IntelliCART

PROJECT SUPERVISOR

Dr. Muhammad Farrukh Shahid

PROJECT TEAM

Syed Ali Jodat Naqvi	20K-0155
Mohammad Basil Ali Khan	20K-0477
Abdul Ahad Shaikh	20K-0319

Submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES
KARACHI CAMPUS

MAY 2024

Project Supervisor	Dr. Muhammad Farrukh Shahid
Project Team	Syed Jodat Ali Naqvi 20K-0155 Basil Ali Khan 20K-0477 Abdul Ahad Shaikh 20K-0319
Submission Date	2nd May 2024

Dr. Muhammad Farrukh Shahid __________________________________
Supervisor

Dr. Atif Tahir __________________________________
Head of Department

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Acknowledgement

We would like to express gratitude to our supervisor Dr. Muhammad Farrukh Shahid for his invaluable guidance, support, and motivation throughout final year project. Without his expertise, insights, and motivation, this work would not have been possible.

We would like to express my sincere gratitude to the members of the jury Dr. Muhammad Waqas, Mr. Abuzar Zafar for their invaluable time, effort in reviewing my final year project – I report. Your constructive feedback, comments, and suggestions were critical in improving the quality and impact of my work.

We would also like to express my appreciation to FAST NUCES, Karachi and for providing the resources and facilities necessary to conduct this research. The use of the FYP Lab was essential to the completion of this project.

Thank you all for your support and guidance.

Abstract

The freshness and quality of products such as fruits and vegetables have become valuable to customers in today's continuous evolving world and global market. So our solution referred to as "IntelliCART" that aims to enhance customer trust and convenience while purchasing fruits and vegetable from a cart. Our idea includes technologies including Artificial Intelligence, Deep Learning, Mobile Application development, Global Positioning System (GPS) to provide a platform that offers real-time monitoring of fruit and vegetable Quality Assessment, Price Validation set by government and location-based services for finding nearby-by fruit and vegetable carts. The proposed solution does not only benefit customers, but also to fruit and vegetable cart vendors. By using this, vendors can construct a relationship with their customers by ensuring a pleasant interaction and transparency expanding their consumer base. Furthermore, adhering to government pricing tips can help companies avoid headaches while retaining their reputation. In summary, the "IntelliCART" addresses issues associated with pleasant buying and selling experience and truthful pricing from fruit and vegetable carts. With its freshness assessment characteristic, Government Price Verification System, and user Friendly Mobile Application, along with Global Positioning System (GPS) -based location offerings, it gives a solution that empowers purchasers to make choices while helping nearby vendors. This project brings together technology, agriculture and consumer awareness to create a more transparent and effective fresh produce market.

Table of Contents

List of Figures

List of Tables

1. Introduction

The world's urge for food for clean and healthy meals has increased in recent years as people around the world have come to appreciate the importance of healthy consumption and living a wholesome lifestyle. fruits and vegetables containing nutrients and dietary fiber, an important supply of the human food plan [1]. However, making sure that fruits and veggies are fresh and actual has emerged as a greater project, especially when you're shopping from a fruit and vegetable cart that doesn't have the infrastructure of a conventional retail outlet. Many clients are uncertain about the cost of the fruits and vegetable they're about to shop for. To be able to address those issues and increase transparency in the marketplace for fresh produce, we're introducing a new idea: IntelliCART.

The proposed project makes use of cutting-edge technology, such as Computer vision, Deep Learning, and Mobile Application, as well as Global Positioning System (GPS), to offer purchasers real-time statistics on fruit and vegetable freshness, validate government-selected pricing, and find nearby carts within the area. By combining these functions into a single platform, the goal is to facilitate, help clients make informed shopping choices, and make the fresh produce market more green and reliable.

This notion outlines the primary factors and capabilities of the system and discusses its benefits for customers and cart vendors. The integration of modern technologies with the agriculture and retail industries illustrates the cross-cutting nature of this project. For farmers and the general public, the sophisticated agricultural fruit recognition system with an easy-to-use camera will serve an amazing purpose [2]. With the aid of improving consumer stories and empowering neighborhood providers, the "IntelliCART," is consistent with the wider objectives of promoting healthy consumption styles and supporting nearby economies.

1.1 Background

As the technology is evolving agriculture industry also evolved immensely. This evolvement open doors to many aspects of agriculture such farming, harvesting, and supply chain. Fruits and vegetable have vitamins, minerals and dietary, an important source of the human everyday diet [3]. The quality assessment of fruits and vegetables results a critical factor for both seller and customers. Traditionally the assessment of fruits and vegetable quality heavily relies on experienced manual labor inspection which is time consuming, subjective, and often prone to errors. This way of assessment also results in maintaining consistency, and scalability especially in large scale operations. The emergence of innovative technologies particularly in field of Artificial Intelligence, Machine Learning, and Computer Vision has introduced a way of to revolutionize the assessment of fruits and vegetables. This offer potential to automate and streamline the assessment process, ensuring high accuracy, and scalability

1.2 Importance of Quality Assessment

Fruit and vegetable quality assessment in today's marketplace hold large important across different aspects including customer satisfaction, ensuring food safety, preferences, health, and impacting economy as well

1.2.1 Customer Preferences and Expectations

Fresh, visual appealing, best quality fruits and vegetable is likely to be attract customers when making their purchases. Thus quality assessment plays an important role in meeting these expectations as it provides consumer with satisfaction regarding overall condition of fruits and vegetables. The quality of fruits and vegetable impact consumer perception of values. Fruits and vegetable that are fresh and have high quality tend to fetch better prices and are preferred by customers resulting in increased sales and customer satisfaction. Delivery of high quality fresh produce results in customer loyalty and repeat business. When customers trust the quality of fruits and vegetable purchased they more likely to return to same seller.

1.2.2 Impact on Food Safety and Health

Fresh and high quality fruits and vegetables holds more nutrients contributing to better health outcomes. Poor quality, contaminated or spoiled poses great risk and threat to health of customers. Effective high quality assessment helps in segregating such risks. Thus results in safety from risks caused by consuming low quality fruits and vegetables.

1.2.3 Economic Significance

High quality fruits and vegetables results in better marketability and prices. Vendors selling high quality fruits and vegetable attract more customers and can sell them at good rates resulting in good revenue. Effective quality assessment contributes to more efficient supply chain. By accurately assessing quality it can streamline distribution, reduce inventory holding costs and ensures quality items reach the consumer. Improved quality assessment practices lead to more feasible agriculture practices, improved profitability for sellers and better resource utilization.

1.3 Challenges of Quality Assessment

There are several challenges affecting efficiency and accuracy of assessment and evaluation when assessing fruits and vegetables quality. These challenges involve subjectivity, time consuming and labor intensive indicating importance of more objective and efficient solution.

1.3.1 Subjectivity and Variability

Human assessment of quality assessment of fruits and vegetable is subjective. Difference of judgement exist in assessing which may result in inconsistent grading and classification. Standards can differ in multiple regions, markets, and cities. This differences of standardization results in variability. Fruits and vegetables have natural colors, size, texture, and other attributes variations making it challenge for making standardization criteria for quality assessment.

1.3.2 Time and Labor Intensive

Manual inspection which is time consuming and requires more workforce for large scale operations. This involve sorting, grading each fruit individually leading to time wastage, labor costs and operational difficulties. In areas such as markets the manual assessment of large quantity of fruits and vegetable becomes impractical and not efficient resulting in bottlenecks and delays.

1.3.3 Need for Objective and Efficient Solution

There is need for standardized methods that can ensure standard quality assessment criteria in different markets. The market demands real time evaluation of quality of fruits and vegetable minimize delays in distribution and ensure fresh fruits and vegetables reaching consumers. Manual methods fail in meeting these demands. Using machine learning, artificial intelligence and computer vision can result in solutions for efficient and automated quality assessment These technologies have potential to standardized assessment criteria and reduce human subjectivity.

1.4 Role of Computer Vision and Deep Learning

In agriculture sector the integration of computer vision and deep learning technologies provide transformative solution for fruits and vegetable quality assessment. These technologies provide approach for automation, precision, improved efficiency and accuracy resulting in showing immense potential for integration with industry. Deep neural networks and the most recent advancements in computer vision technology can be employed for semantic picture division and object detection [4].

1.4.1 Automation and Precision

Computer vision provides a way to automate analysis of images of fruits and vegetables allowing for processing large quantity. From that algorithms can identify and extract relevant features such as color, texture, size and defects without human input. By using defined algorithms and computer vision systems we can facilitate standardized evaluation methods minimizing subjectivity. Deep learning models trained on extensive dataset can accurately classify fruits and vegetables based on quality parameters. These models can differentiate between differences in quality such as ripeness, blotch or bruised with high accuracy.

1.4.2 Enhancing Efficiency and Accuracy

Computer vision system equipped with deep learning models can process images at high speed reducing time for quality assessment. Rapid processing enhances efficiency of assessment process. Deep learning models continuously learn and adapt on data leading to improved accuracy over time. As these models are trained on diverse dataset they can accurately identify and classify a wide range of quality parameters that cannot be done through human capabilities.

1.4.3 Potential for Industry Integration

Integrating computer vision system into fruits and vegetables supply chain can optimize various stages from production to distribution. Real time quality assessment allows timely assessing quality and ensuring delivery of high quality fruits and vegetables to customers. Beyond traditional markets computer vision and deep learning find application in different sectors including precision agriculture, food processing industries and online retail. These technologies enable quality assessment resulting evolving consumer demands.

1.5 Contribution to Society

Traditional cart vendor sells fruits at their own will and the customer don't have idea to either trust with quality and prices with vendors. Our proposed solution has significant contribution to society. Many customers do not have knowledge related to quality and freshness of fruits and vegetables, our project ensures that customer have access to high quality fresh fruits and vegetable by providing real time fresh assessments. using this functionality customer can make informed decisions. Second many vendors sell fruits and vegetables on their own choice of prices and not keeping in regard the government mandated prices so our solution promotes fairness and transparency in pricing of fruits and vegetables by inducing government allotted prices. Proposed project enhances transparency in market and customer can access information about freshness and pricing and promoting fair competition among vendors. This will increase trust between customer and vendors. For customer new in an area does not have idea where to find carts to buy fruits and vegetable so with the help of our proposed solution people can have location of those cart with quality assessment and fair pricing.

1.6 Objectives

- Assessment of quality of fruits and vegetables through image analysis using computer vision and deep learning.
- Training and evaluation of models i.e. MobileNet, Inception-V3, ResNet152, AlexNet, VGG-16, VGG-19 for fruit and vegetable classification.
- Comparing and evaluating the performance of each deep learning models across various classes of fruits and vegetables to determine their effectiveness in quality assessment based of defined metrics accuracy, recall and precision.
- Integrate the most effective model into our IntelliCART application for quality assessment of fruits and vegetables
- Integrating Government-Induced Pricing Validation.
- Implementing GPS-Based Location Services.
- Providing a user friendly and efficient solution for quality assessment ultimately leading to increased sales of vendors, increase customer satisfaction and loyalty and improved economic benefits in agriculture industry.

The goal of the project is to develop a complete and innovative "IntelliCART" that is beneficial to both consumers and sellers in the market.

2. Related Work

The state of the art in agriculture, and more specifically, the field of fruit and vegetable quality assessment, is presented in this chapter:

This paper [5] employs neural network techniques to automatically classify and detect fruits. This technology can be utilized in supermarkets, factories, and education for kids. It can also help people learn about different fruits. The following is a description of this work's main contributions: (i) An automatic fruit detection and classification system was created utilizing two datasets: a proprietary dataset consisting of eight fruit categories and the open-source FIDS-30 dataset, which has 30 classes. (ii) We employed ResNet50 and VGG16 neural network models for categorization. Deep learning frameworks YOLOv3 and YOLOv7 have been used to identify various fruits in the picture. (iii) The domain adaption technique is used to enable the suggested fruit categorization model based on deep learning to handle real-world issues across a variety of domains. This technique was used to train and evaluate the suggested model using multiple fruit image sets. (iv) A flask is used to help develop the web foundation for the suggested automatic fruit classification and detection system. (v) An Android smartphone application that instantly detects and recognizes fruits using the phone's camera was created using a Python-based API.

Based on an enhanced YOLOv4 model, this work provides a deep-learning system for multiclass fruit and vegetable categorization that first determines the type of object in an image before classifying it into either fresh or rotten. The suggested approach entails building an image collection of fruits and vegetables, optimizing the YOLOv4 model, arguing data, and assessing performance. Moreover, the Mish activation function was used to improve the core of the suggested model in order to get faster and more accurate identification. A thorough experimental examination of the suggested method can yield a greater average precision than the original YOLOv4 and YOLOv3 with 50.4%, 49.3%, and 41.7%, respectively, in comparison to the preceding YOLO series [6]. The suggested approach can assist visually impaired individuals in selecting fresh food and preventing food illness. It also has excellent potential for developing an autonomous and real-time fruit and vegetable classification system for the food business and marketplaces.

This research focuses on multi-level grading combining Deep Learning, Computer Vision, and Image Processing approaches to increase the accuracy of the automatic mango grading system [7]. In order to identify the mango variety and categorize according to quality, the suggested method is based on the mango maturity ripening stage, shape, textural features, color, and flaws. Convolutional neural networks are used to extract the mango's mature ripening stage (CNN). Shape, texture traits, and flaws are extracted using techniques from computer vision and image processing. In order to determine the mango variety and categorize the mango quality into three groups—Not fit, Average, and Good—the retrieved features are fed into the Random Forest classifier. The most popular types in Tamil Nadu, Banganapalli, Neelam, and Rumani, comprised the dataset created for this study, which was used to validate the system. The accuracy of the suggested system, which used CNN characteristics to increase its functionality, was 93.23% for variety recognition and 95.11% for quality grading. Because of this, the suggested approach is completely automated, economically feasible, and has better accuracy when it comes to variety identification and mango quality rating across several types.

CNN was utilized by Kamble, P. R. et al. (2020) to determine if the fruit in the image was ripe or raw [8]. The authors used a pre-trained network, such as VGG16, to accomplish this for three distinct fruit types: mango, banana, and apple. They classified mango fruit as either ripe or uncooked with 92% accuracy. With a 97.92% accuracy rate, Srinivasan, D. and Yousef, M. (2020) could determine if the apple fruit is rotting or fresh using the pre-trained ReNnet-50 CNN model [9]. In order to classify the 5031 photos, the scientists chose 2088 photographs of fresh apples and 2943 images of decaying apples.

In this research, [10] An innovative design based on deep learning With a test accuracy of 99.6%, Fruit-CNN has been proposed to identify the sort of fruit and evaluate its quality in real-world photos in different visual variants. When compared to the most advanced deep learning models, the suggested architecture requires the least amount of time to train a huge dataset and test fruit photos, demonstrating its broad applicability in precision agriculture. Less parameters can be used to train more photos from different classes, which speeds up model training and reduces processing time.

Recently, the multimodal Faster RCNN approach for fruit detection using RGB and near infrared images was developed by examining early and late fusion methods [11]. In a different study, M. Afonso et al. demonstrated a Mask-RCNN-based deep learning method for tomato fruit detection and counting in production greenhouses using Intel RealSense cameras [12]. A Faster R-CNN based object identification framework was used to practically construct the detection mechanism for the purpose of detecting fruit in orchards, such as apples, almonds, and mangoes [13]. An accuracy of 77.2% was achieved by a model that used ResNet-101 as its foundation to perform semantic segmentation on apples and apple branches [14]. Other approaches either make use of sensors like hyperspectral, 3D, or LWIR, etc., or fundamental computer vision techniques like color-based segmentation.

In a recently published study, Wu et al. [15] used an 11-layer, modified version of the AlexNet model to find and identify flaws in apples. Additionally, they compare the classification using three well-known algorithms: support vector machines (SVM), particle swarm optimization (PSO), and backpropagation neural networks (BP). The dataset is made up of 5472 × 3648-pixel speckle images that are caused by laser backscatter. A laser system with a beam expander, a complementary metal-oxide semiconductor (CMOS) color camera with a zoom lens, and a polarizer were used during the acquisition process. There are 500 apple samples in all, all roughly the same size (equatorial diameter: 80–100 mm). The suggested CNN model for apple detection outperforms other widely used algorithms like BP, SVM, and PSO algorithm, achieving a recognition rate of 92.50%.

Hou et al. [16] Proposed fine-grained visual classification (FGVC) using the VegFru dataset. VegFru is a domain-specific dataset with approximately 160,000 photos that spans 292 subordinate classes and 25 upper-level categories of fruits and vegetables. Additionally, they introduced a framework named HybridNet that uses the FGVC's label hierarchy to segregate categorization using two DCNNs. They contrast HybridNet for the 292, 100, and 200 subclasses of the VegFru Dataset with VGGNet and CBP-CNN. HybridNet performed better in every test, with VGGNet scoring 77.12%–84.46%–72.32%, CBP-CNN scoring 82.21%–87.49%–84.91%, and HybridNet scoring 83.51%–88.84%–85.78%.

Based on the automatic fruit quality inspection system, Manali R. Satpute and Sumati M. Jagdale (2016) developed a tomato defect detection system that allows for tomato sorting and grading as well as the identification of defective tomatoes. The tomato was segmented in the first step using the OTSU algorithm. Extraction of the feature comes next once the tomato has been segmented. Size detection and color detection are the two feature extraction algorithms used by the author. Size detection uses morphological operations such as dilation and erosion. Following that, size detection using form features—small, medium, and large—is employed. To sort tomatoes, employ color detection based on red, green, and yellow tomatoes [17].

In this research, [18] For the purpose of evaluating fruit quality, they have made use of the idea of densely connected convolutional neural networks, or denseNets. The network's ability to address vanishing gradient issues and guarantee feature reuse for the acquisition of significant insights has been made possible by feature propagation towards the deeper layers. Testing on a dataset of 19,526 photos with three quality levels for every fruit, the suggested pipeline produced an astounding 99.67% accuracy rate. The model's suitability for real-world applications was further demonstrated by its comparable performance on tasks including fruit classification and quality assessment, which further tested the model's robustness.

In this research, [19] When evaluating the quality of fruits and vegetables, they apply the idea of transfer learning. The theory of transfer learning involves using a pre-trained Convolutional Neural Network to address a new problem without requiring large-scale training datasets. To assess the quality of fruits and vegetables, eight pre-trained deep learning models—AlexNet, GoogleNet, ResNet18, ResNet50, ResNet101, Vgg16, Vgg19, and NasNetMobile—are adjusted appropriately. In order to assess each fine-tuned model's performance in training and validation, we gather a dataset made up of pictures from twelve different samples of fruits and vegetables. Over a period of five weeks, the dataset grows. The overall number of photographs throughout the course of five weeks is 350, and the total number of images in the dataset is 12*350, or 4200 images, since 70 images are gathered every week. There are a total of 60 classes (12*5) in the dataset. This dataset, as well as an enhanced version, served as the basis for the models' evaluation. According to the model's results, the ResNet18 model scored the highest validation accuracy based on the supplemented dataset with 91.37% accuracy, while the Vgg19 model achieved the maximum validation accuracy over the original dataset with 91.50% accuracy.

Table 1 Overview of most important papers

Published in year	Classes	Method	Features	Classifiers
2022	123 different classes	Object Detection using Deep Learning	Data Pre-processing and augmentation	U-Net ResNet VGG19
2022	6 Fruits and 6 Vegetables	Object Detection	Grids, Blocks, Union Over Intersection (IOU)	YOLOv4 - tiny
2022	Total 30 fruit classes	Convolutional Neural Networks	Object isolation, bounding boxes, feature vectors	ResNet50 VGG16 YOLOv3
2022	Apple, Banana, Guava, Lime, Orange, Pomegranate	Convolutional Neural Networks	Class Activation Map (CAM), texture and color	DenseNet
2021	Fig Fruit	Deep Learning and Object Detection	Colored bounding boxes, confidence threshold	Faster R-CNN YOLOv3 YOLOv4
IRJET 2021	Apple, banana, Mango Disease	K-means clustering technique to cluster the images	color, morphology, Color Coherence Vector (CCV)	Support Vector Machine (SVM).
ICIRCA 2020	Banana disease	identifying diseases in banana plants at an earlier stage to protect neighboring plants from the same diseases.	Feature extraction is primarily dependent on pattern recognition.	Artificial Neural Network (ANN)
2016	3 classes (background, sweet pepper, rock melon)	Deep Convolutional Neural Networks, Object Detection	RGB Colors, Near Infrared (NIR) images, bounding boxes	Faster R-CNN

Table 2 Comparisons with existing products

APPLICATION STARTUP NAMES	FEATURES				
	Fruit/Vegetable Prices	**Government Induces Prices**	**Freshness Monitoring using Computer vision**	**Nearby Cart Location**	**Have Application**
IntelliCART (Our FYP)	✓	✓	✓	✓	✓
FruitSabzi	✓	×	×	×	×
Farmette	✓	×	×	×	×
F2H – Farm to Home	✓	×	×	×	×
FreshSabz	✓	×	×	×	✓
Suzi.pk	✓	×	×	×	×

3. Requirements

3.1 Intended Audience

The primary audience for the "IntelliCART" or "Intelligent Rehri Wala" project would include:

1. **<u>Clients/Stakeholders</u>:** Individuals who have invested in the company. Fast University, the Supervisor and the students that are creating this project are stakeholders of this project.
2. **<u>End Users</u>:** Customers purchasing fruits and vegetables and the vendors selling fruits and vegetable using the technology.
3. **<u>Support and Maintenance Team</u>:** Individuals responsible for providing ongoing support and maintenance for the deployed IntelliCART system.

3.2 Operating Environment

The “IntelliCART” software requires mobile environment, supporting Android (version 7.0 and above) and iOS (version 11.0 and above) platforms. Requires mobile data connectivity, the application uses a cloud-based database for efficient data management. The software works by computer vision algorithms for real-time freshness assessment, GPS services for location-based functionalities. It is designed to seamlessly run on a variety of mobile devices.

3.3 Functional Requirements

3.3.1 Mobile Application Functions

3.3.1.1 User Interface

3.3.1.1.1 Customer

- ✓ Provide an intuitive interface for customers.
- ✓ Display real-time freshness assessments and pricing details.
- ✓ Comparison between government prices and offered prices.
- ✓ Provide customers to choose which cart they choose with the price they agree on.
- ✓ Feedback option for customers.
- ✓ Search for appropriate fruit and vegetable.

3.3.1.1.2 Vendor

- ✓ Provide an intuitive interface for vendors.
- ✓ Add, update, and delete fruit and vegetable that vendor selling
- ✓ Update his location
- ✓ Real time everyday freshness monitoring
- ✓ Add sales and get to know previous sales

3.3.1.2 Location Services

- ✓ Utilize GPS to locate nearby fruit carts.

- ✓ Offer navigation guidance to identified carts

3.3.2 Database Management

3.3.2.1 Data Storage and retrieval

- ✓ Store fruit and vegetable details.
- ✓ Manage government pricing information.
- ✓ Manage cart locations

3.3.2.2 Profile

- ✓ Store and update customer and vendor profiles.

3.3.3 Security and Authentication

3.3.3.1 Data Encryption

- ✓ Secure data transmission and storage

3.3.3.2 User Authentication

- ✓ Verify user identity for access to sensitive information.

3.4 Use Cases

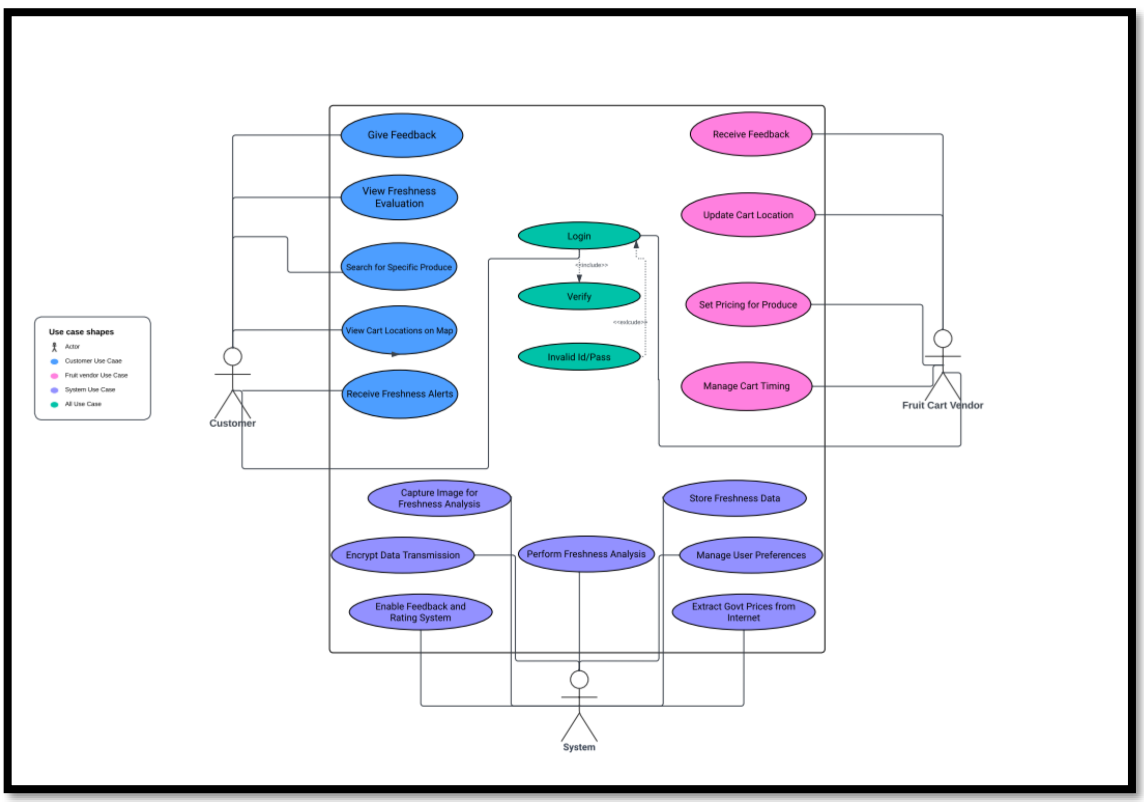


Figure 1 Use Case Diagram

This use case involves users accessing and viewing the freshness evaluation of fruits and vegetables available on nearby carts through the "IntelliCART" mobile application.

There are two primary Actors, Customers and Sellers, and a secondary actor, System.

The customer can view the quality of produce that the nearby carts offer. They can also search for specific produce, such as if they want to buy bananas, they can type it in the search bar. Customers can also view cart locations. Moreover, they can also give feedbacks of the quality that they received.

The Seller can receive feedback on their qualities offered, and can update their cart location. Additionally, they can set pricing for their produce as well.

The system is the secondary actor which will work between the seller and the customer. The overall system will capture image, perform evaluation analysis, encrypt data transmission while sending data to cloud-base database. Moreover, the system will extract daily updated government prices of the produce offered.

Table 3 Use Case table for Freshness Assessment

<table>
<tr><td colspan="3"><Use case Id: IntelliCART></td></tr>
<tr><td colspan="2">Use case Id:</td><td>UC-VFE-001</td></tr>
<tr><td colspan="3">Actors:
Primary: Customers and Cart Vendors (CCV).
Secondary: System.</td></tr>
<tr><td colspan="3">Feature: Real time freshness assessment.</td></tr>
<tr><td colspan="2">Pre-condition:</td><td>1. The user has installed the "IntelliCART" mobile application.
2. The user's location services are enabled to identify nearby carts.</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1</td><td>Customer View Freshness Evaluation</td><td>Extracts the quality that the cart offers.</td></tr>
<tr><td>2</td><td>Customer Gives Feedback</td><td>Shows feedback to Seller and displays to other customers.</td></tr>
<tr><td>3</td><td>Customer Search for specific Produce in search bar</td><td>Gives details of that produce.</td></tr>
</table>

4	Customer View Cart Location	Gives exact address of the fruit cart.
5	CCV Receive Feedbacks	Extracts customers feedback that they receive.
6	CCV updates cart location	Updates cart location for accurate location to customers.
7	CCV Set Pricing for produce	Updates prices of the produce offered by CCV.
8	System Capture image	Evaluates freshness assessment.
9	System Perform Freshness Analysis	Use Deep Learning Models to perform freshness assessment.
10	System Encrypt Data Transmission	Images sent from CCV to customer protected via safe data transfer connection.
11	System Extract Govt. Prices from internet	Extract data from latest uploaded document by government. and make prices available for all customers so that they can match prices offered by CCV and government price.
12	System Enable Feedback and Rating System	Make customer rating available to respect CCV.

Post Conditions	
Step#	***Description***
1	User has viewed the freshness evaluation of selected produce items from nearby carts
2	The customer's feedback is successfully submitted and stored in the system's database
3	The system successfully retrieves and displays results matching the customer's query for specific produce
4	The system shows a map displaying the locations of nearby carts based on selected location.
5	The system presents the freshness in a timely manner, displaying details such as the type of produce available, and cart locations.
6	The system successfully receives and records the feedback provided by customers into system database.
7	The system database reflects the updated location information for the vendor's cart accurately.
9	The updated prices for the produce are now visible to customers through the system's interface.

10	The system generates and stores freshness evaluation data for each produce item based on the analysis.
11	Data transmitted between the application and server is protected against unauthorized access.
12	The updated government prices for the produce are now visible to customers through the system's interface.
13	Users can access and provide feedback on produce quality and vendor service through system interface.

3.5 Non-Functional Requirements

3.5.1 Performance Requirement

3.5.1.1 Speed

The system should provide fast real time responses when end users interact with the system showing fast loadings, updating of cart location, product details.

3.5.1.2 Precision

Accurate in assessment of freshness of fruits and vegetable and product details to users, ensuring precise and reliable data for informed decision making.

3.5.1.3 Concurrency

Multiple user can interact concurrently allow seamless interaction, access and updates for various users browsing cart location.

3.5.1.4 Capacity

Large volume of users and carts data without compromising performance.

3.5.1.5 Safety

Safeguard user data, personal information, ensure encryption, authentication and data protection protocols.

3.5.1.6 Reliability

The system should operate consistently and predictably without unexpected downtime, ensuring high availability for users at all times.

3.5.2 Safety Requirements

- ✓ Implement strong encryption protocols to safeguard user data, preventing unauthorized access or data breaches.

- ✓ Regularly update security measures to mitigate potential risks.
- ✓ In accordance to privacy regulations by anonymizing sensitive information, obtaining user consent for data usage, and providing clear privacy policies.
- ✓ Implement User verification processes to prevent unauthorized access to sensitive data.
- ✓ Thoroughly test system updates or changes in a controlled environment before deployment to prevent unintended disruptions or vulnerabilities.

3.5.3 Security Requirements

- ✓ All user data, including personal information, and preferences, must be encrypted to prevent unauthorized access.
- ✓ Employ data integrity checks to ensure that information remains unaltered and intact during storage, safeguarding against data tampering.
- ✓ Assign different authorization levels to users (e.g., customers, vendors) to control access rights within the system.
- ✓ In accordance to ISO 27001 standards for information security management.

3.5.4 User Documentation

- ✓ Comprehensive guide outlining system functionalities, navigation, and instructions for users to effectively utilize the application.
- ✓ Reference document assisting users in resolving common issues or errors triggered while using the system.
- ✓ Details regarding customer support, including contact information, and procedures for user assistance.

4. Design

4.1 Database Design

4.1.1 ER Diagram

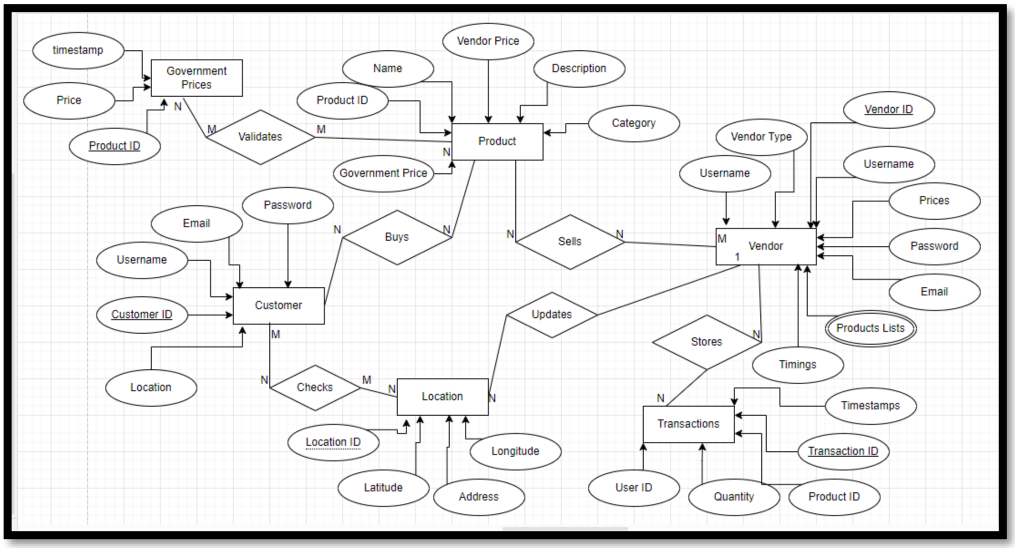


Figure 2 Entity Relationship Diagram

4.1.2 Class Diagram

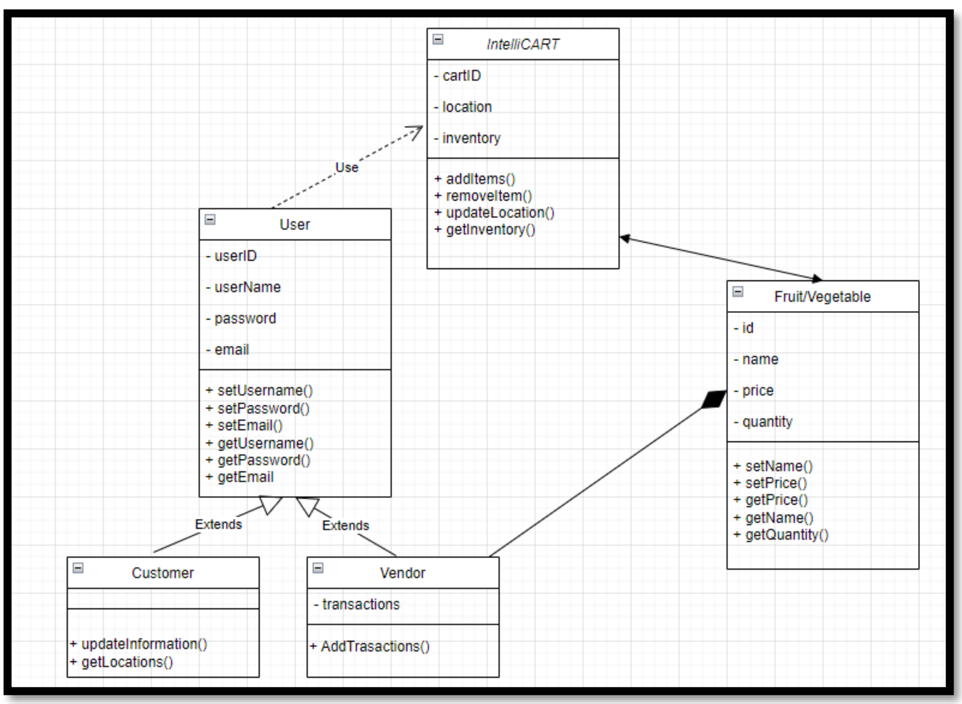


Figure 3 Class Diagram

4.1.3 Data Dictionary

4.1.3.1 Customer

Table 4 Data Dictionary of Customer

Customer						
Name	Customer					
Alias	User					
Where-used/how-used	Used When customer login/signup or buys a product.					
Content description	Composed of people buying fruits and vegetables.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
customer_id	Unique auto number generated number	Integer	12	No	None	PK
username	Name of customer	String	100	No	None	-
email	Unique email of customer	String	100	No	None	-
password	Hashed password	String	200	No	None	-
location	Location of customer	Object	500	No	None	-

4.1.3.2 Vendor

Table 5 Data Dictionary of Vendor

Vendor	
Name	Vendor
Alias	User
Where-used/how-used	Used When vendors login/signup or sells a product.
Content description	Composed of people selling fruits and vegetables.

Column Name	Description	Type	Length	Null able	Default Value	Key Type
vendor_id	Unique auto number generated number	Integer	12	No	None	PK
username	Name of customer	String	100	No	None	-
email	Unique email of customer	String	100	No	None	-
password	Hashed password	String	200	No	None	-
prices	Prices of fruits/vegetables	float	6	No	None	-
vendor_type	Sells fruits or vegetables or both	String	100	No	None	-
location_id	Location of vendor	String	12	No	None	FK
products_lists	Name of products	Object	500	No	None	-
timings	Open and close time	Object	100	No	None	-

4.1.3.3 Product

Table 6 Data Dictionary of Product

Product	
Name	Product
Alias	Fruits and Vegetables
Where-used/how-used	Used When customer buys and vendors sell fruits and vegetables
Content description	Composed of fruits and vegetables.

Column Name	Description	Type	Length	Null able	Default Value	Key Type
product_id	Unique number	Integer	12	No	None	PK
name	Name of customer	String	100	No	None	-
vendor_price	Price set by vendor	Integer	10	No	None	-
description	Description of products	String	500	Yes	None	-
category	Belongs to either Fruits and Vegetable	String	100	No	None	-
government_price	Price set by government	Integer	10	No	No	-

4.1.3.4 Government Prices

Table 7 Data Dictionary of Government Prices

Government Price						
Name	Government Price					
Alias	Govt. Price					
Where-used/how-used	Used When customer buys fruits/vegetable to validate prices.					
Content description	Composed of prices of fruits and vegetable set by government.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
product_id	Unique number	Integer	12	No	None	PK
price	Name of customer	String	10	No	None	-
timestamp	Last updated	Object	100	No	None	-

	price					

4.1.3.5 Location

Table 8 Data Dictionary of Location Services

Location						
Name	Location					
Alias	current location					
Where-used/how-used	Used When customer search for nearby carts or vendors update their location					
Content description	Composed of location of fruit and vegetable vendors.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
location_id	Unique auto number generated number	Integer	12	No	None	PK
latitude	Coordinates of location	Float	100	No	None	-
longitude	Coordinates of location	Float	100	No	None	-

4.1.3.5 Sales

Table 9 Data Dictionary of Sales

Transaction	
Name	Transaction
Alias	Records
Where-used/how-used	Used to store transaction between customer and vendor.
Content description	Composed of record of sell and purchase between vendors and customers

Column Name	Description	Type	Length	Null able	Default Value	Key Type
transaction _id	Unique auto number generated number	Integer	12	No	None	PK
product_id	Product ID	Integer	12	No	None	FK
quantity	Number of product sale	Integer	10	No	None	-
timestamps	Date and time of transaction.	Object	100	No	None	-

4.2 Software Architecture Design

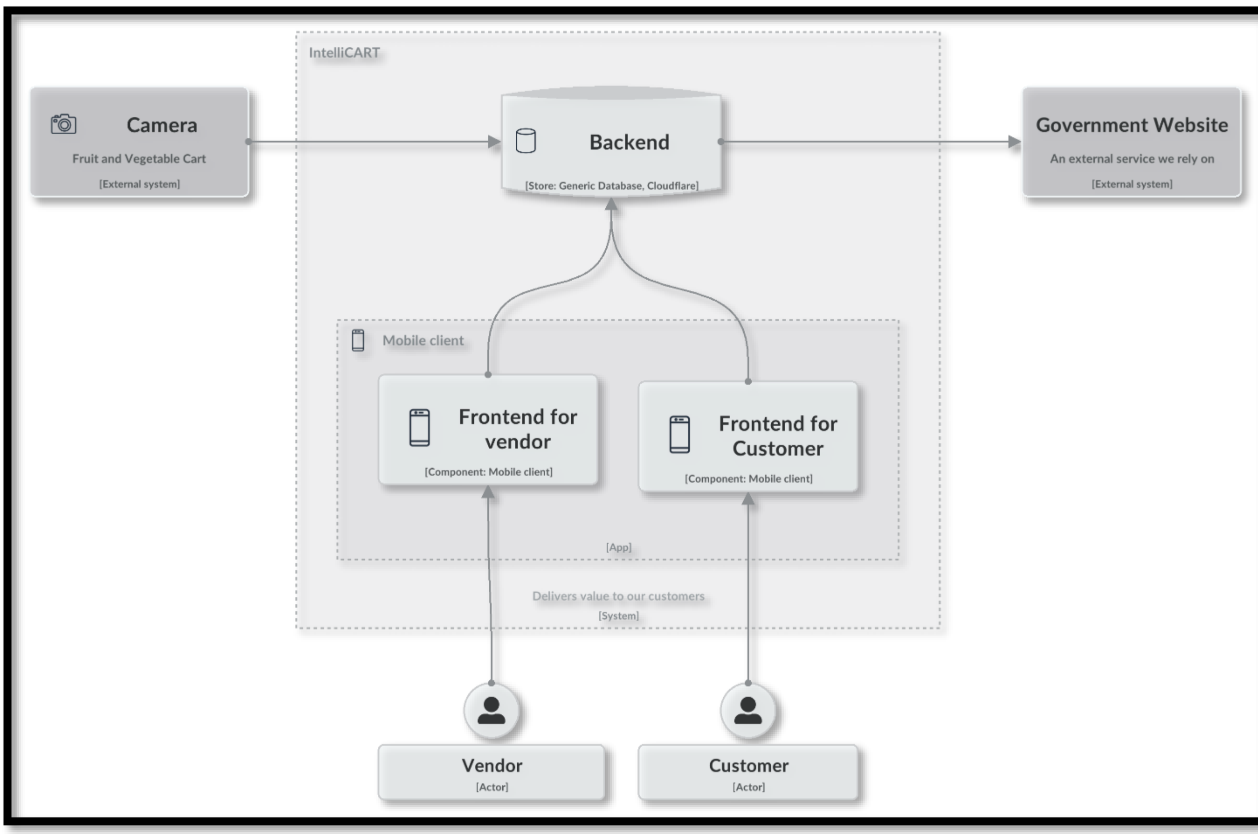


Figure 4 Software Architectural Diagram

4.3 Application Design
4.3.1 Sequence Diagram
4.3.1.1 Customer Registration

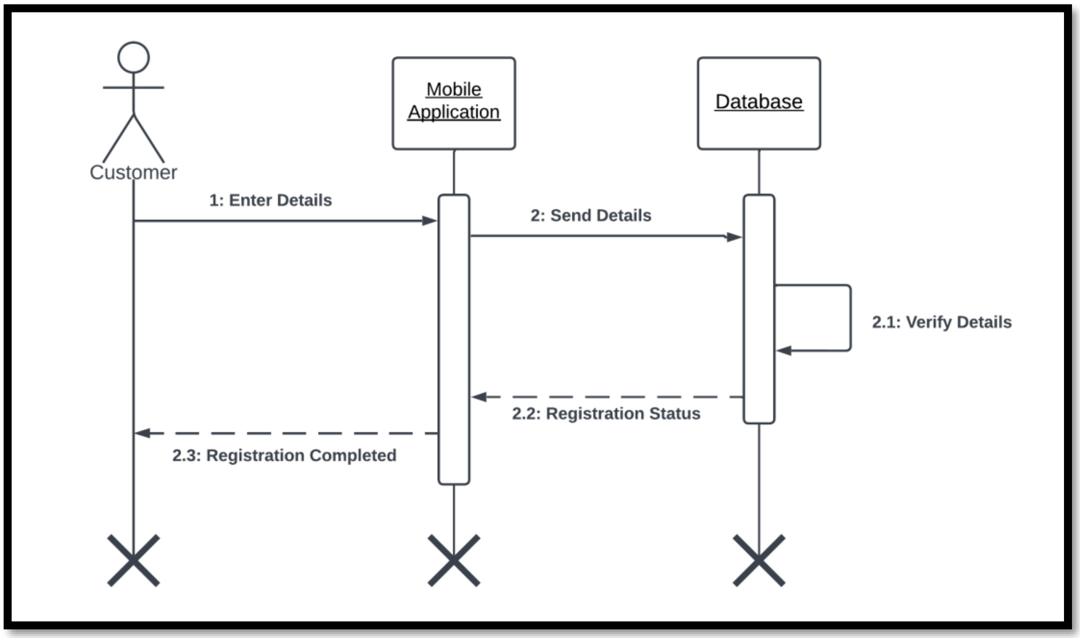


Figure 5 Sequence Diagram for Customer Registration

4.3.1.2 Vendor Registration

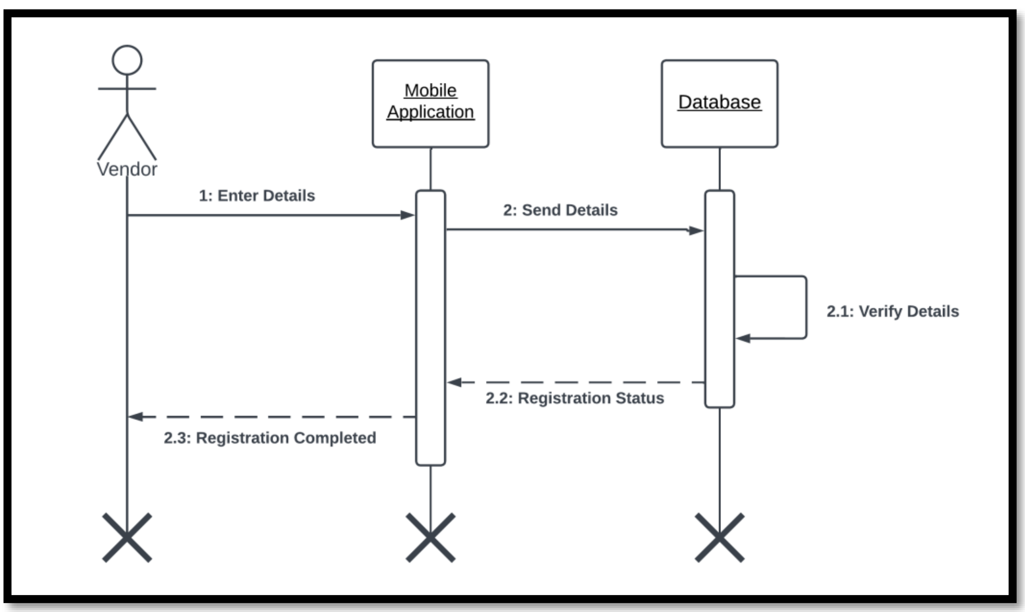


Figure 6 Sequence Diagram for Vendor Registration

4.3.1.3 Customer Login

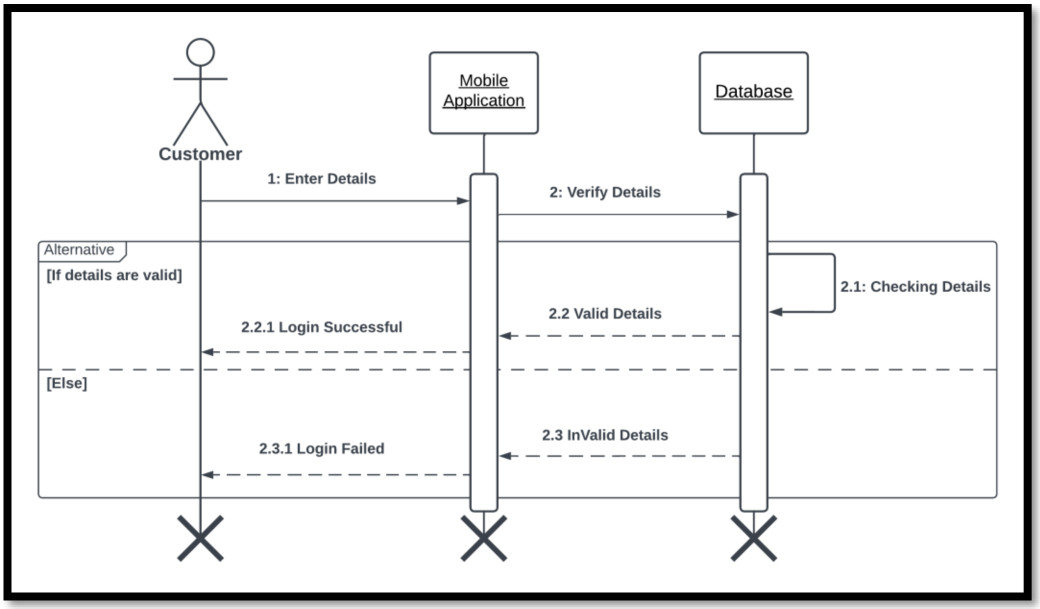


Figure 7 Sequence Diagram for Customer Login

4.3.1.4 Vendor Login

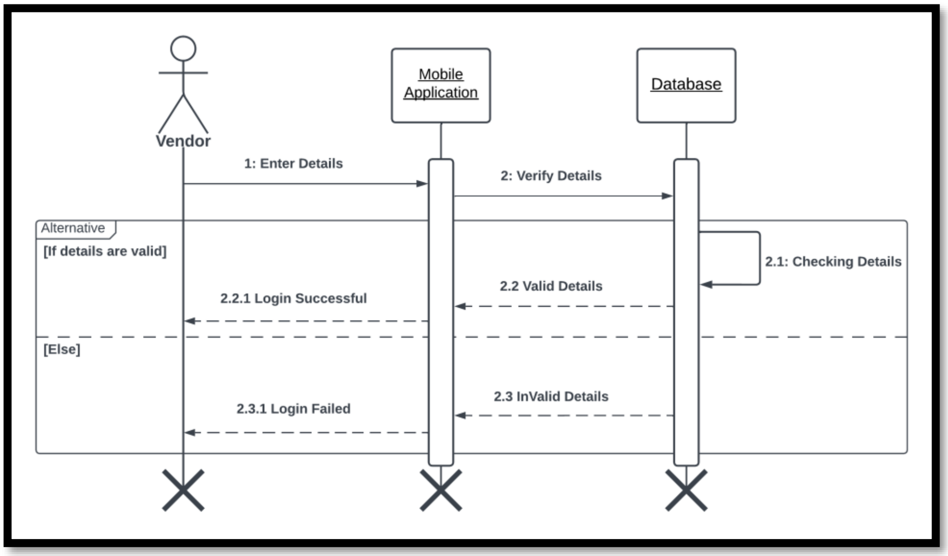


Figure 8 Sequence Diagram for Vendor Login

4.3.1.5 Quality Assessment

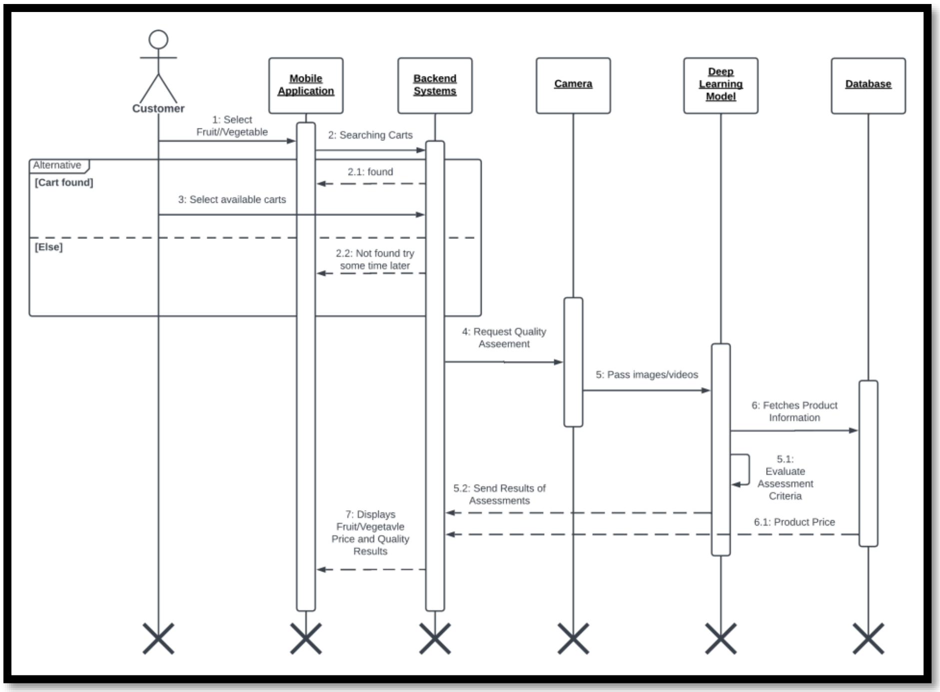


Figure 9 Sequence Diagram for Quality Assessment

4.3.1.6 Price Validation

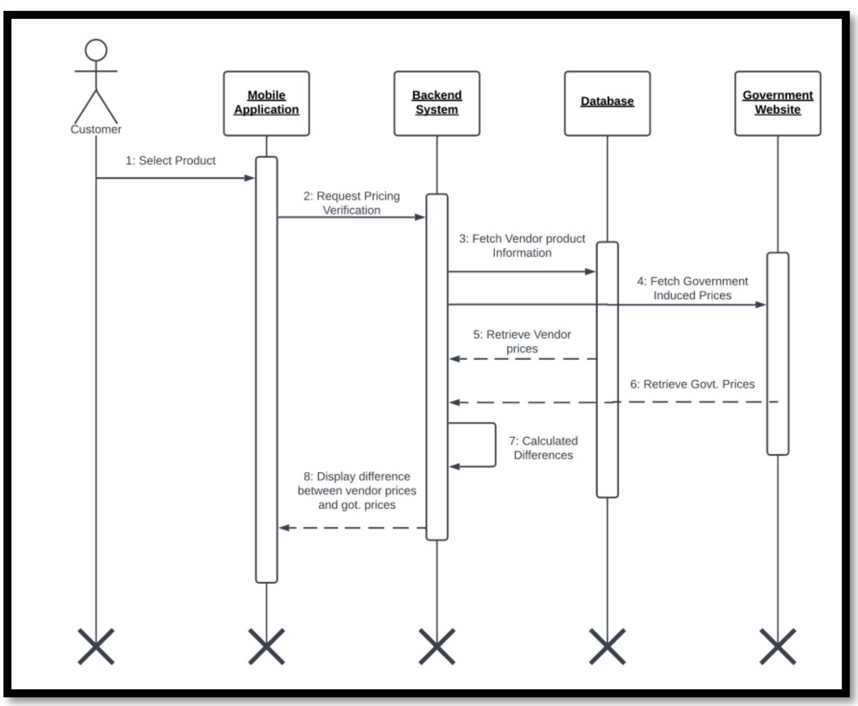


Figure 10 Sequence Diagram for Price Validation

4.3.1.7 Customer Location Nearby Carts

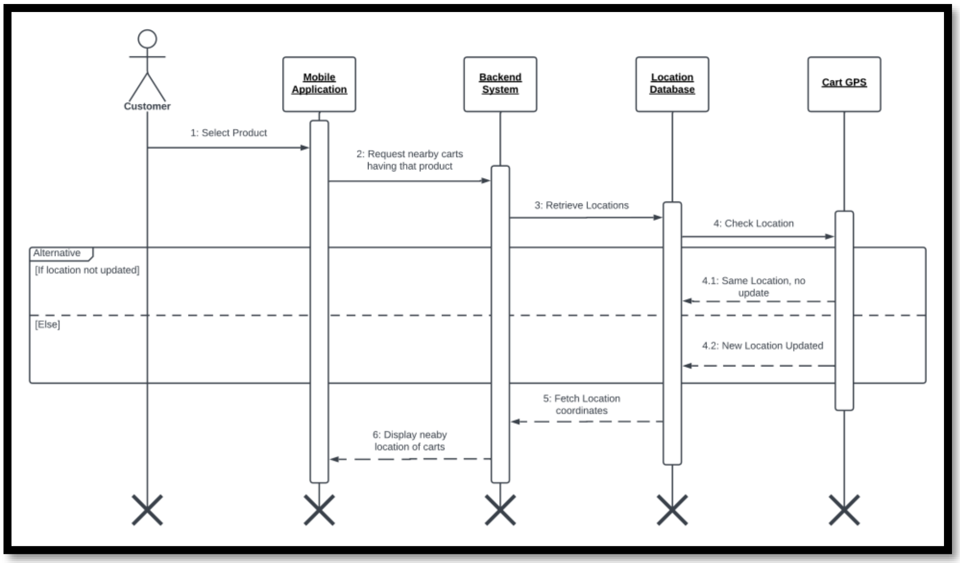


Figure 11 Sequence Diagram for Locating Nearby Carts

4.3.1.8 Customer Providing Feedback

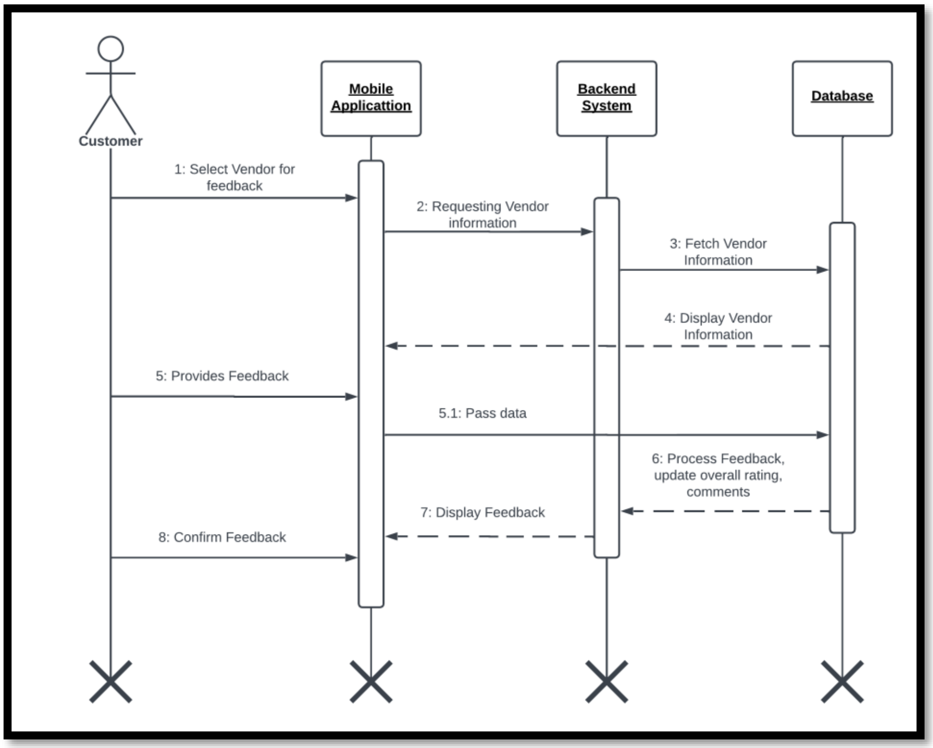


Figure 12 Sequence Diagram for Customer Providing Feedback

4.3.1.9 Vendor Add Product

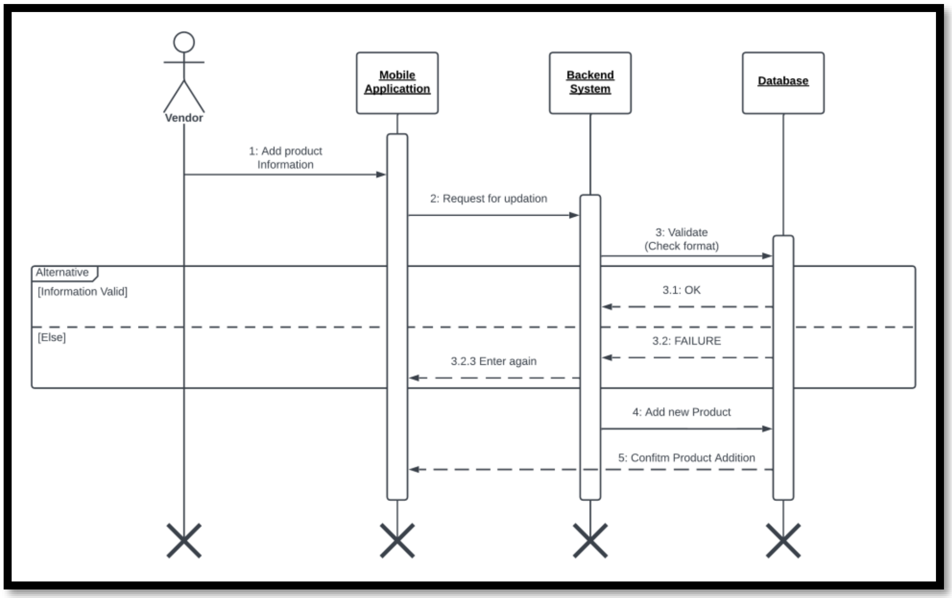


Figure 13 Sequence Diagram for Vendor Adding New Product

4.3.1.10 Vendor Delete Product

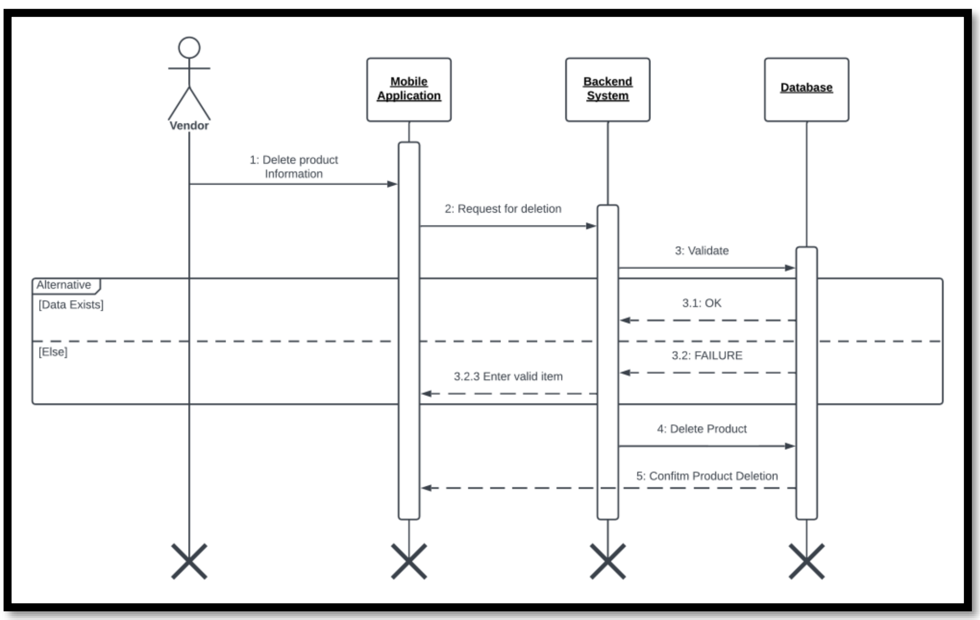


Figure 14 Sequence Diagram for Vendor Deleting Product

4.3.1.11 Transactions

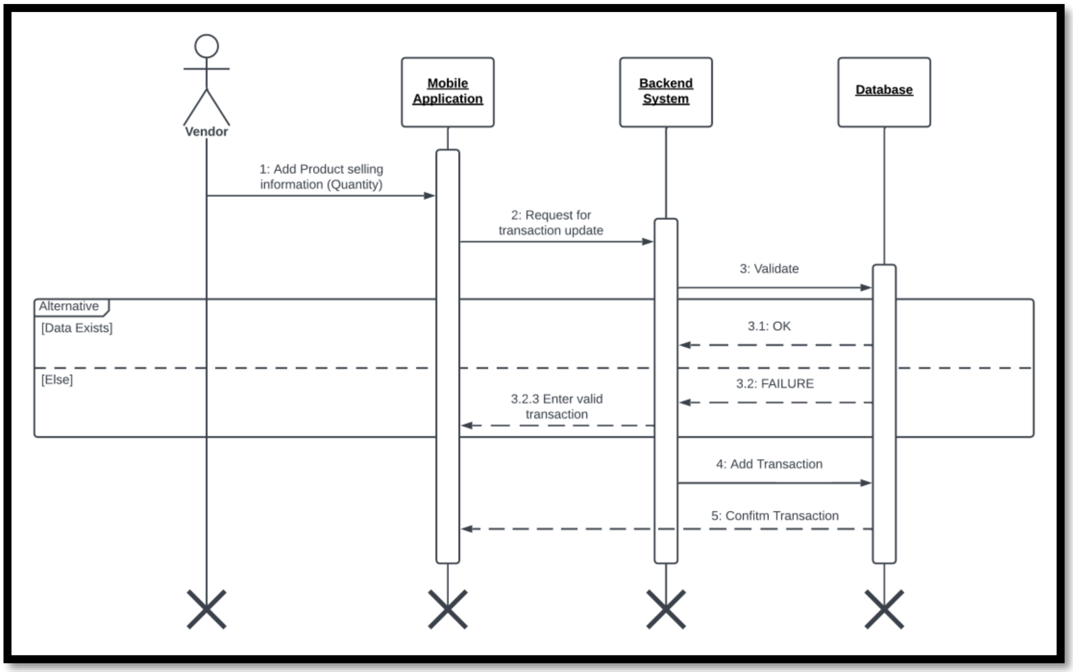


Figure 15 Sequence Diagram for Vendor Transactions

5. Implementation

5.1 Deep Learning

5.1.1 Methodology

The methodology of project involves collecting dataset of fruits and vegetable and preprocessed image to size of 256x256 pixels. On those images trained six deep learning models MobileNet, Inception V3, ResNet152, AlexNet, VGG-16, VGG-19. Using measures such as accuracy, precision, recall, and F1 score we assessed model performance.

To perform real time quality assessment trained a YOLOv8 model on collected dataset. Model was used to detect fruits and vegetable quality.

Finally using best model based on evaluation of metrics integrated in application so that customer or vendor upload image and gets quality of uploaded fruit or vegetable. The app used the trained model for classification and quality assessment and displayed the results to the user in a most user friendly way.

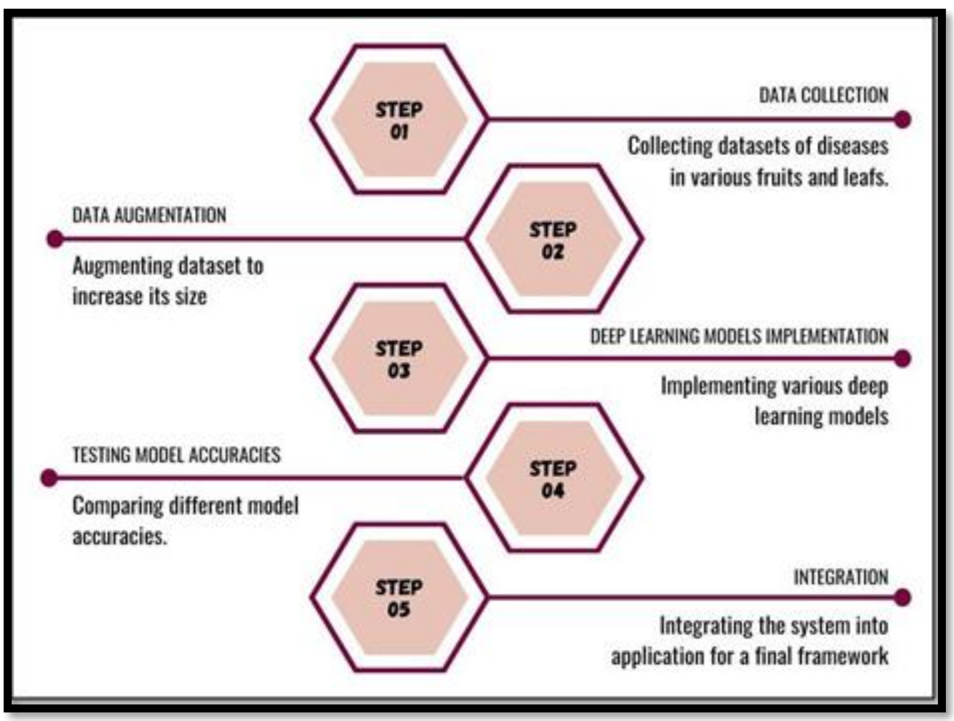


Figure 16 Overview of Methodology

5.1.1.1 Data Collection

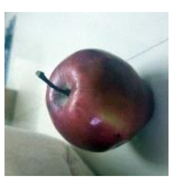
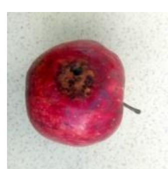
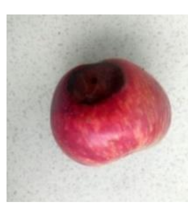
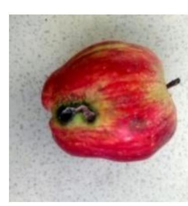
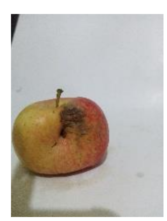
The dataset comprises 10155 images of fruits and vegetables (Apples, Banana, Oranges, Tomato and Green Chili). Some images were taken from online sources while other are taking by going on dataset drives using mobile camera. All images had resolution of 256x256 pixels and were divided into test and train folders.

The dataset contains 19 classes which are used to classify fruits and vegetable quality. Common quality like blotch, scab, rotten, greening and many more as mentioned in below table were included in classes. The images were preprocessed before training to optimize results and ensure consistency and performance

Table 10 Fruits and Vegetable Quality Division in 19 Classes

FRUITS AND VEGETABLES	CLASSES
Apple	Apple Blotch Apple Healthy Apple Scab Apple Rotten
Banana	Banana Heavily Bruised Banana Slightly Bruised Banana Firm
Orange	Orange Greening Orange Healthy Orange Rotten
Tomato	Tomato Damaged Tomato Old Tomato Ripe Tomato Unripe
Green Chili	Green Chili Damaged Green Chili Dried Green Chili Ripe Green Chili Unripe Green Chili Old

Table 11 Apple Dataset

CLASS	SNAPSHOTS			
Apple Blotch				
Apple Scab				

Apple Rotten				
Apple Healthy				

Table 12 Orange Dataset

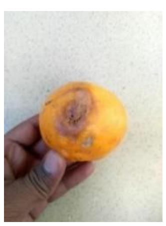
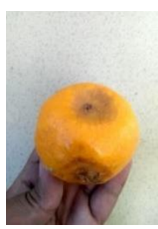
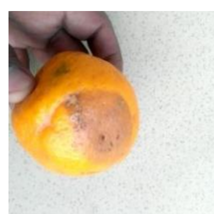
CLASS	**SNAPSHOTS**			
Orange Healthy				
Orange Rotten				
Orange Greening				

Table 13 Banana Dataset

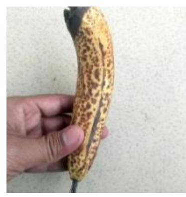
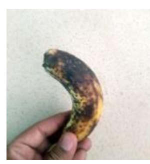
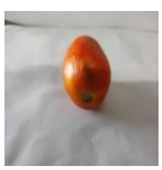
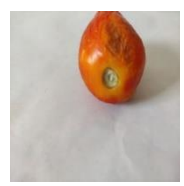
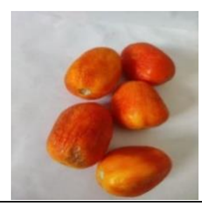
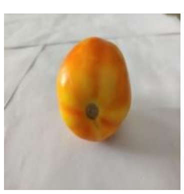
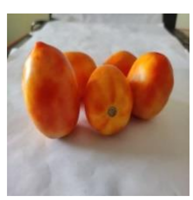
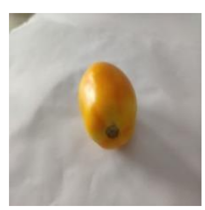
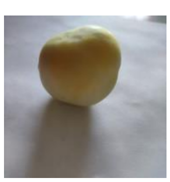
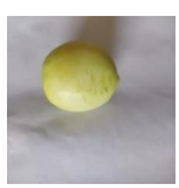
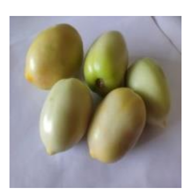
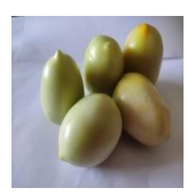
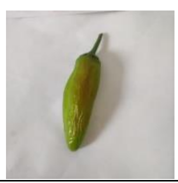
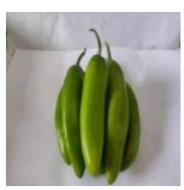
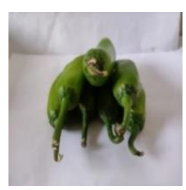
CLASS	SNAPSHOTS			
Banana Firm				
Banana Slightly Bruised				
Banana Heavily Bruised				

Table 14 Tomato Dataset

CLASS	SNAPSHOT			
Tomato Damaged				
Tomato Ripe				
Tomato Unripe				

Tomato Old				

CLASS	SNAPSHOTS			
Green Chili Damaged				
Green Chili Dried				
Green Chili Old				
Green Chili Ripe				
Green Chili Unripe				

5.1.1.2 Data Preprocessing

Preprocessed the data to make sure that it's in format that model can learn before we trained. The dataset undergoes through standardized pixel size, grayscale conversion and normalization of pixel values to increase data diversity and reduce overfitting also applied data augmentation techniques like rotating, flipping and zooming.

Spitted dataset in ratio 80:20 into training and testing respectively. Models trained on training set. And model was assessed on testing set.

In this project, data preprocessing was essential because it made sure the models were trained on high quality data for classifying quality of fruits and vegetables.

5.1.1.3 Deep Learning Models

A variety of deep learning-based neural network frameworks were utilized in the majority of these autonomous fruit classification projects. For instance, in a recent work [20], The authors used computer vision and image processing techniques to create an autonomous model that can identify veggies. Using images, the authors first compare 24 distinct vegetable varieties found in the dataset. The authors preprocessed the photos by shrinking and normalizing them after training the. They then put the convolutional neural network (CNN) into practice [21]. Model selection is important step in deep learning projects as it affects accuracy and performance. We trained six different models and based on evaluation metrics selected best model.

5.1.1.3.1 MobileNet

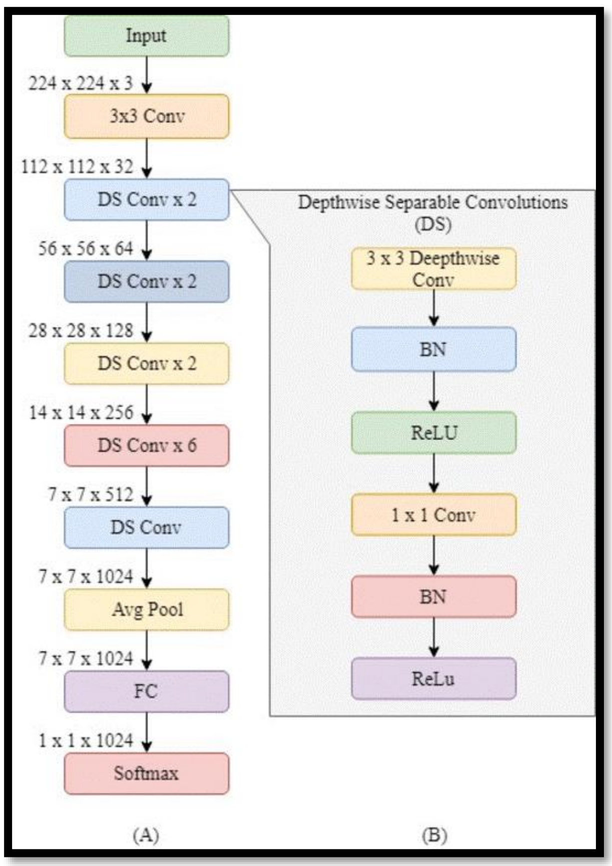


Figure 17 MobileNet Architecture

5.1.1.3.2 Inception V3

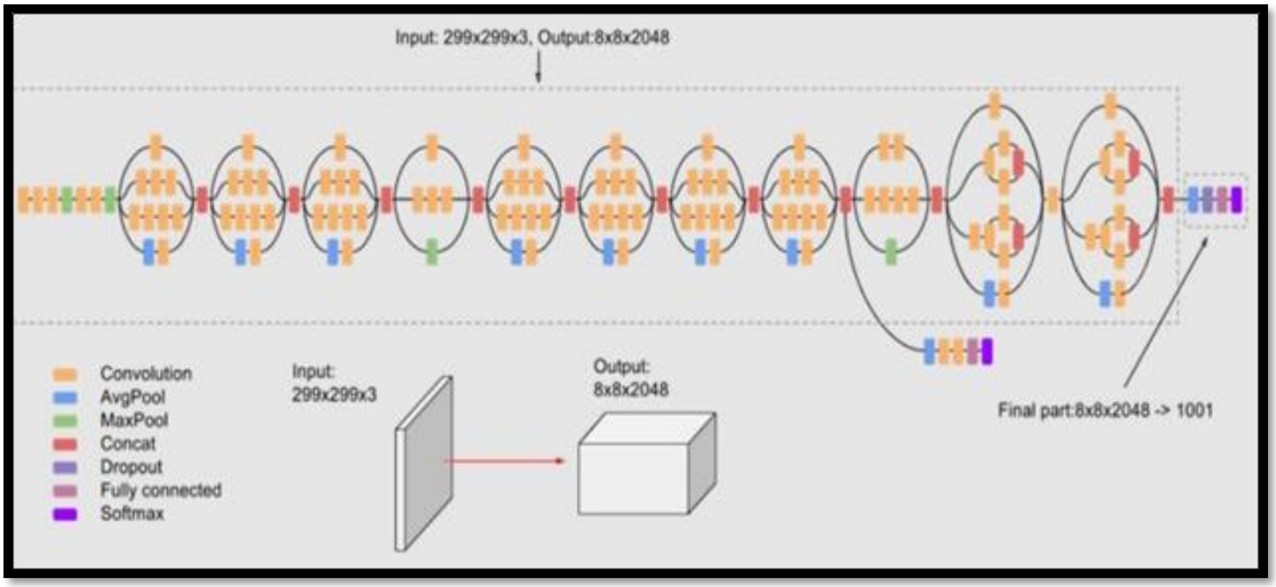


Figure 18 Inception V3 Architecture

5.1.1.3.3 VGG-16

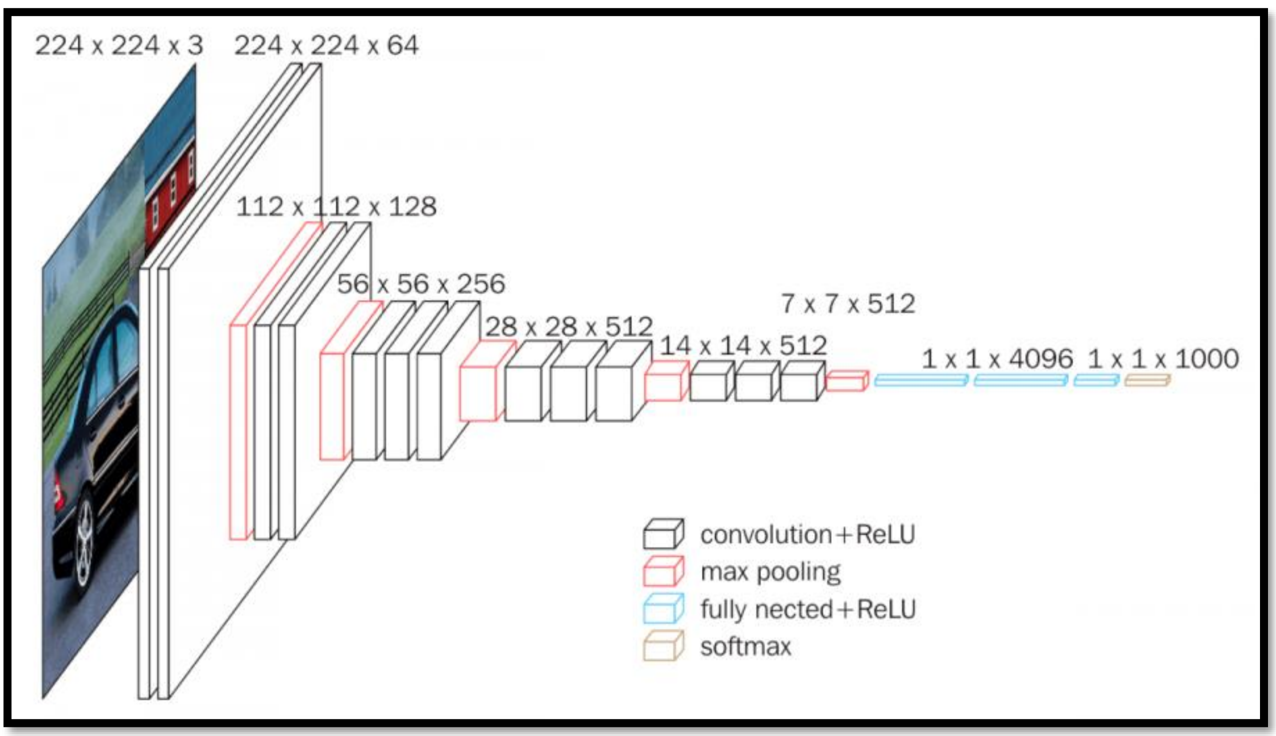


Figure 19 VGG-16 Architecture

5.1.1.3.4 VGG-19

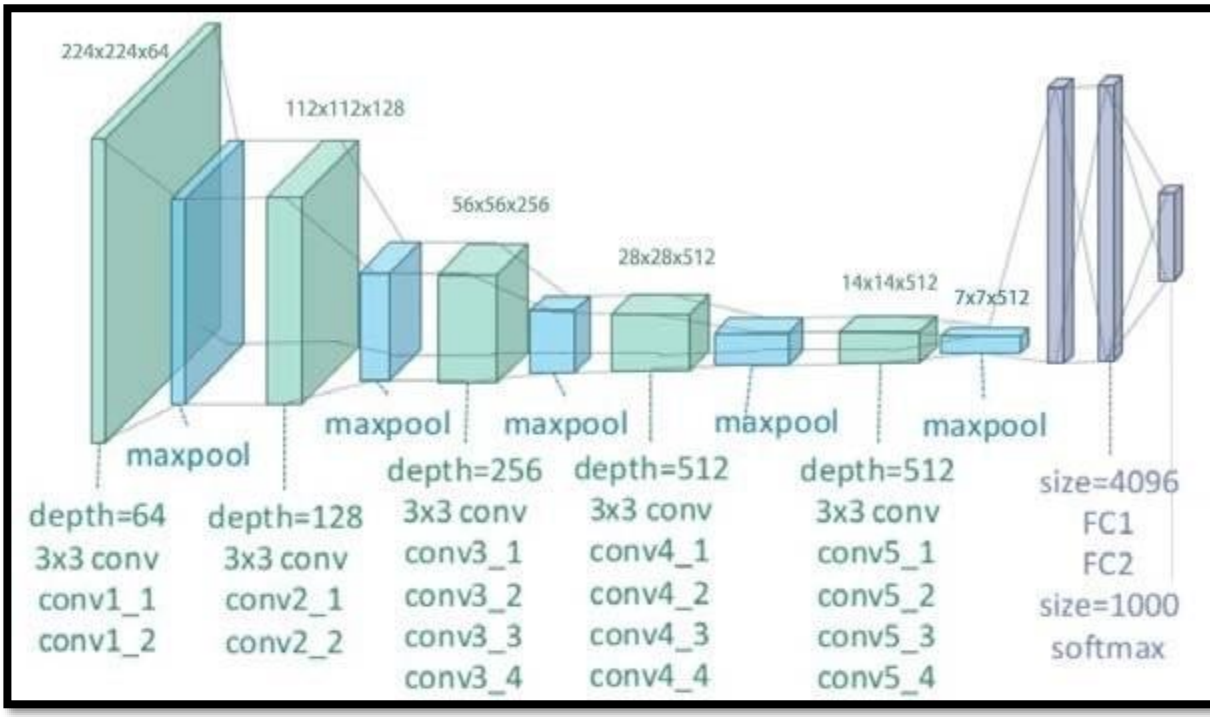


Figure 20 VGG-19 Architecture

5.1.1.3.5 AlexNet

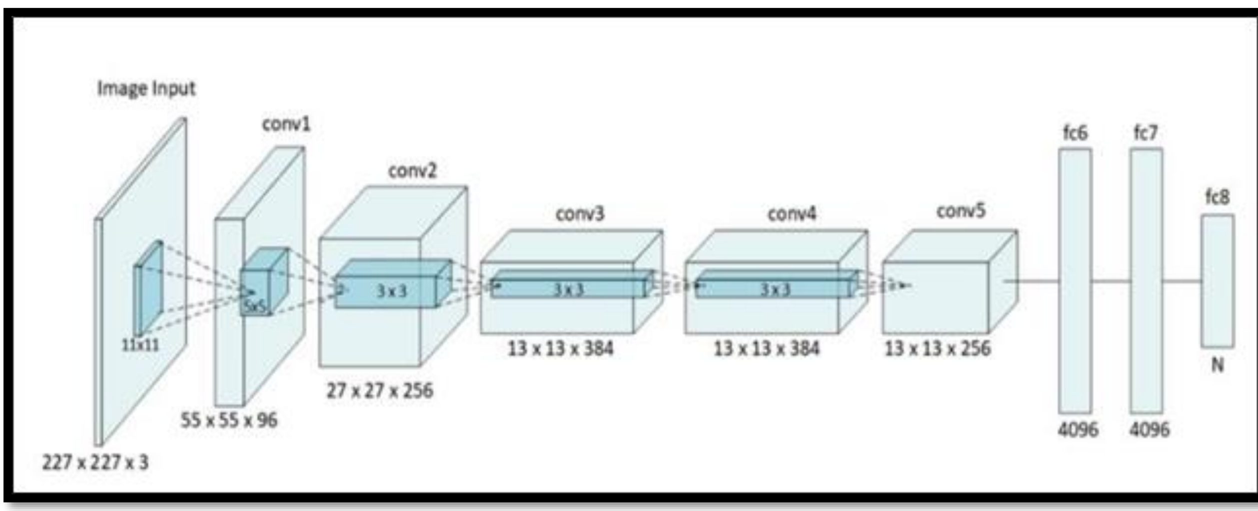


Figure 21 AlexNet Architecture

5.1.1.3.6 ResNet

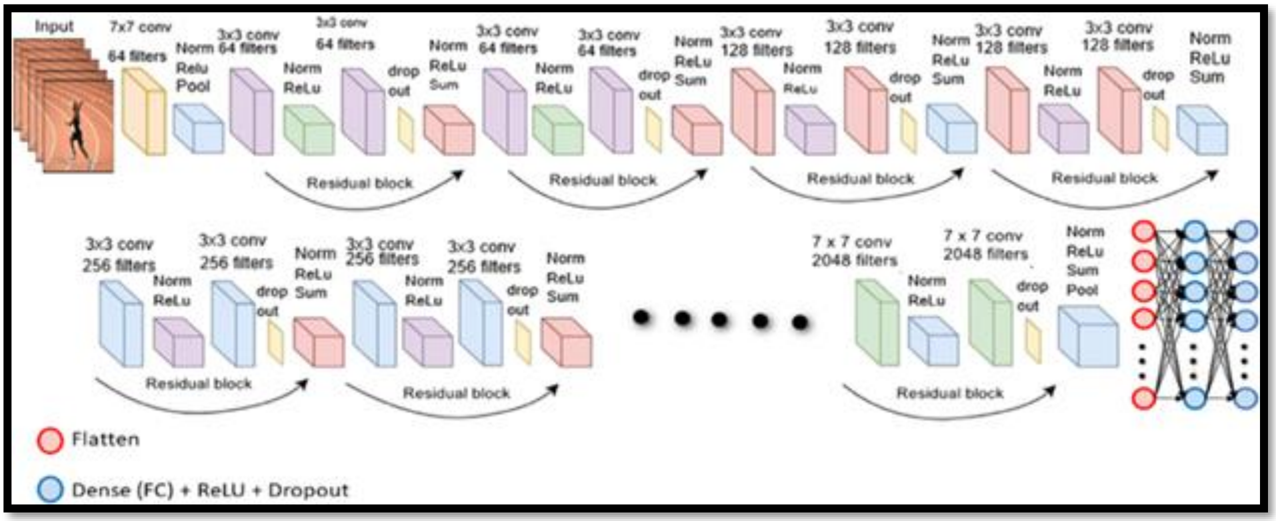


Figure 22 ResNet Architecture

5.1.1.4 Selecting Best model

Once, we had trained and tested all above mentioned six well known Neural Network models (AlexNet, MobileNet, VGG1, VGG19, Inception V3, ResNet 152) on our preprocessed dataset of fruits and vegetables' images, next step was to analyze and compare the results acquired from these models and choose the best model for our use case. The selection criteria for the best model were based on a tradeoff between accuracy and performance. We aimed to choose a model that was accurate in classifying the fruits based on their quality, but also efficient and fast.

After assessing various evaluation metrics, it became evident that MobileNet surpassed other models in terms of both accuracy and performance. Its superiority is attributed to a smaller parameter count and optimization for mobile device usage, resulting in enhanced speed and efficiency. MobileNet achieved an impressive accuracy of 94.93% and a macro F1 score of 0.91, the highest among the models under consideration. With a 28-layer architecture utilizing depth wise convolution for efficient computation and parameter reduction, MobileNet strikes a commendable balance between model accuracy and computational efficiency.

The architecture of MobileNet employs depth wise convolution, a two-step process:

1. **Depth wise Convolution**: This step involves convolving the input volume with a distinct filter for each input channel. Each input channel undergoes independent convolution, yielding feature maps with the same number of channels as the input.
2. **Pointwise Convolution**: Following depth wise convolution, a 1x1 convolution with a set of filters is applied to the output. Pointwise convolution facilitates the mixing and combination of features from different channels, enabling the network to learn intricate interactions between features.

Further scrutiny involved calculating Inter-Quartile Ranges (IQR) for all models based on True Positive Rates (TPR), False Positive Rates (FPR), True Negative Rates (TNR), and False Negative Rates (FNR). MobileNet exhibited the highest TPR and TNR, approaching 1, while maintaining the lowest FPR and FNR, close to 0, outperforming other models.

Table 15 MobileNet IQR Analysis

TPR average	0.9058942945130736
FPR average	0.002858193302300445
TNR average	0.9971418066976995
FNR average	0.0941057054869264

In-depth examination of confusion matrices reinforced the indication that MobileNet excels in classification tasks, solidifying its selection as the optimal model for fruit disease classification. This choice is deemed suitable for real-world applications where both speed and precision are pivotal considerations.

Table 16 Deep Learning Model Used and their Configurations

Model	**Number of Layers**	**Input Image Size**	**Activation Function**	**Convolutional Layers**	**Pooling Layers**	**Fully Connected Layers**
MobileNet	88	224*224	softmax	multiple	multiple	1
VGG-16	16	224*224	softmax	13	5	3
VGG-19	19	224*224	softmax	16	5	3
InceptionV3	159	224*224	softmax	multiple	2 max-pooling layers, 11 inception modules, 1 average pooling layer	2
AlexNet	8	224*224	softmax	5	3	3
ResNet 152	152	224*224	softmax	151	3	1

5.1.1.5 Object Detection

Following the training and evaluation of the deep learning classification models, we trained a highly efficient YOLOv8 model to perform real time object detection of the fruits and vegetables. YOLOv8 model was employed to precisely identify and classify the fruits and vegetables based on their quality classes. YOLOv8 is a popular object detection model that is widely used in computer vision applications. Since it is highly precise and computationally efficient, it is appropriate for real-time applications. We specifically chose YOLOv8 because it has an overall improved accuracy and higher speed than its predecessors. YOLOv8 is an open source model designed by ultralytics, hence it is easily customizable.

To train this object detection model, whole data was annotated by precisely creating bounding boxes around each instance. Then the images were scaled to 640x640 pixel size to make it easily usable input for YOLOv8. Keeping the high performance of YOLOv8 in consideration, we kept the number of epochs minimum to keep it least resource intensive, thus at last 20 epochs were set to be optimal.

Finally, we obtained following results:

(Specially mAP50 i.e. mean Average Precision with IoU threshold at 0.50, obtained was 0.955 after 20 epochs).

```
                   all       1419       1984      0.872      0.892      0.922      0.784

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/20      7.18G     0.5041     0.3682      1.011         21        640: 100% 311/311 [03:02<00:00,  1.70it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 45/45 [00:32<00:00,  1.39it/s]
                   all       1419       1984      0.889      0.892      0.929       0.79

20 epochs completed in 1.284 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/train/weights/best.pt, 52.0MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.221 🚀 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25849024 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 45/45 [00:35<00:00,  1.28it/s]
                   all       1419       1984      0.889      0.892      0.929      0.789
          apple_blotch       1419         72      0.824      0.722      0.853      0.804
         apple_healthy       1419        135      0.834      0.891      0.923      0.883
          apple_rotten       1419        158      0.979      0.981      0.993      0.949
            apple_scab       1419         50      0.989       0.98      0.994       0.94
           banana_firm       1419         80      0.854      0.804      0.892      0.591
  banana_heavilybruised       1419        207      0.902      0.843       0.94      0.672
 banana_slightlybruised       1419         61      0.738      0.785       0.78      0.536
    greenchilli_damaged       1419         24      0.952      0.832      0.917      0.666
       greenchilli_ripe       1419         96      0.917      0.927      0.967      0.703
        orange_greening       1419        219      0.904       0.95      0.981      0.876
         orange_healthy       1419        156      0.757      0.942       0.94      0.864
          orange_rotten       1419        295      0.935      0.971      0.986      0.947
         tomato_damaged       1419         25          1      0.752       0.85      0.744
            tomato_old       1419        110      0.746      0.982      0.928       0.84
           tomato_ripe       1419        209      0.914      0.914      0.931      0.816
         tomato_unripe       1419         87      0.983          1      0.984      0.797
Speed: 0.2ms preprocess, 9.4ms inference, 0.0ms loss, 1.7ms postprocess per image
Results saved to runs/detect/train
```

Figure 23 Results of training YOLOv8 Model on the dataset consisting of apple, oranges, Tomato, Green Chili and bananas comprising of 19 classes.

This trained YOLOv8 model was also tested using real time camera to assess quality of fruits and vegetables. Thus this would be used in a camera mounted or either mobile camera on the fruit and vegetables' cart and would be integrated in a mobile application to help vendors and customers to provide a transparent and smooth selling and buying experience.

Table 17 Initialization Parameters for YOLOv8

Hyper-Parameters	***Value***
Epochs	20
Warm-up bias learning rate	0.1
Batch size	16
Input Image size	640
Weight Decay	0.0005
Momentum	0.937

5.1.1.5.1 Bounded Box

A bounding box is a graphical representation of the position of an object in an image. Each image's bounding box includes the ascribes listed below.

1. Dimensions
2. Measurements
3. Class (rotten_apple, bruised_banana, fresh_oranges, etc.).

Figure 24 Bounded box Example

5.2 Mobile Application

5.2.1 Backend Code

FYP II > API > Yolo-Fruit-classification-API > video_classification_yolo.py > process_frame

```python

app = FastAPI()
model = YOLO("best_final.pt")

@app.post("/upload_video/")
async def upload_video(video_file: UploadFile = File(...)):
    # Save the uploaded video to disk
    save_path = f"videos/{video_file.filename}"
    with open(save_path, "wb") as buffer:
        shutil.copyfileobj(video_file.file, buffer)

    model = YOLO("best_final.pt")
    VIDEO_PATH = "./videos/" + video_file.filename
    video_info = sv.VideoInfo.from_video_path(VIDEO_PATH)
    global complete_video_labels
    target_video_path = "./output_videos/" + video_file.filename
    sv.process_video(source_path=VIDEO_PATH, target_path=target_video_path, callback=process_frame)
    print(complete_video_labels)
    results = count_occurences(complete_video_labels)
    print(results)

    # Save the results to a JSON file
    json_file_path = "./results_json/" + video_file.filename.split(".")[0] + ".json"
    with open(json_file_path, "w") as json_file:
        json.dump(results, json_file, indent=4)

    return results


```

FYP II > API > Yolo-Fruit-classification-API > video_classification_yolo.py > process_frame

```python
from fastapi import FastAPI, File, UploadFile
from ultralytics import YOLO
import json
import supervision as sv        # version 0.20.0
import numpy as np
import shutil

# To execute this API : uvicorn video_classification_yolo:app --reload --host 0.0.0.0 --port 8000

complete_video_labels = []

def process_frame(frame: np.ndarray, _) -> np.ndarray:
    results = model(frame, imgsz=1280)[0]
    detections = sv.Detections.from_ultralytics(results)
    box_annotator = sv.BoxAnnotator(thickness=4, text_thickness=4, text_scale=2)
    labels = []
    global complete_video_labels

    for detection in detections:
        # print(detection)
        print('processing........')
        label = [f"{model.names[detection[3]]} {detection[2]:0.2f}"]
        labels = labels + label
        label_tuple = {"name": model.names[detection[3]], "confidence": f"{detection[2]:0.2f}"}     # just for
        complete_video_labels = complete_video_labels + [label_tuple]

    frame = box_annotator.annotate(scene=frame, detections=detections, labels=labels)
    return frame
```

Figure 25 Backend code for Yolo V8 Real time Freshness Quality Assessment

FYP II > API > Quality-Prediction-FAST-API > predict.py > ...

```python
from fastapi import FastAPI, File, UploadFile
from keras.models import load_model
import numpy as np
from PIL import Image
import io
from datetime import datetime

# To execute this API : uvicorn predict:app --reload --host 0.0.0.0 --port 8000
# Loading CNN model
try:
    mobilenet_model = load_model('mobilenet_FYP_model.h5')    #only works with keras and tensorflow 2.15.0
    print("model loaded.")
except Exception as e:
    print("\n\n Error loading model: ", e)

# func to predict class using CNN model
def predict_class(image_arr):
    predictions = mobilenet_model.predict(image_arr)

    # Get most likely class
    predicted_classes = np.argmax(predictions, axis=1)
    #Assuming the model returns a list of classes with probabilities
    # You may need to adjust this part based on your model's output format
    class_labels = ['Apple_blotch','Apple_healthy', 'Apple_rotten','Apple_scab',
                    'Banana_firm','Banana_heavilybruised','Banana_slightlybruised',
                    'GreenChilli_damaged','GreenChilli_dried','GreenChilli_old','GreenChilli_ripe',
                    'Orange_greening','Orange_healthy','Orange_rotten',
                    'Tomato_old','Tomato_ripe','Tomato_rotten','Tomato_unripe']  # Provide your class names he
    predicted_classes = [class_labels[i] for i in np.argmax(predictions, axis=1)]

```

FYP II > API > Quality-Prediction-FAST-API > predict.py > ...

```python
# preprocess image
def preprocess_image(image):
    image = image.resize((224, 224))  # Resize the image to match the input size of the model
    image_arr = np.array(image) / 255.0  # Normalize pixel values to [0, 1]
    image_arr = np.expand_dims(image_arr, axis=0)  # Add batch dimension
    return image_arr

# Define the FastAPI endpoint
app = FastAPI()

@app.post("/predict/")                                #uvicorn package is used to let your API running as a server
async def predict(image: UploadFile = File(...)):
    # Read the uploaded image file
    contents = await image.read()

    # Convert the image data to a PIL Image object
    img = Image.open(io.BytesIO(contents))

    # Preprocess the image
    img_arr = preprocess_image(img)

    # Make predictions using the model
    predicted_classes = predict_class(img_arr)

    #To print time for logs
    timestamp = datetime.now().strftime("%d %B %Y %H:%M:%S")
    print(str(timestamp) + "  --------  " + str({"predictions": predicted_classes}))
    return {"predictions": predicted_classes}

```

Figure 26 Backend Code for MobileNet Freshness Quality Assessment execute_all_scripts_daily.py > ...

```python
import subprocess

def execute_scripts():
    try:
        # Execute 1
        subprocess.run(["python", "1_download_pdf.py"], check=True)

        # Execute 2 after 1
        subprocess.run(["python", "2_pdf_to_img.py"], check=True)

        subprocess.run(["python", "3_get_table_out_of_img.py"], check=True)

        subprocess.run(["python", "4_get_rows_in_img.py"], check=True)

        subprocess.run(["python", "5_cloudmersive_img_ocr.py"], check=True)

        subprocess.run(["python", "6_produce_final_json.py"], check=True)

    except subprocess.CalledProcessError as e:
        # Handle errors if any script fails
        print(f"Error: {e}")

if __name__ == "__main__":
    execute_scripts()

```

{} price_list.json > ...


```json
{
    "26Mar2024": {
        "applegolden": 219,
        "apple": 166,
        "banana": 176
    },
    "29Mar2024": {
        "applegolden": 228,
        "apple": 175,
        "banana": 170
    },
    "2024-04-15": {
        "applegolden": 204,
        "apple": 175,
        "banana": 155,
        "orange": 307
    },
    "2024-04-16": {},
    "2024-04-17": {
        "apple_golden": 210,
        "apple": 185,
        "banana": 160,
        "orange": 322
    },
    "2024-04-22": {
        "apple_golden": 210,
        "apple": 185,
        "banana": 160,
        "orange": 322
    },
```

```python
serve_prices.py > ...
from fastapi import FastAPI, HTTPException
from datetime import datetime
import json

# To execute this API : uvicorn serve_prices:app --reload --host 0.0.0.0 --port 8001
app = FastAPI()

# Define route to handle date input
@app.get("/{date}")
async def get_price(date: str):
    try:

        # Load data from JSON file
        with open("price_list.json", "r") as file:
            data = json.load(file)

        # Convert input date string to datetime object
        print("Provided date in API call: ", date)
        input_date = date
        timestamp = datetime.now().strftime("%d %B %Y %H:%M:%S")
        # Check if the date exists in the data
        if input_date in data:
            print(str(timestamp) + "  --------  ")
            return {input_date: data[input_date]}
        else:
            print(str(timestamp) + "  --------  ")
            raise HTTPException(status_code=404, detail="Data not found for the given date")
    except ValueError:
        timestamp = datetime.now().strftime("%d %B %Y %H:%M:%S")
        print(str(timestamp) + "  --------  ")
        raise HTTPException(status_code=400, detail="Invalid date format. Please use format DDMMMYYYY (e.g., 26Mar2024)")
```

Figure 27 Backend Code for Extracting Government Prices

5.2.2 Frontend Interface

5.2.2.1 Main Screen (User Type)

Figure 28 Main Screen Select User Type

5.2.2.2 Customer Side

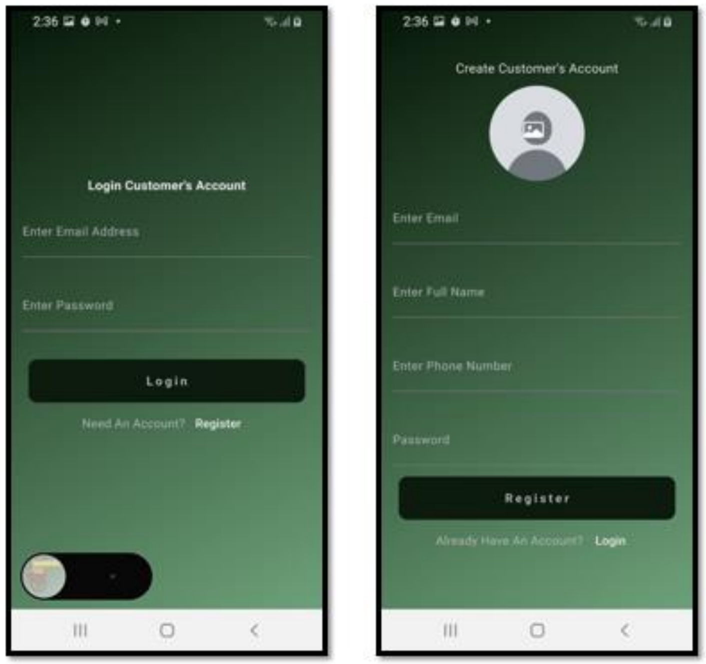


Figure 29 Customer Side Login and Registration Screen

Figure 30 Customer Side Nearby Vendor Cart Screen

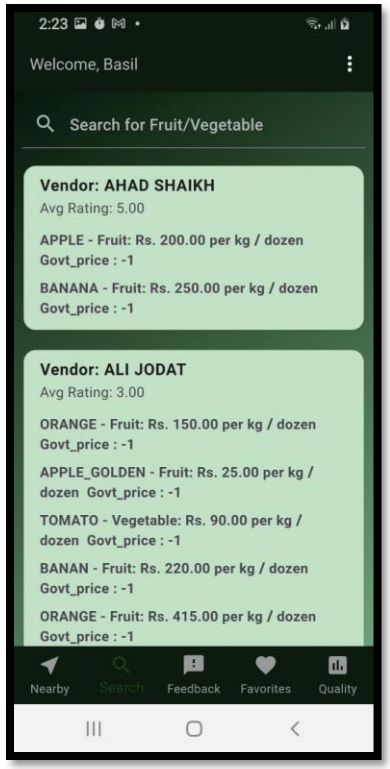


Figure 31 Customer Side Searching for Vendor Carts Screen

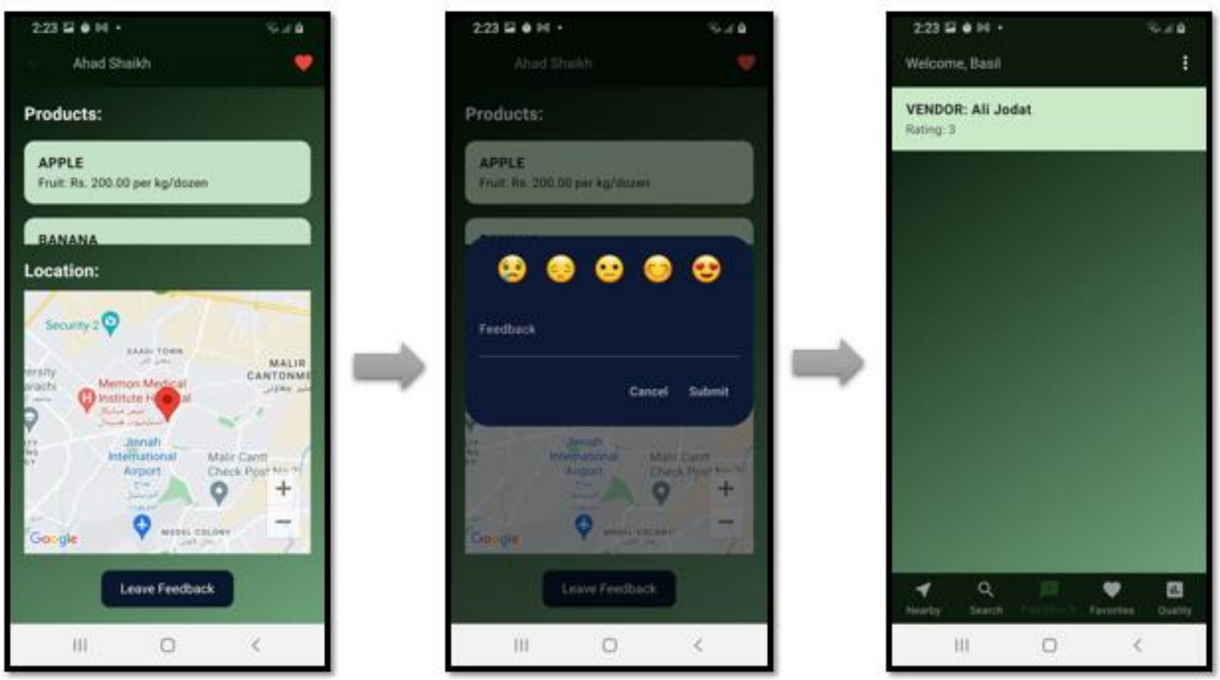


Figure 32 Customer Side Feedback to Vendor Screens

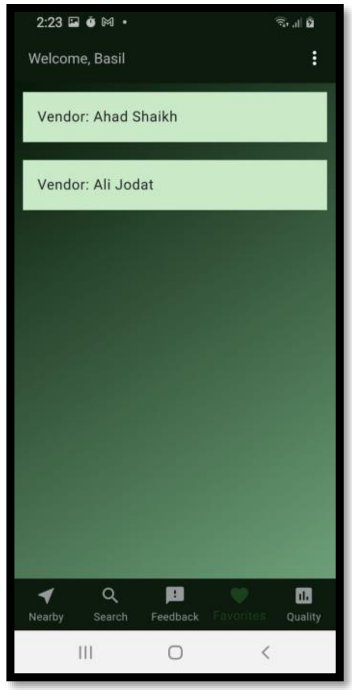


Figure 33 Customer Side Favorites Vendor Screen

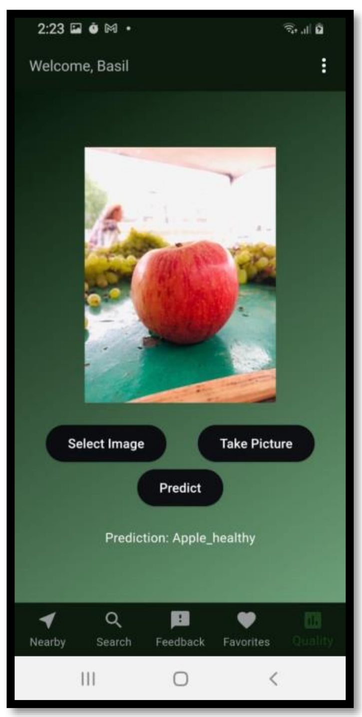


Figure 34 Customer Side Freshness Quality Assessment Screen

5.2.2.3 Vendor Side

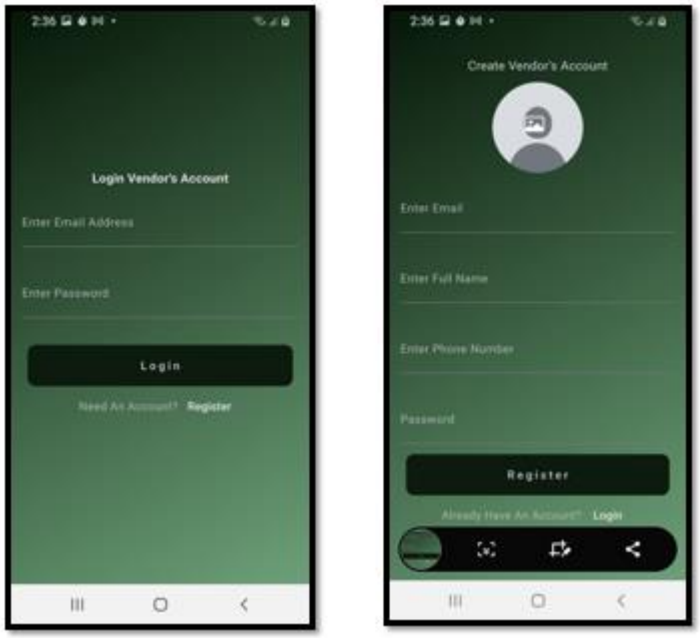


Figure 35 Vendor Side Login And Registration Screen

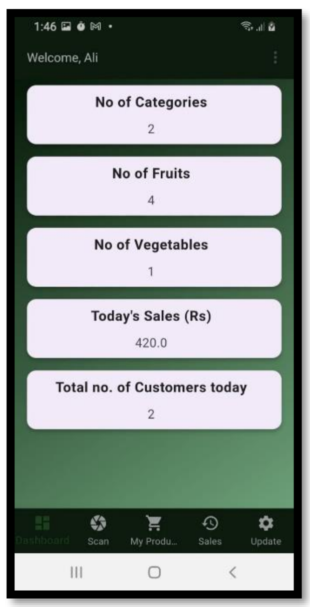


Figure 36 Vendor Side Dashboard Screen

Figure 37 Vendor Side Product Screen

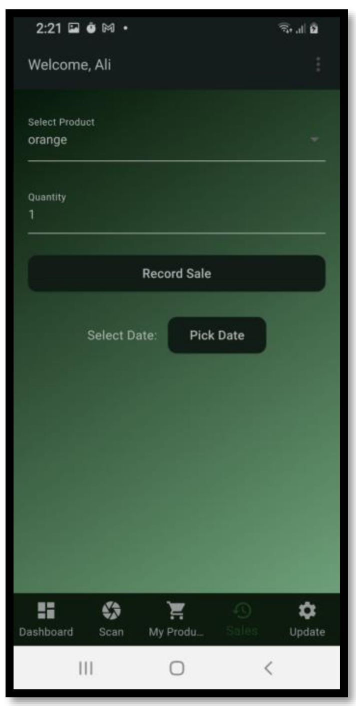


Figure 38 Vendor Side Sales Screen

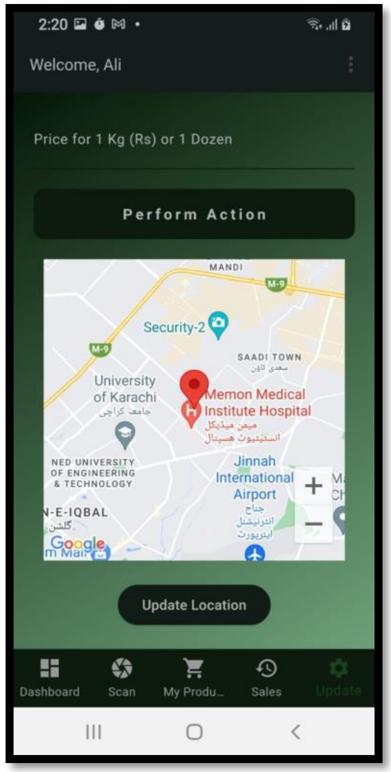


Figure 39 Vendor Side Update Screen

5.2.2.4 Customer Freshness Quality Assessment Flow

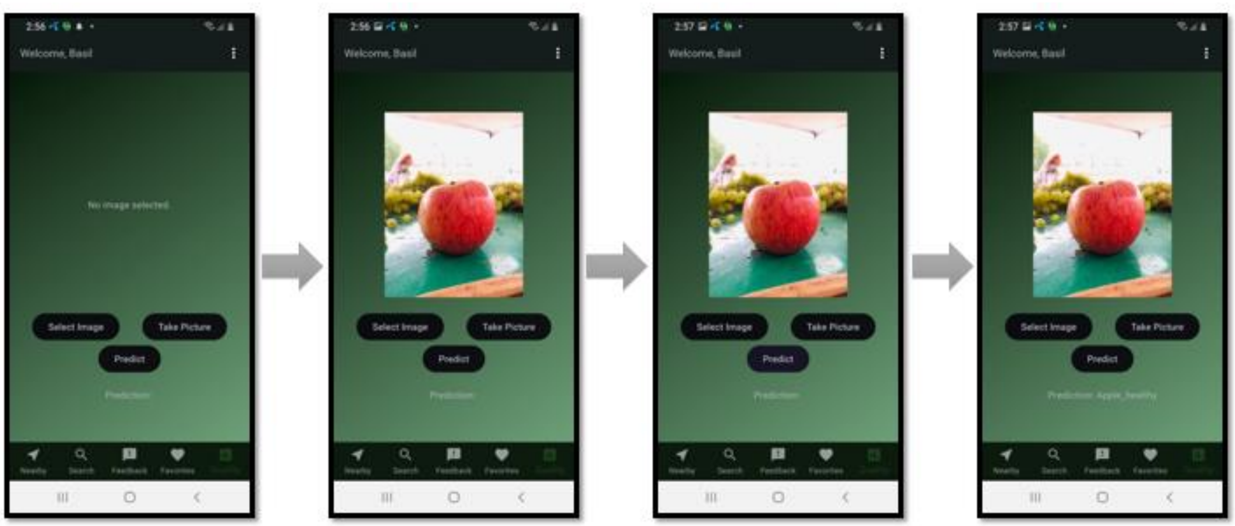


Figure 40 MobileNet Freshness Quality Screens

5.2.2.5 Vendor/Customer Real Time Freshness Quality Assessment Flow

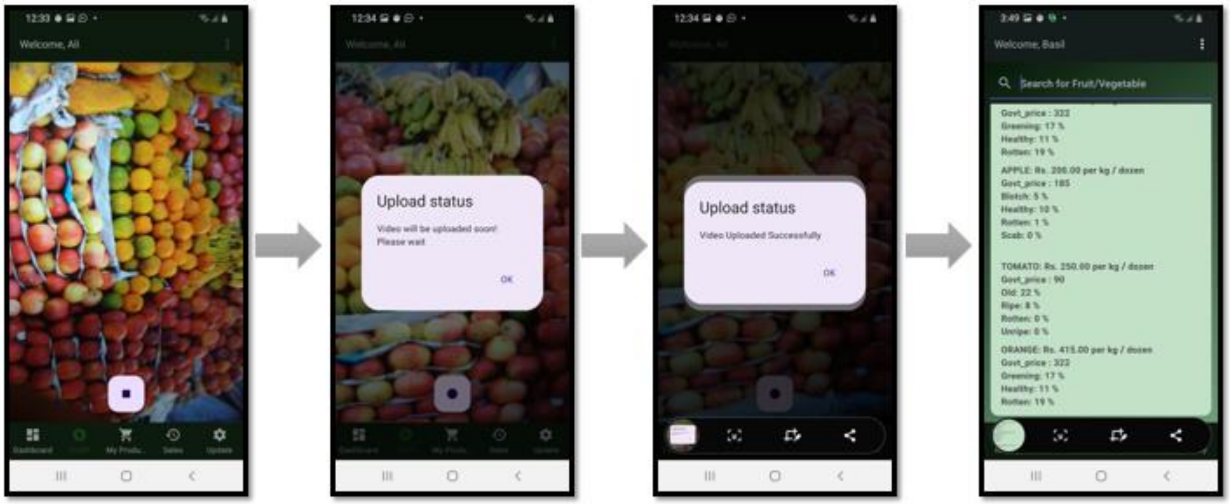


Figure 41 Yolo V8 Real Time Quality Assessment Screens

5.2.2.6 Customer Tracking Vendor Cart Location Flow

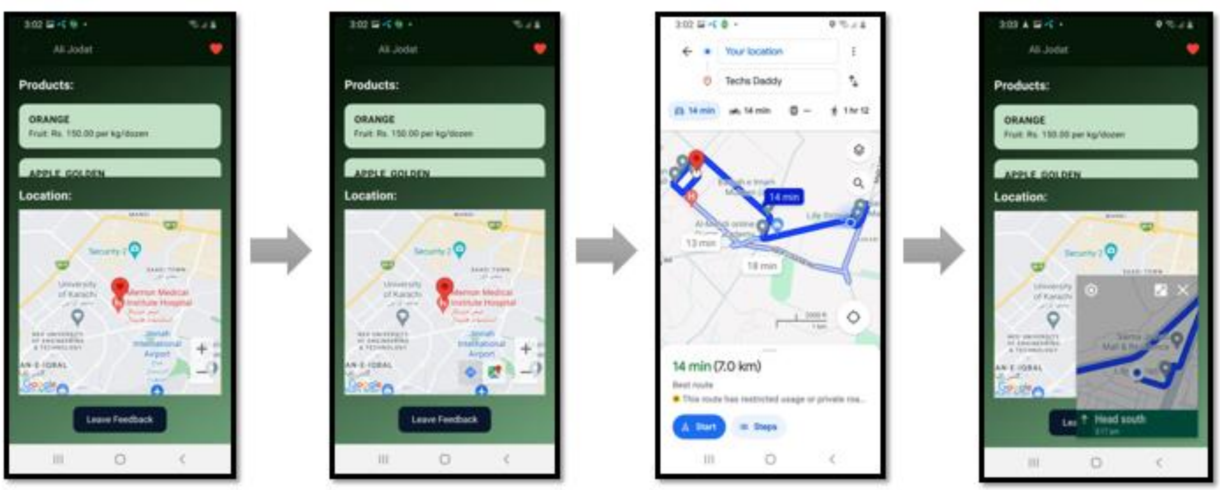


Figure 42 Nearby Vendor Cart Screens

6. Testing and Evaluation

6.1 Deep Learning Model Results

The deep learning results showed that MobileNet has the highest accuracy among the other models.

Table 18 Accuracies of Deep Learning Models Implemented

Model	Epochs executed	Accuracy
MobileNet	10	94.925%
ResNet	10	93.930%
AlexNet	30	92.437%
VGG16	10	92.288%
VGG19	10	92.039%
Inception-V3	10	89.203%

Following are the graph showing their accuracy and loss.

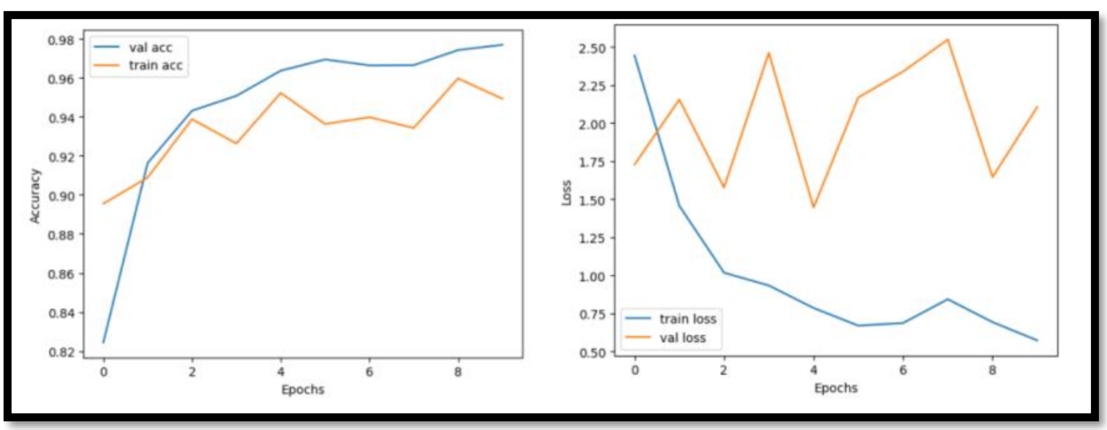


Figure 43 Graphs showing (a) Accuracy and (b) Loss of MobileNet.

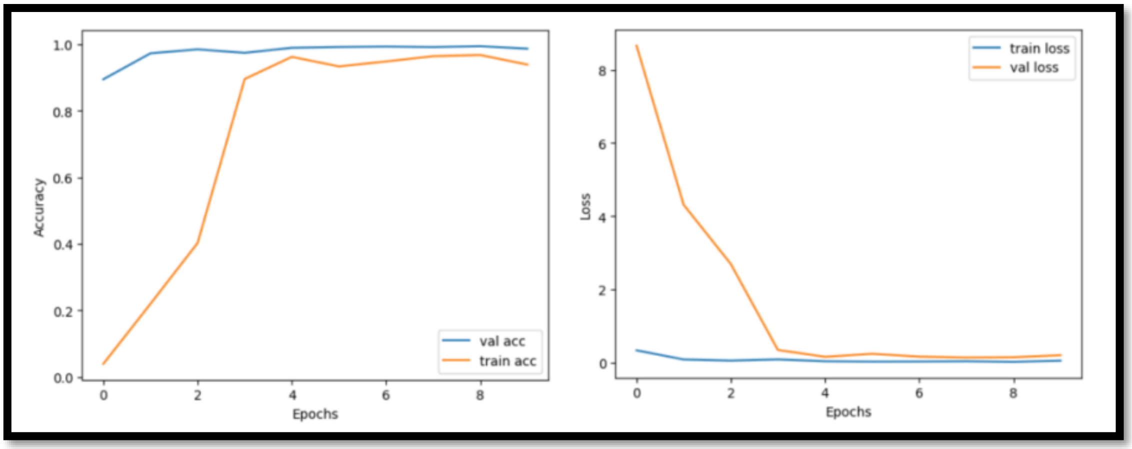


Figure 44 Graphs showing (a) Accuracy and (b) Loss of ResNet.

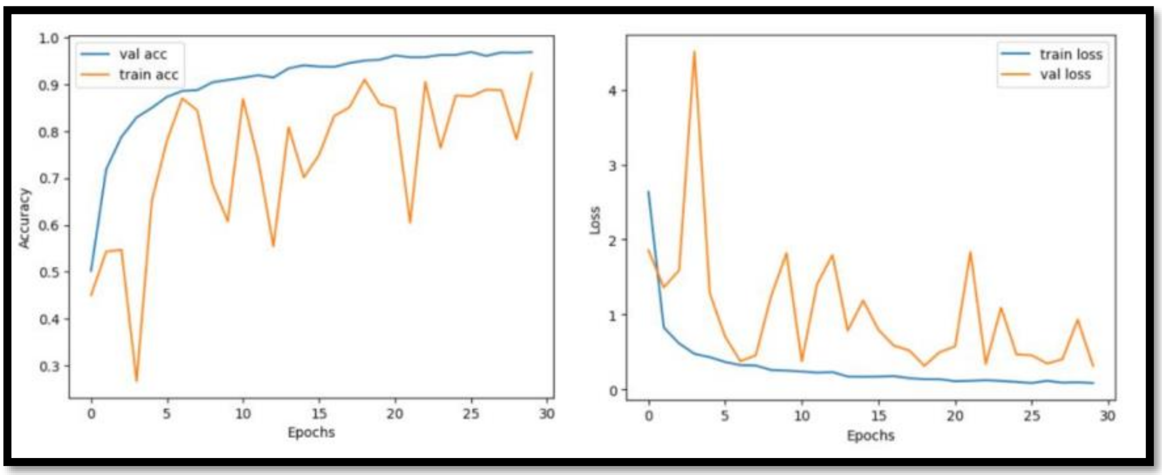


Figure 45 Graphs showing (a) Accuracy and (b) Loss of AlexNet.

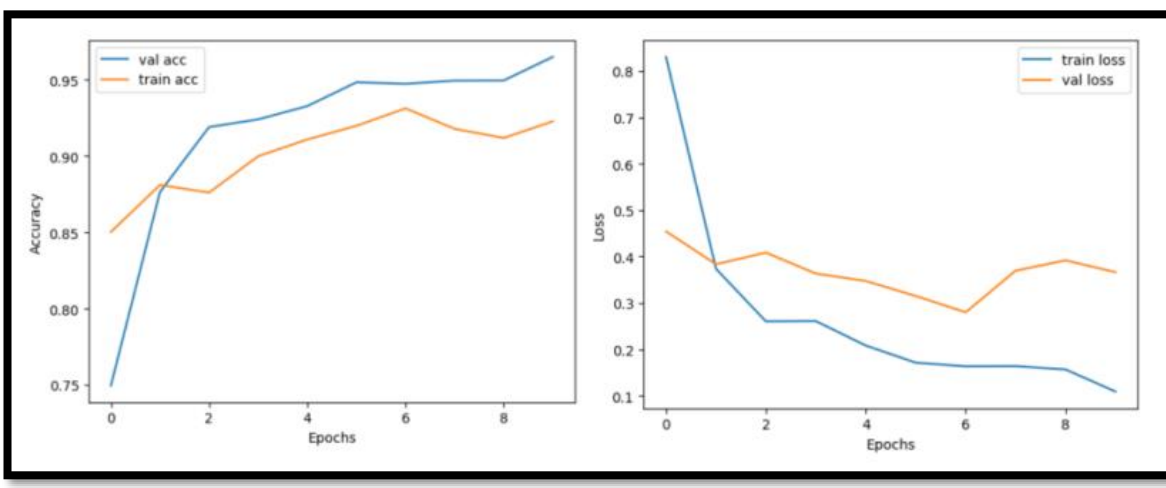


Figure 46 Graphs showing (a) Accuracy and (b) Loss of VGG16.

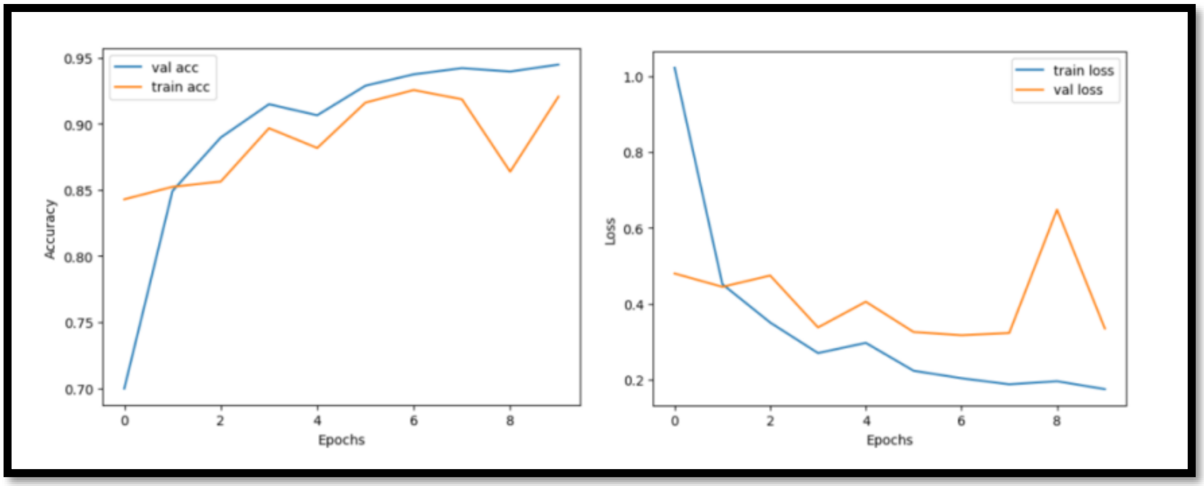


Figure 47 Graphs showing (a) Accuracy and (b) Loss of VGG19.

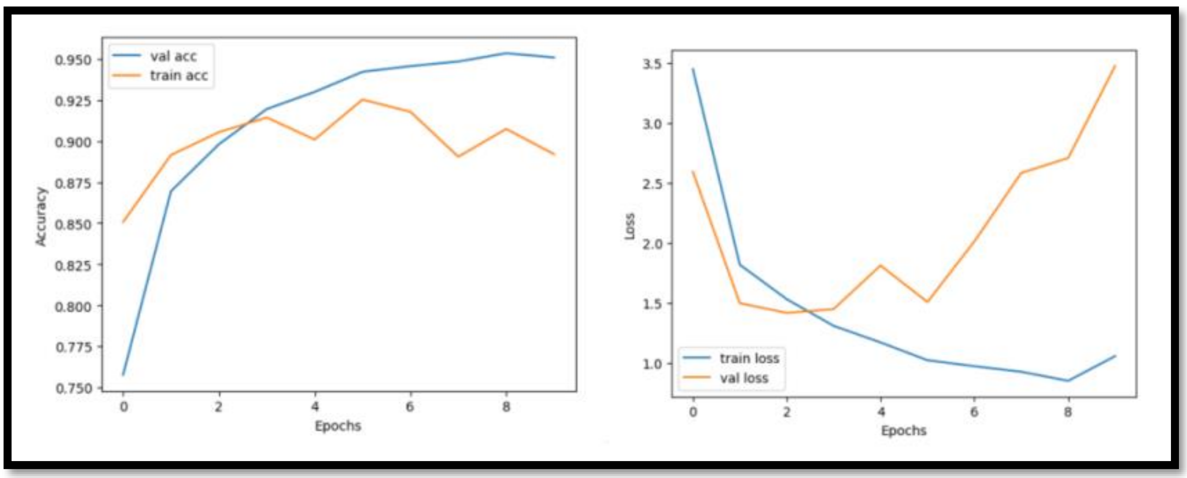


Figure 48 Graphs showing (a) Accuracy and (b) Loss of Inception-V3.

MobileNet, with its simple and efficient architecture, achieved the highest accuracy of all model. As MobileNet is a lightweight model it would be easy to deploy it on a mobile application as well.

6.2 YoloV8 Results

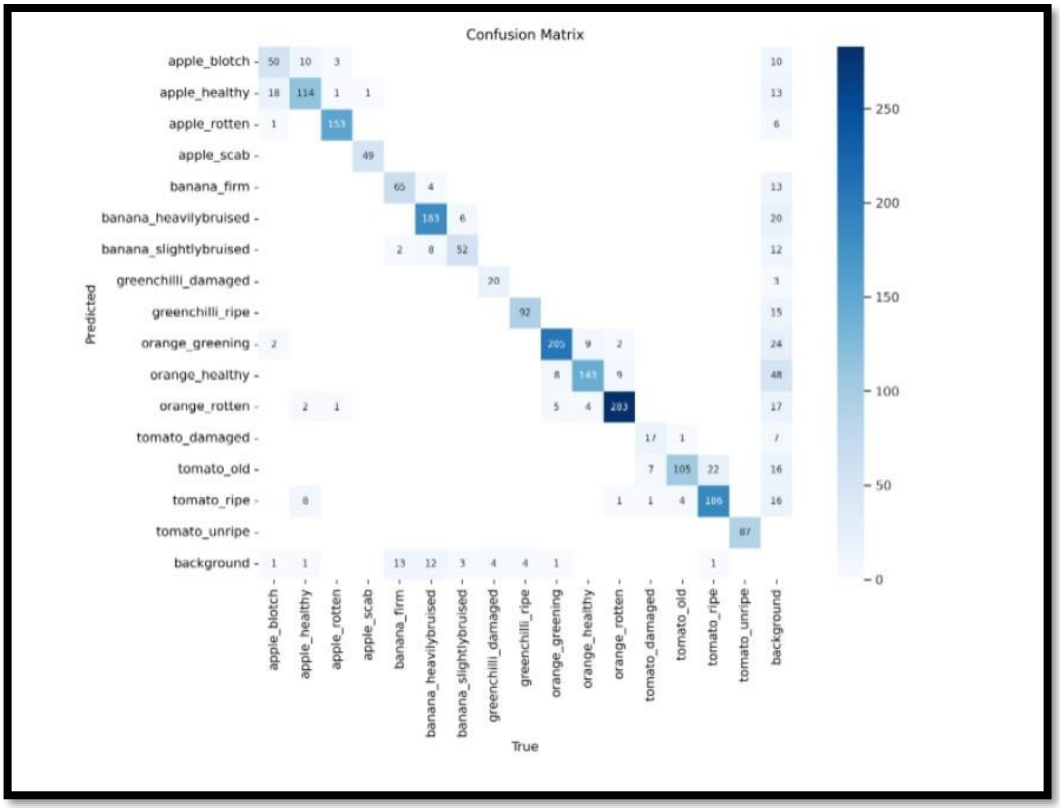


Figure 49 Confusion Matrix of YoloV8 Results

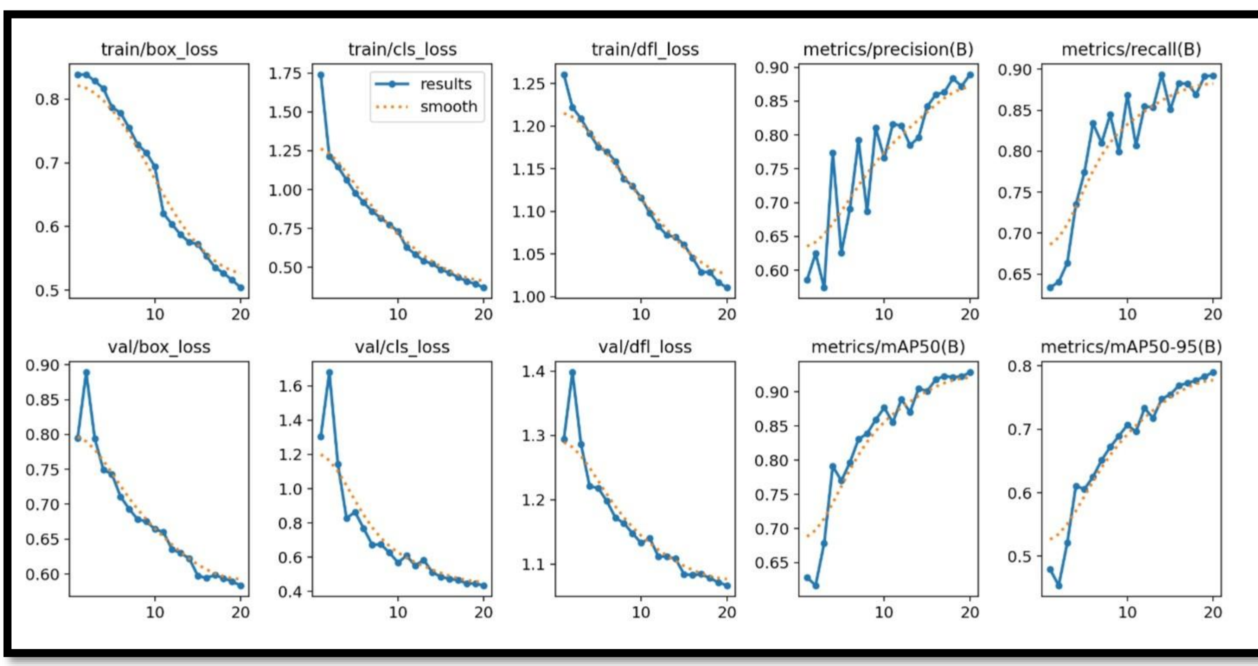


Figure 50 Validation and Loss Results

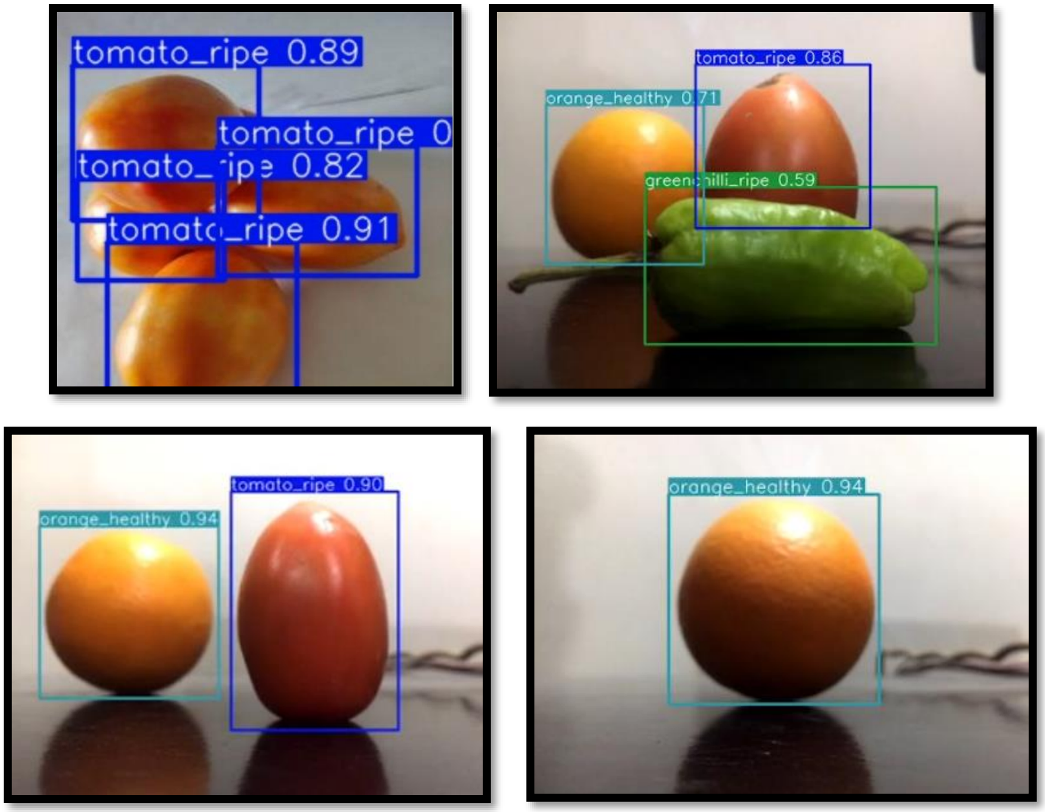


Figure 51 Some Results Sample

6.3 Purpose

The objective of this test plan document is to outline the comprehensive testing approach and strategies for ensuring the quality and functionality of IntelliCART application. By defining clear test objectives, criteria this document aims to facilitate effective coordination among project stakeholders. To achieve product quality. Aims to mitigate risk, detect early defect in development lifecycle and ensure application works well and meet end user expectations and requirements.

6.4 Validation

- Unit Testing: Test individual components of system to check system to function properly.
- Integration Testing: Integration of individual component to test combine functionality.
- System Testing: Entire system test for checking requirements.
- Interface Testing: Test interface as per business needs.
- Performance Testing: Ensure system work under heavy traffic.
- Compatibility Testing: Ensure system to works on different operating system.

6.5 Test Pass/Fail Criteria

If application crashes at any stage of testing, then it is failure else it is pass.

6.6 Test Cases

Table 19 Nearby Vendor Carts GPS Integration Test

TEST CASE ID: TC-1 **DESCRIPTION:** GPS Integration Test				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Enable Location and allow permission to app to access location.	Application should point to nearby carts with specified radius	Displaying nearby carts	Pass
2.	Input updated location by pressing update location button	Application should point to new nearby carts in that radius	Displaying different nearby carts	Pass
3.	Accurately displays nearby carts on map screen and provide tracking directions on map	Application should update real time as user moves and provided assistance in reaching that location	Providing real time assistance	Pass

Table 20 Customer/Vendor Login Test

TEST CASE ID: TC-2 **DESCRIPTION:** Customer/Vendor Login				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Input valid login credentials	User should be able to login	User logged in	Pass
2.	Input invalid login credentials	User should not be able to login and prompt invalid credentials	Prompts invalid credentials	Pass

Table 21 Vendor Product Details Management Test

TEST CASE ID: TC-3 **DESCRIPTION:** Vendor Product Details Management				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Adding new fruit or vegetable to product list	The addition should be reflected accurately in the product list	New fruit or vegetable added	Pass
2.	Updating price of existing fruit or vegetable	The modification should be reflected accurately in the product list	Price updated	Pass
3.	Removal of existing fruit or vegetable	The removal should be reflected accurately in the product list	Existing fruit or vegetable removed	Pass

Table 22 Government Price Transparency Check Test

TEST CASE ID: TC-4 **DESCRIPTION:** Price Transparency Check				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Open selected nearby vendors to check selling price and government allotted price	Mobile application should display selling price following government price	Displaying selling and government price	Pass
2.	Verify government prices are fetched and updated regularly	Should display updated price	Updating and displaying regularly	Pass

Table 23 Real Time Freshness Monitoring Test

TEST CASE ID: TC-5 **DESCRIPTION:** Real Time Freshness Monitoring				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Records video and press upload	Video should be uploaded and process under Yolo model for quality assessment and save results in database	Video being processed and results saved	Pass
2.	Fetch results and display results of quality assessment of fruits and vegetable under each vendor	Results should be displayed in percentages	Quality in percentages is being displayed	Pass

Table 24 Feedback and Favorites Testing

TEST CASE ID: TC-6 **DESCRIPTION: Feedback and Favorites testing**				
No.	**STEPS**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1.	Pressing heart button to add vendor to favorites	Application should add that vendor to favorites	Vendor added successfully	Pass
2.	Selecting feedback button to provide rating to that selected vendor	Application should add rating and calculate average and save that rating to database	Vendor feedback added successfully	Pass

7. Conclusion

Our proposed solution is an important step towards addressing demand for fresh and healthy food with the authenticity of prices as well especially when purchasing from traditional carts. Our innovative solution involves modern cutting edge technologies that includes computer vision, cloud computing, mobile application development, and Global positioning system (GPS) to provide customer with freshness and price validation transparency. All above mentioned technologies merging in single platform will result in customer vendor strong relationship build on trust and fair market competition among vendors.

Summing up the “IntelliCART” provides an exciting and innovative solution with capability of revolutionizing fruit and vegetable market, enhancing trust and transparency and provides health food to customer

8. References

[1] P. S. D. a. K. Jayasimha, "Intra class vegetable recognition system using deep learning," in *Maisamaguda, India, International Conference on Intelligent Computing and Control Systems*, 2018.

[2] E. L. F. Mandracchia, "Potential Use of Mobile Phone Applications for Agriculture," 2019.

[3] D. P. a. R. Jeewon, "Fruit and vegetable intake: benefits and progress of nutrition education," *Iranian Journal of Public Health,* vol. 44, no. 10, p. 1309–1321, 2015.

[4] S. N. N. S. a. G. L. S. Arivazhagan, "Fruit recognition using color and texture features," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 1, no. 1, pp. 90-94, 2020.

[5] S. A. T. R. a. R. K. N.-E.-A. Mimma, "Fruits Classification and Detection Application," Vols. 2022, p. 16, 17 November, p. 16, 17 November 2022.

[6] A. M. a. J. C. M. Mukhiddinov, "Improved Classification Approach for Fruits and Vegetables Freshness Based on Deep Learning," 26 October 2022.

[7] V. V. K. V. N. Gururaj, "Deep grading of mangoes using Convolutional Neural," *Multimedia Tools and Applications ,* vol. 82, p. 39525–39550, 2022.

[8] M. R. J. S. R. S. K. J. K. PR, "Development of an effective system to Identify Fruit ripening Stage for Apple, Banana and Mango," p. 2766–2772, 2020.

[9] D. a. Y. M. Srinivasan, "Apple Fruit Detection and Maturity Status Classification," vol. 9, no. 2, p. 1055–1059, 2020.

[10] R. C. J. M. K. D. M. J. R. B. A. Kumar, "Fruit-CNN: An Efficient Deep learning-based Fruit Classification and Quality Assessment for Precision Agriculture," in *13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, 2021.

[11] Z. G. F. D. B. U. T. P. a. C. M. I. Sa, "DeepFruits: A Fruit Detection System Using Deep Nueral Network," *Sensors,* vol. 16, p. 1222, 2016.

[12] M. A. e. al., "Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning," *Front. Plant Sci,* vol. 11, 2020.

[13] S. B. a. J. Underwood, "Deep fruit detection in orchards," *IEEE International Conference on Robotics and Automation (ICRA),* p. 3626–3633, 2017.

[14] K. a. Chen, "Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards," *Sensors,* vol. 19, no. 20, p. 4599, 2019.

[15] J. Z. a. T. R. A. Wu, "Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network," *Comput. Electric. Eng.,* vol. 81, p. 106454, 2020.

[16] Y. F. a. Z. W. S. Hou, "Vegfru: A domain-specific dataset for fine-grained visual categorization," *IEEE International Conference on Computer Vision,* p. 541–549, 2017.

[17] M. R. S. A. S. M. Jagdale, "Automatic Fruit Quality," in *IEEE*, 2016.

[18] S. A. T. A. M. U. I. A. A. R. M. S. Morshed, "Fruit Quality Assessment with Densely Connected Convolutional Neural Network," in *12th International Conference on Electrical and Computer Engineering (ICECE)*, 2022.

[19] A. A. A. M. A. S. S. Turaev, "Application of Transfer Learning for Fruits and Vegetable Quality Assessment," in *14th International Conference on innovation in Information and Technology (IIT)*, 2020.

[20] P. S. D. a. K. Jayasimha, "Intra class vegetable recognition system using deep learning," in *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS)*, Maisamaguda, India, 2020.

[21] A. T.-G. a. Y. D. Z. A. Nasiri, "Image based deep learning automated sorting of date fruit,," *Post harvest Biology and Technology,* vol. 153, p. 133–141, 2019.