PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM

Undergraduate Thesis
Submitted to the faculty of the
College of Engineering and Information Technology
Cavite State University
Indang, Cavite

In partial fulfilment
of the requirements for the degree
Bachelor of Science in Computer Science

BELANIZO, MERC JAN RALE P.
ENILO, MARK ROBINSON A.
MERCADO, JOHN LOWELL
MORGA, CARLO A.
August 2024

BIOGRAPHICAL DATA

Merc Jan Rale P. Belanizo was born on January 17, 2002, in Bacoor City, Cavite. He is the second child among four children of Ramilo G. Belanizo and Ellen P. Belanizo. Their family is currently residing at Phase 2, Block 5 Lot 32 St. Peregrine Pamayanan ng Bagong Kabitenyo Pasong Kawayan II in General Trias City, Cavite.

He completed his primary education at Pasong Kawayan II Elementary School where he became one of the student athlete awardees in 2014. He then completed his Junior High at Luis Y Ferrer Jr. West National High School with Honors in 2018. Afterwards, he successfully finished his Senior High School at Tanza National Trade School in 2020.

In college, he took Bachelor of Science in Computer Science at Cavite State University – Main Campus and graduated in 2024.

.

BIOGRAPHICAL DATA

Mark Robinson A. Enilo is the second son of Robinson Enilo and Benedicta Enilo. He was born on March 13, 2002, and currently resides at Phase 1 Section 3 Block 10 Lot 34 Paradahan 1, Tanza Cavite.

He finished his primary education at Pasong Kawayan II Elementary School located at Pamayanang Maliksi, General Trias City, Cavite in 2014. He completed his Junior High at Tanza National Trade School located in Paradahan 1, Tanza, Cavite in 2018 and he finished his senior high school in the same school in 2020.

He later enrolled under the Bachelor of Science in Computer Science program at Cavite State University – Main Campus and earned his degree in 2020.

BIOGRAPHICAL DATA

John Lowell Mercado is the only son of Diego Gatdula and Princess Mercado. He was born on December 10, 2001 and currently resides at 024, Bagtas, Tanza, Cavite.

He finished his primary education at Punta Elementary School located at Punta, Tanza, Cavite in 2014. He then completed his junior high school at Tanza National Trade School located at Paradahan I, Tanza, Cavite in 2018. He continued his senior high school in the same school and finally graduated in 2020.

In college, he took up Bachelor of Science in Computer Science at Cavite State University – Main Campus and graduated in 2020.

BIOGRAPHICAL DATA

Carlo A. Morga was born on June 6, 2002, in Pio Duran, Albay. He is the middle child among the three children of Ernesto Morga Jr. and Emma Morga. Their family currently resides at Block 28 Lot 6 Marigold St., Phase 4, Pamayanan ng Bagong Kabitenyo, Pasong Kawayan II, General Trias City, Cavite.

He finished his primary education at Langgam Elementary School in 2014, located in San Pedro, Laguna. He then completed his junior high school at Luis Y Ferrer Jr. West National High School in 2018 and successfully earned his senior high school diploma at Notre Dame of Trece Martirez in Trece Martires City, Cavite in 2020.

He took his Bachelor of Science in Computer Science program at Cavite State University - Main Campus in 2020.

ACKNOWLEDGMENT

The researchers want to extend their deepest gratitude to everyone who helped make their study successful and to those who made substantial contributions in making the seemingly impossible happen.

Dr. Willie C. Buclatin, dean of the College of Engineering and Information Technology, for allowing them to conduct this research and for helping them at every stage of the process;

Prof. Charlotte B. Carandang, chairperson of the Department of Information Technology and Dr. Edwin R. Arboleda, college research and development coordinator, for their prompt assistance with the thesis forms;

Mr. Ace Amiel E. Malicsi, undergraduate thesis professor, for assisting and helping the researchers since the beginning of the project;

Mr. Marlon R. Pereña, for accompanying and overseeing them during the research process, for his invaluable support, encouragement, and insightful guidance, which were instrumental in shaping the direction of the research;

Mr. Stephen Andrei O. Rocillo, technical critic, for his assistance in enhancing the mobile application, which significantly improved the study's quality and effectiveness.

The researchers would also like to express heartfelt gratitude to their parents for their unwavering financial support throughout this endeavor. Their encouragement and sacrifice have been instrumental in enabling the completion of this project.

The vendors who generously assisted them and cooperated in providing valuable insights to their inquiries. Their willingness to share knowledge and expertise

has been instrumental in advancing their research and understanding of the subject matter. They deeply appreciate their support and contributions, which have significantly enriched their project and enabled them to develop a more comprehensive understanding of the factors at play.

The researchers also thank their friends, colleagues, and classmates for their consistent assistance during challenging times. They are equally thankful to all individuals who lent a helping hand, as their kindness and support were greatly valued.

On top of that, the researchers' unwaveringly supportive parents, for helping them in whatever way they could, are deeply appreciated. Their unending understanding is greatly valued and acknowledged.

And foremost, praise be to the mighty God of heaven and earth, Who has not ceased to give them the intelligence and wisdom that served as instruments for the success of this study, and for all the bounties He has always showered upon them.

MERC JAN RALE P. BELANIZO
MARK ROBINSON A. ENILO
JOHN LOWELL MERCADO
CARLO A. MORGA

ABSTRACT

BELANIZO, MERC JAN RALE P., ENILO, MARK ROBINSON A., MERCADO, JOHN LOWELL, and MORGA, CARLO A. "PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM" Undergraduate Thesis. Bachelor of Science in Computer Science. Cavite State University, Indang, Cavite. August 2024. Adviser: Dr. Marlon R. Pereña.

"The Peel Perfect: Development of Banana Ripeness Detection using Image Processing System", conducted at Cavite State University from October 2023 to June 2024, aimed to create a mobile application capable of using image processing to assess banana ripeness (Unripe, Ripe, Overripe) as its primary objective, while also identifying banana types *(Lakatan, Latundan, Saba, and Senorita)* and quantity as secondary functions. It also provides basic information and recommendations for each banana type and its ripeness.

The researchers employed an iterative model approach to the application, which involved step-by-step improvements to design and functionality until the mobile app was completed. The process included requirement gathering and analysis, where researchers gathered a comprehensive and varied dataset of banana images. In the design phase, various tools were utilized for application development. Implementation focused on training the model using YOLOv8 and developing the application features and requirements. Testing encompassed unit testing, integration testing, system testing, and accuracy testing to ensure the implemented code functions correctly and meets specified requirements. Deployment involved deploying the software as a mobile application. Lastly, maintenance involved identifying and fixing any bugs or errors that arose. This process continues until the final application meets all the requirements and features.

The system was evaluated using ISO/IEC 25010 criteria, which evaluates the quality of software products. As for respondents, 20 random banana vendors, 30

random banana consumers, and 10 Information Technology (IT) Professionals used the following criteria: functional suitability, performance efficiency, usability, reliability and portability. The overall result of the evaluation for non-technical was “Very Good”, with an average mean of 4.05. On the other hand, technical evaluation was also “Excellent”, with an average mean of 4.41, which means that the desired output was met.

The Peel Perfect application aims to revolutionize the traditional way of assessing bananas by providing a more precise, and efficient solution. Using image processing and machine learning, it determines the type and ripeness level of bananas. The Peel Perfect application has been beneficial for both consumers and retailers, simplifying the process of banana assessment through a single photo.

The researchers recommend improving the datasets by adding more pictures that show varying levels of brightness and including a wider variety of banana types with different shapes and sizes. This will make the application more useful for different situations. They also suggest allowing users to upload several banana images and using the app to identify how ripe each banana is and what type it is. This way, users can get information about their bananas quickly and easily.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIX FIGURES

LIST OF APPENDIX TABLES

PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM

Merc Jan Rale P. Belanizo
Mark Robinson A. Enilo
John Lowell Mercado
Carlo A. Morga

An undergraduate thesis submitted to the faculty of the Department of Information Technology, College of Engineering and Information Technology, Cavite State University, Indang, Cavite. In partial fulfilment of the requirements for the degree Bachelor of Science in Computer Science with Contribution No. CEIT-2023-24-CS-027
Prepared under the supervision of Mr. Marlon R. Pereña

INTRODUCTION

Project Context

The Philippine banana industry plays a significant role in the economy as it stands out as one of the primary fruit crops commodities in the country. According to the report made by Food and Agriculture Organization (2021), the Philippines is the 4th largest banana producer in Asia trailing behind India, China and Indonesia. In 2018, 84% of the banana production came from the island of Mindanao, with the Davao region, Northern Mindanao, and SOCCSKSARGEN being the top three regions. Notably, majority of banana varieties grown in the country are cavendish, *saba, lakatan, and latundan* which are also widely consumed by the Filipinos across the country. Bananas are known for its versatility and nutritional value, and it serves as a central figure in every Filipino household, frequently appears as a dessert or a nutritious snack. Thus, a key ingredient in different culinary dishes, and a fundamental part of the local breakfast tradition.

On the other hand, ripeness significantly influences postharvest handling and processing since it determines the consumer's acceptability and overall eating experience. In the case of bananas, optimal harvesting practice involves selecting fully mature fruits that have not yet initiated the ripening process. This approach facilitates efficient postharvest handling and transportation, ensuring product integrity and optimal quality upon reaching the consumer (Adebayo *et al.,* 2016). Classifying the ripeness of banana can pose several challenges due to various factors such as varying in ripening stages, external appearance and individual preference. Although yellow color often signifies maturity of the banana, the presence of yellow does not guarantee the consistent ripening stage among bananas. This identification may lead to confusion and result to selection of banana that is either underripe or overripe. In addition, the external appearance of the banana might not accurately reflect the internal ripeness since external factors must be considered such as bruise or blemishes on the peel of the fruit. Another factor is the environmental condition of the fruit because temperature can influence the ripeness of the fruit; banana ripen faster in warmer environment, making it difficult to predict their ripening stage. Relatively, the retailers and consumers in Cavite stated that the diseases of a banana are commonly found on the leaf or inside of the banana and it is not currently available on the peel which the system will scan and evaluate. Lastly, the traditional method of assessing the ripeness of bananas is insufficient in addressing the demand of modern fruit industry, and consumers and retailers are increasingly adopting advanced technologies in identifying ripeness of this fruit. Therefore, the development and implementation of a method for efficiently determining the ripeness of banana is crucial.

To address these problems, researchers explored the possibility of creating a mobile application that can capture bananas and determine their ripeness. The app would be able to identify whether a banana is unripe, ripe, or overripe, and provide users with information on the benefits of each stage. This technology could have a significant impact on both consumers and retailers, allowing them to make informed

decisions on which bananas to buy and how to use them. The primary goal of this research is to develop a reliable and user-friendly mobile app that can identify the ripeness of bananas and their potential benefits.

Objectives of the Study

The researchers aimed to develop a mobile application that uses a mobile phone camera to capture a banana and display the types and its level of ripeness. The level of ripeness will be identified as unripe, ripe, or overripe and the types of bananas will include *Lakatan, Latundan, Saba, and Señorita.* It will be done with the aid of image recognition in YOLOv8. The output will be used to enhance quality control in the banana food industry.

The system was developed consisting of the following modules, Data Collection, Preprocessing, Training, Evaluation and Deployment. These modules collectively ensure the efficiency of the application for banana ripeness detection.

Specifically, it aimed to:

1. Develop a user-friendly mobile application system that allows users to quickly inspect bananas with their mobile phone camera, allowing them to instantly identify types of bananas.

2. Develop a system capable of identifying the following using image processing:
    a. Determine the bananas in terms of color, size and types *(Lakatan, Latundan, Saba and Señorita).*
    b. Determine the ripeness level of banana (Unripe, Ripe, Overripe).
        i. Display recommended food out of banana depending on the ripeness level.
        ii. Display the basic information of bananas.
    c. Determine the number in a cluster of bananas.

d. Determine the estimated number of days for a banana to ripen and provide suggestions on whether it is best for immediate consumption or storage.

3. Develop a system capable of recording the history of all the scanned bananas and allow the user to Update, Delete or Clear the recorded details.

4. Test and evaluate the application through Unit Testing, Integration Testing, System Testing and Accuracy Testing.

5. Evaluate the efficiency of a developed mobile application using the ISO/IEC 25010 Standard Model.

Purpose and Description

Empowering consumers to purchase bananas at ideal ripeness using image processing technology in a smartphone application, thereby conserving valuable agricultural land, is one of the key objectives of the Sustainable Development Goal. This technology also optimizes resource utilization throughout the entire supply chain and marketing, aligning with the principles of efficient resource management and responsible consumption. By becoming active participants in a more sustainable food system through informed purchase decisions, consumers can directly contribute to land preservation and responsible resource management.

By integrating a smartphone application leveraging Smart Engineering ICT, consumers and retailers can leverage image processing tools to classify the type of banana and determine its ripeness stage (unripe, ripe, or overripe), directly contributing to Agri-Fisheries and Food Security. This technology optimizes resource utilization throughout the supply chain. It empowers consumers to make informed

purchasing decisions, ultimately boosting industrial competitiveness within the agricultural technology sector through innovation and market opportunities.

Furthermore, the insights gathered from interviews with retailers showed a strong interest in using technology to figure out the ripeness of bananas in their inventory. At the same time, consumers emphasized the need for a reliable solution to assess banana ripeness before buying. The fact that both retailers and consumers acknowledged these challenges highlights the essential importance of the proposed mobile application. Moreover, it is important to highlight that most interviewees shared the belief that having a mobile application to determine banana ripeness would be quite helpful. This acknowledgment underscores the need to implement this mobile application to address common concerns and improve the overall banana purchasing experience.

Contextually, this study would benefit the following:

Retailers can track the ripeness of their banana inventory by precisely tracking the ripeness of their stock. Additionally, retailers can consistently offer high-quality bananas among their customers.

Consumers can confidently purchase bananas at their peak ripeness, ensuring optimal flavor and texture. This user-friendly system empowers them to easily assess ripeness, eliminating the guesswork and frustration of buying unripe or overripe fruits. Additionally, it allows them to choose bananas based on their desired level of ripeness, their preferences, or their intended use.

Future researchers can use this study as a reference if they decide to further explore the subject.

Scope and Limitations of the Study

The "Peel Perfect" application is set to develop a machine learning algorithm for banana ripeness detection, aiming to revolutionize consumer and retailers' assessment of banana ripeness. The system will detect different banana varieties, including *Lakatan, Latundan, Saba, and Señorita,* popular among Cavite markets. It will assist consumers and retailers in making informed decisions regarding ripeness before purchasing. The system will also recognize and analyze entire banana clusters or bunches. The development process involves creating a reliable algorithm that uses image processing techniques to analyze visual cues like color and size to determine the ripeness level.

Data Collection Module. This module involves gathering a comprehensive and varied datasets of banana images, composed of different varieties (e.g., *Latundan, Lakatan, Saba,* and *Señorita*) and their different ripeness level (e.g., unripe, ripe, and overripe). To ensure the accuracy of the data, using a smartphone camera capable of producing clear and detailed images is crucial. The collection of datasets will be crucial for training and evaluating the machine learning models responsible for detecting banana types and ripeness. This module establishes the groundwork for constructing a machine learning model proficient in identifying the type and ripeness level of bananas.

Preprocessing Module. This module is designed to standardize and enhance the banana images and ensure consistent and optimal performance by the machine learning model. This module handles several crucial tasks to prepare the images for analysis such as image scaling, normalization and standardization. By adjusting the images to a uniform resolution, which is vital for maintaining consistent dimensions across all inputs. This uniformity is essential for reliable processing feature extraction. Furthermore, it ensures that the image data format is compatible with the requirements

of the machine learning model, guaranteeing that the images are correctly interpreted and processed.

Training Module. This module involves using YOLOv8 to learn patterns and features from an image dataset with its annotation of banana images. The training process involves the backbone, neck, and head where the backbone focuses on extracting features in the image using a series of Convolutional Neural Networks. The neck refines the feature extracted from the backbone by combining the features from different levels. Lastly, the head predicts the bounding boxes, class probabilities, and objectness scores for the output. The model created will be converted into Tflite for Android development.

User Interface Module. In this module, the system is designed to be user friendly and effective, specifically in satisfying the needs of the retailers and consumers in Cavite. Developers focus on a user-centric design featuring a simple yet visually appealing layout that is responsive to various mobile devices. The users will be able to capture images of bananas through a dedicated image capture interface and alternatively users will be able to upload pre-existing images from their device's gallery. Lastly, the users will be presented with a simple visualization of the detected banana ripeness stages and types, providing an intuitive overview of the analysis results from the mobile application.

Evaluation Module. This module evaluation will be conducted to the system using separate data sets to assess its ability and performance on new banana images. Performance metrics, including accuracy and precision, are analyzed to pinpoint areas for refinement. Evaluation will involve analyzing the content of the bananas based on the extracted features, such as their type and ripeness level. The output will include basic information about each banana and recommendations for their use, based on the evaluated content.

The application will be specifically designed for smartphones and will be accessible to consumers and retailers in Cavite. The application features a user-

friendly interface for capturing banana images and receiving real-time ripeness feedback. The smartphone's local storage will serve as the primary repository for the images and extracted data, ensuring the information is readily accessible for reference or reanalysis.

The dataset will include statistical data on various bananas, detailing the number of images, banana types (e.g., Lakatan, Latundan, Señorita, Saba), and ripeness stage distribution (unripe, ripe, overripe). Visualizations like bar or pie charts will show the ripeness category proportions, while summary statistics (mean, median, standard deviation) will highlight the dataset's characteristics. This concise presentation aids in understanding the dataset's diversity, crucial for effective machine learning model development.

During the evaluation phase, the researchers conducted a convenience sampling to gather information from banana vendors and consumers in Cavite. The team visited locations such as markets, roadside stands and popular spots where bananas are sold to swiftly gather data from willing participants who are easily accessible.

The system presents significant limitations that utilize the use of image processing and machine learning methods to assess the ripeness and types of bananas. The system only includes four (4) banana varieties and other banana varieties are not considered in the development. Thus, the system's effectiveness may be compromised and not perform optimally for bananas outside the specified types.

Though the system focuses on visual detection methods, such as image processing and computer vision for ripeness detection, other factors influencing banana ripeness and other non-visual methods for assessing banana ripeness are not included. Another limitation of the system is its incapacity to accurately assess the internal ripeness of bananas when it contradicts the external peel indicators. The researchers’ approach relies mainly on analyzing external visual traits such as peel

color, texture, and visible blemishes, but it does not address situations where the external appearance may not accurately reflect the true ripeness of the fruit.

Also, one notable limitation of the system is its inability to accurately diagnose or identify diseases affecting the bananas. While the system is effective in classifying the ripeness of bananas based on visual cues, it does not encompass the detection of any pathological conditions or diseases that might present in the bananas. Banana diseases are often presenting a subtle change that is not detectable by standard imaging processing. Therefore, it may require advance spectral imaging that uses specialized sensors for identification, and this exceeded the capabilities of the ripeness detection system.

Conceptual Framework of the Study

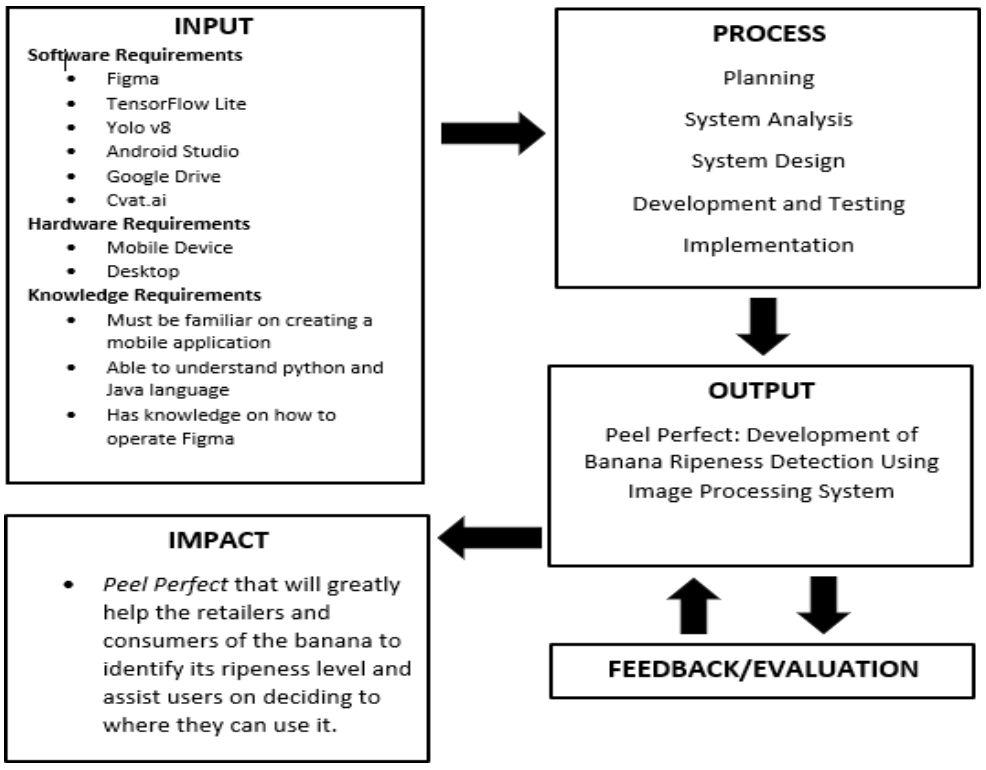


Figure 1. The Conceptual Framework of the Study

A conceptual framework outlines how the variables link to one another by highlighting how the study's pertinent objectives fit together to yield logical findings (Swaen & George, 2022).

The focus of the study is to determine the ripeness level of the banana by processing the image that has been uploaded or captured to help users decide what they do to the processed banana. The limitation is projected using the Input-Process-Output (IPO) model. According to Canonizado (2021), the IPO model comprises everything that makes up a process including the materials and data to be collected, specific details of the operation, and the proposed outcome/s of the study. It gives direction to the researchers on how to administer the study by constructing a series of actions to be done during the conduct of the research.

Moreover, this study was guided by the framework as shown in Figure 1. The first frame from the left represents the input which contains the requirements in order to complete the system. It consists of the software requirement, hardware requirement and knowledge requirement. The second frame refers to the process which includes the planning, system analysis, system design, development and testing and implementation. The third frame is the output tab that consists of the system itself. Feedback and evaluation are also included for the future patch and fix of the system. Lastly, the impact tab shows the potential impact of the system to the community.

Definition of Terms

Convolutional Neural Networks (CNN) are a class of deep learning algorithm commonly used for image processing. CNN extracts key features from images, allowing accurate tasks like classification and object detection.

Image Processing refers to the manipulation or analysis of digital images (Chor *et al.*, 2022), employing computational techniques that encompass tasks such as enhancement, restoration, segmentation, and recognition.

Preferred Installer Program (PIP) is a command-line utility that installs, reinstalls or uninstalls PyPI packages with one simple command: pip.

Python is a programming language for the image processing thesis project and provides a robust foundation with its user-friendly syntax and extensive libraries.

You Only Look Once (YOLO) is a family of algorithms in computer vision excelling at real-time object detection (Redmon *et al.,* 2016). Unlike traditional methods, YOLO employs a single neural network to perform both object classification and bounding box detection in one go.

REVIEW OF RELATED LITERATURE

Technical Background

This chapter serves as a prelude phase in laying the foundation for the development of "Peel Perfect", an advanced mobile application designed to automate the detection of banana ripeness through advanced image processing techniques. The application aspects include a user-friendly interface, and it is designed to perform precise categorization and identification of banana types (Lakatan, Latundan, Saba, and Señorita). Additionally, the system will ascertain the ripeness level, distinguishing between unripe, ripe, and overripe conditions. Finally, the application will present comprehensive information about the banana, including the pieces of banana in a cluster.

To build the system, the following software tools are essential: Android Studio, a powerful integrated development environment (IDE) specifically designed for android app development; TensorFlow Lite, an open-source machine learning framework for building, training and deploying models on android devices; YOLOv8 (You Only Look Once version 8), a popular object detection algorithm known for its real-time processing capabilities; CVAT.ai (Computer Vision Annotation Tool), a versatile platform for annotating images to create training datasets; Visual Studio Code, a versatile code editor with debugging and syntax highlighting; Figma, a collaborative design tool for creating user interfaces and prototypes; and Google Drive, a cloud storage service for storing and sharing project files. These tools collectively facilitate tasks ranging from UI design and machine learning to application deployment, providing a comprehensive foundation for the system's development and functionality. In the literature review, there are scholarly books, research papers, theses, studies, *etcetera.* This includes insights from various studies that employ advanced technologies, including image processing, machine learning, and computer vision, to address the challenges associated with classifying banana ripeness. It aimed to

develop automated systems that can streamline the ripeness detection process, reduce post-harvest losses, and enhance overall efficiency in the agricultural and food industry.

Related Literature

Programming Language

Image Processing with Python Libraries. Python is widely recognized as the optimal programming language for artificial intelligence and more specifically, computer vision, owing to its rich set of libraries that expedite tasks and produce satisfactory outcomes (Asaad *et al.,* 2023). In relation, Computer vision, an essential part of artificial intelligence, lets computers "see" and understand images as humans do. Image processing is a part of computer science, and involves tweaking images to make them better or to get useful information from them. It is like when someone wants a computer program to tell if a picture has a house or not. However, one needs to adjust first the pictures to ensure they are the same size. That is where image processing comes in handy, making sure the data is ready for the computer to understand. This is just one example of why image processing is crucial for many computer vision applications.

Types of Bananas

An Overview of Philippine Bananas. Occupying the prestigious position of the world's second-largest banana exporter, the Philippines stands testament to the profound economic influence wielded by this seemingly simple fruit (Mendiola, 2022). Bananas constitute a crucial component of the nation's economic fabric, acting not only as a significant source of revenue for its farmer population but also spawning an entire byproducts industry, further diversifying and strengthening the economy. Moreover, on a personal level, bananas are adored for their delightful taste and valuable nutritional content, solidifying their multifaceted relevance in the lives of Filipinos. According to

Mendiola (2021), the Philippines has over 20 known varieties of bananas. In terms of export, Cavendish is the top as well as the largest in terms of production. However, Saba, Lakatan, and Latundan are most commonly found on any household's tray.

Level of Ripeness

Banana Ripeness Identification and Classification Using Hybrid Models with RESNET-50, VGG-16 and Machine Learning Techniques. Bananas can be classified into four categories based on their ripeness: unripe, mid-ripe, ripe, and overripe (Matthew *et al.,* 2023). The ripeness of a banana is determined by factors such as its color, texture, and firmness. Unripe bananas are characterized by their green color, firmness, and whether they are ready to eat or not. The color may also indicate that they are not yet fully matured and may have a starchy taste. A mid-ripe banana may have a greener color and a slightly firmer texture. Ripe bananas, on the other hand, have a yellow color with some brown spots and are easy to peel and have a softer texture compared to unripe bananas. Also, they are sweet and possess a pleasant flavor. Overripe bananas are identified by their brown or black color and mushy texture and may have a strong aroma signifying that they have surpassed the stage of being fully ripe and may have a sweeter taste.

Banana Freshness Identification Using Image Processing Techniques. According to a study, evaluating freshness is crucial for maintaining both product integrity and financial viability (Mehendran *et al.,* 2022). The traditional method for measuring the ripeness of a banana by day relies on the visual assessment of knowledgeable specialists but not every moment a specialist is available. As stated by Mehendran *et al.* (2022), one way to measure the freshness of a banana is by using a Support Vector Machine (SVM), which is used to estimate the freshness by training using the selected features from the input images.

Induced Ripening Agents and Their Effect on Fruit Quality of Banana. (Maduwanthi, S. D. T., & Marapana, R. 2019). The life cycle of a fruit is divided into three stages: fruit set, fruit development, and fruit ripening. Fruit ripening marks the beginning of fruit senescence, a genetically programmed and highly coordinated process that transforms an organ from its unripe to its ripe stage, resulting in an appealing and edible fruit.

Fruit ripening involves a combination of physiological, biochemical, and molecular processes that lead to changes in color, sugar content, acidity, flavor, aroma, and texture. Since bananas are climacteric fruits, they are typically harvested at the pre-climacteric stage and artificially ripened for commercial purposes. Artificial ripening allows traders to reduce losses during transportation and release the product at the desired ripening stage. According to Maduwanthi *et al.* (2019), bananas can be artificially ripened using various ripening agents such as ethylene gas, ethephon, acetylene (emitted from calcium carbide), ethylene glycol, and alkyl alcohols (containing 6-14 carbon atoms such as lauryl alcohol). Ethylene gas is the preferred method for banana ripening in developed countries due to its ability to control and accelerate the ripening process. Modern banana ripening rooms ensure consistent fruit quality by precisely regulating temperature, humidity, and ethylene gas concentration. Also, traditional methods of banana ripening include smoke generated from burning green leaves.

Assessment of External Properties for Identifying Banana Fruit Maturity Stages Using Optical Imaging Techniques. (Zhuang et al. 2019). Visual characteristics of bananas, such as color, texture, and shape, can be used to determine their ripeness. For instance, the appearance and distribution of brown spots on the peel can reveal the maturity stage of a banana. The shape of the banana peel changes as it ripens and this change can be used to determine the ripeness of the banana. Based on Zhuang et. al (2019), one way to measure this change is to use a

Histogram of Oriented Gradients (HOG). HOG is a method that can be used to describe local shape information by calculating the distribution of local edge directions and intensity gradients on a dense grid. Therefore, a combination of image feature extraction and pattern recognition techniques could be an alternative method for identifying the maturity stage of bananas.

Banana Ripeness Classification using HSV Colour Space and Nearest Centroid Classifier. (Tamatjita and Sihite 2022). Banana is one of the most common fruits in any market, it's delicious and healthy and can be used for culinary dishes or as a dessert. Even though bananas are widely accessible, numerous people still struggle to identify the ideal ripeness stage for bananas. The research investigated by Tamatjita and Sihite (2022) stated that Hue, Saturation, and Value (HSV) color features can be used for classifying the ripeness stage of bananas. Ambon Lumut, Kepok, and Raja mentioned in their study that bananas are used to test the effectiveness of HSV. Thus, the study successfully identifies four classes of ripeness stage (green, almost ripe, ripe, and overripe), and the overall average classification that was achieved is at the rate of 73.33%. However, the study identifies a potential limitation in separating the ripe and overripe stages due to subtle color tone differences, specifically in the darkness of the peel. Notably, the removal of the background enhances the result of distinguishing the ripeness stages (Tamatjita & Sihite, 2022).

Detection of Banana and Its Ripeness Using Residual Neural Network. (Dhaniswara et al. 2021). Automatic fruit detection utilizing computer vision techniques has been carried out to help the agriculture and plantation industries. The study researches smart systems to detect bananas and ripeness classification utilizing residual neural networks. Furthermore, the method used to detect bananas is transfer learning pre-trained model VGG-19. Sliding windows are used to detect the position of bananas followed by Non-Max Suppression to summarize the results of several detected bananas. The banana ripeness is detected by raw, ripe, and overripe. The

result of banana detection testing percentage using VGG-19 shows 100% for ripe bananas, 99% for raw bananas, and 100% for overripe bananas.

Machine Learning Methods

Deep Learning Based Intelligent Identification System for Ripening Stages of Banana. (Thayumanavan et al. 2023). When bananas are handled, shipped, carried on board, and kept by ripe hands, post-harvest losses increase and trade is hindered. Knowing when something is ripening is crucial to minimizing loss. For bulk handlers as well as the food processing industry, automated, non-destructive techniques for determining ripening stages are essential. The study found that banana fruit can be classified as underripe, ripe, overripe, or unripe. The foundation of this non-destructive categorization method is deep learning. Furthermore, the CNN-XgBoost algorithm is defined as a convolutional neural network (CNN) coupled with an eXtreme Gradient Boosting (XgBoost) technique for the efficient assessment of banana ripening. CNN is utilized as the trainable feature extractor for the images, and XgBoost is employed to determine the ripening stage. Using Linear Discriminant Analysis (LDA) eliminates the need for augmented or large-scale data sets. Consequently, the proposed deep learning technique can perform classification even with a smaller data set, and its total performance accuracy is 91.25%.

Banana Fruit Classification using Convolutional Neural Network. (Samad et al. 2022). There are five stages involved in the classification of banana fruit using a Convolutional Neural Network: input image, preprocessing image, segmentation image, classification image, and performance measure. The input images represent three maturity stages of banana fruits: raw banana, ripe banana, and overripe banana. These images serve as the primary data source for the classification process. The preprocessing stage involves resizing the image, converting it from RGB to grayscale, and filtering it using the median filtering method which is applied to eliminate noise and smoothen the grayscale image, improving its clarity and reducing unwanted artifacts.

and calculating the mean squared error (MSE) and peak signal-to-noise ratio (PSNR). MSE measures the average error between the original and preprocessed images, while PSNR quantifies the ratio of desired signal to noise in the image. The segmentation stage involves segmenting the filtered image using the Sobel edge detection method. The Sobel edge detection method is employed to detect edges and delineate the boundaries of the banana fruit. The classification stage employs a ResNet classifier to categorize the bananas. This powerful neural network model is trained to recognize the distinct features of raw, ripe, and overripe bananas, enabling it to accurately categorize the banana fruit in the image. The performance measure stage calculates the accuracy and error of the classification system. Accuracy measures the proportion of correctly classified images, while error represents the percentage of misclassified images. These metrics provide insights into the effectiveness of the classification process.

Ripeness Classification of Bananas Using an Artificial Neural Network. (Mazen et al. 2019). The freshness and maturity of bananas are crucial factors for both consumers and fruit companies. Accurately and efficiently classifying banana ripeness is essential for determining its quality. Image processing tools can be developed and implemented to effectively categorize the ripeness stage of incoming banana brunches. Banana ripeness plays a significant role in both its taste and market value. An Artificial Neural Network-based framework that utilizes color evaluation, brown-spot recognition, and Tamura texture metrics is employed to classify and determine the ripeness level of bananas.

An Innovative Approach for Fruit Ripeness Classification. (Thakur et al. 2020). Assigning items to their categories can lead to rigid opinions and is more costly in terms of time. Finding a budget yet accurate method for determining the classification of fruits and their ripeness level remains a problem. Surface color, size, and shape are necessary for successful classification. Based on Thakur et al. (2020),

one way to measure the ripeness of a fruit is by using convolutional neural networks. It utilizes surface color, size, and shape as critical features for ripeness determination. The CNN effectively extracts these features from strawberry images and achieves a remarkable accuracy of 91.6%, significantly outperforming manual classification methods. Thus, this approach offers a promising solution for enhancing the efficiency and precision of strawberry ripeness evaluation.

Classification of Banana Ripeness Based on Color and Texture Characteristics. (Kahfi et al. 2023). Another way to identify the maturity level of bananas is by observing the texture and color of the bananas. The peel of the banana acts as an indicator of its maturity, the changes of the vibrant color green to pale yellow and eventually into brown speckles color accompanied by the visual change in texture with the firm. A study by Kahfi et al. (2023), implemented digital image processing techniques that focus on color and texture, for accurate classification of banana ripeness level. By utilizing the K-Nearest Neighbor (KNN) algorithm with a parameter value of 3 achieved a remarkably high accuracy, reaching an overall average of 86%. Furthermore, the digital image extraction techniques that emphasize the extraction of energy, contrast, correlation, and homogeneity values based on color and textures emerge as an important factor that increases the classification of banana ripeness level. The study demonstrates that digital image processing, combined with the KNN algorithm and extraction techniques has shown a precise method for automated banana ripeness assessment (Kahfi et al., 2023).

Evaluating the Ripening Stages of Musa Acuminata × Balbisiana (Saba) using 2D-image Analysis. (Moso et al. 2021). Utilizing 2D- image analysis for banana quality assessment not only offers a non-destructive method but also provides an accurate classification of ripening stages. In a study presented by Moso et al. (2021), the extracted dataset of 90 images employing color and texture features was used to train a Support Vector Machine (SVM) for the classification of 30 Saba banana

samples into unripe, ripe, and overripe categories. Impressively, the utilized features demonstrated a 70% accuracy in the classification of the ripening stages, with SVM achieving a 100% accuracy specifically for ripe banana images. Therefore, 2D-image analysis presents a promising non-destructive alternative for classifying the ripening stages of Saba bananas and potentially other comparable fruits within the food industry (Moso et al., 2021).

Fruit Ripeness identification using YOLOv8 Model. (Xiao *et al.,* 2023). Deep learning-based visual object detection has become a crucial aspect of computer vision, enabling the precise localization and classification of multiple objects within images. This research focuses on the classification of fruits as ripe or overripe using digital imagery. The proposed model uses advanced deep learning techniques to extract visual features from fruit images, specifically analyzing the characteristics of the fruit peel to predict its class. The study uses custom datasets to train two anchor-free models: YOLOv8 and CenterNet. CenterNet uses ResNet-50 for feature map upsampling and DeConv for heatmap prediction. YOLOv8 incorporates CSP and C2f modules for efficient processing. Comparative analysis shows that the C2f module in YOLOv8 significantly enhances classification performance, achieving an impressive accuracy rate of 99.5%.

Review of Related Study

Fuzzy Logic-Based Size and Ripeness Classification of Banana using Image Processing Technique. (Malabag et al. 2022). Achieving the ideal ripening process for bananas involves careful monitoring of the various ripening stages to prevent fruit spoilage and ensure optimal quality. To accurately assess the ripeness of bananas using image processing techniques, the researchers employed the RGB color detection approach in MATLAB. This method enabled the extraction of relevant features from the image, including area, main axis length, and minor axis length, which serve as indicators of banana size and ripeness. According to Malabag *et al.* (2022),

combining MATLAB's image processing toolbox and fuzzy logic effectively classifies bananas based on both size and ripeness. Size classification is achieved using predefined criteria for length and thickness, determined based on the main and minor axis lengths of the banana, measured in pixels. Meanwhile, image processing techniques such as color segmentation and mean color intensity calculation based on RGB values are utilized to classify ripeness.

Banana Ripeness Classification Using Computer Vision-based Mobile Application. (Mohamedon *et al.,* 2021). The integration of smartphone applications and the expanding power of artificial intelligence empowers users to tackle everyday tasks in innovative ways, promoting practicality and efficiency. To accurately and efficiently determine the ripeness of the banana fruits, they implemented a computer vision technique. Image classification is performed by utilizing a pre-trained model to leverage its ability to extract edges and identify patterns in images. The convolutional neural network (CNN) model is used to train the classifier. According to Mohamedon *et al.* (2021), Google Colab is utilized for the execution of the code, and it is well-suited for machine learning, while TensorFlow Lite makes the process simpler when it comes to adapting and converting a neural network model to particularly input data before installing it to an android application.

Determining Banana Types and Ripeness from Image using Machine Learning Methods. (Sabilla *et al.,* 2019). The authors explore several machine learning algorithms to identify the types and ripeness of bananas based on their category from images. The study examined a preprocessing method that involved grayscale and determining the dominance of colors red, Green, and Blue (RGB) at each pixel. The incorporation of PCA (Principal Component Analysis), was explored and this plays a role in processing time speed without sacrificing the accuracy of the algorithm. The findings of the research indicate that the Linear SVM, partnered with preprocessing methods and efficiency of PCA, achieves a remarkable 99.1% accuracy

in classifying bananas based on their classification and outperforms the other algorithm, namely k-Nearest Neighbour (k-NN) and Decision Tree (DT). This research provides significant advancements in the application of machine learning to agricultural and food quality assessment (Sabilla et al., 2019).

Developing a Colorimetric Equation and a Colorimetric Model to Create a Smartphone Application That Identifies the Ripening Stage of Lady Finger Bananas in Thailand. (Tanut *et al.,* 2023) The study develops a colorimetric equation and colorimetric model to create a smartphone application that identifies the ripening stage of lady finger banana (LFB). The system photographs the LFB and automatically analyzes the skin color of the banana and gives the user information about the number of days until the banana ripens and the number of days that it remains edible. According to Tanut *et al.* (2023), the application is called the Automatic Banana Ripeness Indicator (ABRI, pronounced like "Aubrey"), and rapid analysis will be given considering it will be useful for anyone who is involved in the storage and distribution of bananas. Furthermore, the colorimetric equation interprets the skin color with the CIE L*a*b* color model in conjunction with the Pythagorean theorem. The colorimetric model has three parts. First, COCO-SSD is an object detection tool that identifies and locates the banana in the image. Second, the Automatic Power-Law Transformation adjusts the illumination to a standard derived from the average of a set of laboratory images. After the removal of the background image and conversion of the image to L*a*b*. The data are sent to the colorimetric equation to calculate the ripening stage. The results show that ABRI correctly detects a banana with 91.45% accuracy and the Automatic Power-Law Transformation correctly adjusts the image illumination with 95.72% accuracy. In conclusion, ABRI is thus an accurate and robust tool that quickly, conveniently, and reliably provides the user with any LFB's ripening stage and the remaining days for consumption.

Table 1. Table of Comparison

	STUDY	RS 1	RS 2	RS 3	RS 4	RS 5
	Capture Banana	✓	✓	✓	✓	✓
	Classification of Banana	✓			✓	
Features	Determines Ripeness Level	✓	✓	✓	✓	✓
	Gives Recommendation	✓				✓
	Can Detect Cluster of Banana	✓				

Table of Comparison Legend

Research Study 1 - Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

Research Study 2 - Fuzzy Logic-Based Size and Ripeness Classification of Banana using Image Processing Technique.

Research Study 3 - Banana Ripeness Classification Using Computer Vision-based Mobile Application

Research Study 4 - Determining Banana Types and Ripeness from Image using Machine Learning Methods

Research Study 5 - Developing a Colorimetric Equation and a Colorimetric Model to Create a Smartphone Application That Identifies the Ripening Stage of Lady Finger Bananas in Thailand

Synthesis

The study entitled "Fuzzy Logic-Based Size and Ripeness Classification of Banana using Image Processing Technique" uses the fuzzy logic technique of MATLAB, classifying the banana sizes using the metrics of length and thickness and banana ripeness using color segmentation based on RGB in image processing.

Meanwhile, the study entitled "Banana Ripeness Classification Using Computer Vision-based Mobile Application" uses a Convolutional neural network (CNN) to determine the ripeness of bananas by identifying the patterns in the images. Google Collab is also utilized for code execution while TensorFlow Lite makes the process of adapting and converting easier.

The study entitled "Determining Banana Types and Ripeness from Image using Machine Learning Methods" employs a preprocessing method that applies grayscale and classifies the dominance of colors red, green, and blue (RGB) to identify the ripeness of bananas. Three machine learning algorithms were used: Support Vector Machine (SVM) which has the highest accuracy of 99.1% for determining the type of banana, while both k-Nearest Neighbor(k-NN) and SVM achieved the same highest result of 96.6% for determining the banana ripeness leaving the Decision Tree (DT) as the lowest among the three.

The study entitled "Developing a Colorimetric Equation and a Colorimetric Model to Create a Smartphone Application that Identifies the Ripening Stage of Lady Finger Bananas in Thailand" created a mobile application called Automatic Banana Ripeness Indicator (ABRI). It uses a colorimetric model and it has three parts. First, COCO-SSD is object detection to locate and identify the banana in the image. Second, Automatic Power-Law Transformation to adjust the illumination to standard derived from a set of laboratory images. Last is removing the background image and converting it to L*a*b* then will be sent to the colorimetric equation to calculate the ripening stage. The result shows 91.45% accuracy thus ABRI is an accurate and robust tool that provides users the ripening stage and remaining days for consumption of the banana. The strength of this study is its mechanism to support three different types of bananas which are Lakatan, Latundan, Senorita and Saba. The system will be able to detect the ripeness level of bananas and also display information about the different richness of nutrients that the user can get per level of ripeness. It will also give food recommendations for the user to process or cook which will also depend on the

banana's level of ripeness. Hence, the developers intended that the mobile application would be user-friendly and easily accessible by the user.

METHODOLOGY

Requirement Analysis

Peel Perfect is a software system that uses image processing to ensure perfect banana choices, eliminating the guesswork and disappointment of choosing bad ones. This system requires user input to function.

The user simply captures or uploads a picture of the banana, and the system's algorithms spring into action. Analyzing the image with meticulous precision, they unlock the banana's secrets, revealing both its ripeness and type.

Overall, Peel Perfect requires only a picture of a banana as input. The system then analyzes the image and compares it to the trained machine learning model of banana characteristics to provide approximately correct results. Clear and concise results empower the user to make the perfect selection.

Data Flow Diagram

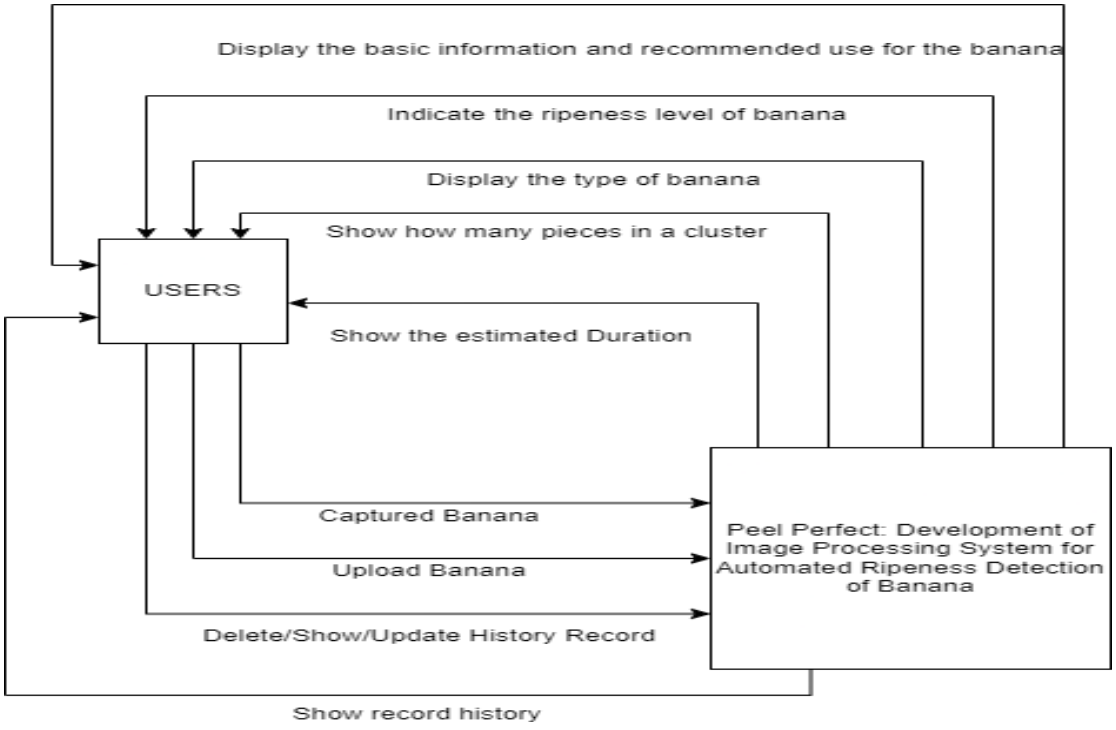


Figure 2. Peel Perfect System Overview Process

Requirement Documentation

The development of Peel Perfect will be crucial to the banana food industry and this documentation will provide an overview of the features and functionalities that will be expected in the mobile application.

Table 2. Main Features

MAIN FEATURES	DETAILED FEATURES
Input Banana Image	Peel Perfect shall get an image of a banana by capturing real time or uploading a photo.
Process the Image	Peel Perfect shall process the collected image by identifying the following: 1. Different types of bananas 2. Level of ripeness of banana
Display the Result	Peel Perfect shall give an output of the following: 1. Display the specific type banana 2. Display the ripeness level of the banana 3. Display the confidence level of the banana 4. Display the recommended use of the banana 5. Display number of pieces in cluster of bananas 6. Show and update the records of banana in the history 7. Show estimated number of days for banana to ripen

Table 3. Quality Attributes

QUALITY ATTRIBUTES	DESCRIPTION
Usability	The Peel Perfect shall be easy to operate and easy to understand.
Accuracy	The Peel Perfect shall be approximately accurate on the information that it will output.
Reliability	The Peel Perfect shall consistently perform its function without failures or errors.
Performance	The Peel Perfect shall exhibit responsive and efficient behavior and with acceptable output.

Table 4. Software Requirements and Significance

PLATFORM	SIGNIFICANCE
Android Studio	The platform used to make the system application.
Yolo v8	Used to train and preprocess the dataset.
TensorFlow Lite	Converted the model to this extension for compatibility of android studio
Cvat.ai	Used to annotate the dataset.
Visual Studio Code	The platform used to code the training of the model using python language.
Figma	The platform used to design the user interface of the application
Google Drive	The platform used to store the datasets for image processing

Design and Methodology

Peel Perfect is an image processing application that purposely made to determine the ripeness of the banana and its variety. Image processing is a process that teaches the machine based on the photos or dataset given. Peel Perfect will use this technology to efficiently capture and provide high accuracy on the distinguished banana.

The application Peel Perfect allows user to capture and upload an image of a banana and outputs its ripeness level (Unripe, Ripe, Overripe), its type (Latundan, Lakatan, Saba, Senorita), the basic information about the captured/uploaded banana, the recommended food for its type and ripeness and the duration going to its new stage. With the help of YOLOv8, the application can successfully identify its ripeness and type. The history page allows the user to update the saved banana.

The trained YOLOv8 model identifies the ripeness and type of a banana by processing the captured image through a series of convolutional neural network layers designed to detect and classify objects. The model uses the annotated and labeled data from the training phase to learn features associated with different ripeness levels (Unripe, Ripe, Overripe) and banana types (Lakatan, Latundan, Saba, Senorita). When an image is included into the system, the model applies these learned features to predict bounding boxes around the banana and assigns labels indicating its ripeness and type, providing efficient classification based on the visual characteristics identified during training. Also, the model is responsible for showing the confidence level of the scanned/captured banana that will be displayed on the output page. The formula in getting the confidence level is class score multiply by bounding box score.

The basic information, recommendation and estimated duration before going to another ripeness stage will be based on the detected type and ripeness. The contents will be different depending on the status and variety of the banana. The recommended dishes have a link that will redirect the user on the tutorial on how to make the certain dish (See Appendix Table 15). The duration will also vary depending

on the ripeness level and the type of the banana. The displayed duration is based on the experimental monitoring of the banana (See Appendix Table 11-14).

The history page lets users update information on captured or uploaded images of bananas. This feature helps track the ripeness stages, allowing users to see how many days each banana will take to reach the next stage. This tool aids in managing banana ripeness and ensuring optimal use.

Machine learning Architecture

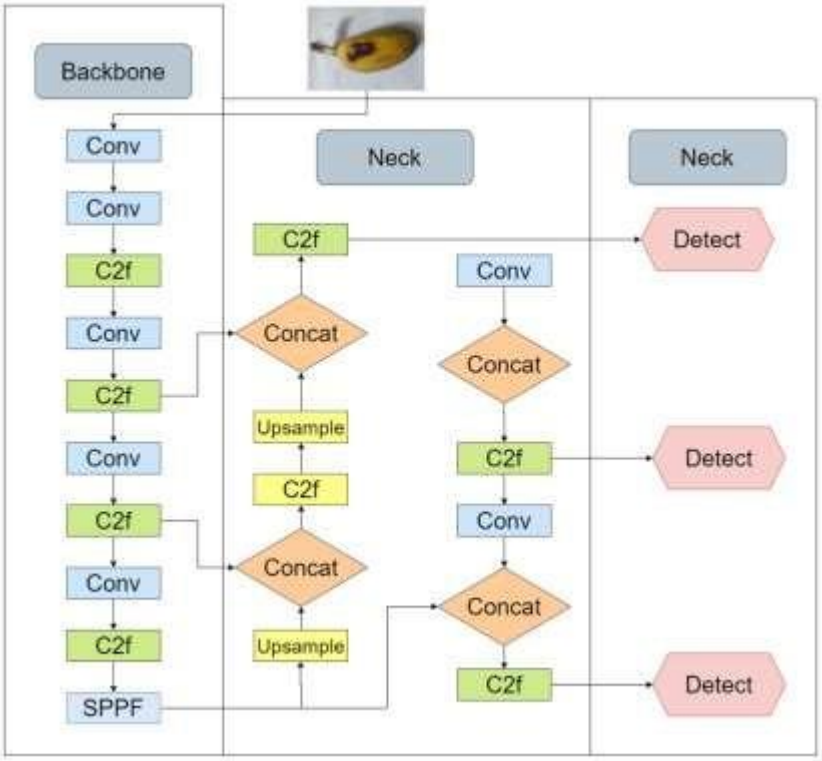


Figure 3. YOLOv8 Model Architecture

The Backbone, Neck, and Head are the three basic components that make up the architecture of the YOLOv8 model. The Backbone is responsible for the initial feature extraction from the input image. Convolution layers use sliding filters across the input image to execute convolutional operations to detect low-level characteristics like edges, textures, and patterns. C2f modules improve information flow while preserving a lightweight structure. These modules, which combine many convolutional

layers, allow the network to capture complicated patterns and correlations in the data. The SPPF (Spatial Pyramid Pooling Fast) module, which pools the feature maps to a fixed size independent of the input size, is the last one used by the Backbone. By utilizing three successive max-pooling layers, this minimizes computing effort and latency while aiding in the summary of feature presence in various picture regions and producing a condensed representation.

The Neck uses a PAN-FPN (Path Aggregation Network - Feature Pyramid Network) structure to improve detection performance for objects of various sizes. The PAN (Path Aggregation Network) merges feature to retain precise location details, while the FPN (Feature Pyramid Network) combines deep semantic information to enrich feature maps with high-level contextual information. This combination of structures allows for the integration of high-level semantic information and low-level spatial details. In order to improve feature representation for object detection, the Neck further processes and refines the features that the Backbone extracted. The Neck improves information flow and feature representation by utilizing C2f modules for further feature extraction and processing. Concatenation operations combine features from several scales and layers to produce feature maps that are richer and more detailed. The detection of tiny objects depends on the integration of data from various network levels. Upsample layers make sure that little details are preserved by improving the spatial resolution of feature maps.

The Head is responsible for detecting and classifying objects in an image using three detection heads corresponding to different scales of features. The detection head is divided into two branches for object classification and bounding box regression. One branch classifies objects into different categories, while the other predicts the coordinates of bounding boxes. Different loss functions are used for these tasks: Binary Cross-Entropy Loss (BCE Loss) for classification, Distribution Focal Loss (DFL) and Complete Intersection over Union (CIoU) Loss for bounding box regression, which

improves the accuracy of predicted bounding boxes and their alignment with actual objects.

System Architecture

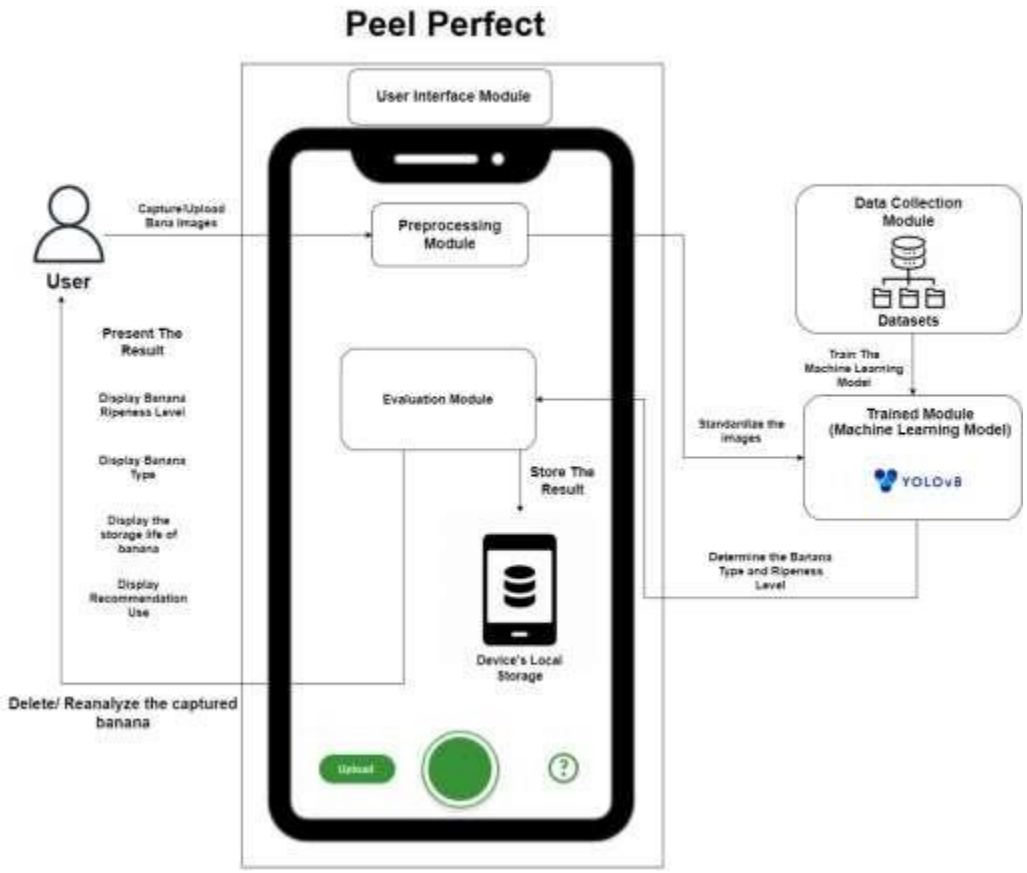


Figure 4. System Architecture of Peel Perfect

The system architecture presents the main structure and functions of each module. The researchers developed a conceptual framework that defines the relationship between different components. Each component serves a specific purpose, contributing to the system's overall functionality and features. The following processes are performed by each element that contributes to the functionality of the system.

The **User Interface Module** facilitates user interaction with the application. The user can access the camera button or upload pre-existing banana images from gallery, initiating the following steps in image processing workflow.

In the **Preprocessing module**, banana's image will be standardized to ensure consistency in format and quality before the analysis. The inputted image will be optimized by the system through various processes like image scaling, normalization and standardization.

The collected **datasets** play a pivotal role in training and evaluating machine learning models, laying the foundation for the development of a proficient model capable of accurately identifying both the type and ripeness level of bananas.

The **Machine learning model**, this analyzes the preprocessed images to identify the type and ripeness level. the **YOLOv8 model** will be employed for image detection and classification of the banana type (Saba, Latundan, Lakatan or Senorita) and Ripeness level (Unripe, Ripe or Overripe). The Model will extract features, form the input image, refine the features, and predict the images with bounding boxes.

The **Evaluation module** will involve analyzing the result from the machine learning model and provide the user feedback, indicating the predicted ripeness level of the banana and the determined type. Additionally, for the approximate duration, recommendation and basic information of banana will be presented based on the analyzed banana. The result will be presented in textual description based on the extracted information from banana.

The **Device's Local Storage** will be used as the primary repository of the captured images and the extracted information about the banana. The users will have an option to save, reanalyze, or reference the captured images as needed. They can also update or delete the recorded history from the history page.

Development and Testing

The researchers employed an iterative model approach to identify design-related flaws at the earliest stages and addressed them in each iteration until the final system was developed.

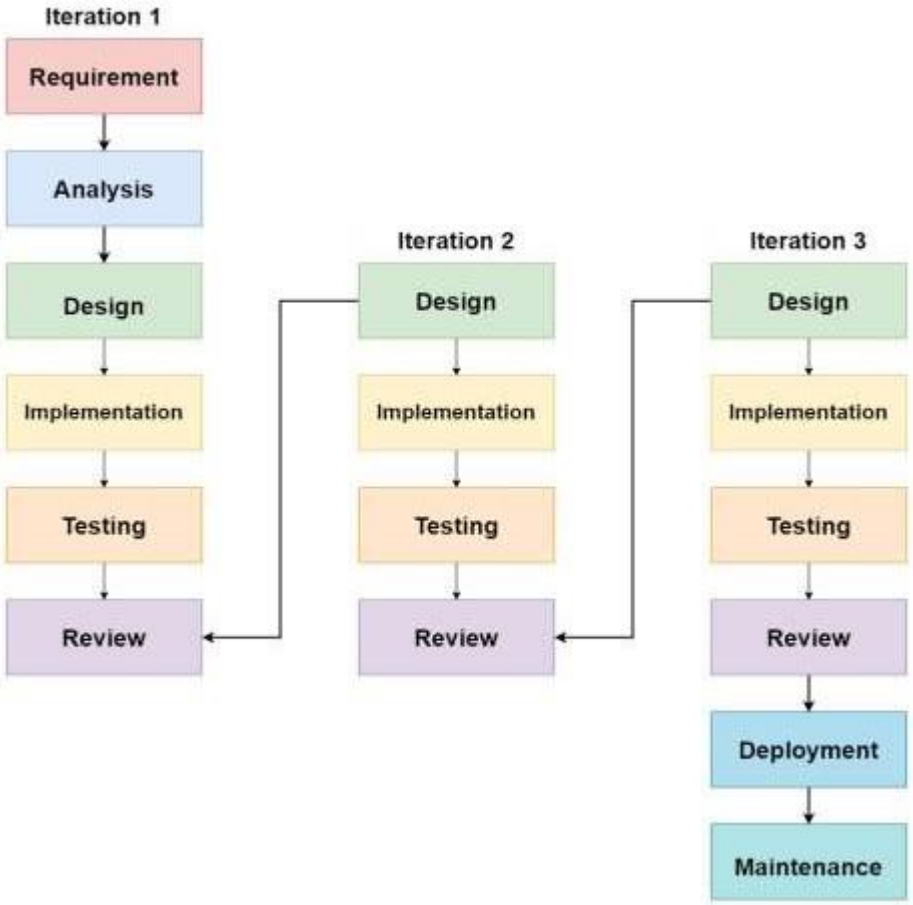


Figure 5. Iterative Model for Development and Testing.

Requirement gathering & analysis: In this phase, the researchers gathered a comprehensive and varied dataset of banana images, including different varieties and their ripeness levels. This dataset is essential for meeting the requirements of developing the application. After completing the dataset collection, the software team proceeds to the next phase of development.

Design: The design phase involves planning and conceptualizing the software requirements, such as the System architecture, User interface of application and its functionalities.

Implementation: This phase involves writing the code and developing the features based on the requirements and design. It is the process of turning the planned design into a working application or system.

Testing: Once an initial implementation is ready, testing begins. Testing ensures that the implemented code functions correctly and meets the specified requirements.

Unit Testing. In this testing phase, each module with its procedures is tested to determine its suitability for the system. This testing can check every unit of an individual module to determine its correctness without encountering any issues.

Integration Testing. Following the unit tests, individual modules are combined into a group and converted into input modules. The input modules undergo testing and processing before being delivered as output modules ready for system testing.

System Testing. This test is conducted for the entire mobile application, encompassing the evaluation of the application against its specific requirements. The scope of system testing includes testing the entire mobile application, and its primary purpose is to detect errors in the Android application.

Accuracy Testing. This testing focuses on ensuring the precision and correctness of the application functionalities, particularly in capturing or uploading pictures to identify the type and ripeness of bananas. This involves capturing data from 60 bananas to determine the average ripeness, type, and the number of bananas per cluster. The data collected from these bananas will be processed using the confusion matrix accuracy testing. This approach ensures that the results are representative and reliable, providing a solid basis for further analysis or application. Through this methodical process, the study aims to offer valuable insights into the average ripeness stage, types, and cluster composition of bananas.

Formula used in accuracy testing:

Legend:
True Positive – TP
True Negative – TN
False Negative - FN
False Positive - FP

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

Review: In this phase, after the product deployment, review phase is performed to check the behavior and validity of the developed product. And if there are any errors found then the process starts again from the requirement gathering.

Maintenance: The maintenance phase involves identifying and fixing any bugs or errors that may arise, as well as implementing new updates to ensure optimal performance of the mobile application.

Evaluation

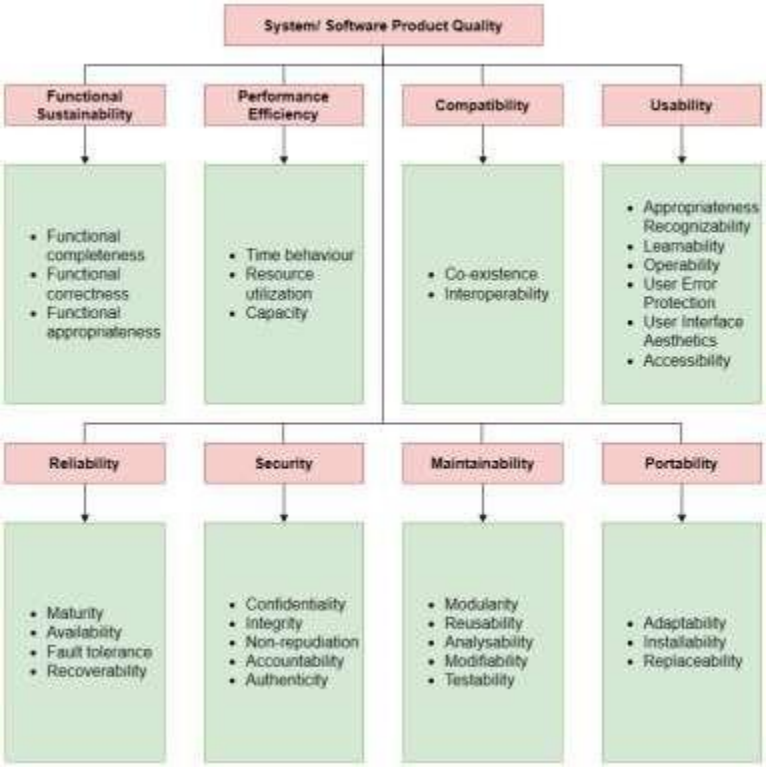


Figure 6. ISO/IEC 25010 Quality Model

To guarantee high-quality performance of the proposed system, the researchers plan to incorporate the ISO/IEC 25010 Standard Model as a fundamental framework to evaluate the system. By employing this widely recognized standard, the researchers aim to systematically assess the various characteristics and sub-characteristics that contribute to the overall quality of the software product. The model uses eight product quality characteristics and 31 sub-characteristics for understanding, following, and refining the selected software as shown in figure.

RESULTS AND DISCUSSION

System Development

In this system, an image processing application capable of identifying banana ripeness as its primary function, while also identifying banana type and quantity as secondary functions were developed. This application intends to replace traditional methods of assessing banana status and variety with cutting-edge technology.

The researchers employed an iterative model approach, which divides the process into iterations, enabling incremental improvements and flexibility in development. This process has phases of software development. The first phase is Requirement gathering and Analysis, the researchers gathered a comprehensive and varied dataset of banana images, including different varieties and their ripeness levels. The researchers gathered 12,000 pictures of bananas for their study. These images were sorted into four types of bananas: *Saba, Lakatan, Latundan, and Señorita,* with each type having 3,000 images. They also classified each banana type into three ripeness stages: unripe, ripe, and overripe, with 1,000 images representing each stage (See Appendix Figure 10).

The second phase was design. This phase involved planning and conceptualizing the software requirements, such as the Data Flow Diagram (DFD), System Architecture, User Interface of the application, and its functionalities. Taking action from this phase, the researchers used software tools to design the system, such as Google Drive as a platform for storing datasets of banana images. The Data Flow Diagram (DFD) was also designed to provide more details about the system, illustrating how data flows between different components. The researchers brainstormed the system architecture of the application on Canva and designed it collaboratively. Figma was the chosen tool for designing the user interface of the application. It allowed for real-time collaboration and streamlined design processes

that made it easier for the researchers to work together efficiently. Additionally, Figma's prototype feature enabled the functionalities of the application to be tested.

The third phase was implementation, which involved writing the code and developing the application and its features based on the requirements and design. This phase required transforming the design phase prototype into a working model. The researchers began developing the actual system. They used various software tools for development, storing datasets on Google Drive and using CVAT.ai to annotate the dataset before proceeding to training. Visual Studio served as the IDE, and Python was the programming language utilized. YOLOv8 was selected as the platform for training the model, leveraging frameworks for computer vision tasks that provided essential functionalities for image processing and machine learning. The model created in YOLOv8 was converted into a TensorFlow Lite model, which was then integrated into Android Studio. Android Studio was the software used to create the application, managing both its UI and functionalities. By using these software tools and frameworks, the researchers successfully developed an initial application.

The fourth phase was testing. Once an initial implementation is ready, testing begins to ensure that the implemented code functions correctly and meets the specified requirements.

The fifth phase was reviewing. In this phase, the researchers conducted an evaluation of the behavior and validity of the developed application, testing its ability to process images, classify banana types correctly, and provide reliable information about the ripeness stage. This assessment focused on ensuring the application met its intended functionality and provided users with dependable results. For any errors found during this evaluation, the researchers initiated problem-solving solutions, which might have involved revisiting the requirement gathering phase to address any discrepancies or identified issues. This phase ensured that the final product met and aligned effectively with the application's objectives.

The sixth phase was deployment. Deployment occurs after the implementation and testing phases, where the fully working application is deployed as a mobile application using Android Studio.

The last phase was maintenance. In this phase it involves actions taken by the researchers to update and modify the software after deployment. The researchers addressed issues such as bug fixes, updates, and enhancements to ensure the software functions effectively and meets user needs. They also monitored performance to maintain the software's functionality and usability. This ongoing process aims to optimize the software's performance and adapt it to changing requirements and environments.

System Overview

This section presents the datasets and sample screenshots of the application.

Figure 7 shows the Landing Page of the Application “Peel Perfect”. It has the Perfect History Card where one can see the captured banana history. The About Card shows the overview and the developers of the application. The Camera Button to proceed to the Camera Page.

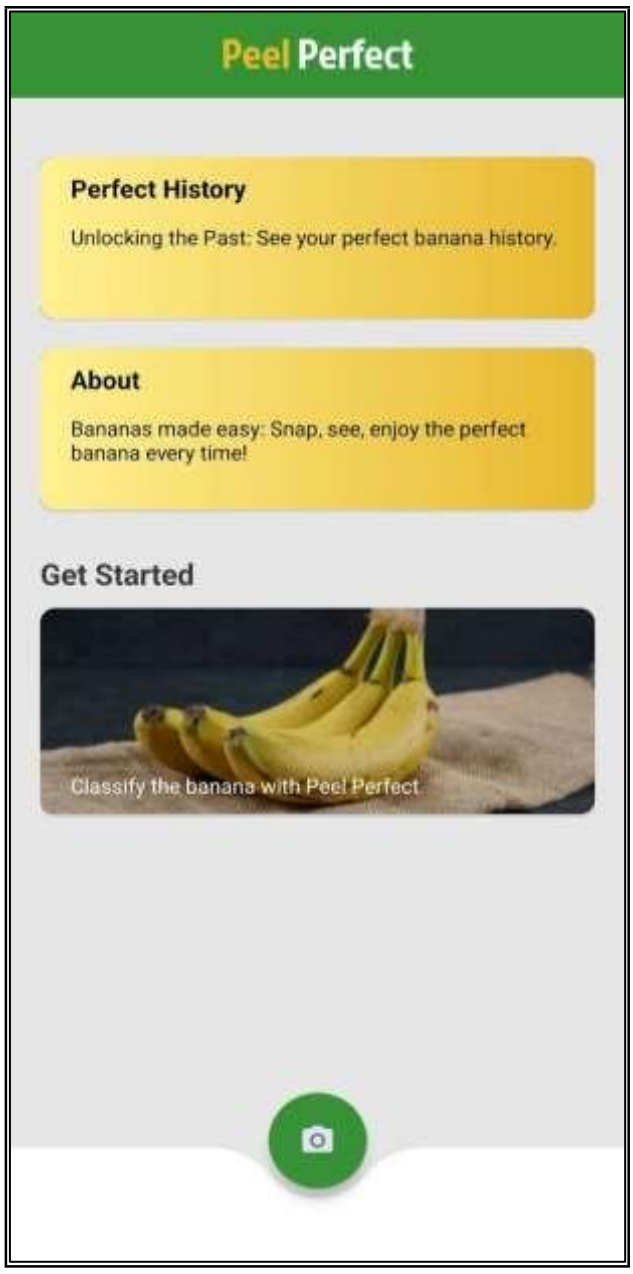


Figure 7. Screenshot of the Landing Page of the Application

Figure 8 shows the Camera page where the users capture the banana and the application will process it. The upload button will prompt the user if they want to upload a picture of the banana. The question mark button is the Capture tips that prompts guide on how to capture a banana.

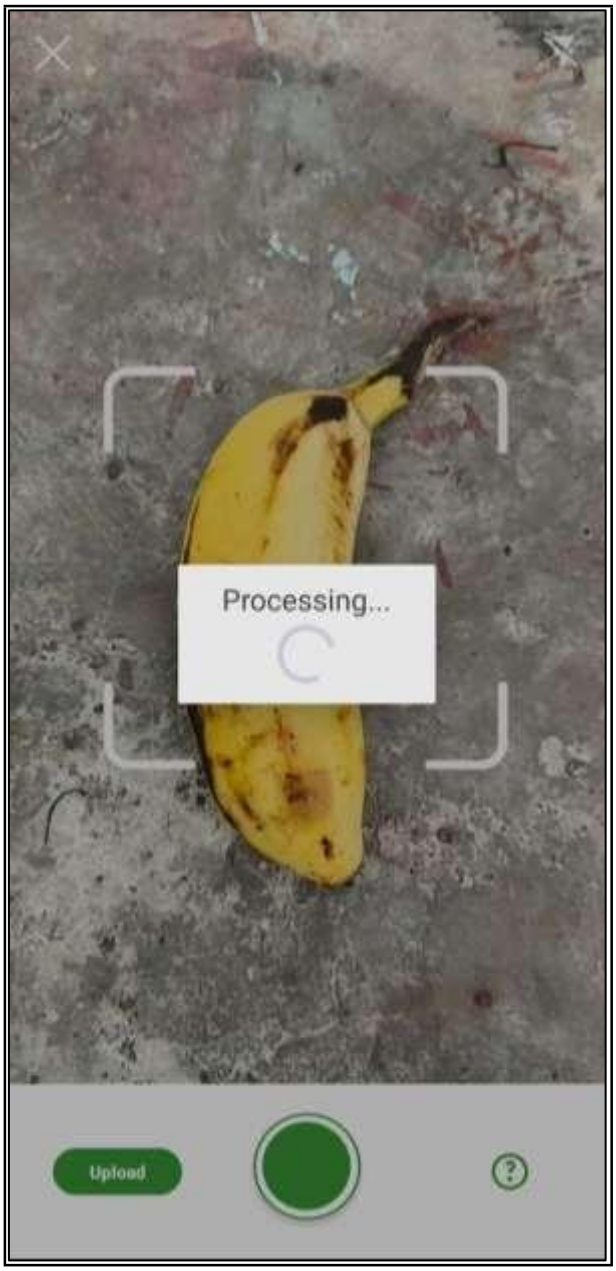


Figure 8. Camera Page for Capturing Banana Image

Figure 9 shows the capture tips card that shows how to perfectly capture a banana.

Figure 9. Screenshot of the Capture Tips

Figure 10 shows the options on where the user wants to get a picture to upload.

Figure 10. Selecting a Gallery Options

Figure 11 shows the gallery of user where they can select an image of a banana.

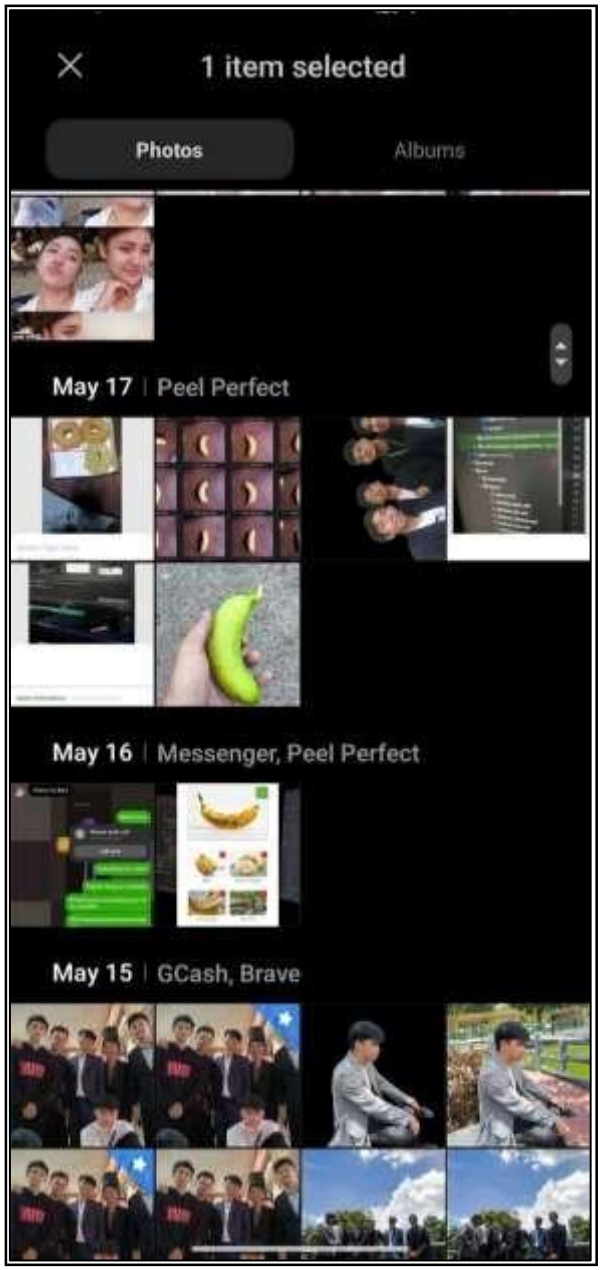


Figure 11. Gallery Selection for Banana Image

Figure 12 shows the Output page where the users can see the type and ripeness of the captured/uploaded banana.

Figure 12. Screenshot of the Output Page of the Application

Figure 13 shows the basic information of the banana. It displays the pieces of the banana, the duration until it will become ripe, unripe or overripe and the description of the captured/uploaded banana.

Figure 13. Basic Information of the Captured/Uploaded Banana

Figure 14 shows the recommendation on how the user can use the banana. It also shows extra tips on to handle the banana**.**

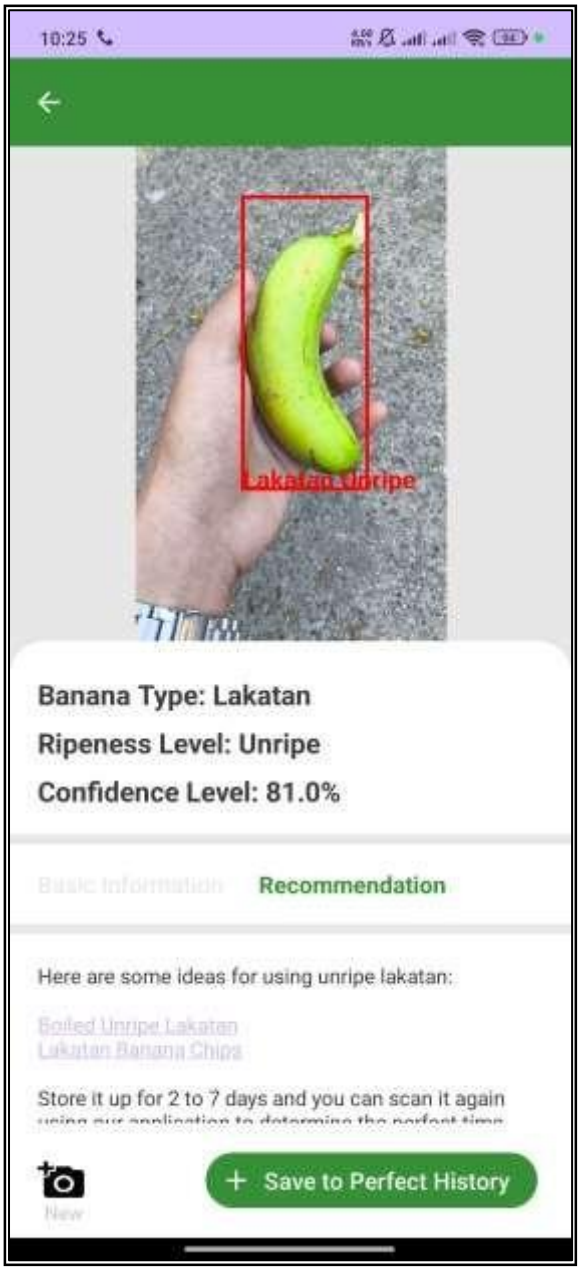


Figure 14. Banana Usage Recommendations

Figure 15 shows the history of the captured/uploaded banana. The users can view the date and time on when they have captured the banana.

Figure 15. Screenshot of the History Page of the Application

Figure 16 shows the update page of the clicked banana history card.

Figure 16. Screenshot of the Update Page of the Application

Figure 17 shows the update output page of the clicked history card.

Figure 17. Update Output Page

Figure 18 shows the delete dialog box where the users can delete the history of the saved captured/uploaded banana.

Figure 18. Delete History

Figure 19 shows that when a banana is not detected in the captured/uploaded image, a prompt reads “Banana is not detected in the image, please take another picture”. The user then can retake the photo by clicking the button.

Figure 19. Prompt for Retaking Photo when Banana is Not Detected

Figure 20 shows the About page where the users can see the overview of the application and the developers who made it.

Figure 20. About Page

Software Testing

The researchers tested the application with selected testers while still in the development stage. The system was tested using four different methods, Unit Testing, Integration Testing, System Testing and Accuracy Testing, with two (2) different versions to test and assess the system.

In Unit Testing, each module is tested to make sure that it is suited for the system (See Appendix 3). In Integration Testing, the system was tested as a combination of two connected modules (See Appendix 3). While in System Testing, the system was tested as a combination of all the modules. System testing is conducted to determine if the system meets all of the specifications and serves its purpose (See Appendix 5).

In Unit Testing version one (1), this phase includes the Data Collection module functionalities which already passed the unit testing phase. On the other hand, the Preprocessing module, Training module, User Interface and Evaluation module failed.

In Unit Testing version two (2), in this phase, all of the Preprocessing module, Training module, User Interface and Evaluation module are already completed and passed the final unit testing phase.

In Accuracy Testing, in this phase, 60 bananas were tested to determine the average accuracy of the ripeness, type, overall accuracy and number of bananas in a cluster of the application. The ripeness accuracy is 95%, the type accuracy is 85%.

Table 5. Type Accuracy Testing

TYPE ACCURACY					
OUTPUT	Saba	Senorita	Lakatan	Latundan	SUM
Saba	14 93.33%	1 6.7%	0	0	15 93.33 % 6.7 %
Senorita	4 26.67%	11 73.33%	0	0	15 26.67% 73.33%
Lakatan	0	0	14 93.33%	1 6.7%	15 93.33 % 6.7 %
Latundan	1 6.7%	1 6.7%	1 6.7%	12 80%	15 6.7% 6.7% 6.7% 80%
SUM	19 73.68 %	13 84.62 %	15 93.33 %	13 92.31%	**Overall Type Accuracy** **85%**

Table 6. Ripeness Accuracy Testing

RIPENESS ACCURACY				
OUTPUT	Unripe	Ripe	Overripe	SUM
Unripe	19 95%	1 5 %	0	20 95% 5%
Ripe	0	20 100%	0	20 100%
Overripe	0	2 10%	18 90%	18 90% 10%
SUM	19 100%	23 86.96%	18 100%	**Overall Ripeness Accuracy** 95.33%

Software Evaluation

Software evaluation was conducted and evaluated with the participation of 20 banana vendors and 30 banana consumers. There was a total of 50 respondents for non-technical evaluation and ten (10) IT professionals for technical evaluation. To determine whether the system met the required requirements (see Appendix 4 for Technical and Appendix 5 for non-Technical), 60 participants evaluated the software for both non-technical and technical based on the following criteria: functional suitability, performance efficiency, usability, reliability and portability.

The researchers provided evaluation forms to the respondents (see Appendix 4 and Appendix 5) to test and rate the system application, as shown in Table 7. Also, table interpretations are presented accordingly through legends, as shown in Table 8. After tallying the software evaluation results (see Appendix Table 7 and Appendix Table 7), the mean was computed to determine the average rating per indicator. The standard deviation was calculated to determine the dispersion of the responses. The computed mean was then verbally interpreted, as shown in Table 8.

Table 7. Software Evaluation Rating

RATING	REMARKS	INTERPRETATION
5	Excellent	The system fully meets and far exceeds the most expectations.
4	Very Good	The system fully meets and exceeds several expectations.
3	Good	The system fully meets all expectations.
2	Fair	The system does not fully meet all expectations.
1	Poor	The system fails to meet expectations to a significant degree in several areas.

Table 8. Legend of the Rating Scale of the Software Evaluation

LEGEND		
4.21 - 5.00	Excellent	The system fully meets and far exceeds the most expectations.
3.41 - 4.20	Very Good	The system fully meets and exceeds several expectations.
2.61 - 3.40	Good	The system fully meets all expectations.
1.81 - 2.60	Fair	The system does not fully meet all expectations.
1.00 - 1.80	Poor	The system fails to meet expectations to a significant degree in several areas.

Table 9-14 indicate the individual result for the non-technical system evaluation for vendors and consumers.

Meanwhile, Table 9 presents that the functional suitability of the application was “Very Good” in terms of functional completeness (M = 4.16, SD = 0.55), functional correctness (M = 4.12, SD = 0.63) and functional appropriateness (M = 3.94, SD = 0.65). It shows that every function of the software was very good and properly working.

Table 9. Consumer/Vendors Evaluation of the Application in terms of its Functional Suitability (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Completeness	4.16	0.55	Very Good
Functional Correctness	4.12	0.63	Very Good
Functional Appropriateness	3.94	0.65	Very Good
AVERAGE	4.07	0.61	Very Good

Table 10 presents that the performance efficiency of the application was “Very Good” in terms of functional time behavior (M = 4.1, SD = 0.68), resource utilization (M = 3.94, SD = 0.59) and capacity (M = 3.94, SD = 0.65). It shows that every function of the software was very good and properly working.

Table 10. Consumer/Vendors Evaluation of the Application in terms of its Performance Efficiency (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Time Behavior	4.1	0.68	Very Good
Resource Utilization	3.94	0.59	Very Good
Capacity	3.94	0.65	Very Good
AVERAGE	3.99	0.64	Very Good

Table 11 presents that the usability of the application was “Very Good” in terms of appropriateness recognizability (M = 4.06, SD = 0.71), user protection error (M = 3.86, SD = 0.70), user interface aesthetics (M = 4.2, SD = 0.83) and accessibility (M = 4.14, SD = 0.68). Also, the data shows “Excellent” in terms of learnability (M = 4.34, SD = 0.56) and operability (M = 4.24, SD = 0.72). Lastly, it explains that every function of the software was very good and properly working.

Table 11. Consumer/Vendors Evaluation of the Application in terms of its Usability (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Appropriateness Recognizability	4.06	0.71	Very Good
Learnability	4.34	0.56	Excellent
Operability	4.24	0.72	Excellent
User Error Protection	3.86	0.70	Very Good
User Interface Aesthetics	4.2	0.83	Very Good

Table 11. Continued.

Accessibility	4.14	0.57	Very Good
AVERAGE	4.14	0.68	Very Good

Table 12 presents that the reliability of the application was “Very Good” in terms of maturity (M = 4.08, SD = 0.67), fault tolerance (M = 3.72, SD = 0.67) and recoverability (M = 3.86, SD = 0.70). Meanwhile, the data reveals “Excellent in terms of availability (M = 4.24, SD = 0.69) and it shows that every function of the software was very good and properly working.

Table 12. Consumer/Vendors Evaluation of the Application in terms of its Reliability (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Maturity	4.08	0.67	Very Good
Availability	4.24	0.69	Excellent
Fault Tolerance	3.72	0.67	Very Good
Recoverability	3.86	0.70	Very Good
AVERAGE	3.98	0.68	Very Good

Table 13 presents that the portability of the application was “Very Good” in terms of adaptability (M = 3.94, SD = 0.62), installability (M = 4.1, SD = 0.74) and replaceability (M = 3.96, SD = 0.57). It shows that every function of the software was very good and properly working.

Table 13. Consumer/Vendors Evaluation of the Application in terms of its Portability (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Adaptability	3.94	0.62	Very Good
Installability	4.1	0.74	Very Good
Replaceability	3.96	0.57	Very Good
AVERAGE	4	0.64	Very Good

Table 14. Summary of the Results of the Evaluation (Consumers/Vendors) & (Non-Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Suitability	4.07	0.61	Very Good
Performance Efficiency	3.99	0.64	Very Good
Usability	4.14	0.68	Very Good
Reliability	3.98	0.68	Very Good
Portability	4	0.64	Very Good
AVERAGE	4.04	0.65	Very Good

Table 15-20. Indicate the individual result for the technical system evaluation for IT Professional.

Table 15 presents that the functional suitability of the application was “Excellent” in terms of functional completeness (M = 4.6, SD = 0.70), functional correctness (M = 4.6, SD = 0.52) and functional appropriateness (M = 4.4, SD = 0.70). It shows that every function of the software was excellent and properly working.

Table 15. IT Professional Evaluation of the Application in terms of its Functional Suitability (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Completeness	4.6	0.70	Excellent
Functional Correctness	4.6	0.52	Excellent
Functional Appropriateness	4.4	0.70	Excellent
AVERAGE	4.53	0.64	Excellent

Table 16 states that the performance efficiency of the application was “Very Good” in terms of functional time behavior (M = 4.2, SD = 0.78), resource utilization (M = 4.3, SD = 0.82) and capacity (M = 4.4, SD = 0.7). It shows that every function of the software was excellent and properly working.

Table 16**.** IT Professional Evaluation of the Application in terms of its Performance Efficiency (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Time Behavior	4.2	0.78	Excellent
Resource Utilization	4.3	0.82	Excellent
Capacity	4.4	0.70	Excellent
AVERAGE	4.3	0.77	Excellent

Table 17 presents that the usability of the application was “Very Good” in terms of appropriateness recognizability (M = 4.2, SD = 0.78). Meanwhile, the data reveals “Excellent” in terms of learnability (M = 4.5, SD = 0.53) and operability (M = 4.6, SD = 0.70), user protection error (M = 4.5, SD = 0.70), user interface aesthetics (M = 4.4, SD = 0.70) and accessibility (M = 4.4, SD = 0.70). It shows that every function of the software was very good and properly working.

Table 17. IT Professional Evaluation of the Application in terms of its Usability (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Appropriateness Recognizability	4.2	0.78	Very Good
Learnability	4.5	0.53	Excellent
Operability	4.6	0.70	Excellent
User Error Protection	4.5	0.70	Excellent
User Interface Aesthetics	4.4	0.70	Excellent
Accessibility	4.4	0.70	Excellent
AVERAGE	4.43	0.69	Excellent

Table 18 states that the reliability of the application was “Excellent” in terms of maturity (M = 4.5, SD = 0.71) and availability (M = 4.5, SD = 0.71). Meanwhile, the data reveals “Very Good” in terms of fault tolerance (M = 4.1, SD = 0.88) and recoverability (M = 4.1, SD = 0.74). It shows that every function of the software was excellent and properly working.

Table 18**.** IT Professional Evaluation of the Application in terms of its Reliability (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Maturity	4.5	0.71	Excellent
Availability	4.5	0.71	Excellent
Fault Tolerance	4.1	0.88	Very Good
Recoverability	4.1	0.74	Very Good
AVERAGE	4.3	0.76	Excellent

Table 19 states that the portability of the application was “Excellent” in terms of adaptability (M = 4.5, SD = 0.53), installability (M = 4.4, SD = 0.70) and replaceability (M = 4.6, SD = 0.52). It shows that every function of the software was very good and properly working.

Table 19. IT Professional Evaluation of the Application in terms of its Portability (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Adaptability	4.5	0.53	Excellent
Installability	4.4	0.70	Excellent
Replaceability	4.6	0.52	Excellent
AVERAGE	4.5	0.58	Excellent

Table 20. Summary of the Results of the Evaluation (IT Professional) & (Technical)

INDICATOR	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Suitability	4.53	0.64	Excellent
Performance Efficiency	4.3	0.77	Excellent
Usability	4.43	0.69	Excellent
Reliability	4.3	0.76	Excellent
Portability	4.5	0.58	Excellent
AVERAGE	4.41	0.69	Excellent

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

Summary

This research developed the system/application entitled Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System. It is an application that uses image processing methods to identify the type and ripeness of the banana. The users can capture or upload an image of a banana and the application will determine the type, ripeness and quantity of the captured/uploaded banana. The users can also save the captured/uploaded banana and can view it on the Perfect History page. The application provides duration on when the banana will become ripe or overripe and also a recommendation on how they can use the captured/uploaded banana. The software was designed with four (4) built modules which are the data collection module, preprocessing module, training module, and user interface module.

The application was done from March 2024 to May 2022 at Cavite State University - Main Campus. Related Literature and studies and significant data were acquired and collected from the Internet. The computer's specifications used in developing the system are Windows 10 operating system, 512 GB NVME SSD, 16GB ram and Ryzen 5 5600G processor. The following software applications were used: VS code for training and obtaining the model, Figma for creating the prototype of the User Interface of the system, CVAT.ai for annotating the images, and Android Studio for creating the application.

As for respondents, 20 random banana vendors, 30 random banana consumers, and 10 IT Professionals used the following criteria: functional suitability, performance efficiency, usability, reliability and portability. The overall result of the evaluation for non-technical was "Very Good", with an average mean of 4.05. On the

other hand, technical evaluation was also “Excellent”, with an average mean of 4.41, which means that the desired output was met.

Conclusion

Most consumers often rely on traditional methods to assess the ripeness level of a banana. By adapting to technology, the researchers developed a mobile application with image recognition. The development of the Peel Perfect application revolutionizes the way consumers select bananas. The Peel Perfect application has been beneficial for both consumers and retailers, simplifying the process of banana assessment through a single photo. It allows the user to easily determine the type and ripeness level of bananas. The result of the system evaluation conducted by the researchers got a total average of 4.05 for non-technical and 4.41 for technical. Therefore, the Peel Perfect application has proven beneficial to both consumers and retailers, as shown by its ability to make banana assessment easier and efficient.

Recommendations

1. The researchers put highly emphasis on enhancing the datasets by incorporating a wider range of images that display different levels of brightness and including more types and varieties of bananas with diverse shapes and sizes to broaden the versatility of the application.
2. The researchers suggest uploading multiple images of bananas and determining the ripeness level and type of the banana of the uploaded images.
3. The researchers recommend including other factors that affect the banana conditions in the market such as pests and diseases as this enhances the application's overall quality in assessing banana conditions within the market.

REFERENCES

Adebayo, S. E., Hashim, N., Abdan, K., Hanafi, M., & Mollazade, K. (2016). Prediction of quality attributes and ripeness classification of bananas using optical properties. Scientia Horticulturae, 212, 171–182. https://doi.org/10.1016/j.scienta.2016.09.045

Asaad, R. R., Ali, R. I., Ali, Z. A., & Shaaban, A. H. (2023). Image Processing with Python Libraries. Academic Journal of Nawroz University, 12(2), 410–416. https://doi.org/10.25007/ajnu.v12n2a1754

Canonizado, I.C. (2021). Input-Process-Output Model. Retrieved from. https://discover.hubpages.com/education/IPO-Model-of Research.

Chor, B., & Rubinstein, A. (2022). Image processing. In Cambridge University Press eBooks (pp. 166–186). https://doi.org/10.1017/9781108178327.014

Dhaniswara, E., Kristian, Y., & Setiawan, E. I. (2021). Detection of banana and its ripeness using residual neural network. JITE (Journal of Informatics and Telecommunication Engineering), 5(1), 188–197. https://doi.org/10.31289/jite.v5i1.4844

Garone, S. (2022, May 11). Doing it ripe: The benefits of eating bananas at every stage of ripeness. Greatist. https://greatist.com/health/banana-stages-benefits#benefits

Gueco, Dr. L., Antig, S., Aguilar, E., Juruena, M., Agoncillo, E., Artes, L., & Banzon, A. (2023). PHILIPPINE BANANA INDUSTRY ROADMAP 2021-2025. Department of Agriculture - Bureau of Agricultural Research. https://www.da.gov.ph/wp-content/uploads/2023/05/Philippine-Banana-Industry-Roadmap.pdf

How Does TensorFlow Work and Why is it Vital for AI? - Spiceworks. (2022, December27). Spiceworkshttps://www.spiceworks.com/tech/devops/articles/what-is-tensorflow/#:~:text=TensorFlow%20is%20defined%20as%20an,deep%20learning%20models%20on%20data.

Kahfi, A. H., Hasan, M., & Hasanah, R. L. (2023). Classification of banana ripeness based on color and texture characteristics. Journal of Computer Networks, Architecture and High-Performance Computing, 5(1), 10–17. https://doi.org/10.47709/cnahpc.v5i1.1985

Maduwanthi, S. D. T., & Marapana, R. (2019). Induced ripening agents and their effect on fruit quality of banana. International Journal of Food Science, 2019, 1–8. https://doi.org/10.1155/2019/2520179

Malabag, B., Santiago Jr, C., Cahapin, E., Reyes, J., & Legaspi, G. (2022). Fuzzy Logic-Based Size and Ripeness Classification of Banana using Image Processing Technique. International Journal of Emerging Technology and Advanced Engineering, 12, 11-18. https://www.researchgate.net/profile/Cereneo-Santiago-Jr/publication/364099201_Fuzzy_Logic-

Based_Size_and_Ripeness_Classification_of_Banana_using_Image_Processing_Technique/links/633aadfa9cb4fe44f3fa0c2f/Fuzzy-Logic-Based-Size-and-Ripeness-Classification-of-Banana-using-Image-Processing-Technique.pdf

Mathew, M. B., Manjunathan, G. S., Gokul, B., Ganesh, K., Sajidha, S. A., & Nisha, V. M. (2023). Banana Ripeness Identification and Classification Using Hybrid Models with RESNET-50, VGG-16 and Machine Learning Techniques. In Lecture notes in electrical engineering (pp. 259–273). https://doi.org/10.1007/978-981-99-0085-5_22

Mazen, F. M. A., & Nashat, A. A. (2019). Ripeness classification of bananas using an artificial neural network. Arabian Journal for Science and Engineering, 44(8), 6901–6910. https://doi.org/10.1007/s13369-018-03695-5

Mehendran Y., Kartheeswaran T., Kodikara N. (2022). Banana freshness identification using image processing techniques. IEEE Xplore

Mendiola, I. (2022, February 8). An overview of Philippine bananas - Food Philippines. Food Philippines. https://foodphilippines.com/story/an-overview-of-philippine-bananas/

Mohamedon M., Rahman F., Mohamad S., Khalifa O. (2021). Banana ripeness classification using computer vision-based mobile application. IEEE Xplore

Moso, I. M., Puda, S. R., Timaan, P., & Jolito, M. D. (2021). Evaluating the ripening stages of Musa acuminata × balbisiana (Saba) using 2D-image analysis. Publiscience, Vol. 4(Issue 1), 93–98.

Sabilla, I. A., Wahyuni, C. S., Fatichah, C., & Herumurti, D. (2019, March 1). Determining Banana Types and Ripeness from Image using Machine Learning Methods. IEEE Conference Publication | IEEE Xplore. Retrieved November 20, 2023, from https://ieeexplore.ieee.org/abstract/document/8834490/authors#authors

Samad, M. a. Z. A. (2022, November 14). Banana Fruit Classification using Convolutional Neural Network. https://publisher.uthm.edu.my/periodicals/index.php/eeee/article/view/8526/2851

Swaen, B. & George, T. (2022). What is a Conceptual Framework? | Tips & Examples. Retrieved from. https://www.scribbr.com/methodology/conceptual- framework/.

Tamatjita, E. N., & Sihite, R. D. (2022). Banana Ripeness Classification using HSV Colour Space and Nearest Centroid Classifier. Information Engineering Express, 8(1), 1. https://doi.org/10.52731/iee.v8.i1.687

Tanut, B., Tatomwong, W., & Buachard, S. (2023). Developing a colorimetric equation and a colorimetric model to create a smartphone application that identifies the ripening stage of lady finger bananas in Thailand. Sensors, 23(14), 6387. https://doi.org/10.3390/s23146387

Thakur R., Suryawanshi G., Patel H., & Sangoi J. (2020). An innovative approach for fruit ripeness classification. IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9121045

Thayumanavan, S., Raj, J. J. R., Kumar, P. S., Sheeba, K. N., & Uma, S. (2023b). Deep learning based intelligent identification system for ripening stages of banana. Postharvest Biology and Technology, 203, 112410. https://doi.org/10.1016/j.postharvbio.2023.112410

Tripathi, A. (2023, October 9). *What is Confusion Matrix in ML - Elements, Examples & More*. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. https://www.analytixlabs.co.in/blog/confusion-matrix/

Xiao, B., Nguyen, M., & Yan, W. Q. (2023). Fruit ripeness identification using YOLOv8 model. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023- 16570-9

Zhuang, J., Hou, C., Tang, Y., He, Y., Guo, Q., Miao, A., Zhong, Z., & Luo, S. (2019). Assessment of external properties for identifying banana fruit maturity stages using optical imaging techniques. Sensors, 19(13), 2910. https://doi.org/10.3390/s19132910

APPENDICES

APPENDIX 1

Approved Letters

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

February 5, 2024

Dear Respondents:

Greetings! I am writing to request an interview with you for my thesis project. I am currently conducting research about Banana Ripeness detection using image processing system as part of my undergraduate thesis at Cavite State University – Main Campus. This technology aims to significantly enhance the efficiency of ripeness detection processes for both retailers and consumers, potentially revolutionizing the way we select and consume bananas.

The purpose of our study is to gather insights and feedback from key stakeholders in the banana supply chain, including retailers who sell bananas and consumers who purchase them. Your participation would involve a brief interview, lasting approximately 5-10 minutes. During the interview, we will discuss your experiences and perspectives on banana ripeness detection, any challenges you currently face in this area, and your views on the potential impact of an image processing system designed for this purpose.

Your invaluable input will play a crucial role in shaping the development of our technology, ensuring it meets the real-world needs and preferences of those it is designed to benefit. Please be assured that all information shared during the interview will be treated with the utmost confidentiality and will be used solely for the purposes of this research.

I am hoping for your kind consideration regarding this matter. Thank you very much. God bless you.

Sincerely,

MERC JAN RALF P. BELANIZO

MARK ROBINSON A. ENILO

JOHN LOWELL MERCADO

CARLO A. MORGA
Researchers

Noted:

MARLON R. PERENA, Phd
Thesis Adviser

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY
Department of Information Technology

To whom it may concern:

Good day!

We are student researchers from BSCS 4-2, working on a study titled **"Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System."** The objective of this project is to develop an innovative system for detecting the ripeness and the types of bananas using advanced image processing techniques.

Specifically, the research focuses on capturing images of bananas at various stages of ripeness and applying algorithms to analyze color, shape and other visual features. The goal is to create a reliable, automated method to assess ripeness. This system aims to enhance the efficiency and accuracy of ripeness detection, reducing waste and improving the quality of bananas available to consumers.

In line with this, we would like to formally invite you to participate in a software evaluation of Peel Perfect. Your in-depth knowledge and experience in the field will provide critical insights into the strengths and weaknesses of the proposed system, and your feedback will be instrumental for enhancement end refinement of this project. The evaluation will involve a brief demonstration of the system's functionalities, followed by a discussion to gather your feedback.

We understand your time is valuable, and we greatly appreciate your willingness to participate in this evaluation. Please let us know your availability for the demonstration at your earliest convenience.

Thank you very much for considering our request.

Sincerely,

Merc Jan Rale P. Belanizo
mercjanrale.belanizo@cvsu.edu.ph

Mark Robinson A. Enilo
markrobinson.enilo@cvsu.edu.ph

John Lowell Mercado
johnlowell.mercado@cvsu.edu.ph

Carlo A. Morga
carlo.morga@cvsu.edu.ph

Noted:

Marlon R. Pereña, PhD
Thesis Adviser

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY
Department of Information Technology

To whom it may concern:

Good day!

We are student researchers from BSCS 4-2, working on a study titled "**Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System**." The objective of this project is to develop an innovative system for detecting the ripeness and the types of bananas using advanced image processing techniques.

Specifically, the research focuses on capturing images of bananas at various stages of ripeness and applying algorithms to analyze color, shape and other visual features. The goal is to create a reliable, automated method to assess ripeness. This system aims to enhance the efficiency and accuracy of ripeness detection, reducing waste and improving the quality of bananas available to consumers.

We are excited to invite you to participate in the evaluation of "Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System." By joining this evaluation, you will have the opportunity to explore the functionality, provide critical feedback, and contribute to the enhancement of a tool that has the potential to benefit consumers and industry professionals alike. Your participation will play a crucial role in ensuring the reliability and effectiveness of the Peel Perfect system, ultimately guiding us towards a successful deployment and impactful real-world application.

We understand your time is valuable, and we greatly appreciate your willingness to participate in this evaluation. Please let us know your availability for the demonstration at your earliest convenience.

Thank you very much for considering our request.

Sincerely,

Merc Jan Rale P. Belanizo
mercjanrale.belanizo@cvsu.edu.ph

Mark Robinson A. Enilo
markrobinson.enilo@cvsu.edu.ph

John Lowell Mercado
johnlowell.mercado@cvsu.edu.ph

Carlo A. Morga
carlo.morga@cvsu.edu.ph

Noted:

Marlon R. Pereña, PhD
Thesis Adviser

APPENDIX 2

Approved CEIT R&E Form

CEIT R&E Form No. 1

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

REQUEST FOR ADVISER AND TECHNICAL CRITIC

NAME OF STUDENT/S: BELANIZO, MERC JAN RALE P.
ENILO, MARK ROBINSON A.
MERCADO, JOHN LOWELL
MORGA, CARLO A.
COURSE: BS COMPUTER SCIENCE
AREA OF STUDY: INTELLIGENT SYSTEM

CONFORME:

MARLON R. PEREÑA, PhD 11/06/23 **STEPHEN ANDREI O. ROCILLO** 11/06/23
Adviser Date Technical Critic Date

RECOMMENDING APPROVAL:

ACE AMIEL E. MALICSI 11/06/23 **CHARLOTTE B. CARANDANG** 11/04/23
Unit Research Coordinator Date Department Chairperson Date

EDWIN R. ARBOLEDA 11/21/23
College Research Coordinator Date

APPROVED:

WILLIE C. BUCLATIN
Dean NOV 22 2023 Date

cc: College R&E Coordinator
Not valid without University seal

CEIT R&E Form No. 2

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

TITLE APPROVAL SHEET

NAME OF STUDENT/S: **MERC JAN RALE P. BELANIZO**
MARK ROBINSON A. ENILO
JOHN LOWELL MERCADO
CARLO A. MORGA

TITLE: **PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM**

APPROVED:

MARLON R. PERENA. PhD 11/14/23
Adviser Date

STEPHEN ANDREI O. ROCILLO 11/14/23
Technical Critic Date

ACE AMIEL E. MALICSI 11/14/2023
Unit Research Coordinator Date

CHARLOTTE B. CARANDANG 11-21-23
Department Chairperson Date

EDWIN R. ARBOLEDA 11/21/23
College Research Coordinator Date

WILLIE Q. BUCLATIN NOV 22 2023
Dean Date

MIRIAM D. BALTAZAR 11-23-2023
Director for Research Date

cc: College and Unit R&E Coordinator

CEIT RA E Form No. 3

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

CAPSULE APPROVAL SHEET

NAME: **MERC JAN RALE P. BELANIZO, MARK ROBINSON A. ENILO, JOHN LOWELL MERCADO, CARLO A. MORGA**

TITLE: **PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM**

APPROVED:

MARLON R. PEREÑA. PhD 11|14|23
Adviser Date

STEPHEN ANDREI O. ROCILLO 11|14|23
Technical Critic Date

ACE AMIEL E. MALICSI 11/14/23
Unit Research Coordinator Date

CHARLOTTE B. CARANDANG 11-21-23
Department Chairperson Date

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

Department of Information Technology

Student(s): **MERC JAN RALE P. BELANIZO, MARK ROBINSON A. ENILO, JOHN LOWELL MERCADO, CARLO A. MORGA**

Type of Study: ◉ Thesis ○ Narrative ○ EDP ○ Design Project ○ Case Study ○ Teaching Portfolio

Title of the Study: **PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM**

	DATE RECEIVED	DATE RELEASED	REMARKS
Marlon R. Pereña, Phd Thesis Adviser		6-24-24	ok
Stephen Andrei O. Rocillo Technical Critic		6-24-24	ok
Ace Amiel E. Malicsi Unit Research Coordinator	6/24/24	6/24/24	
Department OJT Coordinator			
Charlotte B. Carandang Department Chairperson		6/24/24	
REGEL L. MOZOL, MDC English Critic	6-25-24	6-27-24	FOR 2ND READING
	6-27-24	6-29-24	FORMATTING
	6-29-24	7-1-24	ACCOMPLISHED
Dr. Edwin R. Arboleda College Research Coordinator		7/1/24	
Dr. Willie C. Buclatin College Dean			
Dr. Miriam D. Baltazar Director of Research	07/01/2024	07/01/24	

Not valid without University seal

CEIT R&E Form No. 5

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

Department of Information Technology

June 7, 2024

DR. WILLIE C. BUCLATIN
Dean, CEIT
This University

Sir,

We wish to apply for an oral review of our thesis entitled **"Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System"** on June 10, 2024 (tentative).

Thank you.

Respectfully yours,

Merc Jan Rale P. Belanizo
Mark Robinson A. Enilo
John Lowell Mercado
Carlo A. Morga

RECOMMENDING APPROVAL:

MARLON R. PEREÑA, Phd 6-3-24 Date — Adviser
STEPHEN ANDREI O. ROCILLO 6-3-24 Date — Technical Critic
ACE AMIEL E. MALICSI 6/3/24 Date — Unit Research Coordinator
CHARLOTTE B. CARANDANG 6/4/24 Date — Department Chairperson
EDWIN R. ARBOLEDA 6/4/24 Date — College Research Coordinator

APPROVED:

WILLIE C. BUCLATIN 6/4/24 Date — Dean, CEIT

Not valid without University seal

CEIT R&E Form No. 6

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

Department of Information Technology

February 05, 2024

DR. WILLIE C. BUCLATIN
Dean, CEIT
This University

Subject: **RECOMMENDATION FOR OUTLINE APPROVAL**

Dear Dr. Buclatin:

The undersigned hereby recommend the approval of the outline proposal entitled **"Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System"** prepared by **Merc Jan Rale P. Belanizo, Mark Robinson A. Enilo, John Lowell Mercado, and Carlo A. Morga.**

The study has been reviewed and successfully defended on January 15, 2024. The recommendations/suggestions/corrections of the review panel have been considered and incorporated in the revisions made.

MARLON R. PEREÑA, Phd Adviser	02-05-24 Date	**STEPHEN ANDREI O. ROCILLO** Technical Critic	02-06-24 Date
HERNZ N. RAMONES Panelist	02/06/24 Date	**TROY JIMWELL R. PERMA** Panelist	02/06/24 Date
JULIE ANN C. LONTOC Panelist	02-06-24 Date	**ACE AMIEL E. MALICSI** Unit Research Coordinator	2/8/24 Date
CHARLOTTE B. CARANDANG Department Chairperson	2/8/24 Date	**EDWIN R. ARBOLEDA, DEng** College Research Coordinator	2/8/24 Date

Approved By:

WILLIE C. BUCLATIN, PhD ASEAN Engr.
Dean

Date

Not valid without University seal

CEIT R&E Form No. 7

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

Department of Information Technology

OUTLINE APPROVAL SHEET

NAME OF STUDENTS : **MERC JAN RALE P. BELANIZO**
MARK ROBINSON A. ENILO
JOHN LOWELL MERCADO
CARLO A. MORGA

TITLE : **PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM**

MARLON R. PEREÑA, Phd 01-25-24
Adviser Date

STEPHEN ANDREI O. ROCILLO 01-29-24
Technical Critic Date

ACE AMIEL E. MALICSI 2/8/24
Unit Research Coordinator Date

CHARLOTTE B. CARANDANG 2/8/24
Department Chairperson Date

EDWIN R. ARBOLEDA 2/8/24
College Research Coordinator Date

WILLIE C. BUCLATIN 02/8/24
Dean Date

MIRIAM D. BALTAZAR 2/8/24
Director for Research Date

Not valid without University seal

CEIT R&E Form No. 8

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

CERTIFICATE OF COMPLETION

June 24, 2024

To Whom It May Concern:

This is to certify that Mr. Merc Jan Rale P. Belanizo, Mr. Mark Robinson A. Enilo 2, Mr. John Lowell Mercado, and Mr. Carlo A. Morga have successfully defended their undergraduate thesis entitled Peel Perfect: Development of Banana Ripeness Detection Using Image Processing on June 20 2024 at the DIT Building.

MARLON R. PEREÑA, PhD Adviser	6/24/24 Date	**STEPHEN ANDREI O. ROCILLO** Technical Critic	6/24/24 Date
ACE AMIEL E. MALICSI Panelist	6/24/2024 Date	**EZRA MARIE F. RAMOS** Panelist	6/24/2024 Date
HERNAN J. RAMONEZ Panelist	06/24/24 Date	**ACE AMIEL E. MALICSI** Unit Research Coordinator	6/24/2024 Date
CHARLOTTE B. CARANDANG Department Chairperson	6/24/24 Date	**EDWIN R. ARBOLEDA, DEng** College Research Coordinator	7/1/24 Date

Approved by:

WILLIE C. BUCLATIN, PhD ASEAN Engr. 7/1/24
Dean Date

APPENDIX 3

Data Gathering Picture and Interview Questionnaire sample


LAKATAN
55

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

RESEARCH TITLE

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

February 6, 2024

Banana Consumers Interview Questionnaires:

1. How do you currently select bananas for purchase? What signs of ripeness do you look for? (Paano mo kasalukuyang pinipili ang mga saging na bibilhin? Anong mga palatandaan ng kahinugan ang iyong hinahanap?)
   : Madas hilaw yung mga bibibili kong mga saging, Makikita mo yun sa kulay niya tyaka sa lambot.
2. Would you prefer to buy bananas from a store that sells them with accurate ripeness? (Mas gusto mo bang bumili ng saging sa tindahan na nagbebenta ng tamang kahinugan?):
   Doon ako sa mura kasi pang business ko yung, bibili kong saging
3. How do you determine the ripeness of a banana? (Paano mo natutukoy ang kahinugan ng saging?)
   : Makikita mo yun sa kulay niya tyaka sa lambot.
4. Do you think it would be helpful to have a mobile application that can detect the ripeness of a banana? (Sa tingin mo ba ay makakatulong ang pagkakaroon ng mobile application na kayang tukuyin ang kahinugan ng saging?)
   : Mas- maganda yun kung sakali
5. Which language do you prefer in a mobile application? Filipino or English? (Anong wika ang iyong mas pipiliin sa isang mobile application? Filipino o Ingles?)
   : Tagalog nalang para alam ko yung nabobasa ko

6. How do you determine if a banana has a disease? (Paano mo natutukoy kung ang saging ay may sakit?)

: Pagbukas mo lang kyuka mo kng makikita

7. How much trust do you place in technology to accurately determine the ripeness of bananas? (Gaano kalaki ang tiwala mo sa teknolohiya para tumpak na matukoy ang kahinugan ng mga saging?)

: Hindi ko alam pero maaps dapat yan para maganda

8. Do you prefer to receive recommendations from a smartphone application when selecting bananas for purchase? (Gusto mo bang makatanggap ng mga rekomendasyon mula sa isang smartphone application kapag pipili ka ng saging na bibilhin?)

: Mas okay yun para babayaran ko nalang yung napili na saging

MIERNONE LACDAO

Name & Signature

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

RESEARCH TITLE

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

February 6, 2024

Banana Consumers Interview Questionnaires:

1. How do you currently select bananas for purchase? What signs of ripeness do you look for? (Paano mo kasalukuyang pinipili ang mga saging na bibilhin? Anong mga palatandaan ng kahinugan ang iyong hinahanap?)

   : Minsan sa kulay, tinitignan ko sa kulay, pag kulay dilaw pinipisil ko

2. Would you prefer to buy bananas from a store that sells them with accurate ripeness? (Mas gusto mo bang bumili ng saging sa tindahan na nagbebenta ng tamang kahinugan?):

   Oo

3. How do you determine the ripeness of a banana? (Paano mo natutukoy ang kahinugan ng saging?)

   : (Titignan ng Figure, kapa-kapain ko) sa kulay

4. Do you think it would be helpful to have a mobile application that can detect the ripeness of a banana? (Sa tingin mo ba ay makakatulong ang pagkakaroon ng mobile application na kayang tukuyin ang kahinugan ng saging?)

   : Di na siguro, kasi kaya ko naman malaman kung hinog na ang saging

5. Which language do you prefer in a mobile application? Filipino or English? (Anong wika ang iyong mas pipiliin sa isang mobile application? Filipino o Ingles?)

   :

6. How do you determine if a banana has a disease? (Paano mo natutukoy kung ang saging ay may sakit?)

: Hindi pa

7. How much trust do you place in technology to accurately determine the ripeness of bananas? (Gaano kalaki ang tiwala mo sa teknolohiya para tumpak na matukoy ang kahinugan ng mga saging?)

: okay lang

8. Do you prefer to receive recommendations from a smartphone application when selecting bananas for purchase? (Gusto mo bang makatanggap ng mga rekomendasyon mula sa isang smartphone application kapag pipili ka ng saging na bibilhin?)

: mas okay rana

JOCELYN D. LASPRILLAS

Name & Signature

Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

RESEARCH TITLE

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

February 6, 2024

Banana Consumers Interview Questionnaires:

1. How do you currently select bananas for purchase? What signs of ripeness do you look for? (Paano mo kasalukuyang pinipili ang mga saging na bibilhin? Anong mga palatandaan ng kahinugan ang iyong hinahanap?)

   : sa kulay, kung dilaw na, hinog na yon.

2. Would you prefer to buy bananas from a store that sells them with accurate ripeness? (Mas gusto mo bang bumili ng saging sa tindahan na nagbebenta ng tamang kahinugan?):

   syempre, mas okay na tama yung binebenta nila kung hinog na ba.

3. How do you determine the ripeness of a banana? (Paano mo natutukoy ang kahinugan ng saging?)

   : El kulay

4. Do you think it would be helpful to have a mobile application that can detect the ripeness of a banana? (Sa tingin mo ba ay makakatulong ang pagkakaroon ng mobile application na kayang tukuyin ang kahinugan ng saging?)

   : Ah oo makakatulong sya para alam mo na agad kung hinog na

5. Which language do you prefer in a mobile application? Filipino or English? (Anong wika ang iyong mas pipiliin sa isang mobile application? Filipino o Ingles?)

   : Tagalog lang, para malalaman

6. How do you determine if a banana has a disease? (Paano mo natutukoy kung ang saging ay may sakit?)

: wala pa kong naencounter sa buong e

7. How much trust do you place in technology to accurately determine the ripeness of bananas? (Gaano kalaki ang tiwala mo sa teknolohiya para tumpak na matukoy ang kahinugan ng mga saging?)

: sa ngayon kase meron ng application na kung ano ano - kaya tingin ko kaya nanaman mga yon

8. Do you prefer to receive recommendations from a smartphone application when selecting bananas for purchase? (Gusto mo bang makatanggap ng mga rekomendasyon mula sa isang smartphone application kapag pipili ka ng saging na bibilhin?)

: di, mas okay

Ernesto B. Borga Jr.

Name & Signature

APPENDIX 4

Unit Testing

Unit Testing

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

Date: March 15, 2024
Proponents: Belanizo, Merc Jan Rale P., Enilo, Mark Robinson A., Mercado, John Lowell, Morga, Carlo A.
Instructions: Assess the following

Module: Data Collection Module

	Description	Remarks
1. Collection of Saba banana in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Saba for training	**Passed**
2. Collection of Lakatan banana in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Lakatan for training	**Passed**
3. Collection of Latundan banana in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Latundan for training	**Passed**
4. Collection of Señorita banana in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Senorita for training	**Passed**

Tester: Merc Jan Rale P. Belanizo

Date: March 15, 2024
Proponents:
Instructions:

Module: Preprocessing Module

	Description	Remarks
1. Resizing the images	Standardized the sizes of all the images	**Passed**
2. Augmenting the images	Flipping, rotating the images	**Failed**
3. Annotating the images	Selecting the banana on the images	**Failed**

Tester: Merc Jan Rale P. Belanizo

Date: March 15, 2024
Proponents:
Instructions:

Module: Training Module

	Description	Remarks
1. Train the banana for classifying the ripeness	Training the dataset for identifying the ripeness of the banana	**Passed**
2. Train the banana for classifying the type	Training the dataset for identifying the type of the banana	**Failed**
3. Train the banana for counting the quantity of banana in a cluster	Training the dataset for identifying the count of the banana	**Failed**

Tester: Merc Jan Rale P. Belanizo

Date: March 15, 2024
Proponents:
Instructions:

Module: User Interface Module

	Description	Remarks
1. Camera UI	Location of the capture tips button, capture button, upload button, flash and rotate camera button	**Passed**
2. Capture Button	Used to take a picture of the banana	**Passed**
3. Upload Button	Used to upload an image of the banana	**Passed**
4. Save History	Used to save the captured banana	**Failed**
5. Delete Button	Used to delete the saved captured banana	**Failed**

Tester: Merc Jan Rale P. Belanizo

Date: March 15, 2024
Proponents:
Instructions:

Module: Evaluation Module

	Description	Remarks
1. Evaluate banana basic info	Assess the info of banana based on the identified type and ripeness	**Failed**
2. Evaluate the recommendation	Identify the right recommendation for the	**Failed**

	banana	

Tester: Merc Jan Rale P. Belanizo

Unit Testing - version two

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

Date: May 18, 2024
Proponents:
Instructions:

Module: Data Collection Module

	Description	Remarks
1. Collection of Saba banana in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Saba for training	**Passed**
2. Collection of Lakatan bananas in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Lakatan for training	**Passed**
3. Collection of Latundan bananas in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Latundan for training	**Passed**
4. Collection of Señorita bananas in different ripeness levels (Unripe, Ripe and Overripe).	Gathering Senorita for training	**Passed**

Tester: Merc Jan Rale P. Belanizo

Date: May 18, 2024
Proponents:
Instructions:

Module: Preprocessing Module

	Description	Remarks
1. Resizing the images	Standardized the sizes of all the images	**Passed**
2. Augmenting the images	Flipping, rotating the images	**Passed**
3. Annotating the images	Selecting the banana on the images	**Passed**

Tester: Merc Jan Rale P. Belanizo

Date: May 18, 2024

Proponents:
Instructions:

Module: Training Module

	Description	Remarks
1. Train the banana for classifying the ripeness	Training the dataset for identifying the ripeness of the banana	**Passed**
2. Train the banana for classifying the type	Training the dataset for identifying the type of the banana	**Passed**
3. Train the banana for counting the quantity of banana in a cluster	Training the dataset for identifying the count of the banana	**Passed**

Tester: Merc Jan Rale P. Belanizo

Date: May 18, 2024
Proponents:
Instructions:

Module: User Interface Module

	Description	Remarks
1. Camera UI	Location of the capture tips button, capture button, upload button, flash and rotate camera button	**Passed**
2. Capture Button	Used to take a picture of the banana	**Passed**
3. Upload Button	Used to upload an image of the banana	**Passed**
4. Save History	Used to save the captured banana	**Passed**
5. Delete Button	Used to delete the saved captured banana	**Passed**

Tester: Merc Jan Rale P. Belanizo

Date: March 15, 2024
Proponents:
Instructions:

Module: Evaluation Module

	Description	Remarks
1. Evaluate banana basic info	Assess the info of banana based on the identified type and ripeness	**Failed**
2. Evaluate the recommendation	Identify the right recommendation for the banana	**Failed**

Tester: Merc Jan Rale P. Belanizo

APPENDIX 5

INTEGRATION TESTING

Integration Testing
Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

INTEGRATION TESTING		
Modules:	**Integrate Data Collection Module and Preprocessing Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024

Objective: Ensure that the data collected by the Data collection module are correctly preprocessed by the Preprocessing module

Test Case # 1	**Test Case Name:**	**Integrate Data Collection Module and Preprocessing Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**
Verify if the collected images are correctly handed to the Preprocessing Module.	The images are successfully handed to the preprocessing module.	The images are successfully handed to the preprocessing module	**Passed**
Check if the Preprocessing processes are correctly applied to the images	The images are successfully preprocessed.	The images are successfully preprocessed.	**Passed**

INTEGRATION TESTING		
Modules:	**Integrate Preprocessing Module and Training Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024

Objective: Ensure that the Preprocessed data are correctly used by the Training Module.

Test Case # 2	**Test Case Name:**	**Integrate Preprocessing Module and Training Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**
Verify if the Preprocessed images are correctly handed into the machine learning module.	The preprocessed images are successfully handed to the Machine Training module.	The preprocessed images fail to be handed to the Machine Training module.	**Failed**
Ensure if the Model can learn from the extracted data without errors.	The model extracted the data successfully.	The model extracted the data unsuccessfully.	**Failed**

INTEGRATION TESTING		
Modules:	**Integrate Training Module with User Interface Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 03/21/2024

Objective: Ensure that the trained model can be accessed and utilized through the User Interface.

Test Case # 3	**Test Case Name:**	**Integrate Training Module with User Interface Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**

Check if the trained model can be loaded and used by the UI.	The User Interface successfully integrates the Model.	The integration of the trained model with the User Interface was unsuccessful.	**Failed**
Ensure UI can capture images, then process them into the model and display the type of banana and ripeness level.	The User Interface successfully captures the images and displays the type and ripeness.	The User Interface failed to capture images and display the corresponding banana type and ripeness.	**Failed**
Ensure UI can upload images, then process them into the model and display the type of banana and ripeness level.	The User Interface successfully uploads the images and displays the type and ripeness.	The User Interface has failed to upload the images and display the type and ripeness.	**Failed**

Integration Testing second version

Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

INTEGRATION TESTING		
Modules:	**Integrate Data Collection Module and Preprocessing Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024

Objective: Ensure that the data collected by the Data collection module are correctly preprocessed by the Preprocessing module

Test Case # 1	**Test Case Name:**	**Integrate Data Collection Module and Preprocessing Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**
Verify if the collected images are correctly handed to the Preprocessing Module.	The images are successfully handed to the preprocessing module.	The images are successfully handed to the preprocessing module.	**Passed**
Check if the Preprocessing processes are correctly applied to the images	The images are successfully preprocessed.	The images are successfully preprocessed.	**Passed**

INTEGRATION TESTING		
Modules:	**Integrate Preprocessing Module and Training Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024

Objective: Ensure that the Preprocessed data are correctly used by the Training Module.

Test Case # 2	**Test Case Name:**	**Integrate Preprocessing Module and Training Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**
Verify if the Preprocessed images are correctly handed into the machine learning module.	The preprocessed images are successfully handed to the Machine Training module.	The preprocessed images are successfully handed to the Machine Training module.	**Passed**
Ensure if the Model can learn from the extracted data without errors	The model extracted the data successfully.	The model extracted the data successfully.	**Passed**

INTEGRATION TESTING		
Modules:	**Integrate Training Module with User Interface Module**	
Designed by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024
Tested by:	Merc Jan Rale P. Belanizo	Date: 05/18/2024

Objective: Ensure that the trained model can be accessed and utilized through the User Interface.

Test Case # 3	**Test Case Name:**	**Integrate Training Module with User Interface Module**	
Test action	**Expected Result**	**Actual Result**	**Remarks**

Check if the trained model can be loaded and used by the UI.	The User Interface successfully integrates the Model.	The User Interface successfully integrates the Model.	**Passed**
Ensure UI can capture images, then process them into the model and display the type of banana and ripeness level.	The User Interface successfully captures the images and displays the type and ripeness.	The User Interface successfully captures the images and displays the type and ripeness.	**Passed**
Ensure UI can upload images, then process them into the model and display the type of banana and ripeness level.	The User Interface successfully uploads the images and displays the type and ripeness.	The User Interface successfully uploads the images and displays the type and ripeness.	**Passed**

APPENDIX 6

SYSTEM TESTING

SYSTEM TESTING
Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

Date: April 11, 2024

Test action	Expected Result	Actual Result	Remarks
Capture Banana	• Successfully captured clear banana image	• Successfully captured clear banana image	**Passed**
Upload Banana	• Successfully uploaded clear banana image	• Successfully captured clear banana image	**Passed**
Identify Banana Ripeness	• User can view the ripeness level of the captured/uploaded banana	• Incorrect ripeness level	**Failed**
Identify Banana Type	• User can view the type of the captured/uploaded banana	• Incorrect type of the banana	**Failed**
Identify Banana Count	• User can view the quantity of the captured/uploaded banana	• Incorrect count of the banana	**Failed**
Save to History	• User can save and view the captured/uploaded banana	• Cannot save to the history page	**Failed**
Delete History	• User can delete the saved captured/uploaded banana	• Cannot delete the saved banana	**Failed**

SYSTEM TESTING VERSION TWO
Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System

Date: May 18, 2024

Test action	Expected Result	Actual Result	Remarks
Capture Banana	• Successfully captured clear banana image	• Successfully captured clear banana image	**Passed**
Upload Banana	• Successfully uploaded clear banana image	• Successfully captured clear banana image	**Passed**
Identify Banana Ripeness	• User can view the ripeness level of the captured/uploaded banana	• User can view the ripeness level of the captured/uploaded banana	**Passed**
Identify Banana Type	• User can view the type of the captured/uploaded banana	• User can view the type of the captured/uploaded banana	**Passed**
Identify Banana Count	• User can view the quantity of the captured/uploaded banana	• User can view the quantity of the captured/uploaded banana	**Passed**
Save to History	• User can save and view the captured/uploaded banana	• User can save and view the captured/uploaded banana	**Passed**
Delete History	• User can delete the saved captured/uploaded banana	• User can delete the saved captured/uploaded banana	**Passed**

APPENDIX 7

Sample Technical Evaluation Form

PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM

RESEARCHERS: MERC JAN RALE P. BELANIZO, MARK ROBINSON A. ENILO, JOHN LOWELL MERCADO, CARLO A. MORGA

Name(Optional): _____ Age: _____
Sex: ____Male____Female Email: _____
Affiliated Institution: _____
Position/Designation: _____

Directions: Please evaluate the software quality of the "**Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System**" based on the main characteristics of ISO 25010 Software Product Quality Standards. Choose the appropriate box using the Likert scale with corresponding verbal descriptions (5 as the highest and 1 as the lowest).

5 – (Excellent) 4 – (Very Good) 3 – (Good) 2 – (Fair) 1 – (Poor)

A. Functional Suitability	**5**	**4**	**3**	**2**	**1**
1. **Functional completeness** - Does the system cover all the tasks and goals users want to achieve?					
2. **Functional correctness** - How accurately does the system deliver the right results with the required level of precision?					
3. **Functional appropriateness** - How well do the functions help users complete their tasks and achieve their goals?					
B. Performance Efficiency					
1. **Time behavior -** How well the system meets its speed and capacity needs when doing its job?					
2. **Resource utilization -** How well a system's use of resources matches what's needed when it's doing its job.					
3. **Capacity -** How well does the system stay within its limits as needed?					
C. Usability					
1. **Appropriateness recognizability** - How appropriate is the product based on the user's need?					
2. **Learnability** - How well a system helps users learn to use the product effectively, efficiently, safely, and in a specific situation?					

3. **Operability** - Can the user navigate and control the system easily?					
4. **User error protection** - Does the system prevent users from making mistakes?					
5. **User interface aesthetics** - How much a user interface makes interactions enjoyable and satisfying for the user.					
6. **Accessibility** - How well a system works for everyone to achieve a specific goal in a specific situation?					
D. Reliability					
1. **Maturity** - Can the system meet reliability needs during regular use?					
2. **Availability** - How well a system or part is available and usable when needed?					
3. **Fault tolerance** - How well a system keeps working correctly even if there are hardware or software problems?					
4. **Recoverability** - How well a system can recover data and get back to normal after an interruption or failure?					
E. Portability					
1.**Adaptability** - How well the system can be efficiently adjusted for various software and evolving smartphones?					
2.**Installability** - How effectively and efficiently the system can be installed and/or uninstalled?					
3.**Replaceability** - Can the system substitute another specified software product for the same purpose in the same environment?					

Comment and Suggestion:

_____
_____
_____
_____

Respondent's Signature: _____

Date of Evaluation: _____

APPENDIX 8

Sample Non-Technical Evaluation Form (Vendors and Consumers)

PEEL PERFECT: DEVELOPMENT OF BANANA RIPENESS DETECTION USING IMAGE PROCESSING SYSTEM

RESEARCHERS: MERC JAN RALE P. BELANIZO, MARK ROBINSON A. ENILO, JOHN LOWELL MERCADO, CARLO A. MORGA

Name (Optional): _____ Age: _____

Sex: ___ Male___Female Vendor: ☐ Consumer: ☐

Panuto:: Maari mo bang suriin the ang kalidad ng "**Peel Perfect: Development of Banana Ripeness Detection Using Image Processing System**". Pumili ng naaayon kahon gamit ang Likert Scale (5 ang pinakamataas at 1 naman ang pinakamababa).

5 – (Excellent) 4 – (Very Good) 3 – (Good) 2 – (Fair) 1 – (Poor)

A. Functional Suitability	**5**	**4**	**3**	**2**	**1**
1. **Functional completeness** - Nagagawa ba ng app lahat ng mga gawain at layunin na nais ng mga gumagamit na makamit?					
2. **Functional correctness** - Gaano katumpak ang naibibigay na resulta ng app base sa kinakailangan antas?					
3. **Functional appropriateness** - Gaano kahusay ang mga function sa pagtulong sa mga gumagamit para makamit ang layunin nito?					
B. Performance Efficiency					
1. **Time behavior -** Gaano kabilis at kalakas ang app sa paggawa ng trabaho nito?					
2. **Resource utilization -** Nagagamit ba ng app ng ayos ang resources nito habang ginagawa ang trabaho?					
3. **Capacity -** Gaanong kahusay ang app sa pagpapanatili sa loob ng kanyang mga limitasyon kapag kinakailangan??					
D. Usability					
1. **Appropriateness recognizability** - Naayon ba ang produkto ng app sa pangangailangan ng gumagamit?					
2. **Learnability** - Madali bang natututo ang user sa paggamit ng app upang magamit ng epektibo, mabisa at sa ligtas na pamamaraan?					

3. **Operability** - Madali bang nakokontrol at nagagamit ng user ang app?					
4. **User error protection** - Gaano kahusay napipigilan ng app na magkamali ang gagamit nito?					
5. **User interface aesthetics** - Gaano kahusay ang pagkakadisenyo ng app?					
6. **Accessibility** - Gaano kahusay gumagana ang app para makamit ang tiyak na layunin at tiyak na sitwasyon?					
E. Reliability					
1. **Maturity** - Natutugunan ba ng app ang mga pangangailangan sa regular na paggamit?					
2. **Availability** - Ang app ba ay madali lang mabuksan at magamit kapag kinakailangan?					
3. **Fault tolerance** -Patuloy ba na gumagana ang app kahit nagkaroon ng problema sa device?					
4. **Recoverability** - Gaano kahusay ang application na patuloy na gumagana nang tama kung mayroong problema sa hardware o software?					
H. Portability					
1.**Adaptability** - Gaano kaayos ang app sa pag adapt sa ibat ibang selpon?					
2.**Installability** - Gaano kaayos o kadali ang application na i-install sa iyong selpon?					
3.**Replaceability** - Ang application ba ay maaring ipalit sa iba pang application para sa parehas na gamit nito?					

Maari kang magbigay ng iyong komento o suwestyon:
_____
_____
_____

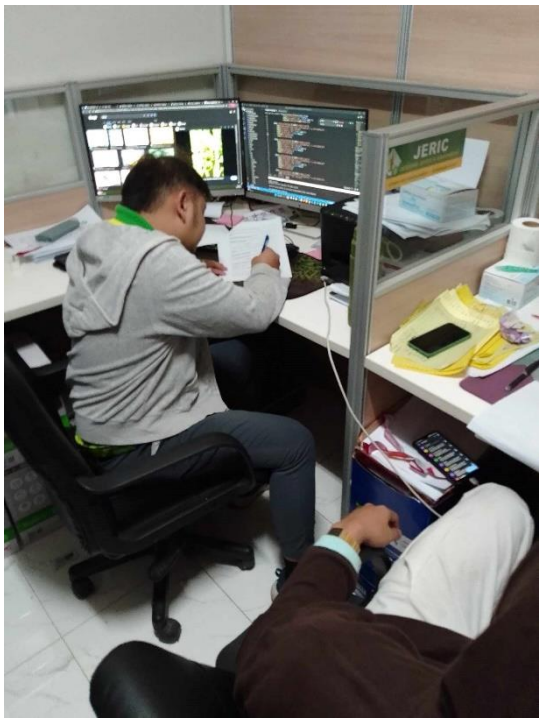
Respondent's Signature: _____

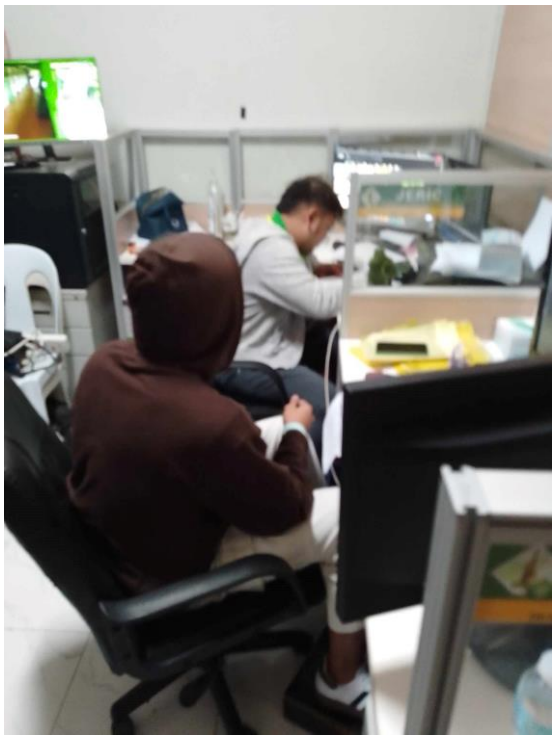
Date of Evaluation: _____

APPENDIX 9

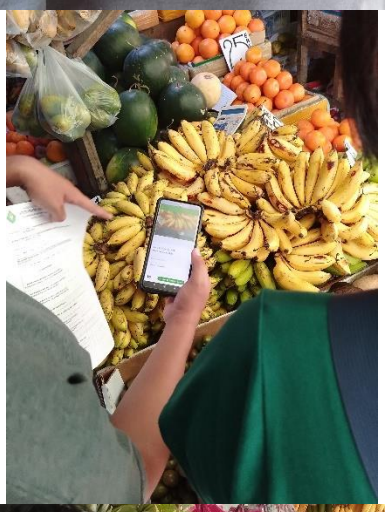
Evaluation Picture Documentation of Technical and Non-Technical Evaluator

Technical Evaluator



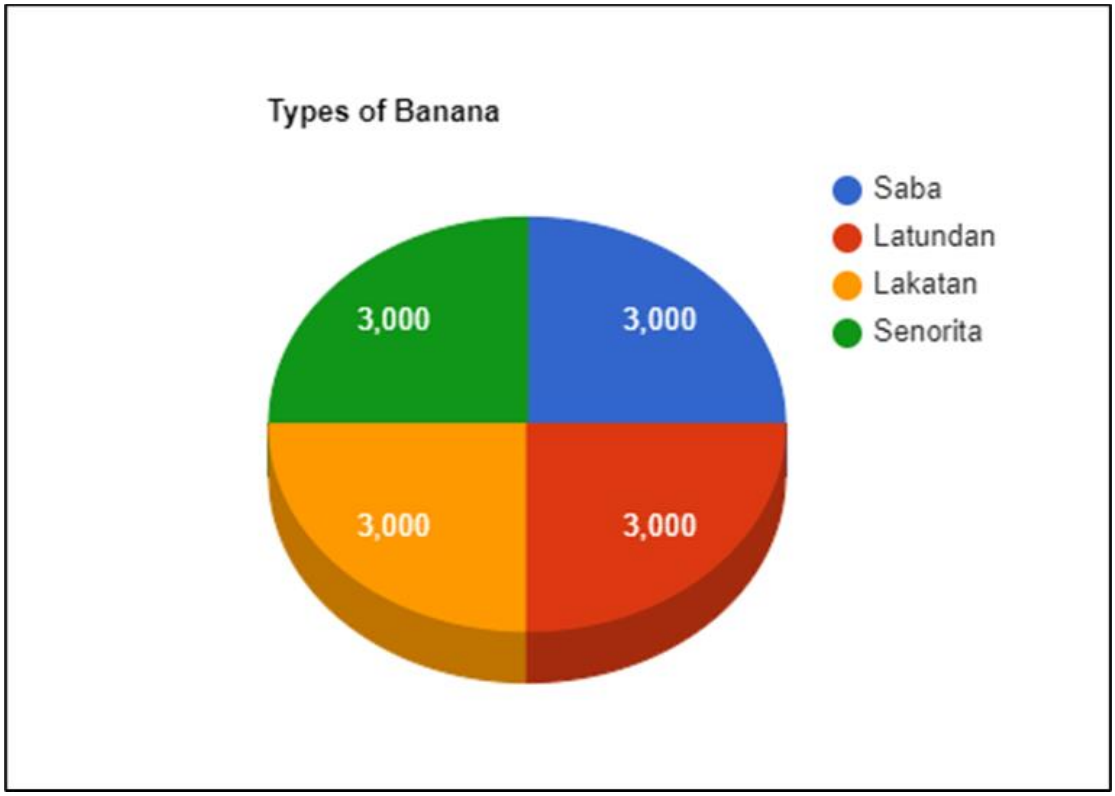


Non-Technical Evaluator

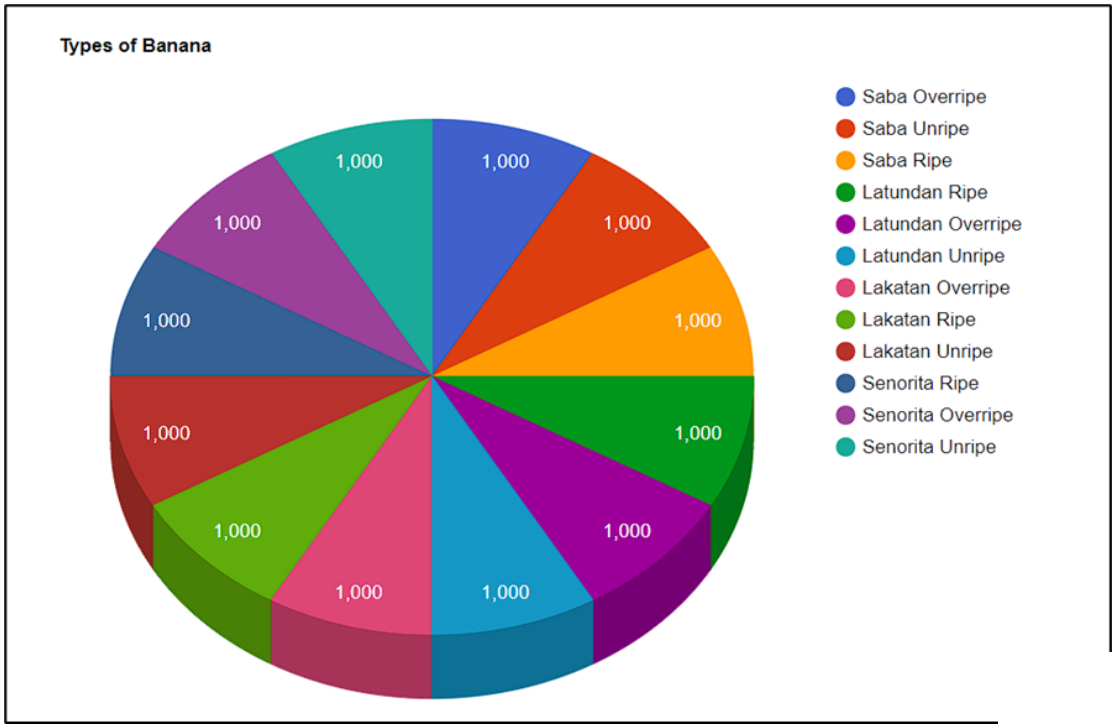


APPENDIX 10

Dataset Distribution Charts



Composition of Banana Types in the Collected Dataset



Composition of Ripeness Levels for Each Banana Type

APPENDIX TABLE

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	12	24%	34	68%	4	8%	0	0	0	0	50	100%
2	13	26%	30	60%	7	14%	0	0	0	0	50	100%
3	9	18%	29	58%	12	24%	0	0	0	0	50	100%

Appendix Table 1. Frequency distribution of scores of the Functional Suitability indicators (non-technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	14	28%	27	54%	9	18%	0	0	0	0	50	100%
2	7	14%	33	66%	10	20%	0	0	0	0	50	100%
3	9	18%	29	58%	12	24%	0	0	0	0	50	100%

Appendix Table 2. Frequency distribution of scores of the Performance Efficiency indicators (non-technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	14	28%	25	50%	11	22%	0	0	0	0	50	100%
2	19	38%	29	58%	2	4%	0	0	0	0	50	100%
3	20	40%	22	44%	8	16%	0	0	0	0	50	100%
4	9	18%	25	50%	16	32%	0	0	0	0	50	100%
5	23	46%	14	28%	13	26%	0	0	0	0	50	100%

6	**12**	**24%**	**33**	**66%**	**5**	**10%**	**0**	**0**	**0**	**0**	**50**	**100%**

Appendix Table 3. Frequency distribution of scores of the Usability indicators (non-technical)

STATEMENT	**5**		**4**		**3**		**2**		**1**		**TOTAL**	
	f	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**F**	**%**
1	**13**	**26%**	**28**	**56%**	**9**	**18%**	**0**	**0**	**0**	**0**	**50**	**100%**
2	**19**	**38%**	**24**	**40%**	**7**	**14%**	**0**	**0**	**0**	**0**	**50**	**100%**
3	**6**	**12%**	**24**	**48%**	**20**	**4%**	**0**	**0**	**0**	**0**	**50**	**100%**
4	**9**	**18%**	**25**	**50%**	**16**	**32%**	**0**	**0**	**0**	**0**	**50**	**100%**

Appendix Table 4. Frequency distribution of scores of the Reliability indicators (non-technical)

STATEMENT	**5**		**4**		**3**		**2**		**1**		**TOTAL**	
	f	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**F**	**%**
1	**8**	**16%**	**31**	**62%**	**11**	**22%**	**0**	**0**	**0**	**0**	**50**	**100%**
2	**16**	**32%**	**23**	**46%**	**11**	**22%**	**0**	**0**	**0**	**0**	**50**	**100%**
3	**7**	**14%**	**34**	**68%**	**9**	**18%**	**0**	**0**	**0**	**0**	**50**	**100%**

Appendix Table 5. Frequency distribution of scores of the Portability indicators (non-technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	7	70%	2	20%	1	10%	0	0	0	0	10	100%
2	6	60%	4	40%	0	0	0	0	0	0	10	100%
3	5	50%	4	40%	1	10%	0	0	0	0	10	100%

Appendix Table 6. Frequency distribution of scores of the Functional Suitability indicators (technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	4	40%	4	40%	2	20%	0	0	0	0	10	100%
2	5	50%	3	30%	2	20%	0	0	0	0	10	100%
3	5	50%	4	40%	1	10%	0	0	0	0	10	100%

Appendix Table 7. Frequency distribution of scores of the Performance Efficiency indicators (technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	4	40%	4	40%	2	20%	0	0	0	0	10	100%
2	5	50%	5	50%	0	0	0	0	0	0	10	100%
3	7	70%	2	20%	1	10%	0	0	0	0	10	100%
4	6	60%	3	30%	1	10%	0	0	0	0	10	100%
5	5	50%	4	40%	1	10%	0	0	0	0	10	100%
6	5	50%	4	40%	1	10%	0	0	0	0	10	100%

Appendix Table 8. Frequency distribution of scores of the Usability indicators

(technical)

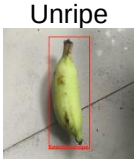
STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	6	60%	3	30%	1	10%	0	0	0	0	10	100%
2	6	60%	3	30%	1	10%	0	0	0	0	10	100%
3	4	40%	3	30%	3	30%	0	0	0	0	10	100%
4	3	30%	5	50%	2	20%	0	0	0	0	10	100%

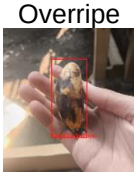
Appendix Table 9. Frequency distribution of scores of the Reliability indicators (technical)

STATEMENT	5		4		3		2		1		TOTAL	
	f	%	f	%	f	%	f	%	f	%	F	%
1	5	50%	5	50%	0	0	0	0	0	0	10	100%
2	5	50%	4	40%	1	10%	0	0	0	0	10	100%
3	6	60%	4	40%	0	0	0	0	0	0	10	100%

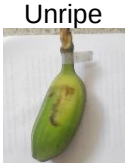
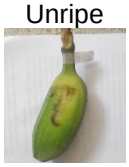
Appendix Table 10. Frequency distribution of scores of the Portability indicators (technical)

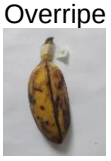
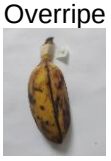
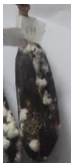
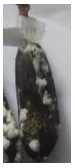
The criteria for determining ripeness levels were as follows: Unripe - green; Ripe - yellow with a few dark spots; Overripe - yellow with many dark spots; and Rotten -excessively dark or black and have mold

Type and Number	Ripeness	Day 0	Day 1	Day 2	Day 3	Day 4	DURATION
Senorita 1	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	Approximately 2 days to ripe
Senorita 2	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Senorita 3	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Senorita 4	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Senorita 5	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Senorita 6	Ripe 	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Approximately 2-4 days to overripe
Senorita 7	Ripe 	Ripe 	Ripe 	Overripe 	Overripe	Overripe	
Senorita 8	Ripe	Ripe	Ripe	Ripe	Ripe	Overripe	
Senorita	Ripe	Ripe	Ripe	Overripe	Overripe	Overripe	

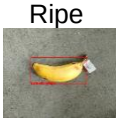
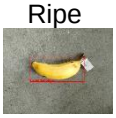
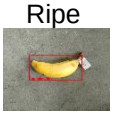
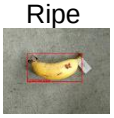
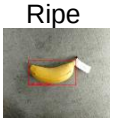
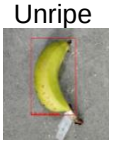
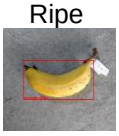
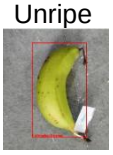
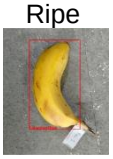
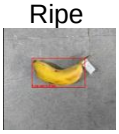
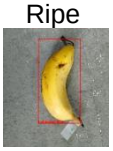
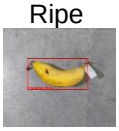
9							
Senorita 10	Ripe 	Ripe 	Ripe 	Overripe 	Overripe 	Overripe 	
Senorita 11	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	Approximately 3-4 days to rot
Senorita 12	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Senorita 13	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	
Senorita 14	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Senorita 15	Overripe 	Overripe 	Overripe 	Overripe 	Overripe	Rotten	

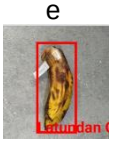
Appendix Table 11. Senorita Duration Table

Type and Number	Ripeness	Day 0	Day 1	Day 2	Day 3	Day 4	DURATION
Saba 1	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	Approximately 2 days to ripe
Saba 2	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Saba 3	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Saba 4	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Saba 5	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Saba 6	Ripe 	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Approximately 2-4 days to overripe
Saba 7	Ripe 	Ripe 	Ripe 	Overripe 	Overripe	Overripe	
Saba 8	Ripe	Ripe	Ripe	Ripe	Ripe	Overripe	
Saba 9	Ripe	Ripe	Ripe	Overripe	Overripe	Overripe	

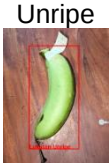
							
Saba 10	Ripe 	Ripe 	Ripe 	Overripe 	Overripe 	Overripe 	
Saba 11	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	Approximately 3 days to rot
Saba 12	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Saba 13	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Saba 14	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Saba 15	Overripe 	Overripe 	Overripe 	Overripe 	Rotten	Rotten	

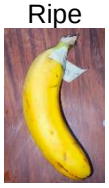
Appendix Table 12. Saba Duration Table

Type and Number	Ripeness	Day 0	Day 1	Day 2	Day 3	Day 4	DURATION
Latundan 1	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	Approximately 1-2 days to ripe
Latundan 2	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Latundan 3	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	Ripe 	
Latundan 4	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Latundan 5	Unripe 	Unripe 	Unripe 	Ripe 	Ripe 	Ripe 	
Latundan 6	Ripe 	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Approximately 2-4 days to overripe
Latundan 7	Ripe 	Ripe 	Ripe 	Ripe 	Ripe	Overripe	
Latundan 8	Ripe	Ripe	Ripe	Ripe	Ripe	Overripe	
Latundan 9	Ripe	Ripe	Ripe	Ripe	Ripe	Overripe	
Latundan 10	Ripe	Ripe	Ripe	Overripe	Overripe	Overripe	

Latundan 11	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	Approximately 3-4 days to rot
Latundan 12	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	
Latundan 13	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	
Latundan 14	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	
Latundan 15	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	Rotten 	

Appendix Table 13. Latundan Duration Table

Type and Number	Ripeness	Day 0	Day 1	Day 2	Day 3	Day 4	DURATION
Lakatan 1	Unripe 	Unripe 	Unripe 	Unripe 	Unripe 	Ripe 	Approximately 4 days to ripe
Lakatan 2	Unripe 	Unripe 	Unripe 	Unripe 	Unripe	Ripe	
Lakatan 3	Unripe	Unripe	Ripe	Ripe	Ripe	Ripe	
Lakatan 4	Unripe	Unripe	Unripe	Unripe	Unripe	Ripe	

							
Lakatan 5	Unripe 	Unripe 	Unripe 	Unripe 	Unripe 	Ripe 	
Lakatan 6	Ripe 	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Approximately 3-4 days to overripe
Lakatan 7	Ripe 	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	
Lakatan 8	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Overripe 	
Lakatan 9	Ripe 	Ripe 	Ripe 	Ripe 	Overripe 	Overripe 	
Lakatan 10	Ripe 	Ripe 	Ripe 	Ripe 	Ripe	Overripe	
Lakatan 11	Overripe	Overripe	Overripe	Overripe	Overripe	Rotten	Approximately 4 days to rot
Lakatan 12	Overripe	Overripe	Overripe	Overripe	Overripe	Rotten	

Lakatan 13	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	
Lakatan 14	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	
Lakatan 15	Overripe 	Overripe 	Overripe 	Overripe 	Overripe 	Rotten 	

Appendix Table 14. Lakatan Duration Table

Type and Ripeness	Recommended Food
Senorita (Unripe)	• Boiled Green Senorita Banana • Baked Green Senorita Banana Reference: https://www.treehugger.com/ways-use-green-bananas-wont-ripen-4864266
Senorita (Ripe)	• Senorita Banana Fritters • Senorita Banana Loaf Reference: https://www.cookalmostanything.com/2010/02/senorita-banana-fritters.html https://www.youtube.com/watch?v=qRDKwNvyXVE
Senorita (Overripe)	• Overripe Senorita Smoothie Reference: https://foodhero.org/healthy-food/beyond-peel-creative-ways-use-overripe-bananas#:~:text=They're%20fun%20to%20make,extra%20fruit%20to%20your%20breakfast!&text=Overripe%20bananas%20add%20sweetness%20and,recipe%20that%20calls%20for%20bananas.
Saba (Unripe)	• Pinakro • Binangkal • Banana Chips Reference: https://www.youtube.com/watch?v=bx_ZRQuxoUw https://www.youtube.com/watch?v=HOTkKsDW7so
Saba (Ripe)	• Bananacue • Turon • Maruya • Caramelized Banana Reference: https://www.youtube.com/watch?v=Mqk2MiJHqno&pp=ygUdYmFuYW5hY3VlIHR1cm9uIG1hcnV5YSByZWNpcGU%3D https://www.youtube.com/watch?v=Fs3gpHsnaKQ
Saba (Overripe)	• Overripe Saba Smoothie • Overripe Saba Pancake Reference: https://www.youtube.com/watch?v=bcuHBR-CtJM https://www.youtube.com/watch?v=ohO1IHUGkG4
Lakatan (Unripe)	• Lakatan Banana Chips • Boiled Unripe Lakatan Reference: https://www.youtube.com/watch?v=MWx-OiUOCsE https://www.youtube.com/watch?v=52QZO11oMww
Lakatan (Ripe)	• Lakatan Banana Pancake • Lakatan Banana Balls Reference: https://www.youtube.com/watch?v=XXv50RTK2SM https://www.youtube.com/watch?v=2MAZq5QxHgk
Lakatan (Overripe)	• Banana Flan • Sinaging Reference: https://www.youtube.com/watch?v=f4qIs_MgyDc https://www.youtube.com/watch?v=hYCSvYJrr_I
Latundan (Unripe)	• Latundan Banana Chips Reference: https://youtu.be/9B6GjSGLt1Y?si=Io4eHMf151F5OhFC

Latundan (Ripe)	• Banana Crunch • Latundan Banana Loaf Reference: https://youtu.be/sjsez8Ec_Sk?si=r_VRy-I7-eS5LLdU https://www.facebook.com/190655255003471/posts/pfbid027yzfMf8Z455wNquUNzNT67yYjq4Z81WFvQwz5zmCt9tttz7aSXCZ5PYrCfnmfSPKl/?app=fbl
Latundan (Overripe)	• Banana Blanca • Banana Pie Reference: https://www.youtube.com/watch?v=mdvIa9PUwpY https://www.youtube.com/watch?v=IScTDCBZ4ME

Appendix Table 15. Recommendation Table

APPENDIX 11

Source Code

about.java

```
package com.example.peelperfect;

import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.content.Intent;
import android.net.Uri;
import android.graphics.Bitmap;

public class about extends AppCompatActivity {

   ImageView bck_btn;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.about);

      bck_btn = findViewById(R.id.about_back);

      bck_btn.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent intent = new Intent(about.this,
MainActivity.class);
            startActivity(intent);
         }
      });

   }
}
```

CameraMonitor.java

```
package com.example.peelperfect;

import android.annotation.SuppressLint;
import android.app.Dialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.Toast;
import android.Manifest;
import android.provider.MediaStore;
import androidx.activity.result.ActivityResult;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import
androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.camera.core.AspectRatio;
import androidx.camera.core.Camera;
import androidx.camera.core.CameraSelector;
import androidx.camera.core.ImageCapture;
import androidx.camera.core.ImageCaptureException;
import androidx.camera.core.Preview;
import androidx.camera.lifecycle.ProcessCameraProvider;
import androidx.camera.view.PreviewView;
import androidx.core.content.ContextCompat;
import
com.google.common.util.concurrent.ListenableFuture;
import java.io.File;
import java.util.Arrays;
import java.util.concurrent.ExecutionException;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.graphics.Typeface;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.nio.MappedByteBuffer;
import java.util.ArrayList;
import java.util.List;

import org.tensorflow.lite.DataType;
import org.tensorflow.lite.Interpreter;
import
org.tensorflow.lite.support.tensorbuffer.TensorBuffer;

import java.nio.ByteBuffer;

import org.tensorflow.lite.support.common.FileUtil;
import org.tensorflow.lite.support.common.ops.CastOp;
import
org.tensorflow.lite.support.common.ops.NormalizeOp;
import org.tensorflow.lite.support.image.ImageProcessor;
import org.tensorflow.lite.support.image.TensorImage;

import java.io.BufferedReader;
import java.io.InputStreamReader;



public class CameraMonitor extends AppCompatActivity {
   ImageButton capture, toggleFlash, flipCamera,
closeCamera, showPopup;
   private PreviewView previewView;
   int cameraFacing =
CameraSelector.LENS_FACING_BACK;
   Button btnGallery;

   private String modelPath = "model.tflite";
   private String labelPath = "labels.txt";
   private Interpreter interpreter;
   private int tensorWidth = 0;
   private int tensorHeight = 0;
   private int numChannel = 0;
   private int numElements = 0;
   private List<String> labels = new ArrayList<>();
   private ImageProcessor imageProcessor = new
ImageProcessor.Builder()
```

```
        .add(new NormalizeOp(INPUT_MEAN,
INPUT_STANDARD_DEVIATION))
        .add(new CastOp(INPUT_IMAGE_TYPE))
        .build(); // preprocess input

  private static final float INPUT_MEAN = 0f;
  private static final float INPUT_STANDARD_DEVIATION =
255f;
  private static final DataType INPUT_IMAGE_TYPE =
DataType.FLOAT32;
  private static final DataType OUTPUT_IMAGE_TYPE =
DataType.FLOAT32;
  private static final float CONFIDENCE_THRESHOLD =
0.3F;
  private static final float IOU_THRESHOLD = 0.5F;
  ArrayList<String> classNames = new ArrayList<>();
  ArrayList<String> confidenceLevel = new ArrayList<>();
  ActivityResultLauncher<Intent> resultLauncher;

  private final ActivityResultLauncher<String>
activityResultLauncher = registerForActivityResult(new
ActivityResultContracts.RequestPermission(), new
ActivityResultCallback<Boolean>() {
    @Override
    public void onActivityResult(Boolean result) {
      if(result){
        startCamera(cameraFacing);
      }
    }
  });

  @SuppressLint("MissingInflatedId")
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.camera_monitor);

    previewView = findViewById(R.id.cameraPreview);
    capture = findViewById(R.id.capture);
    toggleFlash = findViewById(R.id.toggleFlash);
    flipCamera = findViewById(R.id.flipCamera);
    closeCamera = findViewById(R.id.close_cam);
    showPopup = findViewById(R.id.showPopup);

    showPopup.setOnClickListener(v -> {
      showPopup();
    });

    if
(ContextCompat.checkSelfPermission(CameraMonitor.this
, Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED) {

activityResultLauncher.launch(Manifest.permission.CAME
RA);
    } else {
      startCamera(cameraFacing);
    }

    flipCamera.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {
        if(cameraFacing ==
CameraSelector.LENS_FACING_BACK) {
          cameraFacing =
CameraSelector.LENS_FACING_FRONT;
        } else {
          cameraFacing =
CameraSelector.LENS_FACING_BACK;
        }
        startCamera(cameraFacing);
      }
    });

    closeCamera.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent intent = new Intent(CameraMonitor.this,
Monitor.class);
        startActivity(intent);
      }
    });
    //Gallery
    btnGallery = findViewById(R.id.btnGallery);
    registerResult();

    btnGallery.setOnClickListener(view -> pickImage());
  }
  //Pick Image from Gallery
  private void pickImage(){
    Intent iGallery = new Intent(Intent.ACTION_PICK);

iGallery.setData(MediaStore.Images.Media.EXTERNAL_CO
NTENT_URI);
    resultLauncher.launch(iGallery);


  }
  //Gallery Result
  private void registerResult(){
    resultLauncher = registerForActivityResult(
        new
ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {
          @Override
          public void onActivityResult(ActivityResult
result) {
            try {
              // Get image
              Uri imageUri = result.getData().getData();

              //Start

@org.checkerframework.checker.nullness.qual.NonNull
MappedByteBuffer modelFile = null;
              try {
                modelFile =
FileUtil.loadMappedFile(CameraMonitor.this, modelPath);
              } catch (IOException e) {
                throw new RuntimeException(e);
              }
              Interpreter.Options options = new
Interpreter.Options();
              options.setNumThreads(4);
              interpreter = new Interpreter(modelFile,
options);

              // Get input shape
              int[] inputShape =
interpreter.getInputTensor(0).shape();
              tensorWidth = inputShape[1];
              tensorHeight = inputShape[2];

              // Get output shape
              int[] outputShape =
interpreter.getOutputTensor(0).shape();
              numChannel = outputShape[1];
              numElements = outputShape[2];
```

```
                try {
                    InputStream inputStream =
CameraMonitor.this.getAssets().open(labelPath);
                    BufferedReader reader = new
BufferedReader(new InputStreamReader(inputStream));
                    String line;
                    while ((line = reader.readLine()) != null
&& !line.isEmpty()) {
                        labels.add(line);
                    }
                    reader.close();
                    inputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }

                InputStream inputStream =
getContentResolver().openInputStream(imageUri);
                Bitmap bitmap =
BitmapFactory.decodeStream(inputStream);
                Bitmap resizedBitmap =
Bitmap.createScaledBitmap(bitmap, tensorWidth,
tensorHeight, false);
                TensorImage tensorImage = new
TensorImage(DataType.FLOAT32);
                tensorImage.load(resizedBitmap);

                // Preprocess image
                TensorImage processedImage =
imageProcessor.process(tensorImage);
                ByteBuffer imageBuffer =
processedImage.getBuffer();

                // Create an output tensor buffer
                TensorBuffer output =
TensorBuffer.createFixedSize(new int[]{1, numChannel,
numElements}, OUTPUT_IMAGE_TYPE);

                // Run inference
                interpreter.run(imageBuffer,
output.getBuffer());

                float[] outputArray =
output.getFloatArray();
                List<BoundingBox> bestBoxes =
bestBox(outputArray, numElements, numChannel,
CONFIDENCE_THRESHOLD, labels);

                if (bestBoxes != null) {
                    Bitmap resultBitmap =
drawBoundingBoxes(bitmap, bestBoxes);
                    Uri uri = saveBitmap(resultBitmap);

                    System.out.println("Original List: " +
classNames);

                    if (uri != null) {
                        Intent intent = new
Intent(CameraMonitor.this, OutputMonitor.class);
                        intent.putExtra("resultBitmapUri",
uri.toString());
                        intent.putExtra("classNames",
classNames);
                        intent.putExtra("confidenceLevel",
confidenceLevel);
                        startActivity(intent);
                    } else {
                        System.out.println("Failed to save
bitmap");
                    }
                } else {
                    Intent intent = new
Intent(CameraMonitor.this, OutputMonitor.class);
                    intent.putExtra("imageFilePath",
imageUri.toString());
                    startActivity(intent);
                }
            } catch (Exception e) {
                Toast.makeText(CameraMonitor.this, "No
Image Selected", Toast.LENGTH_SHORT).show();
            }
        }

    }
);
}
public void startCamera(int cameraFacing) {
    int aspectRatio =
aspectRatio(previewView.getWidth(),
previewView.getHeight());
    ListenableFuture<ProcessCameraProvider>
listenableFuture =
ProcessCameraProvider.getInstance(this);

    listenableFuture.addListener(() -> {
        try {
            ProcessCameraProvider cameraProvider =
(ProcessCameraProvider) listenableFuture.get();

            Preview preview = new
Preview.Builder().setTargetAspectRatio(aspectRatio).build
();

            ImageCapture imageCapture = new
ImageCapture.Builder().setCaptureMode(ImageCapture.C
APTURE_MODE_MINIMIZE_LATENCY).setTargetRotation(g
etWindowManager().getDefaultDisplay().getRotation()).b
uild();

            CameraSelector cameraSelector = new
CameraSelector.Builder().requireLensFacing(cameraFacing
).build();

            cameraProvider.unbindAll();

            Camera camera =
cameraProvider.bindToLifecycle(this, cameraSelector,
preview, imageCapture);

            capture.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    loading();
                    takePicture(imageCapture);
                    capture.setOnClickListener(null);
                }

            });

            toggleFlash.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    setFlashIcon(camera);
                }
            });
```

```
preview.setSurfaceProvider(previewView.getSurfaceProvi
der());
        } catch (ExecutionException | InterruptedException
e) {
            e.printStackTrace();
        }
    }, ContextCompat.getMainExecutor(this));
  }
  public void takePicture(ImageCapture imageCapture) {
    final File file = new
File(getExternalFilesDir(Environment.DIRECTORY_PICTURE
S), System.currentTimeMillis() + ".jpg");
    ImageCapture.OutputFileOptions outputFileOptions =
new ImageCapture.OutputFileOptions.Builder(file).build();
    imageCapture.takePicture(outputFileOptions,
ContextCompat.getMainExecutor(this), new
ImageCapture.OnImageSavedCallback() {
        @Override
        public void onImageSaved(@NonNull
ImageCapture.OutputFileResults outputFileResults) {
            // Image saved, pass the file path to the next
activity
            // Image was stored in file.getAbsolutePath()

            //Start

@org.checkerframework.checker.nullness.qual.NonNull
MappedByteBuffer modelFile = null;
            try {
                modelFile =
FileUtil.loadMappedFile(CameraMonitor.this, modelPath);
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            Interpreter.Options options = new
Interpreter.Options();
            options.setNumThreads(4);
            interpreter = new Interpreter(modelFile,
options);

            // Get input shape
            int[] inputShape =
interpreter.getInputTensor(0).shape();
            tensorWidth = inputShape[1];
            tensorHeight = inputShape[2];

            System.out.println("Input Shape: " +
Arrays.toString(inputShape));

            // Get output shape
            int[] outputShape =
interpreter.getOutputTensor(0).shape();
            numChannel = outputShape[1];
            numElements = outputShape[2];

            try {
                InputStream inputStream =
CameraMonitor.this.getAssets().open(labelPath);
                BufferedReader reader = new
BufferedReader(new InputStreamReader(inputStream));
                String line;
                while ((line = reader.readLine()) != null &&
!line.isEmpty()) {
                    labels.add(line);
                }
                reader.close();
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }

            Bitmap bitmap =
BitmapFactory.decodeFile(file.getAbsolutePath());
            Bitmap resizedBitmap =
Bitmap.createScaledBitmap(bitmap, tensorWidth,
tensorHeight, false);
            TensorImage tensorImage = new
TensorImage(DataType.FLOAT32);
            tensorImage.load(resizedBitmap);

            // Preprocess image
            TensorImage processedImage =
imageProcessor.process(tensorImage);
            ByteBuffer imageBuffer =
processedImage.getBuffer();

            // Create an output tensor buffer
            TensorBuffer output =
TensorBuffer.createFixedSize(new int[]{1, numChannel,
numElements}, OUTPUT_IMAGE_TYPE);

            System.out.println(imageBuffer);
            // Run inference
            interpreter.run(imageBuffer, output.getBuffer());

            float[] outputArray = output.getFloatArray();
            List<BoundingBox> bestBoxes =
bestBox(outputArray, numElements, numChannel,
CONFIDENCE_THRESHOLD, labels);

            if (bestBoxes != null) {

                Bitmap resultBitmap =
drawBoundingBoxes(bitmap, bestBoxes);
                Uri uri = saveBitmap(resultBitmap);

                System.out.println("Original List: " +
classNames);

                if (uri != null) {
                    Intent intent = new
Intent(CameraMonitor.this, OutputMonitor.class);
                    intent.putExtra("resultBitmapUri",
uri.toString());
                    intent.putExtra("classNames", classNames);
                    intent.putExtra("confidenceLevel",
confidenceLevel);
                    startActivity(intent);
                } else {
                    System.out.println("Failed to save bitmap");
                }
            } else {
                Intent intent = new Intent(CameraMonitor.this,
OutputMonitor.class);
                intent.putExtra("imageFilePath",
file.getAbsolutePath());
                startActivity(intent);
            }
        }

        @Override
        public void onError(@NonNull
ImageCaptureException exception) {
            Toast.makeText(CameraMonitor.this, "Failed to
save: " + exception.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
  }
```

```
  private void setFlashIcon(Camera camera) {
    if(camera.getCameraInfo().hasFlashUnit()) {

if(camera.getCameraInfo().getTorchState().getValue() ==
0) {
          camera.getCameraControl().enableTorch(true);
          toggleFlash.setImageResource(R.drawable.flash);
        } else {
          camera.getCameraControl().enableTorch(false);

toggleFlash.setImageResource(R.drawable.flash_off);
        }
    } else {
      runOnUiThread(new Runnable() {
        @Override
        public void run() {
          Toast.makeText(CameraMonitor.this, "Flash is
not available currently", Toast.LENGTH_SHORT).show();
        }
      });
    }
  }
  private int aspectRatio(int width, int height) {
    double previewRatio = (double) Math.max(width,
height) / Math.min(width, height);
    if (Math.abs(previewRatio - 4.0 / 3.0) <=
Math.abs(previewRatio - 16.0 / 9.0)) {
      return AspectRatio.RATIO_4_3;
    }
    return AspectRatio.RATIO_16_9;
  }
  private void showPopup(){

    Handler handler = new
Handler(Looper.getMainLooper());
    handler.postDelayed(new Runnable() {
      @Override
      public void run() {
        final Dialog dialog = new
Dialog(CameraMonitor.this);

dialog.requestWindowFeature(Window.FEATURE_NO_TIT
LE);
        dialog.setCancelable(true);
        dialog.setContentView(R.layout.dialog_popup);
        dialog.show();
        Button close =
dialog.findViewById(R.id.continueBtn);
        close.setOnClickListener(new
View.OnClickListener() {
          @Override
          public void onClick(View v) {
            dialog.dismiss();
          }
        });
      }
    },100);
  }

  public static class BoundingBox {
    private float x1;
    private float y1;
    private float x2;
    private float y2;
    private float cx;
    private float cy;
    private float w;
    private float h;
    private float cnf;
    private int cls;
    private String clsName;

    public BoundingBox(float x1, float y1, float x2, float
y2, float cx, float cy, float w, float h, float cnf, int cls, String
clsName) {
      this.x1 = x1;
      this.y1 = y1;
      this.x2 = x2;
      this.y2 = y2;
      this.cx = cx;
      this.cy = cy;
      this.w = w;
      this.h = h;
      this.cnf = cnf;
      this.cls = cls;
      this.clsName = clsName;
    }

    public float getX1() {
      return x1;
    }

    public float getY1() {
      return y1;
    }

    public float getX2() {
      return x2;
    }

    public float getY2() {
      return y2;
    }

    public float getCx() {
      return cx;
    }

    public float getCy() {
      return cy;
    }

    public float getW() {
      return w;
    }

    public float getH() {
      return h;
    }

    public Float getCnf() {
      return cnf;
    }

    public int getCls() {
      return cls;
    }

    public String getClsName() {
      return clsName;
    }
  }

  public static List<BoundingBox> bestBox(float[] array, int
numElements, int numChannel, float
CONFIDENCE_THRESHOLD, List<String> labels) {
    List<BoundingBox> boundingBoxes = new
ArrayList<>();
```

```
        for (int c = 0; c < numElements; c++) {
            float maxConf = -1.0f;
            int maxIdx = -1;
            int j = 4;
            int arrayIdx = c + numElements * j;
            while (j < numChannel) {
                if (array[arrayIdx] > maxConf) {
                    maxConf = array[arrayIdx];
                    maxIdx = j - 4;
                }
                j++;
                arrayIdx += numElements;
            }

            if (maxConf > CONFIDENCE_THRESHOLD) {
                String clsName = labels.get(maxIdx);
                float cx = array[c]; // 0
                float cy = array[c + numElements]; // 1
                float w = array[c + numElements * 2];
                float h = array[c + numElements * 3];
                float x1 = cx - (w / 2F);
                float y1 = cy - (h / 2F);
                float x2 = cx + (w / 2F);
                float y2 = cy + (h / 2F);
                if (x1 < 0F || x1 > 1F) continue;
                if (y1 < 0F || y1 > 1F) continue;
                if (x2 < 0F || x2 > 1F) continue;
                if (y2 < 0F || y2 > 1F) continue;

                boundingBoxes.add(new BoundingBox(x1, y1, x2,
y2, cx, cy, w, h, maxConf, maxIdx, clsName));
            }
        }

        if (boundingBoxes.isEmpty()) return null;

        return applyNMS(boundingBoxes);
    }

    private static List<BoundingBox>
applyNMS(List<BoundingBox> boxes) {
        List<BoundingBox> sortedBoxes = new
ArrayList<>(boxes);
        sortedBoxes.sort((box1, box2) ->
Float.compare(box2.getCnf(), box1.getCnf()));

        List<BoundingBox> selectedBoxes = new
ArrayList<>();

        while (!sortedBoxes.isEmpty()) {
            BoundingBox first = sortedBoxes.get(0);
            selectedBoxes.add(first);
            sortedBoxes.remove(first);

            for (int i = 0; i < sortedBoxes.size(); i++) {
                BoundingBox nextBox = sortedBoxes.get(i);
                float iou = calculateIoU(first, nextBox);
                if (iou >= IOU_THRESHOLD) {
                    sortedBoxes.remove(nextBox);
                    i--;
                }
            }
        }
        return selectedBoxes;
    }
    private static float calculateIoU(BoundingBox box1,
BoundingBox box2) {
        float x1 = Math.max(box1.getX1(), box2.getX1());
        float y1 = Math.max(box1.getY1(), box2.getY1());
        float x2 = Math.min(box1.getX2(), box2.getX2());
        float y2 = Math.min(box1.getY2(), box2.getY2());
        float intersectionArea = Math.max(0F, x2 - x1) *
Math.max(0F, y2 - y1);
        float box1Area = box1.getW() * box1.getH();
        float box2Area = box2.getW() * box2.getH();
        return intersectionArea / (box1Area + box2Area -
intersectionArea);
    }

    public Bitmap drawBoundingBoxes(Bitmap bitmap,
List<BoundingBox> boxes) {
        Bitmap mutableBitmap =
bitmap.copy(Bitmap.Config.ARGB_8888, true);
        Canvas canvas = new Canvas(mutableBitmap);
        Paint paint = new Paint();
        paint.setColor(Color.RED);
        paint.setStyle(Paint.Style.STROKE);
        paint.setStrokeWidth(15f);

        Paint textPaint = new Paint();
        textPaint.setColor(Color.RED);
        textPaint.setTextSize(100f);
        textPaint.setTypeface(Typeface.DEFAULT_BOLD);


        for (BoundingBox box : boxes) {
            RectF rect = new RectF(
                    box.getX1() * mutableBitmap.getWidth(),
                    box.getY1() * mutableBitmap.getHeight(),
                    box.getX2() * mutableBitmap.getWidth(),
                    box.getY2() * mutableBitmap.getHeight()
            );
            canvas.drawRect(rect, paint);
            canvas.drawText(box.getClsName(), rect.left,
rect.bottom, textPaint);
            classNames.add(box.getClsName());
            confidenceLevel.add(box.getCnf().toString());
        }

        return mutableBitmap;
    }

    private Uri saveBitmap(Bitmap bitmap) {
        String fileName = "result_image.png";
        File file = new
File(getExternalFilesDir(Environment.DIRECTORY_PICTURE
S), fileName);

        try (FileOutputStream fos = new
FileOutputStream(file)) {
            bitmap.compress(Bitmap.CompressFormat.PNG,
100, fos);
            fos.flush();
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }

        return Uri.fromFile(file);
    }

    private void loading(){
        Handler handler = new
Handler(Looper.getMainLooper());
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                final Dialog dialog = new
Dialog(CameraMonitor.this);
```

```
dialog.requestWindowFeature(Window.FEATURE_NO_TIT
LE);
            dialog.setCancelable(false);
            dialog.setContentView(R.layout.loading);
            dialog.show();

            ProgressBar progressBar =
dialog.findViewById(R.id.progressBar);

            // Show the ProgressBar to indicate loading
            progressBar.setVisibility(View.VISIBLE);

            // Simulate a task by using a Handler to delay the
progress visibility change
            new Handler().postDelayed(new Runnable() {
               @Override
               public void run() {
                  // Hide the ProgressBar after 3 seconds
                  progressBar.setVisibility(View.GONE);
                  dialog.dismiss();
               }
            }, 2000); // 3000 milliseconds = 3 seconds
         }
      },100);
   }
}
```

DataHandler.java

```
package com.example.peelperfect;

import android.graphics.Bitmap;

import java.util.ArrayList;
import java.util.List;

public class DataHandler {
   private static ArrayList<Bitmap> bitmapList = new
ArrayList<>();
   private static ArrayList<String> typeList = new
ArrayList<>();
   private static ArrayList<String> ripenessList = new
ArrayList<>();
   private static ArrayList<String> confidenceList = new
ArrayList<>();
   private static ArrayList<String> countList = new
ArrayList<>();
   private static ArrayList<String> dateList = new
ArrayList<>();
   private static int index = 0;
   private static int page = 0;

   public static void addData(Bitmap bitmap, String type,
String ripeness, String confidence, String count, String
date) {
      bitmapList.add(bitmap);
      typeList.add(type);
      ripenessList.add(ripeness);
      confidenceList.add(confidence);
      countList.add(count);
      dateList.add(date);
   }

   public static List<Bitmap> getAllBitmaps() {
      return bitmapList;
   }
   public static List<String> getAllTypes() {
      return typeList;
   }
   public static List<String> getAllRipeness() {
      return ripenessList;
   }
   public static List<String> getAllConfidence() { return
confidenceList; }
   public static List<String> getAllCount() { return
countList; }
   public static List<String> getAllDate() { return dateList; }


   public static Bitmap soloBitmap(int index) {
      return bitmapList.get(index);
   }
   public static String soloType(int index) {
      return typeList.get(index);
   }
   public static String soloRipeness(int index) {
      return ripenessList.get(index);
   }
   public static String soloConfidence(int index) {
      return confidenceList.get(index);
   }
   public static String soloCount(int index) {
      return countList.get(index);
   }
   public static String soloDate(int index) {
      return dateList.get(index);
   }

   public static void clearData(){
      bitmapList.clear();
      typeList.clear();
      ripenessList.clear();
      confidenceList.clear();
      countList.clear();
      dateList.clear();
   }

   public static void clearDataSolo(int index){
      bitmapList.remove(index);
      typeList.remove(index);
      ripenessList.remove(index);
      confidenceList.remove(index);
      countList.remove(index);
      dateList.remove(index);
   }

   public static void changeIndex(int num) {
      index = num;
   }

   public static int getIndex() {
      return index;
   }

   public static int getPage() {
      return page;
   }

   public static void changePage(int num) {
      page = num;
   }

   // Monitoring

   private static ArrayList<Bitmap> bitmapList2 = new
ArrayList<>();
   private static ArrayList<String> typeList2 = new
ArrayList<>();
```

```
   private static ArrayList<String> ripenessList2 = new
ArrayList<>();
   private static ArrayList<String> confidenceList2 = new
ArrayList<>();
   private static ArrayList<String> countList2 = new
ArrayList<>();
   private static ArrayList<String> dateList2 = new
ArrayList<>();

   public static void addData2(Bitmap bitmap, String type,
String ripeness, String confidence, String count, String
date) {
      bitmapList2.add(bitmap);
      typeList2.add(type);
      ripenessList2.add(ripeness);
      confidenceList2.add(confidence);
      countList2.add(count);
      dateList2.add(date);
   }

   public static void addAllData2(List<Bitmap> bitmap,
List<String> type, List<String> ripeness, List<String>
confidence, List<String> count, List<String> date) {
      bitmapList2.addAll(bitmap);
      typeList2.addAll(type);
      ripenessList2.addAll(ripeness);
      confidenceList2.addAll(confidence);
      countList2.addAll(count);
      dateList2.addAll(date);
   }

   public static List<Bitmap> getAllBitmaps2() {
      return bitmapList2;
   }
   public static List<String> getAllTypes2() {
      return typeList2;
   }
   public static List<String> getAllRipeness2() {
      return ripenessList2;
   }
   public static List<String> getAllConfidence2() { return
confidenceList2; }
   public static List<String> getAllCount2() { return
countList2; }
   public static List<String> getAllDate2() { return
dateList2; }

   public static void clearData2(){
      bitmapList2.clear();
      typeList2.clear();
      ripenessList2.clear();
      confidenceList2.clear();
      countList2.clear();
      dateList2.clear();
   }


   // Monitoring #2

   private static ArrayList<Bitmap> bitmapList3 = new
ArrayList<>();
   private static ArrayList<String> typeList3 = new
ArrayList<>();
   private static ArrayList<String> ripenessList3 = new
ArrayList<>();
   private static ArrayList<String> confidenceList3 = new
ArrayList<>();
   private static ArrayList<String> countList3 = new
ArrayList<>();
   private static ArrayList<String> dateList3 = new
ArrayList<>();

   public static void addData3(Bitmap bitmap, String type,
String ripeness, String confidence, String count, String
date) {
      bitmapList3.add(bitmap);
      typeList3.add(type);
      ripenessList3.add(ripeness);
      confidenceList3.add(confidence);
      countList3.add(count);
      dateList3.add(date);
   }

   public static void addAllData3(List<Bitmap> bitmap,
List<String> type, List<String> ripeness, List<String>
confidence, List<String> count, List<String> date) {
      bitmapList3.addAll(bitmap);
      typeList3.addAll(type);
      ripenessList3.addAll(ripeness);
      confidenceList3.addAll(confidence);
      countList3.addAll(count);
      dateList3.addAll(date);
   }

   public static List<Bitmap> getAllBitmaps3() {
      return bitmapList3;
   }
   public static List<String> getAllTypes3() {
      return typeList3;
   }
   public static List<String> getAllRipeness3() {
      return ripenessList3;
   }
   public static List<String> getAllConfidence3() { return
confidenceList3; }
   public static List<String> getAllCount3() { return
countList3; }
   public static List<String> getAllDate3() { return
dateList3; }

   public static void clearData3(){
      bitmapList3.clear();
      typeList3.clear();
      ripenessList3.clear();
      confidenceList3.clear();
      countList3.clear();
      dateList3.clear();
   }

   // Monitoring #3

   private static ArrayList<Bitmap> bitmapList4 = new
ArrayList<>();
   private static ArrayList<String> typeList4 = new
ArrayList<>();
   private static ArrayList<String> ripenessList4 = new
ArrayList<>();
   private static ArrayList<String> confidenceList4 = new
ArrayList<>();
   private static ArrayList<String> countList4 = new
ArrayList<>();
   private static ArrayList<String> dateList4 = new
ArrayList<>();

   public static void addData4(Bitmap bitmap, String type,
String ripeness, String confidence, String count, String
date) {
      bitmapList4.add(bitmap);
      typeList4.add(type);
```

```
        ripenessList4.add(ripeness);
        confidenceList4.add(confidence);
        countList4.add(count);
        dateList4.add(date);
    }

    public static List<Bitmap> getAllBitmaps4() {
        return bitmapList4;
    }
    public static List<String> getAllTypes4() {
        return typeList4;
    }
    public static List<String> getAllRipeness4() {
        return ripenessList4;
    }
    public static List<String> getAllConfidence4() { return
confidenceList4; }
    public static List<String> getAllCount4() { return
countList4; }
    public static List<String> getAllDate4() { return
dateList4; }

    public static void clearData4(){
        bitmapList4.clear();
        typeList4.clear();
        ripenessList4.clear();
        confidenceList4.clear();
        countList4.clear();
        dateList4.clear();
    }
}
```

MainActivity.java

```
package com.example.peelperfect;


import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.Manifest;
import androidx.activity.EdgeToEdge;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import
androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.camera.core.AspectRatio;
import androidx.camera.core.Camera;
import androidx.camera.core.CameraSelector;
import androidx.camera.core.ImageCapture;
import androidx.camera.core.ImageCaptureException;
import androidx.camera.core.Preview;
import androidx.camera.lifecycle.ProcessCameraProvider;
import androidx.camera.view.PreviewView;
import androidx.cardview.widget.CardView;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.provider.MediaStore;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;
import com.example.peelperfect.R;
import
com.google.common.util.concurrent.ListenableFuture;
import java.io.File;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.Executors;
import android.util.Log;



public class MainActivity extends AppCompatActivity {

    ImageView imageView;
    ImageButton cam_button;

    CardView about_card, peel_card;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = findViewById(R.id.output_img);
        cam_button = findViewById(R.id.camera_button);
        about_card = findViewById(R.id.about_card);
        peel_card = findViewById(R.id.peel_card);


        if
(ContextCompat.checkSelfPermission(MainActivity.this,
                Manifest.permission.CAMERA)!=
PackageManager.PERMISSION_GRANTED){

ActivityCompat.requestPermissions(MainActivity.this,
                    new String[]{Manifest.permission.CAMERA},
101);
        }

        cam_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
camera_ui.class);
                startActivity(intent);
            }
        });
        about_card.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
about.class);
                startActivity(intent);
            }
        });
        peel_card.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
peel_history.class);
                startActivity(intent);
            }
```

```
    });


  }


  }
```

Monitor.java

```
package com.example.peelperfect;

import android.app.Dialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.provider.ContactsContract;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;

public class Monitor extends AppCompatActivity {
  LinearLayout containerLayout;
  ImageView bck_btn, history_trash;
  TextView title;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.history_monitor);

    history_trash = findViewById(R.id.history_trash);
    bck_btn = findViewById(R.id.history_back);
    title = findViewById(R.id.history_title);

    int index = DataHandler.getIndex();

    title.setText("Banana " + (index+1));

    Bitmap bitmap = DataHandler.soloBitmap(index);
    String type = DataHandler.soloType(index);
    String ripeness = DataHandler.soloRipeness(index);
    String confidence =
DataHandler.soloConfidence(index);
    String count = DataHandler.soloCount(index);
    String date = DataHandler.soloDate(index);

    // Get the container layout
    containerLayout =
findViewById(R.id.container_layout);

    // Create a new CardView
    CardView cardView = new CardView(this);
    LinearLayout.LayoutParams cardLayoutParams = new
LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT,

getResources().getDimensionPixelSize(R.dimen.card_heig
ht)); // Set height as defined in resources
    cardView.setLayoutParams(cardLayoutParams); // Set
layout parameters

cardView.setRadius(getResources().getDimensionPixelSize
(R.dimen.card_corner_radius)); // Set corner radius

    // Add bottom margin
    cardLayoutParams.bottomMargin =
getResources().getDimensionPixelSize(R.dimen.margin_bo
ttom); // Set bottom margin

    // Create a new LinearLayout (outer)
    LinearLayout outerLinearLayout = new
LinearLayout(this);
    outerLinearLayout.setLayoutParams(new
LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.MATCH_PARENT)); //
Set layout parameters

outerLinearLayout.setOrientation(LinearLayout.HORIZONT
AL); // Set orientation
    outerLinearLayout.setPadding(40, 20, 0, 20); // Set
padding start

outerLinearLayout.setBackground(getResources().getDraw
able(R.drawable.bg1)); // Set background

    // Create a new LinearLayout (inner)
    LinearLayout innerLinearLayout = new
LinearLayout(this);
    LinearLayout.LayoutParams innerLayoutParams = new
LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.WRAP_CONTENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);

innerLayoutParams.setMargins(getResources().getDimens
ionPixelSize(R.dimen.margin_start), 0, 0, 0); // Set margin
start

innerLinearLayout.setLayoutParams(innerLayoutParams);
// Set layout parameters

innerLinearLayout.setOrientation(LinearLayout.HORIZONT
AL); // Set orientation

    // Create a new ImageView
    ImageView imageView = new ImageView(this);
    LinearLayout.LayoutParams imageLayoutParams =
new LinearLayout.LayoutParams(

getResources().getDimensionPixelSize(R.dimen.image_wi
dth), // Set width as defined in resources
        LinearLayout.LayoutParams.WRAP_CONTENT);
    imageView.setLayoutParams(imageLayoutParams); //
Set layout parameters
    imageView.setImageBitmap(bitmap); // Set image
resource
```

```
        // Add ImageView to outer LinearLayout
        outerLinearLayout.addView(imageView);

        // Create a new LinearLayout (vertical)
        LinearLayout verticalLinearLayout = new
LinearLayout(this);
        LinearLayout.LayoutParams verticalLayoutParams =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);

verticalLinearLayout.setLayoutParams(verticalLayoutPara
ms); // Set layout parameters

verticalLinearLayout.setOrientation(LinearLayout.VERTICA
L); // Set orientation

        // Create a new TextView
        TextView textView = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams.setMargins(20, 10, 0, 0); // Set
margin start and top
        textView.setLayoutParams(textLayoutParams); // Set
layout parameters
        textView.setText("Banana " + (index + 1)); // Set text

textView.setTextColor(getResources().getColor(R.color.bla
ck)); // Set text color
        textView.setTextSize(TypedValue.COMPLEX_UNIT_SP,
20); // Set text size
        textView.setTypeface(null, Typeface.BOLD); // Set text
style to bold

        // Add TextView to inner LinearLayout
        verticalLinearLayout.addView(textView);

        // Create and add the second TextView
        TextView textView2 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams2 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams2.setMargins(20,10, 0, 0); // Set
margin start
        textView2.setLayoutParams(textLayoutParams2); //
Set layout parameters
        textView2.setText("Type: " + type); // Set text

textView2.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView2.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the second TextView to inner LinearLayout
        verticalLinearLayout.addView(textView2);

        // Create and add the third TextView
        TextView textView3 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams3 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams3.setMargins(20,10, 0, 0); // Set
margin start
        textView3.setLayoutParams(textLayoutParams3); //
Set layout parameters
        textView3.setText("Level: " + ripeness); // Set text

textView3.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView3.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the third TextView to inner LinearLayout
        verticalLinearLayout.addView(textView3);

        TextView textView5 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams5 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams5.setMargins(20,10, 0, 0); // Set
margin start and top
        textView5.setLayoutParams(textLayoutParams5); //
Set layout parameters
        textView5.setText("Banana Count: " + count); // Set
text

textView5.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView5.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout.addView(textView5);

        TextView textView6 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams6 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams6.setMargins(20,10, 0, 0); // Set
margin start and top
        textView6.setLayoutParams(textLayoutParams6); //
Set layout parameters
        textView6.setText("Confidence Level: " + confidence +
"%"); // Set text

textView6.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView6.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout.addView(textView6);

        // Create and add the fourth TextView
        TextView textView4 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams4 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams4.setMargins(20,10, 0, 0); // Set
margin start and top
        textView4.setLayoutParams(textLayoutParams4); //
Set layout parameters
        textView4.setText("Time: " + date); // Set text

textView4.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color
```

```
textView4.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

    // Add the fourth TextView to inner LinearLayout
    verticalLinearLayout.addView(textView4);

    // Add vertical LinearLayout to outer LinearLayout
    outerLinearLayout.addView(verticalLinearLayout);

    // Add inner LinearLayout to outer LinearLayout
    outerLinearLayout.addView(innerLinearLayout);

    // Add outer LinearLayout to the CardView
    cardView.addView(outerLinearLayout);

    // Add the CardView to the container layout
    containerLayout.addView(cardView);

    // Create a new Button
    Button myButton = new Button(this);
    myButton.setText("Update");

    // Create a GradientDrawable for the button's
background
    GradientDrawable drawable = new
GradientDrawable();
    drawable.setShape(GradientDrawable.RECTANGLE);

drawable.setColor(getResources().getColor(R.color.green))
; // Set the button color
    drawable.setCornerRadius(75); // Set corner radius

    // Set the drawable as the button's background
    myButton.setBackground(drawable);
    int padding =
getResources().getDimensionPixelSize(R.dimen.padding);
// Get padding value from resources
    myButton.setPadding(padding,0,padding,0);

    // Set Layout Parameters for the button
    LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.WRAP_CONTENT,
        LinearLayout.LayoutParams.WRAP_CONTENT
    );
    layoutParams.gravity = Gravity.CENTER; // Center the
button horizontally
    myButton.setLayoutParams(layoutParams);

    // Set OnClickListener for the button
    myButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(Monitor.this,
CameraMonitor.class);
            startActivity(intent);
        }
    });

    // Add the button to the LinearLayout
    containerLayout.addView(myButton);

    // Create a View to represent the horizontal line
    View horizontalLine = new View(this);
    // Set width and height programmatically
    LinearLayout.LayoutParams layoutParams3 = new
LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT, //
width
        5 // height (in pixels)
    );
    // Set margins programmatically
    int margin =
getResources().getDimensionPixelSize(R.dimen.margin_to
p); // Get padding value from resources
    layoutParams3.setMargins(0, margin, 0, 0); // left,
top, right, bottom

    horizontalLine.setLayoutParams(layoutParams3);
    // Set background color

horizontalLine.setBackgroundColor(getResources().getCol
or(R.color.black));

    // Add the horizontal line to the container layout
    containerLayout.addView(horizontalLine);

    // Create a TextView
    TextView textView8 = new TextView(this);
    textView8.setText("Last Update:"); // Set the text for
the TextView

textView8.setTextColor(getResources().getColor(R.color.bl
ack)); // Set the text color
    textView8.setTextSize(18); // Set the text size
    textView8.setTypeface(textView8.getTypeface(),
Typeface.BOLD); // Set the font weight to bold

    // Set Layout Parameters for the TextView
    LinearLayout.LayoutParams layoutParams4 = new
LinearLayout.LayoutParams(
        ViewGroup.LayoutParams.WRAP_CONTENT, //
width
        ViewGroup.LayoutParams.WRAP_CONTENT //
height
    );

    // Add margins programmatically
    layoutParams4.setMargins(0, 20, 0, 20); // left, top,
right, bottom

    textView8.setLayoutParams(layoutParams4); // Set
layout parameters for the TextView

    // Add the TextView to the LinearLayout
    containerLayout.addView(textView8);

    String uriString =
getIntent().getStringExtra("resultBitmapUri");
    String bananaType =
getIntent().getStringExtra("bananaType");
    String ripenessLevel =
getIntent().getStringExtra("ripenessLevel");
    String bananaCount =
getIntent().getStringExtra("bananaCount");
    String confidenceLevel =
getIntent().getStringExtra("confidenceLevel");

    // Get Current Date
    LocalDateTime currentTime = LocalDateTime.now();

    // Define a custom date and time formatter
    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("MM-dd-yyyy HH:mm:ss");
    String date1 = currentTime.format(formatter);

    if (uriString != null) {
```

```
        Uri uri = Uri.parse(uriString);
        Bitmap bitmap1 =
BitmapFactory.decodeFile(uri.getPath());

        // Add data
        if (index+1 == 1){
           DataHandler.addData2(bitmap1, bananaType,
ripenessLevel, confidenceLevel, bananaCount, date1);
        } else if (index+1 == 2){
           DataHandler.addData3(bitmap1, bananaType,
ripenessLevel, confidenceLevel, bananaCount, date1);
        } else if (index+1 == 3){
           DataHandler.addData4(bitmap1, bananaType,
ripenessLevel, confidenceLevel, bananaCount, date1);
        }

     } else {
        System.out.println("Uri string is null");
     }

     // Retrieve data
     List<Bitmap> lastBitmaps =
DataHandler.getAllBitmaps();
     List<Bitmap> allBitmaps =  new ArrayList<>();
     List<String> allTypes =  new ArrayList<>();
     List<String> allRipeness =  new ArrayList<>();
     List<String> allConfidence =  new ArrayList<>();
     List<String> allCount =  new ArrayList<>();
     List<String> allDate =  new ArrayList<>();

     if (index+1 == 1){
        allBitmaps = DataHandler.getAllBitmaps2();
        allTypes = DataHandler.getAllTypes2();
        allRipeness = DataHandler.getAllRipeness2();
        allConfidence = DataHandler.getAllConfidence2();
        allCount = DataHandler.getAllCount2();
        allDate = DataHandler.getAllDate2();
     } else if (index+1 == 2){
        allBitmaps = DataHandler.getAllBitmaps3();
        allTypes = DataHandler.getAllTypes3();
        allRipeness = DataHandler.getAllRipeness3();
        allConfidence = DataHandler.getAllConfidence3();
        allCount = DataHandler.getAllCount3();
        allDate = DataHandler.getAllDate3();
     } else if (index+1 == 3){
        allBitmaps = DataHandler.getAllBitmaps4();
        allTypes = DataHandler.getAllTypes4();
        allRipeness = DataHandler.getAllRipeness4();
        allConfidence = DataHandler.getAllConfidence4();
        allCount = DataHandler.getAllCount4();
        allDate = DataHandler.getAllDate4();
     }

     // Check if history is empty
     history_trash.setOnClickListener(v -> {
        if (lastBitmaps.isEmpty()){
           Toast.makeText(this, "Already Empty.",
Toast.LENGTH_SHORT).show();
        } else {
           history_trash();
        }
     });

     // Create a new CardView
     for (int i = 0; i < allBitmaps.size(); i++) {
        // Create a new CardView
        CardView cardView10 = new CardView(this);
        LinearLayout.LayoutParams cardLayoutParams10 =
new LinearLayout.LayoutParams(
              LinearLayout.LayoutParams.MATCH_PARENT,
getResources().getDimensionPixelSize(R.dimen.card_heig
ht)); // Set height as defined in resources

cardView10.setLayoutParams(cardLayoutParams10); //
Set layout parameters

cardView10.setRadius(getResources().getDimensionPixelS
ize(R.dimen.card_corner_radius)); // Set corner radius

        // Add bottom margin
        cardLayoutParams10.bottomMargin =
getResources().getDimensionPixelSize(R.dimen.margin_bo
ttom); // Set bottom margin

        // Create a new LinearLayout (outer)
        LinearLayout outerLinearLayout10 = new
LinearLayout(this);
        outerLinearLayout10.setLayoutParams(new
LinearLayout.LayoutParams(
              LinearLayout.LayoutParams.MATCH_PARENT,
              LinearLayout.LayoutParams.MATCH_PARENT));
// Set layout parameters

outerLinearLayout10.setOrientation(LinearLayout.HORIZO
NTAL); // Set orientation
        outerLinearLayout10.setPadding(40, 20, 0, 20); //
Set padding start

outerLinearLayout10.setBackground(getResources().getDr
awable(R.drawable.bg1)); // Set background

        // Create a new LinearLayout (inner)
        LinearLayout innerLinearLayout10 = new
LinearLayout(this);
        LinearLayout.LayoutParams innerLayoutParams10 =
new LinearLayout.LayoutParams(
              LinearLayout.LayoutParams.WRAP_CONTENT,
              LinearLayout.LayoutParams.WRAP_CONTENT);

innerLayoutParams10.setMargins(getResources().getDime
nsionPixelSize(R.dimen.margin_start), 0, 0, 0); // Set
margin start

innerLinearLayout10.setLayoutParams(innerLayoutParams
10); // Set layout parameters

innerLinearLayout10.setOrientation(LinearLayout.HORIZO
NTAL); // Set orientation

        // Create a new ImageView
        ImageView imageView10 = new ImageView(this);
        LinearLayout.LayoutParams imageLayoutParams10
= new LinearLayout.LayoutParams(

getResources().getDimensionPixelSize(R.dimen.image_wi
dth), // Set width as defined in resources
              LinearLayout.LayoutParams.WRAP_CONTENT);

imageView10.setLayoutParams(imageLayoutParams10); //
Set layout parameters
        imageView10.setImageBitmap(allBitmaps.get(i)); //
Set image resource

        // Add ImageView to outer LinearLayout
        outerLinearLayout10.addView(imageView10);

        // Create a new LinearLayout (vertical)
        LinearLayout verticalLinearLayout10 = new
LinearLayout(this);
```

```
        LinearLayout.LayoutParams
verticalLayoutParams10 = new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);

verticalLinearLayout10.setLayoutParams(verticalLayoutPar
ams10); // Set layout parameters

verticalLinearLayout10.setOrientation(LinearLayout.VERTI
CAL); // Set orientation

        // Create a new TextView
        TextView textView10 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams10 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams10.setMargins(20, 10, 0, 0); // Set
margin start and top
        textView10.setLayoutParams(textLayoutParams10);
// Set layout parameters
        textView10.setText("Banana " + (index + 1) + "." + (i
+ 1)); // Set text

textView10.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView10.setTextSize(TypedValue.COMPLEX_UNIT_SP,
20); // Set text size
        textView10.setTypeface(null, Typeface.BOLD); //
Set text style to bold

        // Add TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView10);

        // Create and add the second TextView
        TextView textView11 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams11 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams11.setMargins(20,10, 0, 0); // Set
margin start
        textView11.setLayoutParams(textLayoutParams11);
// Set layout parameters
        textView11.setText("Type: " + allTypes.get(i)); // Set
text

textView11.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView11.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the second TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView11);

        // Create and add the third TextView
        TextView textView12 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams12 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams12.setMargins(20,10, 0, 0); // Set
margin start
        textView12.setLayoutParams(textLayoutParams12);
// Set layout parameters
        textView12.setText("Level: " + allRipeness.get(i)); //
Set text

textView12.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView12.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the third TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView12);

        TextView textView13 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams13 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams13.setMargins(20,10, 0, 0); // Set
margin start and top
        textView13.setLayoutParams(textLayoutParams13);
// Set layout parameters
        textView13.setText("Banana Count: " +
allCount.get(i)); // Set text

textView13.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView13.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView13);

        TextView textView14 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams14 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams14.setMargins(20,10, 0, 0); // Set
margin start and top
        textView14.setLayoutParams(textLayoutParams14);
// Set layout parameters
        textView14.setText("Confidence Level: " +
allConfidence.get(i) + "%"); // Set text

textView14.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView14.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView14);

        // Create and add the fourth TextView
        TextView textView15 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams15 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams15.setMargins(20,10, 0, 0); // Set
margin start and top
        textView15.setLayoutParams(textLayoutParams15);
// Set layout parameters
        textView15.setText("Time: " + allDate.get(i)); // Set
text

textView15.setTextColor(getResources().getColor(R.color.
black)); // Set text color

textView15.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size
```

```
        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout10.addView(textView15);

        // Add vertical LinearLayout to outer LinearLayout

outerLinearLayout10.addView(verticalLinearLayout10);

        // Add inner LinearLayout to outer LinearLayout

outerLinearLayout10.addView(innerLinearLayout10);

        // Add outer LinearLayout to the CardView
        cardView10.addView(outerLinearLayout10);

        // Add the CardView to the container layout
        containerLayout.addView(cardView10);
      }

      bck_btn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
          Intent intent = new Intent(Monitor.this,
peel_history.class);
          startActivity(intent);
        }
      });
    }

    private void history_trash(){

      Handler handler = new
Handler(Looper.getMainLooper());
      handler.postDelayed(new Runnable() {
        @Override
        public void run() {
          final Dialog dialog = new Dialog(Monitor.this);

dialog.requestWindowFeature(Window.FEATURE_NO_TIT
LE);
          dialog.setCancelable(true);

dialog.setContentView(R.layout.dialog_trash_output);
          dialog.show();

          TextView yes = dialog.findViewById(R.id.yes);
          TextView no = dialog.findViewById(R.id.no);

          yes.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
              int index = DataHandler.getIndex();
              if (index+1 == 1){

DataHandler.clearDataSolo(DataHandler.getIndex());
                DataHandler.clearData2();

DataHandler.addAllData2(DataHandler.getAllBitmaps3(),
DataHandler.getAllTypes3(),
DataHandler.getAllRipeness3(),
DataHandler.getAllConfidence3(),
DataHandler.getAllCount3(), DataHandler.getAllDate3());
                DataHandler.clearData3();

DataHandler.addAllData3(DataHandler.getAllBitmaps4(),
DataHandler.getAllTypes4(),
DataHandler.getAllRipeness4(),
DataHandler.getAllConfidence4(),
DataHandler.getAllCount4(), DataHandler.getAllDate4());
                DataHandler.clearData4();
              } else if (index+1 == 2){

DataHandler.clearDataSolo(DataHandler.getIndex());
                DataHandler.clearData3();

DataHandler.addAllData3(DataHandler.getAllBitmaps4(),
DataHandler.getAllTypes4(),
DataHandler.getAllRipeness4(),
DataHandler.getAllConfidence4(),
DataHandler.getAllCount4(), DataHandler.getAllDate4());
                DataHandler.clearData4();
              } else if (index+1 == 3){

DataHandler.clearDataSolo(DataHandler.getIndex());
                DataHandler.clearData4();
              }

              Intent intent = new Intent(Monitor.this,
peel_history.class);
              startActivity(intent);
              //containerLayout.removeAllViews();
              dialog.dismiss();
            }
          });

          no.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
              dialog.dismiss();
            }
          });
        }
      },100);
    }
}
```

OutputMonitor.java

```
package com.example.peelperfect;

import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Typeface;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.content.Intent;
import android.net.Uri;
import android.graphics.Bitmap;
import android.widget.RelativeLayout;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.Objects;
```

```
public class OutputMonitor extends AppCompatActivity {
   ImageView outputImage,new_photo, output_back;
   RelativeLayout container, basic_information_layout,
banana_info, banana_main_details;
   TextView new_photo_text, banana_type,
ripeness_level, banana_count, banana_content,
recommendation, basic_info, confidence_level;
   String[] parts;
   String bananaCount;
   double confidence;
   Button save_button;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.output_monitor);

      outputImage = findViewById(R.id.output_img);
      new_photo = findViewById(R.id.new_photo);
      output_back = findViewById(R.id.output_back);
      banana_type = findViewById(R.id.banana_type);
      ripeness_level = findViewById(R.id.ripeness_level);
      confidence_level =
findViewById(R.id.confidence_level);
      banana_count = findViewById(R.id.banana_count);
      banana_content =
findViewById(R.id.banana_content);
      recommendation =
findViewById(R.id.recommendation);
      basic_info = findViewById(R.id.basic_info);
      save_button = findViewById(R.id.save_button);
      basic_information_layout =
findViewById(R.id.basic_information_layout);
      banana_info = findViewById(R.id.banana_info);
      banana_main_details =
findViewById(R.id.banana_main_details);
      new_photo_text =
findViewById(R.id.new_photo_text);
      container = findViewById(R.id.container);

      try {
         Intent intent = getIntent();
         if (intent != null &&
intent.hasExtra("imageFilePath")) {
            String filePath =
intent.getStringExtra("imageFilePath");

            // Check if the file path is a URI
            if (filePath.startsWith("content://") ||
filePath.startsWith("file://")) {
               // This is a URI
               Uri imageUri = Uri.parse(filePath);
               outputImage.setImageURI(imageUri);

               // Hide not necessary layout
               banana_main_details.setVisibility(View.GONE);

basic_information_layout.setVisibility(View.GONE);
               new_photo.setVisibility(View.GONE);
               new_photo_text.setVisibility(View.GONE);

               // Center New Photo Image Button
               RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
save_button.getLayoutParams();

layoutParams.removeRule(RelativeLayout.ALIGN_PARENT
_END);

layoutParams.addRule(RelativeLayout.CENTER_IN_PAREN
T);
               save_button.setLayoutParams(layoutParams);
               save_button.setText("Retake Photo");
               Drawable drawable =
getResources().getDrawable(R.drawable.baseline_photo_
camera_24); // Replace R.drawable.add_icon with your
drawable resource

save_button.setCompoundDrawablesWithIntrinsicBounds
(drawable, null, null, null);

               // Set Border Radius
               Drawable drawable1 =
ContextCompat.getDrawable(this, R.drawable.border);
               banana_info.setBackground(drawable1);

               // Set Text, Move Center
               banana_count.setText("Banana is not detected
in the image, please take another picture.");
               banana_count.setGravity(Gravity.CENTER);

banana_count.setTextColor(getResources().getColor(R.col
or.light_black));
               banana_count.setTextSize(20);
               RelativeLayout.LayoutParams layoutParams4 =
(RelativeLayout.LayoutParams)
banana_count.getLayoutParams();
               layoutParams4.setMargins(0, 80, 0, 0);

banana_count.setLayoutParams(layoutParams4);
            } else {
               // This is a file path to a bitmap
               Bitmap bitmap =
BitmapFactory.decodeFile(filePath);
               outputImage.setImageBitmap(bitmap);

               // Hide not necessary layout
               banana_main_details.setVisibility(View.GONE);

basic_information_layout.setVisibility(View.GONE);
               //save_button.setVisibility(View.GONE);
               new_photo.setVisibility(View.GONE);
               new_photo_text.setVisibility(View.GONE);

               // Center New Photo Image Button
               RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
save_button.getLayoutParams();

layoutParams.removeRule(RelativeLayout.ALIGN_PARENT
_END);

layoutParams.addRule(RelativeLayout.CENTER_IN_PAREN
T);
               save_button.setLayoutParams(layoutParams);
               save_button.setText("Retake Photo");
               Drawable drawable =
getResources().getDrawable(R.drawable.baseline_photo_
camera_24); // Replace R.drawable.add_icon with your
drawable resource

save_button.setCompoundDrawablesWithIntrinsicBounds
(drawable, null, null, null);


               // Set Border Radius
               Drawable drawable1 =
ContextCompat.getDrawable(this, R.drawable.border);
```

```
            banana_info.setBackground(drawable1);

            // Set Text, Move Center
            banana_count.setText("Banana is not detected
in the image, please take another picture.");
            banana_count.setGravity(Gravity.CENTER);

banana_count.setTextColor(getResources().getColor(R.col
or.light_black));
            banana_count.setTextSize(20);
            RelativeLayout.LayoutParams layoutParams4 =
(RelativeLayout.LayoutParams)
banana_count.getLayoutParams();
            layoutParams4.setMargins(0, 80, 0, 0);

banana_count.setLayoutParams(layoutParams4);
          }

        } else {
          Uri uri =
Uri.parse(getIntent().getStringExtra("resultBitmapUri"));
          Bitmap bitmap =
BitmapFactory.decodeFile(uri.getPath());
          outputImage.setImageBitmap(bitmap);

          ArrayList<String> classNames =
(ArrayList<String>)
getIntent().getSerializableExtra("classNames");
          parts = classNames.get(0).split(" ");
          bananaCount = String.valueOf(classNames.size());
          banana_type.setText("Banana Type: " + parts[0]);
          banana_type.setTextColor(
getResources().getColor(R.color.light_black));
          ripeness_level.setText("Ripeness Level: " +
parts[1]);
          ripeness_level.setTextColor(
getResources().getColor(R.color.light_black));

          ArrayList<String> confidenceLevel =
(ArrayList<String>)
getIntent().getSerializableExtra("confidenceLevel");
          double convert =
Double.parseDouble(confidenceLevel.get(0));
          confidence = Math.round(convert * 100);
          confidence_level.setText("Status: " + confidence
+ "%");

confidence_level.setTextColor(getResources().getColor(R.c
olor.light_black));

          // Count the number of pieces
          if (classNames.size() > 1) {
            String bananaCounter = ("Cluster: " +
classNames.size() + " pieces");
            banana_count.setText(bananaCounter);
            banana_count.setTextColor(
getResources().getColor(R.color.light_black));
          } else {
            String bananaCounter = ("Cluster: No");
            banana_count.setText(bananaCounter);
            banana_count.setTextColor(
getResources().getColor(R.color.light_black));
          }

          // Default, Check Banana Type
          if (Objects.equals(parts[0], "Lakatan")) {
            banana_content.setTextColor(
getResources().getColor(R.color.light_black));
            RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
            layoutParams.setMargins(0, 35, 0, 0);
            if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration\n\nOverripe:
approximately 4 days until rot" +
getString(R.string.lakatan));
            }else if (Objects.equals(parts[1], "Ripe")){

banana_content.setText("\nDuration\n\nRipe:
approximately 3 to 4 days until overripe" +
getString(R.string.lakatan));
            }     else{

banana_content.setText("\nDuration:\n\nUnripe:
approximately 4 days until ripe"
+getString(R.string.lakatan));
            }

banana_content.setLayoutParams(layoutParams);

          } else if (Objects.equals(parts[0], "Saba")) {
            banana_content.setTextColor(
getResources().getColor(R.color.light_black));
            RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
            layoutParams.setMargins(0, 35, 0, 0);
            if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration\n\nOverripe:
approximately 3 days until rot" + getString(R.string.saba));
            }else if (Objects.equals(parts[1], "Ripe")){

banana_content.setText("\nDuration\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.saba));
            }     else{

banana_content.setText("\nDuration:\n\nUnripe:
approximately 2 days until ripe" +getString(R.string.saba));
            }

banana_content.setLayoutParams(layoutParams);
          } else if (Objects.equals(parts[0], "Latundan")) {
            banana_content.setTextColor(
getResources().getColor(R.color.light_black));
            RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
            layoutParams.setMargins(0, 35, 0, 0);
            if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration\n\nOverripe:
approximately 3 to 4 days until rot" +
getString(R.string.latundan));
            }else if (Objects.equals(parts[1], "Ripe")){

banana_content.setText("\nDuration\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.latundan));
            }     else{

banana_content.setText("\nDuration:\n\nUnripe:
approximately 1 to 2 days until ripe"
+getString(R.string.latundan));
            }
```

```
banana_content.setLayoutParams(layoutParams);
            }
            else {
                banana_content.setTextColor(
getResources().getColor(R.color.light_black));
                RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
                layoutParams.setMargins(0, 35, 0, 0);
                if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration\n\nOverripe:
approximately 3 to 4 days until rot" +
getString(R.string.senorita));
                }else if (Objects.equals(parts[1], "Ripe")){

banana_content.setText("\nDuration\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.senorita));
                }       else{

banana_content.setText("\nDuration:\n\nUnripe:
approximately 2 days until ripe" +
getString(R.string.senorita));
                }

banana_content.setLayoutParams(layoutParams);
            }
            basic_info.setTypeface(null, Typeface.BOLD);

        }
    } catch (Exception e) {
        System.out.println("Exception: " + e.getMessage());
    }
    new_photo.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(OutputMonitor.this,
CameraMonitor.class);
            startActivity(intent);
        }
    });

    output_back.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(OutputMonitor.this,
MainActivity.class);
            startActivity(intent);
        }
    });

    recommendation.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ArrayList<String> classNames =
(ArrayList<String>)
getIntent().getSerializableExtra("classNames");
            if (classNames.size() > 1){

recommendation.setTextColor(ContextCompat.getColor(O
utputMonitor.this, R.color.green));
                recommendation.setTypeface(null,
Typeface.BOLD);
basic_info.setTextColor(ContextCompat.getColor(Output
Monitor.this, R.color.gray));
                basic_info.setTypeface(null,
Typeface.NORMAL);
                banana_count.setVisibility(View.GONE);
                RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
                layoutParams.setMargins(0, 0, 0, 0);

banana_content.setLayoutParams(layoutParams);

                runRecommendation(DataHandler.getPage());
            } else {
                runRecommendation(0);
            }
        }
    });

    basic_info.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ArrayList<String> classNames =
(ArrayList<String>)
getIntent().getSerializableExtra("classNames");
            if (classNames.size() > 1){

basic_info.setTextColor(ContextCompat.getColor(Output
Monitor.this, R.color.green));
                basic_info.setTypeface(null, Typeface.BOLD);

recommendation.setTextColor(ContextCompat.getColor(O
utputMonitor.this, R.color.gray));
                recommendation.setTypeface(null,
Typeface.NORMAL);
                banana_count.setVisibility(View.VISIBLE);

                runOutput(DataHandler.getPage());
            } else {
                runOutput(0);
            }
        }
    });

    save_button.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ArrayList<String> classNames =
(ArrayList<String>)
getIntent().getSerializableExtra("classNames");
            if (classNames.size() > 1) {
                if (DataHandler.getPage() + 2 >
classNames.size()){
                    DataHandler.changePage(0);
                    runOutput(DataHandler.getPage());
                    basic_info.performClick();
                } else {

DataHandler.changePage(DataHandler.getPage()+1);
                    runOutput(DataHandler.getPage());
                    basic_info.performClick();
                }
            } else {
                try {
                    Uri uri =
Uri.parse(getIntent().getStringExtra("resultBitmapUri"));
```

```
                Intent intent = new
Intent(OutputMonitor.this, peel_history.class);
                if (uri != null) {
                    intent.putExtra("resultBitmapUri",
uri.toString());
                    intent.putExtra("bananaType", parts[0]);
                    intent.putExtra("ripenessLevel", parts[1]);
                    intent.putExtra("bananaCount",
bananaCount);
                    intent.putExtra("confidenceLevel",
String.valueOf(confidence));
                    System.out.println("Intent is not null.");
                    startActivity(intent);
                } else {
                    // Handle the case where the intent is null
                    System.out.println("Intent is null.");
                }
            } catch (Exception e) {
                Intent intent = new
Intent(OutputMonitor.this, camera_ui.class);
                startActivity(intent);
                System.out.println("Exception: " +
e.getMessage());
            }
        }
    }
  });
}

  public void runOutput(int index){
    ArrayList<String> classNames = (ArrayList<String>)
getIntent().getSerializableExtra("classNames");
    parts = classNames.get(index).split(" ");
    bananaCount = String.valueOf(classNames.size());
    banana_type.setText("Banana Type: " + parts[0]);
    banana_type.setTextColor(
getResources().getColor(R.color.light_black));
    ripeness_level.setText("Ripeness Level: " + parts[1]);
    ripeness_level.setTextColor(
getResources().getColor(R.color.light_black));

    ArrayList<String> confidenceLevel =
(ArrayList<String>)
getIntent().getSerializableExtra("confidenceLevel");
    double convert =
Double.parseDouble(confidenceLevel.get(index));
    confidence = Math.round(convert * 100);
    confidence_level.setText("Status: " + confidence +
"%");

confidence_level.setTextColor(getResources().getColor(R.c
olor.light_black));

    // Count the number of pieces
    if (classNames.size() > 1) {
      String bananaCounter = ("Cluster: " +
classNames.size() + " pieces");
      banana_count.setText(bananaCounter);
      banana_count.setTextColor(
getResources().getColor(R.color.light_black));
    } else {
      String bananaCounter = ("Cluster: No");
      banana_count.setText(bananaCounter);
      banana_count.setTextColor(
getResources().getColor(R.color.light_black));
    }

    // Default, Check Banana Type
    if (Objects.equals(parts[0], "Lakatan")) {
      banana_content.setTextColor(
getResources().getColor(R.color.light_black));
      RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
      layoutParams.setMargins(0, 35, 0, 0);
      if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration:\n\nOverripe:
approximately 4 days until rot" +
getString(R.string.lakatan));
      }else if (Objects.equals(parts[1], "Ripe")){
        banana_content.setText("\nDuration:\n\nRipe:
approximately 3 to 4 days until overripe" +
getString(R.string.lakatan));
      }     else{
        banana_content.setText("\nDuration:\n\nUnripe:
approximately 4 days until ripe"
+getString(R.string.lakatan));
      }
      banana_content.setLayoutParams(layoutParams);

    } else if (Objects.equals(parts[0], "Saba")) {
      banana_content.setTextColor(
getResources().getColor(R.color.light_black));
      RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
      layoutParams.setMargins(0, 35, 0, 0);
      if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration:\n\nOverripe:
approximately 3 days until rot" + getString(R.string.saba));
      }else if (Objects.equals(parts[1], "Ripe")){
        banana_content.setText("\nDuration:\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.saba));
      }     else{
        banana_content.setText("\nDuration:\n\nUnripe:
approximately 2 days until ripe" +getString(R.string.saba));
      }
      banana_content.setLayoutParams(layoutParams);
    } else if (Objects.equals(parts[0], "Latundan")) {
      banana_content.setTextColor(
getResources().getColor(R.color.light_black));
      RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
      layoutParams.setMargins(0, 35, 0, 0);
      if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration:\n\nOverripe:
approximately 3 to 4 days until rot" +
getString(R.string.latundan));
      }else if (Objects.equals(parts[1], "Ripe")){
        banana_content.setText("\nDuration:\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.latundan));
      }     else{
        banana_content.setText("\nDuration:\n\nUnripe:
approximately 1 to 2 days until ripe"
+getString(R.string.latundan));
      }
      banana_content.setLayoutParams(layoutParams);
    }
    else {
      banana_content.setTextColor(
getResources().getColor(R.color.light_black));
```

```
        RelativeLayout.LayoutParams layoutParams =
(RelativeLayout.LayoutParams)
banana_content.getLayoutParams();
        layoutParams.setMargins(0, 35, 0, 0);
        if (Objects.equals(parts[1], "Overripe")){

banana_content.setText("\nDuration:\n\nOverripe:
approximately 3 to 4 days until rot" +
getString(R.string.senorita));
        }else if (Objects.equals(parts[1], "Ripe")){
           banana_content.setText("\nDuration:\n\nRipe:
approximately 2 to 4 days until overripe" +
getString(R.string.senorita));
        }      else{
           banana_content.setText("\nDuration:\n\nUnripe:
approximately 2 days until ripe"
+getString(R.string.senorita));
        }
        banana_content.setLayoutParams(layoutParams);
     }
     basic_info.setTypeface(null, Typeface.BOLD);
  }

  public void runRecommendation(int index){
     ArrayList<String> classNames = (ArrayList<String>)
getIntent().getSerializableExtra("classNames");
     parts = classNames.get(index).split(" ");

     //Content
     if (Objects.equals(parts[0], "Lakatan")) {
        if (Objects.equals(parts[1], "Overripe")) {
           banana_content.setText(Html.fromHtml("Here
are some ideas for using overripe lakatan:<br><br><a
href='https://www.youtube.com/watch?v=f4qIs_MgyDc'>
Banana Flan</a><br><a
href='https://www.youtube.com/watch?v=hYCSvYJrr_I'>Si
naging</a><br><br>If you like to explore more recipe
options, you can search online for \"overripe lakatan
recipes\" or \"banana desserts using lakatan\".
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Ripe")) {
           banana_content.setText(Html.fromHtml("Here
are some ideas for using ripe lakatan:<br><br><a
href='https://www.youtube.com/watch?v=2MAZq5QxHgk
'>Lakatan Banana Pancake</a><br><a
href='https://www.youtube.com/watch?v=XXv50RTK2SM'
>Lakatan Banana Balls</a><br><a
href='https://www.youtube.com/watch?v=KmKpZR9d9PE'
>Lakatan Banana Fritter (Maruya)</a><br><a
href='https://www.tiktok.com/@kayejchannel/video/725
4920493311020294?lang=en'>Lakatan
Pudding</a><br><br>if you like to explore more recipe
options, you can search online for \"ripe lakatan recipes\"
or \"banana desserts using lakatan\". <br><br>Practical
Tips:<br><br>-To slow down ripening, store bananas in a
cool, dry place away from other fruits.<br>-To speed up
ripening, place bananas in a paper bag with an apple or
another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Unripe")) {
           banana_content.setText(Html.fromHtml("Here
are some ideas for using unripe lakatan:<br><br><a
href='https://www.youtube.com/watch?v=MWx-
OiUOCsE'>Boiled Unripe Lakatan</a><br><a
href='https://www.youtube.com/watch?v=52QZO11oMw
w'>Lakatan Banana Chips</a><br><br>Store it up for 2 to
7 days and you can scan it again using our application to
determine the perfect time to consume/use it.
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }
     }else if (Objects.equals(parts[0], "Saba")) {
        if (Objects.equals(parts[1], "Overripe")) {
           banana_content.setText(Html.fromHtml("Here
are some ideas for using overripe saba:<br><br><a
href='https://www.youtube.com/watch?v=ohO1IHUGkG4'
>Saba Smoothie</a><br><a
href='https://www.youtube.com/watch?v=bcuHBR-
CtJM'>Saba Pancake</a><br><br>If you like to explore
more recipe options, you can search online for \"overripe
saba recipes\" or \"banana desserts using saba\".
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Ripe")) {
           banana_content.setText(Html.fromHtml( "Here
are some ideas for using ripe saba:<br><br><a
href='https://www.youtube.com/watch?v=Mqk2MiJHqno'
>Banana Cue</a><br><a
href='https://www.youtube.com/watch?v=Mqk2MiJHqno'
>Banana Roll(Turon)</a><br><a
href='https://www.youtube.com/watch?v=Mqk2MiJHqno'
>Banana Fritter (Maruya)</a><br><a
href='https://www.youtube.com/watch?v=Fs3gpHsnaKQ'>
Caramelized Banana</a><br><br>If you like to explore
more recipe options, you can search online for \"ripe saba
recipes\" or \"banana desserts using saba\".
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Unripe")) {
           banana_content.setText(Html.fromHtml("Here
are some ideas for using unripe saba:<br><br><a
href='https://www.youtube.com/watch?v=bx_ZRQuxoUw'
```

```
>Pinakro</a><br><a
href='https://www.youtube.com/watch?v=bx_ZRQuxoUw'
>Binangkal</a><br><a
href='https://www.youtube.com/watch?v=HOTkKsDW7so'
>Banana Chips</a><br><br>Store it up for 2 to 7 days and
you can scan it again using our application to determine
the perfect time to consume/use it. <br><br>Practical
Tips:<br><br>-To slow down ripening, store bananas in a
cool, dry place away from other fruits.<br>-To speed up
ripening, place bananas in a paper bag with an apple or
another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }
      }else if (Objects.equals(parts[0], "Latundan")) {
        if (Objects.equals(parts[1], "Overripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using overripe latundan:<br><br><a
href='https://www.youtube.com/watch?v=IScTDCBZ4ME'
>Banana Pie</a><br><a
href='https://www.youtube.com/watch?v=mdvIa9PUwpY'
>Banana Blanca</a><br><br>If you like to explore more
recipe options, you can search online for \"overripe
latundan recipes\" or \"banana desserts using latundan\".
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Ripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using ripe latundan:<br><br><a
href='https://www.youtube.com/watch?v=sjsez8Ec_Sk'>B
anana Crunch</a><br><a
href='https://www.facebook.com/100075687441734/post
s/842157866519870/'>Latundan Banana
Loaf</a><br><br>if you like to explore more recipe
options, you can search online for \"ripe latundan
recipes\" or \"banana desserts using latundan\".
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit." ));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Unripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using unripe latundan:<br><br><a
href='https://www.youtube.com/watch?v=9B6GjSGLt1Y'>
Latundan Banana Chips</a><br><br>Store it up for 2 to 7
days and you can scan it again using our application to
determine the perfect time to consume/use it.
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }
      }else {
        if (Objects.equals(parts[1], "Overripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using overripe senorita:<br><br>Can
be fed to your pet pigs<br><a
href='https://foodhero.org/healthy-food/beyond-peel-
creative-ways-use-overripe-
bananas#:~:text=They're%20fun%20to%20make,extra%20
fruit%20to%20your%20breakfast!&text=Overripe%20ban
anas%20add%20sweetness%20and,recipe%20that%20call
s%20for%20bananas.'>Senorita Smoothie</a><br><br>If
you like to explore more recipe options, you can search
online for \"overripe senorita recipes\" or \"banana
desserts using senorita\". <br><br>Practical
Tips:<br><br>-To slow down ripening, store bananas in a
cool, dry place away from other fruits.<br>-To speed up
ripening, place bananas in a paper bag with an apple or
another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Ripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using ripe senorita:<br><br><a
href='https://www.cookalmostanything.com/2010/02/sen
orita-banana-fritters.html'>Senorita Banana
Fritters(Maruya)</a><br><a
href='https://www.youtube.com/watch?v=qRDKwNvyXVE'
>Senorita Banana Loaf</a><br><br>If you like to explore
more recipe options, you can search online for \"overripe
senorita recipes\" or \"banana desserts using senorita\"
<br><br>Practical Tips:<br><br>-To slow down ripening,
store bananas in a cool, dry place away from other
fruits.<br>-To speed up ripening, place bananas in a paper
bag with an apple or another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));

banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }else if (Objects.equals(parts[1], "Unripe")) {
          banana_content.setText(Html.fromHtml("Here
are some ideas for using unripe senorita:<br><br><a
href='https://www.treehugger.com/ways-use-green-
bananas-wont-ripen-4864266'>Boiled Green Senorita
Banana</a><br><a
href='https://www.treehugger.com/ways-use-green-
bananas-wont-ripen-4864266'>Baked Green Senorita
Banana</a><br><br>Store it up for 2 to 7 days and you
can scan it again using our application to determine the
perfect time to consume/use it. <br><br>Practical
Tips:<br><br>-To slow down ripening, store bananas in a
cool, dry place away from other fruits.<br>-To speed up
ripening, place bananas in a paper bag with an apple or
another ethylene-producing fruit."));

banana_content.setTextColor(getResources().getColor(R.c
olor.light_black));
```

```
banana_content.setMovementMethod(LinkMovementMe
thod.getInstance());
        }
    }
  }
}
```

peel_history.java

```
package com.example.peelperfect;

import android.app.Dialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.TypedValue;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import java.io.ByteArrayOutputStream;
import java.text.SimpleDateFormat;
import java.time.format.DateTimeFormatter;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.time.LocalDateTime;
// Import DataHandler
import com.example.peelperfect.DataHandler;


public class peel_history extends AppCompatActivity {

  LinearLayout containerLayout;
  ImageView bck_btn, history_trash;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.history_toolbar);

    bck_btn = findViewById(R.id.history_back);
    history_trash = findViewById(R.id.history_trash);

    String uriString =
getIntent().getStringExtra("resultBitmapUri");
    String bananaType =
getIntent().getStringExtra("bananaType");
    String ripenessLevel =
getIntent().getStringExtra("ripenessLevel");
    String bananaCount =
getIntent().getStringExtra("bananaCount");
    String confidenceLevel =
getIntent().getStringExtra("confidenceLevel");

    // Get Current Date
    LocalDateTime currentTime = LocalDateTime.now();

    // Define a custom date and time formatter
    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("MM-dd-yyyy HH:mm:ss");
    String date = currentTime.format(formatter);

    if (uriString != null) {
      Uri uri = Uri.parse(uriString);
      Bitmap bitmap =
BitmapFactory.decodeFile(uri.getPath());

      // Add data
      DataHandler.addData(bitmap, bananaType,
ripenessLevel, confidenceLevel, bananaCount, date);

    } else {
      System.out.println("Uri string is null");
    }

    // Retrieve data
    List<Bitmap> allBitmaps =
DataHandler.getAllBitmaps();
    List<String> allTypes = DataHandler.getAllTypes();
    List<String> allRipeness =
DataHandler.getAllRipeness();
    List<String> allConfidence =
DataHandler.getAllConfidence();
    List<String> allCount = DataHandler.getAllCount();
    List<String> allDate = DataHandler.getAllDate();

    // Check if history is empty
    history_trash.setOnClickListener(v -> {
      if (allBitmaps.isEmpty()){
        Toast.makeText(this, "Already Empty.",
Toast.LENGTH_SHORT).show();
      } else {
        history_trash();
      }
    });

    // Get the container layout
    containerLayout =
findViewById(R.id.container_layout);

    // Create a new CardView
    for (int i = 0; i < allBitmaps.size(); i++) {
      // Create a new CardView
      CardView cardView = new CardView(this);
      LinearLayout.LayoutParams cardLayoutParams =
new LinearLayout.LayoutParams(
          LinearLayout.LayoutParams.MATCH_PARENT,

getResources().getDimensionPixelSize(R.dimen.card_heig
ht)); // Set height as defined in resources
      cardView.setLayoutParams(cardLayoutParams); //
Set layout parameters

cardView.setRadius(getResources().getDimensionPixelSize
(R.dimen.card_corner_radius)); // Set corner radius

      // Add bottom margin
      cardLayoutParams.bottomMargin =
getResources().getDimensionPixelSize(R.dimen.margin_bo
ttom); // Set bottom margin

      // Set CardView to be clickable
      cardView.setClickable(true);
      cardView.setFocusable(true);
```

```
        // Set a tag with index to get the index of clicked
ViewCard
        cardView.setTag(i);

        cardView.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Retrieve the index from the tag
                int index = (int) v.getTag();

                // Create an Intent to start a new activity
                Intent intent = new Intent(peel_history.this,
Monitor.class);
                DataHandler.changeIndex(index);
                startActivity(intent);
            }
        });

        // Create a new LinearLayout (outer)
        LinearLayout outerLinearLayout = new
LinearLayout(this);
        outerLinearLayout.setLayoutParams(new
LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.MATCH_PARENT,
                LinearLayout.LayoutParams.MATCH_PARENT));
// Set layout parameters

outerLinearLayout.setOrientation(LinearLayout.HORIZONT
AL); // Set orientation
        outerLinearLayout.setPadding(40, 20, 0, 20); // Set
padding start

outerLinearLayout.setBackground(getResources().getDraw
able(R.drawable.bg1)); // Set background

        // Create a new LinearLayout (inner)
        LinearLayout innerLinearLayout = new
LinearLayout(this);
        LinearLayout.LayoutParams innerLayoutParams =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);

innerLayoutParams.setMargins(getResources().getDimens
ionPixelSize(R.dimen.margin_start), 0, 0, 0); // Set margin
start

innerLinearLayout.setLayoutParams(innerLayoutParams);
// Set layout parameters

innerLinearLayout.setOrientation(LinearLayout.HORIZONT
AL); // Set orientation

        // Create a new ImageView
        ImageView imageView = new ImageView(this);
        LinearLayout.LayoutParams imageLayoutParams =
new LinearLayout.LayoutParams(

getResources().getDimensionPixelSize(R.dimen.image_wi
dth), // Set width as defined in resources
                LinearLayout.LayoutParams.WRAP_CONTENT);
        imageView.setLayoutParams(imageLayoutParams);
// Set layout parameters
        imageView.setImageBitmap(allBitmaps.get(i)); //
Set image resource

        // Add ImageView to outer LinearLayout
        outerLinearLayout.addView(imageView);

        // Create a new LinearLayout (vertical)
        LinearLayout verticalLinearLayout = new
LinearLayout(this);
        LinearLayout.LayoutParams verticalLayoutParams =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);

verticalLinearLayout.setLayoutParams(verticalLayoutPara
ms); // Set layout parameters

verticalLinearLayout.setOrientation(LinearLayout.VERTICA
L); // Set orientation

        // Create a new TextView
        TextView textView = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams.setMargins(20, 10, 0, 0); // Set
margin start and top
        textView.setLayoutParams(textLayoutParams); //
Set layout parameters
        textView.setText("Banana " + (i + 1)); // Set text

textView.setTextColor(getResources().getColor(R.color.bla
ck)); // Set text color

textView.setTextSize(TypedValue.COMPLEX_UNIT_SP, 20);
// Set text size
        textView.setTypeface(null, Typeface.BOLD); // Set
text style to bold

        // Add TextView to inner LinearLayout
        verticalLinearLayout.addView(textView);

        // Create and add the second TextView
        TextView textView2 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams2 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams2.setMargins(20,10, 0, 0); // Set
margin start
        textView2.setLayoutParams(textLayoutParams2); //
Set layout parameters
        textView2.setText("Type: " + allTypes.get(i)); // Set
text

textView2.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView2.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the second TextView to inner LinearLayout
        verticalLinearLayout.addView(textView2);

        // Create and add the third TextView
        TextView textView3 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams3 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams3.setMargins(20,10, 0, 0); // Set
margin start
        textView3.setLayoutParams(textLayoutParams3); //
Set layout parameters
```

```
        textView3.setText("Level: " + allRipeness.get(i)); //
Set text

textView3.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView3.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the third TextView to inner LinearLayout
        verticalLinearLayout.addView(textView3);

        TextView textView5 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams5 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams5.setMargins(20,10, 0, 0); // Set
margin start and top
        textView5.setLayoutParams(textLayoutParams5); //
Set layout parameters
        textView5.setText("Banana Count: " +
allCount.get(i)); // Set text

textView5.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView5.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout.addView(textView5);

        TextView textView6 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams6 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams6.setMargins(20,10, 0, 0); // Set
margin start and top
        textView6.setLayoutParams(textLayoutParams6); //
Set layout parameters
        textView6.setText("Confidence Level: " +
allConfidence.get(i) + "%"); // Set text

textView6.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView6.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout.addView(textView6);

        // Create and add the fourth TextView
        TextView textView4 = new TextView(this);
        LinearLayout.LayoutParams textLayoutParams4 =
new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
        textLayoutParams4.setMargins(20,10, 0, 0); // Set
margin start and top
        textView4.setLayoutParams(textLayoutParams4); //
Set layout parameters
        textView4.setText("Time: " + allDate.get(i)); // Set
text

textView4.setTextColor(getResources().getColor(R.color.bl
ack)); // Set text color

textView4.setTextSize(TypedValue.COMPLEX_UNIT_SP,
13); // Set text size

        // Add the fourth TextView to inner LinearLayout
        verticalLinearLayout.addView(textView4);


        // Add vertical LinearLayout to outer LinearLayout
        outerLinearLayout.addView(verticalLinearLayout);

        // Add inner LinearLayout to outer LinearLayout
        outerLinearLayout.addView(innerLinearLayout);

        // Add outer LinearLayout to the CardView
        cardView.addView(outerLinearLayout);

        // Add the CardView to the container layout
        containerLayout.addView(cardView);
    }

    bck_btn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(peel_history.this,
MainActivity.class);
            startActivity(intent);
        }
    });
  }
  private void history_trash(){

    Handler handler = new
Handler(Looper.getMainLooper());
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            final Dialog dialog = new
Dialog(peel_history.this);

dialog.requestWindowFeature(Window.FEATURE_NO_TIT
LE);
            dialog.setCancelable(true);
            dialog.setContentView(R.layout.dialog_trash);
            dialog.show();

            TextView yes = dialog.findViewById(R.id.yes);
            TextView no = dialog.findViewById(R.id.no);

            yes.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    DataHandler.clearData();
                    DataHandler.clearData2();
                    DataHandler.clearData3();
                    DataHandler.clearData4();
                    containerLayout.removeAllViews();
                    dialog.dismiss();
                }
            });

            no.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    dialog.dismiss();
                }
            });
```

```
        }
    },100);
  }
}
```

about.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  xmlns:tools="http://schemas.android.com/tools"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".about"
  >

  <androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/green"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme">

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/arrow_back"
      android:layout_alignParentStart="true"
      android:id="@+id/about_back"/>


  </androidx.appcompat.widget.Toolbar>

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/gray"
    android:paddingHorizontal="30dp">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/about"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="40dp"
      android:textSize="30dp"
      android:textColor="@color/light_black"
      android:textStyle="bold"
      />
    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/about1"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="90dp"
      android:textSize="14dp"
      android:textColor="@color/light_black"
      />

    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/about2"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="140dp"
      android:textSize="14dp"
      android:textColor="@color/light_black"
      />

    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/about3"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="220dp"
      android:textSize="14dp"
      android:textColor="@color/light_black"
      />

    <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginTop="50dp">
    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/peel_team"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="290dp"
      android:textSize="20dp"
      android:textColor="@color/light_black"
      android:textStyle="bold"
      />

    <ImageView
      android:layout_width="250dp"
      android:layout_height="200dp"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="285dp"
      android:src="@drawable/peel_team"
      />
    </RelativeLayout>
  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="bottom"
    android:paddingBottom="50dp">
    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/contact_us"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="520dp"
      android:textSize="16dp"
      android:textColor="@color/light_black"
      android:textStyle="bold"
      />
    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/contact_info"
      android:layout_centerHorizontal="true"
      android:layout_alignParentTop="true"
      android:layout_marginTop="550dp"
      android:textSize="12dp"
      android:textColor="@color/light_black"
      />
    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
        android:text="@string/contact_info1"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="565dp"
        android:textSize="12dp"
        android:textColor="@color/light_black"
        />
      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/email"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="580dp"
        android:textSize="12dp"
        android:textColor="@color/blue"
        />
   </RelativeLayout>
   </RelativeLayout>
</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:background="@color/gray"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
  <include layout="@layout/toolbar"/>
   <include layout="@layout/cards"/>
   <include layout="@layout/bottom_bar"/>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

bottom_bar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_gravity="bottom"
   tools:context=".MainActivity">

<com.google.android.material.bottomappbar.BottomApp
Bar
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_gravity="bottom"
      android:id="@+id/bottomAppBar"
      app:fabCradleMargin="7dp"
      app:fabCradleRoundedCornerRadius="50dp"
      app:fabCradleVerticalOffset="5dp"
      app:fabAlignmentMode="center"
      app:fabAnchorMode="cradle"
      android:backgroundTint="@color/white"
      >

</com.google.android.material.bottomappbar.BottomApp
Bar>

<com.google.android.material.floatingactionbutton.Floati
ngActionButton
      android:id="@+id/camera_button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:backgroundTint="@color/green"

android:contentDescription="@string/add_new_photo"

android:src="@drawable/baseline_photo_camera_24"
      app:layout_anchor="@+id/bottomAppBar"
      app:fabCustomSize="65dp"

app:shapeAppearanceOverlay="@style/FabShapeStyle"
      />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

bottom_camera.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:gravity="bottom"
   >
   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="83dp"
      android:layout_marginTop="7dp"
      android:background="@color/green">

      <ImageButton
         android:id="@+id/camera_button"
         android:layout_width="wrap_content"
         android:layout_height="50dp"
         android:layout_marginStart="80dp"
         android:layout_centerVertical="true"
         android:background="@color/green"
         android:src="@drawable/camera_png" />

      <ImageButton
         android:id="@+id/upload_button"
         android:layout_width="wrap_content"
         android:layout_height="48dp"
         android:background="@color/green"
         android:layout_alignParentEnd="true"
         android:layout_marginEnd="80dp"
         android:layout_centerVertical="true"
         android:src="@drawable/file_upload"

         />

   </RelativeLayout>
</RelativeLayout>
```

bottom_nav.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:gravity="bottom">
```

```
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="83dp"
        android:layout_marginTop="7dp"
        android:background="@color/white"
        android:id="@+id/container">


        <ImageButton
            android:layout_width="35dp"
            android:layout_height="30dp"
            android:layout_centerVertical="true"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:background="@drawable/add_photo"
            android:id="@+id/new_photo"

android:contentDescription="@string/add_new_photo">

        </ImageButton>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="25dp"
            android:layout_marginTop="55dp"
            android:layout_marginRight="25dp"
            android:id="@+id/new_photo_text"

            android:text="@string/add_new_photo">

        </TextView>

        <Button
            android:id="@+id/save_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="25dp"
            android:backgroundTint="@color/green"
            android:drawableLeft="@drawable/add_icon"
            android:paddingStart="10dp"
            android:paddingEnd="20dp"
            android:textAllCaps="false"
            android:textColor="@color/white"
            android:textSize="17sp"
            tools:ignore="RelativeOverlap" />

    </RelativeLayout>
</LinearLayout>
```

bottom_nav_output.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="bottom">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="83dp"
        android:layout_marginTop="7dp"
        android:background="@color/white"
        android:id="@+id/container">


        <ImageButton
            android:layout_width="35dp"
            android:layout_height="30dp"
            android:layout_centerVertical="true"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:background="@drawable/add_photo"
            android:id="@+id/new_photo"

android:contentDescription="@string/add_new_photo">

        </ImageButton>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="25dp"
            android:layout_marginTop="55dp"
            android:layout_marginRight="25dp"
            android:id="@+id/new_photo_text"

            android:text="@string/add_new_photo">

        </TextView>

        <Button
            android:id="@+id/save_button"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="25dp"
            android:backgroundTint="@color/green"
            android:paddingStart="10dp"
            android:paddingEnd="20dp"
            android:text="Update"
            android:textAllCaps="false"
            android:textColor="@color/white"
            android:textSize="17sp"
            tools:ignore="RelativeOverlap" />

    </RelativeLayout>
</LinearLayout>
```

camera_monitor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

xmlns:androidx.camera="http://schemas.android.com/ap
k/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".camera_ui">

    <androidx.camera.view.PreviewView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/cameraPreview"
        />

    <ImageView
```

```
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:src="@drawable/qr_box_white"
      android:layout_marginBottom="100dp"
      />


    <ImageButton
      android:layout_width="40sp"
      android:layout_height="40sp"
      android:layout_alignParentEnd="true"
      android:layout_margin="10sp"
      android:src="@drawable/flash_off"
      app:tint="#fff"
      android:id="@+id/toggleFlash"

android:background="?attr/selectableItemBackgroundBor
derless"
      />

    <ImageButton
      android:layout_width="40sp"
      android:layout_height="40sp"
      android:layout_alignParentStart="true"
      android:layout_margin="10sp"
      android:src="@drawable/close_cam"
      android:id="@+id/close_cam"
      app:tint="#fff"

android:background="?attr/selectableItemBackgroundBor
derless"/>

    <ImageButton
      android:layout_width="40sp"
      android:layout_height="40sp"
      android:layout_below="@+id/toggleFlash"
      android:layout_marginEnd="10sp"
      android:layout_alignParentEnd="true"
      android:id="@+id/flipCamera"
      android:src="@drawable/baseline_360_24"

android:background="?attr/selectableItemBackgroundBor
derless"
      app:tint="#fff"/>

    <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="120dp"
      android:layout_alignBottom="@id/cameraPreview"
      android:background="@color/white"
      >
      <ImageButton
        android:layout_width="80sp"
        android:layout_height="80sp"
        android:layout_centerHorizontal="true"

        android:layout_centerVertical="true"
        android:id="@+id/capture"
        android:src="@drawable/cam_btn"
        android:scaleType="centerCrop"
        android:background="@color/white"/>

      <Button
        android:id="@+id/btnGallery"
        android:layout_width="wrap_content"

        android:layout_height="40dp"
        android:layout_centerVertical="true"
        android:layout_marginStart="30dp"
        android:backgroundTint="@color/green"
        android:text="Upload"

        android:textAllCaps="false"
        android:textColor="@color/white"
        android:textSize="12sp"
        android:textStyle="bold" />

      <ImageButton
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_marginStart="64dp"
        android:layout_centerVertical="true"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="50dp"
        android:background="@color/white"
        android:scaleType="centerCrop"
        android:src="@drawable/tips_btn"
        android:id="@+id/showPopup"/>


    </RelativeLayout>
</RelativeLayout>
```

cards.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingTop="100dp"
  android:paddingLeft="20dp"
  android:paddingRight="20dp">

 <androidx.cardview.widget.CardView
   android:layout_width="match_parent"
   android:layout_height="110dp"
   app:cardCornerRadius="10dp"
   android:id="@+id/peel_card">

   <RelativeLayout
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:background="@drawable/bg1"
     android:orientation="vertical">

     <RelativeLayout
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:orientation="vertical">

       <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="@string/perfect_history"
         android:textColor="@color/black"
         android:textStyle="bold"
         android:textSize="17sp"
         android:layout_marginStart="20dp"
         android:layout_marginTop="10dp" />

       <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginStart="20dp"
         android:layout_marginTop="45dp"
```

```
        android:text="@string/history_content"
        android:textColor="@color/black"
        android:textSize="14sp"
        />

    </RelativeLayout>

  </RelativeLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
   android:layout_width="match_parent"
   android:layout_height="110dp"
   app:cardCornerRadius="10dp"
   android:layout_marginTop="20dp"
   android:id="@+id/about_card">

  <RelativeLayout
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:background="@drawable/bg1"
     >

    <RelativeLayout
       android:layout_width="wrap_content"
       android:layout_height="match_parent"
       >

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="About"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="17sp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp" />

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:text="@string/about_content"
        android:layout_marginTop="45dp"
        android:textColor="@color/black"
        android:textSize="14sp" />

    </RelativeLayout>

  </RelativeLayout>
</androidx.cardview.widget.CardView>

<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="30dp"
   android:text="Get Started"
   android:textSize="20sp"
   android:textStyle="bold"
   android:textColor="@color/light_black" />

<androidx.cardview.widget.CardView
   android:layout_width="match_parent"
   android:layout_height="140dp"
   app:cardCornerRadius="10dp"
   android:layout_marginTop="10dp">

  <LinearLayout
     android:layout_width="match_parent"
     android:layout_height="match_parent"

android:background="@drawable/banana_getstarted"
     android:orientation="vertical">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="match_parent"
      android:layout_margin="10dp"
      android:gravity="bottom"
      android:paddingStart="10dp"
      android:text="Classify the banana with Peel
Perfect"
      android:textColor="@color/white"
      android:textSize="14sp" />

  </LinearLayout>
 </androidx.cardview.widget.CardView>

</LinearLayout>
```

dialog_popup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:paddingTop="35dp"
   android:paddingBottom="30dp"
   android:paddingLeft="30dp"
   android:paddingRight="30dp"
   android:background="@color/white">

   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      >
      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Tips"
        android:textStyle="bold"
        android:textSize="22sp"
        android:layout_centerHorizontal="true"
        android:textColor="@color/black"
        />
   </RelativeLayout>

   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      >
      <ImageView
        android:layout_width="300dp"
        android:layout_height="199dp"
        android:layout_centerHorizontal="true"
        android:src="@drawable/banana_check" />
   </RelativeLayout>


   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="10dp">

      <ImageView
        android:id="@+id/imageView2"
```

```
    android:layout_width="110dp"
    android:layout_height="70dp"
    android:layout_alignParentStart="true"
    android:src="@drawable/banana_wrong1" />

  <ImageView
    android:id="@+id/imageView3"
    android:layout_width="110dp"
    android:layout_height="70dp"
    android:layout_alignParentEnd="true"
    android:src="@drawable/banana_wrong2" />

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView2"
    android:layout_alignEnd="@+id/imageView2"
    android:layout_alignParentStart="true"
    android:layout_marginStart="2dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="3dp"
    android:paddingStart="40dp"
    android:paddingEnd="40dp"
    android:text="Blur" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView3"
    android:layout_alignStart="@+id/imageView3"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="16dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="16dp"
    android:text="Many Object" />


</RelativeLayout>



<RelativeLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:paddingTop="20dp">

  <ImageView
    android:id="@+id/imageView4"
    android:layout_width="110dp"
    android:layout_height="70dp"
    android:layout_alignParentStart="true"
    android:src="@drawable/banana_wrong3" />

  <ImageView
    android:id="@+id/imageView5"
    android:layout_width="110dp"
    android:layout_height="70dp"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="1dp"
    android:src="@drawable/banana_wrong4" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView4"

    android:layout_alignEnd="@+id/imageView4"
    android:layout_alignParentStart="true"
    android:layout_marginStart="24dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="25dp"
    android:text="Too close" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView5"
    android:layout_alignStart="@+id/imageView5"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="32dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="33dp"
    android:text="Too far" />

  </RelativeLayout>

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp">
    <Button
      android:id="@+id/continueBtn"
      android:layout_width="wrap_content"
      android:layout_height="40dp"
      android:layout_centerHorizontal="true"
      android:paddingStart="40dp"
      android:paddingEnd="40dp"
      android:text="Continue"
      android:textAllCaps="false"
      android:backgroundTint="@color/green"
      android:textColor="@color/white"
      android:textSize="12sp"
      />
  </RelativeLayout>
</LinearLayout>
```

dialog_trash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:orientation="vertical"
  android:padding="25dp"
  android:background="@color/gray">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:text="Are you sure you want to delete all the
history?">
  </TextView>

  <RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">
  <TextView
    android:id="@+id/no"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="55dp"
    android:layout_alignParentEnd="true"
    android:background="@color/green"
```

```
    android:textStyle="bold"
    android:paddingVertical="5dp"
    android:paddingHorizontal="14dp"
    android:textSize="20dp"
    android:textColor="@color/white"
    android:text="No">
  </TextView>

  <TextView
    android:id="@+id/yes"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/white"
    android:background="@color/red"
    android:layout_alignParentEnd="true"
    android:textStyle="bold"
    android:paddingVertical="5dp"
    android:paddingHorizontal="10dp"
    android:textSize="20dp"
    android:text="Yes">
  </TextView>

  </RelativeLayout>

</LinearLayout>
```

dialog_trash_output.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:orientation="vertical"
  android:padding="25dp"
  android:background="@color/gray">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:text="Are you sure you want to delete?">
  </TextView>

  <RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">
  <TextView
    android:id="@+id/no"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="55dp"
    android:layout_alignParentEnd="true"
    android:background="@color/green"
    android:textStyle="bold"
    android:paddingVertical="5dp"
    android:paddingHorizontal="14dp"
    android:textSize="20dp"
    android:textColor="@color/white"
    android:text="No">
  </TextView>

  <TextView
    android:id="@+id/yes"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/white"
    android:background="@color/red"
    android:layout_alignParentEnd="true"
    android:textStyle="bold"
    android:paddingVertical="5dp"
    android:paddingHorizontal="10dp"
    android:textSize="20dp"
    android:text="Yes">
  </TextView>

  </RelativeLayout>

</LinearLayout>
```

history_content.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    android:orientation="vertical"
    android:id="@+id/container_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="100dp"
    android:paddingLeft="30dp"
    android:paddingRight="30dp">
  </LinearLayout>
</ScrollView>
```

history_monitor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:background="@color/gray"
  >
  <include layout="@layout/history_content"/>

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:background="@color/green">


    <TextView
      android:id="@+id/history_title"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginStart="35dp"
      android:textSize="20sp"
      android:textColor="@color/white"
      android:textStyle="bold"/>

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/arrow_back"
```

```
        android:layout_alignParentStart="true"
        android:id="@+id/history_back"/>
    <ImageView
        android:id="@+id/history_trash"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/trash"
        android:layout_alignParentEnd="true"/>

  </RelativeLayout>
</RelativeLayout>
```

history_toolbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:background="@color/gray"
  >
  <include layout="@layout/history_content"/>

  <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:background="@color/green">


    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/perfect_history"
      android:layout_marginStart="35dp"
      android:textAllCaps="true"
      android:textSize="20sp"
      android:textColor="@color/white"
      android:textStyle="bold"/>

    <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/arrow_back"
      android:layout_alignParentStart="true"
      android:id="@+id/history_back"/>

    <ImageView
      android:id="@+id/history_trash"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/trash"
      android:layout_alignParentEnd="true"/>

  </RelativeLayout>
</RelativeLayout>
```

loading.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:orientation="vertical"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:paddingVertical="10dp"
  android:paddingHorizontal="30dp"
  android:background="@color/gray">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Processing..."
    android:textSize="20sp"
    android:textColor="@color/light_black">
  </TextView>

  <ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:visibility="gone"
    android:indeterminate="true"/>
</LinearLayout>
```

main_bottom_navbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <FrameLayout
    android:layout_width="0dp"
    android:layout_height="0dp"

app:layout_constraintBottom_toBottomOf="@+id/mainB
ottomNav"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

  </FrameLayout>

<com.google.android.material.bottomnavigation.BottomN
avigationView
    android:background="@color/green"
    android:id="@+id/mainBottomNav"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

output_monitor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:layout_editor_absoluteX="1dp"
  tools:layout_editor_absoluteY="1dp">

  <!-- AppBarLayout to fix the Toolbar at the top -->
  <com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:theme="@style/ThemeOverlay.AppCompat.Actio
nBar">

    <androidx.appcompat.widget.Toolbar
      android:id="@+id/toolbar"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="@color/green"
      android:minHeight="?attr/actionBarSize"
      android:theme="?attr/actionBarTheme">

      <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/arrow_back"
        android:layout_alignParentStart="true"
        android:id="@+id/output_back"/>

    </androidx.appcompat.widget.Toolbar>
  </com.google.android.material.appbar.AppBarLayout>

  <!-- Scrollable content -->
  <ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="56dp"

app:layout_behavior="@string/appbar_scrolling_view_be
havior">

    <LinearLayout
      android:id="@+id/main"
      android:background="@color/gray"
      android:orientation="vertical"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      tools:context=".output">

      <!-- banana image-->
      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="500dp">
        <ImageView
          android:id="@+id/output_img"
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          android:scaleType="centerInside" />
      </LinearLayout>

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="140dp"
        android:background="@drawable/border"
        android:layout_marginTop="-20dp"
        android:id="@+id/banana_main_details">

        <!-- types and ripeness -->
        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_marginTop="25dp"
          android:paddingLeft="20dp"
          android:paddingRight="20dp"
          android:textSize="20sp"
          android:textStyle="bold"
          android:id="@+id/banana_type">
        </TextView>

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_marginTop="60dp"
          android:paddingLeft="20dp"
          android:paddingRight="20dp"
          android:textSize="20sp"
          android:textStyle="bold"
          android:id="@+id/ripeness_level">
        </TextView>

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_marginTop="95dp"
          android:paddingLeft="20dp"
          android:paddingRight="20dp"
          android:textSize="20sp"
          android:textStyle="bold"
          android:id="@+id/confidence_level">
        </TextView>
      </RelativeLayout>

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="58dp"
        android:layout_marginTop="7dp"
        android:background="@color/white"
        android:id="@+id/basic_information_layout">


        <TextView
          android:layout_width="wrap_content"
          android:layout_height="match_parent"
          android:text="@string/basic_info"
          android:textSize="17sp"
          android:textColor="@color/green"
          android:gravity="center_vertical"
          android:paddingLeft="20dp"
          android:paddingRight="20dp"
          android:id="@+id/basic_info">

        </TextView>

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="match_parent"
          android:text="@string/recommendation"
          android:textSize="17sp"
          android:gravity="center_vertical"
          android:layout_marginStart="160dp"
          android:layout_marginEnd="1dp"
          android:paddingLeft="20dp"
          android:paddingRight="20dp"
          android:id="@+id/recommendation">

        </TextView>

      </RelativeLayout>

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="350dp"
        android:layout_marginTop="7dp"
        android:background="@color/white"
        android:id="@+id/banana_info">

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:textSize="17sp"
```

```
            android:paddingTop="14dp"
            android:paddingLeft="20dp"
            android:paddingRight="20dp"
            android:textStyle="bold"
            android:id="@+id/banana_count">
        </TextView>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:padding="20dp"
            android:textSize="15sp"
            android:layout_marginTop="15dp"
            android:id="@+id/banana_content">

        </TextView>
      </RelativeLayout>
    </LinearLayout>
  </ScrollView>

  <!-- Bottom Navigation Bar fixed at the bottom -->
  <include
    android:id="@+id/bottom_nav"
    layout="@layout/bottom_nav_output"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"/>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

toolbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:background="@color/green">

  <RelativeLayout
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp">


    <ImageView
      android:layout_width="match_parent"
      android:layout_height="20dp"
      android:src="@drawable/peel_perfect"
      android:layout_centerVertical="true"/>

  </RelativeLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```