BANANA DISEASES DETECTION MOBILE APPLICATION

BY

LEE WAI HIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Banana Diseases Detection Mobile Application

Academic Session: JAN 2022

I LEE WAI HIN
(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:
NO 532, LALUAN SIMEE 9,
KAMPUNG SIMEE,
31400 IPOH, PERAK

Ts LAI SIEW CHENG
Supervisor's name

Date: 20/04/2022

Date: 21/4/2022

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF __INFORMATION AND COMMUNICATION TECHNOLOGY__

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __20/04/2022__

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***LEE WAI HIN***______ (ID No: __***18ACB02736***__) has completed this final year project/ dissertation/ thesis* entitled "_____*Banana Diseases Detection Mobile Application* _" under the supervision of ___Ts_LAI SIEW CHENG____ (Supervisor) from the Department of ___Computer Science___, Faculty/Institute* of _Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*LEE WAI HIN*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**BANANA DISEASES DETECTION MOBILE APPLICATION**" 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 : LEE WAI HIN

Date : 20/04/2022

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and thanks to my supervisors, Ts Lai Siew Cheng and my moderator, Prof. Dr Leung Kar Hang who has given me the opportunity to engage in a computer vision task mobile application development project. Other than that, they have guided me in the progress of my project development. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

Banana crop yields have significantly reduced due to the continuously exploitation of the banana disease in the plantation which happening in Malaysia. Farmers have to take action in order to resolve this matter. However, in a conventional way is that they will always perform trial and error to examine the symptoms of the banana disease of their banana plant. This action may require a lot of time and it is not efficient for an early-stage diagnosis mechanism. Other than that, not most of the farmers willing to pay a huge expenditure for advisory in their plantation to resolve the disease matter. An early-stage diagnosis and low budget expenditure proposed method are required to resolve problems that farmers faced. Due to the high affordability of mobile phone and advanced of computer vision task, a mobile application can be developed that deploy with the deep learning model can acts as a medium for farmers to perform classification on the banana leave to detect the banana disease within the banana plant. From literature reviews, researchers are carrying out study on techniques of using CNN architectures to perform classification of banana diseases and deploying into a mobile-based solution. There are several available systems on the market information written in literature review section of the report. A clearer understanding regarding of computer vision task on mobile application will be presented in this report based on the reviews of strengths and weaknesses of existing system on market. The final product of this project is a developed mobile application that capable to classify disease of the infected banana leaf images. The banana disease detector is based on the TensorFlow Lite model which converted from TensorFlow Keras model of EfficientNetV2-B0 by transfer learning. The EfficientNetV2-B0 model has achieved overall 95.00% on F1-score and 95.26% in testing score. F1-score is a score combined with precision and recall score in a single metric used to evaluate performance of classifiers The banana disease detector can classify infected banana leaf images under image condition such as low light intensity with flashlight on and minor obstacles covering infected banana leaf. However, it does have some limitation where unable to classify if the image was extremely dark, blurry, and taken from far distance. Other than that, it equipped with a comprehensive community feature allowed users to share their problems and thoughts and provided with different kind of banana diseases information such as disease symptoms, causal and prevention measures. The application backend is supported by the services from the Firebase platform that working with Java programming language in Android Studio.

TABLE OF CONTENTS

APPENDIX G: FYP2 REPORT CHECKLISTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
IT	Information Technology
ANN	Artificial Neural Network
DTL	Deep Transfer Learning
API	Application Programming Interface
mAP	Mean Average Precision
CM	Confusion Matrix
IoU	Intersection Over Union
SSD	Single Shot Multi-box
CIAT	International Center for Tropical Agriculture
CGIAR	The Consultative Group on International Agricultural Research
APP	Application
PEAT	Progressive Environmental & Agricultural Technologies
Xcm	Xanthomonas Campestris pv. Musacearum
BBW	Banana Bacterial Wilt
CMV	Cucumber Mosaic Virus
BBTV	Banana Bunch Top Virus
BBS	Banana Black Sigatoka
IDE	Integrated Development Environment
GPU	Graphical Processing Unit
TPU	Tensor Processing Unit
aka	Also Known As
TFLITECONVERTER	TensorFlow Lite Converter

Chapter 1 Introduction

This chapter is to present background and motivation of this project, contributions to the study field and the outline of the thesis.

1.1 Problem Statement and Motivation

Over the years, Malaysia banana production yields has drastically declined due to the severe impact of banana diseases[1]. On top of that, the local banana farmers are traditionally using labour forces and seek the help from the government advisory agricultural parties to identify the banana diseases occurrence for each banana tree in their banana plantations. Such conventional methods and techniques will greatly deplete those local farmers' time and funds. Thus, to greatly reduce the occurrence of diseases outbreak in the banana plantations, an early-stage diagnosis is mandatory and as a crucial first step to be implemented by the local farmers[2]. Other than that, the development of an automated and guided system which capable to detect banana diseases via the banana leaves will come in handy and able to reduce the funds and time consumption in larger margin.

With the use of computer vision and machine learning, it is possible to develop a system of detecting the banana diseases by examining the banana leaves. Other than that, due to vast usage of smartphone of this modern era; majority of the Malaysia citizen are capable to afford smartphones including those local farmers as well. Hence, based on these two factors are sufficient to motivate the establishment of this project to develop a mobile application that deploy with the banana disease detection system. Local farmers are able to perform their early-stage banana diseases diagnosis by utilizing the developed mobile application. Therefore, this project has been proposed and banana disease detection mobile application has been developed as the result of this project.

1.2 Project Scope

This project is proposed to develop a banana diseases detection mobile application based on deep learning approach. This mobile application can detect and classify the banana disease within the banana tree by classifying the banana leaves. Other than that, this mobile application is capable to provides some recommendations on the treatment to cure the banana tree from the diseases. This mobile application is expected to be an early-stage diagnosis technique for the banana plantation farmers without spending any heavy fees on banana diseases advisory from any related society, business company and advisory department from the government. The targeted user in this project is the banana plantation farmers. This mobile application is expected to support on Android platform.

1.3 Project Objectives

- **To detect banana diseases using EfficientNetV2 architecture as the deep learning approach for construct neural network training model**
  In this project, a training model need to be construct in order for the use of banana diseases detection. A deep learning approach will be selected when constructing the training model. The dataset used for the training model to train will be the affected banana leaves images dataset. A compiled and constructed training model can be trained and perform the classification purpose. Hence, the trained model should be able to classify each class of the affected banana leaves.

- **To deploy the banana disease detection trained model into the mobile application**
  Normally, it would be a huge size of trained model to deploy into a mobile application environment. The trained model needs to be converted to a more optimized version of trained model. Hence, the optimized version of trained model can perform classification more efficiently in mobile application.

- **To implement a community feature in the mobile application for users perform information exchange purpose**
  Community feature would be pleasant and informative interface for the users to interact with. Users can upload photos or images of their banana plant for other users to interact and reference. Other than that, users can comment and give opinions in the discussion which regarding information of banana diseases, crops and some related important tips. Hence, information exchange can be achieved by all of the users when they are interacting with.

- **To implement a crop guide cycle in the mobile application for guiding users to do maintain their crop**
  A crop guide cycle feature will be a great information source for guiding the users on each related phases of banana crop cycle. Users will be informed with the banana crop cycle sequences and different phases diseases happening for the

banana crop cycle. Hence, users can follow the guide and perform with the correct methods when they are handling the crop cycle of banana plant.

- **To implement a banana diseases prevention and control measures information feature for alert and recommend users**

  Banana diseases prevention and control measures feature is to provide users the information regarding the banana diseases symptoms. Other than that, users can receive more details on the recommendation in preventive measures. Moreover, users are informed with the recommendation information of control measures in terms of chemical control and organic control. Therefore, they are equipped with necessary knowledge of control and prevention measures when dealing with the banana diseases.

1.4 Impact, Significance, and Contribution

As mentioned before in a previous section, the banana diseases happening in Malaysia have drastically impact and decline the banana production yields over the years. Local banana farmers have been implementing the conventional methods such as the using labour forces and spent heavy funds on seek the government advisory agricultural parties to resolve the banana diseases outbreak. However, it seems these methods doesn’t produce significant improvement of the banana production yields over the years. Hence, the local farmers can no longer capable to prevent further loss of their resources and funds due to the banana diseases outbreak.

The contributions of this project are including the developed mobile application of banana disease detection system with implemented by the idea of using convolution neural network (CNN) as known as the subset of deep learning to train a model that can perform a high accuracy on banana disease detection via banana leaf classification for the great reduction on financial loss currently face by those local banana farmers. Other than that, the mobile application banana disease detection system is focusing on the major and chronic banana diseases which are actively occur in Malaysia. Furthermore, this mobile banana disease detection also contributes the accurate treatment recommendations according to detected banana diseases. Besides, this proposed mobile application can increase the user-experience and user-reliability because it is lightweight and consume lesser power usage and consume lesser memory of the phone. Moreover, this mobile application is suitable to run on any Android devices. In conclusion, this proposed mobile application is expected to be an early-stage diagnosis technique to detect the banana diseases, reduce financial loss and increase the banana yields for the banana plantation farmers.

1.5 Report Organization

This report is organized into total of 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology/Approach, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion and Chapter 7 Conclusion and Future Work. The first chapter is the introduction of this project which includes problem statement and motivation, project scope, project objectives, impact, significance, and contribution and report organization. The second chapter is the literature review on chronological works in neural network and existing systems. The third chapter is to show diagram of the system methodology. The fourth chapter is to the overall project flow, machine learning pipeline and mobile application system designs. The fifth chapter is the setup of hardware and software, settings and configuration and system operation. The sixth chapter is the system evaluation and discussion on machine learning model performance, deployed mobile application model performance, project challenges and objectives evaluation. The last chapter is conclusion and future work.

Chapter 2 Literature Review

2.1 Neural Network

A neural network is a network which mimic as same as the behavior of a human brain. Neural network allowed computer programs to perform recognition on the patterns. Other than that, it also capable to solve common problems in vast fields of Artificial Intelligence, for example including the deep learning and machine learning as well.

There are several published papers that surrounding the study with neural networks. In the year of 1943, Walter Pitts and Warren S. McCulloch have published a paper titled “A logical calculus of the ideas immanent in nervous activity. This research paper is to investigate and understand how neurons or connected brain cells in human brain could produce such complex patterns. A Boolean logic compared to neurons with a binary threshold study work is one of the main ideas proposed by them[3].

About 15 years later after research paper proposed by Walter Pitts and Warren S. McCulloch, Frank Rosenblatt took Pitts and McCulloch’s research work into further step by introducing weights into the proposed equation. Other than that, he was credited with the perceptron development that documented in his research paper titled “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain” [4].

In the year of 1989, Yann LeCun along with other researchers have published a paper titled “Backpropagation Applied to Handwritten Zip Code Recognition”. This research paper was illustrating the use of constraints in backpropagation and its integration into neural network architecture can be implemented to train algorithms [5].

Neural networks sometimes can be known as artificial neural network (ANNs) which are a subset of machine learning and can be as a core of the deep learning algorithms.

The structure of an artificial neural networks (ANN) contained of node layers such as input layer, one or more hidden layers and lastly an output layer shown in Figure 2.1.1. Each artificial neuron is connected to another with threshold and associated weight. Node will be activated if one individual node output is above specified threshold value. Once node activated, it will send data to next layer of the network [6].

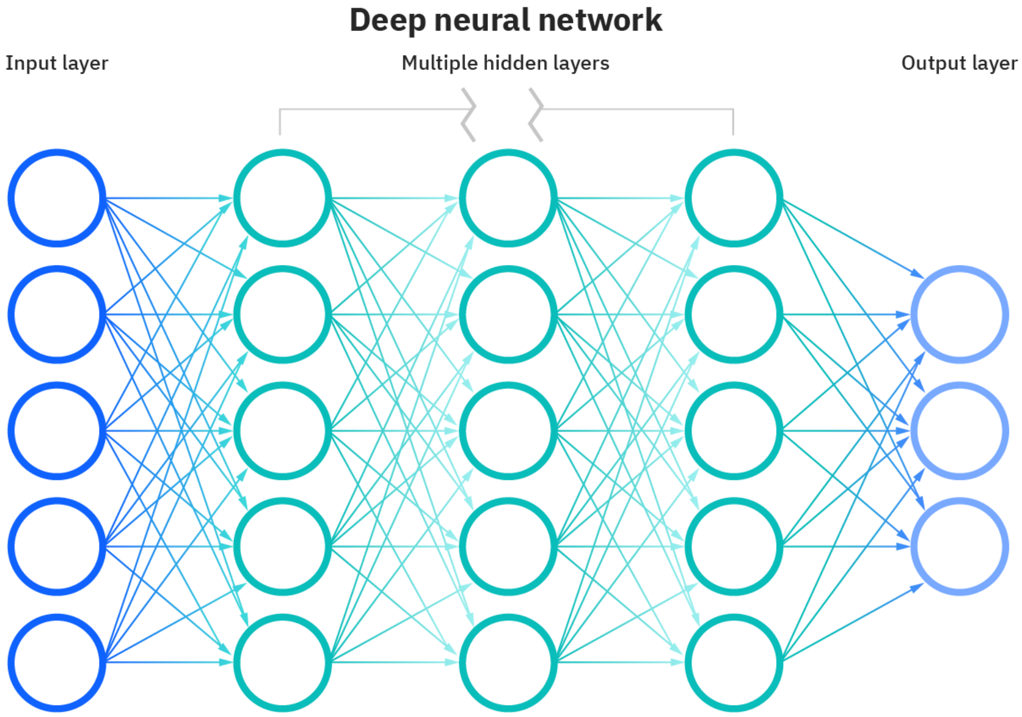


Figure 2.1.1 Structure of Artificial Neural Network [6]

Neural networks are relatively dependent on training data to learn and improve the neural network accuracy over time. These neural networks could be the most reliable and powerful tools in computer science tasks and artificial intelligence as well after successfully fine-tuned for accuracy. Computer vision tasks such as speech recognition and image recognition will be consuming only few minutes compared to hours by manual identification performed by human experts [6].

2.2 Deep Learning

One of the subsets in machine learning would be the deep learning. Deep learning is a field that based on continuously learning and improving on its own by executing and examining the computer algorithms. Comparison between machine learning and deep learning is that machine learning is uses much more simpler concepts while deep learning is work on artificial neural network which designed in a way that mimic the way of human learn and think. Neural networks were limited in complexity and by the computational power. Fortunately, The Big Data analytics advancement have grew larger of the sophisticated neural networks which enable computers to learn, react and observe to high complexity situations faster than ordinary human. Hence, deep learning has guided a lot of computer science tasks such as speech recognition, language translation and image classification [7].

Artificial neural networks which composed of many layers can drive the deep learning purpose. Thus, Deep Neural Networks (DNNs) was produced where each layer able to perform complex operations such as abstraction and representation which bring the meaning and senses of sound, text and images. Therefore, companies around the globe have created new business model and business direction based on the usage of deep learning technology [7].

The Deep Neural Network (DNNs) would be having much more deeper based network based on the layers number it has. Other than that, DNNs would have much heavier weighted node that exert more effect on next layer of nodes in the network. However, such heavy layers of nodes present, deep learning systems indeed require a more powerful hardware where also a large amount of data being processed. Moreover, it does involve complex mathematical calculations. In order for deep learning system to produce accurate results, indeed that it needed to takes a huge datasets of huge amount of data [7].

2.3 Convolutional Neural Network

Nowadays, machines are enabled to view the world as what humans do. Machines also able to perceive the view in similar way and uses knowledges for various of computer sciences tasks such as image analysis, image classification, media recreation, video recognition and natural language processing, etc. A particular algorithm that leaded the advancements in computer vision with deep learning is the Convolutional Neural Network (CNN).

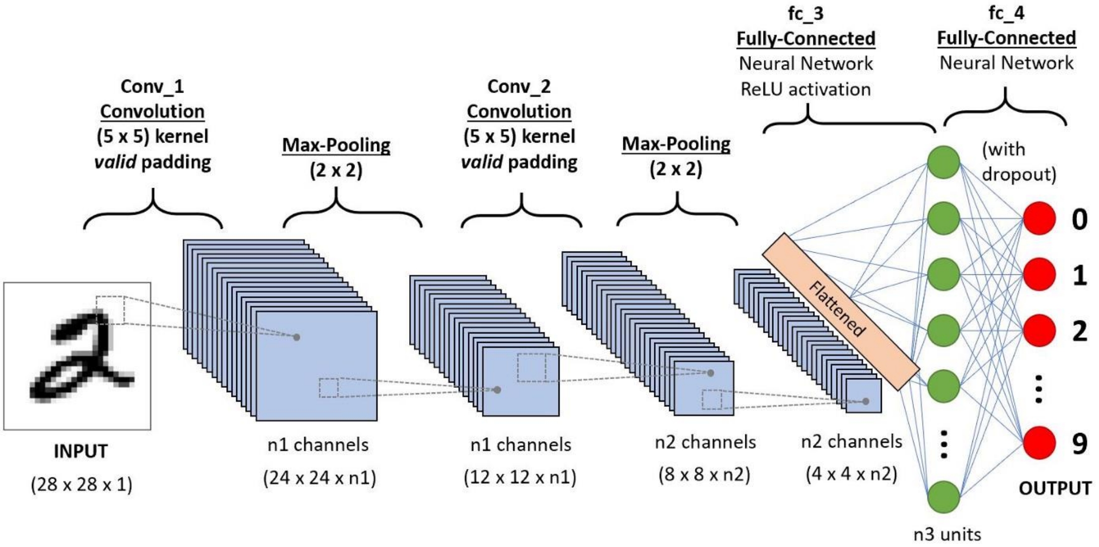


Figure 2.3.1 An example of CNN sequence on classify handwritten digits [8]

The Deep Learning algorithm, Convolutional Neural Network that can receive an input image and assigned it with importance of biases and learnable weights to various aspects in image which normally we called it as feature for differentiate one from the other [8]. One of the benefits using Convolutional Neural Network is require lower need of pre-processing for images such as image segmentation, augmentation, etc. Because Convolutional Neural Network posses the ability of learn the characteristics (features) of each image have taken in as input. Furthermore, the Convolutional Neural Network can be trained to understand the complexity of the image even better due to reduction in parameters used and weights reusability.

The Input Image

Figure 2.3.2 show the three separated color planes which are Red, Green and Blue as the RGB image input into the Convolutional Neural Network. Convolutional Neural Network is such a role that tend to reduce image into simpler form which benefited in process while without suffer from features loss that critical for resulting accurate prediction.

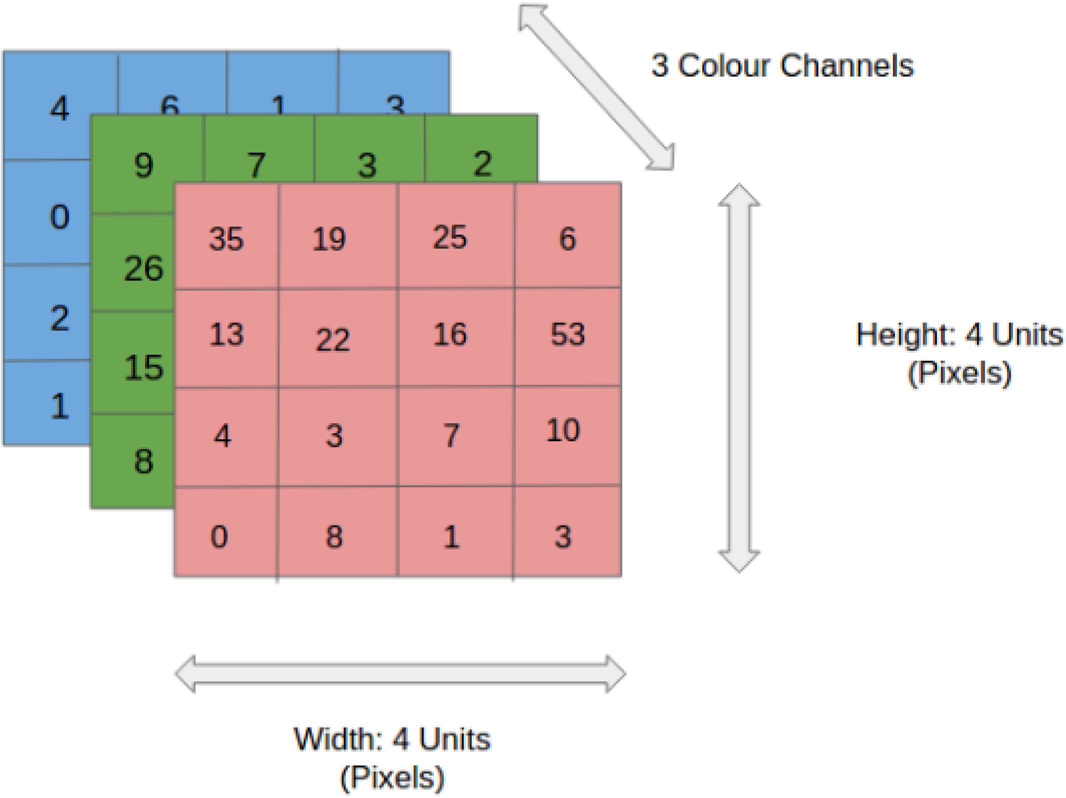


Figure 2.3.2 A 4x4x3 RGB Image input into the Convolutional Neural Network [8]

The Convolutional Layer (The Kernel)

Figure 2.3.3 show a 5x5x1 input image with green section, is being carry out convolution operation by the Kernel as known as Filter which represented as yellow color in 3x3x1 matrix.

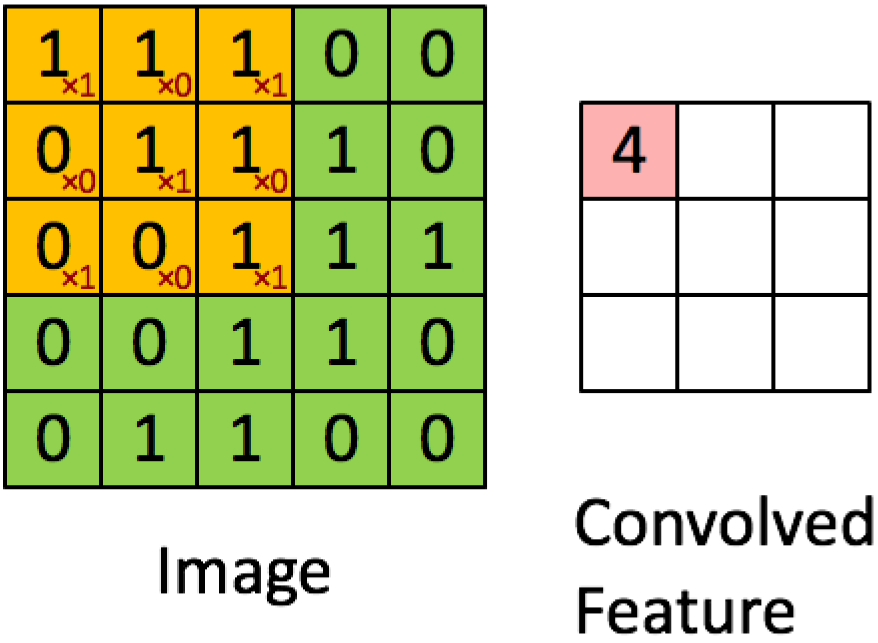


Figure 2.3.3 A 5x5x1 image being convoluting into 3x3x1 convolved feature by 3x3x1 kernel/filter [8]

The movement pattern of the Kernel can be visualized as shown in Figure 2.3.4. There are total of 9 shifts happened during convoluting the input image. The Kernel will always move to right with a certain Stride value until it completed to the most width. After that, it will jump down with begin from the left side of the image with same Stride value length of 1. This kernel movement pattern will be continued until traversed entire image [8].

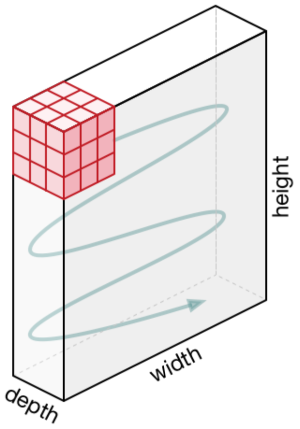
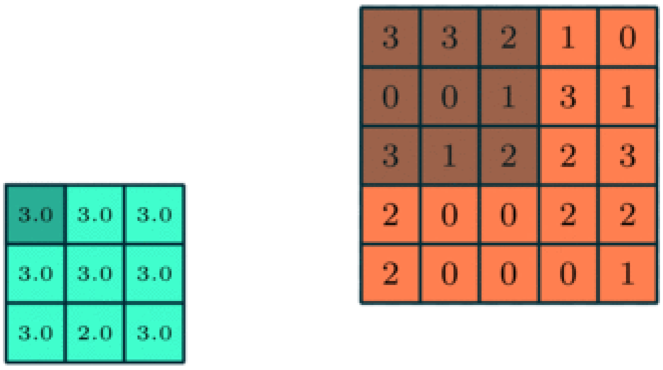


Figure 2.3.4 Movement pattern of the Kernel when traversing the input image [8]

High-level features extraction is the objective of the Convolution Operation. Normally, first convolution layer is responsible to capture Low-level features such as color, edges and gradient orientation. Along with more added layers, the architecture has successfully adapted to capture High-Level features which in result to a network that have deep understanding the images in dataset. The operation may produce two types of results which are convolved feature reduced in dimensionality compared to input. The other one is the dimensionality either remain same or increased. Valid Padding will apply to the one with dimensionality reduced and Same Padding will be applied to the dimensionality that either increased or remain the same [8].

The Pooling Layer

The way pooling layer worked will be similar to convolutional layer as well shown in Figure 2.3.5. Pooling layer is targeted to reduce spatial size of convolved feature. This implementation is to decrease computational power used for data processing via the reduction in dimensionality. Moreover, extracting dominant features such as positional and rotational invariant would be useful in this layer as it can maintain the model training effectiveness.



3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Figure 2.3.5 A 5x5 convolved feature get pooling over by 3x3 [8]

Max Pooling aimed to returns maximum value from the image part that traversed by the Kernel. Besides that, Max Pooling also worked as a Noise Suppressant. Its job is to expel those noisy activations and performs de-noising with reduction in dimensionality as well. Other the other hand, Average Pooling is to return average of all values from image portion that traversed by the Kernel. Average Pooling can be said it is work similar as the Max Pooling in noise suppressing mechanism [8]. However, Max Pooling can perform much better than Average Pooling. Both the Pooling layer types can be illustrated as shown in Figure 2.3.6. With the above process, the training model is able to understand the features of the input image. Further on, the final output needs to be flattened and feed to Neural Network for classification.

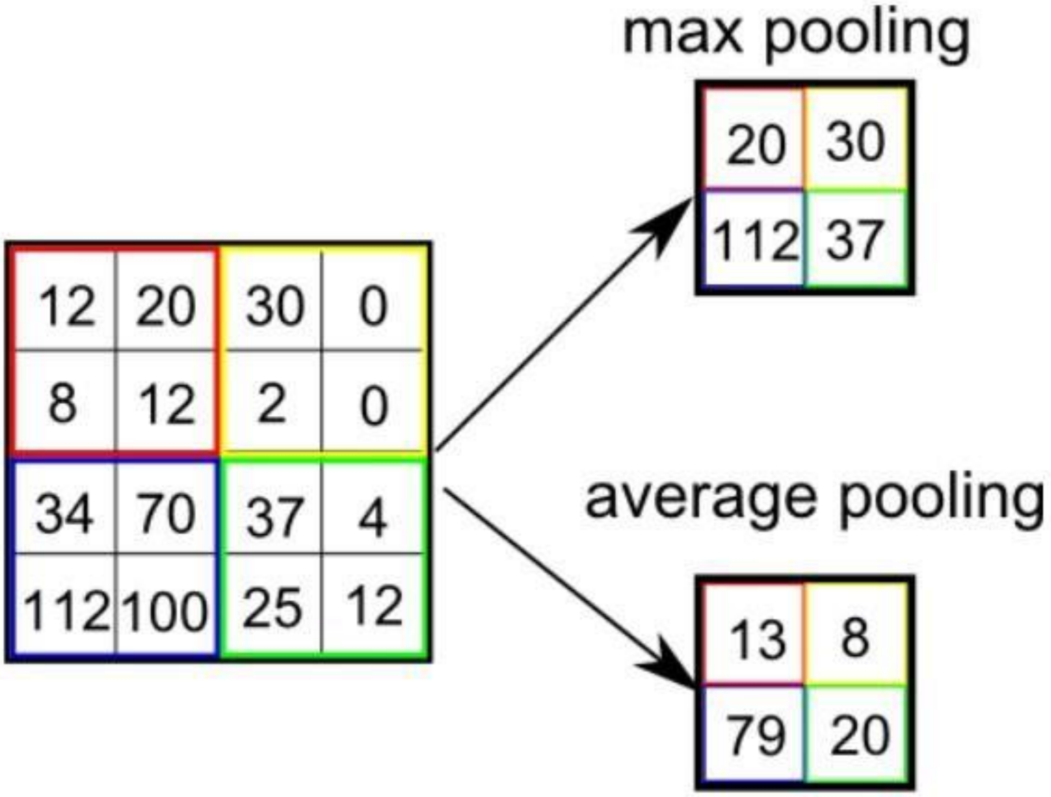


Figure 2.3.6 Comparison between Max Pooling and Average Pooling [8]

The Classification (Fully Connected Layer, FC Layer)

Implementation of adding a Fully-Connected layer is normally categorized as low cost for learning the high-level features non-linear combinations that represented by convolutional layer output. Other than that, a possibly non-linear function could be learned by the Fully-Connected layer.

Converted input image shall be flatten into a column vector. Feed-forward neural network needs to feed with the flattened output. Each iteration of the training can be applied with the backpropagation. The training model is capable to distinguish between low-level features and dominating features in images over the series of epochs. The Softmax Classification technique is used to classify features in distinct result shown in Figure 2.3.7 [8].

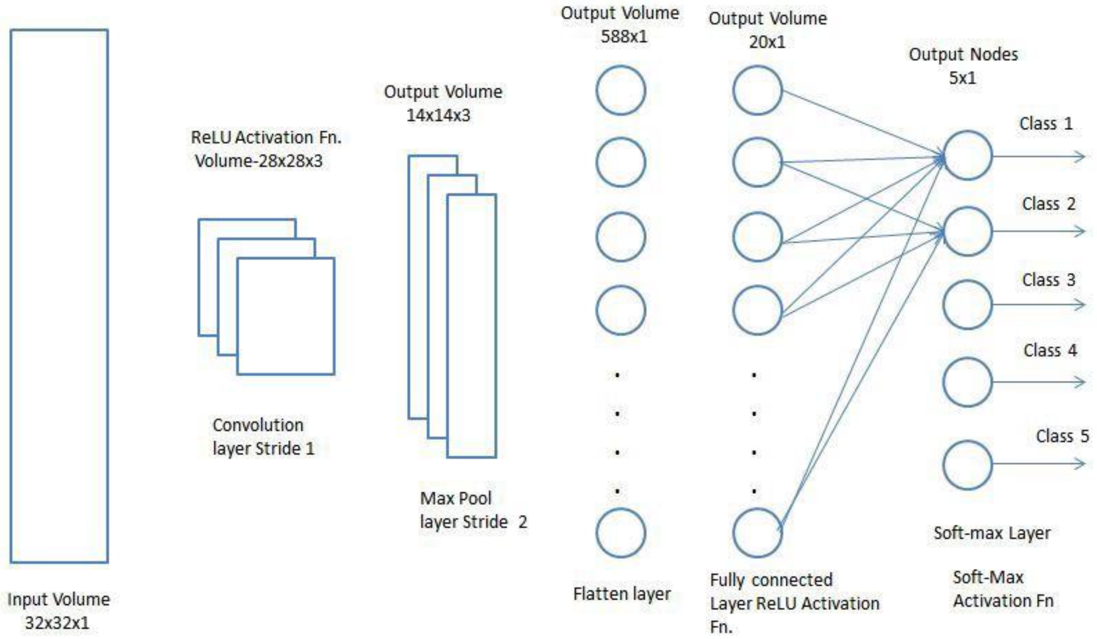


Figure 2.3.7 Clearer picture of Fully-Connected layer and Softmax classification activation layer [8]

2.4 MobileNet

Howard et al, 2017 have introduce the MobileNets in their research paper titled "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". Their paper proposed a class of network architectures which allowed model developer to choose small network that matched resource restrictions in terms of size and latency for developing their application. MobileNets that they proposed focus on latency optimization, yet small networks are possible to yield.

The core layers built in MobileNet are the depthwise separable convolution. The depthwise separable convolutions have significantly reduces parameters number compared others regular convolutional neural network with the same depth. Hence, MobileNet can be resulted in a lightweight deep neural network. Pointwise and Depthwise convolution are the two operations in a depthwise separable convolution [9].

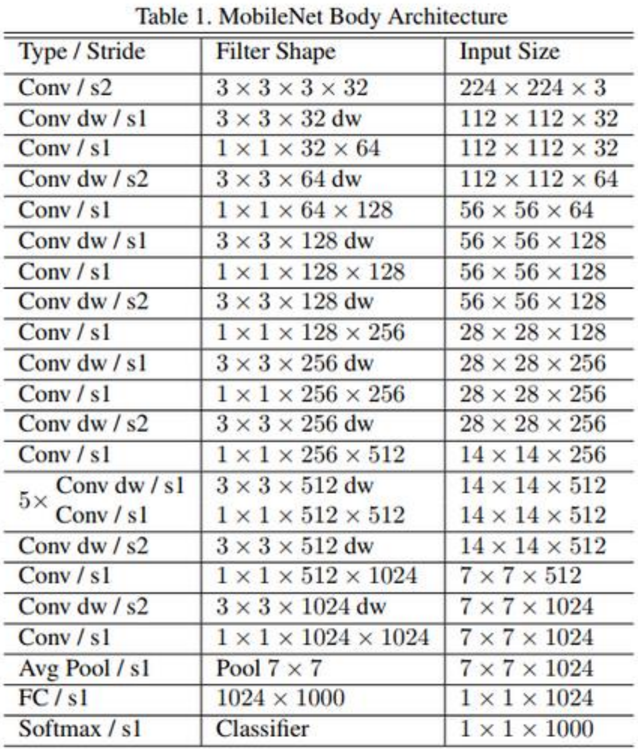
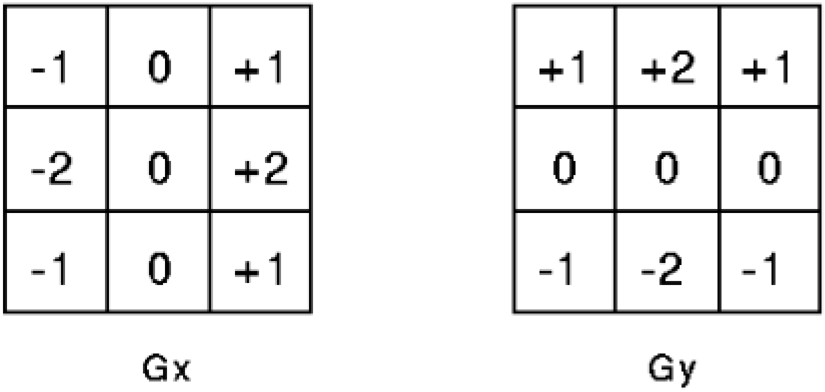


Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1 Conv / s1	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Figure 2.4.1 Body architecture of the MobileNet [9]

The depthwise separable convolution is based on the idea of a filter’s spatial and depth dimension which can be separated [10]. A Sobel filter can be an example for the convolution shown in Figure 2.4.2. Filters’ height and width dimensions can be separated. The Gx filter can be perceived as matrix product of [1 2 1] that transposed with [-1 0 1]. The filters itself would have 6 parameters instead of 9 due to the height and width dimension separation. This is the same idea that can be applied to the depthwise separable convolution.



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 2.4.2 Gx, for vertical edge & Gy, for horizontal edge detection. A Sobel Filter [10]

Figure 2.4.3 can show a clear picture on how the structure of depthwise separable convolution where depthwise convolution is followed by pointwise convolution.

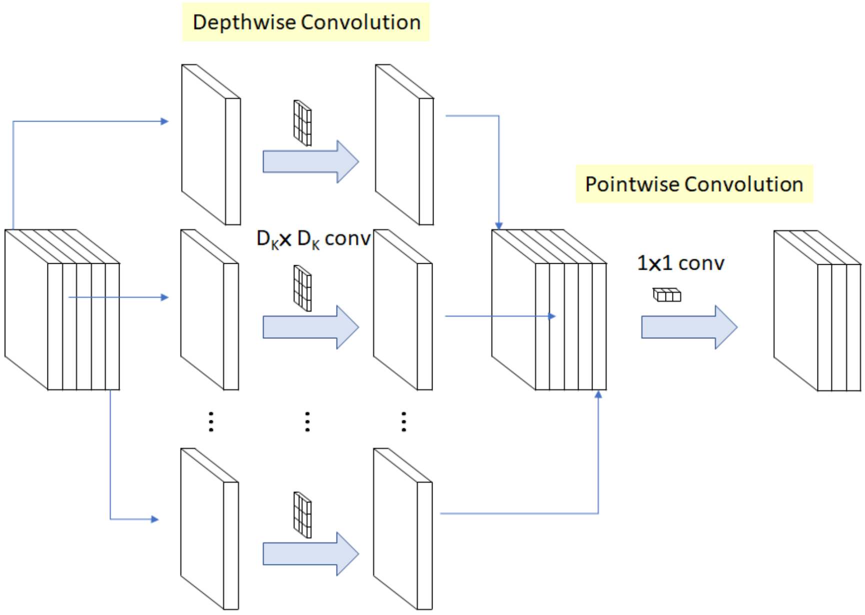


Figure 2.4.3 Depthwise Convolution and Pointwise Convolution [10]

The channel-wise D_kxD_k spatial convolution is the depthwise convolution. On each input channel, there will be a map of single convolution separately. Hence, number of input channels should be the same as the output channels number shown in Figure 2.4.5 [9]. The formulated computational cost is:

$$Df^2 * M * D_k^2$$

Figure 2.4.4 Depthwise convolution computational cost formula

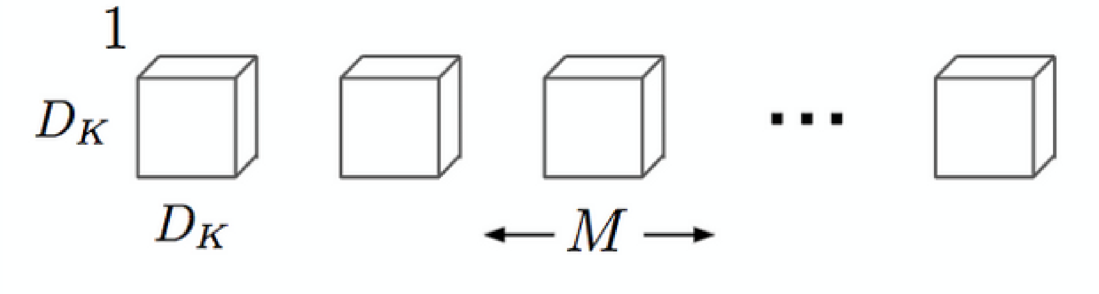


Figure 2.4.5 Depthwise Convolution block structure [9]

The depthwise convolution have created the pointwise convolution with a kernel size of 1x1 which simply combines the features shown in Figure 2.4.7 [9]. The formulated computational cost of it will be:

$$M * N * Df^2$$

Figure 2.4.6 Pointwise convolution computational cost formula

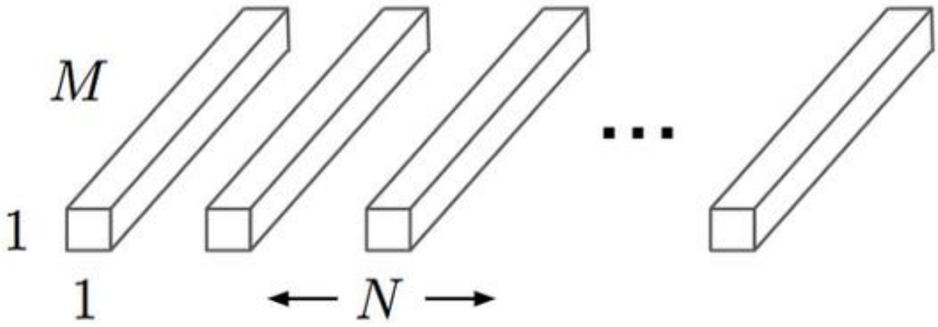


Figure 2.4.7 Pointwise convolution block structure [9]

The comparison between a standard convolutional layer with batch normalization and ReLU and the MobileNet architecture is the main difference in the separation of the convolution. MobileNet architecture is targeted to separate convolution into a 1x1 pointwise convolution and a 3x3 depthwise convolution shown in Figure 2.4.8 [9].

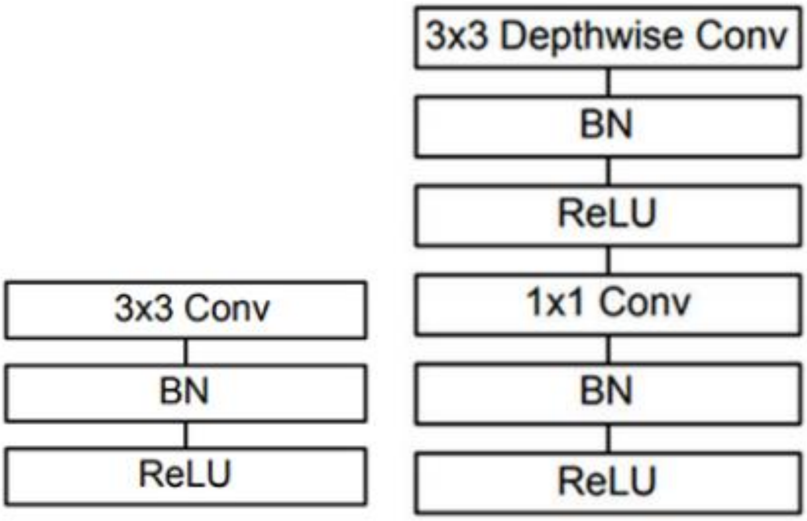


Figure 2.4.8 Differences between Standard CNN and MobileNet [9]

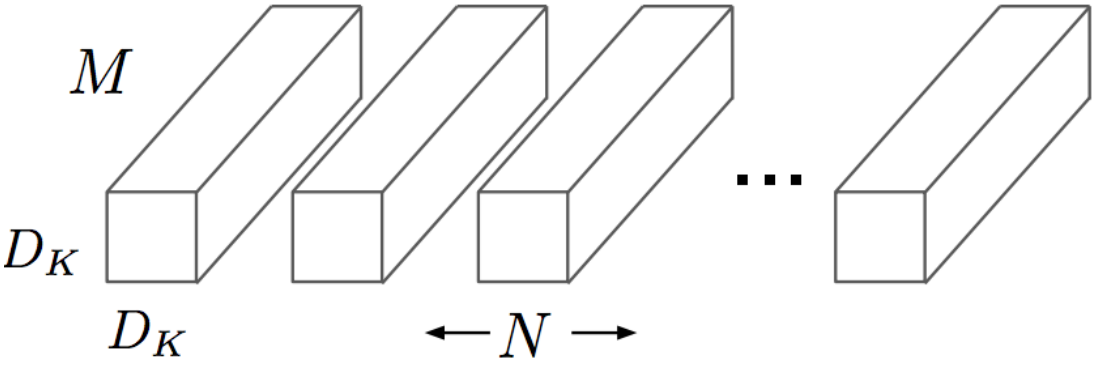


Figure 2.4.9 Standard convolution filters [9]

2.5 EfficientNet

Quoc V.Le and Mingxing Tan are the researchers from Google Research, Brain team proposed EfficientNet model in their research paper titled “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. Moreover, their research paper has presented in International Conference on Machine Learning in the year of 2019. Both researchers devoted themselves in studying model scaling and customizing the network “EfficientNet” in terms of depth, width, and resolution to reach better performance [11].

Based on their study, a new scaling method is proposed which achieve to uniformly scales depth, width and resolution all dimensions of the network. Furthermore, a new baseline network can be design and able to scale it in obtaining deep learning model family based on the use of neural architecture search [11]. The family of deep learning models called EfficientNets, capable to perform better accuracy and efficiency by comparing to previous Convolutional Neural Networks.

Other than that, both researchers acknowledged the compound scaling method used in their work is the method that can significantly improve the model’s efficiency and accuracy. Compound scaling method is mainly to scale the network dimensions. In this method, the strategy of applied grid search can reveal the relationship between the baseline network with different scaling dimensions under a fixed resource constraint [11]. Both researchers are able to find the appropriate scaling coefficients for each of dimensions to scaled-up by implementing this strategy. Hence, the baseline network scaled by desired size by using these scaling coefficients.

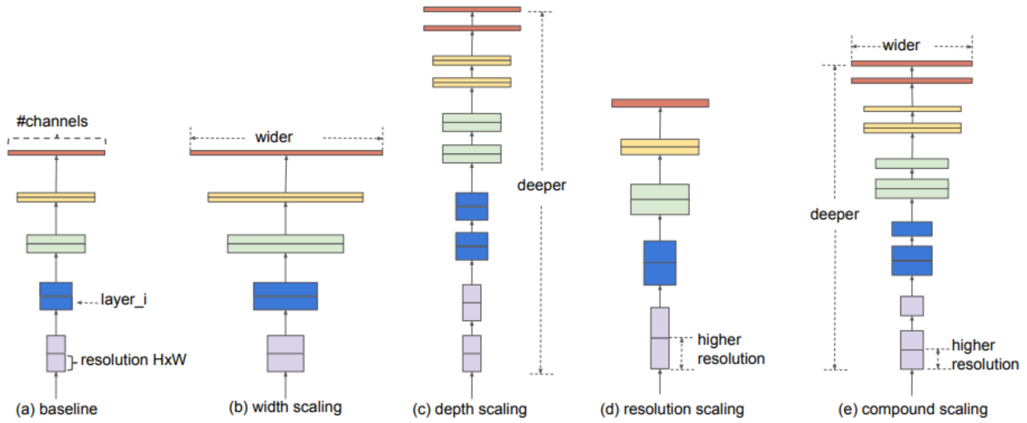


Figure 2.5.1 EfficientNet dimension scaling by applying with compound scaling [11]

The architecture of EfficientNets is based on the automation technique of neural network design which performed as neural architecture search to design the initial baseline network. The baseline network optimized both efficiency and accuracy which it measured on floating-point operations per second (FLOPS) basis. Mobile inverted bottleneck convolution (MBConv) implemented by the developed architecture [11]. Hence, with baseline network scaled up realized the deep learning models.

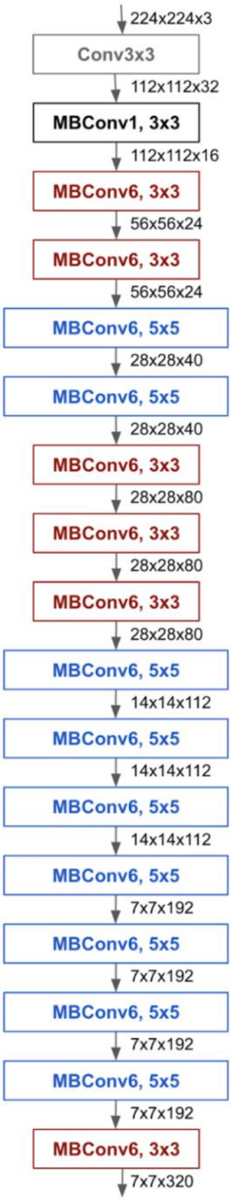


Figure 2.5.2 EfficientNet architecture [11]

Besides of that, both researchers have presented the performance comparison between EfficientNets and the other powerful transfer learning models with the same weight of working on ImageNet dataset [11]. EfficientNet-B7 shown the EfficientNets capabilities of achieving highest accuracy among other powerful learning models with lesser number of parameters shown in Figure 2.5.3.

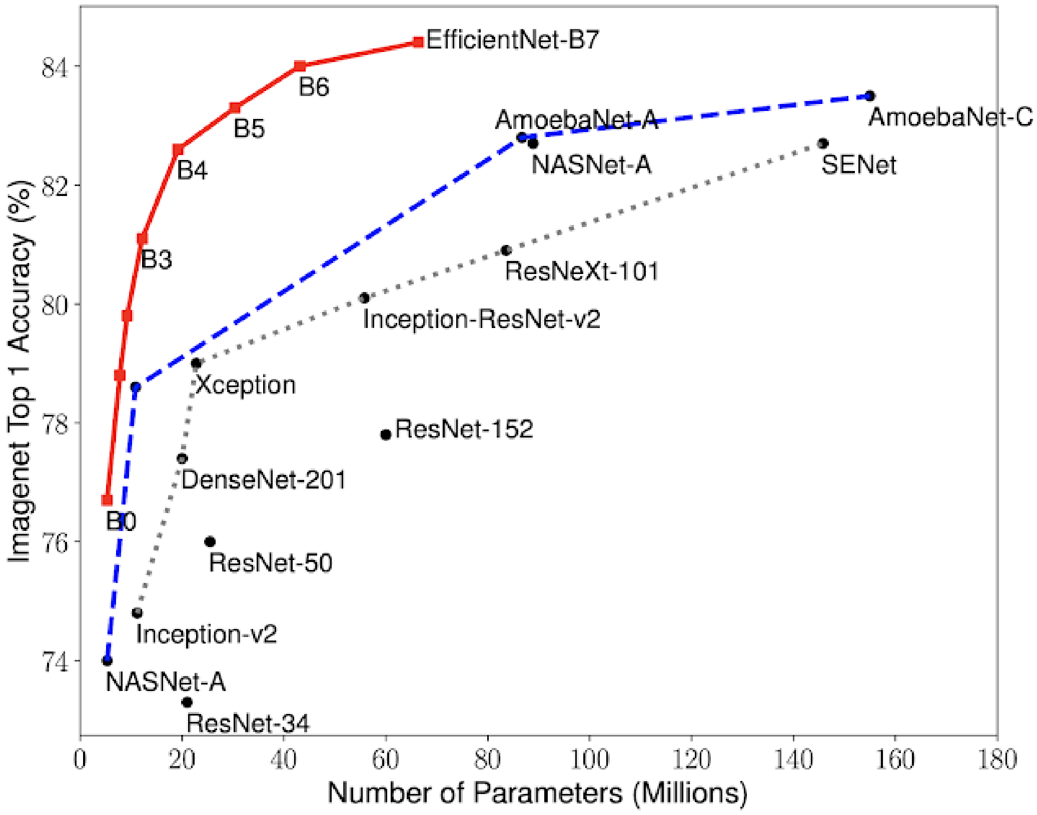
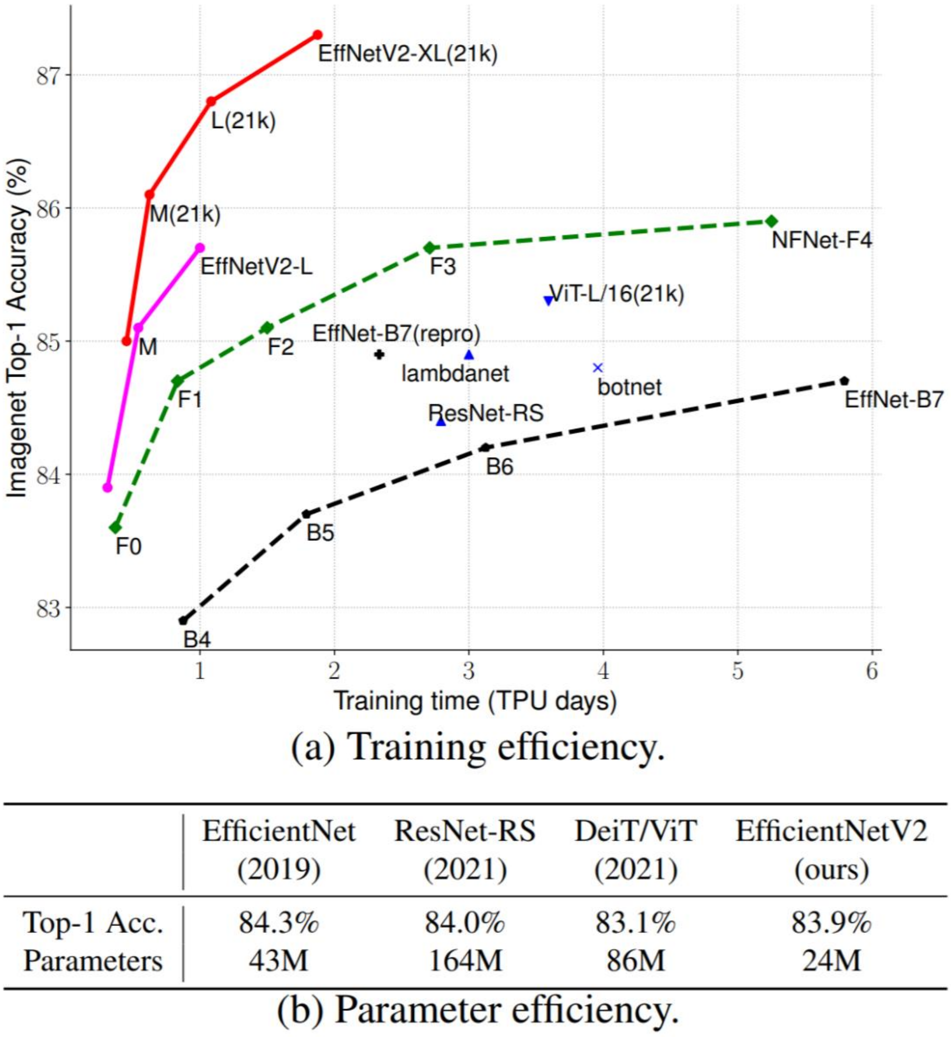


Figure 2.5.3 Performance difference between EfficientNets and other transfer learning model on working with ImageNet dataset [11]

After the huge success of EfficientNet in 2019, Quoc V.Le and Mingxing Tan proposed their research paper to introduce the EfficientNetV2 as a new family of convolutional networks that have faster training speed and better parameter efficiency than previous models from EfficientNetB0 to EfficientNetB7 [12].

In their research paper, it introduced new EfficientNetV2 architecture, and an improved progressive learning method is proposed which image size is to adjust the regularization. In Figure 2.5.4, EfficientNetV2 architecture compared to previous version EfficientNet and other network to have 11 times faster training speed and around 6.8 times better parameter efficiency [12].



(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Figure 2.5.4 Performance comparison between new EfficientNetV2 architecture and other network architecture [12]

For EfficientNetV1 architecture, the depth-wise convolutions in MBConv have fewer parameters and its FLOPs higher than other regular convolutions. However, it did not fully utilize the modern accelerators which stating that FLOPs reduction does not necessarily lead training speed improvement. EfficientNetV2 architecture utilized Fused-MBConv to replace depth-wise 3x3 convolution and use regular 3x3 convolution to replace the expansion 1x1 convolution in MBConv shown in Figure 2.5.5. In Figure 2.5.6, the performance improvement and two building blocks comparison shown the difference between Fused-MBConv and MBConv (No fused). Fused-MBConv that applied from stages 1 to 3 can improve the training speed with lower FLOPs and lesser overhead parameters. Moreover, there is no literal answer of which kind of situation utilize MBConv or Fused-MBConv. The researchers suggested that it is better to search for best combination [12].

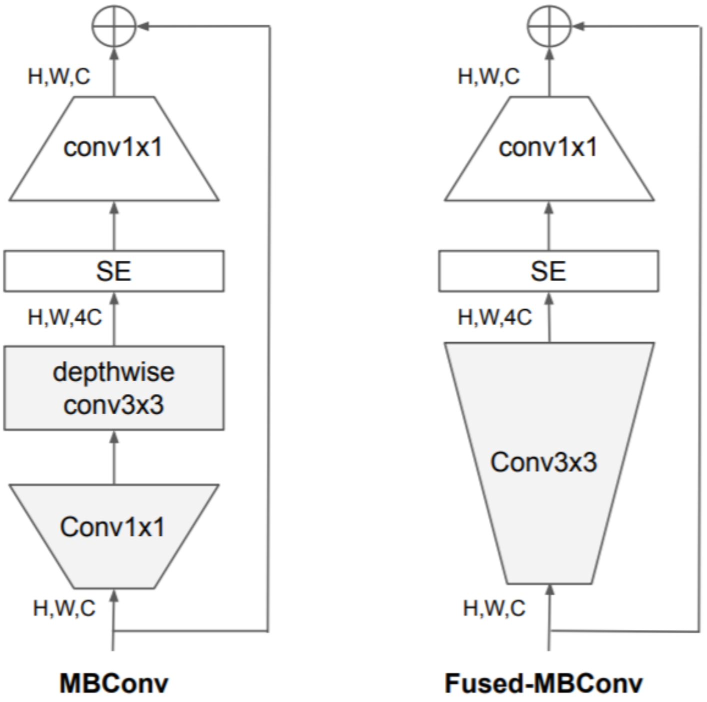
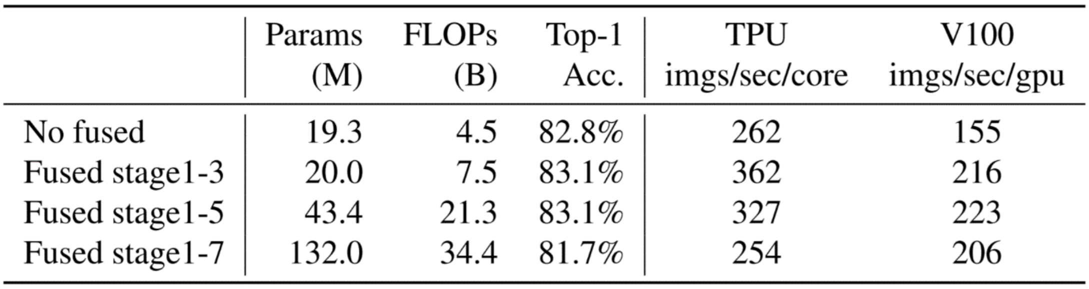


Figure 2.5.5 Fused-MBConv compare to MBConv [12]

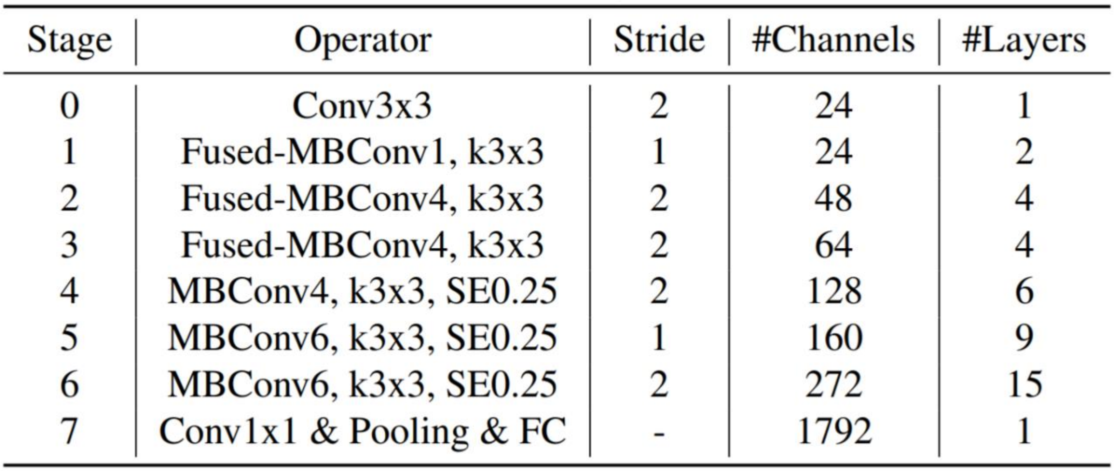


	Params (M)	FLOPs (B)	Top-1 Acc.	TPU imgs/sec/core	V100 imgs/sec/gpu
No fused	19.3	4.5	82.8%	262	155
Fused stage1-3	20.0	7.5	83.1%	362	216
Fused stage1-5	43.4	21.3	83.1%	327	223
Fused stage1-7	132.0	34.4	81.7%	254	206

Figure 2.5.6 Comparison between MBConv(No fused) and Fused-MBConv [12]

Other than that, it is sub-optimal to equally scaling up every stage using simple compound scaling rule in EfficientNetV1 architecture. The scaled stages are not equally contributed to efficiency of parameter and faster training speed. Hence, both researchers implemented a non-uniform scaling strategy to gradually add more layers for the EfficientNetV2 architecture [12].

In EfficientNetV2 architecture, the researchers implemented both MBConv and Fused-MBConv in the early layers. Other than that, EfficientNetV2 use 3x3 kernel sizes instead of 5x5 as in EfficientNetV1. The last stride-1 stage in EfficientNetV1 is completely removed implemented in EfficientNetV2 architecture [12]. These settings of EfficientNetV2-S architecture can be shown in Figure 2.5.7.



Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	272	15
7	Conv1x1 & Pooling & FC	-	1792	1

Figure 2.5.7 EfficientNetV2-S architecture [12]

2.6 Detection of Banana Leaf and Fruit Diseases Using Neural Networks

Brief

Most country populations are greatly depended on their agricultural production. However, the existences of the plant diseases which possibly transmitted by the insects that caused their agricultural farm yields greatly reduce. Therefore, they realized that they are required an advanced and automated techniques which enable them to resolve this problem by having an early-stage diagnosis method and help them to greatly reduce their agricultural yields loss. Other than that, they also realized the knowledge in IT field is one of the effective keys to implement such automate technique. Image processing used in banana plant classification and plant diseases detection are effective and with minimum cost to help analyses plant growth automatically and effectively. To resolve those stated problems, [13] Saranya et al. proposed a system which capable in detect the diseases at the early-stage health of the banana tree by using the image processing and able to classify diseases via the artificial neural networks (ANN) algorithm. Furthermore, their proposed system consists of several important steps such as image acquisitions, image pre-processing, feature extraction, detection of diseases and the diseases classification by the artificial neural network [13].

Strengths

The proposed system by [13] Saranya et al. have implemented the impressive and effective image pre-processing and image segmentation steps to reach accurate disease classification. In their image pre-processing part, they used the histogram based equalization which able to enhance images by tuning the images intensities such as the brightness of the images. Therefore, this histogram based equalization provides better image quality and helps distinct distinguish of affected area of banana leaf and fruit images after any previous transformation steps taken in the early image pre-processing steps. In image segmentation part, they applied the fuzzy c-means clustering technique which helps them to segment out the homogeneity and heterogeneity of the banana fruits and leaves images by clustering the data as one cluster.

Weaknesses

The treatment recommendations in their proposed system after detected the diseases are lack of further information on the disease symptoms and guidelines which may cause the users have misinterpretation on the solution they provided. As it may not be an effective system for helping the users about how they can resolve the problem by using the proposed system by Saranya et al.

Recommendations

Saranya et al. (2020) are encouraged to take the consideration of their proposed system should include more explanation on the disease detected and further recommendation on the treatment methods such as the chemical control guideline information, cultivating control measures information, any further elaborations and more banana expert’s treatment ideas.

2.7 Mobile-based Deep Learning Models for Banana Disease Detection

Brief

Smallholder banana farmers in most of the country are the important personnel that contribute to the banana production. Unfortunately, their banana production is threatening by the existence of pests and diseases. Hence, they acknowledged that banana diseases early detection is important to identify the banana diseases before their productions get damaged and reduce in yields. [14] Sanga et al. proposed to build a tool aimed for banana disease early detection via deep learning approach. For their deep learning approach, they used five different deep learning architectures which are Vgg16, Resnet18, Resnet50, Resnett152 and InceptionV3 to implement the models for the banana disease detection. Their InceptionV3 model able to achieve high accuracy as 95.41% and Resnet152 was 99.2% of accuracy. [14] Sanga et al. decided to use InceptionV3 for the mobile development since InceptionV3 required less memory. Their developed mobile application having a confidence of 99% in detecting capabilities for a captured leaf from the real environment. Therefore, their developed mobile application able to help the smallholder banana farmers for an early detection of the banana diseases which also help in improving their productivity [14].

Strengths

The mobile deployment proposed by [14] Sanga et al. achieved a greater coverage on different Deep Convolution Neural Network models. These models help them in deciding which models to be used in the mobile development. They also used transfer learning techniques which the model developed for a task can be reused as for a initiate point for the model on second task. Therefore, they selected InceptionV3 as for their mobile deployment since it is performed greatly with a low computational cost. Other than that, they also used the augmentation technique for their image data generator in the image pre-processing step which enhance their images for the better training outcome.

Weaknesses

However, the mobile development tool proposed by [14] Sanga et al. are limited in provide any form of information on the treatment for encountering the diseases and pest damages. Furthermore, their proposed mobile application was unable to be found on the internet and Google Play Store which indicate their proposed system may have fault and flaws.

Recommendations

[14] Sanga et al. are recommended to includes more treatment information which is important in the mobile application for the banana farmers to help them in curing their banana tree. Sanga et al. should give a statement on which platform they had published their mobile application as it is important to allow the users and farmers to utilize their system and to help their banana plantation in early-stage disease diagnosis.

2.8 AI-powered system for banana diseases and pest detection

Brief

Smallholder farmers are always in the risk of facing the abiotic and biotic constraints for their crops such as banana. The pests and diseases of banana have caused a huge amount of destruction towards the farmers financial and yield losses across their owned plantation which consecutively bringing a significant threat to the global food security. It is no doubt that farmers are required to take the first crucial step which is the diseases and pests early-stage detection in their plantation. However, most of the smallholder farmers doesn't equipped with advanced knowledge on the bio-chemical structuring of their crops where they only can rely on their empirical knowledge which also leads them in having a hard time to overcome any farming challenges. Hence, an early detection of those pests and diseases able to reduce the impacts on the yields losses as well as food supply chains. [2] Selvaraj et al. proposed the idea of considering the use of Artificial intelligence (AI) with the deep learning models that capable in identify plant diseases by providing the plant's visual symptoms and appearance. Other than that, [2] Selvaraj et al. also take the consideration on the smartphone-based AI apps which aimed to alert the farmers and executing the disease diagnosis which results possible prevention of diseases and pests outbreak. Therefore, [2] Selvaraj et al. proposed the study was to claim the state-of-the-art deep learning techniques in detecting the visible banana disease and pest symptoms on the banana tree distinct parts by considering the adaptation of pre-trained deep learning CNN models that using large dataset of plant experts' pre-screened images collected from India and Africa.

Strengths

The study proposed by [2] Selvaraj et al. have a stronger stand and advantages as they considered six different models and 18 different classes that aimed to achieve maximum accuracy in their Deep Transfer Learning (DTL) system dataset. The DTL approach is to generates new framework for predictive analytics and digital image processing, that perform in greater accuracy and great potential in plant disease detection. Furthermore, DTL provides a convincing avenue for disease identification by using huge amount of trained image datasets and have a fast access to developed models offered by mobile applications. Furthermore, the data labelling step in proposed study by [2] Selvaraj et al. apply the image tagging process via LabelImg software which this software can be found in a GitHub. Moreover, [2] Selvaraj et al. have a great focus on the performance metrics such as loss function, mAP score and confusion matrix. The object detection API used in study by [2] Selvaraj et al. is to measure model performance by estimate the total loss function. The mean average precision (mAP) is act as the validation metric for both disease and pest banana detection. The confusion matrix (CM) in this study is computed based on the object detection script, in each detection algorithm consider the ground-truth boxes and classes with detected boxes, classes and scores of Intersection over Union (IoU) with detection score of more than 0.5 as the threshold. In this study, [2] Selvaraj et al. applied the Single Shot Multi-box (SSD) model selected with the MobileNetV1 which this combination considered the fastest object detection models.

Weaknesses

This study lack of the model usage distribution by training the model for the banana disease identification on a wider field for the application. Furthermore, this study may not be ready to explore in other recognized plant diseases as this study did the DTL as using for generating fresh framework.

Recommendations

[2] Selvaraj et al. are encouraged to consider the broad structure development that consists of the server-side machinery that contain trained model and mobile application with features such as displaying different identified diseases in other crops. Furthermore, [2] Selvaraj et al. also encouraged to merges the aerial images of the banana growing plantation regions which captured from the air by drones or any vehicles with the convolution neural network for multiple diseases instant segmentation.

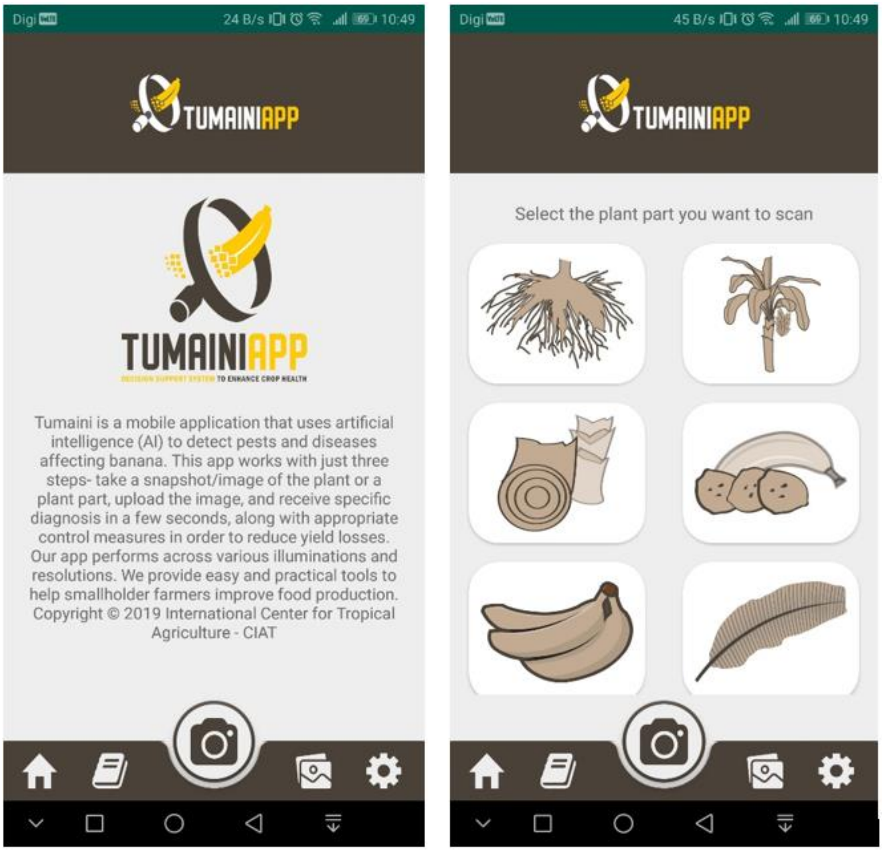
2.9 Existing Mobile Application Review

2.9.1 Tumaini App

To potentially protect the banana production yields from the outbreak extent of pests and diseases, an advanced tool is important to help those smallholder farmers to overcome the farming challenges they faced. The Figure 2.9.1.1 depicts the interfaces of the Tumaini App, a mobile application that capable in linking both artificial intelligence and the competence on banana genetic information which can classify the pest or disease that threatening the banana plant.

Moreover, this mobile application allowing smallholder farmers increase their productivity in detecting their banana crops health status. Moreover, this mobile application is a collaboration product by the International Center for Tropical Agriculture (CIAT) and CGIAR researchers from the Alliance of Bioversity International. The name of the mobile application “Tumaini” is means “hope” in Swahili, Swahili is a native language used in Kenya, Uganda and Tanzania. This mobile application is symbolized as a hope that help smallholder farmers in scan their plants to detect out the possible diseases from major five diseases and one common pest which provided in the Tumaini App [15].

Tumaini App is a mobile application used the models that trained in CNN architecture, a deep learning approach to detect the diseases and pests [16]. Other than that, the researchers used the transfer learning technique to retrain the three different convolutional neural network architectures that have been deployed into the Tumaini App [17].



Tumaini is a mobile application that uses artificial intelligence (AI) to detect pests and diseases affecting banana. This app works with just three steps- take a snapshot/image of the plant or a plant part, upload the image, and receive specific diagnosis in a few seconds, along with appropriate control measures in order to reduce yield losses. Our app performs across various illuminations and resolutions. We provide easy and practical tools to help smallholder farmers improve food production. Copyright © 2019 International Center for Tropical Agriculture - CIAT

Select the plant part you want to scan

Figure 2.9.1.1 Tumaini App mobile

Strengths

One of the advantages of using the Tumaini App is that users are allowed to scan or upload different part of the banana plant to have a disease detection diagnosis. The parts detection category provided by the Tumaini App are such as the fruit, bunch and leaf [17]. For the interface of plant part detection, it is referring the right-side image in the Figure 2.9.1.1. The other advantage is that the deep learning models in the application is trained to read the low-quality images and still maximize the detection accuracy [17].

Weaknesses

Even though Tumaini App is claimed to achieve an accuracy about 90% in the detection rate. There is a limitation observed during the detection of a healthy banana leaf where the Tumaini App was not able to give feedback and any result towards the banana leaf health status which illustrated in Figure 2.9.1.2.

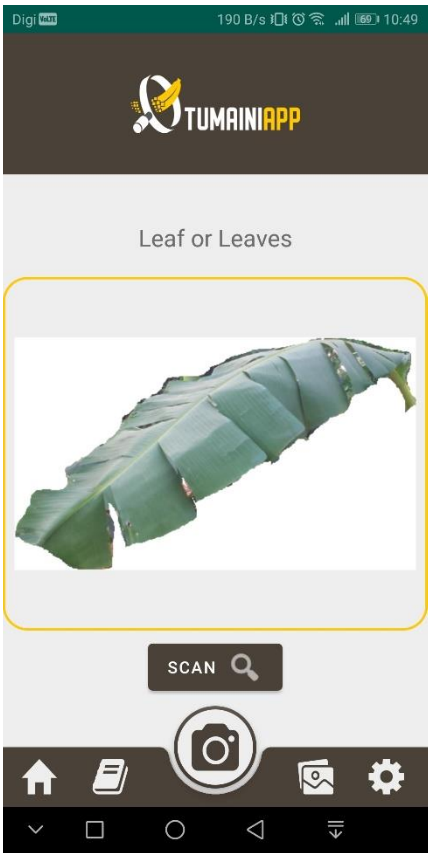


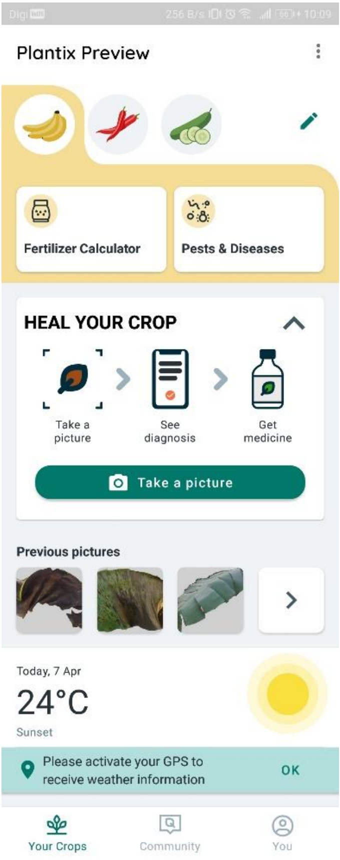
Figure 2.9.1.2 Tumaini App Leaf Part

Recommendations

The researchers are recommended to includes more images of the healthy banana leaf category into the dataset for further training. They are also recommended to enhance their data labelling part with more precise technique in pre-process their dataset images.

2.9.2 Plantix Preview – your crop doctor

Figure 2.9.2.1 depict the interface of Plantix Preview mobile application which is a multi-lingual plant disease and pest diagnostic tools developed by the German AI-based start-up company called Progressive Environmental & Agricultural Technologies (PEAT). This mobile application helps in classifying the plant damage with the technology used of machine learning [18]. Furthermore, this application used AI-powered image recognition for diagnose the pest damages, nutrient deficiencies and plant diseases which possibly threatening the crops [19]. The images uploaded by the users are then sent to their servers, images will be analysed by the PEAT's self-developed algorithms aimed for classifying out possible disease and generates the results such as symptoms, agro-chemicals or biological treatments to users. Moreover, this app provided information of weather system which is specific to the users' location [18].

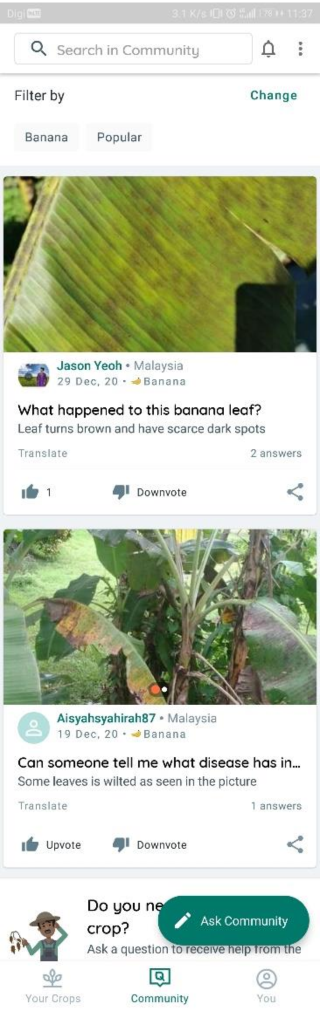


Please activate your GPS to receive weather information

Figure 2.9.2.1 Plantix Preview – your crop doctor mobile application

Strengths

One of the strengths of the Plantix Preview mobile application is the community feature that allowed the farmers and users to seek more helps than uploading the photo of plant for scanning the diseases or pests. The farmers and users are allowed to give comments, upvote, downvote and share out the questions externally which can be depict in Figure 2.9.2.2.



Jason Yeoh • Malaysia
29 Dec, 20 • Banana

What happened to this banana leaf?
Leaf turns brown and have scarce dark spots

Translate 2 answers

1 Downvote

Aisyahsyahirah87 • Malaysia
19 Dec, 20 • Banana

Can someone tell me what disease has in...
Some leaves is wilted as seen in the picture

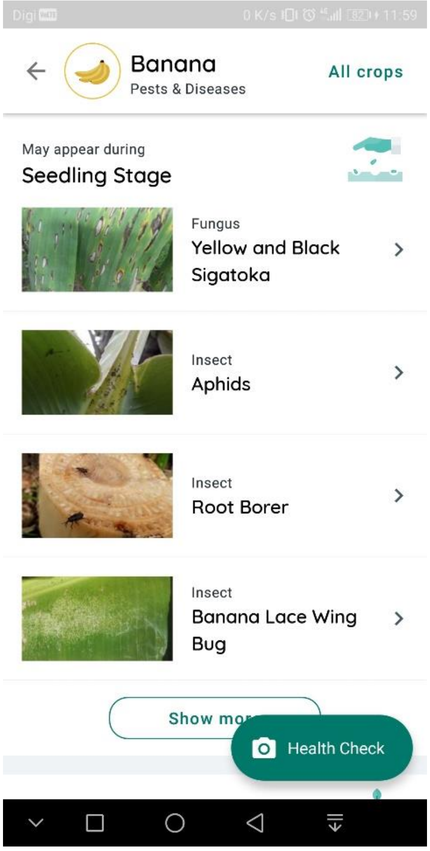
Translate 1 answers

Upvote Downvote

Do you ne
crop?
Ask a question to receive help from the

Figure 2.9.2.2 Community section interface in Plantix Preview Mobile Application

Another strength is in the Figure 2.9.2.3 which depict the crop guide interface for pests and diseases happening in each stage of the banana plant of the crop cycle. This crop guide feature provides possible pests and diseases that may occur during the stages of the crop cycle such as in seedling stage, vegetative stage, flowering stage, fruiting stage and harvesting stage. Therefore, this feature did help a lot for the farmers and users to controlling their crops health status whereby it includes symptoms information and treatment information. Moreover, with the help of PEAT’s software, every photo uploaded by users and stored into their servers which allowed PEAT to receive geodata of the photo. After that, the developers are able to map out the spread of a plant diseases and generates warning messages for the farmers and users to do the prevention steps [20].



Banana
Pests & Diseases

May appear during
Seedling Stage

Fungus
Yellow and Black Sigatoka

Insect
Aphids

Insect
Root Borer

Insect
Banana Lace Wing Bug

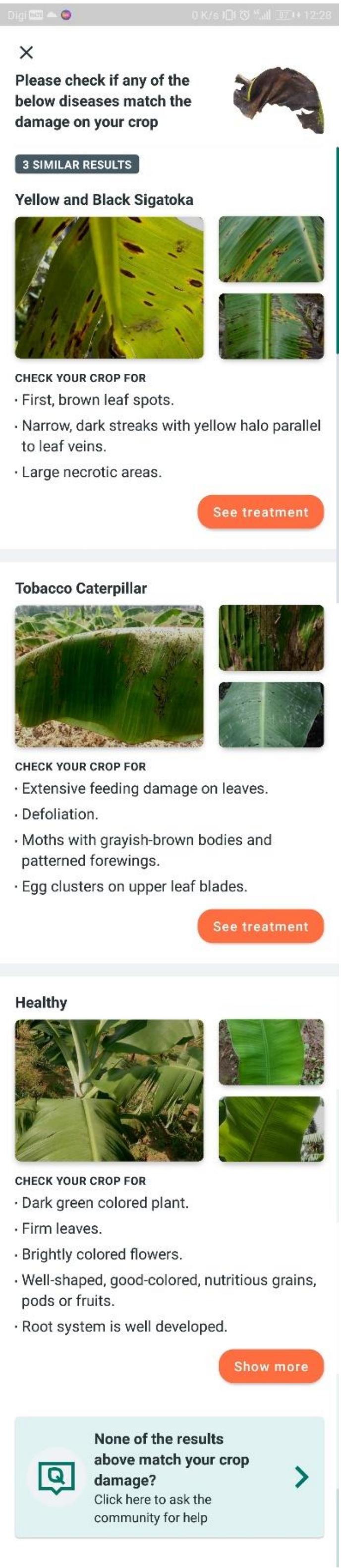
Figure 2.9.2.3 Crop guide interface in Plantix Preview Mobile Application

Weaknesses

There is a weakness found in this Plantix Preview mobile application which is a limitation in generate an accurate disease detection which is in Figure 2.9.2.4. The photo uploaded for classifying the disease is diagnosed as Xanthomonas Wilt of banana disease on the banana leaf. However, Plantix Preview unable to detect the accurate result instead it provides three different results such as yellow and black sigatoka, tobacco caterpillar and healthy. There are few possible reasons behind of this inaccurate detection result are due to lack of the disease for the leaf photos and dependency of total images in the database.

Recommendations

The developers of this application are encouraged to insert more image datasets of the bacterial wilt diseases and other significant diseases in banana plant. Rather than receive images from users uploads, developers are encouraged to request images from the website of Plant Village datasets and the banana experts who visited those banana farms to collect the photos to feeds into their database to increase the total images for developers' deep learning model training whereby it can increase the disease detection accuracy.



Please check if any of the below diseases match the damage on your crop

3 SIMILAR RESULTS

Yellow and Black Sigatoka

CHECK YOUR CROP FOR

- First, brown leaf spots.
- Narrow, dark streaks with yellow halo parallel to leaf veins.
- Large necrotic areas.

Tobacco Caterpillar

CHECK YOUR CROP FOR

- Extensive feeding damage on leaves.
- Defoliation.
- Moths with grayish-brown bodies and patterned forewings.
- Egg clusters on upper leaf blades.

Healthy

CHECK YOUR CROP FOR

- Dark green colored plant.
- Firm leaves.
- Brightly colored flowers.
- Well-shaped, good-colored, nutritious grains, pods or fruits.
- Root system is well developed.

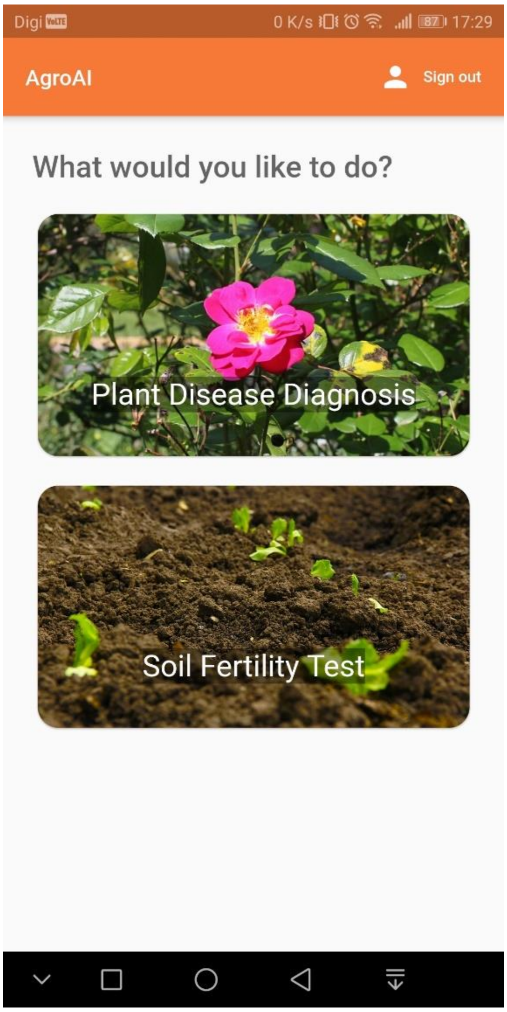
None of the results above match your crop damage?
Click here to ask the community for help

Figure 2.9.2.4 Diagnose results interface in Plantix Preview Mobile Application

2.9.3 AgroAI – Plant Diseases Diagnosis

AgroAI is a mobile application that dedicated to empowering farmers on aspects of plant disease diagnosis and soil fertility test. The latest version updated of this application is on 20 of November 2020. This application is offered by Neural Farms and it is still remain as an early accessible application available on Google Play Store.

With AgroAI, users are able to diagnose their plant's diseases via taking photo of affected plants' leaf. Moreover, this application offered users to check soil fertility and identifying various levels of different soil nutrients. Furthermore, AgroAI is an offline mobile application which aimed to reduce network dependency as it can reduce farmers network expenditures.

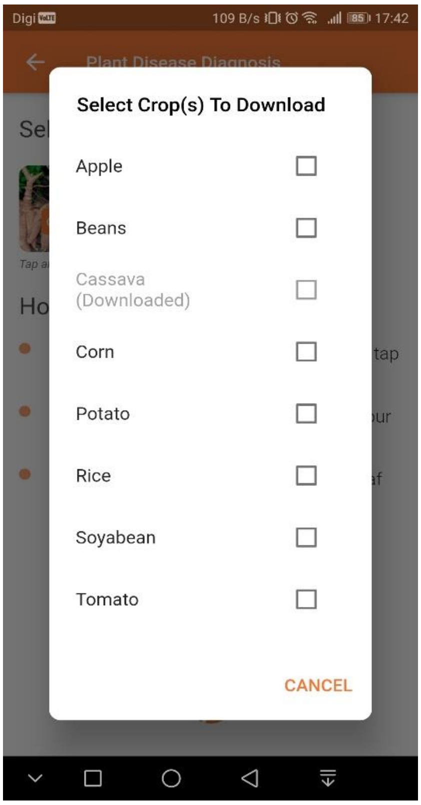


What would you like to do?

Figure 2.9.3.1 AgroAI mobile application main page

Strength

For the plant disease diagnosis feature, AgroAI provided eight different kinds of plant for diagnose the disease of the plants' leaf as shown in Figure 2.9.3.2. For example to diagnose cassava plant after uploaded picture to the application, the application will return the result of the diagnosed disease of the plant' leaf , disease description and possible prevention measures shown in Figure 2.9.3.3.



Select Crop(s) To Download

Apple □

Beans □

Cassava (Downloaded) □

Corn □

Potato □

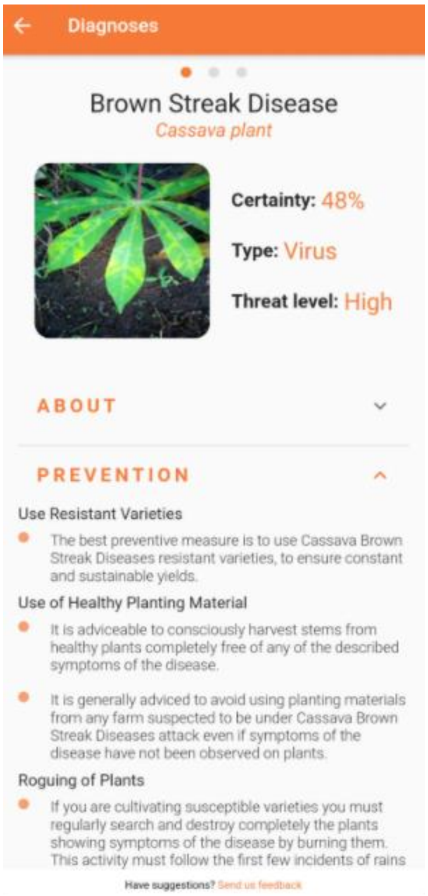
Rice □

Soyabean □

Tomato □

CANCEL

Figure 2.9.3.2 Plant Disease Diagnose Selection Interface



Brown Streak Disease

Cassava plant

Certainty: 48%

Type: Virus

Threat level: High

ABOUT

PREVENTION

Use Resistant Varieties

- The best preventive measure is to use Cassava Brown Streak Diseases resistant varieties, to ensure constant and sustainable yields.

Use of Healthy Planting Material

- It is adviceable to consciously harvest stems from healthy plants completely free of any of the described symptoms of the disease.
- It is generally adviced to avoid using planting materials from any farm suspected to be under Cassava Brown Streak Diseases attack even if symptoms of the disease have not been observed on plants.

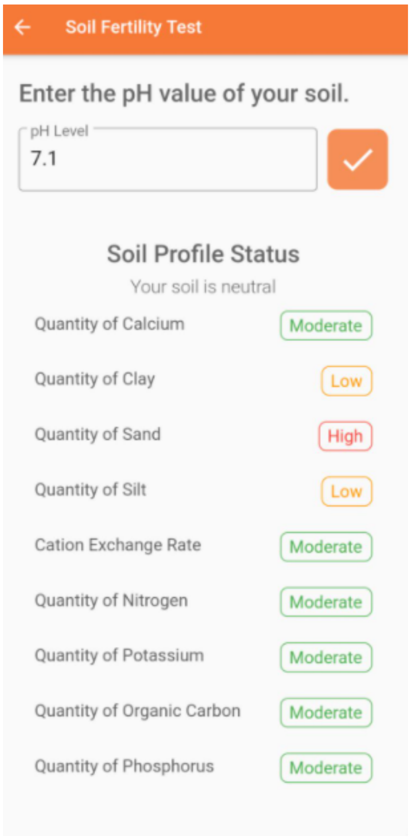
Roguing of Plants

- If you are cultivating susceptible varieties you must regularly search and destroy completely the plants showing symptoms of the disease by burning them. This activity must follow the first few incidents of rains

Have suggestions? Send us feedback

Figure 2.9.3.3 Diagnosed disease result of uploaded affected plant leaf of cassava plant

Other than that, AgroAI provided soil fertility test features that aimed to receive an input of pH level value of the soil. The result will return the soil profile status and different kinds of material and mineral quality existed in soil shown in Figure 2.9.3.4.



Enter the pH value of your soil.

pH Level: 7.1

Soil Profile Status

Your soil is neutral

Quantity of Calcium	Moderate
Quantity of Clay	Low
Quantity of Sand	High
Quantity of Silt	Low
Cation Exchange Rate	Moderate
Quantity of Nitrogen	Moderate
Quantity of Potassium	Moderate
Quantity of Organic Carbon	Moderate
Quantity of Phosphorus	Moderate

Figure 2.9.3.4 Soil Fertility Test interface

Weakness

The soil fertility test should feature does not provide recommended pH value soil for different kinds of plant.

Recommendation

The developer should consider the function of soil fertility test to include pH value recommendation for the soil of different kinds of plant as the application aimed to diagnose those eight kinds of plants. Other than that, provides method for users to improve their soil pH value to reach the recommended pH value corresponding to their plant.

2.9.4 Plant Disease Identification app

Plant Disease Identification app is a mobile application aimed to deliver various plants disease identification feature for users. This mobile application is offered by Fouxa and developed by Vikasdeep Singh. Fouxa is a platform that provides free educational content that aimed as a solution for students who have hardship searching for free educational resources [21].

Plant Disease Identification app is deployed with Machine Learning model to detect plant diseases via the plant leaf. Other than that, it provides variety of important prevention and control measures to aid the users for resolve their problems.

Strengths

This mobile application provided articles for tips in cultivating plants and crops which can benefit users' knowledge.

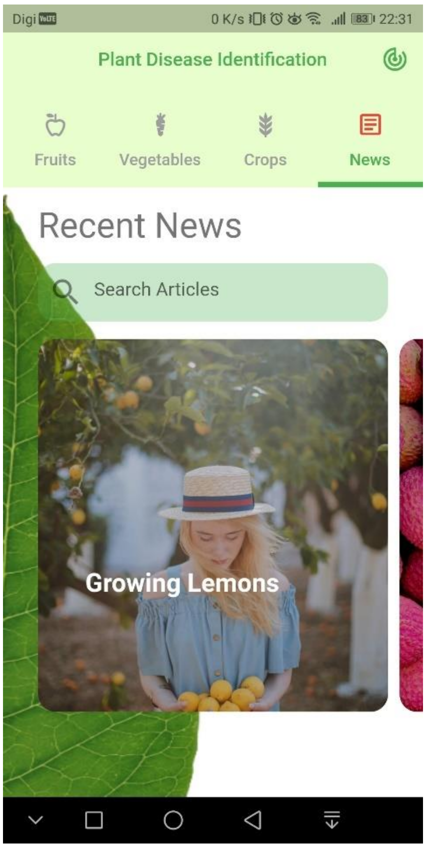
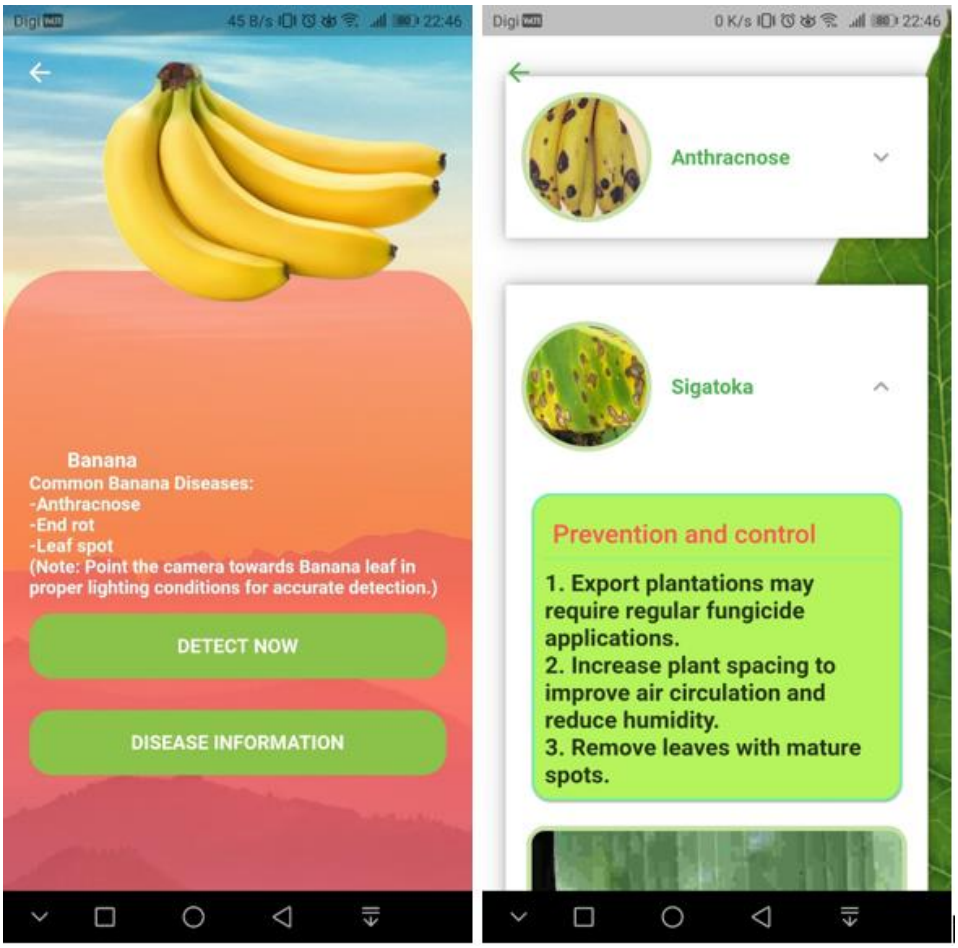


Figure 2.9.4.1 Article news for plants cultivating tips

Other than that, this application provides disease information for each available plant in three category such as crops, fruits and vegetables.



Banana
Common Banana Diseases:
-Anthracnose
-End rot
-Leaf spot
(Note: Point the camera towards Banana leaf in proper lighting conditions for accurate detection.)

Prevention and control

1. Export plantations may require regular fungicide applications.
2. Increase plant spacing to improve air circulation and reduce humidity.
3. Remove leaves with mature spots.

Figure 2.9.4.2 Banana fruit disease information

The machine learning offered in this application capable to classify diseases for fruits and vegetables category plant. There are total of 18 different fruits and 8 different vegetables which the machine learning model can handle and carry out the disease identification process.

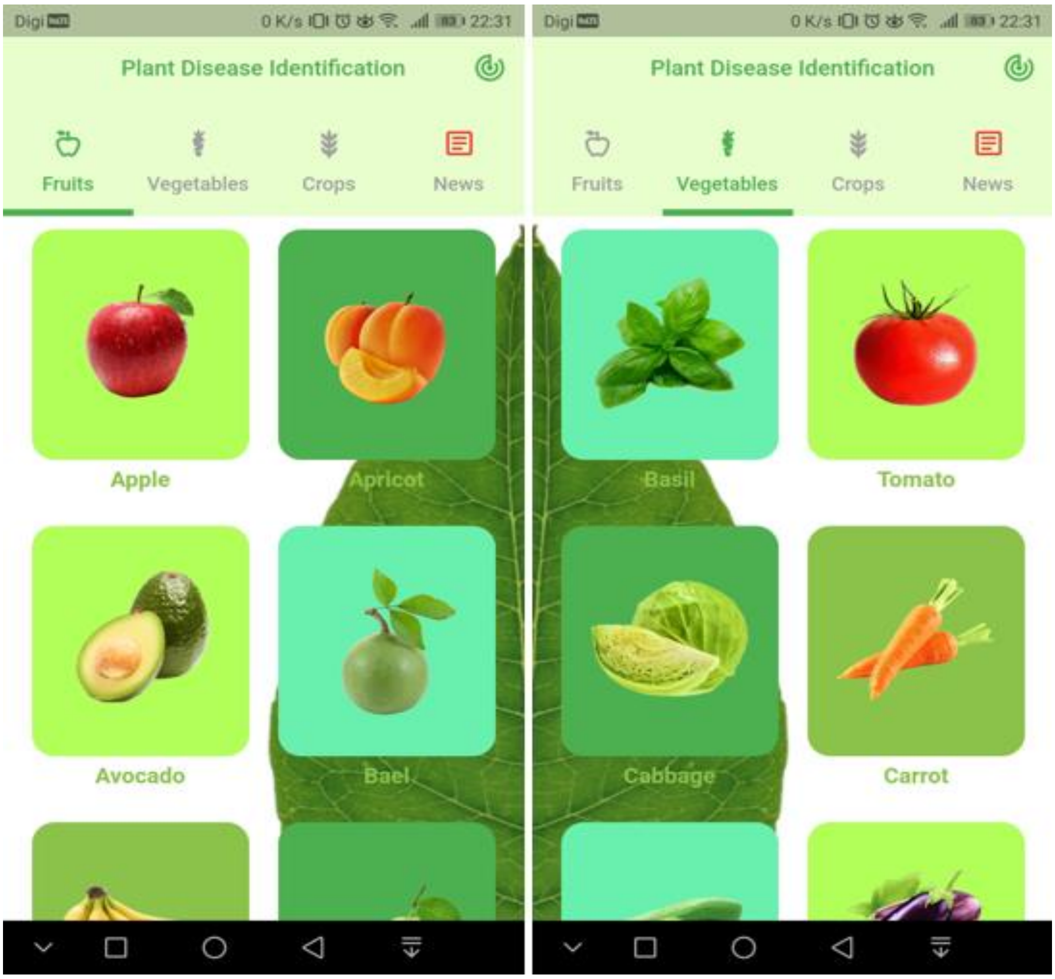


Figure 2.9.4.3 Fruits and vegetables category for plant disease identification

Weakness

The machine learning model unable to identify the disease of the uploaded affected banana leaf photo. The results of the machine learning model can carry out are between rot and blight shown in Figure 2.9.4.4. The uploaded photo disease of the banana is the black sigatoka disease which is a leaf spot disease.

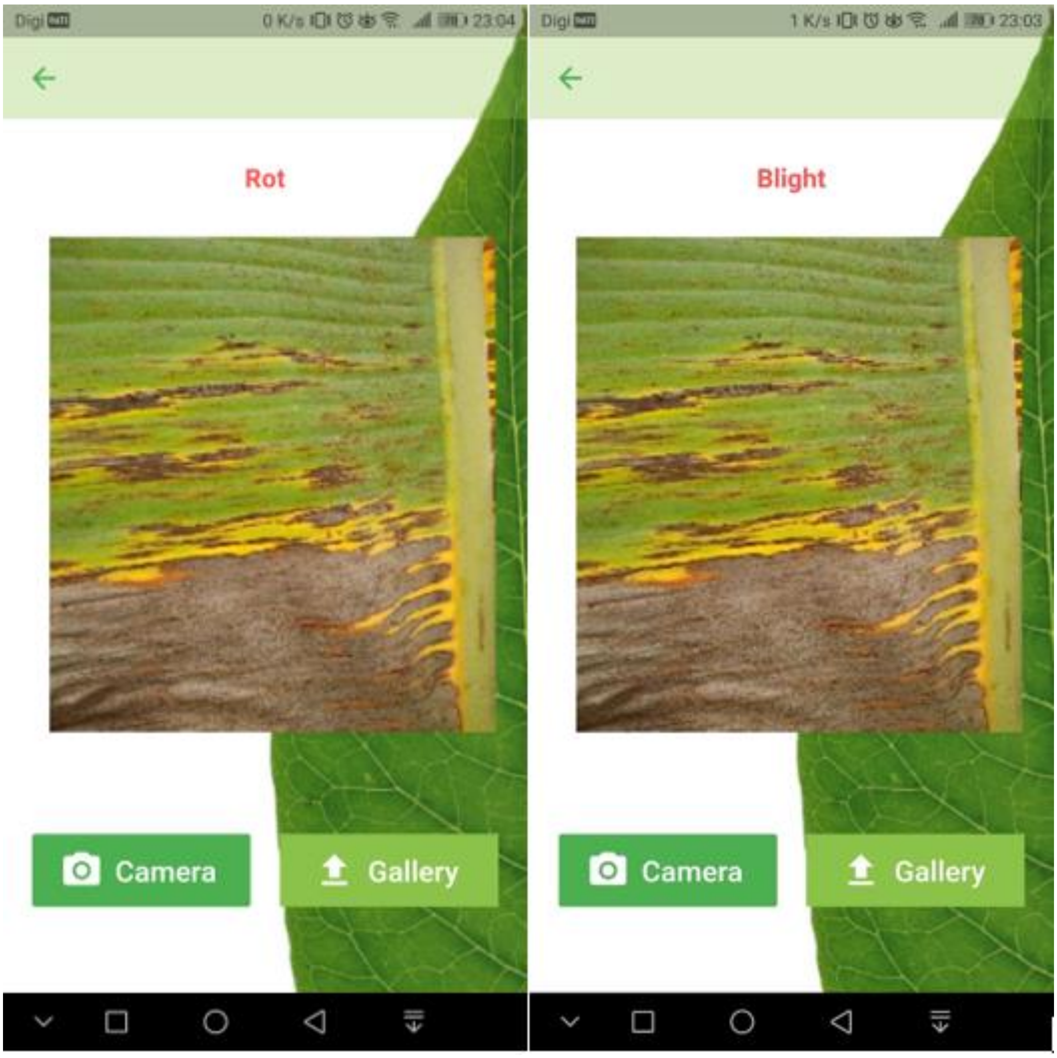


Figure 2.9.4.4 Affected banana leaf results

Recommendation

The developer is suggested to increase the dataset of the disease black sigatoka to improve the machine learning model accuracy and efficiency. Other than that, the result of the disease identification should include along with the disease description and the prevention measures information.

2.10 Critical Remark

Strengths (features)	Plant-related disease detection mobile application (Banana)				
	Tumaini App	Plantix Preview	AgroAI	Plant Disease Identification	Proposed mobile application
Disease detection diagnosis	✓	✓	✓	✓	✓
Community feature	×	✓	×	×	✓
Crop cycle guide	×	✓	×	×	✓
Disease prevention and control measures	✓	✓	✓	✓	✓
Soil fertility test	×	×	✓	×	×

Yes: ✓ No: ×

Table 2.10.1 Comparison of similar and proposed system

After comparing the existing mobile applications which similar to the project proposed mobile application, there are total four features to adapt into the implementation of the mobile application at the end of the project. Soil fertility test will be excluded as it does not convey relevant information or purpose as aligned with the proposed mobile application. Disease detection diagnosis feature is the core feature of the proposed mobile application. Community feature, crop cycle guide and disease prevention and control measures are the additional features that can further improve the mobile application information convey capability and eventually to guide users whenever they are in need.

Chapter 3 System Methodology/Approach

3.1 System Design Diagram

In this chapter, there are three subsection which will show the diagrams that can be represented as the system flow of this project. These diagrams can enhance the visualization on how the system built on.

3.1.1 System Architecture Diagram

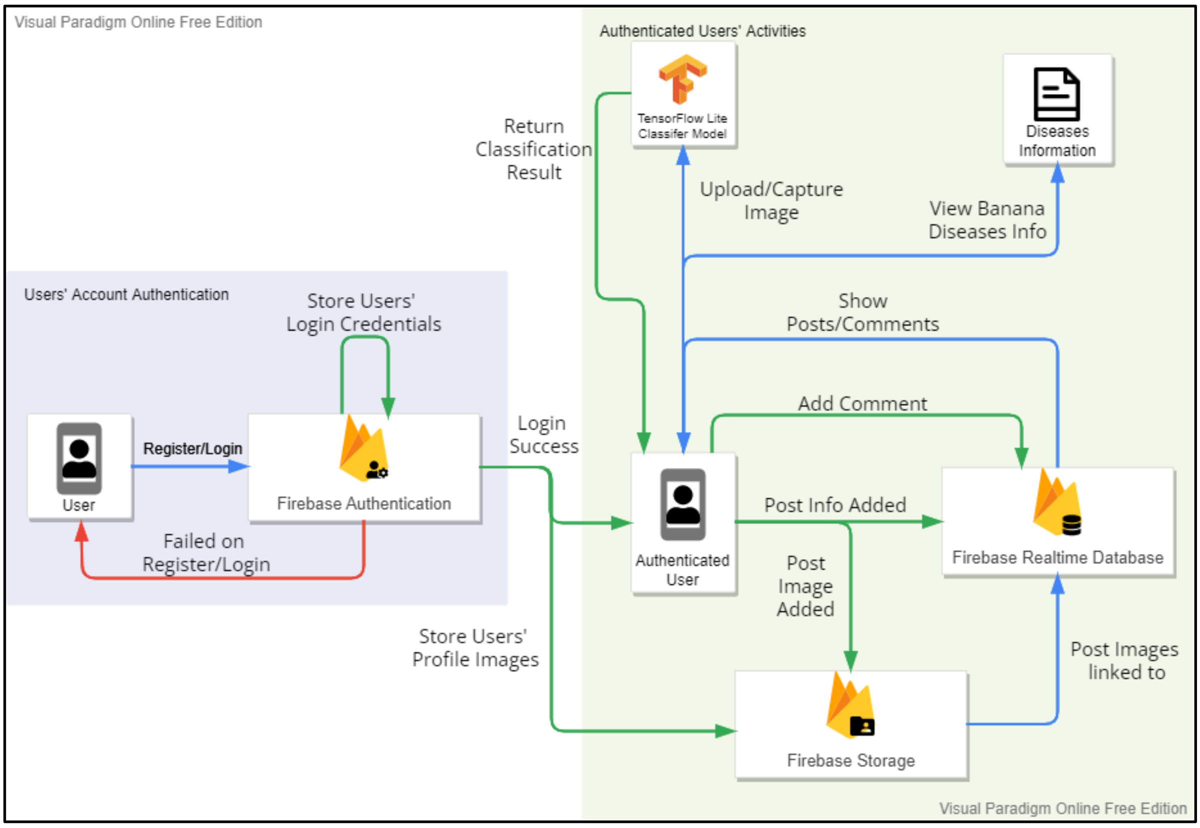

Visual Paradigm Online Free Edition
Authenticated Users' Activities
TensorFlow Lite Classifer Model
Diseases Information
Return Classification Result
Upload/Capture Image
View Banana Diseases Info
Users' Account Authentication
Store Users' Login Credentials
Show Posts/Comments
Add Comment
User
Register/Login
Firebase Authentication
Login Success
Authenticated User
Post Info Added
Firebase Realtime Database
Failed on Register/Login
Post Image Added
Store Users' Profile Images
Post Images linked to
Firebase Storage
Visual Paradigm Online Free Edition

Figure 3.1.1 Banana Disease Detection Mobile App Architecture Diagram

In this system architecture diagram for the developed mobile application, the backend architecture is based on Firebase platform. Other than that, Java programming language used as for the Android mobile application development and Python programming language used for TensorFlow and Keras deep learning model building. First and foremost, there are three different colored arrows with its solely purpose such as blue represent as primary path of the system flow, green represent the successful users' action and red represent the unsuccessful users' action. There are two important sections included which are users' account authentication and authenticated users' activities.

In users' account authentication, users are required to login into the system before allowed to perform authenticated users' action in the system. If users don't have an account for the system, the users can perform user account registration under the backend service of Firebase Authentication. The system will notify the users if their registration or login process failed. On success users' account registration, the Firebase Authentication back-end will store the registered users' login credentials which will used as authentication information to authenticate the users whenever they login into the system in future. Other than that, Firebase Authentication will redirect the stored users' login profile images to store in Firebase Storage.

In authenticated users' activities, once the users have successfully authenticated by the system, the users are allowed to use several different activities such as view posts, add post, add comment, view banana diseases information, and perform classification on infected banana leaf. Firebase Realtime Database is the backend that stored information on posts, comments, and post images linkage. Firebase Realtime Database will retrieve post images from Firebase Storage for displaying posts in the system. Authenticated users are allowed to upload their post with image and adding comment which then the system will insert that information into Firebase Realtime Database and Firebase Storage accordingly. Furthermore, the TensorFlow lite banana diseases classification model is an offline module built-in feature that offered to authenticated users to perform banana diseases classification on the image in the system after they have uploaded image or capture image using camera and system will show the classification result accordingly. Finally, authenticated users are able to read the banana diseases information regarding of diseases symptoms, causes and prevention measures which natively available in the system.

3.1.2 Use Case Diagram and Description

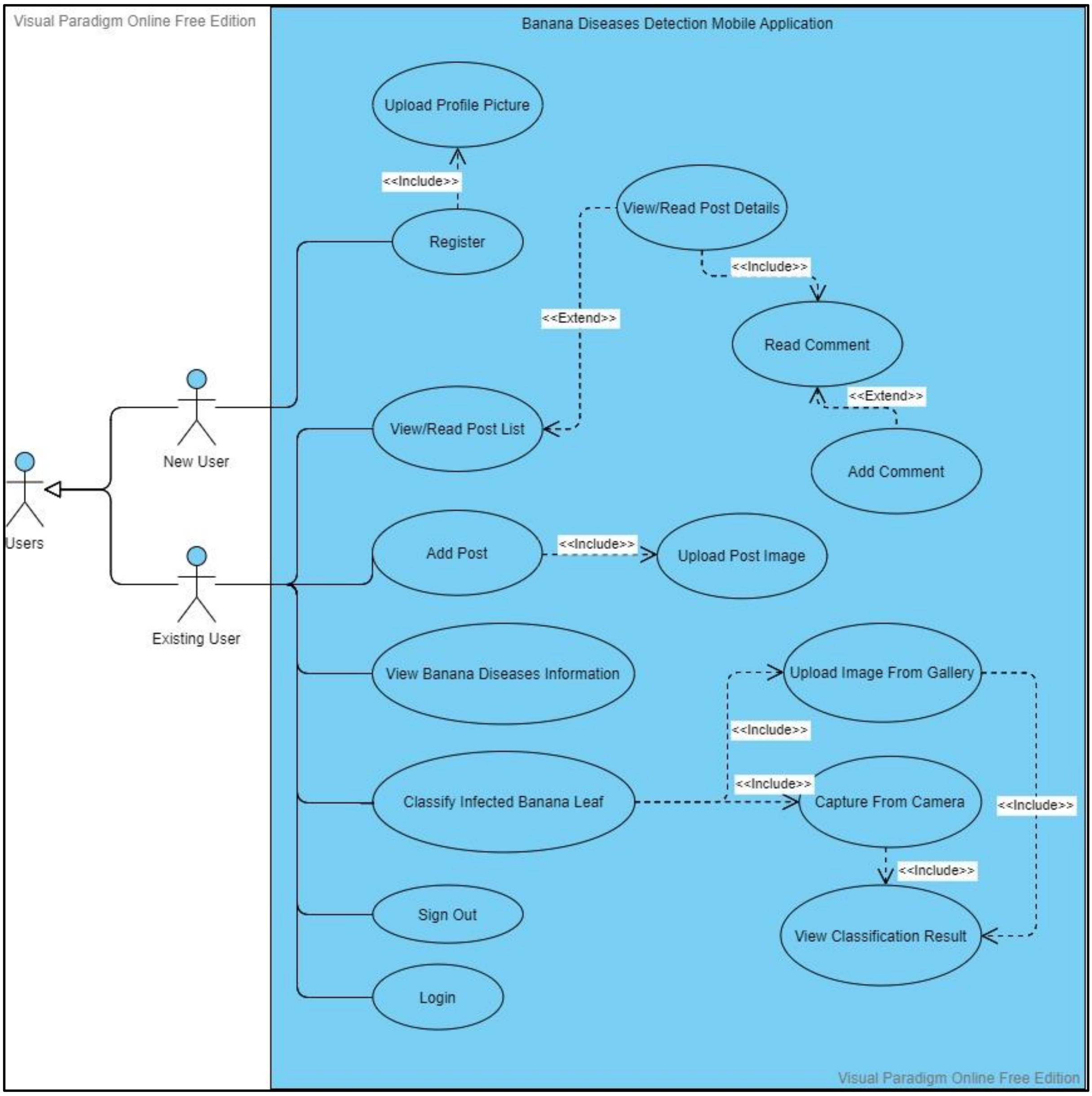


Figure 3.1.2 Banana Diseases Detection Mobile Application Use Case Diagram

3.1.2.1 Use Case Description

Register

Use Case Name: Register	ID: 101	Importance Level: High
Primary Actor: New User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • New User : want to register an account		
Brief Description: • This use case describes on how this system provide new user to register an account		
Trigger: Click on the “Register”		
Relationships: Association: - Include: 201 (Upload Profile Picture) Extend: - Generalization: -		
Normal Flow of Events: 1. New users need to register account by entering required fields. a. SF-A sub flow will be performed 2. System will validate entered fields. 3. New users need to click “Register” to finish the registration. 4. The system will display a message “Account Created” and “Register Complete” accordingly. 5. New users are then redirected by system to homepage of the system.		
Sub Flows: • SF-A: Insert Account Login Details a) New users need to insert his/her username. b) New users need to insert his/her email address. c) New users need to insert his/her password. d) New users need to insert his/her reconfirm password.		
Alternate/Exceptional Flows:		

a) The system will display message “Please Verify and fill in all fields” and “Click on Photo Icon to insert your Profile Picture” accordingly.

Login

Use Case Name: Login	ID: 102	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to login to the system		
Brief Description: • This use case describes on how this system provide existing user to login to the system		
Trigger: Click on the “Login”		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: • Existing user required login by entering their login credentials. a) SF-A will be performed • System will validate the entered login credentials. • Existing user required to click “Login” to proceed login page. • Upon successful login, existing user will be redirected by system to home page.		
Sub Flows: • SF-A: Insert Account Login Credentials a) Existing user needs to insert his/her email address. b) Existing user needs to insert his/her password.		
Alternate/Exceptional Flows: 1. The system will display message “Please Verify and fill in all fields” if fields are not filled in.		

2. The system will display message “Users are not available” if entered email address doesn’t exist in the system. 3. The system will display message “Password incorrect” if entered password doesn’t corresponding to the email address that exist in the system.

View/Read Post List

Use Case Name: View/Read Post List	ID: 103	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to view the list of post displayed in the system		
Brief Description: • This use case describes on how this system display list of posts in home page to existing user to after login to the system		
Trigger: Click on “Home” on navigation drawer or once login to the system		
Relationships: Association: - Include: - Extend: (202) View/Read Post Details Generalization: -		
Normal Flow of Events: • Existing user can scroll over the list of post in home page		
Sub Flows: Not Applicable		
Alternate/Exceptional Flows: Not Applicable		

Add Post

Use Case Name: Add Post	ID: 104	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to upload or add a post		

Brief Description: • This use case describes on how this system offer existing user to upload or add post feature
Trigger: Click on the pen icon button located right bottom in the home page of the system
Relationships: Association: - Include: (203) Upload Post Image Extend: - Generalization: -
Normal Flow of Events: • Existing user can click on the pen icon button located at right bottom in the home page. • Existing user required to fill in the fields. a) SF-A will be performed • System will validate the inserted fields. • Once post successfully added, system will issue a message display “Post added successfully”. • Existing user will be redirected back to the home page in the system.
Sub Flows: • SF-A: Insert Post Details ○ Existing user needs to insert the post title. ○ Existing user needs to insert the post description. ○ Existing user needs to insert the post image.
Alternate/Exceptional Flows: • The system will display message “Please verify all input fields and pick an picture for image post” if one of the input fields are not fill in.

View Banana Diseases Information

Use Case Name: View Banana Diseases Information	ID: 105	Importance Level: High

<table>
<tr><td>Primary Actor: Existing User</td><td>Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="2">Stakeholders and Interests:
• Existing User : want to view banana diseases information</td></tr>
<tr><td colspan="2">Brief Description:
• This use case describes on how this system display different banana diseases information to existing users</td></tr>
<tr><td colspan="2">Trigger: Click on “Banana Crop Diseases” on navigation drawer</td></tr>
<tr><td colspan="2">Relationships:
Association: -
Include: -
Extend: -
Generalization: -</td></tr>
<tr><td colspan="2">Normal Flow of Events:
• Existing user can click on the “Banana Crop Diseases” on navigation drawer.
• Existing user will be redirect to Banana Crop Diseases page.
• Existing user can click on one of the banana diseases to view detail information about the disease.
a) SF-A will be performed</td></tr>
<tr><td colspan="2">Sub Flows:
• SF-A: View Diseases Details
○ Existing user can view disease symptoms information
○ Existing user can view causal of the disease
○ Existing user can view prevention measures information
○ Existing user can click on source link for further details which will redirected to browser</td></tr>
<tr><td colspan="2">Alternate/Exceptional Flows: Not Applicable</td></tr>
</table>

Classify Infected Banana Leaf

Use Case Name: Classify Infected Banana Leaf	ID: 106	Importance Level: High

Primary Actor: Existing User	Use Case Type: Detail, Essential
Stakeholders and Interests: • Existing User : want to classify an infected banana leaf	
Brief Description: • This use case describes on how this system provide banana disease classification feature to existing user	
Trigger: Click on “Detector” on navigation drawer	
Relationships: Association: - Include: (205) Upload Image From Gallery, (206) Capture From Camera Extend: - Generalization: -	
Normal Flow of Events: • Existing user can click on the “Detector” on navigation drawer. • Existing user will be redirect to Detector page. • Existing user can click on one of the buttons, either upload an infected banana leaf image from device gallery or capture from the camera.	
Sub Flows: Not Applicable	
Alternate/Exceptional Flows: Not Applicable	

Sign Out

Use Case Name: Sign Out	ID: 107	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to sign out from the system		
Brief Description: • This use case describes on how this system provide a sign out feature for existing users to exit the system		
Trigger: Click on “Sign Out” on navigation drawer		
Relationships:		

Association: - Include: - Extend: - Generalization: -
Normal Flow of Events: • Existing user can click on the “Sign Out” on navigation drawer. • Existing user will be redirect to Login page.
Sub Flows: Not Applicable
Alternate/Exceptional Flows: Not Applicable

Upload Profile Picture

Use Case Name: Upload Profile Picture	ID: 201	Importance Level: High
Primary Actor: New User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • New User : prompted to upload a picture from their device to complete account registration		
Brief Description: • This use case describes on how this system prompted new user to upload their picture as a profile picture for their account registration purpose.		
Trigger: Click on the photo icon in the register page		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: • New user can click on the photo icon in the register page. • New user will be prompted to allow system to gain permission access to device gallery after a dialog box pop up. a) SF-A will be performed.		

• New user can view the photo icon changed to their selected profile picture from gallery.
Sub Flows: • SF-A: Device Gallery Permission: o New user can either choose to “allow” or “deny” completing the permission process
Alternate/Exceptional Flows: • The system will be display “Please accept for required permission” message if new user click on “deny” for the permission access for gallery.

View/Read Post Details

Use Case Name: View/Read Post Details	ID: 202	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to read/view the details of selected post that existing user clicked from the list of post in home page		
Brief Description: • This use case describes on how this system display selected post details for the existing user to read or view		
Trigger: Click on a post in the list of posts in the home page		
Relationships: Association: - Include: (301) Read Comment Extend: - Generalization: -		
Normal Flow of Events: • Existing user can click on a post in the list of posts in home page. • Existing user will be redirect to the selected post’ Post Details page. • Existing user can read the details. a) SF-A will be performed		

Sub Flows: • SF-A: Read Post Details: o Existing user can view post title. o Existing user can view post description. o Existing user can view which existing users’ post. o Existing user can view the date of the post.
Alternate/Exceptional Flows: Not Applicable

Upload Post Image

Use Case Name: Upload Post Image	ID: 203	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : prompted to upload a picture from their device to complete add post process		
Brief Description: • This use case describes on how this system prompted existing user to upload their picture as a post image for add post process.		
Trigger: Click on the photo pane in the add post pop up window		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: • Existing user can click on the photo pane in the post pop up window. • Existing user will be prompted to allow system to gain permission access to device gallery after a dialog box pop up. a) SF-A will be performed. • Existing user can view the photo pane changed to their selected images uploaded from gallery.		

Sub Flows: • SF-A: Device Gallery Permission: o Existing user can either choose to “allow” or “deny” completing the permission process
Alternate/Exceptional Flows: • The system will be display “Please accept for required permission” message if new user click on “deny” for the permission access for gallery.

Upload Image From Gallery

Use Case Name: Upload Image From Gallery	ID: 204	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : wants to upload an infected banana leaf to use banana disease classification feature and retrieve result		
Brief Description: • This use case describes on how this system provide existing user to upload an infected banana leaf from existing user device to do banana disease classification.		
Trigger: Click on the “From Gallery” button in the Detector page		
Relationships: Association: - Include: (301) View Classification Result Extend: - Generalization: -		
Normal Flow of Events: • Existing user can click on the “From Gallery” button in the Detector page. • Existing user will be prompted to allow system to gain permission access to device gallery after a dialog box pop up. a) SF-A will be performed. • Existing user can view the image pane changed to their selected infected banana leaf image that uploaded from gallery.		

Sub Flows: • SF-A: Device Gallery Permission: ○ Existing user can either choose to “allow” or “deny” completing the permission process
Alternate/Exceptional Flows: • The system will be display “Please accept for required permission” message if new user click on “deny” for the permission access for gallery.

Capture From Camera

Use Case Name: Capture From Camera	ID: 205	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : wants to use camera to capture an infected banana leaf to use banana disease classification feature and retrieve result		
Brief Description: • This use case describes on how this system provide existing user to use camera to capture an infected banana leaf from existing user device to do banana disease classification.		
Trigger: Click on the “Take Picture” button in the Detector page		
Relationships: Association: - Include: (301) View Classification Result Extend: - Generalization: -		
Normal Flow of Events: • Existing user can click on the “Take Picture” button in the Detector page. • Existing user will be prompted to allow system to gain permission access to device camera after a dialog box pop up. a) SF-A will be performed.		

• Existing user can view the image pane changed to their infected banana leaf image that captured from device camera.
Sub Flows: • SF-A: Device Camera Permission: o Existing user can either choose to "allow" or "deny" completing the permission process
Alternate/Exceptional Flows: • The system will be display "Please accept for required permission" message if new user click on "deny" for the permission access for gallery.

Read Comment

Use Case Name: Read Comment	ID: 301	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to read/view the comments in Post Details page		
Brief Description: • This use case describes on how this system display comments of the selected post details for the existing user to read or view		
Trigger: Click on a post in the list of posts in the home page		
Relationships: Association: - Include: - Extend: (302) Add Comment Generalization: -		
Normal Flow of Events: • Existing user can click on a post in the list of posts in home page. • Existing user will be redirect to the selected post' Post Details page. • Existing user can read the list of comments for the post. a) SF-A will be performed		
Sub Flows:		

• SF-A: Read Comments Details: o Existing user can view commentor username. o Existing user can view comments content. o Existing user can view comments' date and time.
Alternate/Exceptional Flows: Not Applicable

Add Comment

Use Case Name: Add Comment	ID: 302	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to add comment in Post Details page		
Brief Description: • This use case describes on how this system provide existing user to add comment in the post		
Trigger: Click and enter comment content in input field and click on send button		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: • Existing user can click and enter the comment content in the input field in Post Details page. • Existing user can click on the send button once they wanted to send out the comment. • Existing user can see a message displayed "Your comment successfully added" upon successful added comment.		
Sub Flows: Not Applicable		
Alternate/Exceptional Flows: Not Applicable		

View Classification Result

Use Case Name: View Classification Result	ID: 303	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Existing User : want to read/view the details of classification result		
Brief Description: • This use case describes on how this system display the classification result of infected banana leaf image uploaded or captured by existing user		
Trigger: After uploaded/captured infected banana leaf image		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: • Existing user can view the classification result in Detector page after uploaded/captured infected banana leaf image. a) SF-A will be performed		
Sub Flows: • SF-A: Read Classification Result: ○ Existing user can view classified banana disease name. ○ Existing user can view percentage of each disease class confidence.		
Alternate/Exceptional Flows: Not Applicable		

3.1.3 Activity Diagram

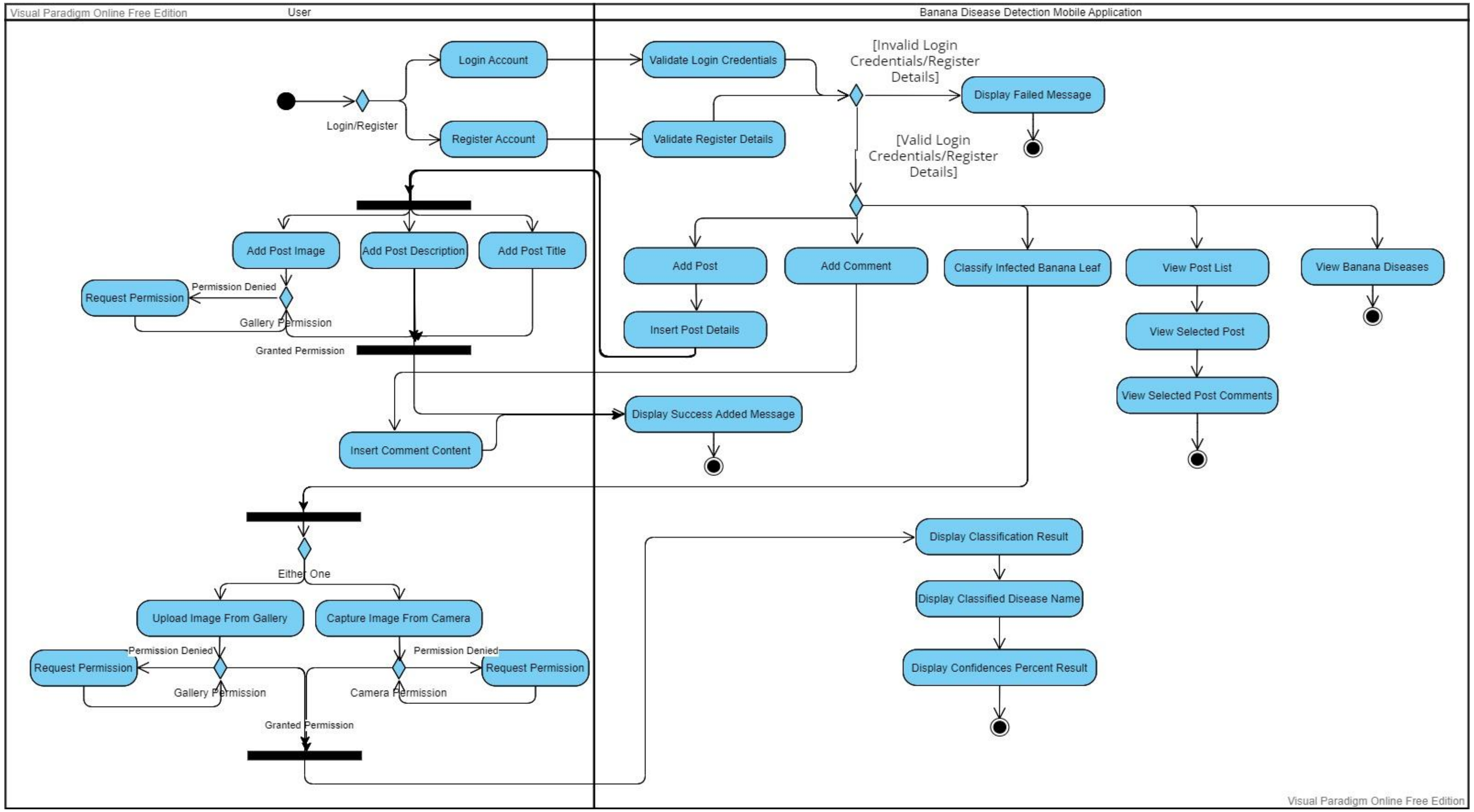

Visual Paradigm Online Free Edition
User
Banana Disease Detection Mobile Application
Login Account
Register Account
Login/Register
Validate Login Credentials
Validate Register Details
[Invalid Login Credentials/Register Details]
Display Failed Message
[Valid Login Credentials/Register Details]
Add Post Image
Add Post Description
Add Post Title
Request Permission
Permission Denied
Gallery Permission
Granted Permission
Add Post
Add Comment
Classify Infected Banana Leaf
View Post List
View Banana Diseases
Insert Post Details
View Selected Post
View Selected Post Comments
Display Success Added Message
Insert Comment Content
Either One
Upload Image From Gallery
Capture Image From Camera
Request Permission
Permission Denied
Gallery Permission
Camera Permission
Permission Denied
Request Permission
Granted Permission
Display Classification Result
Display Classified Disease Name
Display Confidences Percent Result
Visual Paradigm Online Free Edition

Figure 3.1.3 Activity Diagram for Banana Disease Detection Mobile Application

Chapter 4 System Design

In this chapter, the project will be clearly explained and following on how the system will be designed and developed.

4.1 Project Flow Diagram

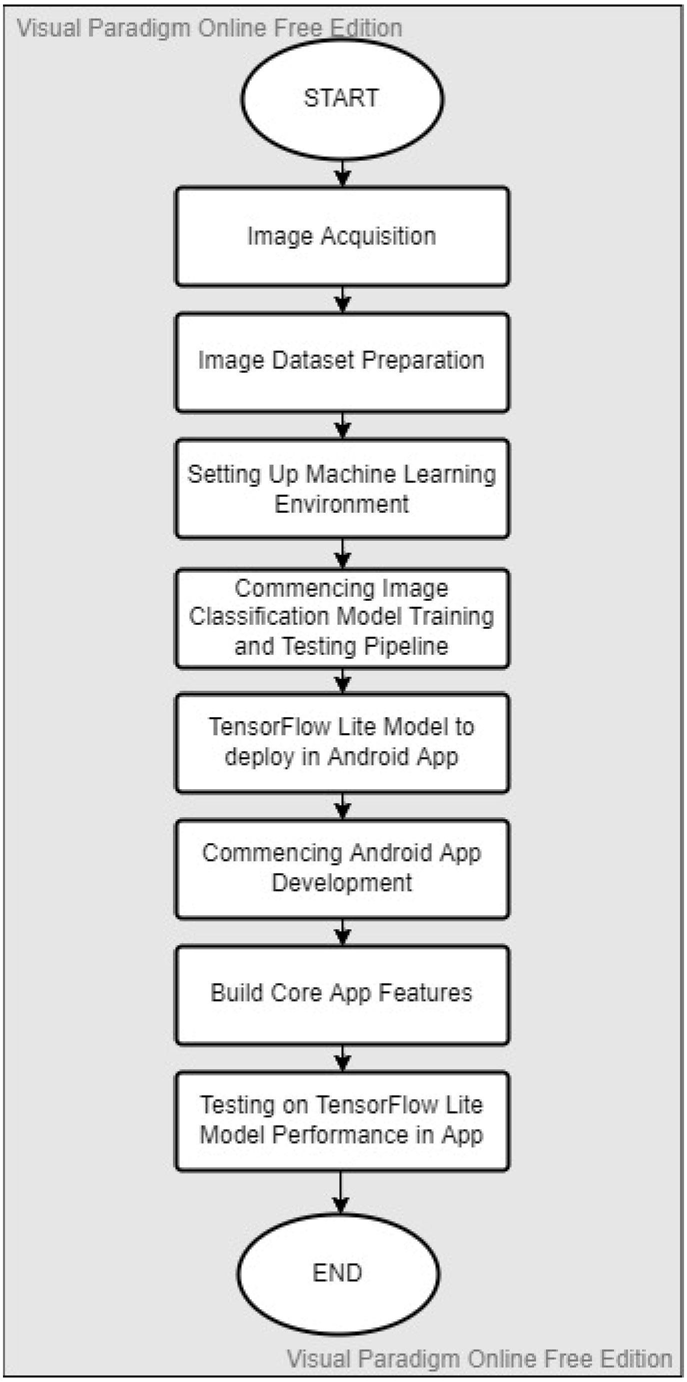

Visual Paradigm Online Free Edition
START
Image Acquisition
Image Dataset Preparation
Setting Up Machine Learning Environment
Commencing Image Classification Model Training and Testing Pipeline
TensorFlow Lite Model to deploy in Android App
Commencing Android App Development
Build Core App Features
Testing on TensorFlow Lite Model Performance in App
END
Visual Paradigm Online Free Edition

Figure 4.1 Project Flow Diagram

The initiation of the project will be crucially dependent on image acquisition step which is to retrieve images of banana leaf that infected with diseases. With all the acquired images, images dataset preparation can be proceeded and need to split into training dataset and test dataset respectively. Further on, an environment for machine learning needed to be setup before proceeding for train a deep learning model. Once the environment has been setup, the training and testing pipeline for deep learning image classification model can commenced. At the end of the pipeline, the trained model is required to convert to a lightweight TensorFlow Lite model which suitable to run in an Android mobile application.

The project will be continued on the Android application development. During the application development, there are several features need to be developed and the core feature will be the banana diseases classifier as the computer vision task in the app. Finally, it is needed to evaluate the performance of the banana diseases classifier to test its limitation and strength.

4.2 EfficientNetV2-b0 Deep Learning Trained TensorFlow Lite Model

In this section, is to show the steps have been implemented for building, training and testing the deep learning model. TensorFlow Lite model converted from trained EfficientNetV2-b0 as the output for deploy into the mobile application. The environment used is the Google Colab with the TensorFlow as the open-source libraries available to use. Besides that, Google Colab provided the free access of GPU which allow this implementation to be successfully accomplish.

4.2.1 TensorFlow Lite Model Building Flowchart

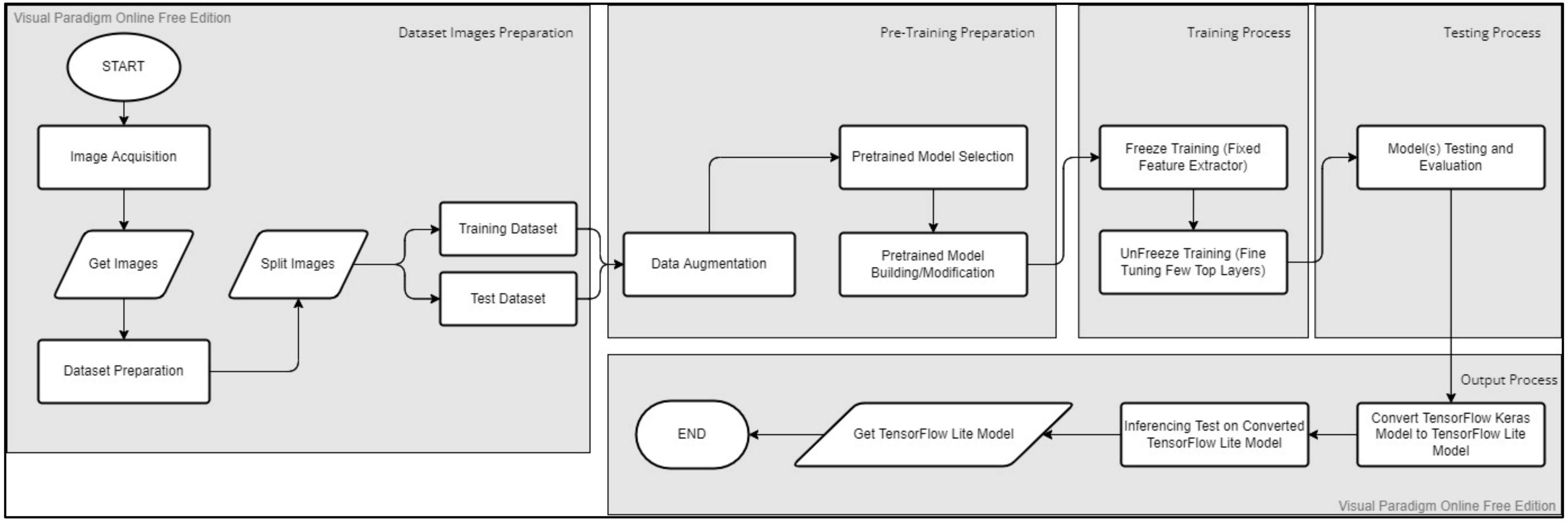

Visual Paradigm Online Free Edition
Dataset Images Preparation
START
Image Acquisition
Get Images
Dataset Preparation
Split Images
Training Dataset
Test Dataset
Pre-Training Preparation
Data Augmentation
Pretrained Model Selection
Pretrained Model Building/Modification
Training Process
Freeze Training (Fixed Feature Extractor)
UnFreeze Training (Fine Tuning Few Top Layers)
Testing Process
Model(s) Testing and Evaluation
Output Process
Convert TensorFlow Keras Model to TensorFlow Lite Model
Inferencing Test on Converted TensorFlow Lite Model
Get TensorFlow Lite Model
END
Visual Paradigm Online Free Edition

Figure 4.2.1 TensorFlow Lite Model Building Flowchart

Based on Figure 4.2.1, the whole sub chapter of 4.2 will be following each process show in the flowchart. There are total of five process to implement the TensorFlow Lite model building according to the flowchart of Figure 4.2.1. At the end of the flowchart, a TensorFlow Lite model will be developed and ready to deploy into the Android application.

4.2.2 Dataset Images Preparation

4.2.2.1 Image Acquisition for Dataset Preparation

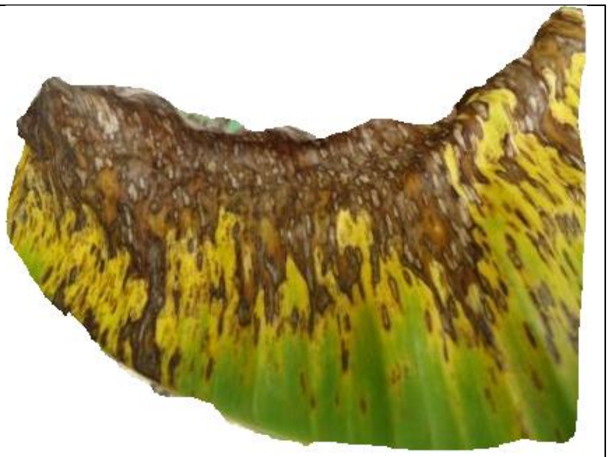
This project dataset will be based on combining three datasets that available on three different online repository platform which are GitHub, Kaggle and Mendeley Data.

godliver/source-code-BBW-BBS (GitHub)

This dataset is provided by the user from GitHub named as Godliver Owomugisha [22]. The dataset is composed of three type of banana leaf which are Black Sigatoka disease, Bacterial Wilt disease and healthy banana leaf. The images inside the dataset have inconsistent resolution scale and all images can be in landscape or portrait.

Types of Banana Leaf Status	Total Number of Images	Cropped	With White Colour Background
BBS (Banana Black Sigatoka)	43	Yes	Yes
BBW (Banana Bacterial Wilt)	220	Yes	Yes
Healthy	360	Yes	Yes

Table 4.2.2.1.1 Different type of banana leaf provided in GitHub

Banana Black Sigatoka	

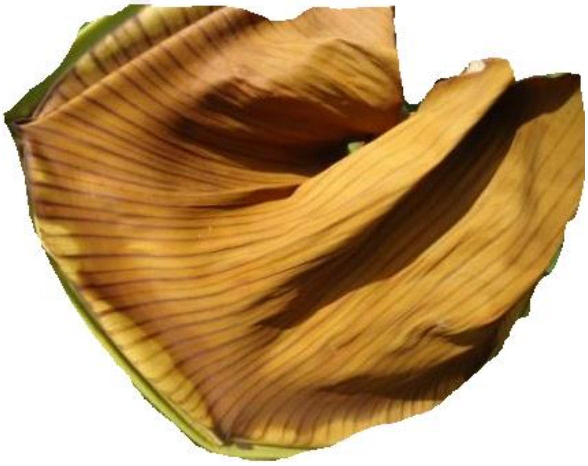
Banana Bacterial Wilt	
Healthy	

Table 4.2.2.1.2 Original Image Examples from GitHub Repository

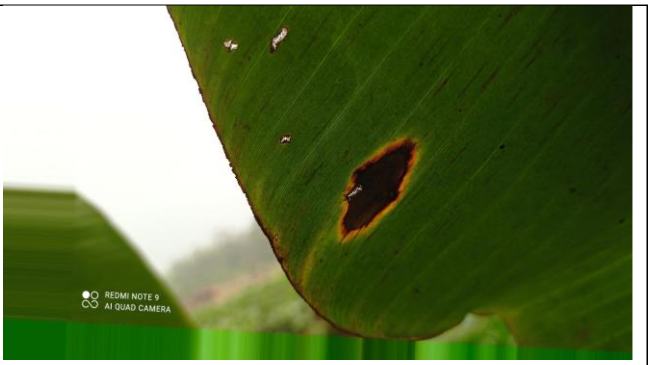
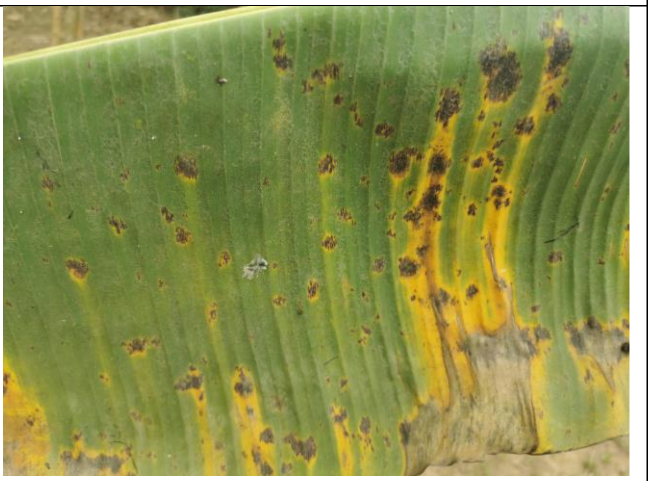
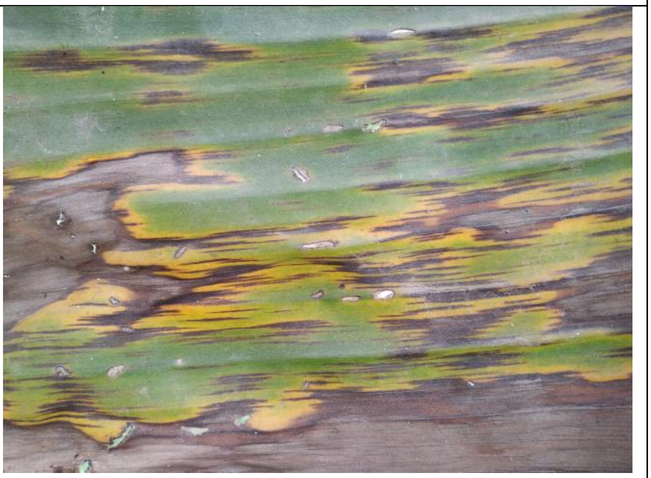
Banana Leaf Dataset (Kaggle)

This dataset is provided by the user in Kaggle.com named as Kaies Al Mahmud [23]. The dataset is composed of four type of banana leaf which are Cordana disease, Pestalotiopsis disease, Black Sigatoka disease and healthy banana leaf as well. The dataset has separation of three set which are test set, training set and validation set. Test set and validation set have the same number of images for each type of banana leaf. The images in the dataset also have the condition of different resolution scale and images are in portrait or landscape.

Types of Banana Leaf Status	Total Number of Images	Test set/Validation set	Training set
Cordana	162	20/20	122
Pestalotiopsis	173	20/20	133
Black Sigatoka	473	20/20	433

Healthy	129	20/20	89

Table 4.2.2.1.3 Different type of banana leaf images provided in Kaggle

Cordana	
Pestalotiopsis	
Black Sigatoka	

Healthy	

Table 4.2.2.1.4 Original Image Examples from Kaggle Repository

Banana Leaf Disease Images (Mendeley Data)

This dataset is retrieved from the Mendeley Data which contributed by Yordanos Hailu [24]. This dataset is composed of 1288 pictures in total that under three categories which are healthy banana leaf, “Xanthomonas” infected leaf can be considered as a type of bacterial wilt disease and “Black sigatoka” infected banana leaf. All the images in the dataset are in (150x113) resolution and “.jpg” file format.

Types of Banana Leaf Status	Total Number of Images	Resolution	File Format
Healthy	155	150 x 113	JPG
(Xanthomonas) Bacterial Wilt	814	150 x 113	JPG
Black Sigatoka	320	150 x 113	JPG

Table 4.2.2.1.5 Different type of banana leaf images provided in Mendeley Data

Healthy	

(Xanthomonas) Bacterial Wilt	
Black Sigatoka	

Table 4.2.2.1.6 Original Image Examples from Mendeley Data Repository

4.2.2.2 Dataset Preparation

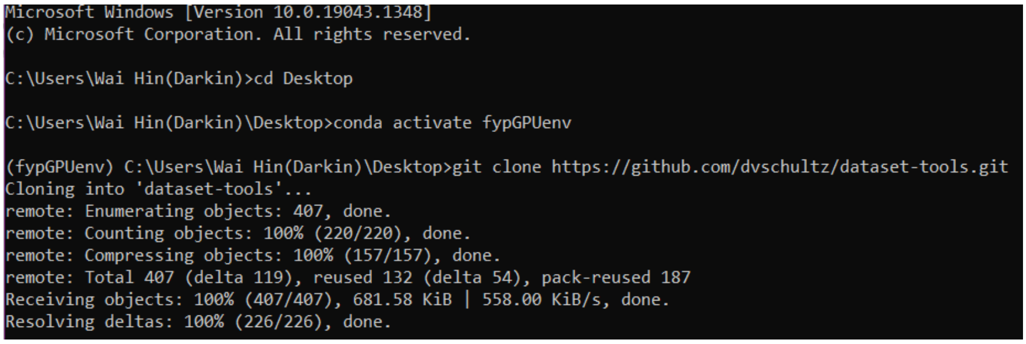
The total three datasets from three different sources are required to combined, resize the images' resolution and reformat the file type. The reason for resizing the images' resolution is to meet the requirement of input size of (224 x 224) by EfficientNetV2-b0 input size layer. Moreover, the initial combined dataset total images number is 2849 which is not adequate for the training and testing purposes. Therefore, a tool or a method is required to increase the total number of images for each type of banana leaf, resize the images and reformat the image file format.

There is one dataset tools can be found in GitHub which contributed by dvschultz, duskvirkus, blueshirt and mudphone from GitHub users who created the (dataset-tools) repository [25]. The dataset tools can be performed in command prompt interface which running in python script.

Step for altering dataset:

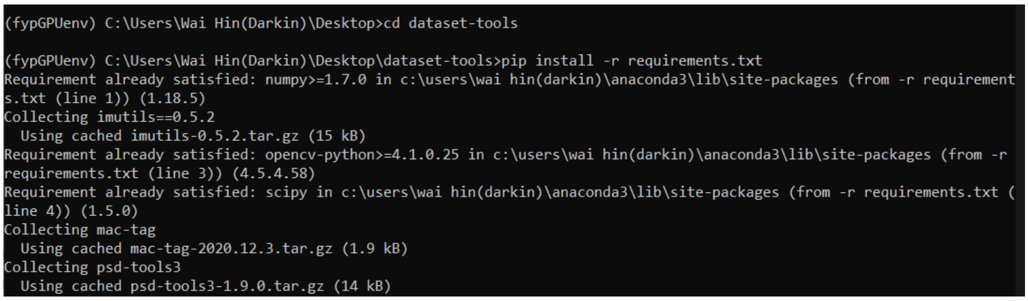
1. Installation:

   Open up command prompt in Window. Change directory to designated as the preference. Activate the anaconda environment for running required python libraries in the current directory. Perform the Git cloning from the online dataset tools repository. Change the directory to the (dataset-tools) folder and install the configuration (requirements.txt) file.



```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wai Hin(Darkin)>cd Desktop

C:\Users\Wai Hin(Darkin)\Desktop>conda activate fypGPUenv

(fypGPUenv) C:\Users\Wai Hin(Darkin)\Desktop>git clone https://github.com/dvschultz/dataset-tools.git
Cloning into 'dataset-tools'...
remote: Enumerating objects: 407, done.
remote: Counting objects: 100% (220/220), done.
remote: Compressing objects: 100% (157/157), done.
remote: Total 407 (delta 119), reused 132 (delta 54), pack-reused 187
Receiving objects: 100% (407/407), 681.58 KiB | 558.00 KiB/s, done.
Resolving deltas: 100% (226/226), done.
```

Figure 4.2.2.2.1 Step 1, part 1

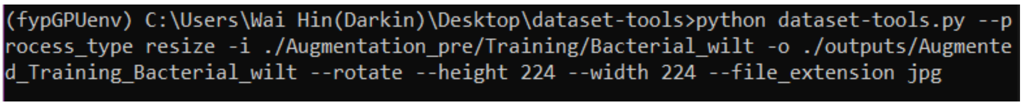


```
(fypGPUenv) C:\Users\Wai Hin(Darkin)\Desktop>cd dataset-tools

(fypGPUenv) C:\Users\Wai Hin(Darkin)\Desktop\dataset-tools>pip install -r requirements.txt
Requirement already satisfied: numpy>=1.7.0 in c:\users\wai hin(darkin)\anaconda3\lib\site-packages (from -r requirement
s.txt (line 1)) (1.18.5)
Collecting imutils==0.5.2
  Using cached imutils-0.5.2.tar.gz (15 kB)
Requirement already satisfied: opencv-python>=4.1.0.25 in c:\users\wai hin(darkin)\anaconda3\lib\site-packages (from -r
requirements.txt (line 3)) (4.5.4.58)
Requirement already satisfied: scipy in c:\users\wai hin(darkin)\anaconda3\lib\site-packages (from -r requirements.txt (
line 4)) (1.5.0)
Collecting mac-tag
  Using cached mac-tag-2020.12.3.tar.gz (1.9 kB)
Collecting psd-tools3
  Using cached psd-tools3-1.9.0.tar.gz (14 kB)
```

Figure 4.2.2.2.2 Step 1, part 2

2. Resize, reformat file format, increase total images number via 90-degree rotation augmentation:

   Example for Bacterial Wilt type banana leaf:-



```
(fypGPUenv) C:\Users\Wai Hin(Darkin)\Desktop\dataset-tools>python dataset-tools.py --p
rocess_type resize -i ./Augmentation_pre/Training/Bacterial_wilt -o ./outputs/Augmente
d_Training_Bacterial_wilt --rotate --height 224 --width 224 --file_extension jpg
```

Figure 4.2.2.2.3 Example dataset-tools.py python command for step 2

After the steps successfully applied, the images for each type of banana leaf have increase in images total number and successfully resize in (224x224) resolution with all in JPG format file. For each type of banana leaf is required to split into two set such as training and test set in 8:2 ratio split showed in Table 4.2.2.2.1.

Types of Banana Leaf Status	Total Number of Images	Training Set	Test Set
Healthy	1000	800	200
(Xanthomonas) Bacterial Wilt	1000	800	200
Black Sigatoka	1000	800	200
Cordana	1000	800	200
Pestalotiopsis	1000	800	200

Table 4.2.2.2.1 Prepared Dataset For Training and Testing

Below table is the sample of training images after the dataset preparation process.

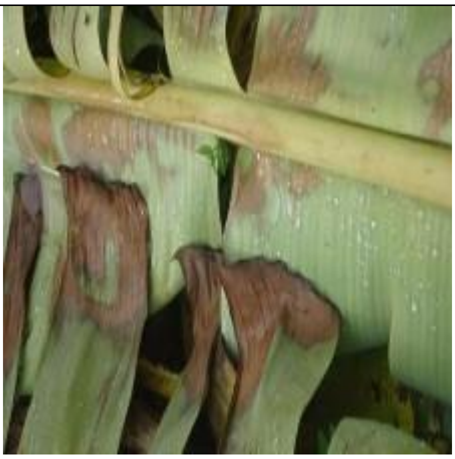
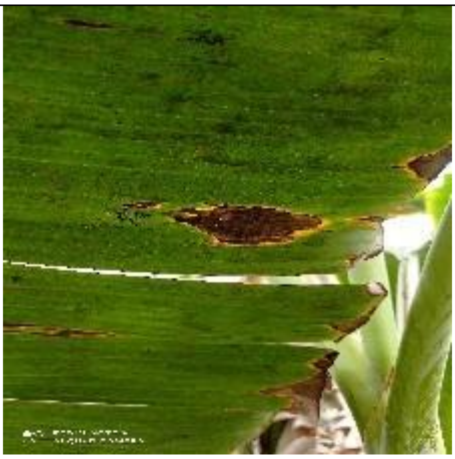
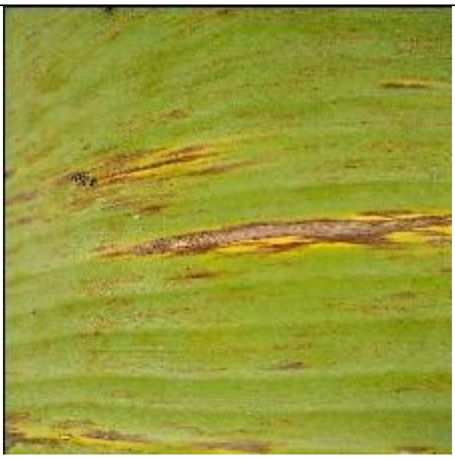
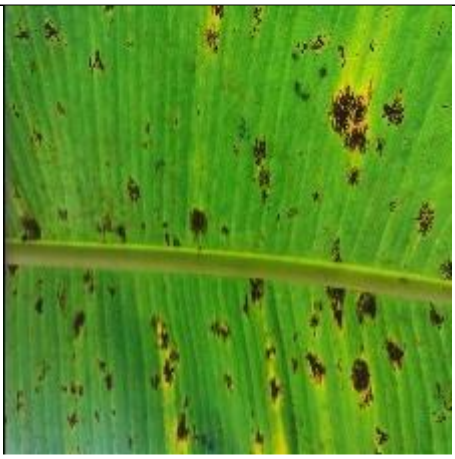
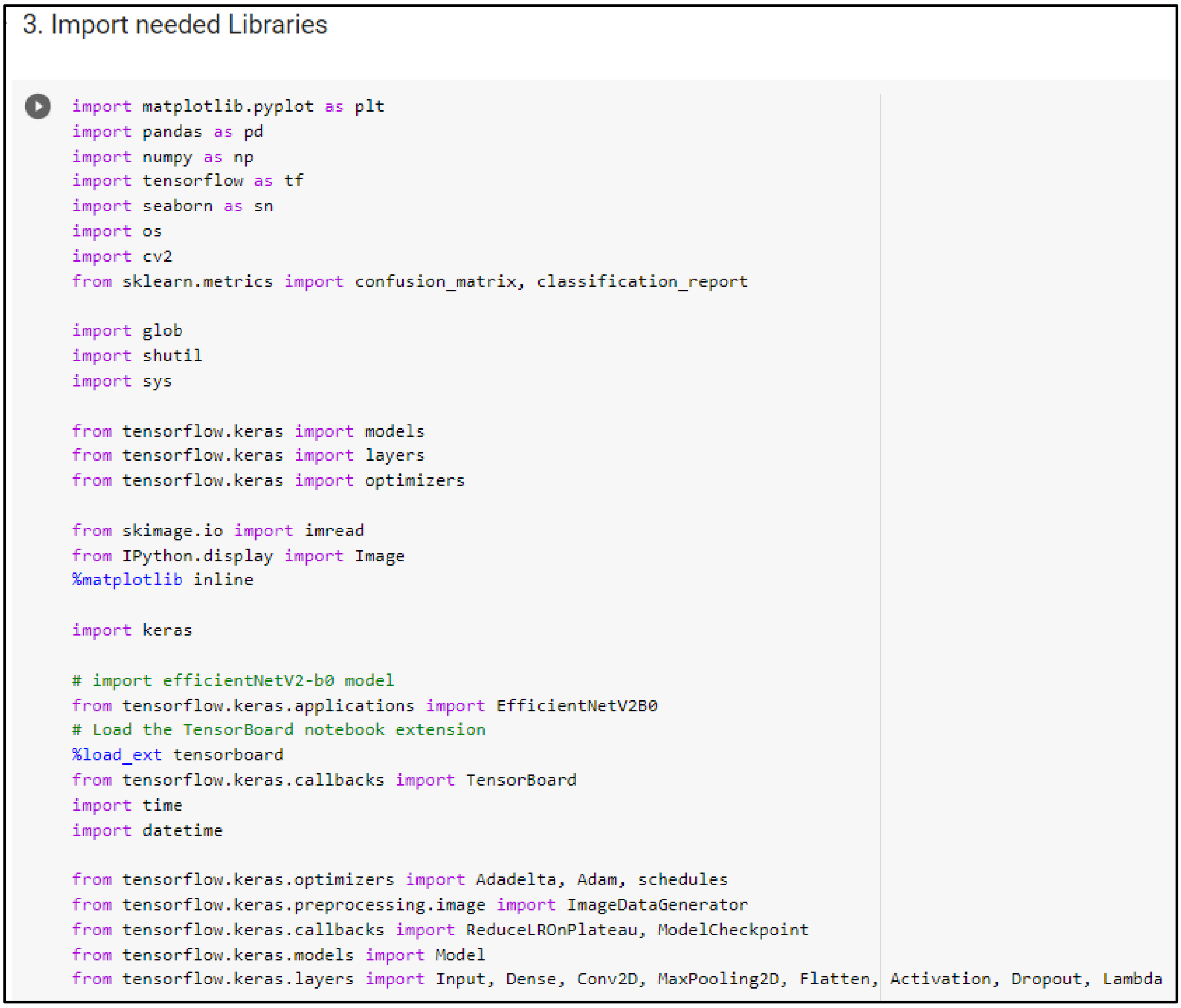
 Bacterial Wilt	 Cordana
 Black Sigatoka	 Pestalotiopsis
 Healthy	

Table 4.2.2.2.2 Sample of Training Set Images For Each Type Of Banana Leaf

4.2.3 Pre-training Preparation

4.2.3.1 Setup Development Environment

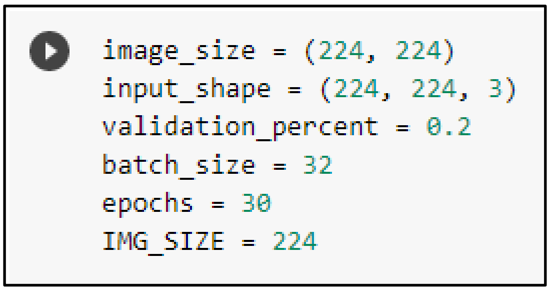
Once image dataset is fully prepared, it can be uploaded to Google Drive via a Google account. Google Colab can be opened in the Google Drive and open a new notebook (.ipynb). After that, all important libraries, image dataset and hyperparameters have executed in the runtime.



3. Import needed Libraries

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import tensorflow as tf
import seaborn as sn
import os
import cv2
from sklearn.metrics import confusion_matrix, classification_report

import glob
import shutil
import sys

from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras import optimizers

from skimage.io import imread
from IPython.display import Image
%matplotlib inline

import keras

# import efficientNetV2-b0 model
from tensorflow.keras.applications import EfficientNetV2B0
# Load the TensorBoard notebook extension
%load_ext tensorboard
from tensorflow.keras.callbacks import TensorBoard
import time
import datetime

from tensorflow.keras.optimizers import Adadelta, Adam, schedules
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Conv2D, MaxPooling2D, Flatten, Activation, Dropout, Lambda
```

Figure 4.2.3.1.1 Import important libraries in runtime Google Colaboratory



```
image_size = (224, 224)
input_shape = (224, 224, 3)
validation_percent = 0.2
batch_size = 32
epochs = 30
IMG_SIZE = 224
```

Figure 4.2.3.1.2 Hyperparameters Setup

4.2.3.2 Data Augmentation

ImageDataGenerator is a class under image preprocessing from Keras. It responsible to augment training dataset in real-time[26]. From the Figure 4.2.3.2, the total images in training dataset are 3200 that belong to 5 classes. Validation dataset total images is 800 belong to 5 classes. Test dataset will have 1000 total images belong to 5 classes. With using ImageDataGenerator, trained model will be more robust and reduce the overhead memory.

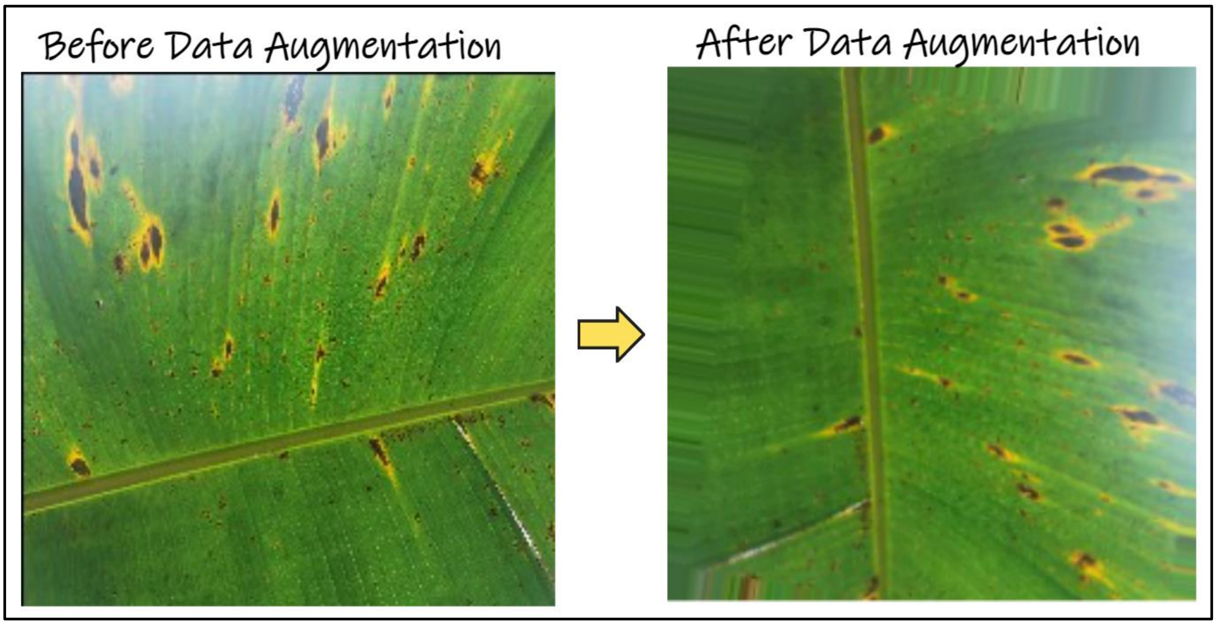


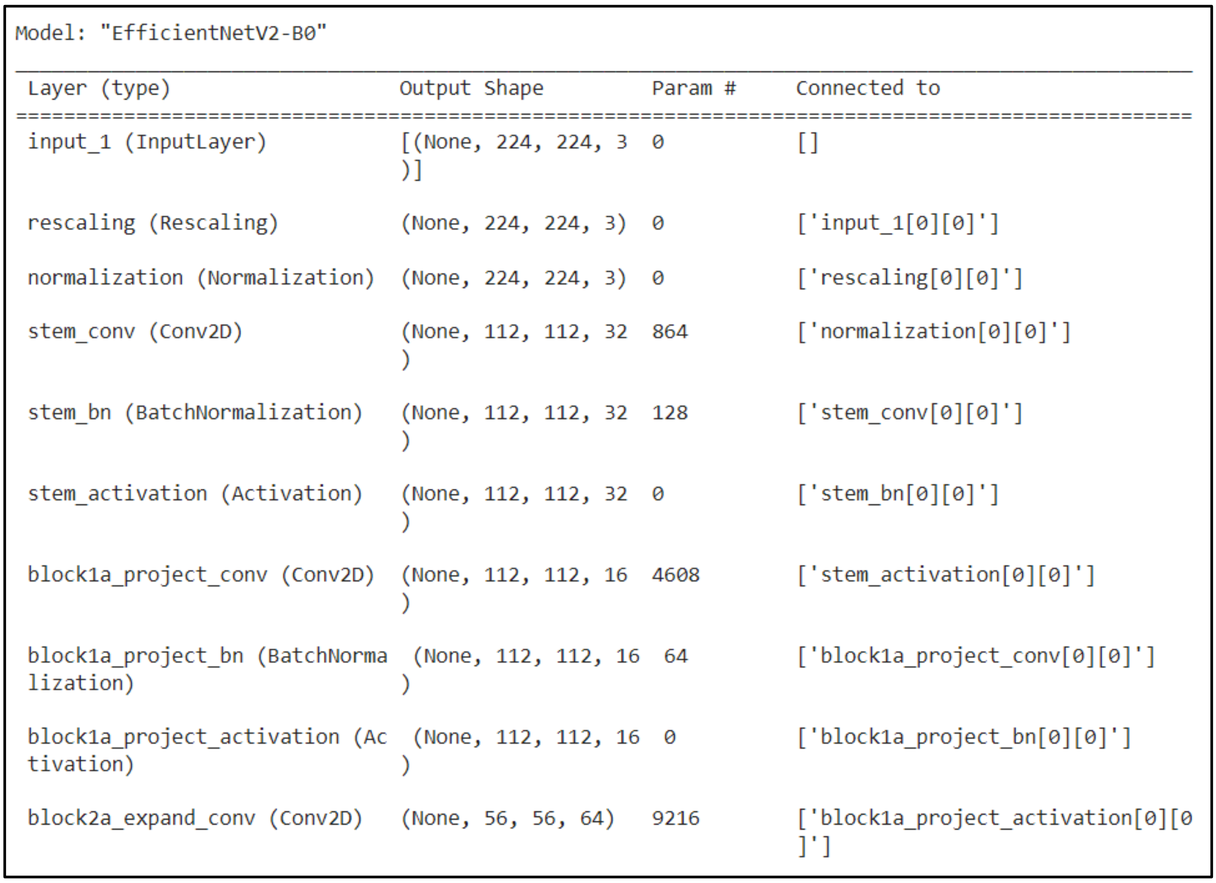
Figure 4.2.3.2 Training Images After Data Augmentation in Google Colaboratory

4.2.3.3 Pretrained Model Selection and Building

This section focused on building and selecting the deep learning model in the Google Colab environment. "EfficientNetV2B0" can be imported into runtime by getting it from the Keras Applications library. Keras Applications provide the deep learning models that have pretrained weights and these models are suitable used for feature extraction, prediction and finetuning as well[27]. EfficientNetV2-b0 is chosen with the weight of ImageNet of an image dataset. ImageNet is the primary dataset for pre-trained deep learning models[28]. The "include_top" attribute needed to state as False to remove the top layer of the EfficientNetv2-b0 classifier to worked as a fixed feature extractor. Other than that, by setting "model.trainable" to False, the model pretrained weights will be freeze as doesn't update the weights parameter during training.

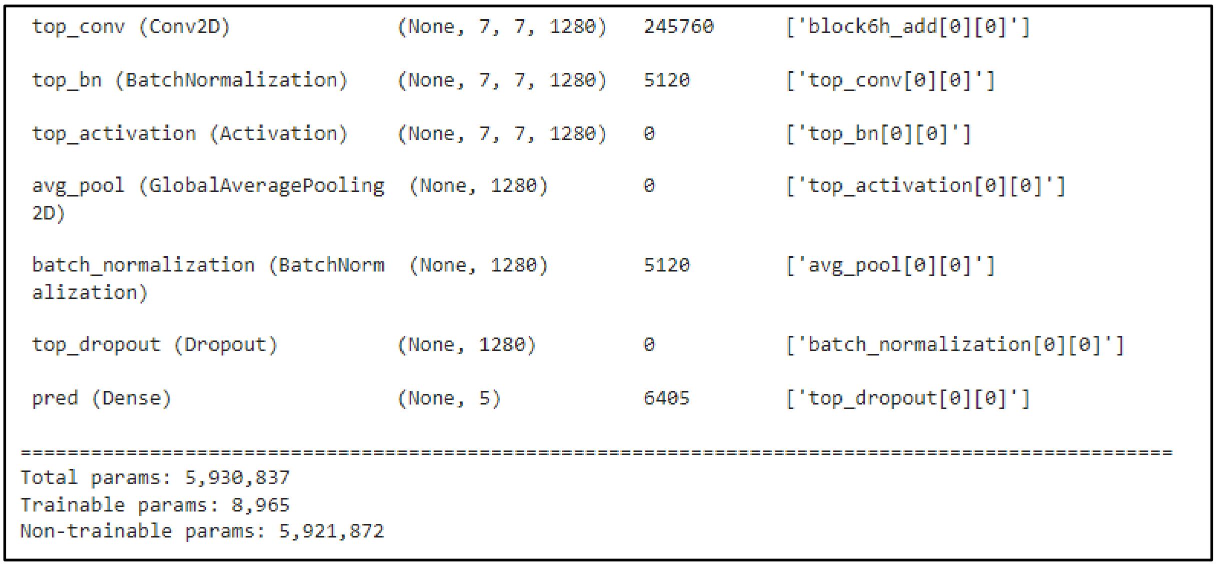
There are total of four layers implemented to rebuild the top layer for the model. The first layer to be rebuild is the “GlobalAveragePooling2D” layer which is a pooling layer. Pooling layers used to reduce the dimensions of feature maps. Other than that, pooling layers have significantly less parameters which make the model easier to train and fine tuning. The next layer to be rebuild is the “BatchNormalization” layer, it is worked as a stabilizer as extra layers are added into the model layers. Continuously, the “Dropout” layer implemented is to prevent the overfitting on training data by nullifies some neurons output towards next layer. Finally, the deeply connected neural network layer is always referred to as the dense layer[29]. “Dense” layer is to receive input from previous layer neurons (“Dropout” layer) and perform matrix-vector multiplication to change vector dimension and create output with 5 units which are same as the total of five classes the model modified to classify. The activation function used in the “Dense” layer is the “softmax” which will convert “logits” score to “probit” to probability distribution for each classes classification result. The “Optimizer” used is the Adam, Adam is the optimizer that will change the learning rate adaptively in every training epoch on backpropagation to eventually change the model weights parameters. The loss function used in this model will be using the “CategoricalCrossEntropy” which will assist the model to decide among five classes in a manner of [0, 1, 0, 0, 0] based on the probability of each class.

Based on Figure 4.2.3.3.2, the model summary showed the rebuilt top layers that have implemented, trainable parameters and non-trainable parameters. The number of trainable parameters which is 8965 is based on the rebuilt top layers as it is needed for fine tune the weights parameters.



```
Model: "EfficientNetV2-B0"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to                     
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []                               
                                )]                                                                
                                                                                                  
 rescaling (Rescaling)          (None, 224, 224, 3)  0           ['input_1[0][0]']                
                                                                                                  
 normalization (Normalization)  (None, 224, 224, 3)  0           ['rescaling[0][0]']              
                                                                                                  
 stem_conv (Conv2D)             (None, 112, 112, 32  864         ['normalization[0][0]']          
                                )                                                                 
                                                                                                  
 stem_bn (BatchNormalization)   (None, 112, 112, 32  128         ['stem_conv[0][0]']              
                                )                                                                 
                                                                                                  
 stem_activation (Activation)   (None, 112, 112, 32  0           ['stem_bn[0][0]']                
                                )                                                                 
                                                                                                  
 block1a_project_conv (Conv2D)  (None, 112, 112, 16  4608        ['stem_activation[0][0]']        
                                )                                                                 
                                                                                                  
 block1a_project_bn (BatchNorma  (None, 112, 112, 16  64         ['block1a_project_conv[0][0]']   
 lization)                      )                                                                 
                                                                                                  
 block1a_project_activation (Ac  (None, 112, 112, 16  0          ['block1a_project_bn[0][0]']     
 tivation)                      )                                                                 
                                                                                                  
 block2a_expand_conv (Conv2D)   (None, 56, 56, 64)   9216        ['block1a_project_activation[0][0
                                                                 ]']                              
```

Figure 4.2.3.3.1 EfficientNetV2-b0 initial layers summary



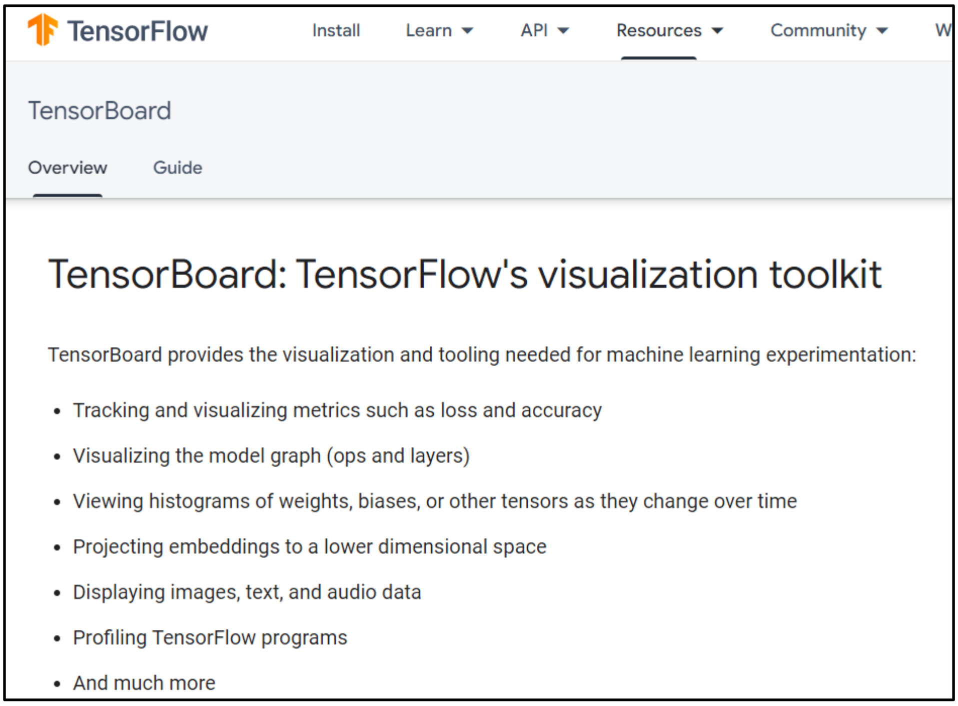
```
 top_conv (Conv2D)              (None, 7, 7, 1280)   245760      ['block6h_add[0][0]']            
                                                                                                  
 top_bn (BatchNormalization)    (None, 7, 7, 1280)   5120        ['top_conv[0][0]']               
                                                                                                  
 top_activation (Activation)    (None, 7, 7, 1280)   0           ['top_bn[0][0]']                 
                                                                                                  
 avg_pool (GlobalAveragePooling  (None, 1280)        0           ['top_activation[0][0]']         
 2D)                                                                                              
                                                                                                  
 batch_normalization (BatchNorm  (None, 1280)        5120        ['avg_pool[0][0]']               
 alization)                                                                                       
                                                                                                  
 top_dropout (Dropout)          (None, 1280)         0           ['batch_normalization[0][0]']    
                                                                                                  
 pred (Dense)                   (None, 5)            6405        ['top_dropout[0][0]']            
                                                                                                  
==================================================================================================
Total params: 5,930,837
Trainable params: 8,965
Non-trainable params: 5,921,872
```

Figure 4.2.3.3.2 EfficientNetV2-b0 rebuilt top layers summary

4.2.4 Training Process

4.2.4.1 Freeze Training (Fixed Feature Extractor)

Before starting the training, it is suggested to utilize the callbacks function available in Keras. TensorBoard is the callbacks function that will record the history on accuracy, loss in each running epoch and saved in a log file.



TensorBoard: TensorFlow's visualization toolkit

TensorBoard provides the visualization and tooling needed for machine learning experimentation:

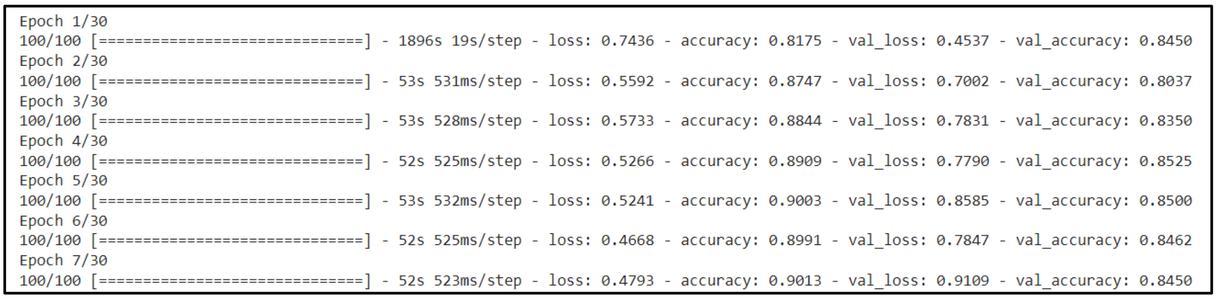
- Tracking and visualizing metrics such as loss and accuracy
- Visualizing the model graph (ops and layers)
- Viewing histograms of weights, biases, or other tensors as they change over time
- Projecting embeddings to a lower dimensional space
- Displaying images, text, and audio data
- Profiling TensorFlow programs
- And much more

Figure 4.2.4.1.1 TensorBoard From TensorFlow

There are two values need to be retrieved which are the steps per epoch and validation steps. Given the total samples size in training set in this project will be 3200 of images and batch size hyperparameter have set to 32. The value of steps per epoch will be 100 which means in every epoch the model will be going through 100 iterations to train each image in training set. Validation set sample size is 800 and batch size will be the same as 32. The value of validation steps will be calculated as 25 which means in every epoch the model will be going through 25 iterations to evaluate for each image in validation set.

Based on Figure 4.2.4.1.2, ".fit" function is the function to start the training of the model. There are several attributes need to be filled up before it can start to train. "dataset_two_train_gen" is the input for the training images set to train the model. "epochs" is the hyperparameter that set on how many times the model going to train to passes through the entire training dataset.

"validation_data" will be the input for the validation dataset. The other attributes have discussed in previous paragraph and will be insert to ".fit" function.

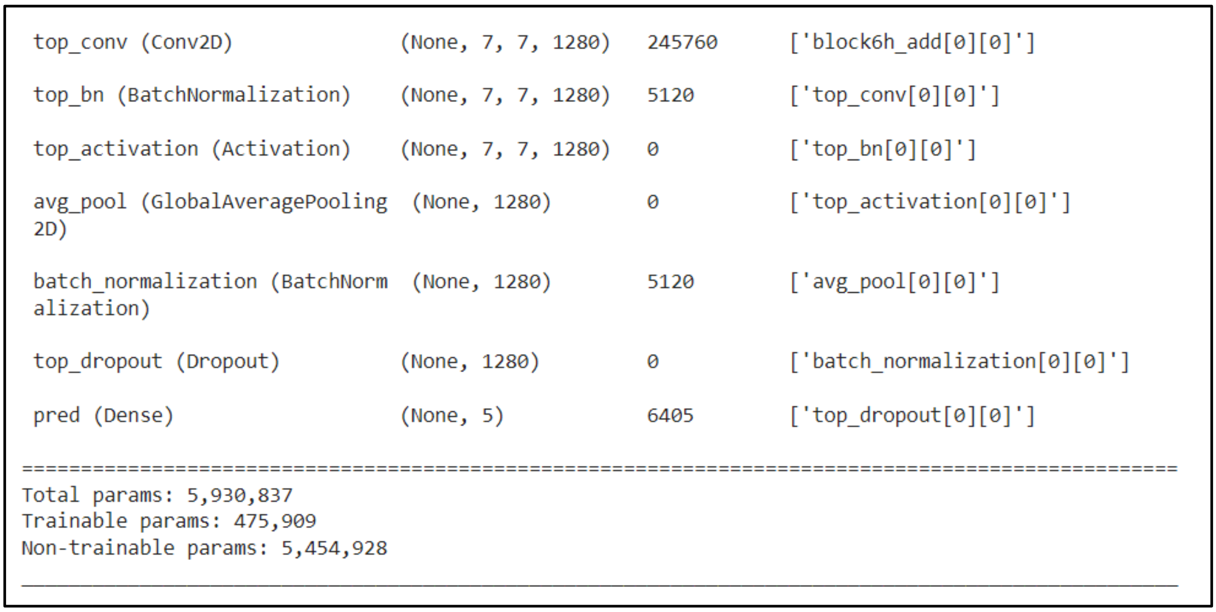


```
Epoch 1/30
100/100 [==============================] - 1896s 19s/step - loss: 0.7436 - accuracy: 0.8175 - val_loss: 0.4537 - val_accuracy: 0.8450
Epoch 2/30
100/100 [==============================] - 53s 531ms/step - loss: 0.5592 - accuracy: 0.8747 - val_loss: 0.7002 - val_accuracy: 0.8037
Epoch 3/30
100/100 [==============================] - 53s 528ms/step - loss: 0.5733 - accuracy: 0.8844 - val_loss: 0.7831 - val_accuracy: 0.8350
Epoch 4/30
100/100 [==============================] - 52s 525ms/step - loss: 0.5266 - accuracy: 0.8909 - val_loss: 0.7790 - val_accuracy: 0.8525
Epoch 5/30
100/100 [==============================] - 53s 532ms/step - loss: 0.5241 - accuracy: 0.9003 - val_loss: 0.8585 - val_accuracy: 0.8500
Epoch 6/30
100/100 [==============================] - 52s 525ms/step - loss: 0.4668 - accuracy: 0.8991 - val_loss: 0.7847 - val_accuracy: 0.8462
Epoch 7/30
100/100 [==============================] - 52s 523ms/step - loss: 0.4793 - accuracy: 0.9013 - val_loss: 0.9109 - val_accuracy: 0.8450
```

Figure 4.2.4.1.2 Commence training process with fit function

4.2.4.2 Unfreeze Training (Fine Tuning Few Top Layers)

A defined "unfreeze_model" function is to accept a model in its argument. In the function, it will run a for loop to going through top 10 layers in the model and excluding any "BatchNormalization" layer to set those layers to be able to train for updating the weights parameters. The reason behind of retain "BatchNormalization" layer to non-trainable is to avoid the representation or the batch statistics learned by the model to get wreak havoc[30]. Based on Figure 4.2.4.2.1, the unfreeze model will have total of 475,909 trainable parameters for fine tuning.



```
 top_conv (Conv2D)           (None, 7, 7, 1280)   245760      ['block6h_add[0][0]']

 top_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['top_conv[0][0]']

 top_activation (Activation)  (None, 7, 7, 1280)  0           ['top_bn[0][0]']

 avg_pool (GlobalAveragePooling  (None, 1280)     0           ['top_activation[0][0]']
 2D)

 batch_normalization (BatchNorm  (None, 1280)     5120        ['avg_pool[0][0]']
 alization)

 top_dropout (Dropout)       (None, 1280)         0           ['batch_normalization[0][0]']

 pred (Dense)                (None, 5)            6405        ['top_dropout[0][0]']

==================================================================================================
Total params: 5,930,837
Trainable params: 475,909
Non-trainable params: 5,454,928
__________________________________________________________________________________________________
```

Figure 4.2.4.2.1 Unfreeze model trainable parameters

By input the previously freeze trained model into the “unfreeze_model” function will result the model which having top 10 layers to set to trainable except for the “BatchNormalization”. The approach of unfreeze the layers for fine tune the model is to significantly improve its performance by incrementally adapting the pretrained features to the training data and aim to retraining it with a lower learning rate[30].

4.2.5 Testing Process

4.2.5.1 Model Evaluation and Testing

There are several techniques implemented to test and evaluate the performance of trained model before converting it to TensorFlow Lite model for deploy into Android application.

TensorBoard

TensorBoard, callbacks function available from TensorFlow which provide it as a powerful visualization tool for research to inspect on the history log of each epoch run during training. The different color in TensorBoard graph line of Figure 4.2.5.1.1 and Figure 4.2.5.1.2 can be refer as shown in Table 4.2.5.1. By using the TensorBoard in callbacks in model(s) training will significantly increase the visualization and understanding towards those trained model.

Color(s)	Training/Validation	Freeze/Unfreeze Model
Red	Training	Unfreeze
Light Blue	Validation	Unfreeze
Orange	Training	Freeze
Dark Blue	Validation	Freeze

Table 4.2.5.1 Color differentiation in TensorBoard

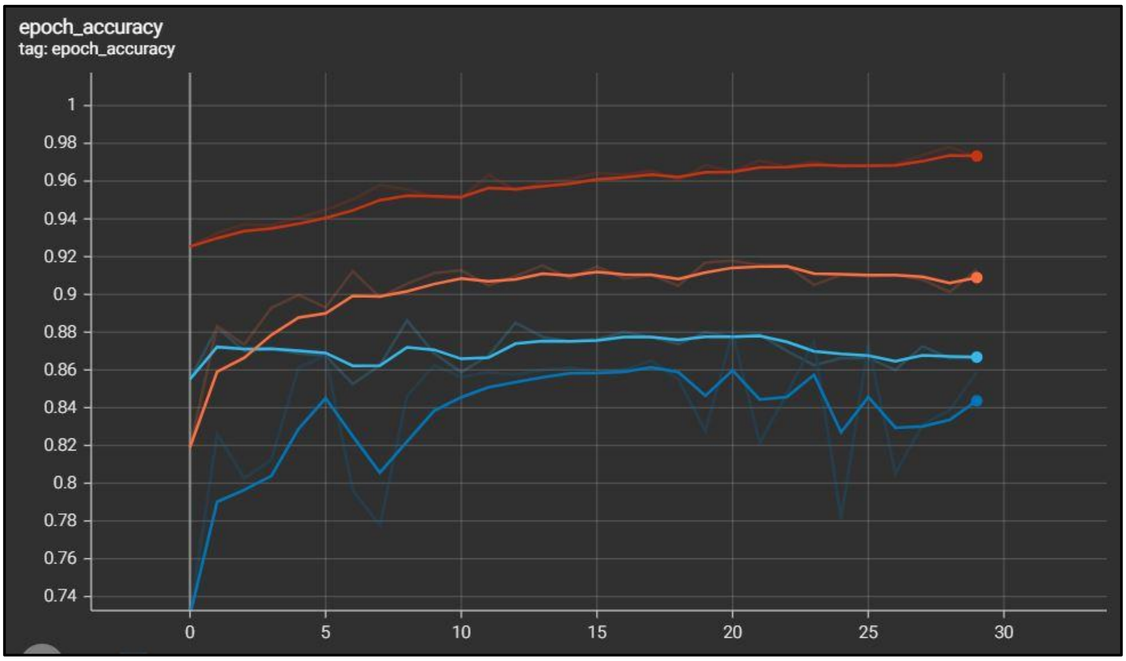


Figure 4.2.5.1.1 Accuracy graph for training and validation in both Freeze and Unfreeze Model

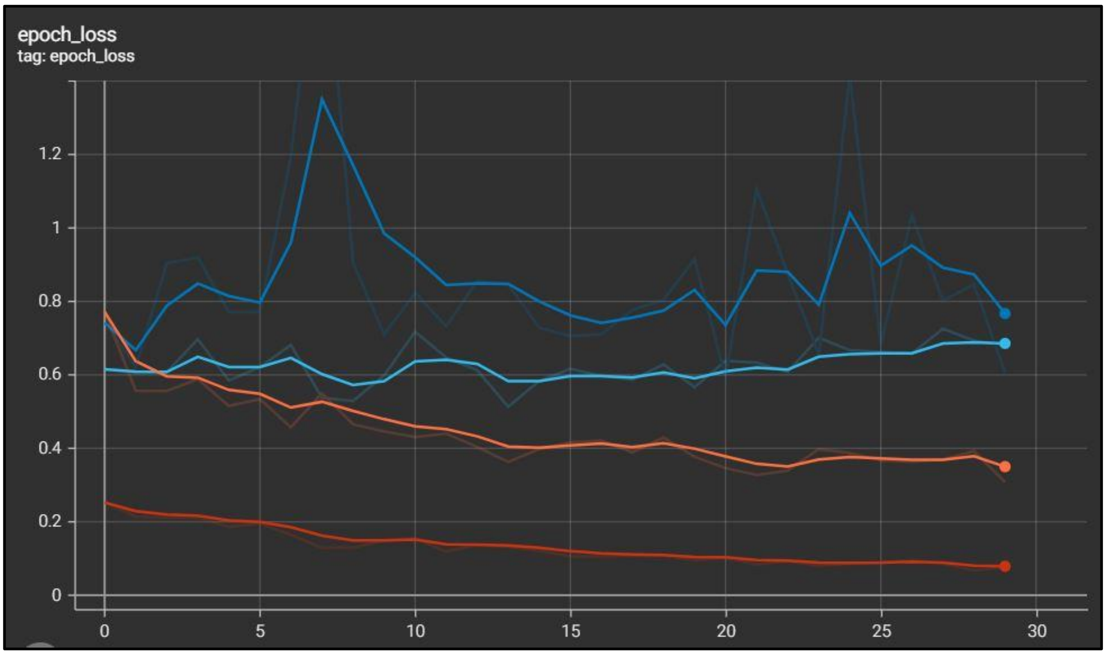
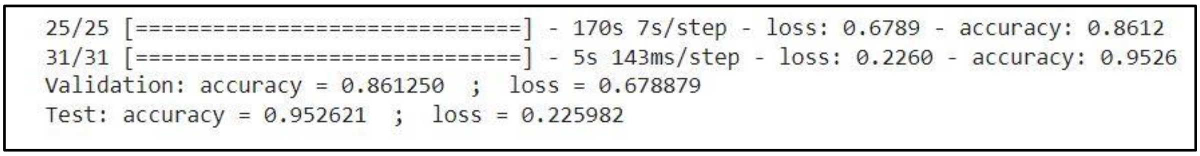


Figure 4.2.5.1.2 Loss graph for training and validation in both Freeze and Unfreeze Model

Evaluate Model With Validation and Testing Set

This is also another technique to evaluate the trained model which also using “.evaluate” function from Model provided by TensorFlow[31]. The function will return two metrics which are accuracy and loss values after finished evaluating all images in validation and test dataset.



```
25/25 [==============================] - 170s 7s/step - loss: 0.6789 - accuracy: 0.8612
31/31 [==============================] - 5s 143ms/step - loss: 0.2260 - accuracy: 0.9526
Validation: accuracy = 0.861250  ;  loss = 0.678879
Test: accuracy = 0.952621  ;  loss = 0.225982
```

Figure 4.2.5.1.7 Performance metrics for Validation and Test dataset

Confusion Matrix

Other than that, confusion matrix is also a good technique to visualize the model evaluation. This can carry out by using ".predict" function in Model class from TensorFlow[31] to retrieve the predicted true and compare to true labels using the confusion matrix from the sklearn library. The test dataset with 200 images each class to be evaluated using the trained model and having the result based on the Figure 4.2.5.1.3. The y-axis of the confusion matrix is representing the True labels and x-axis is the representing the predicted labels. The blue boxes in the confusion matrix are represented wrongly classified. While the low affinity color of orange represented as precisely classified.

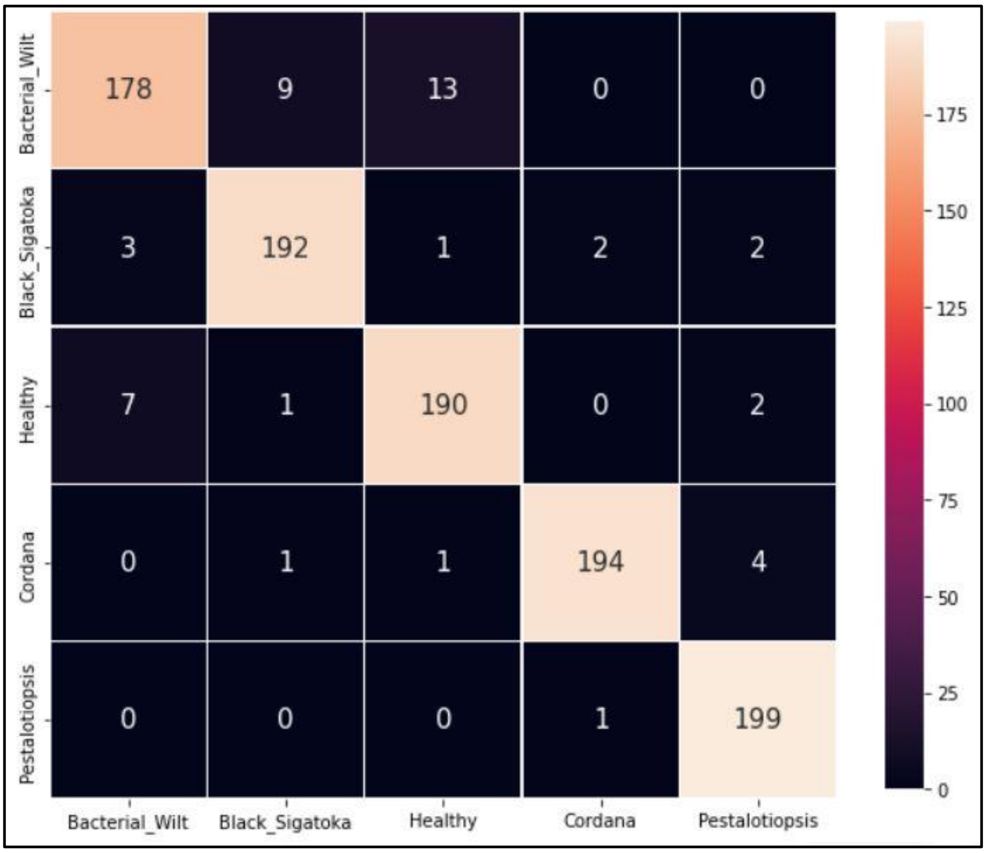
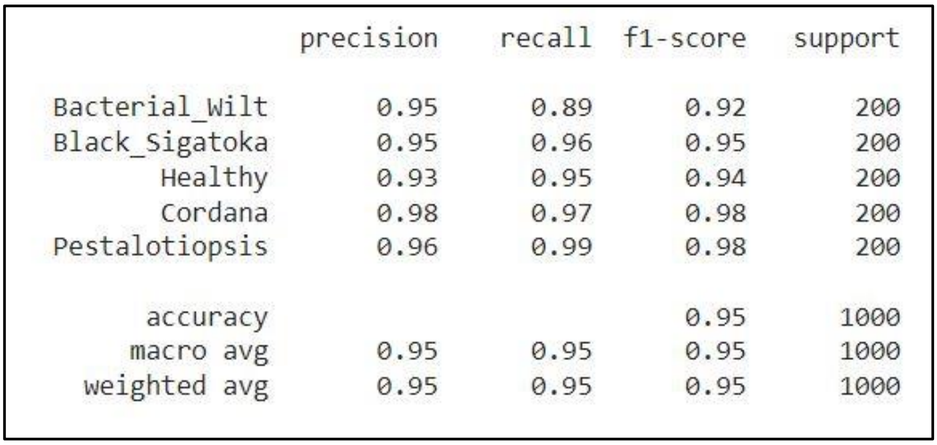


Figure 4.2.5.1.3 Confusion Matrix for Unfreeze Model

Classification Report

Classification report function is also provided in sklearn library to compute the result on each class by considering the predicted labels and true labels. Precision score is aimed to test the model to not label an incorrect sample as a positive label. Recall score is aimed to test the model to retrieve all positive samples. F1-score is the weighted average of both Precision score and Recall score and evaluate how model perform on each different class.

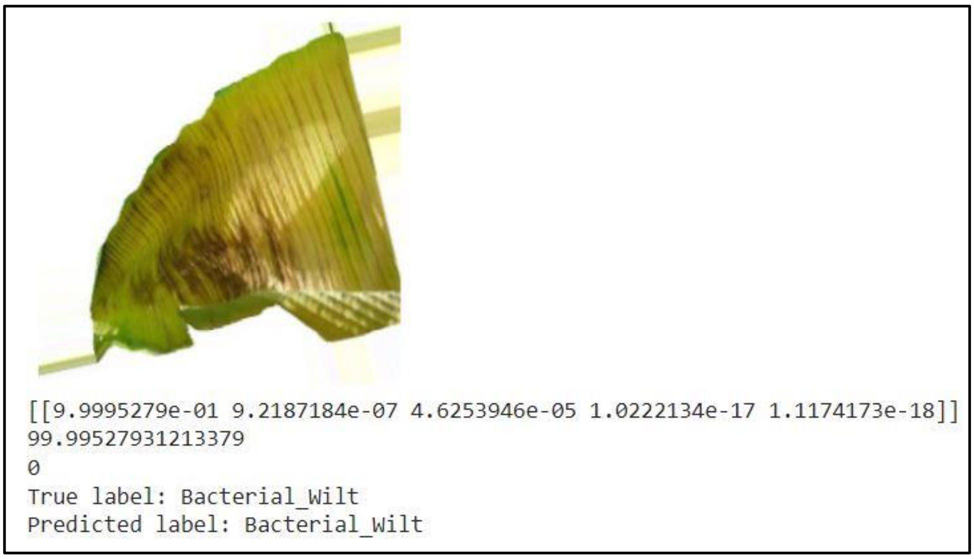


```
                precision    recall  f1-score   support

 Bacterial_Wilt       0.95      0.89      0.92       200
 Black_Sigatoka       0.95      0.96      0.95       200
        Healthy       0.93      0.95      0.94       200
        Cordana       0.98      0.97      0.98       200
 Pestalotiopsis       0.96      0.99      0.98       200

       accuracy                           0.95      1000
      macro avg       0.95      0.95      0.95      1000
   weighted avg       0.95      0.95      0.95      1000
```

Figure 4.2.5.1.4 Classification Report on Unfreeze Model

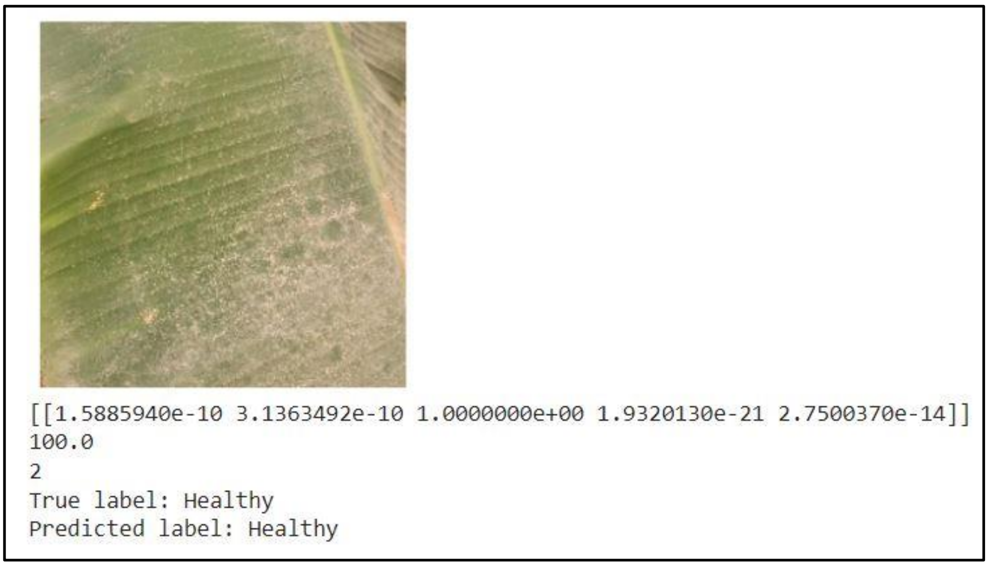
Testing on One Image

This is an optional technique to visualize a random picked image from validation or test dataset to predict the label. Model class from TensorFlow also provided another function ".predict" to predict on single image and result the predicted scores, predicted label and true label as well[31].



```
[[9.9995279e-01 9.2187184e-07 4.6253946e-05 1.0222134e-17 1.1174173e-18]]
99.99527931213379
0
True label: Bacterial_Wilt
Predicted label: Bacterial_Wilt
```

Figure 4.2.5.1.5 Predict an Image from Validation Dataset



```
[[1.5885940e-10 3.1363492e-10 1.0000000e+00 1.9320130e-21 2.7500370e-14]]
100.0
2
True label: Healthy
Predicted label: Healthy
```

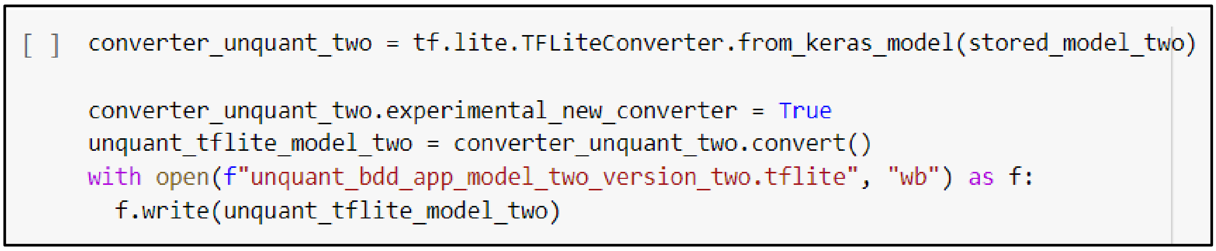
Figure 4.2.5.1.6 Predict an Image from Test Dataset

4.2.6 Output Process

In this output process, the aim is to convert the trained TensorFlow Keras model which have been evaluated and tested to TensorFlow Lite model. Other than that, it also recommended to run inferencing test on the converted TensorFlow Lite model. Therefore, it can be ensuring the performance was great when deploy into the Android application.

4.2.6.1 Convert TensorFlow Keras to TensorFlow Lite

Module used is the TFLITEConverter that provided in TensorFlow Lite to perform the conversion shown in Figure 4.2.6.1.1. The converter will be accepting a TensorFlow Keras model and perform a "convert()" function to result a converted TensorFlow Lite model. The TensorFlow Lite model will be quantized, and the model's parameters will be in 32-bit floating point numbers by default to result smaller size and faster computations after deploying into mobile application[32].

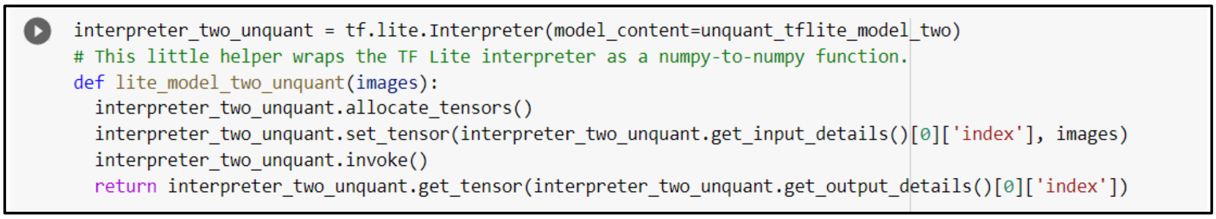


```
converter_unquant_two = tf.lite.TFLiteConverter.from_keras_model(stored_model_two)

converter_unquant_two.experimental_new_converter = True
unquant_tflite_model_two = converter_unquant_two.convert()
with open(f"unquant_bdd_app_model_two_version_two.tflite", "wb") as f:
  f.write(unquant_tflite_model_two)
```

Figure 4.2.6.1 TensorFlow Keras to TensorFlow Lite model

4.2.6.2 Inferencing Test on TensorFlow Lite model

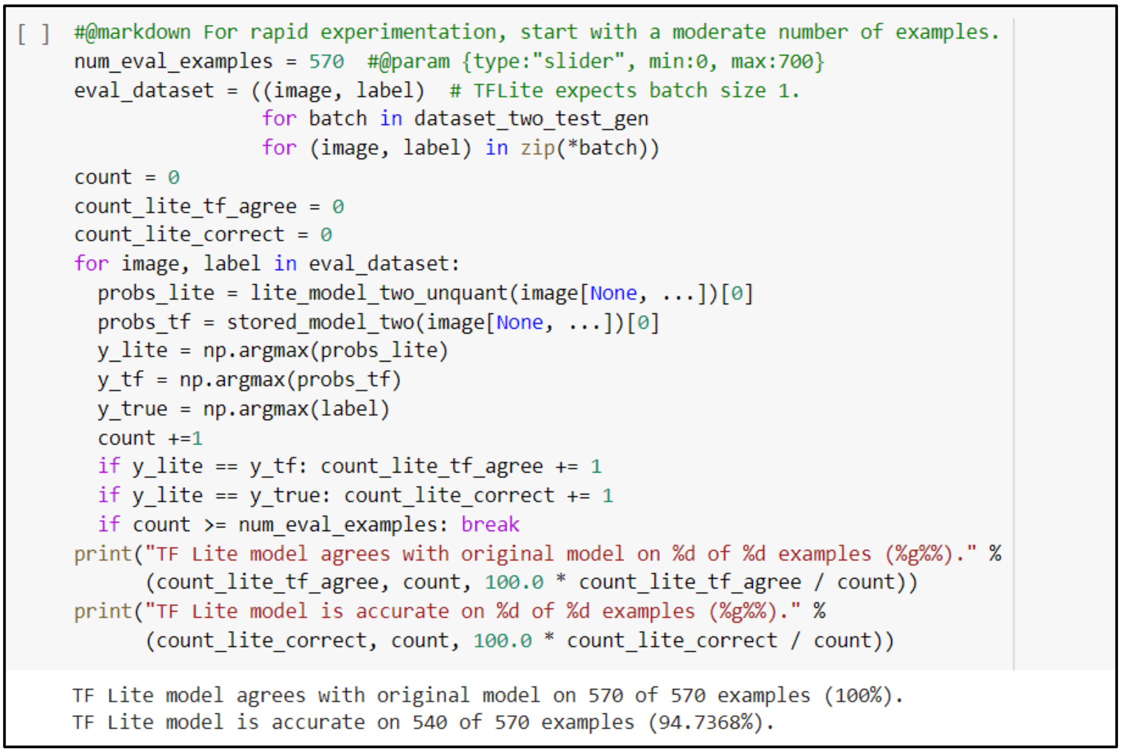
To run the TensorFlow Lite model by normal basis is not possible, however TensorFlow have provided the solution by using the “Interpreter”. For any model obtained from “TfLiteConverter” can run in python with the help from “Interpreter”[33]. Based on Figure 4.2.6.2.1, the defined function is to wrap up the interpreter to work in a numpy-to-numpy condition for predicting the image input.



```
interpreter_two_unquant = tf.lite.Interpreter(model_content=unquant_tflite_model_two)
# This little helper wraps the TF Lite interpreter as a numpy-to-numpy function.
def lite_model_two_unquant(images):
  interpreter_two_unquant.allocate_tensors()
  interpreter_two_unquant.set_tensor(interpreter_two_unquant.get_input_details()[0]['index'], images)
  interpreter_two_unquant.invoke()
  return interpreter_two_unquant.get_tensor(interpreter_two_unquant.get_output_details()[0]['index'])
```

Figure 4.2.6.2.1 Interpreter from TensorFlow

Based on Figure 4.2.6.2.2, the dataset used to do the inferencing test is the test dataset images. “stored_model_two” is the TensorFlow Keras model that have trained previously to cross validate the image prediction with interpreter (TensorFlow Lite model). Since both of the model agreed with 570 image samples class label, hence the accuracy for both of them will be the same. Therefore, the equivalent between predicted label predicted by interpreter and true label is 540 out of 570 with accuracy of 94.7368%.



```
[ ]  #@markdown For rapid experimentation, start with a moderate number of examples.
     num_eval_examples = 570  #@param {type:"slider", min:0, max:700}
     eval_dataset = ((image, label)  # TFLite expects batch size 1.
                     for batch in dataset_two_test_gen
                     for (image, label) in zip(*batch))
     count = 0
     count_lite_tf_agree = 0
     count_lite_correct = 0
     for image, label in eval_dataset:
       probs_lite = lite_model_two_unquant(image[None, ...])[0]
       probs_tf = stored_model_two(image[None, ...])[0]
       y_lite = np.argmax(probs_lite)
       y_tf = np.argmax(probs_tf)
       y_true = np.argmax(label)
       count +=1
       if y_lite == y_tf: count_lite_tf_agree += 1
       if y_lite == y_true: count_lite_correct += 1
       if count >= num_eval_examples: break
     print("TF Lite model agrees with original model on %d of %d examples (%g%%)." %
           (count_lite_tf_agree, count, 100.0 * count_lite_tf_agree / count))
     print("TF Lite model is accurate on %d of %d examples (%g%%)." %
           (count_lite_correct, count, 100.0 * count_lite_correct / count))

     TF Lite model agrees with original model on 570 of 570 examples (100%).
     TF Lite model is accurate on 540 of 570 examples (94.7368%).
```

Figure 4.2.6.2.2 Inferencing Test

4.3 Banana Diseases Detection Mobile Application (System Design)

In this section, is to show the steps that implemented to develop an Android application that worked with a TensorFlow Lite model as the computer vision task feature. Other than that, there are several important features implemented in the application as well. Furthermore, there are several diagrams that able to visualize on how the Android application system designed.

4.3.1 Application System Flowchart

Based on Figure 4.3.1, it is illustrated on how the mobile application system operate including its important features. For the following sub-chapters will elaborate more how features are designed and including the block diagram for in-depth operation visualization.

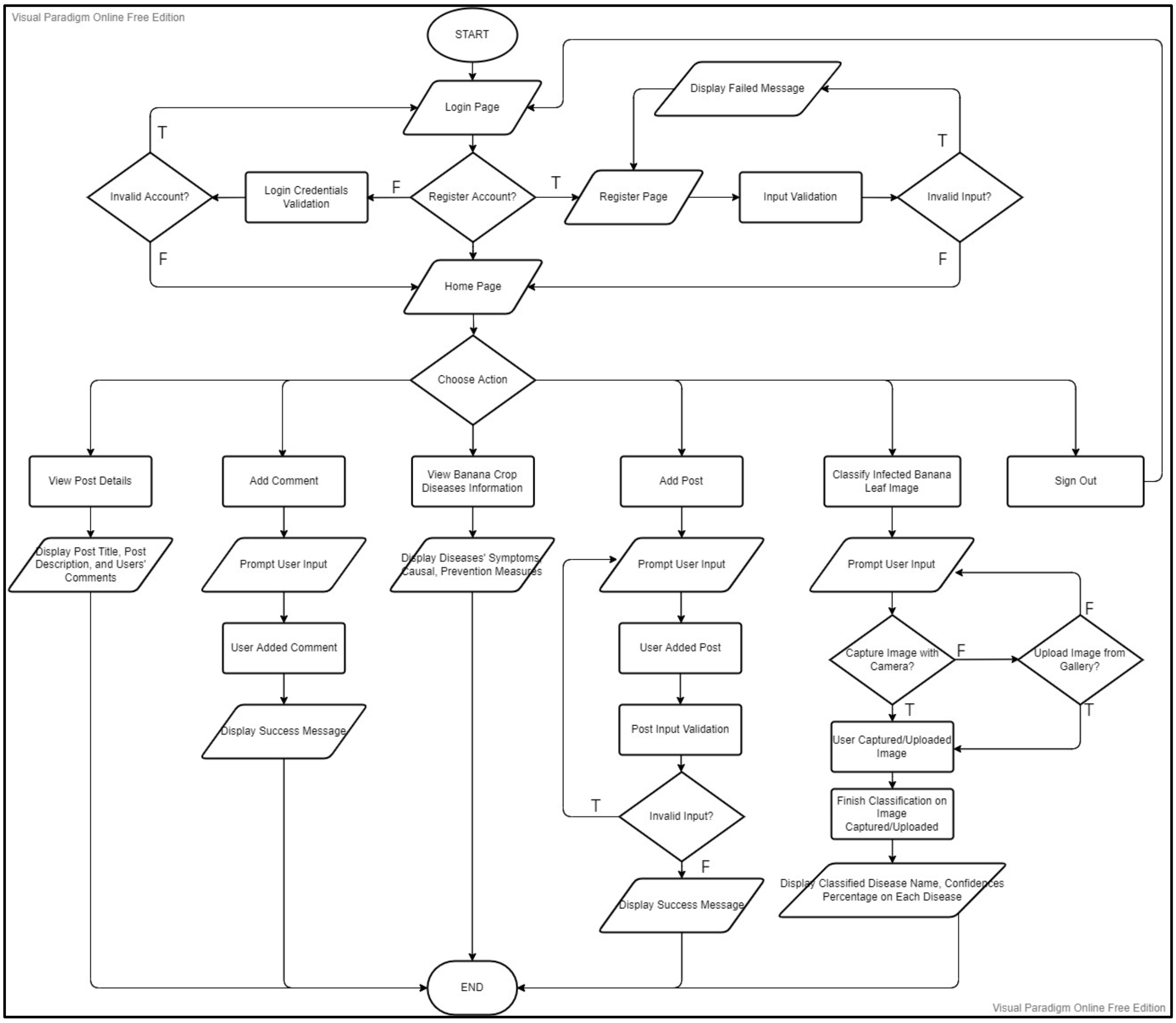

Visual Paradigm Online Free Edition
START
Login Page
Display Failed Message
T
Invalid Account?
Login Credentials Validation
F
Register Account?
T
Register Page
Input Validation
Invalid Input?
F
Home Page
Choose Action
View Post Details
Add Comment
View Banana Crop Diseases Information
Add Post
Classify Infected Banana Leaf Image
Sign Out
Display Post Title, Post Description, and Users' Comments
Prompt User Input
Display Diseases' Symptoms, Causal, Prevention Measures
Prompt User Input
Prompt User Input
User Added Comment
User Added Post
Capture Image with Camera?
Upload Image from Gallery?
Display Success Message
Post Input Validation
User Captured/Uploaded Image
Finish Classification on Image Captured/Uploaded
Invalid Input?
Display Classified Disease Name, Confidences Percentage on Each Disease
Display Success Message
END
Visual Paradigm Online Free Edition

Figure 4.3.1 Application System Flowchart

4.3.2 Account Registration and Login

Account Registration Block Diagram

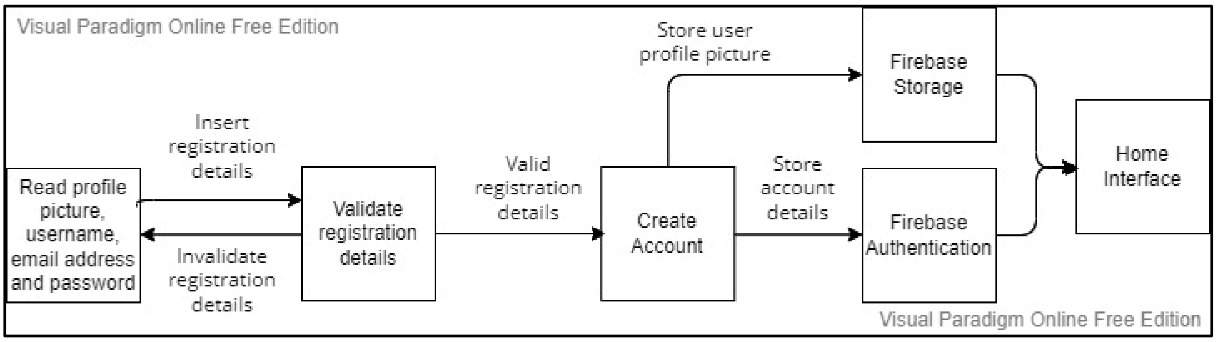

Visual Paradigm Online Free Edition
Read profile picture, username, email address and password
Insert registration details
Invalidate registration details
Validate registration details
Valid registration details
Create Account
Store user profile picture
Store account details
Firebase Storage
Firebase Authentication
Home Interface
Visual Paradigm Online Free Edition

Figure 4.3.2.1 Account Registration Block Diagram

Account Login Block Diagram

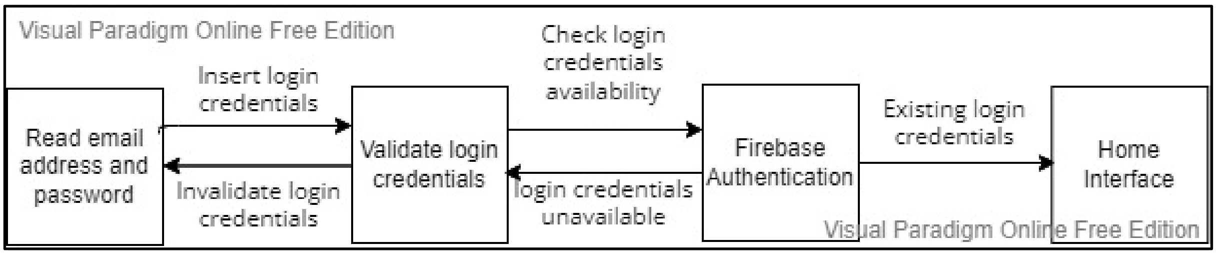

Visual Paradigm Online Free Edition
Read email address and password
Insert login credentials
Invalidate login credentials
Validate login credentials
Check login credentials availability
login credentials unavailable
Firebase Authentication
Existing login credentials
Home Interface
Visual Paradigm Online Free Edition

Figure 4.3.2.2 Account Login Block Diagram

Account Registration Design

Based on Figure 4.3.2.1, registration backend activity design involved using the Firebase Authentication and Firebase Storage service provided by Firebase platform for mobile development. Firebase Authentication is only responsible on creating account during account registration procedure. Firebase Storage used as to store each registered user’s profile picture stored in a folder created. Account registration interface is designed and implemented as an individual Register Activity class extends from “AppCompatActivity” in Java Android application development. Users are required to insert their selected image as profile picture, username, email address and password when interacting with the register interface. The Register Activity class is implemented to validate every single input field from users and display necessary message if anything gone wrong. Once users’ registration details input succeeds the validation, the Register Activity class method will invoke the necessary action. For example, it will store the users’ profile picture in the user photo folder in Firebase Storage and creating an account in Firebase Authentication by using users' account registration details. Once the account registration procedure succeeded, users are redirected to the home interface.

Account Login Design

Based on Figure 4.3.2.2, login backend activity design involved using Firebase Authentication solely. Firebase Authentication is only responsible on validating account login credentials during account login procedure. Account login interface is designed and implemented as an individual Login Activity class extends from "AppCompatActivity" in Java Android application development. Users are required to insert their email address and password when interacting with the login interface. The Login Activity class is implemented to validate every single input field from users and display necessary message if anything gone wrong. Once users' login credentials input succeeds the validation, the Login Activity class method will invoke the necessary action. For example, Firebase Authentication will validate on the users' login credentials and if invalid login credentials will display necessary failed login message. Once the account login procedure succeeded, users are redirected to the home interface.

4.3.3 Home Interface (Community Feature)

Home Interface Block Diagram

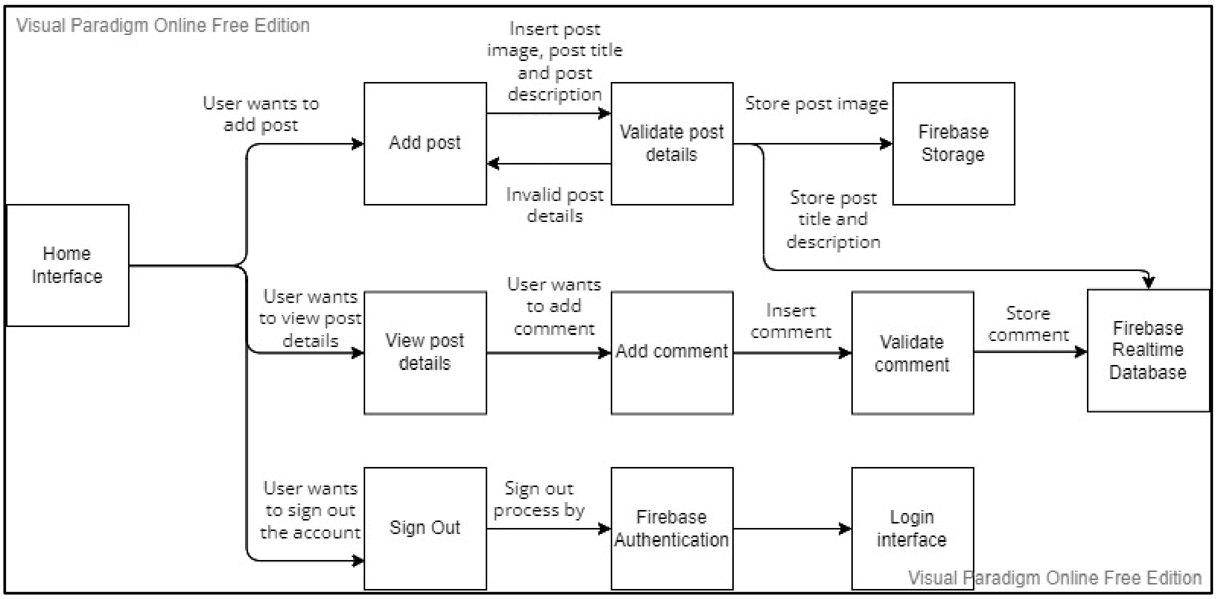

Visual Paradigm Online Free Edition
Home Interface
User wants to add post
Add post
Insert post image, post title and post description
Validate post details
Invalid post details
Store post image
Firebase Storage
Store post title and description
User wants to view post details
View post details
User wants to add comment
Add comment
Insert comment
Validate comment
Store comment
Firebase Realtime Database
User wants to sign out the account
Sign Out
Sign out process by
Firebase Authentication
Login interface
Visual Paradigm Online Free Edition

Figure 4.3.3 Home Interface Block Diagram

Home Interface Design

Based on Figure 4.3.3, the home interface activity backend design involved using the Firebase Storage, Firebase Authentication and Firebase Realtime Database. The Firebase Realtime Database storing object as a JSON object with key and value manner. There are total of four different features available to access from home interface which are view selected post details, adding post, sign out, and adding comment. Once users succeeds either in login or account registration procedure, they are redirected to home interface, where they will be able to view a scrollable list of post available in the system in real-time.

Other than that, they are allowed to click on a post to view the post details which will redirect them to another activity of Post Details Activity class. In post details activity, users can view the image of the post, title, description, the user’s username that added the post, postdate and comment section in the post. Moreover, users can decide whether to add a comment in the post during the post details activity. Post Details Activity class will invoke method to store the added comment into the Firebase Realtime Database as a JSON object if it is not a blank comment and display validation status message. Once it succeeded, the added comment will be display in the comment section in real-time.

Furthermore, the add post feature is represented as a button and a add post window will popup once add post button clicked. The add post window will display current user profile picture, post title input field, post description input field and post image input field and add post execution button. Users are prompted to insert every input field and click the add post execution button if users are satisfied with their input. Every user’s input will be validated and display success message if inputs are valid. Firebase Storage will store the post image and Firebase Realtime Database will store the post details. After that, newly added post will display in the home interface scrollable list of post in real-time.

The last feature available at home interface is the sign out function. The sign out function is to allow user to logout of the application using the Firebase Authentication method on signing out a user. Once the user signed out, user is redirected to the login interface.

4.3.4 Detector Interface (Banana Diseases Detection feature)

Detector Interface Block Diagram

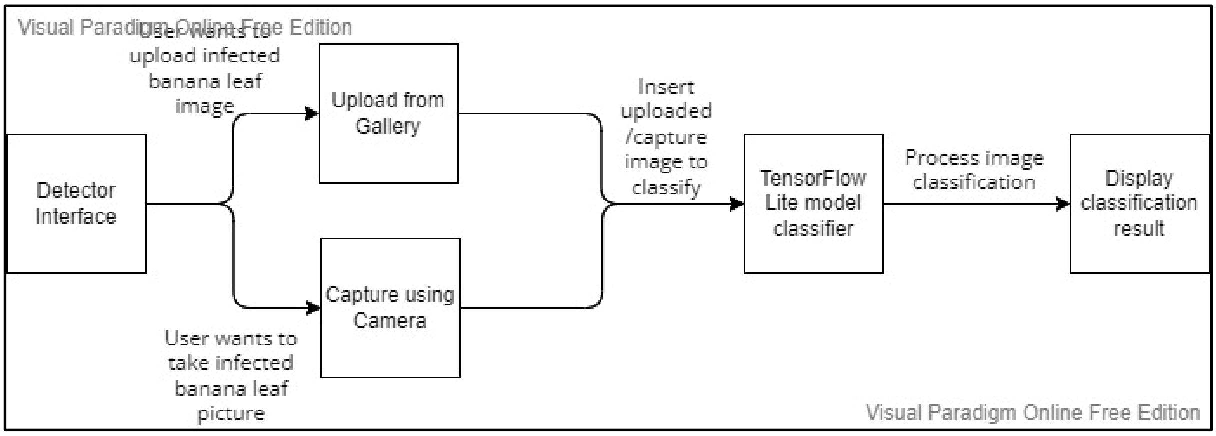

Visual Paradigm Online Free Edition
User wants to upload infected banana leaf image
Upload from Gallery
Insert uploaded /capture image to classify
Detector Interface
TensorFlow Lite model classifier
Process image classification
Display classification result
Capture using Camera
User wants to take infected banana leaf picture
Visual Paradigm Online Free Edition

Figure 4.3.4 Detector Interface Block Diagram

Detector Interface Design

Based on Figure 4.3.4, a TensorFlow Lite model classifier is deployed into the Android application development after its finished trained and converted which stated in sub-chapter of 4.2.6. The deployed TensorFlow Lite model classifier is worked as a classifier class in the application development. Classifier Activity need to be implemented to access the classifier class to perform image classification on the image uploaded or captured by the user and to display the classification result.

In the Detector interface, there are two button selection for the user to choose either upload an image or capture a picture of the infected banana leaf to perform classification and display the classification result. Once the classifier finished the classification process, the interface will display the classified disease name and a list of confidences percentage on each different disease.

4.3.5 Banana Crop Diseases Interface (Banana Diseases Information)

Banana Crop Diseases Interface Block Diagram

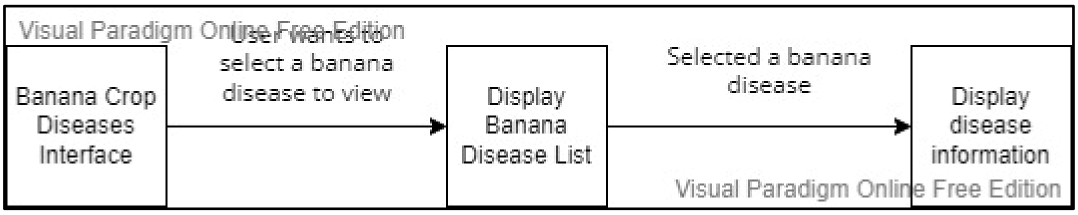

Visual Paradigm Online Free Edition
Banana Crop Diseases Interface
User wants to select a banana disease to view
Display Banana Disease List
Selected a banana disease
Display disease information
Visual Paradigm Online Free Edition

Figure 4.3.5 Banana Crop Diseases Interface Block Diagram

Banana Crop Diseases Interface Design

Based on Figure 4.3.5, this is an interface which implemented to show user the listed banana diseases information. “DiseaseDetailsActivity” Activity implemented to organize the operation of the banana crop diseases interface. Each disease information implemented and store as in-app storage in “strings.xml” which is a string object.

A list of banana diseases name and corresponding banana disease image display in a list format. User can click either one of them to proceed and view disease’s further information. After selected or clicked a particular disease, user will be redirected to page of the disease. The disease’s image, name, symptoms, causal and prevention measures information will be display for user reference.

Chapter 5 System Implementation

5.1 Hardware Setup

The hardware setup used for developing this mobile application are:

1. Acer Predator Helios 300, Laptop

Operating System	Windows 10 64-bit
Processor	Intel® Core^{TM} i5-10300H @ 2.50GHz
RAM	8GB DDR4-2933
Graphic Card	NVIDIA GeForce RTX2060

Table 5.1.1 Laptop Hardware Specification

2. Huawei Nova 2i (RNE-L22), mobile smartphone

Operating System	EMUI 8.0.0 based on Android 8.0.0
Chipset	64bit – HiSilicon Kirin 659 (16nm)
RAM Memory	4GB LPDDR3
Internal Storage	64GB
Display Resolution	1080x2160pixels
Rear Camera	- Dual Camera - 16 megapixels - 4608x3456 pixels
Front Camera	- Dual Camera - 13 megapixels - 4128x3096 pixels

Table 5.1.2 Mobile Phone Hardware Specification

5.2 Software Setup

The software that has been chosen for developing this system are:

1. TensorFlow (v 2.8.0)

Figure 5.2.1 TensorFlow Logo

TensorFlow is an open-source artificial intelligence library developed by Google Inc.. It used data flow graphs to build models and enable developers to create large-scaled neural network with tremendous layers. This library mainly supports for:

- Classification
- Perception
- Understanding
- Discovering
- Prediction
- Creation

2. Keras

Figure 5.2.2 Keras Logo

It is an API for deep learning which written in Python. Keras is highly dependent on machine learning platform TensorFlow 2. Keras is:

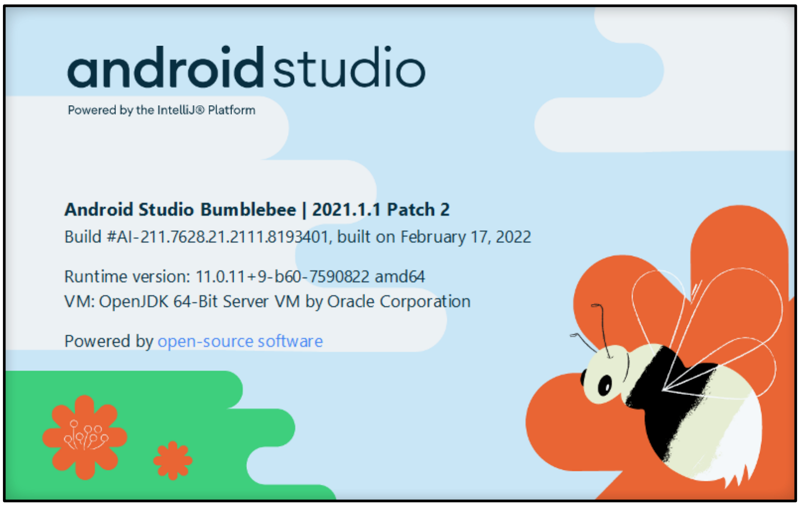
- Simple
- Flexible
- Powerful

3. Google Colaboratory

Figure 5.2.3 Google Colaboratory Logo

Google Colab is a product from Google Research. It is a product that serves anybody to implement their code in python in browser environment provided by Google Colab. Jupyter notebook is used by Google Colab is a hosted service which does not need to setup meanwhile providing the computing resources such as GPUs and TPUs as a free access.

4. Android Studio (Bumblebee, 2021.1.1 Patch 2)



android studio

Powered by the IntelliJ® Platform

Android Studio Bumblebee | 2021.1.1 Patch 2

Build #AI-211.7628.21.2111.8193401, built on February 17, 2022

Runtime version: 11.0.11+9-b60-7590822 amd64

VM: OpenJDK 64-Bit Server VM by Oracle Corporation

Powered by open-source software

Figure 5.2.4 Android Studio Version

Android Studio is based on IntelliJ IDEA to provide Integrated Development Environment (IDE) for Android application development. Other than IntelliJ's powerful code editor and some other developer tools, Android Studio provides various of features which strengthen the productivity during developing Android application, for example:

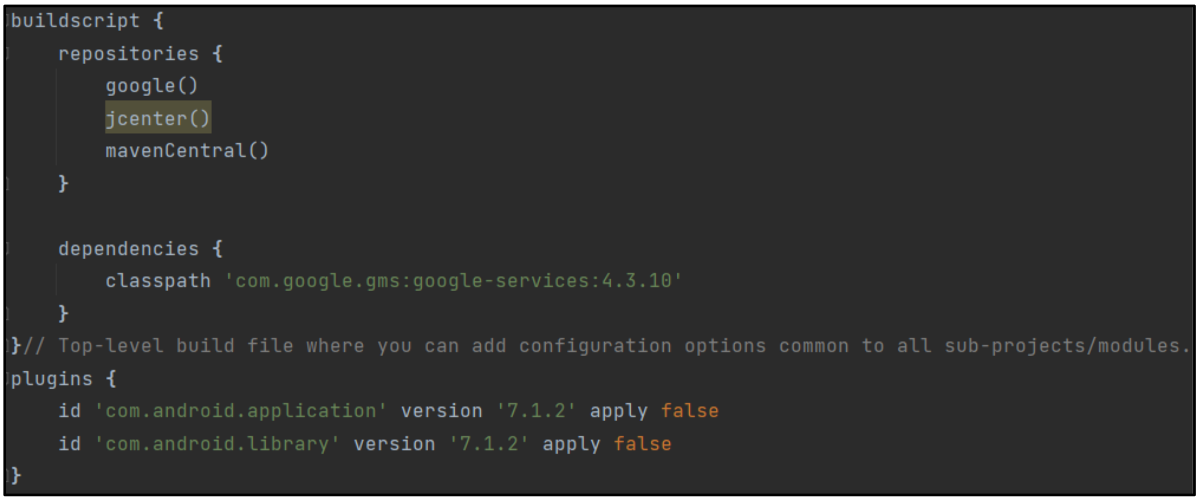
- Emulator which are full of features and efficient
- Gradle-based build system that very flexible
- Allow developer to develop application which suits for all Android devices on the market using a unified environment

5.3 Settings and Configuration

5.3.1 Android Studio Configuration

Build Gradle Project file configuration and setting

The dependency needed in this project is the google services with the version of 4.3.10. It is required to collaborate with the services sync with the Firebase platform.



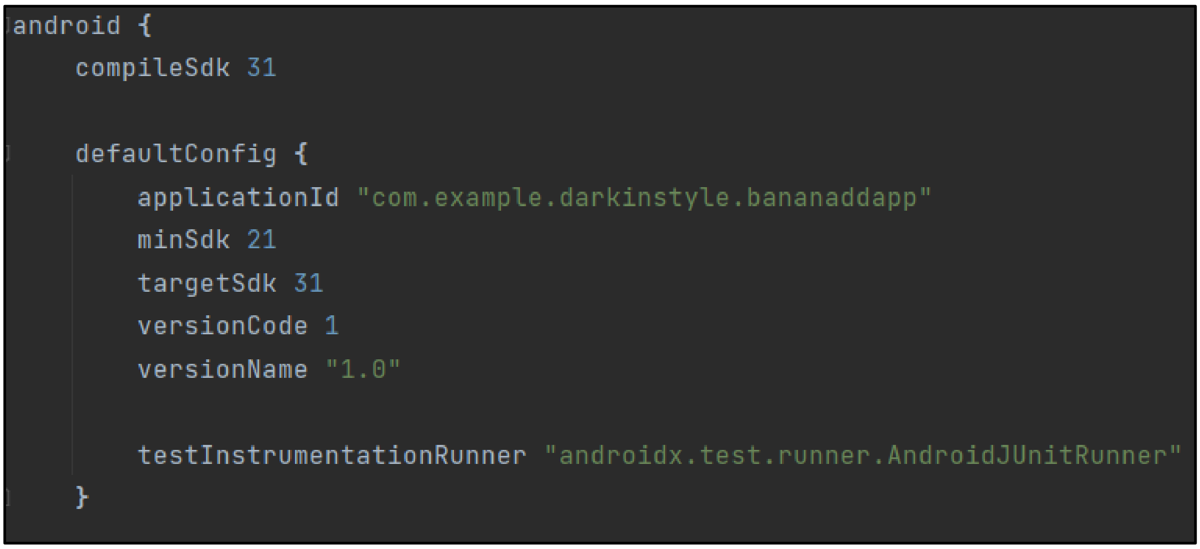
```
buildscript {
    repositories {
        google()
        jcenter()
        mavenCentral()
    }

    dependencies {
        classpath 'com.google.gms:google-services:4.3.10'
    }
}// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
    id 'com.android.application' version '7.1.2' apply false
    id 'com.android.library' version '7.1.2' apply false
}
```

Figure 5.3.1.1 Configuration and Setting on project build Gradle file

Build Gradle App Module file configuration and setting

SDK Version

The developed mobile application can run in minimum operating system requirement of SDK version of 21 (Android 5.0) to maximum 31 SDK version (Android 12).

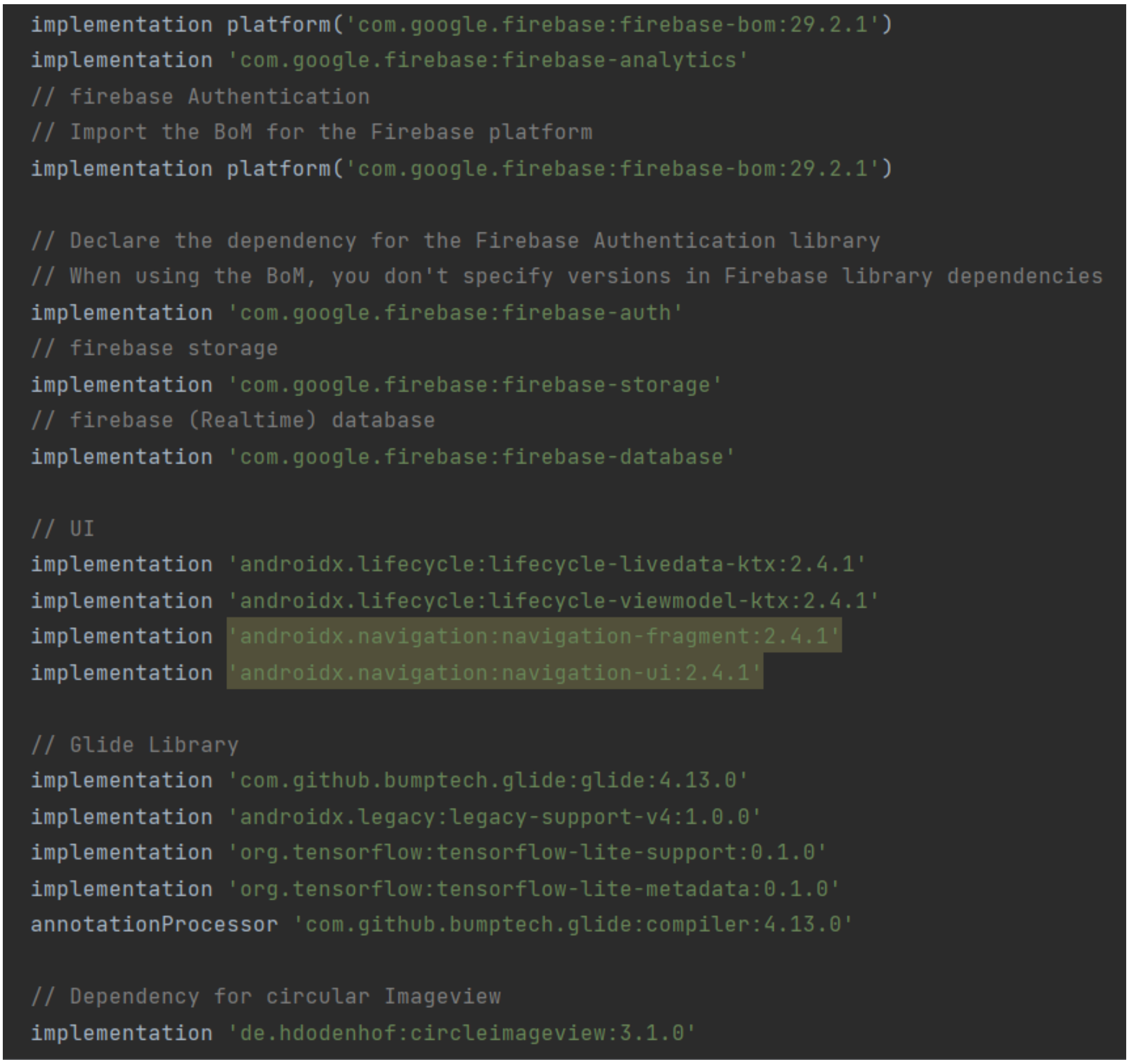


```
android {
    compileSdk 31

    defaultConfig {
        applicationId "com.example.darkinstyle.bananaddapp"
        minSdk 21
        targetSdk 31
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
```

Figure 5.3.1.2 SDK Version of Developed Mobile Application

Dependencies

Based on Figure 5.3.1.3, these are the required dependencies need to be used for the developed mobile application.

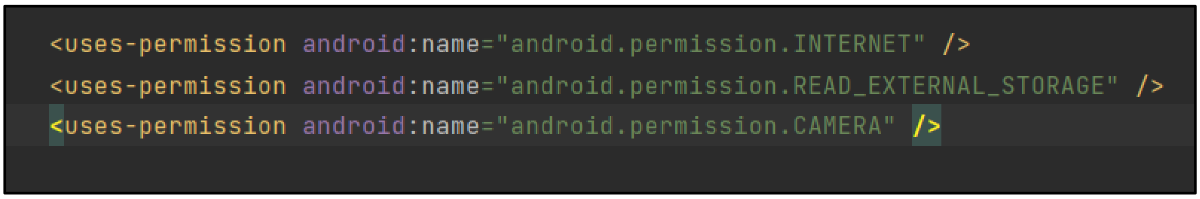


```
implementation platform('com.google.firebase:firebase-bom:29.2.1')
implementation 'com.google.firebase:firebase-analytics'
// firebase Authentication
// Import the BoM for the Firebase platform
implementation platform('com.google.firebase:firebase-bom:29.2.1')

// Declare the dependency for the Firebase Authentication library
// When using the BoM, you don't specify versions in Firebase library dependencies
implementation 'com.google.firebase:firebase-auth'
// firebase storage
implementation 'com.google.firebase:firebase-storage'
// firebase (Realtime) database
implementation 'com.google.firebase:firebase-database'

// UI
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.4.1'
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.4.1'
implementation 'androidx.navigation:navigation-fragment:2.4.1'
implementation 'androidx.navigation:navigation-ui:2.4.1'

// Glide Library
implementation 'com.github.bumptech.glide:glide:4.13.0'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
implementation 'org.tensorflow:tensorflow-lite-support:0.1.0'
implementation 'org.tensorflow:tensorflow-lite-metadata:0.1.0'
annotationProcessor 'com.github.bumptech.glide:compiler:4.13.0'

// Dependency for circular Imageview
implementation 'de.hdodenhof:circleimageview:3.1.0'
```

Figure 5.3.1.3 Dependencies Requirement for Developed Mobile Application

5.3.2 Android Application Permission Access Used

The Android mobile application developed in this project will be requiring the permission access for any device to request internet support, device storage access and camera access as well. Internet is required as the app will interacting with Firebase platform for all services such as authentication, database, and storage. Camera is required to capture picture for perform banana disease detection feature. Read access to device for upload image from device for profile picture, adding post and banana disease detection.



```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CAMERA" />
```

Figure 5.3.2 Android Application Permission Access Used

5.4 System Operation

In this section, every implemented interface and operation in the developed mobile application will be elaborated and simulated.

5.4.1 Account Registration Operation

Successful Account Registration

Based on Figure 5.4.1.1 below, this figure depicts the successful account registration under normal circumstances. When the mobile application is not login, the first landing interface will be the login interface. In the login interface, if users would like to register an account, they are required to click on "register" text button shown below figure of a red bounding box. Once users clicked on it, users will be redirected to register interface and users is prompted to insert their name, email address, password, confirm password and profile picture. Click on sample photo icon under blue bounding box shown in figure below, will request users to insert picture from their device gallery and user must allow permission accessing device storage if prompted. Once all input field inserted, users can be registered, and account is created.

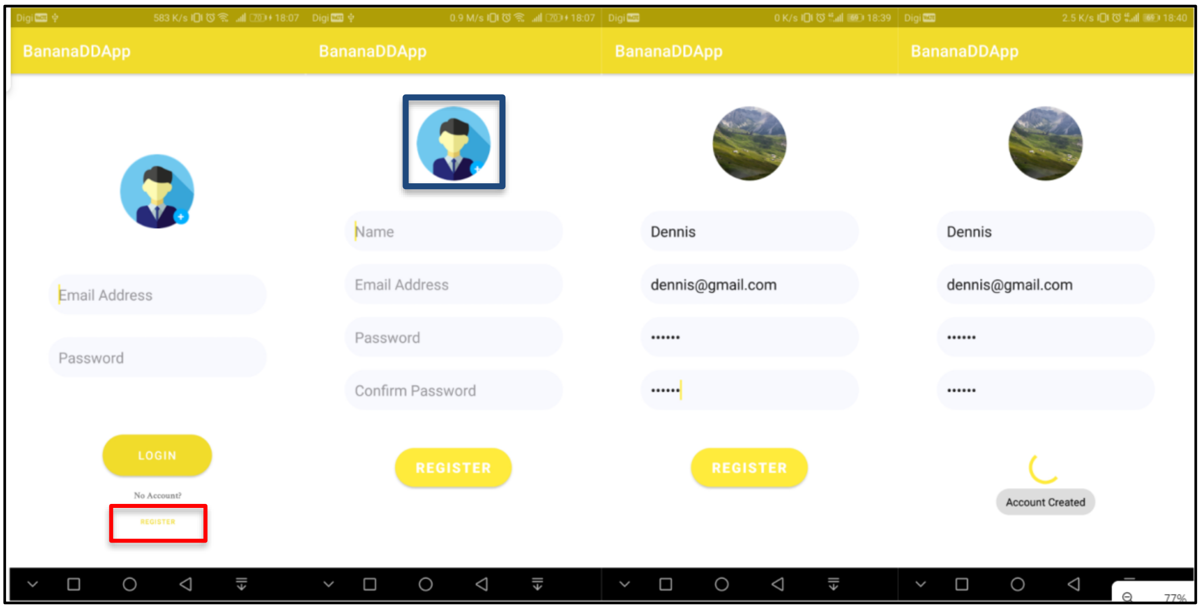
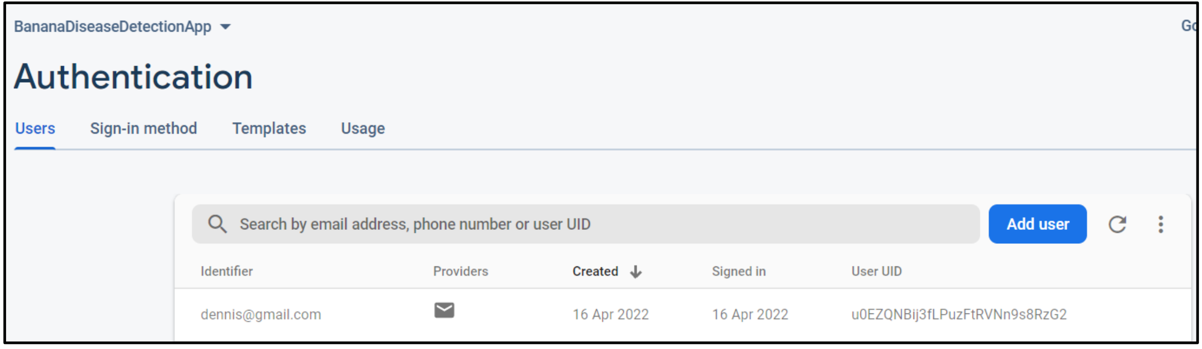


Figure 5.4.1.1 Success Account Registration Operation Flow

Below Figure 5.4.1.2 depicts the account created status and shown in Firebase Authentication in Firebase Project management console.


BananaDiseaseDetectionApp
Authentication
Users
Sign-in method
Templates
Usage
Search by email address, phone number or user UID
Add user

Identifier	Providers	Created	Signed in	User UID
dennis@gmail.com	✉	16 Apr 2022	16 Apr 2022	u0EZQNBij3fLPuzFtRVNn9s8RzG2

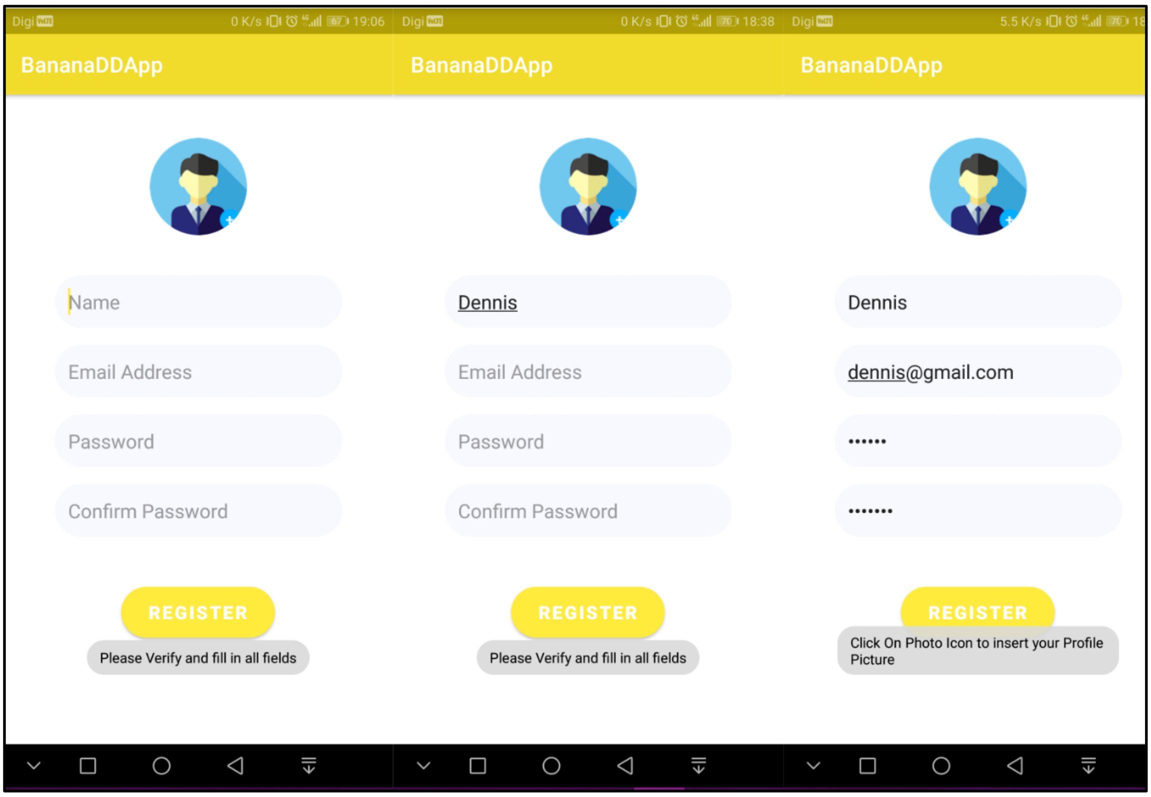
Figure 5.4.1.2 Account Created shown in Firebase Authentication

Unsuccessful Account Registration Circumstances

Users cannot register if:

1. Leave blank for every input field
2. Not allow device storage access permission
3. Incorrect email address format
4. Non-equivalent between password and confirm password entered

Below Figure 5.4.1.3 depicts some of the unsuccessful condition.



Dennis

Please Verify and fill in all fields

Dennis

dennis@gmail.com

••••••

•••••••

Click On Photo Icon to insert your Profile Picture

Figure 5.4.1.3 Unsuccessful Account Registration Operation Flow

5.4.2 Account Login Operation

Successful Account Login

Figure 5.4.2.1 depicts the successful account login operation flow on how users able to login into the mobile application system. The first landing interface of the mobile application is login interface. Users prompted to insert the login credentials in the "EditText" input field for both email address and password. Once user filled up the input field, users can click on the "Login" button to proceed into the mobile application home interface. Once users successfully logged in, system will show a Toast message "Login Success" that verified users have officially logged in. The logged in account is validated and authenticated by the Firebase Authentication service implemented in the mobile application.

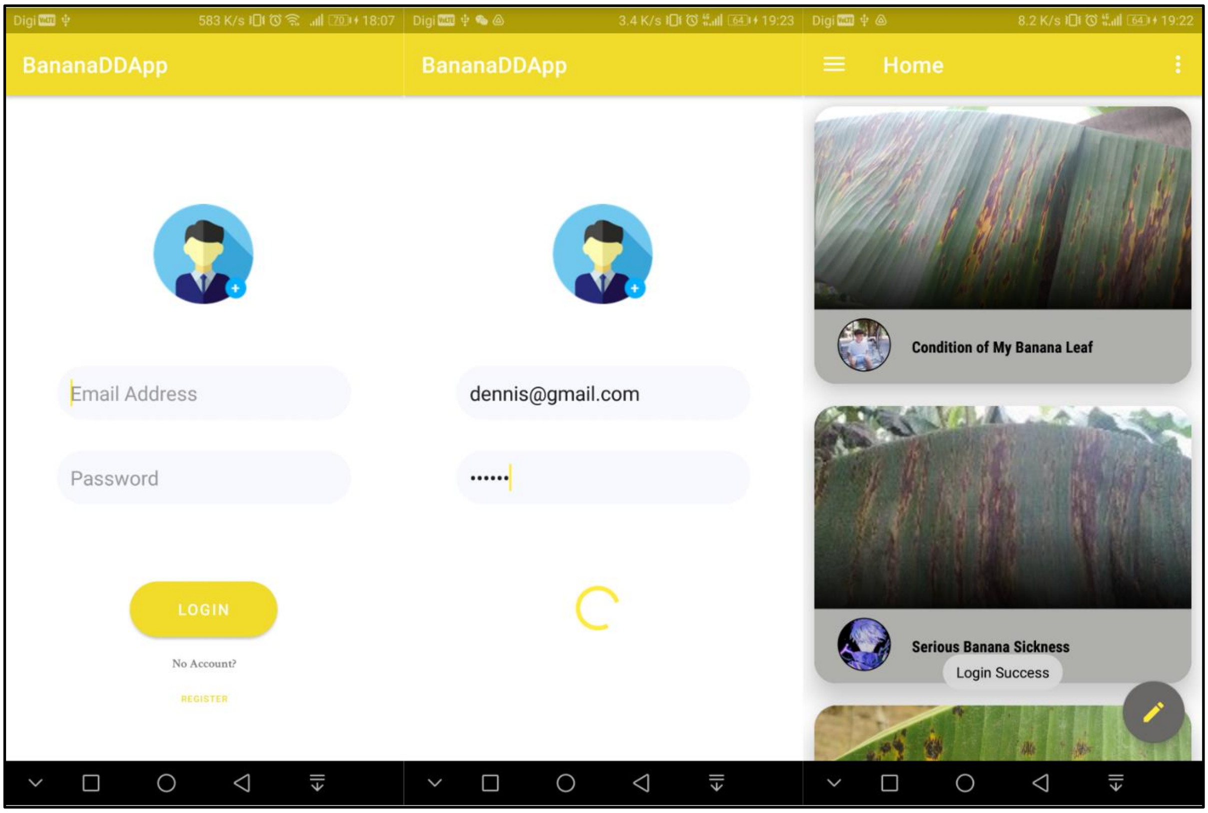


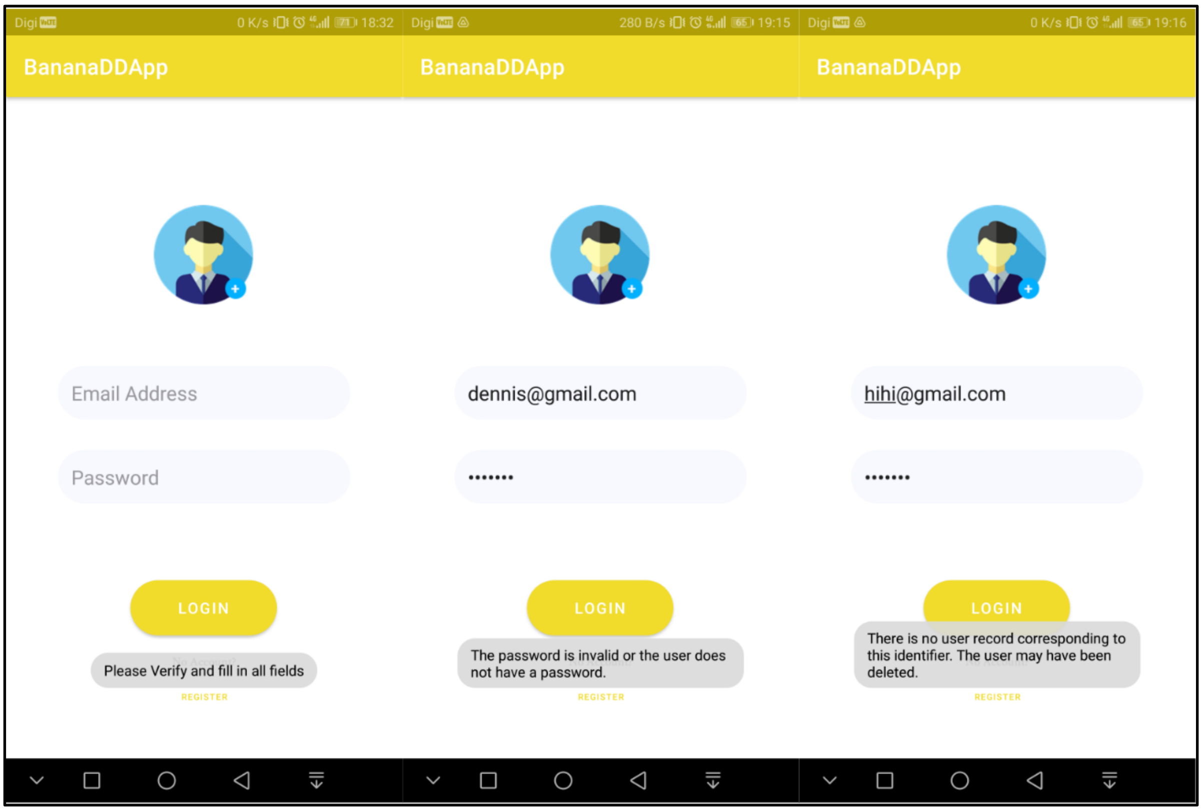
Figure 5.4.2.1 Successful Login Operation Flow

Unsuccessful Account Login Circumstances

Figure 5.4.2.2 depicts the unsuccessful account login circumstances that performed by users during attempted to login.

Circumstances:

1. Leaving blank for both input field
2. Invalid password entered for email address
3. Non-existing login credentials



Please Verify and fill in all fields

dennis@gmail.com

•••••••

The password is invalid or the user does not have a password.

hihi@gmail.com

•••••••

There is no user record corresponding to this identifier. The user may have been deleted.

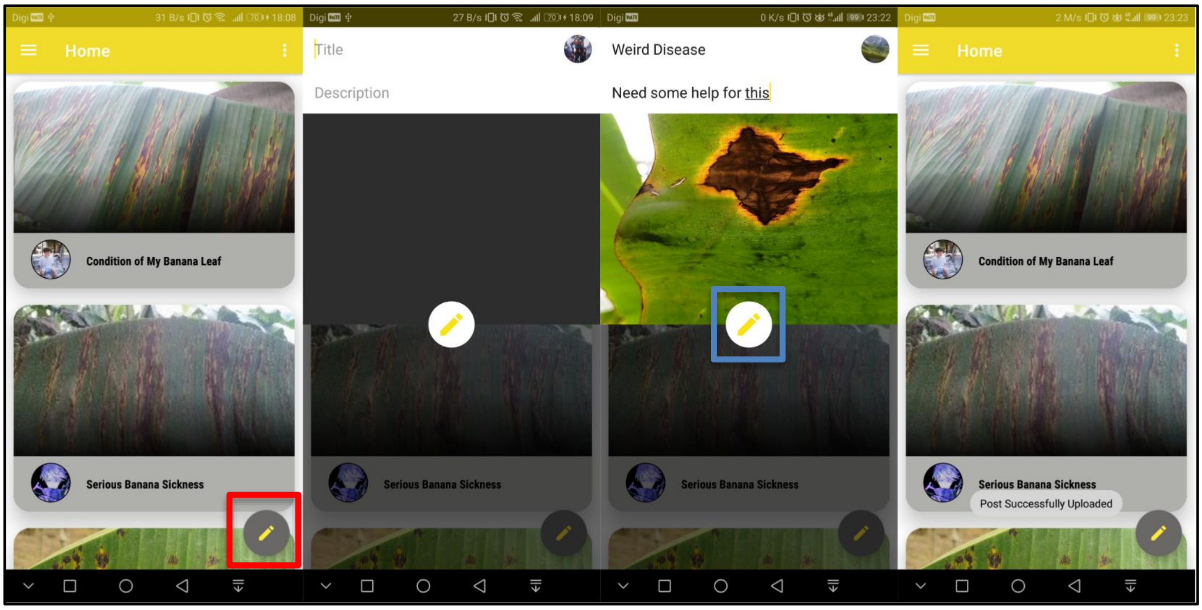
Figure 5.4.2.2 Unsuccessful Account Login Circumstances Operation Flow

5.4.3 Home Interface Operation

Add Post Operation

Successful Add Post Operation

Below Figure 5.4.3.1 depicts the successful add post operation flow on how users able to add their post successfully. Once users authenticated and logged in, users automatically redirected into the home interface. If users would like to add a post, they can click on the floating action button with a pen logo which have a red bounding box shown in figure below. Once the click registered, a dialog window will popup and prompt users to input both “EditText” input field and “ImageView” for image insertion. A dialog message regarding device storage permission access may popup, users must allow the requested access. Once users inserted all required input field for the post details, users required to click on the pen icon button with blue bounding box shown in figure below. Once the button clicked, the add post dialog window will shut off and display “Post Successfully Uploaded” in Toast message. The system will store the post details information into the Firebase Realtime Database and post image will store into the Firebase Storage accordingly.



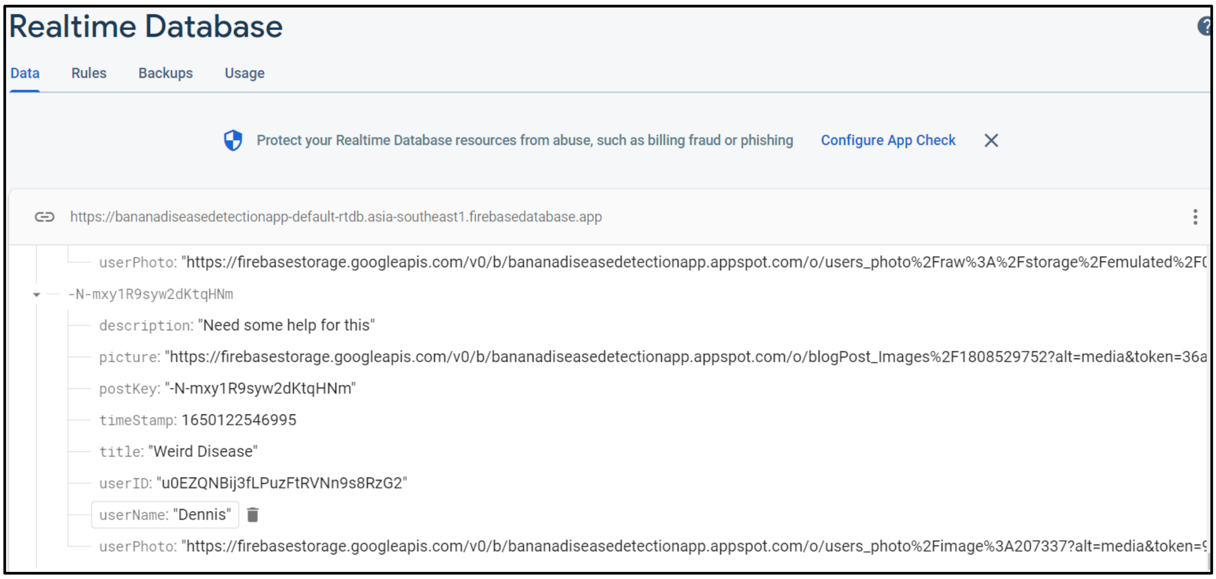
Weird Disease

Need some help for this

Post Successfully Uploaded

Figure 5.4.3.1 Successful Add Post Operation Flow

Figure 5.4.3.2 below shown the successful add post details referred to Figure 5.4.3.1. Post image will be stored in a form of URL link with access token shown in figure below. The URL link is referring to the post image that stored in a folder at Firebase Storage which shown in Figure 5.4.3.3.



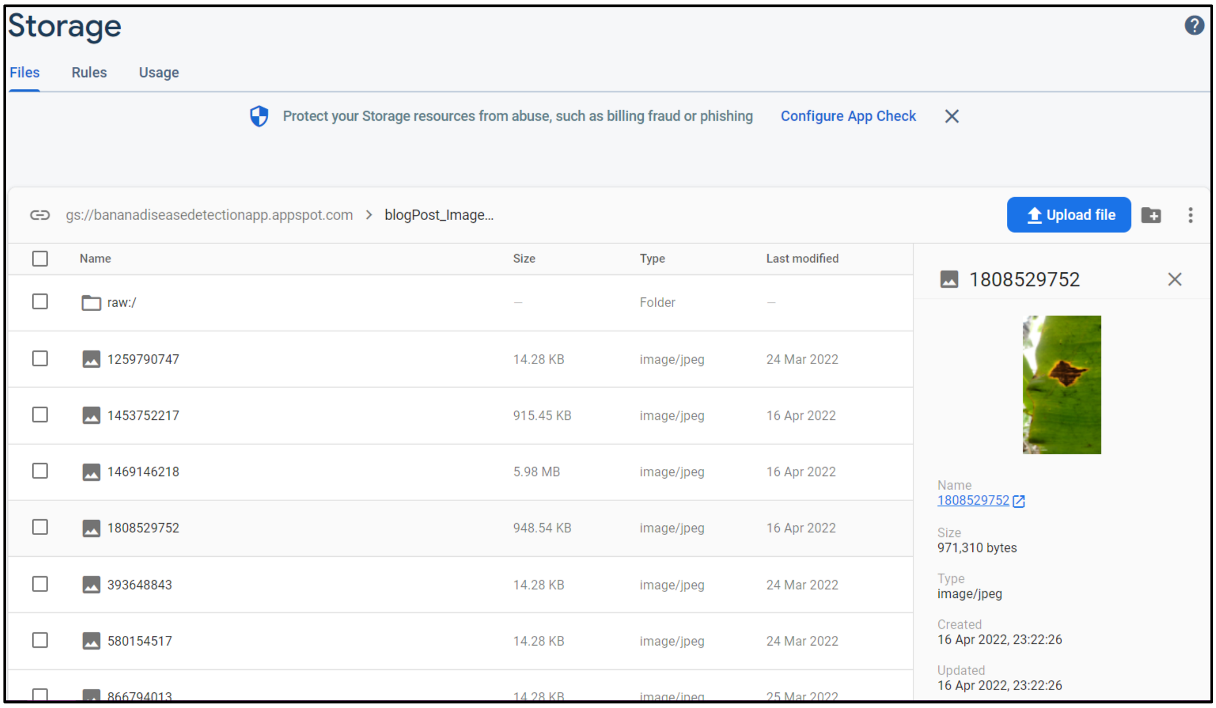
Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check

https://bananadiseasedetectionapp-default-rtdb.asia-southeast1.firebasedatabase.app

```
      userPhoto: "https://firebasestorage.googleapis.com/v0/b/bananadiseasedetectionapp.appspot.com/o/users_photo%2Fraw%3A%2Fstorage%2Femulated%2F(
  -N-mxy1R9syw2dKtqHNm
      description: "Need some help for this"
      picture: "https://firebasestorage.googleapis.com/v0/b/bananadiseasedetectionapp.appspot.com/o/blogPost_Images%2F1808529752?alt=media&token=36a
      postKey: "-N-mxy1R9syw2dKtqHNm"
      timeStamp: 1650122546995
      title: "Weird Disease"
      userID: "u0EZQNBij3fLPuzFtRVNn9s8RzG2"
      userName: "Dennis"
      userPhoto: "https://firebasestorage.googleapis.com/v0/b/bananadiseasedetectionapp.appspot.com/o/users_photo%2Fimage%3A207337?alt=media&token=(
```

Figure 5.4.3.2 Added Post in Firebase Realtime Database



Storage

Protect your Storage resources from abuse, such as billing fraud or phishing Configure App Check

gs://bananadiseasedetectionapp.appspot.com > blogPost_Image...

☐	Name	Size	Type	Last modified
☐	raw:/	–	Folder	–
☐	1259790747	14.28 KB	image/jpeg	24 Mar 2022
☐	1453752217	915.45 KB	image/jpeg	16 Apr 2022
☐	1469146218	5.98 MB	image/jpeg	16 Apr 2022
☐	1808529752	948.54 KB	image/jpeg	16 Apr 2022
☐	393648843	14.28 KB	image/jpeg	24 Mar 2022
☐	580154517	14.28 KB	image/jpeg	24 Mar 2022
☐	866794013	14.28 KB	image/jpeg	25 Mar 2022

1808529752

Name
1808529752

Size
971,310 bytes

Type
image/jpeg

Created
16 Apr 2022, 23:22:26

Updated
16 Apr 2022, 23:22:26

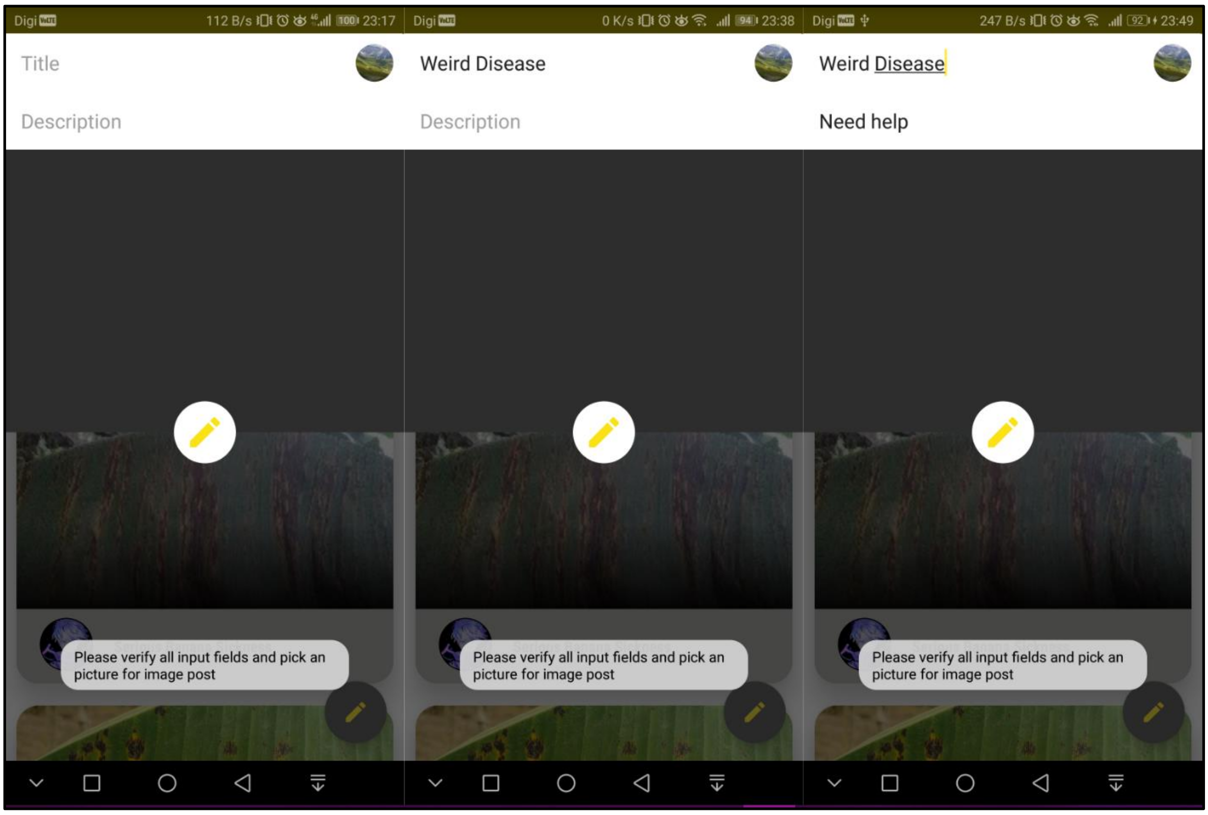
Figure 5.4.3.3 Added Post Image in Firebase Storage

Unsuccessful Add Post Circumstances

Figure 5.4.3.2 depicts the unsuccessful add post circumstances operation flow performed by users when they attempted to add post.

Circumstances:

1. Leave blank for every input field
2. Leave description input field and image input field blank
3. Only image input field blank



Please verify all input fields and pick an picture for image post

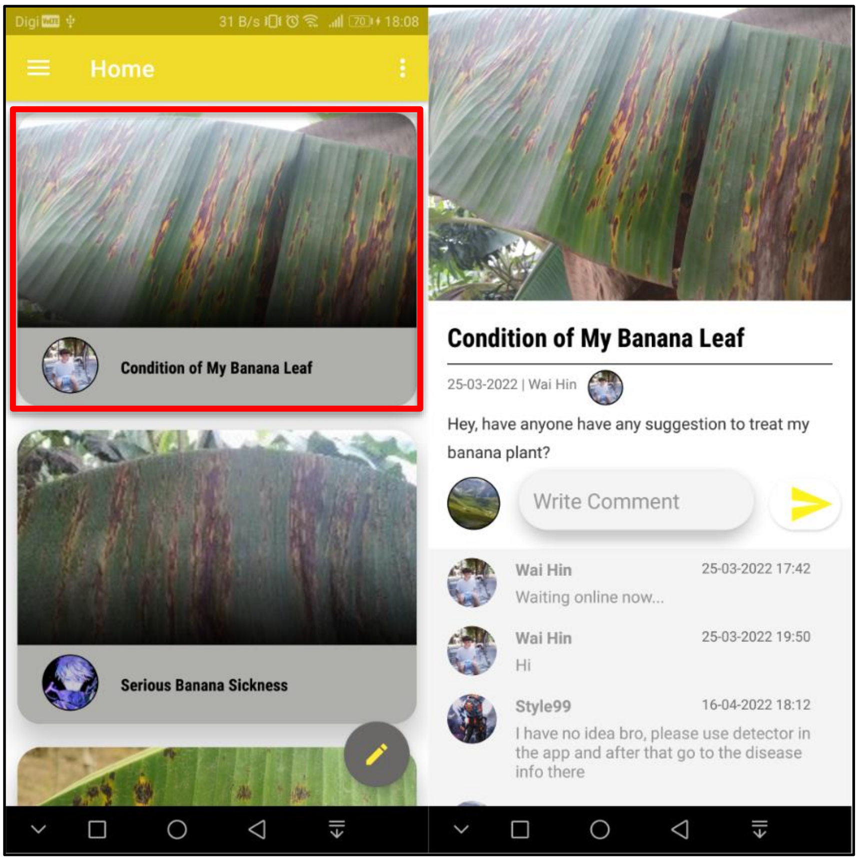
Please verify all input fields and pick an picture for image post

Please verify all input fields and pick an picture for image post

Figure 5.4.3.3 Unsuccessful Add Post Circumstances Operation Flow

View Post Details Operation

Figure 5.4.3.4 below depicts the operation flow of view post details on how users should perform when they would like to view a particular post detail. At home interface, users can click on the post that bounded by red box shown in figure below. The system will redirect users to new activity of selected post details to view the post details such as post title, image, description, user and username that post it, postdate and comment section as well.



Condition of My Banana Leaf

25-03-2022 | Wai Hin

Hey, have anyone have any suggestion to treat my banana plant?

Wai Hin 25-03-2022 17:42
Waiting online now...

Wai Hin 25-03-2022 19:50
Hi

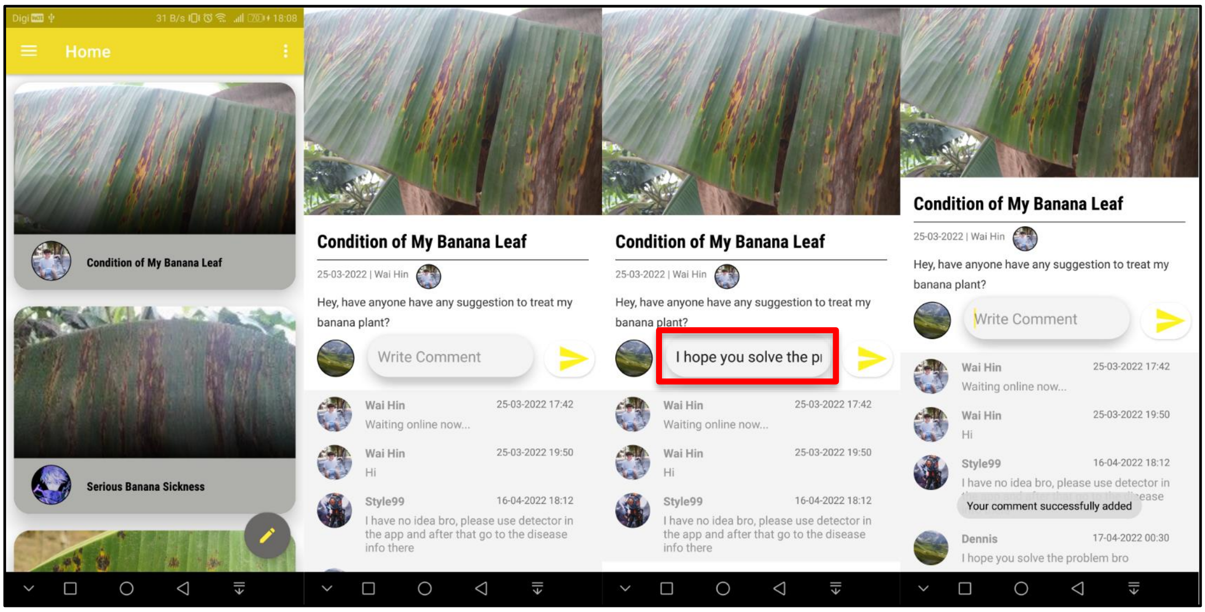
Style99 16-04-2022 18:12
I have no idea bro, please use detector in the app and after that go to the disease info there

Figure 5.4.3.4 View Post Details Operation Flow

Add Comment Operation

Successful Add Comment Operation

Figure 5.4.3.5 below depicts the successful add comment operation flow performed by the users. The initial operation flow will be the same as shown in Figure 5.4.3.4 when users wanted to view the post details. The users can write their comment in the “Edittext” input field with red bounding box shown in figure below. Once users satisfied with their comment content, they can add the comment by click on the send button which located beside the comment input field. Users’ comment, username, timestamp, user profile picture and UID (use for Firebase Authentication) will be stored in the Firebase Realtime Database in real-time shown in Figure 5.4.3.6. Finally, the system will display “Your comment successfully added” in a Toast message.



Condition of My Banana Leaf

25-03-2022 | Wai Hin

Hey, have anyone have any suggestion to treat my banana plant?

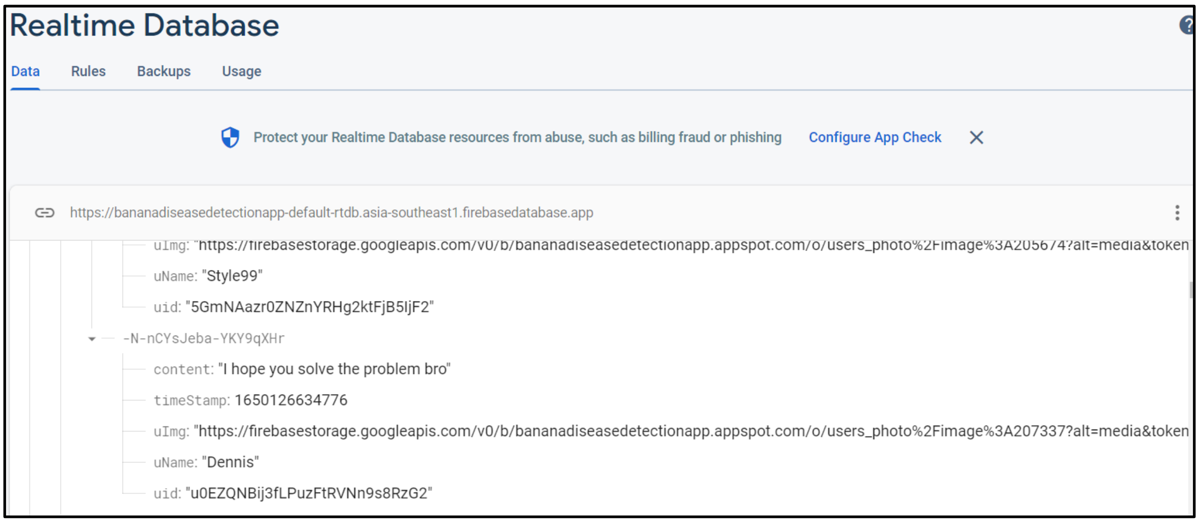
Wai Hin 25-03-2022 17:42
Waiting online now...

Wai Hin 25-03-2022 19:50
Hi

Style99 16-04-2022 18:12
I have no idea bro, please use detector in the app and after that go to the disease info there

Dennis 17-04-2022 00:30
I hope you solve the problem bro

Figure 5.4.3.5 Successful Add Comment Operation Flow



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://bananadiseasedetectionapp-default-rtdb.asia-southeast1.firebasedatabase.app

```
        uImg: "https://firebasestorage.googleapis.com/v0/b/bananadiseasedetectionapp.appspot.com/o/users_photo%2Fimage%3A205674?alt=media&token
        uName: "Style99"
        uid: "5GmNAazr0ZNZnYRHg2ktFjB5IjF2"
    -N-nCYsJeba-YKY9qXHr
        content: "I hope you solve the problem bro"
        timeStamp: 1650126634776
        uImg: "https://firebasestorage.googleapis.com/v0/b/bananadiseasedetectionapp.appspot.com/o/users_photo%2Fimage%3A207337?alt=media&token
        uName: "Dennis"
        uid: "u0EZQNBij3fLPuzFtRVNn9s8RzG2"
```

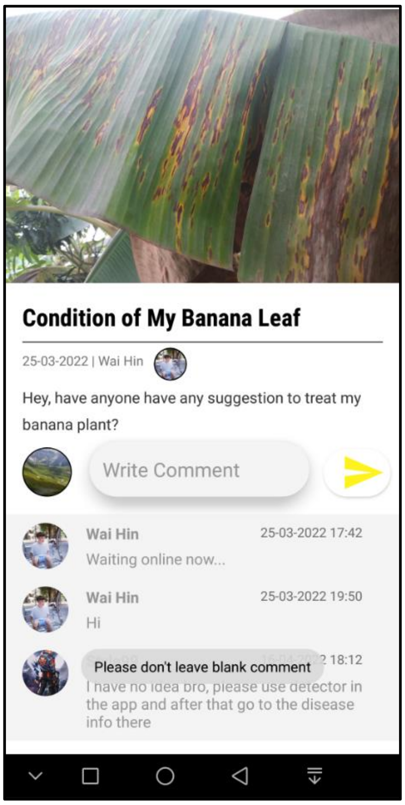
Figure 5.4.3.6 Comment Added into Firebase Realtime Database

Unsuccessful Add Comment Circumstance Operation

Figure 5.4.3.7 depicts the unsuccessful add comment operation performed by users. This circumstance happened when users attempt to:

1. Add a blank comment

The system will display message to inform users to not add a blank comment.



Condition of My Banana Leaf

25-03-2022 | Wai Hin

Hey, have anyone have any suggestion to treat my banana plant?

Wai Hin 25-03-2022 17:42
Waiting online now...

Wai Hin 25-03-2022 19:50
Hi

2 18:12
I have no idea bro, please use detector in the app and after that go to the disease info there

Please don't leave blank comment

Figure 5.4.3.7 Unsuccessful Add Comment Circumstance Operation

Navigation Drawer Operation

Figure 5.4.3.8 below depicts how users can perform navigation drawer operation. Users can click on the “kebab” menu icon which bounded by red box and the navigation drawer will slides out. Navigation drawer displayed the users’ profile picture, username and email address. Other than that, users can click “Home” to go home interface, click on “Detector” to perform image classification on infected banana leaf, click on “Banana Crop Diseases” to view banana diseases information and click on “Sign Out” to sign out off from the system.

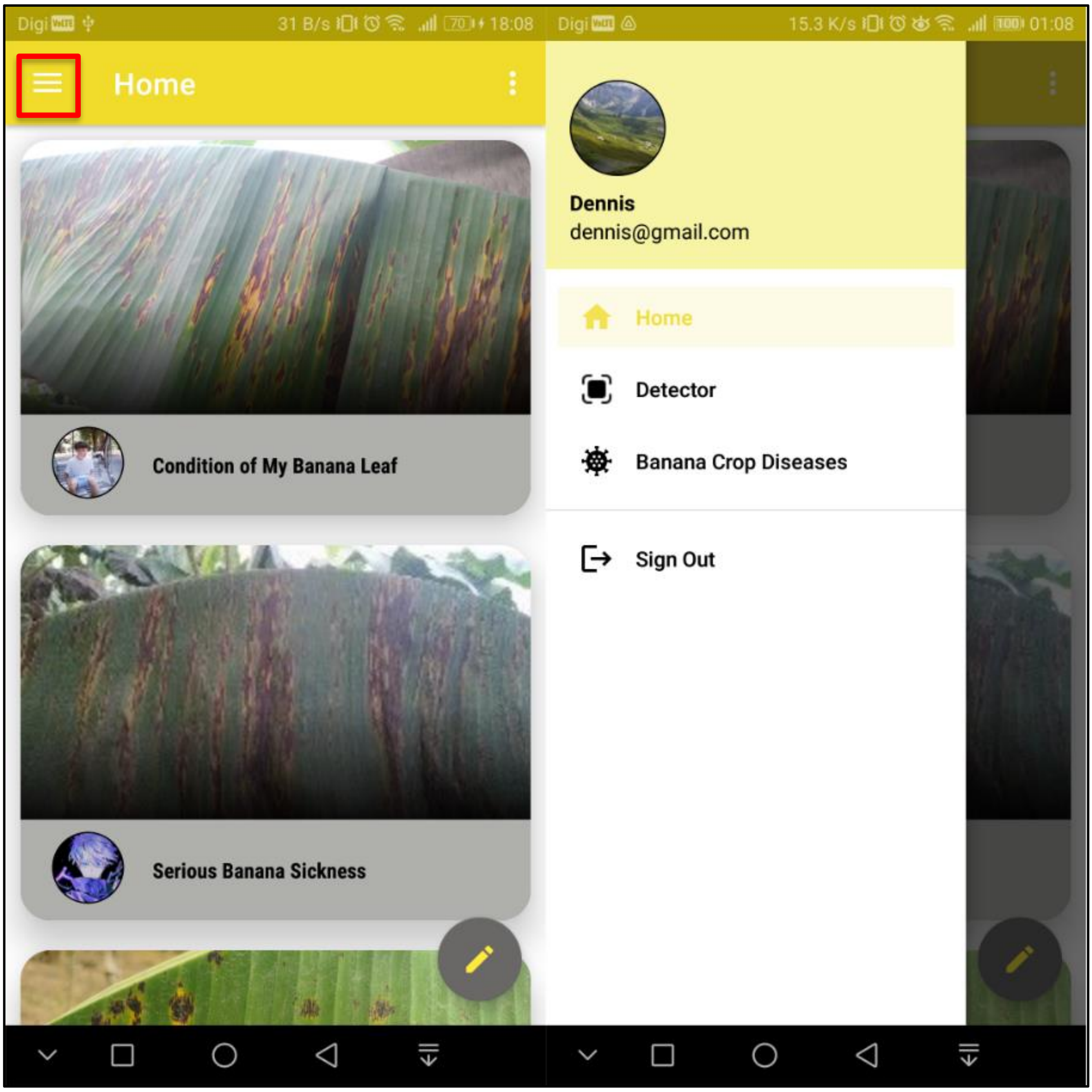


Figure 5.4.3.8 Navigation Drawer Operation

Sign Out Operation

Users can click “Sign Out” in navigation drawer and system process sign out using Firebase Authentication service . Users will be redirected to login interface afterwards. Finally, system will display “Sign Out Successfully” in a Toast message shown in Figure 5.4.3.9 below.

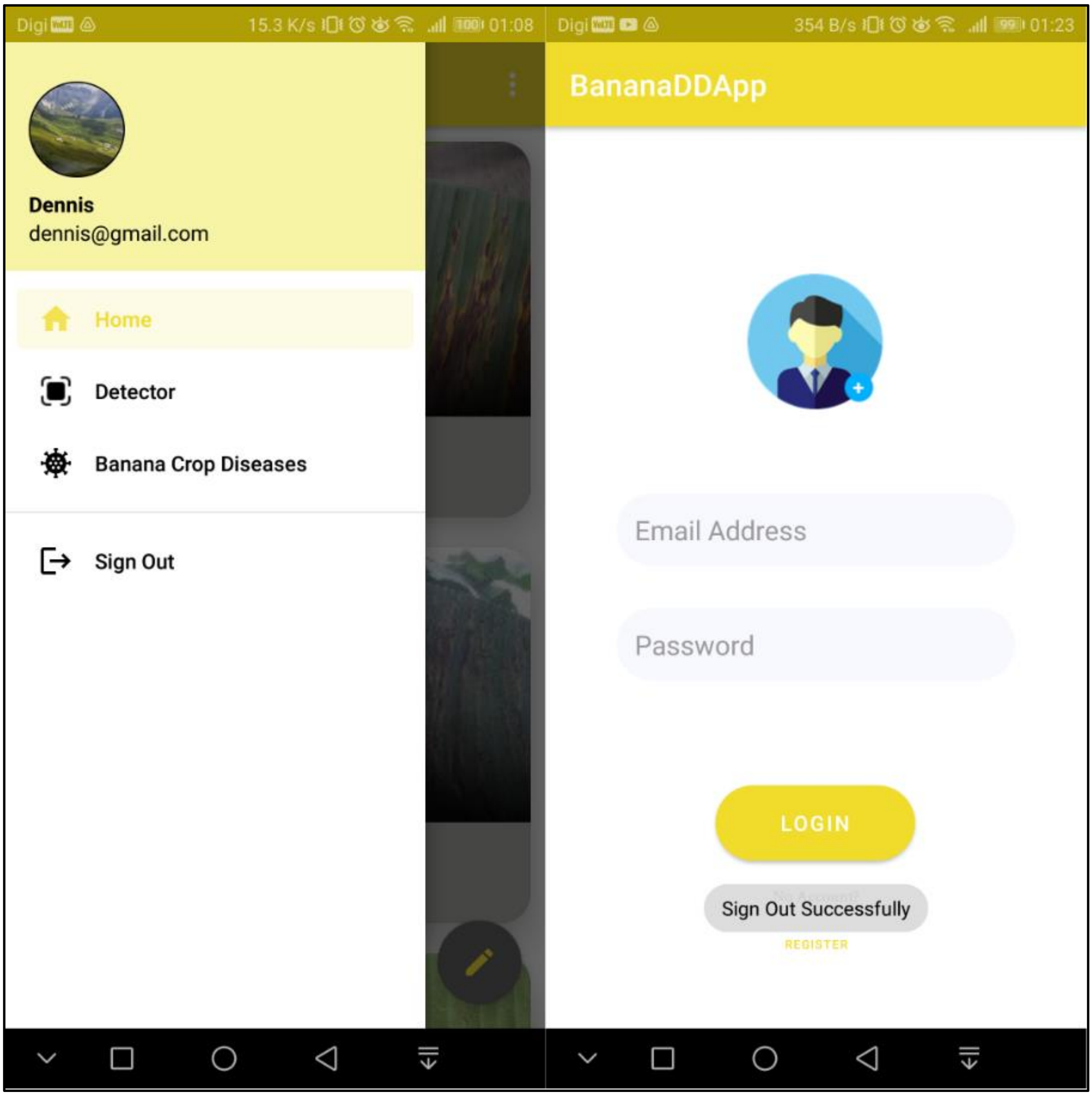


Figure 5.4.3.9 Sign Out Operation Flow

5.4.4 Detector Interface Operation

Navigate to Detector Interface

Figure 5.4.4.1 depicts on how users want to navigate to Detector interface through navigation drawer. Users can click on “Detector” in navigation drawer and will be redirected to the Detector interface straight away. The deployed TensorFlow Lite model inside the mobile application will process banana diseases image classification once receive input from image uploading from device storage or captured picture from device camera.

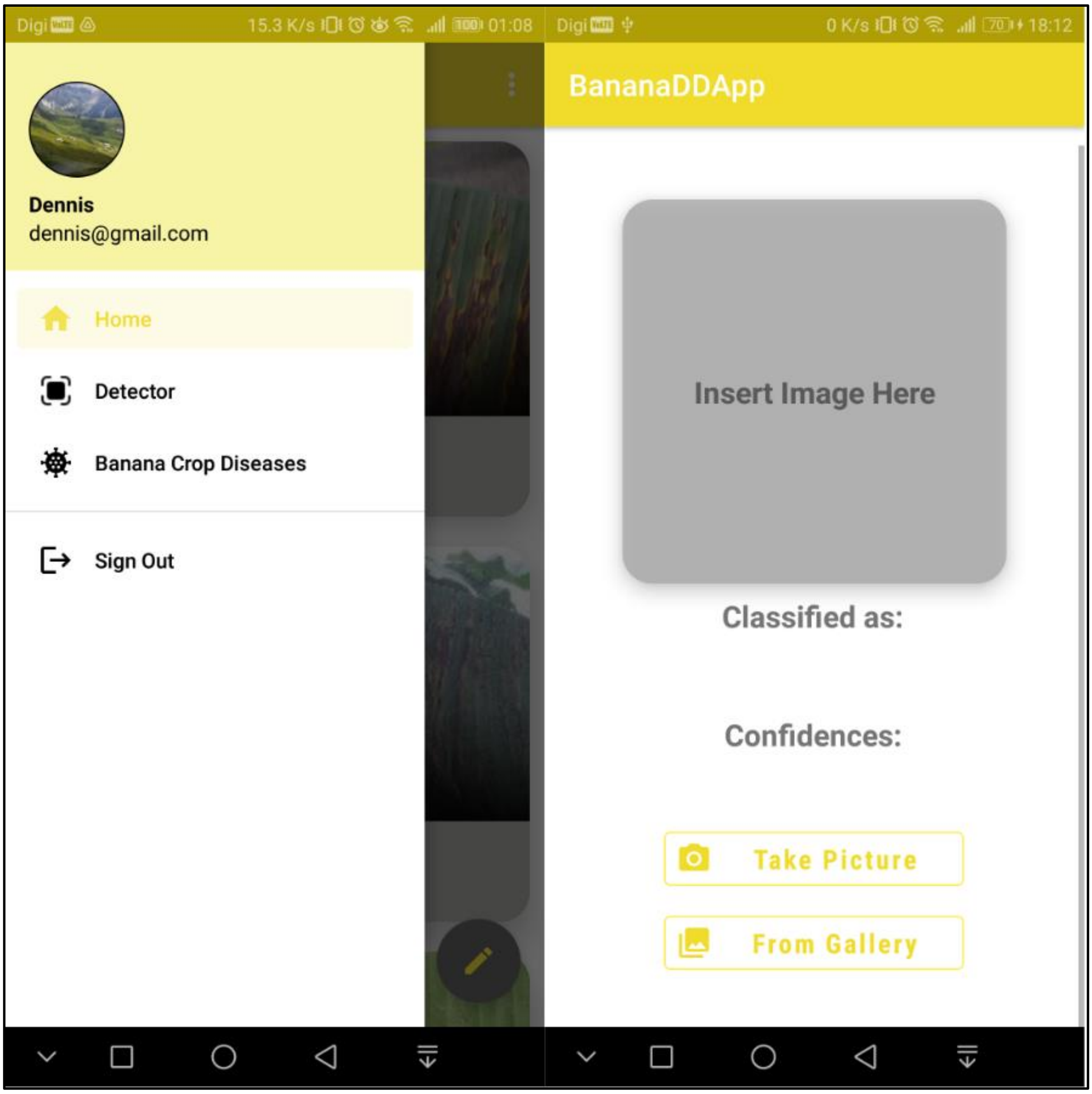
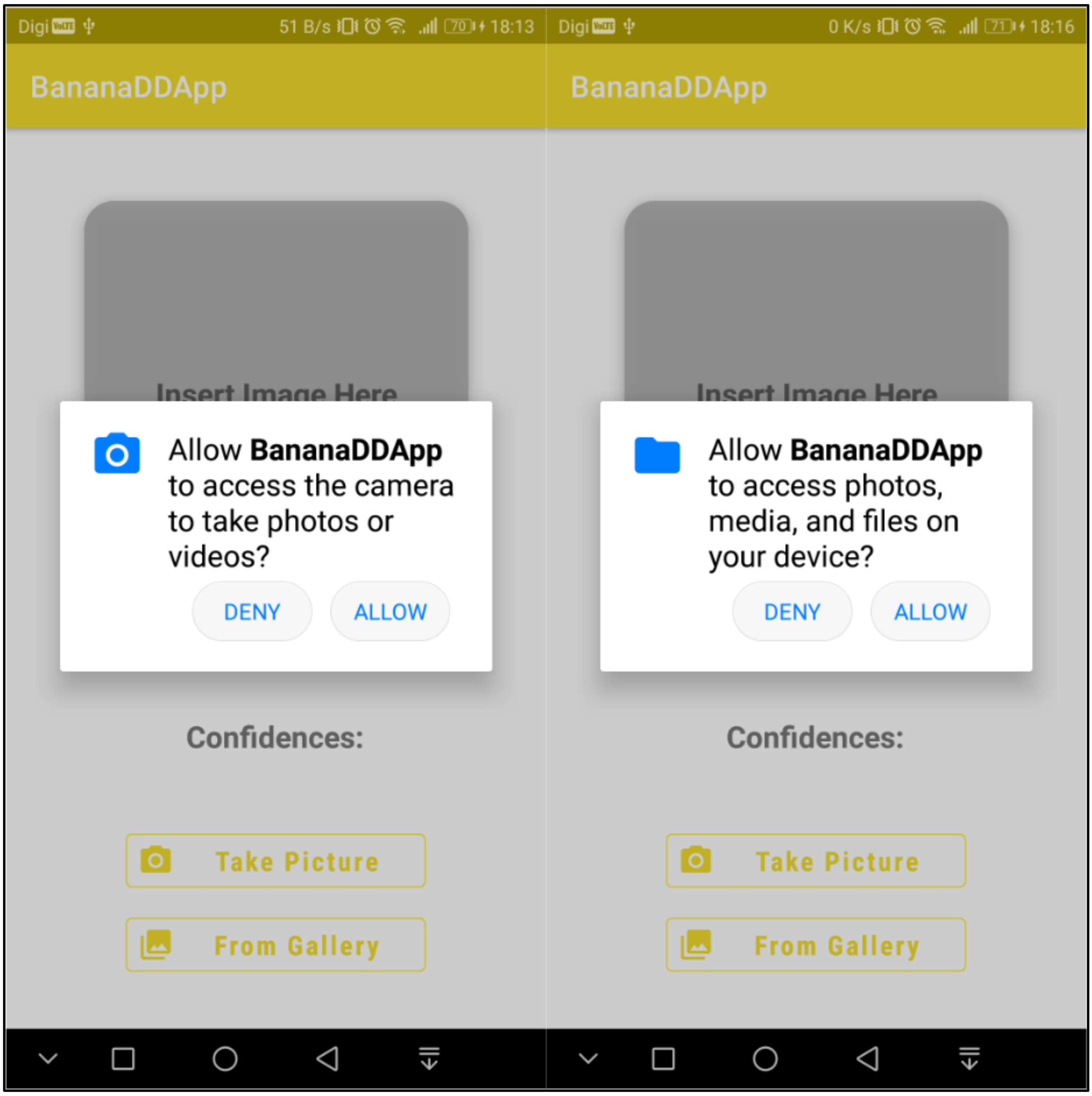


Figure 5.4.4.1 Navigate to Detector Interface

Device Resource Permission Access Request

If users click on either “Take Picture” or “From Gallery”, the app system will prompt users to allow the permission access request shown in Figure 5.4.4.2. Users must choose to “ALLOW” for app system to access the device storage and camera for retrieving image and capture picture. Hence, the system able to perform banana diseases detection on the retrieved image or captured picture.



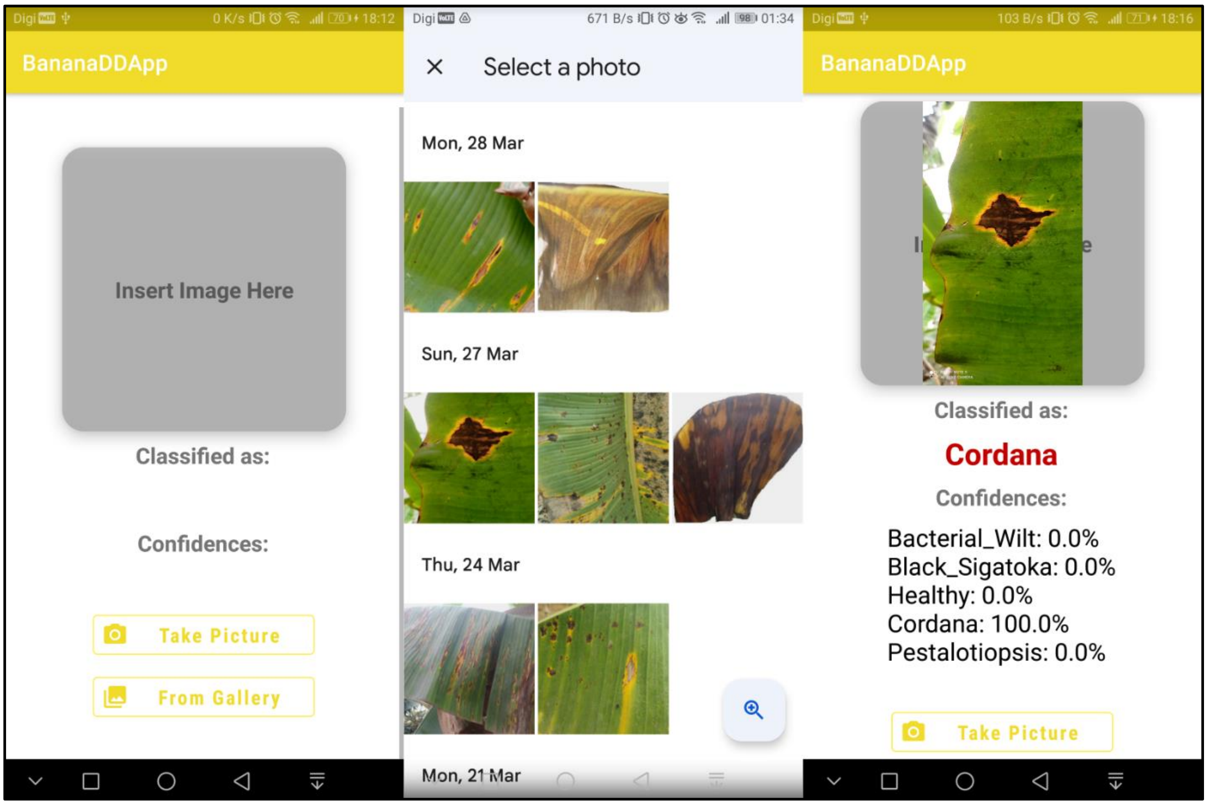
Allow **BananaDDApp** to access the camera to take photos or videos?

Allow **BananaDDApp** to access photos, media, and files on your device?

Figure 5.4.4.2 Device Resource Permission Access Request

Classify Image From Gallery Operation

Figure 5.4.4.3 depicts the users perform the operation of classify an infected banana leaf that retrieved from device storage. In Detector interface, users can click on “From Gallery” icon-text button for accessing device storage to retrieve an image. Users will be redirected to an interface for choosing infected banana leaf image. Once users finished and selected the image, users will be redirected back to Detector interface. The Detector interface will display the classification result such as selected infected banana leaf image, classified disease name, confidences percentage for each disease. In Figure 5.4.4.3, the classified disease name is Cordana, confidence percentage is 100% and zero for others.

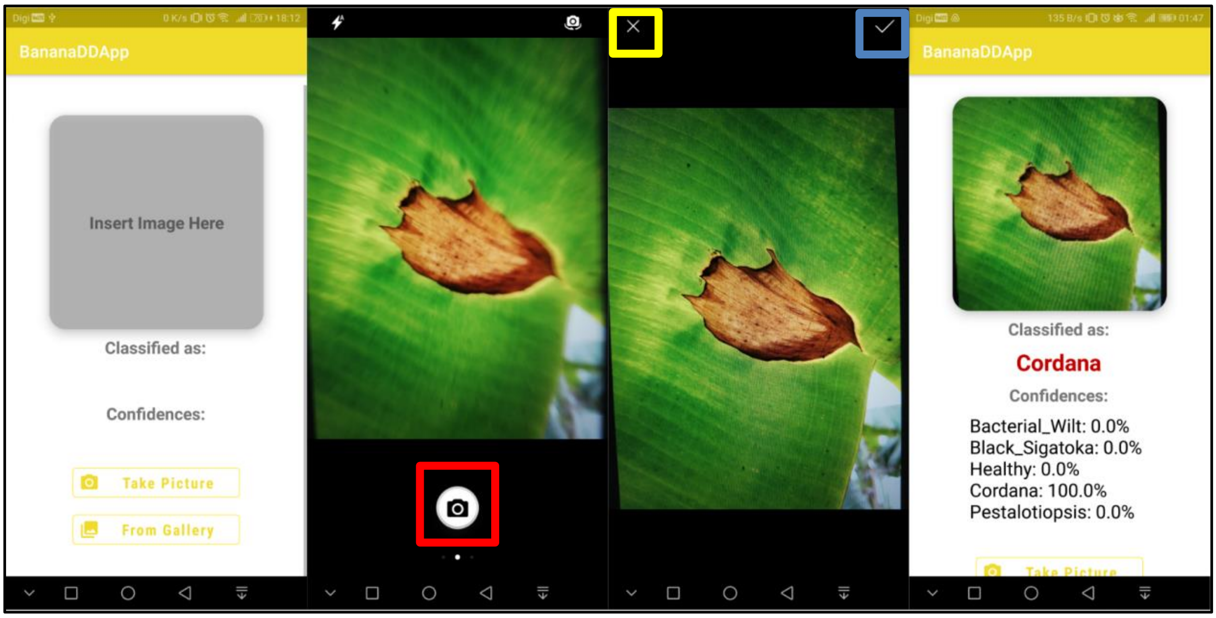


Bacterial_Wilt: 0.0%
Black_Sigatoka: 0.0%
Healthy: 0.0%
Cordana: 100.0%
Pestalotiopsis: 0.0%

Figure 5.4.4.3 Retrieve Image from Device Storage to Classify Operation Flow

Classify Captured Picture From Camera Operation

Figure 5.4.4.4 depicts users perform the operation of classify an infected banana leaf picture captured from the device camera. In Detector interface, users can click on “Take Picture” icon-text button for accessing the camera to capture a picture. Users will be redirected to device camera interface which running in real-time. Users are suggested to fit the target infected banana leaf area inside the camera frame view. After that, users can click on the “camera” icon button to capture the desired picture. Users could click on “tick” icon button with blue bounding box if users satisfied with the captured picture, otherwise users can retry by click on “cancel” icon button with yellow bounding box. Once successfully captured a picture, users will be redirected back to Detector interface which display the classification result which is the same as in Figure 5.4.4.3.



Classified as:

Confidences:

Classified as:

Cordana

Confidences:

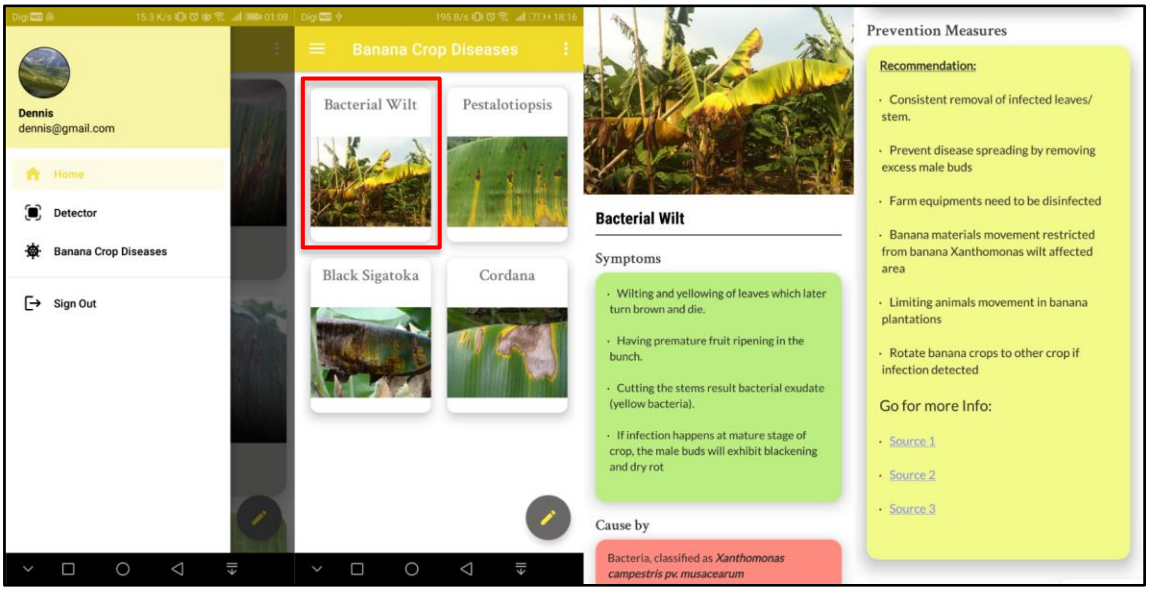
Bacterial_Wilt: 0.0%
Black_Sigatoka: 0.0%
Healthy: 0.0%
Cordana: 100.0%
Pestalotiopsis: 0.0%

Figure 5.4.4.4 Capture Picture from Device Camera to Classify Operation Flow

5.4.5 Banana Crop Diseases Interface Operation

Figure 5.4.5.1 below depicts users perform operation of viewing the banana crop diseases information. Users need to click on “Banana Crop Diseases” in navigation drawer to navigate to Banana Crop Diseases interface. At Banana Crop Diseases interface, users can click either one of the banana disease card view items to view its information. If users click on the “Bacterial Wilt” with red bounding box, users will be redirected to “Bacterial Wilt” disease Activity which display corresponding disease image, name, symptoms, causal and prevention measures.

Figure 5.4.5.2 depicts users click on the “Source 1” with blue bounding box for reaching more information. Users will be requested to use either one browser to access the URL link. Users will be redirected to the selected browser. The selected browser will browse the website and display website content to the users.



Bacterial Wilt

Symptoms

- Wilting and yellowing of leaves which later turn brown and die.
- Having premature fruit ripening in the bunch.
- Cutting the stems result bacterial exudate (yellow bacteria).
- If infection happens at mature stage of crop, the male buds will exhibit blackening and dry rot

Cause by

Bacteria, classified as *Xanthomonas campestris pv. musacearum*

Prevention Measures

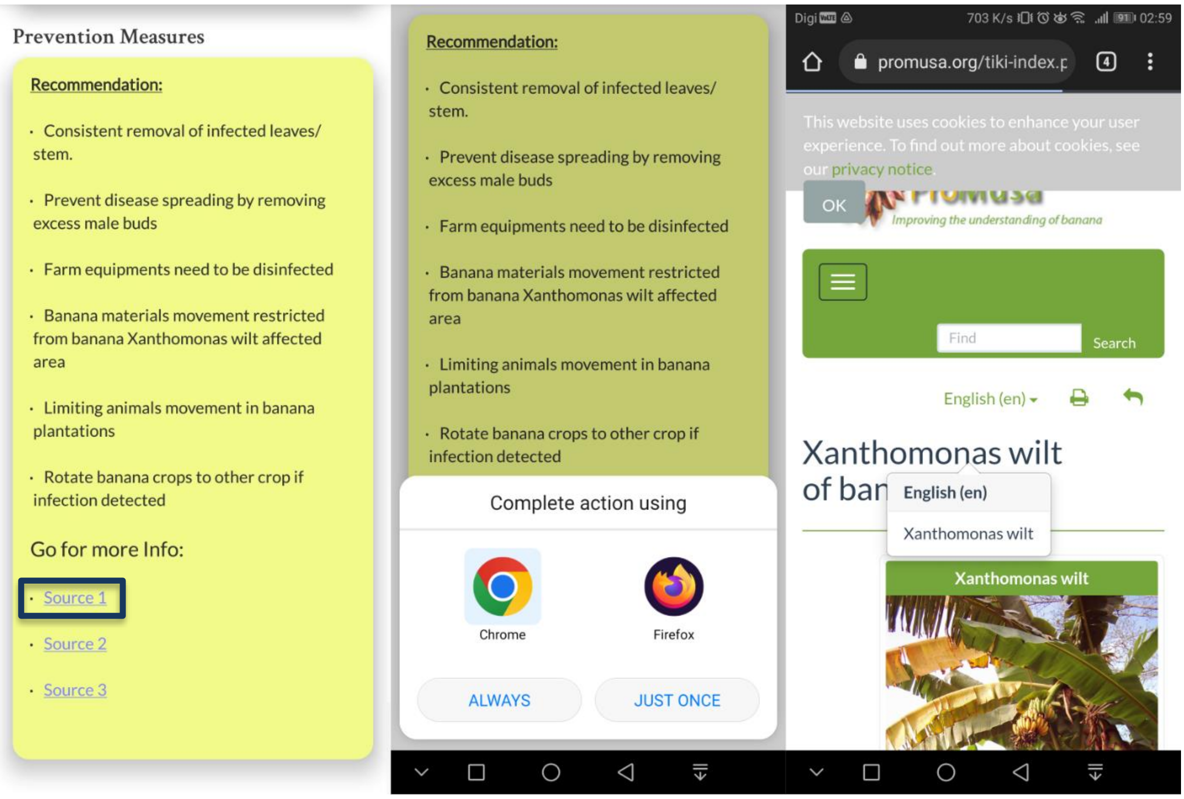
Recommendation:

- Consistent removal of infected leaves/ stem.
- Prevent disease spreading by removing excess male buds
- Farm equipments need to be disinfected
- Banana materials movement restricted from banana Xanthomonas wilt affected area
- Limiting animals movement in banana plantations
- Rotate banana crops to other crop if infection detected

Go for more Info:

- Source 1
- Source 2
- Source 3

Figure 5.4.5.1 Banana Crop Diseases Interface Operation Flow



Prevention Measures

Recommendation:

- Consistent removal of infected leaves/ stem.
- Prevent disease spreading by removing excess male buds
- Farm equipments need to be disinfected
- Banana materials movement restricted from banana Xanthomonas wilt affected area
- Limiting animals movement in banana plantations
- Rotate banana crops to other crop if infection detected

Go for more Info:

- Source 1
- Source 2
- Source 3

Recommendation:

- Consistent removal of infected leaves/ stem.
- Prevent disease spreading by removing excess male buds
- Farm equipments need to be disinfected
- Banana materials movement restricted from banana Xanthomonas wilt affected area
- Limiting animals movement in banana plantations
- Rotate banana crops to other crop if infection detected

Complete action using

This website uses cookies to enhance your user experience. To find out more about cookies, see our privacy notice.

Xanthomonas wilt of ban

Figure 5.4.5.2 More Info Operation Flow

Chapter 6 System Evaluation and Discussion

6.1 TensorFlow Keras Model Performance Evaluation

In this section, it is to elaborate on the technique to evaluate the performance of the TensorFlow Keras Model. Therefore, the model that perform better will be converted to TensorFlow Lite model which will be deployed into the mobile application.

6.1.1 Accuracy and Loss in Training and Validation (TensorBoard)

TensorBoard used as the visualization tool to evaluate the performance of the freeze model training (feature extractor model) and unfreeze model training (fine-tuned model).

Color(s)	Training/Validation	Freeze/Unfreeze Model
Red	Training	Unfreeze
Light Blue	Validation	Unfreeze
Orange	Training	Freeze
Dark Blue	Validation	Freeze

Table 6.1.1.1 Color differentiation in TensorBoard

Unfreeze Model Training Vs Freeze Model Training in Epoch Accuracy

Based on Figure 6.1.1.1, it is observed that the unfreeze model training accuracy is higher than the freeze model training accuracy. The reason of freeze model training accuracy stopped increasing because of weights and bias parameters are not able to fine tune due to every layer before classifier layer are frozen. Referring to Table 6.1.1.2, unfreeze model achieved 0.9728 which is higher than 0.9131 for the freeze model in training accuracy.

Type of model	Training Accuracy
Unfreeze	97.28%
Freeze	91.31%

Table 6.1.1.2 Training Accuracy between Unfreeze and Freeze Model

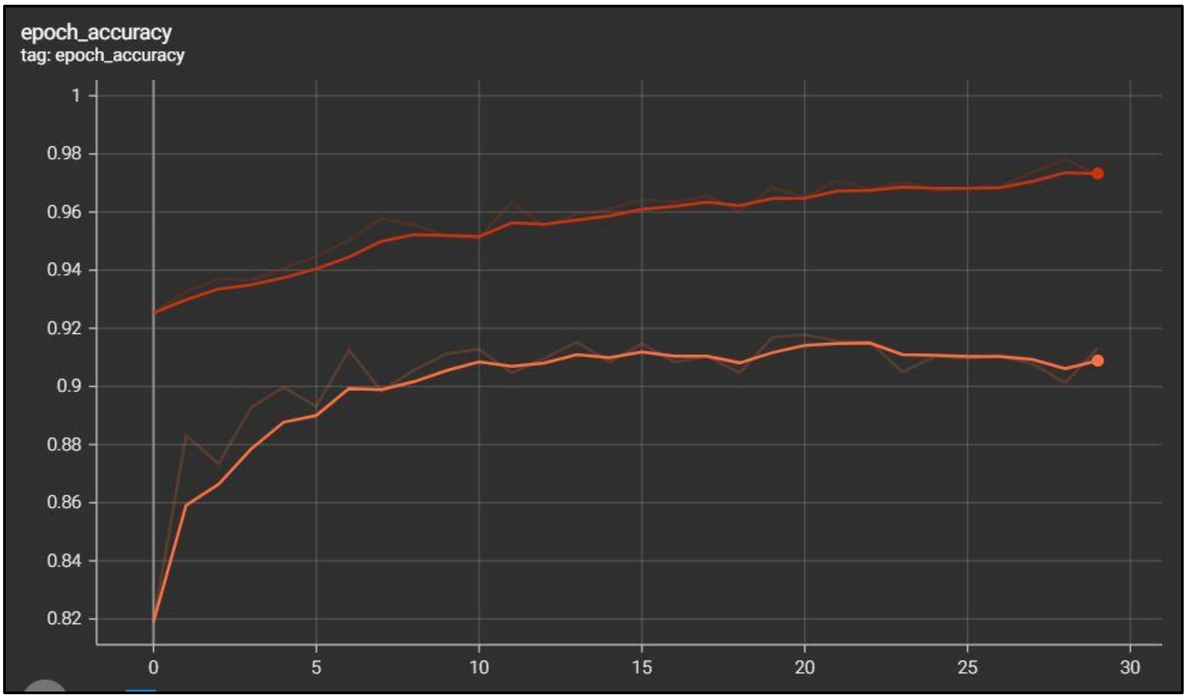


Figure 6.1.1.1 Unfreeze and Freeze Model Training in Epoch Accuracy

Unfreeze Model Validation Vs Freeze Model Validation in Epoch Accuracy

Based on Figure 6.1.1.2, it is observed that the unfreeze model validation accuracy is higher than the freeze model validation accuracy. Other than that, it is observed that freeze model validation accuracy having noticeable fluctuation than unfreeze model validation accuracy. The possible reason for that fluctuation is the freeze model may be classified the samples randomly. Referring to Table 6.1.1.3, unfreeze validation model achieved 0.8662 which is higher than 0.8587 for the freeze model in validation accuracy.

Type of model	Validation Accuracy
Unfreeze	86.62%
Freeze	85.87%

Table 6.1.1.3 Validation Accuracy between Unfreeze and Freeze Model

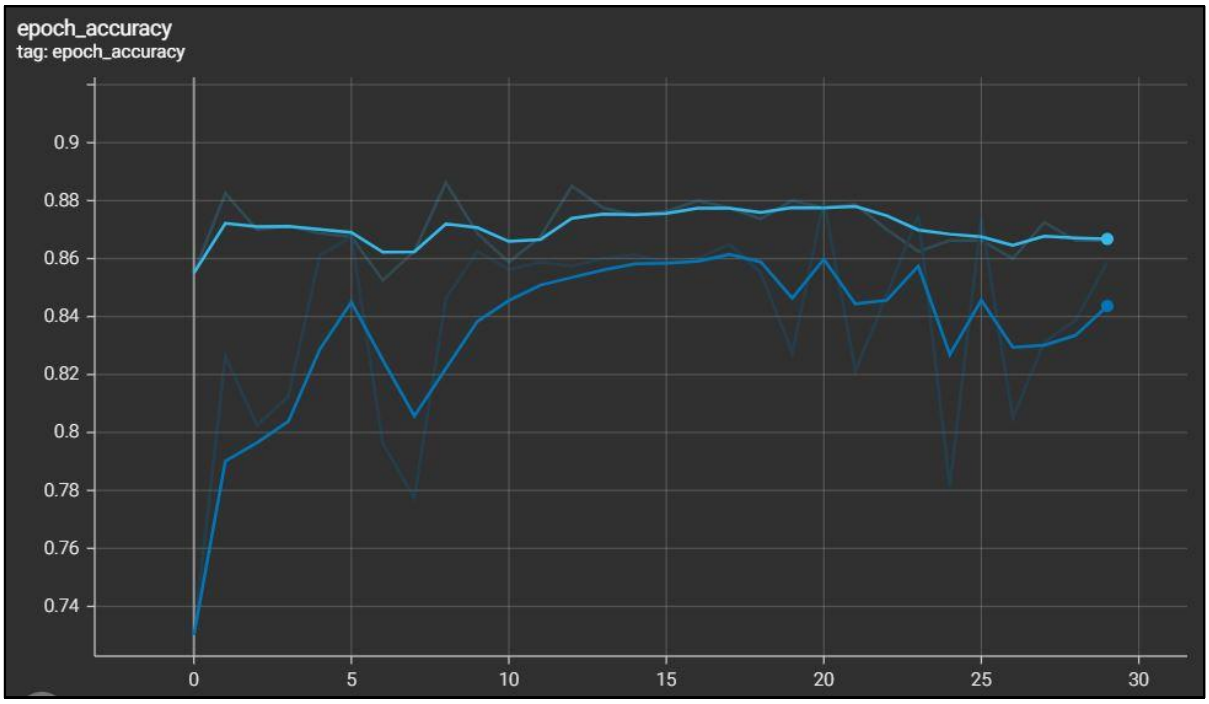


Figure 6.1.1.2 Unfreeze and Freeze Model Validation in Epoch Accuracy

Unfreeze Model Training Vs Freeze Model Training in Epoch Loss

Based on Figure 6.1.1.3, it is observed that the unfreeze model training loss is lower than the freeze model training loss. Referring to Table 6.1.1.4, unfreeze model achieved 0.07662 which is lower than 0.3078 for the freeze training model. Unfreeze model able to achieve a low training loss because of the few top layers of model enabled to update the weights and bias parameters and able to learn the features.

Type of model	Training Loss
Unfreeze	0.07662
Freeze	0.3078

Table 6.1.1.4 Training Loss between Unfreeze and Freeze Model

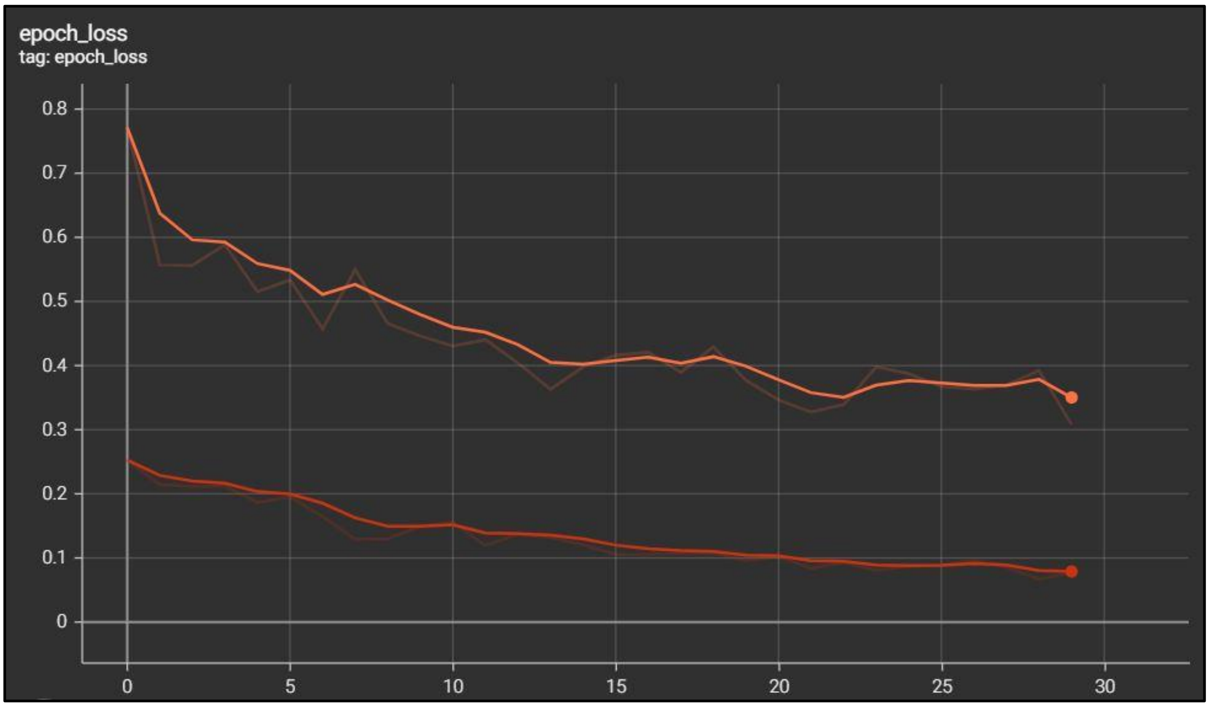


Figure 6.1.1.3 Unfreeze and Freeze Model Training in Epoch Loss

Unfreeze Model Validation Vs Freeze Model Validation in Epoch Loss

Based on Figure 6.1.1.4, it is observed that unfreeze model validation loss is lower than freeze model validation loss. However, by inspecting in the TensorBoard, it is perceived that the validation loss values for both models in the last epoch (at 30th epoch) are the same values as shown in the Table 6.1.1.5. However, it is also perceived that freezer model validation loss tends to have great fluctuation due to the model may classify randomly or sensitive to some noise in the validation dataset. Unfreeze model have a better stability of validation loss in general.

Type of model	Validation Loss
Unfreeze	0.6813
Freeze	0.6072

Table 6.1.1.5 Validation Loss between Unfreeze and Freeze Model

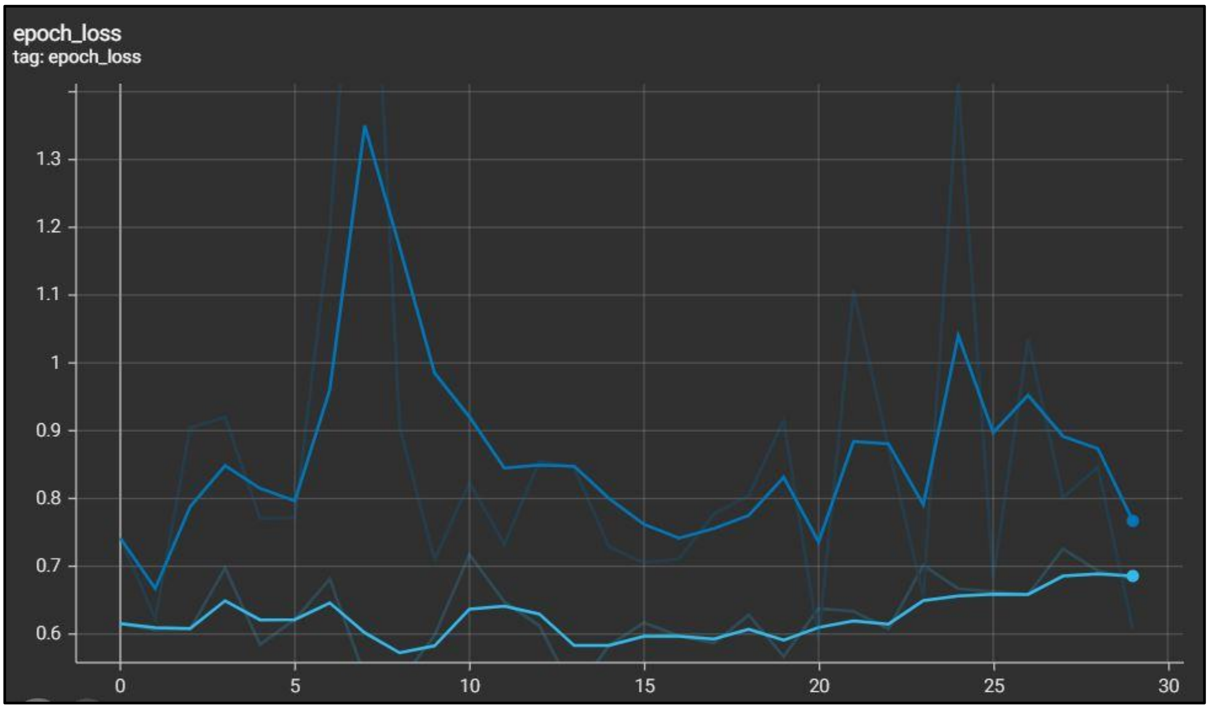


Figure 6.1.1.4 Unfreeze and Freeze Model Validation in Epoch Loss

Evaluation on Training and Validation Loss of Freeze Model

In Figure 6.1.1.5, it is observed that freeze model has great fluctuation in validation loss due to the noise and random classification. Other than that, by comparing to Figure 6.1.1.6, unfreeze model seems to have a better learning curve than freeze model in Figure 6.1.1.5.

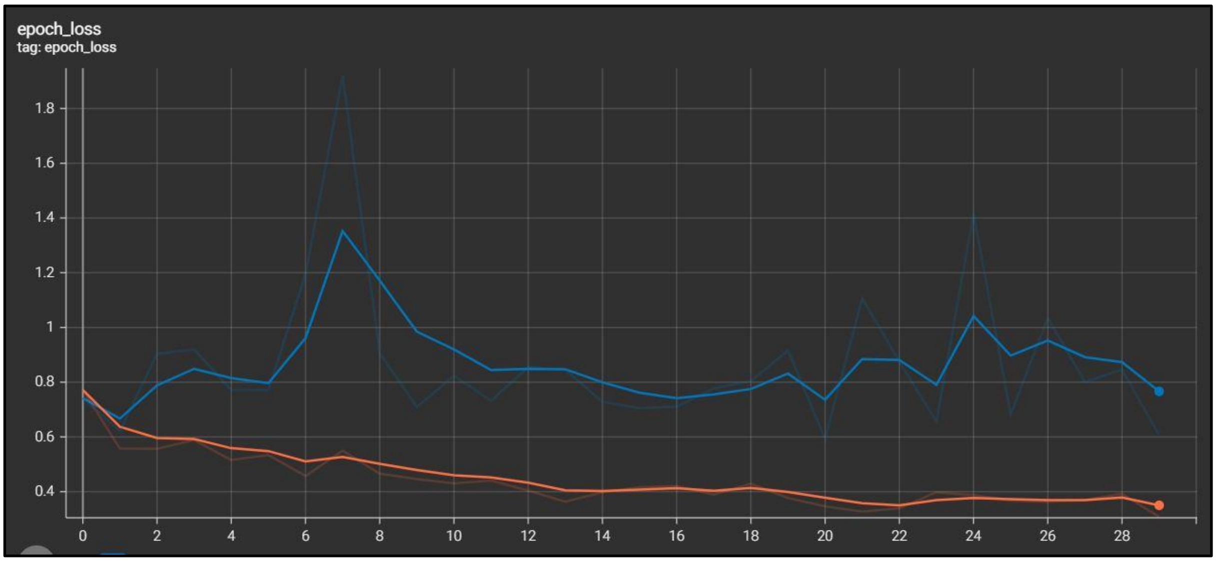


Figure 6.1.1.5 Freeze Model Training and Validation Loss Evaluation

Evaluation on Training and Validation Loss of Unfreeze Model

By observing Figure 6.1.1.6, the result of training and validation loss of the unfreeze model doesn't seem to be a model overfitting nor underfitting. It can be considering the unfreeze model is able to classify the validation samples and didn't classify randomly or not sensitive to the noise within in validation samples.

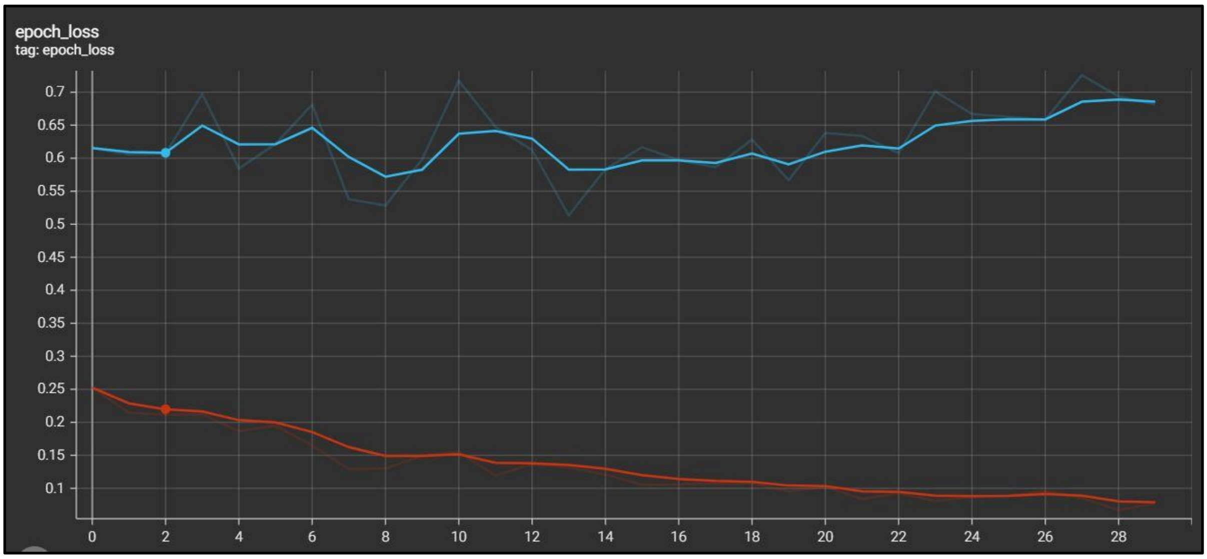


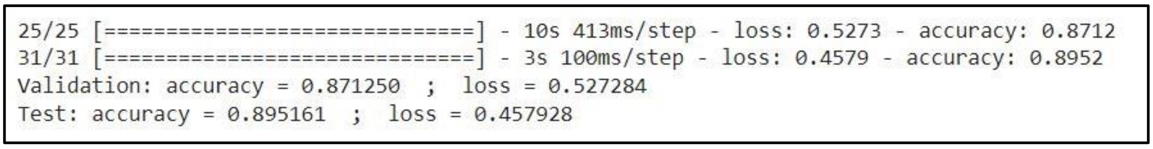
Figure 6.1.1.6 Unfreeze Model Training and Validation Loss Evaluation

6.1.2 Evaluate using Validation and Testing Dataset

Hyperparameters	Values
Epoch	Freeze Model: 30
	Unfreeze Model: 30
Optimizer	Freeze Model: Adam
	Unfreeze Model: Adam
Learning Rate	Freeze Model: 0.01
	Unfreeze Model: 0.0001

Table 6.1.2.1 Hyperparameters Used for Unfreeze and Freeze Model

Freeze Model Performance Metrics



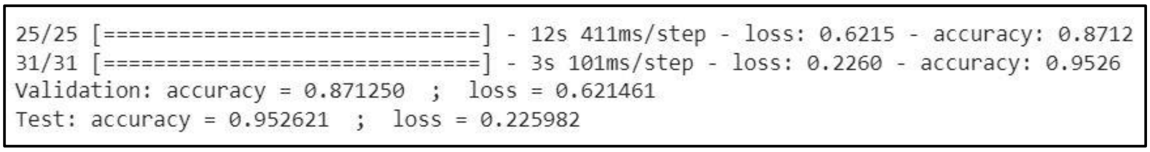
```
25/25 [==============================] - 10s 413ms/step - loss: 0.5273 - accuracy: 0.8712
31/31 [==============================] - 3s 100ms/step - loss: 0.4579 - accuracy: 0.8952
Validation: accuracy = 0.871250  ;  loss = 0.527284
Test: accuracy = 0.895161  ;  loss = 0.457928
```

Figure 6.1.2.1 Freeze Model Validation and Testing Performance Metrics

Type of Dataset/Type of Score	Accuracy	Loss
Validation	87.1250%	0.527284
Test	89.5161%	0.457928

Table 6.1.2.2 Accuracy and Loss of Freeze Model for Validation and Test Dataset

Unfreeze Model Performance Metrics



```
25/25 [==============================] - 12s 411ms/step - loss: 0.6215 - accuracy: 0.8712
31/31 [==============================] - 3s 101ms/step - loss: 0.2260 - accuracy: 0.9526
Validation: accuracy = 0.871250  ;  loss = 0.621461
Test: accuracy = 0.952621  ;  loss = 0.225982
```

Figure 6.1.2.2 Unfreeze Model Validation and Testing Performance Metrics

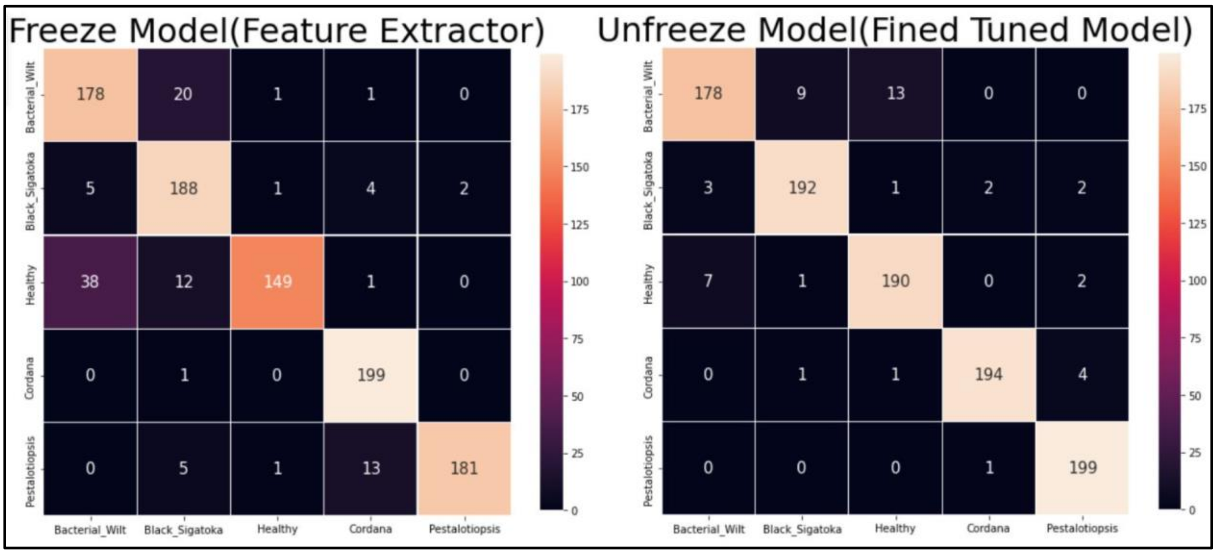
Type of Dataset/Type of Score	Accuracy	Loss
Validation	87.1250%	0.621461
Test	95.2621%	0.225982

Table 6.1.2.3 Accuracy and Loss of Unfreeze Model for Validation and Test Dataset

The performance metrics result for both figures above are carried out by using the ".evaluate()" function provided by TensorFlow Keras Model class. Validation and testing dataset used in this evaluation. By comparing between Table 6.1.2.2 and Table 6.1.2.3, unfreeze model have better result in evaluating for testing dataset than freeze model. Besides that, freeze model only have a lower validation loss than the unfreeze model validation loss. Hence, unfreeze model is the model that perform better.

6.1.3 Comparison in Confusion Matrix

Based on the Figure 6.1.3 below, it is observed that unfreeze model perform better in True Positive value for each class prediction. For Bacterial Wilt prediction, both models have the same count for False Negative. However, freeze model have greater count in False Positive for Bacterial Wilt prediction. Freeze model predicted wrongly may be due to its frozen pretrained parameters and unable to learn the features as Healthy class and Bacterial Wilt may have high similarity in features. In general, unfreeze model still perform better than freeze model.



Freeze Model(Feature Extractor)

	Bacterial_Wilt	Black_Sigatoka	Healthy	Cordana	Pestalotiopsis
Bacterial_Wilt	178	20	1	1	0
Black_Sigatoka	5	188	1	4	2
Healthy	38	12	149	1	0
Cordana	0	1	0	199	0
Pestalotiopsis	0	5	1	13	181

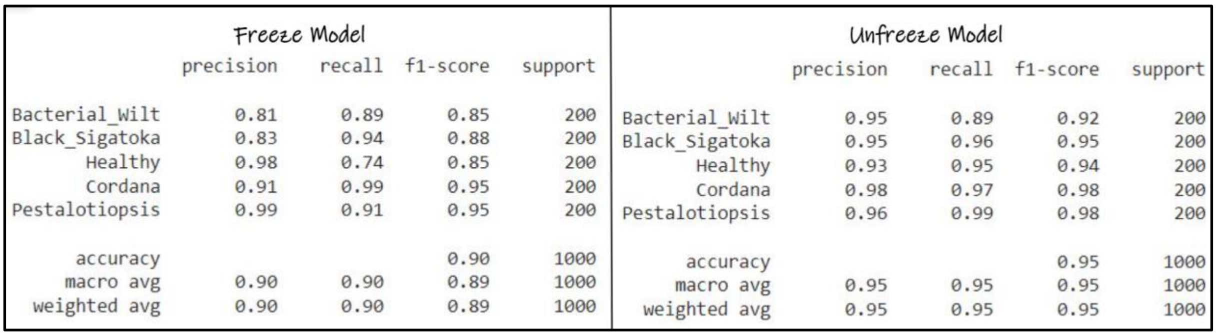
Unfreeze Model(Fined Tuned Model)

	Bacterial_Wilt	Black_Sigatoka	Healthy	Cordana	Pestalotiopsis
Bacterial_Wilt	178	9	13	0	0
Black_Sigatoka	3	192	1	2	2
Healthy	7	1	190	0	2
Cordana	0	1	1	194	4
Pestalotiopsis	0	0	0	1	199

Figure 6.1.3 Confusion Matrix Comparison between Freeze and Unfreeze Model

6.1.4 Comparison in Classification Report

Figure 6.1.4 depicts the classification report result performed by using the “classification_report” function provided by sklearn library. By comparing both model classification report, it is observed that the averaged accuracy of unfreeze model is 0.95 that higher than the averaged accuracy of freeze model. Since F1-score is the weighted average of both Precision score and Recall score and it is used to evaluate how model perform on each different class. Therefore, unfreeze model is the model that perform better when do classification on each different class.



Freeze Model

```
                precision    recall  f1-score   support

Bacterial_Wilt       0.81      0.89      0.85       200
Black_Sigatoka       0.83      0.94      0.88       200
       Healthy       0.98      0.74      0.85       200
       Cordana       0.91      0.99      0.95       200
Pestalotiopsis       0.99      0.91      0.95       200

      accuracy                           0.90      1000
     macro avg       0.90      0.90      0.89      1000
  weighted avg       0.90      0.90      0.89      1000
```

Unfreeze Model

```
                precision    recall  f1-score   support

Bacterial_Wilt       0.95      0.89      0.92       200
Black_Sigatoka       0.95      0.96      0.95       200
       Healthy       0.93      0.95      0.94       200
       Cordana       0.98      0.97      0.98       200
Pestalotiopsis       0.96      0.99      0.98       200

      accuracy                           0.95      1000
     macro avg       0.95      0.95      0.95      1000
  weighted avg       0.95      0.95      0.95      1000
```

Figure 6.1.4 Classification Report Comparison between Freeze and Unfreeze Model

6.1.5 Conclusion in TensorFlow Keras Model Performance Evaluation

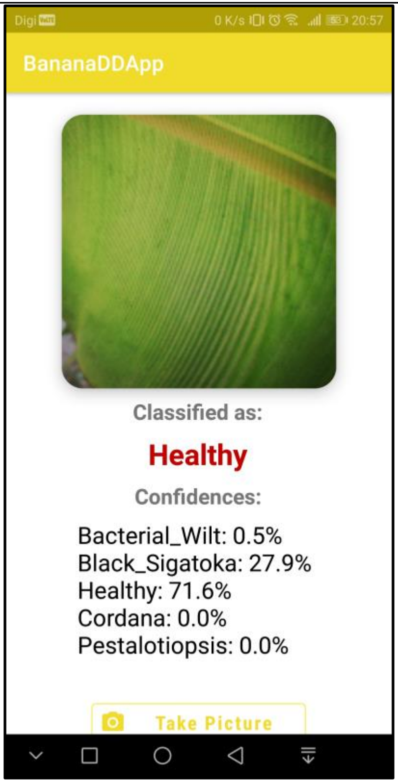
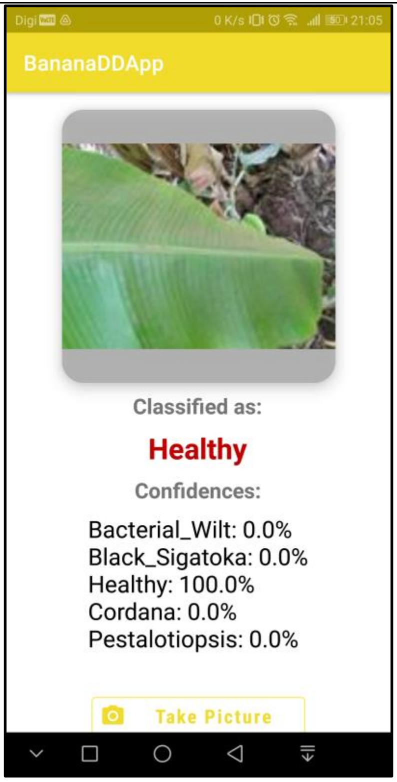
By going through the sub-chapter of 6.1, it is obvious that unfreeze model is the model that perform better, and it is selected to be convert to TensorFlow Lite model. Therefore, the TensorFlow Lite model have deployed into the developed Android mobile application.

6.2 Testing Setup and Result

This section is to show classification test and result of using the Detector developed in the mobile application. Captured picture from device camera and uploaded image from device storage will be considered in this testing setup.

6.2.1 Healthy Classification Test Result

Table 6.2.1 depicts the classification test of Healthy banana leaf in successful and unsuccessful result. Successful classification test result is based on that the image or picture have a satisfied light condition and properties. Unsuccessful classification test result is based on the bad condition in the picture or bad lighting condition during camera capturing.

Successful Classification Result	
 Digi 0 K/s 20:57 BananaDDApp Classified as: Healthy Confidences: Bacterial_Wilt: 0.5% Black_Sigatoka: 27.9% Healthy: 71.6% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Healthy Predicted Label: Healthy Confidence: 71.6% Picture capture from device camera	 Digi 0 K/s 21:05 BananaDDApp Classified as: Healthy Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 100.0% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Healthy Predicted Label: Healthy Confidence: 100.0% Using image from device storage

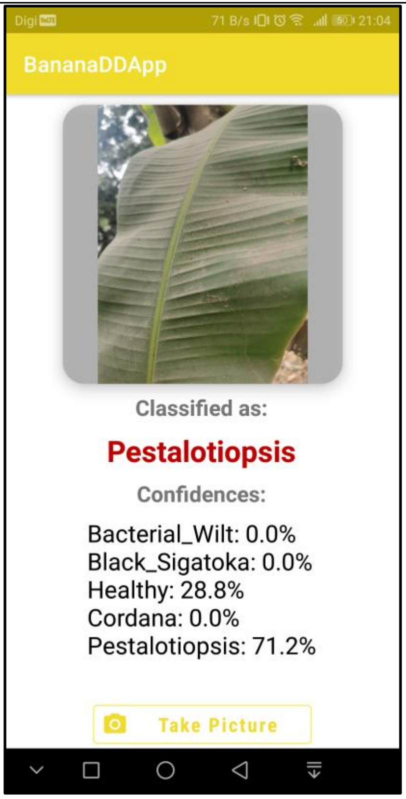
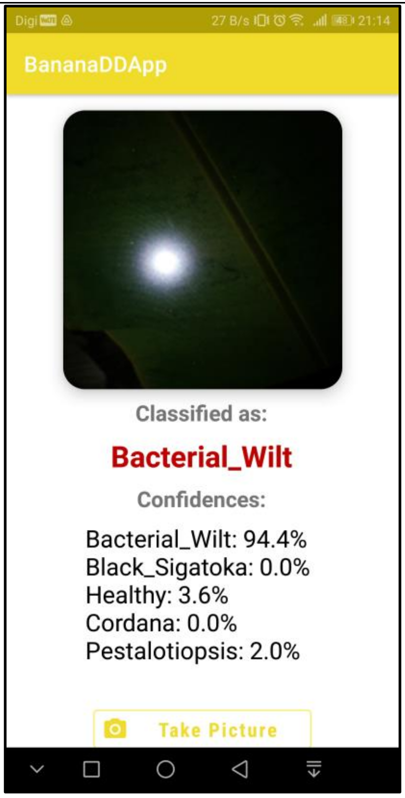
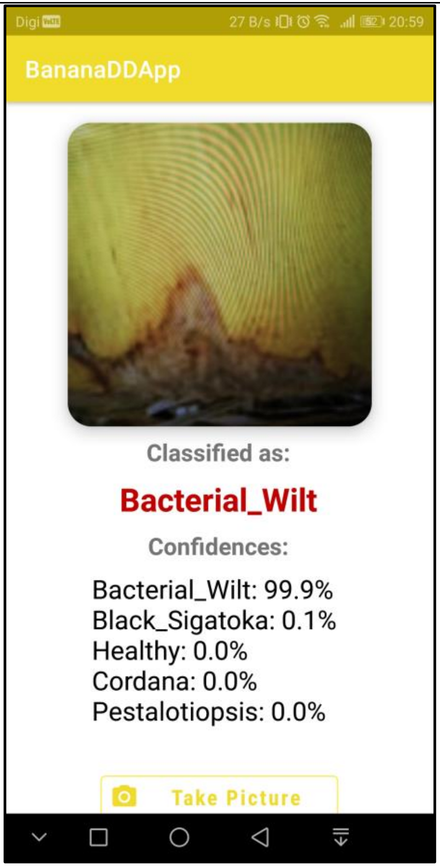
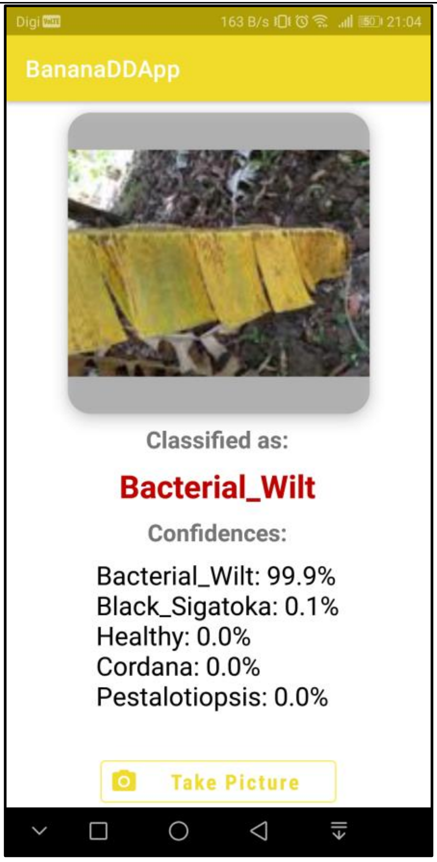
Unsuccessful Classification Result	
 Digi 71 B/s 21:04 BananaDDApp Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 28.8% Cordana: 0.0% Pestalotiopsis: 71.2% Take Picture True Label: Healthy Predicted Label: Pestalotiopsis Confidence (Healthy): 28.8% Confidence (Pestalotiopsis): 71.2% Using image uploaded from device storage. The healthy banana leaf in the image had some dirt or debris that affect the classification accuracy.	 Digi 27 B/s 21:14 BananaDDApp Classified as: Bacterial_Wilt Confidences: Bacterial_Wilt: 94.4% Black_Sigatoka: 0.0% Healthy: 3.6% Cordana: 0.0% Pestalotiopsis: 2.0% Take Picture True Label: Healthy Predicted Label: Bacterial Wilt Confidence (Healthy): 3.6% Confidence (Bacterial Wilt): 94.4% Capture picture using device camera under low light intensity condition with flashlight on

Table 6.2.1 Healthy Classification Test Result

6.2.2 Bacterial Wilt Classification Test Result

Table 6.2.2 depicts the classification test of Bacterial Wilt banana leaf in successful and unsuccessful result. Successful classification test result is based on that the image or picture have a satisfied light condition and properties. Unsuccessful classification test result is based on the picture is taken from far away and blurry picture capturing using camera.

Successful Classification Result	
 Digi 27 B/s 20:59 BananaDDApp Classified as: Bacterial_Wilt Confidences: Bacterial_Wilt: 99.9% Black_Sigatoka: 0.1% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Bacterial Wilt Predicted Label: Bacterial Wilt Confidence: 99.9% Picture capture from device camera	 Digi 163 B/s 21:04 BananaDDApp Classified as: Bacterial_Wilt Confidences: Bacterial_Wilt: 99.9% Black_Sigatoka: 0.1% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Bacterial Wilt Predicted Label: Bacterial Wilt Confidence: 99.9% Using image from device storage
Unsuccessful Classification Result	

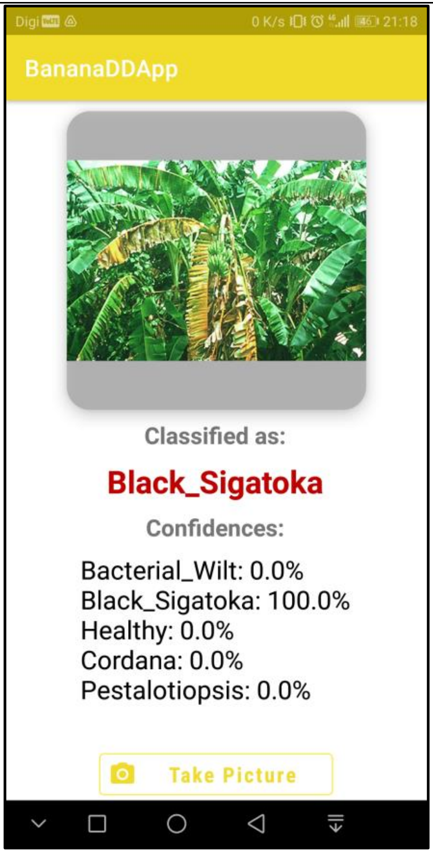
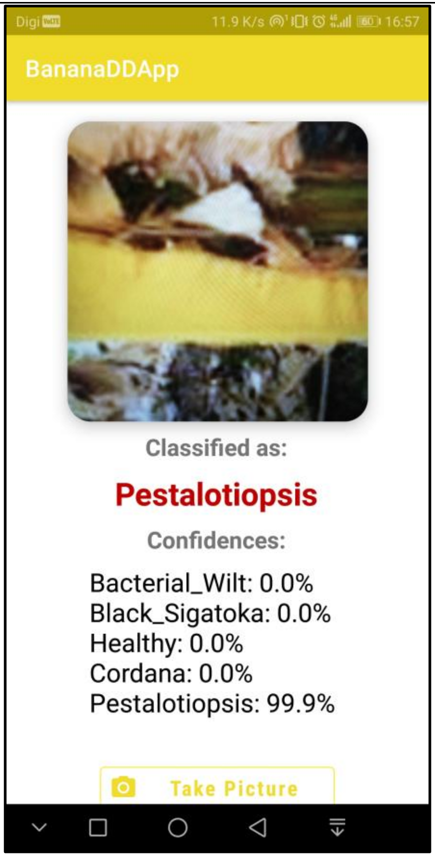
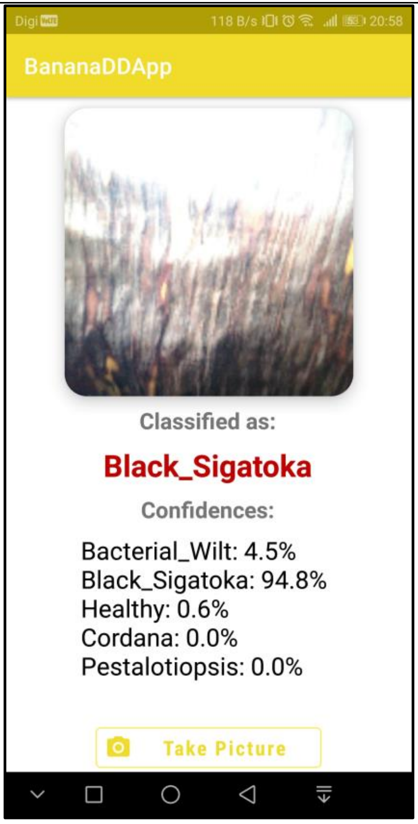
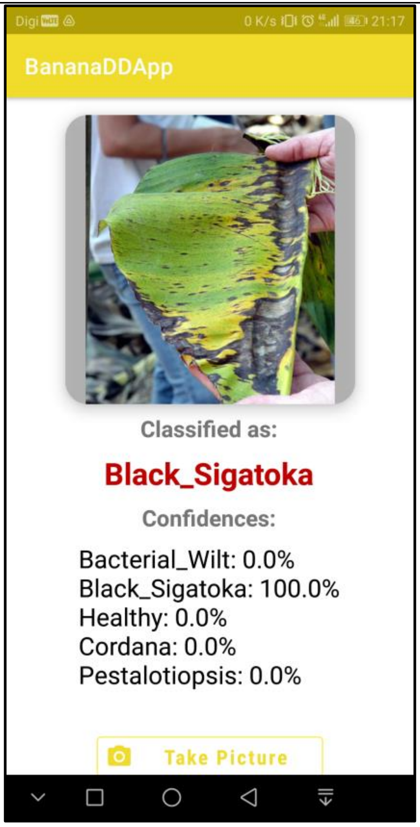
 Classified as: Black_Sigatoka Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 100.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 0.0% True Label: Bacterial Wilt Predicted Label: Black Sigatoka Confidence (Bacterial Wilt): 0.0% Confidence (Black Sigatoka): 100.0% Using image that taken from far away and it is uploaded from device storage	 Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 99.9% True Label: Bacterial Wilt Predicted Label: Pestalotiopsis Confidence (Bacterial Wilt): 0.0% Confidence (Pestalotiopsis): 99.9% Capture a blurry picture using device camera

Table 6.2.2 Bacterial Wilt Classification Result

6.2.3 Black Sigatoka Classification Test Result

Table 6.2.3 depicts the classification test of Black Sigatoka banana leaf in successful and unsuccessful result. Successful classification test result is based on that the image or picture have a high brightness condition and obstacles inside picture. Unsuccessful classification test result is based on blurry picture capturing using camera.

Successful Classification Result	
 Classified as: Black_Sigatoka Confidences: Bacterial_Wilt: 4.5% Black_Sigatoka: 94.8% Healthy: 0.6% Cordana: 0.0% Pestalotiopsis: 0.0% True Label: Black Sigatoka Predicted Label: Black Sigatoka Confidence: 94.8% Picture capture from device camera under high brightness	 Classified as: Black_Sigatoka Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 100.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 0.0% True Label: Black Sigatoka Predicted Label: Black Sigatoka Confidence: 100.0% Using image contains human finger or body as obstacles, uploaded from device storage
Unsuccessful Classification Result	

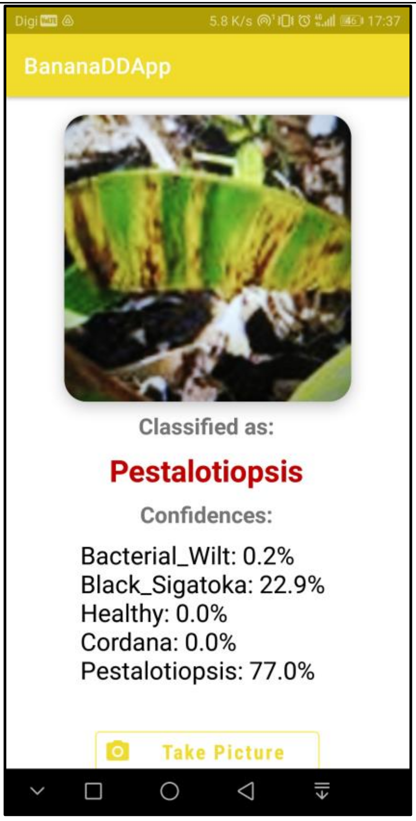
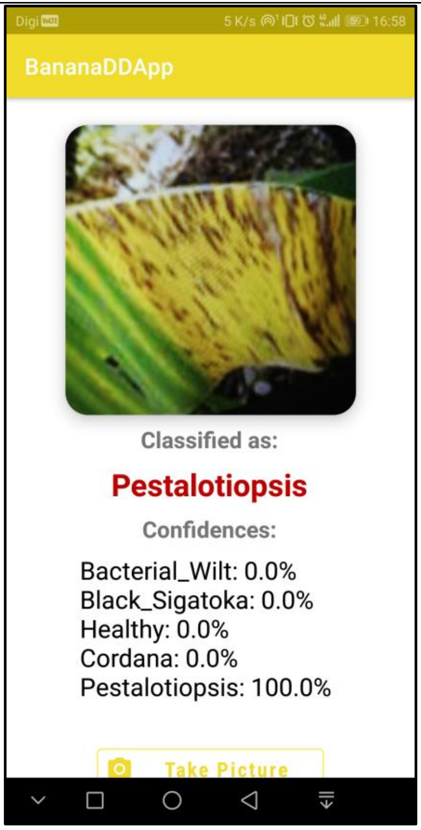
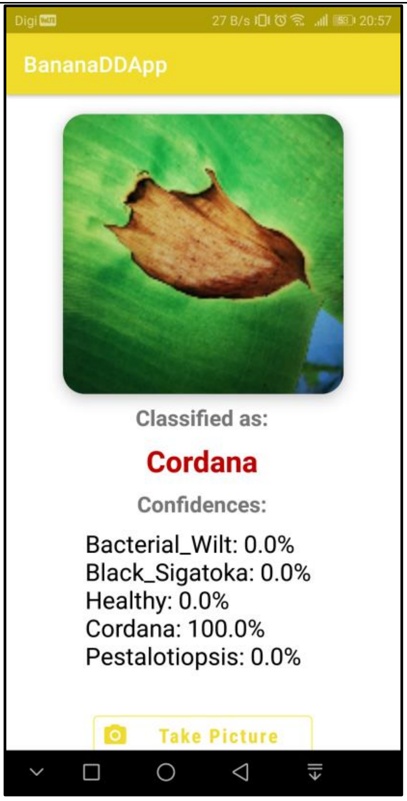
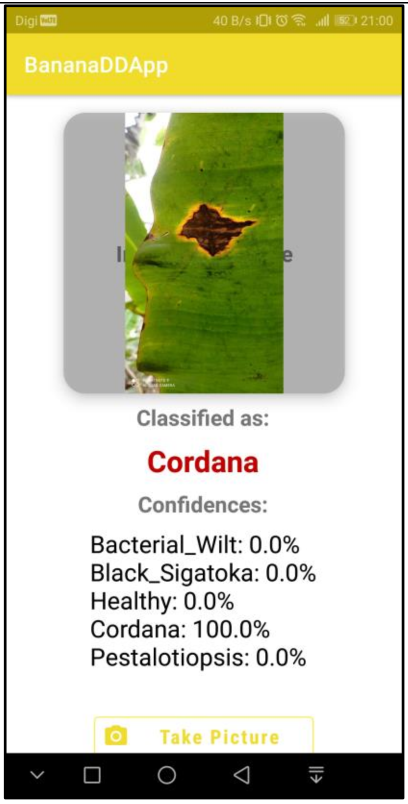
 BananaDDApp Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 0.2% Black_Sigatoka: 22.9% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 77.0% True Label: Black Sigatoka Predicted Label: Pestalotiopsis Confidence (Black Sigatoka): 22.9% Confidence (Pestalotiopsis): 77.0% Using blurry image uploaded from device storage	 BananaDDApp Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 100.0% True Label: Black Sigatoka Predicted Label: Pestalotiopsis Confidence (Black Sigatoka): 0.0% Confidence (Pestalotiopsis): 100.0% Capture blurry picture using device camera

Table 6.2.3 Black Sigatoka Classification Result

6.2.4 Cordana Classification Test Result

Table 6.2.4 depicts the classification test of Cordana banana leaf in successful and unsuccessful result. Successful classification test result is based on that the image or picture have a satisfied light condition and properties. Unsuccessful classification test result is based on blurry picture capturing using camera and bad lighting condition during capturing the picture.

Successful Classification Result	
 Digi 27 B/s 20:57 BananaDDApp Classified as: Cordana Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 0.0% Cordana: 100.0% Pestalotiopsis: 0.0% Take Picture True Label: Cordana Predicted Label: Cordana Confidence: 100.0% Picture capture from device camera	 Digi 40 B/s 21:00 BananaDDApp Classified as: Cordana Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 0.0% Cordana: 100.0% Pestalotiopsis: 0.0% Take Picture True Label: Cordana Predicted Label: Cordana Confidence: 100.0% Using image uploaded from device storage
Unsuccessful Classification Result	

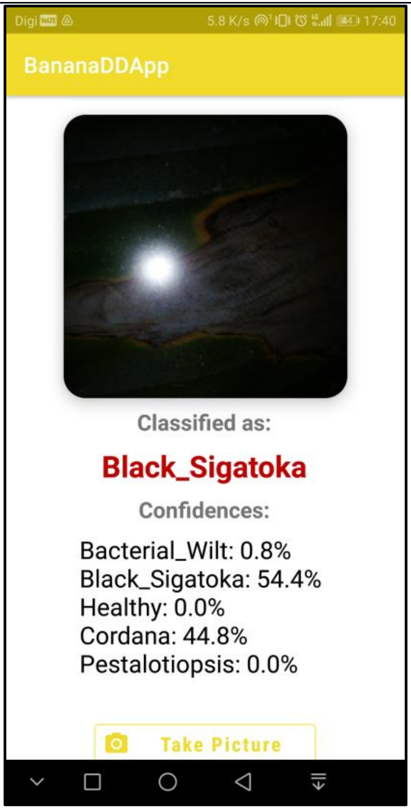
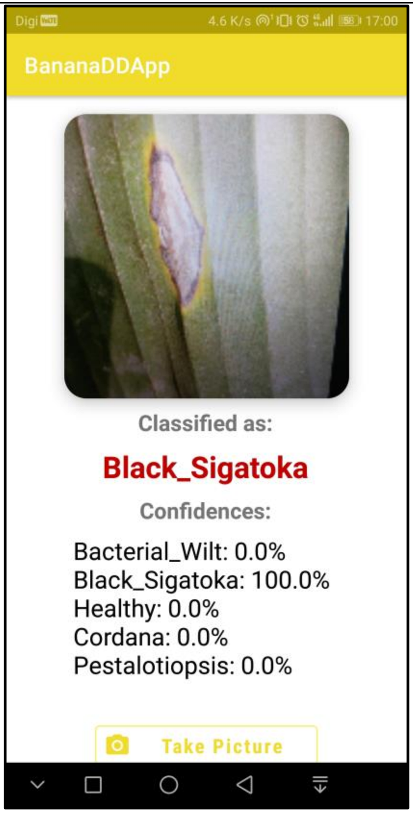
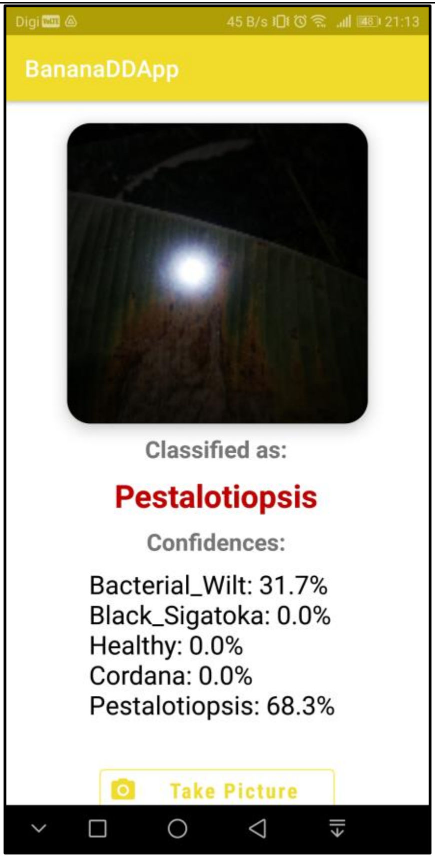
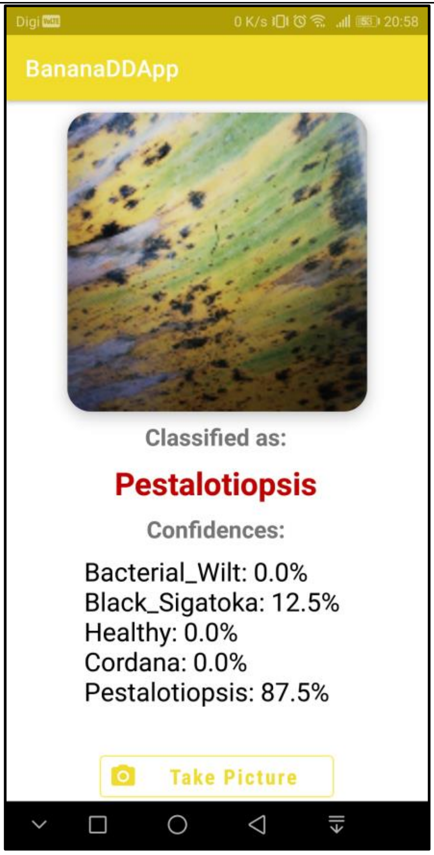
 BananaDDApp Classified as: Black_Sigatoka Confidences: Bacterial_Wilt: 0.8% Black_Sigatoka: 54.4% Healthy: 0.0% Cordana: 44.8% Pestalotiopsis: 0.0% Take Picture True Label: Cordana Predicted Label: Black Sigatoka Confidence (Cordana): 44.8% Confidence (Black Sigatoka): 54.4% Capture the picture using device camera under low light intensity condition	 BananaDDApp Classified as: Black_Sigatoka Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 100.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Cordana Predicted Label: Black Sigatoka Confidence (Cordana): 0.0% Confidence (Black Sigatoka): 100.0% Upload blurry image using device storage

Table 6.2.4 Cordana Classification Result

6.2.5 Pestalotiopsis Classification Test Result

Table 6.2.5 depicts the classification test of Pestalotiopsis banana leaf in successful and unsuccessful result. Successful classification test result is based on that the picture have bad lighting condition during capturing and a satisfied condition. Unsuccessful classification test result is based on bad lighting condition during capturing the picture.

Successful Classification Result	
 Digi 45 B/s 21:13 BananaDDApp Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 31.7% Black_Sigatoka: 0.0% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 68.3% Take Picture True Label: Pestalotiopsis Predicted Label: Pestalotiopsis Confidence: 68.3% Picture capture from device camera under low brightness with flashlight on	 Digi 0 K/s 20:58 BananaDDApp Classified as: Pestalotiopsis Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 12.5% Healthy: 0.0% Cordana: 0.0% Pestalotiopsis: 87.5% Take Picture True Label: Pestalotiopsis Predicted Label: Pestalotiopsis Confidence: 87.5% Using image uploaded from device storage
Unsuccessful Classification Result	

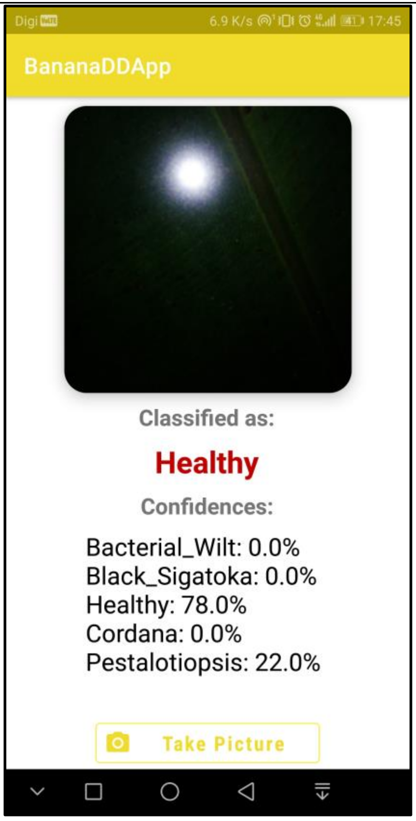
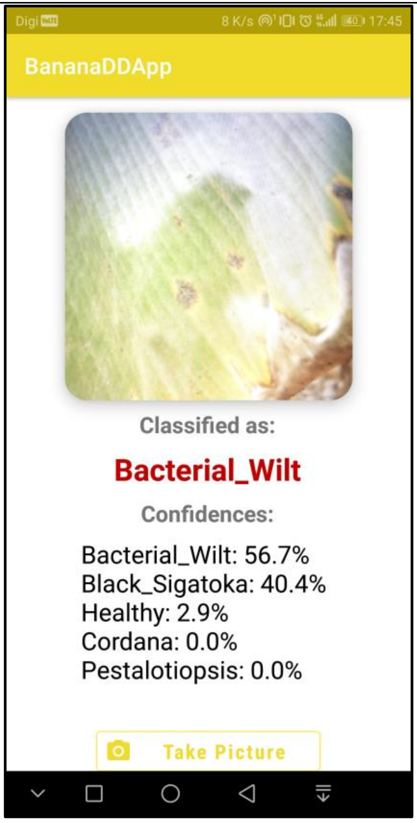
 BananaDDApp Classified as: Healthy Confidences: Bacterial_Wilt: 0.0% Black_Sigatoka: 0.0% Healthy: 78.0% Cordana: 0.0% Pestalotiopsis: 22.0% Take Picture True Label: Pestalotiopsis Predicted Label: Healthy Confidence (Pestalotiopsis): 22.0% Confidence (Healthy): 78.0% Capture picture using device camera under extremely low light intensity condition	 BananaDDApp Classified as: Bacterial_Wilt Confidences: Bacterial_Wilt: 56.7% Black_Sigatoka: 40.4% Healthy: 2.9% Cordana: 0.0% Pestalotiopsis: 0.0% Take Picture True Label: Pestalotiopsis Predicted Label: Bacterial Wilt Confidence (Pestalotiopsis): 0.0% Confidence (Bacterial Wilt): 56.7% Using an image with high brightness uploaded from device storage

Table 6.2.5 Pestalotiopsis Classification Result

6.2.6 Conclusion (Strengths and Weaknesses)

Strengths

The Detector can perform well for the captured pictures or uploaded images that with enough brightness, obstacles exist in picture and moderate low light intensity. Other than that, the Detector also perform exceptionally well if the picture or image is well taken.

Weaknesses

However, it might face some challenges if those images or pictures having the image condition such as extreme low brightness, image that taken from a far distance and extreme blurry image. Some of the class have high similarity features between them, therefore sometimes the classification will have bias on either one. For example, Cordana with Black Sigatoka have high similarity in the features based on the Table 6.2.4. Other than that, Black Sigatoka have high similarity features with Pestalotiopsis as well based on the Table 6.2.3.

6.3 Project Challenges

6.3.1 Resolved Project Challenges

There are several challenges that have increased the difficulty of implementing the process of the mobile application however it was able to be resolved:

1. Several library version incompatibles

   This issue happened when trying to build the project in the Android Studio. Several libraries include the SDK build and platform tools version does not compatible to the requirement of the project. Other than that, a project builds automation management tool named as Gradle project sync does require a lot of libraries update. Hence, this increases the time consumption of start the mobile application development.

2. TensorFlow code understanding

   The study on deep learning neural network does require a lot of understanding of the model selection, pre-trained model mechanisms and construct the layers in the model. Research on the latest deep learning model comparison to produce a final decision on choosing which is the baseline model available in the open source from TensorFlow.

3. Data finding

   This is the most time exhausting process before the start of the mobile application development. The finding of the dataset had spent up two weeks to acquire a satisfied dataset. Two datasets have been found and need to combine. Finalized by separation within the dataset of consisting training set and test set.

6.3.2 Unresolved Project Challenges

However, there are still some challenges are difficult to resolve:

1. TensorFlow Lite Model Accuracy

   The deployed TensorFlow Lite model will still have some misclassification. This maybe caused by limitation of the model generalization power after training. Other than that, it may still require tremendous amount of dataset images to increase the model accuracy.

2. Mobile Application Implementation Bug

   Implemented Java code was logically and performed well during development and actual use. However, sometimes may still creates some bugs. For example, added comments did not display date and time for that particular comment even though the implementation was successful. Other than that, able to post without inserting post image even though it worked well after successfully developed.

6.4 Objectives Evaluation

The proposed system has achieved the objectives stated below:

- To detect banana diseases using EfficientNetV2 architecture as the deep learning approach for construct neural network training model

  In this project, the neural network training model have constructed based on the EfficientNetV2 architecture. The constructed neural network training model have trained with the dataset of infected banana leaf images. The trained neural network model can perform classification on each class of disease.
- To deploy the banana disease detection trained model into the mobile application

  The banana disease detection trained model have successfully converted to an optimized format which suitable to work in Android mobile application.
- To implement a community feature in the mobile application for users perform information exchange purpose

  Community feature have developed in the mobile application. User’s post has uploaded, and they added comment in each post in the community. Information exchange in the application is achieved.
- To implement a banana diseases prevention and control measures information feature for alert and recommend users

  Developed mobile application have equipped with the prevention and control measures information for users to read and understand.

Chapter 7 Conclusion and Future Work

7.1 Conclusion on Project Achievements

A TensorFlow Keras banana disease detection deep learning neural network model has successfully developed. The banana disease detection deep learning neural network model is successfully converted to TensorFlow Lite model. The TensorFlow Lite model is ready to deploy into the mobile application for development. Total of three main features in mobile application have developed such as the community feature, banana diseases prevention measures information and deployed TensorFlow Lite model as main component for Detector feature. The other feature such as the account registration and login feature have developed. The developed mobile application can act as an early-stage diagnosis technique for classifying the banana disease. This allowed farmers to have early warning towards their banana plant health conditions. Other than that, they can perform prevention and control measures as to seize the disease outbreak in their banana plantation. Moreover, this may reduce their financial loss and possible to increase the bananas yields. Besides that, they can share their thoughts and requesting helps by post their banana leaf or relevant problem in the community feature. They also can view the banana diseases information such as prevention measures and symptoms as well.

7.2 Future Work

In terms of future work, it is recommended to find more dataset images for different kind of major banana diseases to increase the capability of the deployed TensorFlow Lite Model. Other than that, the community feature can be enhanced with the functionality such as allow users to edit their post and comment and delete their own added post and comment. Besides that, users can post video in community feature. Recent news related to banana or banana diseases can be implemented as a new feature into the mobile application. This can further increase the usability of the mobile application. New model building and deployment pipeline that receive images or picture uploaded by users which store into a cloud storage for increase the dataset and retrain the model to increase the classification performance.

REFERENCES

[1] S. A. Tumin and A. A. Shaharudin, “Views - Articles - Research - Khazanah Research Institute,” Feb. 28, 2019. http://www.krinstitute.org/Views-@-Banana-;_The_Worlds_Most_Popular_Fruit.aspx (accessed Apr. 13, 2022).

[2] M. G. Selvaraj *et al.*, “AI-powered banana diseases and pest detection,” *Plant Methods*, vol. 15, no. 1, Aug. 2019, doi: 10.1186/s13007-019-0475-z.

[3] W. S. Mcculloch and W. Pitts, “A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY,” *BULLETIN OF MATHEMATICAL BIOPHYSICS*, vol. 5, 1943, Accessed: Apr. 13, 2022. [Online]. Available: https://home.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf

[4] F. Rosenblatt, “THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1,” 1958. Accessed: Apr. 13, 2022. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf

[5] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *AT & T Bell Laboratories Holmdel, NJ 07722 USA*, vol. 1, pp. 541–551, 1989, Accessed: Apr. 13, 2022. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf

[6] IBM Cloud Education, “What are Neural Networks? - Malaysia | IBM,” Aug. 17, 2020. https://www.ibm.com/my-en/cloud/learn/neural-networks#toc-history-of-rIfu5uF2 (accessed Apr. 13, 2022).

[7] Simplilearn, “What is Deep Learning and How Does It Works [Updated],” Feb. 21, 2021. https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning (accessed Apr. 13, 2022).

[8] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science,” *Towards Data Science*, Dec. 16, 2018. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed Apr. 13, 2022).

[9] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Accessed: Apr. 13, 2022. [Online]. Available: https://arxiv.org/pdf/1704.04861v1.pdf

[10] A. Pujara, "MobileNet Convolutional neural network Machine Learning Algorithms | Analytics Vidhya," *Analytics Vidhya*, Jul. 04, 2020. https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470 (accessed Apr. 13, 2022).

[11] M. Tan and Q. v Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks".

[12] M. Tan and Q. v Le, "EfficientNetV2: Smaller Models and Faster Training," 2021, Accessed: Apr. 13, 2022. [Online]. Available: https://github.com/google/

[13] N. Saranya, L. Pavithra, N. Kanthimathi, B. Ragavi, and P. Sandhiyadevi, "Detection of Banana Leaf and Fruit Diseases Using Neural Networks," in *Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020)*, 2020, pp. 493–499. Accessed: Apr. 13, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9183006

[14] S. L. Sanga, D. Machuve, and K. Jomanga, "Mobile-based Deep Learning Models for Banana Disease Detection," 2020. [Online]. Available: www.etasr.com

[15] CGIAR, "Tumaini: an AI-powered mobile app for pests and diseases - CGIAR," 2017. https://www.cgiar.org/innovations/tumaini-an-ai-powered-mobile-app-for-pests-and-diseases/ (accessed Apr. 13, 2022).

[16] CIAT, "Tumaini Mobile App | CIAT." https://ciat.cgiar.org/phenomics-platform/tumaini/ (accessed Apr. 13, 2022).

[17] Vision spectra, "AI-Based Portable App Could Help Growers Prevent Production Losses | The News Wire | Vision Spectra." https://www.photonics.com/Articles/AI-Based_Portable_App_Could_Help_Growers_Prevent/a65020 (accessed Apr. 13, 2022).

[18] CGIAR, "Plant disease diagnosis using artificial intelligence: a case study on Plantix - CGIAR Platform for Big Data in Agriculture." https://bigdata.cgiar.org/digital-intervention/plant-disease-diagnosis-using-artificial-intelligence-a-case-study-on-plantix/ (accessed Apr. 13, 2022).

[19] Fresh Fruit Portal, "Crop doctor app Plantix fills small-holder farmer advisory gap and democratizes knowledge - FreshFruitPortal.com," Dec. 12, 2019.

https://www.freshfruitportal.com/news/2019/12/12/crop-doctor-app-plantix-fills-farmer-advisory-gap-and-democratizes-knowledge/ (accessed Apr. 13, 2022).

[20] Networked India, "German Startup PEAT's Plant Disease App Is Empowering Indian Farmers - Networked India." https://www.networkedindia.com/2016/10/11/german-startup-peats-plant-disease-app-empowering-indian-farmers/ (accessed Apr. 13, 2022).

[21] fouxa.com, "About fouxa." https://fouxa.com/moreinformation/moreinformation.html (accessed Apr. 13, 2022).

[22] G. Owomugisha, "godliver/source-code-BBW-BBS: Implementation-BBW and BBS Diseases," 2014. https://github.com/godliver/source-code-BBW-BBS (accessed Apr. 13, 2022).

[23] K. al Mahmud, "Banana Leaf Dataset | Kaggle," 2019. https://www.kaggle.com/datasets/kaiesalmahmud/banana-leaf-dataset (accessed Apr. 13, 2022).

[24] Y. Hailu, "Banana Leaf Disease Images - Mendeley Data," Sep. 14, 2021. https://data.mendeley.com/datasets/rjykr62kdh/1 (accessed Apr. 13, 2022).

[25] dvschultz, duskvirkus, blueshirt, and mudphone, "dvschultz/dataset-tools: Tools for quickly normalizing image datasets." https://github.com/dvschultz/dataset-tools (accessed Apr. 13, 2022).

[26] Analytics Vidhya, "Image Augmentation Keras | Keras ImageDataGenerator," 2020. https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/ (accessed Apr. 13, 2022).

[27] Keras, "Keras Applications." https://keras.io/api/applications/ (accessed Apr. 13, 2022).

[28] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21K Pretraining for the Masses," Apr. 2021, doi: 10.48550/arxiv.2104.10972.

[29] P. Sharma, "Keras Dense Layer Explained for Beginners - MLK - Machine Learning Knowledge," Oct. 20, 2020. https://machinelearningknowledge.ai/keras-dense-layer-explained-for-beginners/ (accessed Apr. 13, 2022).

[30] fchollet, "Transfer learning & fine-tuning," May 12, 2020. https://keras.io/guides/transfer_learning/ (accessed Apr. 13, 2022).

[31] TensorFlow, "tf.keras.Model | TensorFlow Core v2.8.0." https://www.tensorflow.org/api_docs/python/tf/keras/Model (accessed Apr. 14, 2022).

[32] TensorFlow, “Model optimization | TensorFlow Lite.” https://www.tensorflow.org/lite/performance/model_optimization (accessed Apr. 15, 2022).

[33] TensorFlow, “tf.lite.Interpreter | TensorFlow Core v2.8.0.” https://www.tensorflow.org/api_docs/python/tf/lite/Interpreter (accessed Apr. 15, 2022).

Appendix A

WEEKLY REPORT

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 1**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Confirm project title

2. WORK TO BE DONE

- Study Android mobile application development
- Do report chapter 1
- Modification on TensorFlow model building pipeline

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 2**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Finish Report Chapter 1

2. WORK TO BE DONE

- Modification on TensorFlow model building pipeline
- Study Android mobile application development
- Do report chapter 2

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 3**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Finish report chapter 2

2. WORK TO BE DONE

- Modification on TensorFlow model building pipeline
- Study Android mobile application development
- Finding more images for dataset

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 4**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Modification on TensorFlow model building pipeline

2. WORK TO BE DONE

- Testing the TensorFlow model
- Study Android mobile application development
- Finding more images for dataset

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 5**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE • Finding more images for dataset
2. WORK TO BE DONE • Testing the TensorFlow model • Study Android mobile application development
3. PROBLEMS ENCOUNTERED **No problem encounters**
4. SELF EVALUATION OF THE PROGRESS **MODERATE**

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 6**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Testing the TensorFlow model

2. WORK TO BE DONE

- Convert TensorFlow model to TensorFlow Lite model and do inference testing
- Study Android mobile application development
- Initiate programming on community feature

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 7**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Study Android mobile application development

2. WORK TO BE DONE

- Convert TensorFlow model to TensorFlow Lite model and do inference testing
- Initiate programming on community feature
- Study on using Firebase platform

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 8**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Initiate programming on community feature
- Convert TensorFlow model to TensorFlow Lite model and do inference testing

2. WORK TO BE DONE

- Study on using Firebase platform
- Finish community feature
- Deploy TensorFlow Lite model into the application

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 9**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Study on using Firebase platform
- Finish community feature
- Deploy TensorFlow Lite model into the application

2. WORK TO BE DONE

- Testing the deployed TensorFlow Lite model
- Program detector feature
- Program banana crop diseases information feature

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Slow

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 10**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Testing the deployed TensorFlow Lite model
- Program detector feature
- Program banana crop diseases information feature

2. WORK TO BE DONE

- Do report chapter 3, 4, 5, 6, 7 and other parts
- Mobile application testing and debugging

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 11**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Finish report chapter 3 and 4
- Mobile application testing and debugging

2. WORK TO BE DONE

- Do report chapter 5, 6, 7 and other parts

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 12**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Finish report chapter 5, 6, 7 and other parts

2. WORK TO BE DONE

- Finalizing parts in the project

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no.: 13**
Student Name & ID: Lee Wai Hin (18ACB02736)	
Supervisor: Ts Lai Siew Cheng	
Project Title: Banana Diseases Detection Mobile Application	

1. WORK DONE

- Finalizing parts in the project

2. WORK TO BE DONE

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Moderate

Supervisor's signature

Student's signature

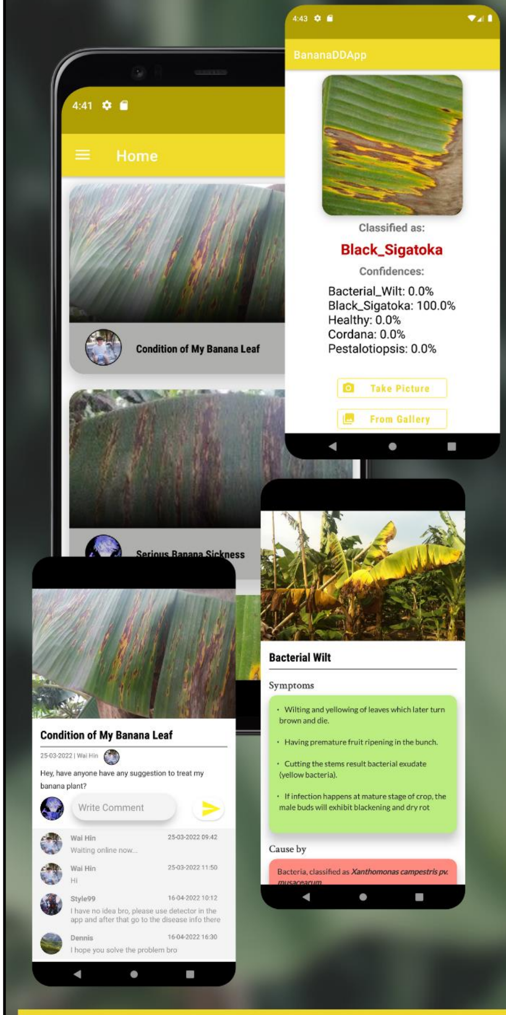
APPENDIX

POSTER

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

BANANA DISEASES DETECTION MOBILE APPLICATION

by Lee Wai Hin
supervised by Ts Lai Siew Cheng



Condition of My Banana Leaf

Serious Banana Sickness

Classified as:
Black_Sigatoka
Confidences:
Bacterial_Wilt: 0.0%
Black_Sigatoka: 100.0%
Healthy: 0.0%
Cordana: 0.0%
Pestalotiopsis: 0.0%

Condition of My Banana Leaf
25-03-2022 | Wai Hin
Hey, have anyone have any suggestion to treat my banana plant?
Write Comment

Wai Hin 25-03-2022 09:42
Waiting online now...

Wai Hin 25-03-2022 11:50
Hi

Style99 16-04-2022 10:12
I have no idea bro, please use detector in the app and after that go to the disease info there

Dennis 16-04-2022 16:30
I hope you solve the problem bro

Bacterial Wilt

Symptoms

- Wilting and yellowing of leaves which later turn brown and die.
- Having premature fruit ripening in the bunch.
- Cutting the stems result bacterial exudate (yellow bacteria).
- If infection happens at mature stage of crop, the male buds will exhibit blackening and dry rot

Cause by

Bacteria, classified as *Xanthomonas campestris pv. musacearum*

Introduction

This project is aimed to develop a mobile application to classify banana disease of infected banana leaf picture using deep learning approach.

Methods

Detector
- Capture or upload infected banana leaf picture
- Generate classification result

Community
- Post your banana plant problem
- Read people post and comments
- Add your thoughts in comment section

Banana Crop Disease
- Display banana diseases information
- Symptoms, causal and prevention measures

Discussion

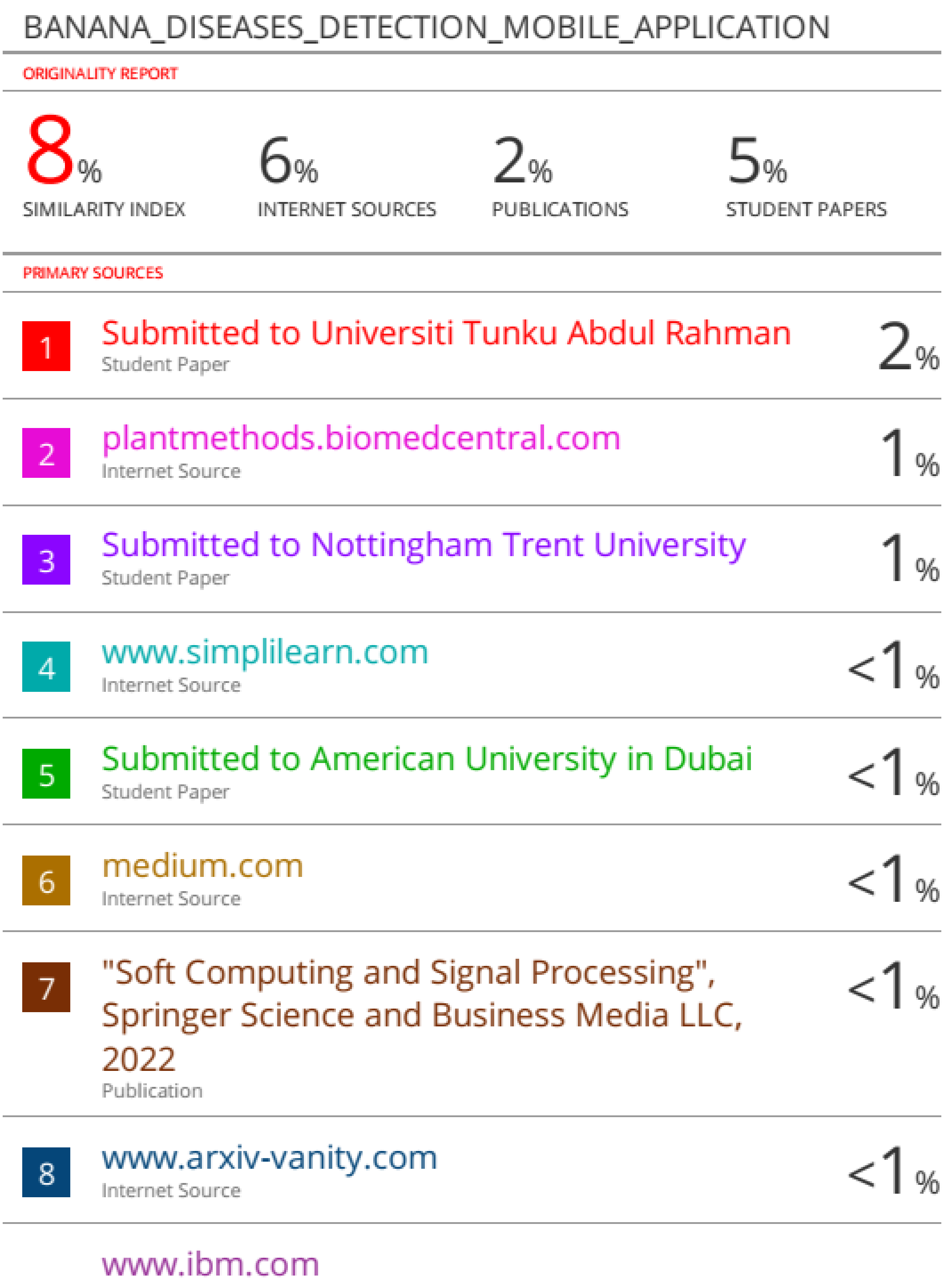
Software Tools Used:
- Google Colaboratory
- TensorFlow & Keras
- Android Studio

Programming Langguage Used:
- Python
- Java

Conclusion

- Easy-to-use banana diseases detector as early-stage disease diagnosis tool
- Platform for sharing their problems and ideas
- Embrace users with more banana diseases information

Appendix E: PLAGIARISM



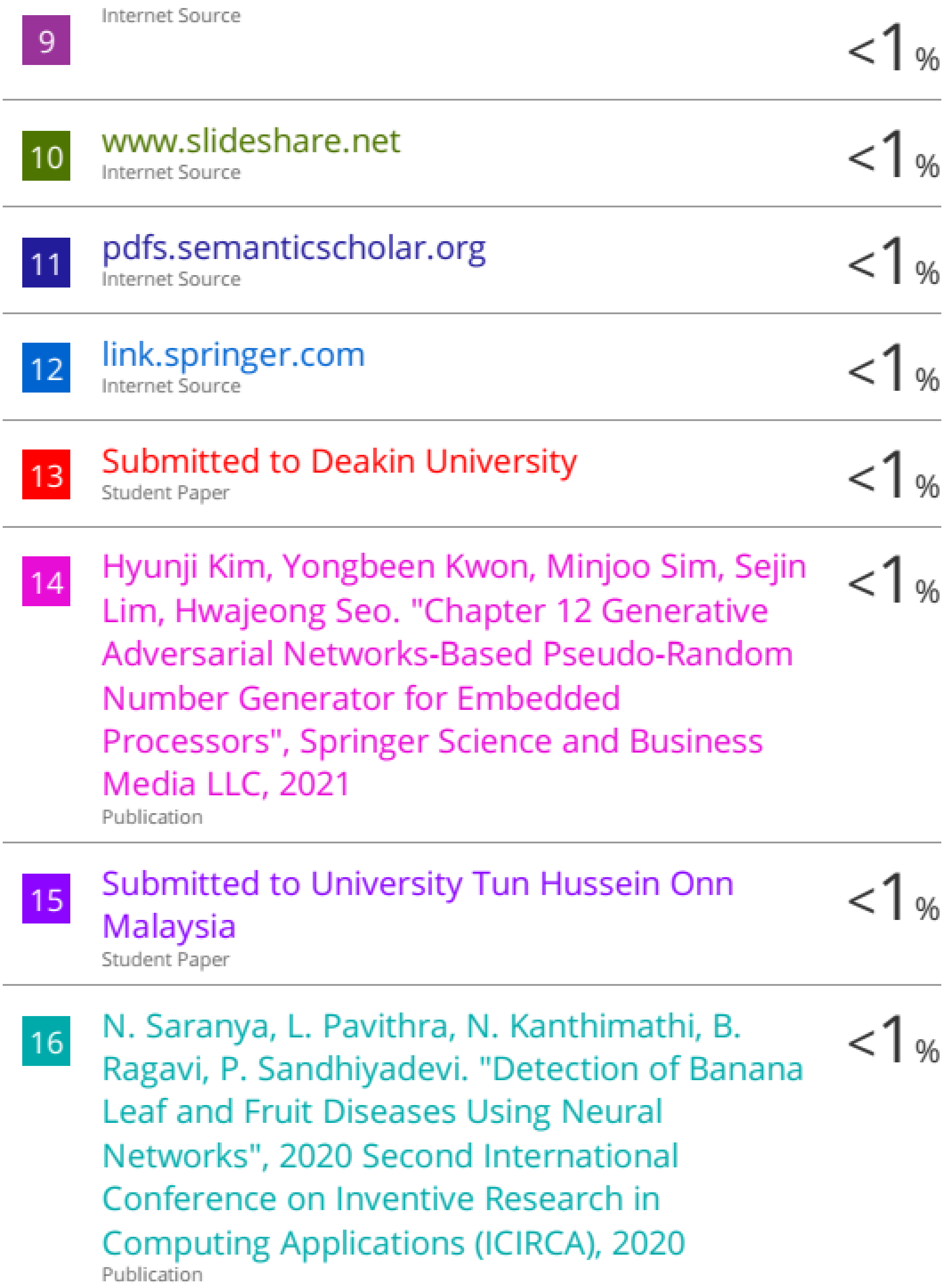
BANANA_DISEASES_DETECTION_MOBILE_APPLICATION

ORIGINALITY REPORT

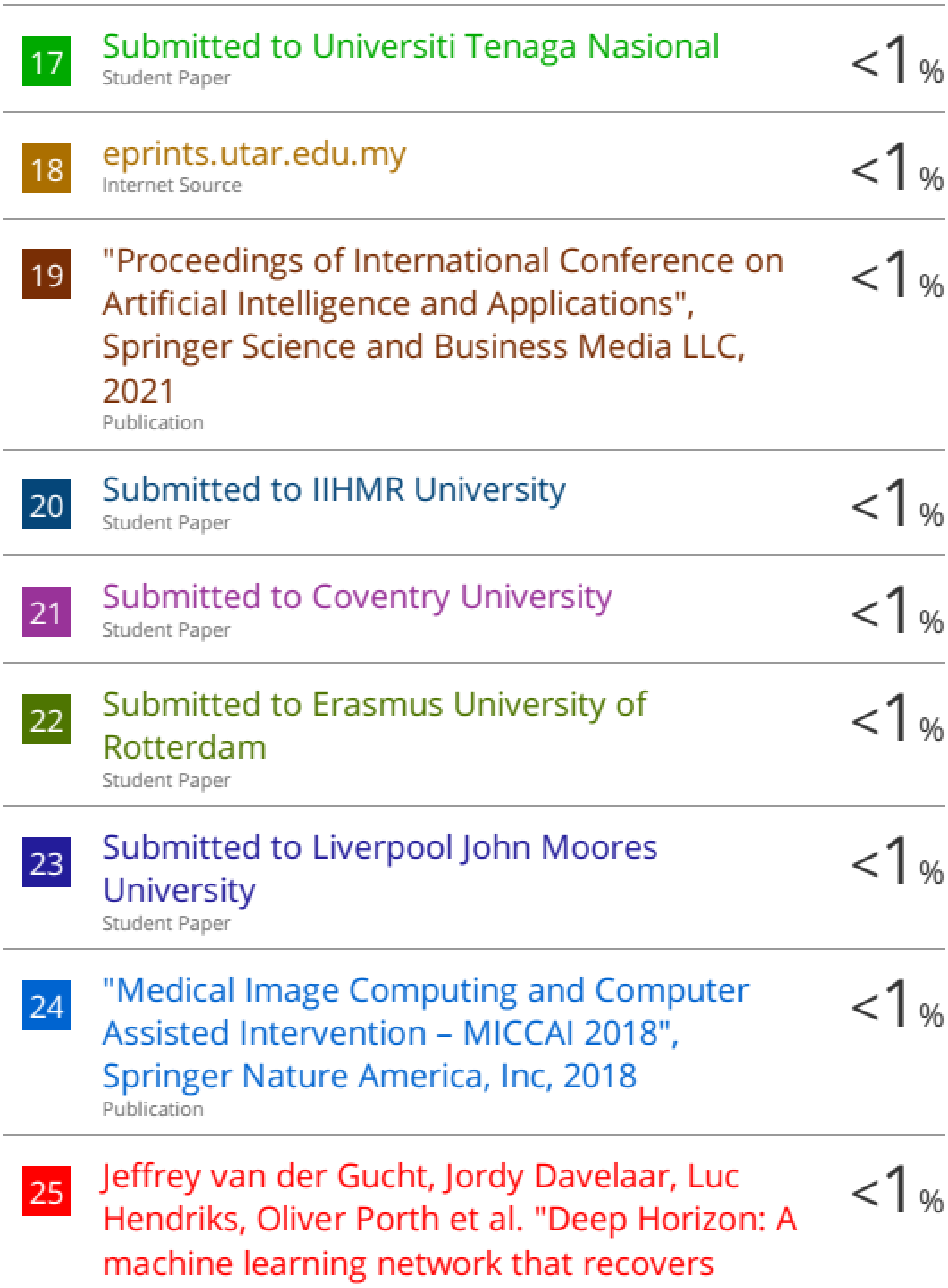
8%	6%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

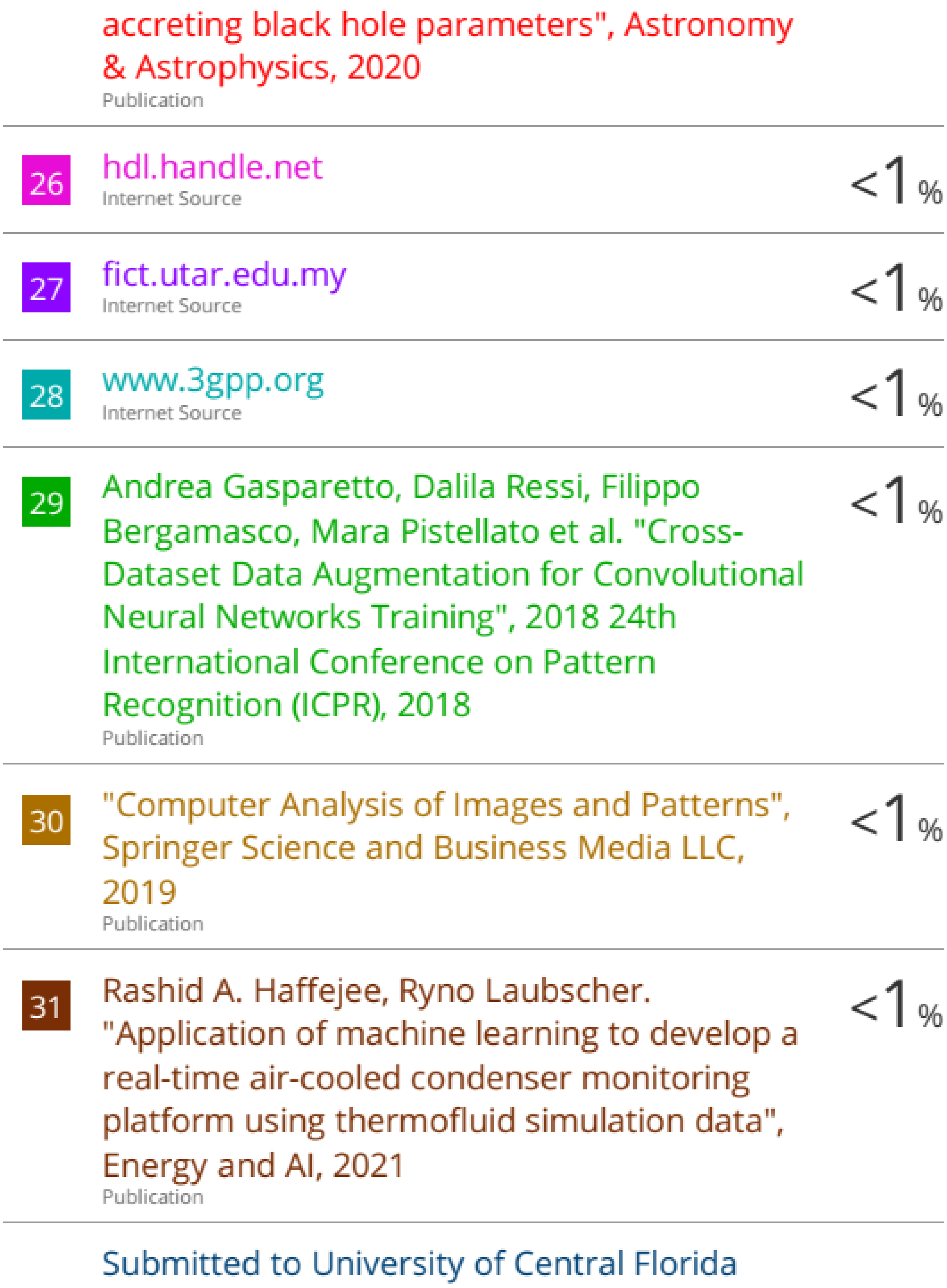
1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	plantmethods.biomedcentral.com Internet Source	1%
3	Submitted to Nottingham Trent University Student Paper	1%
4	www.simplilearn.com Internet Source	<1%
5	Submitted to American University in Dubai Student Paper	<1%
6	medium.com Internet Source	<1%
7	"Soft Computing and Signal Processing", Springer Science and Business Media LLC, 2022 Publication	<1%
8	www.arxiv-vanity.com Internet Source	<1%
	www.ibm.com	



9	Internet Source	<1 %
10	www.slideshare.net Internet Source	<1 %
11	pdfs.semanticscholar.org Internet Source	<1 %
12	link.springer.com Internet Source	<1 %
13	Submitted to Deakin University Student Paper	<1 %
14	Hyunji Kim, Yongbeen Kwon, Minjoo Sim, Sejin Lim, Hwajeong Seo. "Chapter 12 Generative Adversarial Networks-Based Pseudo-Random Number Generator for Embedded Processors", Springer Science and Business Media LLC, 2021 Publication	<1 %
15	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1 %
16	N. Saranya, L. Pavithra, N. Kanthimathi, B. Ragavi, P. Sandhiyadevi. "Detection of Banana Leaf and Fruit Diseases Using Neural Networks", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020 Publication	<1 %



17	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
18	eprints.utar.edu.my Internet Source	<1 %
19	"Proceedings of International Conference on Artificial Intelligence and Applications", Springer Science and Business Media LLC, 2021 Publication	<1 %
20	Submitted to IIHMR University Student Paper	<1 %
21	Submitted to Coventry University Student Paper	<1 %
22	Submitted to Erasmus University of Rotterdam Student Paper	<1 %
23	Submitted to Liverpool John Moores University Student Paper	<1 %
24	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2018", Springer Nature America, Inc, 2018 Publication	<1 %
25	Jeffrey van der Gucht, Jordy Davelaar, Luc Hendriks, Oliver Porth et al. "Deep Horizon: A machine learning network that recovers	<1 %



accreting black hole parameters", Astronomy & Astrophysics, 2020
Publication

26 hdl.handle.net <1%
Internet Source

27 fict.utar.edu.my <1%
Internet Source

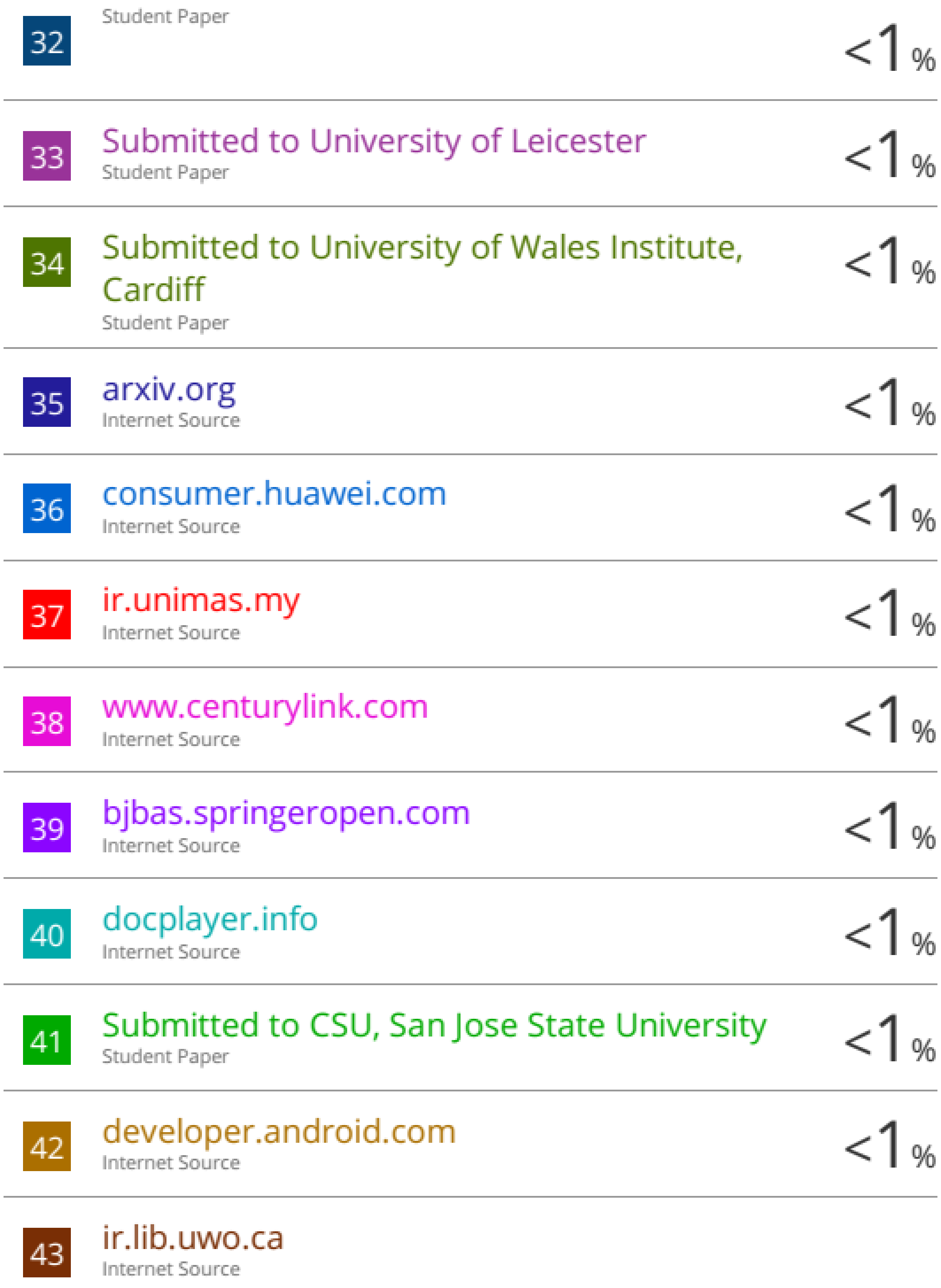
28 www.3gpp.org <1%
Internet Source

29 Andrea Gasparetto, Dalila Ressi, Filippo Bergamasco, Mara Pistellato et al. "Cross-Dataset Data Augmentation for Convolutional Neural Networks Training", 2018 24th International Conference on Pattern Recognition (ICPR), 2018 <1%
Publication

30 "Computer Analysis of Images and Patterns", Springer Science and Business Media LLC, 2019 <1%
Publication

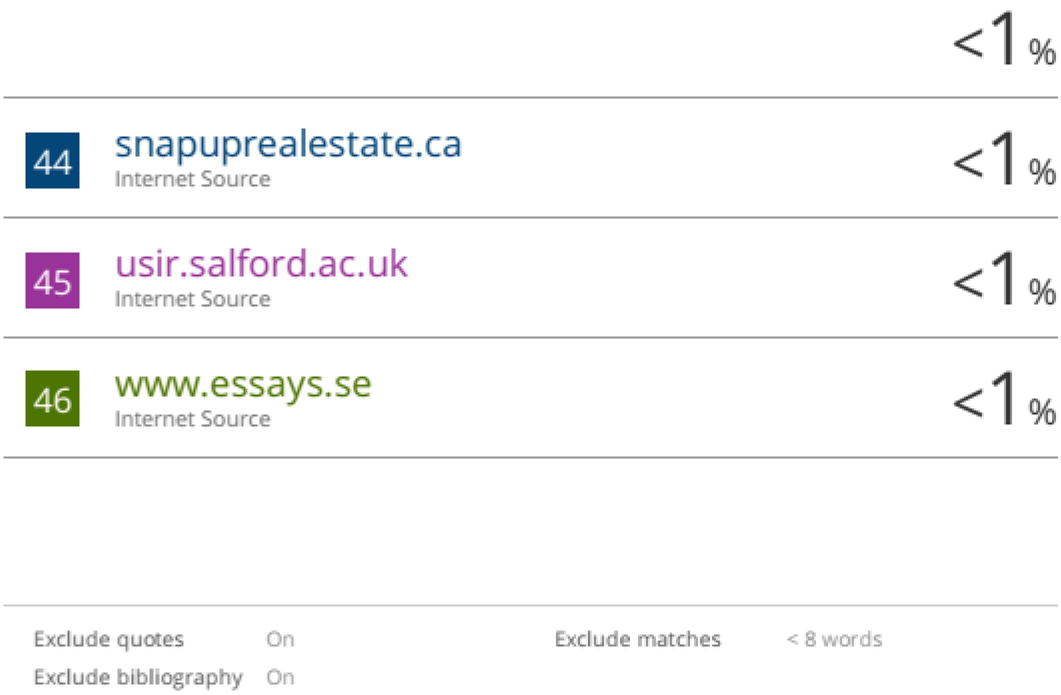
31 Rashid A. Haffejee, Ryno Laubscher. "Application of machine learning to develop a real-time air-cooled condenser monitoring platform using thermofluid simulation data", Energy and AI, 2021 <1%
Publication

Submitted to University of Central Florida



32	Student Paper	<1 %
33	Submitted to University of Leicester Student Paper	<1 %
34	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
35	arxiv.org Internet Source	<1 %
36	consumer.huawei.com Internet Source	<1 %
37	ir.unimas.my Internet Source	<1 %
38	www.centurylink.com Internet Source	<1 %
39	bjbas.springeropen.com Internet Source	<1 %
40	docplayer.info Internet Source	<1 %
41	Submitted to CSU, San Jose State University Student Paper	<1 %
42	developer.android.com Internet Source	<1 %
43	ir.lib.uwo.ca Internet Source	

PLAGIARIMS CHECK RESULT



		<1 %
44	snapuprealestate.ca Internet Source	<1 %
45	usir.salford.ac.uk Internet Source	<1 %
46	www.essays.se Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

Appendix F: PLAGIARISM CHECK RESULT

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)	LEE WAI HIN
ID Number(s)	18ACB02736
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	BANANA DISEASES DETECTION MOBILE APPLICATION

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 8 **%** **Similarity by source** Internet Sources: 6 % Publications: 2 % Student Papers: 5 %	
Number of individual sources listed of more than 3% similarity: N.A.	
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.

PLAGIARIMS CHECK RESULT

Signature of Supervisor

Name: Lai Siew Cheng

Date: 21/4/2022

Signature of Co-Supervisor

Name: ____________________________

Date: ______________________________

Appendix G: FYP2 REPORT CHECKLISTS

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB02736
Student Name	LEE WAI HIN
Supervisor Name	Ts LAI SIEW CHENG

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.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	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)

FYP 2 CHECKLISTS

√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*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.

____

(Signature of Student)

Date: 20/04/2022