemble library is used to import the RandomForestClassifier class. For the classification with random forest, an object of the class is created and passed the arguments mentioned in the table above.

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

Support vector machine

The support vector machine classifier algorithm is implemented in python sklearn libray. The sklearn.svm library is used to import the svc class. For the classification with SVM, an object of the class is created and passed the arguments mentioned in the table above.

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

Fully connected Multilayer Perceptron

The network has three hidden layer consisting of 12,11,10 neurons respectively. The output has 7 neurons with a float value categorizing the image based on the disease type. The neural network is visualized on figure 3.7.

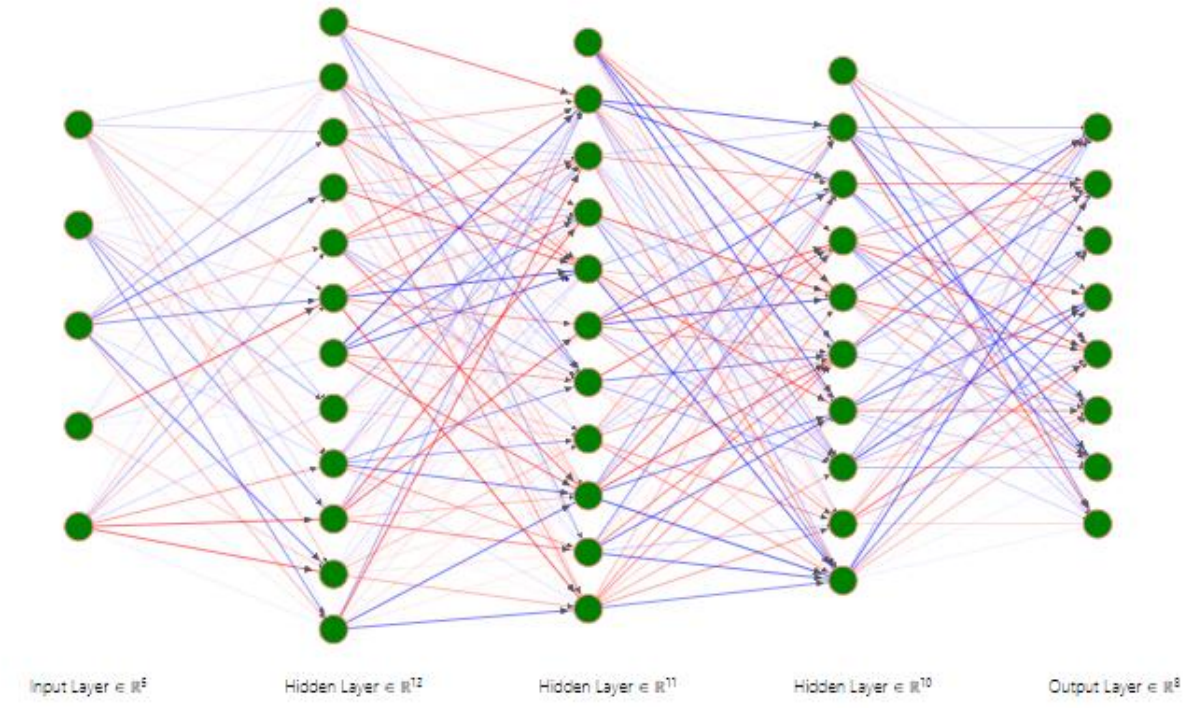


Figure 3.7: Structure of fully connected multilayer perceptron used.

Like SVM, DT, RF, and KNN, the FCMLP have no the potential to extract features from the raw input image. To extract and select the features two different preprocessing functions using the OpenCV package were defined. First one is called image to feature vector, to resize the image into 128x128 , make the RGB images into grey-scale images and then flatten the image into a list of row pixel. Through trial-and-error it was observed that larger input image with the larger amount

of input neurons doesn't increase accuracy, but takes a heavy toll on the processing power of the neural network.

The second one is called feature selection , to select features which are well correlated. The training data is fed to neural network inputs (5 values). The neural network is trained for varied number of epochs after which it is tested with test dataset.

The created neural network is trained with a back-propagation trainer with the classification dataset created previously. Back-propagation trainer trains the parameters of a module according to a supervised dataset by back-propagating the errors. The input parameters to the trainer are very important as they decide the effectiveness of training the neural network.

The trainer takes as input the momentum, learning rate and weight decay. The parameters affect the trainer as follows. The learning rate gives the ratio of which parameters are changed into the direction of the gradient. The learning rate decreases by a factor which is used to multiply the learning rate after each training step. The parameters are also adjusted with respect to momentum, which is the ratio by which the gradient of the last timestep is used. Weight decay corresponds to the weight decay rate, where 0 is no weight decay at all. The trainer is run for 100 epochs as through testing it was deemed that after 100 epochs of training the error rate doesn't increase greatly anymore.

The neural network uses sigmoid layer as it is best fitted for multi-classification in a logistic regression model. Added value is that the probabilities sum will be 1. Once the neural network is trained, it is tested with a dataset of 1201 images of health and diseased maize and potato plants and the accuracy and error rate is recorded. The test data result will be explained in the result and discussion section.

Designing Convolution Neural Network (CNN)

The network architecture of the designed convolutional network is shown in figure 8.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 128, 128, 32)	320
activation_1 (Activation)	(None, 128, 128, 32)	0
batch_normalization_1 (Batch	(None, 128, 128, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 42, 42, 32)	0
dropout_1 (Dropout)	(None, 42, 42, 32)	0

conv2d_2 (Conv2D)	(None, 42, 42, 64)	18496
activation_2 (Activation)	(None, 42, 42, 64)	0
batch_normalization_2 (Batch	(None, 42, 42, 64)	256
conv2d_3 (Conv2D)	(None, 42, 42, 64)	36928
activation_3 (Activation)	(None, 42, 42, 64)	0
batch_normalization_3 (Batch	(None, 42, 42, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 21, 21, 64)	0
dropout_2 (Dropout)	(None, 21, 21, 64)	0
conv2d_4 (Conv2D)	(None, 21, 21, 128)	73856
activation_4 (Activation)	(None, 21, 21, 128)	0
batch_normalization_4 (Batch	(None, 21, 21, 128)	512
conv2d_5 (Conv2D)	(None, 21, 21, 128)	147584
activation_5 (Activation)	(None, 21, 21, 128)	0
batch_normalization_5 (Batch	(None, 21, 21, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 10, 10, 128)	0
dropout_3 (Dropout)	(None, 10, 10, 128)	0
conv2d_6 (Conv2D)	(None, 10, 10, 256)	295168
activation_6 (Activation)	(None, 10, 10, 256)	0
batch_normalization_6 (Batch	(None, 10, 10, 256)	1024
conv2d_7 (Conv2D)	(None, 10, 10, 256)	590080
activation_7 (Activation)	(None, 10, 10, 256)	0
batch_normalization_7 (Batch	(None, 10, 10, 256)	1024
max_pooling2d_4 (MaxPooling2	(None, 5, 5, 256)	0

dropout_4 (Dropout)	(None, 5, 5, 256)	0
flatten_1 (Flatten)	(None, 6400)	0
dense_1 (Dense)	(None, 1024)	6554624
activation_8 (Activation)	(None, 1024)	0
batch_normalization_8 (Batch	(None, 1024)	4096
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 7)	7175
activation_9 (Activation)	(None, 7)	0

Total params: 7,732,039
Trainable params: 7,728,135
Non-trainable params: 3,904

Table 3.4. Shown here is the architecture and positions of the various layers in the designed CNN.

The convolutional neural network consists of seven sets of convolution and pooling. After the seven sets of steps are completed, the resulting values are given to the hidden layer and outputs provide.

Training Model

An input image with a different background, light intensity, orientation and different shape and size is downloaded from the internet so as to improve accuracy and decrease false classification. In the initial training the model has been trained with the 7 classes classification dataset in 50 epochs , using stochastic gradient descent with a starting learning of 10^{-5},weight decay of 0.0003 and a momentum of 0.3 and input image of 512X512. Secondly, the model has been trained with the same classes classification dataset in 80 epochs , using stochastic gradient descent with a starting learning of 10^{-5},weight decay of 0.0005 and a momentum of 0.5 and then the input image scaled down to 256X256.of 512X512. Thirdly, the model has been trained with the same classes classification dataset in 100 epochs , using stochastic gradient descent with a starting learning of 10^{-5},weight decay of 0.0005 and a momentum of 0.5 and then finally the input image is scaled down to 128X128.of 512X512. After the training, the classifier achieves a training accuracy of 89.5%, 93.3% and 98.5% initially, secondly, and thirdly respectively.

Figure 3.8 compares the two most different models for 100 epochs of training. The CNN with augmented data performs better than MLP by far when trained with augmented data. The CNN reaches an a accuracy of 98-99.59% upon validation. The MLP models don't manage to get over 90% accuracy even after 60 epochs of training. Difference between accuracy of MLP and convolutional NN is at some points as high as 10%.

During training, the loss value of each model is recorded. As indicated in figure 3.9 below the loss value of CNN is 0.00001, which is lower than MLP.

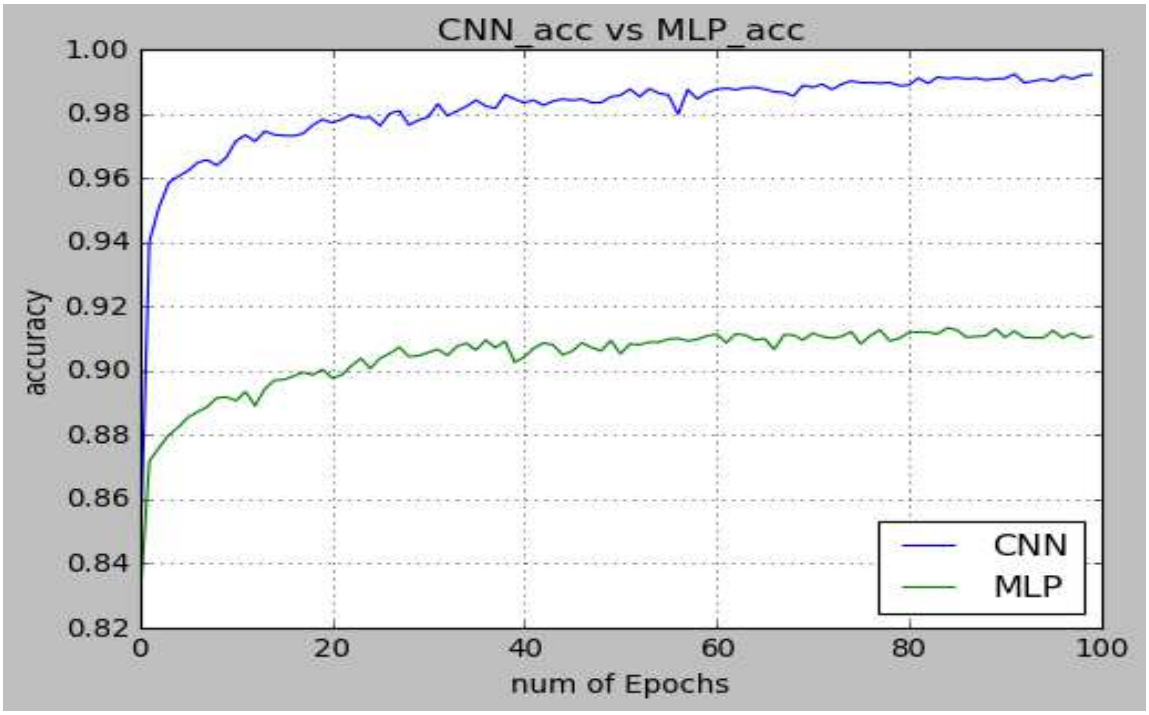


Figure 3.8:10Accuracy of MLP and CNN based on validation after each epoch of training 100 epochs

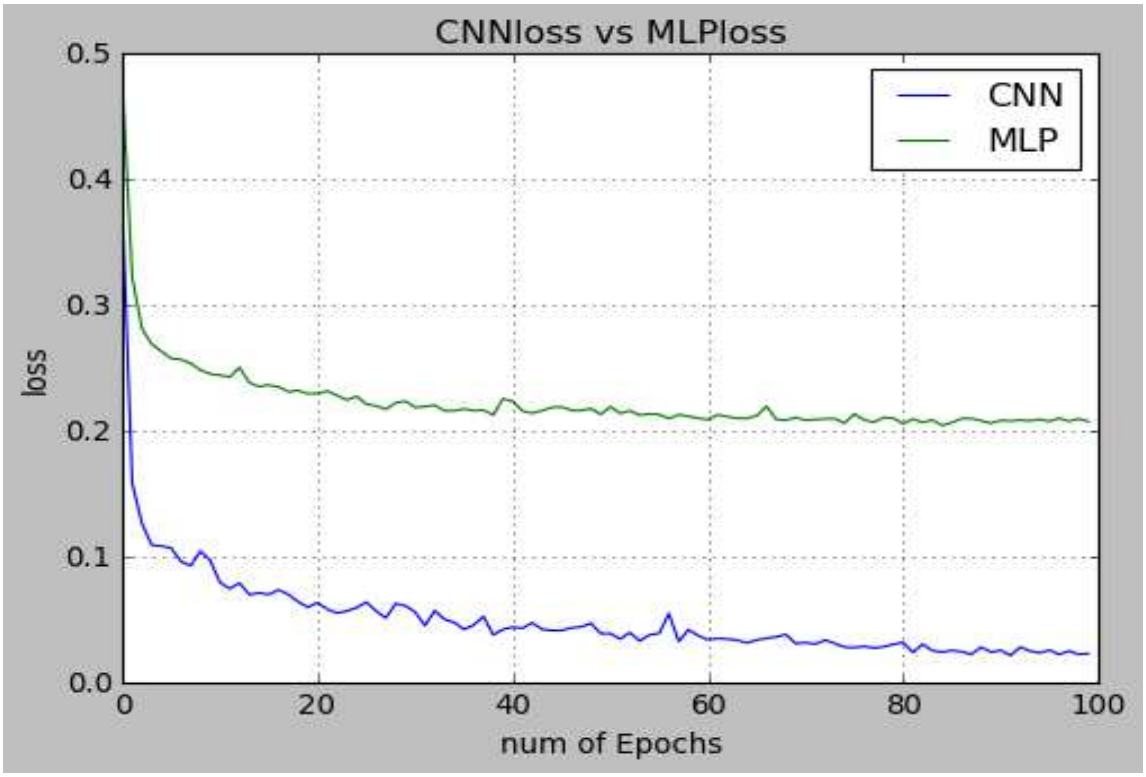


Figure 3.9: Loss value of MLP and CNN based on validation after each epoch of training 100 epochs

3.8 Disease Management techniques

Based on biological characteristics all plant disease along with their management approach have been studied. However, ten thousands of plants are available and remembering the right procedure for mitigating the disease occurred in the plant is challenging one. Once the disease is detected and classified this approach provides the steps to mitigate the disease. Result after the disease management techniques is shown below.

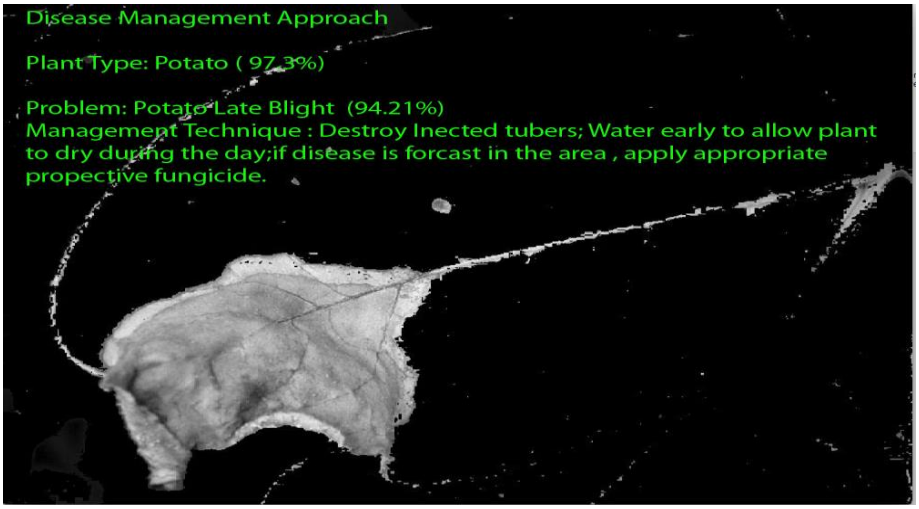


Figure 3.10: Disease Management Approach

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Introduction

For testing and comparing the proposed approach with the different machine learning classifier a total of 31 images of healthy potato, 200 images of early blight, 200 images of late blight, 232 images of healthy maize, 103 images of gray leaf spot, 239 images of common rust, and 197 images northern leaf blight are used. For the conventional machine learning techniques before testing each image is passed through the image pre-processing, masking, and removing the green part of the image, feature extraction and selection phase whereas, for the deep learning each image is passed only through the image pre-processing, masking and removing the green part of the image phases.

In this chapter the resources required for the research, the performance metrics, the expermental results and discussion about the result will be presented.

4.2 Resources used for the research

The experiments and related analysis processes are done on a computer with Intel (R) Core (TM) i7-5500U CPU having 4 cores with each core having 2.4GHz Speed, 8.00 GB RAM, and 750 GB hard disk space. For software, jupyter notebook development IDE is used and program is done with Python 3.3+ language with OpenCV and deep learning framework, keras. OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. The reason for using OpenCV is that it gives easy functionality to do different processes without going into implementations. Keras [7] is the system used in this thesis. It is a modular neural network library written in Python capable of running on top of either TensorFlow or Theano. The library was conceived to let the users start experimenting as fast as they could, being able to go from idea to result with the least possible delay.

The reasons why TF was selected as a backend is that both TF and keras were optimized to perform deep learning tasks. Both systems are implemented in Python which allow the user to work with them in a compact way without having to use multiple files. Both systems can run on top of both CPU and GPU which makes them very fast.

With Keras, the model has to be first defined, which can be selected between a Sequential model or a Graph model. In a Sequential model, the layers are stacked and the output from a layer feeds the input of next layer until it reaches the output layer. In the other hand, the Graph model allows

the users to get the output from a desired layer and feed that output to a desired layer, permitting the generation of multiple output networks or getting the output in an intermediate layer of the model. For this thesis the Sequential model is selected.

Once the model is defined, it has to be compiled in order to start the training. The Keras "compile" function requires two parameters that need to be tweaked. These parameters are:

• Optimizer: This parameter determines the learning and convergence of the model. There a lot of predefined optimizers in Keras, some of them are Stochastic Gradient Descend (SGD), Adam, RMSprop and Adagrad. All the optimizers have parameters that can also be modified, each optimizers has its own parameters, but there is one that is shared between all of them, the Learning Rate. This parameter will define how much the weights are updated after each epoch. For a high Learning Rate, the weight change will be higher than for a small Learning Rate, after each epoch. Also, another important parameter is the weight decay which is an additional term in the weight update rule that causes the weights to exponentially decay to zero, if no other update is scheduled. After critically overviewed and experimented the adam optimize has been selected for this study.

• Loss: The second parameter, define the objective of the training that the model has to optimize. There are many different objectives defined, for example: mean square error (MSE), mean squared logarithmic error (MSLE), categorical cross entropy that computes de logarithmic difference of the output with all the classes, and many more. From these objective functions the categorical cross entropy is selected as it is preferred for classification.

• Metrics: This parameter is optional and allows the user to see the accuracy of the model after each training step. With the model compiled, the training can be started. The Keras function to train a model is called "fit" and has a lot of parameters that can be modified as well, some of them are the following:

- Train Data: A Numpy array of the training data.
- Train Test : A Numpy array of the target data.
- Batch Size: The number of samples per gradient update.
- Number of epochs: The number of times to iterate over the training data.

- Callbacks: This parameter allows the user to save the weights of the network after each epoch if the loss is lower than any previous value.
- Validation Data: Is the sub-set that will validate how the model is performing.
- Shuffle: Allows the user to automatically shuffle the training data after each epoch.

4.3 Performance metrics

Confusion Matrix also known as contingency table, provides a comprehensive overview by summarizing the classification results. It shows the individual results for each of the categories by tabulating the predicted and actual categories. Calculating confusion matrix gives better idea of what classification model is getting right and what types of errors it is making.

Different metrics are used to measure the performance of each methods. Some of the metrics used for evaluating the proposed approach and five conventional machine learning classifier are accuracy, precision, recall, and F1 score.

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

$$\text{Accuracy} = \frac{\text{TN}+\text{TP}}{\text{TN}+\text{TP}+\text{FP}+\text{FN}}$$

Where:

TP refers to True positive: As an example of it, if the result of a common rust test is positive and the plant has that disease, then it's called a true positive.

TN refers to True negative: As an example of it, if the result of a common rust test for a maize disease is negative for maize does not have that disease, then it is called a true negative.

FP refers to False positive: As an example of it, if the result of a common rust test for a maize disease is positive for maize doesn't have that disease, then it's called a false positive.

FN refers to False negative: As an example of it, if the result of a common rust test is negative and the plant is infected by common rust, then it's called a false negative.

Precision:

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

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$$

Recall:

Recall is a measure for the true positive rate and is defined as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

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

F-score

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

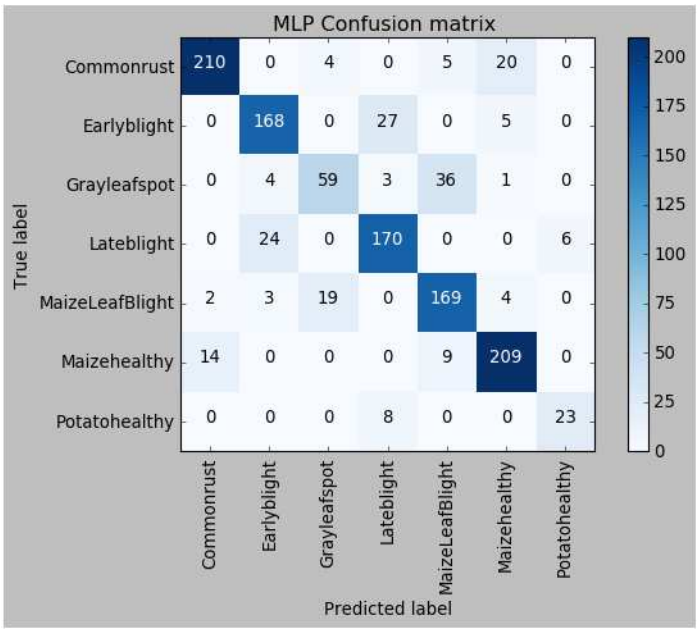
$$\text{F-score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision + Recall}}$$

4.4 Result Analysis

Experiment 1:

In this experiment, the proposed approach and the five conventional machine learning classifier were analyzed in terms of confusion matrix, average precision, average recall and average F-measure as mentioned in section 4.3. each algorithm has been run three times to select the best model comparably using the original test data. The figures below show the confusion matrix, Precision, Recall and F-Measure obtained of each techniques.

Fully connected multilayer perceptron had the third best performance classification result from the other conventional machine learning techniques. FCMLP obtained highest F1 score for mazie common rust while lowest F1 score has been obtained for Maize gray leaf spot. For Maize gray leaf spot most of their samples were classified into another maize leaf blight that indicates many similarity's were found with the maize northern blight. The average precision, recall, and f1 score for fully connected multilayer perceptron is 84.0%.



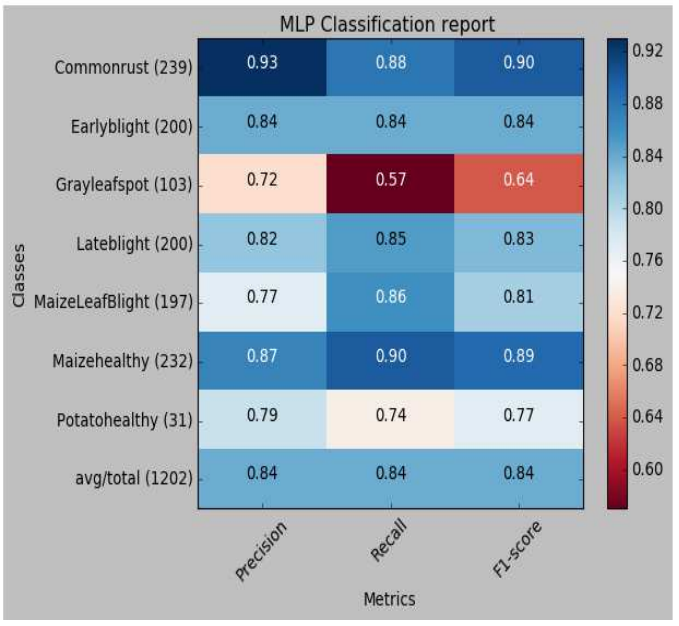
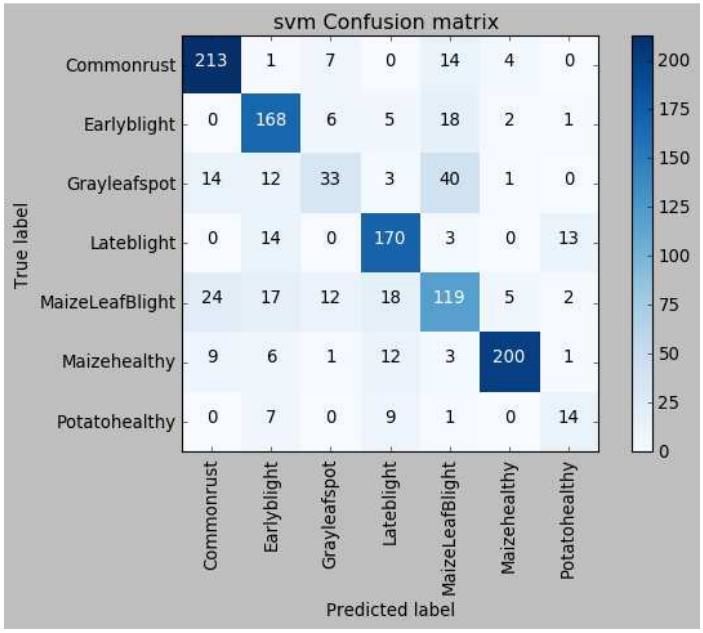


Table 4.1: Shows the Confusion Matrix, and Classification Performance metrics value of FCMLP.



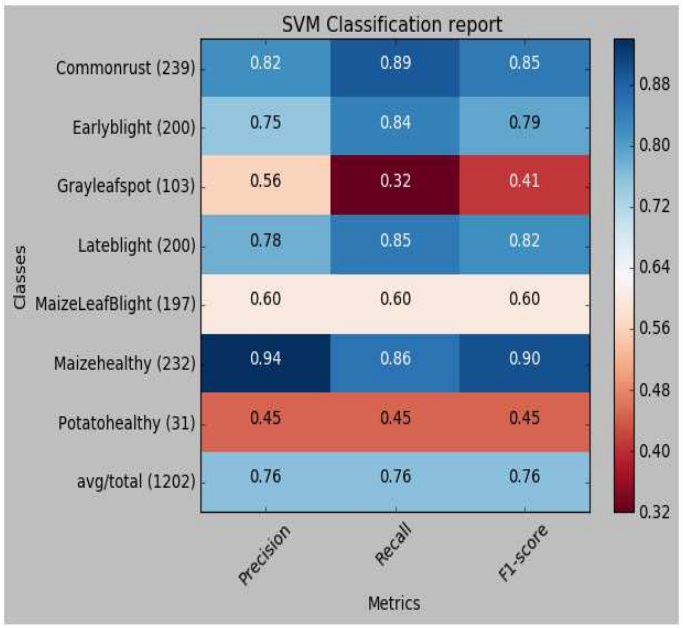


Table 4.2: Shows the Confusion Matrix, and Classification Performance metrics value of SVM.

The classification performance result of SVM show again there is much misclassification on the miaze gray leaf spot and potato healthy, but all examples are kind of classified to the correct labels except with some miner errors. A focus should be given to the maize gray leaf spot and potato healthy as it seems that a small percent is detected and classified correctly but most of the examples

are classified to other categories, so this is not the best machine learning algorithm to use it for detecting and classifying plant diseases for this kind of categories.

Decision tree algorithm classification performance result show that there is no much misclassification but only minor misclassification especially for the gray leaf , maize leaf blight and potato healthy meaning that some of the examples for this algorithm have some similarity with other classes.

Random forest had the best classification performance results from all other conventional machine learning techniques. As it can be indicated from the confusion matrix, and classification performance metrics below except maize gray leaf spot, maize leaf blight and potato healthy, the performance classification report for the other categories is above 93%. According to table 4.4 best F 1 score was achieved for maize healthy while the lowest F1 score has been obtained for the maize gray leaf spot.

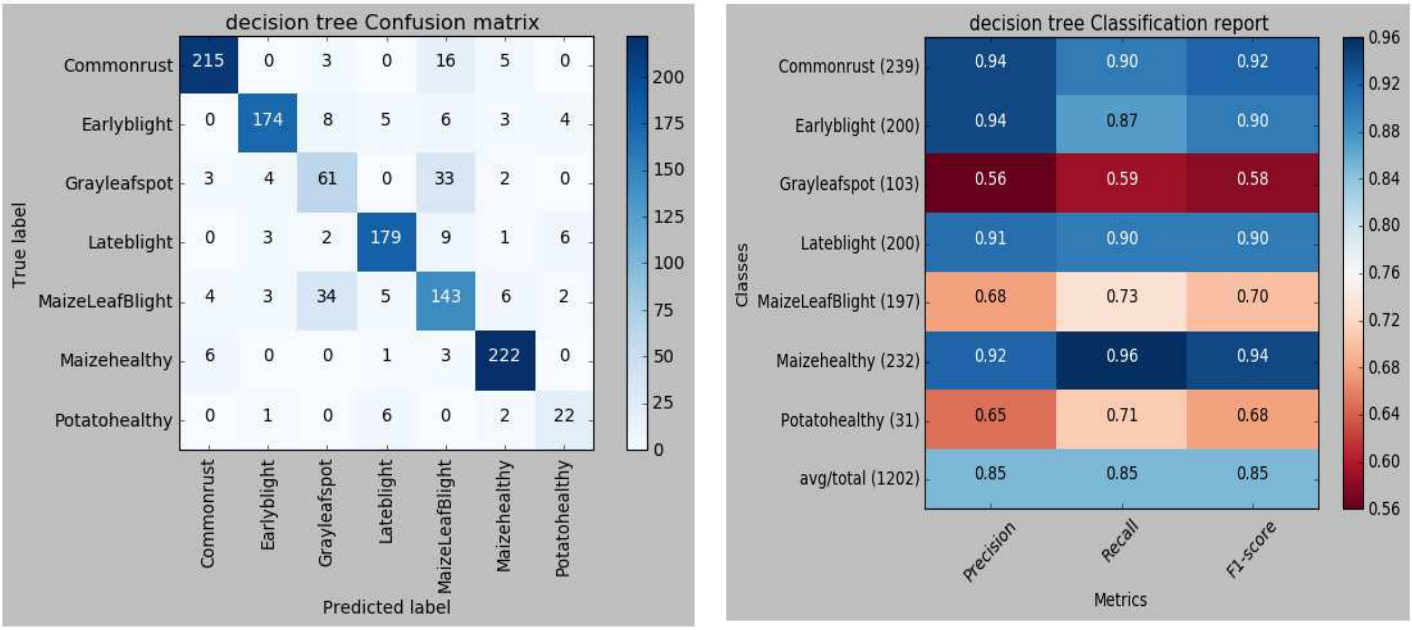
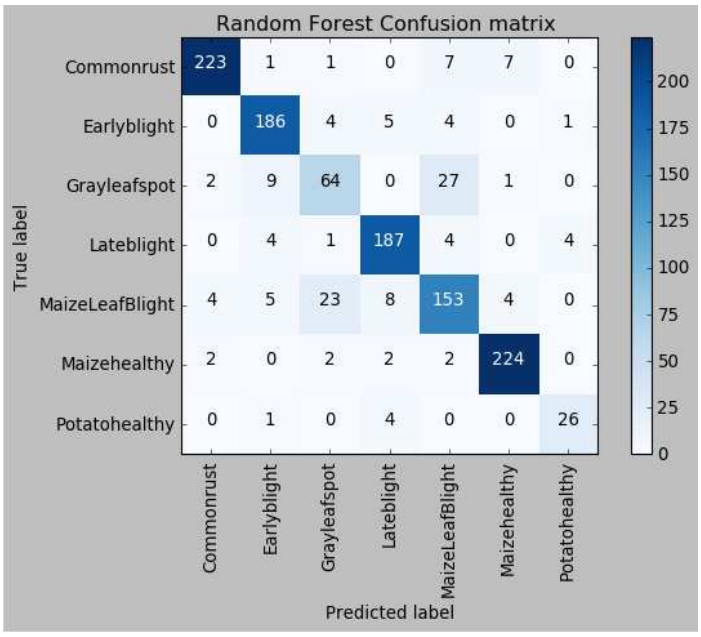


Table 4.3: Shows the Confusion Matrix, and Classification Performance metrics value of DT

KNN had the worst classification performance results form all other machine learning techniques. As it can be indicated from the below confusion matrix, and classification performance metrics , they are too misclassified so it can not categorize correctly in to the corresponding class especially maize gray, potato healthy ,and maize leaf blight. From plant disease detection and classification

perspective the problem was not approached properly; it may be considered that this is not best algorithm for this problem.



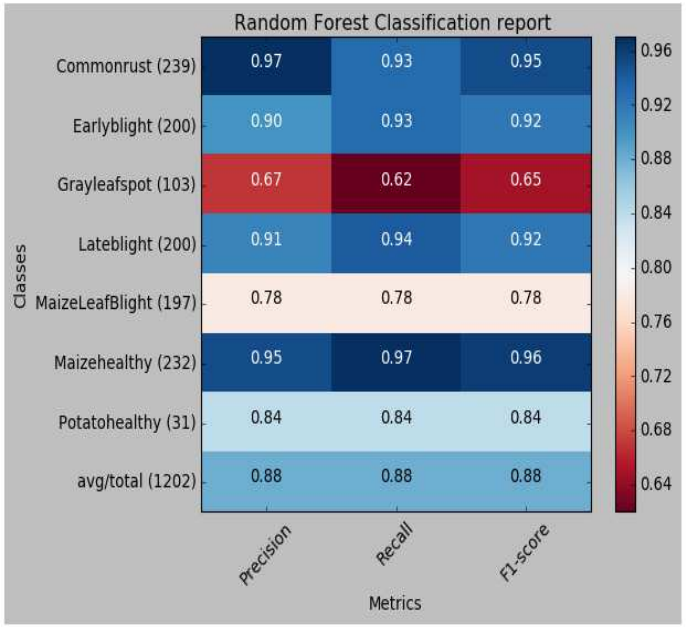
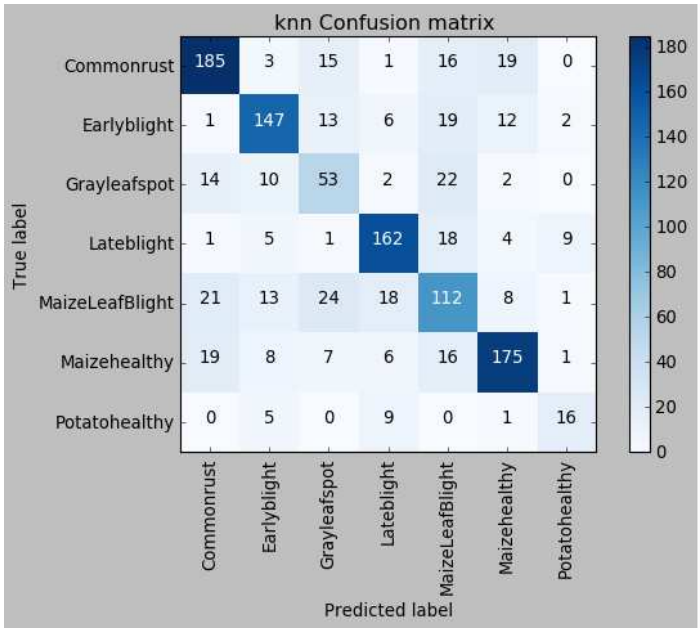


Table 4.4: Shows the Confusion Matrix, and Classification Performance metrics value of RF.



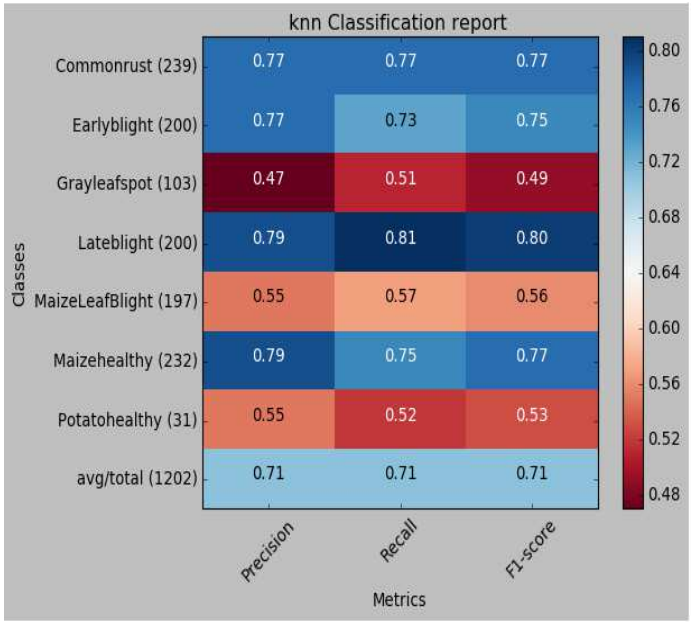


Table 4.5: Shows the Confusion Matrix, and Classification Performance metrics value of KNN.

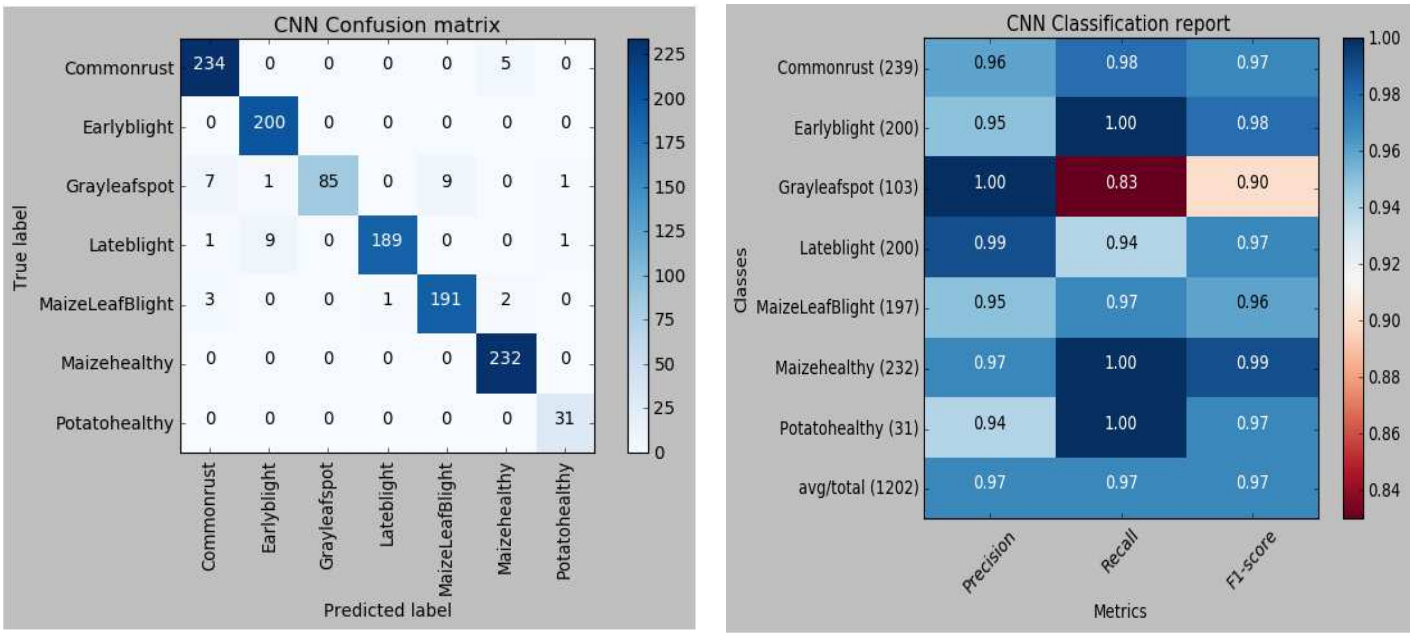


Table 4.6: Shows the Confusion Matrix, and Classification Performance metrics value of CNN.

Shown above in table 4.6 is the confusion matrix and classification performance metric value generated for the proposed approach. This model presented with precision, recall, and F1 score of 97.0%, which was a very high results. The accuracy of this model at Early blight, Maize healthy and potato healthy was specifically high, at 100.0% suggesting that this model was especially good at recognizing these categories. The average precision, recall and F1 score achieved for the proposed approach is 97.0%; except maize gray leaf spot all other categories have achieved precision, recall, and F1 score of above 96.0%.

As it is shown above in the confusion matrix of each classifier, Values on the main diagonal of the confusion matrix represents the number of categories correctly classified to the corresponding category. In this study to generate the confusion matrix, and performance classification report of each classifier the sklearn package is used.

Referring to the results obtained, as shown in Figure 4.1, general conclusion can be drawn on the performance of classifier. Result of individual classifiers, shown in Figure 4.1, illustrated the performance of comparison of each classification techniques. The proposed approach performance worked quite well in comparison with other techniques.

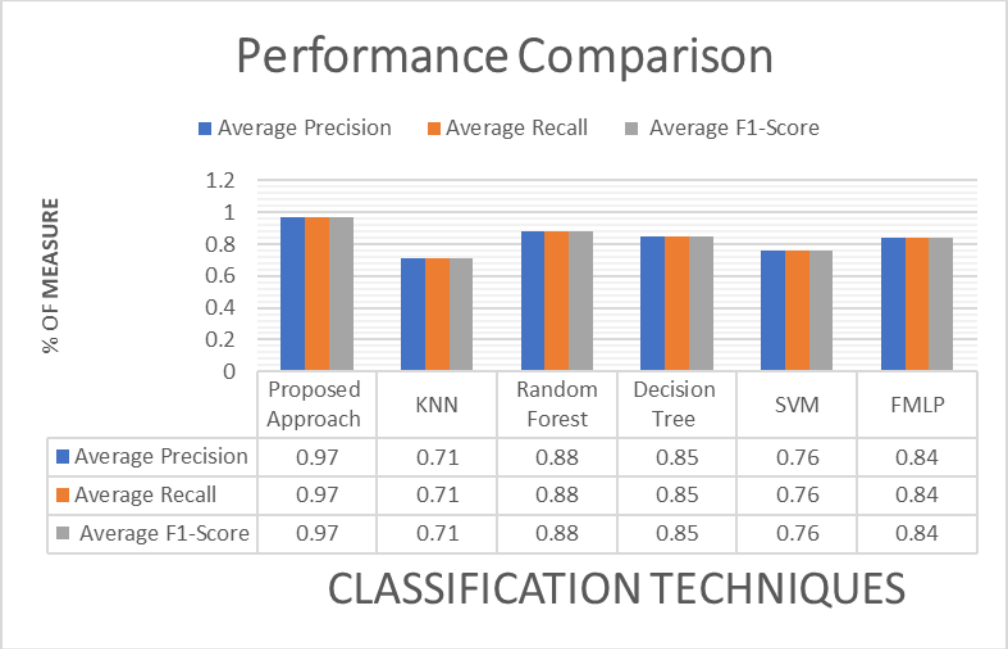


Figure 4.1: Comparison of CNN, KNN, RF, DT, SVM, FMLP base on Precision, Recall and F-Measure

Experimental Analysis 2

The above experimental results represent only the performance measures that can be used for each technique. However, it is not enough to compare algorithm alone, the study must provide a statistical evidence of the result from the evaluation. This experiment analysis involves the use of student's t test statics [40] for checking whether the proposed approach is significantly and statistically outperform other conventional machine learning technique.

This experimental analysis is necessary in order to compare the means of the proposed approach with each conventional machine learning classifier. Since each classifier has been evaluated with the same dataset, the means of each conventional machine learning technique can be directly compared with the proposed approach. T tests, and related non-parametric tests can compare the proposed approach with each conventional machine learning techniques. The results of applying a T-test to the means of the metrics from the Table 4.7 are as follow: The T–test results in table 4.7 show that there is a significant difference in each performance metrics referred to precision, recall, and F 1 score of the proposed approach with each conventional classifier ($p<0.05$) and yet the other questions proof to be considered, by conventional criteria, not statistically significant ($\rho > 0.05$).

Here, a clear differences are visible and though, the need to perform a T–test for further evaluation of the results was present. How the test was performed is explained for the creation task below. The null–hypothesis is defined as:

$$H_0: \bar{X}_{CNN} = \bar{X}_{MLP} = \bar{X}_{DT} = \bar{X}_{RF} = \bar{X}_{KNN} = \bar{X}_{SVM}$$

and means that the mean answers are equal for the entire performance metrics of each classifier.

Metrics	$\bar{X}$ CNN	$\bar{X}$ RF	$\bar{X}$ SVM	$\bar{X}$ MLP	$\bar{X}$ KNN	$\bar{X}$ DT	σ CNN	σ RF	σ SVM	σ MLP	σ KNN	σ DT
Precision	0.96571	0.86	0.7	0.82	0.67	0.8	0.00050	0.01120	0.02883	0.00473	0.0196	0.02670
Recall	0.96	0.85857	0.68714	0.80571	0.66571	0.80857	0.00377	0.01545	0.05326	0.01340	0.01626	0.01785
F1-Score	0.96286	0.86	0.68857	0.81143	0.66714	0.80286	0.00086	0.0127	0.04011	0.00771	0.0179	0.02112

Table 4.7: Values obtained for each performance metrics after training all classifier three times.

The alternative hypothesis, which directly correlates to the hypothesis of the present work, is defined as

$$H_1: \bar{X}_{CNN} > \bar{X}_{MLP}\ ;\ \bar{X}_{CNN} > \bar{X}_{DT}\ ;\ \bar{X}_{CNN} > \bar{X}_{RF}\ ;\ \bar{X}_{CNN} > \bar{X}_{KNN}\ ;\ \bar{X}_{CNN} > \bar{X}_{SVM}$$

and means that the mean answers of the proposed approach for the entire performance metrics of each classifier is greater than the conventional classifier, which in turn could indicate that the proposed approach is better. To reject the null–hypothesis, the results of the performance metrics have been used as a sample population. For instance the sample means of MLP and CNN have the values $\bar{X}_{CNN} = 0.96571;\ and\ \ \bar{X}_{MLP} = 0.81143$

The sample size for each classifier is equal to $n_{MLP} = n_{CNN} = 3$. The last thing needed for the T–test equation is the variance. The equation is defined as

$$S^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

where the sum with x_i values, the variances equal is cycling through all values for each run. The variances for each technology are: S^2_{MLP}= 0.00771 and S^2_{CNN} =0.00086

They are therefore almost equal, which makes the T–test of type 2: Different sets of runs for each classifier with equal variances. The equation for calculating the actual T–value is then defined as[41]:

$$t = \frac{\bar{X}_{CNN} - \bar{X}_{MLP}}{\sqrt{\frac{(n-1)S^2_{MLP} + (n-1)S^2_{CNN}}{n+n-2} \cdot \frac{n+n}{n.n}}}$$

t= 2.77645

That makes the significance level (ρ)[42]:

ρ= 0.02500

Which means that the difference between the means is highly significant ($\rho < 0.05$) which in turn causes the null-hypothesis ($\bar{X}_{CNN} = \bar{X}_{MLP}$) to be rejected and shows that the MLP Classifier is significantly less with the proposed approach.

The results generated from each classifier after three times training were given to the JMP statistics software and generated the following tables.

	CNN	RF	DT	MLP	SVM	KNN
CNN	-0.02138	0.06862	0.09862	0.11195	0.18862	0.2459
RF	0.06862	-0.02138	0.00862	0.02195	0.09862	0.15529
DT	0.09862	0.00862	-0.02138	-0.00805	0.06862	0.12529
MLP	0.11195	0.02195	-0.00805	-0.02138	0.05529	0.11195
SVM	0.18862	0.09862	0.06862	0.05529	-0.02138	0.03529
KNN	0.2459	0.15529	0.12529	0.11195	0.03529	-0.02138

Table 4.8: Least significance difference

As shown above in table 4.8 positive values show pairs of means that are significant different, whereas the negative values show pairs of means that are not significantly different. The larger positive LSD value the more significantly different.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
CNN	KNN	0.2666667	0.0098131	0.245286	0.2880475	<.0001*
CNN	SVM	0.2100000	0.0098131	0.188619	0.2313808	<.0001*
RF	KNN	0.1766667	0.0098131	0.155286	0.1980475	<.0001*
DT	KNN	0.1466667	0.0098131	0.125286	0.1680475	<.0001*
CNN	MLP	0.1333333	0.0098131	0.111952	0.1547142	<.0001*
MLP	KNN	0.1333333	0.0098131	0.111952	0.1547142	<.0001*
CNN	DT	0.1200000	0.0098131	0.098619	0.1413808	<.0001*
RF	SVM	0.1200000	0.0098131	0.098619	0.1413808	<.0001*
CNN	RF	0.0900000	0.0098131	0.068619	0.1113808	<.0001*
DT	SVM	0.0900000	0.0098131	0.068619	0.1113808	<.0001*
MLP	SVM	0.0766667	0.0098131	0.055286	0.0980475	<.0001*
SVM	KNN	0.0566667	0.0098131	0.035286	0.0780475	<.0001*
RF	MLP	0.0433333	0.0098131	0.021952	0.0647142	0.0008*
RF	DT	0.0300000	0.0098131	0.008619	0.0513808	0.0100*
DT	MLP	0.0133333	0.0098131	-0.008048	0.0347142	0.1992

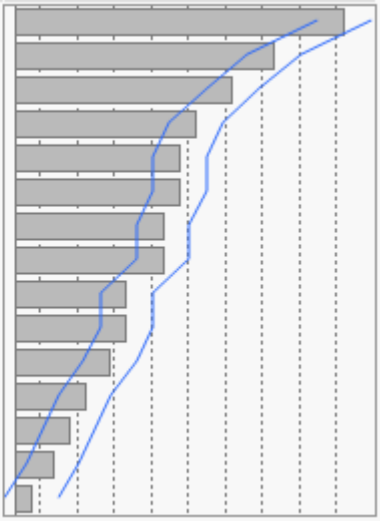


Table 4.9.: Values obtained for each metrics after evaluating all classifiers over 3 runs applying a T-test. Above result indicate the p value between each classifier generated by the software. Any p value less than 0.05 indicates that there is significant difference between the classifier.

Experiment Analysis 3:

This experimental analysis involves testing the generalization capability of the proposed approach in classifying plant disease that affect other plants, but not used for training purpose. For checking the generality potential of the proposed approach in classifying the same diseases which were used in training the model, but affect other plants. Total 200 leaves which are affected by early and late blight tomato disease are used for experiment. As indicated in the confusion matrix below from the total 140 and 154 of these are correctly classified as early and late blight respectively, whereas 60 and 46 of them are misclassified.

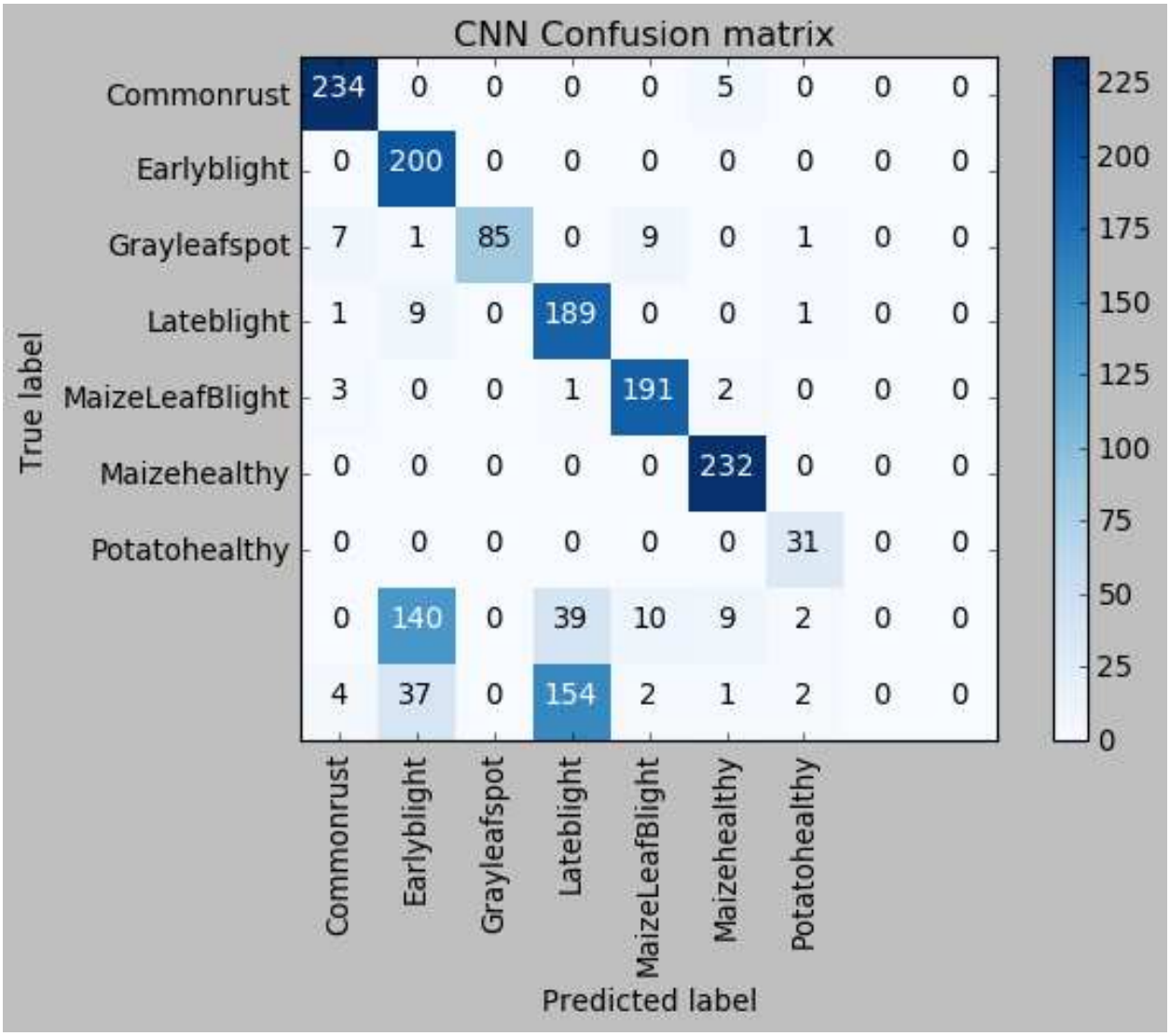


Table 4.10: Shows the Confusion Matrix of CNN for the visualization of the generalization potential.

4.5 Answers to Research Questions

In the first chapter of this study two research questions which are considered relevant in evaluating the success of the study were formulated. The answer to these research questions are examined as follow:

Research question 1: Does convolutional neural networks out-perform other machine learning techniques for plant disease detection and classification?

The experimental results obtained as shown in figure 4.1, indicates that, in many cases the classification report improvement achieved by the proposed approach is large compared to existing approaches. For instance, proposed approach achieves highest 26.0%, 26%, 26% performance improvement and at least 9.0%, 9.0%, 9.0% improvement in Precision, Recall and F Score respectively compared to other the existing machine learning classifier.

The proposed approach gives comparable performance in comparison with the conventional machine learning methods because the proposed approach extract and select high level features from the given raw input images. Therefore, from the experimental results obtained it can be concluded that the proposed approach improves the performance of classifying plant disease into the corresponding category in terms of the classification report.

Research question 2: Does convolutional neural networks adapt for disease detection and classification of any plant with the help of transfer learning?

From the total 200 leaves which are threaten by early blight tomato disease are used for experiment. In those leaves 60 images were misclassified and the other 140 images are correctly classified. In addition, Total 200 leaves which are affected by tomato late blight disease are used for experiment. In those leaves, 56 images were misclassified and the other 154 images are correctly classified. It is therefore safe to conclude that yes, the proposed approach can detect and classify plant disease which was not used during training the model.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Introduction

The purpose of this study is to build a model that capable of automatically detecting and classifying plant diseases on leaves using artificial neural networks. This chapter of the study deals with the conclusion and summaries of the results obtained by the researcher. The chapter continues to give suggestions and recommendations made by the researcher based on the findings of the research to help other researchers who wanted to do research on the same field of study. These recommendations would help future researchers to enhance and improve the study and even go into the details of the study.

5.2 Summary

This research contains five main chapters with different subtopics. Chapter one of this study gives a brief introduction and background of the study. It further explains the purpose of the study which is automatically plant disease detection and classification on leaves using an artificial neural network. It also includes topics like the significance of the study, research questions, objectives, hypothesis, scope and limitations of the study, methodology, research contribution, organization of the thesis.

Chapter two also deals with a review of related studies. This is a review of the work of other researches are analyzed and reviewed by discussing the general structure of the study as well as the experimental evidence of the study.

Again, various concepts such as definition of machine learning, machine leaning techniques in context with plant disease detection and classification using neural network were thoroughly discussed. The chapter covered the review of existing literature in relation to the study. Books, articles and other materials which are written by other persons or authors associated with the topic were considered. The main items considered in the chapter include Image processing, Uses of image processing, Image processing techniques, convolutional neural network and other machine learning techniques

Chapter three of the research deals with the materials, and methodologies used to get the desired result of the study. These include algorithms, flowcharts, and other resources to get the outcome of the study. It includes the process for preprocessing the images, methods of masking and removing the green part of the image, and look up algorithm for the disease management techniques of the classified disease.

Chapter four of the research deals with the results, analysis, and explanation of the data collected and the results obtained from the research.

This chapter gives an explanation of the methodology used to conduct the study. The methodology was put in place with the aim that they will help the reader for masking and removing the green part of the images. The results were presented in simple forms using graphs, tables, charts, etc, for easy reading and analysis.

The last chapter of the research summarizes the entire thesis by explaining the details of the study. It further provides an overview of the research by explaining the main topics of the study. Some of the topics considered in this chapter include summary, conclusion, and recommendations.

5.3 Conclusion

This research was meant to a build model that could automatically detect and classify plant disease appeared on plant leaves using artificial neural network. The research took into consideration the negative impact of leaf diseases on plants. Many studies have shown that the effect of plant diseases is huge and difficult to deal with. However, plant diseases, especially those appeared on the leaf part of the plant could be detected and classified by combining the use of image processing and machine learning techniques so as to ensure the proper and appropriate use of chemical for curing the diseases and decrease the production loss of agriculture. Different image processing techniques were applied in order to improve and enhance the quality of the image. Besides, masking and removing the green part of the image is carried out to separate the diseased region from the heathy one. The methodology involves image collection, image pre-processing , masking and removing the green part of the input image, After the masking and removing the green part of the input image phase texture features from GLCM using haralick were extracted and features that have the same values were neglected especially for the conventional machine learning methods, applying CNN machine learning techniques for classification purpose for extracting and selecting high level features from the green part removed image. All the input images will be passing through the image pre-processing, masking and removing the green image, feature extraction and selection steps before it proceeds to the conventional machine learning techniques, whereas, for the deep learning all the input images will be passing through the image preprocessing , and masking and removing the green part of the image phase before it proceeds to the CNN. The methodology used in this study proves to be one of the simples ways of detecting and classifying leaf disease in plants. Consequently, it also proves to be one of the best in terms of accuracy because it works up to 97% accuracy in terms of results.

After that, For the classification purpose an experimental analysis using the state-of-the-art machine learning methods have been conducted. The last step in this study is the disease management technique. The disease management technique let the user how to cure the plant from the disease and what pesticide to use for curing it from the disease being attacked.

5.4 Recommendations

It is known that; agriculture is the backbone of every country no matter how well it is developed. Therefore, there is the need to pay much attention to it so that the right output will be obtained from it. In our country where there is limited application of technology in agriculture, extension worker and users find it very difficult to produce high yields due to factors specifically plant diseases. It is therefore suggested that much concern will be given to the treatment of plant diseases to reduce the production loss in the agriculture sector. In addition, the government should focus on training the agricultural extension worker, users, and pathologist to use and apply the nondestructive way of detecting and classifying plant disease early. In addition, farmers must also be given the required attention and training as to how they should go about their farming practices specifically the use of pesticides and fungicides. Again, right amount of fungicides and pesticides must be applied to plants that are affected with various forms of diseases to avoid excessive toxic waste in food crops and contamination of groundwater bodies.

5.5 Future work

The primary objective of this study has been achieved and the research questions have been answered. Although, the researcher has tried his best to get the desired output and the accuracy value is quite good for this method, yet there is still room for improvement as long as the accuracy is not exactly 100 percent. As a future work , researchers should increase the number of the images and the classes used in the dataset which will have an impact on the detection and classification. In addition through this investigation , a conclusion were drawn that deep learning techniques heavily rely on the input parameters. By tuning the CNN input parameters and layers, the accuracy might be improved. The future work should be to bring the plant disease detection and classification accuracy over 99.7%.

In this thesis, only the most common disease that affect the two plants were used as a source, but more data could be accumulated if more disease would be added. Also, more exhaustive testing on traditional machine learning methods and transfer learning methods should be done, as this

study tested only a few of the methods. It might be that some different methods in machine learning are more suitable for this kind of data.

In this study, only texture features especially for the conventional machine learning algorithms have been extracted. As future work, researchers should try to use meta features so that it can be optimized to a achieve a higher recognition accuracy.

Finally , future researchers should try to include in to their work how the farmers, users, and pathologist can measure the right quantity and concentration of fungicides and pesticides based on the severity of the plant disease before applying them on their crops to prevent ground water pollutions due to excessive toxic waste and agricultural production losses.

References

[1] JENNIFERC, “The Importance of Plants to Life on Earth,” 2014. [Online]. Available: https://blog.udemy.com/importance-of-plants/. [Accessed: 12-May-2018].

[2] B. J. Staskawicz, M. B. Mudgett, J. L. Dangl, and J. E. Galan, “Common and Contrasting Themes of Plant and Animal Diseases,” *Science (80-.).*, vol. 292, no. 5525, pp. 2285–2289, Jun. 2001.

[3] S. Chakraborty and A. C. Newton, “Climate change, plant diseases and food security: an overview,” *Plant Pathol.*, vol. 60, no. 1, pp. 2–14, Feb. 2011.

[4] M. Ray *et al.*, “Fungal disease detection in plants: Traditional assays, novel diagnostic techniques and biosensors,” *Biosens. Bioelectron.*, vol. 87, pp. 708–723, Jan. 2017.

[5] B. Zechmann and G. Zellnig, “Rapid diagnosis of plant virus diseases by transmission electron microscopy,” *J. Virol. Methods*, vol. 162, no. 1–2, pp. 163–169, Dec. 2009.

[6] M. Zhang, X. Liu, and M. O’Neill, “Spectral discrimination of Phytophthora infestans infection on tomatoes based on principal component and cluster analyses,” *Int. J. Remote Sens.*, vol. 23, no. 6, pp. 1095–1107, Jan. 2002.

[7] U. F. Abdulhamid and M. A. Aminu, “Detection of Soya Beans Ripeness Using Image Processing Techniques and Detection of Soya Beans Ripeness Using Image Processing Techniques and Artificial Neural Network,” no. March, 2018.

[8] Z. Ramedani, M. Omid, A. Keyhani, S. Shamshirband, and B. Khoshnevisan, “Potential of radial basis function based support vector regression for global solar radiation prediction,” *Renew. Sustain. Energy Rev.*, vol. 39, pp. 1005–1011, Nov. 2014.

[9] S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, and P. Kulkarni, ““Diagnosis and classification of grape leaf diseases using neural networks”,” in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–5.

[10] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, “Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance,” *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, Oct. 2010.

[11] A. George N., “Introduction to Plant Diseases,” 2017. [Online]. Available: https://www.academicscope.com/introduction-plant-diseases/. [Accessed: 12-Apr-2018].

[12] H. S. and P. Chaube, “CROP DISEASES AND THEIR MANAGEMENT - H. S. CHAUBE, V. S. PUNDHIR - Google Books,” *PHI Learning Pvt. Ltd*, 2005. .

[13] Y. Mazengia, “Smallholders commercialization of maize production in Guangua district , northwestern Ethiopia,” vol. 58, pp. 65–83, 2016.

[14] P. Olivera *et al.*, “ Phenotypic and Genotypic Characterization of Race TKTTF of Puccinia graminis f. sp. tritici that Caused a Wheat Stem Rust Epidemic in Southern Ethiopia in 2013–14 ,” *Phytopathology*, vol. 105, no. 7, pp. 917–928, 2015.

[15] A. M. Tesfaw Abay Birsh, Fekadu Gebretensay Mengistu, Amsalu Nebiyu3, “Response of Potato Varieties for Extended Harvesting at Kulumsa, Southeast Ethiopia,” 2018. [Online]. Available: https://www.academicresearchjournals.org/ARJASR/Abstract/2018/June/Birsh et al.htm.

[Accessed: 13-Sep-2018].

[16] Phillip Wharton, “Potato Diseases: Early Blight (E2991) - MSU Extension,” 2015. [Online]. Available: https://www.canr.msu.edu/resources/potato_diseases_early_blight_e2991. [Accessed: 13-Sep-2018].

[17] Muhammad Jamil Moughal, “Which Machine Learning algorithm to use? – Muhammad Jamil Moughal – Medium,” 2018. [Online]. Available: https://medium.com/@mjamilmoughal786/which-machine-learning-algorithm-to-use-bd9f7dc479c4. [Accessed: 13-Oct-2018].

[18] and S. Y. Fei-Fei Li, Justin Johnson, “CS231n Convolutional Neural Networks for Visual Recognition,” *Jan.* , 2017. [Online]. Available: http://cs231n.github.io/neural-networks-1/. [Accessed: 15-Jul-2018].

[19] and S. Y. Fei-Fei Li, Justin Johnson, “CS231n Convolutional Neural Networks for Visual Recognition,” *Jan.* , 2017. [Online]. Available: http://cs231n.github.io/neural-networks-1/. [Accessed: 15-Aug-2018].

[20] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, “A Taxonomy of Deep Convolutional Neural Nets for Computer Vision,” *Front. Robot. AI*, vol. 2, Jan. 2016.

[21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015.

[22] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” Mar. 2016.

[23] Savan Patel, “Chapter 2 : SVM (Support Vector Machine) — Theory – Machine Learning 101 – Medium,” *May 3,* 2017. [Online]. Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72. [Accessed: 15-Oct-2018].

[24] S. Raschka, *Python Machine Learning*. 2016.

[25] Abhay Padda, “Introduction to Random Forest.,” 2018. [Online]. Available: https://analyticsdefined.com/introduction-random-forests/. [Accessed: 16-Feb-2018].

[26] TAVISH SRIVASTAVA, “Introduction to KNN, K-Nearest Neighbors,” *Mar 27th,* 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Accessed: 16-May-2018].

[27] Rajesh S. Brid, “Decision Trees — A simple way to visualize a decision,” *Oct 25,* 2018. [Online]. Available: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb. [Accessed: 16-Sep-2018].

[28] U. Abdulhamid, S. Daniel, and U. Babawuro, “Classification of Soya Beans Based Image Processing Techniques and Artificial Neural Network,” *J. Adv. Math. Comput. Sci.*, vol. 26, no. 6, pp. 1–9, Mar. 2018.

[29] K. Ko Zaw, Z. Ma Ma Myo, and D. Thae Hsu Thoung, “Support Vector Machine Based Classification of Leaf Diseases,” 2018.

[30] “Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance,” *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, Oct. 2010.

[31] A. Kliamenakis, “Automatic Flower Disease Identification Using Image Processing,” p. 112, 2011.

[32] Habtamu Minassie Aycheh, “Image analysis for Ethiopian coffee classification,” *Chem. …*, no. January, pp. 1–11, 2008.

[33] S. Wallelign, M. Polceanu, and C. Buche, “Soybean Plant Disease Identification Using Convolutional Neural Network,” *Thirty-First Int. Flairs Conf.*, May 2018.

[34] M. Reyalat, M. Braik, and Z. Alrahamneh, “Fast and Accurate Detection and Classification of Plant Diseases,” vol. 17, no. 1, pp. 31–38, 2011.

[35] J. (Eds. . Espinoza, L. & Ross, “Maize (corn) | Diseases and Pests, Description, Uses, Propagation.” [Online]. Available: https://plantvillage.psu.edu/topics/corn-maize/infos. [Accessed: 19-Mar-2019].

[36] M. Rohit Verma and J. Ali, “A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques,” 2013.

[37] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Shadow identification and classification using invariant color models,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 3, pp. 1545–1548.

[38] G. Tripathi, “Review on color and texture feature extraction techniques,” 2014.

[39] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 2007.

[40] Web Center for Social Research Methods, “The T-Test,” 2017. [Online]. Available: http://www.socialresearchmethods.net/kb/stat_t.php. [Accessed: 24-Mar-2019].

[41] David R. Caprette, “‘Student’s’ t Test (For Independent Samples),” 2016. [Online]. Available: http://www.ruf.rice.edu/~bioslabs/tools/stats/ttest.html. [Accessed: 20-Mar-2019].

[42] social science statistics, “T-Test Calculator for 2 Independent Means.” [Online]. Available: https://www.socscistatistics.com/tests/studentttest/. [Accessed: 24-May-2019].

[43] Choosing the optimal model,"Data Science, Machine Learning and Statistics,implmented in python "[Online].Available:https://xavierbourretsicotte.github.io/subset_selection.html. [Accessed 28-May-2019].