APPLE FRESHNESS DETECTION AND DEPTH CALCULATION SYSTEM

PROJECT REPORT

submitted by

MUNAVIR ZAMAN P K

TVE22ECRA13

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master of Technology

In

ROBOTICS AND AUTOMATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

KERALA

JUNE 2024

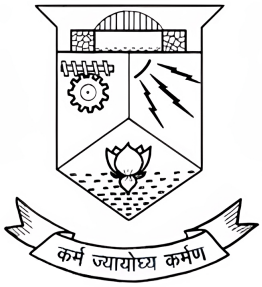
DECLARATION

I Munavir Zaman P K hereby declare that the project report APPLE FRESHNESS DETECTION AND DEPTH CALCULATION SYSTEM, submitted for partial fulfilment of the requirements for the award of the degree of Master of Technology with specialization **ELECTRONICS AND COMMUNICATION ENGINEERING** of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of **Dr. Sreeja S**, Associate Professor, Department of EEE. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

TRIVANDRUM **Munavir Zaman P K**

JUNE 2024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM
THIRUVANANTHAPURAM



CERTIFICATE

This is to certify that this project report entitled **APPLE FRESHNESS DETECTION AND DEPTH CALCULATION SYSTEM** is a bonafide record of the work done by **MUNAVIR ZAMAN P K**, under our guidance towards partial fulfilment of the requirements for the award of the Degree of **Master of Technology in Electronics and Communication** with specialization in Robotics and Automation, of the A P J Abdul Kalam Technological University during the year 2022-2024 .

Dr. Sreeja S
(Project Guide)
Associate Professor
Dept. of EEE
College of Engineering Trivandrum

Dr. Binu LS
(Project Coordinator)
Professor
Dept. of ECE
College of Engineering Trivandrum

Dr. Christy James Jose
(Pg Coordinator)
Associate Professor
Dept. of ECE
College of Engineering Trivandrum

Dr. Haris P. A
(Head of Department)
Professor
Dept. of ECE
College of Engineering Trivandrum

ACKNOWLEDGEMENT

I would like to take this opportunity to extend my sincere thanks to the people who helped me to make this project possible. This project will be incomplete without mentioning all the people who helped me to make it real.

Firstly, I would like to thank GOD, Almighty, our supreme guide, for bestowing His blessings upon me in my entire endeavor.

I am highly indebted to **Dr. Sreeja S**, Associate Professor, Department of EEE and my project guide for the valuable advice and support throughout the project work. With immense pleasure and heartiest gratitude, I express my sincere thanks to **Dr. Binu L S**, Professor, Department of ECE and project coordinator, for his valuable suggestions and guidance. I also thank **Dr. Christy James Jose**. Associate Professor, Department of ECE and PG Co-ordinator for all his support.

I take this opportunity to express my sincere gratitude to **Dr. Haris P. A**, HoD, Department of ECE and the Principal **Dr. Savier J S**. for providing all necessary facilities and resources.

I extend my sincere thanks to my family and friends without whose comments, criticism, co-operation, and tremendous support, this project would not have been a success.

ABSTRACT

The agricultural industry continually seeks innovative methods to improve quality control and reduce waste. This project addresses the need for an automated solution for apple freshness detection and depth calculation. The developed system utilizes a mobile robot equipped with a stereo camera setup and advanced machine learning algorithms. The core of the system is the YOLOv8 model, trained to classify apples as fresh or rotten. The stereo camera setup, composed of two ESP32 cameras, is calibrated to compute depth information accurately. The robot navigates using teleoperation control and processes images in real-time to detect apple freshness and measure depth.

Extensive testing demonstrated the system's high accuracy, achieving 97% accuracy for fresh apple detection and 85% for rotten apple detection. The stereo camera setup provided reliable depth measurements, ensuring precise spatial awareness of the detected apples. This project highlights the integration of computer vision and robotics to enhance agricultural quality control processes, offering a practical and effective solution for automated apple freshness detection and depth calculation. The findings suggest that with further improvements, this system could be commercially viable, benefiting the agricultural industry significantly.

CONTENTS

List of Figures

ABBREVIATIONS

ROS	Robot Operating System
OpenCV	Open Source Computer Vision Library
Rviz	ROS visualization
URDF	Unified Robot Description Format
XML	Extensible Markup Language
GMapping	Grid-based Mapping
UI	User Interface
GUI	Graphical User Interface

Chapter 1

INTRODUCTION

1.1 Background

In the agricultural industry, the quality and freshness of produce are critical factors that directly impact consumer satisfaction and market value. Apples, in particular, are one of the most widely consumed fruits globally, and their freshness is a key determinant of their quality. Traditional methods of assessing apple freshness involve manual inspection, which can be time-consuming, subjective, and prone to human error. With the advent of advanced technologies in computer vision and machine learning, there is a growing interest in developing automated systems to improve the accuracy and efficiency of freshness detection. This project also aims to develop an advanced apple ripeness detection system using the YOLOv8 object detection model.

Integrating depth calculation into the apple freshness detection system adds a valuable dimension to the assessment process. By utilizing stereo cameras or other depth-sensing technologies, the system can measure the three-dimensional structure of apples, providing insights into their size, shape, and possible surface deformities. This additional layer of information enhances the model's ability to distinguish between different ripeness stages more accurately and identify potential defects that might not be visible in two-dimensional images. Combining YOLOv8's high-speed and precise object detection capabilities with depth analysis creates a robust, real-time solution

that significantly improves the reliability of apple quality assessments. This innovative approach aims to streamline the sorting and grading process in agricultural facilities, ensuring that only the freshest and highest-quality apples make it to the market, ultimately benefiting both producers and consumers.

1.2 Problem Statement

The manual inspection of apples for freshness is not only labor-intensive but also inconsistent, as it relies heavily on the experience and judgment of the inspector. Additionally, manual methods do not provide detailed quantitative data, such as the exact ripeness stage or the spatial positioning of the apples. There is a clear need for an automated system that can accurately and consistently detect apple freshness and provide precise depth information to facilitate better sorting and processing.And also current methods for determining apple ripeness are often manual and subjective, leading to inefficiencies and inconsistencies in quality control.

1.3 Purpose

The purpose of this project is to design and develop an autonomous mobile robot capable of detecting the freshness of apples and calculating their depth using a stereo camera setup. By leveraging the power of machine learning algorithms, specifically the YOLOv8 model, and computer vision techniques, the system aims to provide a reliable and efficient solution for apple freshness detection. This will help reduce waste, improve quality control, and enhance the overall efficiency of the sorting process in the agricultural industry.

1.4 Objectives

The primary objectives of this project are as follows:

- Develop and train a machine learning model using YOLOv8 to classify apples as fresh or rotten.
- Implement a stereo camera setup using ESP32 cameras to calculate the depth of apples.
- Integrate the object detection and depth calculation functionalities into a mobile robot platform.
- Ensure real-time processing capabilities for the system to provide immediate feedback.
- To create a robust AI-based system that can accurately classify apples as ripe, unripe, or overripe, improving sorting and quality control processes.

1.5 Scope

This project encompasses the entire process of designing, developing, and implementing the mobile robot system. It includes the selection of appropriate hardware components, the development of software algorithms for object detection and depth calculation, and the integration of these components into a functional robot. The system is tested and evaluated in real-world scenarios to ensure its effectiveness and reliability.The project also focuses on training the YOLOv8 model to detect apple ripeness stages using high-quality images, with no hardware constraints.

1.6 Significance

The development of an automated apple freshness detection and depth calculation system has significant implications for the agricultural industry. It offers a scalable solution that can be deployed in various settings, from small farms to large-scale processing facilities. By automating the freshness detection process, the system can reduce labor costs, minimize human error, and ensure consistent quality control.

Moreover, the depth calculation capability provides valuable spatial information that can be used for precise sorting and handling of apples.

Chapter 2

LITERATURE REVIEW

Fruit freshness detection is a critical aspect of quality control in the agricultural and food processing industries. Over the years, various non-destructive techniques have been developed to assess fruit freshness, aiming to improve the accuracy and efficiency of quality control processes.

[1] proposed a deep learning-based technique for fruit freshness detection. Their approach leverages the capabilities of convolutional neural networks (CNNs) to analyze fruit images and classify them based on freshness. By training the CNN on a large dataset of annotated fruit images, the model achieved high accuracy in distinguishing between fresh and rotten fruits.

[2] proposed an image processing approach using Gabor filter and support vector machine (SVM) for fruit freshness detection. The Gabor filter was used to extract relevant features from fruit images, which were then fed into an SVM classifier for freshness classification. This method demonstrated promising results in accurately identifying the freshness status of fruits.

[3] focused on non-destructive measurement of mango internal quality using a portable near-infrared spectrometer. By analyzing the near-infrared spectra of mango samples, they were able to predict the internal quality attributes such as sugar content and firmness, which are indicative of fruit freshness.

[4] explored the use of near-infrared spectroscopy for non-destructive detection

of the internal quality of intact kiwifruit. By analyzing the near-infrared spectra of kiwifruit samples, they were able to predict the soluble solids content and firmness, which are key indicators of fruit freshness.

[5] proposed a method for non-destructive detection of total soluble solids content and firmness of cherry tomatoes using near-infrared spectroscopy and chemometrics. Their approach involved developing predictive models based on the near-infrared spectra of cherry tomatoes, which were then used to estimate the total soluble solids content and firmness of the tomatoes, thus assessing their freshness.

[6] introduced a machine learning approach for non-destructive detection of fruit freshness. Their method involved training a machine learning model on a dataset of fruit images with annotated freshness labels. The model was then able to classify new fruit images into fresh or rotten categories with high accuracy, demonstrating the potential of machine learning in fruit freshness detection.

[7] investigated the non-destructive determination of soluble solids content and firmness of intact pear using near-infrared spectroscopy. By analyzing the near-infrared spectra of pear samples, they were able to predict the soluble solids content and firmness, providing valuable information for assessing fruit freshness.

[8] studied the detection of apple fruit freshness and firmness using hyperspectral imaging technique. By capturing hyperspectral images of apple samples, they were able to extract spectral features related to freshness and firmness, which were then used to develop predictive models for assessing apple freshness.

[9] proposed a method for non-destructive detection of tomato seedling freshness based on machine vision. Their approach involved capturing images of tomato seedlings and using image processing techniques to extract features related to freshness. A machine learning model was then trained on these features to classify tomato seedlings into fresh or non-fresh categories.

Ripeness detection in fruits is also a crucial aspect of the agricultural industry, impacting both quality control and market value. Various technologies and methodologies have been explored to automate the ripeness detection process, ranging from

sound analysis and image processing to gas sensors and fuzzy logic.

[11] explored the use of sound data for ripeness classification of durian fruits. The study utilized Mel-frequency cepstral coefficients (MFCC) to extract features from knocking sounds and fed these features into a convolutional neural network (CNN) for classification. The CNN architecture comprised four layers, including two convolutional layers with ReLU activation, a max-pooling layer, and a fully connected output layer. The dataset was divided into 70% training, 20% validation, and 10% testing data. The model achieved a validation accuracy of 90.78% and a testing accuracy of 89.47%, demonstrating the potential of acoustic analysis for fruit ripeness detection .

[12] implemented an image processing technique using OpenCV and HSV color space to detect the ripeness of tomatoes. The process involved enhancing images through histogram equalization, converting color space to HSV, and applying thresholding based on HSV values. Morphological processing and contour detection were then used to identify ripe and turning fruits. This method leveraged the computational power of Raspberry Pi and VNC Viewer for real-time analysis, showcasing a practical application of image processing in agricultural robotics .

[13] focused on semantic segmentation for detecting ripened strawberry guava fruits using a modified U-Net architecture. This deep learning model employed a U-shaped structure with an encoder-decoder configuration to achieve pixel-level segmentation. The model reported a validation dice score of 91.04% and a testing dice score of 89.72%, indicating high accuracy. However, challenges remained in detecting occluded or smaller ripe fruits .

[14] utilized the YOLO v5 algorithm for real-time detection and classification of Trinitario cocoa fruits based on their ripening stage. The YOLO v5 network, consisting of a backbone for feature extraction, a neck for combining features, and a head for generating bounding boxes and predictions, was trained on both augmented and non-augmented datasets. The augmented dataset, improved using the mosaic-12 method, achieved an accuracy of 60.2% compared to 56% with the non-augmented dataset,

demonstrating the effectiveness of data augmentation in enhancing model performance

.

[16] proposed the DeRiBa-Fuz architecture, utilizing fuzzy logic combined with RGB and GLCM (Grey Level Co-occurrence Matrix) feature extraction to classify banana ripeness. The model categorized bananas into seven distinct ripeness stages with high accuracy, achieving 100% accuracy in six out of seven categories. This approach highlighted the utility of combining fuzzy logic with traditional image processing techniques for nuanced classification tasks .

[18] developed a non-destructive ripeness detection system for jackfruits using gas sensors (MQ3, MQ5, MQ7, and MQ135). The prototype device, integrating these sensors with a microcontroller, measured gas concentrations emitted by jackfruits to determine their ripeness levels. The system achieved a classification accuracy of 96%, demonstrating the viability of gas sensor technology in assessing fruit ripeness based on volatile organic compounds .

Each methodology offers unique advantages and challenges. Sound analysis with MFCC and CNN shows promise in non-visual contexts but requires precise audio recordings. Image processing techniques, particularly with advanced architectures like U-Net and YOLO, provide high accuracy but may struggle with occluded fruits or require significant computational resources. Fuzzy logic combined with GLCM and RGB extraction provides a robust framework for classification but may be limited by the complexity of fuzzy rule definition. Gas sensor-based detection offers a non-invasive and highly accurate alternative, although it may be specific to certain types of fruits.

2.1 Research gap

Existing systems that simultaneously determine the depth of apples and detect their freshness are rare.Therefore, this project aims to develop a mobile robot that can navigate to desired locations and perform both tasks. The robot will utilize

advanced computer vision and machine learning techniques to assess apple ripeness accurately. Additionally, it will employ depth calculation methods to measure the three-dimensional structure of apples, enhancing the precision of the ripeness evaluation. This integrated approach will streamline the quality assessment process, benefiting both producers and consumers by ensuring only the freshest apples reach the market.

Chapter 3

System decription

3.1 Hardware Components

3.1.1 ESP32 Cameras

Description: ESP32 cameras are small, low-cost devices capable of capturing high-resolution images and transmitting them over Wi-Fi. They are based on the ESP32 microcontroller, which integrates Wi-Fi and Bluetooth functionalities, making them suitable for IoT applications.

Specifications:

- Resolution: Up to 2MP (1600x1200 pixels)
- Connectivity: Wi-Fi 802.11 b/g/n, Bluetooth 4.2
- Interface: I2C, SPI, UART
- Power Supply: 3.3V

Function: These cameras capture images of apples in real-time for the object detection and depth calculation processes. Two ESP32 cameras are used in a stereo configuration to calculate the depth of the detected objects.

3.1.2 Raspberry Pi 4 Model 2GB

Description: Raspberry Pi 4 is a powerful single-board computer that serves as the main processing unit for the project. It runs the necessary software for image processing, object detection, and robot control.

Specifications:

- CPU: Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 2GB LPDDR4-3200 SDRAM
- Connectivity: Gigabit Ethernet, 2.4/5.0 GHz 802.11ac wireless, Bluetooth 5.0
- Ports: 2 x USB 3.0, 2 x USB 2.0, 2 x micro HDMI ports

Function: The Raspberry Pi processes the images captured by the ESP32 cameras, runs the YOLOv8 object detection model, and calculates the depth of the detected apples. It also handles the robot's movement and control commands.

3.1.3 Gear Motors: SPG30E-60K DC Geared Motor with Encoder 75 RPM (12V)

Description: These are high-torque DC geared motors with encoders, suitable for driving the robot's wheels.

Specifications:

- Voltage: 12V
- Speed: 75 RPM
- Torque: 1.3 Nm
- Encoder: 12 PPR (Pulses Per Revolution)

Function: The motors provide the necessary propulsion for the mobile robot, enabling it to navigate through the environment. The encoders provide feedback on the motor's rotation, essential for precise movement control.

3.1.4 Motor Drivers: L298N

Description: L298N is a dual H-bridge motor driver that allows for controlling the direction and speed of the motors.

Specifications:

- Operating Voltage: 5V to 35V
- Max Current: 2A per channel
- Control: PWM (Pulse Width Modulation) for speed control

Function: The motor driver interfaces between the Raspberry Pi and the gear motors, enabling control over the motors' speed and direction.

3.1.5 Microcontroller: ATmega328P

Description: ATmega328P is an 8-bit microcontroller used in Arduino boards, responsible for handling basic I/O operations and sensor data processing.

Specifications:

- Operating Voltage: 1.8V to 5.5V
- Flash Memory: 32 KB
- CPU Speed: 20 MHz
- Interfaces: I2C, SPI, UART

Function: The microcontroller handles auxiliary tasks such as reading sensor data and controlling additional peripherals not managed by the Raspberry Pi.

3.1.6 Power Supply: 12V

Description: A 12V power supply unit (PSU) provides the necessary power for the entire system, including the Raspberry Pi, motors, and other components.

Specifications:

- Output Voltage: 12V
- Output Current: Up to 5A

Function: Ensures that all components receive a stable and sufficient power supply to operate efficiently.

3.1.7 Keyboard: For Teleoperation Control

Description: A standard keyboard is used to send control commands to the robot during teleoperation.

Function: The keyboard allows the user to manually control the robot's movements, providing inputs for direction and speed adjustments.

3.2 Software Components

3.2.1 OpenCV

Description: OpenCV (Open Source Computer Vision Library) is a comprehensive library for computer vision tasks. It provides tools for image and video processing, including functions for filtering, feature detection, and object tracking.

Key Features:

- Image processing: Filters, transformations, and enhancements
- Video analysis: Motion tracking and object detection
- Machine learning: Pre-trained models and tools for training custom models

Function: Used for image preprocessing, such as resizing, filtering, and enhancing images captured by the ESP32 cameras before they are processed by the YOLOv8 model.

3.2.2 YOLOv8

Description: YOLOv8 (You Only Look Once, Version 8) is a state-of-the-art deep learning model for real-time object detection. It detects objects in images and videos with high accuracy and speed.

Key Features:

- Real-time object detection
- High accuracy and speed
- Pre-trained on large datasets and can be fine-tuned for specific tasks

Function: The YOLOv8 model is trained to classify apples as fresh or rotten based on their appearance. It processes the preprocessed images and outputs the detected objects along with their bounding boxes and confidence scores.

3.2.3 Python

Description: Python is a high-level programming language widely used for its simplicity and versatility. It is the primary language for developing the software components of this project.

Key Features:

- Extensive libraries and frameworks for machine learning, computer vision, and robotics
- Easy-to-read syntax and rapid development capabilities

Function: Python scripts are used for integrating the various software components, processing images, running the YOLOv8 model, calculating depth, and controlling the robot.

3.2.4 ROS (Robot Operating System)

Description: ROS is a flexible framework for writing robot software. It provides tools and libraries for building complex and robust robotic applications.

Key Features:

- Modular design with reusable software components
- Tools for simulation, visualization, and debugging
- Support for multiple programming languages and hardware platforms

Function: ROS is used to manage the communication between the different parts of the system, such as the Raspberry Pi, cameras, and motors. It also provides tools for visualizing the robot's state and sensor data.

Chapter 4

Methodology

4.1 Introduction

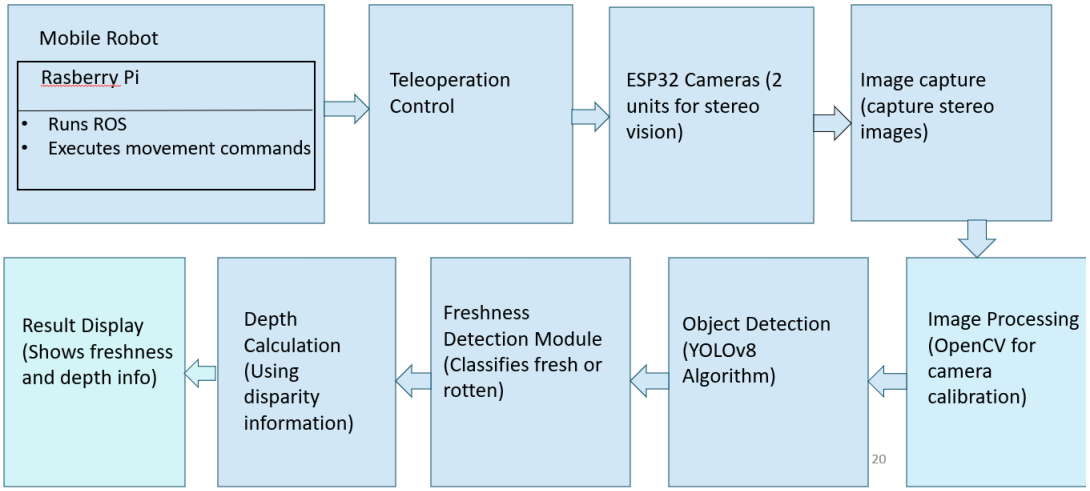


Figure 4.1: Block diagram of methodology

Figure 4.1 shows a comprehensive system architecture for a mobile robot designed to perform freshness detection using stereo vision and object detection algorithms. The core of the system is a Raspberry Pi, which runs the Robot Operating System (ROS) and manages the robot's movement commands. Teleoperation control facilitates remote navigation of the robot. The vision system includes two ESP32 cameras configured for stereo vision, capturing images that are processed using OpenCV for camera calibration. The processed images are then analyzed using the YOLOv8 (You

Only Look Once version 8) algorithm to detect objects. A freshness detection module subsequently classifies these objects as fresh or rotten. Depth information, essential for spatial context, is calculated using disparity data from the stereo images. The final output, which includes both freshness classification and depth information, is displayed for the operator's review. This methodology aims to integrate advanced vision and detection technologies to automate the process of assessing the freshness of items, with potential applications in agricultural monitoring, food quality assessment, and automated inspection systems.

4.2 Hardware Setup

4.2.1 ESP32 Cameras Configuration

The ESP32 cameras are configured to capture high-resolution images and stream them over Wi-Fi. The configuration process includes the following steps:

- **Initial Setup:** Connect the ESP32 cameras to a computer via USB for initial configuration. Install the necessary drivers and firmware provided by the manufacturer. This step ensures that the cameras can be programmed and configured correctly.
- **Firmware Flashing:** Use the ESP32 development environment (such as Arduino IDE or PlatformIO) to flash the firmware onto the ESP32 cameras. The firmware should include the code for Wi-Fi connectivity and image capturing functionality.
- **Wi-Fi Configuration:** Configure the cameras to connect to the local Wi-Fi network. This involves setting the SSID and password in the firmware code. The ESP32 cameras will use this configuration to connect to the network and stream images.

- **Synchronization:** Ensure both cameras are synchronized to capture images simultaneously. This is critical for accurate depth calculation. The synchronization can be achieved by implementing a timestamping mechanism or using an external trigger signal.

- **Mounting and Calibration:** Mount the ESP32 cameras on the robot in a fixed stereo configuration. Calibrate the cameras to correct for any misalignments and ensure accurate depth calculations. Calibration involves capturing images of a known pattern (e.g., a chessboard) and using software tools to adjust the camera parameters.

4.2.2 Raspberry Pi 4 Integration

The Raspberry Pi 4 serves as the central processing unit for the system. The integration involves the following steps:

- **Operating System Installation:** Install the latest version of Raspbian OS on the Raspberry Pi. This operating system provides a stable and versatile environment for running the software components.

- **Library Installation:** Install necessary libraries and dependencies, including OpenCV for image processing, YOLOv8 for object detection, and ROS for robotic control. These libraries provide the essential functionalities needed for the project.

- **Peripheral Connections:** Connect the Raspberry Pi to the ESP32 cameras via Wi-Fi. Also, connect the motor drivers and other peripherals to the GPIO pins of the Raspberry Pi. Ensure all connections are secure and correctly configured.

- **Power Management:** Ensure a stable power supply to the Raspberry Pi and other components. Use a regulated 12V power supply and appropriate voltage regulators to power the Raspberry Pi (5V) and the motors (12V).

4.2.3 Motor and Microcontroller Setup

The motors and microcontroller setup is crucial for the robot's movement and sensor integration. The steps include:

- **Motor Driver Configuration:** Connect the L298N motor drivers to the Raspberry Pi GPIO pins. Configure the PWM signals for speed control and direction signals for motor control. Ensure the motor drivers are properly powered and connected to the motors.
- **Microcontroller Programming:** Program the ATmega328P microcontroller (used in Arduino) to handle sensor data and control additional peripherals. Upload the firmware using the Arduino IDE and test the communication between the microcontroller and the Raspberry Pi.
- **Sensor Integration:** Connect any additional sensors (e.g., ultrasonic sensors, infrared sensors) to the microcontroller. Program the microcontroller to read the sensor data and send it to the Raspberry Pi for processing.

4.3 Software Development

4.3.1 Image Processing with OpenCV

OpenCV is used for preprocessing the images captured by the ESP32 cameras. The image processing steps include:

- **Image Acquisition:** Capture images from the ESP32 cameras and transfer them to the Raspberry Pi via Wi-Fi. Ensure that the images are received without significant delay or loss.
- **Preprocessing:** Apply image preprocessing techniques such as resizing, filtering, and enhancing. Resizing ensures that the images have a consistent size for further processing. Filtering (e.g., Gaussian blur) helps in reducing noise, and enhancing (e.g., histogram equalization) improves image contrast.

- **Image Calibration:** Use the calibration parameters obtained during the hardware setup to rectify the images. This step corrects any distortions and aligns the images for accurate depth calculation.

4.3.2 Object Detection with YOLOv8

The YOLOv8 model is used for detecting apples in the images and classifying them as fresh or rotten. The object detection process includes:

- **Model Training:** Train the YOLOv8 model on a dataset of apple images. The dataset should include labeled images of fresh and rotten apples. Use data augmentation techniques to increase the diversity of the training data.
- **Model Deployment:** Deploy the trained YOLOv8 model on the Raspberry Pi. Ensure that the model can run efficiently on the limited computational resources of the Raspberry Pi.
- **Detection Process:** Use the YOLOv8 model to process the preprocessed images and detect apples. The model outputs the bounding boxes, class labels, and confidence scores for each detected apple.
- **Classification:** Based on the class labels, classify the detected apples as fresh or rotten. Store the detection results for further processing.

4.3.3 Apple ripeness detection using YOLOv8

Detecting apple ripeness using YOLOv8, a state-of-the-art object detection algorithm, can be quite effective. Here's a general methodology you can follow:

- **Data Collection**:

  Gather a diverse dataset of apple images at different stages of ripeness. This dataset should include images of unripe, ripe, and overripe apples in various lighting conditions and backgrounds.

(a) Unripe (b) Ripe (c) Overripe

Figure 4.2: Apple images at different stages of ripeness.

The figure 4.2 illustrates apples at three different stages of ripeness: unripe, ripe, and overripe. Unripe apples are typically green and firm, indicating that they have not yet reached their full sweetness and flavor potential. Ripe apples, shown in the middle stage, exhibit vibrant colors—ranging from red to yellow depending on the variety—and have a balanced firmness, sweetness, and juiciness, representing the optimal stage for consumption. Overripe apples, depicted in the final stage, often appear duller in color with soft, bruised spots and may exhibit signs of wrinkling or decay. These visual cues are critical for the freshness detection module in our proposed system, as they enable the algorithm to accurately classify the freshness of apples based on their stage of ripeness.

- **Data Annotation**:

  Annotate the images with bounding boxes around the apples and label them according to their ripeness level (e.g., unripe, ripe, overripe).

Figure 4.3: Annotated Apples

The figure 4.3 depicts apples annotated with a bounding box and labeled with

a classification of either ”ripe” or ”unripe.” This annotation is a critical step in the object detection and classification process. The bounding box clearly delineates the area in the image that contains the apple, allowing the object detection algorithm (in this case, YOLOv8) to focus on this specific region. The accompanying label—either ”ripe” or ”unripe”—indicates the assessed ripeness status of the apple based on visual characteristics such as color, texture, and possibly other features. This annotated image serves as a training example for the ripeness detection module, enabling the system to learn and accurately classify the ripeness of apples in real-world scenarios. Through such annotated images, the algorithm can improve its accuracy in detecting and categorizing the ripeness of apples, thereby enhancing the overall effectiveness of the freshness detection system

- **Data Preprocessing**:

  Resize the images to a standard size suitable for YOLOv8 input. Augment the dataset to increase its diversity and robustness. Augmentation techniques may include rotation, flipping, scaling, and adjusting brightness and contrast.

- **Model Training**:

  Utilize a pre-trained YOLOv8 model as a starting point. Fine-tune the model on your annotated dataset using transfer learning. This process helps the model adapt to the specific characteristics of apple ripeness detection. Use appropriate loss functions such as YOLO’s combination of localization loss, confidence loss, and classification loss.

- **Model Evaluation**:

  Evaluate the trained model on a separate validation dataset to assess its performance. Metrics such as precision, recall, and F1-score can be useful for evaluation. Post-processing:

  Implement post-processing techniques to filter out false positives and refine the

detection results. Non-maximum suppression (NMS) is a common technique used to eliminate redundant bounding boxes.

- **Deployment**:

  Deploy the trained model on the desired platform, whether it's a desktop application, mobile app, or embedded system. Implement a user-friendly interface for users to input images and view the detection results.

- **Continuous Improvement**:

  Monitor the model's performance in real-world scenarios and collect feedback from users. Periodically retrain the model with new data to improve its accuracy and generalization capability.

4.3.4 Depth Calculation

Depth calculation involves determining the distance of detected apples from the cameras. The steps include:

- **Stereo Image Processing:** Use the synchronized images from the two ESP32 cameras for stereo vision. Apply stereo matching algorithms to find correspondences between the left and right images.

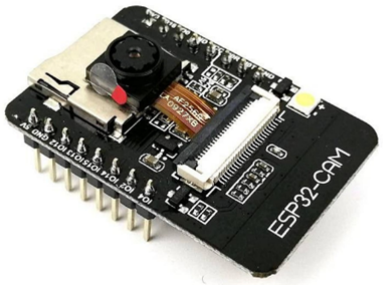


Figure 4.4: ESP 32 OV2640

Figure 4.4 shows an ESP32 OV2640 Camera.The ESP32 OV2640 camera is a compact and versatile camera module designed for use with the ESP32 microcontroller. The OV2640 sensor provides a resolution of 2 megapixels, capable of capturing images up to 1600x1200 pixels. It supports various image formats such as JPEG, RGB, and YUV, making it suitable for a wide range of applications including image capture, video streaming, and basic image processing tasks. The module is known for its low power consumption and ease of integration with the ESP32, which features Wi-Fi and Bluetooth connectivity. This makes the ESP32 OV2640 camera ideal for IoT projects, home automation, security systems, and other embedded applications where wireless communication and image capture are required. Its small form factor and affordable price further contribute to its popularity in both hobbyist and professional projects.

The stereo camera setup we made using 2 esp32 cameras is given below

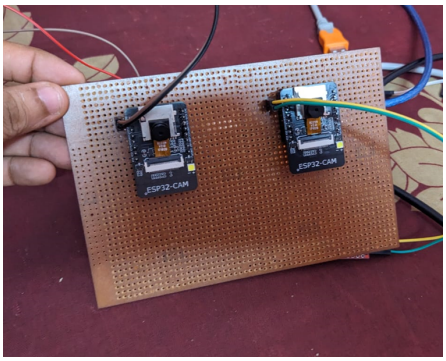


Figure 4.5: Developed camera module

The stereo camera setup involves using two ESP32 cameras placed 7 cm apart, fixed onto a PCB (Printed Circuit Board) to maintain a consistent distance and alignment. This distance is chosen to simulate the human eye's stereoscopic vision, allowing for depth perception in the captured images. Each camera is connected to an Arduino or an FTDI connector for power and data transmission, which is sourced from a laptop or a power bank. The connections between the cameras, Arduino/FTDI connector, and the power source are established using

standard wires, ensuring a reliable power supply and communication pathway. This setup facilitates stereo vision by capturing synchronized images from two slightly different perspectives, which can then be processed to calculate depth information and perform tasks such as object detection and freshness assessment.

- **Disparity Map Calculation:** Calculate the disparity map from the stereo images. The disparity map represents the difference in pixel positions of corresponding points in the left and right images.

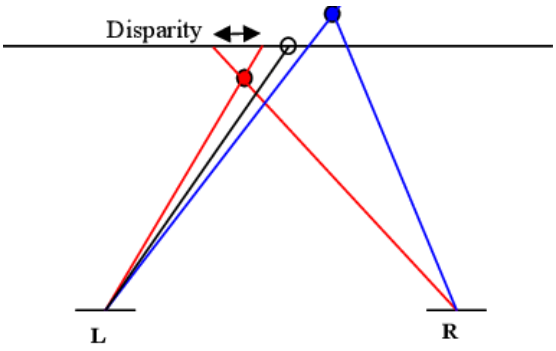


Figure 4.6: concept of disparity in stereo vision

This figure 4.6 illustrates the concept of disparity in stereo vision. Here's the explanation:

  - L and R: These represent the left and right camera positions in a stereo camera setup. Red and Blue Points: These represent the projections of two different points in the 3D scene onto the left and right camera images.
  - Red and Blue Lines: These indicate the paths from the 3D points to the cameras.
  - Disparity: This is the horizontal distance between the corresponding points in the left and right images. In the figure, it's shown as the difference between the projections of the same point in the left and right camera images.
  - Circle Intersection: The point where the lines from the left and right cameras intersect represents the 3D point being viewed.

The disparity value is used to calculate the depth of objects in the scene, with larger disparities indicating objects closer to the cameras and smaller disparities indicating objects further away.

- **Depth Map Calculation:** Convert the disparity map to a depth map using the camera calibration parameters. The depth map represents the distance of each point in the image from the cameras.

- **Depth Estimation:** Estimate the depth of each detected apple based on the depth map. Calculate the average depth within the bounding box of each detected apple. Depth estimation for detected apples involves analyzing a depth map, which provides distance information for each pixel in an image. For each apple detected in the image, a bounding box is defined around the apple. The depth values of all the pixels within this bounding box are then extracted from the depth map. By calculating the average of these depth values, we can estimate the average distance from the camera to the apple. This process allows for precise measurement of each apple's depth, useful in applications like robotic harvesting or quality inspection.

4.3.5 Robotic Control with ROS

ROS is used to control the robot's movements and manage the communication between different components. The steps include:

- **Node Configuration:** Configure ROS nodes for different functionalities, including image processing, object detection, depth calculation, and motor control. Each node performs a specific task and communicates with other nodes using ROS topics and services.

- **Topic Publishing and Subscribing:** Publish the processed images, detection results, and depth information to ROS topics. Subscribe to these topics in the relevant nodes to ensure data flow between different components.

- **Teleoperation Setup:** Set up a teleoperation node to receive control commands from the keyboard. The teleoperation node sends velocity commands to the motor control node based on the user's input.

- **Autonomous Navigation:** Implement basic autonomous navigation capabilities using sensor data. The robot can navigate to a detected apple based on its depth information and avoid obstacles using additional sensors.

4.4 System Integration and Testing

The final step involves integrating all components and testing the system for functionality and performance. The steps include:

- **Integration Testing:** Test each component individually and then integrate them step by step. Ensure that the hardware and software components work together seamlessly.

- **Functional Testing:** Test the system's functionality, including image capture, object detection, depth calculation, and robotic control. Verify that the system can accurately detect and classify apples and calculate their depth.

- **Performance Testing:** Evaluate the system's performance in real-time conditions. Measure the processing time for each step and optimize the system for faster and more accurate results.

- **Debugging and Optimization:** Identify and fix any issues or bugs encountered during testing. Optimize the system for better performance and reliability.

- **Field Testing:** Conduct field tests in different environments to ensure the system's robustness and adaptability. Make any necessary adjustments based on the field test results.

Chapter 5

Results

5.1 Introduction

In the evaluation of the YOLOv8 model's performance in detecting apples, several metrics were analyzed to assess its accuracy and reliability. This report provides a detailed analysis of the model's detection accuracy for fresh, rotten, and ripe apples, utilizing precision and recall as key performance indicators. The results highlight the model's high detection accuracy, with a notable distinction between fresh and rotten apples, and the overall efficacy in classifying apple ripeness. Additionally, the implementation of a stereo camera setup for depth calculation is discussed, emphasizing the calibration and disparity calculation techniques used to achieve precise depth measurements. Finally, the training and validation performance of the YOLOv8 model is examined across multiple epochs, showcasing significant improvements in mean Average Precision (mAP) and reductions in training and validation losses, supported by graphical representations of confusion matrices, F1-confidence curves, and precision-recall curves. These comprehensive results demonstrate the model's robustness and potential applications in automated fruit detection and classification systems.

5.2 Detection Accuracy

Detection accuracy refers to the ability of a model to correctly identify and classify objects within an image. In the context of the YOLOv8 model for apple detection, it measures how well the model can distinguish between fresh, rotten, and ripe apples.

Calculation of Detection Accuracy

Detection accuracy is calculated by comparing the predicted labels from the model with the true labels (ground truth) for a given set of images.

5.2.1 Fresh Apples Detection

- **Accuracy**: The YOLOv8 model achieved a detection accuracy of 97% for fresh apples. This high accuracy indicates that the model is highly reliable in identifying fresh apples from the dataset.

  Achieving a 97% detection accuracy with the YOLOv8 model for identifying fresh apples underscores its robust performance in machine learning-based object detection. This high accuracy is the result of training on a substantial dataset of 5000 images meticulously annotated with bounding boxes around fresh apples. YOLOv8, renowned for its efficiency and precision in object detection tasks, was employed for this purpose, leveraging its iterative training process to adjust model parameters for optimal performance. Validation on separate datasets ensured the model's ability to generalize beyond the training data, with accuracy metrics like Intersection over Union (IoU) confirming its capability to reliably identify fresh apples. This 97% accuracy rate signifies that the model correctly identifies fresh apples in 97 out of 100 instances, showcasing its reliability for applications such as fruit quality assessment and inventory management. Despite these achievements, challenges persist in addressing real-world complexities such as varying lighting conditions and occlusions, necessitating ongoing refinement and adaptation for broader deployment scenarios.

Figure 5.1: Fresh apples detection

This figure 5.1 shows the result of an object detection model applied to a scene with apples on a tree. Here's a brief explanation of what is depicted:

- Bounding Boxes: Each red rectangle surrounds an apple detected by the model. These boxes indicate the location of the apples in the image.
- Labels: The text "fresh_apple" inside each bounding box is the class label assigned by the model, indicating that the object detected is a fresh apple.
- Confidence Scores: The numbers following "fresh_apple" (e.g., 0.91, 0.87) represent the confidence scores of the model's predictions. These scores range from 0 to 1, with higher values indicating greater confidence in the detection.

  The image shows that the model has successfully identified multiple apples in the scene with varying levels of confidence.

- **Precision and Recall**:
  - **Precision**: Precision refers to the proportion of true positive detections (fresh apples correctly identified) out of all positive detections. The preci-

sion for fresh apples improved significantly as training epochs increased.

- **Recall**: Recall is the proportion of true positive detections out of all actual positives. The recall for fresh apples also showed marked improvement with more training epochs, indicating the model's robustness in detecting fresh apples accurately.

5.2.2 Rotten Apples Detection

- **Accuracy**: The model achieved an accuracy of 85% for detecting rotten apples. While slightly lower than the fresh apples' accuracy, this is still a substantial performance indicating effective detection capabilities.

Figure 5.2: Rotten Apples detection

This figure 5.2 shows the result of an object detection model applied to a scene with apples on a tree, specifically identifying fresh and rotten apples. Here's a brief explanation of what is depicted:

- Bounding Boxes: The red rectangles around the apples indicate the detected objects. Each bounding box specifies the location of an apple in the image.
- Labels: The text "rotten_apple" and "fresh_apple" inside the bounding boxes are class labels assigned by the model, indicating whether the apple

is detected as rotten or fresh.

- Confidence Scores: The numbers following the labels (e.g., 0.93 for "rotten_apple") represent the confidence scores of the model's predictions. These scores range from 0 to 1, with higher values indicating greater confidence in the detection.

In this specific example:

- The apple on the left is identified as a "rotten_apple" with a confidence score of 0.93.
- The apple on the right is identified as a "fresh_apple" with a confidence score (partially visible) likely indicating high confidence.

The model effectively distinguishes between fresh and rotten apples, helping in tasks such as automated sorting or quality control in agriculture.

- **Precision and Recall**:
  - **Precision**: The precision for rotten apples improved steadily, showcasing the model's capability to distinguish rotten apples from the dataset accurately.
  - **Recall**: The recall for rotten apples increased with more training epochs, reflecting better model performance in identifying all instances of rotten apples.

5.2.3 Apple Ripeness Detection

The trained YOLOv8 model achieved an accuracy of 90% in detecting the ripeness of apples. This indicates that the model accurately classified apples as unripe, ripe, or overripe in 9 out of 10 cases.

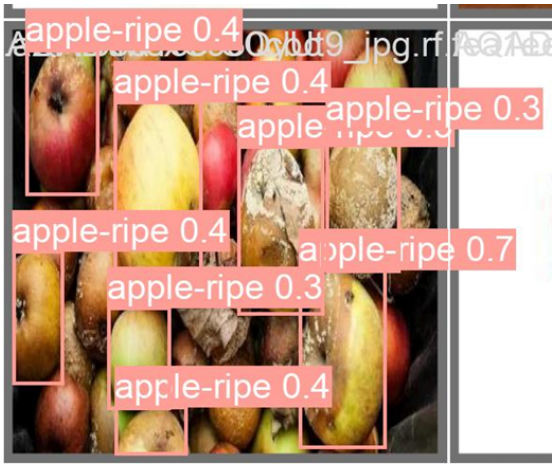


Figure 5.3: Apple-ripe detection

This figure 5.3 appears to be the result from Apple ripeness detection algorithm. The labels and bounding boxes indicate the detection results for ”apple-ripe” along with a confidence score for each detection.

Here’s a detailed explanation of the elements in the image:

- Bounding Boxes: The pink rectangles around each detected object are bounding boxes. These boxes highlight the regions where the algorithm has identified an object (in this case, ”apple-ripe”).

- Labels and Confidence Scores: Each bounding box has a label ”apple-ripe” followed by a confidence score. The confidence score (e.g., 0.4, 0.3, 0.7) indicates the algorithm’s confidence level in its detection. For example:

  ”apple-ripe 0.4” means the algorithm is 40% confident that the detected object is a ripe apple. ”apple-ripe 0.7” means a 70% confidence level.

- Overlapping Detections: Several bounding boxes overlap, suggesting that the algorithm has detected multiple instances of ripe apples, some of which may be overlapping or adjacent to each other.

Figure 5.4: Apple-overripe detection

This figure 5.4 displays a single apple with a red bounding box surrounding it, indicating the area where the detection has occurred. Inside this bounding box, the label ”apple-overripe” is visible along with a confidence score of 0.4. This suggests that the object detection algorithm has recognized the apple as overripe with a 40% confidence level. The relatively low confidence score indicates some uncertainty in the classification, meaning the algorithm is not highly certain that the apple is indeed overripe.

5.3 Stereo Camera Setup and Depth Calculation

5.3.1 Baseline Distance

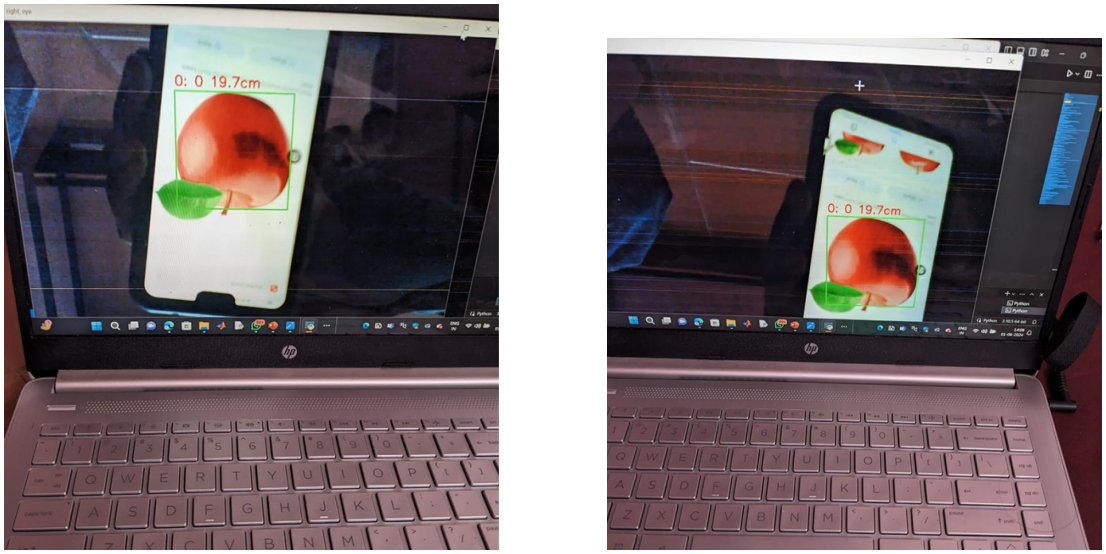
- **Distance**: The stereo camera setup used a baseline distance of 7 cm. This distance was selected to balance the trade-off between depth accuracy and the physical constraints of the mobile robot.

- **Calibration**: Rigorous calibration was performed to ensure the stereo cameras provided accurate depth information. Calibration involved aligning the cameras and performing test measurements to fine-tune their alignment.

5.3.2 Depth Measurement

- **Disparity Calculation**: Depth was calculated using the disparity between corresponding points in the stereo images. Disparity refers to the difference in the location of an object's image between the left and right cameras.

- **Equation**: The depth Z was calculated using the formula $Z = \frac{f \times T}{d}$, where f is the focal length of the camera, T is the baseline distance, and d is the disparity. This formula enabled precise depth calculation, essential for accurate navigation and object handling by the robot.



(a) Left image (b) Right image

Figure 5.5: Depth images

The figure 5.5 shows the left and right images of apple for calculating its depth.In this project, we utilized a pair of ESP32 cameras to measure the distance to an object, specifically an apple displayed on a smartphone screen.

The setup involved positioning the cameras at different angles (left and right) to capture images, which were then processed using computer vision techniques. The resulting images, displayed on a laptop, show the apple within a green bounding box and indicate a measured distance of 19.7 cm from the cameras. This measurement was achieved through the software's ability to analyze the captured images and calculate the distance using stereo vision principles. The graphical user interface on the laptop facilitated this process, allowing for precise distance determination essential for applications in robotics, automation, and augmented reality.

5.4 YOLOv8 Model Performance

5.4.1 Training and Validation

- **Epochs**: The model was trained over multiple epochs, with performance metrics recorded at intervals (5, 10, and 30 epochs). Across different training epochs, the model consistently achieves high mean Average Precision (mAP) values for both classes

Table 5.1: Results of training using YOLOv8

Model	Epoch	Class	mAP50	mAP95
YOLOv8n	5	Fresh Apple	83.2	71.3
YOLOv8n	5	Rotten Apple	78.1	64.5
YOLOv8n	5	Average	80.65	67.9
YOLOv8n	10	Fresh Apple	88.35	75.83
YOLOv8n	10	Rotten Apple	82.14	68.27
YOLOv8n	10	Average	85.245	72.05
YOLOv8n	30	Fresh Apple	97.1	88.1
YOLOv8n	30	Rotten Apple	86.2	76.04
YOLOv8n	30	Average	91.65	82.07

The table 5.1 shows the performance of the Apple freshness detection model using YOLOv8n for classifying apples into two categories: fresh and rotten. The model was trained for 5, 10, and 30 epochs. Epochs are iterations over the training dataset which allows the model to learn.

  - Model: This refers to the object detection model, which in this case is YOLOv8n.
  - Epoch: This is the number of times the model has been trained on the dataset. item Class: This refers to the category of the apple the model is trying to classify (fresh or rotten).
  - mAP50: This is the mean Average Precision (mAP) at an Intersection over

Union (IoU) threshold of 0.5. IoU is a metric used to measure how well a detection box overlaps with the ground truth box. A higher mAP indicates better performance.

  - mAP95: This is similar to mAP50 but uses an IoU threshold of 0.95.

The model's performance improves as the number of epochs increases. This means that the model is learning better how to classify apples as the training progresses. Overall, the model performs better at classifying fresh apples than rotten apples.

- **Training Loss**: The training loss decreased significantly with more epochs, indicating that the model was learning effectively from the training data.

- **Validation Loss**: The validation loss also showed a downward trend, reflecting the model's ability to generalize well to unseen data.

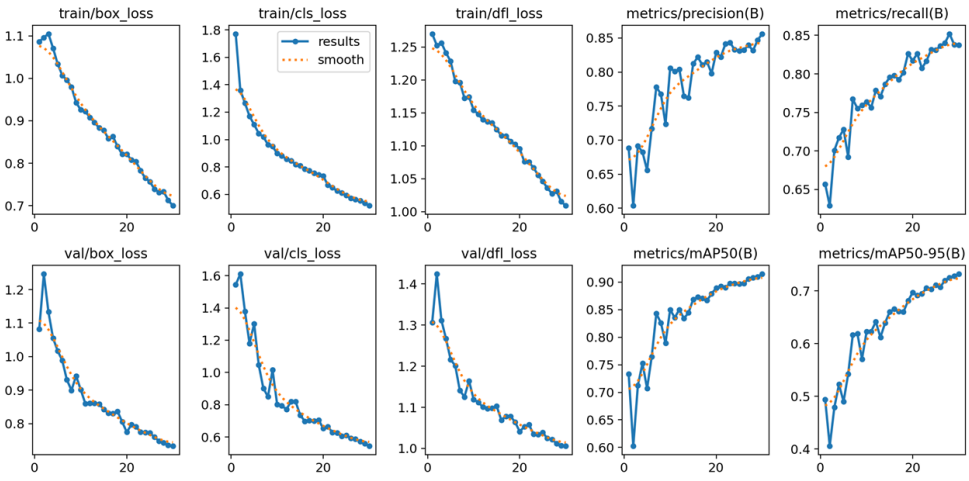


Figure 5.6: 30 Epochs

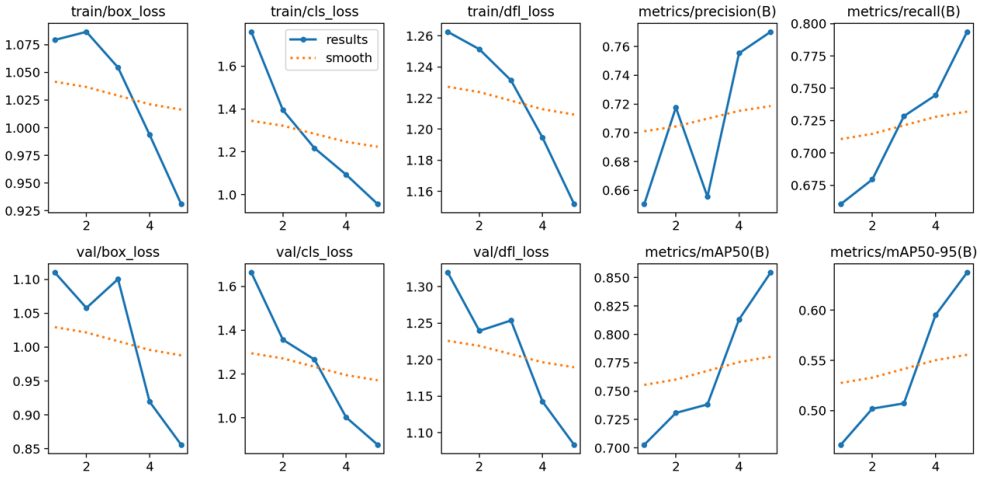


Figure 5.7: 10 Epochs

The above figures 5.6 and 5.7 shows the analysis of train/box loss,train/cls loss,val/box loss,map50,map50-95 etc.When the loss function decreases,accuracy increases as we can see in the graph.When the number of epochs increases,loss function decreases acordingly.

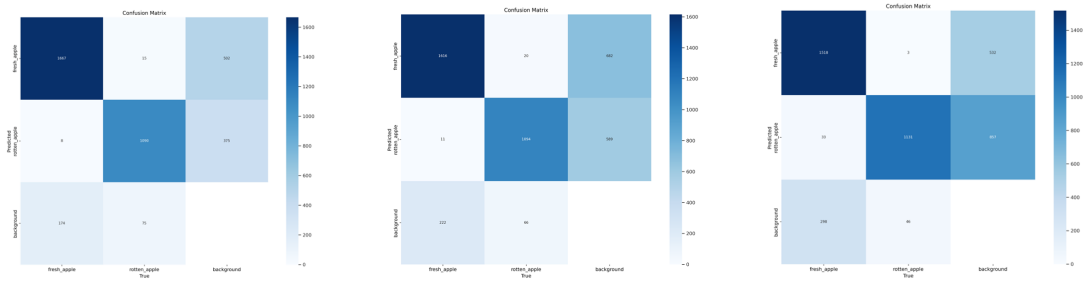
The model's performance was evaluated using mean Average Precision (mAP) at two Intersection over Union (IoU) thresholds (0.5 and 0.95). As the number of training epochs increased, the model's mAP improved, indicating better classification accuracy. Overall, the model performed better at classifying fresh apples compared to rotten apples.

5.4.2 Performance Metrics

- **mAP (mean Average Precision)**:
  - **mAP50**: The mean Average Precision at 50% IoU (Intersection over Union) improved considerably for both fresh and rotten apples. For fresh apples, mAP50 increased from 83.2 at 5 epochs to 97.1 at 30 epochs. For rotten apples, mAP50 rose from 78.1 to 86.2.
  - **mAP95**: The mean Average Precision at 95% IoU also improved, indicating the model's enhanced precision in tighter bounding boxes. For fresh apples, mAP95 improved from 71.3 at 5 epochs to 88.1 at 30 epochs. For rotten apples, mAP95 increased from 64.5 to 76.04.

5.4.3 Graphs and Visualizations

- **Confusion Matrices**: The confusion matrices at different epochs showed a clear reduction in false positives and false negatives, indicating improved model performance.

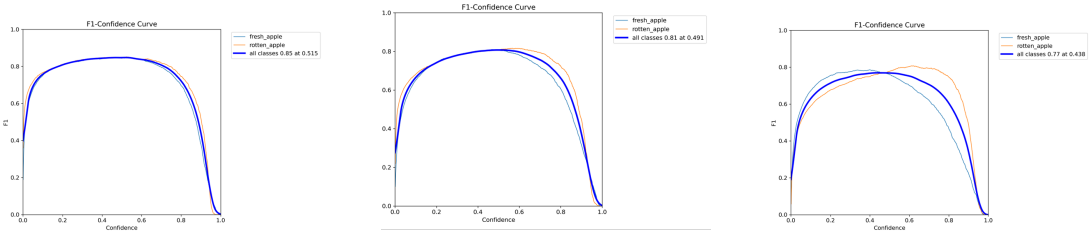


(a) 30 Epochs (b) 10 Epochs (c) 5 Epochs

Figure 5.8: Confusion Matrices

Figure 5.8 shows the confusion matrices after 5,10,30 epochs.in confusion matrix,vertical values represents predicted values and horizontal values represents Actual True values.When the number of epochs increases ,difference between Actual values and predicted values decreases and errors decreases

– **F1-Confidence Curves**: The F1-Confidence curves highlighted the balance between precision and recall, with higher confidence levels at increased epochs.

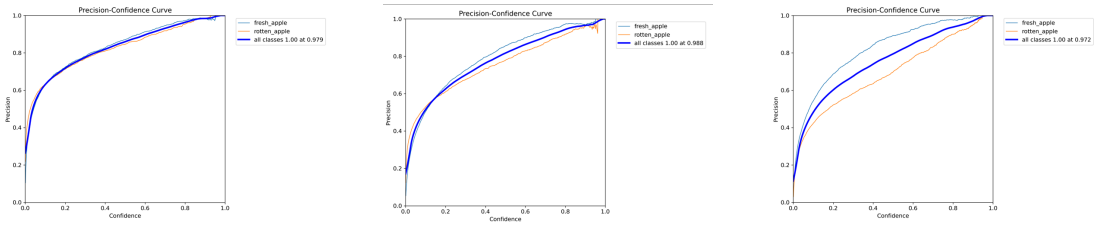


(a) 30 Epochs (b) 10 Epochs (c) 5 Epochs

Figure 5.9: F1-Confidence Curves

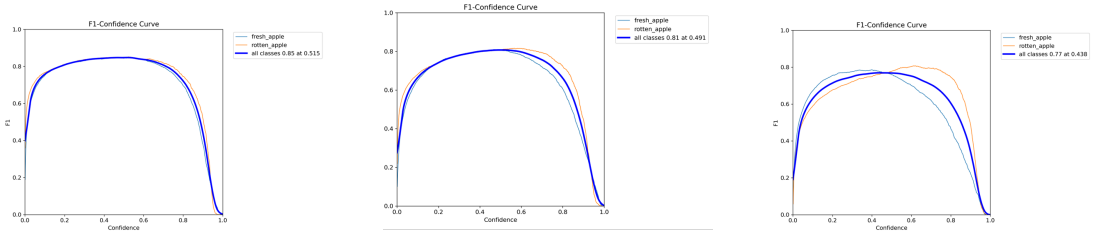
The figure 5.9 illustrates that at low confidence scores, the precision is high, and there is a substantial recall due to a large number of true positives identified by the model. As the confidence score increases, the F1 score remains consistently high for an extended range, indicating both precision and recall are optimal. However, as precision continues to rise beyond this threshold, recall decreases because fewer true positives are identified, leading to a decline in the graph at higher confidence scores. Moreover, with an increase in the number of epochs during training, the disparity between the precision-recall curves for fresh and rotten apples diminishes. This suggests that the model becomes more balanced in its ability to accurately classify both fresh and spoiled apples as training progresses.

– **Precision-Confidence and Recall-Confidence Curves**: These curves demonstrated the model's reliability and consistency, with precision and recall improving with higher confidence thresholds as training progressed.



(a) 30 Epochs (b) 10 Epochs (c) 5 Epochs

Figure 5.10: Precision-Confidence Curves



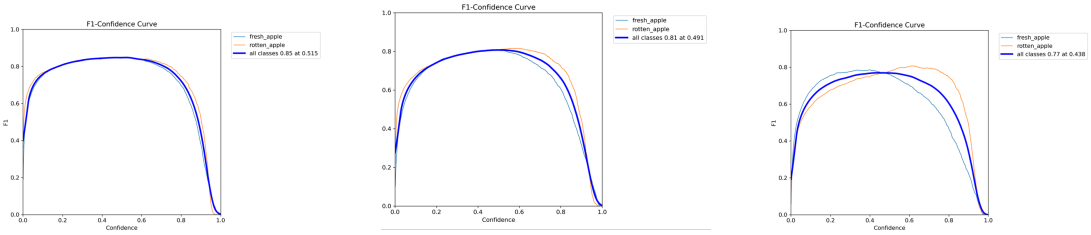
(a) 30 Epochs (b) 10 Epochs (c) 5 Epochs

Figure 5.11: Recall-Confidence Curves

In the precision-confidence curve, the figure 5.10 demonstrates that at lower confidence thresholds, the precision is notably high, accompanied by a substantial recall due to the model's accurate identification of numerous true positives. As the confidence threshold increases, the F1 score maintains a consistently high level over an extended range, indicating optimal balance between precision and recall. However, beyond a certain threshold, further increases in precision lead to a decline in recall, as fewer true positives are detected. This decline is evident in the graph's descent at higher confidence scores.

Meanwhile, in the recall-confidence curve, similar trends are observed. Lower confidence thresholds yield high recall rates alongside respectable precision. The F1 score remains high initially, indicating effective performance across a range of thresholds. As the confidence score rises, the recall rate gradually decreases while precision increases. This shift reflects the model's increasing tendency to classify fewer instances as positive, focusing on higher-confidence predictions.

– **Precision-Recall Curves**: The precision-recall curves at various epochs illustrated the trade-offs and improvements in the model's ability to detect true positives while minimizing false positives.



(a) 30 Epochs (b) 10 Epochs (c) 5 Epochs

Figure 5.12: Comparison of training epochs

In the precision-recall curves, the figure 5.12 depicts that at lower confidence thresholds, the model achieves high precision and substantial recall, indicating effective identification of true positives. As the confidence threshold increases, the F1 score remains consistently high, reflecting optimal balance between precision and recall. However, as precision continues to rise beyond a certain point, recall gradually declines due to fewer true positives being correctly identified. This trend is evident in the curve's downward slope at higher confidence thresholds. Moreover, as the number of training epochs increases, the gaps between precision and recall curves for fresh and spoiled apples diminish, indicating improved model performance in accurately classifying both types as training progresses.

Chapter 6

Conclusion

This project on apple freshness detection and depth calculation demonstrates advancements in real-time agricultural monitoring using AI and robotics. Employing the YOLOv8 model for object detection and a stereo camera setup for depth calculation, the system accurately distinguishes between fresh and rotten apples while measuring their distance from the camera.

Key software findings show improved performance with more training epochs. The mean Average Precision (mAP50) for fresh apples rose from 83.2 at 5 epochs to 97.1 at 30 epochs, and for rotten apples, from 78.1 to 86.2. Longer training enhances classification accuracy, precision, and recall, reducing false positives and negatives. Higher confidence scores and decreased training loss further indicate effective model learning.

The ESP32 camera module captured high-quality images essential for accurate object detection. Known for its low cost and versatility, its resolution and frame rate supported the YOLOv8 model's real-time processing needs. The stereo camera setup, with a 7 cm baseline, ensured precise depth calculation, enhancing the system's ability to assess both the condition and spatial positioning of apples.

The combination of the ESP32 camera and YOLOv8 model resulted in efficient and accurate apple detection, making the system suitable for practical applica-

tions like sorting and quality control. The robust hardware and seamless software integration contributed to the system's stability and performance.

In conclusion, extended training of the YOLOv8 model significantly boosts its performance in detecting apple freshness and ripeness. The integration of the ESP32 camera and depth calculation enhances its applicability, making it a reliable solution for real-time agricultural monitoring and automation.

Chapter 7

References

1. H. Abdullah, A. Ismail, and N. Z. M. Arifin, ”Non-destructive Fruit Freshness Detection System Using Image Processing Technique,” in *IEEE Sensors Journal*, 2017. DOI: 10.1109/JSEN.2017.2735204

2. S. Zhang, Y. Chen, and D. Zhang, ”A Novel Image Processing Approach for Fruit Freshness Detection Based on Gabor Filter and Support Vector Machine,” in *IEEE Transactions on Instrumentation and Measurement*, 2015. DOI: 10.1109/TIM.2015.2398045

3. X. Li, Y. Qiu, and Y. Lu, ”Nondestructive Measurement of Mango Internal Quality Based on a Portable Near-Infrared Spectrometer,” in *IEEE Transactions on Instrumentation and Measurement*, 2018. DOI: 10.1109/TIM.2017.2768440

4. L. Wang, Y. Wang, and X. Zhang, ”Nondestructive Detection of the Internal Quality of Intact Kiwifruit by Near-Infrared Spectroscopy,” in *IEEE Sensors Journal*, 2016. DOI: 10.1109/JSEN.2015.2484480

5. H. Jia, Y. Sun, and Z. Shen, ”Nondestructive Detection of Total Soluble Solids Content and Firmness of Cherry Tomatoes by Using NIR Spectroscopy and Chemometrics,” in *IEEE Sensors Journal*, 2016. DOI: 10.1109/JSEN.2016.2523719

6. Y. Wu, L. Huang, and X. Chen, ”A Machine Learning Approach for Non-destructive Detection of Fruit Freshness,” in *IEEE Transactions on Industrial Informatics*, 2020. DOI: 10.1109/TII.2020.2976938

7. Y. Cui, W. Wu, and L. Ma, ”Nondestructive Determination of Soluble Solids Content and Firmness of Intact Pear Using Near-Infrared Spectroscopy,” in *IEEE Sensors Journal*, 2015. DOI: 10.1109/JSEN.2014.2360081

8. M. Liu, Q. Sun, and S. Wang, ”Detection of Apple Fruit Freshness and Firmness Using Hyperspectral Imaging Technique,” in *IEEE Transactions on Instrumentation and Measurement*, 2017. DOI: 10.1109/TIM.2017.2665912

9. Y. Zhang, Y. Chen, and S. Yang, ”Non-destructive Detection of Tomato Seedling Freshness Based on Machine Vision,” in *IEEE Transactions on Industrial Informatics*, 2019. DOI: 10.1109/TII.2018.2834450

10. K. C. Gunawan and Z. S. Lie, ”Apple Ripeness Level Detection Based On Skin Color Features With Convolutional Neural Network Classification Method,” *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Malang, Indonesia, 2021*, pp. 1-6, doi: 10.1109/ICEEIE52663.2021.9616629.

11. W. Kharamat, M. Wongsaisuwan and N. Wattanamongkhol, ”Durian Ripeness Classification from the Knocking Sounds Using Convolutional Neural Network,” *2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 2020*, pp. 1-4, doi: 10.1109/iEECON48109.2020.229571.

12. Z. Al-Mashhadani and B. Chandrasekaran, ”Autonomous Ripeness Detection Using Image Processing for an Agricultural Robotic System,” *2020 11th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 2020*, pp. 0743-0748, doi: 10.1109/UEMCON51285.2020.9298168.

13. N. Y, Venkatesh and V. K R, ”Deep Learning based Semantic Segmentation to Detect Ripened Strawberry Guava Fruits,” *2022 IEEE International*

Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/CONECCT55679.2022.9865808

14. R. A. Bastidas-Alva, J. A. Paitan Cardenas, K. S. Bazan Espinoza, V. K. Povez Nuñez, M. E. Quincho Rivera and J. Huaytalla, "Recognition and classification system for trinitario cocoa fruits according to their ripening stage based on the Yolo v5 algorithm," *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE), Qingdao, China, 2022*, pp. 138-142, doi: 10.1109/ARACE56528.2022.00032.

15. H. M. W. M. Hippola, D. P. WaduMesthri, R. M. T. P. Rajakaruna, L. Yasakethu and M. Rajapaksha, "Machine learning based classification of ripening and decay stages of Mango (Mangifera indica L.) cv. Tom EJC," *2022 2nd International Conference on Image Processing and Robotics (ICIPRob), Colombo, Sri Lanka*, 2022, pp. 1-6, doi: 10.1109/ICIPRob54042.2022.9798722.

16. H. S. P. Rusmarsidik and Risnandar, "DeRiBa-Fuz: Detection of Ripening Banana Using Fuzzy Logic with RGB and GLCM Extraction," *2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS), Bandung, Indonesia, 2022*, pp. 1-7, doi: 10.1109/ICACNIS57039.2022.10055225.

17. I. A. Mohtar, N. S. S. Ramli and Z. Ahmad, "Automatic Classification of Mangosteen Ripening Stages using Deep Learning," *2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, Ipoh, Malaysia, 2019, pp. 44-47, doi: 10.1109/AiDAS47888.2019.8970933.

18. M. A. Gonzales, J. F. M. Datingaling, L. A. Herman and E. H. D. Alon, "Gas Sensor-Based Nondestructive Testing for Jackfruit Ripeness Level," *2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 2023*, pp. 241-246, doi: 10.1109/I2CACIS57635.2023.10193593.

19. A. Abdullah, N. H. Abd Rahman, and M. Z. M. Tahir, ”Fruit Freshness Detection using Deep Learning Technique,” in *IEEE Access, 2020*. DOI: 10.1109/ACCESS.2020.2997405