

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEM

PROJECT ON ENSET BACTERIAL WILT DISEASE IDENTIFICATION AND SOLUTION SUGGESTION WITH COMPUTER VISION

BY

NO	NAME OF STUDENTS	ID NO
1	MICHAEL GETACHEW	CIR/ 133 /10
2	MUHAMMED NURU	CIR /135 /10
3	MUHAMMED YASIN	CIR/ 234 /09
4	SAMRAWIT LEMMA	CIR/ 143/10

ADVISOR: Mr. Dawit Hassen

Wolkite University, Wolkite, Ethiopia

May 25/05/2021

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEM

ENSET BACTERIAL WILT DISEASE IDENTIFICATION AND SOLUTION SUGGESTION WITH COMPUTER VISION

SUBMITED TO DEPARTMENT OF INFORMATION SYSTEM

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHLER OF SCIENCE IN INFORMATION SYSTEM

BY

NO	NAME OF STUDENTS	ID NO
1	MICHAEL GETACHEW	CIR/ 133 /10
2	MUHAMMED NURU	CIR /135 /10
3	MUHAMMED YASIN	CIR/ 234 /09
4	SAMRAWIT LEMMA	CIR/ 143/10

ADVISOR: Mr. Dawit Hassen

Wolkite University, Wolkite, Ethiopia

May 25 /05/2021

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Dawit Hassen and having the title enset bacterial wilt disease identification and solution suggestion with computer vision is the sole contribution of: Michael Getachew, Muhammed Nuru, Muhammed Yassin And Samrawit Lemma. No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Members:

NO	**NAME OF STUDENTS**	**Signature**
1	Michael Getachew	
2	Muhammed Nuru	
3	Muhammed Yassin	
4	Samrawit Lemma	

Approval Form

This is to confirm that the project report entitled **Enset Bacterial wilt disease identification and solution suggestion with computer vision** submitted to **Wolkite University, College of Computing and informatics Department of information system**.

---	---	---
Advisor Name	Signature	Date
---	---	---
Department Head Name	Signature	Date
---	---	---
Examiner 1 Name	Signature	Date
---	---	---
Examiner 2 Name	Signature	Date
---	---	---
Examiner 3 Name	Signature	Date

ACKNOWLEDGMENT

First of all, we would like to thank our almighty God, who gives us love, patience, healthy, wisdom and ability to walk through all the problems and obstacles during the period of our study. Then we would like to express our special thanks of gratitude to our advisor Mr. Dawit Hassen for his constructive idea and willingness to participate in each part of the project and his effective direction, assistance, and guidance for the accomplishing of this project. As well as we would like to thank Mr. Ashenafi who gave us golden advice and full information about the Enset and guid to get Enset farm area of Wolkite and also thanks to Ato Dinnu who work on wokite Enset farm who helps us to get specially infected Enset for our project. Secondly, we would like thank MR. Yidnekachew Kibru who helps us to do an efficient work in many ways. Finally, we would like to thank all the worker in the area of Enset who helped us when we gather information about the Enset disease.

Table of Contents

DECLARATION II
Approval Form III
ACKNOWLEDGMENT IV
LIST OF ABBREVIATIONS IX
ABSTRACT X
CHAPTER ONE 1
1. INTRODUCTION 1
1.1 Statement of the problem 2
1.2 Objective of the project 3
1.3 Feasibility Analysis 3
1.4 Scope and limitation 4
1.5 The significance of the Project 5
1.6 The Beneficiary of the Project Farmer 5
1.7 Methodology for the Project 6
CHAPTER TWO 12
2. LITERATURE REVIEW 12
2.1 INTRODUCTION 12
2.2 History of Enset and Its Bacterial Wilt Disease 12
2.3 Artificial Intelligence 13
2.5 Convolutional Neural Network (CNN) 15
2.7 Conclusion 16
CHAPTER THREE 17
3. PROPOSED SYSTEM 17
3.1. Functional Requirements 17
3.2. Non-functional Requirements 17
CHAPTER FOUR 23
4. SYSTEM ANALYSIS 23
4.1 System Model 23
4.2. Object Model 34
4.3. Dynamic Model 37
4.4 Model development diagram 44
CHAPTER FIVE 45

5. SYSTEM DESIGN 45
5.1. Design Goals 45
5.3. System Architecture 46
5.4 Packages 52
5.5. Algorithm Design 53
5.5. User Interface Design 55
CHAPTER SIX 56
6. IMPLEMENTATION AND TESTING 56
6.1. Implementation of the Database 56
6.2. Implementation of the Class Diagram 56
6.3. Twilio SMS Server 56
6.4. Configuration of Application Security 58
6.5. Implementation of User Interface 58
6.6. Testing 60
6.7. Model Development 61
6.7. Activation function 69
6.8. Loss function 69
6.9. Optimization Algorithm 70
6.10. Learning Rate 70
6.11. Summary () function 71
6.12. Epoch 73
6.13. Configure the dataset for performance 73
6.14. Discussion of the Result 73
CHAPTER SEVEN 74
7. CONCLUSION AND RECOMMENDATION 74
7.1. CONCLUSION 74
7.2 RECOMMENDATION 75
References 76
APPENDICES 77

LIST OF TABLES

Table 1-1 software tools 11

Table 1-2 hardware tools..... 11

Table 4-1 Actor and use case information 23

Table 4-2 Detect use case description 26

Table 4-3 Feedback use case description..... 28

Table 4-4 Add statistical data use case description ..... .28

Table 4-5 View statistical data use case description 29

Table 4-6 Login use case description..... 30

Table 4-7 Logout use case description..... 31

Table 4-8 View feedback use case description..... 32

Table 4-1 Data dictionary for feedback 35

Table 4-2 Data dictionary for statistical data ..... .36

LIST OF FIGURES

Figure 2-1 Healthy enset and Infected 13
Figure 2-2 Deep learning [12]..... 15
Figure 4-3 Use case diagram for enset disease identification 25
Figure 4-4 Class Diagram 34
Figure 4-5 Sequence diagram for detect disease..... 37
Figure 4-6 sequence diagram for view data 38
Figure 4-7 sequence diagram for feedback..... 39
Figure 4-8 sequence diagram for Login 40
Figure 4-9 Activity diagram for detection 41
Figure 4-10 Activity Diagram for Login..... 42
Figure 4-11 state diagram for detection disease 43
Figure 4-12 state diagram for login 44
Figure 4-13 machine learning model development for EDBW 44
Figure 5-1 System architecture……..... 46
Figure 5-2 System decomposition……..... 48
Figure 5-3 Deployment Diagram…….....49
Figure 5-4 Detail class Diagram……..... 50
Figure 5-5 Persistent Diagram……..... 51
Figure 5-6 Package Diagram……..... 52
Figure 5-7 User Interface…….…….…..... 55
Figure 6.1 Twilio Server Console……. 57
Figure 6-2 Twilio Server SMS log Console…….….58
Figure 6-3 Dataset Visualization from Training Set……62
Figure 6-4 normalization of data set…….….64
Figure 6-5 graph visualization accuracy and loss of the model……...…67
Figure 6-6 user interface uploading data……...…68
Figure 6-7 web app respond to the user infected leaf……...…68
Figure 6-8 web app respond to the user healthy leaf.....…...69

LIST OF ABBREVIATIONS

No	Abbreviation	Description
1	ML	Machine Learning
2	CNN	Convolution Neural Network
3	DCNN	Deep Convolutional Neural Network
4	DNN	Deep Neural Network
5	OOSAD	Object Oriented System Analysis and design
6	AI	Artificial Intelligence
7	UI	User Interface
8	CPU	Central Processing Unit
9	Convnet	Convolutional Net
10	EDIASS	Enset Disease Identification and solution suggestion
11	EBWD	Enset BACTERIAL WILT Disease
12	SGD	Stochastic gradient decent

ABSTRACT

Enset is the one of the most common crop plants in south Ethiopia, this crop plant has a variety of uses for the community for example it has used for medicinal purpose for animals as well as for humans that for broken bones and other health problems but mostly is used as main source of food for the human and their animals. Enset crop plant has several infections disease among them bacterial wilt is the one most dangerous disease of enset crop plant. A bacterial wilt is contagiousness, it can be transmitted with using same sickle to cut enset leaf of the enset. And also, it leads to total farm damage if it's not controlled by the time it noticed on the farm. Nowadays, by applying Artificial Intelligence technique identify different type of disease and computer vision is one of technique. For this project we used Deep Convolutional Neural Network algorithm for the model development. Bacterial wilt disease identification model developed and used in the web application in order to give meaningful response for the users and experts. In this project we apply hybrid methodology. We follow the building basic machine learning workflow from data gathering to optimization of the model step by step then after we use the model on the web application the web application has several functions to give the user information about the status of enset bacterial wilt disease. We develop the web application which can collect the detection status log and give the follow up notification and make statistical data with time interval as well.

CHAPTER ONE

1. INTRODUCTION

Around 80-85% of Ethiopians are engaged in Ethiopia [1]. Enset crop is related to and resembles the banana plant which is an indigenous plant classified under the monocarpic genus Enset and family Musaceae. This is commonly known as false banana, the Ethiopian banana or the Abyssinian banana. Locally the plant is called Enset and botanically, it is named as Enset ventricosum [2]. It has the ability to withstand drought conditions and use as a substantial food item for many millions of people in Ethiopia. South and southwestern parts of Ethiopia are among the potential producers of enset. It is a crop upon which over 20 million people in the southern part of Ethiopia rely on as a staple and co- staple source of food. Enset plant has also other many benefits, as a fiber, animal forage, and medicine for cultural practices. [3]

Ethiopia is one of the grand producers of Enset in African continent countries. There are a number of issues and diseases that try to reduce yield and efficiency. As a matter of facts, visual or manual detections may have defects in terms of accuracy in detection along with lower precision. There is a potential need for technology supported alternative systems to support the manual identification of Enset crop diseases so as to optimize the accuracy for remedial action. If the Enset crop diseases can be minimized then definitely the production will significantly affect the production of enset in positive ways.

To solve this problem, we are planning to develop a web application system. The methodology our team will use in this project is an object-oriented system development methodology (OOSAD) for the design and Deep convolutional neural network for model development. This technique has several phases. By collecting image from the farm to developing model that can identify the infected leaf or not and also the team performs modeling of the function of the system (use case modeling), find and identify the business objects, organize the objects and identify the relationship between them and finally model the behavior of the objects in detail. As well as get suggestion to control the infection contagiousness

1.1 Statement of the problem

Enset is used in south and southwestern areas as the main source of nutrition because this plant disease causes a lot of damage to the community of these areas. These diseases are caused by bacteria or fungi and transmitted by many career on to another enset plants from bacterial diseases the most common known and dangerous type of disease is bacterial wilt this disease cause many damages on enset plant and make it wilt .It transmitted with many insect and by any contact from infected one to the healthy one by this means the disease problem become harsh There are many problems in preventing, managing ,controlling the disease of enset because of several reasons.

Since the majority of Ethiopian farmers are uneducated and do not obtain accurate and complete information about Enset crop diseases, They didn't have awareness of the disease type whether its bacterial , fungal , viral or other insect problem (Detection problems) because of this they need expert advice. Furthermore, in regions where the Enset crop is grown, there is a scarcity of resources and expertise in the field of Enset pathology. Furthermore, crop pathologists are unable to visit every farm, and the crop pathologists are limited even if the farmers aware the enset is affected by some kind of disease they do not perform any recommended method of extraction the infected one they use their own traditional methods In addition to this the Prevention method of this are labor intensive due to the young man power migration to urban areas there is lack of human power and this lead to poor scouting of the enset crop farm. Expertise didn’t record the farm state time to time because of lack of follow up they could not predict what will happen to the farm in future.

The computer vision using deep leaning technology now a day’s become the key technique for the diagnosis of various features of the plant in the areas of agriculture because it minimize confusion and helping the expert in the sectors providing new approach to explore the field of agriculture. In recent years, deep learning models that exploit multiple layers of nonlinear information processing, for feature extraction and transformation as well as for pattern analysis and classification, have been shown to overcome these challenges. Among them, CNNs [4] have become the leading architecture for most image recognition, classification, and detection tasks. [5] Therefore, the implementation of Deep Convolutional Neural Networks in the sector have a

paramount importance to facilitate the recognition of plant diseases activities by increasing efficiency and productivity.

To this end, in this project we used Deep Convolutional Neural Network (DCNN) for the model development and Django for the model deployment. By collecting image from the farm to developing model that can identify the infected leaf and use this model to the web application in order to interact with the user using the web application user can get services like the enset plant status whether it's infected or not as well as get suggestion to control the infection contagiousness.

1.2 Objective of the project

1.2.1 General Objective

The main objective of this project is to develop Enset (ventricosum) bacterial wilt disease identification and solution suggestion system.

1.2.2 Specific Objectives

- To Detect the enset plant infected by the disease or healthy
- To suggest treatment and infection removal method
- To develop deep learning model
- To test and tune the model
- To develop web application
- To alert the farmer by notifications

1.3 Feasibility Analysis

We examined feasibility of our study from different perspectives like, from its technical, operational and economical points of view.

1.3.1 Technical Feasibility

The system is easy to maintain and is very much adaptable for change. Our development team has the required knowledge to develop the functionalities of the system. So, the system will be technically feasible

1.3.2 Economic feasibility

The development of the system requires a minimum amount of cost. The estimated cost of resources that we use to develop this project is not more expensive than the functionality of the system and the advantage to the user.

1.3.3 Operational Feasibility

The system performs all of its tasks in a user-friendly manner. In which, anyone without an advanced technical background can use the system. The user interface will simple and interactive other functionality of the system will work smoothly

1.4 Scope and limitation

1.4.1 Scope of the project

Enset (ventricosum) disease identification and solution suggestion with AI meets the following system and information requirements.

- Detect the disease
- Identify disease type
- Display the disease type
- Suggest the solution
- Get feedbacks

1.4.2 Limitation of the project

- System considers for bacterial wilt disease it is not consider other type of disease.

- Local Language support problem due to the variety of Gurage language.

- The system targets web application

1.5 The significance of the Project

By technological advancement we are rich at the moment we can develop useful applications that make human life easy specially the artificial intelligence advancement have a greater role in the day to day life of human beings. Using different machine algorithms and techniques we can develop applications that have a significant role in identification of bacterial wilt diseases that helps farmers and agricultural experts in several ways. This project will have the following importance: -

- Experts of agriculture will have the knowledge of technology and its advantage
- The farmers could identify the bacterial wilt diseases simply by using this application
- this project will help the farmers to save resources and take a good care of their farm
- the project will help agricultural expert to remotely check the bacterial wilt disease spreading and get better information from the application for measurement and statistical analysis

1.6 The Beneficiary of the Project Farmer

The farmer is the most beneficiary from this web app because the system simply detects and identifies diseases and gets appropriate suggestions from this application.

Agricultural Expertise and Researchers

Agricultural expertise and researchers will also beneficiary from this system because this system helps to identify and understand enset disease in simple and easy way

Community

The other beneficiary of this application is the community if the disease out of control on one farmer farm and the disease is contagious so it will spread to the nearby farms by using the application the farmer will easily detect and take action on the infected enset so the disease will not spread to the other farms therefore the community will have sustainable source of food since enset is their main source of food. The other beneficiary from this project is the Woreda /zone agricultural bureau.

1.7 Methodology for the Project

In order to accomplish this project on time and within the cost, we would follow different procedures which are described below.

1.7.1 Data Collection Tools/Techniques

Data collection is the most important part of this project to find the main requirement of the system the data will collect in the form of images which is dataset and other descriptions in the form of text. We are going to use different methods to collect data.

Since we are developing CNN model that identify the bacterial wilt disease that affects enset crop plant, and we have to collect the disease phenotypic images from infected enset leaf and healthy leaf images from farms that the bacterial wilt disease affect and for healthy leafs we gather the data set from conserved research area by Wolkite university around EMDBER by help of the agricultural expert whose name is MR. Ashenafi and , Dinu supervisor of Wolkite conserved farm to identify the real bacterial wilt disease from the farmers farm the images are taken with high resolution camera .

Among the methods, we use the following: -

Primary data gathering

Knowledge acquisition held in several ways for the data that we want to train the model which is a lot of images we take our data by camera that have 16 mega pixel and cannon 5D mark III which is the 2nd best camera of cannon at this time All the data sets of Enset leaf images are

collected from the village farm that has infected Enset plant from this village YEFEREZIYE (የፈረዝየ) and YERBATYE (የርባትየ) has population of bacterial wilt disease and as we mentioned in the introduction part the healthy leafs are collected from the Wolkite university conserved farm and research area which has quite healthy Enset plants. For the model that we develop to be strong there had to be a lot of images.

Interviews:

To get the basic information about the problem that faced the team members will interview the farmers, experts and the researchers that work on this area and the problems associated with that environment.

Observation:

Even though the interview is very important to gather information, direct observation is simple and we will physically observe information that cannot be maintained from the interview or other sources and also it is important if they are unable to communicate with others because of the difficulties they have to the professional terms (language), And we will take the datasets by capturing images.

Document analysis

We will analyze different documents that deal with the detailed description of enset disease and various researches made on this area. We went through documents that deals with bacterial wilt disease and its characteristics and its controlling management and also prevention methods we look the bacterial wilt disease detection by MR. Yidnekachew and another research kind quite good papers that help us to develop bacterial wilt detection model on enset crop using a deep learning approach by yidnekachew kibru afework.

1.7.2 System Analysis and Design

In this project, our team will use object-oriented system development methodology (OOSAD) for the design. This technique has several phases some of them are:

- During this phase, the team models the function of the system (use case modeling), organizes the objects and identifies the relationship between them and finally models the behavior of the objects in detail.
- During this phase, our team uses Edraw Max software to refine the use case model and those for designing the class diagram, sequence, collaboration, activity diagrams, and state diagram and to model object interactions and behavior that support the use case scenario.

1.7.3 System Development Model

Since we use machine learning to identify disease of the enset and we will use web application to represent the solution we will use the hybrid type of system development method that includes machine learning pipeline and system development approach object-oriented system analysis and design method.

1.7.4 System Testing Methodology

For the model

Machine learning model need to tested and evaluated. Because the performance of the machine learning model should be satisfactory.

ML unit tests

By writing tests code and check the model development has bug on this test the hidden layers in neural network can be tested if they are configuring correctly or not

Adjusting hyper parameters

Adjusting hyper parameters helps to optimize the model this could be held by adjusting the learning rate of the model or Regularization Usually the learning rate of the model are pre valued or already decide by the library this the pre-set library learning rate is not always best for developing our model for this reason we have to adjust the learning rate of the model manually.

Regularization can be used only after model can. L1 regularization is used for reduce the model's size and L2 regularization is used to increase the model stability and drop out regularization is held in case of neural networks

For web application

Unit testing

Using unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness. In this level of testing process.

Sample Tests: -

- Check whether the return type of functions is correct.
- Check how the sub procedures or functions are called correctly.
- Check if the correct output is produced for different inputs.
- Check the efficiency of the code with respect to the memory and CPU time.

System testing

In this level of testing process, we have examined how the whole subsystems came together to achieve the desired goal (user's requirements of the system). The goals of system testing are to detect disease that can only be exposed by testing the entire integrated system or some major part of it. Generally, this testing is mainly concerned with areas such as performance, validation and load/stress. But we will focus more only on function validation and performance. Sample Tests:-

• Check the overall functionality of detection and identification (Computer vision) that achieves the user's requirement.

• Verify the system completeness-based user's requirement.

User Acceptance Testing

It is the type of testing in which users of software or applications are real users. This application is tested in a real environment and considered as a form of an external User Acceptance testing. Beta version of software released to a limited number of users when tested in a Real-Time environment with the help of real users, to obtain feedback on product quality. Beta testing reduces failures, risks and provides increased quality of a product through customer validation. Therefore, we are planning to perform this activity for selected farmers and Agricultural expertise in order to obtain feedback.

1.7.5 Development Tools and Technologies

For web Development

1.7.5.1 Frontend Technologies

- Django

1.7.5.2 Backend Technologies

We will use Django in developing and managing at the back-end.

For Model Development

- Tensor flow
- Pandas
- Scikit Learn
- Keras
- Numpy
- Theano

Numpy provides fundamental building blocks like as vectors, matrices, and operations on them, whereas Panda excels at importing data, particularly from databases, and Matplotlib aids in visualizing that data using common plots such as line charts, scatter plots, and histograms. With Theano and TensorFlow, you can define and train neural network models in just a few lines of code, which is incredible.

1.7.5.3 Documentation and Modeling Tools Software tools:

Software Tools	Description
Python	For training the model
Atom	For editing

Table 1-1 software tools

Hardware tools: describe under the following table

Hardware Tools	Description
Computer (desktop or laptop)	For documentation and implementation
flash disk	For file storage

Table 1-2 hardware tools

1.7.5.4 Deployment Environment

For testing purposes, we want to keep the deployment under localhost. Since its web application for the users it will be hosted in a cloud server.

CHAPTER TWO
2. LITERATURE REVIEW

2.1 INTRODUCTION

The bacterial wilt disease infection identification in the current time is completely manual and the farmers use traditional methods to control and prevent the disease. The experts check the farm and if any plant is infected by bacterial wilt disease, they will guide the farmer about the controlling and management mechanism of the disease. Unless otherwise the farmers take their own action to the infected enset. This method could be completely dangerous because bacterial wilt disease is contagious by its nature. If the farmers controlling method went wrong the disease will spread all over the farm and make the enset plant wilt. Enset diseases are important factors determining the yield and quality of enset. Enset diseases identification can be carried out by means of digital image processing. In recent years, deep learning has made breakthroughs in the field of digital image processing, far superior to traditional methods. How to use deep learning technology to study diseases identification has become a research issue of great concern to researchers. This review provides a definition of enset diseases detection problem, and puts forward a comparison. with traditional detection methods. According to the difference of network structure, our project outlines the research on enset disease detection based on deep learning.

2.2 History of Enset and Its Bacterial Wilt Disease

Enset (*Ensete ventricosum*, Musaceae) is an African crop that currently provides the staple food for approx. 20 million Ethiopians. Whilst wild enset grows over much of East and Southern Africa and the genus extends across Asia to China, it has only ever been domesticated in the Ethiopian Highlands. Here, smallholder farmers cultivate hundreds of landraces across diverse climatic and agro ecological systems. Enset bacterial wilt was first reported by Yirgou and Bradbury (1968) in Ethiopia in 1968 and is currently found in all the enset growing regions and on wild enset plants [6]. Bacterial wilt is a destructive disease in all enset growing areas of Ethiopia. However, the current status and distribution of the disease in Ethiopia is not well assessed. Furthermore, farmers mention that bacterial wilt is more severe at high altitudes [6], but this has not been scientifically tested. Even though the pathogen attacks all stages of the plant [6], the comparative importance at different stages is not determined yet. Enset is propagated by

suckers or shoots rather than by seeds. Three to five years after propagation, enset suckers are ready for transplanting onto permanent fields, depending on agro-ecological conditions and location. Farmers often grow plants in 3 to 4 cycles (growing stages) with 2-3 transplanting (every year or two) in different fields over the course of a crop's lifecycle, increasing spacing in each cycle or stage. Research to determine the incidence of the disease at these stages would generate information that would contribute towards targeted management of the disease.



Figure 2-1 Healthy enset and Infected

2.3 Artificial Intelligence

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

The ideal characteristic of artificial intelligence is its ability to rationalize and take actions that have the best chance of achieving a specific goal. A subset of artificial intelligence is machine learning, which refers to the concept that computer programs can automatically learn from and adapt to new data without being assisted by humans. Deep learning techniques enable this automatic learning through the absorption of huge amounts of unstructured data such as text, images, or video. AI to develop a technology which functions like a human [7].This technology is perpetrated by studying how human brain thinks, how humans learn, make decisions, and work while solving a problem, and on this ground intelligent software and systems are

developed. These software's are fed with training data and further these intelligent devices provide us with desired output for every valid input, just like the human brain. Vast domains including Machine Learning and Deep learning are core part of AI. The industry is turning to Artificial Intelligence technologies to help yield healthier crops, control pests, monitor soil, and growing conditions, organize data for farmers, help with the workload, and improve a wide range of agriculture-related tasks in the entire food supply chain.

2.4.1 Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. [8] Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and deep learning models, machines can accurately identify and classify objects and then react to what they "see." We study and develop computer vision algorithms to assist or fully automate the enset plant identification process. [9] From the machine learning point of view, Computer vision methods for plant recognition have a number of applications, including mobile field guides using computer vision to automate or speed up the identification process, image data processing for biological databases, automatic detection, registration and mapping of plants from publicly available data, automation in agriculture, etc. Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. The studies of the plant diseases mean the studies of visually observable patterns seen on the plant. Health monitoring and disease detection on plant is very critical for sustainable agriculture. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time. Hence, in our project image processing is used for the detection of enset plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification. Within artificial intelligence (AI) and machine learning, there are two basic approaches: supervised learning and unsupervised learning. Supervised learning is one uses labeled data to help predict outcomes, while unsupervised learning uses unlabeled data. [10]

2.4.2 Deep Learning

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep neural networks have recently been successfully applied in many diverse domains as examples of end to end learning. Neural networks provide a mapping between an input such as an image of a diseased plant to an output such as a crop disease pair. The nodes in a neural network are mathematical functions that take numerical inputs from the incoming edges, and provide a numerical output as an outgoing edge. Deep neural networks are simply mapping the input layer to the output layer over a series of stacked layers of nodes. The challenge is to create a deep network in such a way that both the structure of the network as well as the functions (nodes) and edge weights correctly map the input to the output. Deep neural networks are trained by tuning the network parameters in such a way that the mapping improves during the training process. [11]

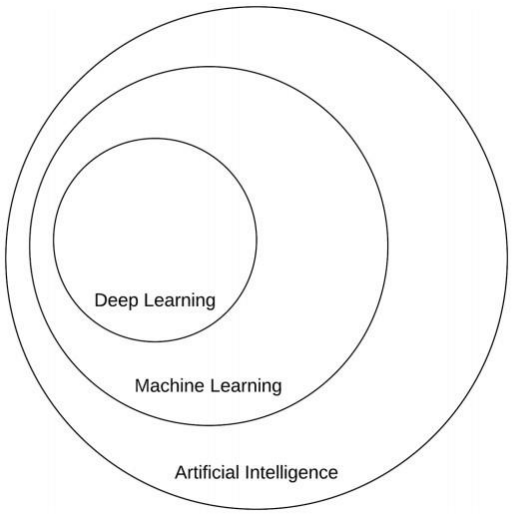


Figure 2-2 Deep learning *[12]*

2.5 Convolutional Neural Network (CNN)

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. The behavior of each neuron is defined by its weights. When fed with the pixel values, the artificial neurons of a

CNN pick out various visual features. When you input an image into a ConvNet, each of its layers generates several activation maps. [13] Activation maps highlight the relevant features of the image. Each of the neurons takes a patch of pixels as input, multiplies their color values by its weights, sums them up, and runs them through the activation function. The first (or bottom) layer of the CNN usually detects basic features such as horizontal, vertical, and diagonal edges. The output of the first layer is fed as input of the next layer, which extracts more complex features, such as corners and combinations of edges. [14]

2.5.1 Deep Convolutional Neural Network

Deep convolutional neural networks (CNN) have become a hot field in medical image segmentation. The key differences between CNN and other deep convolutional neural networks (DNN) are that the hierarchical patch-based convolution operations are used in CNN, which not only reduces computational cost, but abstracts images on different feature levels. [15] Since the patch-based learning is the core operations for both CNN and multiatlas segmentation. It is natural to integrate multiatlas segmentation with deep learning. As a result, significant efforts have been made to proposed label fusion-based CNN/DNN networks to leveraging the segmentation performance

2.7 Conclusion

Generally, we study in this literature about enset bacterial wilt role in our country, roles artificial intelligence has in agriculture, in machine learning types that supervised learning methods we used labeled datasets that we use to train our model and deep learning hidden layers, convolutional neural networks and the subclass of CNN that deep convolutional neural network.in this project we use the convolutional neural network for developing our model because of the training methods the model developing method for bacterial wilt detection follows the development methods and technique of CNN as the literature review of CNN

CHAPTER THREE

3. PROPOSED SYSTEM

3.1. Functional Requirements

The new system is expected to the following functionalities

User module

- The user opens the app and detects the Enset bacterial wilt disease.
- After the system detects the disease then the app displays the solution.
- The user can read statistical data from the app.

Feedback module

- The user gives feedback to the system.

Admin module

- The agricultural bureau can see the collected statistical data about the bacterial wilt disease.

Notification module

- the system gives alert to the farmers and agricultural experts to scout the Enset farm area.

3.2. Non-functional Requirements

The non-functional requirements of our system are: -

3.2.1. User Interface and Human Factors

The proposed system interface is easy to understand, easy to use and user-friendly to use and to carry out its task. Everything stems from getting to know our users, including their goals, skills, preferences, and tendencies. When designing our interface, we take the following factors into account after learning about our user:

- The UI has simplicity, interactive in order to use and reasonable segmentation of contents and has smooth functionality.

3.2.2. Hardware Consideration

The Software product to be developed should run on existing Android-based smartphones. Our system can run in any modern hardware devices.

3.2.3. Security Issues

The system is much secured based on the username and password for all user activity. Nobody can access the system without the authorized person. Passwords are not visible cannot be accessed by anyone because the passwords are encrypted by MD5 encryption.

- The admin should have the username and password.
- The password should be strong.
- The password should be encrypted.
- The system should give sessions and cookies to the user.

3.2.4. Performance Consideration

Response time of the System should be minimum. The system should show no visible deterioration in response time as the users capture images for identification of the disease from healthy. In order to secure authentication, the system considers farmers' knowledge skills.

3.2.5. Error Handling and Validation

The system has the capability to handle user error according the given image inputted. Response to error when the users enter incorrect image or distorted imaged in order to feature extraction. Fault tolerance Error should be considerably minimized and an appropriate error message that guides the user to recover from an error should be provided.

3.2.6. Backup and Recovery

The system includes a secondary database which is a copy of the original database used for a backup purpose that is used for recovery if a problem is occurred with the original one.

3.2.7. Documentation

The well-defined document in our system is designed to keep the system simple. We will also be prepared for system users for a short and accurate aid file. It will provide the system user with a help guide. It shows how they are going to be used.

3.3 Trained model development

3.3.1 Model development

In this study we will follows a basic machine learning workflow:

1. Examine and understand data

2. Build an input pipeline

3. Build the model

4. Train the model

5. Test the model

6. Improve the model

The data set contains two types of leaf images these data sets are labeled as healthy and infected and all are taken with high resolution cameras. Machine learning pipeline is a method of codifying and automating the method of developing a machine learning model. Machine learning pipelines are made up of a series of processes that handle everything from data extraction to preprocessing to model training and deployment.

3.3.1.1 Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. Data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model.

Image Labelling

Data labelling is an essential step in a supervised machine learning task. In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it. Image labeling gives your insight into the content of images. Each label found comes with a score that indicates the confidence the ML model has in its relevance. So, in this study we labeled the image as healthy leaf and infected leaf

Image resizing

The collected image data are taken using digital camera the megapixel size of cameras get larger, it means that the image data will be larger in pixel and document dimension sizes. These larger sizes do matter when it comes to developing model. So, the researcher used image resizing technique using Opencv cv2.resize module. The reason why image resize is importance is to make image files the correct width and height.

Augmentation Process

The main purpose of applying augmentation is to increase the dataset and introduce slight distortion to the images which helps in reducing overfitting during the training stage. In machine learning, overfitting appears when a machine learning model describes random noise or error rather than underlying relationship. Deep networks need large amount of training data to achieve good performance. To build a powerful image classifier using very little training data, image augmentation is usually required to boost the performance of deep networks. Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc. An augmented image generator can be easily created using ImageDataGenerator API in Keras. ImageDataGenerator generates batches of image data with real-time data augmentation. The most basic codes to create and configure ImageDataGenerator and train deep Convolutional Neural Network.

Feature Extraction

Extracting good features is the most significant step in identification of disease in the system. Feature extraction methods are like color, texture and shape as feature vector. Neural networks are very promising technique for feature extraction due to powerful parallel mechanism of computation. Convolutional Neural Networks is used for feature extraction and classification of object. In order to carry out image classification, the neural network must carry out feature extraction. Features are the elements of the data that you care about which are fed through the network. In the specific case of image recognition, the features are the groups of pixels, like edges and points, of an object that the network is analyze for patterns.

3.3.1.2 Training and testing the model on data

For training a model we initially split the model into 3 three sections which are ‘Training data’, ‘Validation data’ and ‘Testing data’. Train the classifier using ‘training data set’, tune the parameters using ‘validation set’ and then test the performance of our model on unseen ‘test data set’. During training the classifier only the training and validation set is available. The test data set must not be used during training the model. The test set will only be available during testing the model.

Training set: The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning that is to fit the parameters of the classifier.

Test set: A set of unseen data used only to assess the performance of a fully-specified classifier.

3.3.1.2 Code our model

To develop the model, we will develop a deep fully-connected ("feed-forward") neural network model that can predict the bacterial wilt disease presence by phenotypic characteristics. To develop our model, we will use the following framework: Tensor Flow, Keras, Theano and Scikit-learn.

Train, evaluate, and tune the model

After setting up the dataset will train and evaluate our model in the Google CoLab. In addition, we will perform hyper parameter tuning functionality to optimize the training process.

When training our model, we feed the data for which the value for the target data attribute (feature). We will run the model to predict those target values for our training data, so that the model can adjust its settings to better fit the data and thus to predict the target value more accurately. Similarly, when evaluating our trained model, we feed it data that includes the target values. Then we will compare the results of our model's predictions to the actual values for the evaluation data and we will use statistical techniques appropriate to our model to gauge its success. During training, we will apply the model to known data to adjust the settings to improve the results. When our results are good enough for the needs of our application, we deploy the model to system and test it. During the testing process, we make adjustments to the model

parameters and hyper parameters based on the results of the testing. We will apply hyper parameter tuning for the model by changing the operations or settings that we use to control the training process, such as the number of training steps to run.

Hyper parameter tuning is one of the fundamental steps in the machine learning routine. Also known as hyper parameter optimization, the method deals finding for the best configuration of hyper parameters to get optimal performance. Machine learning algorithms require user-defined inputs to achieve a balance between accuracy and generalizability

Generally, we develop our model by using our enset data set which is we collect from the farms which is contain the two types of image that is infected and healthy leaf we follow the model building pipe line and develop our model which is feed forward fully connected neural network model during the development of the model we use the validation and test set to validate and test the model accuracy. To optimize the model accuracy, we use hyper parameter optimization method that helps us to find the optimal performance. After the model building the model will be used on the web application that we develop.

CHAPTER FOUR

4. SYSTEM ANALYSIS

4.1 System Model

System model is the conceptual model as a result of system modeling that describes and represents a system.

4.1.1. Use Case Model

4.1.1.1 Use Case Diagram

In this system we identified both actors and use cases (functionalities) as the following:

ACTOR	USE CASE
Farmers	Detect and Give Feedback
Agricultural expert	Login/Logout, Detect, View statistical data, Give Feedback and Add statistical Data
Admin	Detect, Login/Logout, View data, View feedback and Add Statistical Data

Table 4-1 Actor and use case information

Farmers:

- Farmers can detect the disease by using app
- Farmers can give feedback to the system

Agricultural Expert:

- Agricultural expert can login to the system
- Agricultural expert can view statistical data
- Agricultural expert can detect disease type by using app

- Agricultural expert can give feedback to the system
- Agricultural experts can add statistical data.
- Agricultural expert can logout from the system

Admin:

- Admin can login to the system
- Admin can view statistical data
- Admin can view feedback given by user
- Admin can logout from the system

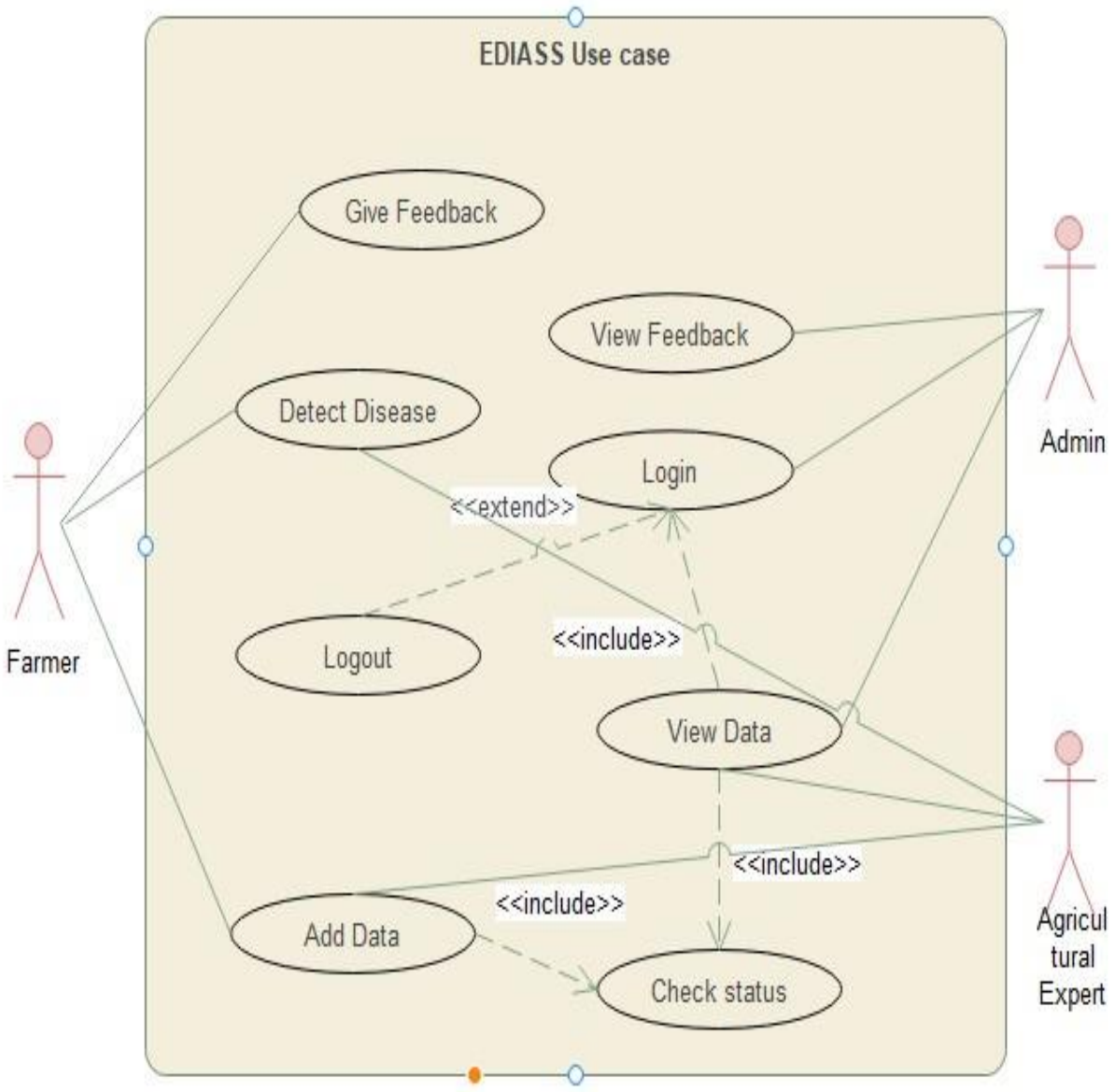


Figure 4-3 Use case diagram for enset disease identification

4.1.1.2. Use Case Description

- **Detect Disease**

Use case ID	UC_1	
Use case name	Detect Disease	
Description	Allow to identify the disease type	
Actor	Farmers and Agricultural experts	
Pre-Condition	The user must have this application and login as farmer or Agricultural expert	
Basic course of action	Actor Action	System response
	Step 1 Users open the application and login as farmer/agricultural experts. **Step 2** Users take the application on to enset to detect the disease.	**Step 3** The Application starts detecting the disease. **Step 4** The app identifies either it got disease or the healthy one.
Alternative course of Action	The app display “there is no disease”	
Post condition	Identify disease	

Table 4-2 Detect use case description

-

Feedback

Use case ID	UC_2	
Use case name	Feedback	
Description	Commenting on the application.	
Actor	Farmers and Agricultural expert.	
Pre-Condition	User first login to the app as farmer or agricultural expert.	
Basic course of action	Actor Action	System response
	Step 1 Users click on the comment link. **Step 3** User enters the comment and clicks on the button.	**Step 2** The system displays comment form. **Step 4** The system validates entered input. **Step 5** The system will display the comment. **Step 6** End of use case.
Alternative course of Action	**Step 4.1** User enters the wrong input or left blank fields.	

	I. The System displays Messages that contain “this field is important” will be displayed. II. User repeat step 2.
Post condition	Comment sent successful

Table 4-3 Feedback use case description

- **Add Statistical Data**

Use case ID	UC_3	
Use case name	Add Statistical Data	
Description	Admin add statistical data about enset disease.	
Actor	Agricultural expert and Admin.	
Pre-Condition	Admin must login to the system with his role.	
Basic course of action	Actor Action	System response
	Step 1 user first open app and upload image on the phone.	**Step 2** data is added on the system when enset plant is detected
Alternative course of Action	**Step 1.1** if image is not detected Data is not added on the system.	
Post condition	Data is added.	

Table 4-4 Add statistical data use case description

-

View statistical data

Use case ID	UC_4	
Use case name	View Statistical Data	
Description	User view statistical data about enset disease.	
Actor	Agricultural expert and Admin	
Pre-Condition	System is ready to display statistical data to the user	
Basic course of action	Actor Action	System response
	Step 1 User clicks statistical data button.	**Step 2** System display statistical data to the user.
Post condition	The system display statistics about enset disease.	

Table 4-5 View statistical data use case description

- **Login**

Use case ID	UC_5
Use case name	Login
Description	Allow Admins enter to the system and perform operation.
Actor	Admin and Agricultural expert

Pre-Condition	User must have valid user name and password in the system.	
Basic course of action	Actor Action	System response
	Step 1 User clicks the button in the home page. **Step 3** User fulfills the username and password and then click on login button.	**Step 2** The system display login page. **Step 4** The system validates entered value. **Step 5** Now Admin has access to his account. **Step 6** End of use case.
Alternative course of action	4.1.user enters invalid username and password I. The System displays messages that contain “field cannot be empty or you entered wrong username or password” will be displayed. II.. user repeat number Step 3.	
Post condition	The user will have access to his account.	

Table 4-6 Login use case description

-

Logout

Use case ID	UC_6	
Use case name	Logout	
Description	Exiting from the system	
Actor	Admin and Agricultural Expert	
Pre-Condition	Login into system	
Basic course of action	Actor Action	System response
	Step 1 user clicks on (log out) button on the home page	**Step 2** The system exit user from the system. **Step 3** End of use case.
Alternative course of action	Nothing	
Post condition	Exit.	

Table 4-7 Logout use case description

- **View feedback**

Use case ID	UC_7	
Use case name	View feedback	
Description	Feedbacks that are sent to developer.	
Actor	Developer	
Pre-Condition	Email	
Basic course of action	Actor Action	System response
	Step 1 Developer clicks on feedback button on home page. **Step 3** User view notification.	**Step 2** The System displays generated feedback from user. **Step 4** Use case end.
Alternative course of Action	Nothing	
Post condition	Get important information.	

Table 4-8 View feedback use case description

4.1.1.3. Use case Scenario

Scenario is description that shows the process/steps while the interaction between system & user.

Scenario 1:

Scenario name: Detect Disease

Participant actor: Admin, Agricultural expert, Farmer

Flow of events

If a user wants to detect bacterial wilt disease first user must have this application and then login as a farmer or Agricultural expert or admin. If user is farmer then by using app detect the disease and know that enset crop is weather healthy or infected but if a user is admin or agricultural expert after they login to the app they fill username and password then by using app camera detect the disease and then he/she identify the disease that occur on that enset crop and give suggestion to the farmer. Admin or agricultural experts also add collected statistical data about enset disease and also view those added data.

Scenario 2:

Scenario name: Give feedback

Participant actor: Agricultural expert, Farmer.

Flow of events

If the user gives feedback about the application. User first login to the app as farmer or agricultural expert. User clicks on the comment link then he/she gives any comment they want then the system validates entered input then after the comment is sent.

4.2. Object Model

An object model is a logical interface, software or system that is modeled through the use of objectoriented techniques. It enables the creation of an architectural software or system model prior to development or programming. In our system we use object model like class diagram and data dictionary.

4.2.1. Class Diagram

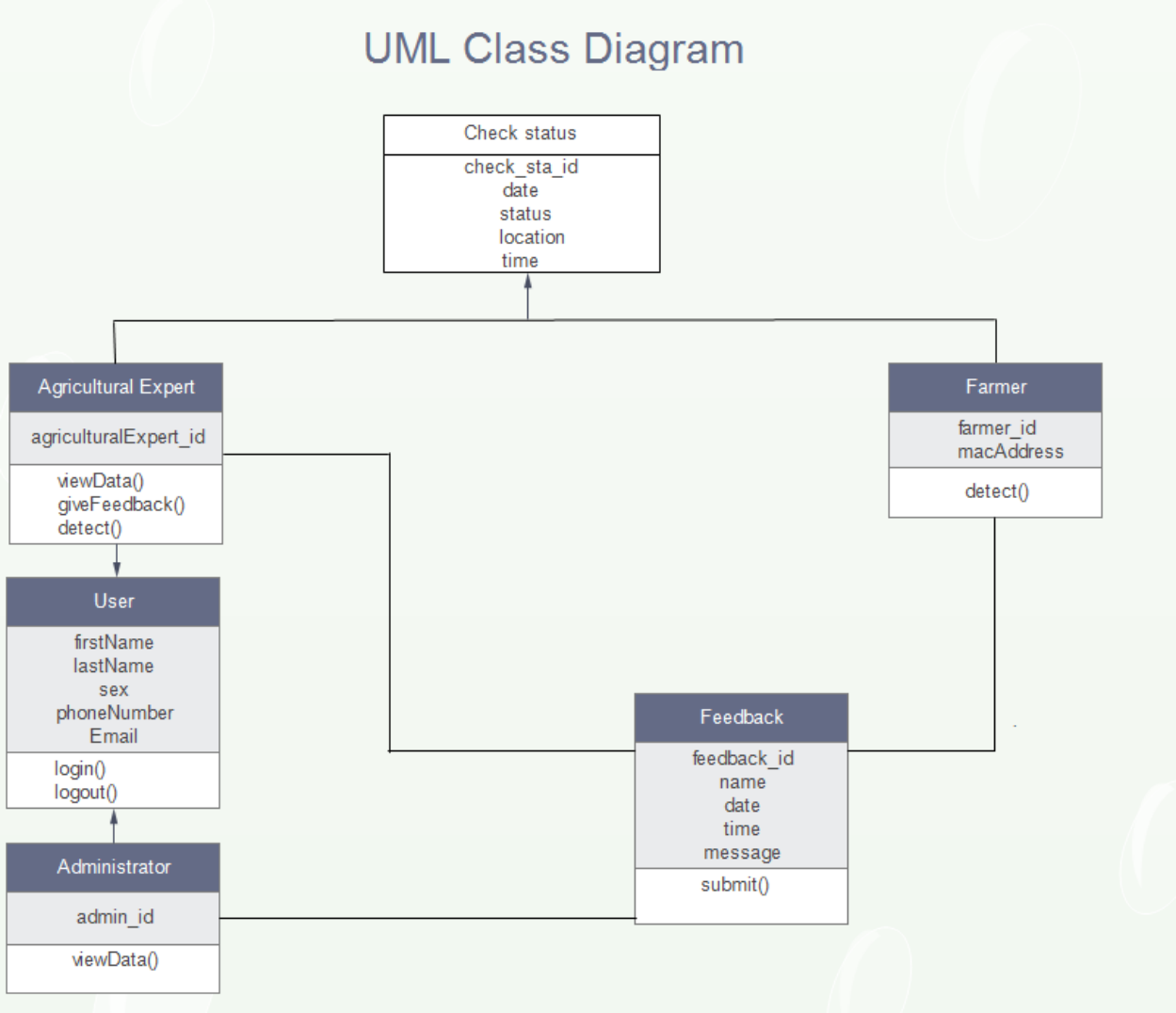


Figure 4-4 Class Diagram

4.2.2. Data Dictionary

As the system able to save/store information about real world entity in to different data files we present data dictionary of the system that describe which data is stored in which data file. Data dictionary is description of database table of the system such as functionality, data types, data size, contents item and table relations.

Attributes	**Data Type**	**Data size**	**Constraints**	**Description**
Feedback_Id	String	30	PRIMARY KEY	Id of feedback
Name	String	30	NOT NULL	Name of feedback giver
Date	String	25	NOT NULL	Date of feedback written
Time	Time	20	NOT NULL	Time of feedback written
Message	String	50	NOT NULL	Message of the feedback

Table 4-1 Data dictionary for feedback

Attributes	**Data Type**	**Data size**	**Constraints**	**Description**
Statistical_Data_Id	String	30	PRIMARY KEY	Id of statistical data
Date	String	30	NOT NULL	Date of statistical data
Time	Time	25	NOT NULL	Time of statistical data
Status	String	20	NOT NULL	Status of the Enset(healthy or infected)
Location	String	50	NOT NULL	Location of statistical data taken

Table 4-2 Data dictionary for statistical data

4.3. Dynamic Model

The dynamic model represents the time dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. It is used where the object's behavior is best described as a set of states that occur in a defined sequence. In our system we use sequence diagram, activity diagram, and State diagram.

4.3.1. Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario so in our project there will be different processes to do specific actions and we include some sequence diagrams to handle these interactions.

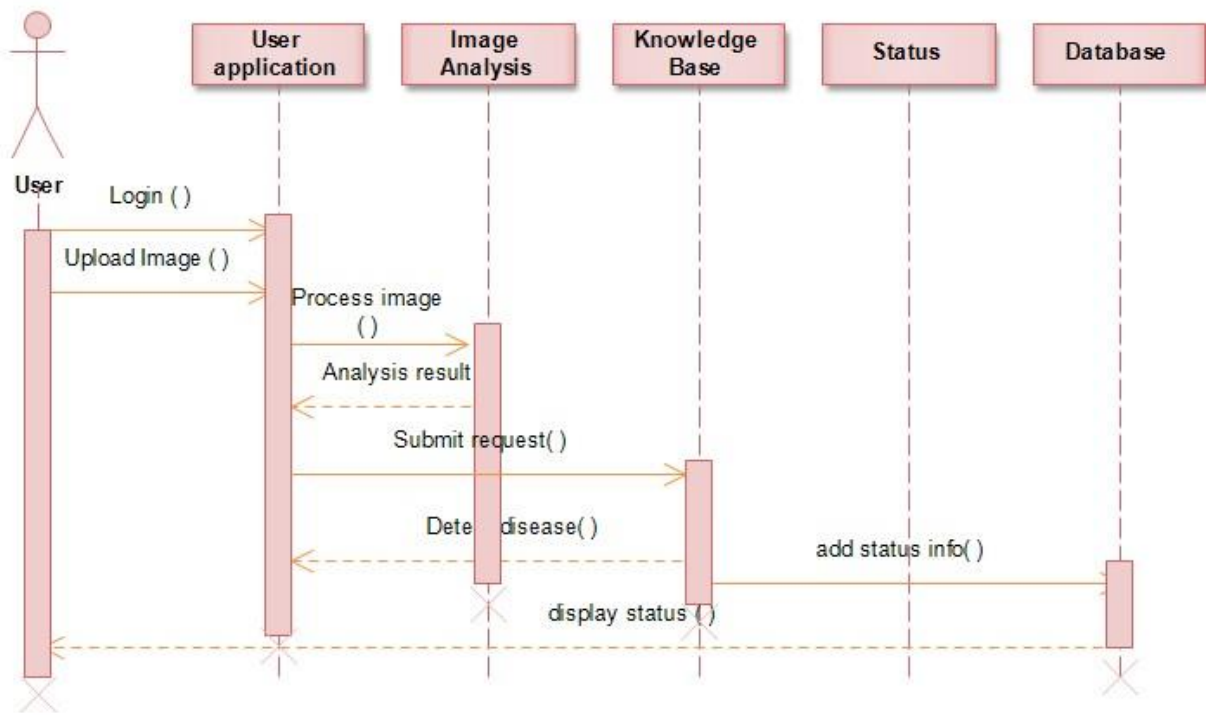


Figure 4-5 Sequence diagram for detect disease

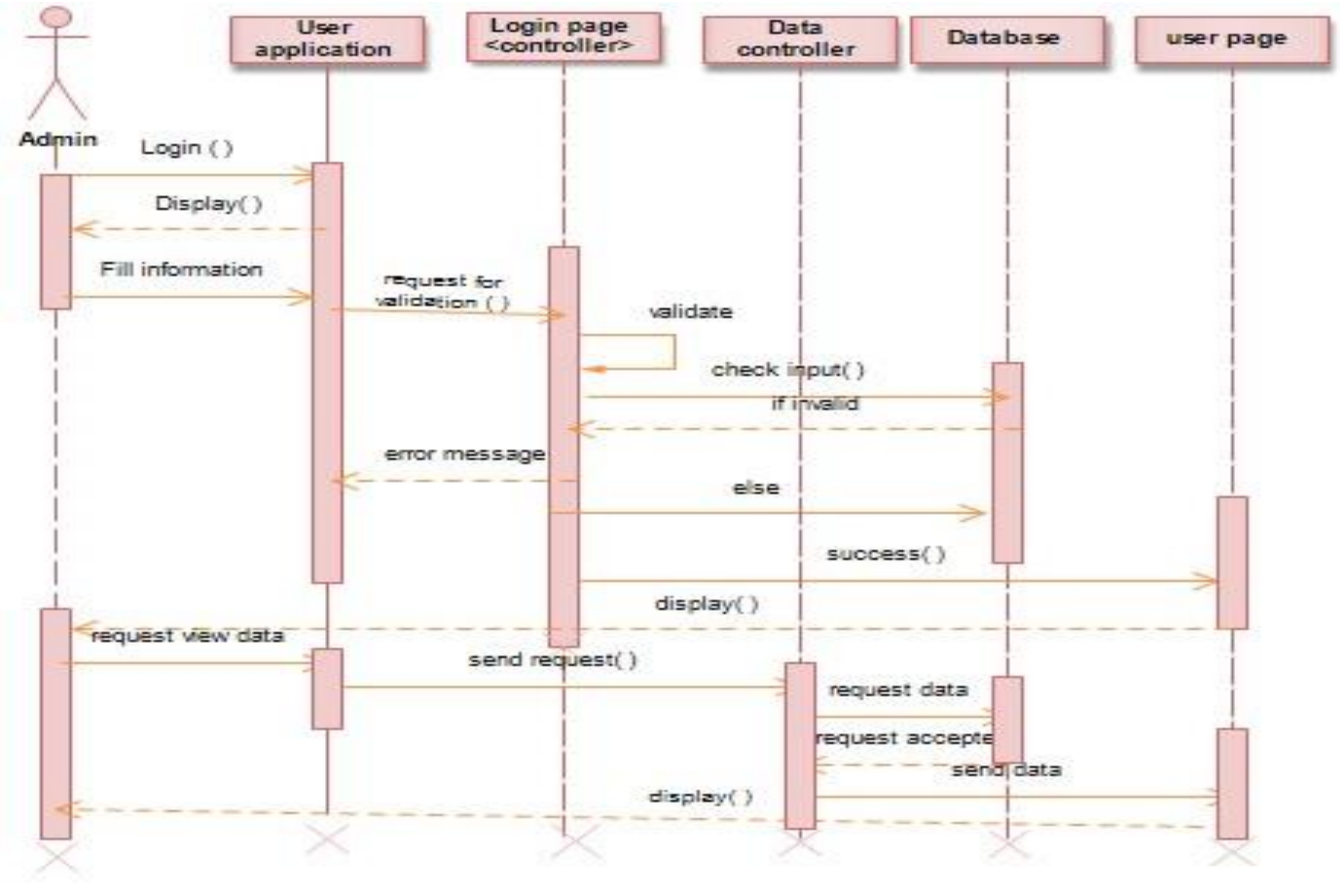


Figure 4-6 sequence diagram for view data

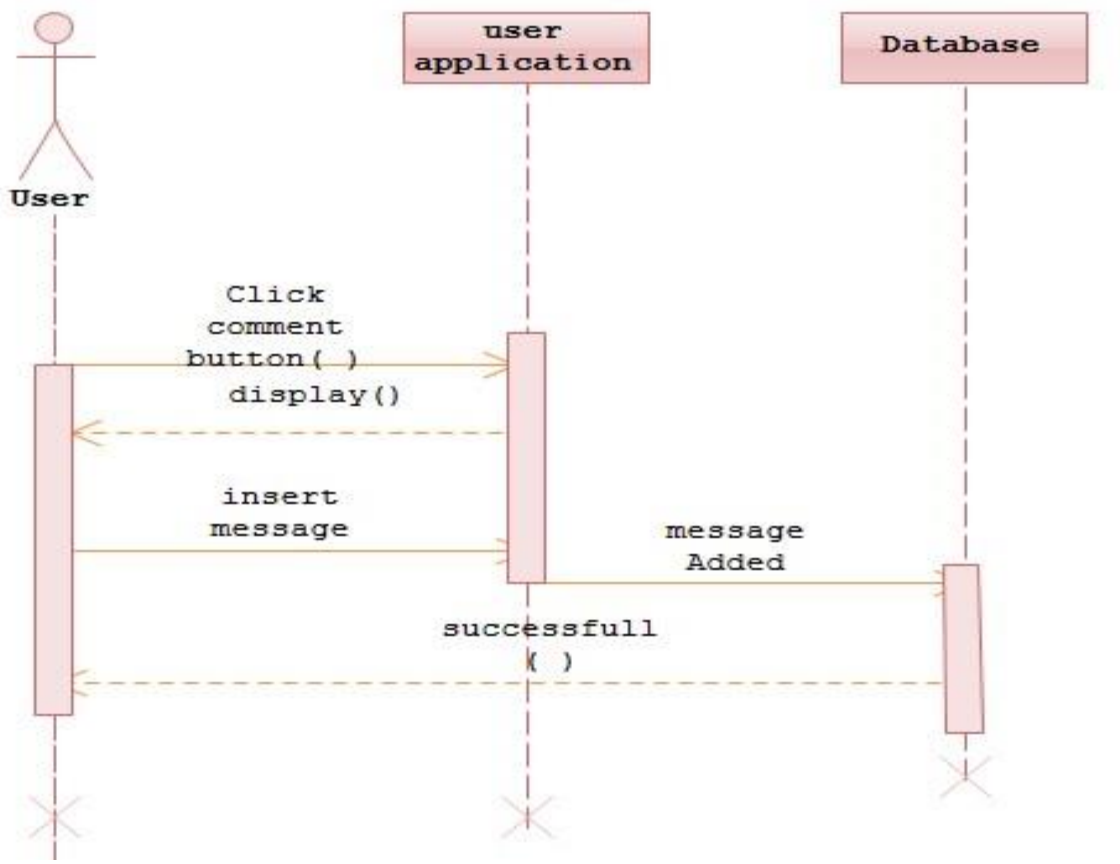


Figure 4-7 sequence diagram for feedback

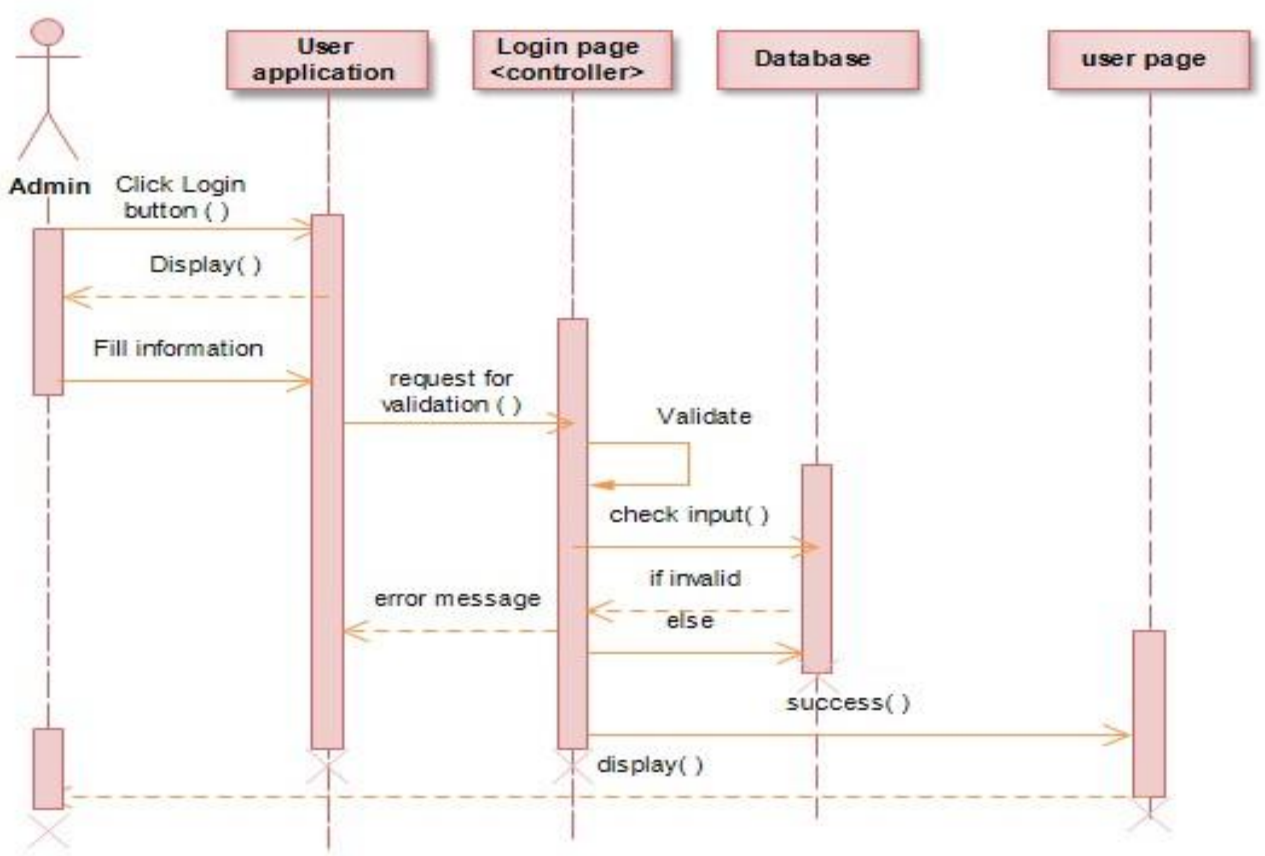


Figure 4-8 sequence diagram for Login

4.3.2. Activity Diagram

Activity diagram is basically a flow chart to represent the flow form one activity to another Activity. The activity can be described as an operation of the system in order to perform many Tasks there should be one or more activity to be done

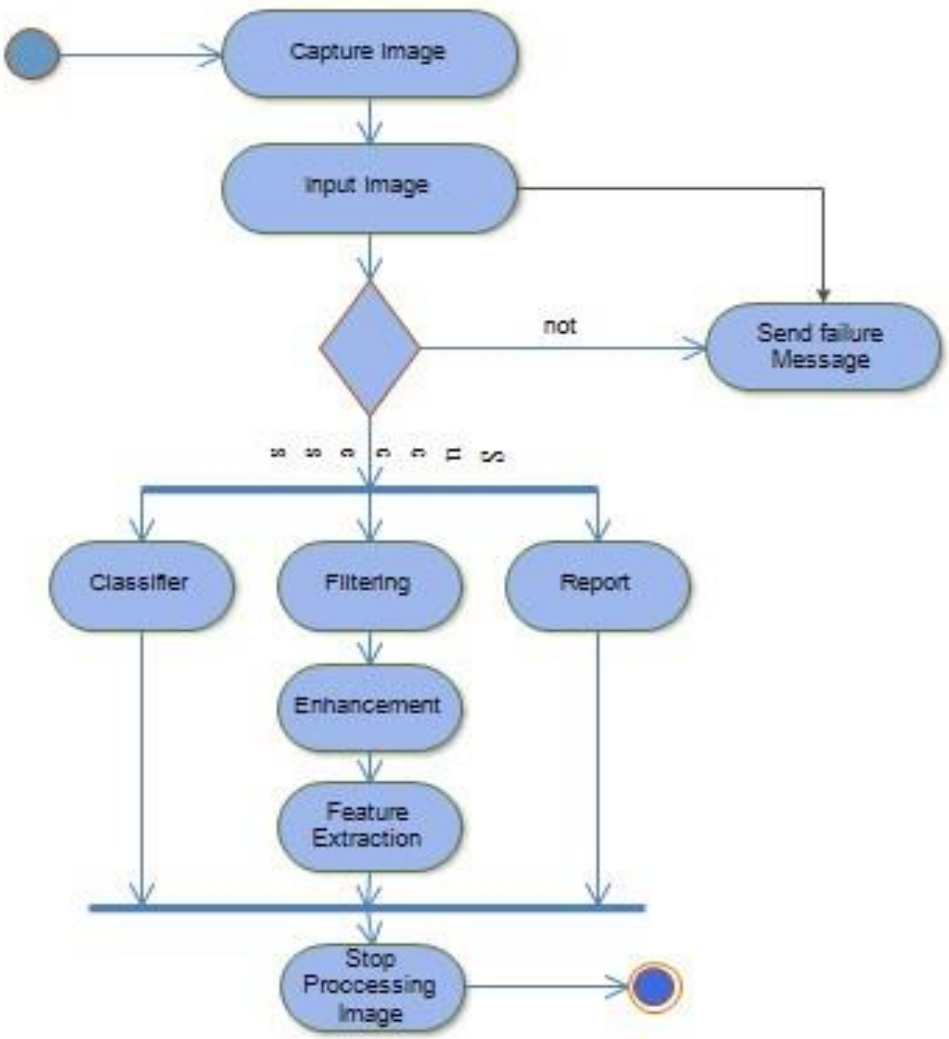


Figure 4-9 Activity diagram for detection

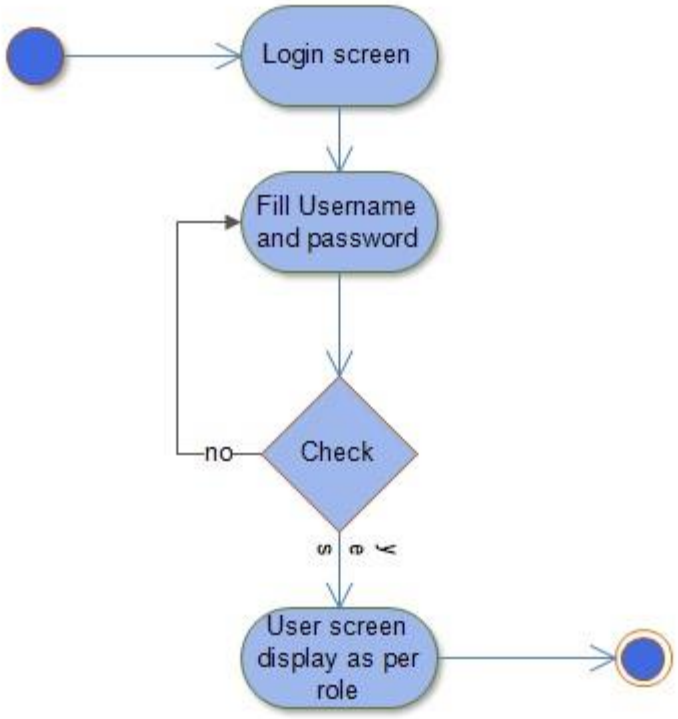


Figure 4-10 Activity Diagram for Login

4.3.3. State Chart Diagram

State chart diagram is to model life time of an object from creation to termination whenever performing different activities there will be some states that are included in activities so in our project we have listed below some states that can be covered during the user interacts with the system.

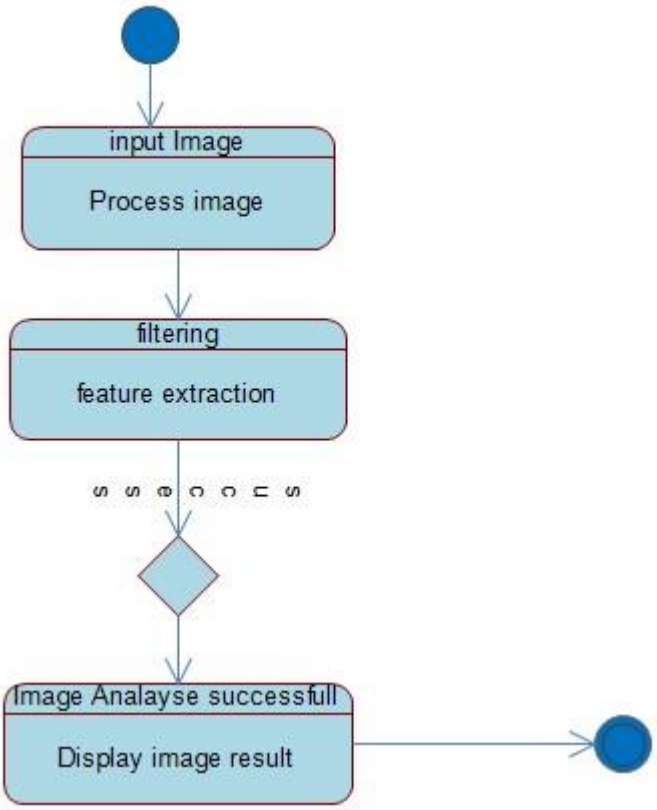


Figure 4-11 state diagram for detection disease

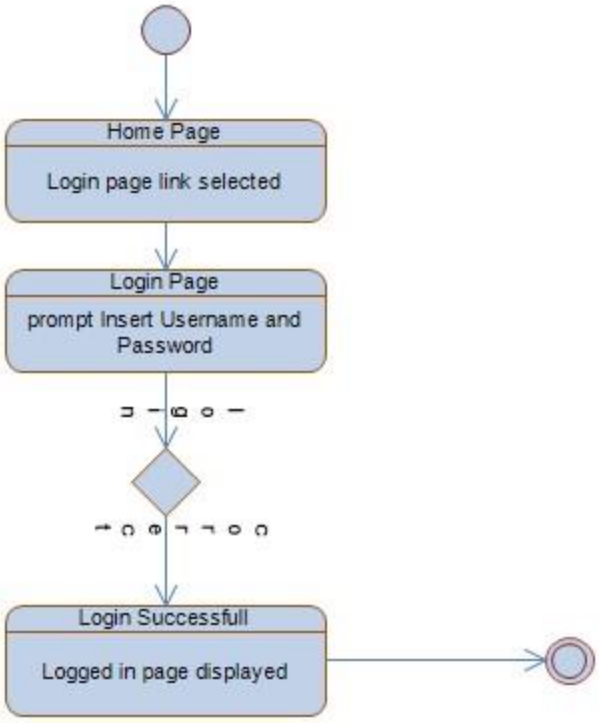


Figure 4-12 state diagram for login

4.4 Model development diagram

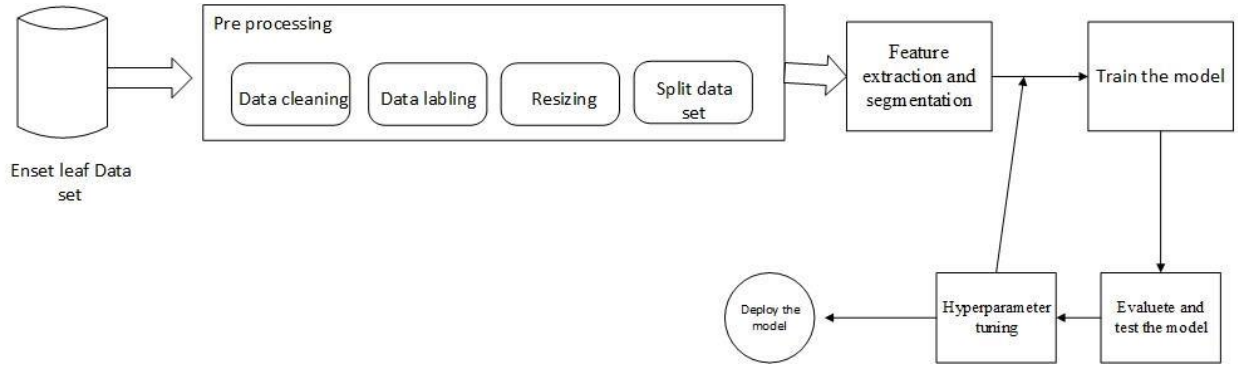


Figure 4-13 machine learning model development for EDBW

CHAPTER FIVE

5. SYSTEM DESIGN

The system consists of a mobile application, which will enable the farmers to take images of plants using their mobile phones and send it to a central server where the central system in the server will analyze the pictures based on visual symptoms using image processing algorithms in order to measure the disease type. An expert group will be available to check the status of the image analysis data and provide suggestions based on the report and their knowledge, which will be sent to the farmer as a notification in the application.

5.1. Design Goals

Design goals specify the quality of the system that should be achieved and addressed during the design of the system. The objectives of design are to model the system with high quality. It also addresses non functionality of the system.

Correct: Our system is designed correctly in which all functional and nonfunctional requirements are satisfied.

Robust: in order to user detect the disease by using an app when there is error the system tolerates due to future extraction from enset plant.

Performance: Response time of the System should be minimum. The system should show no visible deterioration in response time as the users capture images for identification of the disease from healthy. In order to secure authentication, the system considers farmers' knowledge skills.

Usability: The system should provide an easy-to-use graphical interface so users can easily learn how they use the App. The system considers farmers' knowledge in order to use the application.

Availability: The system should be available for any users at any time at any place. The system should be available 24 hours a day, 7 days a week.

Security: the system has two ways of security evaluation methods. The first way is farmers login way the other one is admins and agricultural expert’s login way.

Scalability: since the system is modular it is assumed that the system will be scalable for future use.

Accessibility: Accessibility is the ability of the system to be accessible to users. Any user can access the system anywhere in the enset farm.

Fault tolerance: Whenever the user commits errors intentionally or unintentionally the system should continue performing its task without interruption by displaying user-friendly error message.

5.3. System Architecture

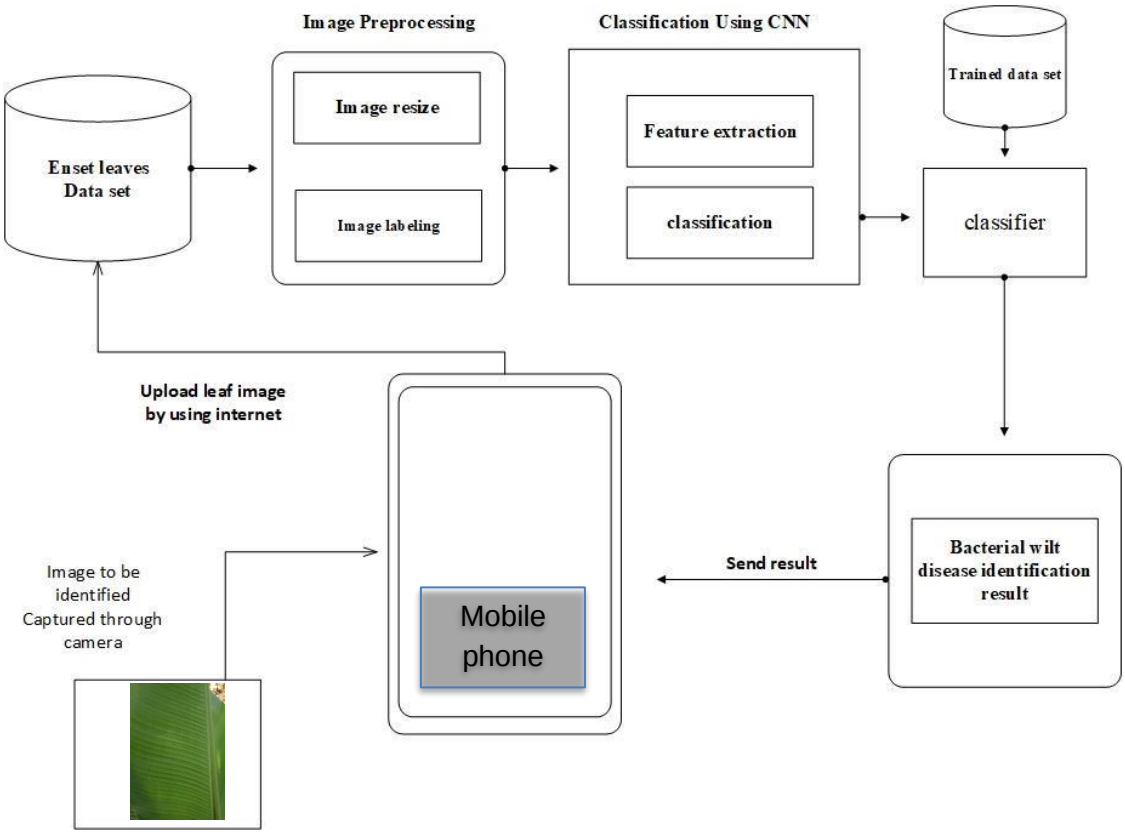


Figure 5-1

Figure 5-1 system architecture

5.3.1. Subsystem Decomposition and Description

Subsystem decomposition is the process of dividing the system in to manageable subsystems from the analysis model of the proposed system. The goal of the system decomposition is to reduce the complexity of design model and to distribute the class of the system in to large scale and cohesive components. Components are generally units of computation or data stores in the system. A component has a name, which is generally chosen to represent the role of component or the function it performs. The different components of a system are likely to interact while the system is in operation to provide the service expected of the system.in our system the following subcomponent is available.

Administrator subsystem: responsible for handling action that administrator

Performs.

- login
- view statistical data
- view feedback
- logout

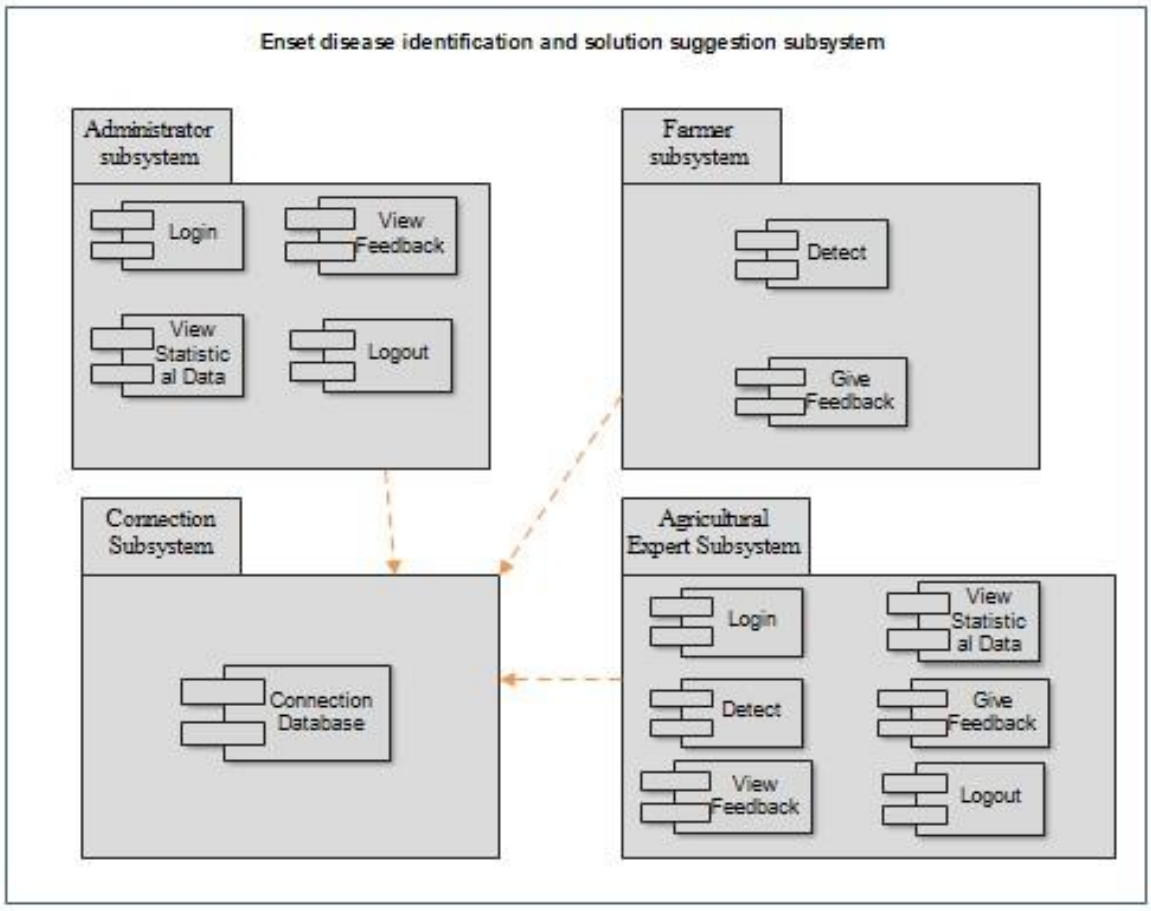
Farmers Subsystem: responsible for handling action that Farmers perform.

- detect
- give feedback

Agricultural Expert Subsystem: perform for some activity that System available the viewer and handling actions that experts perform.

- login
- view statistical data
- detect
- give feedback
- Add statistical data.

- logout



5-2 Subsystem Decomposition

5.3.2. Hardware/Software Mapping

Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware Administrator subsystem Login View Statistical Data View Feedback Logout Enset disease identification and solution suggestion subsystem Farmer detect Give feedback connection subsystem Connection database Agricultural Expert Login detect view statistical data view feedback Give feedback Logout used to connect the disparate machines to one another. You want to create a deployment diagram for applications that are deployed to several machines. It also shows how the software and the hardware components work together. Deployment diagram used to show the hardware of the system, the software that is installed in the hardware and also the middleware that is used to connect the disparate machines to one and other.it also shows how the software and hardware components of the system work together.

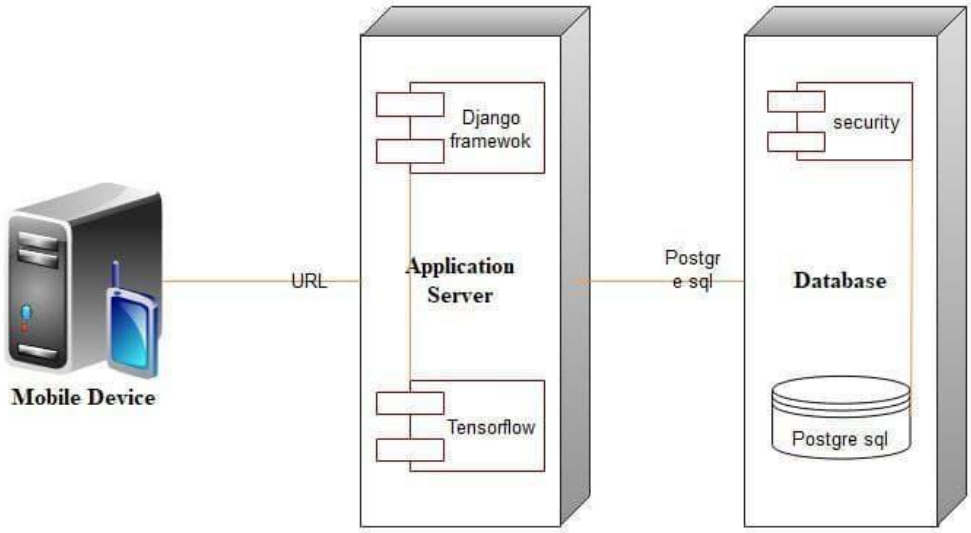


Figure 5 3 Deployment Diagram

5.3.3. Detailed Class Diagram

Class diagram shows the static structure of data and the operations that act on the data, i.e.it shows the static structure of an object-oriented model the object class, their internal structure, and the relationships in which they participate.

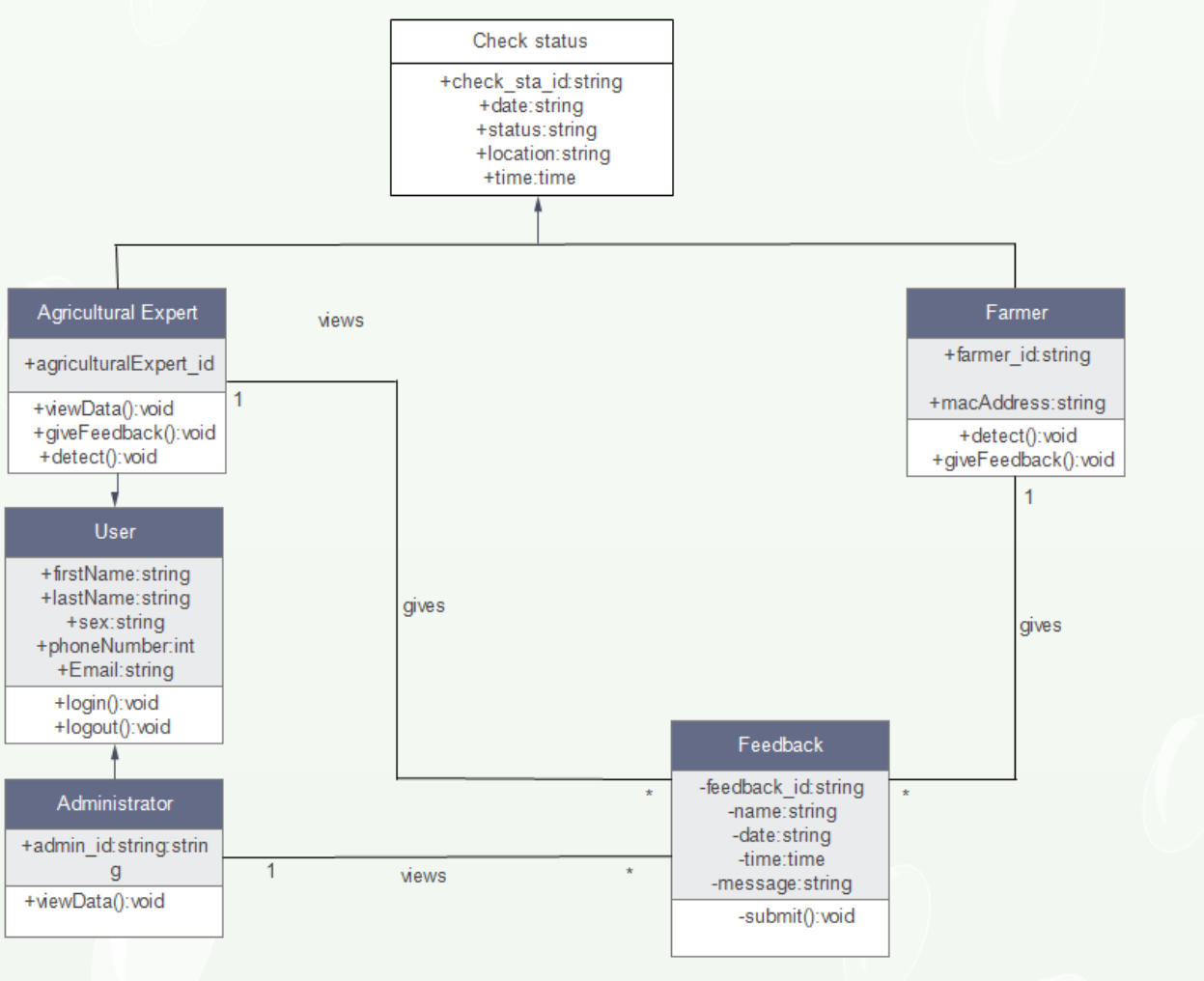


Figure 5 4 Detail Class Diagram

5.3.4. Persistent Data Management

Objects in the models are persistent and provide clean separation points between subsystems with well-defined interfaces. In order to store information persistently we map objects into tables and the attributes into fields to the specific table based on the objects found on the system. This part is to describe and show the necessary relationships among the tables, which are selected to store the data persistently in the system. The following tables indicate the persistence data management of the system.

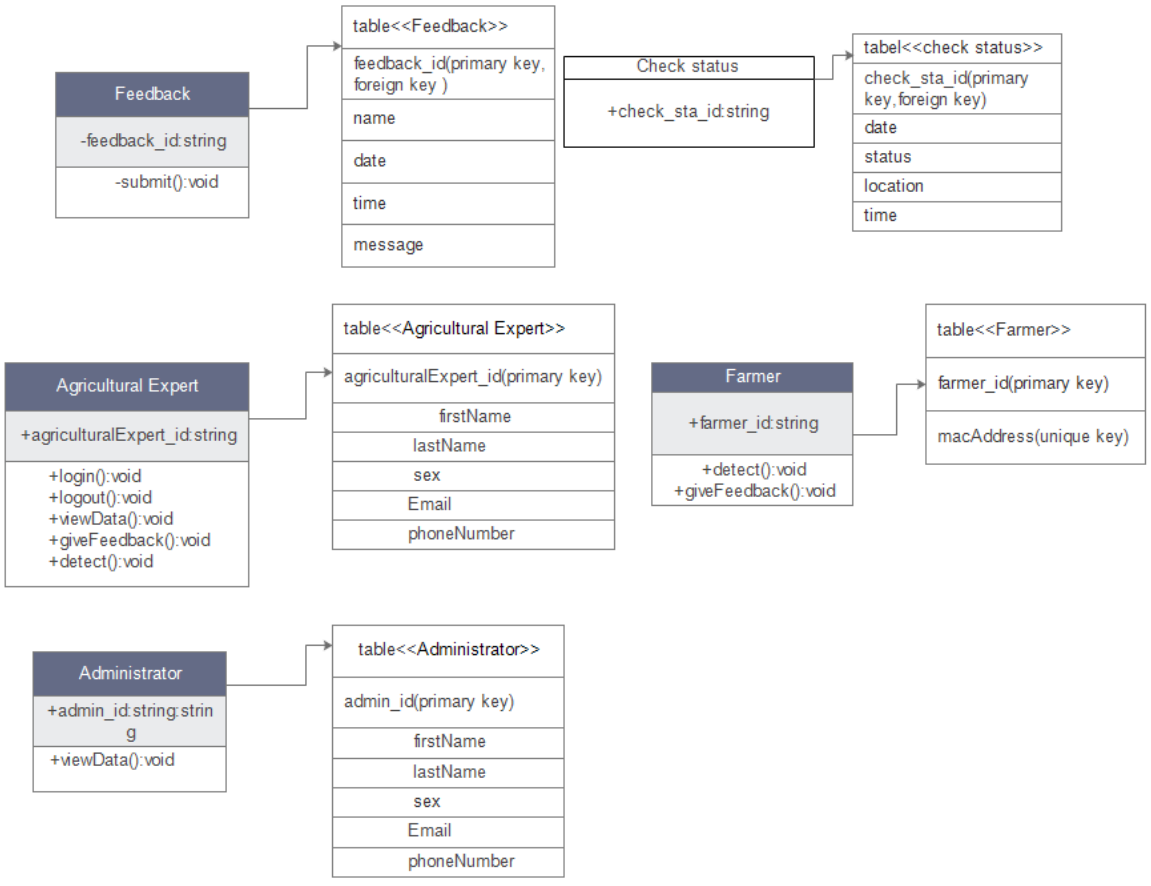


Figure 5.5 Persistent Data Management di

5.4 Packages

Package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality. In our system we category into Feedback package management, Machine learning Model Identifier and Account Management package.

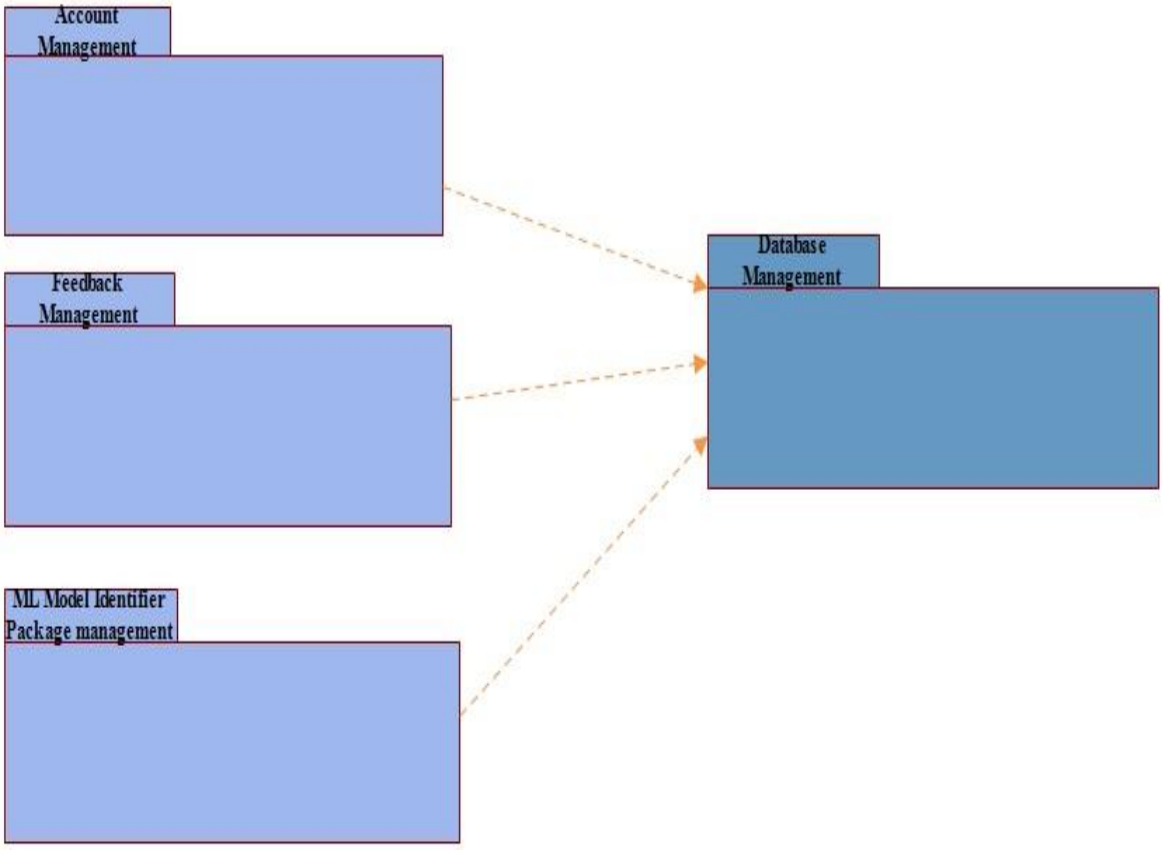


Figure 5-6 Package Diagram

5.5. Algorithm Design

For making the Model

- Import data set
- If image path is not correct throw error message
- Elseif import the image from the location given
- Convert the image in to array
- Then return the array
- Convert the labels of the images
- Save the images instance using pikel library
- Scale the data points from [0,255] to the range [0,1]
- Split trading and test sets from the data imported by the ratio 3:1
- Augment the data set (rotate, crop, flip, contrast)
- Create the model using filters and relu activation
- Create two steps of conventional layer
- Then optimize the model
- Train the model
- For the training and validation accuracy plot graph
- Then calculate accuracy of the model

Login

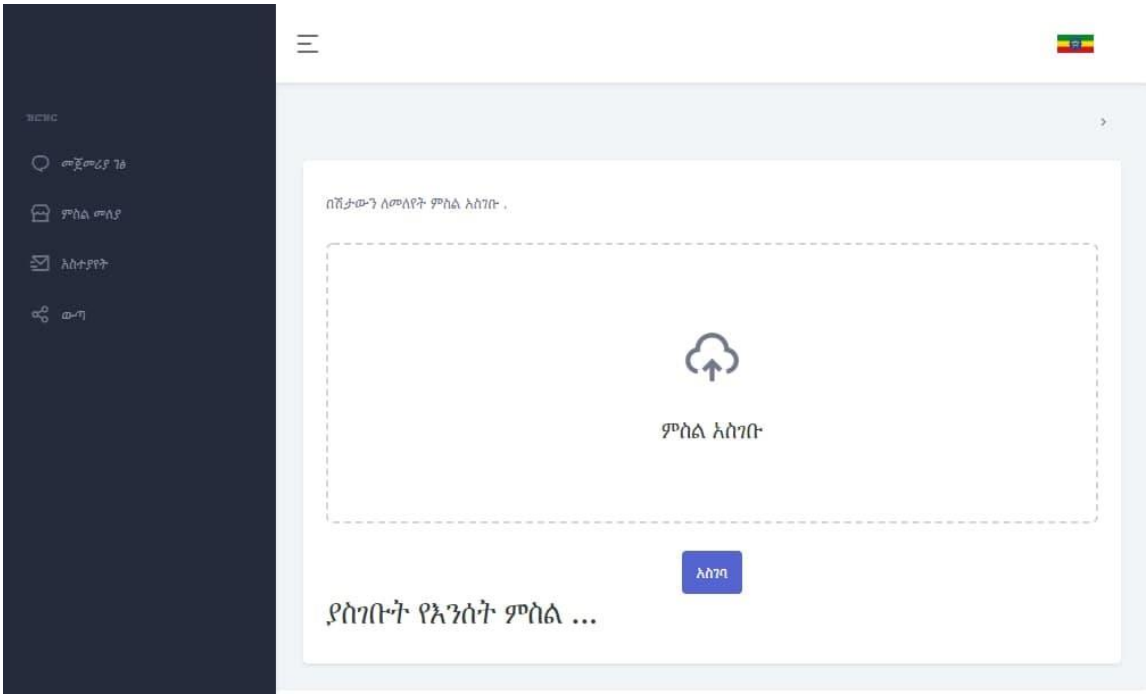
- Home page displayed.
- User click on farmer or expert
- If user click farmer and
- if mac address to the registered before create session and enter to the system
- Else register mac address and create session
- If the user clicks on the expert
- System display sign in page.
- User enter user email and password.
- **If** user email and password is correct, **then**
- **If** user is farmer, **then** system display farmer page.
- **Else if** user is expert**, then** system display expert page.
- **Else if** user is admin, **then** system display admin page.

- **Else** user email and password is not correct, then system display error Message and redisplay login page

Real Time Detection

- Landing page displayed
- Click detect button
- Detection process instantiate camera
- Then the captured image will transfer to the server for identification
- If the leaf is not infected by bacterial wilt disease this message will display “healthy leaf”
- Else this message will display “this enset is infected by bacterial wilt disease” and the status will save in to the data base

5.5. User Interface Design



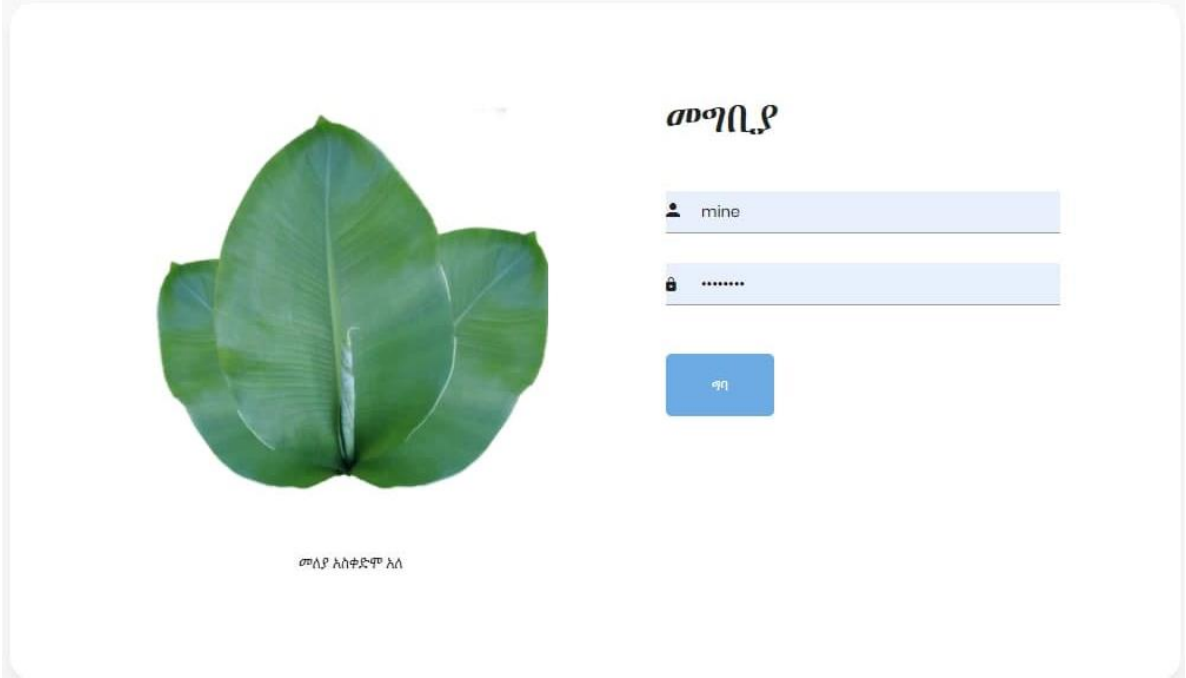


Figure 5-7 User interface Design

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

Implementation phase is the most essential stage to develop the required system. It is the phase in which the design and analysis of the system is transformed into a tangible system. This is done by writing code and makes it operational. Before we go to coding part we gathered datasets form village's Yerbatye (የርባትየ) and Yeferezye (የፈረዝየ) located in wolkite emdeber road that far from our university about 24 km at wolkite university conserved farm. By using Samsung M11, Samsung digital camera and Canon 5D mark III we gather that of infected and healthy datasets then we augment and we get 4000 datasets.

6.1. Implementation of the Database

We use PostgreSQL Database PostgreSQL is an advanced, enterprise-class, and open-source relational database system. PostgreSQL supports both SQL (relational) and JSON (non-relational) querying. PostgreSQL is primarily used as a robust back-end database that powers many dynamic websites and web applications. In order to do this we create tables for registration, feedback and for statistical data.

6.2. Implementation of the Class Diagram

In the implementation of class diagrams we define attributes with the appropriate data type and access level visibility (i.e. private, protected and public). We define all methods with the appropriate return type, parameter and the corresponding proper data type and access level visibility.

6.3. Twilio SMS Server

We use twilio sms service for farmers give information about their enset farm and first of all we create an account and buy phone number and phone number has token and SID and through API get sms twilio message is sent to the phone number of the user (farmer) in order to check his farm.

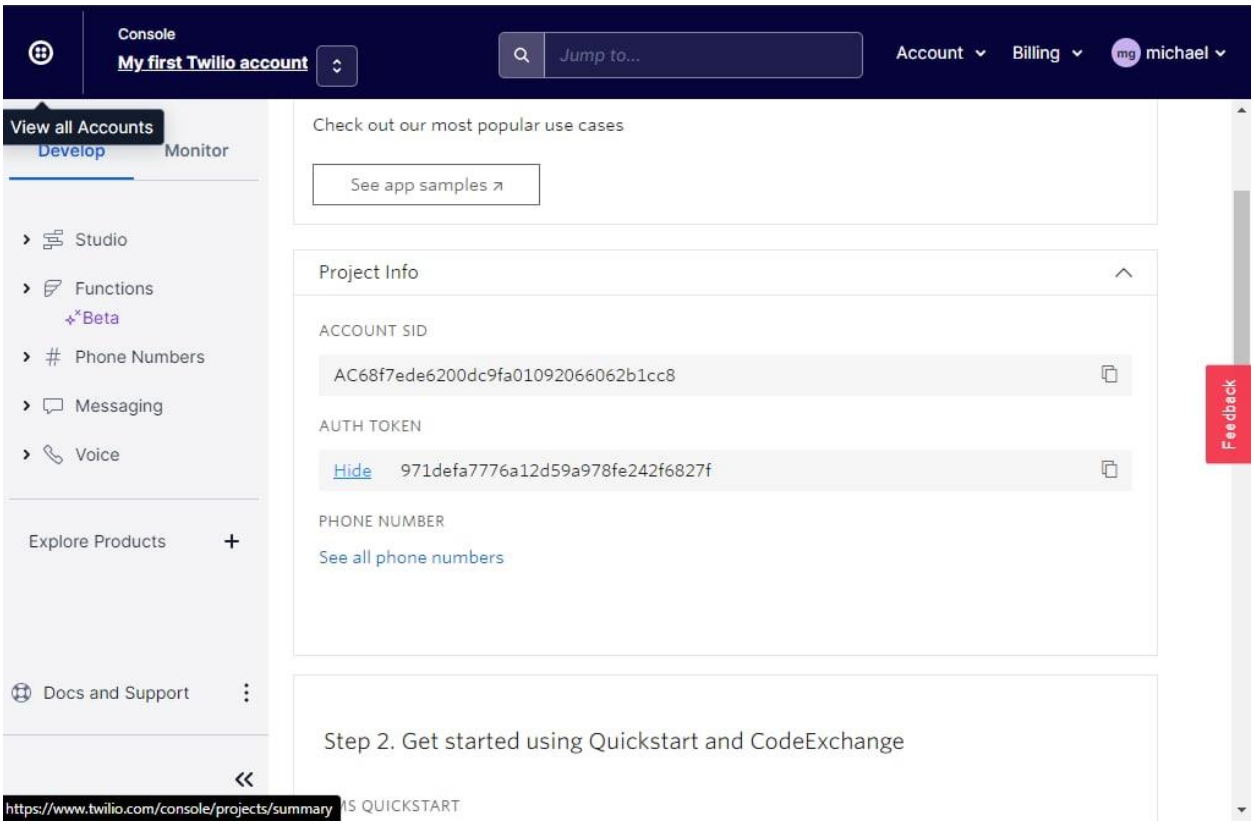


Figure 6.1 Twilio Server Console

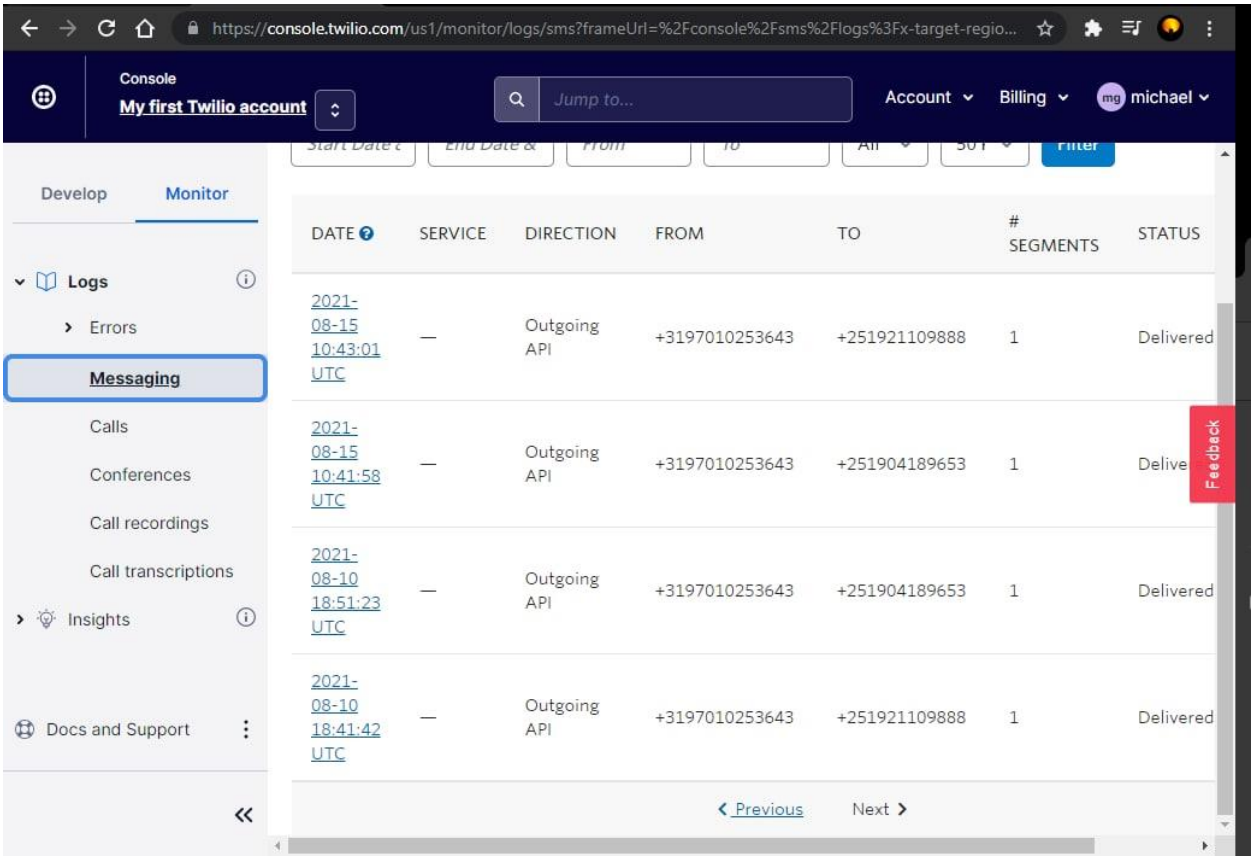


Figure 6-2 Twilio Server SMS log Console

6.4. Configuration of Application Security

We used multi-level Authentication. Multi-level authentication implies different levels of permissions for different users. Each authenticates user levels differently and we set up users as administrators, users and agricultural experts. We have implemented all input validations properly in order to secure our system. When the user invalid inputs or empty, the system notifies the user to enter valid input.

6.5. Implementation of User Interface

The user interface that we have developed is:

- Consistent and stable which does not create any confusion for users.
- Can be easily understandable with clear and steady navigation.
- Consistent and dependable.

- Attractive and eye-catching for users.
- Is compatible, well-matched and friendly

Account Module

For the Account module we customize a predefined Django user model and migrate the database to postgre DB. In order to overwrite the Auth_user mode we have to import Abstract base user and Base user manage. Abstract Base user is for creating new models instead of user models. Base user Abstract is for the Account manager in order to manage and contrat the model. Using Base user Abstract we specify the creation of super user and handling some error messages. Using Abstract Base User: creating the model, specifying default username and required fields are Handles.

View Data

Importing all models from the model and accepting posts and Get requests and responding to them as they request. In view if the method is post registration if the request is registration or login and validating the data and checking requirements if the operation is successful, feature give message if the request is get the view simply retrieve. The template that we specify the render.

Template

Template is the HTML and CSS. The form for registration and login is very elegant and has an image of an Enset that makes the UI look Awesome. For forms we use the form previews' generate a form Development the outlook with HTML and CSS and accepting the data the dynamic part is held by Django forms and pass the values by using post method for views and process the data often that the page will Redirected to login form if the page uses registration if the page is Automatically redirect to the home page if the user is not admin. But if admin the dashboard will appear.

6.6. Testing

6.6.1 Test Case

Feature to be tasted:-

- Input and output functions:-checking what types of our system should take as input and produce expected output.
- Graphical user interface
- Subsystem communication:-the communication between decomposed system moduls should be tasted.
- Database transactions:-ensuring the transactions.
- Security: - only authorized users are allowed to access our system.

Sample test case: Detection

- User ID
- Status
- Date

6.6.2. Testing Tools and Environment

Test case: Detection Form

Test case objective: To identify Ensete leaf

Test case description: - user upload image by using choose image button then image is preprocesses then give result based on its class either infected or healthy.

6.6.3. Unit Testing

As a team we tested each major functionality independently and used any technique of fault finding which helped us to fix any incorrect functionality at the right time.

6.6.4. Integration Testing

In an integration testing we integrate and test all components of the system models that a program is ready for testing purpose.

6.6.5. System Testing

As its name indicates, to test the entire system we use system testing. It helps us to ensure the functional and nonfunctional requirement correctness, performance, accuracy, function validation and security have been met correctly. This testing includes the whole forms, code and modules that we integrate properly. In this phase all functionality, model and error of the system are tested.

6.6.6. Acceptance Testing

It is a test for acceptability in which we ensure that the system is satisfies user need. As unit testing system testing has binary result which is pass or fail. And we test our system by farmer when it is deployed.

6.7. Model Development

Developing tensor flow models that can classify the Ensete leaf is healthy or infected by bacterial wilt disease.

6.7.1. Data visualization

From the data set we have after importing and specifying the path we have to explore the data so we visualize data its rumbling and name above it.

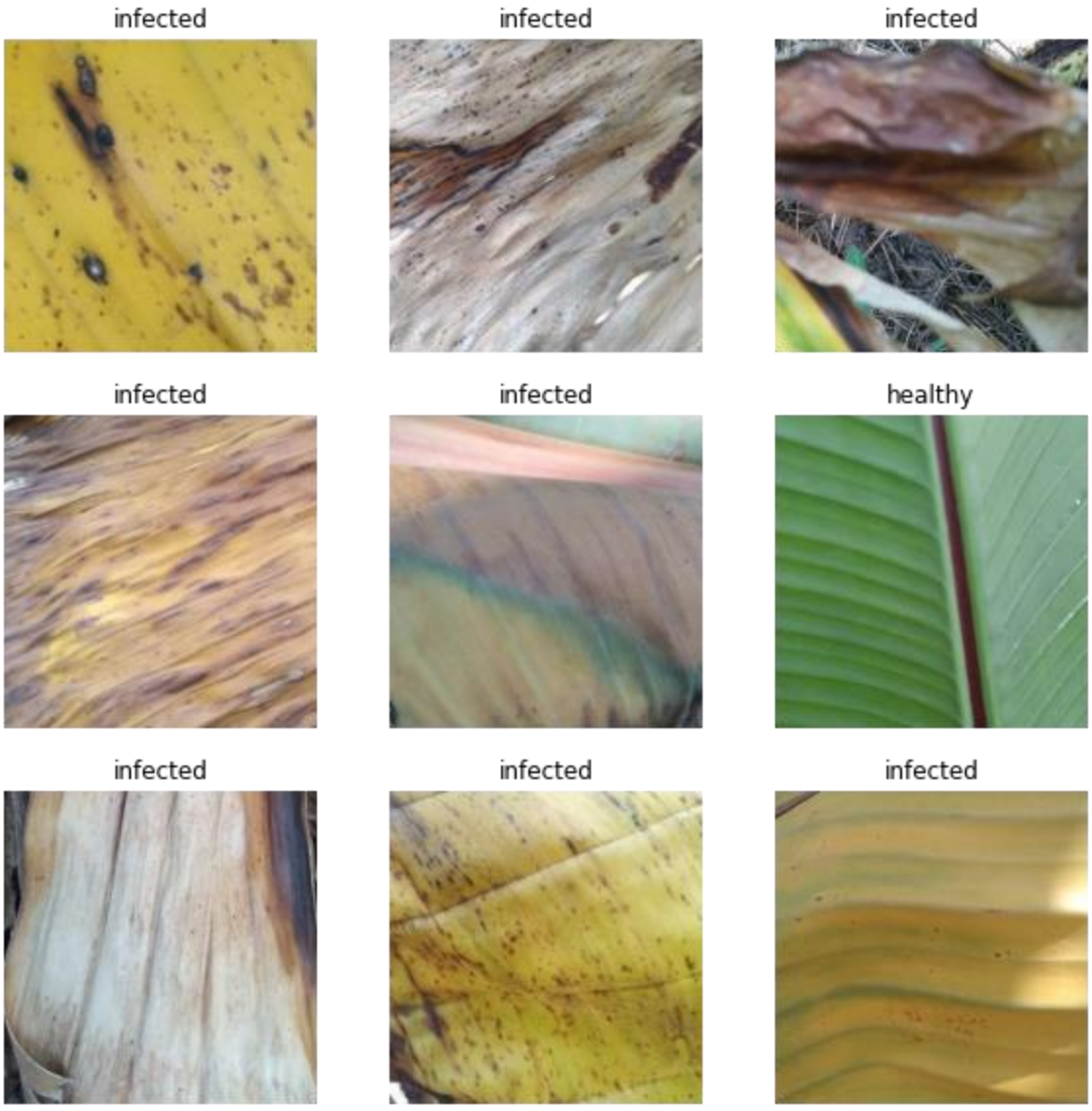


Figure 6-3 Dataset Visualization from Training Set

6.7.2. Convolutional layer

Convolutional layers are the main building blocks used in convolutional neural network. A convolution is a mathematical term that describes a dot product multiplication between two sets of elements. Within deep learning the convolution operation acts on the filters/kernels and image data array within the convolutional layer. Therefore a convolutional layer is simply a layer that houses the convolution operation that occurs between the filters and the images passed through a convolutional neural network.

Convolutional neural networks apply a filter to an input to create a feature map that summarizes the presence of detected features in the input. “Invariance to local translation can be a very useful property if we care more about whether some feature is present than exactly where it is [16]. We use conv2d with the same padding and relu activation conv 2d filter is two- dimensional. In our model definition context we have 8 convolutional layers.

*#Initializing th*e model

```
Model = Sequential()
  # Convolution Step 1

  layer.add(Convolution2D(16, 3, padding='same', activation='relu', input_shape=(256×256, 3))
```

6.7.3. Batch Normalization layer

Batch Normalization is a technique that mitigates the effect of unstable gradients within a neural network through the introduction of an additional layer that performs operations on the inputs from the previous layer. The operations standardize and normalize the input values, after that the input values are transformed through scaling and shifting operations

The RGB channel values are in the [0, 255] range this is not ideal for neural network so the data values must be fall between 0 and 1 this mean grayscale

normalization_layer= layers.experimental.preprocessing.Rescaling (1. /255)

```
normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
image_batch, labels_batch = next(iter(normalized_ds))
first_image = image_batch[0]

print(np.min(first_image), np.max(first_image))
```

```
0.07450981 0.79215693
```

Figure 6-4 normalization of data set

6.7.4. Max Pooling layer

Max pooling is a variant of sub-sampling where the maximum pixel value of pixels that fall within the receptive field of a unit within a sub-sampling layer is taken as the output. Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter above it the max-pooling operation below has a window of 2x2 and slides across the input data, outputting an average of the pixels within the receptive field of the kernel.

```
# Max Pooling Step 1

model.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2), padding = 'valid'))
model.add(BatchNormalization())
```

6.7.5. Flatten layer

Flatten layer Takes an input shape and flattens the input image data into a one-dimensional array.

After the execution of all the convolutional layers we have to flatten in to one- dimensional array in order to create the pooled feature map to a single long feature vector after doing the convolutional layers and max-pooling and dropouts for handling overfit flattening the feature comes next to our tensorflow model of classification.

```
# Flattening Step

model.add(Flatten())
```

6.7.6. Dense Layer

on dense layer each neuron in the dense layer receives input from all neurons of its previous layer this helps the model to lean features from all the combination of the features of the previous layers we have dense layer of with softmax activation and at last we have a dense layer with the class size of the dataset.A dense layer has an embedded number of arbitrary units/neurons within Each neuron is a perceptron.

```
# Full Connection Step
model.add(Dense(units = 2, activation = 'softmax'))
model.summary()
```

6.7.7. Train the network compilation processes

To train the network, we have to compile it. The compilation processes involve specifying the following parameters:

1. Define Network.
2. Compile Network.
3. Fit Network.
4. Evaluate Network.
5. Make Predictions.

Step 1. Define Network

The first step is to define neural network. Neural networks are defined in Keras as a sequence of layers. The container for these layers is the Sequential class. In our case we use a sequential model from the TensorFlow library. The first layer in the network must define the number of inputs to expect. Our model network has 8 hidden layers with max pooling and dropout. The input

layer takes the dataset and changes it to the shape we gave it and transforms its color range from 0-255 to 0-1. After the hidden layers we flatten images from 2d-array to a 1d- array.

Step 2. Compile Network

Once we have defined our network, it must compile. Compilation is an efficiency step. It transforms the simple sequence of layers that we defined into a highly efficient series of matrix transforms in a format intended to be executed depending on how Keras is configured. Whether it's cpu or gpu. Compilation requires a number of parameters to be specified, specifically tailored to training your network. Specifically, the optimization algorithm to use to train the network and the loss function used to evaluate the network that is minimized by the optimization algorithm. In our case we use Adam optimization algorithm that requires the tuning of the learning rate.

Step 3. Fit Network

Once the network is compiled, it can be fit, which means adapt the weights on a training dataset.The network is trained using the backpropagation algorithm and optimized according to the optimization algorithm and loss function specified when compiling the model. The backpropagation algorithm requires that the network be trained for a specified number of epochs or exposures to the training dataset.

Step 4. Evaluate Network

We can evaluate the performance of the network on a separate dataset, unseen during testing. This will provide an estimate of the performance of the network at making predictions for unseen data in the future. The following image shows the training and validation accuracy and loss

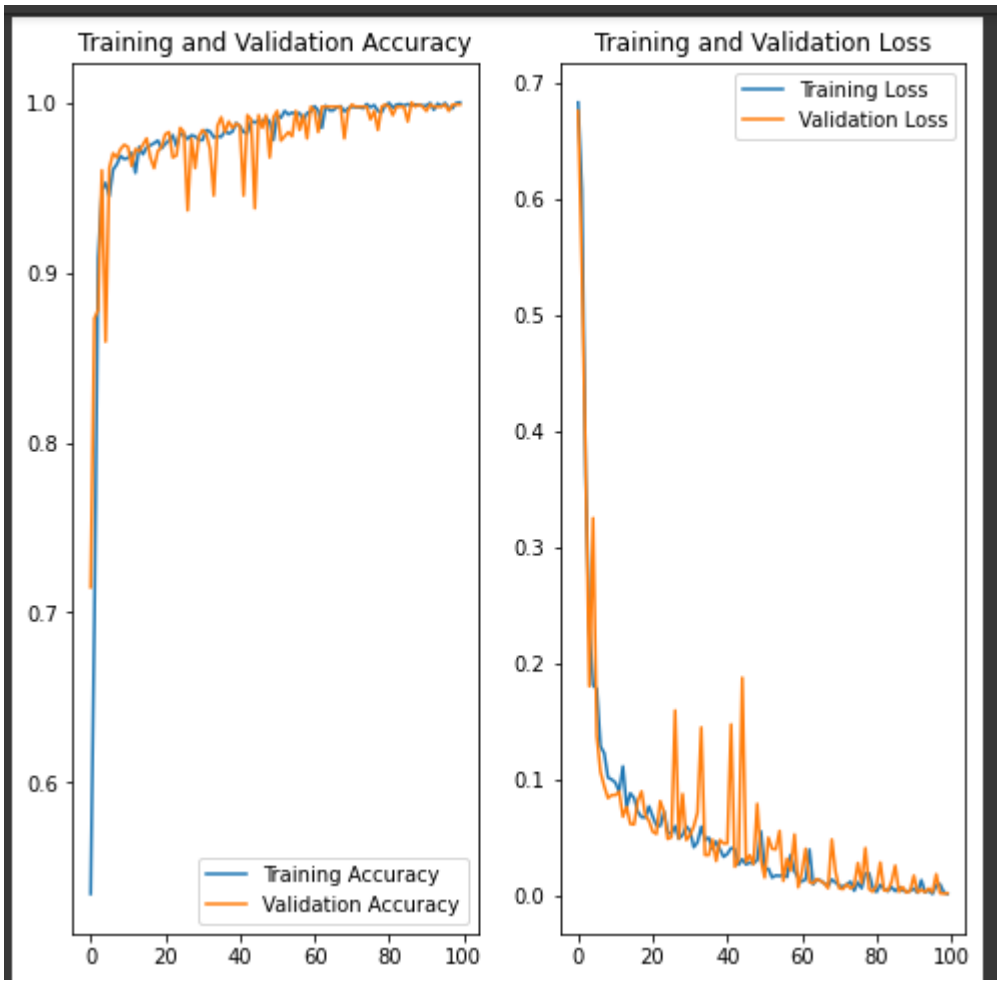


Figure 6-5 graph visualization accuracy and loss of the model

Step 5. Make Predictions

Finally, once we are satisfied with the performance of our fit model, we can use it to make predictions on new data.

Deployment is the method by which we integrate a machine learning model into an existing production environment to make practical identification based on data. It is one of the last stages in the machine learning life cycle and can be one of the most. In order to start using a model for practical predication, it needs to be effectively deployed into production.

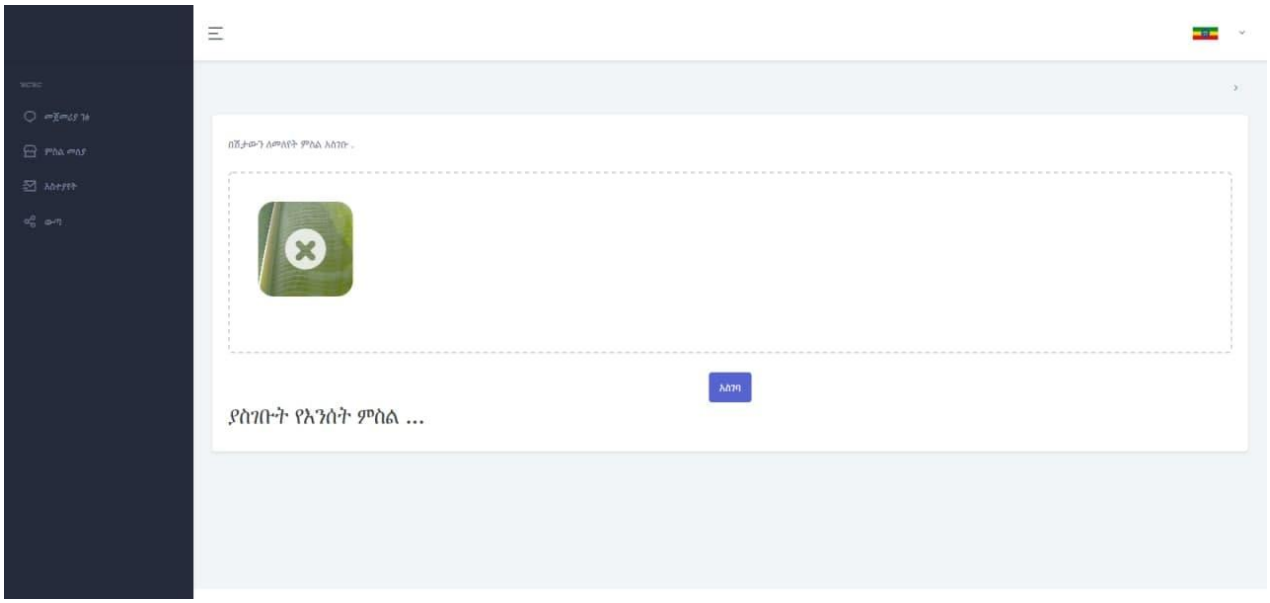


Figure 6-6 user interface uploading data

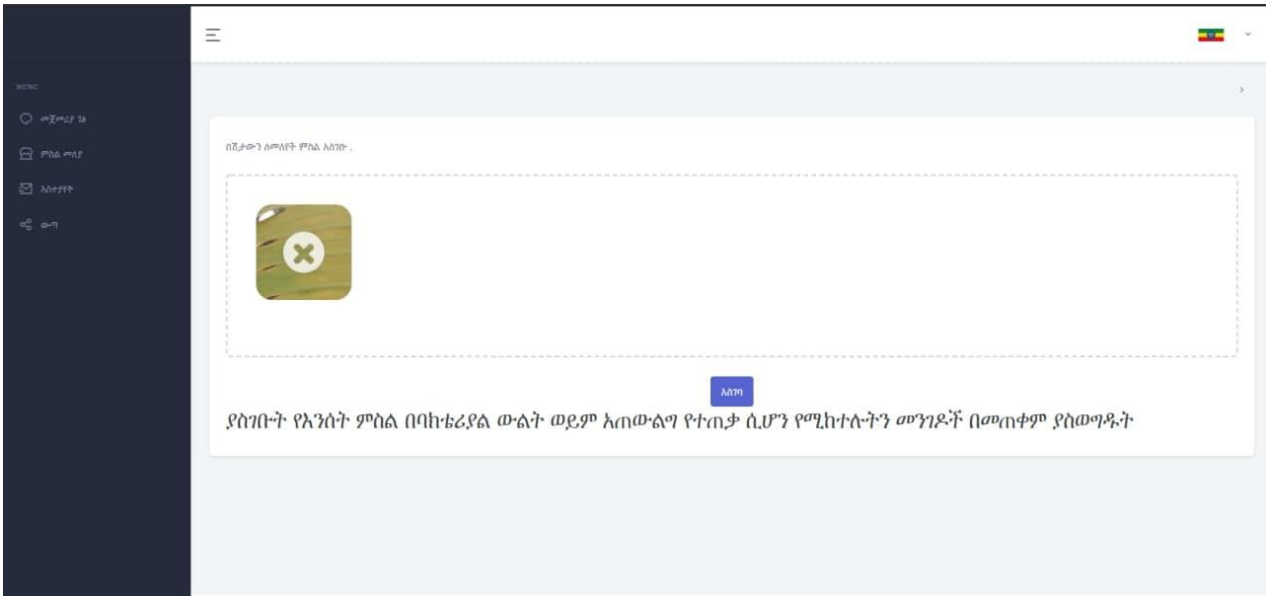


Figure 6-7 web app respond to the user infected leaf

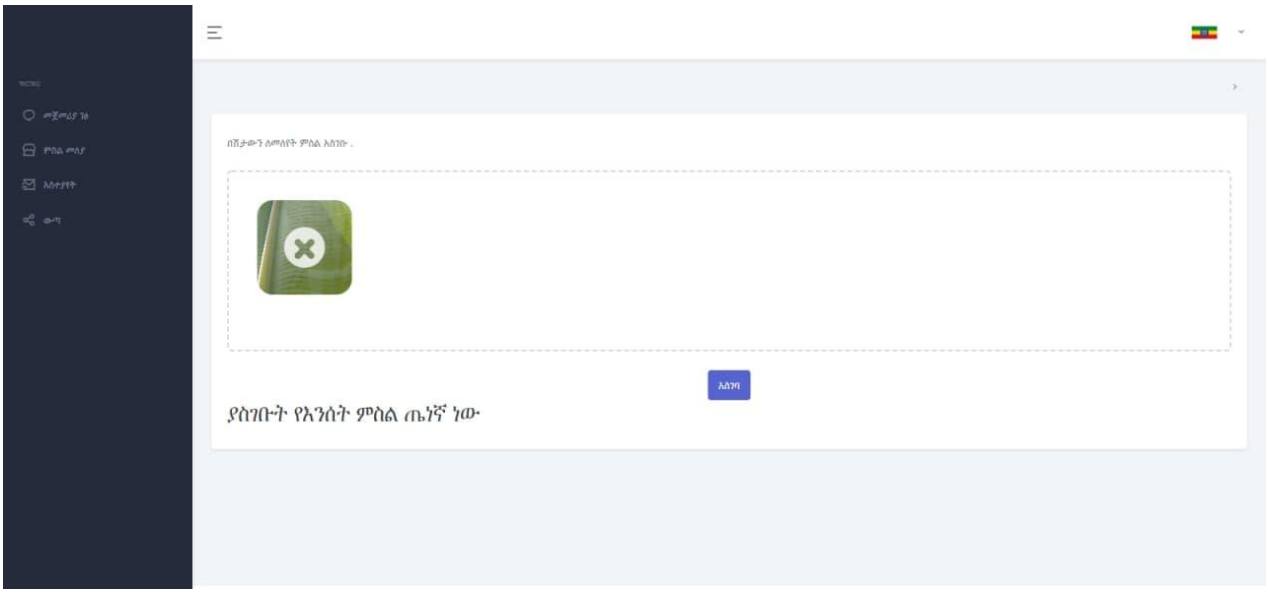


Figure 6-8 web app respond to the user healthy leaf

For the model deployment we used the Django webserver. For the model development we used Google Colab, after we train the data we converted the model to a model.h5 model to create a webapp to identify bacterial wilt disease.

6.7. Activation function

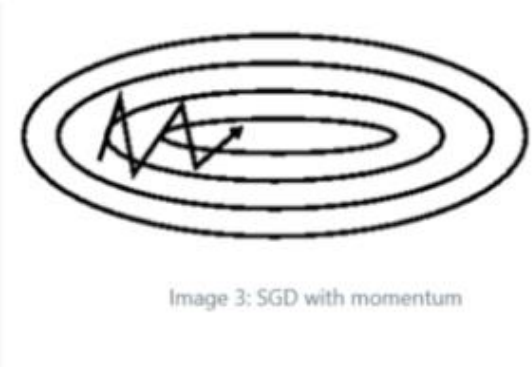
Rectifier Linear Function ReLu activation functions that transform a summed signal from each neuron in a layer can be extracted and added to the Sequential as a layer-like object called Activation.

6.8. Loss function

A method that quantifies „how well“ a machine learning model performs. The quantification is an output (cost) based on a set of inputs, which are referred to as parameter values. The parameter values are used to estimate a prediction, and the „loss“ is the difference between the predictions and the actual values. Since we have infected and healthy its Binary classification so we use binary_crossentropy from tensorflow.keras library.

6.9. Optimization Algorithm

Stochastic Gradient Descent is used to optimize our classification model .Importing SGD from tensorflow.keras implement it with learning rate and momentum. SGD with momentum



accelerates to converge faster and reduce oscillation.

Figure 6-9 Sgd With Momentum

An optimizer within a neural network is an algorithmic implementation that facilitates the process of gradient descent within a neural network by minimizing the loss values provided via the loss function. To reduce the loss, it is paramount the values of the weights within the network are selected appropriately. For this study we use SGD optimization algorithm. SGD is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.

opt = SGD(lr=0.001,momentum=0.9)

6.10. Learning Rate

The amount that the weights are updated during training is referred to as the step size or the learning rate. Learning rate is a configurable hyper parameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.The learning rate controls how quickly the model is adapted to the problem. “The learning rate is perhaps the most important hyper parameter. If you have time to tune only one hyper parameter, tune the learning rate [17]**.** The optimization initial learning rate is 0.001 because if the learning rate is small it can fall to local minimum more precisely.

6.11. Summary () function

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_1 (Rescaling)      (None, 256, 256, 3)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 256, 256, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 128, 16)      2320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 16)        0
_________________________________________________________________
dropout (Dropout)            (None, 64, 64, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 64, 32)        4640
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 64)          0
```

```
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 64)          36928
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 2, 2, 128)         147584
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 1, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 256)               33024
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 514
=================================================================
Total params: 327,058
Trainable params: 327,058
Non-trainable params: 0
```

Total of 327,058 params were found from out tensorflow sequential classification model and all are tainable

6.12. Epoch

The number of epochs is a hyper parameter that defines the number times that the learning algorithm will work through the entire training dataset. 200 epoch is used to train this classification model

6.13. Configure the dataset for performance

Dataset.Cache() keeps the images in memory after they're loaded off disk during the first epoch. This will ensure the dataset does not become a bottleneck while training your model. If your dataset is too large to fit into memory, you can also use this method to create a performant on-disk cache.

6.14. Discussion of the Result

After preprocessing of the dataset, the final dataset consisted of a total of 4006 images. For training and testing the proposed CNN, the dataset was partitioned into two subsets. The training dataset contained 3213 enset leaf images.The testing dataset similarly contained 803 images.As mentioned in the proposed architecture of the CNN model, it consisted of 23 layers in which 1 input layer, 8 are convolutional, 8 max pooling layers, 4 dropout layers, activation function layers, batch normalization layers, 1 flattening layer, and 2 fully connected layers. The CNN model thus achieved an extraordinary performance with an accuracy of 99% with the test data subset used from the processed dataset of this study with a precision of 0.99

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. CONCLUSION

Deep learning methods have recently become popular for computer vision and image identification. Enset has a substantial food item for many millions of people in Ethiopia and has many benefits, as a fiber, animal forage, and medicine for cultural practices but currently enset bacterial wilt affects so many enset leaf. This has different problems in terms of declining of economy growth, reduce food supplies and so many drawbacks arises in community also country. Enset diseases cause significant reduction in both quality and quantity of Enset production. Computer vision is a field that deals with methods for acquiring, processing, analyzing, and understanding images. Deep convolutional neural networks (CNNs) have seen great success in the past few years on a variety of machine learning problems. Deep Convolutional Neural Networks have been applied for end to end learning. In this project we build enset bacterial wilt disease identification and solution suggestion system has a great contribution to announce the disease for those agricultural experts and to guide farmer by give suggestions about how to remove the infected enset and controlling mechanisms systematically ,our system identifies and suggests for disease using machine learning techniques and this system solve communities problem and country also.. A typical CNN architecture consists of several convolutions, pooling, and fully connected layers. This project has been done a deep learning approach for the identification of Enset samples of diseased and healthy. The complete procedure was described, from collecting the images used for training and validation to image preprocessing and augmentation and finally the procedure of training the deep CNN.

The image data was collected from places be considered purposively. The dataset was preprocessed by image resizing,. The model were developed in both original and augmented versions of the Enset dataset. For the model development we used Tensor Flow, Keras, Theano, Matplotlib, Numpy and OpenCV tools. Furthermore, for the experiment the input image size was necessarily resized to 256 x 256 for Deep Convolutional Neural Network (CNN) due to hardware limitations. DL models were trained by using Stochastic Gradient Descent (SGD) a deep learning optimizers. Finally, we develop a Machine Learning System that will identify the healty and disease leaf.

Therefore, the model applies information technology to agricultural production and is favorable to sustainable development of smart agriculture. Some challenges have been encountered in data collection and preparation. We expect that our Machine Learning system will make a significant contribution to the agriculture area.

Generally, we conclude that Enset has a substantial food item for many millions of people in Ethiopia and has many benefits, as a fiber, animal forage, and medicine for cultural practices but currently enset bacterial wilt affects so many enset leaf. This has different problems in terms of declining of economy growth, reduce food supplies and so many drawbacks arises in community also country so our project we build enset bacterial wilt disease identification and solution suggestion system has a great contribution to announce the disease for those agricultural experts and to guide farmer by give suggestions about how to remove the infected enset and controlling mechanisms systematically ,our system identifies and suggests for those disease using machine learning techniques and this system solve communities problem and country also. Generally our system almost achieved those objectives.

7.2 RECOMMENDATION

In the feature the team believe that this system should be expanded and automated with additional functionalities by integrating with many new technologies. The system expanded to an Android system where farmers use mobile applications and also by using sensors they expand more functionality of the system. it will be fully operationally by adding some functionality that is not included in the system.

Apply different types, sizes and styles by effective feature extraction algorithms. The most important quality characteristics of are size, color, shape, and texture.

By deploying these models in mobile environments, both experts and farmers will be able to quickly diagnose the Enset diseases and take the necessary intervention to improve productivity.

References

[1] https://www.farmafrica.org, "https://www.farmafrica.org/ethiopia/ethiopia," 2021. [Online].

[2] G. T. Kibru Abera Ganore, "International Journal of Intelligent Information Systems," *Ethiopian Enset Diseases Diagnosis Model Using Image Processing and Machine Learning Techniques,* p. 5, 17 june 2020.

[3] K. A. YIDNEKACHEW , "DEVELOPING BACTERIAL WILT DETECTION MODEL ON ENSET CROP USING A DEEP LEARNING APPROACH," p. 91, OCTOBER 2019.

[4] L. e. al, Deep Learning, 2015.

[5] G. &. S. salman, 2018.

[6] B. e. al, The Tree Against Hunger, 1997.

[7] j. e. a. Parekh et al, Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides, 2019/2020.

[8] "https://www.sas.com/," 2021. [Online].

[9] "https://u2b.com/," 2021. [Online].

[10] G. T. Kibru Abera Ganore, "http://www.sciencepublishinggroup.com/," 1 nov 2020. [Online].

[11] "https://www.investopedia.com/," 2021. [Online].

[12] "https://master-iesc-angers.com/," [Online].

[13] "https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/," 2020. [Online].

[14] "https://www.medicaldeviceseurope.com/2za92/draw-convolutional-neural-network2beda5," [Online].

[15] E. B.V., "https://www.sciencedirect.com/topics/computer-science/deep-convolutionalneural-networks," 2021. [Online].

[16] "Deep learning," 2016, p. page 342.

[17] "Deep Learning," 2016, p. page 429.

APPENDICES

Appendix I: Sample Source Code

```
img_height,img_width =256,256

classes =['healthy','infected']

model_graph =Graph()

with model_graph.as_default():

        tf_session = Session()

        with tf_session.as_default():

           model = load_model('./models/EBW_MODEL2.h5')
```

Predict image with django

```
def predictImage(request):

    print(request)

    if request.method== 'POST':

   #  print(request.FILES['filePath'])

       fileobj = request.FILES['file']

       fs = FileSystemStorage()

       filepathname=fs.save(fileobj.name,fileobj)

       filepathname =fs.url(filepathname)

       testimage =  '.'+filepathname

       img=tf.keras.preprocessing.image.load_img(testimage,target_size=(img_height,img_width))
```

```
    x=tf.keras.preprocessing.image.img_to_array(img)
    x=np.expand_dims(x, axis=0)
```

Classification model development code

```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers,models
from tensorflow.keras.models import Sequential
sample code for import and count images
image_count = len(list(data_dir.glob('*/*.jpg')))
print(image_count)
```

Test and Train Test

```
batch_size = 32
img_height = 256
img_width = 256
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
```

```
  labels = 'inferred',
  label_mode = "int",
  subset="training",
  # color_mode ='grayscale',
  seed=123,
  image_size=(img_height, img_width),
  shuffle = True,
  batch_size=batch_size)
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  labels = 'inferred',
  label_mode = "int",
  validation_split=0.2,
  subset="validation",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  labels = 'inferred',
  label_mode = "int",
  validation_split=0.2,
```

```
  subset="validation",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

Data visualization

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.title(class_names[labels[i]])
    plt.axis("off")
```

Model defining

```
num_classes = 2
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
```

```
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Conv2D(128, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(128, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
 layers.Dense(256, activation='relu'),  # /fully connect layer
  layers.Dense(num_classes)
])
```

Optimizer

```
opt = SGD(lr=0.001,momentum=0.9)
model.compile(optimizer=opt,
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        # loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
        metrics=['accuracy'])
```