DEEP LEARNING DETECTOR FOR PESTS AND PLANT DISEASE RECOGNITION

BY

ILELADEWA OLUWATIMILEHIN ADEKUNLE

A REPORT

SUBMITTED

TO

Universiti Tunku Abdul

Rahman

in partial fulfillment of the

requirements for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication

Technology (Kampar Campus)

MAY 2020

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: **DEEP LEARNING DETECTOR FOR PLANTS AND PESTS DISEASE RECOGNITION**

Academic Session: **MAY 2020**

I **ILELADEWA OLUWATIMILEHIN ADEKUNLE**

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
231, JALAN 14, TAMAN
PERDANA, 31900 KAMPAR
NEGERI PERAK.

Ng Hui Fuang
Supervisor's name

Date: **10/09/2020**

Date: **10/09/2020**

DEEP LEARNING DETECTOR FOR PESTS AND PLANT DISEASE RECOGNITION

BY

ILELADEWA OLUWATIMILEHIN ADEKUNLE

A REPORT

SUBMITTED

TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology (Kampar Campus)

May 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DEEP LEARNING DETECTOR FOR PESTS AND PLANT DISEASE RECOGNITION**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________

Name : **ILELADEWA OLUWATIMILEHIN ADEKUNLE**

Date : **08/09/2020**

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Ng Hui Fuang and who has given me this bright opportunity to engage in a deep learning based object detection project. It is my first step to establish a career in a deep learning field. A million thanks to you.

Also, I say a big thanks to my parents and my family for their love, unending support and continuous encouragement through the course of my program.

ABSTRACT

Pests and diseases in plants have been a major challenge factor in the agricultural field. And developing a quick and accurate model could help in detecting pests and diseases in plants. Meanwhile, evolution in deep convolutional neural networks for image classification has rapidly improved the accuracy of object detection, classification and system recognition. However, in this project, deep learning techniques are used in developing a model for diseases and pest detection in plants, and then train and test the model before eventually integrating the model into a mobile application. The goal of this project is to develop a framework that can classify the class of a plant, and detect areas of the plant leaf already affected by diseases, and eventually deploy the framework on a mobile application. In order to find a suitable meta-architecture for the aim of the project, we use the combination of Single Shot MultiBox Detector and MobileNet (SSD MobileNet) where Single Shot MultiBox Detector (SSD) is the algorithm that takes a single shot to detect multiple objects within an image, and mobilenet is a neural network for recognition and classification. The system is trained using a large dataset containing different classes of both diseased and healthy images of plants from PlantVillage. Final results of the project reveal that our proposed system can recognize and detect various type of pests and diseases that have been trained in the model, with the ability to handle the complexity of a plant’s surrounding area.

TITLE PAGE **i**
DECLARATION OF ORIGINALITY **iii**
ACKNOWLEDGEMENTS **iv**
ABSTRACT **v**
TABLE OF CONTENTS **vi**
LIST OF FIGURES **vi**
LIST OF TABLES **ix**
LIST OF ABBREVIATIONS **x**

CHAPTER 1 INTRODUCTION **1**
1.1 Project Background **1**
1.2 Problem Statement and Motivation 2
1.3 Project Scope 3
1.4 Project Objectives 4
1.5 Proposed approach 5
1.6 Highlight of what has been achieved 6
1.7 Report Organization 6

CHAPTER 2 LITERATURE REVIEW **7**
2.1 Related Works 7
2.1.1 Deep Convolutional Neural Network (CNN) 8
2.1.2 Image Processing 9
2.1.3 Single-Shot Detector MobileNet (SSD MobileNet) 11
2.2 Data Collection 12
2.3 Strengths and weaknesses of related works 13

CHAPTER 3 SYSTEM DESIGN **14**
3.1 Mobile application of object detection framework 17

CHAPTER 4 IMPLEMENTATION AND EXPERIMENTS **25**

4.1 Implementation 25
4.2 Tools to use 26
4.2.1 Hardware and Software
4.3 Experiments by computing metric calculations of trained model 26
4.4 Implementation issues 29

CHAPTER 5 RESULTS AND DISCUSSION 30

5.1 Image results of Faster RCNN of Project I 30
5.2 Image results of SSD MobileNet in Google Colaboratory of Project II 31
5.3 Results of trained model on Android mobile device 35
5.4 Testing mobile detector accuracy 40

CHAPTER 6 CONCLUSION 44
6.1 Project Review 44
6.2 Problems encountered 44
6.3 Discussions and Conclusions 44
6.4 Potential benefits and impacts 45

REFERENCES 46
POSTER 50
PLAGIARISM CHECK 51
CHECKLIST FOR FYP2 THESIS SUBMISSION 83

LIST OF FIGURES

Figure Number	Title	Page
Figure 1	Country’s most threatened by invasive species to be from sub-Saharan Africa	3
Figure 2	Implementing Single-Shot Detector Network Architecture.	5
Figure 3	Architecture of Deep Convolutional Neural Network (CNN)	9
Figure 4	Input and output images using image processing technique	11
Figure 5	Flowchart of training (Google Colab) program code.	15
Figure 6	System block diagram	16
Figure 7	Flowchart of mobile application	17
Figure 8	Activity diagram of mobile application	18
Figure 9	Screen guide for users	19
Figure 10	Diagnosing a plant disease.	20
Figure 11	Lists details of plant diseases	20
Figure 12	Search feature in application	21
Figure 13	Different features in application	21
Figure 14	Representation of pests and diseases and plants under different conditions	22
Figure 15	Annotated Image. Recognition and localization of plant diseases and pests: Our system aims to detect both class (what) and location (where) of the affected areas in the image	23
Figure 16	LabelImg used for labeling diseased plant images	25
Figure 17	Formula for precision, recall, and f1 scores.	26
Figure 18	Faster RCNN image results from Project I	30
Figure 19	SSD MobileNet image results on Google Colab	35
Figure 20	Mobile detector image results	40
Figure 21	Formula for precision, recall, and f1 scores.	41

LIST OF TABLES

Table Number	Title	Page
Table 1	Metric of plant leaf disease	27
Table 2	Confusion matrix of mobile detector	43

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
Faster R-CNN	Faster Region-Based Convolutional Neural Network
SSD MobileNet	Single-Shot Detector MobileNet
GoogleNet	Google Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory

CHAPTER 1

1.0 INTRODUCTION

1.1 Project Background

In recent years, crops in agriculture are being affected by a wide variety and different species of pests and infectious diseases, causing production and economic losses in the agricultural industry in every part of the world. Crops are usually grown on a large scale commercially for consumption purposes, however several problems which could be as a result of various environmental factors and undetermined changes in climate, and in the atmosphere has really affected the tropical and temperate regions of the world, and could easily attack the crops to a great extent.

Plants are exposed to a wide-range of pests, pathogens and viruses which result in several diseases in various interactions, and occurrences or events. The problem of plant diseases has been a worldwide issue, increasing dramatically in recent years and causing great losses to farmers, and as well threatened food security. The transboundary of plant pests and diseases has become a sudden rise leading to outbreaks and upsurges in different regions of the world, hereby becoming a threat to the livelihood of farmers and the rest of the whole population at a time. A wide research that was conducted on determining the effects and characteristics of diseases on plants in different parts of the world and published by BBC News concluded that the issue of plant pests and diseases has been posing an enormous threat to crops hereby, destabilizing food security globally accounting for about 40% loss and causing many significant losses, till today, indicating that the issue must be treated with special attention and monitored closely. Agricultural sector plays an important role in the economic and social life of several nations. In Ethiopia, a survey conducted on using Image processing approach for identification of plant diseases showed that around 80% to 85% of the people are dependent in agriculture.

A major factor which has been detrimental to humans, crops, and livestock are the harmful and destructive creatures called pests and a these invasive pest species (e.g. phytophthora infestans fungus) have been discovered to cause serious and life-threatening diseases on crop production which result into famines most of the time. So indirectly, it tends to decrease the population by an unreasonably high percentage when consumed because it can lead to deaths during the period. Infestations made on plants by most of these pests result in more than 70% damage, making products unmarketable and inedible. At any one time, because of weather conditions, or emergence of resistance to chemical control, pests can surge into prominence unexpectedly, take control over farm lands, and result into many significant losses. Diseases on the other hand (like coffee wilt disease), cause high losses in the earnings of farmers, and it is believed that such diseases haven't stopped spreading.

In this chapter, we present the background and motivation of our research, with our contributions to the course field, and the outline for an accurate and easier detection of pests and diseases in plants, which will be helpful in the development of a treatment technique while economic losses reduce substantially. Our project goal is to find a deep-learning framework that fits the purpose of our task. We train and test our systems end- to-end on diseased and healthy plant dataset, which contains various images of plants infected by pests and diseases, including several inter-class and extra-class variations, such as infection status and location of infected area in the plant. Testing of our framework on a mobile device through an application that accepts random sample plant images including both diseased and healthy reveals that our system to be developed will be able to recognize effectively different types of plants crops affected by diseases and pests, regardless of the plant's surrounding area because of the ability to deal with complex scenarios.

1.2 Problem Statement and Motivation

As the world becomes more connected, pests species keep invading and disseminating all over into various places, and while these species keep posing as dangerous threats to our biological community of interacting organisms and the physical environment, they arguably pose an even greater threat to our agriculture and food security. Insect pests like silverleaf whitefly, asian gypsy moth, and khapra beetle, are all ranked as major threats and can have negative and far-reaching impacts on agriculture and forest industries in different parts of the world. After estimating the negative impacts of several invasive species on agricultural industries, the most vulnerable countries to crop losses tend to be in some developing nations, with minimal knowledge and investments in crop health management, with African countries in the sub-Saharan areas in particular being the most vulnerable, as shown in Figure 1.

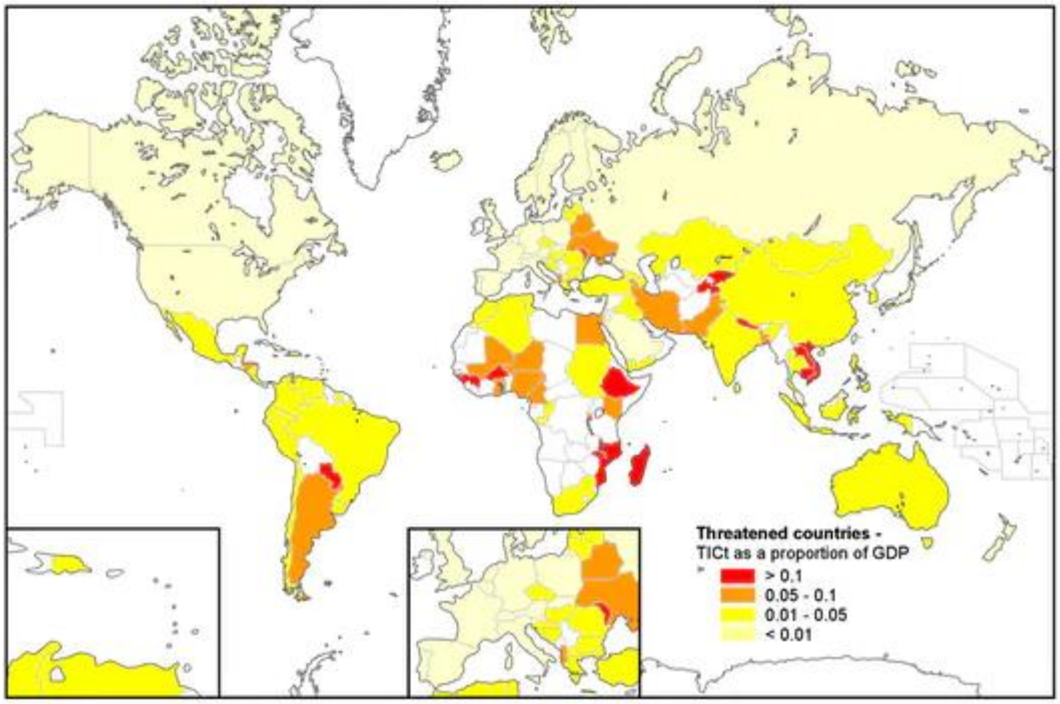


Figure 1. Country's most threatened by invasive species to be from sub-Saharan Africa [1].

In the real world, farmers and other agriculture experts visually carry out inspections on agricultural crops for instance, fruits and vegetables which are likely to be affected by different diseases. However, the process is tedious, time consuming, and does not guarantee an accurate recognition and classification of the plant pests or diseases. By making a global analysis through analyzing thousands of species and several countries, it would be easy to figure out the potential impact of all the species that could invade into a particular community or country, which could as well travel to establish itself in other countries if trading occurs between them. With the aid of this analysis, we can control the emergence of diseases that continue to develop to become uncontrollable hereby, jeopardizing food security.

1.3 Project Scope

Our goal in this project is to devise a method by developing a framework that is able to localize, and detect areas of the plant leaf already affected by diseases. Our system makes use of plant diseases and pest images taken in an intended position and by this means, we avoid using other approaches in some related works like collecting samples and analyzing them in the laboratory. Deep learning has been a breakthrough in image processing, identification, and classification. It can efficiently deal with different light illumination conditions, objects size, background variations, and the surrounding area of the plant. Furthermore, our approach uses input images captured with different camera devices including digital cameras with various resolutions. In addition, after developing the framework, it will be made to be more effective by using smart-phone assisted disease diagnosis system which won't only recognize and spot out the diseases on the infected areas quickly, but will also give hints on how the particular disease can be diagnosed in order to control it from spreading to other parts and areas. This will provide a practical real-time application that can be used in agricultural and several other fields without employing complex technology.

1.4 Project Objectives

Plant pests and diseases are very dangerous and harmful to food crops which could affect a large percentage of people in a particular region, and cause production and economic losses in the agricultural industry in different parts of the world. Besides, due to unsettled climatic and environmental conditions, the outbreaks of plants pests and diseases have become more frequent. Therefore, an early automated system which can detect, diagnose and aid in decreasing huge losses caused by plant diseases must be implemented. One of the aims of our system is to help in preventing the diseases from spreading speedily to other areas. Moreover, developing a plant disease identification and diagnosis system would be beneficial to users who have little knowledge about agricultural pests and diseases, and provides them with hints on how to diagnose and address the problems before they spread to other areas of the plant leaves, and it also benefits those who have limited access to agricultural experts.

With the evolution in computer technology, there has been a deep learning advancement in image plant-based disease and pattern recognition. Firstly, the aim of this project is to develop an effective model for image-based automatic detection, classification, and segmentation of plant diseases. Although training large neural networks could be time-consuming, the trained model will be able to detect and classify images within a fast pace of time, making it more efficient when deployed on mobile devices like smartphones. And secondly, the next aim of this project is to implement the developed framework on a mobile application which will serve as a tool for farmers, and a large percentage of the population, enabling a fast and effective plant disease identification and diagnosis, bringing about an easy decision-making process for controlling the harmful effects of diseases.

1.5 Proposed approach

As food security continues to be a very major concern with the world population growth expected to be more than 9.7 billion by the year 2050. Plant disease has remained a threat to food security, and there has been tremendous efforts in curbing the spread with fertilization, mulch, and good sanitation. Yet, the growing expectations of how agricultural crops should be leaves us with the opportunity of designing some methods that could help in identifying the specific disease so that appropriate measures can be taken to prevent them from spreading, hereby not threatening the lives of people anymore, and getting rid of the problem of economic loss. By developing a fast and accurate model for detecting plant diseases that can be used in mobile devices provided the images, can thus mitigate the issue of food insecurity.

In addition, the mobile application developed in this project will be a valuable tool for farmers, including individuals involved in agriculture, and especially for those living in areas with inadequate infrastructure and limited services for the provision of agronomic advice, enabling them all to be able to control and reduce the rise of life- threatening plant diseases.

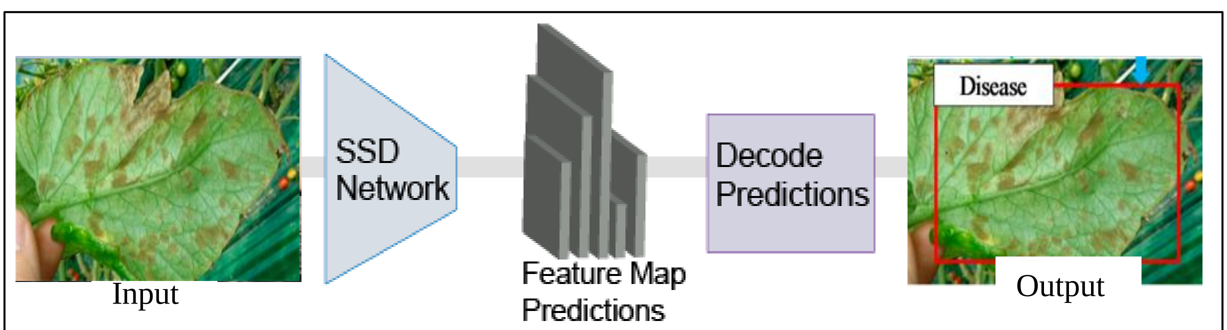


Figure 2. Implementing Single-Shot Detector Network Architecture.

1.6 Highlight of what has been achieved

Since the commencement of the project in my Project I, different architectures were first used in the development of the framework which are the Faster RCNN and SSD MobileNet. The focus of the project has been to develop a framework or trained model that will be able to detect the names diseases of plants given images as the input, and then the model is then integrated into a mobile application with the feature of diagnosing specific plant diseases which can then be used on any mobile device (iOS or Android) by anyone. So before the commencement of Project II, the architecture to be used for training the dataset of all diseased images has been concluded to be the SSD MobileNet, a user interface of the mobile application to be developed has been designed, and the labelling of all plant images (both diseased and healthy) was done.

1.7 Report Organization

The details of this research are shown in the following chapters. In Chapter 2, some related backgrounds and meta-architectures are reviewed. Then, the system design, flow charts, block diagram, and image results are presented in Chapter 3. And then, Chapter 4 and 5 further discusses our methodology, technologies, implementation and experimental results involved during the development of the project. Furthermore, the potential benefits and impacts of implementing our system is defined in the final chapter.

Chapter 2	Related Works: Deep Convolutional Neural Network, Image Processing, SSD MobileNet, Data Collection, Strengths and weaknesses of related works
Chapter 3	System Design: Mobile application of object detection framework
Chapter 4	Implementation: Tools to use, Hardware and Software, Experiments and Implementation issues.
Chapter 5	Image results of Faster RCNN of Project I and Project II, Checking mobile detector accuracy
Chapter 6	Project Review, Discussions and Conclusions, Benefits and impacts

CHAPTER 2

2.0 LITERATURE REVIEW

2.1 Related Works

Plant diseases identification has been an important research topic in the agricultural field, being driven by the demand in producing healthy food. Over the years, several approaches have been adopted and used extensively. Concentration of existing works focused on applying several Deep Learning models and systematic observation, ranging from scientific-studying methods to Convolutional Neural Network (CNN) for image recognition and classification, and deep learning system image classifiers. During the use of technologies to monitor plant health, scientific imaging methods were introduced to calculate the pressure on plants produced mainly by insect attack, increased gases, and radiation. In the determination of plant pathogens on plants, chemical substances were applied on plants and then cultivated to assess their effects on the crops and their abilities to stand against attacks from pests and pathogens. As proposed during the development of a deep learning based detector for real-time tomato plant diseases recognition where the meta-architectures in CNN were used at the beginning with a particular feature extractor (MobileNet, Residual Neural Networks ResNet, etc.), we also apply a feature extractor together with an architecture which will serve as a base model on which our framework will be developed. Although, using scientific study methods show an exceptionally good performance, however, these are conducted using expensive techniques in a laboratory, and do not provide a highly accurate solution for estimating pests and diseases in affected plants in a real-time manner. In our project, one of our main approaches is focusing on a cost-effective way that makes use of datasets and images as our source and main reference.

2.1.1 Deep Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a popular algorithm used for classifying images comprising of a convolution layer which generates a feature map at the end. Extensive research has been conducted on developing and using machine learning and pattern recognition. The techniques involve using various deep learning models and frameworks with other image processing methods for plant disease recognition. These advanced techniques were applied to many crops such as rice, wheat, mango, and maize. Mohanty et al. (2016) introduced a smart-phone assisted disease diagnosis system which employs the method of using image-processing technique. Several researchers who worked along this line adopted the idea of using a deep learning method of creating neural networks to train model frameworks to identify crops species and diseases using images. CNN performs both feature extraction and classification itself. In using deep learning for image-based disease detection article, CNN is used to train the framework to identify 14 crop species and 26 diseases varieties using a dataset of 54,306 images of both healthy and infected leaves of plant crops from Plant Village dataset. Similarly, Sladojevic et al. (2016) used Deep CNN to train their model to distinguish plants from their surroundings achieving an average result of 96.1% accuracy at the end.

Dyrmann et al. (2016) introduced a method of recognizing weeds and plant species making use of colored images, through CNN model. In several plant disease detection, different CNN approaches are usually evaluated by the authors based on neural network like AlexNet and GoogleNet to distinguish several diseases included in 14 different crops using Plant Village Dataset. Due to the advancement in Machine Learning, the results of using CNN model showed a high percentage of accuracy in its results when used in the recognition of different crops. However, some of the leaf images of the dataset used to train the models could have been previously captured with a camera without various resolutions affecting the quality of the images. Thus, this project doesn't only aim at identifying different leaf diseases by using a large dataset of images, but also tries to identify its position and location for the development of a real-time system with various light resolutions. The Convolutional Neural Network (CNN) performs feature extraction from raw inputs provided in a systematic way.

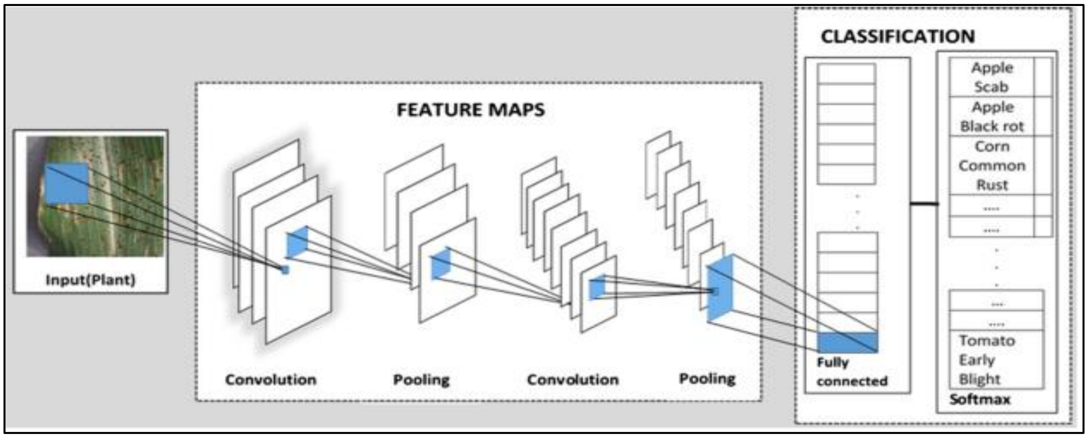


Figure 3. Architecture of Deep Convolutional Neural Network (CNN) [4].

2.1.2 Image Processing

The diagnosis and detection of diseases in plant leaves had always been through a scientific method for quite a number of years, but the evolution of computers has changed our ways of approach using several methods.

Digital image processing is a technique that has been largely employed in the detection of diseases in plants. The detection of diseases in plant leaves has been approached using various methods by many researchers [20], [21], [23], [24]. Dubey et al. (2013) proposed K-mean clustering technique for only the apple fruit, which could not be extended to other various plants, and it involves grouping of the pixels in an image. The approach consisted of feature extraction, classification, and segmentation. Dheeb Al Bashish et al. (2016) introduced using a framework for detecting and classifying plant leaves diseases, employing K-means technique for segmentation.

In performing segmentation using image processing, authors developed various software based on image processing techniques beginning from image acquisition to its classification, so as to automatically detect and classify plant leaf diseases. Masood (2016) used a genetic algorithm which was developed to be efficient in disease detection, and used a cotton leave for experimentation. With the development of computer systems and technology, image processing and computer vision technology has been an automated tool for rapid identification of plant pests and diseases. In a model collaboration for temporal stage classification, a multi-layer classification framework was used to annotate temporal stages for plant images. And to correctly extract disease features, Gray processing image algorithm can be adopted, because the symptoms of different kinds of diseases are different. Equally, in terms of leaves with complicated image background, using image enhancement in advance would be necessary. Camargo and Smith, (2009) introduced a machine vision system for identifying the visual symptoms of plant diseases from colored images using support-vector-machine (SVM). Barbedo (2013), conducted a survey on using various methods that employ digital image processing techniques according to detection, and classification of diseases of plant leaves from digital images in the visible spectrum.

According to the various research methods used, implementing image processing technology to identify crop diseases has several advantages compared to basic traditional methods in terms of bringing direct and instant results. However, the methods still lack accuracy in result in certain cases, and only a few plant diseases were covered. Plant diseases have become a critical issue that affect the leaves, stems, and also parts of the roots. In this project, a number of training samples of plant diseases will be tested and used, because disease symptoms vary from one plant to another and the diseases need to be predicted more accurately.

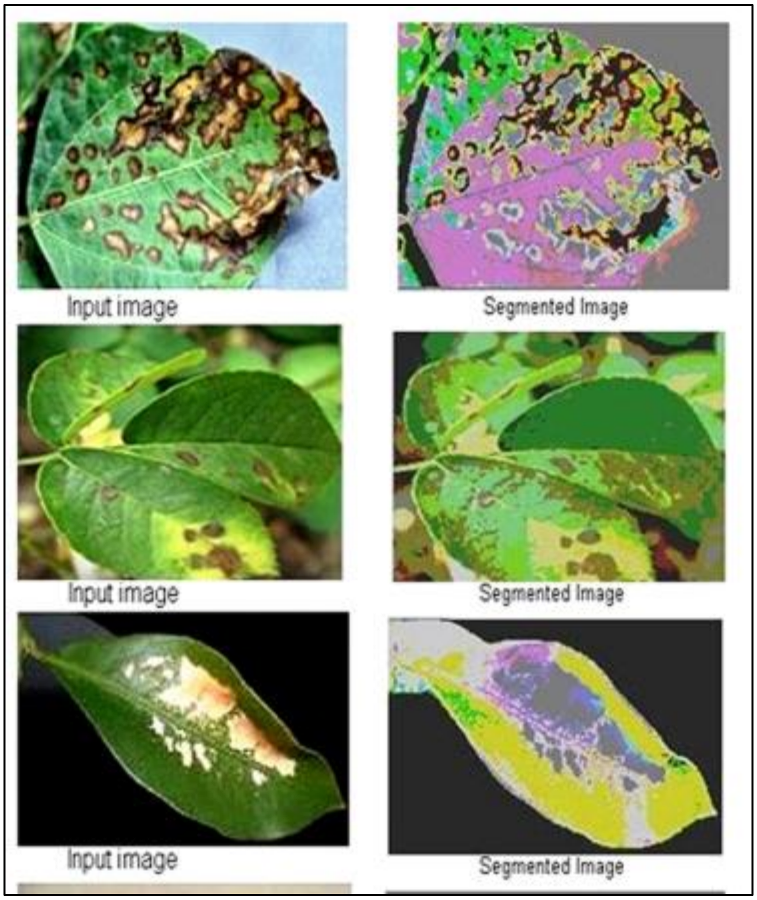


Figure 4. Input and output images using image processing technique [13].

2.1.3 Single-Shot Detector MobileNet (SSD MobileNet)

SSD MobileNet an architecture implemented in this project. SSD uses one single-shot to detect several objects within an image, while the MobileNet is used for feature extraction. Although, SSD uses bounding boxes similar to Faster R-CNN and the implementation of both architectures are almost the same, but SSD runs efficiently on mobile devices with the aid of TensorFlowLite unlike Faster R-CNN which we used in the first part of our project. The bounding boxes have different sizes and aspect ratios depending on the size of object to be detected. Examples are human being (bounding box can be a rectangle), computer mouse (bounding box can be a square), different plant leaf sizes, and so on. Each bounding box has its own class score or confidence score of how likely the detected object is the correct actual object. SSD MobileNet is also considered generally faster than Faster RCNN because models trained using SSD with MobileNet feature extractor has higher mean average precision values (mAP) in other words higher accuracy than when it is trained using regional proposal network (RPN) based approaches and an example of RPN based approach is Faster RCNN.

Nevertheless, the training using SSD MobileNet produces results which appear to be very accurate using our trained detector. Therefore, we use SSD MobileNet in our project due to its high accuracy, and its compatibility with mobile devices.

2.2 Data Collection

Deep learning models are evaluated and trained making use of a large set of images. In this project, we train our deep learning framework using images of plant leaves with the aim of classifying and identifying plant disease on different kinds of images that the model has not seen before. The dataset contains images with several diseases in many different plants. And for this detector, we consider some of the commercial crops, cereal crops, and vegetable crops and fruit plants such as apple, maize, potato, tomato, rice, and so on. Diseased leaves and healthy leaves of various food classes were collected including the aforementioned crops from various datasets. Datasets from PlantVillage (Hughes and Salathe, 2015) were used for this study. PlantVillage dataset has 54,306 images, with 26 diseases for 14 crop plants. The images are originally colored images of varied sizes. From the continuation of the first part of our project that focused only on grape crop plants, now we will be concentrating on all the crop diseases except Blueberry, Orange, and Soybean because only their healthy images are in the dataset. Each image is fixed at a dimension of 256×256. To begin the training of the detector, 80% of the images will be used for training while 20% will be used for testing the detector.

2.3 Strengths and weaknesses of related works

Deep learning is a research area that has been used extensively with various meta-architectures like Residual Network (ResNet), Densely Connected Convolutional Network (DenseNet), GoogleNet, etc., by several researchers in this field, which includes using the architectures for plant species classification.

In (Mohanty et al, 2016), the architecture used is AlexNet and GoogleNet. They trained both healthy and diseased plant leaves using a convolutional neural network, achieving a high percentage of 99.35% accuracy on identifying 14 crop species and 26 diseases through images. However, their results were poor when it was tested on several images under various light conditions, even though the result of the highest level of development was generated.

Dyrmann et al. (2016) in their research work, introduced a method for plant species and weed recognition using a total of 10,413 colored images with 22 weeds and crop species, achieving an overall result of 86.2% accuracy. But, there were some problems encountered during the classification of the plant species, which was caused because of the low number of training samples used for training the species.

CHAPTER 3

3.0 SYSTEM DESIGN

Using deep learning approach in our project, we must beware of drawing incorrect conclusions. To ensure high accuracy in the learning process, the model has to go through a lot of training which requires high-performance computer systems. But to make out system to be re-build easily and understandable even to people outside this field, we make use of more cloud and internet resources like Google Drive, Google Colab, and GitHub.

Firstly, we create a Google colaboratory (Colab) notebook in our Google Drive account to be easily accessible from any system. The colaboratory notebook layout is quite similar to jupyter notebook, where we write the python codes needed for training the model, and for more accuracy we use the external cloud GPU provided by Google Colab. During the development and beginning of the project, the training of the detector was slowing down the performance of the computer system and in order to reduce the workload, we make use of cloud resources like Google Colab to make our training faster and easier but due to very large training files and Google Colab's policy that the notebook used for training will be automatically stopped at some time interval (8 hours in my case), I mounted my google drive account to the Google Colab notebook so that all files generated during training will be automatically stored in my google drive, so anytime I had to restart the runtime of the notebook, I can always continue from where the training previously stopped.

The programming language used for writing the codes for training is Python and the colab notebook of our project which can be found at https://colab.research.google.com/github/CodeT1m/agrik_obj_detect/blob/master/AgrikAI.ipynb contains the coding with detailed explanation of how it works and runs.

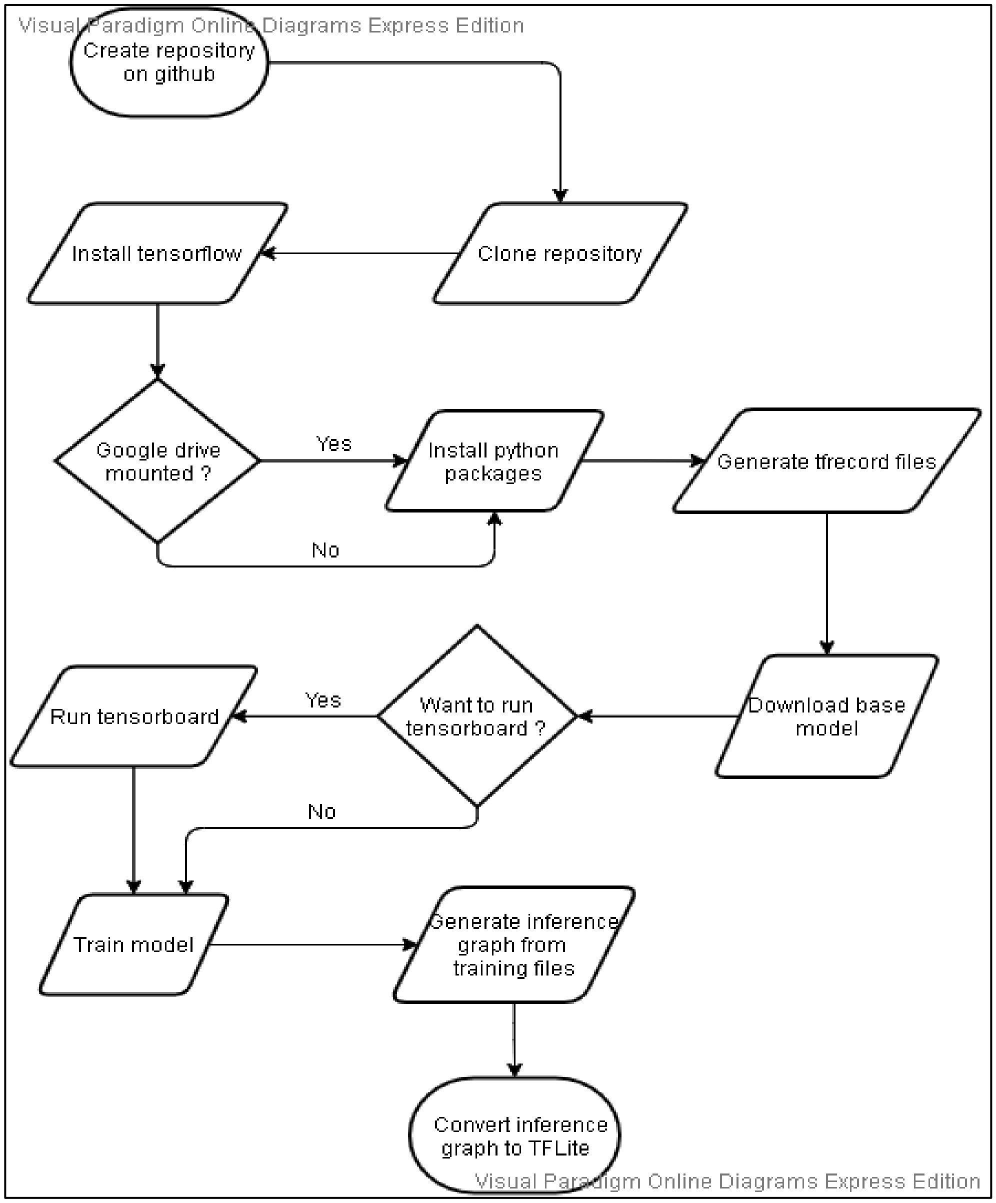


Figure 5. Flowchart of training (Google Colab) program code.

First, we create a repository of out project on github in order to clone the project easily to the Google Colab notebook, this way we do not make much use of the computer system memory. Then we install tensorflow in the notebook and also enable GPU even though it's optional, but we make use of it to better our training model. After that we can choose to mount our google drive or not but mounting it will enable all the large training files to be stored automatically in our google drive. The repository created can be found at https://github.com/CodeT1m/agrik_obj_detect.

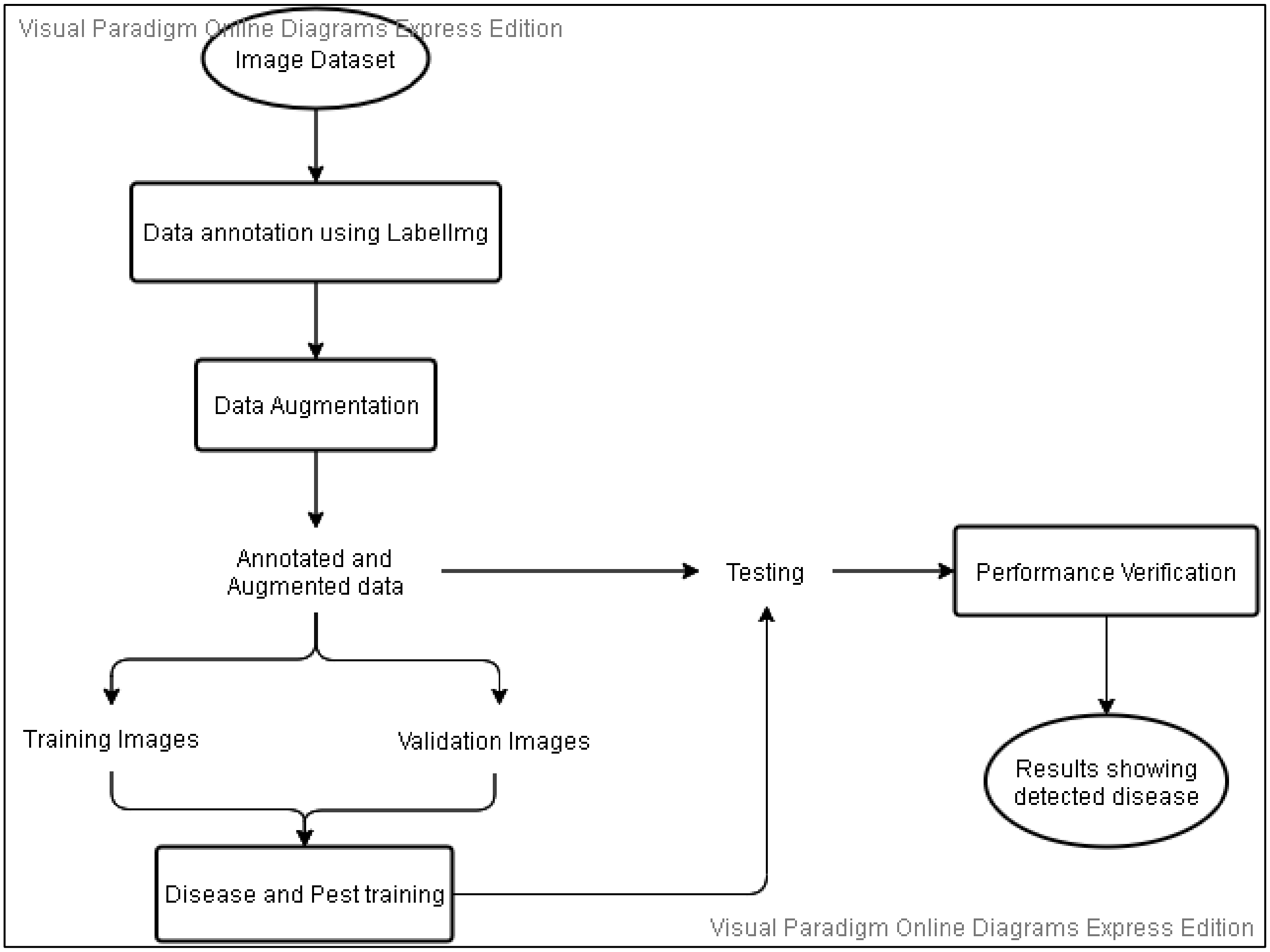


Figure 6. System block diagram

Firstly, diseased and healthy plant images are gotten from PlantVillage dataset, and are labelled with an annotation tool 'LabelImg' generation XML annotation files. Data Augmentation is then performed which increases the diversity of the available data images without actually collecting new data.The XML files are converted to CSV files with a python command in TensorFlow and training and validation is then performed on the images using Faster R-CNN and SSD. An after training, the developed models are deployed into a mobile device. Due to the very long names of the diseases, we use acronyms during the labelling to make it clearer which are the first letters of the names of each disease.

3.1 Mobile application of object detection framework

After training, the image results can be seen on bottom page of Colab notebook but to fulfill our task of integrating the model into a mobile application, we need to convert the model into a Tensorflow Lite (TFLite) which will be added into our mobile application for the model to work on a mobile device. So, we code the application in Flutter framework using Dart language making our application to be able to work on both Android and iOS devices. Section 4.2 lists and describes the software we used in developing the mobile application.

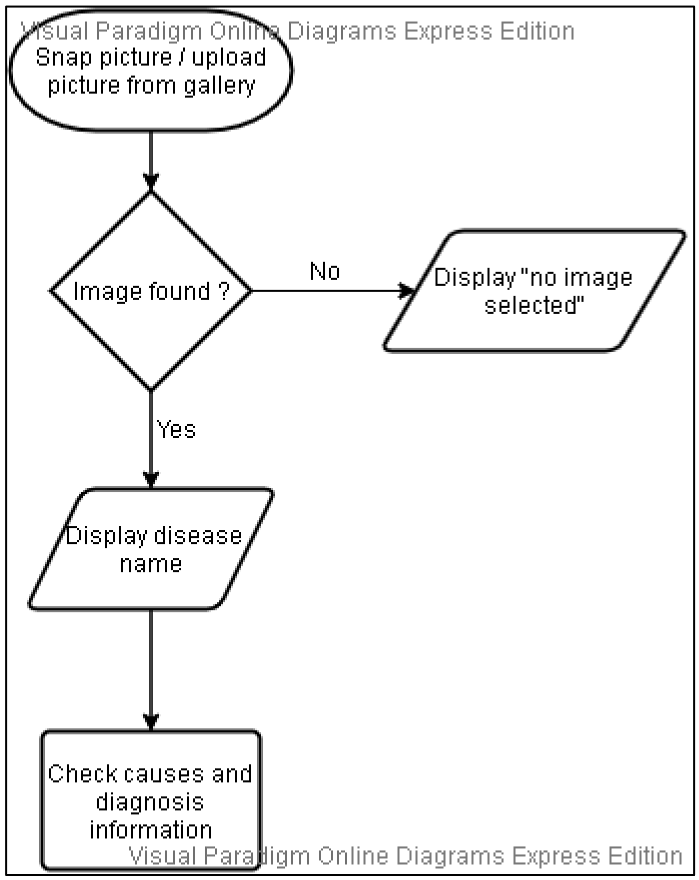


Figure 7. Flowchart of mobile application

When the mobile application is opened on a device, first the user is prompted to either snap the picture of a diseased plant or upload it from the device gallery. When an image has been snapped or uploaded, the model immediately starts processing the image for the prediction of the disease of the plant showing both the name acronym and confidence score. After the formed acronym of the name of the plant disease is shown on a button, the user can now click on the button to see more information about the disease like the causes and how to control, with a link also provided for user to check for more information online.

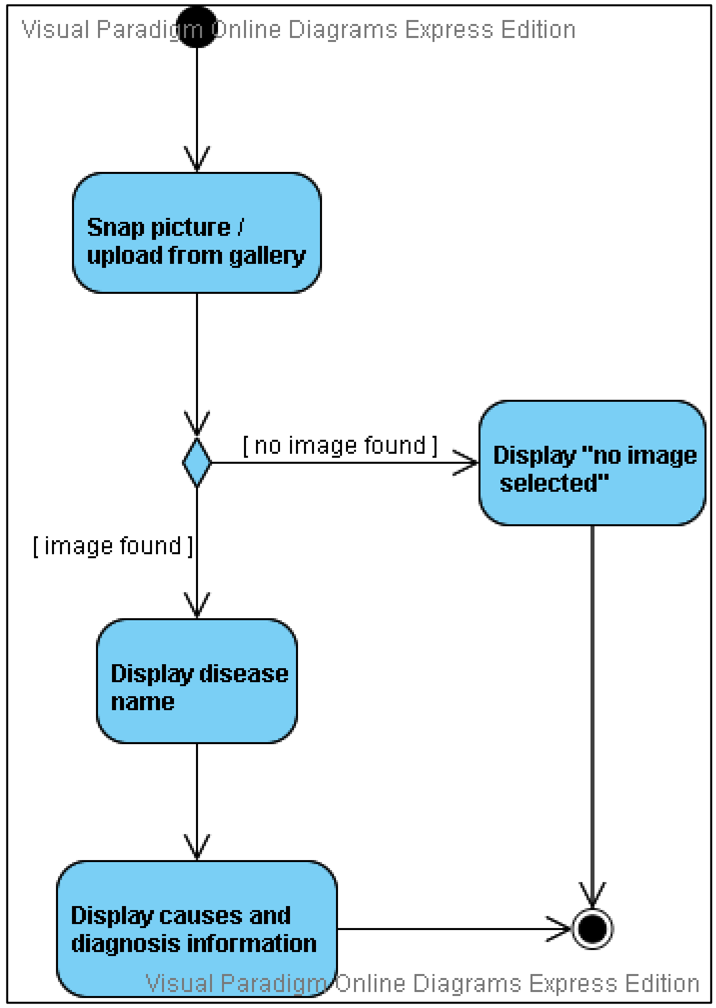


Figure 8. Activity diagram of mobile application

To convert and integrate the model into a working mobile application, we use Flutter framework and dart language for coding the application in android studio and visual studio code software.

Understanding that for first-time users of the application, the flow might not be clear. So to make it easier, a 3 screen guide is created to explain the main feature of how the application works. The images are shown in Figure 9.

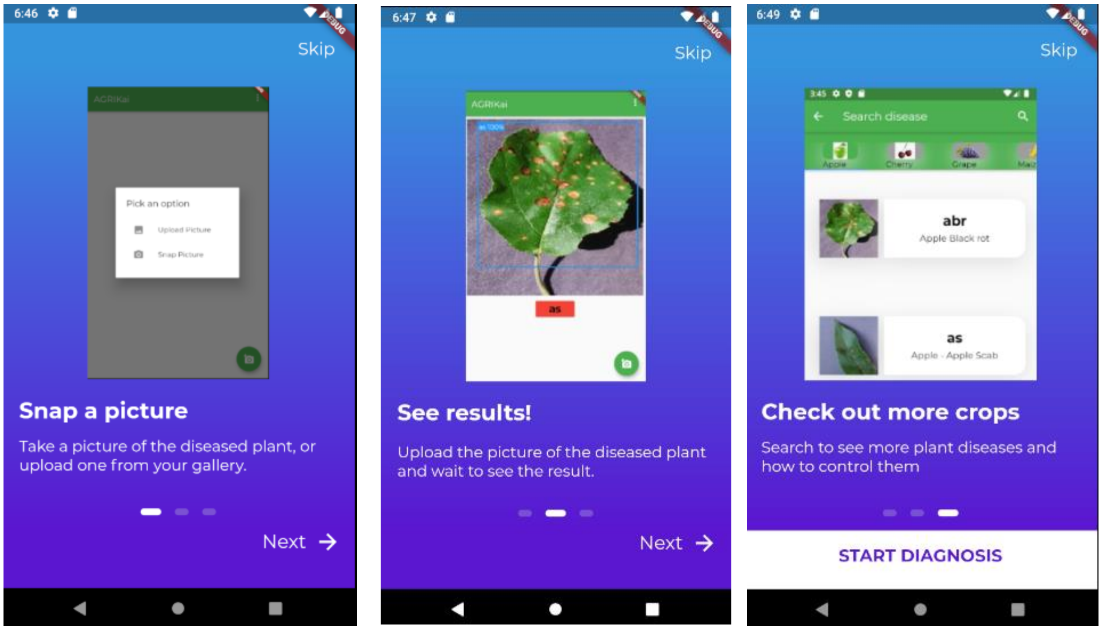


Figure 9. Screen guide for users

The results of a diseased crop plant is shown above with the shortened name in other words acronym, and confidence value of the likely disease of the parts of the plant. And clicking on the red button will display the second image showing more details, and the “Disease” text in the third image navigates to another page that shows all plant food crops that can be detected as show in images in Figure 10.

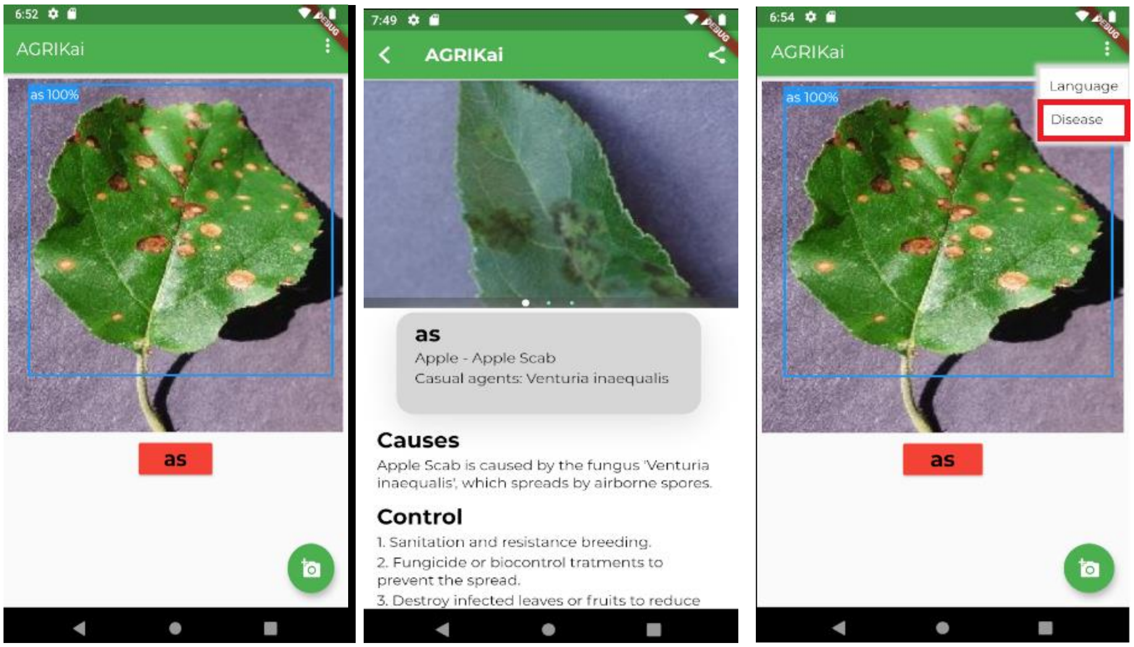


Figure 10. Diagnosing a plant disease.

The below interfaces displays tabs of the 14 fruit crops that the framework in the application can detect, with each tab showing the list, images, and names of each plant disease. Clicking on any plant disease will show extra information about the particular disease with a link provided at the bottom page to see extra information online (this particular screenshot is shown and highlighted in red below).

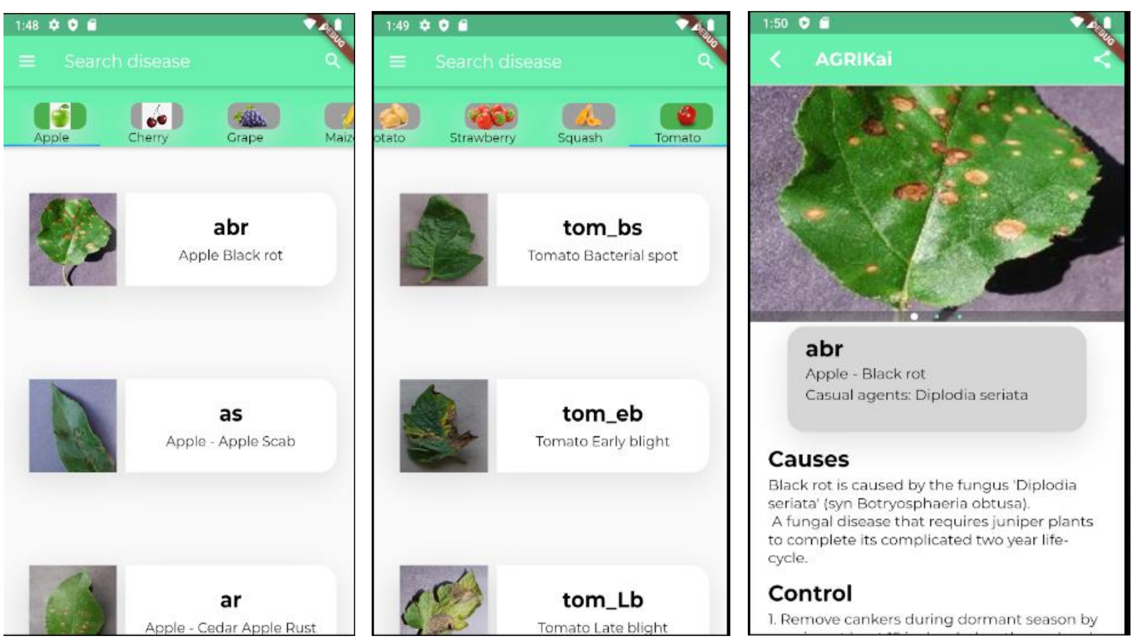


Figure 11. List details of all plant diseases.

The application also has a search function to search for any plant disease quickly. And each page that shows information about a particular disease has a button that when pressed by user, will show more information about the disease in the browser.

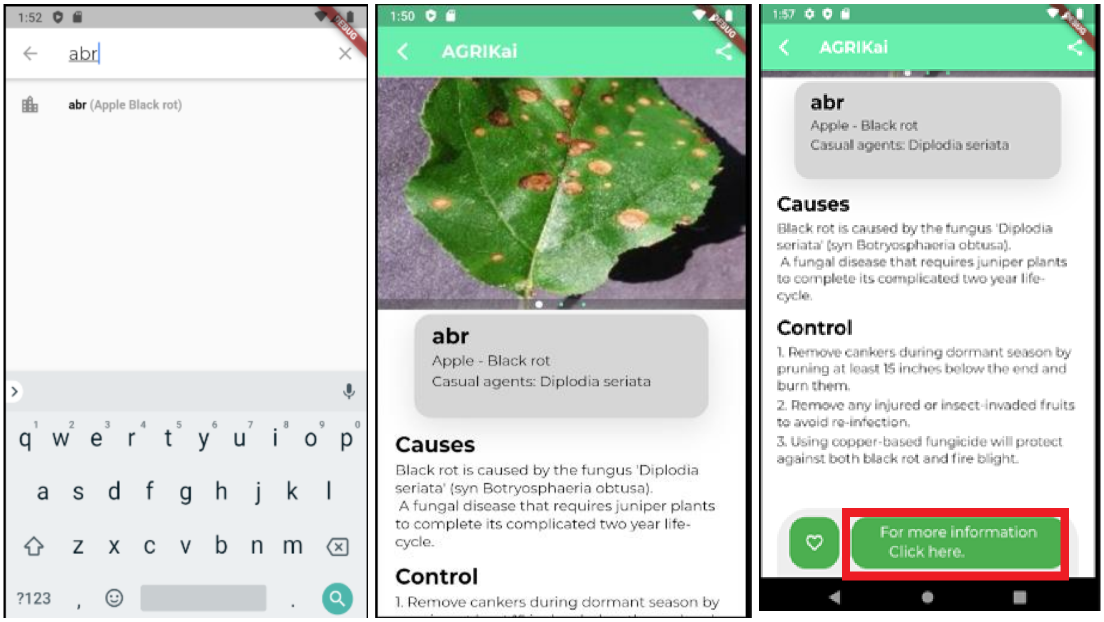


Figure 12. Search feature in application.

For the application to reach a wider audience, the application has a feature of supporting multiple languages which won't only be useful to local farmers but also other individuals in a society. And currently, the application supports English, Malay language, and Chinese Mandarin.

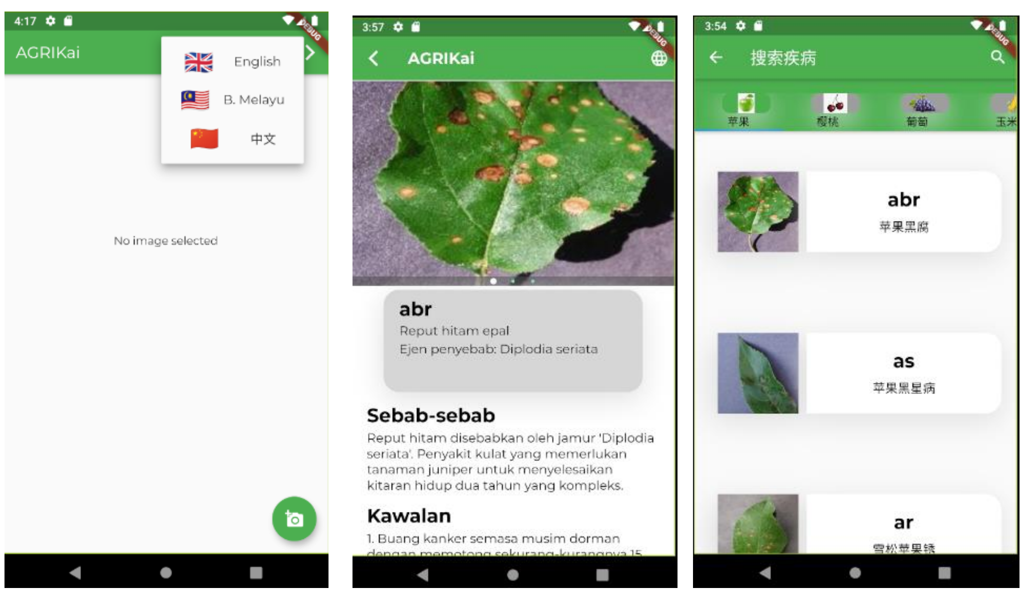


Figure 13. Different language feature in application.

CHAPTER 4

4.0 IMPLEMENTATION AND EXPERIMENTS

Plant crops are easily attacked and affected by pests and diseases which are caused by several reasons like environmental conditions, pests that feed on different plants also spread the diseases from plant to plant, and engaging in commercial trade with other countries will also make the diseases spread to other parts of the world.

Since the aim of our project is to control the spread with a function of diagnosing the diseases, we will be using a lot of plant images (both infected and healthy plants) to train a model with a deep learning architecture (SSD MobileNet), which will study the patterns of the leaf either on the front side or back side because most of the plants show different colors and shapes at different stage of infection. And at the end of the training, the model will be able to detect the location of the diseased parts of any plant given the plant image as input, and to make it more efficient, it will be deployed into a mobile application that can be used on Android and iOS devices.

To achieve a better result, we use Google Colaboratory with Google Drive mounted on it because of the large training files, and to avoid having to restart the training every 8 hours due to Google Colab policy that notebooks used for training will be stopped automatically after 8 – 12 hours. Section 4.2 also describes the tools we used in developing our model and mobile application.

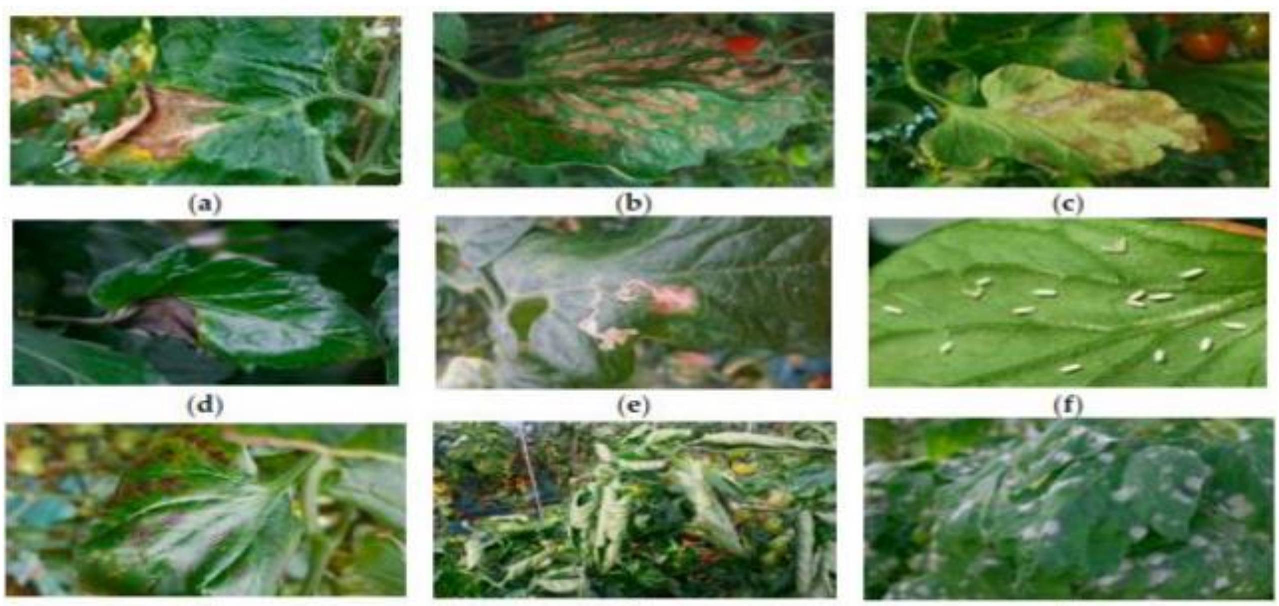


Figure 14. Representation of pests and diseases and plants under different conditions [8].

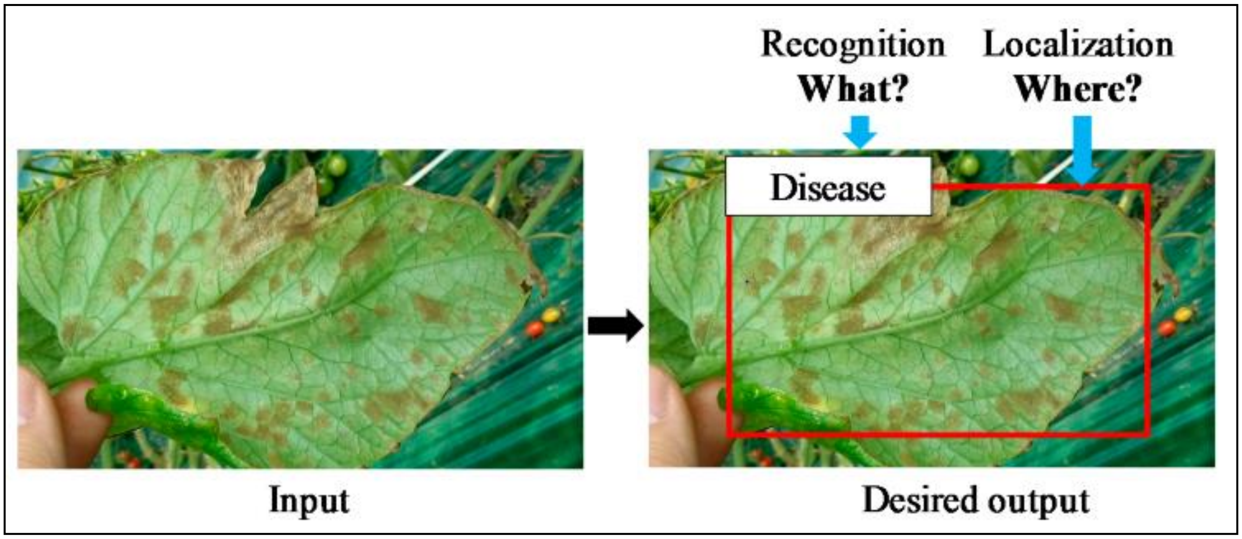


Figure 15. Annotated Image. Recognition and localization of plant diseases and pests: Our system aims to detect both class (what) and location (where) of the affected areas in the image [28].

4.1 Implementation

The plant disease detector is built upon the Tensorflow framework and integrated into a mobile application. The SSD MobileNet model is trained and fine-tuned using the plant disease images from the dataset. The training is executed on Google Colab that comes with a free GPU. During the training, many of the training files happen to be large and are being saved automatically to google drive which occupied 22GB space in total, and after the training of the model, it is then converted into a TFLite which will be integrated in the mobile application that will be able to run on Android and iOS devices. The detector model cannot be trained using just images, but we need to also provide annotation of the location and size of the diseased parts of a plant in each image. Although the Plant Village dataset does not provide the annotation files, we use LabelImg which is an image annotation software tool used to generate XML files of each corresponding image which will be fed into the model for training. An example shown below in Fig. 6. During preparation for training of the model, a total of 23,436 images are used with 18,738 of the images used for the training, and 4,698 of the images used for testing the plant disease detector. A plant leaf is considered to be infected when the detector model detects one or more spots in the image, with the detector indicating them with bounding boxes and name of disease with confidence value.

4.2 Tools to use

4.2.1 Hardware and Software

The training is carried out on Google Colaboratory (Google Colab) on the cloud in order to use the free external GPU provided to make out detector more accurate. And the specifications of the computer used has a memory of 4GB, with a Windows 10 operating system. In order to reduce workload on computer system, all training files were saved to cloud on Google Drive which occupied a total space of 22GB.

Python: Python is a high-level programming language considered very comfortable for Machine Learning and Deep Learning programming, and due to its robustness it is the main language used for our project.

TensorFlow: Tensorflow allows users perform various functions by providing powerful libraries including an API which can be used for object detection on images and videos.

LabelImg: LabelImg is an image annotation tool for labelling objects in images by drawing a bouding box round the object. The annotations are saved as XML files which will be used for training by the architecture.

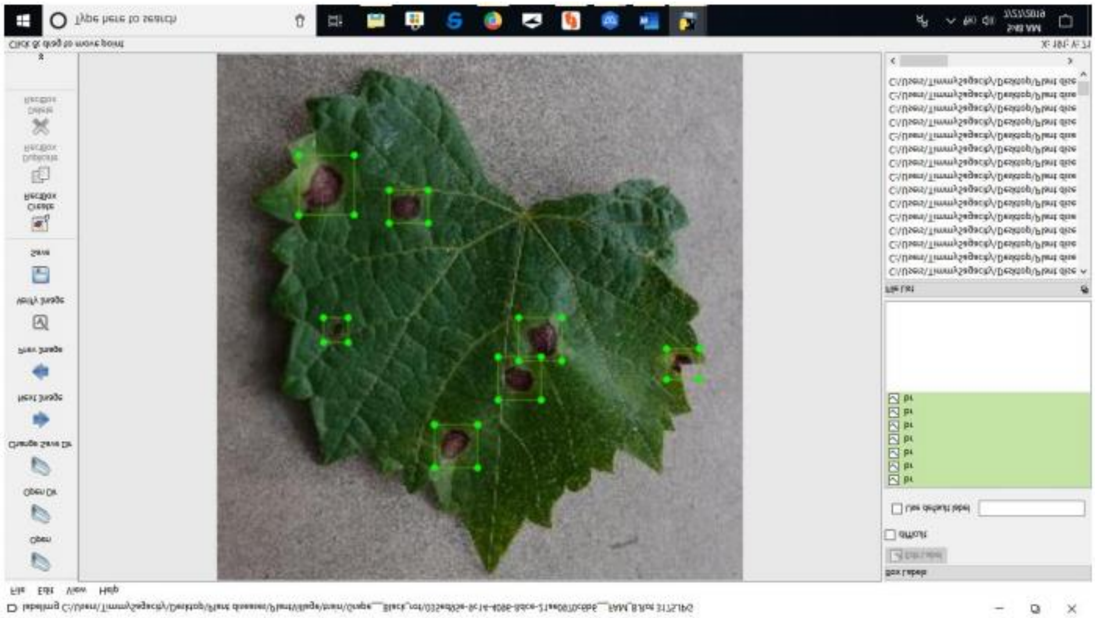


Figure 16. LabelImg used for labeling diseased plant images

Android Studio: an integrated development environment for developing the mobile application for the project but it takes a lot of RAM making other opened applications in the computer system to start working very slowly.
Visual Studio Code: Similar to Android Studio but can be used with many frameworks and also code in multiple programming languages. We also make use of this software because it does not occupy much (Random access memory) RAM space and does not interfere with the performance of other opened applications on the system.

4.3 Experiments by computing metric calculations of trained model

Due to the large testing data containing 4,698 images, the classification metrics are generated first by running an *infer_detections.py* script in the inference folder from tensorflow framework which runs through our trained model and then saves the results in a detection record file. Then we run a *confusion_matrix.py* script that computes the precision scores, recall scores, and f1 scores using the formular shown below in Figure 17, using python commands. The results show there is high and consistent precision and recall rates among the diseases and this a desirable behavior for a detector.

TP (True Positive) – number of positive sample images that are correctly predicted as positive

TN (True Negative) – number of positive sample images that are falsely predicted as negative

FP (False Positive) – number of negative sample images that are correctly predicted as negative

FN (False Negative) - number of negative sample images that are falsely predicted as positive

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned}$$

Figure 17. Formula for precision, recall, and f1 scores.

	Category	**precision**	**recall**	**F1 score**
Apple	Abr (Apple black rot)	0.910714286	0.871794872	0.890829694
	Ah (Apple healthy)	0.994152047	0.965909091	0.979827089
	Ar (Cedar apple rust)	0.962962963	0.753623188	0.845528455
	As (Apple scab)	0.885350318	0.684729064	0.772222222
Cherry	Ch (Cherry healthy)	0.945	0.73828125	0.828947368
	Cpm (Cherry powdery mildew)	0.943089431	0.628726287	0.754471545
Grape	g_Lb (Grape leaf blight)	0.953608247	0.728346457	0.825892857
	g_br (Grape black rot)	0.933333333	0.562043796	0.701594533
	g_ebm (Grape esca black measles)	0.960199005	0.923444976	0.941463415
	grape_h (healthy)	1	0.985507246	0.99270073
Corn (maize)	mLb (Maize leaf blight)	0.878787879	0.572368421	0.693227092
	mLs (Maize leaf spot)	0.696202532	0.709677419	0.702875399
	Mcr (Maize common rust)	0.868512111	0.581018519	0.696255201

	Mh (Maize healthy)	0.974137931	0.8828125	0.926229508
Peach	Pbs (Peach bacterial spot)	0.87529976	0.677179963	0.763598326
	peach_h (Peach healthy)	0.888888889	0.727272727	0.8
Potato	po_Lb (Potato late blight)	0.890710383	0.848958333	0.869333333
	po_eb (Potato early blight)	0.90530303	0.788778878	0.84303351
	potato_h (Potato healthy)	0.891891892	0.75	0.814814815
Pepper	pp_bs (Pepper bacterial spot)	0.93220339	0.44534413	0.602739726
	pp_h (Pepper healthy)	0.95398773	0.840540541	0.893678161
Raspberry	rasp_h (Raspberry healthy)	0.948717949	0.787234043	0.860465116
Strawberry	sberry_Ls (Strawberry leaf scorch)	0.98630137	0.894409938	0.938110749
	sberry_h (Strawberry healthy)	1	1	1
Squash	squ_pm (Squash powdery mildew)	0.878594249	0.650118203	0.747282609
Tomato	t_sLs (Tomato septoria leaf spot)	0.868852459	0.532663317	0.660436137
	t_smt (Tomato spider mite)	0	0.7517	0.83794
	t_tmv (Tomato mosaic virus)	0.855555556	0.77	0.810526316
	t_ts (Tomato target spot)	0.923076923	0.529968454	0.673346693
	t_ty (Tomato yellow leaf virus)	0.917874396	0.733590734	0.815450644
	tom_Lb (Tomato late blight)	0.919463087	0.631336406	0.74863388
	tom_Lm (Tomato late mold)	0.946428571	0.75177305	0.837944664
	tom_bs (Tomato bacterial spot)	0.931914894	0.734899329	0.821763602
	tom_eb (Tomato early blight)	0.833333333	0.621890547	0.712250712
	tom_h (Tomato healthy)	0.939716312	0.892255892	0.91537133

Table 1. Metrics of plant leaf disease

4.4 Implementation Issues

During the development and beginning of the project, using Google Colab appeared to make our training faster and easier but due to Google Colab's policy that the notebook used for training will be automatically stopped at some time interval (8 hours in my case), this made all the files generated during the training to be automatically deleted when it is restarted after it stops after 8 hours. This made us begin training afresh every day for around 2 weeks and store the training files manually on the computer and some of these files were very large and took many hours to download. In order to overcome this difficulty, I mounted my google drive account to the Google Colab notebook so that all files generated during training will be automatically stored in my google drive, so anytime I had to restart the notebook, I can always continue from where the training previously stopped.

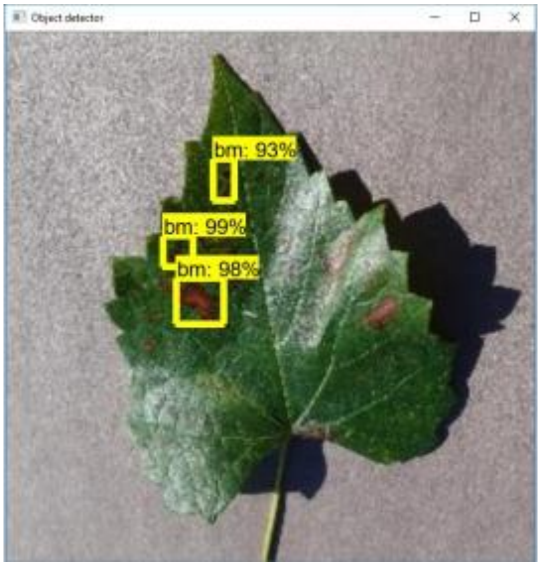
CHAPTER 5

5.0 RESULTS AND DISCUSSION

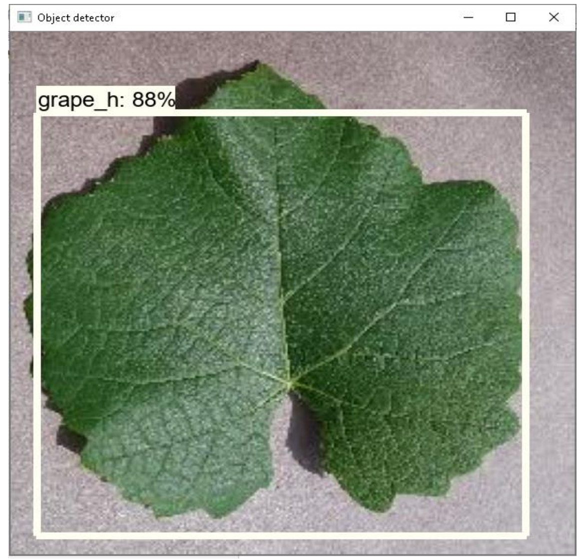
Over the weeks, on the commencement and progress towards the goal of this project and to be certain of the accuracy of the models for the project, including the compatibility with mobile devices after being put into a mobile application. We use the testing data images from PlantVillage consisting of 14 species, each with a sub-class of different disease. For the first part of the project, the grape plant crop with three different diseases was trained for detection using Faster R-CNN and SSD MobileNet. And healthy images of the Grape leaves were also trained to make the model detect the name of the plant disease, location of the disease, and also specify if a particular leaf is healthy or not. Due to the available computer used for Project I, Faster R-CNN took approximately a week to train all the images, while SSD MobileNet took more than 2 weeks and results gotten from the first part of the training shows the results of Faster R-CNN had a high accuracy, and meanwhile SSD MobileNet was also able to predict the diseases and classify the leaves correctly and quickly, the percentage wasn't as high as Faster R-CNN. And now in the second part of the project, much improvements have been made as all 14 crop plant images consisting of infected and healthy images are used to train the detector model using SSD MobileNet. After the training was completed, we first checked the results of the trained model on Google Colab to determine if it requires more training or not, before we exported the detector to a tflite model which is integrated into a mobile application workable on Android and iOS devices which can detect name of plant disease, location of the disease, and specify if it is healthy or not given the input image. With the aim that at the end of the project, a plant disease detector and diagnosis system has been developed which will can localize, and detect areas of the plant leaf already affected by diseases, with a feature of giving causes and how to control each specific disease or how it can be diagnosed to prevent them from spreading.

5.1 Image Results from Faster RCCN model of Project I

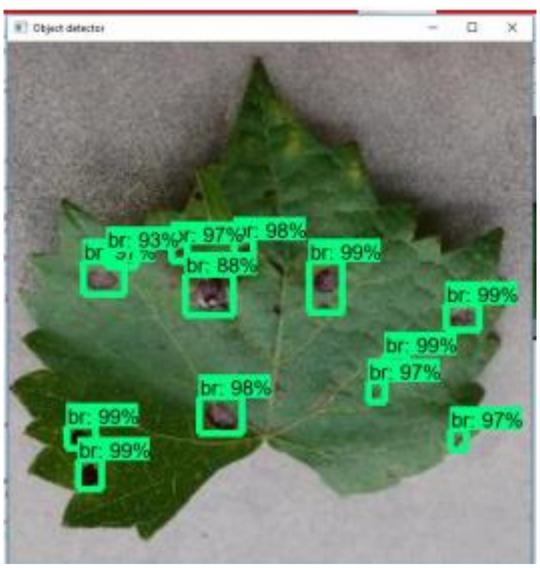
Result images with the disease name acronyms and confidence scores are gotten when we run the notebook used for training our model.



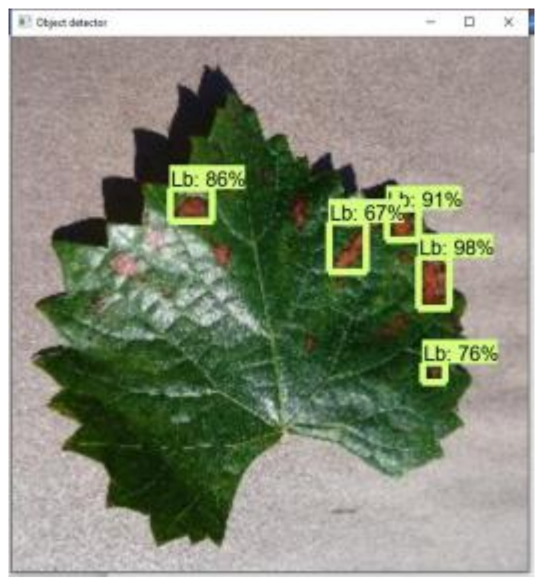
Grape Black measles



Grape healthy



Grape black rot

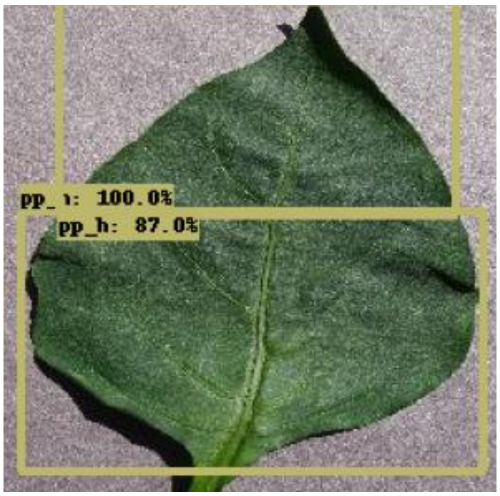


Grape leaf blight

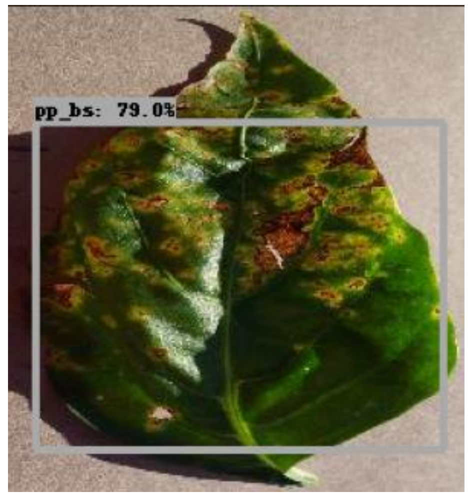
Figure 18. Faster RCNN image results from Project.

5.2 Results of trained model on Google Colaboratory of Project II

Below are some of the image results of the trained model on google colab before integrating it into a mobile application. The disease name acronyms and confidence scores are gotten when we run the notebook used for training our model found at https://colab.research.google.com/github/CodeT1m/agrik_obj_detect/blob/master/AgrikAI.ipynb



Pepper healthy

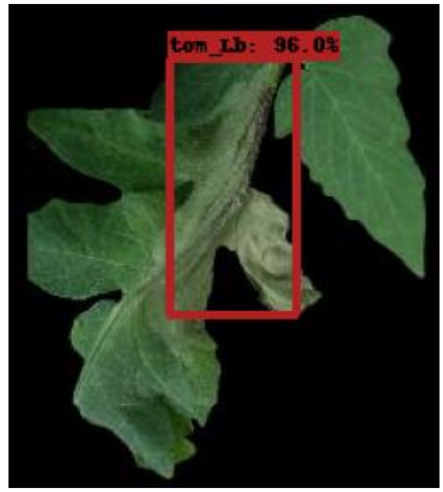


Pepper bacterial spot

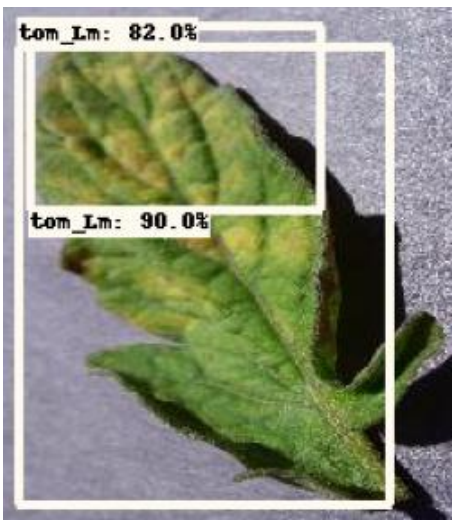
Potato Late blight

Strawberry healthy

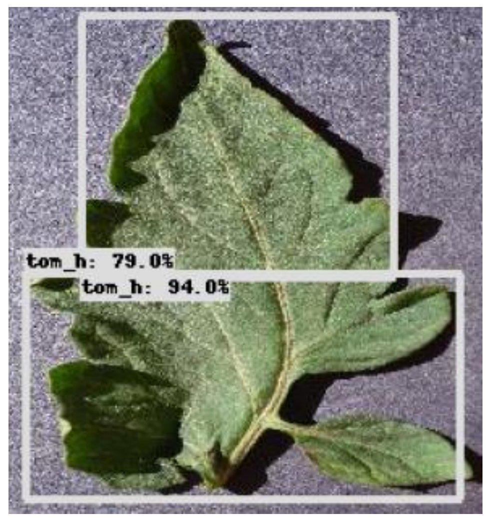
Strawberry leaf scorch



Tomato Late blight

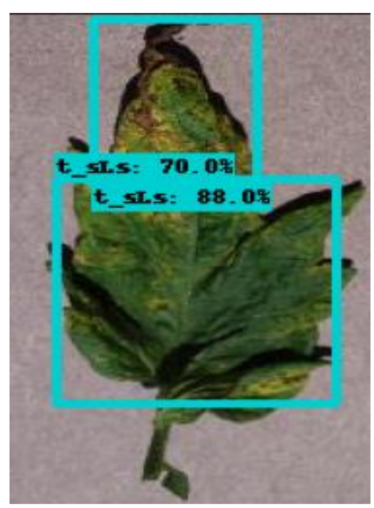


Tomato Late mold



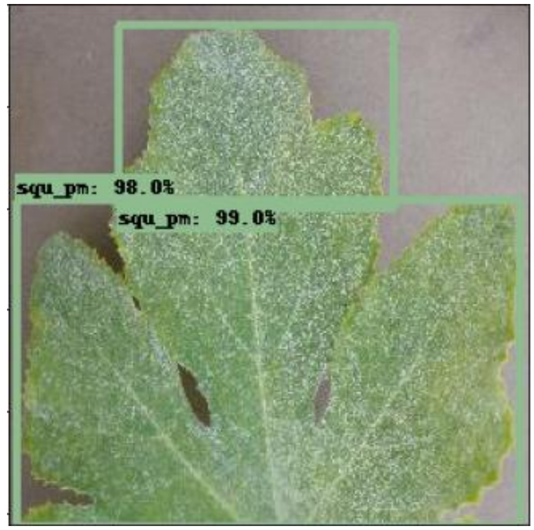
Tomato healthy

Tomato Early blight



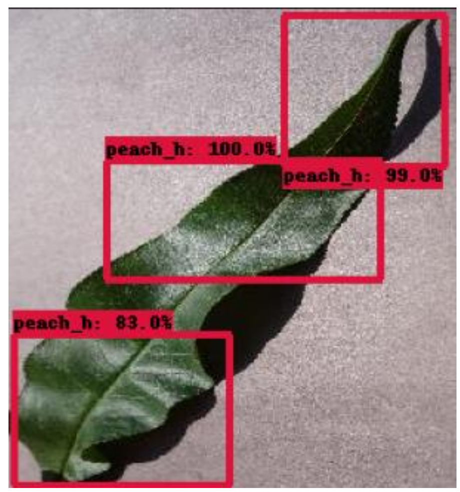
Tomato septoria leaf spot

Raspberry healthy

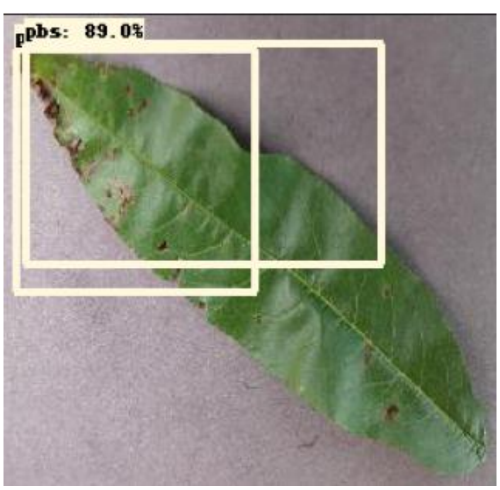


Squash Powdery mildew

Potato Early blight

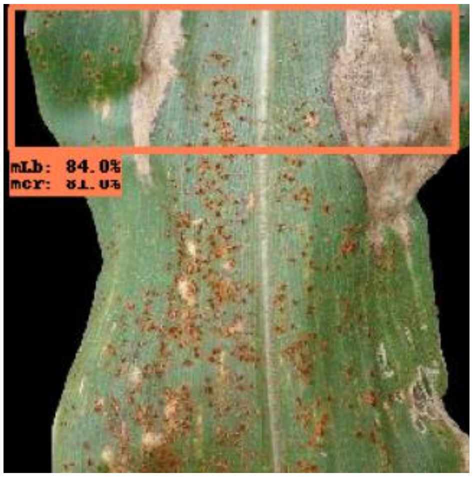


Peach healthy

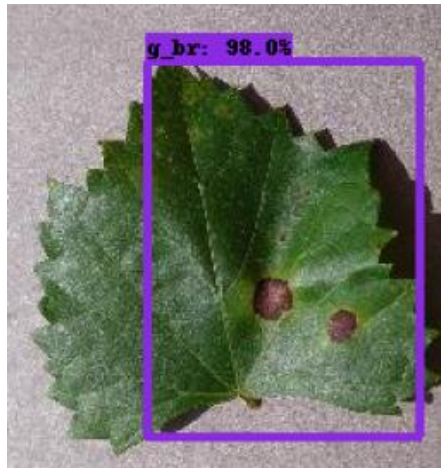


Peach bacterial spot

Maize gray leaf spot

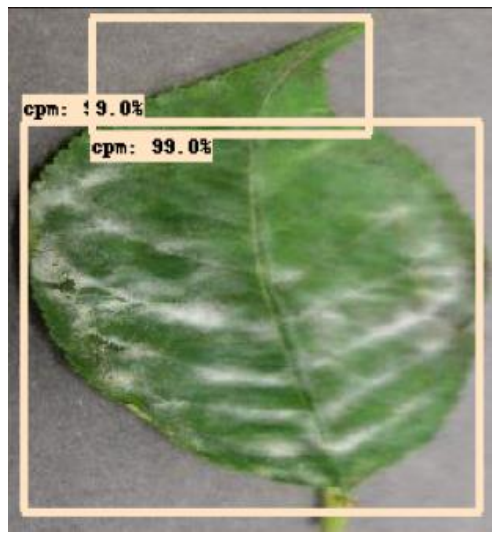


Maize common rust

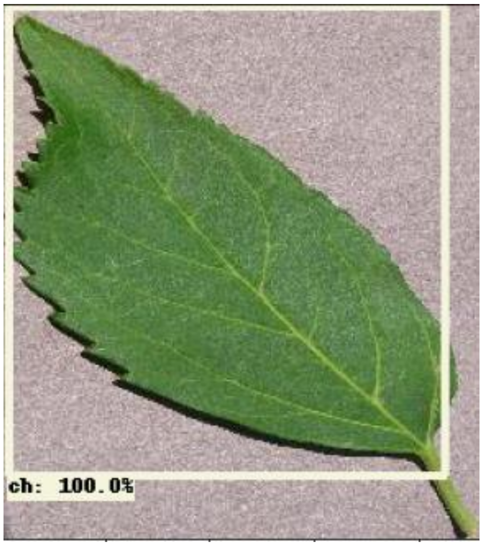


Grape back rot

Grape black measles



Cherry powdery mildew



Cherry healthy

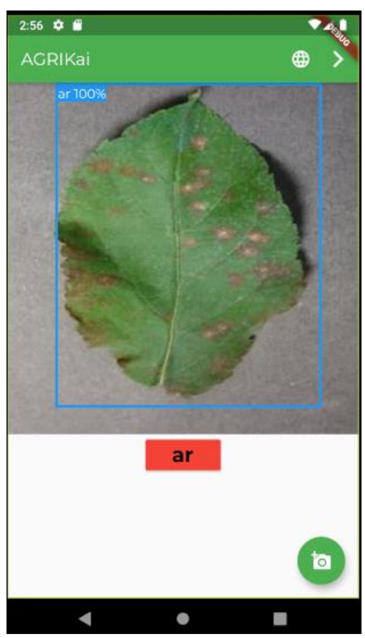
Apple healthy

Figure 19. SSD MobileNet image results on Google Colab

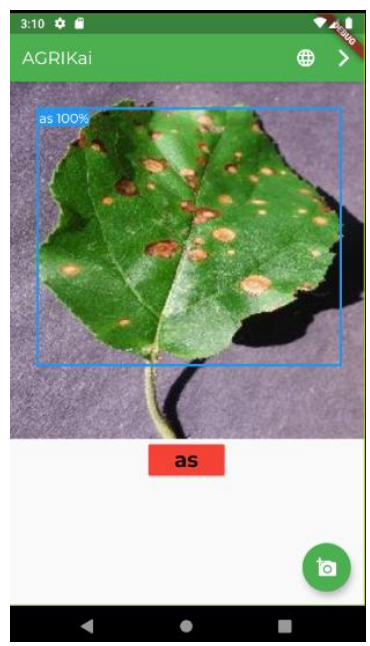
5.3 Results of trained model on Android mobile device

Below are some of the image results of the trained model on google colab after integrating it into a mobile application from an Android device.

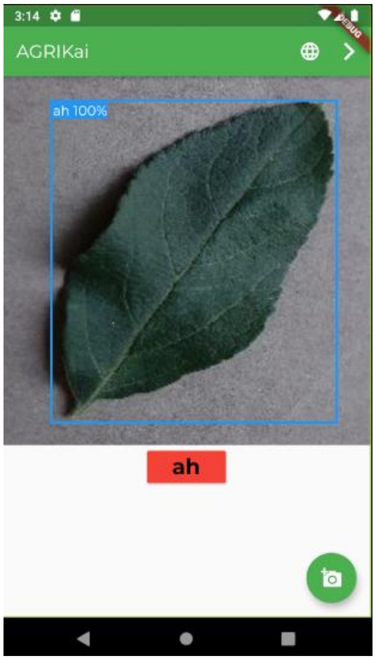
Apple black rot



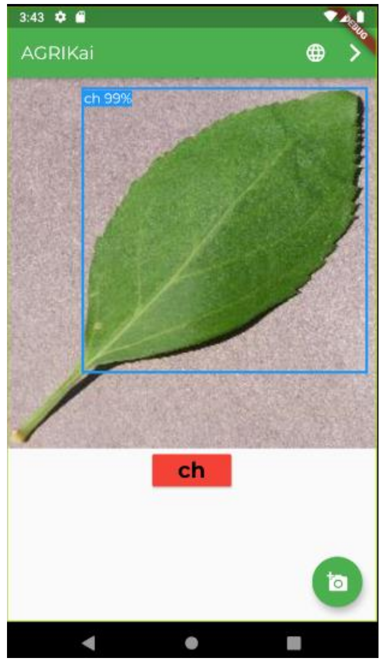
Cedar apple rust



Apple scab

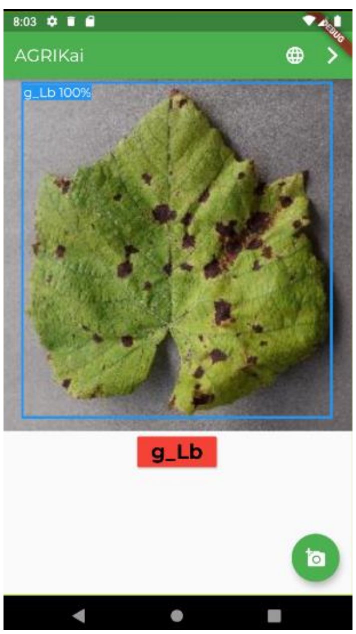


Apple healthy

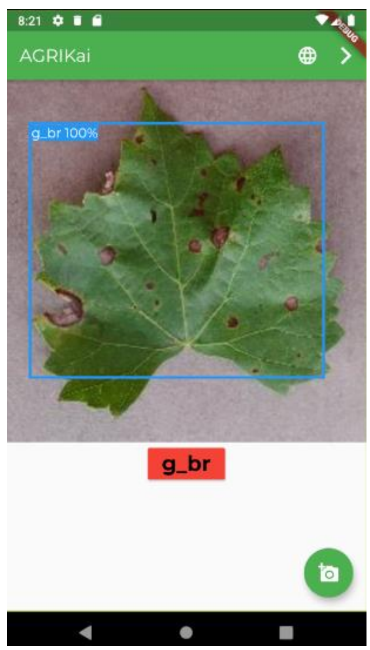


Cherry healthy

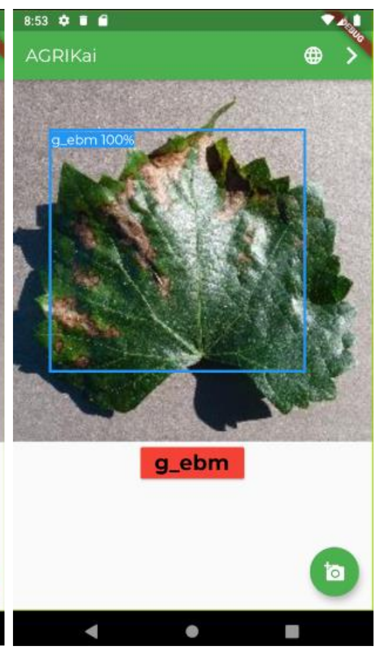
Cherry powdery mildew



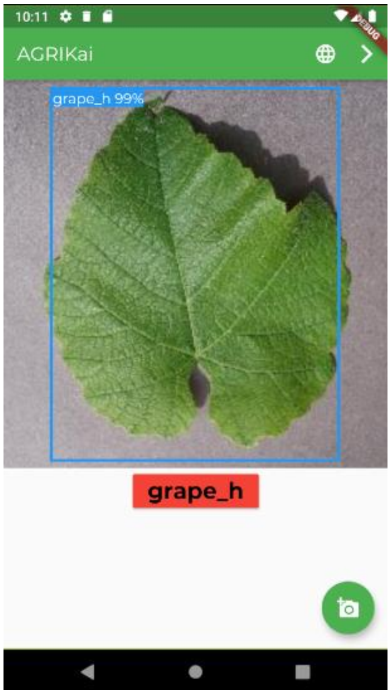
Grape Leaf blight



Grape Black rot



Grape esca (black measles)



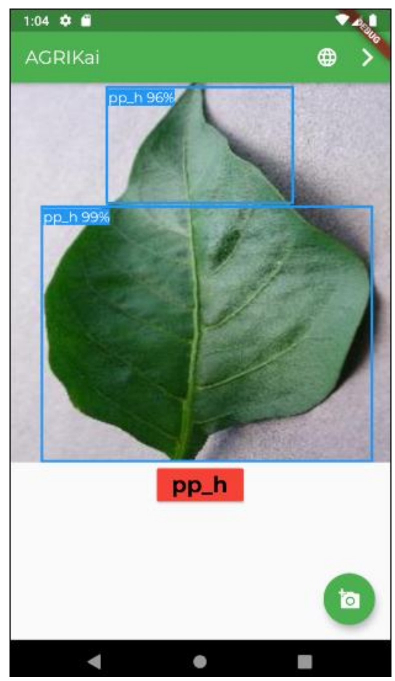
Grape healthy

Maize Leaf blight

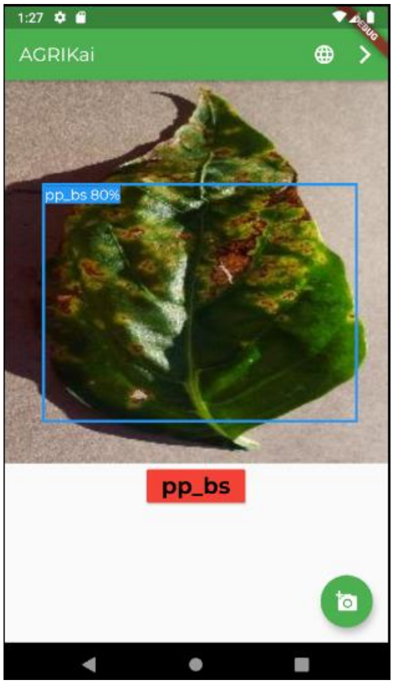
Maize Leaf Spot (Cercospora)

Strawberry leaf scorch

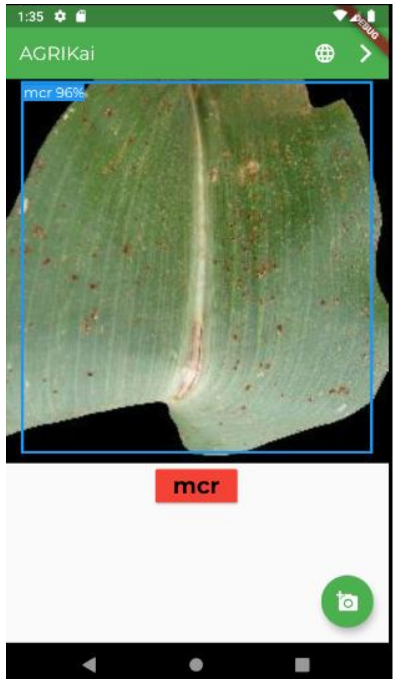
Strawberry healthy



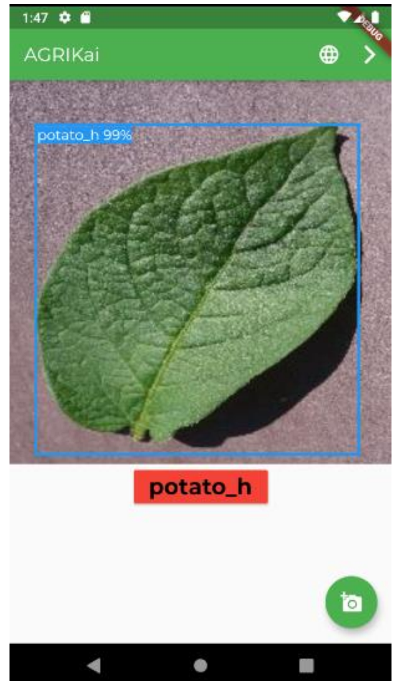
Pepper healthy



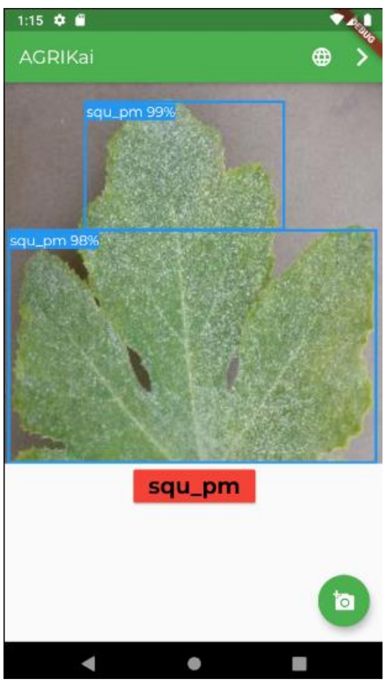
Pepper bacterial spot



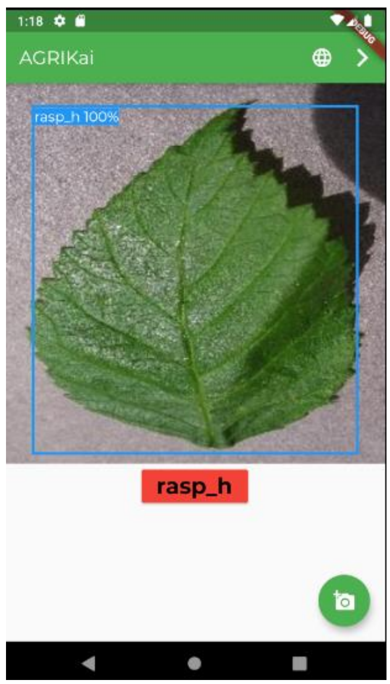
Maize common rust



Potato healthy

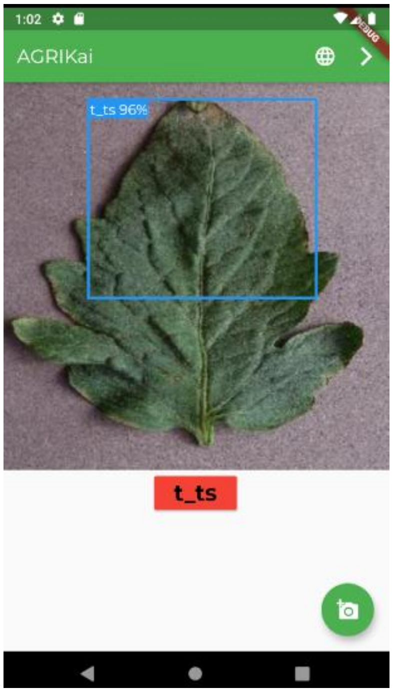


Squash powdery mildew

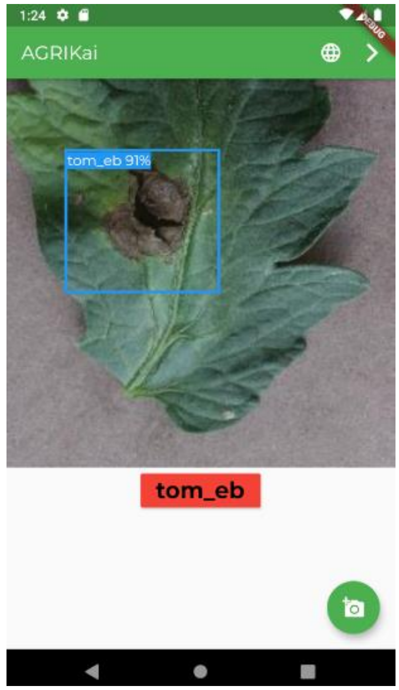


Raspberry healthy

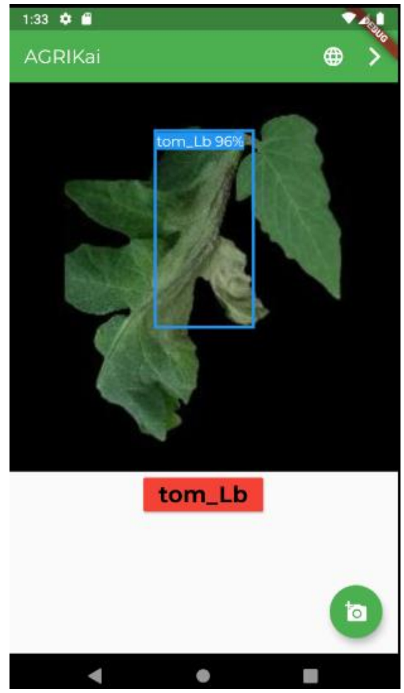
Maize healthy



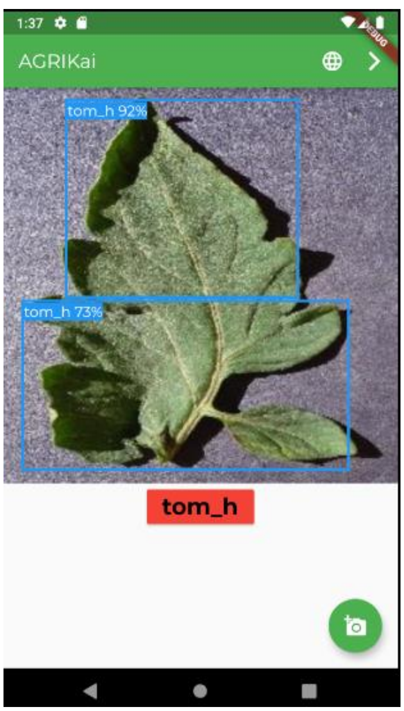
Tomato target spot



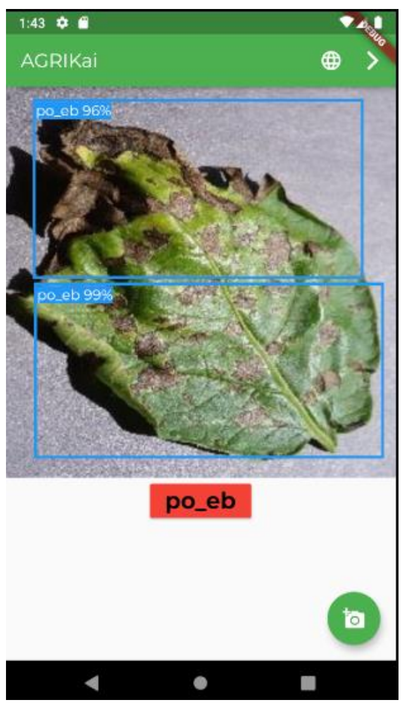
Tomato early blight



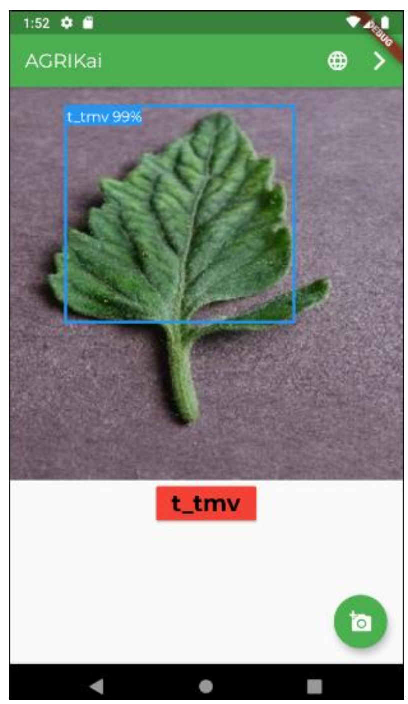
Tomato late blight



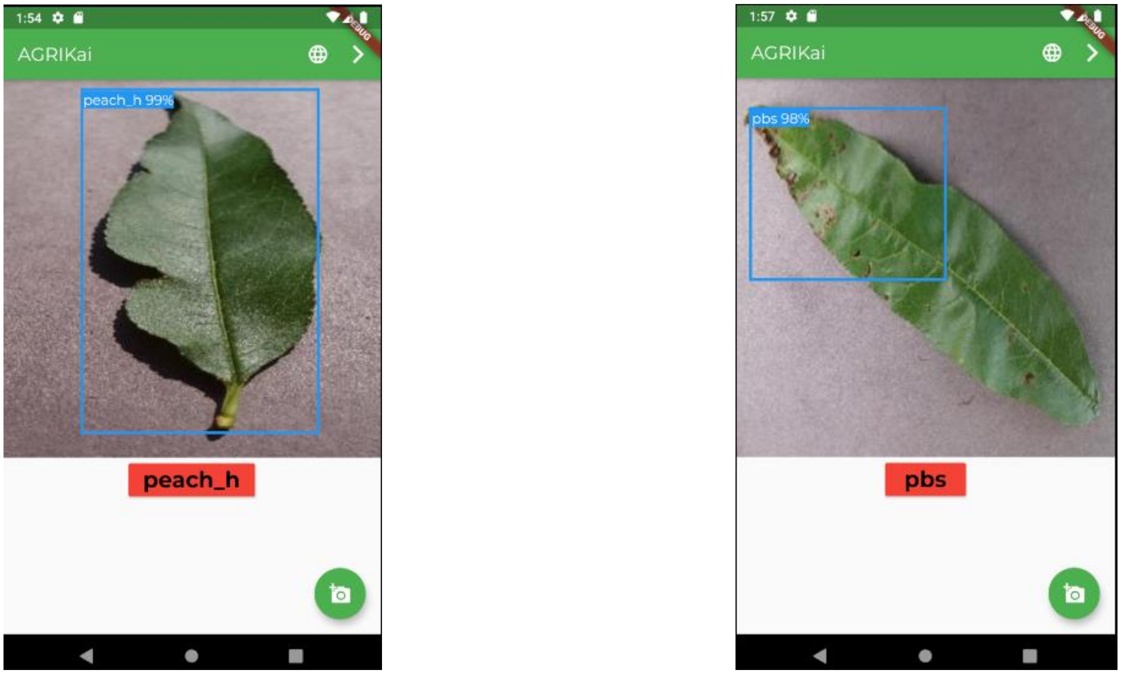
Tomato healthy



Potato early blight



Tomato mosaic virus



Peach healthy Peach bacterial spot

Figure 20. Mobile detector image results.

5.4 Testing mobile detector accuracy

To test the accuracy of the integrated model in the mobile application, we calculate the confusion matrix and classification metrics by selecting 50 images each from each plant disease and running them in the application. Below are the stepson how a confusion matrix is calculated.

1. For each detection record, the algorithm extracts from the processed input file image the ground-truth boxes and classes (what should be predicted by the model), along with the detected boxes, classes, and scores (what the model predicts).
2. Detections with a confidence score greater or equal than 0.5 are considered. While anything below is discarded.
3. For each ground-truth box, the algorithm generates IoU (Intersection over Union) with every detected box. A match is found if both boxes have an IoU greater or equal than 0.5.
4. The list of matches is pruned to remove duplicates (ground-truth boxes that match with more than one detection box or vice versa). If there are duplicates, the best match

(greater IoU) is always selected. This way, an image won't contain too much bounding boxes.

5. The confusion matrix is updated to reflect the resulting matches between ground-truth and detections.

6. Objects that are part of the ground-truth but weren't detected are counted in the last column of the matrix (in the row corresponding to the ground-truth class). Objects that were detected but aren't part of the confusion matrix are counted in the last row of the matrix (in the column corresponding to the detected class).

$$
\begin{aligned}
precision &= \frac{TP}{TP+FP} \\
recall &= \frac{TP}{TP+FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP+TN}{TP+FN+TN+FP}
\end{aligned}
$$

Figure 21. Formula for precision, recall, and f1 scores.

TP (True Positive) – number of positive sample images that are correctly predicted as positive

TN (True Negative) – number of positive sample images that are falsely predicted as negative

FP (False Positive) – number of negative sample images that are correctly predicted as negative

FN (False Negative) - number of negative sample images that are falsely predicted as positive

Important: Number of images used for testing is 50 images of each disease

	Category	**TP**	**FN**	**TN**	**FP**	**accuracy**
Apple	Abr (Apple black rot)	0	50	50	50	0.67 = 67%
	Ah (Apple healthy)	50	0	50	0	1 = 100%
	Ar (Cedar apple rust)	49	1	50	0	
	As (Apple scab)	0	50	50	50	0.67 = 67%
Cherry	Ch (Cherry healthy)	50	0	50	0	1 = 100%

	Cpm (Cherry powdery mildew)	50	0	50	0	1 = 100%
Grape	g_Lb (Grape leaf blight)	50	0	50	0	1 = 100%
	g_br (Grape black rot)	50	0	50	0	1 = 100%
	g_ebm (Grape esca black measles)	50		50	0	1 = 100%
	grape_h (healthy)	50	0	50	0	1 = 100%
Corn (maize)	mLb (Maize leaf blight)	47	3	50	0	0.97 = 97%
	mLs (Maize leaf spot)	49	0	50	3	0.971 = 97%
	Mcr (Maize common rust)	50	0	50	1	0.99 = 99%
	Mh (Maize healthy)	50	0	50	0	1 = 100%
Peach	Pbs (Peach bacterial spot)	49	1	50	0	0.99 = 99%
	peach_h (Peach healthy)	50	0	50	0	1 = 100%
Potato	po_Lb (Potato late blight)	49	1	50	2	0.971 = 97%
	po_eb (Potato early blight)	48	2	50	1	0.970 = 97%
	potato_h (Potato healthy)	50	0	50	0	1 = 100%
Pepper	pp_bs (Pepper bacterial spot)	45	5	50	0	0.95 = 95%
	pp_h (Pepper healthy)	50	0	50	0	1 = 100%
Raspberry	rasp_h (Raspberry healthy)	50	0	50	0	1 = 100%
Strawberry	sberry_Ls (Strawberry leaf scorch)	47	3	50	0	0.97 = 97%
	sberry_h (Strawberry healthy)	50	0	50	0	1 = 100%
Squash	squ_pm (Squash powdery mildew)	50	0	50	0	1 = 100%
	t_sLs (Tomato septoria leaf spot)	48	2	50	4	0.92 = 92%
	t_smt (Tomato spider mite)	46	4	50	0	0.96 = 96%
	t_tmv (Tomato mosaic	46	4	50	0	0.96 = 96%

Tomato	virus)					
	t_ts (Tomato target spot)	49	1	50	4	0.952 = 95%
	t_ty (Tomato yellow leaf virus)	50	0	50	2	0.98 = 98%
	tom_Lb (Tomato late blight)	48	2	50	1	0.97 = 97%
	tom_Lm (Tomato late mold)	50	0	50	0	1 = 100%
	tom_bs (Tomato bacterial spot)	49	1	50	1	0.98 = 98%
	tom_eb (Tomato early blight)	49	1	50	2	0.971 = 97%
	tom_h (Tomato healthy)	50	0	50	0	1 = 100%

Table 2. Confusion matrix of mobile detector.

CHAPTER 6

6.0 CONCLUSION

6.1 Project Review

During the course of the project, the main goal has been to develop a mobile application with a detector integrated in it to detect plant diseases through images. Chapter 3, 4 and 5 have explained in detail how we managed to accomplish the goal also going further to put our work on Github with steps to guide and make it easier for future update and reference. Also, the mobile application developed during the project is capable of running on both Android and iOS devices meaning we can reach a larger audience to make use of our detector from their smartphones.

6.2 Problems encountered

Some problems were encountered during the project and were able to be resolved like using cloud resources (Google Colab, Google Drive), after a few days it was discovered that the files generated during training all happen to be very large which would be too much for just a computer system memory to handle, and Google Colab's policy which made all our files deleted automatically on the cloud after 8-12 hours of training but mounting my Google Drive account and storing the files there automatically also resolved this issue.

6.3 Discussions and Conclusions

In this project, we address pests and disease identification by introducing the application of the SSD MobileNet architecture. In the real world, farmers and other agriculture experts go through tedious and time consuming processes of visually carrying out inspection of agricultural crops such as fruits, and vegetables which are likely to be affected by different diseases and doesn't guarantee an accurate recognition and classification of the plant pests or diseases [20]. These diseases could as well travel to establish itself in other countries if trading occurs between them [1]. So, we basically develop a system that successfully recognizes different pests and diseases gathered in real scenarios. Moreover, our system is able to deal with complex tasks like infected location in plant (e.g. leaves, stem), sides of leaves, different background conditions and so on. Through this way, we can control the emergence of diseases from becoming uncontrollable and jeopardizing food security.

6.4 Potential benefits and impacts

Food security is a major concern with the world population growth expected to be more than 9.7 billion by 2050. Plant disease has been a threat to food security. Therefore, accurate methods need to be applied in order to identify the diseases and appropriate measures can be executed. This project develops a fast and accurate model for plant disease identification and detection so that appropriate measures can be applied early, thus mitigating the issue of food security.

In addition, the mobile application developed in this project will be a valuable tool for farmers, especially for those live in areas that are lacking the appropriate infrastructure and have limited services for the provision of agronomic advice.

REFERENCES

1. Dean, P. (2016). *Global agriculture study finds developing countries most threatened by invasive pest species*. [online] The Conversation. Available from: <https://theconversation.com/global-agriculture-study-finds-developing-countries-most-threatened-by-invasive-pest-species-61280> [Accessed 4 Apr. 2019].

2. Fao.org., *Plant pests and diseases: FAO in Emergencies*. [online] Available from: <http://www.fao.org/emergencies/emergency-types/plant-pests-and-diseases/en/> [Accessed 4 Apr. 2019].

3. BBC News. (2011). *Plant pests: The biggest threats*. [online] Available from: <https://www.bbc.com/news/science-environment-15623490/> [Accessed 4 Apr. 2019].

4. Too, E., Yujian, L., Njuki, S. and Yingchun, L. (2018). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, pp.1-2.

5. Liew, O., Chong, P., Li, B. and Asundi, A. (2008). Signature Optical Cues: Emerging Technologies for Monitoring Plant Health. *Sensors*, vol. 8(5), pp.3205-3239.

6. Mazarei, M., Teplova, I., Hajimorad, M. and Stewart, C. (2008). Pathogen Phytosensing: Plants to Report Plant Pathogens. *Sensors*, vol. 8(4), pp.2628-2641.

7. Ryant, P., Dolezelova, E., Fabrik, I., Baloun, J., Adam, V., Babula, P. and Kizek, R. (2008). Electrochemical Determination of Low Molecular Mass Thiols Content in Potatoes (Solanum tuberosum) Cultivated in the Presence of Various Sulphur Forms and Infected by Late Blight (Phytophora infestans). *Sensors*, vol. 8(5), pp.3165-3182.

8. Fuentes, A., Yoon, S., Kim, S. and Park, D. (2017). A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors*, vol. 17(9), p.2022.
9. Arivazhagan, S., Vineth Ligi, S. (2018). Mango Leaf Diseases Identification Using Convolutional Neural Network, vol. 167
10. R.Preethi, S.Priyanka, U.Priyanka , A.Sheela, (2015), Efficient knowledge based system for leaf disease detection and classification, International Journal of Advance Research In Science And Engineering, Vol. 4.
11. Revathi, P. and Hemalatha, M. (2012). Homogenous Segmentation based Edge Detection Techniques for Proficient Identification of the Cotton Leaf Spot Diseases. *International Journal of Computer Applications*, vol. 47(2), pp.18-21.
12. Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, vol. 267, pp.378-384.
13. Khairnar, K. and Dagade, R. (2014). Disease Detection and Diagnosis on Plant using Image Processing A Review. *International Journal of Computer Applications*, vol. 108(13), pp.36-38.
14. Zhang, L. and Yang, B. (2014). Research on Recognition of Maize Disease Based on Mobile Internet and Support Vector Machine Technique. *Advanced Materials Research*, 905, pp.659-662.
15. Mohanty, S., Hughes, D. and Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7.

16. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience*, 2016, pp.1-11.

17. Dyrmann, M., Karstoft, H. and Midtiby, H. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, vol. 151, pp.72-80.

18. Krizhevsky, A., Sutskever, I. and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, vol. 60(6), pp.84-90.

19. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. *Computer Vision and Pattern Recognition*, pp. 1–9.

20. Pujari, J., Yakkundimath, R. and Byadgi, A. (2015). Image Processing Based Detection of Fungal Diseases in Plants. *Procedia Computer Science*, vol. 46, pp.1802-1808.

21. Kashyap, Y., Sharma, T. and Shahnawaz, S. (2017). Analysis of Plant Diseases with Detection using Image Processing Methods. *International Journal of Computer Applications*, vol. 166(7), pp.28-31.

22. Dubey, S., Dixit, P., Singh, N. and Gupta, J. (2013). Infected Fruit Part Detection using K-Means Clustering Segmentation Technique. *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2(2), p.65.

23. Baghel, J. and Jain, P. (2016). Disease Detection in Soya Bean using K-Means Clustering Segmentation Technique. *International Journal of Computer Applications*, 145(9), pp.15-18.

24. Masood, R., A. Khan, S. and N.A. Khan, M. (2016). Plants Disease Segmentation using Image Processing. *International Journal of Modern Education and Computer Science*, vol. 8(1), pp.24-32.

25. Xu, P., Wu, G., Guo, Y., chen, X., Yang, H. and Zhang, R. (2017). Automatic Wheat Leaf Rust Detection and Grading Diagnosis via Embedded Image Processing System. *Procedia Computer Science*, vol. 107, pp.836-841.

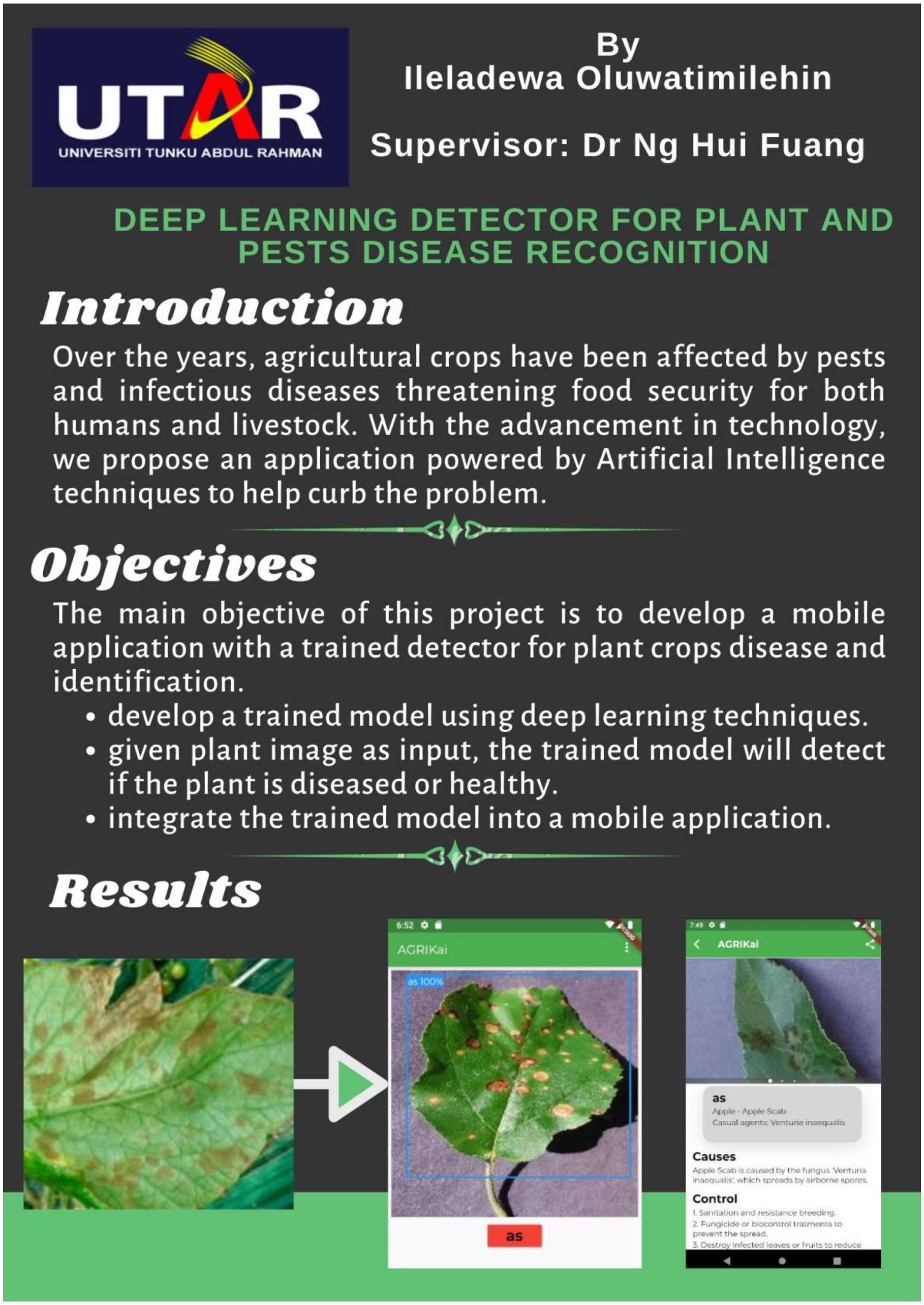
26. Meng, T. and Shyu, M. (2014). Multi-layer Model Collaboration for Bioimage Temporal Stage Classification. *International Journal of Semantic Computing*, 08(02), pp.123-144.

27. Debasu Mengistu, A., Mengistu, S. and Alemayehu, D. (2018). An Automatic Coffee Plant Diseases Identification Using Hybrid Approaches of Image Processing and Decision Tree. *Indonesian Journal of Electrical Engineering and Computer Science*, vol.9(3), p.806.

28. Akila M., Deepan, P., (2018). Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm. *International Journal of Engineering Research and Technology (IJERT)*, vol 88, p 902.

29. Gandhi, R. (2019). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. [online] Medium. Available at: https://towardsdatascience.com/r-cnn-fast-r- cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e [Accessed 26 Jul. 2019].

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
By
Ileladewa Oluwatimilehin
Supervisor: Dr Ng Hui Fuang
DEEP LEARNING DETECTOR FOR PLANT AND PESTS DISEASE RECOGNITION
Introduction
Over the years, agricultural crops have been affected by pests and infectious diseases threatening food security for both humans and livestock. With the advancement in technology, we propose an application powered by Artificial Intelligence techniques to help curb the problem.
Objectives
The main objective of this project is to develop a mobile application with a trained detector for plant crops disease and identification.
• develop a trained model using deep learning techniques.
• given plant image as input, the trained model will detect if the plant is diseased or healthy.
• integrate the trained model into a mobile application.
Results
6:52
AGRIKai
as 100%
as
7:49
AGRIKai
as
Apple - Apple Scab
Casual agents: Venturia inaequalis
Causes
Apple Scab is caused by the fungus 'Venturia inaequalis', which spreads by airborne spores.
Control
1. Sanitation and resistance breeding.
2. Fungicide or biocontrol tratments to prevent the spread.
3. Destroy infected leaves or fruits to reduce

FYP Report Check

ORIGINALITY REPORT

11%	6%	9%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Edna Chebet Too, Li Yujian, Sam Njuki, Liu Yingchun. "A comparative study of fine-tuning deep learning models for plant disease identification", Computers and Electronics in Agriculture, 2019 Publication	2%
2	mafiadoc.com Internet Source	2%
3	www.shiftedup.com Internet Source	1%
4	supervise.ly Internet Source	1%

5	theconversation.com Internet Source		1%
6	Alvaro Fuentes, Sook Yoon, Sang Kim, Dong Park. "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition", Sensors, 2017 Publication		1%
7	Waleed Al-Ghaith. "Developing Lexicon-based Algorithms and Sentiment Lexicon for Sentiment Analysis of Saudi Dialect Tweets", International Journal of Advanced Computer Science and Applications, 2019 Publication		<1%
8	Md. Helal Sheikh, Tahmina Tashrif Mim, Md. Shamim Reza, Most. Hasna Hena. "Leaf Diseases Detection for Commercial Cultivation of Obsolete Fruit in Bangladesh using Image Processing System", 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), 2019 Publication		<1%

9 Haoxu Yang, Lutao Gao, Niansheng Tang, Po Yang. "Experimental analysis and evaluation of wide residual networks based agricultural disease identification in smart agriculture system", EURASIP Journal on Wireless Communications and Networking, 2019
Publication <1%

10 wiredspace.wits.ac.za
Internet Source <1%

11 Geetharamani G., Arun Pandian J.. "Identification of plant leaf diseases using a nine-layer deep convolutional neural network", Computers & Electrical Engineering, 2019
Publication <1%

12 Jayme Garcia Arnal Barbedo. "Digital image processing techniques for detecting, quantifying and classifying plant diseases", SpringerPlus, 2013
Publication <1%

13 Peng Jiang, Yuehan Chen, Bin Liu, Dongjian He, Chunquan Liang. "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks", IEEE Access, 2019 <1%
Publication

14 Sharada P. Mohanty, David P. Hughes, Marcel Salathé. "Using Deep Learning for Image-Based Plant Disease Detection", Frontiers in Plant Science, 2016 <1%
Publication

15 Radhika Bhagwat, Yogesh Dandawate. "Indian plant species identification under varying illumination and viewpoint conditions", 2016 Conference on Advances in Signal Processing (CASP), 2016 <1%
Publication

16 www.ncbi.nlm.nih.gov <1%
Internet Source

17 res.mdpi.com <1%
Internet Source

18 Alvaro F. Fuentes, Sook Yoon, Jaesu Lee, Dong Sun Park. "High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank", Frontiers in Plant Science, 2018

Publication

<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	ILELADEWA OLUWATIMILEHIN ADEKUNLE
ID Number(s)	1705130
Programme / Course	Computer Science
Title of Final Year Project	DEEP LEARNING DETECTOR FOR PESTS AND PLANT DISEASE RECOGNITION

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 11 % **Similarity by source** Internet Sources: 6 % Publications: 9 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as follows:
- **(i) Overall similarity index is 20% and below, and**
- **(ii) Matching of individual sources listed must be less than 3% each, and**
- **(iii) Matching texts in continuous block must not exceed 8 words**

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Ng Hui Fuang

Date: 10/09/2020

Signature of Co-Supervisor

Name:

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	1705130
Student Name	Ileladewa Oluwatimilehin Adekunle
Supervisor Name	Dr. Ng Hui Fuang

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. Ainca ______________________ (Signature of Student) Date: 10/09/2020	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ______________________ (Signature of Supervisor) Date: 10/09/2020