A Mobile Application for Food Quality Recognition.

BY

Connie Tang Ming Xin

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)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: A Mobile Application for Food Quality Recognition

Academic Session: May 2024

I CONNIE TANG MING XIN

(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,

Saw Seow Hui

(Author's signature) (Supervisor's signature)

Address:

Lot 6135, Lorong
Permata 11, Vista
Perdana Phase 2,
98000 Miri, Sarawak

(Ts Dr Saw Seow Hui)

Date: 11 September 2024 **Date**: 9/12/2024

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 11 September 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Connie Tang Ming Xin **(ID No:** 21ACB06403 **) has completed this final year project entitled** "A Mobile Application for Food Quality Recognition" **under the supervision of** Ts Dr Saw Seow Hui **(Supervisor) from the Department of** Computer Science , **Faculty of** Information and Communication .

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

Yours truly,

(Connie Tang Ming Xin)

DECLARATION OF ORIGINALITY

I declare that this report entitled **"A Mobile Application for Food Quality Recognition"** is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : Connie Tang Ming Xin

Date : 11/09/2024

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my supervisor, **Ts Dr Saw Seow Hui**, for providing me with the golden opportunity to work on the project of ‘A Mobile Application for Food Quality Recognition’. This project has been my initial step towards establishing myself in the field of food quality recognition application design. I cannot thank you enough for this.

I am deeply grateful to all my uni friends, for their unwavering patience, support, and love. Their presence during challenging times has been a pillar of strength for me. Lastly, I owe a debt of gratitude to my parents and family for their constant love, support, and encouragement throughout this journey. Their faith in me has been a continuous source of motivation.

ABSTRACT

This project seeks to design and develop a mobile application that has the ability to discern with quality of different foods in response to the emerging issue of food quality and its effects on health. One of them is that people have a common difficulty in evaluating the qualities of foods and thus suffer some adverse health effects and the act of food waste among the various generations especially the young ones. This project uses the concepts of computer vision, machine learning, and artifical intelligence to develop an application that rates the quality of foods; particularly fruits and vegetables. The approach entails collections of large datasets of images in food items and using AutoKeras and EfficientNet B7 towards creating models. After that, these models are incorporated into the mobile application interface to help the users to navigate through the app. The utilization of the application was most importantly in validating the application's capability to recognize food quality and the stage of ripeness of the fruits and vegetables. Thus, by way of revealing details about food quality, the app contributing to food safety. Another strength is its introduction of sophisticated technologies into a basic tool by also providing knowledge in a utilitarian approach. This mobile application can be considered as an innovative solution that goes beyond providing food safety and improving sustainability to acting as an informant that increases people's understanding of proper eating patterns and how to achieve better food sustainability. Therefore, this work can be indeed considered as a useful contribution to public health, environmental conservation and food security in general.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNNs	Convolutional neural networks
ResNets	Residual Networks
DenseNests	Dense Convolution Networks
RGB	Red, Green and Blue
GCH	Global Change Climate and History
CVV	Card Verification Value
BIC	Bayesian information Criterion
UNSER	Unsupervised Learning
LBP	Local Binary Pattern
ISADH	Improved Sum and Difference Histogram
MSVM	Multiclass Support Vector Machine
HSV	Hue, Saturation and Value
SVM	Support Vector Machines
KNN	K-Nearest Neighbors Algorithm
LSTM	Long Short-Term Memory
MLK	Multi-Kernel Learning Region
PLS	Partial Least Squares Regression
DNN	Deep Neural Networks
PCA	Principal Component Analysis
MLP	Multi-layer Perceptron
MEC	Mobile Edge Computing
APIs	Application Programming Interfaces
UI	User Interface

CHAPTER 1
Project Background

This chapter present the introduction (background information), problem statement and motivation, project scope and direction, project objectives and impact, significance and contributions that relate to the title “A Mobile Application for Food Quality Recognition”.

1.1 Introduction

With the advancement of technology, people can now easily have food delivered to their homes through various delivery services [1]. This has resulted in an increasing number of young people being unable to recognize if certain foods, such as fruits or vegetables, are spoiled. This is because they often receive food that has already been processed and rarely go to the supermarket to buy raw ingredients for cooking.

Additionally, most of them also prefer to eat out and pre-packaged meals [2]. As a result, the younger generation is also becoming more and more unable to cook [3]. This causes them to miss out on valuable recognition of some food quality due to cooking experience. This may eventually lead them to face food safety problems due to a lack of experience in recognizing food quality.

Therefore, food safety is a major factor affecting public health and the well-being of society [4].

1.1.1 Definition

1. **Mobile Application:**

   Mobile Application is a software program created expressly for use on tiny, wireless computing like smartphones and tablets [5]. Instead, they are designed for use on desktop or laptop computers [5].

2. **Food Quality Recognition:**

   Food Quality Recognition is an important method for ensuring food safety [6].

1.1.2 Mobile Application for Food Quality Recognition

Mobile Application for Food Quality Recognition is a software program that can be installed on smartphones or another mobile device to help users recognize the quality of food.

1.2 Problem Statement

The central challenge addressed by this project revolves around food quality recognition. However, many people are concerned about the quality and safety of the food they consume, but they often lack ability to recognize the quality of different types of food. This might cause health risks, such as food poisoning and allergies etc.

Therefore, there is a need for a mobile application that can help users recognize and assess the quality of food using various technologies, such as computer vision (image processing), machine learning, and artificial intelligence. This application can improve users' food recognition skills and enable them to make informed quality food choices.

1.3 Motivation

Recognizing food quality is not only about food safety, but it also reduces food waste. For example, consumers and capitalists expect their fruits and vegetables to be of perfect shape, size and color [5]. Therefore, fruits with the slightest imperfection or "ugly" fruits and vegetables are eliminated and discarded. But these fruits and vegetables are perfectly edible, and it is shameful that they should be discarded and wasted simply because they are substandard.

So, if people can accurately assess the freshness of food, they are less likely to discard it prematurely, thus contributing to a more sustainable consumption of food.

1.4 Project Scope and Direction

The scope of the project is centered on recognizing and assessing the quality of food.

1.4.1 Project Deliverables

1. **Mobile Application for Food Quality Recognition**

   The main deliverable of this project is a fully functional mobile application that runs on smartphones and other mobile devices. The application will serve as a tool for users to identify the quality of food including fruits, vegetables and etc.

2. **User documentation**

   In addition to the mobile application, comprehensive user documentation will be provided to guide users on how to effectively use the application. The documentation will include step-by-step instructions, troubleshooting guides and tips for maximising the utility of the application.

3. **Training dataset**

   A dataset will be created that includes a variety of food products with quality labels (e.g., Good, Bad, Level of ripeness, spoiled etc.) This dataset will be used as the basic for training machine learning models in the application.

1.5 Project Objectives

This subsection will outline the specific objectives of the project. These objectives will be aligned with the broader goal of creating food quality recognition mobile application.

1.5.1 Main Objective

The main objective of this project is to develop a mobile application that will enable users to assess the quality of food and promote food safety.

1.5.2 Sub-Objectives

The main objective will break down into 3 different sub-objectives, each contributing to the overall objective:

1. **Image Data Collection**

   Collect image datasets of food, categorised by quality attributes such as quality and ripeness.

2. **Model Development**

   Use the collected datasets to create and train machine learning models (e.g. AtuoKeras, CNNs, etc.) for food quality recognition.

3. **Mobile Application Development**

   Develop a user-friendly mobile application that integrates the trained models and provides real-time quality assessment.

1.6 Impact, Significance and Contributions

The mobile application for food quality recognition represents a significant innovation by integrating computer vision, machine learning, and mobile technology to improve food safety and reduce waste. It serves as a practical tool and educational resource for sustainable and informed food consumption.

1.6.1 Improve Food Safety

Through the application, it can greatly contribute to food safety by helping users identify potentially spoiled or ripe of fruits and vegetables. This can help in reducing the risk of foodborne illnesses like food poisoning, which can have serious health implications.

1.6.2 Educational tool

The app can be used as an educational tool by helping users understand what to look for in quality food and encouraging them to use more fresh, healthy ingredients in their cooking or eating. This knowledge can be extended beyond the app to help users make informed decisions when purchasing food.

In conclusion, this project, through food quality identification, not only promotes food safety and reduces waste but also educates the public about food quality. This can have a profound impact on public health, sustainable development, and society, making it a worthwhile and significant endeavor. It's not just an app; it's a step towards a healthier and more sustainable world.

1.7 Report Organization

This report is organized into 7 chapters. The first chapter is the introduction of this project, which includes the problem statement, motivation, project scope and direction, project objectives, impact, significance and contributions, and report organization.

While in the second chapter, the literature review on the food recognition, food quality recognition, quality recognition approaches have been conducted. Besides that, this chapter also included a review of the existing project for food recognition mobile applications, food quality recognition mobile application, and quality recognition mobile applications.

Next, the third chapter is discussing the overall system methodology/approach of this project. It conducts design specification and system design diagram which include system architecture diagram, use case diagram and activity diagram.

While chapter 4 of this report is about the system implemetation and design of this project. It presents what have been implemented into the system (e.g. dataset) and system flowchart.

In addition, chapter 5 is regarding the details of how to setting and configuration of the system setup and operation. Example, hardware and software setup.

Furthermore, the sixth chapter is about the evaluation and discussion of the system. Here will show all the testing and evaluation result of the models.

Lastly, chapter 7 which is the last chapter of this report is about the conclusion of this project, it included the project review and discussion, the novelties of this project and the recommendation that can improve this project in future.

CHAPTER 2
Literature Review

This chapter presents previous work on food quality recognition, the application of image processing in food, food recognition systems, real-time quality assurance of food and real-time mobile applications for food recognition. Also, their limitations and suggested solutions are also presented.

2.1 Using Machine Learning Approaches for Food Quality Detection (Jumming Han et al) [6]

2.1.1 Review Summary

Through this paper, food quality detection is an important method for ensuring food safety and reducing waste [6]. However, traditional methods for food quality detection are slow, costly and require professionals. For example, when faced with a dramatic increase in fruit production, the selection, transport and quality control of fruit has become an important issue [6]. The disadvantages of traditional methods will become more apparent at this point. Therefore, the aim of this paper is to find an efficient and non-destructive method to identify the quality of fruits. This is because fruit freshness detection is a hot and difficult research area, as fruits are high-value and nutritious foods that are prone to spoilage.

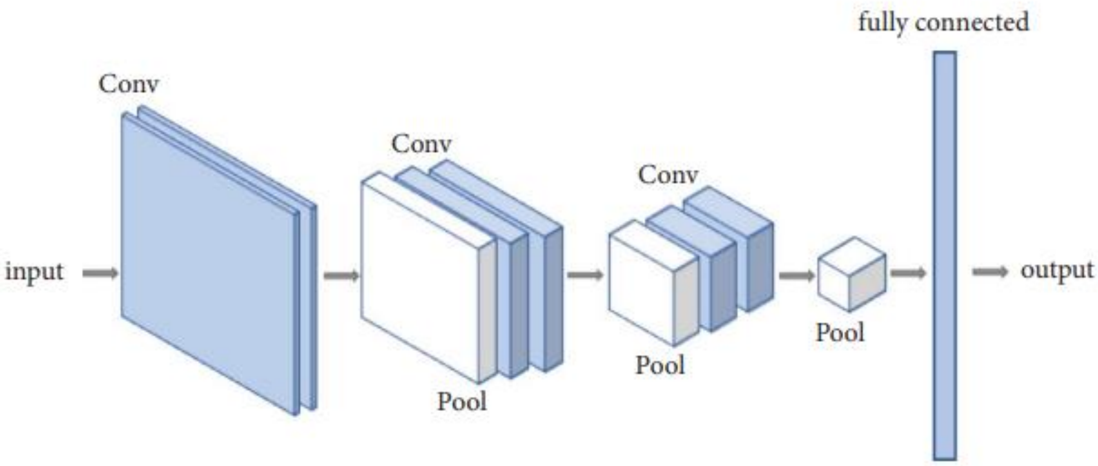


Figure 2.1.1 Typical architecture of CNN model [6].

For this issue, the paper proposes using machine learning algorithms (Convolutional neural networks - CNNs) to classify fruit freshness based on RGB images. CNN is a feed-forward neural network that contains convolutional computations with a deep structure [6]. In other words, CNN is a type of deep learning model that can extract features from images and perform regression or classification tasks [7]. This paper has introduced two improved versions of CNNs, which are "Deep Residual Networks (ResNets)" and "Dense Convolution Networks (DenseNets)".

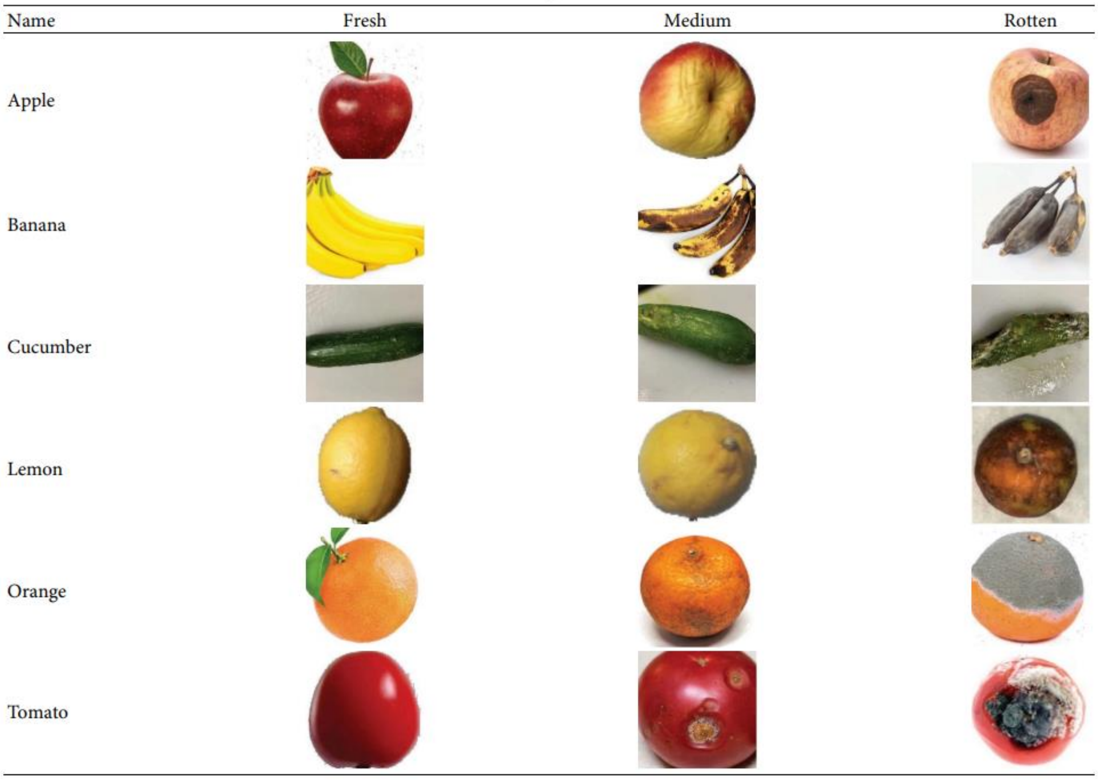


Figure 2.1.2 Example of datasets from paper [6].

Additionally, Figure 2.1.2 shows the dataset for experiments and testing use of this paper. The experiments and testing from this paper have used a large dataset from Kaggle (kaggle.com) and other websites that contain RGB images of six fruits (apples, banana, cucumber, lemon, orange and tomato) at three classification experiments (fresh, medium and rotten) and eighteen classification experiments ([fresh, medium and rotten] multiples [apple, banana, cucumber, lemon, orange and tomato]) [6].

Method	Classification number	Data types	Precision (%)	Recall (%)	F1 score (%)
ResNets	18	RGB image	95.69	94.72	95.07
DenseNets	18	RGB image	95.55	94.73	95

Table 2.1.1: Experimental and testing results from the paper [6].

The experimental and testing results show that ResNets and DenseNets achieve high precision, recall and F1 scores without overfitting. This is because ResNets and DenseNets use only RGB images as input data, whereas several other methods use hyperspectral, laser backscattering, or infrared video to improve the evaluation accuracy of the models [6].

2.1.2 Limitations and Solutions

From this paper, there has been mention a major limitation in CNNs. Which is in CNNs, a degradation problem is exposed when the deep network is able to start converging. This problem is that whenever the accuracy saturates as the depth of the network increases, it then degrades rapidly [8]. This problem is also known as degradation. Therefore, the paper used ResNets and DenseNets to solve this problem.

1. **ResNets:**

ResNets is an improved version of CNN developed by Kaiming He et al [8] for dealing with this degradation problem in CNN.

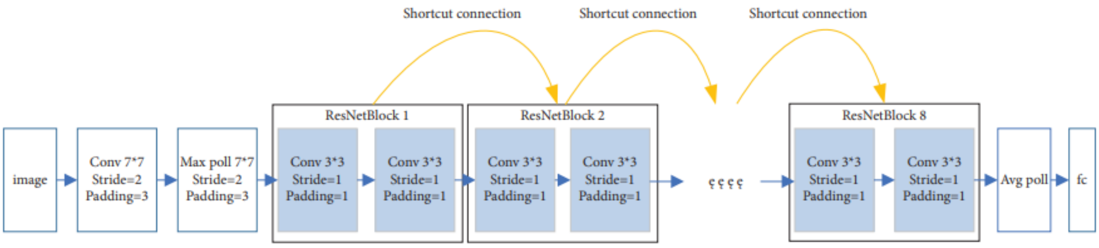


Figure 2.1.3 Shortcut connections in ResNets *[6].*

Figure 2.1.2 shows the joins achieved by simple identity mapping, whose output is added to the output of the stacked layers; which this pure identity mapping is used as a bypass. This means that with sufficient depth, ResNets can improve the performance and combine image features by summing them before they enter the layer [6].

(i) Weakness / Limitation of ResNets:

However, ResNets has weakness too, including complexity, vulnerability to overfitting and restricted interpretability [9].

2. DenseNets:

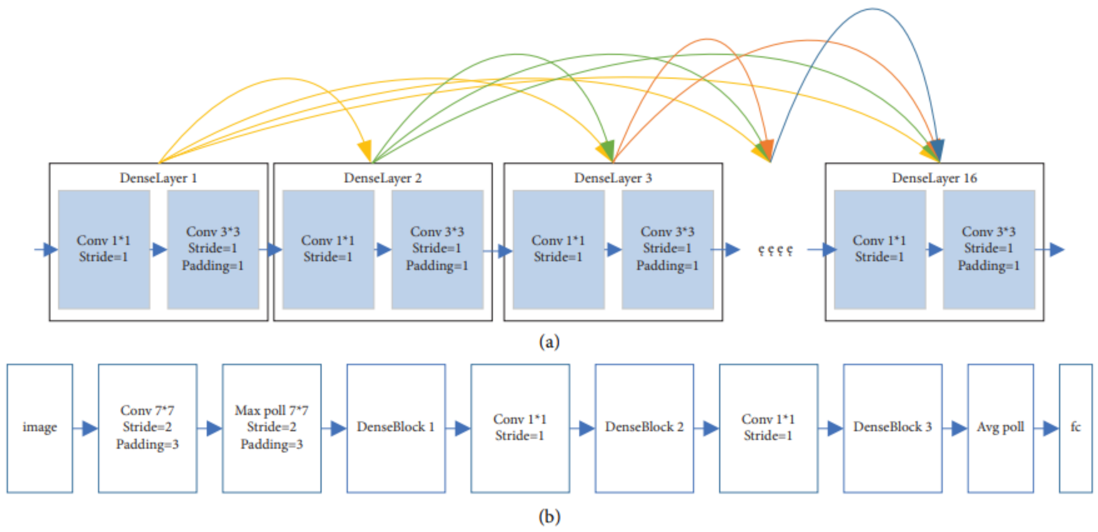


Figure 2.1.4 Concatenate in DenceNets, they are listed as (a) Concatenate is DenceNets, (b) Architecture of DenseNets [6].

If a convolutional network contains short connections between layers close to the input and layers close to the output, the depth, accuracy, and training efficiency of the convolutional network will be greatly improved. DenseNets is a model designed through this idea by Gao Huang et al [10]. Figure 2.1.3 shows that DenseNets use a feed-forward approach to concatenate feature maps learnt from different layers so that for each layer, the input feature maps are the outputs of the feature maps of the previous layers [6].

(ii) Weakness / Limitation of DenceNets:

However, DenceNets has a weakness too which is feature maps of each layer are spliced with the previous layer, and the data is replicated multiple times. The number of model parameters expands linearly with the number of network layers, eventually resulting in an exponential rise in compute and memory overhead during training [11].

2.1.3 Proposed Solutions

1. **ResNets**

The solution that can resolve this is transfer learning with a modified architecture. This is started with a pre-trained ResNets model as a base architecture, which benefits from the complexity and representation power learned on a large dataset like ImageNet. Then, modify and fine-tune this base model to suit your specific task while addressing the other limitations.

2. **DenseNets**

Implement a dense block compression technique to reduce the number of feature maps within each dense block. This can be achieved by applying dimensionality reduction techniques such as 1x1 convolutions or pooling operations to decrease the channel dimensions of feature maps. By reducing the number of feature maps within a dense block, you can maintain expressive power while controlling the model's growth in parameters [12].

2.2 Application of Image Processing in Fruit and Vegetable Analysis: A Review (Shiv Ram Dubey and Anand Singh Jalal) [13]

2.2.1 Review Summary

According to the paper, images are an essential source of data and information for digital image analysis and image processing technology [13]. However, the paper mentions that the existing dedicated imaging system on the market, where users can get results by pressing a few key buttons is not very versatile and mostly they are costly. Moreover, it is difficult for the user to understand how the results are generated due to a lack of basic knowledge of imaging systems.

Additionally, the paper also mentions that the recognition system is a grand challenge for the computer vision to achieve near-to-human levels of recognition [13]. This might be because image-based analysis of fruit and vegetables, such as variations in shape, color, texture, illumination, background, and defects. Therefore, there needs to trade-offs between accuracy, speed, and cost of different methods.

Consequently, this paper focuses on 3 areas, which are fruit and vegetable recognition and classification, fruit disease recognition and classification and the multi-class classification method.

1. **For fruit and vegetable recognition and classification,** the paper surveys the literature for the techniques used for recognizing and classifying different types of fruits and vegetables based on their color, texture, shape, size, density, and other features [13]. It also covers the application of image processing in food quality, safety, and nutrition assessment.

2. **For fruit disease recognition and classification,** the paper reviews the approaches proposed for detecting and identifying fruit diseases using images [13]. It also covers the use of advanced techniques such as thermal imaging, hyperspectral imaging, and multispectral imaging for detecting segmentation and classification.

3. **For the multi-class classification method,** the papers propose a method to classify fruits and vegetables into multiple categories using binary classifiers and unique IDs for each class [13]. It uses the "minimum distance" of the binary outcome to the class IDs as the criterion.

Lastly, the paper compared the different features and classifiers between two datasets, which are fruit and vegetable classification (15 categories and 2615 images) and fruit disease recognition (4 categories and 391 images). So, the paper compares them by using colour and texture features such as GCH, CVV, BIC, UNSER, LBP, CLBP and ISADH. Also, it uses SVM and KNN as the base learners for them [13].

1. **Using GCH, CVV, BIC, UNSER, LBP, CLBP and ISADH Features Considering MSVM as a Base Learner in the RGB and HSV Color Spaces:**

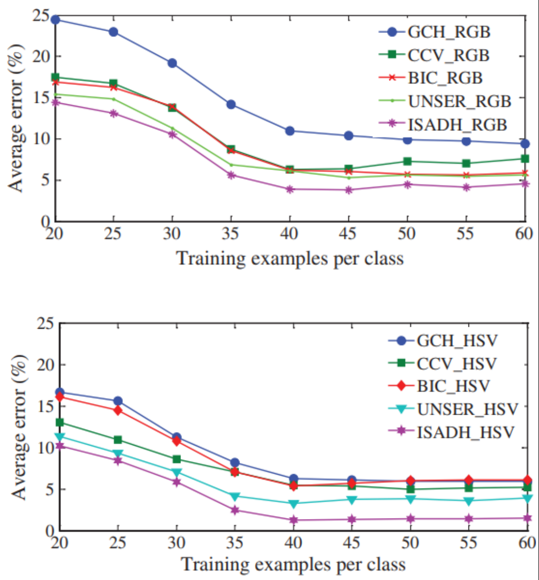


Figure 2.2.1: Average Fruit and Vegetable Classification Error using GCH, CVV, BIC, UNSER, LBP, CLBP and ISADH Features Considering MSVM as a Base Learner in the RGB and HSV Color Spaces [13].

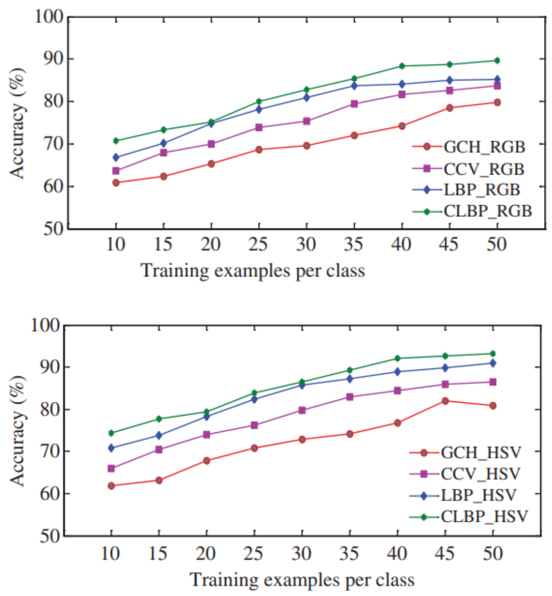


Figure 2.2.2: Apple Fruit Disease Average Classification Accuracy using GCH, CVV, BIC, UNSER, LBP, CLBP and ISADH Features Considering MSVM as a Base Learner in the RGB and HSV Color Spaces [13].

The experimental result above, it shows that the average fruit and vegetable classification error for RGB and HSV colour spaces are decreasing. While for apple fruit disease average classification accuracy for RGB and HSV colour spaces is increasing.

2. **Comparison of SVM and KNN Classifier using the ISADH Texture Feature in Both the RGB and HSV Color Spaces:**

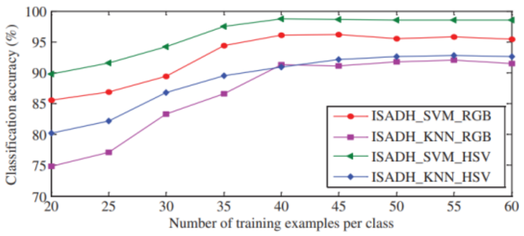


Figure 2.2.3: Comparison of SVM and KNN Classifier using the ISADH Texture Feature in Both the RGB and HSV Color Spaces for Fruit and Vegetable Classification Problem [13].

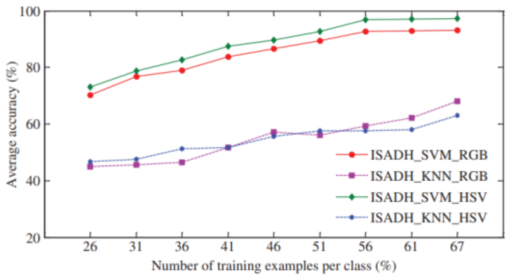


Figure 2.2.4: Comparison of SVM and KNN Classifier using the ISADH Texture Feature in Both the RGB and HSV Color Spaces for Fruit Disease Recognition Problem [13].

However, through a comparison of SVM and KNN classifiers results for fruit and vegetable classification are increasing by more than 85%. While fruit disease recognition is increasing only SVM results for RGB and HSV are more than 80%. Other than that, KNN results for RGB and HSV are only more than 40%.

2.2.2 Limitations and Solutions

1. Multiclass Recognition of Varied Fruits and Vegetables

Because of the input image may contain multiple varieties fruits or vegetables in any position and in any number [13]. Also, many fruits and vegetables vary significantly in shape, texture and colour depending on their ripeness. For example, from the paper, oranges can be green and yellow or mottled and brown [13]. Therefore, it may not be sufficient to use just one image feature to classify fruits and vegetables, so it is necessary to extract and combine those features that help to identify produce.

(i) Weakness / Limitation of combined features:

It will increase the computational complexity of the recognition system and may require significant computational resources. Also, it might overfit if combining a large number of features or using complex models without proper regularisation or validation. Therefore, this might reduce the ability of the model to generalise to new and unseen data.

2. Fruits and Vegetables in Plastic Bag

This means that for recognition it might need to add hue shifts and specular reflections while the fruits and vegetables might put inside a plastic bag [13]. Therefore, it needs to different classifiers that can produce different results. However, it needs settle the problem of choosing the type of classifiers.

(i) Weakness / Limitation of color correction:

Accurately correcting for hue shifts and specular reflections caused by plastic bags is a challenging task especially when the plastic bags are made of different materials.

3. Performance of Fruit Disease Recognition System

The paper mentions that its performance depends on defect segmentation, hence the need for accurate defect segmentation [13]. While the more training examples there are, the longer the training time of the system will be [13]. Therefore, the performance of the system is bound to be better when the system is trained using fewer training examples only.

(i) Weakness / Limitation of defect segmentation:

Developing accurate defect segmentation techniques can be complex can may require manual labelling of training data, which can be both time-consuming and expensive to obtain.

2.2.3 Proposed Solution

1. **Multiclass Recognition of Varied Fruits and Vegetables**

Deep learning models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks can be used for multiclass fruit and vegetable categorization [14].

2. **Fruits and Vegetables in Plastic Bag**

To identify fruits in plastic bags, a block classification-based technique has been suggested. Based on the findings of edge detection of R-G grayscale pictures, this approach segments original images into irregular chunks. Based on the colour and texture information that were retrieved from blocks [15], a support vector machine then divides these blocks into fruit blocks and non-fruit blocks.

3. **Performance of Fruit Disease Recognition System**

Use Semi-supervised learning approaches, where a small amount of data is manually labeled and the remainder is automatically classified based on the manually labeled data, can be used to accurately separate defects [16]. This can minimize training time, approaches like transfer learning may be utilized, where a pre-trained model is fine-tuned on the fresh data [16].

2.3 Study for Food Recognition System Using Deep Learning (Nareen O. M. Salim et al) [17]

2.3.1 Review Summary

The paper has mention that the techniques have been commonly used to detect food characteristics, including electronic noses, computer vision, spectroscopy, and spectral imaging [17]. A large amount of numerical data related to food characteristics can be collected by these methods. Therefore, it is important to analyse the data for these methods because the huge amount of data includes a lot of repetitive and irrelevant material [17]. However, to handle such a huge amount of data and extract useful features from the acquired data and how to apply these methods to the complexity of the real world is challenging [17]. Also, the paper mention that their device while recognises food in real time at the customer's end is without contact with external computer resources while the others' image recognition process takes place on a server, so it causes not allow for interactive operations due to contact delay [17].

Additionally, the paper also mentions that food recognition is a challenging puzzle because foods are diverse (sometimes they are different within the same group of foods) [17]. Therefore, the paper uses fine-grained image classification and multi-food image recognition system (Multi-Kernel Learning Region (MLK)) to identify colour, texture, gradient and SIFT extracted by multiple detectors. However, there are still limitations in placing food on a white plate and taking photographs of the food to calculate the quantity of food as well as using a checkerboard grid.

To handle this, the paper mentions that there is several data analysis methods (PLS, SVM, Random Forests and k-NN) have been developed for modelling and processing large amount of data [17]. So this paper is focusing on the concept and challenges of food recognition and reviews various techniques and application of deep learning in this field.

In addition, this paper uses datasets, algorithms, implementation systems, accuracy rates and significant goal satisfaction methods to analyse and compare successful results and commonalities of food recognition methods and presents them in tabular form. So through the table it can shows that the researcher relied on different datasets (Mangosteen Detection, Food 101, Indian Food, Food Dataset and etc.) rather than other dataset and CNN, DNN, SVM, PCA, MLP and KNN are the most used algorithms in this flied [17]. Therefore, the system that implements this paperwork is stall movement and computer system rather than other systems. However, the accuracy of the work of this paper is in the range of 70% to 100%.

In conclusion, by using these method and techniques, the paper has powerful structures, frameworks, and functions. However, the research of this paper is more oriented towards the field of modern food identification.

2.3.2 Limitations

1. Constraints on Food Placement and Photography

To improve the performance of real-time food recognition, it is recommended that user to place the food on white dishes and captured using a checkerboard [17]. This shows that the current system for food recognition and quantity estimation has constraints in terms of how the food is presented and photographed.

(i) Weakness / Limitation of solution using white dishes and captured using a checkerboard:

For some situations like there is no white dishes or user didn’t capture the food using a checkerboard, this can be cumbersome and may not always be practical for users. This limitation can impact the system’s usability and flexibility since it imposes specific conditions on users when they want to use the technology for food recognition and quantity estimation.

2. Server-Based Image Recognition

The paper mentions that many existing systems rely on performing image recognition processes on servers, which can introduce delays due to data transmission and processing [17]. This can hinder real-time and interactive usage of food recognition applications. Therefore, the paper mentions that identify food products in real-time on the customer side without require contact with external computer resources which allows customers to use it interactively [17].

(i) Weakness / Limitation of real-time on customer side without require contact with external computer resources:

However, customer side devices (smartphones or tablets) may have limited processing power and memory which can performing complex image recognition tasks in real-time can consume a lot of resources, potentially leading to performance degradation or even crashes on less capable devices.

2.3.3 Proposed Solution

1. **Constraints on Food Placement and Photography**

Using “image segmentation” algorithms to isolate the food from the backdrop and estimate its volume and calories independent of the dish's colour or pattern is one way to circumvent the restrictions on food placement and photography. Utilizing interactive segmentation techniques, which enable users to manually modify the food zone and fix any issues with the automated segmentation, is an additional remedy [18], [19].

2. **Server-Based Image Recognition**

Using Mobile Edge Computing (MEC) methods is one way to address the flaw or restriction of real-time image recognition on customer-side devices. MEC allows for dispersed data processing close to the source, which lowers latency and bandwidth usage [20]. Another option is to apply deep neural network "model compression" techniques to scale down their size and complexity, making them more appropriate for portable devices [21]. These solutions can fulfil the need for internet picture transmission, enhance real-time image identification precision, and ease core network load [20].

2.4 Food Handlers' Food Safety Knowledge, Attitudes, and Practices in Taman Negara, Kuala Tahan [22]

2.4.1 Review Summary

The place of study, hosted in Malaysia, at the famous tourist location of Taman Negara, centralized on food safety knowledge, attitude and practices of food handlers. The paper highlighted the pivotal role of food handlers in providing clean, safe food and decreasing the rate of foodborne illnesses.

Key findings of the study include:

- **Food Safety Knowledge:** The Food handlers' knowledge of Food safety and Hygiene in Kuala Tanah was good, with a mean score of over 4.50 [22]. This shows that food handlers in these groups strongly concur that food handling must comply with health standards [22]. The results of this research indicate that this literacy is important to guarantee the protection of the food that catered to the public, especially in those places like tourist attractions.
- **Hygiene practices:** The investigation identified food handlers' hygienic practices and established that staff generally observed Good Hygiene Practices (GHP). This included cases of proper hand washing, and food preparation areas that were safe from pests. Because of this, the incidences of diseases and famine were lessened. The study emphasized on the fact that things could be done to further clean the kitchen floors alongside the fact that they were clean.
- **Training and certification:** The study showed that the food handlers with certification had better food safety knowledge in comparison to those who did not possess the certification. The findings show that certification programs influence the engagement of food handlers which involves continuous education and training and skill development.
- **Data analysis:** The discussion was conducted with the use of SPSS IBM version 22 [22]. The development of such a software helped the researcher to interpret the gathered data on the food handlers correctly and, eventually, derive useful conclusions.
- **Room for Improvement:** Students can use flash cards to go over the vocabulary words and definitions that were taught in the class. This will allow the vocabulary to become part of students' language and help them remember it for a longer time. Therefore, this research underlines the importance of awareness and supervision to keep the level of food safety at the highest level may work.

In conclusion, this study concludes that although food handlers in Kuala Lumpur Mahanta Manigala generally have good food safety knowledge and practices, there is still a need for improvement in certain areas. The study emphasised the importance of continuous education and supervision in maintaining high standards of food safety [22]. The study also highlighted the role of certification schemes in equipping food handlers with the necessary knowledge and skills to ensure food safety. The findings of this study are particularly important given the region's popularity as a tourist destination.

2.4.2 Limitations and Proposed Solutions

1. **Limited Scope of Training Content**

   The findings of the study indicate that the training materials given by the Ministry of Health are quite generic and may not be able to specifically voice out the situation that they are facing [22]. Various individuals who migrate are born in other locations; they have attended different educational institutions; and some of them not familiar with the language of the location where they reside. Currently, a 3-hour training is widely used to train the entire team across food handlers in the respective areas of concern.

   - **Weakness / Limitation**: The current training material only cater for a general approach, and hence, it may fail to address the food handlers' specific area of interest, causing both inadequate understanding and different interpretation by them. This could prove to be a problem for them if the said information does not correspond to their concept or does not make much sense to them due to their level of education or English language proficiency resulting in a gap in the essential knowledge of food safety that they should have.
   - **Proposed Solution**: In order to deal with this challenge, the first step should include designing highly specialized training schemes. These programs should be structured following an assessment of food safety knowledge, attitude and practice and it should be inclusive of all demographics, for instance, young, old, churchgoing, female, poorly educated and the poor [22]. Individual training courses catering to understanding and applying food safety policies by food handlers will enhance the overall situation.

2. Lack of Comprehensive Waste Management Systems

The study points out that the restaurants at Taman Negara have inadequate solid waste disposal systems which lead to disposed waste ending in rivers. This is a two-fold challenge because it means that the residents of the settlement now have to endure their waste contaminating the river that they depend on in terms of food, water, and livelihoods [22].

- **Weakness / Limitation**: With an inefficient waste management system, our environment will be prone to pollution, and can also affect the cleanliness and hygiene standards restaurants will be conducting their operations. The neglect may bring wide range of health impacts to the aquatic environment and to the people who regularly visit the restaurants.
- **Proposed Solution**: It is vital that these restaurants develop and implement a good waste strategy which will undoubtedly play a key role. In this interim they need to make up their minds on what actions to take which should be aimed on limiting any possible negative consequences. It entails that no waste flows into the river and that the established hygiene routes are adhered to as regards the cleanness in a dining area.

2.5 Development of Safety Assessment System for Food Premises [23]

2.5.1 Review Summary

Carried out by the servicemen from Universiti Pahang Malaysia and other local institutions, the paper canters on the design of a safe workplace for foodstuffs service [23]. It reminisces the need of food safety and the magnitude of food handlers' involvement in preventing foodborne illnesses [23].

Key findings of the study include:

- **Food Safety Concerns**: The work highlights the fact that safety in the food industry is a worldwide issue, and it is estimated by the WHO to be connected to 300 million hospitalizations and 420 thousand deaths resultant from contaminated food. It shows the key role of food-workers in the practice of food safety that cross the production, processing, storage, and preparation system [23].
- **Safety Assessment System:** The research will produce a safety system that tests samples of food premises using an excel sheet that will be used to establish its reliability. The system is expected to bring efficiency in auditing food premises. This system works to assist in the ergonomic determination of the respective scores and grades, which enhances the recording and recovering easiness. [23].
- **Usability Testing**: Survey with the System Usability Scale (SUS) among expert panels and food premise owners conducted which scored 69.6 which is greater than the average. This suggests that the system is usable and is likely to meet its intended purpose.
- **Recommendations for Improvement**: The study proves a thumbs up but at the same time, it shows that there is still a long way to go when aiming at optimizing the system to secure better success.

In conclusion, the study successfully develops and validates a new safety assessment system for food premises, which could potentially streamline the assessment process and contribute to higher food safety awareness and standards [23]. The study also calls for continuous improvement and user feedback to refine the system.

2.5.2 Limitations and Proposed Solutions

1. **Limited Training Content Adaptability**

'Plating' refers to a concept or two involving coming up with food for the photographic purposes of instruction or promotional stuff. These restrictions may result from unmet conditions such as the positioning of the camera, lighting, or the deduction to adhere to guidelines that make the food look tasty and reflect the exact dish. Limiting the power of these instruments may cause their beneficial properties to be lost. Ultimately, it is better to find ways to overcome them and at the achieving of the aims.

- **Weakness / Limitation**: There is no aspect of training related to food quality recognition in the current setting that fit users from different demographic settings in the society. It would be a failure to consider user's individual needs, since the skills conveyed may simply not be appropriate or fit to the context (rate of learning, cultural).
- **Proposed Solution**: Therefore, it is recommended to establish a system of lessons for individual training purposes. The courses enabled by such initiatives must provide tutoring and academic support that caters to different user profiles through provision of the necessary materials, differentiated techniques and multiple languages. The objective is to draw users' attention to all the necessary materials to help them conceptualize and practically apply safety food measures.

2. **Constraints on Food Placement and Photography**

These occasions represent the confinements of whenever you are working with food for photos of that you would like to put in in your training or promotional material. The lack of the ability to precisely reproduce the colour, texture, and the appearance of the food may be caused by the setting of equipment, lighting conditions, or adherence to guidelines to make the food look attractive and reflect the recipe. Overcoming the mentioned disadvantages is the key to creating visual aid systems that are also both meaningful and visually appealing.

- **Weakness / Limitation**: A specified system has to be used for the food image recognition purpose and it requires serving food with white dishes and placing the forms in the checkerboard for the system to be able to process the food. There are cases where users are denied access to some public zones just to comply with the system's conditions, effectively setting a limit to the applicability of the system [23].

- **Proposed Solution**: The paper presents the universal solution on how to overcome this issue by implementing state-of-the-art image segmentation techniques. The purpose of this approach will be to distinguish the food items from the reference frame to transparently give exact volumes and calories without the influence of the item's colour or pattern. On the other hand, interactivities could be applied as a derivative technique of enabling users to correct recognized selections, thus ensuring higher precision [23].

3. Server-Based Image Recognition

Server-based image recognition involves processing and analysing images on a central server rather than a local device. This method is typically used in applications that require powerful computational resources to recognise complex patterns or details in an image. It is a common approach in situations where real-time analysis is not important or where the device capturing the image has limited computational power.

- **Weakness / Limitation**: The reliance on server-based image recognition in existing systems can cause delays, affecting the system's ability to provide real-time feedback and hindering its practical use.
- **Proposed Solution**: The research recommends the adoption of Mobile Edge Computing (MEC). MEC processes data closer to the data source, which can significantly reduce latency and bandwidth usage. Furthermore, the paper suggests applying model compression techniques to deep neural networks, making them more efficient and suitable for deployment on mobile devices, thus facilitating faster and more reliable image recognition.

2.6 Smart Supervision for Food Safety in Food Service Establishments in China: Challenges and Solutions [24]

2.6.1 Review Summary

The paper furthered micro-issues of food safety in FSEs in China and demonstrated barriers in policy implementation among the control of the government. A new 'smart supervision,' method to solve the problem will be proposed. Study brings these facts to the fore on account of worldwide occurrence of foodborne diseases and implementing health policies suitable for food safety. [24].

Key findings of the study include:

- **Food Safety Policies**: The article will focus on how China seeks to achieve this noble goal by implementing policies such as scheduled inspection and criteria for grading that are risk based. Among the benefits that this policy report cites are the poor implementation of these policies and ascribe it to factors like poor monitoring plus government officials’ turnover matters.
- Smart Supervision Solution: Facing these problems, there occurs a smart supervision system based on the web and 2D barcode technique [24]. This system is devised with three main objectives to wit consecutively – quickening the implementation process, reducing workload of an individual and facilitating proper policies implementation.
- **Pilot Application:** As a trial supervision system in Jilin province, this model has worked, providing highly positive impacts on governance in social perspective and a more competent regulatory activity [24].
- **Implications for Policy Implementation**: The results indicate that the incorporation of the technology in food safety supervision can play a critical role in implementing the policies effectively The research, too, provides the reasons for government backing homes and also demand a stable funding to sustain project work.

In conclusion, the study provides valuable insights into the obstacles hindering effective food safety supervision in China and offers a technologically driven solution to overcome these challenges. It serves as a reference for other regions and governments seeking to implement similar food safety policies, stressing the importance of innovation, government backing, and stakeholder involvement in ensuring food safety [24]. The paper’s implications extend beyond China, offering a framework for global application in the pursuit of reducing foodborne illnesses.

2.6.2 Limitations and Solutions

1. **Heavy Individual Workload and High Turnover Rate of FSEs**

   The study pinpoints that one of the main obstacles to policy implementation is overload of inspectors and high fluctuation level of FSEs tour visits. This in turn confounds the task of inspectors to remain on top of the compliance obligations.

- **Weakness / Limitation**: The individual workload can increase significantly to an extent of making the inspectors become mentally and physically tired which might make them inefficient. As another point to consider, the astronomical rates at which FSEs recede the field adds as an additional complication as regulators find it difficult to observe the landscape of FSEs since the scene is always changing.
- **Proposed Solution:** To confront the work force and labor drain, the study proposes the introduction of the smart overseer systems. Two 2D barcoding on mobile Internet would be employed for the inspection system in which the inspection process is simplified [24]. It might potentially reduce the workload by functioning as a performance enhancement tool as well as a method of developing employees through automating some tasks and keeping records of the decrease in the turnover rates.

2. **Lack of a Monitoring and Evaluation System**

Furthermore, another shortcoming is unveiled in a monitoring and evaluation system without proper robustness. This interrupts tracing and monitoring food safety measures. It thus becomes difficult to estimate the policy's performance.

- **Weakness / Limitation**: Without having a monitoring and evaluation process, it s hard to say whether their policies were a hit or just a miss, and also to know to what areas for improvement. Feedback is the process that conveys feedback from consumer and therefore this mechanism is neglected that resulted in continuous practices which may be not effective as to the assurance of food safety.
- **Proposed Solution:** The development of the indicators and ability to monitor and control the process are the main aspects of implementing this system into the smart supervision framework. That would also entail the designing of a digital tool for conducting surveillance over the compliance of this measure and the performance of food safety rules in a real-time mode.

2.7 Fruit Quality Recognition using Deep Learning Algorithm [25]

2.7.1 Review Summary

Having the researchers from the College of Electronics and Telecommunication at MITWPU in Pune, India as the study co-Author, this study has designed a model that uses deep learning techniques for the purpose of fruit quality recognition [25]. According to the study, factory fruit sorting automation is a useful method for industrial settings to raise productivity and achieve higher accuracy levels.

Key findings of the study include:

- **Deep Learning Application:** Fruit classification model was successfully organized along CNN lines showing clearly visibly performance [25]. The model obtained 95% accuracy after applying of 50 epochs to the fruits 360 dataset for training it and which has 131 different fruit and vegetable classes [25].
- **Fruit Classification**: The model had, indeed, made it very easy to sort fruits by choices of whether they are: good, raw, damaged. This showed how deep learning could be used to automate the sorting processes in industrial applications.
- **Dataset Utilization:** Showing the ability of the CNN for image recognition based on a broad dataset of data underlines its significance for correct fruit classification because of the complexity of its nature.
- **Automation in Agriculture:** Research has shown the reason for using automation in agriculture, especially developing country, which are mostly manual during pre-harvest operations and post-harvest stages.

In conclusion, the paper presents a promising approach to fruit quality recognition using deep learning, which could significantly benefit the agricultural industry by enhancing the sorting and grading processes. The study's findings are particularly relevant for countries with labour-intensive agricultural practices, offering a technological solution to improve quality control and efficiency.

2.7.2 Limitations and Solutions

1. Limited Dataset Variability

This restriction is with the study only because of the size of the dataset and not the general situation. In this case, the model was trained on a dataset that only included three categories of fruits: Hollywood, on the other hand, usually puts more emphasis on spectacle, speed, and visual effects. Therefore, there is a likelihood that the extracted key features might not be those that are traditionally considered to be the most important for discerning the fruity character of wine. As an example, many fruits carry a lot of cosmetic flaws such as bruises, being discoloured or infected by a disease, which are not considered by the three categories. Subsequently, the model would struggle in terms of classification when presented with fruits that were not in the mode’s training dataset. The suggested approach is integrating the variety of the dataset that contemplates the fruit types and that account those conditions for the fruits in its broader spectrum. This depends on this will help to enhance the model performance and its capacity to generalize.

- **Weakness / Limitation:** The study utilized the fruits 360 dataset that includes 131 different fruit and vegetable varieties [25] to practically bring the different combinations of red, yellow, and green colors to the screen. However, the model was trained only on three categories of fruits: genuine, adorable, and wearable pieces of clothes [25]. On the downside of this narrow case there may not be the necessary range to accurately represent the broad spread of fruit conditions encountered in real life.
- **Proposed Solution:** For future research into the model's robustness and applicability, consideration must involve utilization of a wider range of data that consists of more variety and includes different fruit conditions and types. This will in part support the models precision and generalization.

2. Model Generalization

Model generalization is the focal area of machine learning. Here a model is said to show good generalization capability when it performs efficiently in predicting new unseen data. In this study, the deep learning model performed very well in predicting the class of a given fruit, when it reached the 95% accuracy, however, the model's generalization to fruits it was not trained on is still not clear. Moreover, the research fails to provide information on how it will perform in areas where unseen fruit types or conditions are present which is an important detail in real-world applications. Thus, suppose the model was produced based on the photos of apple, orange, and banana. It may have not been as effective when faced with images of pineapple and mango. However a modeling that achieves such performance on training data only faces a usual problem in machine learning known as overfitting, where a model performs well on the training set but gives a poor result on new, unseen data. The suggested solution is to execute the next tests and studies with both fruits that may not have been included in the training dataset after the tests that only included those fruits presented on the training dataset are done. This could involve multi-source data validation or the actual testing to make sure the model retain performance while trying to categorize fruits of different classes.

- **Weakness / Limitation:** Although the model demonstrated an accuracy of 95% in predicting the shelf life of the fruits that it was trained on, the level at which it will generalize to other fruits that were not part of the training set could not be proven. The unknown response of the device to various fruit types and forming conditions is missed, what is necessary for a practical application.
- **Proposed Solution:** Lastly, this validation must be done on greater number of fruits, this includes fruits that are not present in the training set. This could involve such as a cross-validation technique or original simulation in order to check the model's ability as far as a diversity of fruit classes is concerned.

2.8 Pattern Recognition Techniques In Food Quality And Authenticity: A Guide On How To Process Multivariate Data In Food Analysis [26]

2.8.1 Review Summary

Given by a group of researchers from all over the world, the study represents a compendium of basic principles and algorithms of chemometric tools of pattern recognition applied in food analysis for the purposes of quality evaluation and authenticity determination. The paper advances the position that food authenticity and safety can be guaranteed only through the practice of these techniques.

Key findings of the study include:

- **Food Integrity:** Study deal with the food integrity term in which quality, safety and adulteration regards to food commodities are included [26]. The concept of food security is crucial in terms of realizing the product's adherence of labels on quality, including a declaration of desired constituents and omission of unwanted substances [26].
- **Analytical Techniques:** Some of the analytical methods and acquisition principles of food analysis are addressed in the article. Traditionally, these methods are divided into two groups: spectroscopy, separation techniques, sensors, and image-based methods. They are key as means of getting multivariable data useful in morphology.
- **Chemometric Tools:** The study focusses on the main pattern recognition chemometric techniques which include Principal Component Analysis (PCA) and Hierarchical Cluster Analysis (HCA) that have been used to accomplice the differentiation and interpretation of the complex food analyzes.
- **Applications and Advantages:** For simpler explanation, multivariate data processing in food analysis is explained with practical examples and the advantages and disadvantages of this method are also presented [26]. Such registration of the certain parameters occurs against the background of databases like the EU-Wine one to detect the natural variances within the determined qualities of a commodity [26].

In conclusion, the study asserts that the application of pattern recognition techniques in food analysis is indispensable for ensuring food quality and authenticity [26]. It calls for continuous improvement and development of analytical methods and protocols to keep up with the evolving challenges in food integrity assessment.

2.8.2 Limitations and Solutions

1. Complexity of Food Systems

The food systems consist of the stages from food formation until its utilization, consisting of growing, gathering, processing, packing, transporting, selling, and wasting it. This is due to complexity of systems because of the vast extent of possibilities that include poor handling and mismanaging at any stage. Promoting commendable food quality, which refers to quality, safety, and prevention of misbranding of the foodstuffs is difficult due to this elusiveness [26]. It's crucial for the product to adhere to its label in terms of quality, including the presence of specified constituents and the absence of undisclosed substances [26].

- **Weakness / Limitation:** This article brings to light the complexity of food systems. Moreover, it elaborates how difficult it is to maintain the authenticity of food due to the fact the almost everything can be tampered with or adulterated. This complication within this subject matter is somehow making the designers of these analyses verily difficult to be consistent in their work concerning all aspects of the quality and safety of food products.
- **Proposed Solution:** Such lines may encourage the usage of advanced analytical units and multivariate data modeling in order to get better monitoring systems for food safety. It does not omit the note that a comprehensive multifaceted health of food systems should be achieved by means of cooperation of different approaches.

2. Data Interpretation Challenges

Analytical techniques applied in the modern food realm generate massive volumes of data (input parameter, processing, and output statistics). Understanding these datasets can become hard though you get no help from chemometrics which do statistical and mathematical work related to chemistry. The tools described above make the task of separating the wheat from the chaff of intricate data much easier, thus progressing the calibration of food safety and genuineness. While the instruments make the data collection simple, the tools require experience, and one has to be knowledgeable about data interpretation, creating a problem.

- **Weakness / Limitation:** Furthermore, the article underscores that the comprehensive analytics yielding massive data outputs may not be understood without the support of chemometrics—the right combination of statistical mathematical tools to help sum up the data.
- **Proposed Solution:** It is proposed to fight this issue with the use of pattern recognition chemometric techniques; these approaches can distinguish and interpret the tough data resulting in improved examination of food's quality and authenticity.

2.9 Recognition of Food Material and Measurement of Quality using YOLO and WLD-SVM [27]

2.9.1 Review Summary

2.9.2 Implemented by engineers from the Politeknik Elektronika Negeri Surabaya, the study will report on the development of the system [27]. to differentiate types of food materials and detect their quality and defects visually via camera. The system makes use of the YOLOv3-tiny for detection and which is a combination of Weber Local Descriptor (WLD) features and Support Vector Machine (SVM) for quality assessments [27].

Key findings of the study include:

- **Food Material Recognition:** The approach got the score of 82.02% that was used by the YOLOv3-tiny [27] in object detection [27].
- **Quality Measurement:** The realization of good food quality measure is 83.33 % by WLD-SVM which is a strong indication of the efficiency and effectiveness of the system in classifying foods as fresh or not.
- **Technological Advancement:** The work confirms the use of computer vision and artificial intelligence, a combination that represents a viable and economical approach for food quality determination.
- **Practical Implications:** The system is capable of identifying, recognizing and evaluating the quantities of different food products simultaneously. This fact makes it beneficial, as being used for food safety as well as quality assessments in different places [27].

In conclusion, the paper details the creation of an innovative system that combines food material recognition with quality measurement [27]. The system's high accuracy rates demonstrate its potential utility in food safety applications, although there is room for improvement in detection accuracy and quality differentiation. The study underscores the importance of continuous development in the field of computer vision and machine learning to enhance food quality assessment methods.

2.9.3 Limitations and Solutions

1. Inconsistent Detection Accuracy

Furthermore, identifying the true nature of food ingredients through the use of a (such) system is subject to loss in accuracy. The system was implemented on a dataset resulting in an assurance rate of 82.02%. Nevertheless, in the actual field work, the precision is likely to be affected by different factors such as the kind of lighting in that environment, colours and level of shadows present [27]. This kind of erratic condition might be caused by the system which performs the analysis process and might lead to the deterioration of the detection accuracy.

- **Weakness / Limitation:** Precision of system in detection of food items also fluctuates, and it can detect them with altogether 82.02% of accuracy. Consistency was disrupted due to varied environments between trained model and real-world condition like lighting, object colors or shadows [27].
- **Proposed Solution:** Next up, the dataset could be upgraded to accommodate more realistic variations and fashion stronger algorithms that could safeguard the model against different environmental conditions.

2. Mixed Texture Descriptor Interference

The problem emerges when one Weber local descriptor (WLD) feature is resigned in combination with SVM to be useful for the evaluation of all meal types. The texture classifier based on the WLD-SVM model aims to represent the textures of broad variety food items. Yet the model that is frequently applied to different types of food may be the root of the problem. Therefore, the model could mistakenly classify fresh and unfresh parts coming from water chemistry, or the inside of food into varied samples, with the accuracy shown on the results. This interference is termed as property irrelevant mixed texture.

- **Weakness / Limitation:** System is based on multiplied WLD-SVM model, this leads to mixing up when pointing qualities of food substance. These causes provide an impossibility to distinguish between newly scanned and past scanned images, which reduce the quality of scanner's classification [27].
- **Proposed Solution:** To ensure that the model is able to deal with all food types a separate WDL-SVM model for each food type could be developed, or the existing model could be fine-tuned so that it can distinguish similar textures of different food materials.

2.10 A Review of the Discriminant Analysis Methods for Food Quality Based On Near-Infrared Spectroscopy and Pattern Recognition [28]

2.10.1 Review Summary

The article presents a critical review of discriminant analysis techniques including NIRS (Near-Infrared Spectroscopy) and pattern recognition technologies in food quality assessment [28]. It reveals the importance of these procedures as one of the non-destructive food quality discrimination methods, which is very critical for the food safety and television screens.

Key findings of the study include:

- **Food Safety and Quality:** This review shows the relevance of that NIRS combined with a pattern recognition as a disruptive method of the food quality process [28]. This phylogenetic approach avoids intellectual hardship, is accurate, and does not damage the samples, unlike traditional methods.
- **Technological Advancements:** It talks about the history of NIRS technology such as a near-infrared hyperspectral imaging that is the recent development (NIRHSI). This thing can detect chemical and spatial information from different substances in one sample [28].
- **Methodological Insights:** The article explores techniques used for pre-processing, traditional recognition techniques, and deep learning methods (Convolution Neural Networks, the abbreviation of CNN) that help enhance food quality analysis [28].
- **Future Directions:** The paper analyzes the limitations and advocates for future studies, highlighting the role of pattern recognition in addressing those limitations. The problems of NIRS-based food discrimination are solved with the use of pattern recognition.

In conclusion, the paper presents a detailed analysis of the current state and advancements in NIRS and pattern recognition for food quality assessment, pointing out the benefits and areas for future improvement in this field.

2.10.2 Limitations and Solutions

1. **Limited Data Preprocessing Techniques**

Referring to the case of an NIRS (Near-Infrared Spectroscopy), data preparation techniques are utilized for cleaning and processing the spectral data to make it ready for the purpose of further analysis. These techniques can comprise element disparities, baseline correction, normalization and more. Conversely, underlying technologies capable against never ever may not very well work for all interferences data can be in the data, such as instrument noise or environmental conditions. This is pulling down the accuracy of the predictive methods which depend on the variable selected for food quality judgment by this discriminant analysis.

- **Weakness / Limitation:** The paper discusses that the noise that is coming from the NIRS device itself, cannot be eliminated and this will in turn effect how the modeling is done. This means poor stress reduction of the current approach and may not be able to solve the nuisance of background noise completely.
- **Proposed Solution:** Future studies could focus on developing advanced preprocessing methods that are more effective in eliminating noise and other uncertainties from the spectral data, thus enhancing the robustness of the NIRS-based discriminant models.

2. Challenges in Pattern Recognition

Pattern widely recognized is a process of distinguishing trends and repetitions from data. It is one of the fundamental principles as well as one of the most used techniques in non-invasive reflectance spectroscopy-based quality analysis of food items [28]. Even though this is an advantage, there is a lot of ground to cover in this domain. The principle problem is the so- called "Curse of dimensionality", which appears to be a disguise of the complexity and difficulties that is to process the high-dimensional data, encountered with the NIRS spectral data. In addition to this, pitfalls such as the models fitting the training data only and not performing well on unseen data (also known as overfitting) and the need to use documentation reduction methods which will decrease the dimensionality of the data play an important role.

- **Weakness / Limitation:** The review states the “curses of dimensionality” as a challenge in chemometrics due to the confounding measurement of thousands of wavelengths in the NIRS method. This can become the reason of which the discrimination power of the patter recognition methods will decline.
- **Proposed Solution:** Research could be made to narrow down the feature extracting methods and simultaneously to decrease the spectra diagonal. As a result, deeper learning techniques such as convolutional neural networks (CNN) have the potential to offer more advanced and sophisticated tools for recognizing the good food patterns with NIRS-based determination of its quality.

2.11 Summary

This section provides a comprehensive summary of the reviews, identifies limitations, and proposes solutions for all the reviewed papers/studies, including a comparison with the current title “A Mobile Application for Food Quality Recognition”.

No	Name of papers (Main author - year)	Review Summary	Limitations	Solutions
1.	Using Machine Learning Approaches for Food Quality Detection (Jumming Han et al - 2022)	Explores machine learning for non-destructive fruit quality detection using CNNs, ResNets, and DenseNets.	CNNs face degradation problems with network depth.	Utilize transfer learning with modified architecture and dense block compression.
2.	Application of Image Processing in Fruit and Vegetable Analysis: A Review (Shiv Ram Dubey et al - 2015)	Discusses image-based analysis for fruit and vegetable recognition and disease classification.	Challenges in multiclass recognition and handling fruits in plastic bags.	Suggests deep learning models and block classification-based techniques.
3.	Study for Food Recognition System Using Deep Learning (Nareen O. M. Salim et al - 2014)	Reviews techniques for food recognition systems and challenges in real-time applications.	Constraints on food placement and server-based recognition.	Recommends image segmentation algorithms and Mobile Edge Computing (MEC).
4.	Food Handlers’ Food Safety Knowledge, Attitudes, and Practices in Taman Negara, Kuala Tahan (Chemah Tamby Chik et al -2023)	Focuses on food handlers’ knowledge and practices in Kuala Tahan, emphasizing continuous education.	Limited scope of training content and waste management systems.	Proposes specialized training programs and comprehensive waste management strategies.
5.	Development of Safety Assessment System fod Food Premises (Nazmi ‘Imran Makhilan et al - 2021)	Develops a safety assessment system for food premises to enhance food safety awareness.	Limited training content adaptability.	Suggests the development of adaptable training materials and advanced image segmentation.

6.	Smart Supervision for Food Safety in Food Service Establishments in China: Challenges and Solutions (Taibo Chen - 2020)	Introduces a smart supervision system for FSEs in China to improve policy implementation.	Heavy workload and high turnover rate of FSEs.	Implements a smart supervision system using mobile Internet and 2D barcode technology.
7.	Fruit Quality Recognition using Deep Learning Algorithm (Prof Sarika Bobde - 2021)	Develops a deep learning model for fruit quality recognition with high accuracy.	Limited dataset variability and model generalization.	Incorporates a more diverse dataset and conducts further testing for generalization.
8.	Pattern Recognition Techniques In Food Quality And Authenticity: A Guide On How To Process Multivariate Data In Food Analysis (Adriano de Araújo Gomes - 2023)	Provides an overview of pattern recognition techniques in food analysis for quality and authenticity.	Complexity of food systems and data interpretation challenges.	Advanced analytical techniques and multivariate data modeling are needed.
9.	Recognition of Food Material and Measurement of Quality using YOLO and WLD-SVM (Bima Sena Bayu Dewantara - 2021)	Presents a system for recognizing food materials and measuring quality using YOLO and WLD-SVM.	Inconsistent detection accuracy and mixed texture descriptor interference.	Enhances training datasets and develops separate models for different food types.
10.	A Review of the Discriminant Analysis Methods for Food Quality Based on Near-Infrared Spectroscopy and Pattern Recognition (Jian Zeng - 2021)	Reviews discriminant analysis methods for food quality assessment using NIRS and pattern recognition.	Limited data preprocessing techniques.	Application of pattern recognition chemometric tools for data classification.

Table 2.11.1 Summary of reviewed papers

11.	Proposed Application:	Project Overview	Innovative Approach	Limitation

	A Mobile Application for Food Quality Recognition (Connie Tang Ming Xin - 2024)	The project aims to develop a mobile application for recognizing food quality and focusing on food safety.	The project stands out for its comprehensive approach to food quality recognition, integrating machine learning models into a mobile app for real-time recognition.	Challenges include ensuring data privacy, addressing model bias, and obtaining user consent for data usage.

Table 2.11.2 Comparison of this project with the above reviewed papers/studies

The reviewed studies address a wide range of studies made on food safety and proper identification just by the application of machines that can make use of computer linguistics and digital image processing. This point discusses the issues at hand and then gives propositions to how they can be fixed. Richings the text with examples of CNNs, SVMs, and image segmentation, the technical solution puts a strong point on the significance of such instruments for precise food quality assessment. Furthermore, the mentioned literature is about the huge contributions of these technologies in food security and food products amount minimization.

CHAPTER 3
Proposed Methodology/Approach

This chapter explores the design and complexity of a food quality identification app. It starts with the design specifications, covering the development framework and approach. Then, it presents the system architecture, use case, and activity diagrams. These visuals offer a clear view of the system's structure, user interactions, and workflows, giving key insights into the app's design and operations.

3.1 Design Specifications

3.1.1 Development Framework and Methodology

1. **Research**
   - Conduct thorough research on the food tracking/recognition systems available on the market.
   - Identify approaches relevant for image processing, object detection and recognition models, APIs and tools which are important.

2. **Design**
   - Design a user-friendly mobile application interface.
   - Develop model for object and quality detection.
   - Define data for user to view more detailed information about the quality of the food.

3. **Implementation**
   - Leverage upon the programming language and framework that match the criteria of the mobile app development.
   - Apply the develop model into application to optimize the application for performance and resource efficiency on mobile devices.

4. **Testing**
   - Conduct one stringent testing for the deviation from food recognition model.
   - Check the application performance upon the different scenarios and condition alternatives.

3.2 System Design Diagram

3.2.1 System Architecture Diagram

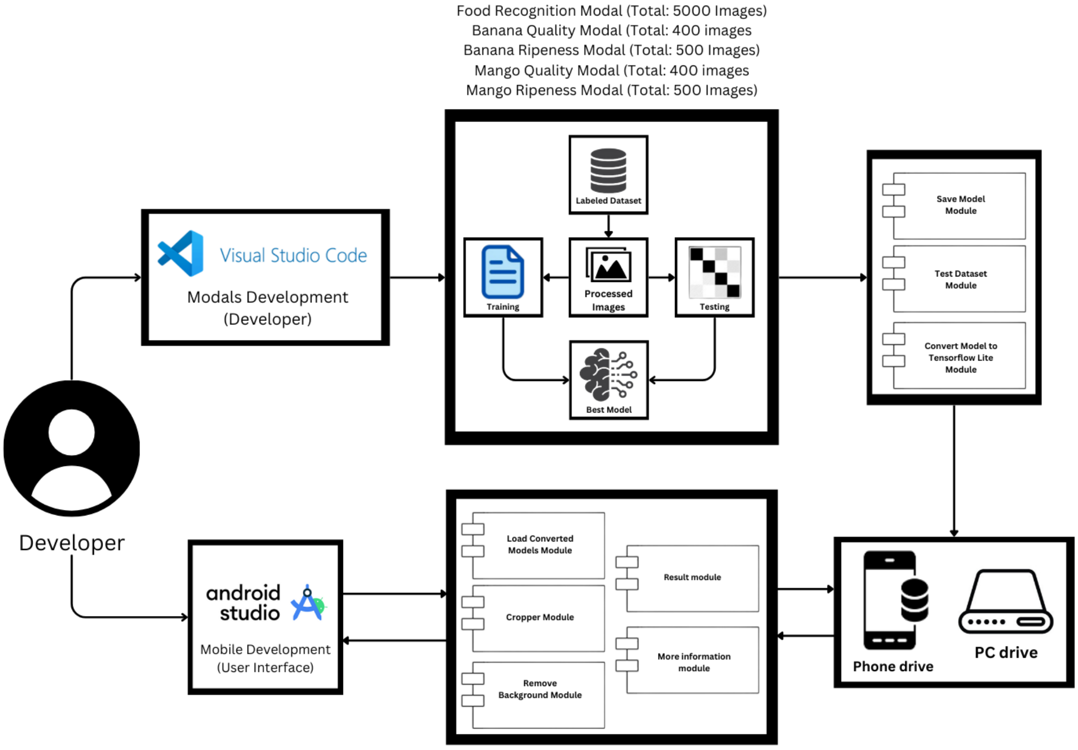


Figure 3.2.1 System Design Diagram

The Figure 3.1.1.1 illustrates the workflow for developing a Food Quality Recognition Mobile Application. The process is divided into two main phases: model development and mobile application development. These phases are handled using Visual Studio Code for training machine learning models and Android Studio for building the mobile application.

1. Models training and development

The model training and development are carried out using Python 3.8.19, managed through an Anaconda environment in Visual Studio Code. The following five machine learning models are developed:

- **Food Recognition Model:** This model recognizes fruits and vegetables from uploaded images and identifies their type.
  - **Classes:** "apple", "avocado", "banana", "beetroot", "bitter gourd", "blueberry", "cabbage", "capsicum", "carrot", "cauliflower", "chilli pepper", "corn", "cucumber", "dragon fruit", "durian", "eggplant", "finger lady", "garlic", "ginger", "grapes", "guava", "honeydew", "jackfruit", "kiwi", "lemon", "lettuce", "longan", "lychee", "mango", "mangosteen", "onion", "orange", "papaya", "peach", "pear", "peas", "pineapple", "pomegranate", "potato",

"pumpkin", "raddish", "rambutan", "soy beans", "spinach", "strawberry", "sweetpotato", "tomato", "turnip", "watermelon", "yam"

- **Banana Quality Model & Mango Quality Model:** These models assess the quality of bananas and mangoes.
    - **Quality:**
        - 1. Good
        - 2. Bad
- **Banana Ripeness Model** & **Mango Ripeness Model:** These models assess the ripeness level of bananas and mangoes.
    - **Ripeness level:**
        - 1. Level 1 (not ripe)
        - 2. Level 2 (ripe)
        - 3. Level 3 (over ripe)
        - 4. Level 4 (spoiled)

a. <u>Image processing</u>

Before training, all images in the datasets are preprocessed. Preprocessing includes:

- Resizing the images to 256x256 pixels.
- Data Augmentation to enhance the dataset and improve model robustness.
- Background Removal (except for the food recognition dataset).

b. <u>Model Performance Evaluation</u>

After training, the best-performing models are saved locally by "Save Model Module". These models are tested on the entire dataset, and metrics such as confusion matrix, accuracy, precision, recall, and F1-score are generated for each model by "Test Dataset Module". These performance result will be manually saved locally.

Once satisfactory performance is achieved, the models are converted into TensorFlow Lite format by "Convert Model to Tensorflow Lite Module", making them suitable for integration into mobile applications.

2. Mobile applications development

The next phase is the development of the mobile application using Android Studio. The TensorFlow Lite models are integrated into the app for real-time inference. The mobile application consists of six activity pages:

- **Home Page:** This is the main interface where users can choose to upload an image for prediction by either taking a photo using the camera or selecting an image from the gallery. Users can also view their previous scan results on this page and access additional information by proceeding to the More Information1 Page.
- **Cropper Page:** This page allows users to crop the uploaded images for better analysis using the "Cropper Module."
- **Remove Background Page:** After cropping the image, users proceed to this page where the uploaded image undergoes background removal via the "Remove Background Module." This step removes unnecessary elements, focusing only on the relevant food item. Users will see a loading indicator, but the image with the background removed is not displayed. Once the background removal process is complete, users are automatically redirected to the Result Page.
- **Result Page:** On this page, users can view the selected image and the model's predictions for food quality and ripeness, generated by the "Result Module." There is also a 'Show Details' button that allows users to navigate to the More Information2 Page for additional insights.
- **More Information1 Page & More Information2 Page:** These pages, managed by the "More Information Module," have the same layout and purpose. They provide users with more detailed information about the food's quality and ripeness. Additionally, links are provided for users to read related articles and learn more about the food.

All results generated by the models from the uploaded images are saved to the user's local phone storage, allowing the scanned food history to be displayed on the Home Page.

3.2.2 Use Case Diagram

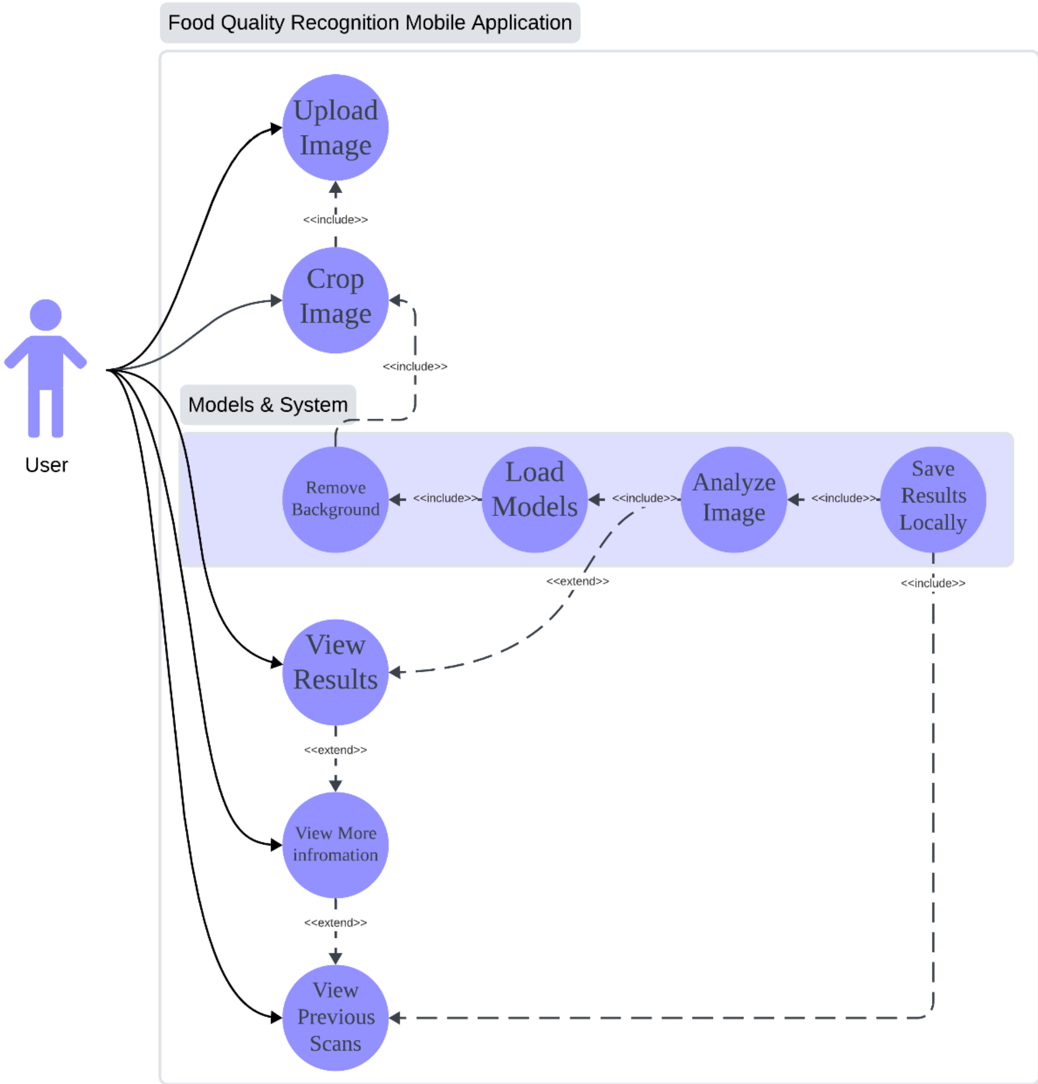


Figure 3.2.2 Use Case Diagram

The above use case diagram shown in Figure 3.1.2 effectively captures the core functionalities and interactions within the Food Quality Recognition Mobile Application. It clearly outlines the steps involved in image analysis, result viewing, and saving.

1. Key Use Cases and Relationships

- **Upload Image:** The primary interaction where the user starts the process by providing an image.
- **Crop Image:** A mandatory step included within "Upload Image," indicating that cropping is essential before analysis.
- **Remove Background:** Another mandatory step extended within "Crop Image," suggesting that background removal is necessary for accurate analysis.
- **Load Models:** A system-level use case indicating that the necessary models are loaded for analysis.

- **Analyze Image:** The core functionality where the system processes the image using the loaded models.
- **Save Results Locally:** A system-level use case for storing results on the device.
- **View Results:** A user-facing use case where the user can see the analysis outcomes.
- **View More Information:** An optional extension of "View Results," providing additional details.
- **View Previous Scans:** A user-facing use case for reviewing past analyses.

2. Sequence of use cases

- Upload Image -> Crop Image -> Remove Background -> Load Models -> Analyze Image -> Save Results Locally -> View Results -> View More Information.
- View Previous Scans -> View More Information.

Overall, the use case diagram in Figure 3.1.2 effectively illustrates the main features and interactions of the Food Quality Recognition Mobile App. It highlights the sequence of actions from image upload, cropping, and background removal to model loading, image analysis, result viewing, and saving. Optional actions like viewing more information or previous scans enhance user experience, providing a clear and streamlined process flow for food quality recognition.

3.2.3 Activity Diagram

1. Activity Diagram for Uploading Image from Gallery

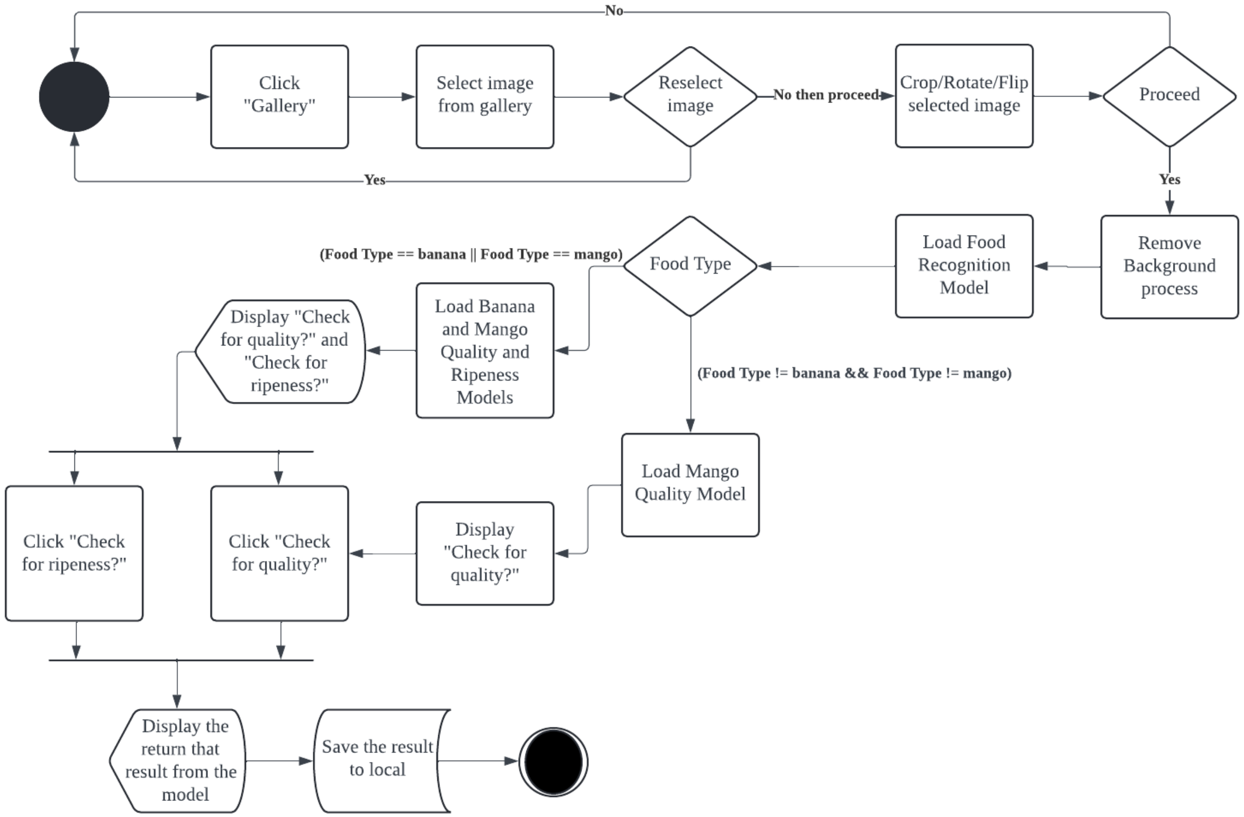


Figure 3.2.3 Activity Diagram for Uploading Image form Gallery

2. Activity Diagram for Uploading Image by Capture from Camera

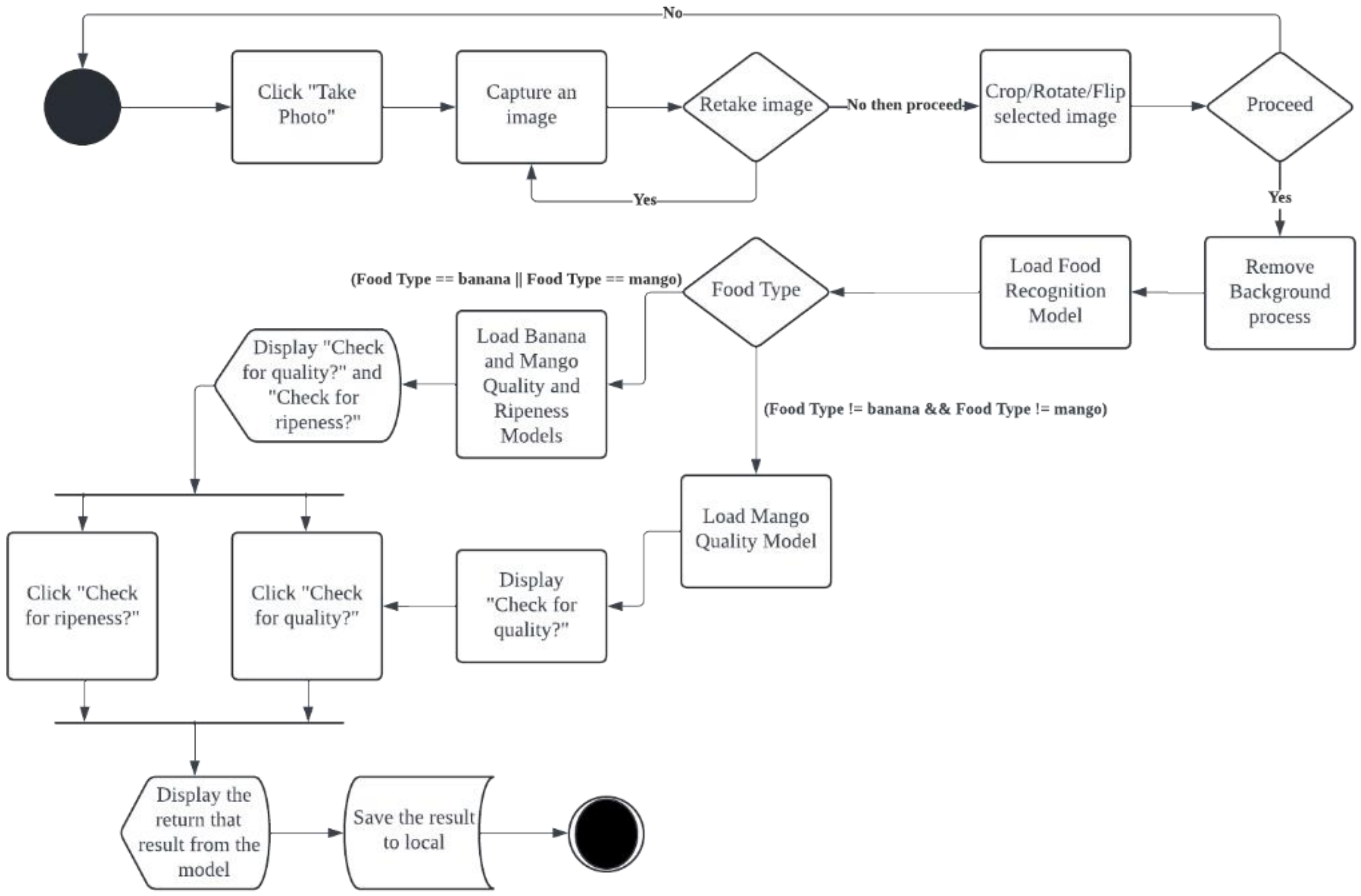


Figure 3.2.4 Activity Diagram for Uploading Image from Camera

The above activity diagrams in Figure 3.1.3.1 and Figurer 3.1.3.2 are similar. Their outlines the steps involved in uploading an image from a gallery or take photo by camera in a food quality recognition mobile application. It demonstrates the decision points and actions taken based on the user's choices and the type of food being analyzed.

Key Steps and Decisions

- **Start:** The process begins.
- **(Gallery) Click "Gallery":** The user clicks the "Galley" button to select an image.
- **(Gallery) Select Image from Gallery:** The user chooses an image from their phone's gallery.
- **(Camera) Click "Take Photo":** The user click the "Take Photo" button to capture an image of the food.
- **Crop/Rotate/Flip (Optional):** The user has the option to crop, rotate or flip the selected image.
- **Proceed:** The user confirms their selection and proceeds to the next step.
- **Remove Background:** The background of the image is removed.
- **Food Type:** The system determines the type of the food in the image by Food Recognition Model.
- **Load Models:** The appropriate models are loaded based on the food type.
    - If banana or mango, then loads both their quality and ripeness models.
        - **Display Questions:** The user is presented with options to check for quality and ripeness of the food.
    - Else then only load mango quality model.
        - **Display Question:** The user is presented only option to check for quality of the food.
- **Analyze Image:** The system analyzes the image using the loaded models.
- **Display Result:** The analysis result is displayed to the user.
- **Save Result to Local:** The result is saved to the phone's local storage.
- **End:** The process concludes.

Overall, the activity diagrams provide a clear visualization of the steps involved in uploading an image, analyzing its quality and ripeness, and saving the results for user review from history.

3. Activity Diagram for More Information

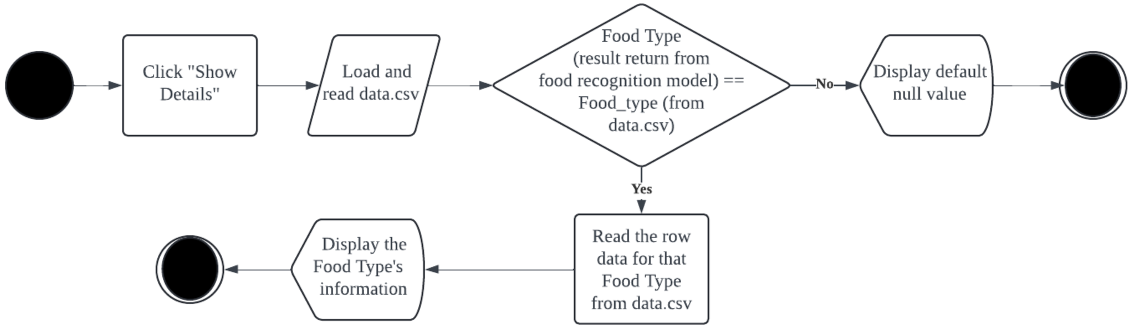


Figure 3.2.5 Activity Diagram for More Information

This activity diagram illustrates the process of retrieving and displaying additional information about the recognized food item based on the result from the Food Recognition Model.

Key Steps and Decisions

- **Start:** The process begins.
- **Click "Show Details":** The user initiates the process by clicking the "Show Details" button.
- **Loads and read data.csv:** The system loads and reads the contents of the data.csv file.
  - Data includes (below is first 5 examples in data.csv):

	A	B	C	D	E	F	G	H	I	J	K
1	Food_type	Vitamin	Image_name	pick	bad	more					
2	apple	Vitamin C, Vitamin A, Vitamin K	apple_good_bad	https://www.wikihow.com/Choose-an-Apple	null	https://helpmechoosemyfruitandveg.com/apple.html					
3	avocado	Vitamin E, Vitamin K, Vitamin C, Vitamin B6	avocado_good_bad	https://www.wikihow.com/Tell-if-an-Avocado-Is-Ripe	https://www.wikihow.com/Tell-if-an-Avocado-is-Bad	https://helpmechoosemyfruitandveg.com/avocado.html					
4	banana	Vitamin B6, Vitamin C, Vitamin A	banana_good_bad	https://www.thespruceeats.com/banana-selection-and-storage-1807738	null	https://helpmechoosemyfruitandveg.com/banana.html					
5	beetroot	Vitamin C, Vitamin B6, Folate	beetroot_good_bad	https://www.mrishtanna.com/how-long-do-cooked-beets-last/#how-long-do-beets-last	null	https://helpmechoosemyfruitandveg.com/beet.html					

Figure 3.2.6 data.csv

    - Food_type (the food name),

        - Vitamin (the vitamins of the food),
        - Image_name (use for display correct image based on the Food_type),
        - pick (a link bring user to view how to pick the food),
        - bad (a link bring user to view how to know the food gone bad),
        - more (a link bring user to view more information about the food).
- **Check Food Type:** The system compares the food type recognized by the model with the food types listed in the data.csv file.
    - **Display Default Null Value:** If the recognized food type is not found in the data.csv file, a default null value or message is displayed. Possible reason is because the wrong spelling so cannot be found.
    - **Display Food Type's Information:** If a match is found, the system reads the row data in the data.csv file corresponding to the recognized food type. The relevant information for that food type is then extracted and displayed to the user.

Overall, this activity diagram outlines the steps involved in retrieving and displaying more information about the recognized food item, ensuring that user has access to relevant details when needed.

4. **Activity Diagram for Off Page Link**

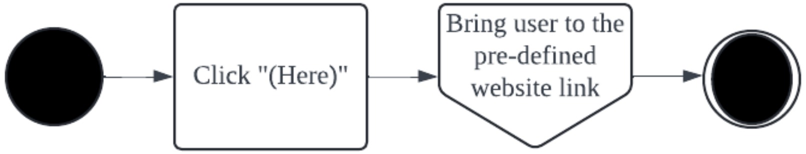


Figure 3.2.7 Activity Diagram for Off Page Link

The above activity diagram illustrates a simple process where a user clicks "(Here)" is a link to be redirected to predefined external website.

Key Steps and Decisions

- **Click "(Here)":** the user initiates the process by clicking on it. This "(Here)" is a link that associated with a predefined URL. So, after clicking will triggers a redirection to the external website.
- **Bring User to the Pre-defined Website Link:** The system then navigates the user to the specific external website.

Overall, this activity diagram outlines the simple process of redirecting the user to an external website based on the click event.

CHAPTER 4
System Implementation and Design

This chapter focuses on the system implementation and design aspects of the Food Quality Recognition Mobile Application. It presents what have implement into the system and system overview design/flowchart. These offered insights into the application's design and functionality.

4.1 System Implementation

4.1.1 Rembg [29]

1. **Introduction**

Rembg is a tool to remove the background from images. This processing step ensure that irrelevant or messy backgrounds do not interfere with the model training process.

By eliminating the background, Rembg helps to enhance model accuracy. With a clean and uniform background, the model can focus more effectively on the surface of the object, leading to better feature extraction and more precise learning of object details.

Example Comparison: Below is an example illustrating the effect of using Rembg.

Figure 4.1.1 Rembg (before and after)

- This is one of the images from the dataset.
- The after image's background is null, not white color.

Therefore, the model can focus on the surface of the object from the dataset to learn the details and their different during the model training.

2. Implementation

This Rembg was implemented in the part of processing image from the dataset. Firstly, the images from the dataset will be resize to 256*256, then apply together for the data augmentation (to increase the dataset up to total 100 images for each class) and this Rembg (to remove the image background). After these all the processed image will be saved into "PreprocessedImages" directory. Example below is the original images and processed images for quality dataset:

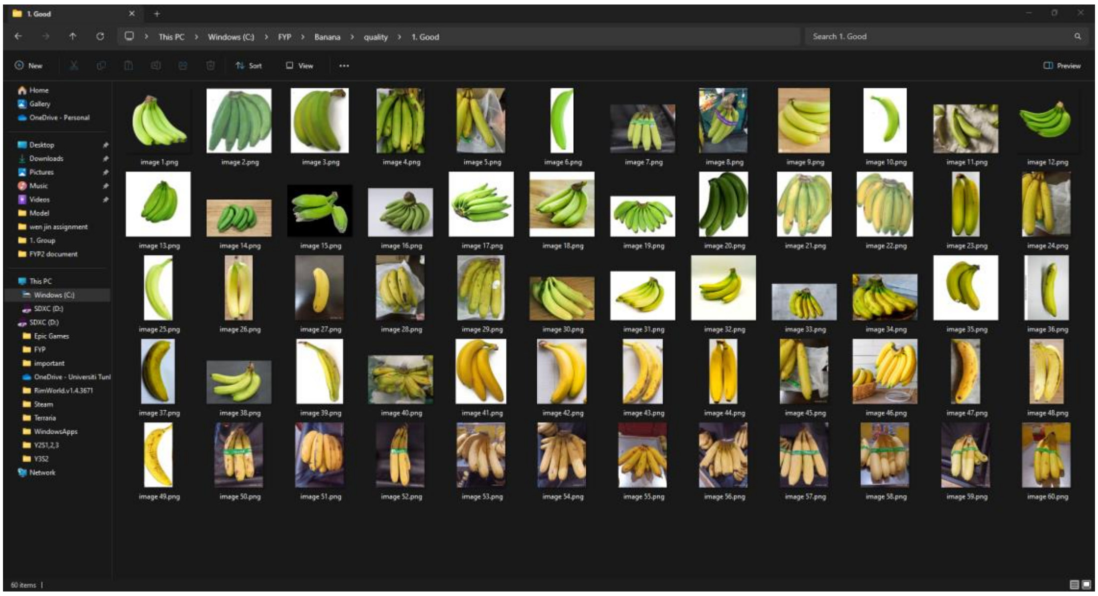


Figure 4.1.2 Original images

Figure 4.1.3 Processed images

4.1.2 AutoKeras [30]

1. Introduction

Below is the logo of AutoKeras and its short introduction:

Figure 4.1.4 Logo of AutoKeras

- AutoKeras is an AutoML (Auto Machine Learning) system which a framework automate create the best neural networks build on top of Keras that fits best to the dataset.
- It is developed by DATA Lab at Texas A&M University.
- The goal of AutoKeras is to make machine learning accessible to everyone.

This tool is used to develop the model for core system.

During the development, this tool is helping in the following tasks:

- **Image Classification model building:** It automates the process of constructing images classification models, exploring various architectures and hyperparameters to find the best neural network or the best model with the most effective configurations.
- **Model training and evaluation:** The tools streamlines the training process by automating hyperparameter tuning and model selection. It also evaluates the model's performance using validation datasets, ensuring that the best possible model is identified.

Overall, AutoKeras is a machine learning library built on Keras and TensorFlow that simplifies model building and hyperparameter tuning using KerasTuner [31]. Its workflow involves analyzing training data, constructing a suitable search space for neural architectures and hyperparameters, and finding high-performing values. AutoKeras supports state-of-the-art models like EfficientNet and BERT, allowing for the use of pretrained weights. In addition to optimizing model architecture, it tunes preprocessing steps and training parameters such as data augmentation, feature encoding, optimizers, and learning rates. By automating these tasks, AutoKeras not only enhances model accuracy and reliability but also reduces development time, optimizes resource usage, and accelerates the iteration and deployment process, improving overall efficiency.

2. Implementation

This AutoKeras was implemented in the part of model training. Firstly, install the package by "**pip install autokeras**" and then import the library by "**import autokeras as ak**".

Example code snippet of a model training by using AutoKeras:

```
# Model Training
print("Starting Quality Model Training")
quality_clf = ak.ImageClassifier(
    max_trials=3,
    overwrite=True,
    objective="val_accuracy",
    project_name="banana_quality_model",
    directory=quality_model_dir
)

early_stopping_callback = EarlyStopping(monitor='val_accuracy', patience=5, restore_best_weights=True)
reduce_lr_callback = ReduceLROnPlateau(monitor='val_accuracy', factor=0.2, patience=3, min_lr=0.00001)
callbacks = [early_stopping_callback, reduce_lr_callback]

quality_clf.fit(train_dataset, validation_data=val_dataset, epochs=50, callbacks=callbacks)
```

Figure 4.1.5 AutoKeras implemented for model training

Based on above code snippet is a model training for banana quality model. The model running search trial in 3 times. Each time will be 50 epochs, but will be stop training and reduce the learning rate by the callbacks (EarlyStopping and ReduceLROnPlateau).

a. Trial

```
Search: Running Trial #1

Value             |Best Value So Far |Hyperparameter
vanilla           |vanilla           |image_block_1/block_type
True              |True              |image_block_1/normalize
False             |False             |image_block_1/augment
3                 |3                 |image_block_1/conv_block_1/kernel_size
1                 |1                 |image_block_1/conv_block_1/num_blocks
2                 |2                 |image_block_1/conv_block_1/num_layers
True              |True              |image_block_1/conv_block_1/max_pooling
False             |False             |image_block_1/conv_block_1/separable
0.25              |0.25              |image_block_1/conv_block_1/dropout
32                |32                |image_block_1/conv_block_1/filters_0_0
64                |64                |image_block_1/conv_block_1/filters_0_1
flatten           |flatten           |classification_head_1/spatial_reduction_1/reduction_type
0.5               |0.5               |classification_head_1/dropout
adam              |adam              |optimizer
0.001             |0.001             |learning_rate
```

Figure 4.1.6 Running Trial 1

```
Search: Running Trial #2

Value             |Best Value So Far |Hyperparameter
resnet            |vanilla           |image_block_1/block_type
True              |True              |image_block_1/normalize
True              |False             |image_block_1/augment
True              |None              |image_block_1/image_augmentation_1/horizontal_flip
True              |None              |image_block_1/image_augmentation_1/vertical_flip
0                 |None              |image_block_1/image_augmentation_1/contrast_factor
0                 |None              |image_block_1/image_augmentation_1/rotation_factor
0.1               |None              |image_block_1/image_augmentation_1/translation_factor
0                 |None              |image_block_1/image_augmentation_1/zoom_factor
False             |None              |image_block_1/res_net_block_1/pretrained
resnet50          |None              |image_block_1/res_net_block_1/version
True              |None              |image_block_1/res_net_block_1/imagenet_size
global_avg        |flatten           |classification_head_1/spatial_reduction_1/reduction_type
0                 |0.5               |classification_head_1/dropout
adam              |adam              |optimizer
0.001             |0.001             |learning_rate
```

Figure 4.1.7 Running Trial 2

```
Search: Running Trial #3

Value              |Best Value So Far |Hyperparameter
efficient          |vanilla           |image_block_1/block_type
True               |True              |image_block_1/normalize
True               |False             |image_block_1/augment
True               |None              |image_block_1/image_augmentation_1/horizontal_flip
False              |None              |image_block_1/image_augmentation_1/vertical_flip
0                  |None              |image_block_1/image_augmentation_1/contrast_factor
0                  |None              |image_block_1/image_augmentation_1/rotation_factor
0.1                |None              |image_block_1/image_augmentation_1/translation_factor
0                  |None              |image_block_1/image_augmentation_1/zoom_factor
True               |None              |image_block_1/efficient_net_block_1/pretrained
b7                 |None              |image_block_1/efficient_net_block_1/version
True               |None              |image_block_1/efficient_net_block_1/trainable
True               |None              |image_block_1/efficient_net_block_1/imagenet_size
global_avg         |flatten           |classification_head_1/spatial_reduction_1/reduction_type
0                  |0.5               |classification_head_1/dropout
adam               |adam              |optimizer
2e-05              |0.001             |learning_rate
```

Figure 4.1.8 Running Trial 3

Here is the breakdown of the hyperparameters tried about:

- **Block Type & Version**
    - **Vanilla:** The value vanilla suggests that a basic, standard CNN architecture is being used. Vanila neural networks refer to straightforward, fully connected networks without any advanced architectures like residual blocks or attention mechanisms.
    - **ResNet 50:** The value ResNet 50 is an architecture includes skip connections that help the model "skip" certain layers and pass information directly to deeper layers, allowing for very deep networks to be trained effectively.
    - **Efficient B7:** The value Efficient b7 is an architecture, which is more efficient CNN that scales both depth and width of the model along with resolution. This architecture achieves a better balance between accuracy and computational efficiency, allowing for high performance with fewer parameters compared to older architectures like ResNet.
- **Normalization = True**
    - "True" means that the model is normalizing the input data, a crucial preprocessing step. Normalization adjust the range of pixel values (e.g. between 0 and 1), which helps the neural network converge faster during training.
- **Augmentation**
    - Only Trial 1 is "False" means that data augmentation is not applied in vanilla configuration.
    - Then Trial 2 and 3 is "True" means that data augmentation is applied in the configuration.

- Augmentation creates variations of the input data (e.g. flips, rotations, contrast adjustment, etc.) to help the model generalize better by training on slightly modified images.

- **Image Augmentation Techniques:** These specific augmentation techniques help create different versions of images to improve model robustness.
  - **Horizontal Flip:** Both Trial 2 and 3 is "True" mean that the random horizontal flips are applied to images.
  - **Vertical Flip:** Only Trial 2 is "True" mean that the random vertical flips are applied to images.
  - **Contrast Factor:** Both Trial 2 and Trial 3 is "0" mean that no contrast adjustment are applied to images.
  - **Rotation Factor:** Both Trial 2 and Trial 3 is "0" mean that no rotation adjustment are applied to images.
  - **Translation Factor:** Both Trial 2 and Trial 3 is "0.1" mean that images are translation (shifting) randomly by small factor from 0% up to 10%.
  - **Zoom Factor:** Both Trial 2 and Trial 3 is "0" mean that no zoom adjustment are applied to images.
- **Vanilla's parameters and classification head**

  (i) Parameters:
  - These parameters define the structure of the convolutional layers (the layers) that process image in the network.
  - **"kernel_size: 3"** this refers to the size of the filter used in convolutional layers. A 3*3 kernel is a standard choice that balances the complexity and efficiency of the feature extraction.
  - **"num_block: 1"** this indicates that there is a single convolutional block. Each block typically consists of multiple layers (like convolution activation, etc.).
  - **"num_layers : 2"** this is suggesting that there are two layers inside the convolutional block.
  - **"max_pooling: True"** this means that max pooling is being used, which helps reduce the spatial dimensions of the feature maps, thereby down-sampling the input to make the model more computationally efficient and to capture important features.
  - **"separable: False"** this means that separable convolutions are not used. Separable convolutions decompose the convolution operation into simple operations, which can reduce the number of parameters and computation.

- **"dropout: 0.25"** this means that there is 25% of the units in the convolutional layer are randomly dropped out. Dropout is a regularization technique to prevent overfitting by randomly setting a fraction of the layer's output to zero during training.
- **"filter_0_0: 32"** and "filter_0_1: 64" these specify the number of filters (e.g. output channels) in the first and second convolutional layers of the block. More filters allow the model capture more features at different levels of abstraction.

(ii) Classification Head:

- **"spatial_reduction1/reduction_type: flatten"** this is a flatten operation is used to covert the 2D feature maps into a 1D vector, which is necessary before epassing the data into fully connected (dense) layers for classification.
- **"classification_head_1/dropout: 0.5"** this means that dropout is also applied in the classification head, and with 50% of units being randomly dropped out to prevent overfitting.

- **ResNet 50's parameters and classification head**

  (i) Parameters:

  - **"Pretrained: False"** this means that the model is not initialized with pre-trained weights from ImageNet in this Trial 2.
  - **"Version: resnet50"** this means this Trial 2 confirms that the ResNet-50 version of the architecture is used. ResNest 50 also known for depth (50 layers) and its ability to learn detailed features, making it very effective for image classification tasks.
  - **"Imagenet Size: Ture"** this means the model expect the input images to match the sized used in ImageNet, which is 224*224 pixels.

  (ii) Classification Head:

  - **"Spatial Reduction: global_avg"** this means that the Global Average Pooling layer is used to reduce the spatial dimensions of the output before passing it to the fully connected layers. Global average pooling replaces fully connected layers and reduces overfitting by averaging the feature maps, thus producing a 1D vector.
  - **"Dropout: 0"** this means that no dropout regularization is applied in the classification head.

- **Efficient B7's parameters and classification head**

  (i) Parameters:

- **"Pretrained: True"** this means that the model is initialized with pre-trained weights from ImageNet in this Trial 3. Using pre-trained models enables the model to start with a better set of weights, which can significantly speed up training and improve performance, especially when fine-tuning.
- **"Version: b7"** this means this Trial 3 confirms that the EfficientNet B7 version of the architecture is used. B7 version is one of the larger and more powerful version in the EfficienNet family. It has more parameters and layers than smaller variants like B0 or B1, allowing it to learn more complex patters in the data.
- **"Trainable: True"** this means that Trial 3 indicates that the pre-trained layers of the EfficienNet B7 model are set to be trainable. In fine-tuning, pre-trained layers can be either frozen (non-trainable) or trainable, depending on whether you want to adjust those layers to your specific dataset. Here, they are fine-tuned.
- **"Imagenet Size: Ture"** this means the model expect the input images to match the sized used by EfficientNet B7, which is 600*600 pixels.

(ii) Classification Head:

- **"Spatial Reduction: global_avg"** this means that the Global Average Pooling layer is used to reduce the spatial dimensions of the output before passing it to the fully connected layers. Global average pooling replaces fully connected layers and reduces overfitting by averaging the feature maps, thus producing a 1D vector.
- **"Dropout: 0"** this means that no dropout regularization is applied in the classification head.

- **Optimizer**
  - Trial 1, 2 and 3 are all used "Adam". It adjusts the learning rate based on the gradients during training, leading to faster and more stable convergence.
- **Learning Rate**
  - Trial 1 and 2 are set to 0.001, which controls the size of the steps the optimizer takes when updating the model weights. A small learning rate like this is to prevents the model from overshooting the optimal weights during training.
  - Trial 3 is set to 2e-05, which a very small learning rate is applied. This is common when fine-tuning pre-trained models/ Smaller learning rates allow the model to make finer adjustments to the pre-trained weights, improving performance without drastically altering the learned features.

Conclusion:

- **Trial 1** is a simple vanilla Convolutional Neural Network (CNN) architecture was tested. The model consists of two convolutional layers per block, using 3x3 filters, followed by max pooling and dropout regularization to prevent overfitting. The final layer is flattened before reaching the classification head. The Adam optimizer was employed with a low learning rate, a standard practice for image-based tasks. However, Vanilla neural networks are basic fully connected architectures, lacking advanced features like convolutional layers, making them less effective for tasks such as image recognition [32].
- **Trial 2** is a ResNet-50 architecture replaces the basic CNN. ResNet-50, with 50 layers and skip connections, excels in image classification by retaining key features in deeper networks. The model also uses data augmentation techniques like flips and rotations to enhance robustness. Although not pre-trained, it is designed to handle ImageNet-sized inputs, a common practice for image classification tasks. ResNet-50 effectively tackles the vanishing gradient problem using skip connections, allowing for the training of deeper networks, often pre-trained on large datasets like ImageNet [33].
- **Trial 3** is a powerful variant of the EfficientNet family, which is EfficientNet B7. It offers state-of-the-art performance by efficiently scaling depth, width, and resolution, achieving high accuracy with fewer parameters compared to older models like ResNet. The trial utilizes pre-trained weights from ImageNet and fine-tunes them to adapt to the current dataset. Data augmentation, including horizontal flips and small rotations, is used to improve generalization. A very small learning rate (2e-05) is applied to prevent overfitting during fine-tuning. EfficientNet models optimize accuracy and efficiency by balancing depth, width, and resolution. EfficientNet-B7, a larger variant, is often fine-tuned with pre-trained weights to enhance performance and generalization [34] [35].
    - In conclusion, all the models developed by using EfficientNet B7 because after every time trial search it always get the best result so it was the best model after training.

b. Callbacks

- **EarlyStopping callback:** it will stop training if the validation accuracy doesn't improve for 5 epochs
- **ReduceLROnPlateau callback:** it will reduce the learning rate by a factor of 0.2 if the validation accuracy doesn't improve for 3 epochs, but the the learning rate will never go below 0.00001.

- Overall, these callbacks are helped to prevent overfitting when the model is not improving on the validation set and improve convergence when the model is stuck in a local minimum. Therefore, by suing these callbacks can often achieve better model performance and save training time.

4.1.3 Python packages

- The installed python packages that will be used for development the core system includes pandas, numpy, matplotlib, seaborn, scikit-learn, tensorflow, keras-tuner, autokeras, keras, opencv-python, ipywidgets and rembg.
- These packages collectively enhance system performance by improving the accuracy of models through better data handling, numerical computation, and visualization; increasing efficiency with optimized computational tools and automated processes; and enhancing user interaction through more intuitive and responsive interfaces.

4.1.4 Dataset

1. **Food Recognition Dataset**
   - Below here is the previous dataset in Project 1:

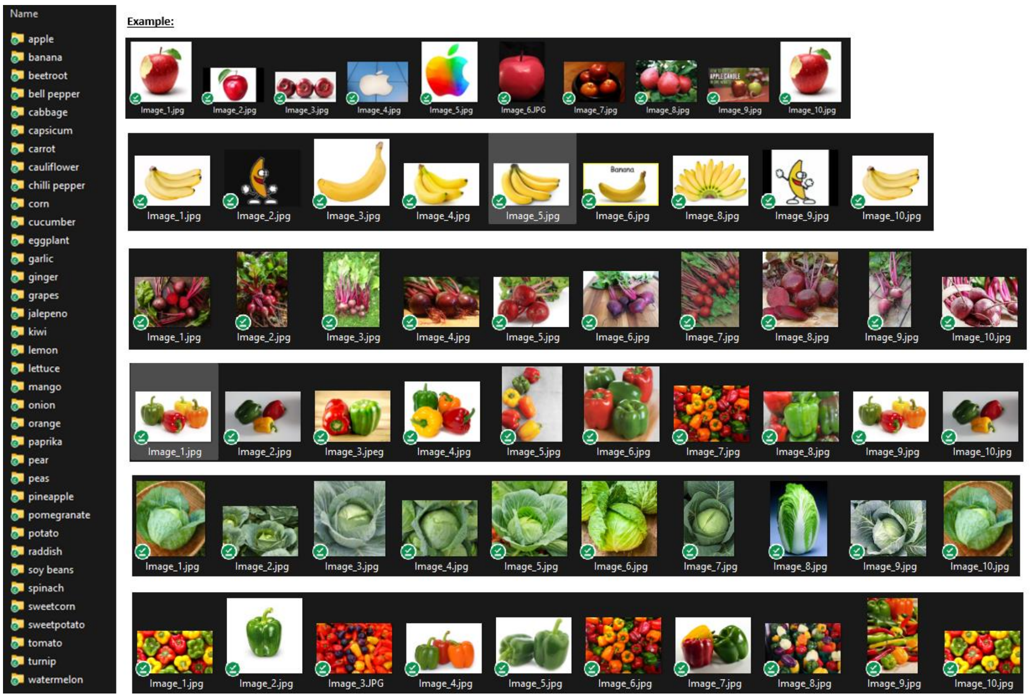


Figure 4.1.9 Food recognition dataset project 1

Found 36 different common fruits and vegetables for dataset use for continuous development. Each fruit and vegetable have 10 different images in the file. This dataset is from Kaggle website [36].

Then in Project 2, developer increase the dataset from 36 types of food to 50 types of food and each type of food have 100 different images in the file so total 5000 images. Here is the 50 classes for this dataset: "apple", "avocado", "banana", "beetroot", "bitter gourd", "blueberry", "cabbage", "capsicum", "carrot", "cauliflower", "chilli pepper", "corn", "cucumber", "dragon fruit", "durian", "eggplant", "finger lady", "garlic", "ginger", "grapes", "guava", "honeydew", "jackfruit", "kiwi", "lemon", "lettuce", "longan", "lychee", "mango", "mangosteen", "onion", "orange", "papaya", "peach", "pear", "peas", "pineapple", "pomegranate", "potato", "pumpkin", "raddish", "rambutan", "soy beans", "spinach", "strawberry", "sweetpotato", "tomato", "turnip", "watermelon", "yam".

- **Below here is a helper code that help developer to find and download the images for increase the dataset:**

```
#pip install bing-image-downloader

from bing_image_downloader import downloader

def download_images(search_term, num_images, output_dir):
    downloader.download(
        search_term,
        limit=num_images,
        output_dir=output_dir,
        adult_filter_off=True,  # Optionally turn off adult content filter
        force_replace=False,  # Optionally overwrite existing images
        timeout=60,  # Timeout for requests
        verbose=True
    )

if __name__ == "__main__":
    # Set the search term and number of images
    search_term = "bitter gourd fruit and vegetable"  # Change this to your desired search term
    num_images = 100  # Change this to the number of images you want to download
    output_dir = r"C:\FYP\Food recognition dataset\bitter gourd"   # Directory where images will be saved

    # Download the images
    download_images(search_term, num_images, output_dir)

```

Figure 4.1.10 Helper code

2. Banana Quality Dataset

- Dataset for banana quality is find from Internet and capture real life images from supermarket. Example:

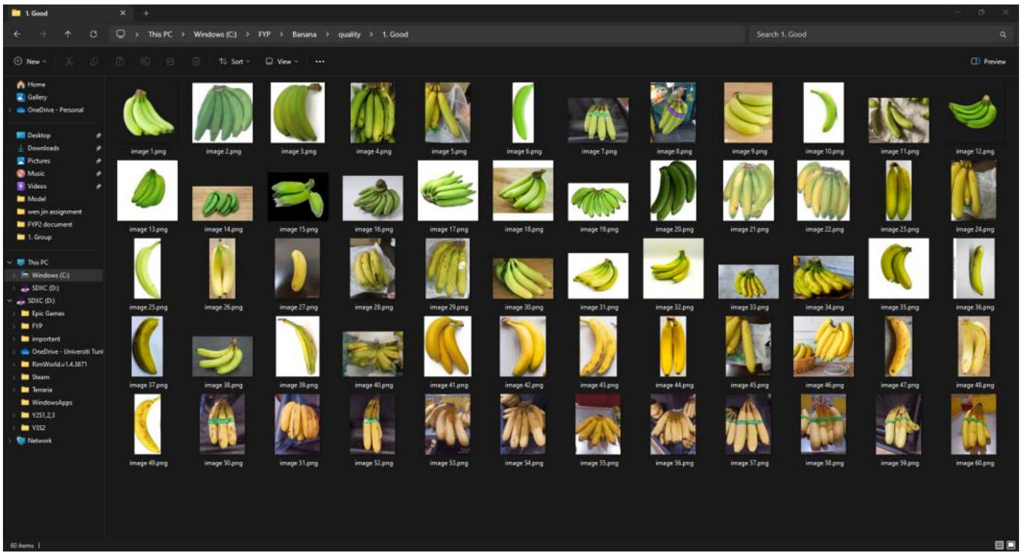


Figure 4.1.11 Banana good quality

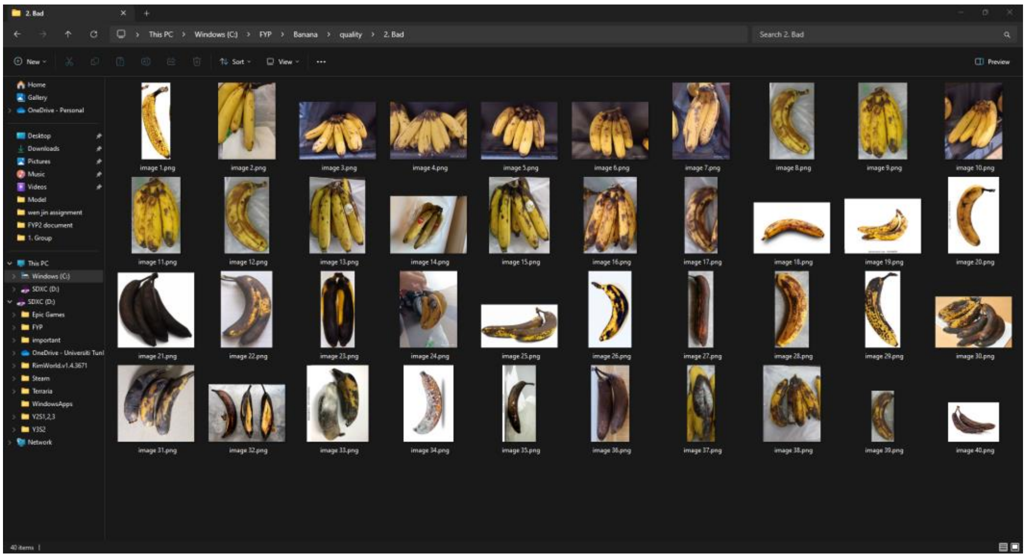


Figure 4.1.12 Banana bad quality

- Total: 60 + 40 = 100 images

3. Banana Ripeness Dataset

- Dataset for banana ripeness is find from Internet and capture real life images from supermarket. Example:

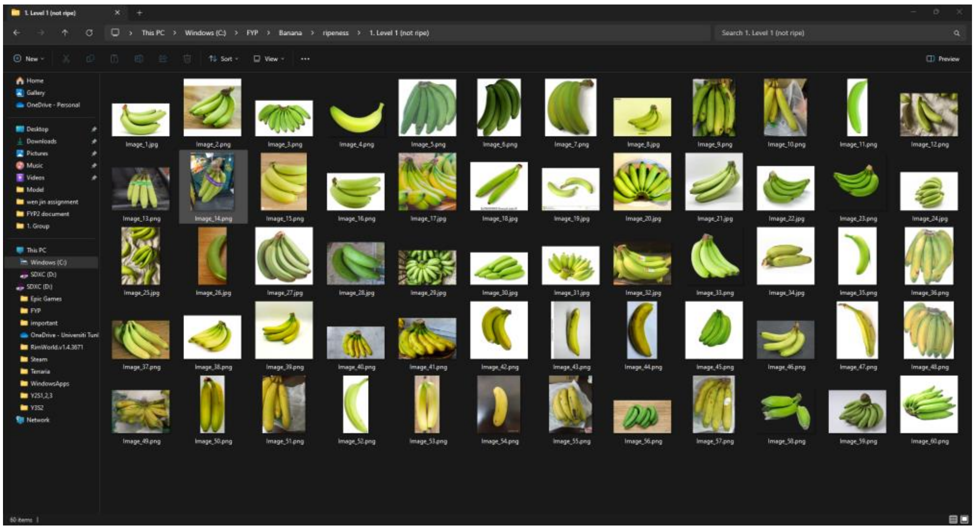


Figure 4.1.13 Banana level1

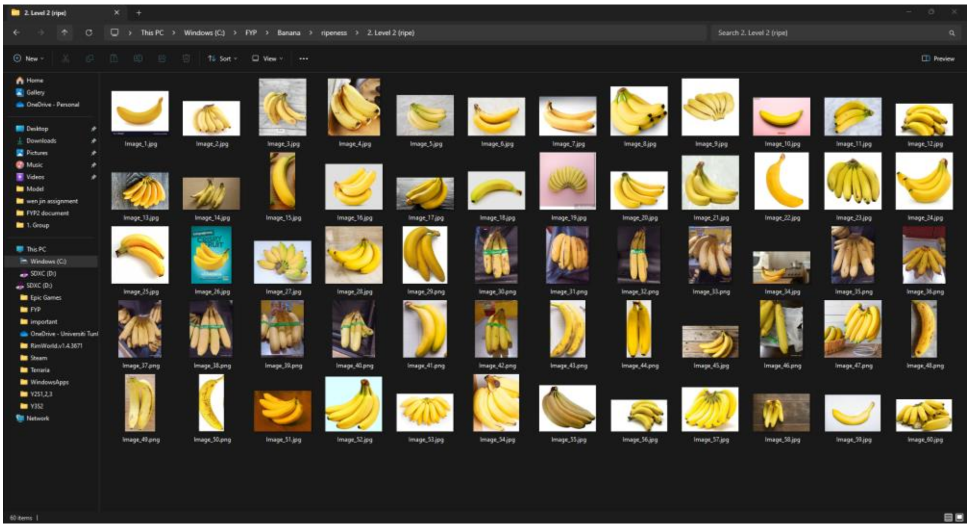


Figure 4.1.14 Banana level2

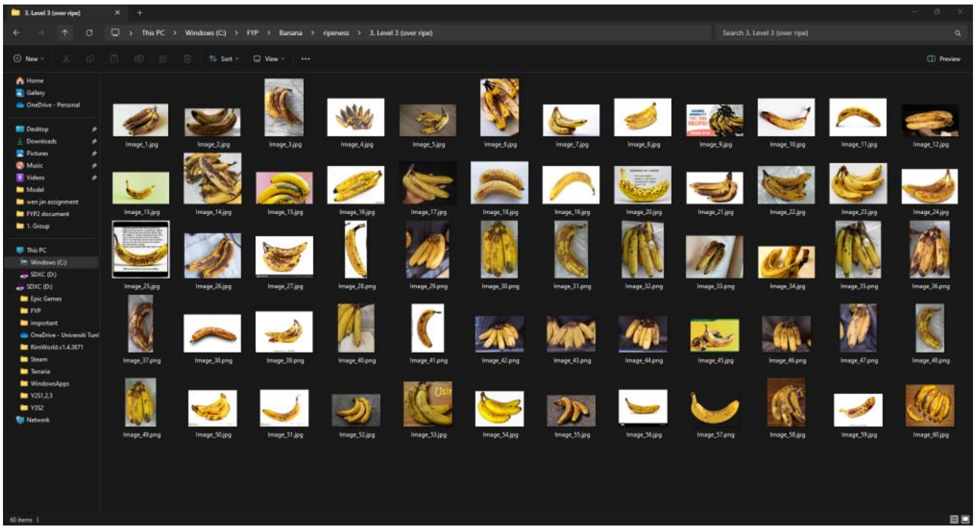


Figure 4.1.15 Banana level3

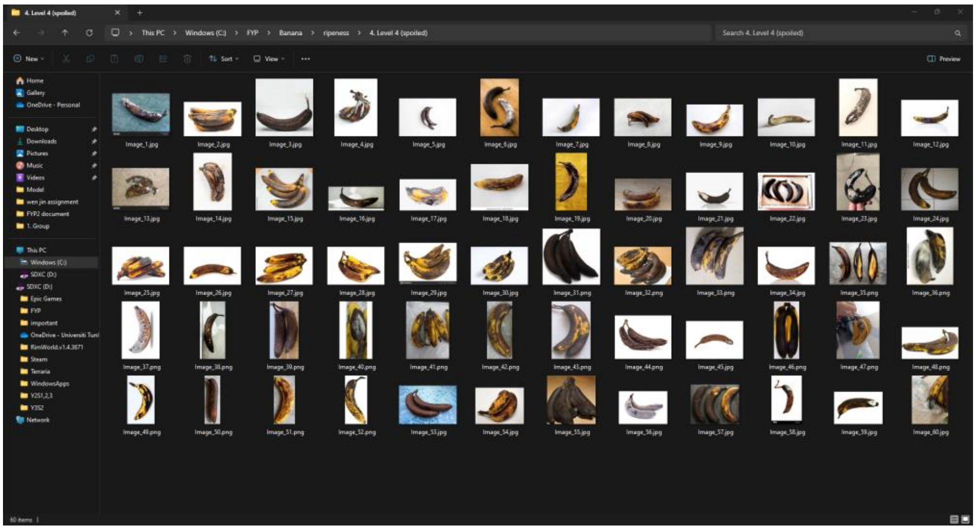


Figure 4.1.15 Banana level4

- Total: 60 + 60 + 60 + 60 = 240 images
- Some of the images is get from Google and Roboflow [37].

4. Mango Quality Dataset

- Dataset for mango quality is find from Internet and capture real life images from supermarket. Example:

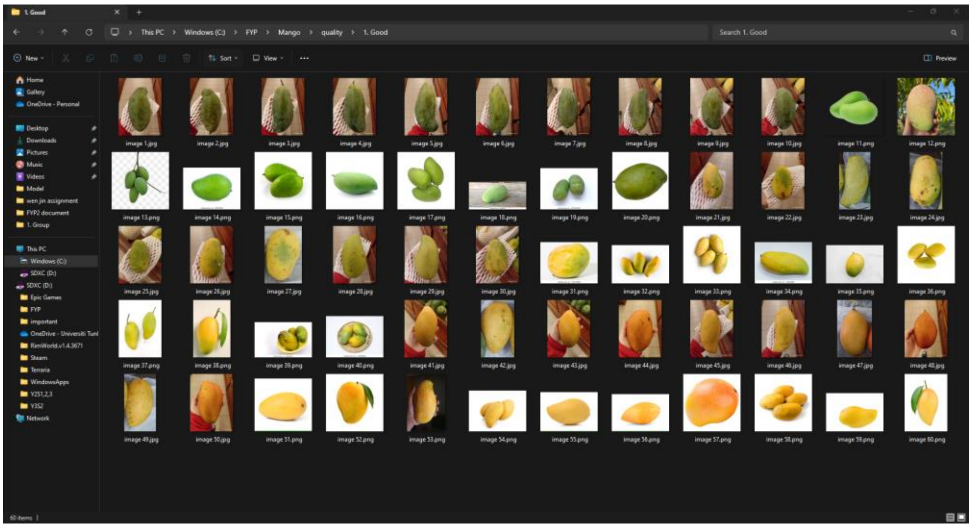


Figure 4.1.16 Mango good quality

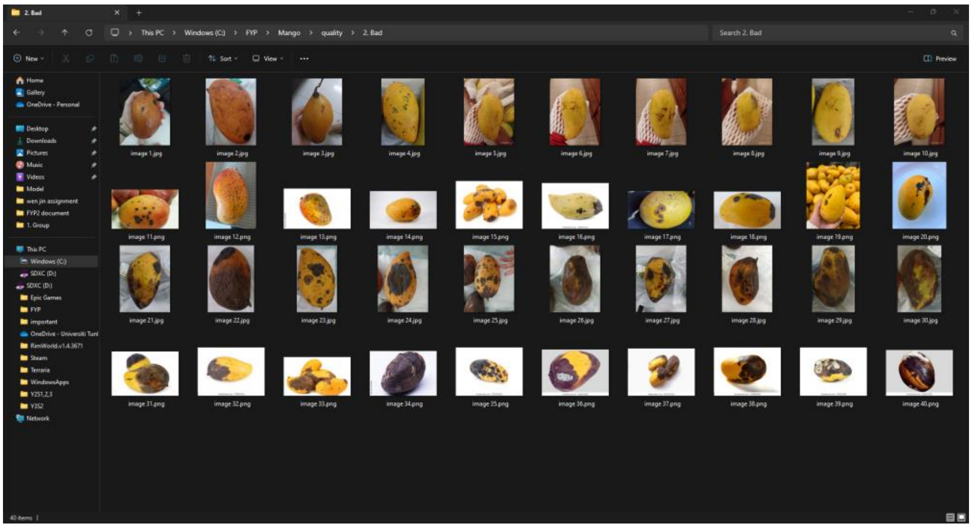


Figure 4.1.17 Mango bad quality

- Total: 60 + 40 = 100 images

5. Mango Ripeness Dataset

- Dataset for mango ripeness is find from Internet and capture real life images from supermarket. Example:

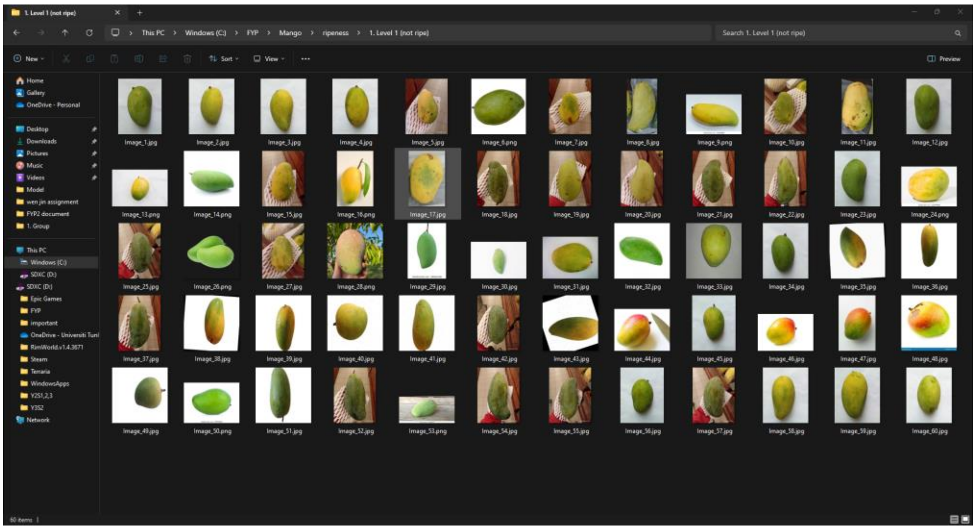


Figure 4.1.18 Mango level1

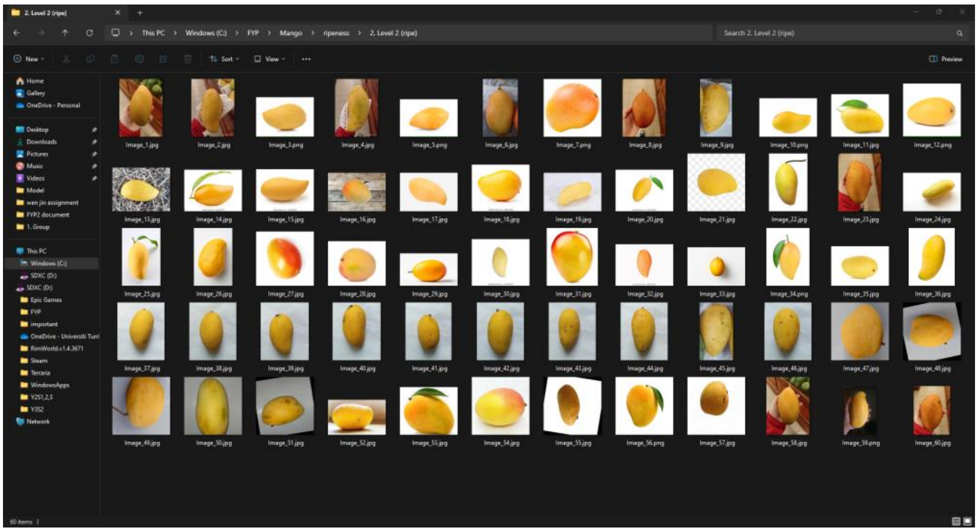


Figure 4.1.19 Mango level2

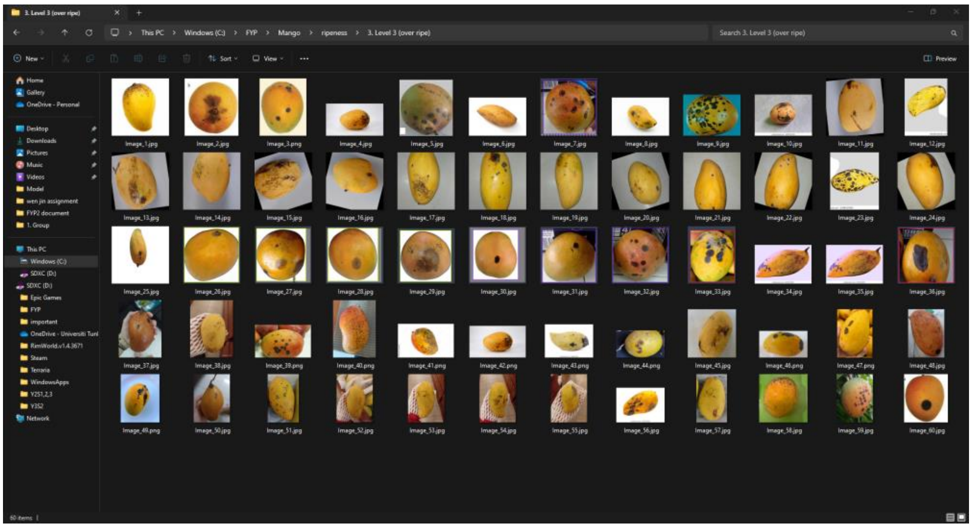


Figure 4.1.20 Mango level3

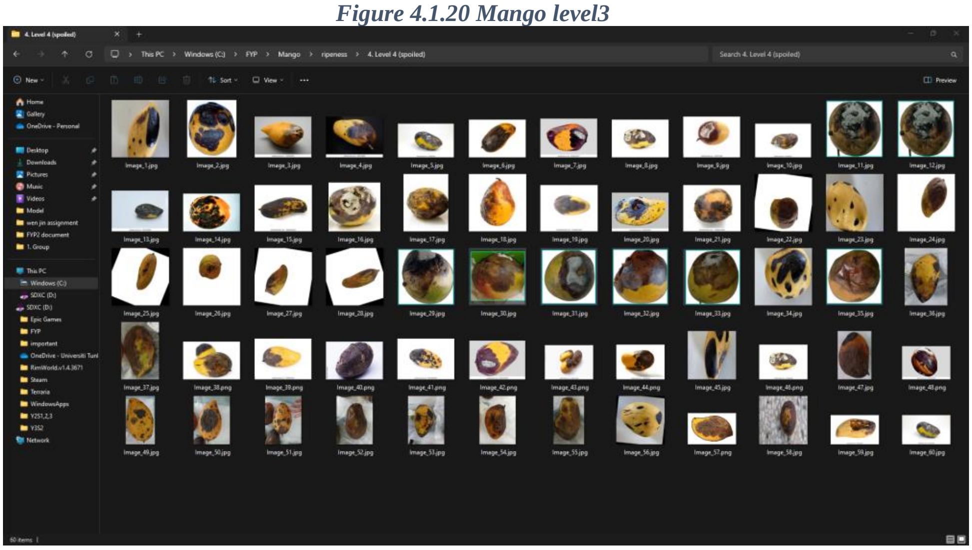


Figure 4.1.15 Banana level4

- Total: 60 + 60 + 60 + 60 = 240 images
- Some of the images is get from Google, Roboflow and Kaggle [38] [39] [40].

4.2 System Design/Overview

In this section, a comprehensive overview of the system design for the mobile application aimed at food quality recognition is provided. This includes outlining the architecture, functionality, and interaction of various components within the application.

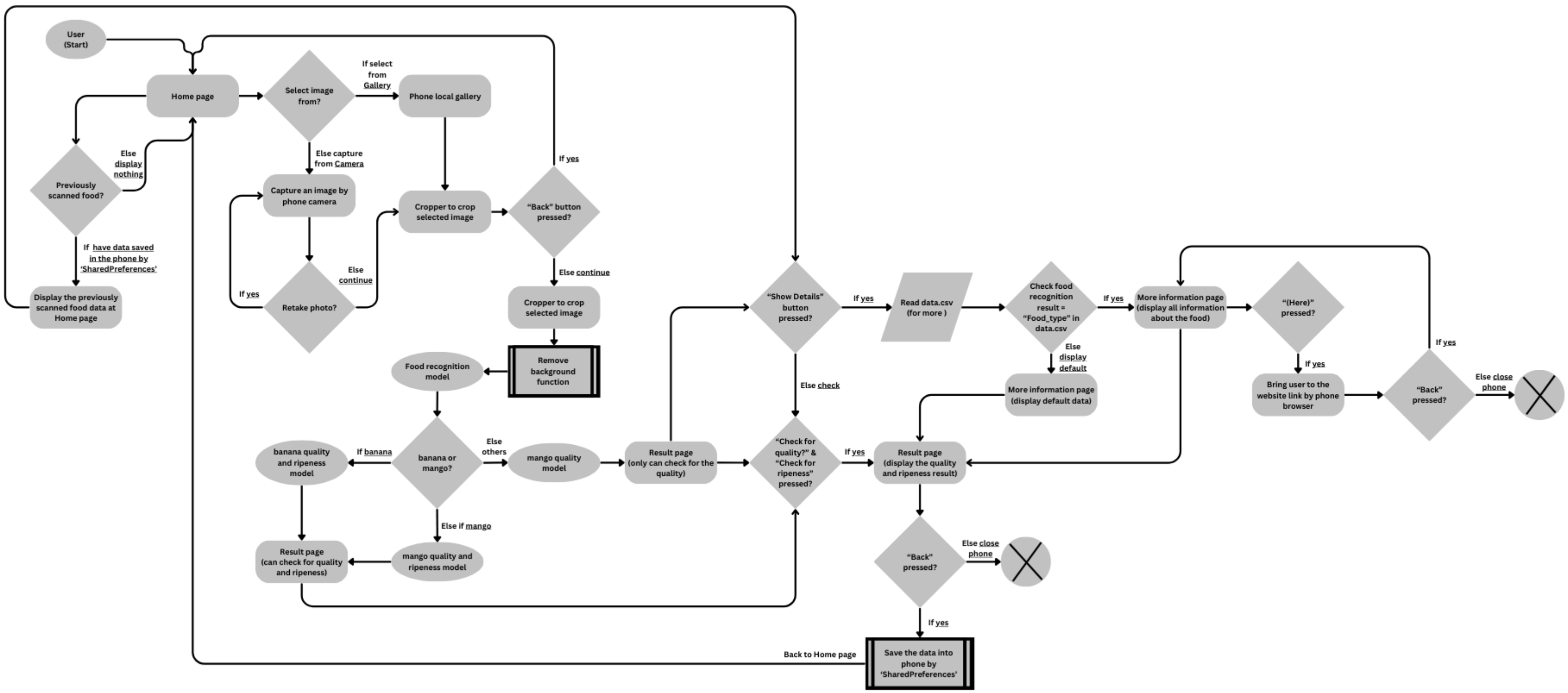


Figure 4.2.1 System flowchart

Here's a breakdown of the process of how the system work based on Figure 4.2.1 above:

1. **Start (User Interaction Begins)**
    - User starts the app on their mobile device, beginning the flow from the home page.

2. **Home Page (Main Menu)**
    - **i. Selecting Image Source**
        - **Camera**
            - Users capture food image via their phone's camera.
            - After capture the image:
                - If user pressed ✕ will recapture again the image.
                - Else if ✓, then forwarded this capture image to cropper module.
        - **Gallery**
            - Users select images from their gallery.
            - The image is then forwarded to the cropper module.
    - **ii. Previously scanned food**
        - The app checks if there is any previously scanned food data stored in the phone's by '**`Shared Preferences`**'.
            - If YES, it displays the previously scanned food data on the home page.
                - User can press "Show Details" button to view more information about that food.
            - Else, there will be empty with no display anything for that part.

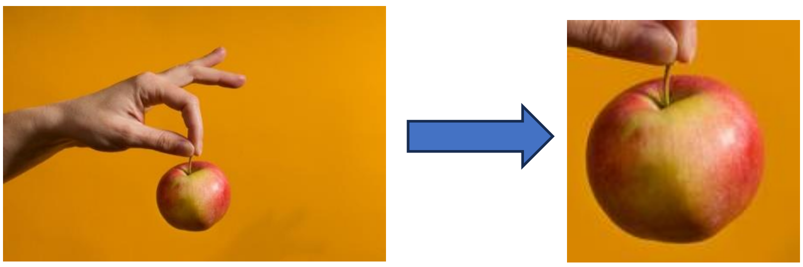
3. **Cropper module**
    - In this module, user can either crop/rotate/flip the uploaded image for increase the accuracy prediction result of the food image. This action is optional, user can do nothing and just proceed to the next step by clicking "NEXT" button.
    - Below here is an example an image why cropper is useful for increasing the correct result accuracy:
        - 

Figure 4.2.2 Image process module [41]

- This is done by "Android-Image-Cropper", an open-source form GitHub [42].
- With the above example the cropped image can be more focus on the food item without disturbing by the background.

- **During cropper page:**
  - If user pressed the "BACK" button, then will take back to the home page.
  - Else if user pressed the "NEXT" button, then will passing the cropped image to Remove background module, which develop by Rembg.

4. Remove background module

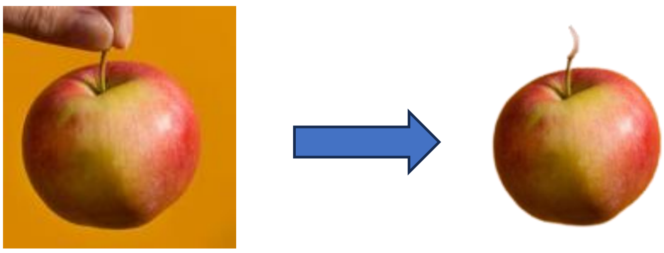
- In this module the cropped image will be remove the background by Rembg.
- e.g.
  - 

Figure 4.2.3 Image process module [41]

- Then the image will pass to Food Recognition Model.

5. Food Recognition Module

- In this module, the processed image will be predicted by developed Food Recognition Model.
- The predicted result will be passed to check it is banana or mango.
  - If is banana or mango then will trigger their quality and ripeness models to check the quality and level of ripeness.
  - Else will only trigger mango's quality model to check the quality. This mango quality model is well to predict the quality for different food that not only mango.
- The results will be passed to the Result Page to display.

6. Banana and Mango Quality and Ripeness Module

- **Quality:**
  - Banana Quality Model will only trigger if the predicted result from Food Recognition Model is "banana".

- Else if the predicted result is "mango" or others result then will trigger Mango Quality Model.
- **Ripeness:**
    - Banana Ripeness Model will only trigger if the predicted result from Food Recognition Model is "banana".
    - Mango Ripeness Model will only trigger if the predicted result from Food Recognition Model is "banana".
- These predictions result will be pass to Result Page to display.

7. **Result Page**
    - This page is to display the quality and ripeness results and a short briefing "More Information" related to the predicted food to user.
    - **Condition:**
        - When the predicted result from the Food Recognition Model is "banana" or "mango" then will display "Check for quality?" and "Check for ripeness?" for user to interact with and get the quality and ripeness results.
        - Else will only display "Check for quality?" for user to interact with and get the quality result.
    - **The "More Information" part:**
        - Firstly, here will need to load and read the data.csv to get the related data to display.
            - The data will be need to get is based on if the predicted result from the Food Recognition Model equal to the Food_type in data.csv, then its row data will be read and get to display.
        - Will display the predicted result from the Food Recognition Model and vitamin details of the food.
    - When the back action trigger by user then the data and result will be saved into phone local storage by '`SharedPreferences`' so that the results can be displayed under "Previous scanned food" at Home Page.
        - Data saved includes image after cropper (as original image), removed background image (as processed image), result from Food Recognition Model, quality result, ripeness result (if don't have then will be null as default), vitamin details from "More Information" part.

8. More Information Page

- This page will be trigger when user clicks on “Show Details” from Home Page under “Previous scanned food” part or Result Page under “More Information” part.
- After user clicked the “Show Details” button, then will bring user to this More Information Page.
- In this page, system will load and read the data.csv and check the predicted result get from Food Recognition Model equal to Food_type in data.csv.
    - If found, then will read the row data for that food type to display at this page.
        - **Here user can view these information:**
            - The name of the food type
            - Vitamin details
            - The comparison good and bad quality image of the food type. **Example:**
                - 

Figure 4.2.4 Good and bad quality image

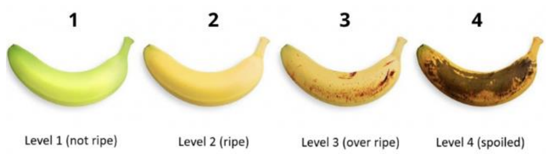
            - Common way to check is 'Good' quality?
            - Common way to check is 'Bad' quality?
            - The comparison of ripeness level image. This will only display when the food type is equal to ‘banana’ or ‘mango’. Else will not display this part. **Example:**
                - 

Figure 4.2.5 Level of ripeness image

            - Provided pre-defined links for user to browser more information that related to the food type. Most of the links are find from wikiHow (because this platform provide step by step with a lot images to teach the reader) and helpmechoosemyfruitandveg.com by Moe Salama [43]. **Example:**

- How to pick?

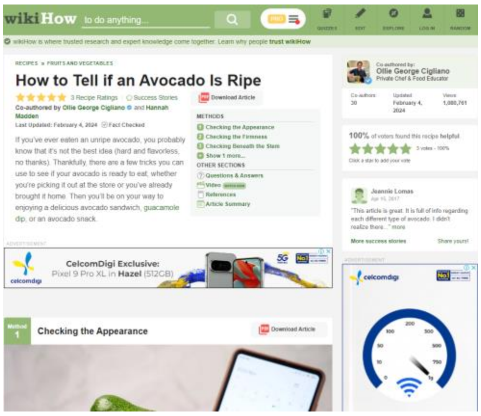


Figure 4.2.6 Example for Avocado [44]

- How to know gone bad?

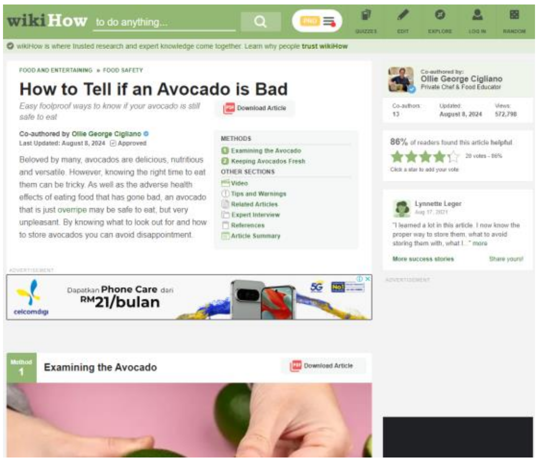


Figure 4.2.7 Example for Avocado [45]

- More information details.

Figure 4.2.8 Example for Banana [46]

- Else not found, then this page will display the default null value.

9. End when user exits the app.

CHAPTER 5
System Setup and Operation

This chapter focuses on the system setup dan operation for Food Quality Recognition Mobile Application development. It presents what the hardware and software setup, setting and configuration, system operation, and implementation issues and challenges. These offered insights into the application’s design and functionality.

5.1 Hardware Setup

The hardware involved in this project consists of a laptop and mobile device for real-time testing. The laptop is used to perform various task such as data pre-processing, model training and application development. The specifications of the laptop used in this project are detailed in Table 5.1.1

Description	**Specifications**
Model	HP Laptop 14s-dk0xxx
Processor	AMD Ryzen 5 35000 U with Readeon Vega Mobile Gfx 2.10GHz
Operating System	Windows 11
Graphic	AMD Radeopn(TM) Vega 8 Graphics
Memory	12GB (4 + 8)
Storage	256GB

Table 5.1.1 Specifications of laptop

5.2 Software Setup

The software involved in this project 2 includes Figma, Visual Studio Code, Android Studio, Anaconda, and GitHub.

5.2.1 Figma

Figure 5.2.1 Logo of Figma

Figma is used to design an overview of the mobile app, which can help create a simple prototype of the application [47].

5.2.2 Visual Studio Code

Figure 5.2.2 Logo of Visual Studio Code

Visual Studio Code is used for developing the core system (food quality recognition system), including preprocessing the dataset, model training, model testing etc. [48].

5.2.3 Android Studio

Figure 5.2.3 Logo of Android Studio

Android Studio is used to develop the mobile application so that to apply the developed core system that can be used by mobile phone with pretty design and friendly user experience. [49].

5.2.4 Anaconda

Figure 5.2.4 Logo of Anaconda

Anaconda is used to create a new environment for developing the system. The cores system was using python 3.8.16 which an environment that created by Anaconda.

5.2.5 GitHub

Figure 5.2.5 Logo of GitHub

GitHub is used to save/backup every changing or version of development/project. It is easy for the process and version control of development and also avoiding issue that sudden the device dies or data missing. Therefore, any bad issues happen still can clone to the GitHub repository and continue the development. Example, both model development and mobile application development are saved/backup on GitHub:

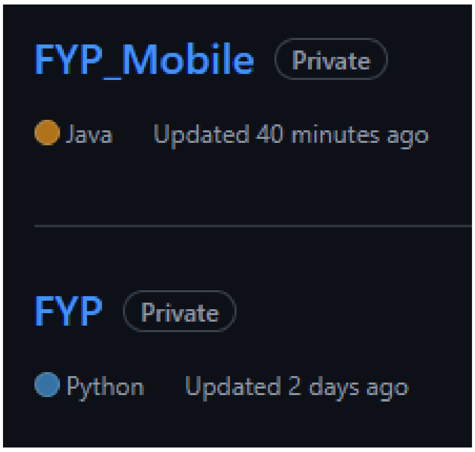


Figure 5.2.6 Screenshot of the developments/projects uploaded on GitHub

5.3 Setting and Configuration

5.3.1 Model Development (Visual Studio Code)

1. Create environment and install all the necessary package

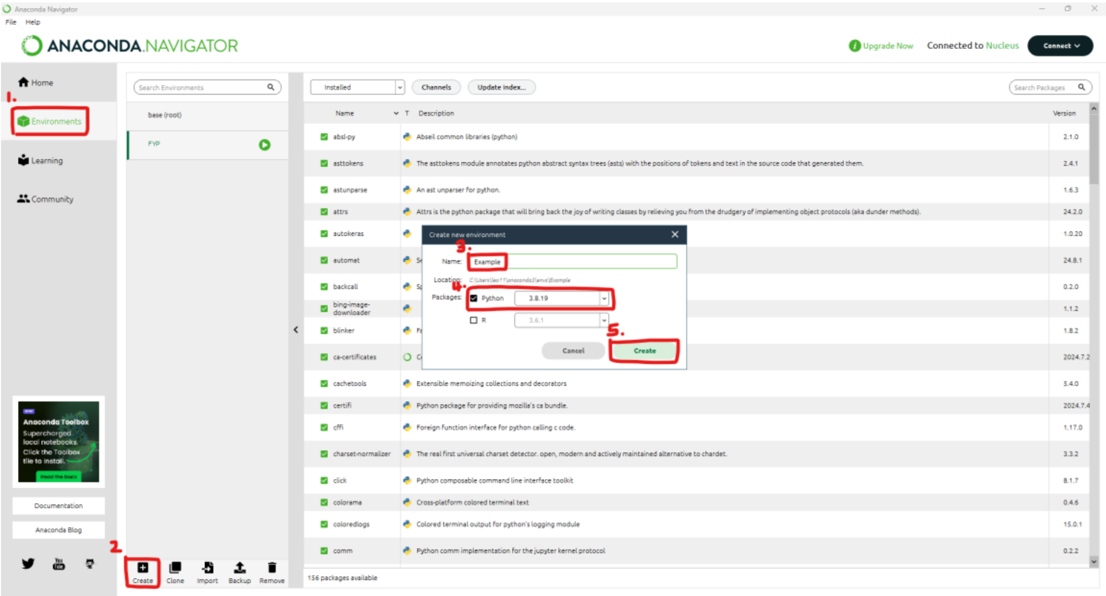


Figure 5.3.1 Create development environment

- Environment -> Create -> Name (Define a name) -> Python (Choose version 3.8.19) -> Create.

2. Select/apply the environment

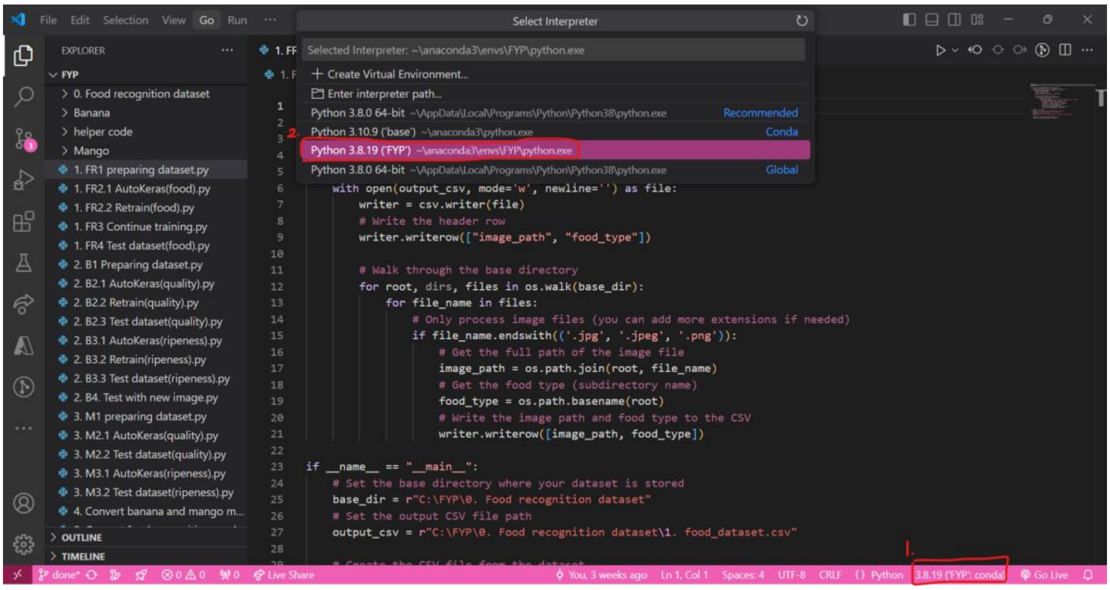


Figure 5.3.2 Create development environment

- In the python file, the right down corner click and change to the created environment.

3. Installation python packages

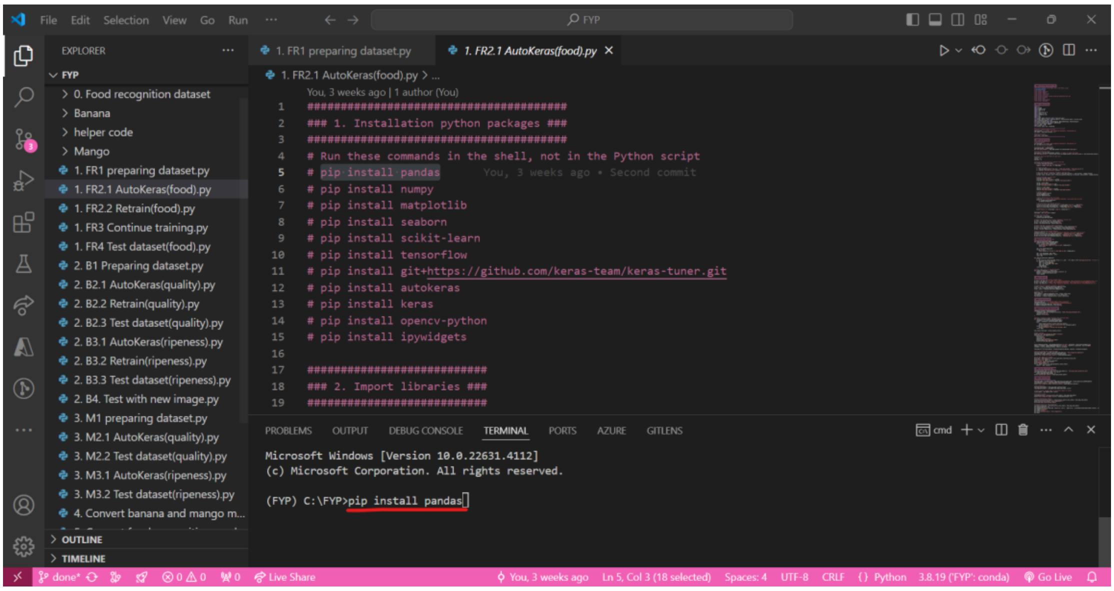


Figure 5.3.3 Create development environment

- Packages need to be install:
  - **`pip install pandas`**
  - **`pip install numpy`**
  - **`pip install matplotlib`**
  - **`pip install scikit-learn`**
  - **`pip install tensorflow`**
  - **`pip install git+https://github.com/keras-team/keras-tuner.git`**
  - **`pip install autokeras`**
  - **`pip install keras`**
  - **`pip install opencv-python`**
  - **`pip install ipywidgets`**
  - **`pip install bing-image-downloader`**

5.3.2 Mobile Application Development (Android Studio)

1. Create new project

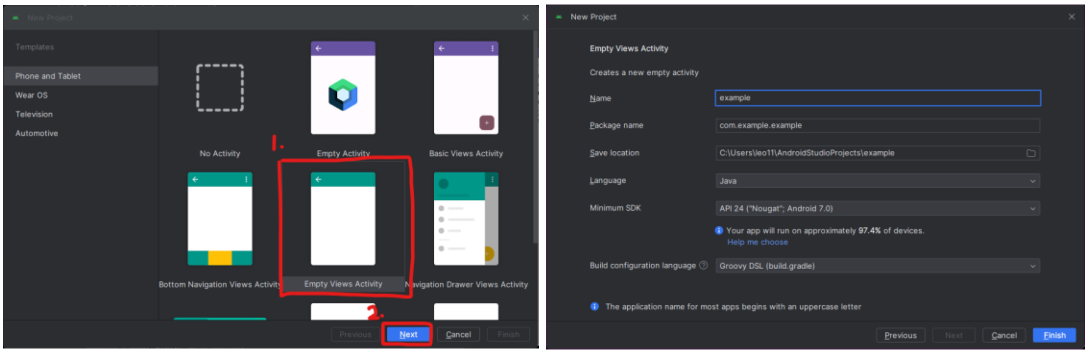


Figure 5.3.4 Create new project

- Select "Empty View Acitivity".
- Optional (Define new one or continue with the default).
- Click "Finish" to create the project.

2. Manifest

```
<uses-sdk tools:overrideLibrary="dev.eren.removebg" />

<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />

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

Figure 5.3.5 Manifest

- Add these before the "<application".
- "**uses-sdk**" is used for remove background module.
- "**uses-features**" is used for capture image by camera.
- "**uses-permission**" is used for getting permission for accessing:
  - Internet
  - Read and write external storage
  - Camera

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="${applicationId}.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths" />
</provider>
```

Figure 5.3.6 Provider

- Add this provider in the “<application”.
    - This is a file provider that allow the app to securely share files with other apps by creating a content URI, instead of exposing the file system path directly. This is essential for sharing files (images from gallery and took from camera) between the app while maintaining privacy and security, as app shouldn’t access each other’s file systems directly.

3. build.gradle (:app)

- **`plugins { id ‘org.jetbrains.kotlin.android’ }`**. Added this to apply Kotlin Android plugin so that can using Kotlin instead of Java.
- **`buildFeatures { mlModelBinding true }`**. Adde this to enable TensorFlow Lite model binding.
- **`kotlinOption { jvmTarget = “1.8” }`**. Added this for the required when using Kotlin and want to set the JVM target version.
- **`dependencies {`**

```
// for Tensorflow Lite:
implementation 'org.tensorflow:tensorflow-lite-support:0.4.4'
implementation 'org.tensorflow:tensorflow-lite-metadata:0.4.4'
implementation 'org.tensorflow:tensorflow-lite-gpu:2.16.1'
// for cropper:
implementation 'com.vanniktech:android-image-cropper:4.6.0'
// for remove background:
```

```
implementation
'com.github.erenalpaslan:removebg:1.0.4'
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.8.1'
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.8.1'
// for Kotlin coroutines:
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.8.1'
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.8.1'
// for CSV file handling:
implementation 'com.opencsv:opencsv:5.5.2'
```

}. Added these implementations so that can import the library and apply them into app.

4. **build.gradle ('project')**
   - **plugins { id 'org.jetbrains.kotlin.android' version '2.0.20' apply false }**. Added this to apply Kotlin.

5. **setting.gradle ('project')**
   - **pluginManagement { repositories { maven { url 'https://jitpack.io' } } }**. Added this for the remove background.
   - **dependencyResolutionManagement { repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS) repositories { maven { url 'https://jitpack.io' } } }**. Added this for the remove background.

5.4 System Operation (with Screenshot)

5.4.1 UI Prototype (Created by Figma)

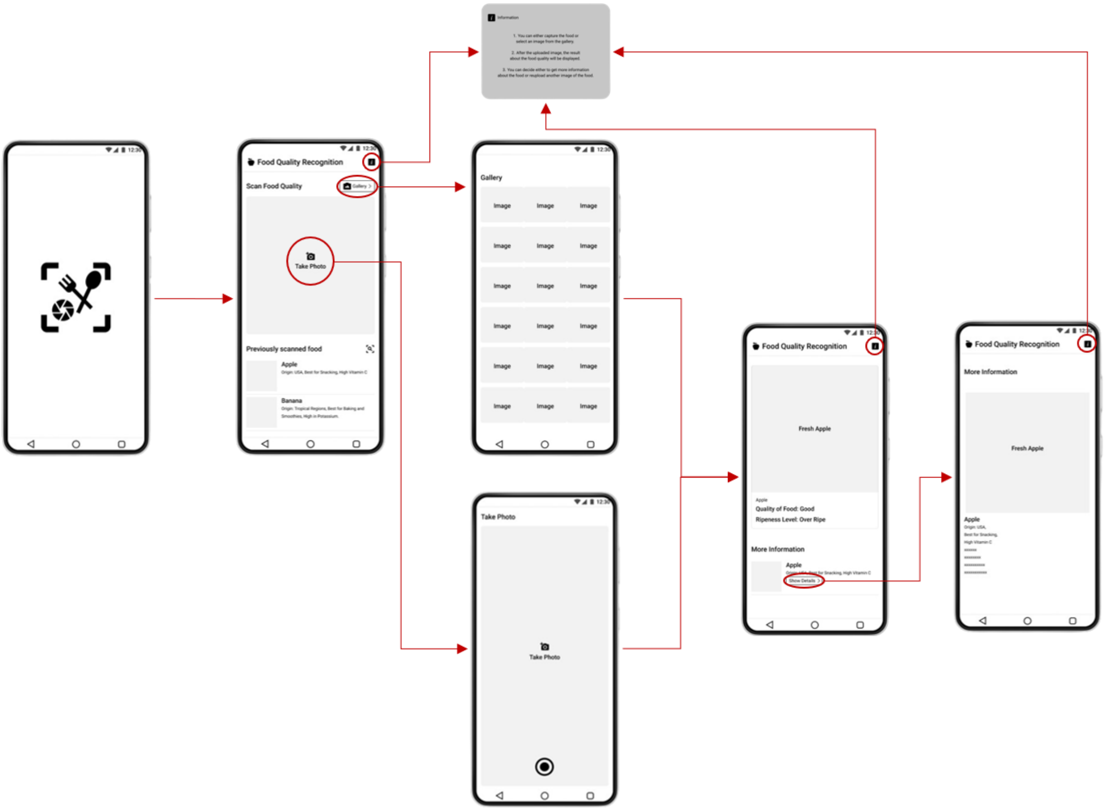


Figure 5.4.1 UI

The above Figure 5.4.1 is the UI prototype design in project 1 that created by Figma. It is homelessness information app that sketched a proto-type UI that designed with Figma. The screen mock-up incorporates a UI design for user interaction features that begin with opening the app and concludes with recognizing the food quality by the provided UI. It has the screen splash of the application where home page and image capture or image gallery page are provided. As well as food quality recognition page (result page) and more information page.

5.4.2 Mobile Application Operation View (Android Studio)

1. Home page:

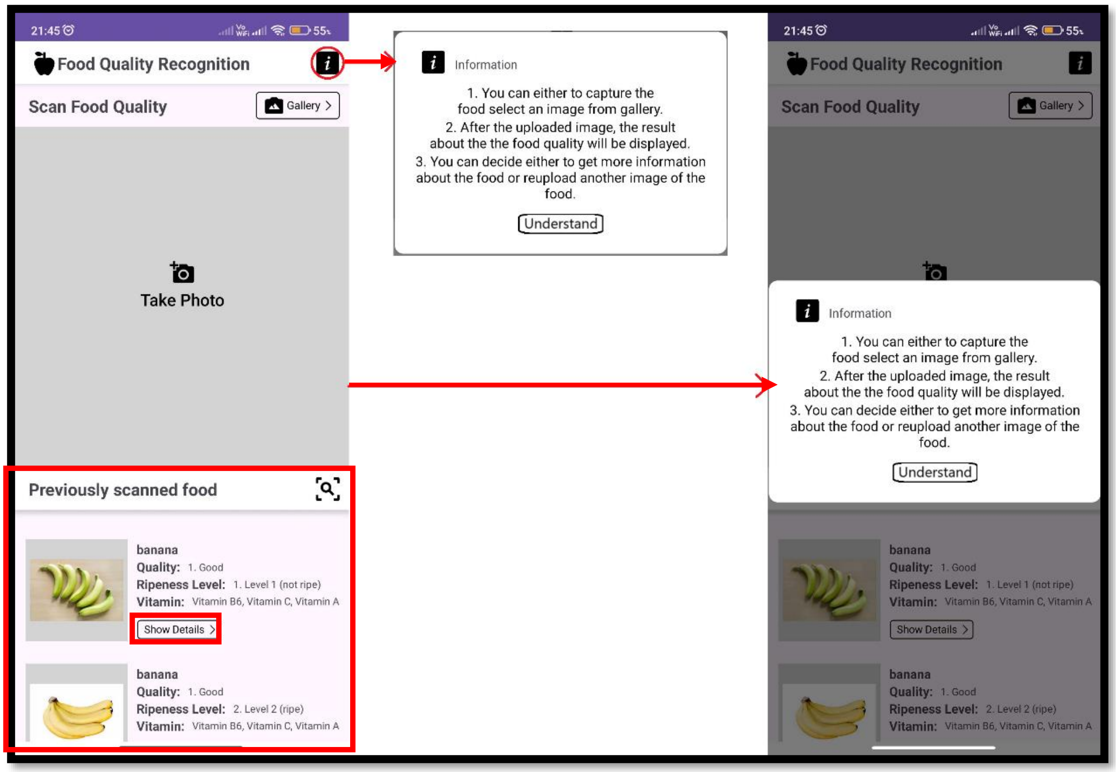


Figure 5.4.2 Home Page

When this icon “ i ” is being clicked, an information dialog box containing the following information will get displayed at the centre of the page. After the establishment of this box, the detailed instructions and guidelines that will help user to use the tool properly will be included in this box.

Furthermore, the users will be able to view the “Previously scanned food” items at the Home Page. The “Show Details >” button for each scanned food item will bring user navigate to More Information Page to get more information that related to the food item. Thus, it is the attribute through which users can check their record of the scanned results and more information be available to view at any time.

The "Take Photo" and "Gallery" options will help him or her navigate to either camera to capture a new food image or an image from their gallery. Having a chosen image placed into the system for processing in the end, the user will decide depending on an outcome page.

2. Take Photo or Gallery

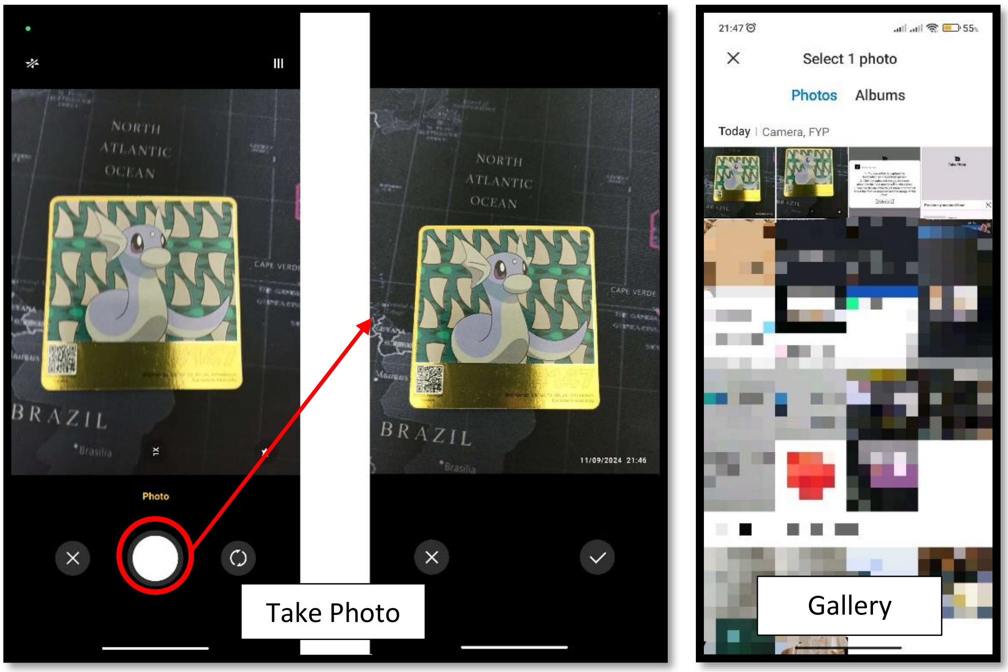


Figure 5.4.3 Image capture page and Gallery page

Users have two methods to input food images into the system. They can either capture a photo directly using the camera or select an existing image from their gallery.

1. Copper

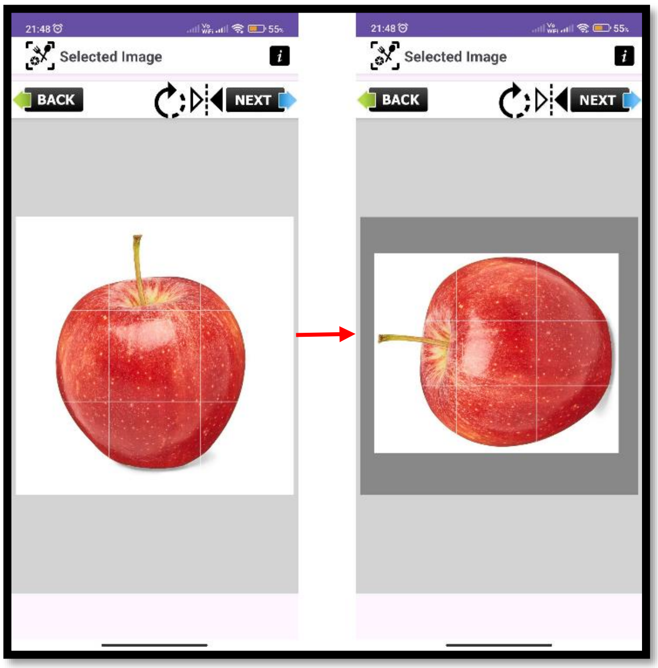


Figure 5.4.4 Cropper

In this page user can choose to crop, rotate right 90 degree each time, flip horizontally. This section is optional, user can click the "" button to proceed.

2. Remove Background

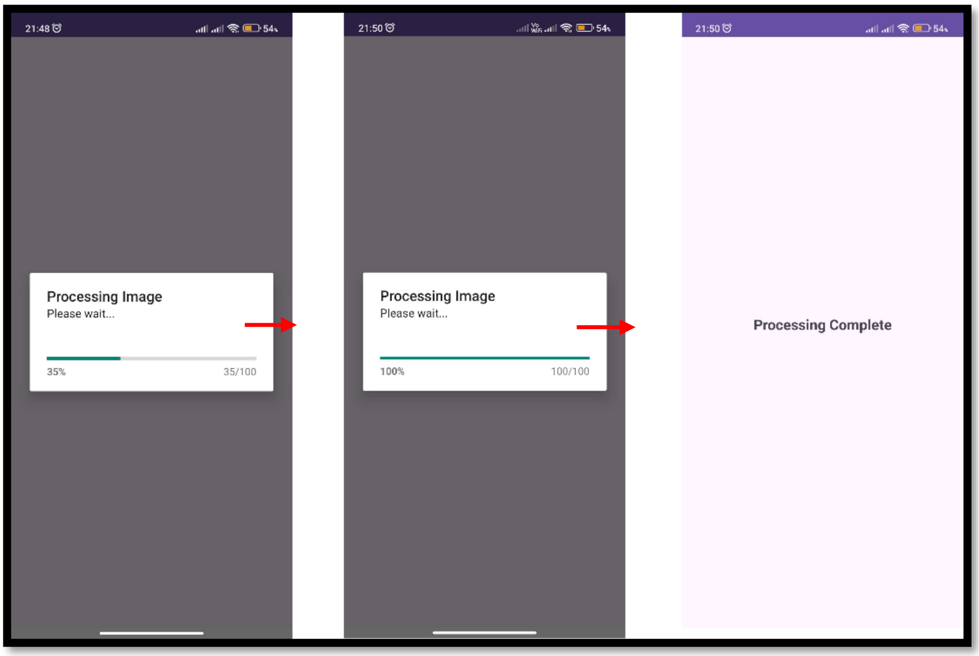


Figure 5.4.5 Remove background

The cropper image will be passed to remove background module to remove the background of the image. In this page the process is automatically run after user click the " 

" button from the cropper page. Also, will display this loading box for user knows it was in processing please wait. It will start from 0% until 100% in 1 to 2 seconds. After it the processing completed, then will automatically passed the removed background image to the next step and also will bring user intent to Result Page.

3. Result Page

The outcome of the Food Recognition, Quality and Ripeness Models' result will be displayed on the result page.

a. Example 1 (Quality and Ripeness Results)

- Only when the result return from Food Recognition Model == 'banana' or 'mango' then can check for their quality and ripeness results by loading their quality and ripeness models.
- The user only clicked "Check for quality?" and "Check for ripeness level?" then the quality and ripeness results will be displayed. Else if not clicked and exit this page then the results will be saved as null value. Then under "Previous scanned food" part at Home Page will be display as "Forgot to click to check the result". Example:

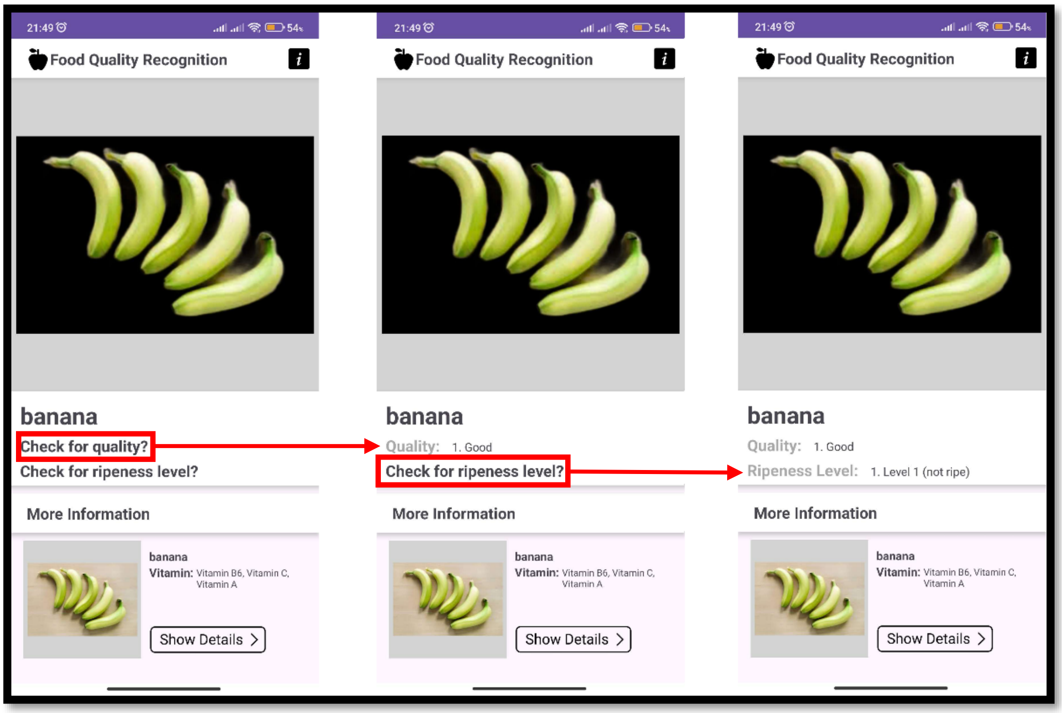


Figure 5.4.6 Example 1

b. Example 2 (Quality Results only)

- When the result return from Food Recognition Model !== 'banana' or 'mango', then can only check for their quality result by loading the Mango Quality Model.
- Same as Example1 the user only clicked "Check for quality?" then the quality result will be displayed. Else result will be saved as null value. Then under "Previous scanned food" part at Home Page will be display as "Forgot to click to check the result". Example:

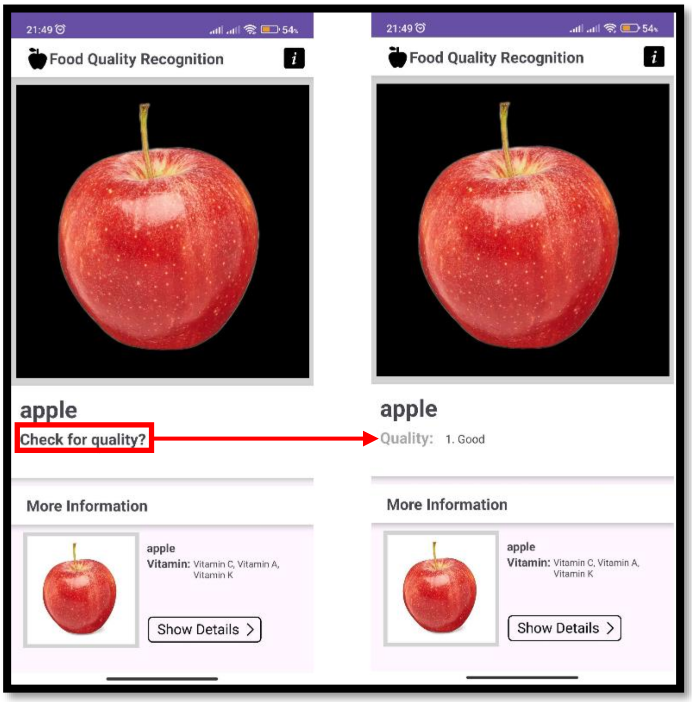


Figure 5.4.7 Example 2

Additionally, users will have the option to click "Show Details". This button will bring user navigate to a More Information Page where user can access additional information about the recognized food.

4. More information page

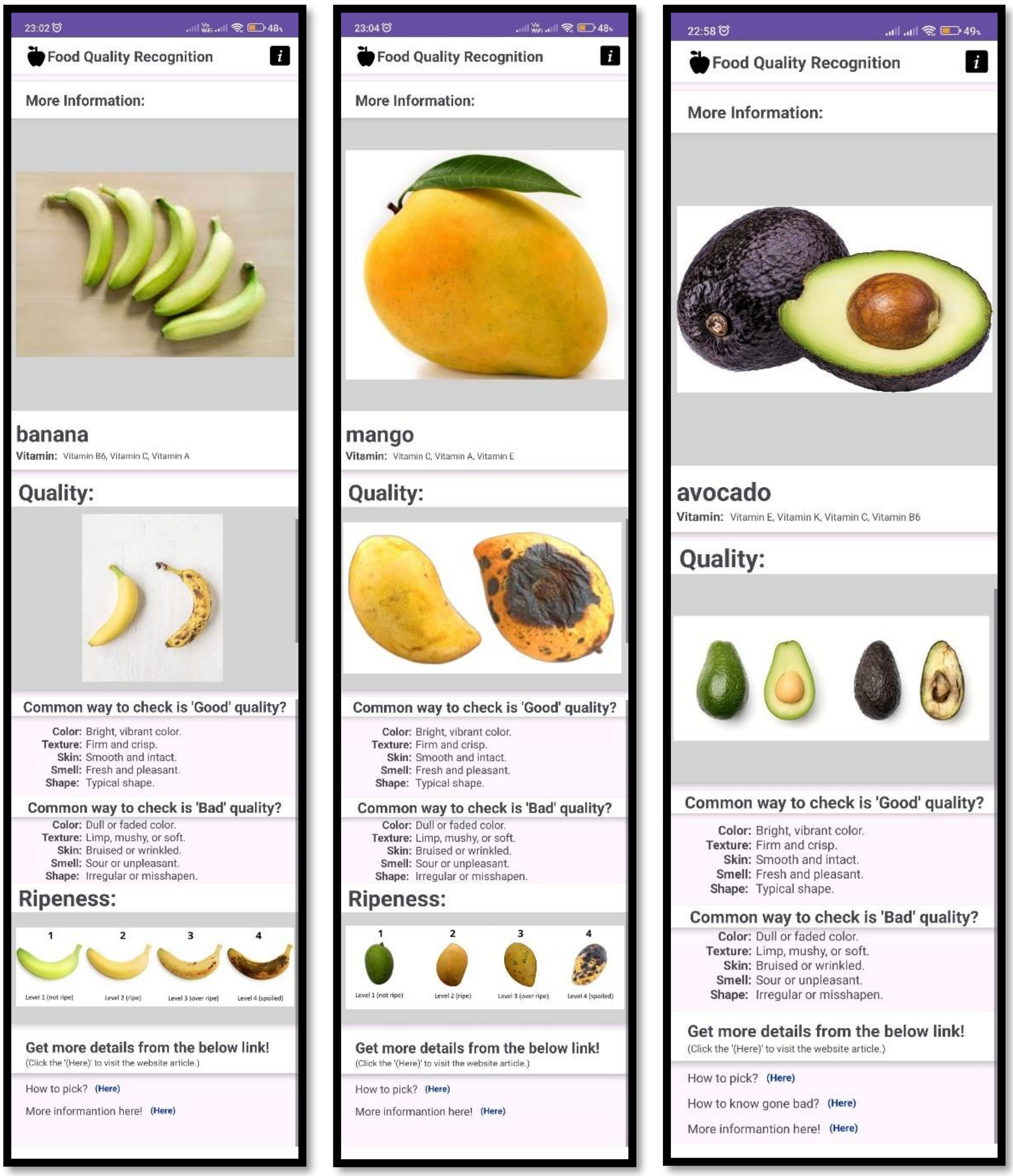


Figure 5.4.8 More information page

In this More Information Page, the information will be display includes the name of the food, vitamin details, comparison good and bad image, common way to check is 'Good' and 'Bad' quality, 4 different levels of ripeness image and pre-defined links for user to browser more information about the food.

Since only 'banana' and 'mango' will load their ripeness to check the level of ripeness, so in More Information Page will display their 4 different levels of ripeness. Then for other result will only showing the comparison good and bad images of the food.

Example the links:

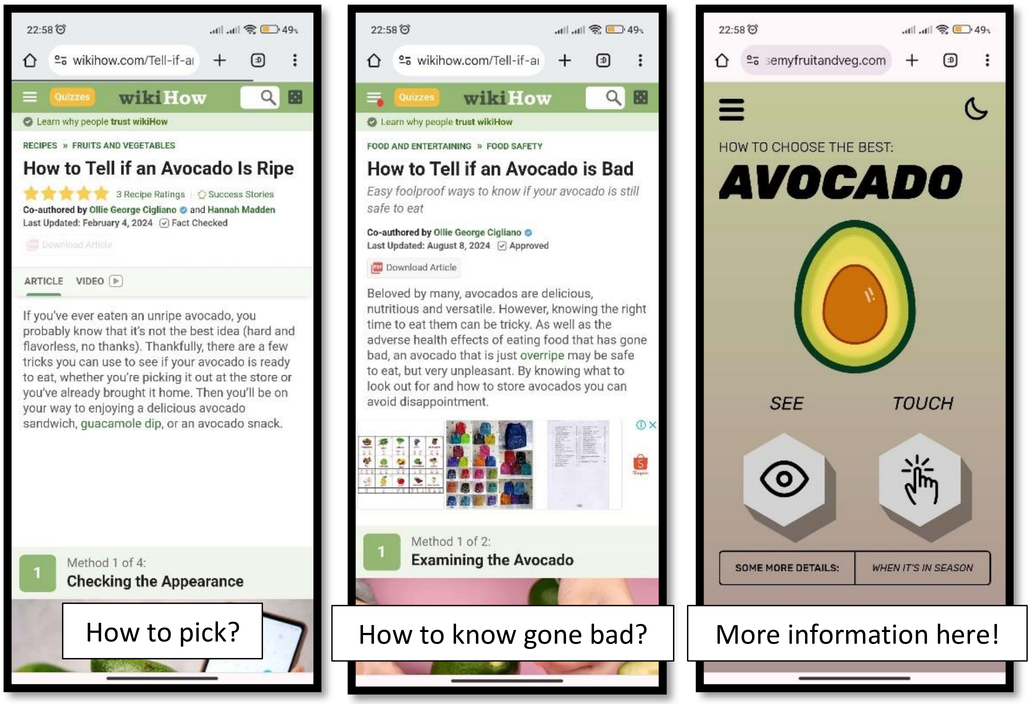


Figure 5.4.9 Example for Avocado [44] [45] [50]

5.5 Implementation Issues and Challenges

5.5.1 Import the converted model (TFLite) into Android Studio

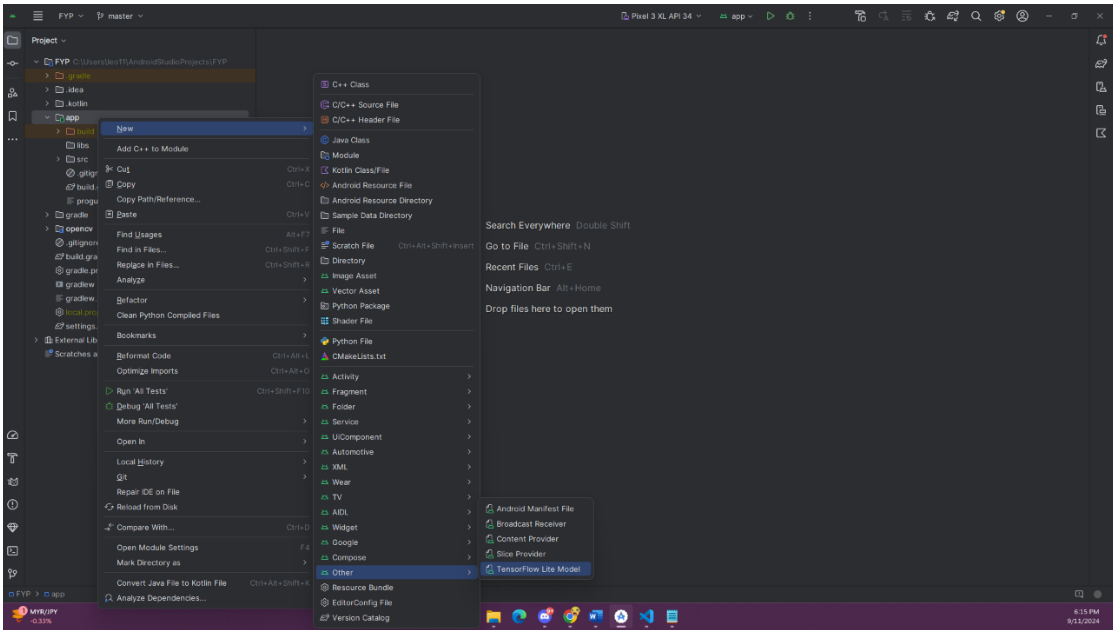


Figure 5.5.1 Import TFLife model into Android Studio

1. **File size issue**
   - The models' file size are over the maximum able file size = 200MB so need to reduce the size during convert section. To fix this problem, need to add this line "**`converter.optimizations = [tf.lite.Optimize.DEFAULT]`**". After this, the file size of the model become 68,540KB.

2. **Invalid file**
   - Converted TFLite model of Food Recognition Model cannot import into Android Studio due to "**`'dtype': <class 'numpy.float64'`**". Android Studio only accept with 'float32'. To fix this problem, need to add these:
     - **Rebuild the Model with New Input Layer:**
     -

```
# Rebuild model with a new input layer of dtype float32
inputs = tf.keras.Input(shape=(256, 256, 3), dtype='float32')
x = model(inputs, training=False)
model = tf.keras.Model(inputs=inputs, outputs=x)
```

Figure 5.5.2 Rebuild the model with new input layer

- **Define a Representative Dataset for Quantization:**

```python
# Representative dataset generator
def representative_dataset_generator():
    for _ in range(100):  # Adjust the number of samples if needed
        yield [np.random.uniform(0.0, 1.0, [1, 256, 256, 3]).astype(np.float32)]

converter.representative_dataset = representative_dataset_generator
```

Figure 5.5.3 Define a representative dataset for quantization

- After these, the converted TFLite model can be import into Android Studio.

The above issues/challenges were solved by referring from this "How to reduce the size of the tflite model below 200mb while converting h5 to tflite" discussion on GitHub [51].

5.5.2 Remove background module problem

Since Rembg is used during model development, it also needs to be applied in Android Studio for processing user-uploaded images. However, Rembg is an open source that in Python language and isn't supported in Android Studio. This challenge almost led to abandoning this step, meaning the model would receive image with unremoved background for prediction. Although this could result in lower prediction accuracy, it would still allow the system to function.

Developer tried 3 different approaches and ultimately succeeded in integrating Rembg into Android Studio, ensuring that the prediction accuracy matched the results achieved in the Visual Studio Code environment.

1. **chaquopy (run python file in Android Studio)**
   - chaquopy is a Python SDK for Android. It provides everything need to include Python components in an Android app [52]. Due to this, developer try to import the functional Rembg python file by following step by step from LaptopML on YouTube [53]. This python file will be run while get the user-uploaded image from the app and return the removed background image to the app. Unfortunately, it fails by cannot receive the return removed background image by the python file.

2. **U2Net model**
   - Since Rembg also use this U2Net model for remove the background image, so developer try find the model and download it. The model was downloaded from the Rembg repository on GitHub [29]. The downloaded model is onnx model so need to convert into TFLite so that can apply in Android Studio. Unfortunately, the convert process was fail due unknown reasons.

3. **removebg (Open source on GitHub allow applying Rembg into Android Studio)**
   - After two failed attempts, the developer, on the verge of giving up, discovered an open-source solution which provide the background removal library for Android. Upon testing it, they found that it produced the same results as in the

Visual Studio Code environment. However, another challenge arose: the Android Studio project was based on Java, while the open-source solution was written in Kotlin. The developer attempted to convert the Kotlin code to Java but encountered issues because the data returned was in Kotlin, which Java couldn't handle. As a result, the developer had to configure Kotlin within the Android Studio project to enable the background removal functionality. Below is an example of how to integrate Java and Kotlin in a single Android Studio project:

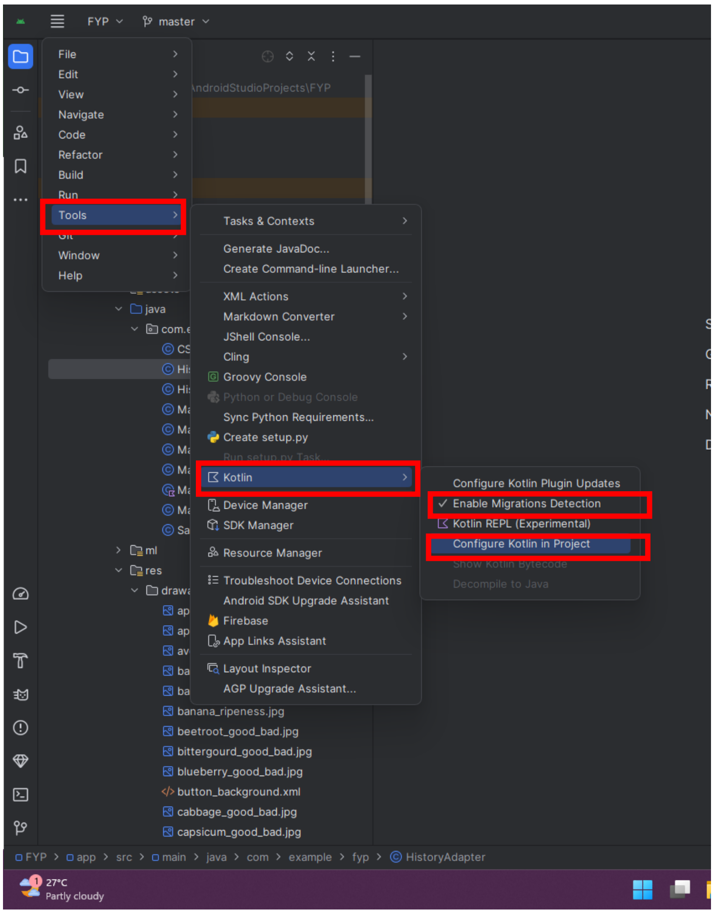
- 

Figure 5.5.4 Configure Kotlin in project

5.6 Concluding Remark

5.6.1 Summarize of the project

1. **Developed models include:**
   - Food Recognition Model, can recognize 50 types of fruits and vegetables.
   - Banana Quality Model & Mango Quality Model, detect good or bad quality.
   - Banana Ripeness Model & Mango Ripeness Model, detect 4 different levels of ripeness includes: 1. Level 1 (not ripe), 2. Level 2 (ripe), 3. Level 3 (over ripe), and 4. Level 4 (spoiled).

2. **Mobile application developed 5 pages:**
   - Home Page
   - Cropper Page
   - Remove Background Page
   - Result Page
   - More Information Page

5.6.2 Highlight key challenges and solutions

1. **Fail import converted TFLite models into Android Studio**
   - Fix by reduce the size below 200MB and make sure that the dtype is "**`'dtype': <class 'numpy.float32`**".

2. **Remove background module problem**
   - Implement chaquopy but fail due to cannot receive the return moved background image by the python file.
   - Import U2Net model but fail due to failing in covert the U2Net.onnx to TFLite model.
   - Implement removebg an open-source solution which provide the background removal library for Android. However, face an issue that the Android Studio project is Java based but this open source is Kotlin language. Tried convert Kotlin to Java but fail due to the return data is in Kotlin so Java cannot handle it. Fix this issue by configure Kotlin into the project.

5.6.3 Future direction

This development project didn't use any APIs or database services so allow it run offline. While this is good feature but it has resulted in the mobile application file size exceeding

500MB above. This is mainly due to the large number of images inserted into the project. Also, the needed data are saved into phone's local storage by '`Shared Preferences`' which this will increase the file size. If the "Previous scanned food" saved a lot of numbers of items, then every time lunch the app may take times to find, read and display the items' data. This will cause bad user experience. However, the solution now to avoid lunch the app takes times is only clear all data, but this will remove the previous scanned data too. Therefore, need to find a way to solve this issue for better user experience.

Secondly, the model currently only recognizes 50 common types of fruits and vegetables but still quite limited. Additionally, there only banana and mango are supported for ripeness detection. In the future, the goal is to expand the model to recognize more types of fruits and vegetables and allow ripeness and quality checks across a wider variety of food types.

CHAPTER 6
System Evaluation and Discussion

This chapter focuses on the system evaluation and discussion. It presents the system testing and performance results.

6.1 System Testing and Performance Metrics

6.1.1 Food Recognition Model

```
32/32 [==============================] - 416s 13s/step
Food recognition Model Accuracy: 0.80
Classification Report (Food recognition):
              precision    recall  f1-score   support

           0       0.92      0.55      0.69        20
           1       0.93      0.70      0.80        20
           2       1.00      0.90      0.95        20
           3       0.86      0.95      0.90        20
           4       1.00      0.75      0.86        20
           5       0.67      1.00      0.80        20
           6       0.94      0.85      0.89        20
           7       0.86      0.90      0.88        20
           8       0.83      1.00      0.91        20
           9       0.86      0.90      0.88        20
          10       0.73      0.80      0.76        20
          11       0.82      0.90      0.86        20
          12       1.00      0.50      0.67        20
          13       0.94      0.80      0.86        20
          14       0.95      0.95      0.95        20
          15       0.89      0.85      0.87        20
          16       0.65      1.00      0.78        20
          17       0.68      0.95      0.79        20
          18       0.90      0.95      0.93        20
          19       0.82      0.90      0.86        20
          20       0.81      0.85      0.83        20
          21       0.78      0.70      0.74        20
          22       0.93      0.65      0.76        20
          23       0.82      0.90      0.86        20
          24       0.81      0.85      0.83        20
          25       0.68      0.95      0.79        20
          26       0.90      0.90      0.90        20
          27       0.47      0.85      0.61        20
          28       0.83      0.50      0.62        20
          29       1.00      0.70      0.82        20
          30       0.92      0.55      0.69        20
          31       0.73      0.80      0.76        20
          32       0.82      0.70      0.76        20
          33       0.85      0.55      0.67        20
          34       0.78      0.90      0.84        20
          35       1.00      0.70      0.82        20
          36       0.95      1.00      0.98        20
          37       0.83      0.95      0.88        20
          38       0.37      0.90      0.52        20
          39       0.83      0.95      0.88        20
          40       0.93      0.65      0.76        20
          41       0.86      0.95      0.90        20
          42       0.88      0.75      0.81        20
          43       0.88      0.75      0.81        20
          44       1.00      0.30      0.46        20
          45       0.64      0.45      0.53        20
          46       0.78      0.90      0.84        20
          47       0.88      0.75      0.81        20
          48       0.83      0.95      0.88        20
          49       1.00      0.80      0.89        20

    accuracy                           0.80      1000
   macro avg       0.84      0.80      0.80      1000
weighted avg       0.84      0.80      0.80      1000
```

1. **First training**
   - Dataset has 50 classes, each class has 100 images, so total 5000 images.
   - Each class splits into 60 images as train set, 20 images as valid set and the left 20 images are test set.
   - Time takes for model training: 20 hours.
   - Model Accuracy: 80%.

Figure 6.1.1 Food recognition model first training

```
32/32 [==============================] - 412s 12s/step
Food recognition Model Accuracy: 0.86
Classification Report (Food recognition):
              precision    recall  f1-score   support

           0       1.00      0.70      0.82        20
           1       1.00      0.75      0.86        20
           2       0.95      0.90      0.92        20
           3       0.95      0.90      0.92        20
           4       0.95      0.95      0.95        20
           5       0.91      1.00      0.95        20
           6       0.90      0.90      0.90        20
           7       0.95      0.90      0.92        20
           8       0.87      1.00      0.93        20
           9       0.95      0.95      0.95        20
          10       0.94      0.80      0.86        20
          11       0.78      0.90      0.84        20
          12       1.00      0.80      0.89        20
          13       0.86      0.95      0.90        20
          14       0.95      0.95      0.95        20
          15       0.85      0.85      0.85        20
          16       0.77      1.00      0.87        20
          17       0.68      0.95      0.79        20
          18       0.95      0.90      0.92        20
          19       0.90      0.90      0.90        20
          20       0.95      0.95      0.95        20
          21       0.83      0.75      0.79        20
          22       0.81      0.85      0.83        20
          23       0.95      0.90      0.92        20
          24       0.61      0.95      0.75        20
          25       0.75      0.75      0.75        20
          26       1.00      1.00      1.00        20
          27       0.94      0.85      0.89        20
          28       0.59      0.65      0.62        20
          29       0.90      0.95      0.93        20
          30       0.93      0.70      0.80        20
          31       0.86      0.60      0.71        20
          32       0.88      0.75      0.81        20
          33       0.92      0.60      0.73        20
          34       0.85      0.85      0.85        20
          35       0.79      0.95      0.86        20
          36       1.00      1.00      1.00        20
          37       0.86      0.95      0.90        20
          38       0.60      0.90      0.72        20
          39       0.82      0.90      0.86        20
          40       0.88      0.70      0.78        20
          41       0.90      0.95      0.93        20
          42       0.79      0.95      0.86        20
          43       0.74      0.85      0.79        20
          44       0.95      0.90      0.92        20
          45       0.92      0.60      0.73        20
          46       1.00      0.90      0.95        20
          47       0.84      0.80      0.82        20
          48       0.94      0.85      0.89        20
          49       0.95      0.95      0.95        20

    accuracy                           0.86      1000
   macro avg       0.88      0.86      0.86      1000
weighted avg       0.88      0.86      0.86      1000
```

2. Second retrain

- Dataset has 50 classes, each class has 100 images, so total 5000 images.
- Each class splits into 60 images as train set, 20 images as valid set and the left 20 images are test set.
- Time takes for model training: 10 hours.
- Model Accuracy: 86%.

Figure 6.1.2 Food recognition model second training

```
157/157 [==============================] - 4777s 30s/step
Model Accuracy on all images: 0.93
Classification Report:
              precision    recall  f1-score   support

        apple       0.94      0.84      0.89       100
      avocado       0.99      0.91      0.95       100
       banana       0.96      0.96      0.96       100
     beetroot       0.98      0.94      0.96       100
 bitter gourd       0.99      0.96      0.97       100
    blueberry       0.91      1.00      0.95       100
      cabbage       0.95      0.95      0.95       100
     capsicum       0.92      0.92      0.92       100
       carrot       0.93      0.99      0.96       100
  cauliflower       0.97      0.98      0.98       100
chilli pepper       0.95      0.90      0.92       100
         corn       0.94      0.95      0.95       100
     cucumber       0.97      0.93      0.95       100
 dragon fruit       0.94      0.98      0.96       100
       durian       0.95      0.96      0.96       100
     eggplant       0.95      0.89      0.92       100
  finger lady       0.89      1.00      0.94       100
       garlic       0.84      0.98      0.90       100
       ginger       0.98      0.96      0.97       100
       grapes       0.94      0.96      0.95       100
        guava       0.96      0.98      0.97       100
     honeydew       0.95      0.89      0.92       100
    jackfruit       0.94      0.94      0.94       100
         kiwi       0.97      0.96      0.96       100
        lemon       0.75      0.97      0.84       100
      lettuce       0.89      0.88      0.88       100
       longan       0.94      0.99      0.97       100
       lychee       0.96      0.96      0.96       100
        mango       0.85      0.82      0.83       100
   mangosteen       0.95      0.98      0.97       100
        onion       0.99      0.84      0.91       100
       orange       0.92      0.73      0.82       100
       papaya       0.95      0.88      0.91       100
        peach       0.98      0.82      0.89       100
         pear       0.97      0.96      0.96       100
         peas       0.90      0.95      0.92       100
    pineapple       0.99      0.98      0.98       100
  pomegranate       0.91      0.99      0.95       100
       potato       0.74      0.91      0.82       100
      pumpkin       0.94      0.96      0.95       100
      radish        0.93      0.85      0.89       100
     rambutan       0.97      0.98      0.98       100
    soy beans       0.87      0.98      0.92       100
      spinach       0.88      0.93      0.90       100
   strawberry       0.99      0.97      0.98       100
  sweetpotato       0.95      0.75      0.84       100
       tomato       0.99      0.96      0.97       100
       turnip       0.94      0.92      0.93       100
   watermelon       0.97      0.95      0.96       100
          yam       0.97      0.97      0.97       100

     accuracy                           0.93      5000
    macro avg       0.94      0.93      0.93      5000
 weighted avg       0.94      0.93      0.93      5000
```

3. Test all images from dataset

- 50 classes, each class tested 100 images so total 5000 images.
- Time takes for test all images: 3 hours.
- Model Accuracy: 93%.

Figure 6.1.3 Food recognition model test dataset

4. Confusion Matrix for test all images from dataset

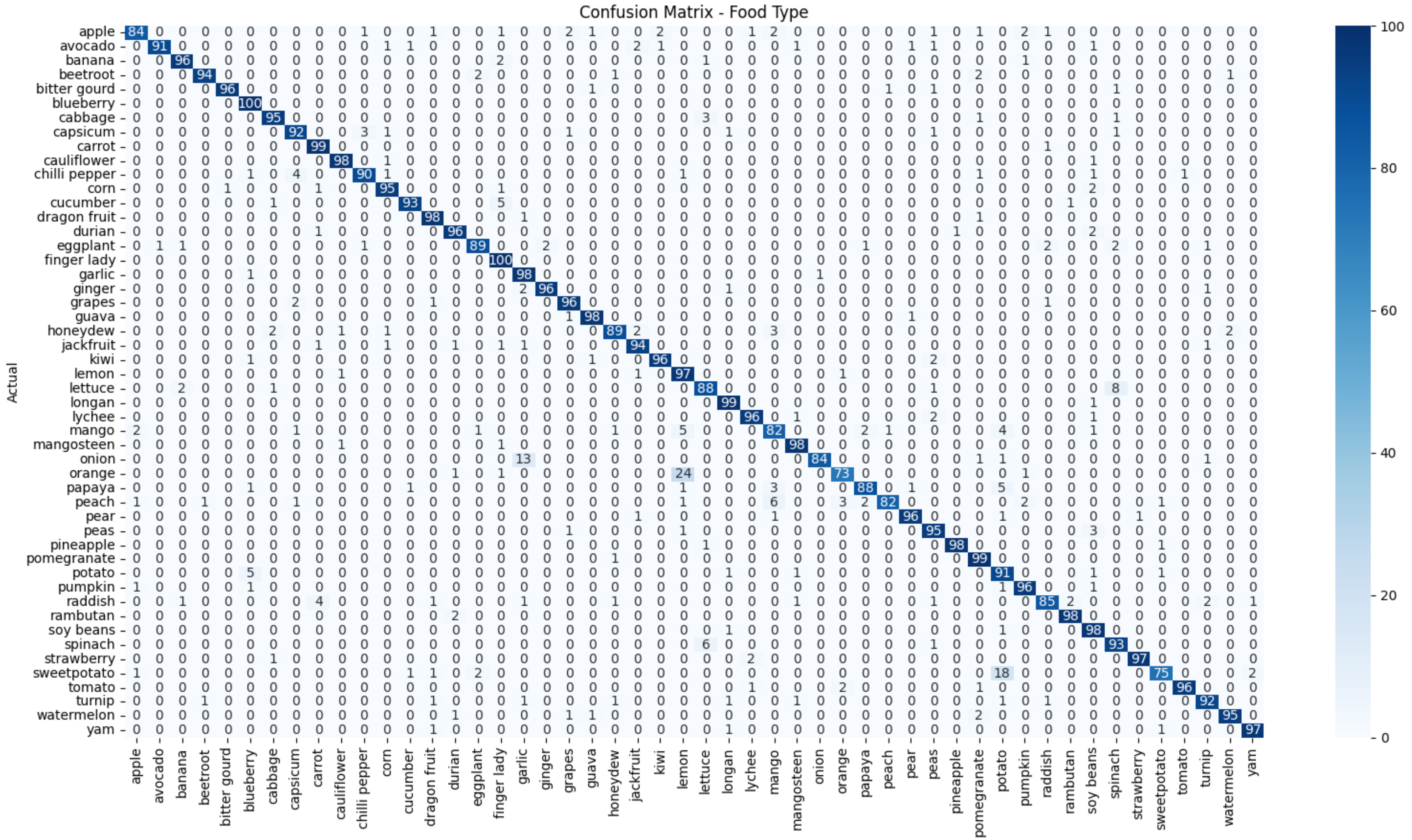


Figure 6.1.4 Confusion Matrix

5. Conclusion based on the above

First time training the model accuracy reach 80%, then the second time retrain improve to 86%. While test all the images form dataset, the accuracy reaches up to 93%. The highest accuracy of food type is blueberry, which reach 100%. The lowest accuracy of food type is orange, which reach 73% because most of the images are predicted as lemon.

6.1.2 Banana Quality Model

1. First time training

```
Training set class distribution: quality
1. Good    120
2. Bad     120
Name: count, dtype: int64
Validation set class distribution: quality
1. Good    40
2. Bad     40
Name: count, dtype: int64
Test set class distribution: quality
1. Good    40
2. Bad     40
Name: count, dtype: int64
Loading image: C:\FYP\Banana\PreprocessedImages\quality\1. Good\image 3_augmented_1.png
Loading image: C:\FYP\Banana\PreprocessedImages\quality\1. Good\image 54_augmented_1.png
Loading image: C:\FYP\Banana\PreprocessedImages\quality\1. Good\image 24_augmented_3.png
```

```
10/10 [==============================] - 58s 5s/step - loss: 0.2689 - accuracy: 0.9500
Quality Model Evaluation Results: [0.2689087688922882, 0.949999988079071]

1/3 [=========>....................] - ETA: 54s
2/3 [===================>..........] - ETA: 19s
3/3 [==============================] - ETA: 0s
3/3 [==============================] - 58s 15s/step
Quality Model Accuracy: 0.50
Classification Report (Quality):
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        40
           1       0.00      0.00      0.00        40

    accuracy                           0.50        80
   macro avg       0.25      0.50      0.33        80
weighted avg       0.25      0.50      0.33        80
```

Figure 6.1.5 Banana quality model first training

- Original dataset only has total 100 images for Good and Bad quality.
- Dataset was increased after data augmentation, so each class become 200 images so total 400 images.
    - So new dataset split into each class 120 as train set, 40 as valid set and the left 40 as test set.
- Time takes for model training: 5 hours.
- Model evaluation results is 95% but the actual accuracy is only 50%.
- This is because all the true labels are predicted as wrong.

2. Second time training

```
3/3 [==============================] - 15s 5s/step - loss: 0.3709 - accuracy: 0.9000
Quality Model Evaluation Results: [0.37093469500541687, 0.8999999761581421]

1/1 [==============================] - ETA: 0s
1/1 [==============================] - 22s 22s/step
Quality Model Accuracy: 0.90
Classification Report (Quality):
              precision    recall  f1-score   support

           0       1.00      0.85      0.92        13
           1       0.78      1.00      0.88         7

    accuracy                           0.90        20
   macro avg       0.89      0.92      0.90        20
weighted avg       0.92      0.90      0.90        20
```

Figure 6.1.6 Banana quality second training

- After fine tune the model can recognize Good and Bad quality.
- Time takes for model training: 1 hour.
- Even though the model evaluation result is 90% and the actual accuracy also 90%.

3. Test all images from dataset

```
1/4 [======>.......................] - ETA: 1:01
2/4 [==============>...............] - ETA: 28s
3/4 [=====================>........] - ETA: 14s
4/4 [==============================] - ETA: 0s
4/4 [==============================] - 52s 11s/step
Quality Model Accuracy on all images: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.97        60
           1       0.91      1.00      0.95        40

    accuracy                           0.96       100
   macro avg       0.95      0.97      0.96       100
weighted avg       0.96      0.96      0.96       100

Total execution time: 150.01 seconds
```

Figure 6.1.7 Banana quality test dataset

- Use the original dataset to test the model performance.
- Time takes for all images: 2.5 minutes
- Model Accuracy: 96%

4. Confusion Matrix for test all images from dataset

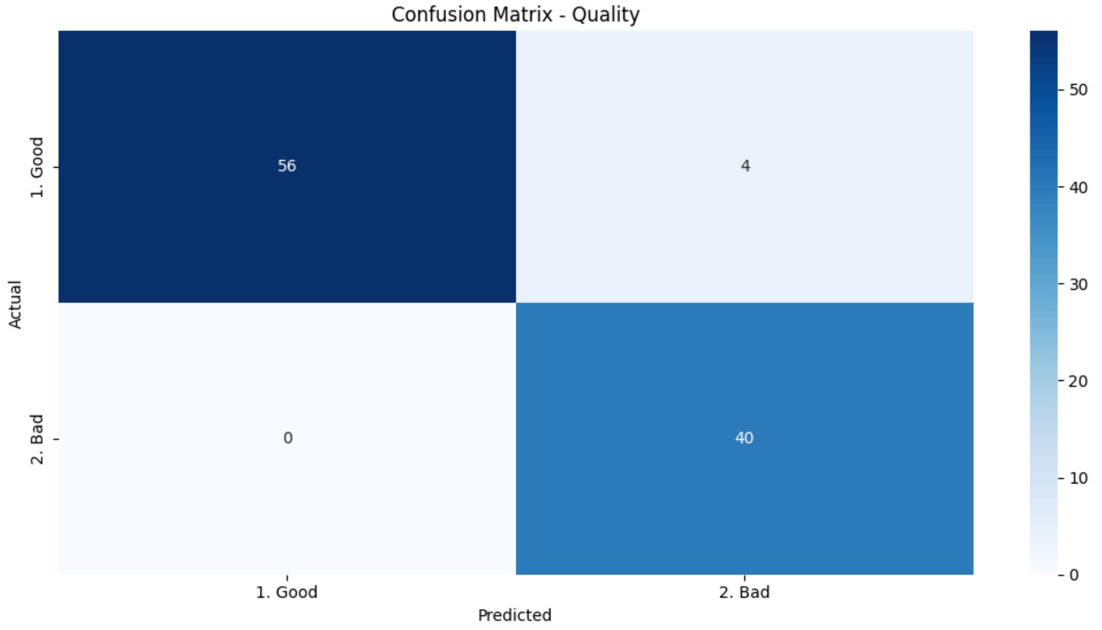


Figure 6.1.8 Banana quality confusion matrix

- In 100 images only 4 images are predicted wrong.

5. Wrong predictions

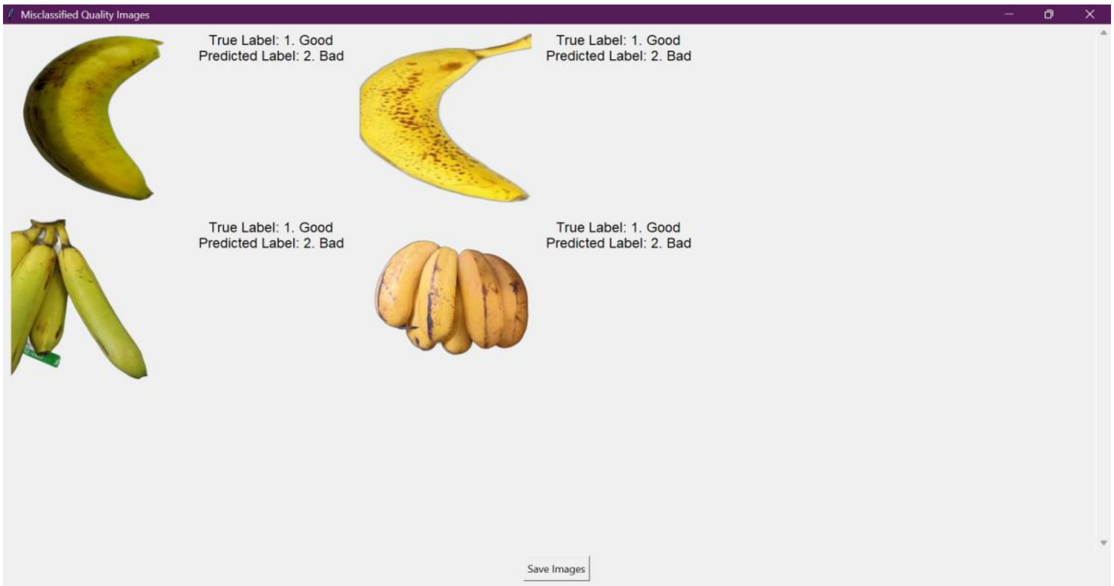


Figure 6.1.9 Model predict different result with actual label

- Above Figure 6.1.9 showing the 4 wrong predictions are which images.

6. Conclusion

The initial training of the banana quality model faced challenges, primarily due to a small dataset of only 100 images split evenly between Good and Bad quality. After data augmentation, the dataset was expanded to 400 images, improving the training set to 120 images per class, with 40 for validation and 40 for testing. However, the first training yielded poor results, with an evaluation accuracy of 95% but an actual accuracy of only 50%, indicating that the model incorrectly predicted all the true labels. In the second round of training, fine-tuning the model improved its ability to correctly classify Good and Bad quality bananas. This reduced the training time to 1 hour and significantly improved both the evaluation and actual accuracy to 90%. Further testing on the original dataset confirmed the model's enhanced performance, with an accuracy of 96% in just 2.5 minutes. The confusion matrix revealed only 4 misclassifications out of 100 images, and the wrongly predicted images were identified and highlighted, illustrating the model’s overall effectiveness with room for slight improvements.

6.1.3 Banana Ripeness Model

1. First time training

```
Training set class distribution: ripeness
1. Level 1 (not ripe)     72
3. Level 3 (over ripe)    72
4. Level 4 (spoiled)      72
2. Level 2 (ripe)         72
Name: count, dtype: int64
Validation set class distribution: ripeness
4. Level 4 (spoiled)      24
2. Level 2 (ripe)         24
3. Level 3 (over ripe)    24
1. Level 1 (not ripe)     24
Name: count, dtype: int64
Test set class distribution: ripeness
4. Level 4 (spoiled)      24
2. Level 2 (ripe)         24
3. Level 3 (over ripe)    24
1. Level 1 (not ripe)     24
Name: count, dtype: int64
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\1. Level 1 (not ripe)\Image_9.png
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\3. Level 3 (over ripe)\Image_34.png
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\4. Level 4 (spoiled)\Image_47_augmented_1.png
```

```
12/12 [==============================] - 53s 4s/step - loss: 0.5547 - accuracy: 0.8438
Ripeness Model Evaluation Results: [0.5547290444374084, 0.84375]

1/3 [=========>....................] - ETA: 37s
2/3 [===================>..........] - ETA: 13s
3/3 [==============================] - ETA: 0s
3/3 [==============================] - 45s 13s/step
Ripeness Model Accuracy: 0.84
Classification Report (Ripeness):
              precision    recall  f1-score   support

           0       0.95      0.79      0.86        24
           1       0.72      0.96      0.82        24
           2       1.00      0.62      0.77        24
           3       0.83      1.00      0.91        24

    accuracy                           0.84        96
   macro avg       0.87      0.84      0.84        96
weighted avg       0.87      0.84      0.84        96
```

Figure 6.1.10 Banana ripeness model first training

- Original dataset only has total 60 images for each class.
- Dataset was increased after data augmentation, so each class become 120 images so total 480 images.
    - So new dataset split into each class 72 as train set, 24 as valid set and the left 24 as test set.
- Time takes for model training: 7 hours.
- Model evaluation results is 84%.

2. Second time training

```
Training set class distribution: ripeness
4. Level 4 (spoiled)      36
1. Level 1 (not ripe)     36
3. Level 3 (over ripe)    36
2. Level 2 (ripe)         36
Name: count, dtype: int64
Validation set class distribution: ripeness
3. Level 3 (over ripe)    12
4. Level 4 (spoiled)      12
2. Level 2 (ripe)         12
1. Level 1 (not ripe)     12
Name: count, dtype: int64
Test set class distribution: ripeness
3. Level 3 (over ripe)    12
4. Level 4 (spoiled)      12
2. Level 2 (ripe)         12
1. Level 1 (not ripe)     12
Name: count, dtype: int64
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\4. Level 4 (spoiled)\Image_6.png
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\1. Level 1 (not ripe)\Image_11.png
Loading image: C:\FYP\Banana\PreprocessedImages\ripeness\3. Level 3 (over ripe)\Image_2.png
```

```
6/6 [==============================] - 40s 6s/step - loss: 0.4962 - accuracy: 0.8333
Ripeness Model Evaluation Results: [0.4961577355861664, 0.8333333134651184]

1/2 [==============>...............] - ETA: 28s
2/2 [==============================] - ETA: 0s
2/2 [==============================] - 38s 9s/step
Ripeness Model Accuracy: 0.83
Classification Report (Ripeness):
              precision    recall  f1-score   support

           0       1.00      0.75      0.86        12
           1       0.73      0.92      0.81        12
           2       0.89      0.67      0.76        12
           3       0.80      1.00      0.89        12

    accuracy                           0.83        48
   macro avg       0.86      0.83      0.83        48
weighted avg       0.86      0.83      0.83        48
```

Figure 6.1.11 Banana ripeness second training

- Time takes for model training: 1.33 hour.
- After the second training the accuracy drop from 84% to 83%, so decide use the first training model.

3. Test all images from dataset

```
1/8 [==>...........................] - ETA: 3:09 - loss: 0.3026 - accuracy: 0.9375
2/8 [======>.......................] - ETA: 1:40 - loss: 0.3685 - accuracy: 0.8750
3/8 [==========>...................] - ETA: 1:22 - loss: 0.3535 - accuracy: 0.8958
4/8 [==============>...............] - ETA: 1:05 - loss: 0.3850 - accuracy: 0.8750
5/8 [=================>............] - ETA: 50s - loss: 0.3764 - accuracy: 0.8813
6/8 [=====================>........] - ETA: 34s - loss: 0.3584 - accuracy: 0.8906
7/8 [=========================>....] - ETA: 16s - loss: 0.3265 - accuracy: 0.9062
8/8 [==============================] - ETA: 0s - loss: 0.3306 - accuracy: 0.9083
8/8 [==============================] - 137s 16s/step - loss: 0.3306 - accuracy: 0.9083
Ripeness Model Evaluation Results: [0.3306102752685547, 0.9083333611488342]

1/8 [==>...........................] - ETA: 2:43
2/8 [======>.......................] - ETA: 1:48
3/8 [==========>...................] - ETA: 1:28
4/8 [==============>...............] - ETA: 1:09
5/8 [=================>............] - ETA: 51s
6/8 [=====================>........] - ETA: 34s
7/8 [=========================>....] - ETA: 16s
8/8 [==============================] - ETA: 0s
8/8 [==============================] - 134s 16s/step
Ripeness Model Accuracy on all images: 0.91
Classification Report (Ripeness):
                        precision    recall  f1-score   support

 1. Level 1 (not ripe)       0.96      0.87      0.91        60
     2. Level 2 (ripe)       0.84      0.93      0.88        60
3. Level 3 (over ripe)       0.94      0.85      0.89        60
  4. Level 4 (spoiled)       0.91      0.98      0.94        60

              accuracy                           0.91       240
             macro avg       0.91      0.91      0.91       240
          weighted avg       0.91      0.91      0.91       240

Total execution time: 751.05 seconds
```

Figure 6.1.12 Banana ripeness test dataset

- Use the original dataset to test the model performance.
- Time takes for all images: 12.5 minutes
- Model Accuracy: 91%

4. Confusion Matrix for test all images from dataset

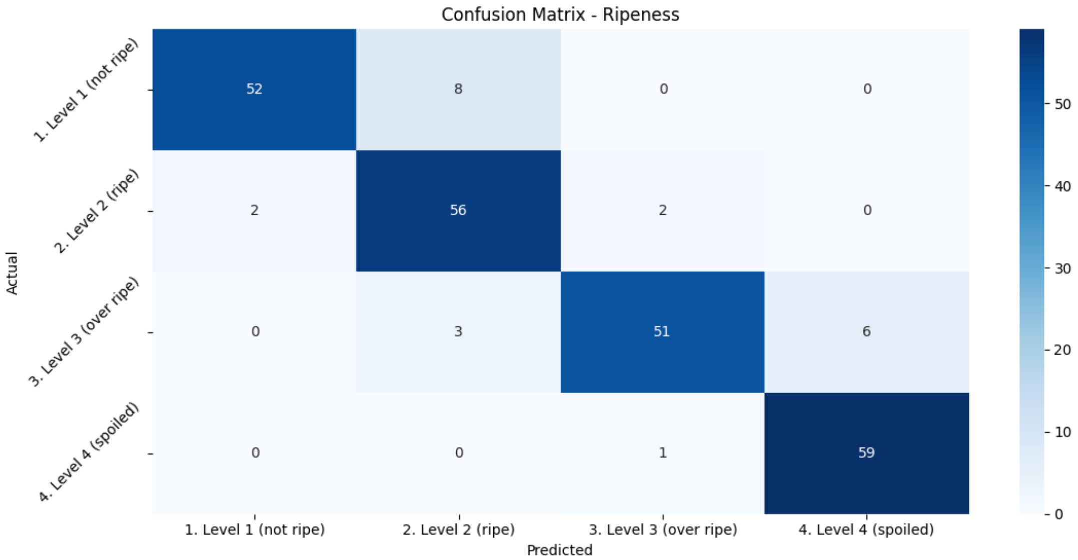


Figure 6.1.13 Banana ripeness confusion matrix

- Total 240 images, wrong predictions have 20 images.
- 8 images from Level 1 predicted as Level 2.
- 2 images from Level 2 predicted as Level 1, and 2 images form Level 2 predicted as Level 3.
- 3 images from Level 3 predicted as Level 2, and 6 images from Level 3 predicted as Level 4.
- Only 1 image from Level 4 predicted as Level 3.
- So in summary, Level 4 have the highest accuracy but Level 3 is the lowest.

5. Wrong predictions

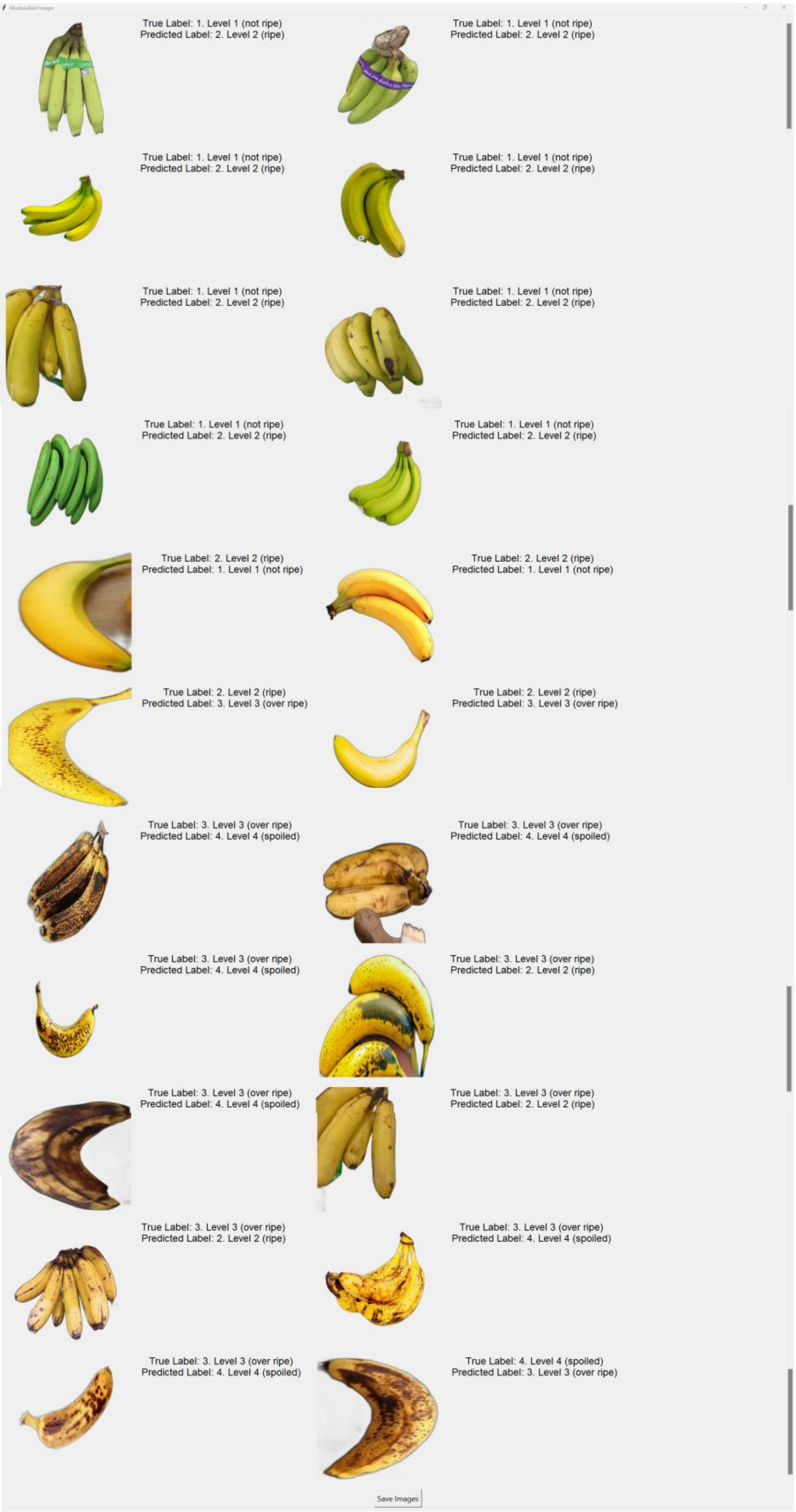


Figure 6.1.14 Model predict different result with actual label

- Above Figure 6.1.14 showing the 22 wrong predictions are which images.

6. Conclusion

The banana ripeness model was first trained on an augmented dataset, increasing the image count per class from 60 to 120, resulting in a total of 480 images. The initial training took 7 hours and achieved an evaluation accuracy of 84%. However, after retraining, the accuracy slightly dropped to 83%, leading to the decision to use the model from the first training. When tested on the original dataset, the model achieved an accuracy of 91% in 12.5 minutes. A confusion matrix revealed 20 incorrect predictions out of 240 images. Most errors occurred in predicting Level 3 ripeness, which had the lowest accuracy, while Level 4 had the highest accuracy. Specifically, Level 1 was often confused with Level 2, and Level 3 was sometimes misclassified as Level 2 or Level 4. In summary, despite a small drop in accuracy after retraining, the first model was selected for its overall better performance. The model demonstrated high accuracy with some areas for improvement, particularly in distinguishing certain ripeness levels.

6.1.4 Mango Quality Model

1. First time training

```
Training set class distribution: quality        You, 3 weeks ago • Mango quality done
1. Good     120
2. Bad      120
Name: count, dtype: int64
Validation set class distribution: quality
1. Good     40
2. Bad      40
Name: count, dtype: int64
Test set class distribution: quality
1. Good     40
2. Bad      40
Name: count, dtype: int64
Loading image: C:\FYP\Mango\PreprocessedImages\quality\1. Good\image 40_augmented_1.png
Loading image: C:\FYP\Mango\PreprocessedImages\quality\1. Good\image 54_augmented_1.png
Loading image: C:\FYP\Mango\PreprocessedImages\quality\1. Good\image 26_augmented_2.png
```

```
10/10 [==============================] - 50s 4s/step - loss: 0.0680 - accuracy: 0.9750
Quality Model Evaluation Results: [0.06801983714103699, 0.9750000238418579]

1/3 [=========>....................] - ETA: 38s
2/3 [===================>..........] - ETA: 13s
3/3 [==============================] - ETA: 0s
3/3 [==============================] - 40s 10s/step
Quality Model Accuracy: 0.97
Classification Report (Quality):
              precision    recall  f1-score   support

           0       0.95      1.00      0.98        40
           1       1.00      0.95      0.97        40

    accuracy                           0.97        80
   macro avg       0.98      0.97      0.97        80
weighted avg       0.98      0.97      0.97        80
```

Figure 6.1.15 Mango quality model first training

- Original dataset only has total 100 images for Good and Bad quality.
- Dataset was increased after data augmentation, so each class become 200 images so total 400 images.
    - So new dataset split into each class 120 as train set, 40 as valid set and the left 40 as test set.
- Time takes for model training: 5 hours.
- Model evaluation results is 97%.

2. Test all images from dataset

```
1/4 [======>.......................] - ETA: 1:15
2/4 [=============>................] - ETA: 24s
3/4 [====================>.........] - ETA: 13s
4/4 [==============================] - ETA: 0s
4/4 [==============================] - 55s 10s/step
Quality Model Accuracy on all images: 0.95
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96        60
           1       0.97      0.90      0.94        40

    accuracy                           0.95       100
   macro avg       0.95      0.94      0.95       100
weighted avg       0.95      0.95      0.95       100

Total execution time: 297.20 seconds
```

Figure 6.1.16 Mango quality test dataset

- Use the original dataset to test the model performance.
- Time takes for all images: 5 minutes
- Model Accuracy: 95%

3. Confusion Matrix for test all images from dataset

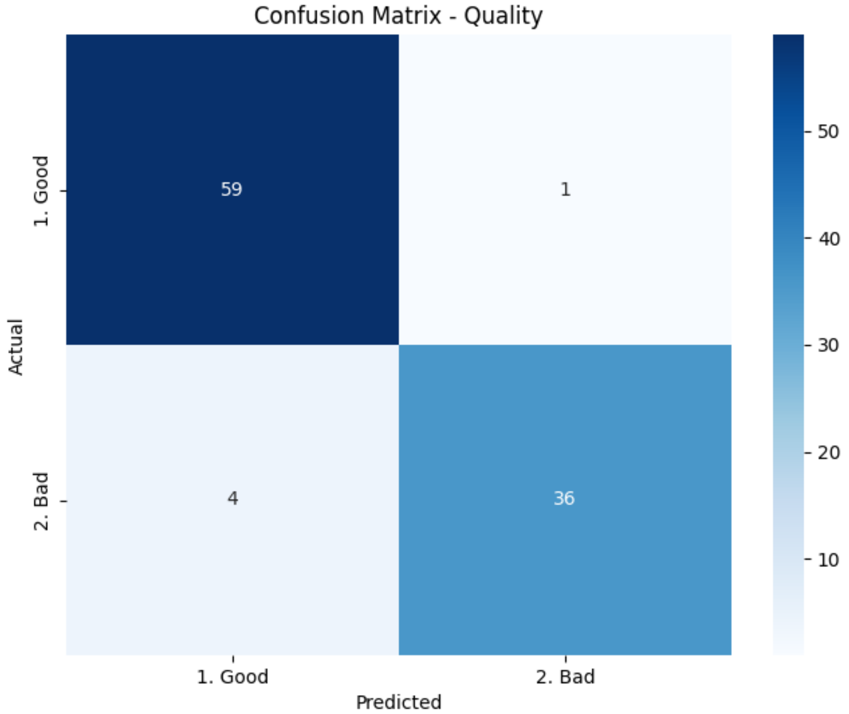


Figure 6.1.17 Mango quality confusion matrix

- In 100 images have 5 images are predicted wrong.
- 1 image from Good quality predicted as Bad quality.

- 4 images form Bad quality predicted as Good quality.

4. Wrong predictions

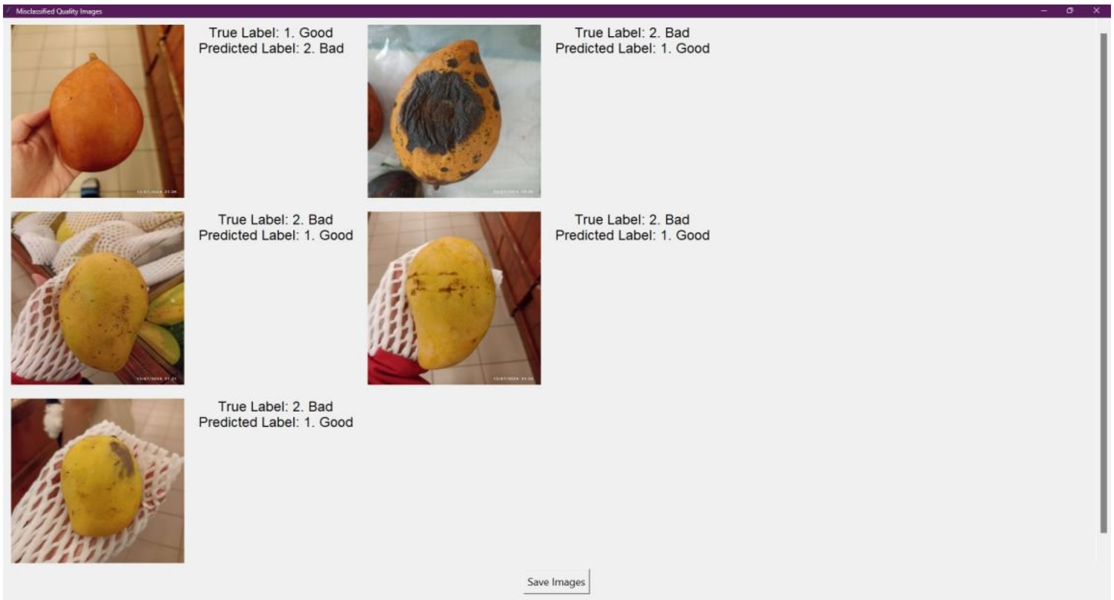


Figure 6.1.18 Model predict different result with actual label

- Above Figure 6.1.18 showing the 5 wrong predictions are which images.

5. Conclusion

The mango quality model was initially trained on an augmented dataset, increasing the number of images per class from 100 to 200, resulting in a total of 400 images. The first training session took 5 hours and achieved a strong evaluation accuracy of 97%. When tested on the original dataset, the model demonstrated an accuracy of 95% within 5 minutes. The confusion matrix showed that out of 100 test images, 5 predictions were incorrect. Specifically, 1 Good quality image was predicted as Bad, while 4 Bad quality images were predicted as Good. In conclusion, the mango quality model performed well, with high overall accuracy. However, there is a slight bias toward misclassifying Bad quality images as Good, indicating an area for improvement.

6.1.5 Mango Ripeness Model

1. First time training

```
Training set class distribution: ripeness
1. Level 1 (not ripe)     72
3. Level 3 (over ripe)    72
4. Level 4 (spoiled)      72
2. Level 2 (ripe)         72
Name: count, dtype: int64
Validation set class distribution: ripeness
4. Level 4 (spoiled)      24
2. Level 2 (ripe)         24
3. Level 3 (over ripe)    24
1. Level 1 (not ripe)     24
Name: count, dtype: int64
Test set class distribution: ripeness
4. Level 4 (spoiled)      24
2. Level 2 (ripe)         24
3. Level 3 (over ripe)    24
1. Level 1 (not ripe)     24
Name: count, dtype: int64
Loading image: C:\FYP\Mango\PreprocessedImages\ripeness\1. Level 1 (not ripe)\Image_9.png
Loading image: C:\FYP\Mango\PreprocessedImages\ripeness\3. Level 3 (over ripe)\Image_34.png
Loading image: C:\FYP\Mango\PreprocessedImages\ripeness\4. Level 4 (spoiled)\Image_47_augmented_1.png
```

```
12/12 [==============================] - 66s 5s/step - loss: 0.4178 - accuracy: 0.8646
Ripeness Model Evaluation Results: [0.41779765486717224, 0.8645833134651184]

1/3 [=========>....................] - ETA: 51s
2/3 [===================>..........] - ETA: 15s
3/3 [==============================] - ETA: 0s
3/3 [==============================] - 60s 17s/step
Ripeness Model Accuracy: 0.86
Classification Report (Ripeness):
              precision    recall  f1-score   support

           0       0.95      0.83      0.89        24
           1       0.68      0.96      0.79        24
           2       0.94      0.71      0.81        24
           3       1.00      0.96      0.98        24

    accuracy                           0.86        96
   macro avg       0.89      0.86      0.87        96
weighted avg       0.89      0.86      0.87        96
```

Figure 6.1.19 Mango ripeness model first training

- Original dataset only has total 60 images for each class.
- Dataset was increased after data augmentation, so each class become 120 images so total 480 images.
    - So new dataset split into each class 72 as train set, 24 as valid set and the left 24 as test set.
- Time takes for model training: 7 hours.
- Model evaluation results is 86%.

2. Test all images from dataset

```
1/8 [==>...........................] - ETA: 3:29 - loss: 0.4496 - accuracy: 0.8125
2/8 [======>.......................] - ETA: 1:50 - loss: 0.2917 - accuracy: 0.8906
3/8 [==========>...................] - ETA: 1:38 - loss: 0.2320 - accuracy: 0.9062
4/8 [==============>...............] - ETA: 1:29 - loss: 0.2431 - accuracy: 0.9141
5/8 [=================>............] - ETA: 1:10 - loss: 0.2540 - accuracy: 0.9125
6/8 [=====================>........] - ETA: 44s - loss: 0.2807 - accuracy: 0.9062
7/8 [=========================>....] - ETA: 21s - loss: 0.2540 - accuracy: 0.9152
8/8 [==============================] - ETA: 0s - loss: 0.2386 - accuracy: 0.9208
8/8 [==============================] - 167s 20s/step - loss: 0.2386 - accuracy: 0.9208
Ripeness Model Evaluation Results: [0.23860543966293335, 0.9208333492279053]

1/8 [==>...........................] - ETA: 3:02
2/8 [======>.......................] - ETA: 1:43
3/8 [==========>...................] - ETA: 1:32
4/8 [==============>...............] - ETA: 1:16
5/8 [=================>............] - ETA: 1:05
6/8 [=====================>........] - ETA: 45s
7/8 [=========================>....] - ETA: 23s
8/8 [==============================] - ETA: 0s
8/8 [==============================] - 178s 22s/step
Ripeness Model Accuracy on all images: 0.92
Classification Report (Ripeness):
                        precision    recall  f1-score   support

 1. Level 1 (not ripe)       0.98      0.88      0.93        60
     2. Level 2 (ripe)       0.79      0.97      0.87        60
3. Level 3 (over ripe)       0.95      0.87      0.90        60
  4. Level 4 (spoiled)       1.00      0.97      0.98        60

              accuracy                           0.92       240
             macro avg       0.93      0.92      0.92       240
          weighted avg       0.93      0.92      0.92       240

Total execution time: 527.49 seconds
```

Figure 6.1.20 Mango ripeness test dataset

- Use the original dataset to test the model performance.
- Time takes for all images: 9 minutes
- Model Accuracy: 92%

3. Confusion Matrix for test all images from dataset

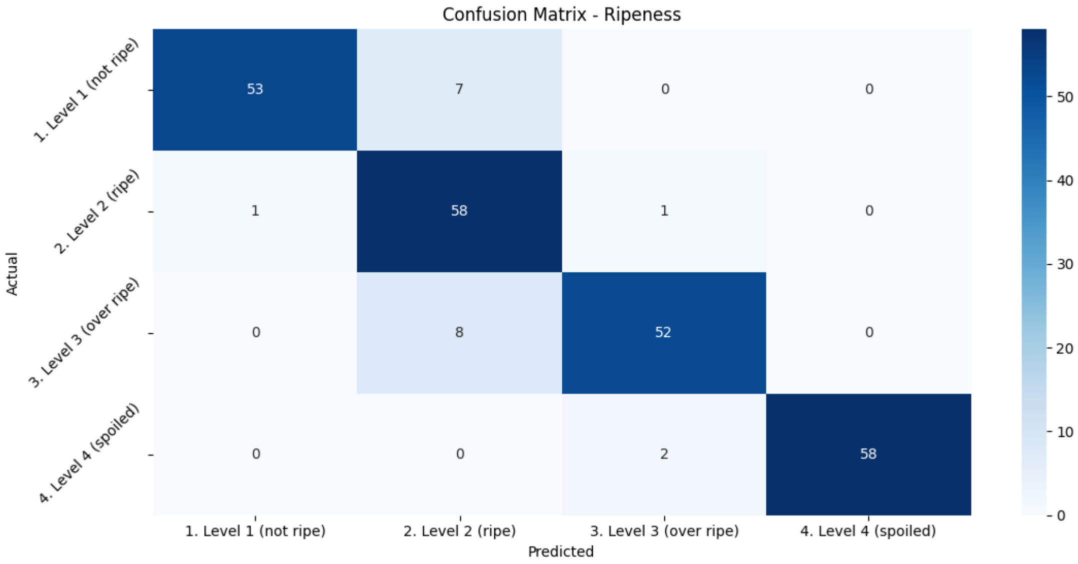


Figure 6.1.21 Mango ripeness confusion matrix

- Total 240 images, wrong predictions have 19 images.
- 7 images from Level 1 predicted as Level 2.
- 1 images from Level 2 predicted as Level 1, and 1 images form Level 2 predicted as Level 3.
- 8 images from Level 3 predicted as Level 2.
- 2 images from Level 4 predicted as Level 3.
- So in summary, Level 2 and Level 4 have the highest accuracy but Level 3 is the lowest.

4. Wrong predictions

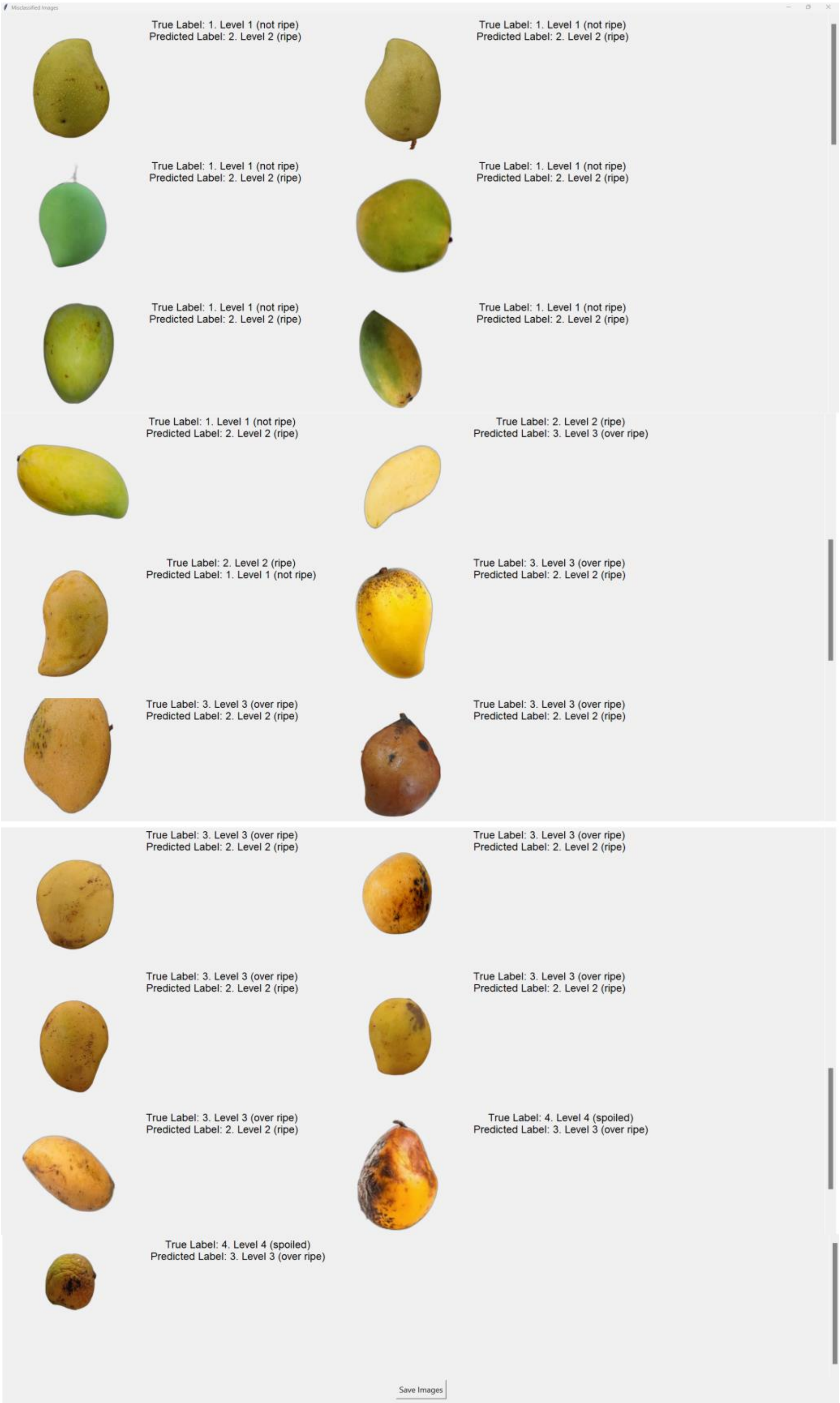


Figure 6.1.22 Model predict different result with actual label

- Above Figure 6.1.22 showing the 19 wrong predictions are which images.

5. Conclusion

The mango ripeness model was trained on an augmented dataset, increasing the number of images per class from 60 to 120, resulting in a total of 480 images. The first training session took 7 hours, with the model achieving an evaluation accuracy of 86%. When tested on the original dataset, the model showed an improved accuracy of 92% in just 9 minutes. The confusion matrix revealed that 19 out of 240 images were predicted incorrectly. Most of the errors occurred with Level 1 images being classified as Level 2, and Level 3 images being misclassified as Level 2 or Level 4. Levels 2 and 4 had the highest accuracy, while Level 3 was the most challenging for the model. In conclusion, the mango ripeness model performed well with high accuracy, though improvements are needed to better differentiate certain ripeness levels, particularly Level 3.

6.2 Testing Setup and Result

6.2.1 Banana

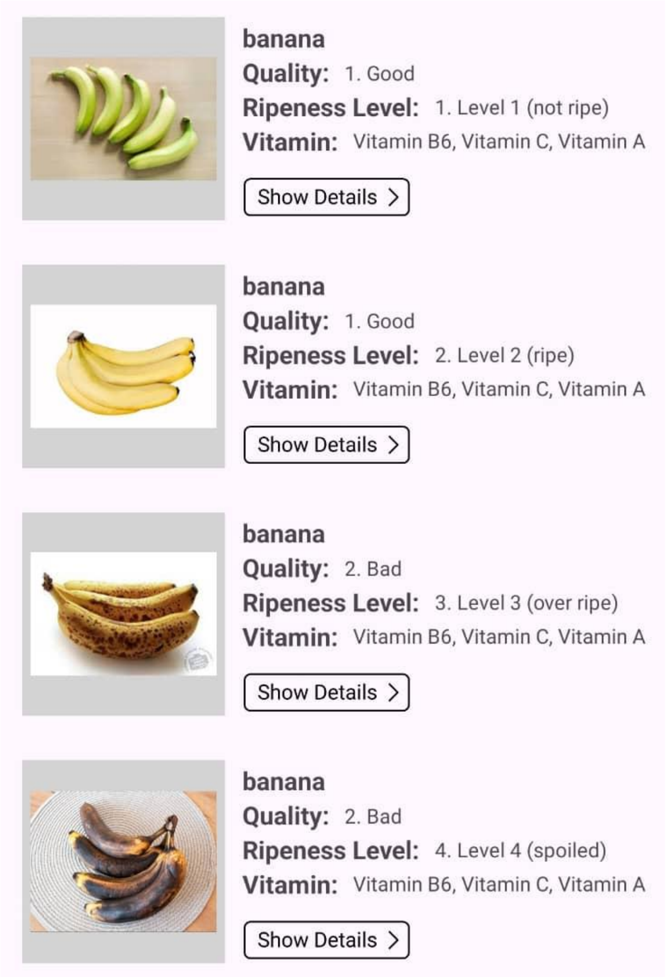


Figure 6.2.1 Banana testing

The above images are new images for models, and the models' predicted results are all correct. By the testing above means that Food Recognition Model is good to recognize the food is banana. This is important because if wrong recognize the food then the prediction for quality and ripeness may be wrong also.

6.2.2 Mango

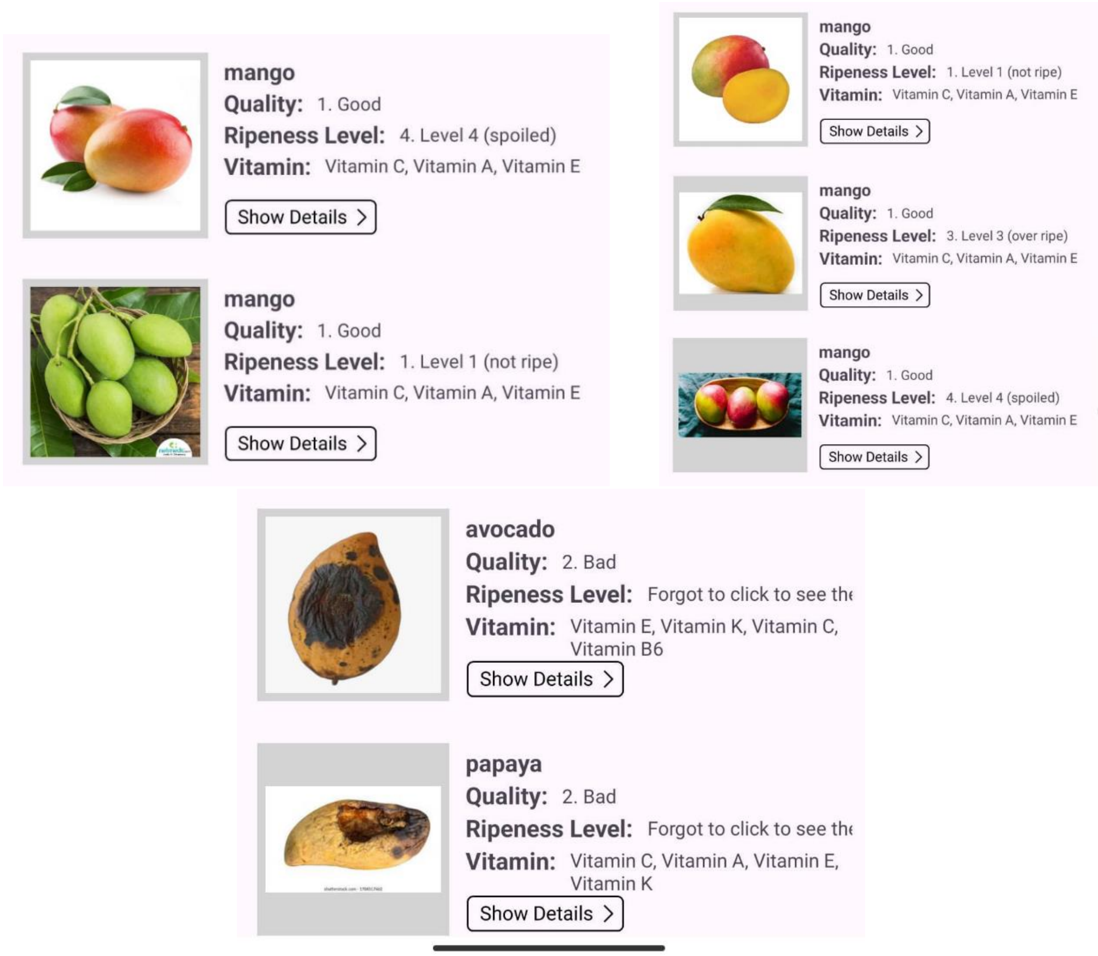
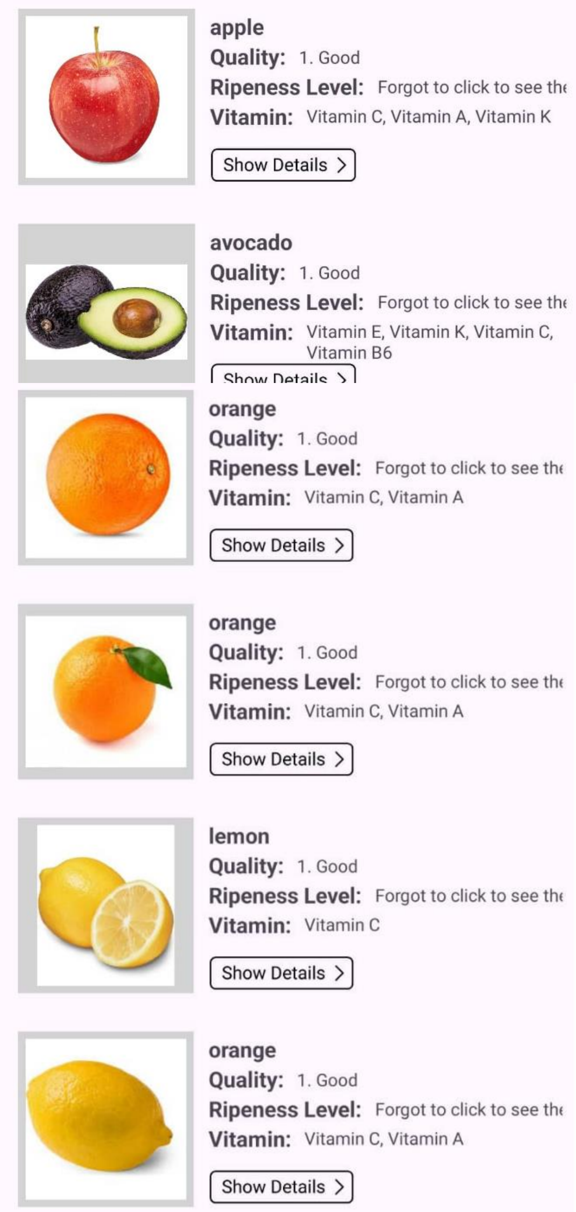


Figure 6.2.2 Mango testing

The above testing result that the Food Recognition Model still not good enough to predict this food is mango. However, the quality results are all correct means that the Mango Quality Model can define Good and Bad quality. Unfortunately, for ripeness result only 100% correct for one result only. The others can guess why but not logic.

6.2.3 Others

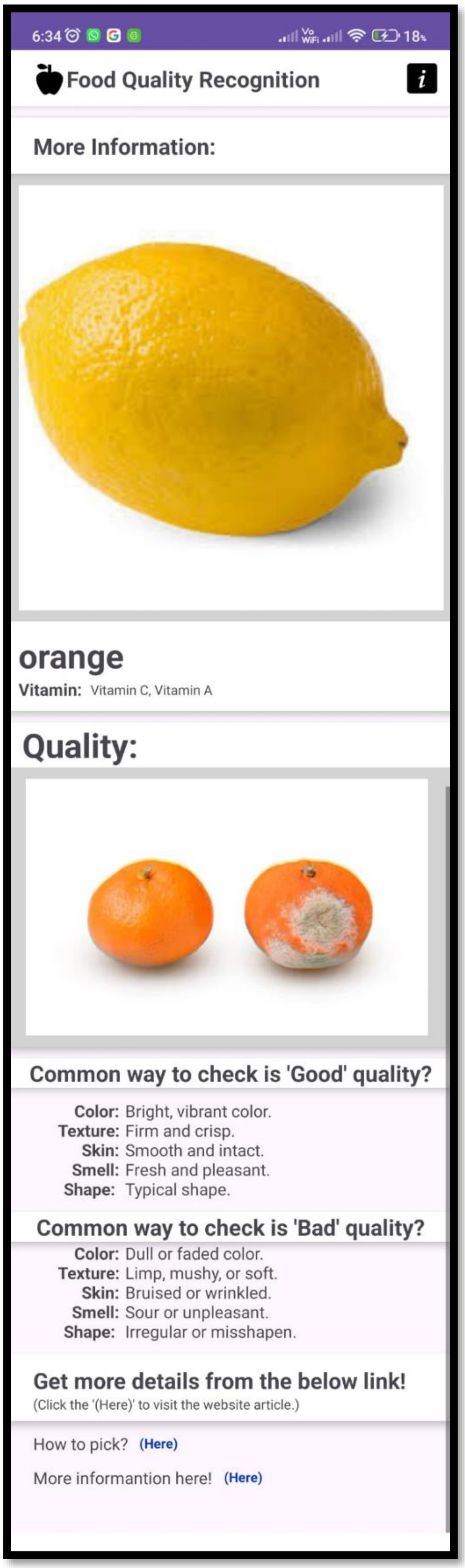


Here is used to test the Food Recognition Model:

- Randomly tested 2 types of food to check the performance, apple and avocado given correct result.
- After that try orange and lemon since above explain that most orange images are predicted as lemon.
- Tired 4 times of different orange and lemon images, really an lemon image predicted as orange.
- This mean that the Food Recognition still need to be improve so that low accuracy getting wrong result.

Figure 6.2.3 Others testing

6.2.4 Example when Food Recognition Model gives wrong predicted result



This image actual food is lemon, but the Food Recognition Model predicted it as orange.

So due to this all the information about orange will be display even the image is lemon actually.

- The vitamin details is for orange.
- The comparison good and bad image displayed for orange
- The links provided also bring user to the article websites that about orange too.

Therefore, Food Recognition Model is the first important model for avoid this kind of issue happen.

Figure 6.2.4 Example wrong information

Conclusion: There still have a big improve for the models.

6.3 Project Challenges

The challenge of this project includes Limited Hardware Resources, Insufficient Storage Capacity, Data Collection and Labeling. Learning Model Development from Scratch, Mobile Application Integration, and Overall Resource and Time Constraints.

The primary challenge in developing this project has been working with limited resources. Model development was conducted on an old laptop, even upgraded to 12GB of RAM (4GB+8GB), which still required almost an entire day to train the models since the process was run on a CPU. During training, the system would become extremely slow, making it difficult to work on other tasks, such as writing reports, finding dataset for other models while waiting for the program to complete. Debugging and testing the models also consumed a significant amount of time.

In addition, the laptop had only 256GB of storage, which proved problematic. On one occasion, after waiting for hours for a program to finish running, the model failed to save due to insufficient disk space. The total space for this project reached 25.787GB, as five different models were developed, each with its own dataset.

Collecting and labeling the datasets was another significant time-consuming task. Unlike some public datasets used by others (such as official skin cancer datasets), food quality datasets were harder to find online. As a result, I manually sourced images by visiting supermarkets, taking pictures of fruits and vegetables, and purchasing items like bananas and mangoes to document their ripeness over time by photographing them daily. This manual collection process, in addition to labeling the images, took a considerable amount of effort.

Being new to model development also posed a challenge. Starting from scratch, I had to learn how to build models while simultaneously developing them. To simplify this process, AutoKeras was used, which helped automate hyperparameter tuning and model selection, reducing the complexity of manually testing configurations.

Finally, after the models were developed, integrating them into a mobile application presented another challenge. I had to research and learn how to effectively apply the models in a mobile environment, which added another layer of complexity to the project.

In conclusion, the most significant challenges have been the limited time, hardware, and financial resources available to complete the project.

6.4 Objective Evaluation

1. **Objective 1: Achieving High Model Accuracy**

The primary goal for the machine learning models is to achieve an accuracy of at least 90% on all images in the dataset. This benchmark ensures that the models are reliable and effective in food quality recognition, providing users with trustworthy results based on the input images.

2. **Objective 2: High Accuracy for Good and Bad Quality Detection**

For the detection of good and bad quality food, it is essential that the model achieves a very high accuracy. Since human eyes can easily distinguish between good and bad quality, the model must be equally effective in identifying food that is visibly spoiled or in poor condition. Achieving high precision in this task is crucial, as even small mistakes could impact the usefulness of the application.

3. **Objective 3: Accurate Classification of Ripeness Levels**

For ripeness detection, the model aims to accurately classify at least three distinct stages: unripe, ripe, and spoiled. Although there are four levels of ripeness defined, the primary focus is on ensuring that the model can effectively differentiate between unripe, ripe, and spoiled stages. These stages are visually distinct and should be easily recognizable by the model, as they exhibit clear differences that can be identified by the human eye. Achieving accurate classification in these key stages is crucial for providing reliable ripeness assessments.

4. **Objective 4: Model Integration into the Mobile Application**

All developed models must be fully integrated into the mobile application. The features and processes applied during model training—such as background removal—should also be implemented within the app to ensure consistency in the results. This seamless integration is vital for maintaining the accuracy and performance achieved during development in the final mobile version.

5. **Objective 5: Adhering to the UI Prototype**

The mobile application’s design should closely follow the UI prototype created during the planning phase. The layout and functionality must be preserved, with improvements made only to enhance the user experience, not to change or remove the core elements of the interface. This consistency ensures that the app delivers a user experience in line with the original design vision.

6. **Objective 6: Offline Functionality and User-Friendly Interface**

The mobile application is expected to function offline, allowing users to access its features without an internet connection. In addition, it must be user-friendly, with a clear and intuitive interface that helps users easily understand its features and how to interact with the app. Meeting this objective ensures that the application is both accessible and convenient for users in various environments.

6.5 Concluding Remark

In conclusion, there are several challenges and objectives formulated in this work to create this mobile application for the food quality recognition. Despite these challenges, much has been achieved working with limited resources, and with all the uncertainties, intricacies that accompany the model training and its integration to the system. The goal of the project is set high accuracy of food quality detection, guarantee successful ripeness classification, and create a mobile application with the intuitive interface that corresponds to the prototype. By achieving these goals, the application will serve the usefulness of the food quality insight and be useful to the users, thus, the successful demonstration of the real-life use of machine learning.

CHAPTER 7
Conclusion and Recommendation

This chapter is the last chapter of this report which is about the conclusion of this project. It is an outline provides a comprehensive overview of the project's conclusion and offers practical recommendations for future enhancements.

7.1 Conclusion

This project focused on developing a mobile application for food quality recognition, utilizing machine learning models to evaluate food based on images.

7.1.1 Project Review and Discussion

1. **Achievement of Objectives:**
   The project successfully met its core objectives. The machine learning models achieved an accuracy of over 90% on the dataset, effectively distinguishing between good and bad quality food. The ripeness classification, while initially set for four levels, focused on accurately defining the key stages of unripe, ripe, and spoiled, ensuring reliable differentiation among these visually distinct stages.

2. **Challenges Overcome:**
   Significant challenges were encountered, including limited hardware resources, insufficient storage capacity, and the need for extensive manual data collection and labeling. Despite these constraints, the project adapted by using AutoKeras for automated hyperparameter tuning and effectively integrating models into the mobile application.

3. **Integration and Usability:**
   The integration of models into the mobile application was a crucial milestone. The application was developed to function offline, ensuring accessibility and user convenience. The UI design, based on the initial prototype, was preserved and refined to enhance user experience while maintaining consistency with the original vision.

4. **Project Novelties:**
   This project is notable for its approach to food quality recognition using machine learning in a mobile context. The ability to perform accurate quality and ripeness detection with limited resources showcases the effectiveness of the chosen methodologies and technologies. Additionally, the focus on offline functionality and user-friendly design adds practical value for end-users.

7.2 Recommendation

To further improve the project and address some of the limitations encountered, the following recommendations are proposed:

1. **Model Accuracy and Stability:**
   Future work should focus on improving the accuracy and stability of the models. Although the current models achieved reasonable accuracy, their performance is not yet fully stable, particularly when predicting new images. Exploring more advanced model architectures, fine-tuning techniques, and additional training data can enhance accuracy. Implementing robust evaluation metrics and incorporating techniques such as cross-validation and ensemble learning could also improve model stability and generalizability.

2. **Hardware and Resource Upgrades:**
   Upgrading hardware resources, including GPUs, would expedite model training and improve overall efficiency. Increasing storage capacity would also address issues related to disk space, allowing for more effective data management and model handling.

3. **Data Collection Enhancement:**
   Expanding and diversifying the dataset is crucial for improving model performance. Collecting more images from varied sources and including additional types of food or ripeness stages can enhance model robustness. Collaborating with organizations for access to larger datasets or leveraging public datasets could also be beneficial.

4. **User Interface and Experience:**
   While the current UI meets the project's goals, future iterations could benefit from user feedback to further refine the interface. Conducting usability testing and incorporating user suggestions can improve the overall user experience and satisfaction.

5. **Mobile Application Features:**
   Future versions of the application could include additional features, such as real-time quality assessment or integration with external databases for more comprehensive information. Exploring cloud-based solutions for data synchronization and updates could enhance the application's capabilities and user experience.

6. **Scalability and Maintenance:**

   Planning for scalability and long-term maintenance will ensure the application's continued relevance and performance. Developing a roadmap for future updates, addressing emerging challenges, and incorporating new technologies will contribute to the application's ongoing success.

In conclusion, while the project has achieved its primary objectives and demonstrated significant progress, addressing these recommendations can lead to further advancements and improvements. Enhancing model accuracy and stability, along with refining the application's features and user experience, will pave the way for a more effective and impactful food quality recognition tool.

References

[1]	K. Y. Koay, "A model of online food delivery service quality, customer satisfaction and customer loyalty: a combination of PLS-SEM and NCA approaches," emerald insight, 15 February 2022. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/BFJ-10-2021-1169/full/html. [Accessed 07 September 2023].
[2]	B. Okumus, "Generation Y's dining out behavior," emerald insight, 20 January 2021. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/IHR-07-2020-0023/full/html. [Accessed 07 September 2023].
[3]	L. University, "Warning over poor cooking skills among teenagers," Science News, 17 June 2015. [Online]. Available: https://www.sciencedaily.com/releases/2015/06/150617104206.htm. [Accessed 07 September 2023].
[4]	H. Yousef, "Intelligent food packaging: a review of smart sensing technologies for monitoring food quality," ACS Publications, 13 March 2019. [Online]. Available: https://pubs.acs.org/doi/full/10.1021/acssensors.9b00440?casa_token=d3OR_Be3t-MAAAAA%3AOc1Nf0r0Yba7_vJ2b0AsvcPCxtUR1Ye_Lm54KfbKG0A97P3nEy93Q7w83sUg8JBLO3_NJGykiIbIWZlUzA. [Accessed 07 September 2023].
[5]	Synoracki, "6 Reasons Why We Waste So Much Food (and How We Can Stop)," Conservation Law Foundation, 19 May 2021. [Online]. Available: https://www.clf.org/blog/why-we-waste-food/. [Accessed 07 September 2023].
[6]	J. Han, "Using Machine Learning Approaches for Food Quality Detection," Hindawai, 31 December 2022. [Online]. Available: https://www.hindawi.com/journals/mpe/2022/6852022/. [Accessed 07 September 2023].
[7]	Gorlaparna1520, "Convolutional Neural Network (CNN) in Machine Learning," GeeksForGeeks, 03 February 2023. [Online]. Available: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/. [Accessed 07 September 2023].
[8]	K. He, "Deep Residual Learning for Image Recognition," IEE Xplore, 30 June 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7780459. [Accessed 07 September 2023].

[9]	"RESNET: Residual Network - JavatPoint," JavatPoint, [Online]. Available: https://www.javatpoint.com/resnet-residual-network#:~:text=However%2C%20ResNet%20has%20limitations%2C%20such,these%20drawbacks%20should%20be%20considered.. [Accessed 07 September 2023].
[10]	G. Huang, "Densely Connected Convolutional Networks," IEE Xplore, 26 July 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8099726. [Accessed 07 September 2023].
[11]	T. Zhou, "Dense Convolutional Network and Its Application in Medical Image Analysis," Hindawi, 22 April 2022. [Online]. Available: https://www.hindawi.com/journals/bmri/2022/2384830/. [Accessed 07 September 2023].
[12]	G. Huang, "Densely Connected Convolutional Networks," Cornell University, 28 January 2018. [Online]. Available: https://arxiv.org/abs/1608.06993v5. [Accessed 07 September 2023].
[13]	Jalal, "Application of Image Processing in Fruit and Vegetable Analysis: A Review," Degruyter, 14 November 2014. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/jisys-2014-0079/html?lang=en. [Accessed 07 September 2023].
[14]	M. Mukhiddinov, "Improved Classification Approach for Fruits and Vegetables Freshness Based on Deep Learning," MDPI, 23 October 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8192. [Accessed 07 September 2023].
[15]	X. Liu, "The recognition of apple fruits in plastic bags based on block classification," SpringerLink, 05 December 2017. [Online]. Available: https://link.springer.com/article/10.1007/s11119-017-9553-2. [Accessed 07 September 2023].
[16]	P. Dhiman, "Smart Disease Detection System for Citrus Fruits Using Deep Learning with Edge Computing," MDPI, 03 March 023. [Online]. Available: https://www.mdpi.com/2071-1050/15/5/4576. [Accessed 07 September 2023].
[17]	N. O. M. Salim, "Study for Food Recognition System Using Deep," IOPsscience, 07 July 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1963/1/012014/meta. [Accessed 07 September 2023].
[18]	G. Ciocca, "Food Recognition and Leftover Estimation for Daily Diet Monitoring," SpringerLink, 01 January 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-23222-5_41. [Accessed 07 September 2023].

[19]	G. Tahir, "A Comprehensive Survey of Image-Based Food Recognition and Volume Estimation Methods for Dietary Assessment," Crnell University, 21 June 2021. [Online]. Available: https://arxiv.org/abs/2106.11776. [Accessed 07 September 2023].
[20]	Z. Lv, "Real-time image processing for augmented reality on mobile devices," SpringerLink, 23 March 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11554-021-01097-9. [Accessed 07 September 2023].
[21]	T. Shen, "The analysis of intelligent real-time image recognition technology based on mobile edge computing and deep learning," SpringerLink, 22 October 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11554-020-01039-x. [Accessed 07 September 2023].
[22]	C. T. Chik, "Food Handlers' Food Safety Knowledge, Attitudes, and Practices in Taman Negara, Kuala Tahan," UiTM, 01 April 2023. [Online]. Available: https://myjurnal.mohe.gov.my/filebank/published_article/140496/Article_4.pdf. [Accessed 16 April 2024].
[23]	N. '. Makhilan, "Development of Safety Assessment System fod Food Premises," Universiti Malaysia Pahang, 17 May 2021. [Online]. Available: https://myjurnal.mohe.gov.my/filebank/published_article/114786/Development_of_safety_assessment_system_for_food_premises.pdf. [Accessed 16 April 2024].
[24]	T. Chen, "Smart Supervision for Food Safety in Food Service Establishments in China: Challenges and Solutions," ScienceDirect, 6 June 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0362028X22055399?ref=pdf_download&fr=RR-2&rr=86988f4f9f6b7132. [Accessed 16 April 2024].
[25]	S. Bobde, "Fruit Quality Recognition using Deep Learning Algorithm," IEEE Xplore, 30 October 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9645793. [Accessed 16 April 2024].
[26]	A. d. A. Gomes, "Pattern recognition techniques in food quality and authenticity: A guide on how to process multivariate data in food analysis," ScienceDirect, 01 July 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165993623001929. [Accessed 16 April 2024].
[27]	B. S. B. Dewantara, "Recognition of Food Material and Measurement of Quality using YOLO and WLD-SVM," IEEE Xplore, 08 November 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9593949. [Accessed 16 April 2024].
[28]	J. Zeng, "A Review of the Discriminant Analysis Methods for Food Quality Based on Near-Infrared Spectroscopy and Pattern Recognition," MDPI, 01 February 2021. [Online]. Available: https://www.mdpi.com/1420-3049/26/3/749. [Accessed 16 April 2024].

[29]	Danielgatis, "GitHub - danielgatis/rembg: Rembg is a tool to remove images background," GitHub, [Online]. Available: https://github.com/danielgatis/rembg. [Accessed 15 August 2024].
[30]	F. C. Q. S. a. X. H. Keras-Team (Haifeng Jin, "AutoKeras: An AutoML Library for Deep Learning," AutoKeras, Juanuary 2019. [Online]. Available: https://autokeras.com/. [Accessed 15 August 2024].
[31]	H. L. L. Z. J. Z. Z. Z. Z. &. J. Y. Feng, "AutoKeras: An AutoML system based on deep learning," 2023. [Online]. Available: https://www.jmlr.org/papers/volume24/20-1355/20-1355.pdf. [Accessed 11 September 2024].
[32]	L. Holzbauer, "A Quick Introduction to Vanilla Neural Networks," 19 December 2019. [Online]. Available: https://blog.insightdatascience.com/a-quick-introduction-to-vanilla-neural-networks-b0998c6216a1. [Accessed 11 September 2024].
[33]	P. Potrimba, "What is ResNet-50?," 13 March 2024. [Online]. Available: https://blog.roboflow.com/what-is-resnet-50/#:~:text=One%20such%20architecture%20is%20called,it%2C%20and%20categorize%20them%20accordingly.. [Accessed 11 September 2024].
[34]	"EfficientNet B0 to B7," Keras, 2023. [Online]. Available: https://keras.io/api/applications/efficientnet/. [Accessed 11 September 2024].
[35]	Y. Fu, "Image classification via fine-tuning with EfficientNet," Keras, 30 June 2020. [Online]. Available: https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/. [Accessed 11 September 2024].
[36]	K. Seth, "Fruits and Vegetables Image Recognition Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition. [Accessed 16 April 2024].
[37]	F. Ripening, "Fruit Ripening Process Computer Vision Project," Roboflow, 2022. [Online]. Available: https://universe.roboflow.com/fruit-ripening/fruit-ripening-process. [Accessed 11 September 2024].
[38]	MUTI, "Ripeness Detection of Mango," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/mutiurrehman80/ripeness-detection-of-mango. [Accessed 11 September 2024].

[39]	project, "project Computer Vision Project," Roboflow, January 2024. [Online]. Available: https://universe.roboflow.com/project-khw09/project-airhk/dataset/1. [Accessed 11 September 2024].
[40]	M. Yolov8, "Mango detection system yolov8 Computer Vision Project," Roboflow, March 2024. [Online]. Available: https://universe.roboflow.com/mango-yolov8/mango-detection-system-yolov8. [Accessed 11 September 2024].
[41]	"Hand Holding Apple Images – Browse 91,913 Stock Photos," Adobe Stock, [Online]. Available: https://t3.ftcdn.net/jpg/05/05/02/82/360_F_505028202_xVkzVlSvhALhjuz026BFr61tbnHlyofC.jpg. [Accessed 16 April 2024].
[42]	the-alchemist-07, "Android-Image-Cropper," GitHub, 2016. [Online]. Available: https://github.com/CanHub/Android-Image-Cropper. [Accessed 09 09 2024].
[43]	M. Salama, "Help Me Choose My Fruit & Veg!," [Online]. Available: https://helpmechoosemyfruitandveg.com/. [Accessed 11 September 2024].
[44]	O. G. Cigliano, "How to Tell if an Avocado Is Ripe," wikiHow, 04 February 2024. [Online]. Available: https://www.wikihow.com/Tell-if-an-Avocado-Is-Ripe. [Accessed 11 September 2024].
[45]	O. G. Cigliano, "How to Tell if an Avocado is Bad," wikiHow, 08 August 2024. [Online]. Available: https://www.wikihow.com/Tell-if-an-Avocado-is-Bad. [Accessed 11 September 2024].
[46]	M. Salama, "Banana," [Online]. Available: https://helpmechoosemyfruitandveg.com/banana.html. [Accessed 11 September 2024].
[47]	syma, "How does Figma help make mobile app development easier?," Figma, 1 Jan 2023. [Online]. Available: https://forum.figma.com/t/how-does-figma-help-make-mobile-app-development-easier/35901. [Accessed 16 April 2024].
[48]	vscode, "Why did we build Visual Studio Code?," Visual Studio Code, 4 April 2024. [Online]. Available: https://code.visualstudio.com/docs/editor/whyvscode. [Accessed 24 April 2024].
[49]	Google, "Meet Android Studio," Google, [Online]. Available: https://developer.android.com/studio/intro#:~:text=Android%20Studio%20provides%20performance%20profilers,performance%2C%20and%20analyze%20network%20requests.. [Accessed 16 April 2024].

[50]	M. Salama, "Avocado," [Online]. Available: https://helpmechoosemyfruitandveg.com/avocado.html. [Accessed 11 September 2024].
[51]	Saduf2019, "How to reduce the size of the tflite model below 200mb while converting h5 to tflite," GitHub, 03 September 2020. [Online]. Available: https://github.com/tensorflow/tensorflow/issues/42898. [Accessed 11 September 2024].
[52]	M. Smith, "chaquopy," GitHub, 25 December 2023. [Online]. Available: https://github.com/chaquo/chaquopy. [Accessed 11 September 2024].
[53]	LaptopML, "PYTHON Java Together: Setup WORKING Android | 2024 Chaquopy," YouTube, 22 December 2023. [Online]. Available: https://www.youtube.com/watch?v=igJgXeV82b4. [Accessed 11 September 2024].

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 2**
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE

- First meet with supervisor and discuss following work for Project II.
- Bought banana and start record the quality and ripeness of banana until spoiled.

2. WORK TO BE DONE

- Data collection
- Prepare dataset

3. PROBLEMS ENCOUNTERED

- No problems, everything is in the progress smoothly.

4. SELF EVALUATION OF THE PROGRESS

- Visit around supermarket to decide a food type to focus on.

Saw Seow Hui

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 4
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE
- Done created dataset with labeling for banana.
- Bought mango and start record the quality and ripeness of banana until spoiled.

2. WORK TO BE DONE
- Continue data collect, labeling and preparing dataset for mango and food recognition dataset.
- Start develop the models and models training.

3. PROBLEMS ENCOUNTERED
- Confusion how to start build models.

4. SELF EVALUATION OF THE PROGRESS
- Research and learning how to build models, start developing the models.

Saw Seow Hui

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 6**
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE

- Finish data collection, labeling and dataset is prepared ready.
- Banana quality models is done develop.

2. WORK TO BE DONE

- Continue develop models.
- Continue models training.

3. PROBLEMS ENCOUNTERED

- Model training takes times.
- Spend whole day waiting the training program run finished.
- When the program running, nothing can do like write report, research and so on.

4. SELF EVALUATION OF THE PROGRESS

- Fully completed at least one model.

Saw Seow Hui

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 8**
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE

- Banana ripeness model done develop.

2. WORK TO BE DONE

- Continue develop the models.
- Continue model training.

3. PROBLEMS ENCOUNTERED

- Not enough sleep because need to keep tracking on the model training when code is running.

4. SELF EVALUATION OF THE PROGRESS

- Even the progress is slower, but still can manage to complete the planning of the week.

Saw Seow Hui

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 10**
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE

- Banana and mango quality and ripeness models is done.

2. WORK TO BE DONE

- Continue develop the last model.
- Continue the last model training.
- Start develop mobile application and configure the developed models into the mobile application.

3. PROBLEMS ENCOUNTERED

- Not enough time to tried out more for increase the accuracy of the models due to submission is coming.

4. SELF EVALUATION OF THE PROGRESS

- All models' development and training can be done on planned time.

Saw Seow Hui

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 12**
Student Name & ID: Connie Tang Ming Xin, 21ACB06403	
Supervisor: Ts Dr Saw Seow Hui	
Project Title: A Mobile Application for Food Quality Recognition	

1. WORK DONE

- Model development and training is done.
- Successfully apply into mobile application.
- Mobile application development done.

2. WORK TO BE DONE

- Project II report.
- Prepare for presentation in the following week 13 or week 14, including PPT slide and demonstration of the prototype.

3. PROBLEMS ENCOUNTERED

- No problems, everything is in the progress smoothly.

4. SELF EVALUATION OF THE PROGRESS

- Fully completed the task planning of the week.

Saw Seow Hui

Supervisor's signature

Student's signature

APPENDIX

POSTER

PLAGIARISM CHECK RESULT

Connie_FYP2_Plagiarism checking.pdf

ORIGINALITY REPORT

8%	%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Noor Adnin binti Ab Aziz, Suzaily Wahab, Rosnah binti Sutan, Muhammad Adib Baharom, Amirul Danial Azmi, Siti Azirah binti Asmai. "Correlation between religiosity and family functioning among secondary school students in high-risk residing areas and factors associated with substance use", Cold Spring Harbor Laboratory, 2024 Publication	3%
2	Shiv Ram Dubey, Anand Singh Jalal. "Application of Image Processing in Fruit and Vegetable Analysis: A Review", Journal of Intelligent Systems, 2015 Publication	1%
3	Nareen O. M. Salim, Subhi R.M. Zeebaree, Mohammed A. M. Sadeeq, A. H. Radie, Hanan M. Shukur, Zryan Najat Rashid. "Study for Food Recognition System Using Deep Learning", Journal of Physics: Conference Series, 2021 Publication	1%

4	Junming Han, Tong Li, Yun He, Quan Gao. "Using Machine Learning Approaches for Food Quality Detection", Mathematical Problems in Engineering, 2022 Publication	<1 %
5	Leonard Garrison. "Notes", Oxford University Press (OUP), 2024 Publication	<1 %
6	Adriano de Araújo Gomes, Silvana M. Azcarate, Ivan Špánik, Liudmyla Khvalbota, Héctor C. Goicoechea. "Pattern recognition techniques in food quality and authenticity: A guide on how to process multivariate data in food analysis", TrAC Trends in Analytical Chemistry, 2023 Publication	<1 %
7	TAIBO CHEN, KAIFANG DING, ZIYIN YU, GENDAO LI, YI DONG. "Smart Supervision for Food Safety in Food Service Establishments in China: Challenges and Solutions", Journal of Food Protection, 2021 Publication	<1 %
8	"Internet of Things. Advances in Information and Communication Technology", Springer Science and Business Media LLC, 2024 Publication	<1 %

9	Bima Sena Bayu Dewantara, Azifah Zusrina Devy, Mochamad Mobed Bachtiar, Setiawardhana. "Recognition of Food Material and Measurement of Quality using YOLO and WLD-SVM", 2021 International Electronics Symposium (IES), 2021 Publication	<1 %
10	K. Makanyadevi, Praveena S, Sudharsan R, Swetha S. "Survey on Customized Diet Assisted System based on Food Recognition", 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), 2023 Publication	<1 %
11	Swati Sood, Kecheng Li, Claire Sand, Lokendra Pal, Martin A. Hubbe. "Fruit wastes: a source of value-added products", Elsevier BV, 2024 Publication	<1 %
12	Mustafa Mhamed, Muhammad Hilal Kabir, Zhao Zhang. "Chapter 1 Developments of the Automated Equipment of Apple in the Orchard: A Comprehensive Review", Springer Science and Business Media LLC, 2023 Publication	<1 %
13	Sandeep Reddy Vermareddy, Teerapat Sanguankotvhakorn. "MODIFICATION OF LEACH PROTOCOL FOR IMPROVING THE STABILITY PERIOD AND LIFETIME OF	<1 %

WIRELESS SENSOR NETWORKS", IndiaRxiv, 2019
Publication

14 Yifan Huang, Taiga Mori, Udara E. Manawadu, Mitsuhiro Kamezaki et al. "Machine Learning Based Skill-Level Classification for Personal Mobility Devices Using Only Operational Characteristics", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018 <1%
Publication

15 Mudduluru Charan Teja, Gaddam Sai Jayachandra, Rudresh Deepak Shirwaikar, B. Shashank Adiga, Chirag Ganesh Padubidri. "Chapter 16 Automated Fish Classification Using Convolutional Neural Networks", Springer Science and Business Media LLC, 2023 <1%
Publication

16 Arthur Lombardi Campos. "Design of a low-power 10-bit 12-MS/s asynchronous SAR ADC", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2020 <1%
Publication

17 Maha Elhini, Hoda Hassaballa, Nicholas P. Simpson, Maha Balbaa et al. "The land degradation and desertification- <1%

	socioeconomic nexus in Egypt's delta region: A case study on Alexandria and Beheira", Heliyon, 2024 Publication	
18	Ton Duc Thang University Publication	<1 %
19	Khairul Amin Bin Mohamad Sukri. "Android-based Home Security Application", Research Square Platform LLC, 2023 Publication	<1 %
20	Ibraheem Ghali-Mohammed, Tajudeen Opeyemi Isola, Ismail Adewuyi Adeyemo, Rafiu Adebisi Kadir et al. "Food safety knowledge and attitudes among fish vendors in informal markets in Ilorin, Nigeria: A cross-sectional study", Springer Science and Business Media LLC, 2024 Publication	<1 %
21	Tao Zhou, XinYu Ye, HuiLing Lu, Xiaomin Zheng, Shi Qiu, YunCan Liu. "Dense Convolutional Network and Its Application in Medical Image Analysis", BioMed Research International, 2022 Publication	<1 %
22	Dubey, Shiv Ram, and Anand Singh Jalal. "Species and variety detection of fruits and vegetables from images", International Journal of Applied Pattern Recognition, 2013.	<1 %

Publication

23	Sushant Kaushal, Dushyanth Kumar Tammineni, Priya Rana, Minaxi Sharma, Kandi Sridhar, Ho-Hsien Chen. "Computer vision and deep learning-based approaches for detection of food nutrients/nutrition: New insights and advances", Trends in Food Science & Technology, 2024 Publication	<1 %
24	Cecilia Fissore, Francesco Floris, Marina Marchisio, Sergio Rabellino. "Learning analytics to monitor and predict student learning processes in problem solving activities during an online training", 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), 2023 Publication	<1 %
25	Lixiong Gong, Xiao Huang, Yinkang Chao, Jialin Chen, Binwen Lei. "An enhanced SSD with feature cross-reinforcement for small-object detection", Applied Intelligence, 2023 Publication	<1 %
26	Theodore Simos, George Maroulis. "Recent Progress in Computational Sciences and Engineering", CRC Press, 2019 Publication	<1 %
27	Umer Amin, Muhammad Imran Shahzad, Aamir Shahzad, Mohsin Shahzad, Uzair Khan,	<1 %

Zahid Mahmood. "Automatic Fruits Freshness Classification Using CNN and Transfer Learning", Applied Sciences, 2023
Publication

28	Hanie Yousefi, Hsuan-Ming Su, Sara M. Imani, Kais Alkhaldi, Carlos D. M. Filipe, Tohid F. Didar. "Intelligent Food Packaging: A Review of Smart Sensing Technologies for Monitoring Food Quality", ACS Sensors, 2019 Publication	<1%
29	Nada Ahmad Alqurashi, Anushree Priyadarshini, Amit K. Jaiswal. "Evaluating Food Safety Knowledge and Practices among Foodservice Staff in Al Madinah Hospitals, Saudi Arabia", Safety, 2019 Publication	<1%
30	Shola Usharani, Telugu Puneeth, Karthikeyan Sugavanan, Sivakumar Depuru. "Food Waste Management using IoT & Android Interface", 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), 2023 Publication	<1%
31	Agnieszka Muszynska. "Rotordynamics", CRC Press, 2019 Publication	<1%
32	"Borrelia burgdorferi", Springer Science and Business Media LLC, 2024	<1%

Publication

33	Jean-Philippe Martin-Flatin. "Web-Based Management of IP Networks and Systems", Wiley, 2002 Publication	<1 %
34	Sarika Bobde, Sarthak Jaiswal, Pradnya Kulkarni, Omkar Patil, Pranav Khode, Rishabh Jha. "Fruit Quality Recognition using Deep Learning Algorithm", 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021 Publication	<1 %
35	Soung Yue Liew, Andrew Khin Huat Tan, Elok Robert Tee. "A Time Slotted Model for All-Optical Variable-Length Packet Switching without Packet Alignment", 2008 Second Asia International Conference on Modelling & Simulation (AMS), 2008 Publication	<1 %
36	"Image and Graphics", Springer Science and Business Media LLC, 2017 Publication	<1 %
37	"Innovations in Electrical and Electronic Engineering", Springer Science and Business Media LLC, 2021 Publication	<1 %

38	"Recent Challenges in Intelligent Information and Database Systems", Springer Science and Business Media LLC, 2024 Publication	<1 %
39	Eradah O Hamad. "Psychological Well-Being among Saudi Adults During the Context of COVID-19 Lockdown: A Psychometric Analysis of the 12-Item General Health Questionnaire", Research Square Platform LLC, 2022 Publication	<1 %
40	Erich Kasper, Jinzhong Yu. "Silicon-Based Photonics", Jenny Stanford Publishing, 2020 Publication	<1 %
41	Karla Holmboe. "Frontal Cortex Functioning in Infancy", Thesis Commons, 2017 Publication	<1 %
42	Marcantonio Gagliardi. "Programming The Mind – The Seven Dimensions of Attachment in Personality and Psychopathology", PsyArXiv, 2024 Publication	<1 %
43	Ray Tricker. "ISO 9001:2015 for Small Businesses", Routledge, 2019 Publication	<1 %
44	Boqiang Li, Baiyang Chen, Zongsu Wei. "Challenging established norms: The unanticipated role of alcohols in UV/PDS	<1 %

radical quenching", Journal of Hazardous Materials, 2024
Publication

45	Constantine Stephanidis, Gavriel Salvendy. "Foundations and Fundamentals in Human-Computer Interaction", CRC Press, 2024 Publication	<1 %
46	Hanoi National University of Education Publication	<1 %
47	Janne Hukkinen. "Institutions in Environmental Management - Constructing Mental Models and Sustainability", Routledge, 2019 Publication	<1 %
48	Johan Woltjer, Ernest Alexander, Angela Hull, Matthias Ruth. "Place-Based Evaluation for Integrated Land-Use Management", Routledge, 2016 Publication	<1 %
49	Jun-Hwa Kim, Namho Kim, Chee Sun Won. "Global–local feature learning for fine-grained food classification based on Swin Transformer", Engineering Applications of Artificial Intelligence, 2024 Publication	<1 %
50	Lameck Mbangula Amugongo, Alexander Kriebitz, Auxane Boch, Christoph Lütge.	<1 %

"Mobile Computer Vision-Based Applications for Food Recognition and Volume and Calorific Estimation: A Systematic Review", Healthcare, 2022
Publication

51	Mathisen, Ralph W.. "Ancient Roman Civilization: History and Sources", Oxford University Press Publication	<1 %
52	Mikhail F. Limonov, Richard De La Rue. "Optical Properties of Photonic Structures - Interplay of Order and Disorder", CRC Press, 2019 Publication	<1 %
53	Ofelia Rodríguez Alzueta. "2'-Methoxyacethopenone as DNA Photosensitiser for Mono and Biphotonic Processes", Universitat Politecnica de Valencia, 2020 Publication	<1 %
54	Sebastian Castro-Alvarez. "Chronicle of a Measurement Unforetold", Thesis Commons, 2022 Publication	<1 %
55	Xingwei Duan, Yuhao Lin, Lixia Li, Fujie Zhang, Shanshan Li, Yuxin Liao. "Hierarchical Detection of Gastrodia elata Based on Improved YOLOX", Agronomy, 2023	<1 %

FYP 2 REPORT MARKING SHEET

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)	Connie Tang Ming Xin
ID Number(s)	21ACB06403
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	A Mobile Application for Food Quality Recognition

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: __8__ **%** **Similarity by source** Internet Sources: __-__ % Publications: __8__ % Student Papers: __-__ %	The percentage meets the requirement.
Number of individual sources listed of more than 3% similarity: __1__	She frequently consults this paper that are directly relevant to his work. Thus, the percentage is high.

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.

Saw Seow Hui

Signature of Supervisor	Signature of Co-Supervisor
Name: Ts Dr Saw Seow Hui	Name: ________
Date: 9/12/2024	Date: ________

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student ID	21ACB06403
Student Name	Connie Tang Ming Xin
Supervisor Name	Ts Dr Saw Seow Hui

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
No this	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)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

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

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 11/09/2024