MOBILE APPLICATION FOR FRUITS FRESHNESS/RIPENESS DETECTION

By

Cheah Ui Shaun

Supervised By
Dr. Ng Hui Fuang

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)

JANUARY 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Mobile Application for Fruit Freshness/Ripeness Detection

Academic Session: 01/2021

I CHEAH UI SHAUN

(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,

Shaun

[illegible]

(Author's signature)	(Supervisor's signature)
Address:	
30, Lembah Taman Bukit,	
Taman Bukit, 14000 Bukit	Ng Hui Fuang
Mertajam, Penang, Malaysia	Supervisor's name
Date: 15.4.2021	**Date**: 16-04-2021

MOBILE APPLICATION FOR FRUITS FRESHNESS/RIPENESS DETECTION

By

Cheah Ui Shaun

Supervised By
Dr. Ng Hui Fuang

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)

JANUARY 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled **Mobile Application for Fruits Freshness/Ripeness Detection** 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	:	Shaun
Name	:	Cheah Ui Shaun
Date	:	15-04-2021

ABSTRACT

Fruits play a crucial role in our diet as they contain nutrients that are vital for our health. These nutrients are important as they help us protect against chronic diseases. Fruits that are not fresh will not contain as many nutrients than when it was fresh. Thus, it is important to ensure that only fresh and ripe fruits are consumed. However, there exist a large number of consumers that do not know how to select fruits that are fresh when purchasing. Besides that, the fruit industry in Malaysia uses harmful chemicals in order to perform fruit inspections which makes the fruit to lose its nutrients.

To solve the problems stated above, this paper proposes a mobile application that can detect freshness in fruits. To do this, this project utilizes Deep Learning technologies in conjunction with a mobile application in order to predict the freshness of fruits. Although there are several applications that can perform fruit freshness prediction, they require the user to have several external devices in order to accurately predict its freshness. Therefore, this project will focus on developing an application that can do fruits freshness prediction accurately without needing extra devices.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to those who have helped in completing this project. A special thanks to my supervisor, Dr. Ng Hui Fuang, for his guidance and insights throughout the lifetime of this project. It is my pleasure to carry out my project under his guidance.

Not forgetting to mention my course mates, for their valuable feedback and time in helping me test my mobile application.

Last but not least, I would like to thank my family for their love and support through my studies

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A.I	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
ELU	Exponential Linear Unit
ER	Entity Relationship
FRESHION	Fruits Freshness Detection Application
GAP	Global Average Pool
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HSV	Hue, Saturation, Value
IDE	Integrated Development Environment
JDK	Java Development Kit
JVM	Java Virtual Machine
MSE	Mean Squared Error
QC	Quality Control
UI	User Interface
ReLU	Rectifier Linear Unit
RGB	Red, Blue, Green
SDK	Software Development Kit
UML	Unified Modelling Language

Chapter 1 INTRODUCTION

1.1 Problem Statement and Motivation

1.1.1 Problem Statement

Learning how to detect freshness in fruits before purchasing can be a problem to people who are new to shopping for groceries. To know how to select fresh fruits to purchase from the heap of fruits in markets, it would require a lot of experience to do so. While there are many people that already have the experience of picking fresh fruits from markets, there are still some inexperience people who are still new to shopping for fruits. That is why, it would be great to have an application that can accurately detect the freshness from fruits before they are purchased. This will greatly benefit the group of inexperienced consumers who do not know how to pick fresh fruits from markets. Besides that, it will also greatly **reduce the number of people that get sick after eating rotten fruits**. Since there are cases in which people get sick from eating spoiled or rotten fruits due to not knowing how to pick the fruit that is fresh and edible.

Furthermore, not every consumer has the knowledge on how to select all kinds of fruits that are fresh. Even an **experienced grocery shopper** that happened to be interested in eating a new fruit can **have trouble knowing how to pick the right fruit** that is actually fresh. This is because not everyone has the experience of selecting fresh fruits from every type of fruits in this world.

Currently, **harmful chemicals and substances are used to determine the freshness of fruits in the agriculture industry.** (Ibrahim et al., 2016) states that, to determine the ripeness of a banana, a few drops of harmful chemical are required which will damage the fruit. By damaging the fruits during the freshness determination process, it will result in a huge lost for the agriculture industry. This loss can be easily be avoided by selecting a non-destructive method to determine the ripeness/freshness of these fruits. An example a non-destructive method is to have **workers to inspect each of the fruits to determine its freshness** according to its colour and texture. However, this method is not practical as it is **time consuming, subjective and inconsistent** (Bhargava and Bansal, 2018). To increase the efficiency of quality checking in the fruit agriculture industry, an automation technique should be used.

1.1.2 Motivation

In this modern world, we humans have automated many types of process in our daily lives. Examples of automation technologies in our daily life would be toaster, water heater, microwave and many more. What if we could apply that to when we are grocery shopping for fruits as well?

Now, with the existence of object classification technology through deep learning methods, we are able to detect the freshness of fruits with ease. In addition to that, if we can use deep learning techniques in conjunction with mobile application development technology, we can make it so that anyone with the application can detect freshness of fruits whenever and wherever they want.

Therefore, I would like to develop a mobile application for fruits freshness detection which will be named as FRESHION which is an abbreviation of Fruits Freshness Detection. With FRESHION, it will be able to provide its users with an efficient way for fruits freshness detection on a mobile device with Android OS. To determine the freshness of fruits, deep learning technology will be used. To make this technology portable, the deep learning model that is trained will be integrated into the mobile application.

1.2 Project Scope

To solve the problems stated in 1.1, the final product of this project will be a mobile application that users can use to do fruit quality inspections. In this application, users will be able to upload/take a photo of the fruit to find out the freshness quality of the fruit. Currently, the planned platform for the mobile application to be developed will be Android as there are more people from around the world that uses android phones. Besides that, only 3 types of fruits inspection are planned to be included in the final product of this project. These fruits are banana, mango and papaya. The reason for choosing these 3 fruits is because they are among the most common types of fruits that are eaten by consumers around in Malaysia.

1.3 Project Objectives

The main objective of this project is to reduce the number of consumers that are getting sick from eating fruits that are not fresh. Furthermore, the final product of the project is to help consumers that do not know how to select fruits that are fresh. Besides helping consumers, this project also aims to help in providing quality inspections of fruits in the agriculture industry around the world. Other than that, with the final product of the project, the agriculture industry would not have to rely on destructive methods as stated in 1.1 to perform fruit quality inspection.

In short, the objectives are as follows:

I. Build a machine learning model that is able to classify freshness of 3 types of fruits (banana, mango and papaya).

II. To develop a mobile application that can utilize the machine learning model built in (I) to classify freshness of fruit based on fruit image captured by user.

1.4 Impact, Significance and Contribution

The mobile application created through the outcome of this project will be able to detect the freshness in fruits so that consumers that lack the expertise or experience can pick fruits that are fresh when shopping for groceries. Incidentally, this will reduce the number of people that will get sick from eating fruits that are not fresh. Besides that, with this application, it will provide the agriculture industry with a non-destructive and efficient method for fruits freshness determination.

Next, this mobile application can be considered as an improvement in the method for fruits freshness detection that the agriculture industry currently uses. Currently, the agriculture industry uses harmful chemicals to determine the freshness of fruits whereby with this mobile application, it uses machine learning methods to achieve the same result. Furthermore, with this mobile application, fruits inspectors are able to inspect fruits more efficiently as it only requires very little human inputs.

1.5 Background Information

Fruits are one of the important foods in our diet as it contains many nutrients such as dietary fibre, vitamins, minerals and electrolytes. These nutrients will help reduce the risk of contracting many non-communicable diseases (Pem & Jeewon, 2015). However, fruits that are spoiled often has lost its nutrients. Therefore, it is important to ensure that the fruit is fresh before consuming them to ensure that essential nutrients are consumed as well.

Currently, the fruit industry uses a destructive method for detecting the quality freshness of fruits. According to (Ibrahim et al., 2016), a sample of the flesh of the banana is required to determine its freshness quality based on the juice contents inside. However, this method of quality checking will cause destruction on the fruit and is not a very practical way of determining the quality of the fruit. Recently, image processing and machine learning have been proposed as a non-destructive method of fruit quality assessment in the fruit industry (Monovar, 2011; Rizam, 2009).

Deep learning is part of the field of A.I. Deep learning is a function that tries to emulate how the human brain work in processing data by creating patterns to use in decision making (Brownlee, 2019). Basically, deep learning teaches machines how to solve problems by learning from example. Image processing is the study of processing through images to extract data for further analysis (Silva & Mendonça, 2015). By using image processing in conjunction with deep learning, we are able to automate the process for fruits freshness detection. By automating the quality inspection process for fruits, we are able to reduce human error and greatly increase the productivity of the fruit quality inspectors.

Currently, there are only a few mobile applications that can perform quality inspection to determine freshness/ripeness of fruits. These applications are able to grade the freshness of fruits by having the user to capture photos of the fruit. These captured photos will then be analysed in the cloud by using computer vision algorithm. After that, the application will grade the captured fruit according to attributes that are acquired by the algorithm like colour and size.

1.6 Proposed Approach

The project can be divided into two different sections, which are developing the deep learning model to classify images according to their ripeness/freshness and also integrating the model into a mobile application platform to use the model.

Developing Deep Learning Model for Fruits Freshness/Ripeness Classification

Recently, CNN has been used widely with image processing with the purpose to solve classification problems due to the great improvements in recent years (Xin & Wang, 2019). Therefore, CNN is selected as our approach to perform fruits freshness/ripeness classification. CNN, also known as Convolution Neural Network, is basically a network with convolution layers that contain a specific number of neurons in order to retrieve features that will be used to classify the fruits at the end of the network. In between each convolution layer, there will be a pooling layer that reduces the resolution of the image before passing on to the next convolution layer, this is to reduce the number of parameters and the computation required to be performed. At the end of the network, the fully connected layer will be used to classify the input image according to the number of classes specified.

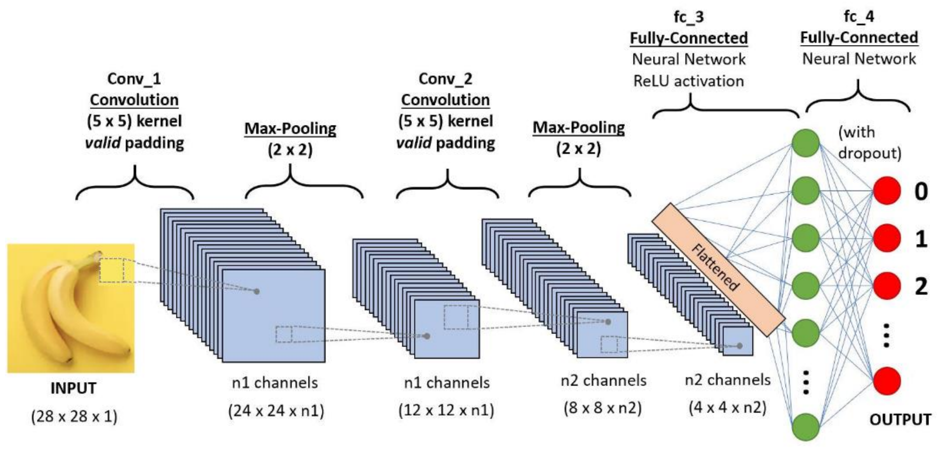


Figure 1.1: Convolutional Neural Network Architecture

Integrating the trained model & developing the mobile application

The model is trained in TensorFlow Keras libraries, we will need to convert it to a TensorFlow Lite file in order to use it in the mobile platform. After converting it using TFLiteConverter, a function imported from TensorFlow library, the model will then be loaded into mobile platform using Android Studio’s TensorFlowLite library. After importing the model that we have converted earlier, an interface is created so that the user can interact with the system to perform fruit inspections to know the freshness of fruit.

1.7 Highlight of achievement

- Able to classify 3 types of fruits (mango, banana and papaya) according to their ripeness using a CNN Model.
- Able to classify whether if the fruit captured/given by the user is one of the 3 fruits that can be classified according to their ripeness using a CNN Model.
- Developed a mobile application that is able to perform freshness inspection using the deep learning models developed stated above.

1.8 Report Organization

Chapter	Description
Chapter 1: Introduction	• Define the problem statement, motivation, scope, objectives and background of the project. • State the impact and contribution of the project. • Explain the proposed approach to the solution.
Chapter 2: Literature Review	• Review of existing applications that are similar to this project. • Review of the existing papers exploring in the use of different technologies to solve the problems stated in Chapter 1
Chapter 3: System Design	• Contains the architecture diagrams • Shows the application layout of the final product • Shows the timeline for the entire project
Chapter 4: Methodology	• States the methodology used to develop the system • States the programming tools and hardware used to complete the project • States the user requirement of the system • Shows the specification of the system
Chapter 5: Implementation & Testing	• Shows the two different implementations attempted during the lifetime of the project • Shows the test cases for different types of fruit inspection
Chapter 6: Conclusion	• Review of the entire project • Novelties and contribution of the project • Future work of the project

Chapter 2 LITERATURE REVIEW

2.1 Review of Existing System

There are only a few similar applications that can perform freshness/ripeness detection in fruits that are available in the internet whereby two of them are taken for review.

2.1.1 ClariFruit

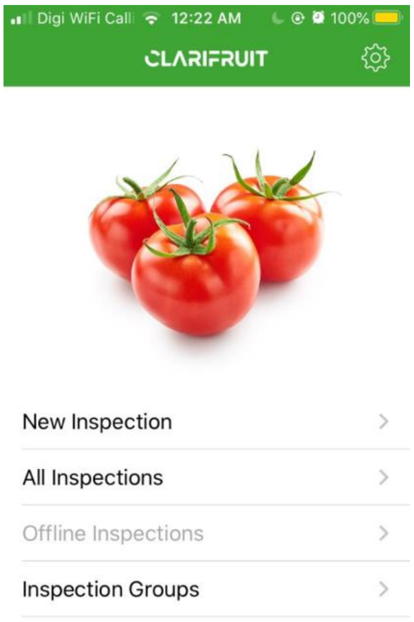
This application started developing in Israel in the year of 2018. ClariFruit is still currently in its pre-release beta version. However, interested users are able to request to join their beta program where you will be given an account to be able to access their application. This application can be downloaded from both the AppStore and PlayStore from android and iOS platforms respectively. The purpose of this application is to perform quality control and provide users with a data analytic platform for fresh produce.

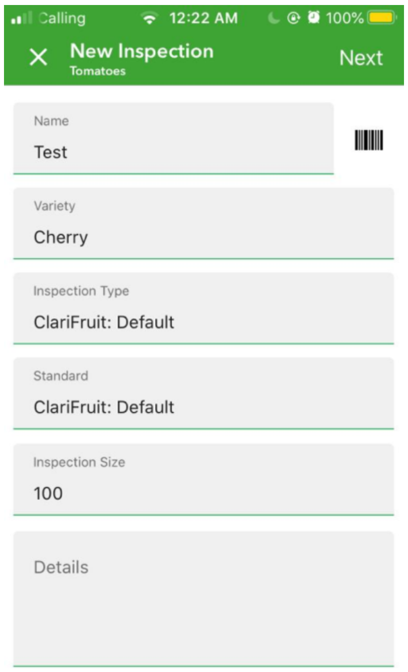
In the settings page, users are able to pair with several external devices such as a pocket spectrometer and refractometer. The pocket spectrometer and refractometer are devices that are used to estimate metrics to measure fruit quality. Examples of these metrics are dry matter, Brix value, acidity and colour. By pairing with these devices, users are able to use these devices to read internal attributes of the fruit to retrieve the metrics as mention before to enter those data automatically. However, if a user does not have those devices, they can simply just enter the data manually. Currently, this application is only able to detect freshness of tomato as it is still in the beta stage.

To start detecting freshness of tomato, users will have to select “New Inspection” in the homepage of the application. It is here that users are able to select the variety type of tomato to be inspected, standard of inspection (United States Department of Agriculture standard / ClariFruit standard), and inspection size (based on inspection standard of QC). After that, users can fill up the defect (i.e. colour, skin, freezing damage and etc) of the tomato by either selecting numbers from 1 to 5 or take a photo of the tomato to have it filled automatically. Next, users can choose to select the firmness group that the tomatoes are in. Furthermore, users are asked to enter the Brix value of the tomatoes which is the sugar content of an aqueous solution. However, if the user has a pocket spectrometer or a refractometer connected, they can use those devices to automatically key in the value instead.

Finally, users are asked to take at least 5 photos of the tomatoes to produce the final result. The final result page shows the user the grade of the tomato (from A to F), diameter, stem colour, size group and many more. Other than that, a copy of this result will be sent to the user's registered email for ease of access. The user can view the previous results in the history page by clicking at the "All Inspections" icon at the homepage.

Overall, this application is very detailed in the result of inspection of tomatoes. It shows the user the factors that affects the final result and grading of the inspection. Besides that, it also contains special features such as being able to pair to different external devices to key in value instantly after using them. However, the bad thing about this application is that it is too complex to use. There are no guidelines to teach users on how to use the application. However, the application makes up for its weakness by having YouTube videos that teach users on how to use their application.





Name
Test

Variety
Cherry

Inspection Type
ClariFruit: Default

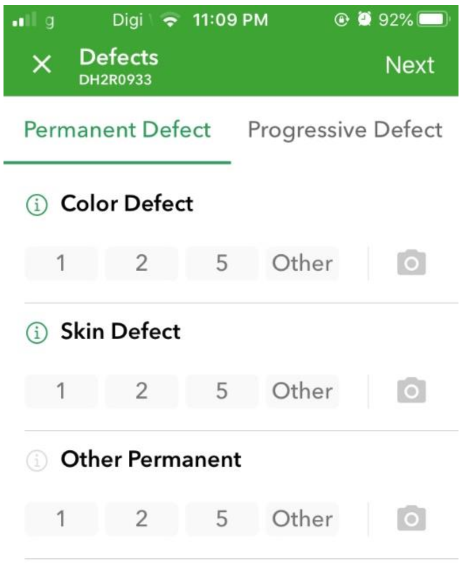
Standard
ClariFruit: Default

Inspection Size
100

Details

Figure 2.1.1: ClariFruit Homepage

Figure 2.1.2: New inspection page



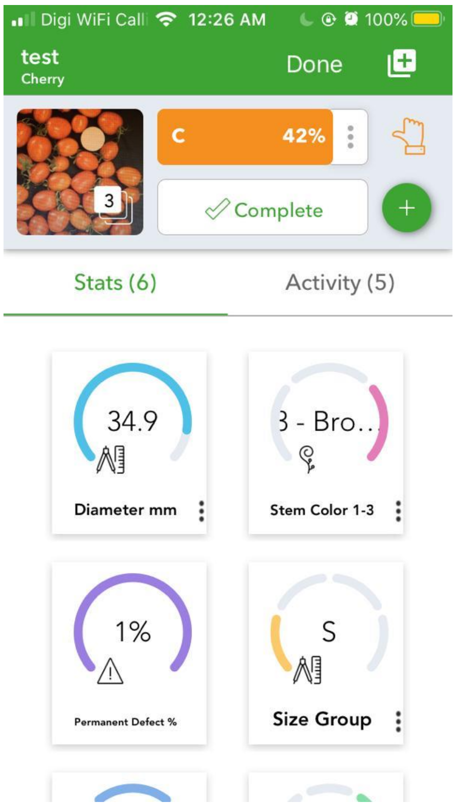


Figure 2.1.3: Form for Defect of fruit

Figure 2.1.4: Result of Inspection

2.1.2 Fruit & Veg Detector

This application is developed by “Dragster Production” in the google Play Store. The application is developed in Pakistan and was last updated in August 4, 2020. There are over five thousand downloads of this application and have a review of 4.6 stars. This application is very simple to use, whereby you just need to point your camera at the fruit that you are trying to detect to obtain the result instantly.

In the homepage of this application, users are able to choose to detect freshness of fruit or vegetable. After that they are then asked to allow the application’s access to the phone camera. Furthermore, users can then choose to analyse the fruit through two modes: detection mode and feature mode. Detection mode just uses your normal camera without adding any special filters to retrieve features of the fruit. Any special features detected will be circled yellow with a white dot. Other than that, in feature mode, the application is also able to detect features of the fruit by showing them as white lines on the photo. After detecting the features, the application will show how fresh the fruit is by showing the percentage for freshness on the screen.

Overall, this application is too simple and lack interesting features (i.e. view the factors that affect the result of the inspection). The accuracy of the inspection is questionable as sometimes it will show result as fresh even though the object that is under inspection is not a fruit/vegetable as shown in Figure 2.1.9. The strength of this application is that it is very straightforward and is very easy to use.

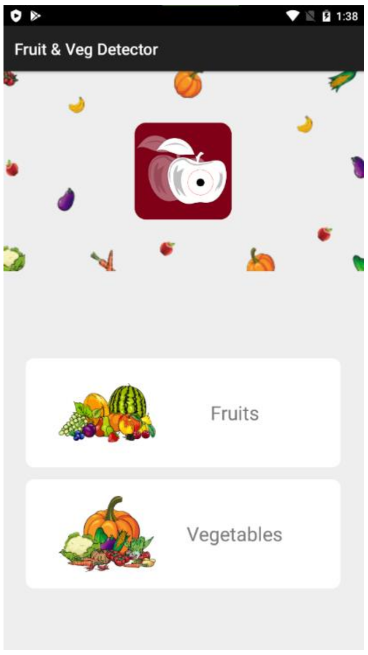
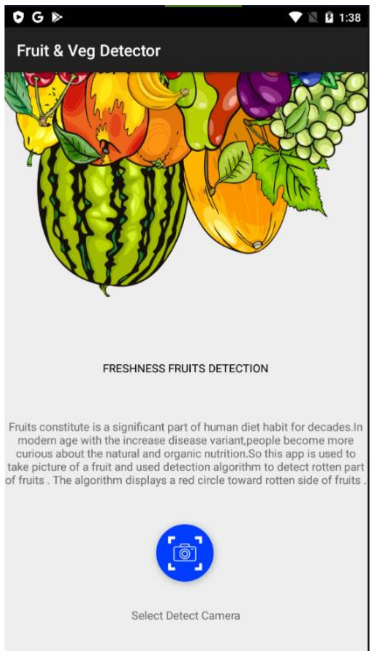


Figure 2.1.5: Homepage of application



FRESHNESS FRUITS DETECTION

Fruits constitute is a significant part of human diet habit for decades.In modern age with the increase disease variant,people become more curious about the natural and organic nutrition.So this app is used to take picture of a fruit and used detection algorithm to detect rotten part of fruits . The algorithm displays a red circle toward rotten side of fruits .

Figure 2.1.6: Select Camera page

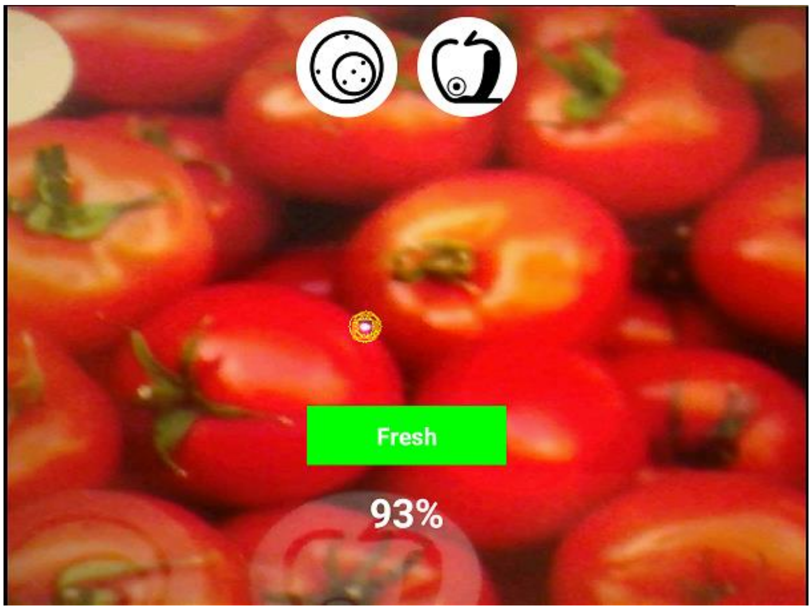


Figure 2.1.7: Detection Mode

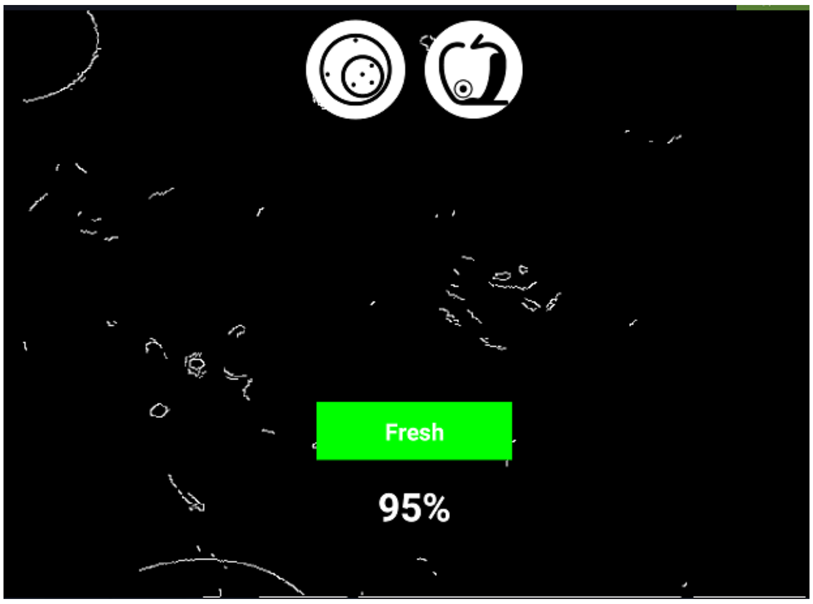


Figure 2.1.8: Feature Mode

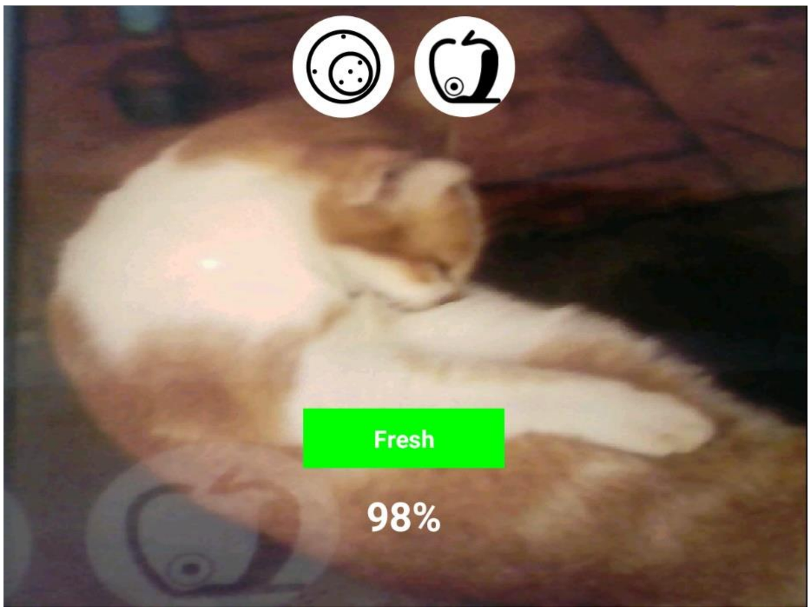


Figure 2.1.9: Questionable Accuracy Example

2.2 Comparison Analysis

2.2.1 Requirements

Application / Requirements	ClariFruit	Fruits & Veg Detector	Proposed Application (FRESHION)
Camera	Yes	Yes	Yes
Internet Access	Partially	No	No
Bluetooth	Yes	No	No
Simplicity	Moderate	Easy	Easy
Hardware Requirement	Low	Low	Low

Table 2.2.1: Requirements Comparison Analysis

2.2.2 Features

Application / Features	ClariFruit	Fruits & Veg Detector	Proposed Application (FRESHION)
Compares with previous inspections	Yes	No	Yes
Freshness Grading	Yes	No	Yes
Display factors that affect inspection result	Yes	No	Yes
Predict how long fruit lasts	No	No	Yes

Table 2.2.2: Feature Comparison Analysis

2.3 Review of other Fruit Quality detection methods

2.3.1 Machine Learning Methods

1. Recognizing the Ripeness of Bananas using Artificial Neural Network based on Histogram Approach (Saad, H., Ismail, A., Othmand, N., Jusoh, M., Naim, N. and Ahmad, N.)

First of all, the project starts by collecting 3 sets of different bananas (unripe, ripe and overripe). These images of the bananas are captured using a webcam and are then resized to 352x288. The reason for doing so is due to the original image being too big. By having a large image, it will take a lot longer for the GPU to process the images. After that, the project moves into the RGB component extraction stage to determine the colour intensity of the image. A histogram is then plotted based on the RGB value as shown in Figure 2.3.2 below.

To train the ANN model, the project uses a propagation that involves in 3 stages (Feed-forward of input training pattern, Back-propagation of error and Weight adjustment). When training the model, the output is compared with target value to calculate appropriate error. The error factor will then be distributed back to the hidden layer. After network is trained, the application will only involve with the feed-forward phase. To minimize the squared error of the output, the gradient descendent method will adjust its weight accordingly. Finally, a GUI is created so that the user is able to use the ANN model to detect ripeness of banana.

Although the ANN model created was able to classify 25 correct samples out of 28 samples, there are still several ways that can be done to increase the effectiveness of the system. First of all, the camera used previously to capture the banana images cannot capture high resolution images. So, to fix this we can use a high pixel camera that is able to capture high quality images to increase accuracy of model. Besides that, the images used to train the model contains background. The background contains unnecessary components that will reduce the accuracy of the model produced. So, the solution is to remove it. However, another solution is to capture the images of the banana in a black background so that it won’t affect the RGB values captured. Finally, to further increase the accuracy of the model, the article suggested to increase the number of colour intensity group (currently the article only uses 3).

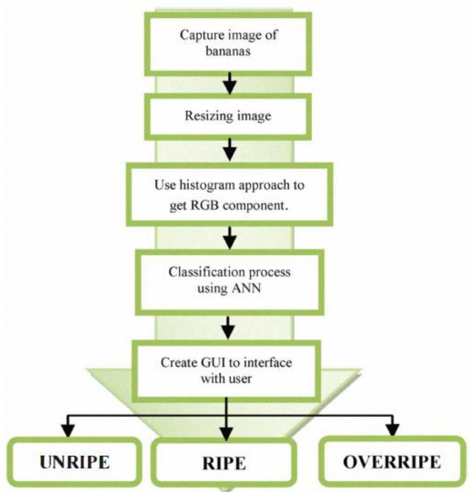

Capture image of bananas
Resizing image
Use histogram approach to get RGB component.
Classification process using ANN
Create GUI to interface with user
UNRIPE
RIPE
OVERRIPE

Figure 2.3.1: Flow chart of project development

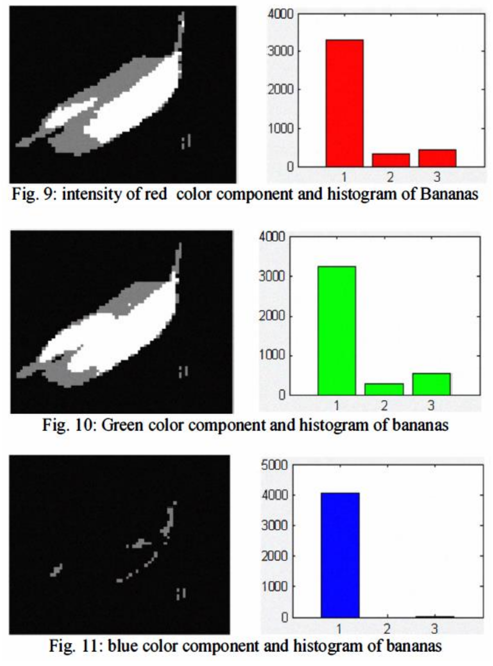


Figure 2.3.2: Histogram plot of the RGB intensity of banana image

2. Ripeness Classification of Bananas using an ANN (Mazen, F. and Nashat, A.)

The proposed method for doing ripeness/freshness detection for this article is by using an ANN. Before training the ANN model, the article begins with the data collection process. The dataset used consists of 300 bananas with varying ripening levels (unripe, ripe and overripe). The images of bananas in the dataset are captured using a Samsung Note 3 camera. Next, the images are then pre-processed before training and testing the ANN. After that, an HSV model is used to describe the RGB colours of the banana images. Furthermore, morphological filtering is done on the banana images to enhance them. In addition to that, Otsu's method is used to remove the background of the banana image to increase the accuracy of the model.

To determine the ripeness of fruit, it will calculate the ripeness factor of the banana fruit based on the number of brown spots on the image. The formula to calculate the ripeness factor is shown in Figure 2.3.4 below. Besides the colour analysis method, the article also used statistical texture analysis to determine the ripeness of the fruit. To train the ANN, the Levenberg-Marquardt backpropagation algorithm is used. The input and output layer of the network consists of 4 neurons. Besides that, the model also contains 10 hidden layers in the middle. The block diagram of the proposed ANN is as shown in Figure 2.3.5 below. The partitioning of the dataset to train and test the data is 70:30. Whereby 70% of the data will be used for training and 30% of the data will be used for testing. The reason for this split is so that we are able to calculate the accuracy and precision of the model to ensure that it actually works.

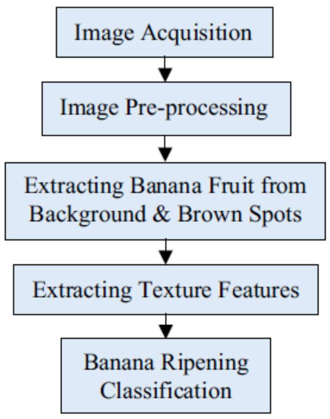


Image Acquisition
Image Pre-processing
Extracting Banana Fruit from Background & Brown Spots
Extracting Texture Features
Banana Ripening Classification

Figure 2.3.3: Steps for the proposed algorithm

$$\text{Banana's Ripeness Factor (RF)} = \frac{\text{Total Area of the Brown Spots}}{\text{Total Area of the Banana}},$$

Figure 2.3.4: Banana Ripeness Factor formula

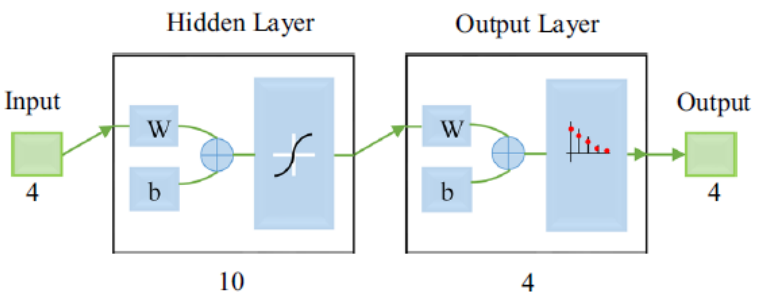


Figure 2.3.5: Architecture of the proposed ANN

<u>Conclusion</u>

After reviewing the journal articles implementing Machine Learning techniques, we can conclude that the articles above supports using ANN to perform classification for fruit quality. This is because ANN can recognise the similar patterns contained in the input images in order to classify them.

ANN has a network of neurons with different layers that are used to simulate how the human brain works as shown in Figure 2.3.6 below. These layers of neurons are used to extract a certain feature/attribute from the input data. Similar to how the human brain processes information before performing certain actions (i.e. Hand moving away from boiling pot). ANN works similar as well, where input data is processed in each layer to obtain elements of the data and then finally produce an output based on the elements obtained from the input data.

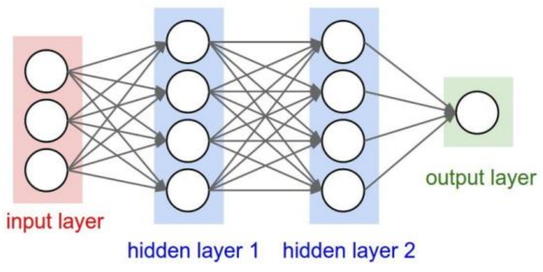


Figure 2.3.6: ANN model example

2.3.1 Deep Learning Methods

1. On line detection defective apples using computer vision system combined with deep learning methods (Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., Zhang, C. and Huang, W.)

Based on a journal article written by Fan et al. (2020), proposed a method of using a 4-lane sorting system to grade the quality of apples. In this 4-lane sorting system, a conveyer will be utilised to move and rotate fruits freely to ensure that the entire surface of the fruit is captured as much as possible by two cameras. Basically, the image of each fruit will be captured in 3 different orientation by having them rotate around for each image capture. This is so that we are able to capture all of the “faces” of the fruit. Besides that, two linear lights are used to ensure that the image captured by the cameras are bright enough. To increase accuracy by blocking out stray light, all of the components stated previously are placed inside a light chamber.

After the images are acquired, an image processing method called Otsu’s method is used to remove the background of the image captured. Without removing the background, it may cause unnecessary noise that will reduce the accuracy of the CNN model. In addition to that, the images are then resized to 80 by 80 pixel by using the resize function in OpenCV.

After the image pre-processing methods, a CNN layer is used to extract features of images. At each layer, a feature map is produced as an output which contains key representations of features extracted from the images. Next, to speed up the training of deep network, a batch normalization layer will be used to act as a regularizer. Furthermore, ELU is used to accelerate learning in deep neural network by pushing mean unit activations closer to zero. After that, a pooling function will be used to modify output of results (from low-level feature representation to high-level feature representation). By doing this, we are able to increase efficiency by extracting the apparent defective regions from an image. Other than that, GAP is then later used to reduce the overfitting in neural networks by generating a feature map for each category of classification task. Finally, a Softmax function is used to produce multiple outputs to obtain the final results for classification of the freshness of fruit.

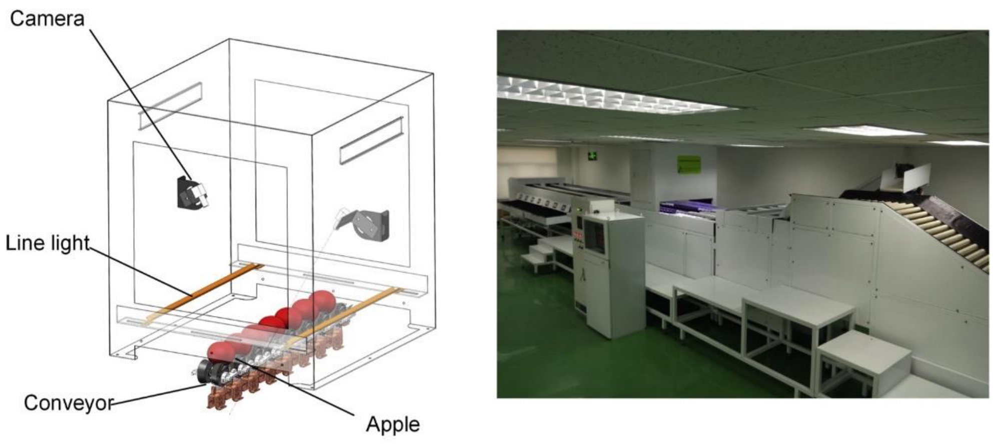


Figure 2.3.7: Schematic of Light Chamber & 4-lane fruit sorting machine

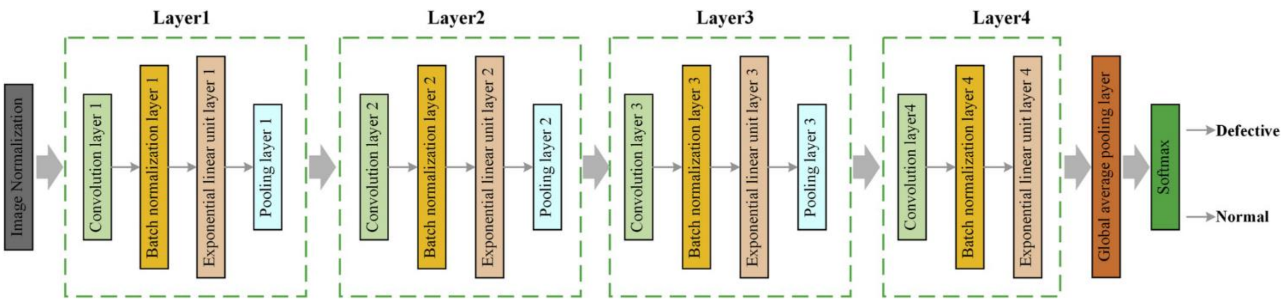

Layer1
Layer2
Layer3
Layer4
Image Normalization
Convolution layer 1
Batch normalization layer 1
Exponential linear unit layer 1
Pooling layer 1
Convolution layer 2
Batch normalization layer 2
Exponential linear unit layer 2
Pooling layer 2
Convolution layer 3
Batch normalization layer 3
Exponential linear unit layer 3
Pooling layer 3
Convolution layer4
Batch normalization layer 4
Exponential linear unit layer 4
Global average pooling layer
Softmax
Defective
Normal

Figure 2.3.8: Architecture of proposed CNN

2. An efficient approach to Fruit Classification and Grading using Deep CNN (Pande, A., Munot, M., Sreeemathy, R. and Bakare, R.)

According to the journal article written by Pande et al. (2019), a similar method is proposed to do fruits quality grading. The method is an automated system to grade fruits by placing them on a conveyor belt to help transport the fruit while a camera captures images of them. After that, the images captured will be sent to Raspberry PI to pre-process them for the DCNN classifier to run on GPU.

During the pre-processing stage, the background of the images is removed to cancel out the “noise” that contains unwanted features in the images. To do this, the s-plane of image will be multiplied with the original image to retrieve the region of the image that contains the fruit. Without removing these “noise”, the CNN model might learn the irrelevant features contained in the background thus making it less accurate.

This article states that Inception V3 (architecture of CNN) is used to classify fruits and their grading. Besides that, transfer learning is implemented to reduce the computing power and the training data required to do classification. Transfer learning is able to do this by reusing a trained model from another related task on a new model. To train the network, the dataset is divided in a ratio of 60:20:20 whereby 60 of the images are used to train the network; 20 will be used to test the accuracy of network and the unused 20 will be used to obtain the final accuracy of network.

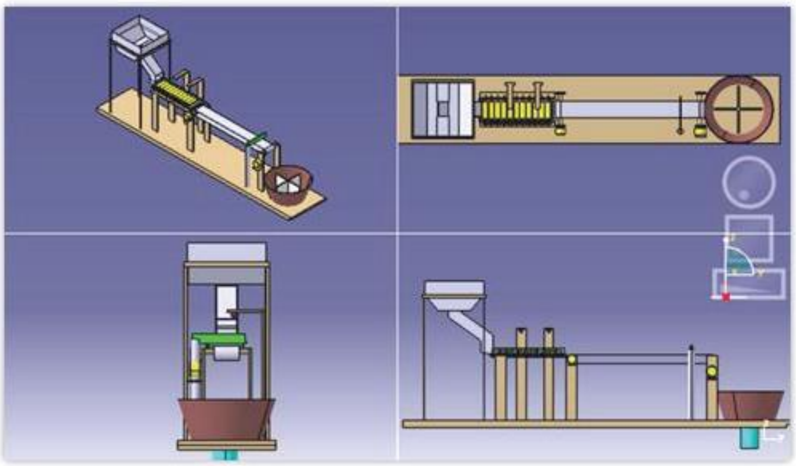


Figure 2.3.9: Conveyor System prototype design

3. Fruit Freshness Detection using CNN Approach (Harsh, A., Jha, K., Srivastava, S., Raj, A. and S, R.)

Harsh, A. et al. (2020) proposed using CNN in order to detect freshness of fruits. CNN is a type of ANN that is specialised in image analysis. CNN is utilized so that the spatial relationship between pixels can be saved. These pixels have mathematical operators that are stacked on top of each other in order to generate the layers in the network.

In this article, the authors used a two-track deep neural network as its architecture. This architecture contains two layers whereby the first layer contains deep learning algorithm and the second layer contains connected layers to do classification. This neural network has 4 layers of hidden neurons.

There are four main operations in the CNN which are Convolution, Non-Linearity (ReLU), Pooling & Sub Sampling and Classification. Convolution is an instance of the CNN that is used to extract useful features from an input image. Besides that, the reason for using ReLU is to insert non-linearity in the CNN to get it used to real world data. Furthermore, the max-pooling layers is able to classify images from the features obtained from convolution layer.

There are several real-world challenges that the authors have considered when collecting dataset. These challenges are different lighting condition, number of fruits, multiple kind of fruits and different quality in the images. To train the network, the dataset used consists of 12 different categories with 4 different fruits. Other than that, the dataset contains ripe, unripe and overripe fruits. In addition to that, the images in the dataset are taken in different conditions (i.e. in a dark/bright room). With these properties, the authors are able to increase the irregularity of the of dataset to get the network used to a more practical scenario.

Conclusion

After reviewing the journal articles that are using Deep Learning methods, it is found that all of the literature reviewed above supports using CNN for fruit freshness classification. The reason being CNN is the most suitable model for image classification due to its high accuracy. CNN consists of 4 main operations as shown in Figure 2.3.10 below.

1. Convolution

Convolution is an instance in CNN that extracts the features/attributes from the input image.

2. Non-Linearity (ReLU)

ReLU is used to insert non-linear data so that the CNN model is able to get used to real world data.

3. Pooling & Sub Sampling

The pooling layers are used to extract high level features from the input image.

4. Classification (Fully Connected Layer)

The Fully Connected Layer is used to classify the images according to the features extracted from the previous layers.

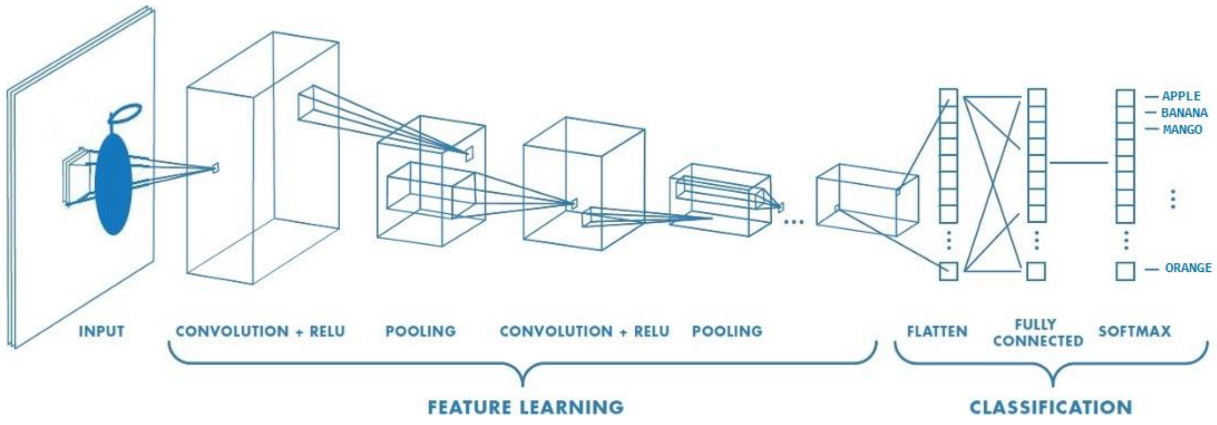


Figure 2.3.10: 4 Main Operations in CNN

Chapter 3 SYSTEM DESIGN

3.1 Flowchart of the Overall Flow of System

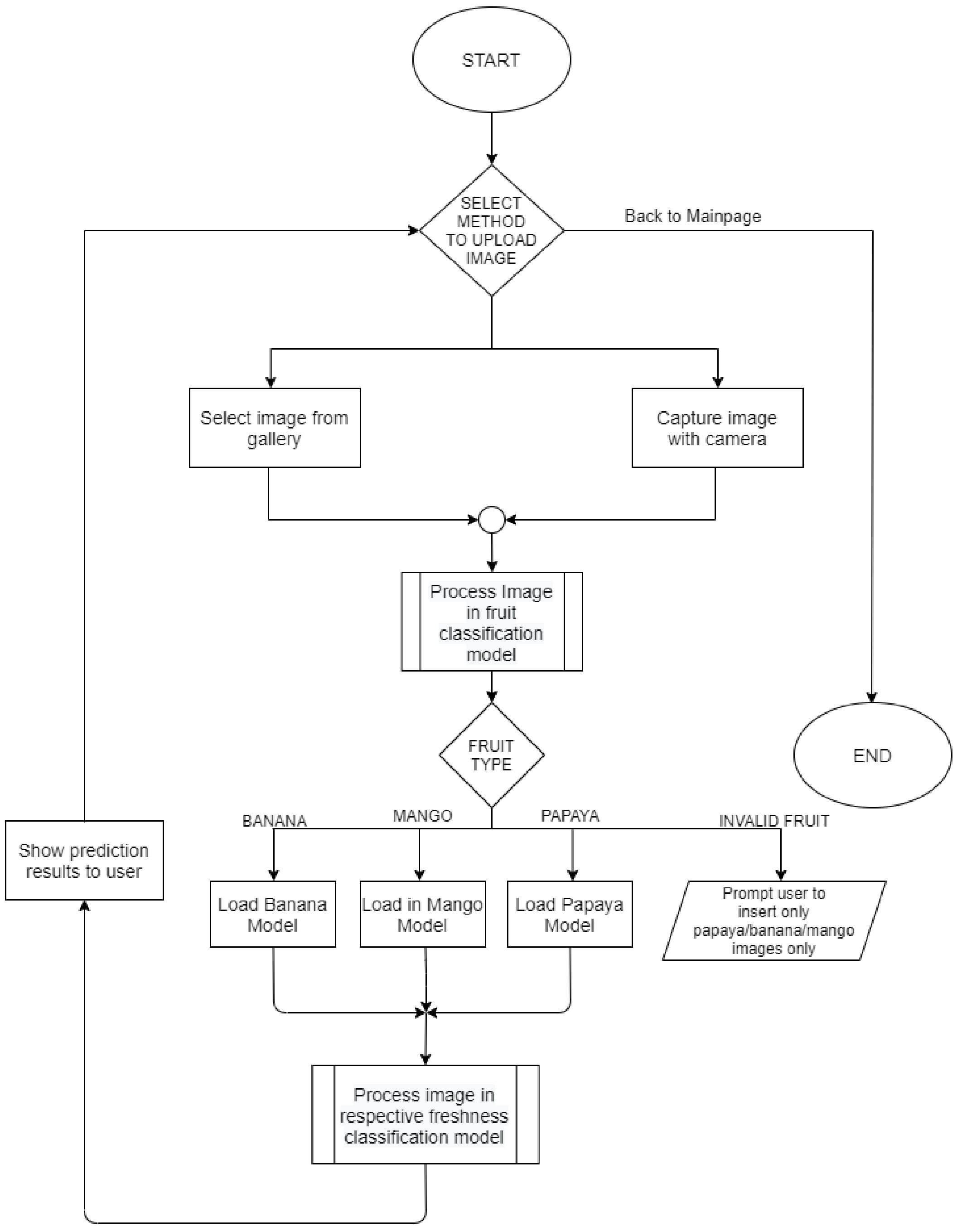

START
SELECT METHOD TO UPLOAD IMAGE
Back to Mainpage
Select image from gallery
Capture image with camera
Process Image in fruit classification model
FRUIT TYPE
END
BANANA
MANGO
PAPAYA
INVALID FRUIT
Show prediction results to user
Load Banana Model
Load in Mango Model
Load Papaya Model
Prompt user to insert only papaya/banana/mango images only
Process image in respective freshness classification model

Figure 3.1.1: Flowchart of the system

The diagram above shows the overall flow of the application in detail. At the main page of the application, the user is able to proceed to the inspection page. After that, the user is prompted to either select an image from the gallery or take a photo using their camera. To take the photo, the user is required to allow the application to access the camera application. The camera application is a default app that is installed on every phone. Next, after the image is uploaded to the system, the system will run the image in a deep learning model to classify the type of fruit uploaded and then load in the freshness/ripeness classification model according to the type of fruit detected. The image will then be run in the respective freshness/ripeness model. After that, the results of the inspection will be shown to the user such as ripeness, type of fruit and also suggestion as to what to do with the fruit. Furthermore, the user can continue doing a new inspection in the same page by selecting another image from gallery or taking a new photo with camera.

3.2 Use Case Diagram

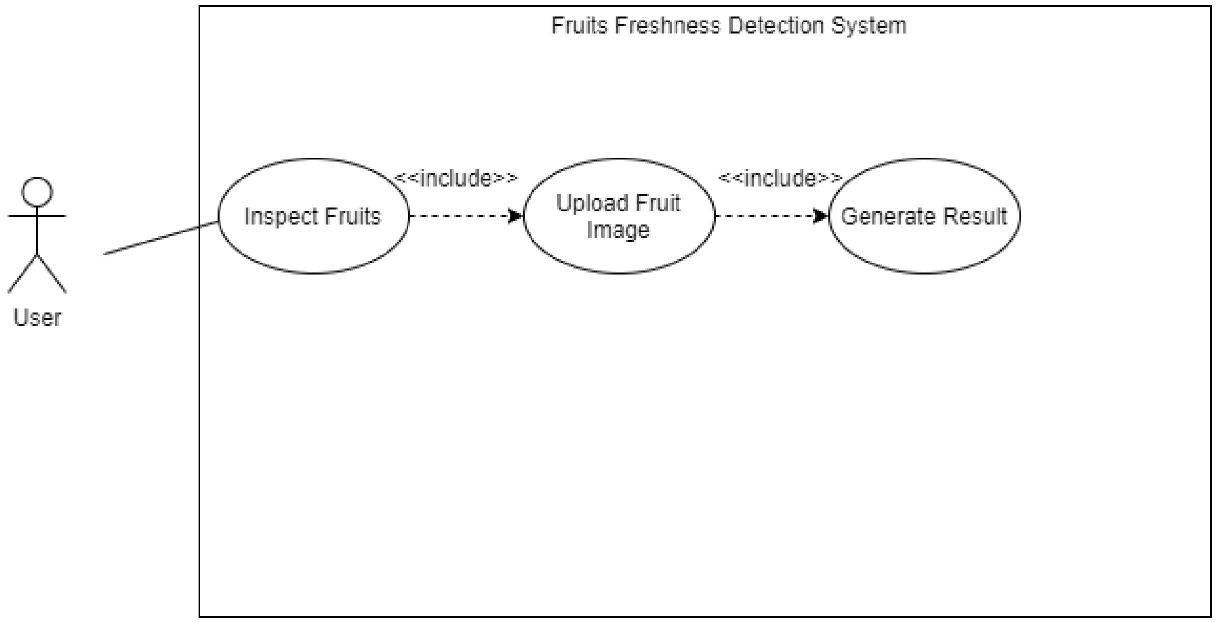


Figure 3.2.1: Use Case Diagram of the Application

The flow of event for inspect fruits are as follows:

<u>Normal Flow of events</u>:

1. System prompts user to upload image of fruit.
2. User uploads image of fruit.
3. System displays the result of inspection.

3.3 Sequence Diagram

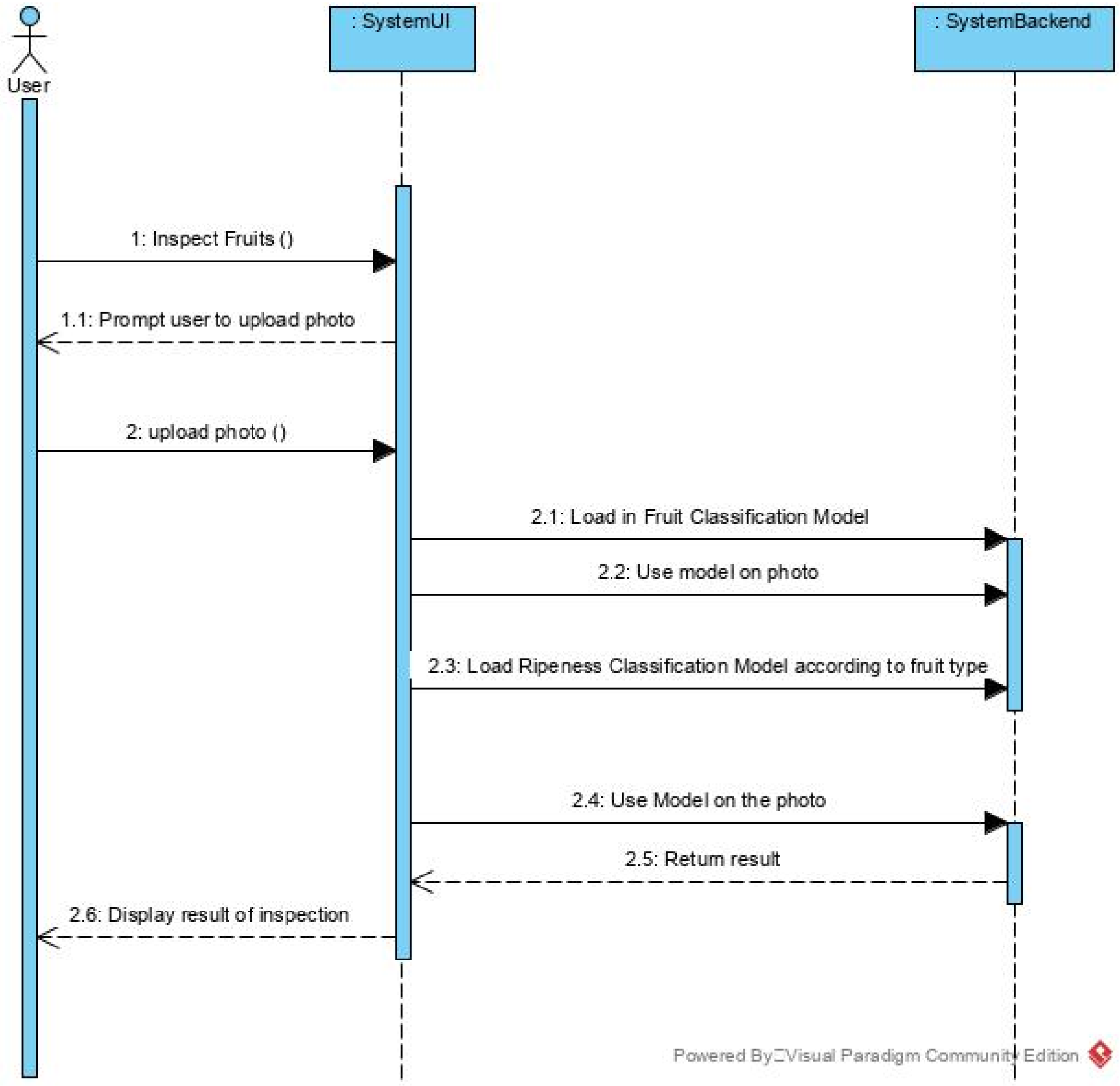

User
: SystemUI
: SystemBackend
1: Inspect Fruits ()
1.1: Prompt user to upload photo
2: upload photo ()
2.1: Load in Fruit Classification Model
2.2: Use model on photo
2.3: Load Ripeness Classification Model according to fruit type
2.4: Use Model on the photo
2.5: Return result
2.6: Display result of inspection
Powered By Visual Paradigm Community Edition

Figure 3.3.1: Sequence Diagram for Fruit Inspection

Similar to the flow chart, the above shows the flow of how users can use the application to perform fruit inspections. First of all, the system will prompt the user to upload a photo. Then, it will load in a fruit classification model that will be used to get the fruit type. After that, it will load the deep learning model of the detected fruit type to detect freshness. Furthermore, the system will run the image on the freshness model to generate the results. Finally, the system will display the result of the inspection showing the user the type of fruit and also the level of ripeness/freshness.

3.4 Application Layout

The application only contains two pages, the main page and the inspection page. The main page only serves the purpose of redirecting the user to the inspection page. Meanwhile, the inspection page is where the user will perform fruits freshness inspection. When the user enters the inspection page for the first time, the user will be informed that only 3 types of fruits can be inspected. After the user uploads an image through gallery/capture from phone, the system will first detect the fruit that the user has uploaded to decide which fruit freshness model should be used to run the image on. After running the image on the freshness model, the result will be shown to the user such as fruit type, freshness/ripeness level and also the suggestion as shown in Figure 3.4.3 below. However, if the user has uploaded a fruit that is not within the scope of the 3 fruits (banana, mango, papaya) specified in the project scope, the system will inform the user that only these 3 fruits can be inspected at the moment. This is shown in Figure 3.4.4 below.

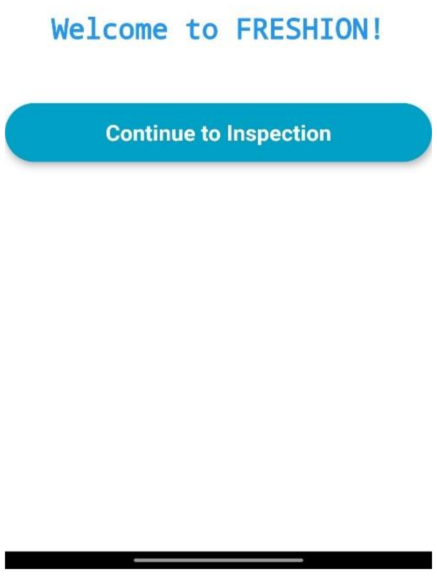
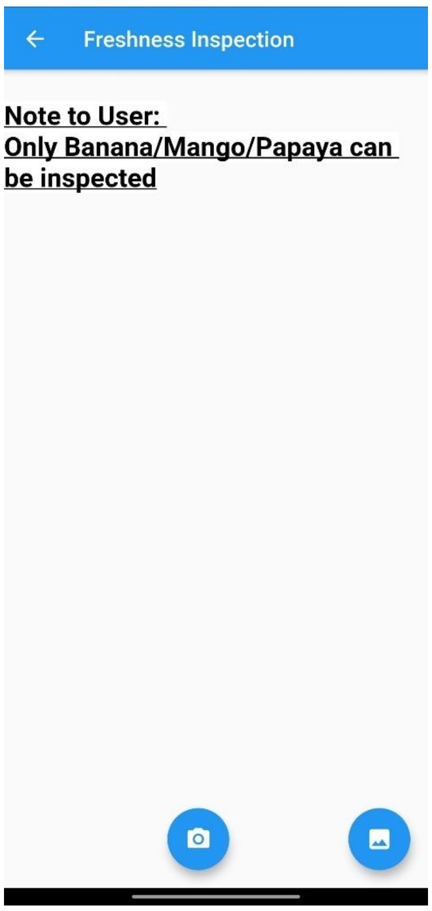
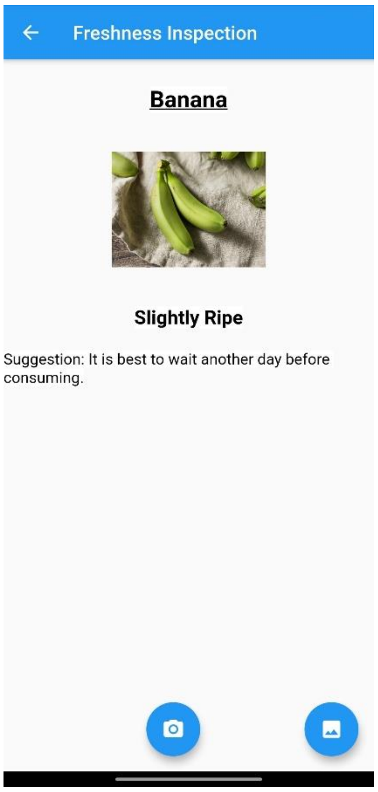


Figure 3.4.1: Main Page



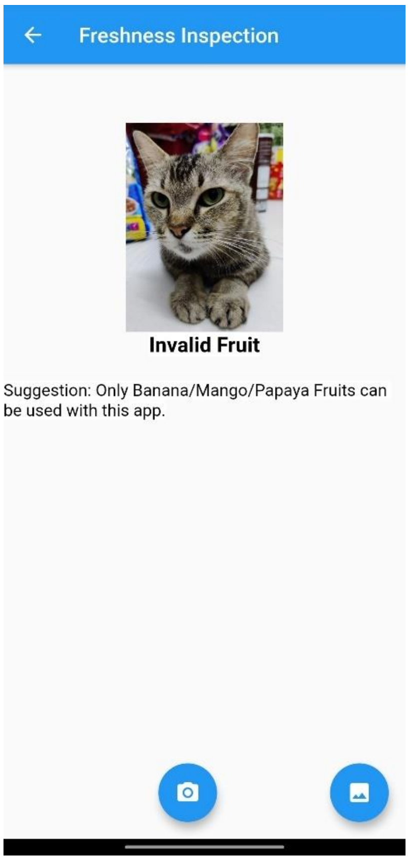
Note to User:
Only Banana/Mango/Papaya can be inspected

Figure 3.4.2: Inspection Page



Suggestion: It is best to wait another day before consuming.

Figure 3.4.3: Results shown in Inspection Page

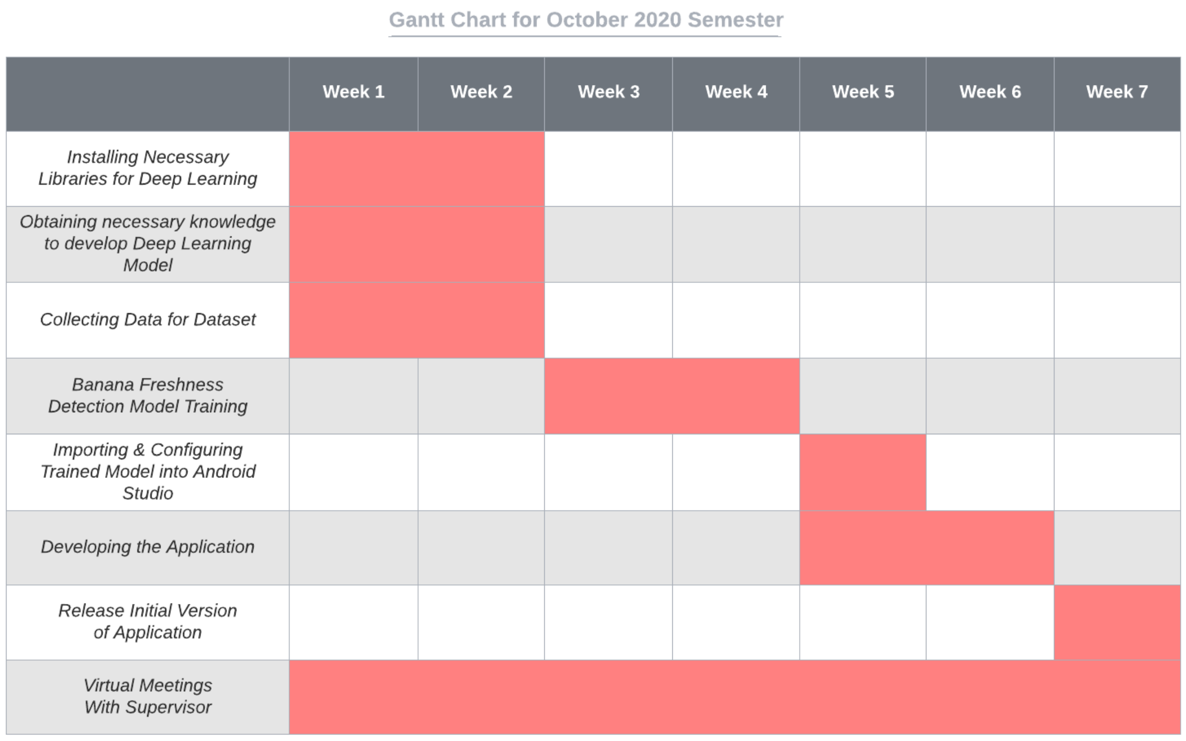


Suggestion: Only Banana/Mango/Papaya Fruits can be used with this app.

Figure 3.4.4: Invalid fruits

3.5 Timeline

3.5.1 October 2020 Semester



Gantt Chart for October 2020 Semester

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Installing Necessary Libraries for Deep Learning	🟥	🟥					
Obtaining necessary knowledge to develop Deep Learning Model	🟥	🟥					
Collecting Data for Dataset	🟥	🟥					
Banana Freshness Detection Model Training			🟥	🟥			
Importing & Configuring Trained Model into Android Studio					🟥		
Developing the Application					🟥	🟥	
Release Initial Version of Application							🟥
Virtual Meetings With Supervisor	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Figure 3.5.1: Timeline for October 2020 Semester

For this semester, the plan is to produce an initial version of the application at the end of the semester that can detect freshness of one fruit. The first two weeks are spent on researching how to develop a deep learning model, installing necessary libraries and collecting the data for dataset. The two weeks after that will be spent on training the deep learning model. During the 5th week, more effort will be spent importing and configuring the deep learning model into Android Studio than developing the application. The 6th week will be spent on, testing and continuing the development from the previous week. During the final week, the initial version of the application will be released and can be used to detect freshness of bananas.

3.5.2 January 2021 Semester

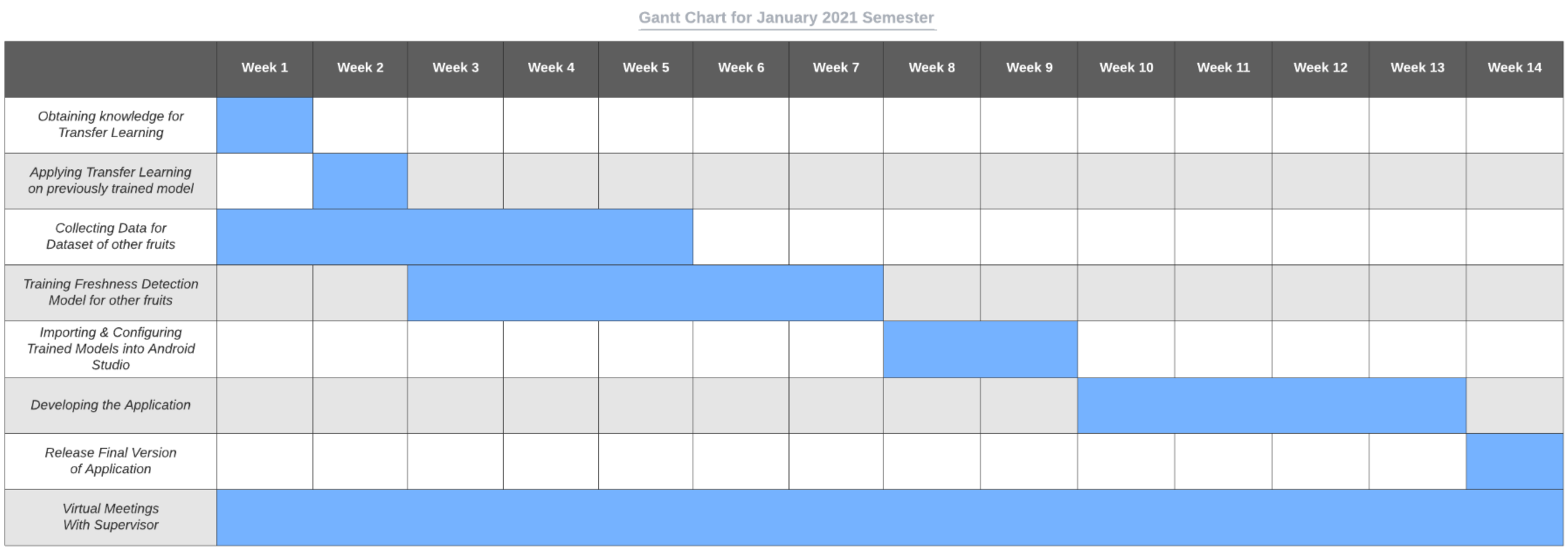


Figure 3.5.2: Gantt Chart for January 2021 Semester

For this semester, the first week will be spent on researching about transfer learning to improve the model trained from the previous semester. Furthermore, the week after that will be used to apply the knowledge learnt on improving the model. During these two weeks and the three weeks after that will be spent on collecting the data for the dataset for other fruits. At the start of week 3 will be used to train the model for fruits other than banana and the knowledge obtained from the first two weeks will be applied as well. After that, the next two weeks will be spent on importing and configuring the models trained into Android Studio. At the start of week 10, the application will start its development and testing will be done on the final week of development. During the final week, the application will release its final version which can be used to detect freshness of banana, mango and papaya.

CHAPTER 4 Methodology

4.1 Methodology

4.1.1 Methodology Used

Iterative development is the development approach used for FRESHION after careful consideration of several aspects such as user feedback, development time and cost of accommodating changes. Iterative development is a development approach which exposes the system to user comment and evolving it after several iterations of development cycle until a suitable system is developed. For FRESHION, it is important to develop an app that users are satisfied with as there will be many user interactions with the app. Besides that, through the feedback from users, future improvements can be made until the final system is completed. Since the users will be able to use an initial version of the application that is working, users will be able to give rapid feedback which leads to better application development. For example, errors or glitches can be discovered by users when they are using the initial version of the application. With this, the developers are able to satisfy the needs and requirements of the users.

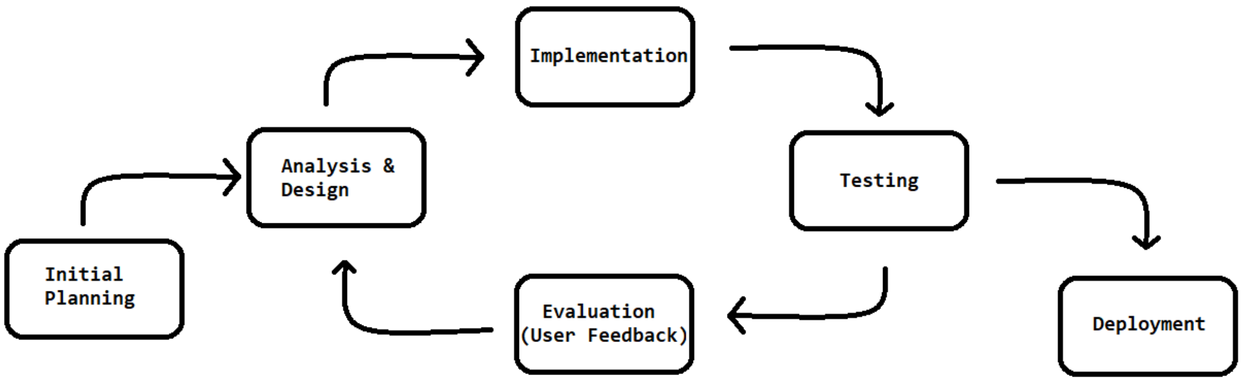


Figure 4.1.1: Iterative Development

4.1.2 General Work Procedures

1. Initial Planning

The objective of this phase is to determine the functionalities from the initial user requirements. One way of doing this is to interview the users that will use the application, in this case the consumers of fruits. Although interviewing the stakeholders of the system is important as well, it is much better if we can get the perspective of other users too. When going through the interview, it is important to record all information from interviewee and give allow them to ask questions as well. Before the interview, the interview questions should be designed beforehand. As for the design type of the questions, it can be close-ended, open-ended or probing.

2. Analysis & Design

The goal of this phase is to gather and analyse the requirements/feedback obtain from the previous phase to create a design which describes the features and functions of the system.

3. Implementation

In this phase, the documents gathered from the previous phases are used to write the necessary code to build the application. During the first few iteration of the development, the UI design and code will be simple and minimalistic to elicit feedback and ideas from users. To develop the application, Android Studio and a number of libraries will be used.

4. Testing

During this phase, the application developed from the previous phase will be tested to ensure it works properly according to the requirements obtained from the previous/initial iteration.

5. Evaluation

The goal of this phase is to expose the version of the application developed to the users in order for them to evaluate it. After obtaining the feedback from users, this feedback will be noted down and will be used to improve the application in the next iteration of development cycle.

6. Deployment

In this phase, the application will be deployed where the actual users can begin to utilize it. In other words, the application will be ready for real world environment where every end user can use it. However, this is not the end of development as bugs and glitches are inevitable in every application. Furthermore, the stakeholders or users might even ask for future improvements as well. For this reason, the developer will have to perform occasional maintenance on a regular basis.

4.2 Tools to use

4.2.1 Programming Language

1. Python

Python is a high-level programming language that is created by Guido van Rossum and was released in 1991. It comes with built in data structures and dynamic binding & typing which makes it a popular choice for Rapid Application Development. The syntax of python is beginner friendly as it emphasizes readability with notable uses of significant whitespace. This language encourages modular programming and code reuse by supporting packages and modules. Python is available in many operating systems like Windows, Linux and macOS. Besides that, Python also consists of many standard libraries that can perform many types of function. However, the main reason for using this language deep learning and image processing libraries that are needed for the solution.

2. Java

Java is a popular object-oriented programming language that many applications uses. Originally, Java was developed by James Gosling and was released in 1995. Furthermore, Java is owned by Oracle Corporation. Java applications can typically run on any type of computer architecture as long as the JVM is installed. Instead of using an SDK, Java has its own development kit called JDK. The JDK consists of a JVM and some resources to complete the development of a Java application. Furthermore, Java is an open-sourced software whereby the source code for the Java compiler is released to the public by Oracle. The reason Java was chosen is because it can be used to develop the functionalities in the mobile application. Java can be used to create mobile applications that is able to run on any Android device.

4.2.2 Software Requirements

1. TensorFlow

TensorFlow is a free and open-sourced software library that has a machine learning framework. TensorFlow is created by Google Brain, an A.I research team in google. TensorFlow is able to help developers train and develop a Machine Learning model. This library is used in the solution as it is able to help develop and train the CNN model. The model developed with TensorFlow can be further converted to TensorFlow Lite which can be further implemented and used in mobile applications.

2. Jupyter Notebook

Jupyter Notebook is a web application used to create documents that contain blocks of code and explanation texts. It is able to perform various useful functions such as data visualisation and many more. In this project, it is used as the platform for developing our CNN model as it is compatible with Python which contains TensorFlow libraries.

3. Android Studio

Android Studio is an Integrated Development Environment (IDE) for the Android OS. It is owned by Google and was released in 2014. This IDE is popular among developers to develop mobile applications. It provides users with useful tools such as design view to make the development process faster. This IDE will be used to develop the mobile application for our solution.

4. Draw.io

Draw.io is a free online diagram software to create UML, ER and workflow diagrams. This software will be used to draw the Gantt chart, ER diagram and UML diagram.

5. Visual Paradigm

Visual Paradigm is another diagramming software that can be used to create UML, ER and workflow diagrams as well. The main purpose of using this software is to draw the sequence diagram.

4.2.3 Hardware Requirements

1. Laptop

Model: ASUS ROG GL552

CPU: Intel® Core™ i7-7700HQ CPU @ 2.80GHz, Quad-core

GPU: NVIDIA GeForce GTX 950M

RAM: 12GB

System Type: 64-bit Operating System

2. Mobile Device

Model: One Plus Nord

Operating System: OxygenOS 10.5 (Based on Android 10)

CPU: Octa-core (1x2.4 GHz Kryo 475 Prime & 1x2.2 GHz Kryo 475 Gold & 6x1.8 GHz Kryo 475 Silver)

GPU: Adreno 620

Camera: 48MP (Primary)

4.3 User Requirement

4.3.1 Functional Requirements

1. As a user, I should be able to access my camera to detect freshness of fruits.
2. As a user, I should be able to select photo from my gallery to detect freshness of fruits.
3. As a user, I should be able to see the result of inspection right after uploading the photo.
4. As a user, I should be informed with the type of fruits that can perform inspection
5. As a user, I want all the buttons in the application to be working.

4.3.2 Non-Functional Requirements

1. As a user, I should be able to run the application on any Android device.
2. As a user, I want the fruit freshness prediction to be 99 percent accurate of the time.
3. As a user, I should be able to use the application without crashing.

4.4 Specification

4.4.1 Analysis

The objectives of the project is carefully analysed in order to ensure the final product can reach its maximum potential in order to help users that lack experience in selecting fresh fruits. In order to do this, existing systems/applications are analysed and reviewed in order to find new and existing features to be included in the final product of this project. For example, the two applications that are reviewed, ClariFruit and Fruit & Veg Detector did not have the feature to automatically detect the type of fruit before inspecting it. However, the final product of this project is able to detect the type of fruit before performing freshness inspection. Therefore, this is why through analysing existing systems, the final product of the project is improved at the end.

4.4.2 Design

The design of the mobile application is designed from scratch. Therefore, every page in the application needs to have elements that are arranged neatly so that the layout of the page appears neat and tidy. Although the design of the application may look simple, it still gets to job done and can be improved in the future when the time comes.

4.4.3 Verification Plan

In order to verify the correctness and completeness of the application, an initial version of the mobile app is released to a small group of developers from University Tunku Abdul Rahman for testing purposes. This group of people will then provide feedback and suggestions to the developer to improve the application. Then, after developing a newer version, the next version is then released to the same group of people for testing and feedback again. This method is known as iterative development whereby the application is being constantly improved through various update and feedback until the final version is released.

CHAPTER 5 Implementation & Testing

5.1 Flow of developing the CNN Model

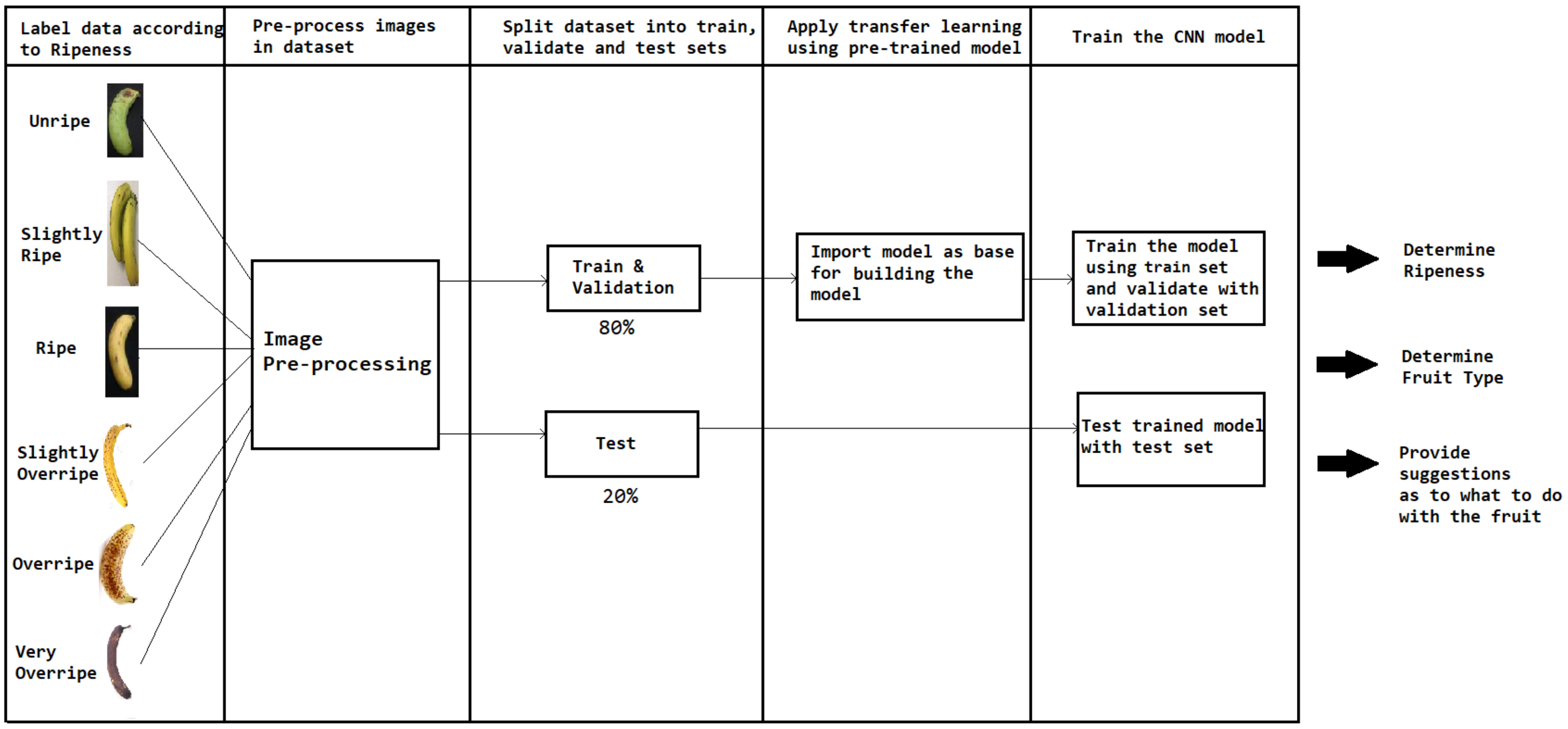

Label data according to Ripeness
Pre-process images in dataset
Split dataset into train, validate and test sets
Apply transfer learning using pre-trained model
Train the CNN model
Unripe
Slightly Ripe
Ripe
Slightly Overripe
Overripe
Very Overripe
Image Pre-processing
Train & Validation
80%
Test
20%
Import model as base for building the model
Train the model using train set and validate with validation set
Test trained model with test set
Determine Ripeness
Determine Fruit Type
Provide suggestions as to what to do with the fruit

Figure 5.1.1: An overview of flow for developing CNN Model

The above figure shows the flow for developing the CNN model to be used in the application to determine the freshness of fruits. First of all, the dataset is collected from various sources such as Kaggle, iStock, gettyimages and many more. After collecting the dataset, each fruit image is labelled according to its ripeness which are unripe, slightly ripe, ripe, slightly overripe, overripe and too overripe. After that, the images will be resized to a smaller size to help the GPU process them faster. In addition to that, the images are split into training, validation and test set with a ratio of 70:10:20 respectively. The reason for the validation set is used to estimate the accuracy of the model trained and the training set is to apply the developed model to real-world data to test its accuracy. After that, import a pre-trained model to serve as the base of our CNN model. Next, with the base model, adjustments and additions will be made to make the model to perform fruit freshness detection classification. Next, the model will be trained and tested to see if it works properly or not for real world data. Finally, we are able to use the CNN model to determine the level of ripeness of fruit, freshness of fruit and how long the fruit has before it is spoilt.

5.2 Fruit Dataset Preparation

First of all, the fruit images in the dataset are collected from many different sources from the internet such as Kaggle, iStock, gettyimages and so on. Besides the internet, the images are also collected by taking photos of fruits in markets and grocery stores that sells fruits. After that, the fruit images will be separated into different folders for different fruits. Note that the number of images collected for each fruit should be around 400. Besides that, the training and testing dataset will be split manually by 80:20 into different folders. So, if there are 400 images in the dataset, 40 images will be for testing and 360 will be for training.

The next part will be slightly tedious, where the each of the fruit images will be categorised manually into different folders according to their ripeness. Figure 3.4.2 below shows one of the charts used as the base for manually categorising the fruits. After that, the number of images for each class should not be extremely far apart from each other. The reason being that we have to make sure that there a plentiful amount of data for each class so that the dataset is not imbalanced. This is particularly important as an imbalanced dataset will cause the model to not learn how to classify the class that is greatly lacking data.

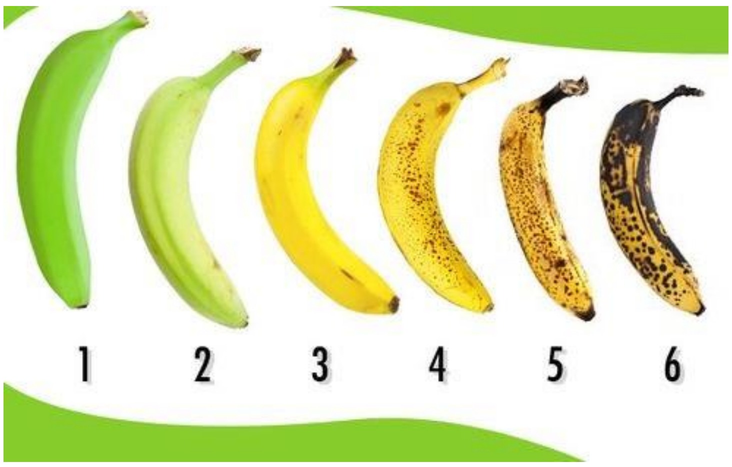
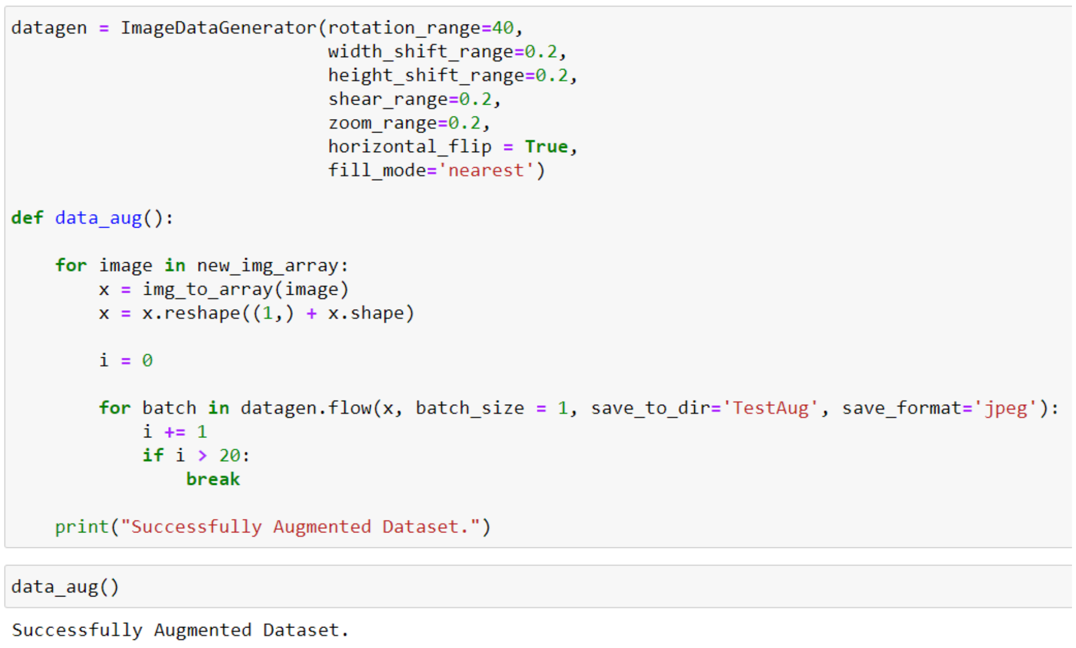


Figure 5.2.1: Banana Ripeness Chart

After manually categorising the fruits into different folders, the next step is to do data augmentation to increase the number of images in our current dataset. Data augmentation is a useful method used in machine learning to increase the number of data in the dataset when the data is too less. Basically, what data augmentation does is to take an image in our dataset and duplicate it with different orientation for each image. Thus, this will greatly increase the number of data that we have in our dataset. After performing data augmentation on our dataset, the number of data in our dataset should be increased tenfold for every class. Data augmentation is done by using the function shown in the figure below.


```
datagen = ImageDataGenerator(rotation_range=40,
                             width_shift_range=0.2,
                             height_shift_range=0.2,
                             shear_range=0.2,
                             zoom_range=0.2,
                             horizontal_flip = True,
                             fill_mode='nearest')

def data_aug():

    for image in new_img_array:
        x = img_to_array(image)
        x = x.reshape((1,) + x.shape)

        i = 0

        for batch in datagen.flow(x, batch_size = 1, save_to_dir='TestAug', save_format='jpeg'):
            i += 1
            if i > 20:
                break

    print("Successfully Augmented Dataset.")
```

```
data_aug()
```

```
Successfully Augmented Dataset.
```

Figure 5.2.2: Data Augmentation

Many types of data augmentation techniques are used to increase the number of data in the dataset. These techniques are rotation, flip, zoom, shifting and many more. Increasing the number of data in the dataset will greatly improve our model’s accuracy and can also help to reduce the model from overfitting as well.

5.3 Regression CNN

5.3.1 Experiments with different pre-trained model (Regression CNN)

Initially, there was an idea to turn the current fruit classification problem into a regression problem. Basically, by changing the problem domain of the project to regression, we can train the CNN model to output a rough freshness level value. With this freshness level value, we can use it to predict its ripeness value as well by specifying a freshness scale. Several CNN regression models have been developed using different pre-trained models such as VGG16, MobileNetV2 and ResNet50. After using these pre-trained models as our base model, the fully connected layer of the base models is then modified to fit the current problem domain. The training results are as shown below:

1. VGG16

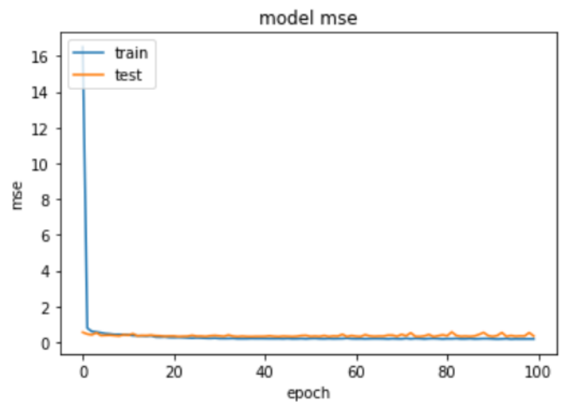


Figure 5.3.1: Training Result of VGG16 CNN Regression

The result shown is quite good, where the mean squared error (MSE) decreases with each epoch at the beginning of training phase and slowly remains at a low value around 0.1~0.2 MSE. Besides that, the validation MSE also remains at a low value similar to the training MSE. This shows us the model is training well and can be used as the model for our regression problem.

2. MobileNetV2

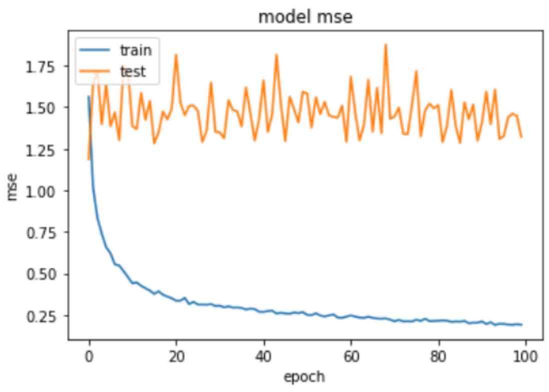


Figure 5.3.2: Training Result of MobileNetV2 CNN Regression

The training result shown in the figure above is quite bad, the model is definitely overfitting. This is because the difference between the train and test MSE is vastly different. Besides that, even though the train MSE is constantly decreasing with each epoch, the validation MSE has very large fluctuations as the number of epochs increases.

This might mean the learning rate is too high. However, given that the train and validation MSE is so far apart from each other, we can conclude that even if we lower the learning rate, the validation MSE will still be too high. Therefore, this model is not suitable to be used as it is overfitting.

3. ResNet50

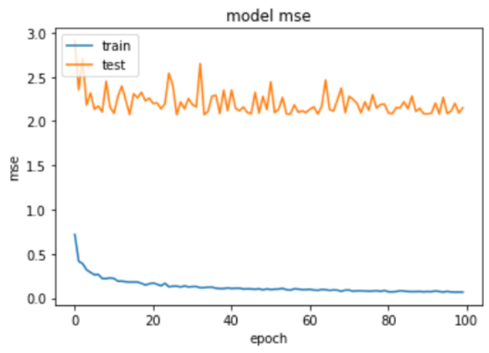


Figure 5.3.3: Training Result of ResNet50 CNN Regression

As observed from the diagram above, the model trained is not suitable to be used as it is overfitting due to its MSE being significantly different when comparing its training one with its validation counterpart. Therefore, we should not consider this model to be used for our regression CNN.

5.3.2 Result of experiments

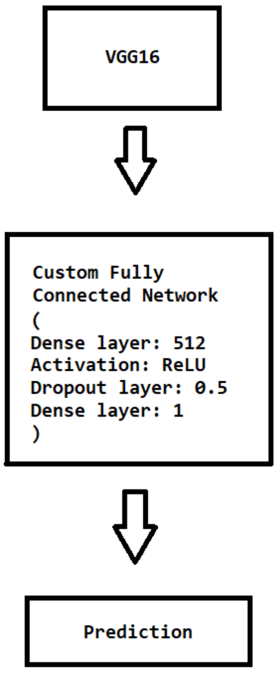
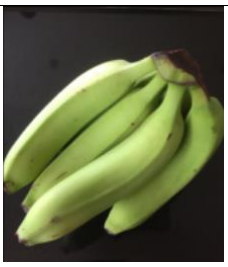
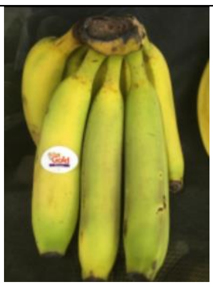
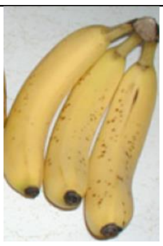
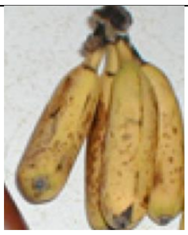


Figure 5.3.4: Architecture of the custom VGG16 network

According to the diagram above, the VGG16 pre-trained model is used as the base model for our architecture. After that, a custom fully connected layer is created through adding a few layers such as dense and dropout layer. The dropout layer is added to reduce overfitting of our network and the activation function ReLU is to squish the values of the output from 0 to positive infinity. With this we will only get positive output from the network.

5.3.3 Test the Regression Network on real data

Image	**Actual Category**	**Predicted Value**
	Unripe	5.7693
	Slightly Ripe	8.1615
	Ripe	7.4148
	Slightly Ripe	4.0719

	Overripe	2.1447
	Very Overripe	12.9891

Table 5.3.1: Result of testing the Regression Model Trained

According to the table shown above, the result of the regression model trained is not as what is expected. Basically, we are unable to find a scale that is suitable to classify the ripeness/freshness of the fruit according to the value given by the model. As we can observe from the table, the value for slightly overripe and overripe banana is very low, while the value of the ripe and slightly ripe is too high. So, we can conclude that the model although trains well, however the output is not suitable for us to create a scaling to predict freshness. Therefore, regression CNN does not suit our problem domain and classification CNN should be used instead.

5.4 Classification CNN

To develop the CNN Model for classification, several pre-trained models are selected in order to perform transfer learning. Basically, in transfer learning, the pre-trained model will be used as a base model for the network and its fully connected layer will be modified to fit the classification problem that we have. For our case, the network needs to be able to classify 6 different freshness level (unripe, slightly ripe, ripe, slightly overripe, overripe, very overripe). However, using different pre-trained models will result in different accuracy and different model size. Which is why the best model selected should have a high accuracy and a small model size (in terms of how much it takes up storage space). The reason why we need to find a model size that is small is because this model will be integrated to the mobile platform, therefore, it is crucial to ensure that the model size remains as small as possible. Otherwise, it might take too long for the mobile application to load in the model. Two pre-trained models (ResNet18 and MobileNetV2) have been taken for testing using the banana dataset.

5.4.1 Experiment on different pre-trained models (Classification CNN)

1. ResNet18

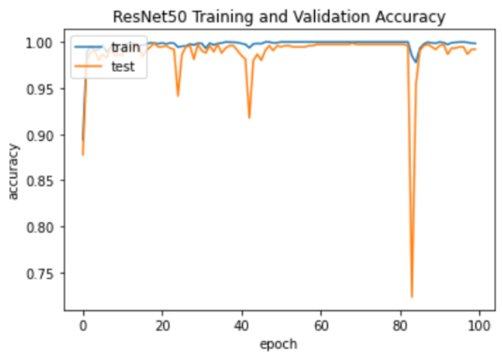


Figure 5.4.1: ResNet18 Accuracy graph

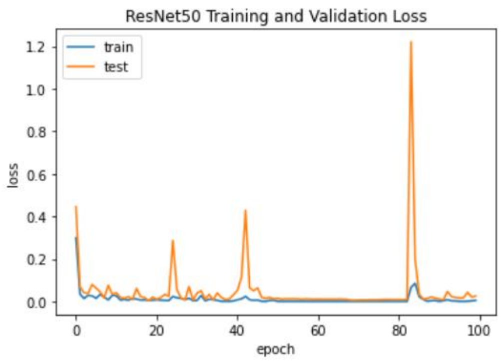
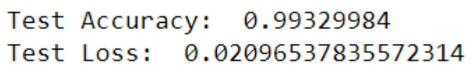


Figure 5.4.2: ResNet18 Loss graph

Observing the accuracy diagram above shows us that the validation accuracy has a few significant drops in several epochs during training notably in the ~20th, ~40th and ~80th epochs. This may be due to several reasons such as overfitting or the model choses the batch that contains “more difficult” input images to be classified, which causes the validation accuracy to be lower in some epochs. The similar thing can be said for the loss graph, in several instances of epochs such as the ~20th, ~40th and ~80th, the model has suddenly high loss value due to the reason stated in the previous sentence as well.



```
Test Accuracy:  0.99329984
Test Loss:  0.020965378355723l4
```

Figure 5.4.3: Test Results of ResNet18

The above diagram shows us the accuracy and loss of the model after testing it with the test dataset prepared previously. This shows us the model is performing very well, as the accuracy is very high at 99.32% and the loss of the model is very small at only 0.021. Which means in terms of performance ResNet18 definitely perform well.

banana(resnet18).h5	H5 File	719,327 KB
banana(resnet18).tflite	TFLITE File	239,264 KB

Figure 5.4.4: File size of the saved model

The above diagram shows us the file size of the model after saving it. The file size is quite huge with a whopping 719MB. Even after converting the model to a

TensorFlow Lite format, the model is still quite huge, with 239MB. This means that if we were to load this model in the mobile platform, it might take some time in order to do so.

2. MobileNetV2

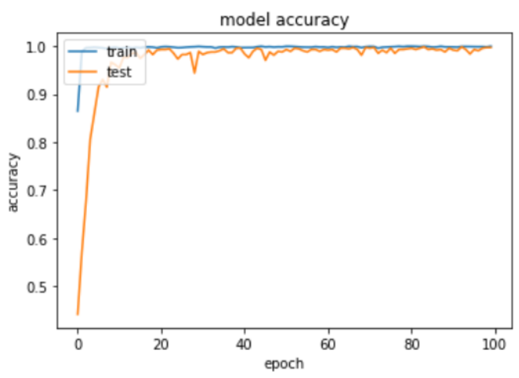


Figure 5.4.5: MobileNetV2 Accuracy graph

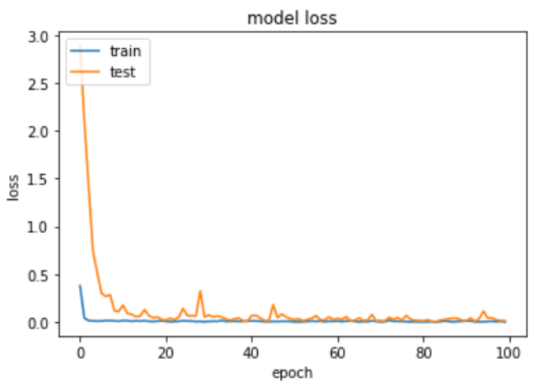
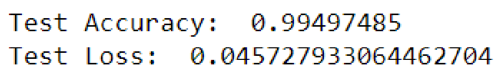


Figure 5.4.6: MobileNetV2 Loss graph

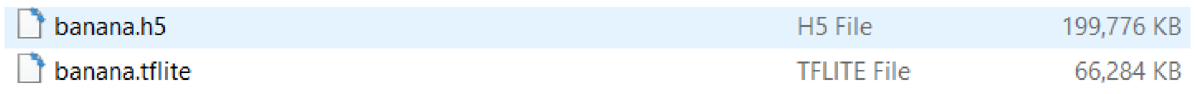
According to the accuracy diagram above, we can see the model is training fine whereby the model shows no signs of overfitting and underfitting. The train and validation accuracy remains almost similar with each other which means that the model is not overfitting. The model is not underfitting since the accuracy is high as shown in the same diagram above. Furthermore, the same can be said with the loss graph where the training and validation loss remains quite similar with each other with minute differences.



```
Test Accuracy:  0.99497485
Test Loss:  0.045727933064462704
```

Figure 5.4.7: Test Results of MobileNetV3

According to the test results shown in the diagram above we can conclude that the model trained is performing well with real world data. The test accuracy and loss above is a proof of that, as the accuracy is 99.49% when testing on the test set and the loss is only at 0.0457 which is quite low. Therefore, this model is good enough in terms of performance to be used in the project.



banana.h5	H5 File	199,776 KB
banana.tflite	TFLITE File	66,284 KB

Figure 5.4.8: File size of the saved model (MobileNetV3)

According to the diagram above, it shows us that model size is not that big, as the file size is only about 200MB in size, and after the conversion, it is quite small with only 66MB. Which means the model can be loaded in the mobile platform in a short amount of time.

5.4.2 Comparison

Model	**Performance**	**Testing Accuracy**	**Testing Loss**	**File Size**
ResNet18	Almost Similar	99.32%	0.0210	Large
MobileNetV2	Almost Similar	99.49%	0.0457	Small

Table 5.4.1: Comparison of different pre-trained models experimented

After comparing the two models experimented, both models perform almost similarly well. When comparing the test loss, ResNet18 is definitely better, but if we compare the model using test accuracy, MobileNetV2 performs better. In conclusion, we can say both of them perform similarly well. However, since MobileNetV2 has a smaller file size (3 times smaller than ResNet18), it is selected in the end.

5.4.3 Architecture of the model selected

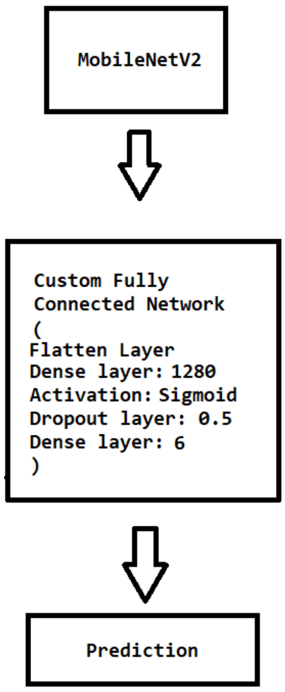


Figure 5.4.9: Custom Network using MobileNetV2

The above diagram shows the network that is built using MobileNetV2 as the base model. A custom fully connected network is added at the end in order to classify the fruits according to their ripeness. In the custom fully connected network, the first layer is the flatten layer which is used to convert a multi-dimension tensor/matrix into a single array before classifying them in the later layers. After that, a dense layer of 1280 neurons is added whereby it will feed the outputs from the final layer of MobileNetV2 into its own neurons.

Furthermore, a Sigmoid activation function is added to squish the output value of the previous layer to only values between 0 and 1. The reason for doing so is because for a classification problem, we are trying to find the probability of each class as an output using the model. Since probability only exist between values of 0 and 1, the Sigmoid function is used to get the probability of each class.

After the sigmoid function, a dropout layer is added in order to prevent the model from overfitting. How it works is by randomly supressing a specified amount of neurons which forces the model to learn more useful features. Lastly, a dense layer of 6 output neurons is added. The reason for having 6 output neurons is because we have

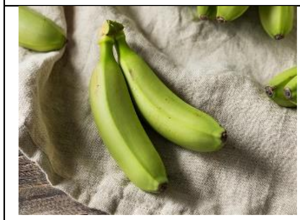
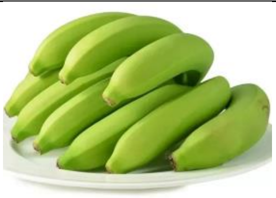
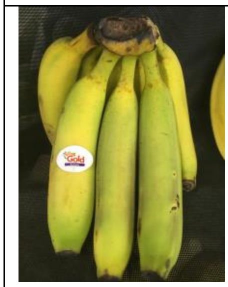
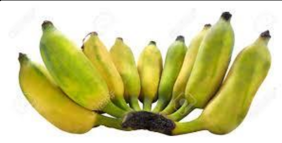
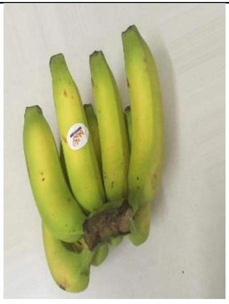
6 classes (unripe, slightly ripe, ripe, overripe, slightly overripe and very overripe) in total that we are trying to classify.

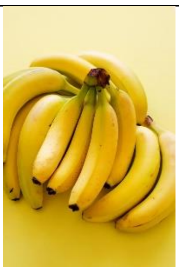
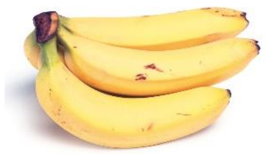
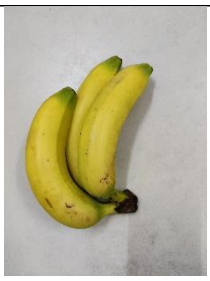
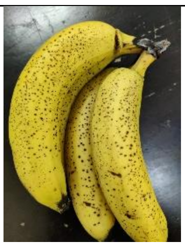
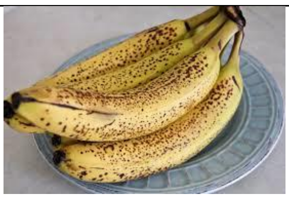
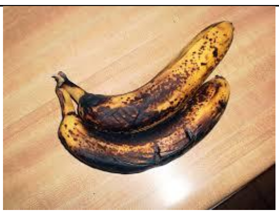
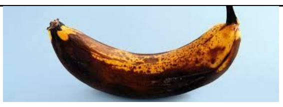
In conclusion, this custom network is used to train different type of fruit models in order to classify its freshness/ripeness according to the type of fruit. Besides that, a fruit classification model is also developed using the similar network in order for the system to use this model to determine the type of fruit model that is to be loaded before performing fruit freshness detection.

5.5 Testing

The below table shows us the test cases for the mobile application on various types of fruit and their respective freshness.

5.5.1 Banana Test Cases

Image	**Actual Ripeness**	**Predicted Ripeness**	**Predicted Type**	**True/False**
	Unripe	Unripe	Banana	True
	Unripe	Unripe	Banana	True
	Slightly Ripe	Slightly Ripe	Banana	True
	Slightly Ripe	Unripe	Banana	False. Incorrectly predicted as unripe
	Slightly Ripe	Slightly Ripe	Banana	True

	Ripe	Ripe	Banana	True
	Ripe	Ripe	Banana	True
	Ripe	Slightly Ripe	Banana	False. Incorrectly predicted as Slightly Ripe
	Slightly Overripe	Slightly Overripe	Banana	True
	Slightly Overripe	Slightly Overripe	Banana	True
	Overripe	Overripe	Banana	True
	Overripe	Very Overripe	Banana	False. Incorrectly predicted as Very Overripe

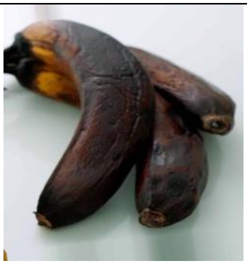
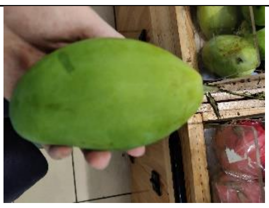
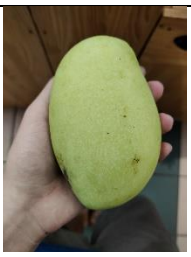
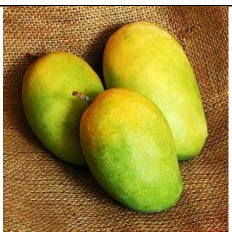
	Very Overripe	Very Overripe	Banana	True

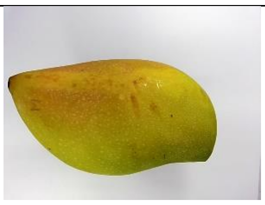
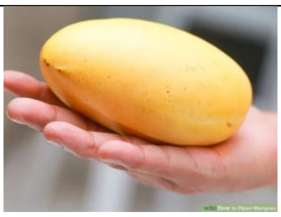
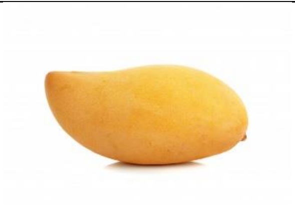
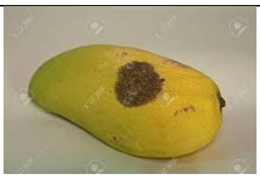
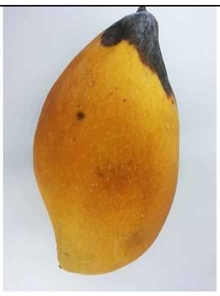
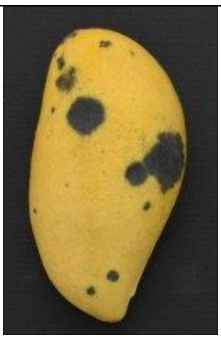
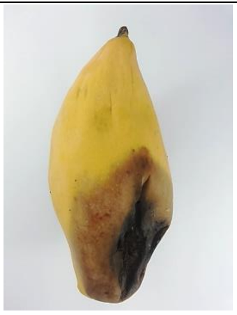
Table 5.5.1: Banana Test Case

Total Cases = 13; Passed Cases = 10; Failed Cases = 3

According to the table above, among the 13 test cases above, only 3 test cases has wrongly predicted the ripeness of the banana. However, the prediction given by the application is not that far apart from the actual ripeness. So, it can still be said to be accurate in a way. For example, the test case where the application has wrongly predicted a slightly ripe banana as unripe, which is wrong but is not that far from the actual ripeness. Therefore, we can say that the banana ripeness/freshness classification model has passed the testing phase.

5.5.2 Mango Test Cases

Image	**Actual Ripeness**	**Predicted Ripeness**	**Predicted Type**	**Pass/Fail**
	Unripe	Unripe	Mango	Pass
	Unripe	Unripe	Mango	Pass
	Slightly Ripe	Slightly Ripe	Mango	Pass

	Slightly Ripe	Slightly Ripe	Mango	Pass
	Ripe	Ripe	Mango	Pass
	Ripe	Ripe	Mango	Pass
	Slightly Overripe	Slightly Overripe	Mango	Pass
	Slightly Overripe	Slightly Ripe	Mango	Fail, Incorrectly predicted fruit as slightly ripe
	Slightly Overripe	Slightly Overripe	Mango	Pass
	Overripe	Slightly Overripe	Mango	Fail, Incorrectly predicted fruit as slightly overripe

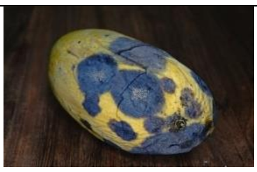
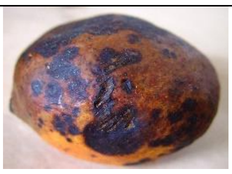
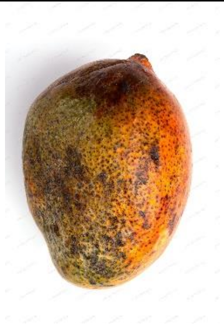
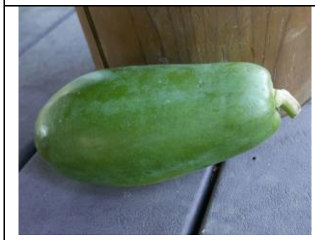
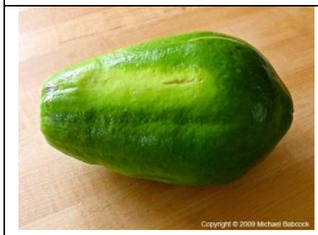
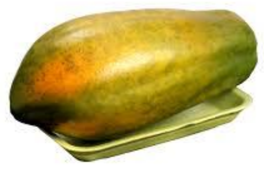
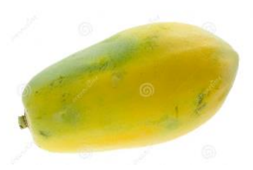
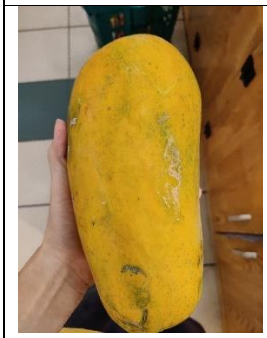
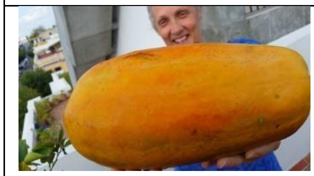
	Very Overripe	Very Overripe	Mango	Pass
	Very Overripe	Very Overripe	Mango	Pass
	Very Overripe	Very Overripe	Papaya	Fail, Incorrectly predicted fruit type as papaya

Table 5.5.2: Mango Test Case

Total Cases = 13; Passed Cases = 10; Failed Cases = 3

The above table shows us the test case for fruit inspection for mango. Among the 13 test cases, 10 cases have passed while 3 cases have failed. For the failed cases, one of them in particular can be said as a very bad prediction. The failed test case for predicting a slightly overripe mango as a slightly ripe one tells us that the mango ripeness model did not perform as it expected. Furthermore, another failed test case tells us the fruit classification model didn't perform as well as expected is the final test case where the mango is incorrectly predicted as papaya. However, it still has passed most of the test cases which means it can still be used for the final release of the system. Nevertheless, the mango ripeness model definitely can be improved by increasing the number of dataset for each class in order to achieve a better accuracy.

5.5.3 Papaya Test Cases

Image	**Actual Ripeness**	**Predicted Ripeness**	**Predicted Type**	**Pass/Fail**
	Unripe	Unripe	Papaya	Pass
	Unripe	Unripe	Papaya	Pass
	Slightly Ripe	Slightly Ripe	Papaya	Pass
	Slightly Ripe	Slightly Ripe	Papaya	Pass
	Ripe	Ripe	Papaya	Pass
	Ripe	Ripe	Papaya	Pass

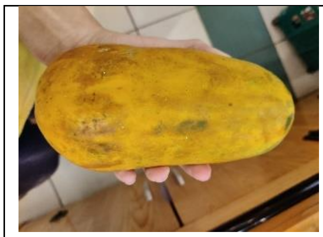
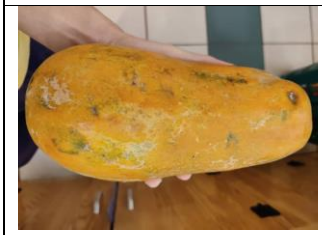
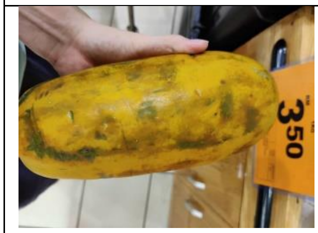
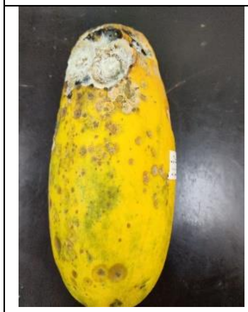
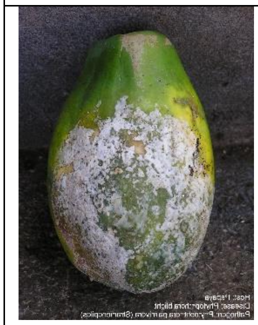
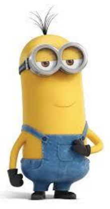
	Slightly Overripe	Slightly Overripe	Papaya	Pass
	Slightly Overripe	Slightly Overripe	Papaya	Pass
	Overripe	Overripe	Papaya	Pass
	Very Overripe	Very Overripe	Papaya	Pass
	Very Overripe	Very Overripe	Papaya	Pass

Table 5.5.3: Papaya Test Case

Test Cases = 11; Passed Cases = 11; Failed Cases = 0

According to the table above, we are able to conclude that the papaya model for fruits freshness detection performs quite well and can be used for the final release of the application. There are no failed cases, but however it does not mean that it won't ever fail in a "real-world" situation. Nevertheless, the test cases above shows us that the model is competent enough to be used for the system.

5.5.4 Invalid Fruit Test Cases

Image	**Actual Ripeness**	**Predicted Ripeness**	**Predicted Type**	**Pass/Fail**
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	Ripe	Banana	Fail, Incorrectly predict fruit as banana

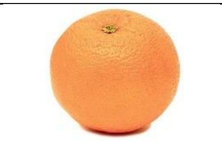
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass
	-	-	Invalid Fruit	Pass

Table 5.5.4: Invalid Fruit Test Case

Test Cases = 10; Passed Cases = 9; Failed Cases = 1

According to the table above, it shows us that fruit detection model used in the application performs quite well. Only one object has been incorrectly classified as a banana. This fruit detection model is used to detect the fruit type of the input image from the user, and load the respective model if it is available, if it is not then informing the user that that fruit type cannot be inspected yet. All in all, the fruit detection model works fine, so it will be included in the final version of the system.

CHAPTER 6 Conclusion

6.1 Project Review

At the end of the project, all of the objectives have been achieved such as the development of a CNN model to classify fruits according to their fruit type, development of CNN models for each respective fruit to classify them according to their freshness/ripeness and also a mobile application that is able to load in these models to allow users of the application to perform fruits freshness inspection.

However, in order to achieve these objectives, several difficulties and challenges have been faced as well.

1. Exporting the CNN model to the Android platform

There are some alternatives with different configurations required to convert the TensorFlow model to TensorFlow Lite model that can run on Android. Some tedious trial and error is required in order to find out which one is the appropriate conversion so that the accuracy of the model remains the same in the mobile platform as well.

2. Collecting dataset to develop CNN model

In order to develop a model that is able to accurately perform fruit inspections in the real world, a lot of data needs to be collected to ensure that the model developed is accurate enough when used in the real world. Furthermore, since there are 6 classes for each fruit, we still need to make sure the number of data in each class are similar with each other. If we don’t do this, there might be data imbalance which will hurt our model performance later on.

6.2 Novelties and Contributions

6.2.1 Novelties

There are several unique aspects in this project which are:

1. Creating and compiling a dataset containing fruit images that are further categorised by their ripeness.

With this dataset compiled, data scientists can use this dataset for future research and development. This will eliminate the tedious process of collecting the dataset and more time can be focused on future research and development instead.

2. Developed a CNN models to categorise fruits by their ripeness and make this technology portable by integrating it into a mobile application.

With this, every consumer with the application installed can use it to classify fruits by their freshness during their grocery shopping run. This will help them with their decision when picking fruits for purchase. Furthermore, with this application, the agriculture industry can perform fruit inspections without using harmful chemicals and substances that will harm the fruit.

6.2.3 Contributions

Currently, are only a few mobile applications in the market that is able to perform fruit freshness detection. However as reviewed in Chapter 2, these existing applications are not as good in terms of performance and complexity. With this project, consumers will be able to use the application to help them select fresh fruits before purchasing them. This may be even useful for consumers that lack the knowledge from selecting fresh fruits. Besides that, the application can also be used by the agriculture industry to do fruit inspection which does not harm the fruit during inspection. Users that use the application only needs to capture image/upload image from gallery to perform inspection. The result of the inspection will be shown to the user after running the fruit image on the respective model loaded.

6.3 Future Work

Since the final version of the application can only perform fruit inspection of 3 types of fruit (mango, banana and papaya), the application still can be improved by adding more fruit inspections. This can be done by collecting dataset of the respective fruit and train the CNN model using the CNN architecture stated in Chapter 5. Besides that, the fruit classification model needs to be retrained by adding a new class that contains the new fruit. This is so that the system is able to know the type of fruit the user has uploaded in order to load the respective model.

Furthermore, the UI design of the mobile application can be further improved. For example, adding a dark mode for users that prefer viewing the application in the dark. Other than that, the design can also be further beautified since the current design used is quite simple.

Bibliography

Bhargava, A. & Bansal, A., 2018, “Fruits and vegetables quality evaluation using computer vision: A review”, *Journal of King Saud University - Computer and Information Sciences*, vol. 6 no.10 pp. 637-640.

Brownlee. J, 2019, “What is Deep Learning?”, *Machine Learning Mastery*, viewed 26 July 2020, https://machinelearningmastery.com/what-is-deep-learning/

Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., Zhang, C. and Huang, W., 2020. “On line detection of defective apples using computer vision system combined with deep learning methods”. *Journal of Food Engineering*, p.286.

Harsh, A., Jha, K., Srivastava, S., Raj, A. and S, R., 2020. “Fruit Freshness Detection using CNN Approach”. *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2 no. 6, pp.456-463.

H.M. Monovar, R. Alimardani and M. Omid, 2011, “Detection of red ripe tomatoes on stem using Image Processing Techniques”, *Journal of American Science,* vol. 7 no. 7.

Mazen, F. and Nashat, A., 2019. “Ripeness Classification of Bananas Using an Artificial Neural Network”. *Arabian Journal for Science and Engineering*, vol. 44 no. 8, pp.6901-6910.

N.U.A. Ibrahim, S. Abd Aziz and N.M. Nawi, 2017, “Novel impedance measurement technique for soluble solid content determination of banana”, *Pertanika Journal Science & Technology*, vol. 25 no. 2 pp. 519-526.

Pande, A., Munot, M., Sreeemathy, R. and Bakare, R., 2019. “An Efficient Approach to Fruit Classification and Grading using Deep Convolutional Neural Network”. *5th International Conference for Convergence in Technology*. Pune, India. 29-31 March.

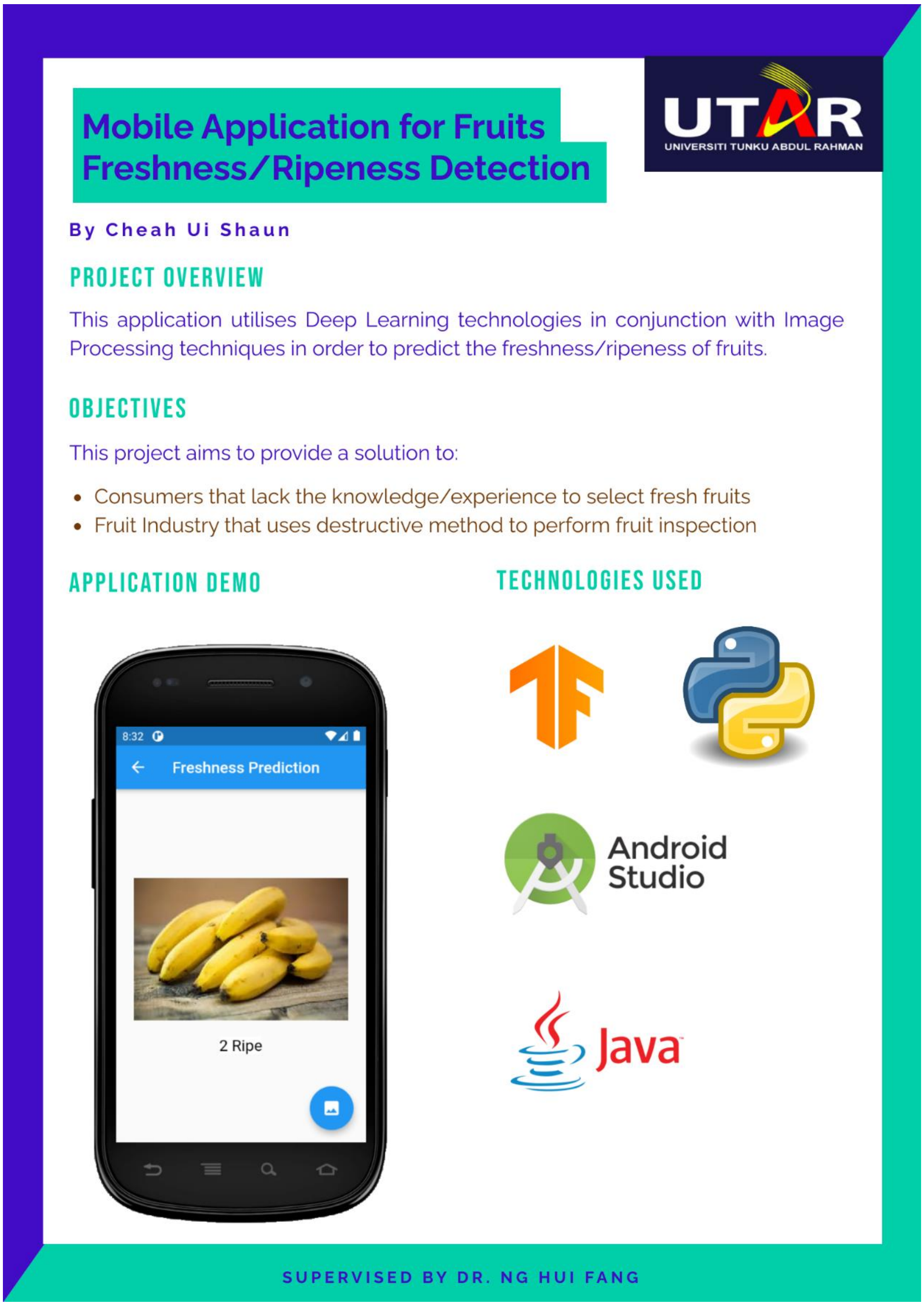
Pem, D. & Jeewon, R. 2015, “Fruits and Vegetable Intake: Benefits and Progress of Nutrition Education Interventions”, *Iran J Public Health,* vol. 44 no. 10 pp.1309-1321.

Rizam, M., Yasmin, A., Ihsan, M. and Shazana, K., 2009, “Non-destructive watermelon ripeness determination using image processing and artificial neural network (ANN)”, *International Journal of Electrical and Computer Engineering,* vol. 3 no. 2, pp. 332-336.

Saad, H., Ismail, A., Othmand, N., Jusoh, M., Naim, N. and Ahmad, N., 2009. “Recognizing the Ripeness of Bananas using Artificial Neural Network based on Histogram Approach”. *IEEE International Conference on Signal & Image Processing Applications (ICSPA09).* Kuala Lumpur, Malaysia. 18-19 November.

Silva, Eduardo A.B. & Mendonça, Gelson, V., 2005, “Digital Image Processing”, *The Electrical Engineering Handbook,* Academic Press, pp. 891-910.

Poster



Mobile Application for Fruits Freshness/Ripeness Detection

UNIVERSITI TUNKU ABDUL RAHMAN

By Cheah Ui Shaun

PROJECT OVERVIEW

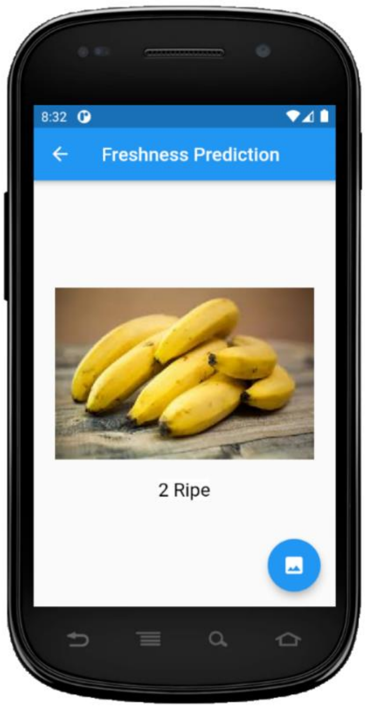
This application utilises Deep Learning technologies in conjunction with Image Processing techniques in order to predict the freshness/ripeness of fruits.

OBJECTIVES

This project aims to provide a solution to:

- Consumers that lack the knowledge/experience to select fresh fruits
- Fruit Industry that uses destructive method to perform fruit inspection

APPLICATION DEMO

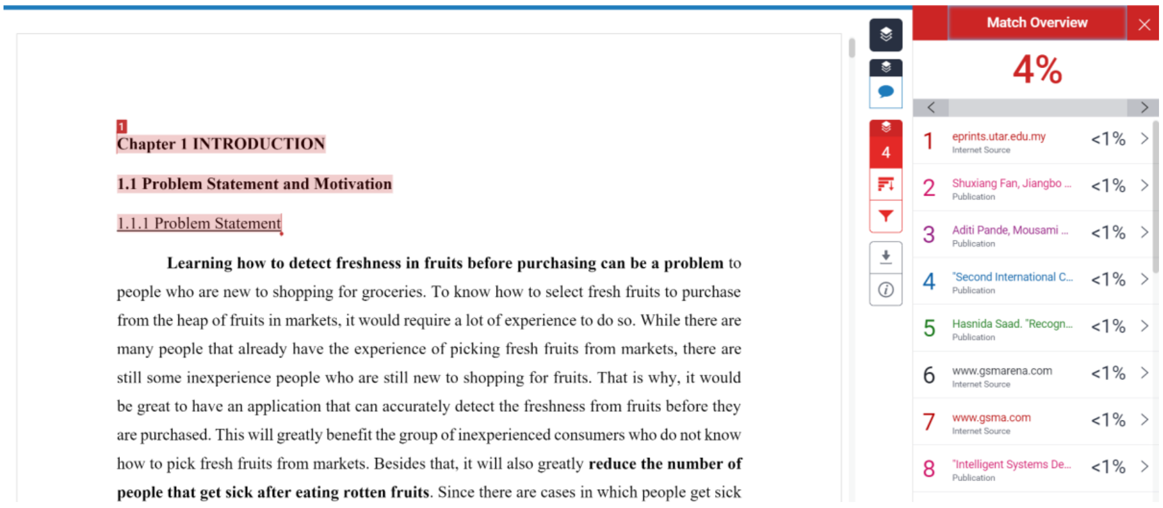


TECHNOLOGIES USED

Android Studio

Java

SUPERVISED BY DR. NG HUI FANG



Chapter 1 INTRODUCTION

1.1 Problem Statement and Motivation

1.1.1 Problem Statement

Learning how to detect freshness in fruits before purchasing can be a problem to people who are new to shopping for groceries. To know how to select fresh fruits to purchase from the heap of fruits in markets, it would require a lot of experience to do so. While there are many people that already have the experience of picking fresh fruits from markets, there are still some inexperience people who are still new to shopping for fruits. That is why, it would be great to have an application that can accurately detect the freshness from fruits before they are purchased. This will greatly benefit the group of inexperienced consumers who do not know how to pick fresh fruits from markets. Besides that, it will also greatly **reduce the number of people that get sick after eating rotten fruits**. Since there are cases in which people get sick

Match Overview

4%

1	eprints.utar.edu.my Internet Source	<1%
2	Shuxiang Fan, Jiangbo ... Publication	<1%
3	Aditi Pande, Mousami ... Publication	<1%
4	"Second International C... Publication	<1%
5	Hasnida Saad. "Recogn... Publication	<1%
6	www.gsmarena.com Internet Source	<1%
7	www.gsma.com Internet Source	<1%
8	"Intelligent Systems De... Publication	<1%

Final Year Project - Mobile Application for Fruits Ripeness Detection

ORIGINALITY REPORT

4%	2%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	<1 %
2	Shuxiang Fan, Jiangbo Li, Yunhe Zhang, Xi Tian, Qingyan Wang, Xin He, Chi Zhang, Wenqian Huang. "On line detection of defective apples using computer vision system combined with deep learning methods", Journal of Food Engineering, 2020 Publication	<1 %

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Cheah Ui Shaun
ID Number(s)	17ACB02900
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Mobile Application for Fruit Freshness/Ripeness Detection

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 4 % **Similarity by source** Internet Sources: 2 % Publications: 2 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

[illegible]

Signature of Supervisor

Name: Ng Hui Fuang

Date: 16-04-2021

Signature of Co-Supervisor

Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB02900
Student Name	Cheah Ui Shaun
Supervisor Name	Dr. Ng Hui Fuang

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	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)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

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

I, the author, have checked and confirmed all the items listed in the table are included in my report. Shaun ____________________ (Signature of Student) Date: 15.04.2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. __ [illegible] ___ (Signature of Supervisor) Date: 16-04-2021

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 1**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Developed banana freshness classification model using transfer learning

2. WORK TO BE DONE

-Attempt to change the classification problem to a regression one

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

-

[illegible] Shaun

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 3**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Show result of changing the problem domain to a regression one

2. WORK TO BE DONE

-Change the problem domain back to classification

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

-

[illegible] Shaun

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 5**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE [Please write the details of the work done in the last fortnight.] -Show result of the banana freshness classification model using MobileNetV3
2. WORK TO BE DONE -Collect mango and papaya dataset, then train respective fruit freshness models
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS -

[illegible]

Supervisor's signature

Shaun

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 7**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE [Please write the details of the work done in the last fortnight.] -Completed mango and papaya freshness classification model
2. WORK TO BE DONE -Integrating the models trained into the mobile application platform
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS -

[illegible] Shaun

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 9**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE [Please write the details of the work done in the last fortnight.] -Integrating the fruit freshness models to the mobile app
2. WORK TO BE DONE -Develop a fruit classification model that classify between different type of fruits
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS -

[illegible]

Supervisor's signature

Shaun

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 11**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE [Please write the details of the work done in the last fortnight.] -Integrate the fruit classification model to the mobile application
2. WORK TO BE DONE -Show more information to the user during inspection
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS -

[illegible]

Supervisor's signature

Shaun

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 13**
Student Name & ID: Cheah Ui Shaun 1702900	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Mobile Application for Fruits Freshness/Ripeness Detection	

1. WORK DONE [Please write the details of the work done in the last fortnight.] -Integrated all the models into the mobile platform
2. WORK TO BE DONE -Write the Report, Refine the application
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS -

[illegible] Shaun

Supervisor's signature

Student's signature