



**Strathmore**  
UNIVERSITY

**SU+ @ Strathmore**  
**University Library**

---

**Electronic Theses and Dissertations**

---

2020

# A Computer vision based model for tomato plant nutrient and disease classification

Kiyegga, R. Paul  
*Faculty of Information Technology*  
*Strathmore University*

## **Recommended Citation**

Kiyegga R. P. (2020). *A Computer vision based model for tomato plant nutrient and disease classification* [Thesis, Strathmore University]. <http://hdl.handle.net/11071/12134>

Follow this and additional works at: <http://hdl.handle.net/11071/12134>

# **A Computer Vision Based Model for Tomato Plant Nutrient and Disease Classification**

By

RAYMOND PAUL KIYEGGA

112701

A Thesis Submitted to the Faculty of Information Technology in partial fulfilment of the requirements for the award of the Degree of Master of Science in Information Technology.

Master of Science in Information Technology

**STRATHMORE UNIVERSITY**

June 2020

## Declaration and Approval

I, Raymond Paul Kiyegga declare that this research has not been submitted to any other university for the award of the Degree of Master of Science in Information Technology.

Student Name: Raymond Paul Kiyegga



Sign: \_\_\_\_\_

Date: 26<sup>th</sup> June 2020

Supervisor's Name: Dr. Bernard Shibwabo

Sign: B.K.S

Date: 26<sup>th</sup> June 2020

## **Acknowledgements**

I would like to extend my sincere gratitude to my supervisor Dr. Shibwabo for his relentless efforts, encouragement, and support to see that this research is completed successfully. Special thanks to Prof. Cecilia for the data and assistance accorded to me during data collection. I would like to thank the Faculty of Information Technology at Strathmore University for sponsoring my master's course. Sincere thanks to my family for the prayers and encouragement. To my friends Federico, Bakanja and all those in Satima for all the assistance they accorded me. In a special way, I would like to extend my heartfelt gratitude to Doreen for her constant encouragement, reminders and support accorded to me during this journey.

## Abstract

Determination of disease and nutrients in plants is still a new concept. Despite efforts from researchers to come up with improved techniques of detecting diseases and nutrients, many have been limited to only specific plant images and no other data such as weather, surrounding conditions to back up the decision. Plant disease identification is very crucial to food production and security, however current practices in Africa include visual identification and microscopy. Visual methods are greatly affected by cognitive error while microscopy is time consuming. It is difficult to detect plant disease unless one is guided by expert knowledge. Therefore, there is a need to apply machine learning techniques to make use of this expert knowledge. Current practices include use of spectral images to achieve this in fruits and other applications to help farmers without access to this knowledge to diagnose plant diseases. One notable challenge is determining nutrient content using images. Current applications require a farmer to look at the provided image and compare with what he sees on the plant. This research work proposes a machine learning model that can automatically detect the disease affecting a tomato plant as well as the nutrition level in the plant leaves. The farmer captures an image on their phone while in the plantation, based on the features from the leaf, the model analyses the image and returns the details of the classification in terms of type of disease and presence of deficiency of nutrients. The model was built on convolution neural network and achieved an accuracy of 85% using a learning rate of 0.001. It trained on 8000 samples using 30 epochs. The model was trained, validated and tested.

**Keywords:** Artificial intelligence, deficiency, disease, nutrient, neural networks.

## **Definition of Terms**

### **Classification Model**

A term that describes drawing a conclusion from an input value given for training and can predict the category for new data (Sarker, 2019).

### **Mineral deficiency**

A term that describes the physical symptoms and conditions of a plant when insufficient levels of a specific nutrient or nutrients are present in the soil or grow medium in which the plant grows (Lauchli, 2012).

### **Neural Networks**

This consists of thousands or even millions of simple processing nodes that are densely interconnected. They are a means of machine learning in which a computer learns to perform a task by analysing specific examples (Ian et al., 2016).

### **Nutrient**

This is a chemical element that is essential to the nourishment of plant health. The three main nutrients are nitrogen, potassium and phosphorous (Allen, 2016).

### **Plant disease**

This a disorder of structure or function in a plant, especially one that produces specific symptoms or that affects a specific location and is not simply a direct result of physical injury (Thind, 2019).

## Table of Contents

|  |     |
|--|-----|
| Declaration and Approval .....   | ii  |
| Acknowledgements .....   | iii |
| Abstract .....   | iv  |
| Definition of Terms .....  | v   |
| Table of Contents .....  | vi  |
| List of Acronyms .....   | xi  |
| List of Figures .....  | xii |
| List of Tables .....   | xiv |
| Chapter 1 : Introduction .....   | 1   |
| 1.1 Background .....   | 1   |
| 1.2 Problem Statement .....  | 3   |
| 1.3 Aim .....  | 3   |
| 1.4 Research Questions .....   | 3   |
| 1.5 Specific Objectives .....  | 4   |
| 1.6 Justification .....  | 4   |
| 1.7 Scope and Limitation .....   | 4   |
| Chapter 2 : Literature Review .....  | 6   |
| 2.1 Overview .....   | 6   |
| 2.2 Challenges Affecting Farmers in Plant Disease Identification .....       | 6   |
| 2.3 Challenges Affecting Farmers in Nutrient Deficiency Identification ..... | 7   |
| 2.4 Diseases Affecting Tomato Plants .....                                   | 7   |
| 2.4.1 Early Blight .....   | 7   |
| 2.4.2 Target spot .....  | 8   |

|                                       |   |    |
|---------------------------------------|---|----|
| 2.4.3                                 | Tomato Mosaic Virus .....   | 8  |
| 2.4.4                                 | Tomato Yellow Leaf Curl virus .....   | 9  |
| 2.4.5                                 | Bacterial Spot.....   | 10 |
| 2.4.6                                 | Late Blight .....   | 10 |
| 2.4.7                                 | Leaf Mold.....  | 11 |
| 2.4.8                                 | Septoria Leaf Spot.....   | 11 |
| 2.5                                   | Nutrient Deficiency in Tomato Plant .....   | 12 |
| 2.6                                   | Techniques for Identifying Plant Disease .....  | 13 |
| 2.6.1                                 | Polymerase Chain Reaction .....   | 13 |
| 2.6.2                                 | Immunofluorescence.....   | 13 |
| 2.6.3                                 | Gas Chromatography .....  | 14 |
| 2.6.4                                 | Human Vision .....  | 14 |
| 2.6.5                                 | Thermography.....   | 14 |
| 2.6.6                                 | Fluorescence Imaging .....  | 15 |
| 2.6.7                                 | Hyperspectral Techniques.....   | 15 |
| 2.7                                   | Expert System .....   | 15 |
| 2.7.1                                 | Components of an Expert System.....   | 16 |
| 2.8                                   | Computer Vision Based Prediction Systems.....   | 17 |
| 2.8.1                                 | Machine Learning Techniques used in Classification of Plant Disease and Nutrient Deficiency ..... | 17 |
| 2.9                                   | Architectures and Designs.....  | 23 |
| 2.10                                  | Conceptual Framework.....   | 25 |
| 2.11                                  | Summary.....  | 26 |
| Chapter 3 : Research Methodology..... |   | 28 |
| 3.1                                   | Introduction .....  | 28 |

|   |                                     |    |
|---|-------------------------------------|----|
| 3.2   | Research Design.....                | 28 |
| 3.3   | Population and Sampling .....       | 28 |
| 3.3.1   | Population .....                    | 28 |
| 3.3.2   | Sampling.....                       | 29 |
| 3.3.3   | Data Collection .....               | 29 |
| 3.4   | Data Analysis Methods .....         | 29 |
| 3.5   | System Development Methodology..... | 30 |
| 3.5.1   | Agile System Development .....      | 30 |
| 3.5.2   | System Analysis.....                | 30 |
| 3.5.3   | System Design .....                 | 30 |
| 3.6   | System Implementation.....          | 31 |
| 3.6.1   | Model .....                         | 31 |
| 3.6.2   | Test Bed .....                      | 31 |
| 3.6.3   | Programming Language.....           | 31 |
| 3.7   | Research Quality .....              | 31 |
| 3.8   | Ethical Considerations.....         | 31 |
| Chapter 4 : System Analysis, Design and Architecture..... |                                     | 32 |
| 4.1   | Introduction .....                  | 32 |
| 4.2   | Requirement Analysis .....          | 32 |
| 4.2.1   | Functional Requirements .....       | 32 |
| 4.2.2   | Non-functional Requirements.....    | 32 |
| 4.3   | System Architecture .....           | 33 |
| 4.3.1   | System Model Architecture .....     | 33 |
| 4.3.2   | Partial Domain Model.....           | 33 |
| 4.3.3   | Use Case Diagram.....               | 34 |

|  |   |    |
|--|---|----|
| 4.4  | Sequence Diagram.....                         | 38 |
| 4.5  | Context Diagram .....                         | 39 |
| Chapter 5: Implementation and Testing.....                   |   | 40 |
| 5.1  | Introduction .....                            | 40 |
| 5.2  | Development Environment .....                 | 40 |
| 5.3  | Hardware Resources.....                       | 40 |
| 5.4  | Software Resources .....                      | 41 |
| 5.5  | Model Components .....                        | 41 |
| 5.5.1  | Neural Network Components .....               | 41 |
| 5.6  | Image Processing.....                         | 42 |
| 5.6.1  | Fetching Images from the Root Directory ..... | 42 |
| 5.6.2  | Transforming Image Labels.....                | 42 |
| 5.6.3  | Data Augmentation .....                       | 43 |
| 5.7  | Model Implementation .....                    | 44 |
| 5.8  | Training and Testing .....                    | 44 |
| 5.8.1  | Model Training .....                          | 44 |
| 5.9  | Model Testing .....                           | 46 |
| Chapter 6: Discussions.....                                  |   | 47 |
| 6.1  | Validation of the Model .....                 | 47 |
| 6.2  | Classification Report of the Model .....      | 49 |
| 6.3  | Research Contribution.....                    | 51 |
| 6.4  | Achievement of the Objectives .....           | 52 |
| 6.5  | Limitations to the Research.....              | 54 |
| Chapter 7: Conclusion, Recommendations and Future Work ..... |   | 56 |
| 7.1  | Conclusion.....                               | 56 |

|     |                       |    |
|-----|-----------------------|----|
| 7.2 | Recommendations ..... | 56 |
| 7.3 | Future Work .....     | 57 |
|     | References.....       | 58 |
|     | Appendices.....       | 66 |

## List of Acronyms

|        |                             |
|--------|-----------------------------|
| AI     | Artificial Intelligence     |
| ANN    | Artificial Neural Networks  |
| CNN    | Convolution Neural Networks |
| CovNet | Convolution Neural Network  |
| DNA    | Deoxyribonucleic acid       |
| KNN    | K-Nearest Neighbour         |
| ML     | Machine Learning            |
| SVM    | Support Vector Machine      |
| VOC    | Volatile Organic Compounds  |

## List of Figures

|   |    |
|---|----|
| Figure 2.1: Leaves Infected with Early Blight.....                                | 8  |
| Figure 2.2: Leaf Infected with Target Spot.....                                   | 8  |
| Figure 2.3: Leaves Infected with Mosaic Virus.....                                | 9  |
| Figure 2.4: Leaves Infected with Tomato Yellow Leaf Curl.....                     | 9  |
| Figure 2.5: Leaves Infected with Bacterial Spot.....                              | 10 |
| Figure 2.6: Leaves Infected with Late Blight .....                                | 11 |
| Figure 2.7: Leaves Infected by Leaf Mold.....                                     | 11 |
| Figure 2.8: Leaves Infected with Septoria Leaf Spot .....                         | 12 |
| Figure 2.9: Components of an Expert System .....                                  | 17 |
| Figure 2.10: Image Classification Steps .....                                     | 18 |
| Figure 2.11: Structure of a Neural Network .....                                  | 19 |
| Figure 2.12: Simple Structure of ANN .....  | 20 |
| Figure 2.13: Image Showing Convolution Process.....                               | 22 |
| Figure 2.14: A Deep Learning Representation by a Digit-Classification Model ..... | 22 |
| Figure 2.15: Showing Steps Followed in Disease Classifier Architecture .....      | 24 |
| Figure 2.16: Evolution of Various Deep Learning Architectures.....                | 25 |
| Figure 2.17: Conceptual Framework .....   | 26 |
| Figure 3.1: Image Classification Process.....                                     | 30 |
| Figure 4.1: System Architecture .....   | 33 |
| Figure 4.2: Partial Domain Model .....  | 34 |
| Figure 4.3: Use Case Diagram .....  | 35 |
| Figure 4.4: Sequence Diagram.....   | 38 |
| Figure 4.5: Context Diagram .....   | 39 |
| Figure 5.1: Showing the Label Binarizer Output.....                               | 43 |
| Figure 5.2: Output Classes from Label Binarizer .....                             | 43 |
| Figure 5.3: Training and Validation Accuracy .....                                | 45 |
| Figure 5.4: Training and Validation Loss .....                                    | 45 |
| Figure 6.1: Precision Metrics for the Model.....                                  | 49 |
| Figure 6.2: Recall Metrics for the Model.....                                     | 50 |
| Figure 6.3: F1 Score Metrics for the Model .....                                  | 51 |

Figure 6.4: Researcher at a Demonstration Green House Field in the University of Nairobi ..... 52

## List of Tables

|  |    |
|--|----|
| Table 4.1: Descriptions of each Actor .....                      | 35 |
| Table 4.2: Normal Course of Events of each Actor.....            | 36 |
| Table 4.3: Description of the Use Cases.....                     | 36 |
| Table 4.4: Preconditions Describing the State of the System..... | 37 |
| Table 4.5: Post Conditions of each Use Case .....                | 37 |
| Table 5.1: Hardware Requirements of the Proposed Model.....      | 40 |
| Table 5.2: Software Requirements of the Proposed Model .....     | 41 |
| Table 6.1: Confusion Matrix.....                                 | 47 |

## Chapter 1 : Introduction

### 1.1 Background

Tomato is a vegetable plant grown for its edible fruit. Its botanical name is "*Lycopersicon esculentum*". Tomatoes can erect with short stems to a height of 0.7 to 2m high which are vine-like. The fruit is harvested only after one growing season therefore must be well taken care of to ensure good productivity. It is the second most consumed vegetable in the world (Foolad, 2007).

Pests and diseases cost farmers billions of shillings in wastage and poor produce. There is currently a global crisis in climatic changes and this has resulted into new attacks on plants making the effect adverse in unprepared regions (Ali et al., 2017). These changes include but not limited to rainfall, temperature, sunshine and relative humidity. This is a big threat to the second sustainable development goal, "end hunger, achieve food security and improved nutrition and promote sustainable agriculture" (Brooks, 2016).

Plant disease affects the entire ecosystem of survival and this can have a great impact on the economy. It can affect the quality of produce and the total production of food and agricultural materials. To reduce this impact, early disease detection techniques have been developed, however, they are labour intensive and time consuming (Dongyi et al., 2019).

Plant disease identification is very crucial to food production and security. However, the current practices in Africa include visual identification and microscopy. Visual examination is easily affected by cognitive error while microscopy is time consuming.

In addition, there are automatic methods which rely on digital images but require expensive devices to capture the images (Sankaran et al., 2010; Bock et al., 2010). These methods focus on a specific disease and neglect other plant problems such as healthy tissue, nutritional problems and pests (Barbedo et al., 2015).

These conventional methods also require experienced users and sophisticated instruments. The use of visual digital techniques to recommend ways of fighting the disease is still a new concept in early detection of plant disease and most of the applications are still in the experimental stages (Sindu & Bashirahamad, 2017).

Disease and nutrient identification is important as it can feed into management decisions. Early detection of disease and nutrient deficiency enables a farmer to get rid of the affected plants before the disease can spread to other plants all together. The farmer can also plan for his fertilizer input to manage crop yield based on which nutrients are deficient in the plantation. Farmers apply nutrients to their plants through chemical fertilizers and manure. Excess in this or deficiency in this can cause harm to the plant. Early detection reduces disease spread and thus improves on a farmer's production (Avila, 2006; Zhang et al., 2002).

It is difficult to detect plant disease unless one is guided by expert knowledge. Therefore, there is a need to apply machine learning techniques to make use of this expert knowledge to help farmers without access to this knowledge to diagnose plant diseases. With such a system, disease detection is automated, cost effective and non-destructive.

In the recent years, systems have been built to aid in detection of plant disease. These systems make use of image processing techniques to come up with a classification. The images are captured from a camera device and run through algorithms to detect a disease on a plant (Khadabadi, 2015).

Convolution Neural Networks have been used in several applications. They have become very popular in recent computer vision tasks such as animal classification. It is on the same premise that this research is built on. The main advantage of CNNs is their ability to identify a class from the images by automatically identifying the features without human supervision. This makes CNNs computationally efficient (Rikiya, 2018). Furthermore, Sachin (2020) suggested that deep learning solves issues to deal with handcrafted features which can affect a classification while using raw data.

Convolution Neural Networks have been used in several applications. They have become very popular in recent computer vision tasks such as animal classification. It is on the same premise that this research is built on. The main advantage of CNNs is their ability to identify a class from the images by automatically identifying the features without human supervision. This makes CNNs computationally efficient (Rikiya, 2018). Sachin (2020) suggested that deep learning solves issues to deal with handcrafted features which can affect a classification while using raw data.

The symptoms for nutrient deficiency and disease presence are not very evident to a non-plant expert. Therefore, these difficulties cause a delay in decision making hence wide spread of diseases

and reduction in crop yield. Hence this research aims to combine benefits from convolution neural networks and vision-based techniques to achieve nutrient deficiency detection and disease classification in tomato plants.

## **1.2 Problem Statement**

For a long time, farmers have depended on visual examination of their plants to detect the diseases destroying their plants. The situation gets more complicated when it comes to nutrient deficiency identification. In modern agriculture such as soilless culture, there have been limitations in determining nutrient deficiencies since common practices detect this from the soil (Beede et al., 2005; Manjula, 2016).

Deficiency of nutrient in tomatoes can have an effect on the final fruit produced. Consequences include; stunted growth, chlorosis and reduction in immunity to resist pest and disease attack (Trung-Tin et al., 2019). Plants need nutrients to grow well and produce good yield. Lack of an essential nutrient element can result into deficiencies which can be determined by visual symptoms. Farmers from rural areas are also limited in expert knowledge to detect these deficiencies and diseases at the early stages before damage. Ghaiwat et al. (2014) points out error tendencies in visual examination of crops to determine disease infection.

The result of this research can be a guide to ensure efficient use of fertilizer intake and detection of diseases affecting tomatoes. With a computer vision based model for identifying both nutrient deficiencies and diseases in plants, misdiagnose in tomatoes can be reduced. Current vision based models can have accuracy that can be enhanced by exploiting other information sources such as climate aspect. They also cannot detect multiple diseases in one image (Marko et al., 2019). Neural networks yield efficient solutions for complex problems and it is upon these efficiencies that quick diagnose of tomato leaves shall be done (Vijai, 2017).

## **1.3 Aim**

The aim of the project is to develop a computer vision model that classifies nutrient deficiency and diseases in tomato plants using vision based sensing.

## **1.4 Research Questions**

- (i) What problems are associated with the current methods of identifying tomato plant diseases?

- (ii) What are the current visual methods for plant disease identification?
- (iii) How will the vision-based tomato disease and nutrient classification model be designed?
- (iv) How will the vision-based tomato disease and nutrient classification model be validated?

### **1.5 Specific Objectives**

- (i) To investigate characteristics associated with plant disease and nutrient deficiencies
- (ii) To review existing techniques and models for predicting plant diseases
- (iii) To appraise algorithms that can be applied in computer vision based sensing for plant disease.
- (iv) To develop a computer vision-based model for identifying tomato leaf diseases and nutrient deficiencies
- (v) To test the validity of the developed nutrient and disease classification model

### **1.6 Justification**

Firstly, the model from the research can assist non-technical farmers not only to carry out inspections faster and easily but also recommend what ought to be done to the plants during fertilizer administration.

Secondly, plant health workers can benefit from the research by being able to easily diagnose plant deficiencies. This can enhance service delivery in agricultural administrative areas.

Thirdly, farmers can benefit from this research through identifying major diseases and nutrients affecting their crop yield. With this data, they can plan accordingly.

Lastly, the results of this research can be useful to academicians and future researchers by contributing to the local and international body of literature. It can be value added to existing literature by providing techniques based on new ways of applying ICTs in modern agriculture.

### **1.7 Scope and Limitation**

This research is limited to classifying eight tomato plant diseases and deficiency of one nutrient in the tomato plant leaf using a model of a computer application. The nutrient in focus is nitrogen since it is one of the most applied fertilizers in crop production. The diseases in focus include; mosaic virus, curl virus, bacterial spot, early blight, late blight, leaf mold, septoria leaf spot and tomato target spot.



## **Chapter 2 : Literature Review**

### **2.1 Overview**

This chapter is divided into sections motivated by the objectives of the research listed in chapter one. Focus shall be on challenges farmers face in identifying diseases and nutrients deficient in their crops. An overview of plant disease and nutrient deficiencies in plants. In addition, the chapter provides in-depth characteristics and details of tomato plant diseases and nutrients.

A general overview of expert systems and computer vision-based prediction systems is explained here. Algorithms of interest for this research and their application in identifying plant disease and nutrient deficiency are also described below with their objectives, methods and weaknesses and how they relate to the objectives of this research.

### **2.2 Challenges Affecting Farmers in Plant Disease Identification**

Plant disease identification is a fundamental activity in growing crops that any farmer would not take for granted. In Kenya it is mostly done visually by the human eye. However, commercial farmers apply microscopy to get in depth results usually in plant clinics and research institutes. The main drawback to visual identification is human error and cognitive prodigy which often biases the conclusions (Jasmeet et al., 2016). In regard to laboratory examinations that would prove to be very accurate, this is often time consuming and might not resonate with timely identification of diseases.

Diagnose and treatment of plant disease requires experience which most farmers in Kenya lack. Owuor (2009) observed that farmers entirely depend on extension workers to help them identify plant diseases. These too use visual methods to carry out the assessment whose drawbacks are mentioned above.

The current Kenyan system to help farmers diagnose plant disease has cost implications and this hinders agricultural growth in the region. The system suggests that an extension officer has to be physically present to observe the symptoms of the plant to identify the disease. This cost is usually passed onto the farmer if it is privately arranged, none the less, government support which ideally should be free is also subsidized (FAO, 2005). These challenges are greatly affected by low budgetary allocation by the Government (Nyang'anga, 2015).

### **2.3 Challenges Affecting Farmers in Nutrient Deficiency Identification**

Plants require more than 13 essential nutrients from the soil to produce good yield. Nutrient deficiency can have adverse effects on the growth and production of a plant. The symptoms are normally visible in leaves and include; chlorosis, distorted edges and reduction in leaf size (Jeyalakshmi & Radha, 2017).

Muhammad (2017) noted that plant nutrient deficiencies are manually conducted by experts who can have wrong judgement prone to factors that are human such as fatigue, sickness or psychological states. These in a way limit the accuracy of the diagnose making the whole process unreliable. In addition, this also involves financial costs incurred by the farmer to hire these experts. Small scale farmers might not be able to regularly afford this exercise. A shortcut to this is making use of past experience to conclude on which fertilizer to apply on the farm.

Therefore, an image-based system for tomato nutrient deficiency is relevant to significantly make the process faster, cheaper and precise. This can benefit many farmers to allocate their pesticide resources prudently.

### **2.4 Diseases Affecting Tomato Plants**

Tomato plants are affected by different diseases and this research focuses on eight of them that affect the leaves of the plant. These include target spot, mosaic virus, curl virus, bacterial spot, early blight, late blight, leaf mold and septoria leaf spot.

#### **2.4.1 Early Blight**

Early blight is a fungal disease caused by *Alternaria*. This pathogen produces many secondary metabolites (substance necessary for metabolism) which affect the host plant. It is estimated that economic-yield losses due to this disease are at 79% (Pragya, 2017). Visual signs of the disease are shown in Figure 2.1.

Early Blight disease is characterised by;

- i. Oval shaped lesions
- ii. Yellow chlorotic region across lesion
- iii. Leaf tissue between veins is destroyed



**Figure 2.1: Leaves Infected with Early Blight (David & Marcel, 2016)**

General management of this disease involves the use of adequate fertilization, fungicide application and rotating crops every 2-3 years. Early Blight can be very dangerous to the tomato plant.

#### **2.4.2 Target spot**

Infected leaves show small, pinpoint water-soaked spots. In later stages, the spots enlarge to become necrotic lesions with circles. It is also characterised by dark margins and light brown centres surrounded by a yellow halo as shown in Figure 2.2. Target spot disease is caused by *Corynespora cassiicola* (Adam et al., 2019).



**Figure 2.2: Leaf Infected with Target Spot (David & Marcel, 2016)**

#### **2.4.3 Tomato Mosaic Virus**

This is one of the viral diseases most common in the tomato plant. It is a member of the *genus tobamovirus* and belongs to the family *virgaviridae* (Najeeb et al., 2017).

Symptoms of this disease can surface at any part of the plant. The infected leaves turn dark green or mosaic and sometimes yellowing of the leaves as shown in Figure 2.3.



**Figure 2.3: Leaves Infected with Mosaic Virus (David & Marcel, 2016)**

Management of this disease is done through removing the destroyed plant to prevent further spread. In addition, it is advised that the tomatoes are planted on a 2-year rotation (Najeeb et al., 2017).

#### **2.4.4 Tomato Yellow Leaf Curl virus**

Tomato leaf curl is a viral disease belonging to the *geminivirus* family spread by white flies. It has a vast effect on the yield of the crop if not well taken care of (Henryk, 2007).

The disease is characterized by the following signs are shown in Figure 2.4:

- i. reduction of leaf size and upward curling
- ii. leaves appear crumpled
- iii. yellowing of veins and leaf margins
- iv. internodes are shorter and plant appears stunted and bushy



**Figure 2.4: Leaves Infected with Tomato Yellow Leaf Curl (David & Marcel, 2016)**

Control of the whiteflies using insecticides is effective in the control of the virus. Management techniques such as host free periods can also play a part in this. They are small bodied insects with white wings given them the name “whiteflies” (Srinivasan, 2010).

### 2.4.5 Bacterial Spot

This is a bacterial disease which starts as water-soaked spots. It is caused by four species of *xanthomonas bacteria* (*X. euvesicatoria*, *X. gardneri*, *X. perforans*, and *X. vesicatoria*). The disease development is influenced by high levels of humidity and the longer the plant is wet, the more likely the infection can occur (Portnis et al., 2015). In general, it appears as dark spots as shown in Figure 2.5.

The disease is characterised by:

- i. Small dark spots on leaves with greasy appearance and may appear transparent
- ii. Necrotic areas form around the spots on the leaves



**Figure 2.5: Leaves Infected with Bacterial Spot (David & Marcel, 2016)**

### 2.4.6 Late Blight

Nelson (2008) studied that late blight is caused by *phytophthora infestans* which is a fungus-like organism. In rainy seasons, the disease spreads rapidly and can affect the entire crop yield in a few days. General management of the disease suggests that crops be grown in glass houses where there is humidity control.

This disease mainly affects the aerial parts of the plant. As shown in Figure 2.6, it is characterised by the following:

- i. Water soaked areas on the leaf appearing as green to black shades
- ii. Leaves may become shrivelled and brown



**Figure 2.6: Leaves Infected with Late Blight (David & Marcel, 2016)**

#### **2.4.7 Leaf Mold**

This fungal disease is highly influenced by high levels of humidity and mostly appears in green house tomatoes. It is caused by the fungus *passalora fulva*. At later stages of the disease, the leaves turn yellow as shown in Figure 2.7 and drop off the main plant (Dominique, 2012).



**Figure 2.7: Leaves Infected by Leaf Mold (David & Marcel, 2016)**

It is characterised by the following symptoms.

- i. Leaves have pale green to yellow spots without clear cut margins
- ii. The spots eventually show a green to brown velvet-like fungal growth
- iii. The spots coalesce and turn brown in mature stages

#### **2.4.8 Seporia Leaf Spot**

This disease is caused by fungus *seporia lycopersici*. It is named to be the most destructive disease to tomato leaves as shown in Figure 2.8. This too is influenced by high levels of humidity (Dominique, 2012).

The symptoms of this disease are characterized by the following.

- i. the symptoms start as tiny, water-soaked spots
- ii. the spots on the leaves have a gray centre with a dark margin and can coalesce
- iii. grayish-white spots may appear under the leaves



**Figure 2.8: Leaves Infected with Septoria Leaf Spot (David & Marcel, 2016)**

## **2.5 Nutrient Deficiency in Tomato Plant**

The aim of this paper is to be able to classify the disease affecting the tomato plant as well as the nitrogen deficiency by analysing the leaf surface. Diseases in plants can be caused by several factors such as lack of nutrients, temperatures and other abiotic factors (Mabvakure et al., 2016).

Crop yield and quality is very dependent on the nutrient presence to a plant which is done through the provision of nutrients to a plant at optimal levels. This is often done through fertilizer application which is a regular practice in farming. These elements can be beneficial or detrimental to the metabolism of the plant. The response of a plant such as tomato to a specific nutrient may change with cultural practices, substrate and environmental conditions (Horald et al., 2007).

A lot of biological and chemical activity happens in the soil, but most farmers do not pay attention to these details that can have significant effects on the plant growth. Plant nutrients exist in two states; positive charges called anions and negative charges called cations. This charge difference determines the chemical reactions that keep living things alive. To be specific, nitrogen (N) is the only nutrient plants receive in both positive and negative form (Forde & Clarkson, 1999).

Nitrogen plays a major role to plant growth and its presence enhances the uptake of other macronutrients relevant for successful growth. Fertilizers and chemicals are added to plants that exhibit deficiencies to correct this abnormality. Wrong administration of these chemicals can have adverse effects on the plant growth. Furthermore, there is a problem of environmental pollution

from these chemicals that can be controlled if the nutrient deficiency is detected in a timely manner (Yuanyuan et al., 2018).

Pathological symptoms might be like those of nutrient deficiencies in tomatoes such as interveinal chlorosis, but they differ from those of nutrient deficiencies. Their overall pattern development and detail differentiate them from the latter. It is very likely that nitrogen deficiency in a plant can happen across a larger population as opposed to a disease that attacks a smaller population. This is one way in which the two symptoms are differentiated (Dominique, 2012).

## **2.6 Techniques for Identifying Plant Disease**

There are several techniques of detecting plant diseases in leaves. Some techniques are destructive where the examination is direct such as polymerase chain reaction, immunofluorescence and gas chromatography. Others are non-destructive which include human vision, thermography, fluorescence imaging and hyperspectral techniques.

### **2.6.1 Polymerase Chain Reaction**

This technique is highly used in detection of viral and bacterial pathogens in plants. This technique uses monoclonal antibodies and amplification of DNA structures of the pathogens using the technology of polymerase chain reaction (Cai et al., 2014).

Although this technique provides sensitive information about a pathogen, it is limited by lack of robustness. Its accuracy is also highly dependent on the efficiency of the DNA extraction (Van der Wolf et al., 2001).

### **2.6.2 Immunofluorescence**

This a technique carried out using a microscope to view the microbiological samples of the leaf. The leaf tissue is studied under a microscope to determine pathogenic characteristics in it. The leaf tissue sample is placed on the microscope slide and a dye is conjugated on the antibody. The detection is done by observing the spread of the studied molecule throughout the sample (Ward et al., 2004).

The main limitation to this technique is the loss of colour on the pigments when illuminated. This leads to false results from the observation. It is also destructive in nature.

### **2.6.3 Gas Chromatography**

Disease in plants could result in the release of specific volatile organic compounds (VOC). This technique works in such a way that it profiles the signatures of the volatile chemicals to detect the type of stress affecting the plant (Fang et al., 2014).

This technique provides very accurate results of the type of disease and the level of infection. It can also be used to detect the diseases at any stage of growth of the plant. However, it requires collection of VOC data which is sampled and analyzed. This process limits field application (Yi & Ramaraja, 2015).

### **2.6.4 Human Vision**

This is the most commonly used method to detect presence of a disease on the plant. The farmer checks for abnormalities on the plant leaf and other parts of the plant. The conclusion on the type of disease affecting the plant is based in the farmer's expert knowledge on the signs and symptoms (Christine, 2016).

Monica (2014) mentions that if a good inspection is done, possible causes of the disease can be deduced by eliminating any environmental causes that could produce similar signs. These factors include; temperature, air quality and moisture among others.

Human vision is considered to be very tedious and time consuming. This method can also be misleading causing wrong diagnose of leaf diseases since conclusions are based on the farmer's experience with the disease. This misdiagnose consequently leads to wrong pesticide management and disease control measurements on the plants (Christine, 2016; Rothe & Kshirsagar, 2014).

### **2.6.5 Thermography**

This method involves emission of an infrared radiation on the leaf which is captured by thermographic cameras. This allows imaging of the variations in temperatures on the surface of the leaf (Yi & Ramaraja, 2015).

Studies show that plant disease pathogens affect the stomata of the leaf affecting the loss of water from the leaf (Chaerle, 2007). Thus, the target disease can be monitored through thermography and resultant transpiration of water can be determined without external factors. This combined with the temperature changes aid in determining the infection.

The limitation to this technique is its practical application due to susceptibility to environmental conditions that would influence the measurements. Furthermore, for diseases that exhibit similar thermographic patterns, this technique cannot differentiate them (Yi & Ramaraja, 2015).

### **2.6.6 Fluorescence Imaging**

In this technique, the level of chlorophyll fluorescence is measured against the incident light on the plant leaf. Based on the changes on the photosynthetic apparatus and the transport reactions from the photosynthetic electron, the variations in the fluorescence parameters are what are used to arrive to an infection on the leaf (Bürting et al., 2011; Kuckenberg et al., 2009). The field application of fluorescence imaging is limited despite the fact measuring using this technique offers precise estimations of photosynthesis irregularities (Chaerle et al., 2009).

### **2.6.7 Hyperspectral Techniques**

This technique provides detailed information about the leaf surface and further information not visible to the human eye. It extracts features of the leaf using the full spectrum. The technique measures reflected light from the plant in numerous narrow bands. Its application spans through categorizing, modelling and detecting plant disease (Mahlein et al., 2012; Peyman et al., 2017). The main limitation is that the equipment is bulky and difficult to come by.

Most studies on identification of plant disease use digital images. Recent studies show that spectral images present more information than normal photographs. However, capturing these types of images is expensive and requires bulky equipment which might not be easily accessible to small scale farmers. With a system like this in study, access to conventional cameras and smartphone devices makes it feasible and sustainable.

## **2.7 Expert System**

An expert system is a computer tool used for solving complex decision problems in a way that an expert can. It uses both facts and heuristics based on knowledge from an expert. It is a computer program that simulates the thought process of a human expert within a particular domain. An expert system grows in accuracy as it learns from user feedback. Expert systems are very important because of the limitations of conventional human thinking. Tomato disease identification is very difficult for an individual without experience in this domain. Advantages of such a system are that human expertise is scarce, mental workload wears humans down, critical

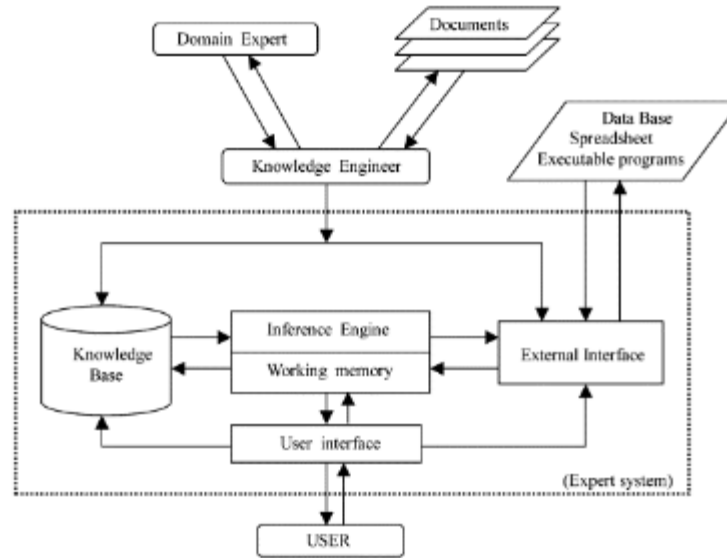
details of an observation can be taken for granted and inconsistency in human decision making (Swati & Ritika, 2013).

### **2.7.1 Components of an Expert System**

Collopy and Armstrong (2001) suggest two ways of developing an expert system. The first is that one can infer the rules the experts used to develop a model. This approach presents more accurate levels than an unaided expert. The second is studying how an expert decides in order to develop a model that represents the process. Swati and Gupta (2013) categorise expert systems into:

- a) Interpreting and identifying
- b) Predicting
- c) Diagnosing
- d) Planning
- e) Monitoring
- f) Debugging and testing
- g) Instructing and training
- h) Controlling

Jayme (2016) pointed out that effective use of an expert system to detect plant disease requires an expert to make informed decisions. An expert system can help by using its inference engine to achieve a desired outcome, as seen in Figure 2.9. It consists of an inference engine, knowledge base, knowledge gaining subsystem, human-machine interface and interpreter (GuanHuiLing, 2000).



**Figure 2.9: Components of an Expert System (Swati & Ritika, 2013)**

The inference engine and the knowledge base are the most important components of an expert system. The inference engine is the brain of the system and a program that derives answers from a knowledge base while the knowledge base stores knowledge and rules from experts (Chen et al., 2009).

## **2.8 Computer Vision Based Prediction Systems**

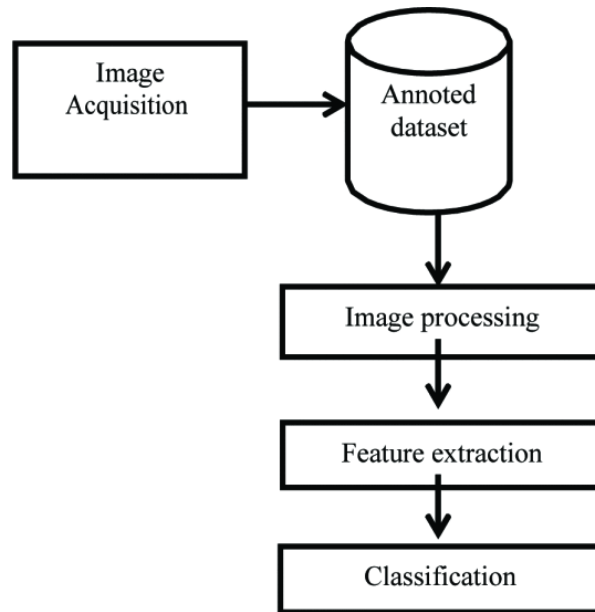
Over the years, researchers have attempted to make identification and control of plant disease easier and more accurate. Agriculture is moving away from the use of traditional methods to manage their crop yield to more sophisticated methods such as computer vision. The classification systems depend on machine learning (ML) techniques for analysis. These techniques have been used in other fields to solve complex problems such as speech recognition.

### **2.8.1 Machine Learning Techniques used in Classification of Plant Disease and Nutrient Deficiency**

Image classification has been applied in several fields and it is a process that can categorise a given image into a certain class according to the visual elements present. Machine learning algorithms offer the computer algorithm the ability to automatically learn without explicitly being programmed (David, 2015). These include support vector machine, artificial neural networks

(ANN), K nearest neighbour (KNN) classifier, and convolution neural networks (CNN). This research is focused on CNN as the algorithm to be used.

Plant diseases and nutrient deficiencies can be identified visually by observing features on the leaf, stem or fruit depending on the stress the plant has. The image processing techniques involve these key basic steps as shown in Figure 2.10 that help to arrive to the detection.



**Figure 2.10: Image Classification Steps (Shruthi, 2019)**

### 2.8.1.1 Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that analyses data used for classification through transformation to find an optimum boundary between its outputs. Its principle is based on unknown and nonlinear dependency (mapping, function)  $y = f(x)$ , where  $x$  is input vector and  $y$  is output vector.

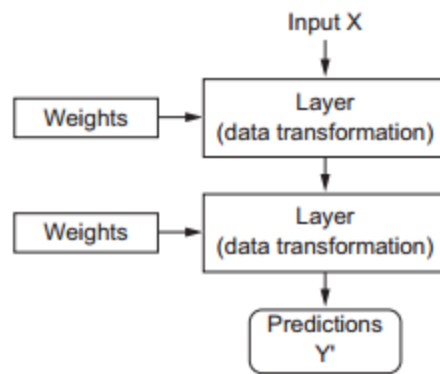
The only information used is a training dataset  $D = \{(x_i, y_i) \in X \times Y\}$ ,  $i=1, l$ , where  $l$  stands for the number of the training data pairs and hence size of the training dataset (Wang, 2005).

Arivazhagan, et al. (2013) proposed detection of unhealthy region and classification using texture features on several species of plants ; tomato, banana, mango, beans, jackfruit and sapota with an accuracy of 94.74% using SVM.

Carmago and Smith (2009) used SVM to identify visual symptoms in cotton plant disease from analysis of colored images. The algorithm flags the diseased location with different ranges of light distribution. Its limitation is that it cannot correctly classify the image from the parameters.

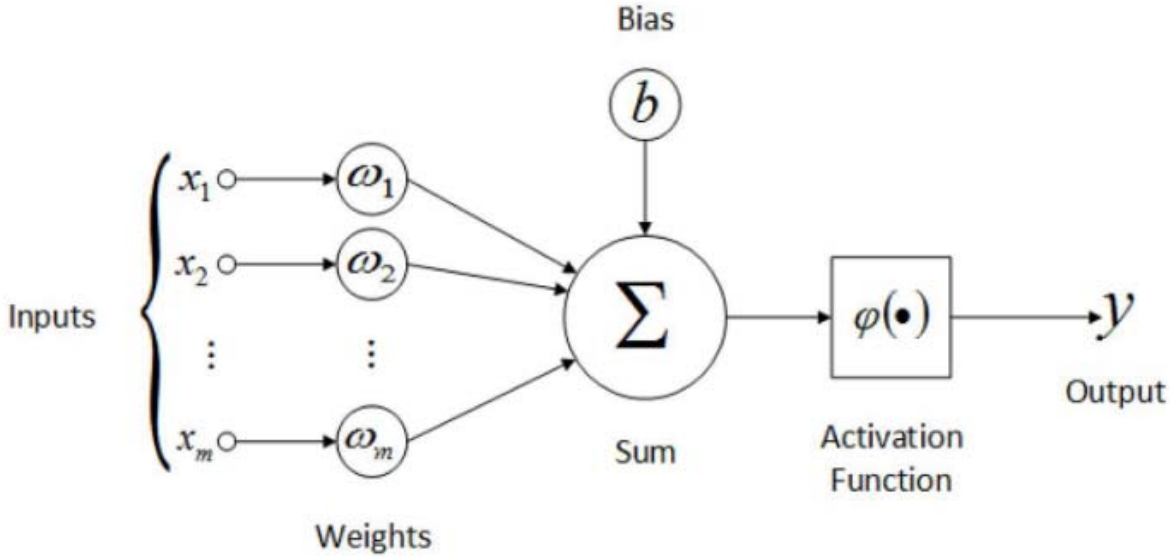
### 2.8.1.2 Artificial Neural Network (ANN)

ANNs have their genesis on the neural networks of the human brain. The algorithm learns from what it experiences. ANN uses a group of nodes referred to as artificial neurons. It has layers that specify what the network does. This process is regulated by the weights also known as the parameters of the layer as shown in Figure 2.11.



**Figure 2.11: Structure of a Neural Network (Francois, 2017)**

A single neuron in the layer which receives stimuli as input sends it to the neighboring neurons after processing is done. A simple structure of the ANN model is shown in Figure 2.12.



**Figure 2.12: Simple Structure of ANN (Bapu, 2018)**

Wang et al. (2008) proposed an ANN algorithm for detecting late blight infection spectrally on tomatoes using artificial neural network (ANN). This algorithm used gradient descent learning algorithm on two experimental samples.

### 2.8.1.3 KNN Classifier

KNN is one of the most popular algorithms in the field of pattern recognition and it is a supervised machine learning algorithm. The classification rules in this algorithm are produced by only the training dataset. KNN predicts the test class according to the K training samples which are the closest neighbours to the test sample. The prediction is made based on the largest class probability. The probability is given by equation (2.1).

$$(2.1) \quad P(X, C_j) = \sum_d \text{SIM}(X, d_i) \cdot y(d_i, C_j)$$

Where  $y(d_i, C_j)$  is a category attribute function

The similarities are calculated between all training samples and X.  $i^{\text{th}}$  sample  $d_i(d_{i1}, d_{i2}, \dots, d_{im})$

$$(2.2) \quad \text{And } \text{SIM}(X, d_j) = \frac{\sum_{j=1}^m X_j d_{ij}}{(\sqrt{(\sum_{j=1}^m X_j)^2} \sqrt{(\sum_{j=1}^m d_{ji})^2})}$$

Training categories  $C_1, C_2, C_3, \dots, C_j$  and sum of training samples is N

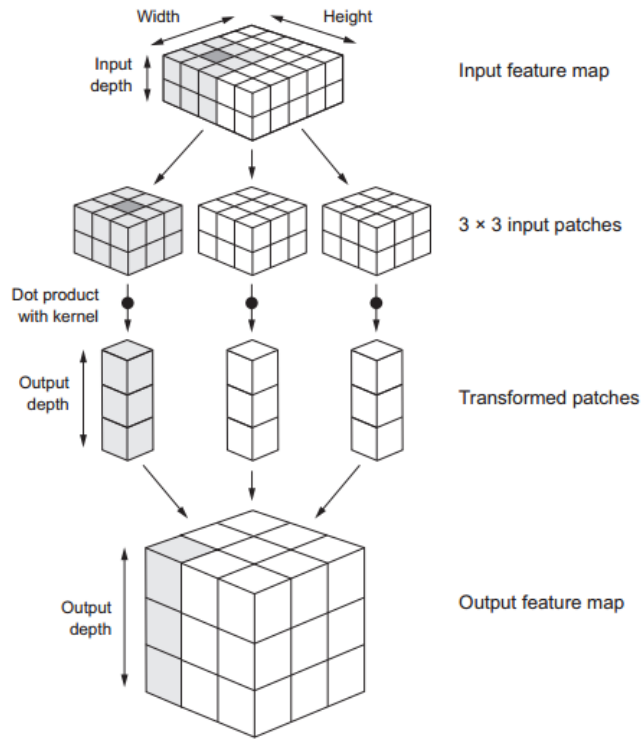
Sample X is the same feature vector of the form  $(X_1, X_2, X_3, \dots, X_m)$

Al-Hiary et al. (2011) propose KNN for classification and identification of plant disease using KNN. Their focus was on identifying the disease and future works on developing a hybrid algorithm. This algorithm was tested on five diseases such as early and late scorch, cottony, ashen mold and tiny whiteness with accuracy levels between 83% and 93%.

Suguna (2010) points out that KNN has limitations such as high calculation complexity, dependency on the training set and no weight difference which limits its application. Ranjith and Vishnu (2015) say that its main disadvantage is being a slow learner.

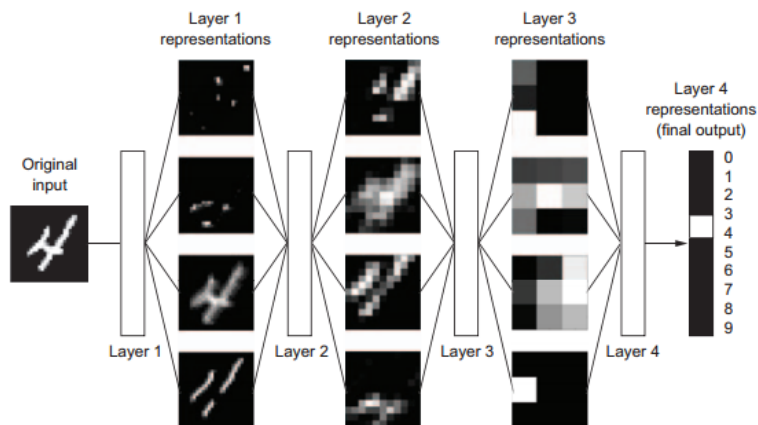
#### **2.8.1.4 Convolution Neural Network (CNN)**

In this study, deep learning plays a major role in achieving the objectives of the research. Francois (2017) defines deep learning as a mathematical framework which emphasizes the understanding of successive layers of data representations in terms of how many layers contribute to a model as shown in Figure 2.13. These layers are learned through neural networks which are described in section 2.8.1.2.



**Figure 2.13: Image Showing Convolution Process (Chollet, 2018)**

The information goes through layered filters and is outputted as a purified sample. The algorithm transforms the data image into representations different from the original image as shown in Figure 2.14.



**Figure 2.14: A Deep Learning Representation by a Digit-Classification Model (Chollet, 2018)**

The Convolution Neural Network (CovNet) is widely used for implementing the deep learning technique. It comprises of layers for feature extraction and classification. The advantage of CNN is its ability to identify the image features without supervision from the human which is good for timely detection of plant disease by non experts (Manoj et al.,2018).

Ramcharan et al. (2017) proposed a deep learning model for predicting three diseases in cassava and two types of pests. This study applied transfer learning to train the deep learning model. Transfer learning is where a model has been trained on another large dataset and retrained of new classes. The method avoids the tedious work of feature extraction. Future works on this study include widening the scope and more disease identification. It also does not have the ability to detect nutrient deficiency.

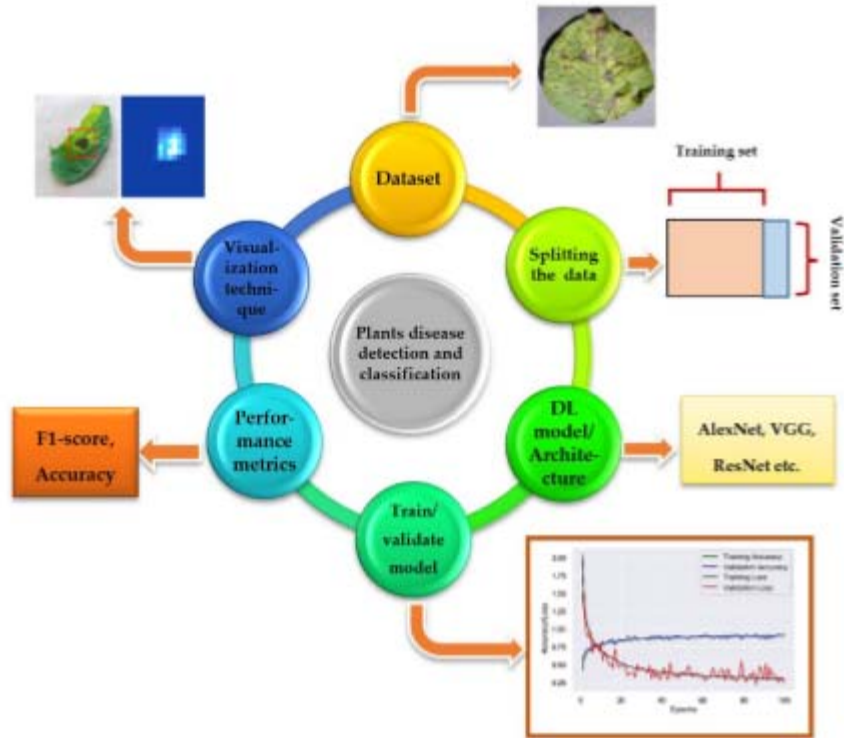
Jihen and Bassem (2017) focus on using CNN to detect two famous banana diseases based on two concepts; deep learning classification and image preprocessing. The research makes use of LeNet architecture of CNN to achieve the classification. The model learns the leaf and disease features directly from the images. Further research includes widening the scope of the classes and does not have the ability to detect nutrient deficiency.

Robert and Sook (2017) propose a deep-learning-based detector that recognises nine different diseases and pests in images. Their aim was to identify regions of interest using meta-architectures which correspond to the diseases. This research did not consider nutrient deficiencies.

## **2.9 Architectures and Designs**

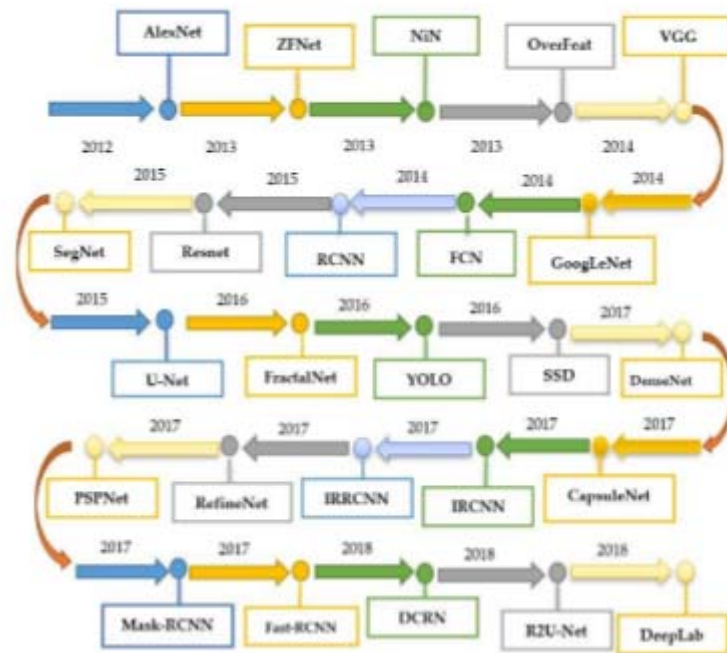
Architectures used in the design of plant disease detection systems are accompanied by several machine learning techniques. Hall et al. (2015) proposed a classification architecture through the use of a modified CNN and Random Forest (RF) classifier. This type of classification was done on 32 species whose performance was evaluated at 97.3%. Its major draw back was detection of occluded objects.

For the implementation of the architecture, a number of steps are followed to achieve a good classification as shown in Figure 2.15.



**Figure 2.15: Showing Steps Followed in Disease Classifier Architecture (Muhammad et al., 2019)**

Since the development of the AlexNet architecture, many state of the art architectures were developed and Figure 2.16 clearly shows the evolution of these architectures.



**Figure 2.16: Evolution of Various Deep Learning Architectures (Muhammad et al., 2019)**

Simple CNN architectures like AlexNet, GoogleNet and ResNet were employed for identifying tomato leaf diseases. Amongst all these architectures, ResNet showed the most promising results (Zhang et al., 2018). However, Durmus (2017) found AlexNet to be the best architecture against SqueezeNet v1.1 in terms of accuracy to classify tomato plant disease. AlexNet is known as the first CNN architecture and is supported by ReLU to achieve better performance.

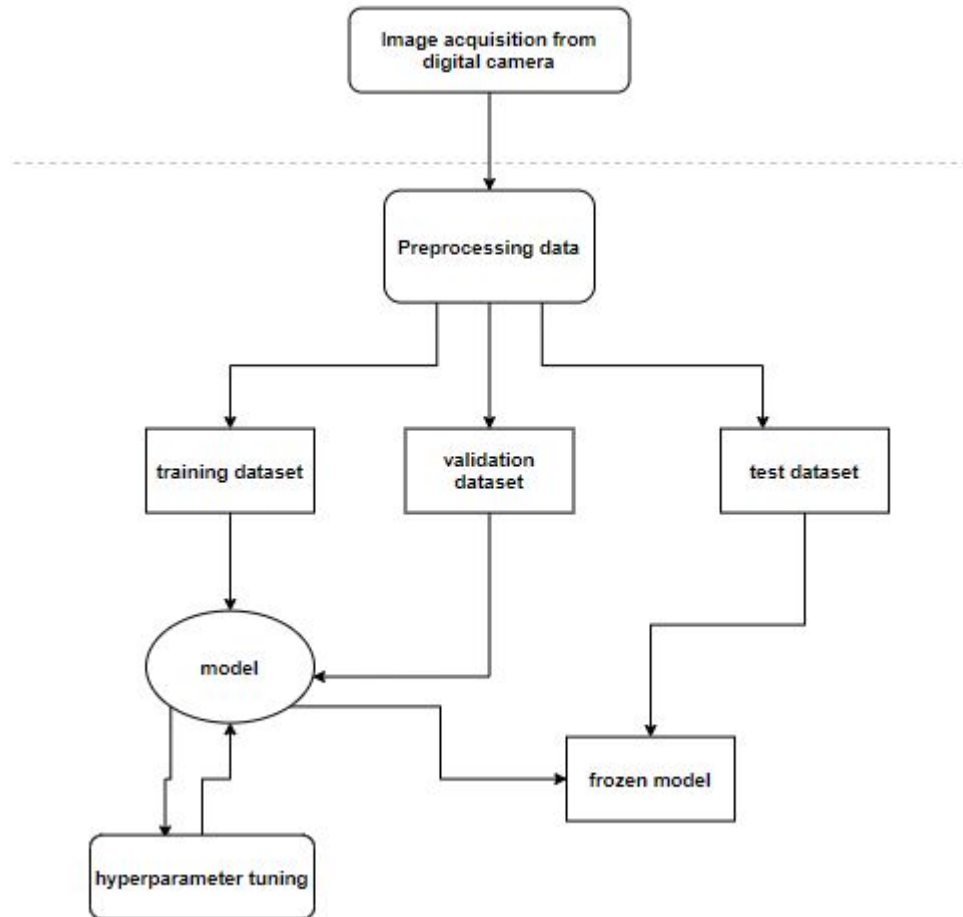
In order to classify diseases in banana leaves, LeNet architecture was implemented whose evaluation was done in color and gray scale modes (Amara, Bouaziz, & Algergawy, 2017). The downside of LeNet is that it has limited capability of computation and few parameters are used compared to other CNN architectures. There is also Inception-v3 that was used to detect cassava disease. Amongst all these architectures, no visualization technique was administered to spot the symptoms of the disease.

## 2.10 Conceptual Framework

Based on the literature review, the framework shown in Figure 2.17 combines techniques sited to come up with a model that can achieve the objectives of this research. The independent variables

are images for the diseases and nutrient deficiency and dependent variables are the diseases and deficiency type.

This framework was established to provide a solution to the problem in study in interpreting the disease and nutrient deficient symptoms in the same way a human expert would.



**Figure 2.17: Conceptual Framework**

## 2.11 Summary

Plant disease detection and nutrient deficiency is still a major challenge while using the human eye. One would require expert knowledge to achieve this. The algorithms discussed in the preceding sections describe recent research on plant disease identification through the use of images.

Understanding the characteristics of plant diseases is very important to come up with better inspection methods. Most of the techniques used require drilling and penetrating the inner surfaces, which leave the pole with further damage and potential breeding ground for microorganisms to attack.

This research proposed to combine the merits of existing techniques to come up with a model that can classify tomato plant disease and nutrient deficiency. This was achieved by harnessing the power of machine learning.

## **Chapter 3 : Research Methodology**

### **3.1 Introduction**

This chapter presents a framework of how the research was conducted. It highlights the research design and the method of choice used to carry out this research. The research was guided by the objectives intended to be met. Furthermore, the analysis of the data and the methods used in the study are also introduced together with their advantages. Lastly, ethical issues to bear in mind during this research are also summarized in this chapter.

### **3.2 Research Design**

The objectives of this research were to design a model for classifying tomato diseases and nutrient deficiency, to investigate characteristics associated with plant disease and nutrient deficiencies and to review existing techniques and models for predicting plant diseases.

The research is quantitative in nature and it sought to classify plant diseases in tomatoes. The system uses information from the camera to classify the data. The research paradigm is quantitative in nature and the data collected from the images taken can be quantified in numbers to represent the output. The method is correlational in nature because features from the leaf images can be detected to predict the outcome from an image.

### **3.3 Population and Sampling**

#### **3.3.1 Population**

Several studies have been done on plant images for disease classification which has provided large datasets on leaf images of interest in this research. Tomato leaf images was the target population mainly from the plantvillage dataset. The population consists of images of 1591 healthy tomato leaves, 1404 target spot, 373 mosaic virus, 3209 curl virus, 2127 bacterial spot, 1000 early blight, 1909 late blight, 952 leaf mold, 1771 septoria leaf spot diseases. 100 leaf images of nitrogen deficiency was collected from University of Nairobi at their Department of Plant Science and Crop Protection this was a choice since the plants were in a controlled environment on tests of nitrogen deficiency.

### **3.3.2 Sampling**

The sampling technique used was simple random sampling representation of the data. This is because probability sampling supports the law of statistical regularity which states that if on average if a random sample is chosen, it can carry the same characteristics as the population (Kothari, 2004). Sampling was done because of the random nature of the images used in the study. These images were pre-processed using data augmentation techniques.

The images collected were divided into two datasets. 80% of the images were for training the model whereas 10% for testing and 10% for validating the model.

### **3.3.3 Data Collection**

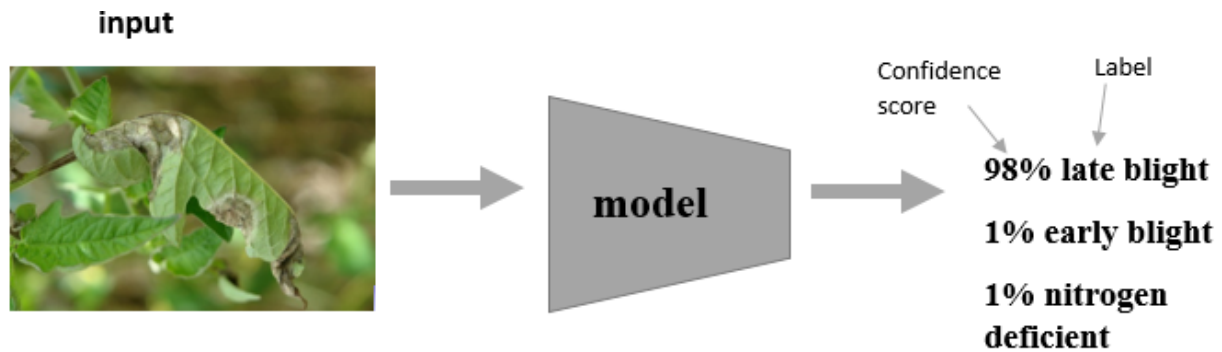
Secondary data for the tomato diseases was used in this research from a crowd data source by plantvillage and the nitrogen deficient images from University of Nairobi, Department of Plant Science and Crop Protection. The algorithm was tested on real world leaf images. The primary devices for data collection from the sources were cameras and smartphone devices which is the target device for the application. During testing, the users take pictures of the plant image using cameras or phone cameras and upload the images to the application.

The quality of the images really matters in order to build a good algorithm, therefore the data collected was labelled in their respective disease types which include target spot, mosaic virus, curl virus, bacterial spot, early blight, late blight, leaf mold, septoria leaf spot. The format of the labelling was in this format “disease\_name (number of image)”. The nitrogen deficiency images were labelled in the same manner. Since the users are not experts, the application recognises the images on their behalf to inform them of what particular disease is affecting the plant.

## **3.4 Data Analysis Methods**

The images collected were analysed using a deep learning technique to be able to classify the plant diseases. These images served as an input to the algorithm for classification. Keras and TensorFlow were the main libraries used in to build the deep learning model. Deep learning was used because the hand engineered features are time consuming. Each image had 133,225 features from their initial image size of 365x365. The features were learnt from the raw images to express ideas for plant disease classification and also modular programming which created a simple way of organising the building blocks for the model.

The model input was an image and the output provided information such as the class name of the image (i.e late blight) with a confidence score that indicates the likelihood of the image containing the class as shown in Figure 3.1. The model is a multiclass recognition model.



**Figure 3.1: Image Classification Process**

### **3.5 System Development Methodology**

#### **3.5.1 Agile System Development**

An agile development methodology was used in this research because it is incremental and iterative. Most importantly it has a theoretical background which is consistent with problem solving approaches tackled in this research.

#### **3.5.2 System Analysis**

The system detected the features on the plant images and classified the disease and whether or not there was presence of nitrogen in the plant. The camera was the primary device used to capture the images needed for analysis. Therefore, the application indicated whether a particular disease has been detected in addition to nutrient deficiency or classify it as healthy.

#### **3.5.3 System Design**

The system architecture comprised of both hardware components and software components. The hardware components are camera for capturing images and a processor whereas the software components are the plant disease classification algorithm, the software tools used such as Google Colab, programming language and its corresponding libraries.

## **3.6 System Implementation**

### **3.6.1 Model**

The research required building a representation of the model. Sketches of the system were drawn on paper models using the design thinking model.

### **3.6.2 Test Bed**

The test bed was required to test the model. This can include hardware and software. The hardware can include a mobile phone with a camera. Software components included plant disease classification algorithm used to detect the plant disease and diseased leaves to test the model. The test bed was Google Colab where all test images were inputted into the model to predict their classes. The testing was done on unknown images to the model under the test dataset.

### **3.6.3 Programming Language**

The programming language used in this research was python because of its wide array of open libraries that were easily accessed. This language easily works through the Keras and TensorFlow libraries. The algorithm can run on the GoogleColab which offered graphics processing units for running the decay algorithm.

## **3.7 Research Quality**

This was a very important aspect of maintaining accuracy in this research, therefore, validity, objectivity and reliability of the research were important. This research achieved this by limiting bias in the data collected. The research minimized the effects of extraneous independent variables that were undesirable. This was achieved by using of deep learning techniques that focussed on every aspect of the features of the images and the training can minimize their effects. The choice of the method was well explained in section 3.4.

## **3.8 Ethical Considerations**

This research upheld honesty and truthfulness and avoided any form of plagiarism, falsification and fabrication of data. In regard to the technical responsibility, the topic was related to a technical problem caused by lack of expert knowledge to identify a tomato disease. The results of the research were technically valuable and expected to benefit farmers and researchers in the field of modern agriculture.

## **Chapter 4 : System Analysis, Design and Architecture**

### **4.1 Introduction**

In this section, key system requirements are presented, both functional and non-functional used in the research. The section also discusses the system architecture and design. This includes the diagrammatic representation of the sequence diagram, context diagram and data flow diagram.

### **4.2 Requirement Analysis**

#### **4.2.1 Functional Requirements**

- i. The application should identify the type of leaf from the image
- ii. The application should detect the foliar symptoms of the diseases on the leaves
- iii. The application should identify the disease type from the image fed into it
- iv. The application should present the F1-score using the proposed algorithm for prediction accuracy levels
- v. The application should give an output of the class of image identified

#### **4.2.2 Non-functional Requirements**

##### **4.2.2.1 Usability**

The proposed users of the application are tomato farmers who would like to manage their plant yield disease identification process.

##### **4.2.2.2 Scalability**

The application should gradually improve on the accuracy levels. It should also adapt to new datasets from different environmental conditions. This can ensure good application response on new data.

##### **4.2.2.3 Storage**

The application should store the F1-score each time a prediction is made. This can be used to further train the model.

### 4.3 System Architecture

#### 4.3.1 System Model Architecture

The application uses images as the only input. The images are passed to the model for classification of the type of disease on the leaf image. In order to achieve high levels of accuracy, single leaf images should be fed into the application. The model is then trained and tested on the input data as shown in Figure 4.1.

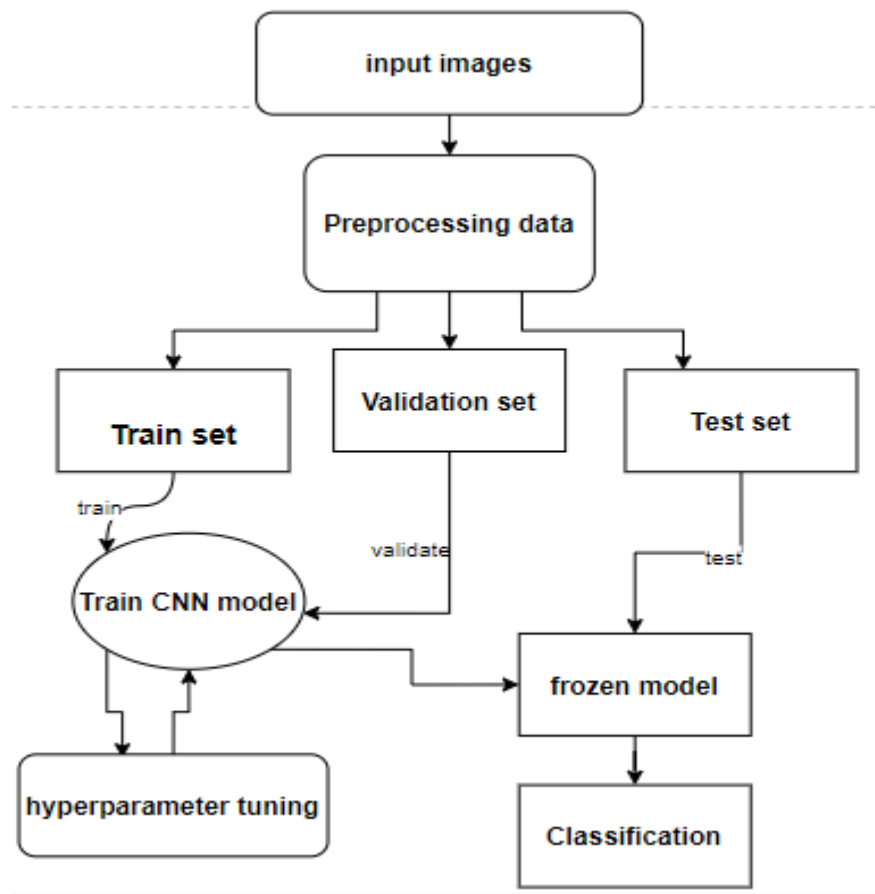
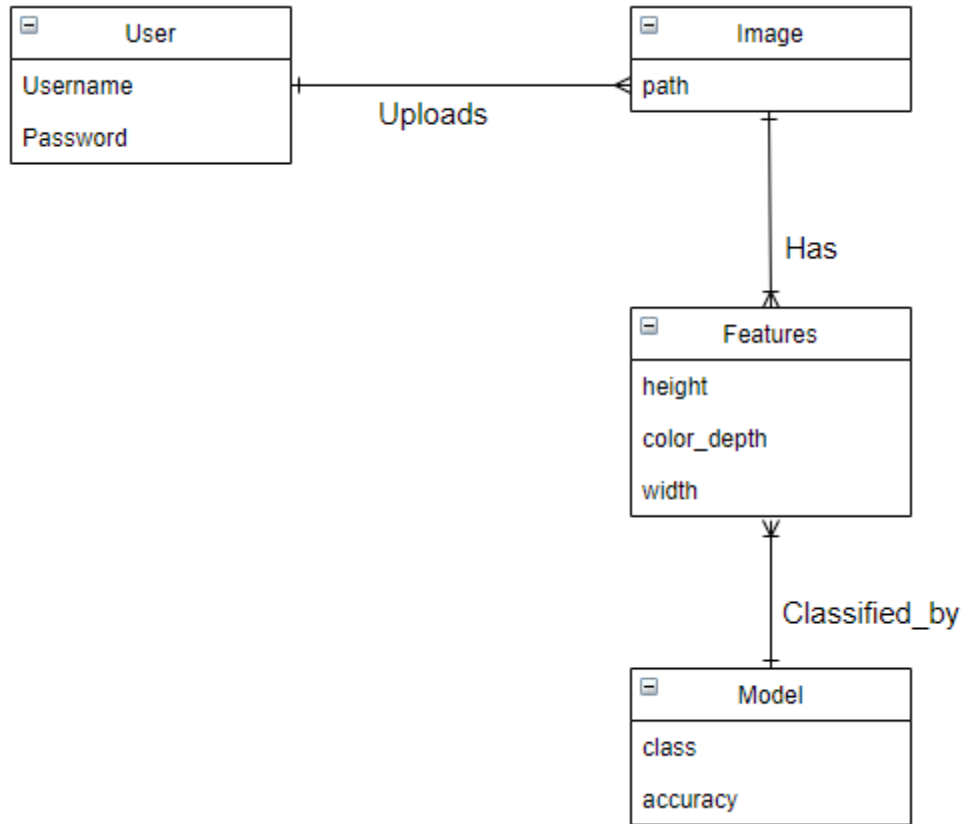


Figure 4.1: System Architecture

#### 4.3.2 Partial Domain Model

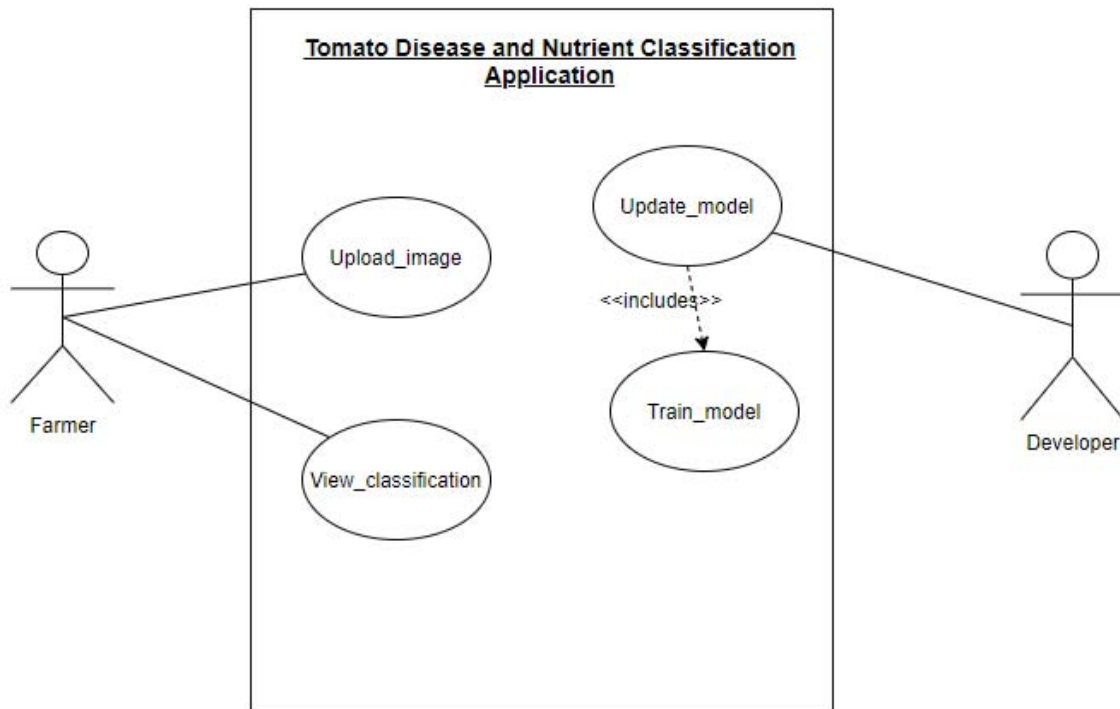
The partial domain model gives details of relevant concepts of the application. Among these are the user and how they interact with the application, conceptual classes and attributes. It is a diagrammatic representation of real-world objects such as the farmer and leaf images. The Partial Domain Model of this application is shown in Figure 4.2.



**Figure 4.2: Partial Domain Model**

### 4.3.3 Use Case Diagram

The Use Case Diagram is used to show how the users interact with the application. In this research, there are mainly two users who directly interact with the application. The first being the developer who can make system changes and the user being the farmer. The primary actor is the farmer responsible for uploading the images into the application. The developer constantly updates the application and the model for tomato disease classification as shown in Figure 4.3.



**Figure 4.3: Use Case Diagram**

#### 4.3.3.1 Use Case Basic Flows

These show the basic roles of the actors used in the use case as applied in the use case diagram. Table 4.1 clearly shows what role each actor plays while using the system.

Table 4.1: Descriptions of each Actor

| Actors    | Description   |
|-----------|---|
| Farmer    | The farmer uses the application to identify the type of disease or nutrient deficiency in the leaf of the tomato plant using images.  |
| Developer | The role of the developer is to update the model and train it on new data. The role of the farmer is to upload images and view the classifications made by the system. The use case ends when the system presents a result on the classification. |

**Normal Course of Events:**

The course of events follows the normal process in which the users interact with the system. Table 4.2 clearly describes step by step walk through on how each actor interacts with the system.

Table 4.2: Normal Course of Events of each Actor

| Actors   | Description  |
|--|--|
| <p><b>Farmer:</b> If the farmer wanted to identify the disease affecting the plant</p> | <ul style="list-style-type: none"> <li>a. The application requests for an image of the leaf to be uploaded</li> <li>b. The user uploads the image</li> <li>c. The model pre-processes the image and does the classification through the application</li> <li>d. The application returns results on the classification made</li> <li>e. The use views the results</li> <li>f. The application requests for an image of the leaf to be uploaded</li> <li>g. The user uploads the image</li> <li>h. The model pre-processes the image and does the classification through the application</li> <li>i. The application returns results on the classification made</li> <li>j. The use views the results</li> </ul> |
| <p><b>Developer:</b> If the developer wanted to make changes in the system</p>         | <ul style="list-style-type: none"> <li>a. The developer can update the model by making changes in the existing parameters</li> <li>b. The model is trained on new data</li> <li>c. The model is saved with the changes made by the developer</li> </ul>  |

The use case list in Table 4.3 clearly shows the description of each use case used.

Table 4.3: Description of the Use Cases

| Use Case Name | Brief Description |
|---------------|-------------------|
|---------------|-------------------|

|                     |  |
|---------------------|--|
| Upload_image        | The user uploads an image from the field   |
| Update_model        | The developer changes model parameters   |
| Train_model         | This is included in the update_model use case, the model is trained on new data after the update and changes saved |
| Save_model          | The model is saved   |
| View_classification | The user views the information about the prediction made   |

### Step by Step Description

### Preconditions

These represent the state of the system when the case begins as presented in Table 4.4. If these conditions are met, then the expected results are provided. These are values of system attributes.

Table 4.4: Preconditions Describing the State of the System

| Use case            | Precondition                         |
|---------------------|--------------------------------------|
| Upload_image        | Image should exist                   |
| Update_model        | Model should exist                   |
| Train_model         | New model and data should exist      |
| View_classification | Classification should have been done |
| Save_model          | New model should exist               |

### Post conditions

These include only changes made to the system. The post condition of each use case are shown in Table 4.5.

Table 4.5: Post Conditions of each Use Case

| Use case | Postcondition |
|----------|---------------|
|----------|---------------|

|                     |   |
|---------------------|---|
| Upload_image        | New image uploaded                              |
| Update_model        | New model created                               |
| Train_model         | New model trained                               |
| View_classification | Plant disease or nutrient deficiency identified |
| Save_model          | New model saved                                 |

### Exceptions

These result of either an unmet precondition or an error with the input. If image is not uploaded, the application presents an error message requesting for an input image.

### 4.4 Sequence Diagram

The sequence diagram as shown in Figure 4.4 is a combination of various objects which shows how their interactions occur to achieve the task of classifying the diseases and nutrients deficiency from the input images. The proposed application has the following classes; user, upload, pre-processor, model, save and classifier.

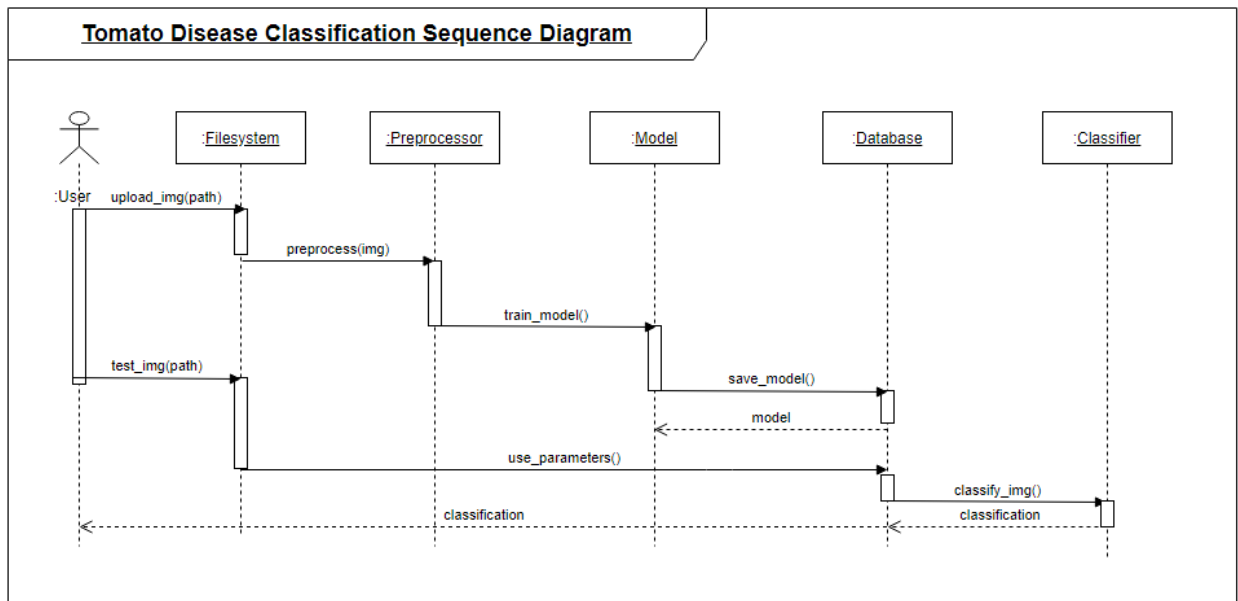
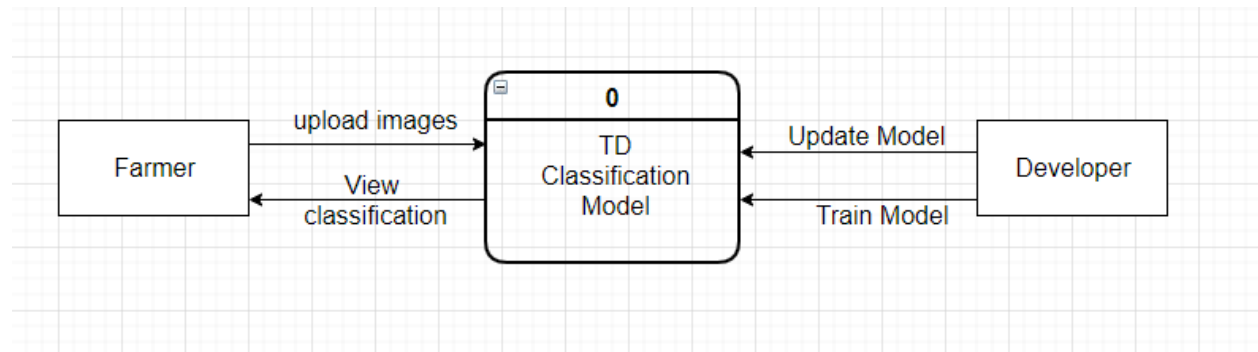


Figure 4.4: Sequence Diagram

## 4.5 Context Diagram

This captures the high-level data flow and how the users interact with the model. It shows the various types of inputs and outputs from the classification as shown in Figure 4.5. The farmer and the developer being the main users.



**Figure 4.5: Context Diagram**

## Chapter 5: Implementation and Testing

### 5.1 Introduction

This chapter describes the layout of the system. The implementation stage describes data preparation, the various steps taken to build the model and its testing. In addition, the programming tools, experimental environment are all discussed in this chapter. As described in the conceptual model in the research methodology, the model entails of training the data, carrying out a validation and testing it on a new dataset to test its accuracy. The results should be the type of plant disease or nutrient deficiency detected.

### 5.2 Development Environment

The development of the model was carried out on the Google Colab (Google Colaboratory) platform. Google Colab is a free cloud service which offers free GPU (Graphics Processing Unit) that were essential for building this model. The application of deep learning required very expensive hardware and powerful libraries such as Keras, OpenCV and TensorFlow. This was created using the Google account for resource access and was the test bed environment. All training and validations were used in the same environment.

### 5.3 Hardware Resources

The hardware requirements consisted of the main components used in Google Colab. Being a cloud-based service, there was limited control of the resources. Table 5.1 shows the details of each respective resource.

Table 0.1: Hardware Requirements of the Proposed Model

| Hardware     | Details |   |
|--------------|---------|---|
| Google Colab | GPU     | 1xTesla K80 , having 2496 CUDA cores,<br>compute 3.7, 12GB(11.439GB Usable)<br>GDDR5 VRAM |
|              | CPU     | 1xsingle core hyper threaded  |

|  |      |   |
|--|------|---|
|  |      | i.e(1 core, 2 threads) Xeon Processors @2.3Ghz<br>(No Turbo Boost) , 45MB Cache |
|  | RAM  | 12.6 GB Available   |
|  | Disk | 320 GB Available  |

**5.4 Software Resources**

The software consisted of the main language used in the development and the respective libraries used to build the model. Table 5.2 shows the main libraries of python that were used. Jupyter notebook on Google Colab was used for code writing of the machine learning model. As described in section 3.6.3, the programming language in use was python and all the corresponding software components used to build the model responded to this language.

Table 0.2: Software Requirements of the Proposed Model

| Software   | Details    |           |
|------------|------------|-----------|
| Python 3.6 | Libraries  | Version   |
|            | TensorFlow | Version 1 |
|            | Keras      | Version 2 |
|            | Numpy      | 1.16.2    |

**5.5 Model Components**

**5.5.1 Neural Network Components**

**5.5.1.1 Storage**

The images collected were stored using the Google drive platform. The drive is a storage platform developed by Google to enable storage of files and can be synchronized across different devices. This choice enabled faster loading of the large dataset which was called easily on the Google Colab

platform. The Jupyter notebooks were also stored on the drive. This efficient way of storage was cheaper since one does not require purchase of hardware components to store the data.

### **5.5.1.2 Input Layer**

This was the first layer of the network which interacts with the external environment. This is the first layer which received the images from the file system storage as described in chapter four. This layer expected input of images of size (256,256,3). The images were in the form of 2D arrays.

### **5.5.1.3 Output Layer**

This was the last layer of the network. Within it were the 10 classes of the diseases and nitrogen which the model classified. It received all the outputs from the hidden layers and presented its output to the external environment.

## **5.6 Image Processing**

The CovNet's input were tensors of shape (image height, image width and number of channels), in this model, the input layer taken in was (256,256, 3) since the images are RGB as shown in Appendix A. This was the original shape of the pictures. In order to avoid distortion of the resolution, the size was maintained as they were fed into the model.

Nitrogen images which were collected from test fields in the University of Nairobi were converted to the same size as the plant village dataset (256,256). All the 100 images were converted and added to the file system.

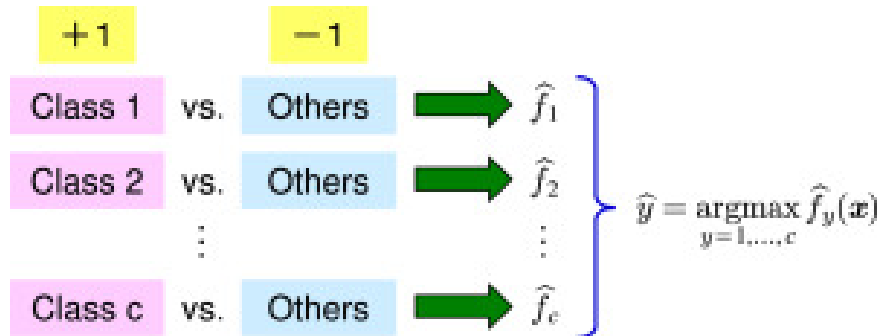
### **5.6.1 Fetching Images from the Root Directory**

From the root directory, the image labels were extracted based as the names of the diseases and these labels were stored as a list of classes. In total they were 10 classes based on the diseases and deficiency to be detected by the model. The total number from all classes consisted of 10,000 images belonging to 10 classes.

### **5.6.2 Transforming Image Labels**

The image labels were transformed into binary labels using the scikit learn label binarizer. The solution was a multiclass classification model, therefore, the classes had to be assigned binary labels. The purpose of this was to create a label indicator matrix which was introduced to reduce

multiclass classification into binary classification. The method used was the one versus all shown in Figure 5.1. It considered one possible outcome for each class.



**Figure 0.1: Showing the Label Binarizer Output (Bishop, 2006)**

The code to initiate the label binarizer is shown in Figure 5.2. The output is a matrix of 0s and 1s representing the different labels hot encoded into binary.

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer, open('label_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

```
print(n_classes)
print(image_labels)
```

```
10
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 1]
 [0 0 0 ... 0 0 1]
 [0 0 0 ... 0 0 1]]
```

**Figure 0.2: Output Classes from Label Binarizer**

### 5.6.3 Data Augmentation

The training dataset and validation data set were augmented so that the image does not train on the same pictures more than once, to avoid over fitting. The code snippet below shows the different

alterations that each image had to undergo. The model cannot view the same image more than twice.

```
train_datagen= ImageDataGenerator(  
    rescale=1./255,  
    rotation_range=40,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
)
```

## 5.7 Model Implementation

The model was compiled on a loss function of binary cross entropy. This was the choice of function because the images represented a multiclass grouping. Meaning they were mutually exclusive. The Adam optimizer was used. Adam was the best choice because of its ability to combine the benefits of AdaGrad and RMSProp which can manage noisy datasets in deep learning.

The hidden layers consisted of convolution layers, batch normalization, dropout and max pooling layers. The main purpose of the hidden layers was to provide maximum output to what was expected in the output layer. These were stacked in between the input layer and the output layer as shown in Appendix B.

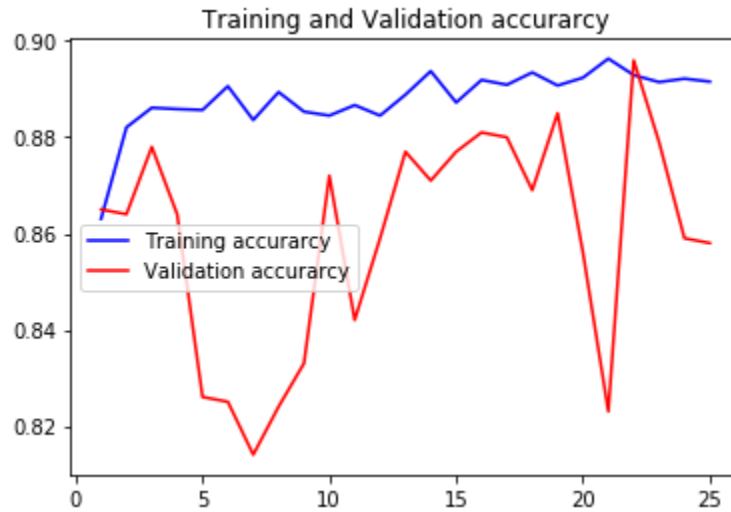
## 5.8 Training and Testing

### 5.8.1 Model Training

The model was compiled on a loss function of binary cross entropy. This was the choice of function because the images represented a multiclass grouping whose labels were transformed into binary through the label binarizer. The Adam optimizer was used. This is the method through which the network updates itself.

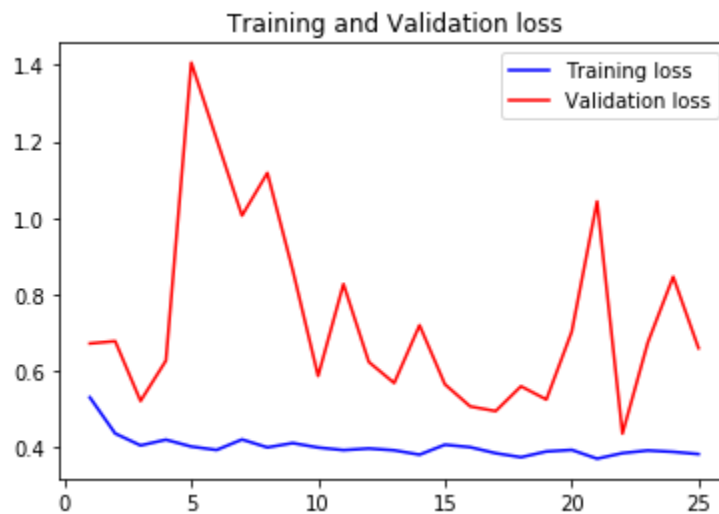
The progress of the model training is shown in Figure 5.3. The training accuracy was higher than the validation accuracy. The validation accuracy curve had several dips. This showed the model

did not overfit the data. The performance was tracked over the time as shown in Appendix C. The x and y axis represent epochs and accuracy values respectively.



**Figure 0.3: Training and Validation Accuracy**

In the progress of the model training is shown in Figure 5.4, the training loss was lower than the validation loss. The validation loss curve had several dips. The performance was tracked over the time. The x and y axis represent epochs and loss values respectively.



**Figure 0.4: Training and Validation Loss**

### 5.8.1.1 Hyper Parameter Tuning

This was done by adjusting mainly three variables: the optimization, the learning rate and the number of epochs. The learning rate took values of [0.00001,0.0001,0.001]. The optimizers used were SGD, Adam and Adagrad. The epochs used in training consisted of [20,30,50]. This was possible because of the use of Colab which gave accessibility to its GPUs.

Hyper parameters determined how a model is structured. They can be used for controlling system requirements such as cost on memory which was an important aspect to a model. The hyper parameters adjusted in this model are as follows:

- i. **Learning Rate:** this is the most important of all hyper parameters, it assigns a quantity to the progress of learning the learning rate quantifies the learning progress of a model in a. If the learning rate is of a low value, updates are few and optimization speed is reduced whereas a high value, the optimization diverges. The learning rate used for the model was 0.001.
- ii. **Epochs:** this is a parameter which separates the training into different stages. The higher the number of epochs, the higher the probability of having an over fitted model. The epochs used for the model were 20
- iii. **Optimizer:** adaptive moment estimation Adam is the most commonly used for deep learning models for computer vision. It works in such a way that it calculates an exponential moving average of the gradient and the squared gradient.

## 5.9 Model Testing

The evaluate function is used to test the model accuracy across the test data that is stored in the test directory. The model achieved an accuracy of 85%. This was done on timings of 3ms/step.

## Chapter 6: Discussions

This research's main objective was to come up with a model for predicting 9 diseases that commonly affect tomatoes and nitrogen deficiency in the leaves using computer vision. This presented a total of 10 target classes for the model. The part of the plant used in this study was leaves. In this section, we shall discuss the validation and how the research questions have been answered.

### 6.1 Validation of the Model

The validation of the model was carried out using the confusion matrix. This presented the values of images correctly identified for each class. It is a general way in which the number of instances of class **A** are classified as class **B**, each row in the confusion matrix represents a single class of the diseases (Géron, 2019). For example, where the number of instances a healthy tomato leaf is classified as yellow leaf disease. This was the performance measure of the classification used for the 10 classes. From the matrix in Appendix D, the classes with fewer images presented low accuracy levels compared to classes with higher samples. By viewing the matrix, one can have a broader picture of how the model is performing. This was done on a total of 100 instances. Table 6.1 presents the matrix of the classifications.

Table 0.1: Confusion Matrix

| Input          | Bacterial spot | EarlyBlight | Healthy | LateBlight | LeafMold | MosaicViruses | Nitrogen | SeptoriaLeaf | TargetSpot | YellowLeaf |
|----------------|----------------|-------------|---------|------------|----------|---------------|----------|--------------|------------|------------|
| Bacterial spot | 14             | 1           | 0       | 0          | 0        | 0             | 0        | 0            | 0          | 0          |
| EarlyBlight    | 0              | 2           | 0       | 1          | 0        | 0             | 0        | 1            | 0          | 0          |
| Healthy        | 0              | 0           | 6       | 0          | 0        | 0             | 0        | 0            | 1          | 0          |
| LateBlight     | 1              | 1           | 0       | 15         | 0        | 0             | 0        | 4            | 0          | 1          |
| LeafMold       | 0              | 0           | 0       | 0          | 6        | 0             | 0        | 0            | 0          | 1          |
| MosaicViruses  | 0              | 0           | 0       | 0          | 0        | 1             | 0        | 0            | 0          | 0          |
| Nitrogen       | 0              | 0           | 0       | 5          | 0        | 0             | 1        | 0            | 0          | 0          |
| SeptoriaLeaf   | 1              | 0           | 0       | 2          | 0        | 0             | 0        | 11           | 2          | 0          |

|            |   |   |   |   |   |   |   |   |   |    |
|------------|---|---|---|---|---|---|---|---|---|----|
| TargetSpot | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0  |
| YellowLeaf | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 |

There were a total of 100 instances that were used to validate the model. A total of 15 instances of bacterial spot disease target were presented to the network and 14 were correctly classified and 1 was wrongly classified. 4 instances of early blight disease were presented to the network and 2 were classified correctly, 1 was wrongly classified as late blight disease and the other classified as septoria leaf disease. 7 instances of healthy leaves were presented and 6 were classified correctly whereas 1 was wrongly classified as target spot disease. 22 instances were presented to the network as late blight disease and 15 of them were classified correctly, however, 4 were wrongly classified as septoria leaf disease, 1 as yellow leaf disease, 1 as bacterial spot disease and 1 as early blight disease. 7 instances of leaf mold disease were presented, 6 were classified correctly and 1 was classified wrongly. Mosaic virus disease was presented with 1 instance which was correctly classified. 6 instances of nitrogen were presented to the model, 1 was classified as nitrogen and 5 were classified as late blight disease. 16 instances of septoria leaf disease were presented, 11 were classified correctly, and 2 were classified wrongly as target spot disease, 2 as late blight disease and one as bacterial spot disease. 6 instances were presented as target spot disease, 3 were classified correctly whereas 1 was classified wrongly as bacterial spot disease, early blight disease and healthy. Lastly, 16 instances of yellow leaf disease were presented to the model, 15 were correctly identified as yellow leaf disease and 1 was incorrectly identified as bacterial spot disease.

From all these, mosaic virus was 100% accurately identified due to the single image presented to it. The second was nitrogen that had poor performance mainly due to variations in the source of the data since this presented varying conditions as opposed to the diseases. Most of the wrong classifications ended up in bacterial spot, which meant that this particular disease had similar characteristic features with most of the diseases presented in this research. For a farmer, this could be very difficult to differentiate by use of eyesight. One advantage of the model is that only one image presented was wrongly classified as healthy, which means that model accurately differentiates diseased from non-diseased leaves.

## 6.2 Classification Report of the Model

The classification report provided the summary of how the model performed for each particular class of images. This report measured the degree of authenticity of the predictions that the model made. In the methods reviewed in the literature, it is common to base on only accuracy for validity of a model. However, more metrics such as precision, recall and F1-score provide an in depth analysis of the model performance on new data (Géron, 2019). With this in place, the model was validated on test data to view the true predictions and false predictions in terms of true positives, false positives, true negatives and false negatives.

### Precision

This presents the accuracy and positive predictions of each class given by the equation 6.1:

$$(6.1) \quad \text{precision} = \frac{TP}{TP+FP} \quad (\text{Géron, 2019})$$

where TP stands for True positive and FP stands for False positive. Figure 6.1 shows the precision metrics from the model.

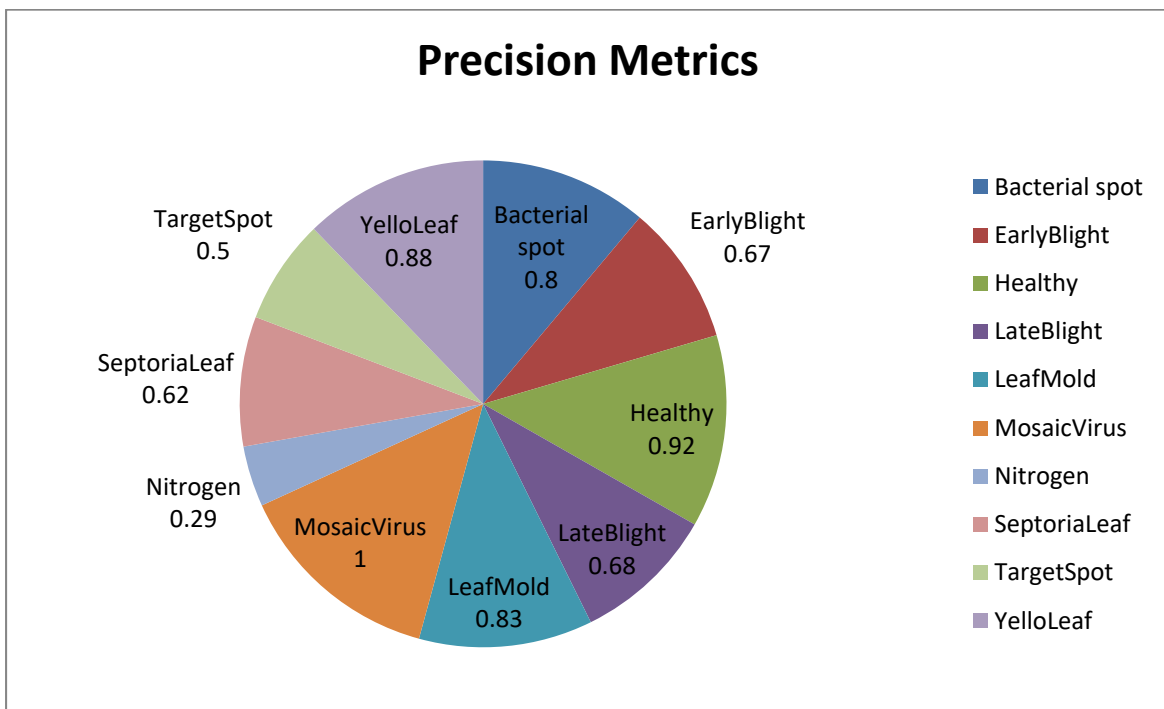


Figure 0.1: Precision Metrics for the Model

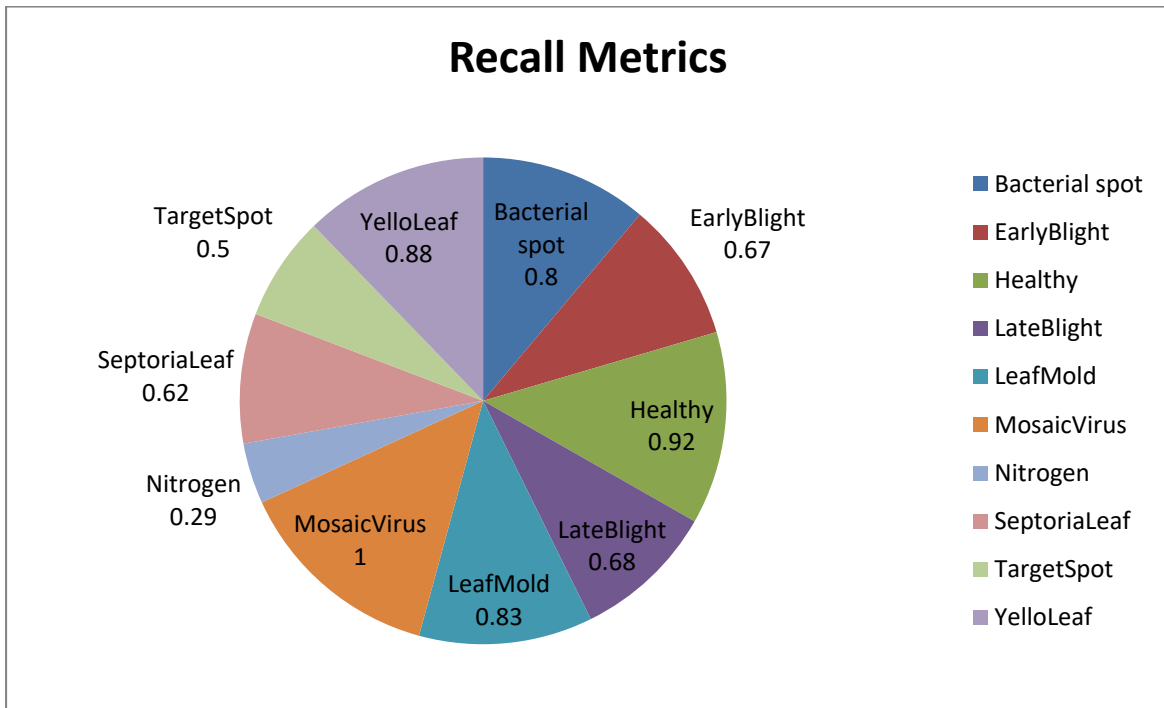
Mosaic Virus and Healthy presented the highest precision score because they were the most correctly predicted. Nitrogen performed worst in terms of prediction and presented the worst precision score. Precision alone might not present the best of conclusions of the model performance, therefore recall was introduced to back it up (Géron, 2019).

### Recall

This gives the ration of positive instances which are correctly identified by the model. It is given by the equation 6.2:

$$(6.2) \quad \text{recall} = \frac{TP}{TP+FN} \quad (\text{Aurélien, 2019})$$

where FN stands for false negative. Figure 6.2 shows the recall metrics from the model.



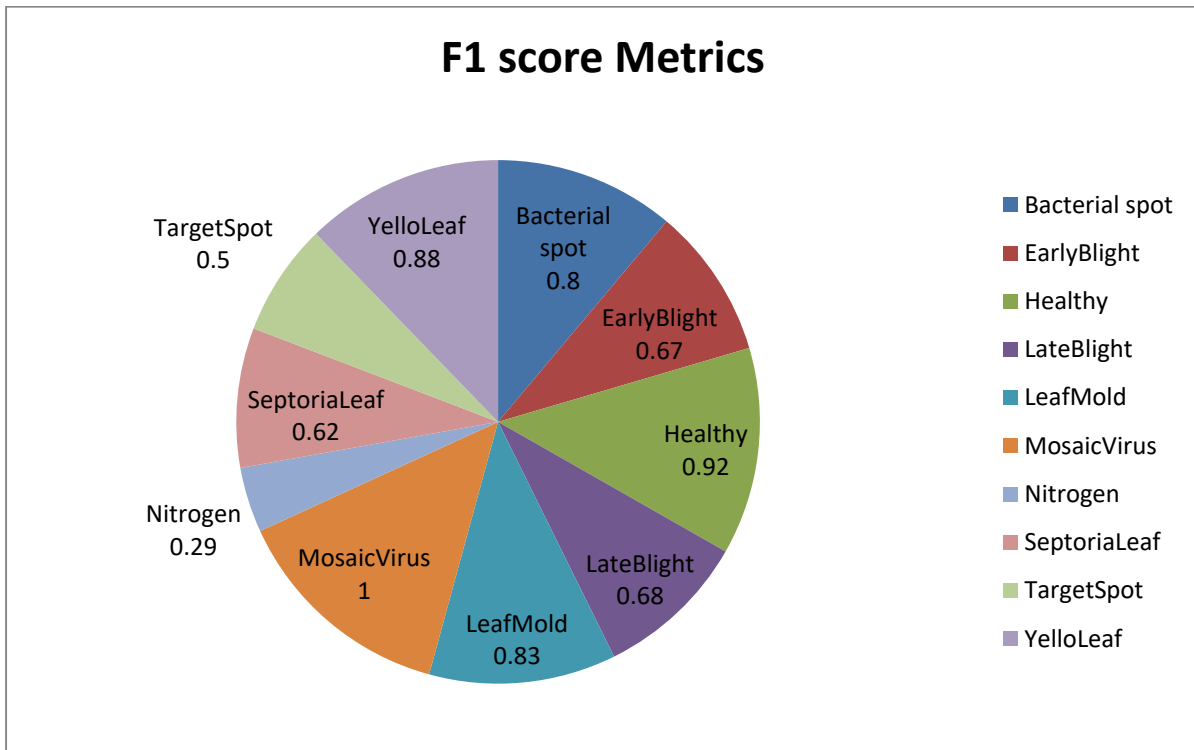
**Figure 0.2: Recall Metrics for the Model**

Mosaic Virus and Healthy presented the highest recall score because they were the most correctly predicted. Nitrogen performed worst in terms of prediction and presented the worst recall score.

## F1-Score

This gave the harmonic mean of the precision and recall values. Harmonic mean assigns more weight to low values. Whereas ordinary mean treats all values equally. Therefore a target gets a high F1-score if both its precision and recall are high. The F1 score is given by the equation 6.3 and shown in Figure 6.3:

$$(6.3) \quad F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2x \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{FN+FP}{2}} \quad (\text{Géron, 2019})$$



**Figure 0.3: F1 Score Metrics for the Model**

Mosaic Virus presented the highest F1 score because it had the highest precision and recall values.

### 6.3 Research Contribution

The model provided a more efficient way in which farmers can be able to detect diseases and nitrogen deficiency in their fields. Detection of nitrogen deficiency can facilitate prevention of diseases. This eventually can lead to more yield from the plantation hence contributing to food security in the long run since nitrogen is a limiting factor that does not exist in the soil.

Automatic detection of diseases eliminated the use of expert workers who sometimes might incorrectly diagnose plant diseases. This also contributed the reduction in cost of managing crop yield since the farmer can regularly check on the crop leaves for potential deficiency and disease without spending money on an expert. The primary data collected from University of Nairobi also contributed to data availability for further research to academics interested in similar research.

#### **6.4 Achievement of the Objectives**

In this section, a summary of how the objectives of this research were achieved. All the objectives were outlined in section 1.4 and their achievements are mentioned as follows:

The first objective was to investigate the characteristics associated with plant disease and nutrient deficiencies. A visit to the University of Nairobi, Department of Plant Science and Crop Protection provided the researcher with deeper insights in the relationship between plant disease and nutrient deficiency as shown in Figure 6.4.



**Figure 0.4: Researcher at a Demonstration Green House Field in the University of Nairobi**

Nitrogen is a limiting factor which is not produced by the tomato plant therefore the most common way of ensuring that nitrogen is present in the soil is by growing different crops in every season. For example, beans produce nitrogen in the soil and planting tomatoes in the next season would be a good strategy. Detecting nitrogen would enhance yield and would avoid potential disease cause to the tomato plant. Lack of nitrogen weakens the plant's immunity and presents them to disease causing organisms easily.

The plant diseases in question share certain similarities with each other. For example, certain features were common across all the diseases and the probability of the classes given by the model gave a very good prediction to what extent a particular image. The features include interveinal chlorosis, marginal chlorosis, uniform chlorosis, necrosis, distorted edges, and reduction in size of the leaf. The nitrogen deficient leaves present a yellow coloring as discussed in the literature review, these same symptoms could be present in diseased leaves such as tomato yellow leaf curl virus. For a farmer, these two could present similar signs and would be difficult to differentiate, however, with the model; the images fed in can be a good ground for identifying the differences.

The second objective was to review existing techniques and models for predicting plant diseases. This fed into the third objective which was to appraise algorithms that can be applied in computer vision-based sensing for plant disease. Their review in chapter 2 helped the researcher to understand the existing solutions that are currently in use and how their application has helped agricultural modernization. The models in review were noticed to have been applied on several other datasets. This enhanced the scope of the research and common areas of interest to researchers. Their weaknesses and strengths were discussed and their shortfalls aided in development of this model used on tomato leaves. The main model was the convolution neural network which was applied in this research combining nutrient deficiency together with plant disease as a contributor to poor yield in tomato plants. It was also noted that the use of human vision is far much less efficient compared to computer vision due to human limitations and lack of prior expert knowledge of potential symptoms of disease on leaves.

The fourth objective was to develop a computer vision-based model for identifying tomato leaf diseases and nutrient deficiencies. The detection of nutrient and disease affecting a tomato plant was implemented in an automatic way which was the main objective of the study. The research methodology was clearly stated in chapter 3 on how the development was done. The system

analysis then followed in chapter 4 on how farmers will be able to interact with the system to detect the diseases. The final implementation was clearly stated in chapter 5. This gave an account of how the model was built, trained and tested.

The model output made it easier for farmers without expert knowledge to be able to determine the type of disease affecting the tomato plant as well as the nutrient deficiency. There is a direct correlation between nutrient deficiency and plant disease. Lack of nitrogen which is a limiting factor in tomatoes makes the plant more susceptible to disease attack. The absence of nutrients in plants will reflect in the leaves. This is one area of a plant in which a disease can manifest itself where expert workers can be able to conclude on a disease attack or nutrient deficiency. Since nitrogen deficiency weakens the plant, it was important to detect its absence in the leaves using the computer vision model. The images that were fed into the model presented similar characteristics as those written on in the literature review.

The model developed in this research made disease and nutrient deficiency detection more accurate as opposed to the methods described in the literature review. The use of a machine learning algorithm gives the farmer a faster and more accurate feedback.

The fifth objective was to test the validity of the developed model. The model was tested and achieved an accuracy of 85%. This accuracy level was good and acceptable compared to existing plant computer vision models. The testing was done using the test dataset which was unseen data to the model. The validity was carried out using 3 metrics; the precision, recall and F1 score. All these metrics' results were outputted in pie chart form for the researcher to analyse the model performance.

## **6.5 Limitations to the Research**

- i. The model did not consider other parts of a plant in which diseases and nutrient deficiency could manifest through.
- ii. The model was limited to only one type of nutrient being nitrogen.
- iii. The model does not provide solutions to manage the disease nor the nutrient deficiency.



## **Chapter 7: Conclusion, Recommendations and Future Work**

### **7.1 Conclusion**

The main purpose of this project was to develop a computer vision model to detect presence of disease and nutrient deficiency in tomato plant leaves. The 10 classes trained on the model included diseases: mosaic virus, curl virus, bacterial spot, early blight, late blight, leaf mold, septoria leaf spot and tomato target spot and nitrogen deficiency.

The research was divided into milestones to complete this project. These included literature review where challenges facing farmers and computer vision prediction systems were reviewed. These have all been very fundamental in the research arena. The research methodology in chapter 3 described the framework used in carrying out the research. The key system requirements in chapter 4, the implementation of the model and testing were carried out in the preceding chapter and finally validation of the model in chapter 6. The analysis on the datasets was carried and the output of the model was the type of class identified by the model.

With such a solution, the model can adapt to new data in different environments. From these, the project studied the various complex features that are present in leaves which an ordinary farmer might not be able to detect using a photo sheet of even.

Addressing the issue of food security in the world today, effective ways of enhancing crop yield are coming up. The use of technology is also on the rise thus, there are increasing numbers in data collection which ordinary systems might not be able to process. Hence the need for machine learning models that can be able to process this data faster.

### **7.2 Recommendations**

Plant disease affects the entire ecosystem which can affect the total production of food in the country. It would be important to combine the results from this research with existing solutions in order to achieve sustainability and food security. In addition, the model will have to be trained regularly using new data in order to learn more. The researcher would also recommend that farmers embrace these technologies since they are more accurate than human vision methods that are currently practiced. This would enhance early disease detection to improve yield.

### **7.3 Future Work**

The limitations of this research present further areas for future works that would improve further application of this model.

- I. The model should accommodate other parts of the plant in which diseases can manifest such as the stem and fruit.
- II. The model should be trained on other nutrient deficient images such as magnesium, potassium and phosphorus which compose full fertilizers.
- III. The model should be enhanced with best practices to manage disease and nutrient deficiency once detected.
- IV. More data on overall field images should be collected for a wider scope of the disease extent of nutrient extent.

## References

- Al Hiary, H., Bani Ahmad, S., Reyalat, M., Braik, M., & Alrahamneh, Z. (2011). Fast and Accurate Detection and Classification of Plant Disease. (pp. vol. 17, no.1, pp 31-38). Wageningen Academic publishers.
- Allen, V. B., & David, J. P. (2016). *Handbook of Plant Nutrition*. London: CRC Taylor and Francis.
- Amara, J., Bouaziz, B., & Algergawy, A. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. In *Proceedings of the BTW (Workshops)*, (pp. 79-88). Stuttgart, Germany.
- Arivazhagan, S., Newlin, S. R., Ananthi, S., & Vishnu, V. S. (2013). Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Commission Internationale du Genie Rural(CIGR) journal*, vol.15, no.1, pp:211-217.
- Avila, Y. (2006). Evaluation of *Frankliniella bispinosa*. *Florida Entomologist*, 89, 204-207.
- Barbedo, J., Tibola, C., & Fernandes, J. (2015). Detecting Fusarium head blight in wheat kernels using hyperspectral imaging. *Biosystems Engineering*, 131, 65-76.
- Beede, R. H. (2005). Diagnosing and correcting nutrient deficiencies. In L. Ferguson, *Pistachio Production Manual. 4<sup>a</sup> Ed.* University of California, Davis.
- Bishop, C. . (2006). *Pattern Recognition and Machine Learning*. . Springer.
- Blancard, D. (2012). Identification, Biology and Control. In D. Blancard, *Identification, Biology and Control: A Colour Handbook, Second Edition* (p. 378). CRC Press.
- Bock, C. P. (2010). Plant disease severity estimated visually by digital photography and image analysis and by hyperspectral imaging. *Critical Reviews in Plant Sciences*, 29,59-107.
- Brooks, J. (2016). Food Security and the Sustainable Development Goals. In L. Patrick, *Debate the Issues: New Approaches to Economic Challenges*. Paris: OECD Publishing.

- Bürling, K., Hunsche, M., & Noga, G. ( 2011 Sep 15). Use of blue-green and chlorophyll fluorescence measurements for differentiation between nitrogen deficiency and pathogen infection in winter wheat. *J Plant Physiol*, 168(14):1641-8.
- Cai, H., Caswell, J., & Prescott, J. (2014 ). Nonculture molecular techniques for diagnosis of bacterial disease in animals: a diagnostic laboratory perspective. *Vet Pathol.* , 51(2):341-50.
- Camargo, A., & Smith, J. S. (January 2009). An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems Engineering*, Volume 102, Issue 1, Pages 9-21.
- Chaerle, L., Leinonen, I., Jones, H., & Van Der Straeten, D. (2007). Monitoring and screening plant populations with combined thermal and chlorophyll fluorescence imaging. *J Exp Bot*, 58(4):773-84.
- Chaerle, L., Lenk, S., Leinonen, I., Jones, H., Van Der Straeten, D., & Buschmann, C. (2009 Aug). Multi-sensor plant imaging: Towards the development of a stress-catalogue. *Biotechnol J*, 4(8):1152-67.
- Collopy, F. A., & Armstrong, J. S. (2001). Expert Systems for Forecasting. In S. J. Armstrong, *Principles of Forecasting*. Kluwer: Kluwer Academic Publishers.
- Czosnek, H. (2007). *Tomato Yellow Leaf Curl Virus Disease*. Dordrecht: Springer Netherlands.
- David, K., & Dong, P. Z. (2015). Image Classification. *Heterogeneous Computing with OpenCL 2.0*.
- David, P. H., & Marcel, S. (2016). *An open access repository of images on plant health to enable the development of mobile disease diagnostics*. Retrieved from Cornell University: <https://arxiv.org/abs/1511.08060>
- Dongyi, W., Robert, V., & Holmes, M. ( 2019). Early Detection of Tomato Spotted Wilt Virus by Hyperspectral Imaging and Outlier Removal Auxiliary Classifier Generative Adversarial Nets (OR-AC-GAN). *Scientific Reports*, 9, Article number: 4377.

- Esteban, L. G., Fernández, F. G., Palacios, P., & Conde, M. (2009). Artificial neural networks in variable process control: application in particleboard manufacture. *Investigacion Agraria-Sistemas Y Recursos Forestales*, 18(1), 92-100.
- Fang, Y., Umasankar, Y., & Ramasamy, R. (2014). Electrochemical detection of p-ethylguaiacol, a fungi infected fruit volatile using metal oxide nanoparticles. *Analyst*, 139(15):3804-10.
- Foolad, M. (2007). Genome mapping and molecular breeding of tomato. *International Journal on Plant Genom.*
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* . Beijing: Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 9547.
- Ghaiwat, N. S., & Arora, P. (2014). Detection and Classification of Plant Leaf Diseases Using Image Processing Techniques. *A Review. International Journal of Recent Advances in Engineering & Technology*, 2347 - 2812, Vol. 2, No. 3.
- GuanHuiLing, H. (2000). *Principle and Practice of Facility Diagnosis*. Beijing: China Science Publishing House.
- Hall, D., McCool, C., Dayoub, F., Sunderhauf, N., & Upcroft, B. (2015). Evaluation of features for leaf classification in challenging conditions. *In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision* (pp. 797–804). Waikoloa Beach, HI, USA: IEEE.
- Harold, C. P., Ioannis, C. K., Penelope, J. B., & Savvas, D. (2007). A Review of Recent Research on Tomato Nutrition, Breeding and Post-Harvest Technology with Reference to Fruit Quality. *The European Journal of Plant Science and Biotechnology*.
- Hillary, N. T. (2015). *Artificial Intelligent System for Diagnosis and Management of Maize Pest in Uasin Gishu County, Kenya*. Nairobi: University of Nairobi.
- Ian, G., Yoshua, B., & Aaron, C. (2016). *Deep Learning*. Massachusetts: MIT Press.

- Iqbal, H. S., Kayes, A. S., & Watters, P. (2019). Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data*, Article Number 57.
- Jasmeet, K., Raman, C., Shvani, T., & Ramanpreet, K. (2016, April). A REVIEW PAPER ON PLANT DISEASE DETECTION USING IMAGE PROCESSING AND NEURAL NETWORK APPROACH. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, pp. 758-761.
- Jayme, G. (2016). Expert Systems Applied to Plant Disease Diagnosis: Survey and Critical View. *IEEE Latin America Transactions* , 14(4):1910-1922.
- Jeyalakshmi, S., & Radha, R. (2017, May). A REVIEW ON DIAGNOSIS OF NUTRIENT DEFICIENCY SYMPTOMS IN PLANT LEAF IMAGE USING DIGITAL IMAGE PROCESSING. *ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING*, pp. 1515-1530.
- Jihen, A., & Bassem, B. A. (2017). Deep Learning Based Approach for Banana Leaf Diseases Classification. *Datenbanksysteme für Business, Technologie und Web*. Stuttgart, Germany: Datenbanken und Informationssysteme (DBIS).
- Kamei, A., Dutta, S., Sarker, K., & Das, S. (2019). Target leaf spot of tomato incited by *Corynespora cassiicola* , an emerging disease in tomato production under Gangetic alluvial region of West Bengal, India. *Archives of Phytopathology and Plant Protection*, 1-10.
- Khadabadi, G. C. (2016). Disease Detection in Vegetables Using Image Processing Techniques: A Review. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, 1353 (14) 2.
- Konstantinos, D., Wlodek, D., & Lazaros, S. I. (September, 2010). Artificial Neural Networks- ICANN 2010. *20th International Conference*. Thessaloniki, Greece: Springer.
- Kuckenber, J., Tartachnyk, I., & Noga, G. ( 2009). Use of blue-green and chlorophyll fluorescence measurements for differentiation between nitrogen deficiency and pathogen infection in winter wheat. *Precis. Agric.*, 10:34–44.

- Lauchli, A., & Bielecki, R. (2012). *Inorganic Plant Nutrition*. New York: Springer-Verlag.
- Mabvakure, B., Martin, D., Kraberger, S., Cloete, L., Van Bruschoot, S., Geering, A., & Thomas, J. (2016). Ongoing geographical spread of Tomato yellow leaf curl virus. *Virology*, 498, 257–264.
- Mahlein, A., Oerke, E., Steiner, U., & Dehne, H. (2012). Recent advances in sensing plant diseases for precision crop protection. *European Journal Plant Pathology*, 133:197–209.
- Maina, N. C. (2016). *Vision-based model for maize leaf disease identification: a case study in Nyeri County*. Nairobi: Strathmore University Library.
- Manjula, N. (2016). Diagnosing Nutrient Deficiencies. *Integrated Pest Management*, University of Missouri.
- Manoj, K., Neelima, M., & Harshali, M. (2018). Image Classification using Deep Learning. *International Journal of Engineering and Technology*, 7(2.7), 614-617.
- Marko, A., Mirjana, K., Srdjan, S., & Andras, A. (2019). Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection. *Symmetry*.
- Monica, E., Ken, P., & Aaron, P. (2014). *Guidelines to Identification and Management of Plant Disease Problems: Part I. Eliminating Insect Damage and Abiotic Disorders*. Florida: University of Florida.
- Muhammad, A., Nur, D. K., Amar, F. Z., & Siti, A. N. (2017, MAY ). A Fuzzy Inference System For Diagnosing Oil Palm Nutritional Deficiency Symptoms. *ARPJ Journal of Engineering and Applied Sciences*, pp. VOL(12), NO.10, 3244-3257.
- Muhammad, H. S., Johan, P., & Khalid, M. A. (2019). Plant Disease Detection and Classification by Deep Learning. *Plants MDPI*.
- Nelson, S. C. (2008). *Late Blight of Tomato (Phytophthora infestans)*. Hawaii: Department of Plant and Environmental Protection Sciences, University of Hawaii at Manoa.
- Nils, J. N. (2016). *Principles of Artificial Intelligence*. Berlin: Springer-Verlag Berlin Heidelberg.

- Organisation, F. a. (2005). *Modernizing national agricultural extension systems: a practical guide for policy-makers of developing countries*. Netherland: FAO printing Press.
- Owuor, B. W., Ogada, J. O., Poulton, C., & Argwings-Kodhek, G. (2010). The Role and Performance of the Ministry of Agriculture in Eldoret West District. *Future Agricultures Consortium*. Sussex.
- Peyman, M., Daniel, W., Ethan, G., Srimal, J., Pavan, S., & Emili, H. (2017). Plant Disease Detection Using Hyperspectral Imaging. *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. Sydney, NSW, Australia: IEEE.
- Potnis, N., Timilsina, S., & Strayer, A. (2015). Bacterial spot of tomato and pepper: diverse *Xanthomonas* species with a wide variety of virulence factors posing a worldwide challenge. *Molecular Plant Pathology*, Vol. 16(9), 907-920.
- Pragya, A., Yeonyee, O., & Dilip, R. P. (2017). Current Status of Early Blight Resistance in Tomato: An Update. *International Journal of Molecular Sciences*.
- Rikiya, Y., Mizuho, N., Do, R. K., & Kaori, T. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, pages 611–629.
- Rothe, R. P., & Kshirsagar, V. R. (2014). SVM-based Classifier System for Recognition of Cotton Leaf Diseases. *International Journal of Emerging Technologies in Computational and Applied Sciences*.
- Rumpf, T. e. (2010). Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Computers and Electronics in Agriculture* , 74, 91–99 .
- Sachin, B. J., Sanjay, B. P., & Vishwanath, R. U. (2020). Identification of plant diseases using convolutional neural networks. *International Journal of Information Technology*.
- Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, 72, 1–13.

- Shruthi, U., & Nagaveni, V. (2019). A Review on Machine Learning Classification Techniques for Plant Disease Detection. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. Coimbatore, India: IEEE.
- Sindhu, B., & Bashirahamad, F. M. (2018). Detection of Diseases on Crops & Design of Recommendation Engine: A Review. *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 390-398). Springer.
- Srinivasan, R. (2010). .Safer tomato production methods: A field guide for soil fertility and pest management. *AVRDC-The World Vegetable Center* (pp. No.10-740. 97). Shanhua, Taiwan: VRDC Publication.
- Suguna, N., & Thanushkodi, K. (2010). An Improved k-Nearest Neighbor Classification Using Genetic Algorithm. *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 4.
- Swati, G., & Ritika, S. (2013). Fundamentals and Characteristics of an Expert System. *International Journal on Recent and Innovation Trends in Computing and Communication*, 1(3), 110-113.
- Thind, B. S. (2019). *Phytopathogenic Bacteria and Plant Diseases*. Punjab , India: CRC Press.
- Trung-Tin, T., & Jae-Won, C. T.-T. (2019). A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. *Applied Sciences*.
- Van der Wolf, J., van Bechhoven, J., Bonants, P., & Schoen, C. (2001). New technologies for sensitive and specific routine detection of plant pathogenic bacteria. In *Plant Pathogenic Bacteria* (pp. 75–77). Berlin, Germany: Springer.
- Vijai, S. (2017, March). Information Processing in Agriculture. *Detection of plant leaf diseases using image segmentation and soft computing techniques*, pp. Vol 4, Issue 1, Pages 41-49.
- Wang, L. (2005). *Support Vector Machines: Theory and Applications* . Singapore: Springer.

- Wang, X., Zhang, M. Z., & Geng, S. (2008). Spectral prediction of *Phytophthora infestans* infection on tomatoes using artificial neural network (ANN). *International Journal of Remote Sensing*, 1693-1706.
- Ward, E., Foster, S., Fraaije, B., & McCartney, H. (2004). Plant pathogen diagnostics: Immunological and nucleic acid-based approaches. *Ann. Appl. Biol.*, 145:1–16.
- Yi, F., & Ramaraja, P. R. (2015). Current and Prospective Methods for Plant Disease Detection. *Biosensors*, 5(3): 537–561.
- Yuanyuan, S., Cheng, T., Shan, H., Ke, W., & Lisu, C. (2018 ). Identification of Nitrogen, Phosphorus and Potassium Deficiencies Based on Temporal Dynamics of Leaf Morphology and Color. *Sustainability*, Vol. (10), 762 .
- Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can Deep Learning Identify Tomato Leaf Disease. *Adv. Multimed*, 10.
- Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. *Computers and electronics in agriculture* , 36, 113–132 .

## Appendices

### Appendix A: CNN Model

```
model=models.Sequential()  
model.add(Conv2D(32, (3, 3), padding="same",input_shape=(256,256,3)))  
model.add(Activation("relu"))  
model.add(BatchNormalization(axis=chanDim))  
model.add(MaxPooling2D(pool_size=(3, 3)))  
model.add(Dropout(0.25))  
model.add(Conv2D(64, (3, 3), padding="same"))  
model.add(Activation("relu"))  
model.add(BatchNormalization(axis=chanDim))  
model.add(Conv2D(64, (3, 3), padding="same"))  
model.add(Activation("relu"))  
model.add(BatchNormalization(axis=chanDim))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
model.add(Conv2D(128, (3, 3), padding="same"))  
model.add(Activation("relu"))  
model.add(BatchNormalization(axis=chanDim))  
model.add(Conv2D(128, (3, 3), padding="same"))  
model.add(Activation("relu"))  
model.add(BatchNormalization(axis=chanDim))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
model.add(Flatten())  
model.add(Dense(1024))  
model.add(Activation("relu"))  
model.add(BatchNormalization())  
model.add(Dropout(0.5))  
model.add(Dense(n_classes))  
model.add(Activation("softmax"))
```

## Appendix B: Model Summary

Model: "sequential\_4"

| Layer (type)                                | Output Shape         | Param #  |
|---|----------------------|----------|
| conv2d_4 (Conv2D)                           | (None, 256, 256, 32) | 896      |
| activation_2 (Activation)                   | (None, 256, 256, 32) | 0        |
| batch_normalization_1 (Batch Normalization) | (None, 256, 256, 32) | 128      |
| max_pooling2d_1 (MaxPooling2D)              | (None, 85, 85, 32)   | 0        |
| dropout_3 (Dropout)                         | (None, 85, 85, 32)   | 0        |
| conv2d_5 (Conv2D)                           | (None, 85, 85, 64)   | 18496    |
| activation_3 (Activation)                   | (None, 85, 85, 64)   | 0        |
| batch_normalization_2 (Batch Normalization) | (None, 85, 85, 64)   | 256      |
| conv2d_6 (Conv2D)                           | (None, 85, 85, 64)   | 36928    |
| activation_4 (Activation)                   | (None, 85, 85, 64)   | 0        |
| batch_normalization_3 (Batch Normalization) | (None, 85, 85, 64)   | 256      |
| max_pooling2d_2 (MaxPooling2D)              | (None, 42, 42, 64)   | 0        |
| dropout_4 (Dropout)                         | (None, 42, 42, 64)   | 0        |
| conv2d_7 (Conv2D)                           | (None, 42, 42, 128)  | 73856    |
| activation_5 (Activation)                   | (None, 42, 42, 128)  | 0        |
| batch_normalization_4 (Batch Normalization) | (None, 42, 42, 128)  | 512      |
| conv2d_8 (Conv2D)                           | (None, 42, 42, 128)  | 147584   |
| activation_6 (Activation)                   | (None, 42, 42, 128)  | 0        |
| batch_normalization_5 (Batch Normalization) | (None, 42, 42, 128)  | 512      |
| max_pooling2d_3 (MaxPooling2D)              | (None, 21, 21, 128)  | 0        |
| dropout_5 (Dropout)                         | (None, 21, 21, 128)  | 0        |
| flatten_3 (Flatten)                         | (None, 56448)        | 0        |
| dense_4 (Dense)                             | (None, 1024)         | 57803776 |
| activation_7 (Activation)                   | (None, 1024)         | 0        |

|   |              |       |
|---|--------------|-------|
| batch_normalization_6 (Batch Normalization) | (None, 1024) | 4096  |
| dropout_6 (Dropout)                         | (None, 1024) | 0     |
| dense_5 (Dense)                             | (None, 10)   | 10250 |
| activation_8 (Activation)                   | (None, 10)   | 0     |
| =====                                       |              |       |
| Total params: 58,097,546                    |              |       |
| Trainable params: 58,094,666                |              |       |
| Non-trainable params: 2,880                 |              |       |

## Appendix C: Training Progress

Train on 8000 samples, validate on 2000 samples

```
Epoch 1/30
8000/8000 [=====] - 30s 4ms/step - loss: 0.4643 -
acc: 0.8737 - val_loss: 0.3572 - val_acc: 0.8967
Epoch 2/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.4232 -
acc: 0.8827 - val_loss: 0.3813 - val_acc: 0.8937
Epoch 3/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.4086 -
acc: 0.8860 - val_loss: 0.3798 - val_acc: 0.8927
Epoch 4/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3977 -
acc: 0.8883 - val_loss: 0.3679 - val_acc: 0.8927
Epoch 5/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3933 -
acc: 0.8893 - val_loss: 0.3575 - val_acc: 0.8950
Epoch 6/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3911 -
acc: 0.8901 - val_loss: 0.3549 - val_acc: 0.8952
Epoch 7/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3879 -
acc: 0.8902 - val_loss: 0.3487 - val_acc: 0.8960
Epoch 8/30
8000/8000 [=====] - 28s 3ms/step - loss: 0.3865 -
acc: 0.8908 - val_loss: 0.3499 - val_acc: 0.8937
Epoch 9/30
8000/8000 [=====] - 28s 3ms/step - loss: 0.3844 -
acc: 0.8911 - val_loss: 0.3919 - val_acc: 0.8903
Epoch 10/30
8000/8000 [=====] - 28s 3ms/step - loss: 0.3846 -
acc: 0.8915 - val_loss: 0.3480 - val_acc: 0.8932
Epoch 11/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3808 -
acc: 0.8909 - val_loss: 0.3326 - val_acc: 0.8982
Epoch 12/30
8000/8000 [=====] - 28s 3ms/step - loss: 0.3784 -
acc: 0.8916 - val_loss: 0.4166 - val_acc: 0.8861
Epoch 13/30
8000/8000 [=====] - 28s 3ms/step - loss: 0.3801 -
acc: 0.8920 - val_loss: 0.3802 - val_acc: 0.8899
Epoch 14/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3784 -
acc: 0.8926 - val_loss: 0.3307 - val_acc: 0.8990
Epoch 15/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3745 -
acc: 0.8930 - val_loss: 0.3617 - val_acc: 0.8902
Epoch 16/30
8000/8000 [=====] - 28s 4ms/step - loss: 0.3799 -
acc: 0.8924 - val_loss: 0.3751 - val_acc: 0.8918
Epoch 17/30
```

8000/8000 [=====] - 28s 4ms/step - loss: 0.3735 -  
acc: 0.8932 - val\_loss: 0.3911 - val\_acc: 0.8890  
Epoch 18/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3712 -  
acc: 0.8932 - val\_loss: 0.3513 - val\_acc: 0.8947  
Epoch 19/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3712 -  
acc: 0.8937 - val\_loss: 0.3838 - val\_acc: 0.8895  
Epoch 20/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3716 -  
acc: 0.8936 - val\_loss: 0.3282 - val\_acc: 0.8994  
Epoch 21/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3683 -  
acc: 0.8938 - val\_loss: 0.3330 - val\_acc: 0.8977  
Epoch 22/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3709 -  
acc: 0.8940 - val\_loss: 0.3442 - val\_acc: 0.8948  
Epoch 23/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3673 -  
acc: 0.8942 - val\_loss: 0.4492 - val\_acc: 0.8833  
Epoch 24/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3717 -  
acc: 0.8939 - val\_loss: 0.3498 - val\_acc: 0.8952  
Epoch 25/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3674 -  
acc: 0.8942 - val\_loss: 0.3378 - val\_acc: 0.8972  
Epoch 26/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3677 -  
acc: 0.8936 - val\_loss: 0.4046 - val\_acc: 0.8866  
Epoch 27/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3641 -  
acc: 0.8947 - val\_loss: 0.4428 - val\_acc: 0.8836  
Epoch 28/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3644 -  
acc: 0.8945 - val\_loss: 0.3661 - val\_acc: 0.8934  
Epoch 29/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3654 -  
acc: 0.8941 - val\_loss: 0.3782 - val\_acc: 0.8921  
Epoch 30/30  
8000/8000 [=====] - 28s 4ms/step - loss: 0.3664 -  
acc: 0.8943 - val\_loss: 0.3488 - val\_acc: 0.8966

## Appendix D: Classification Report

| Classification Report |           |        |          |         |
|-----------------------|-----------|--------|----------|---------|
|                       | precision | recall | f1-score | support |
| Bacterial_spot        | 0.70      | 0.93   | 0.80     | 15      |
| EarlyBlight           | 0.60      | 0.75   | 0.67     | 4       |
| Health                | 1.00      | 0.86   | 0.92     | 7       |
| LateBlight            | 0.64      | 0.73   | 0.68     | 22      |
| LeafMold              | 1.00      | 0.71   | 0.83     | 7       |
| MosaicVirus           | 1.00      | 1.00   | 1.00     | 1       |
| Nitrogen              | 1.00      | 0.17   | 0.29     | 6       |
| SeptoriaLeaf          | 0.69      | 0.56   | 0.62     | 16      |
| TargetSpot            | 0.50      | 0.50   | 0.50     | 6       |
| YellowLeaf            | 0.83      | 0.94   | 0.88     | 16      |
| accuracy              |           |        | 0.73     | 100     |
| macro avg             | 0.80      | 0.71   | 0.72     | 100     |
| weighted avg          | 0.75      | 0.73   | 0.72     | 100     |

# Appendix E: Turnitin Report

feedback studio Raymond Paul Kiyegga A COMPUTER VISION BASED MODEL FOR TOMATO PLANT NUTRIENT AND DISEASE CLASSIFICATION

**A COMPUTER VISION BASED MODEL FOR TOMATO PLANT NUTRIENT AND DISEASE CLASSIFICATION**

**Match Overview**

**18%**

| Rank | Source                                      | Percentage |
|------|---|------------|
| 1    | mc.ai<br>Internet Source                    | 2%         |
| 2    | Submitted to University...<br>Student Paper | 2%         |
| 3    | Submitted to Strallemor...<br>Student Paper | 1%         |
| 4    | www.mdpi.com<br>Internet Source             | 1%         |
| 5    | pdfs.semanticscholar...<br>Internet Source  | 1%         |
| 6    | Submitted to University...<br>Student Paper | <1%        |
| 7    | eprints.utm.my<br>Internet Source           | <1%        |
| 8    | "Tomato Leaf Disease ..."<br>Publication    | <1%        |

## Appendix F: Ethical Review Approval



9<sup>th</sup> January 2020

Mr Kiyegga, Raymond,  
rkiyegga@strathmore.edu

Dear Mr Kiyegga,

**RE: A Vision Based Model for Tomato Plant Nutrient and Disease Classification**


This is to inform you that SU-IERC has reviewed and **approved** your above research proposal. Your application approval number is **SU-IERC0588/19**. The approval period is **9<sup>th</sup> January, 2020 to 8<sup>th</sup> January, 2021**.

This approval is subject to compliance with the following requirements:

- i. Only approved documents including (informed consents, study instruments, MTA) will be used
- ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
- iii. Death and life threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 72 hours of notification
- iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 72 hours
- v. Clearance for export of biological specimens must be obtained from relevant institutions.
- vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
- vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) <https://oris.nacosti.go.ke> and also obtain other clearances needed.

Yours sincerely,

*for*  
  
Dr Virginia Gichuru,  
Secretary; SU-IERC

Cc: Prof Fred Were,  
Chairperson; SU-IERC

