TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

KATHMANDU ENGINEERING COLLEGE
KALIMATI, KATHMANDU

A
Major Project Proposal
Defense Report
On

"Advancing Tomato Harvesting: An Automation with Machine

Learning Approach"

Submitted By:

Mausam Gurung (KAT076BEI014)

Shalon Maharjan (KAT 076BEI027)

Shubham Bista(KAT076BEI028)

Symon Shrestha (KAT076BEI030)

Submitted To:

**DEPARTMENT OF ELECTRONICS, COMMUNICATION
AND INFORMATION ENGINEERING**

August-2023

ACKNOWLEDGEMENT

ABSTRACT

The agricultural industry continually seeks innovative solutions to enhance productivity, address labor shortages, and optimize crop harvesting processes. In this context, the development of an autonomous tomato harvesting robot equipped with machine learning capabilities presents a promising approach. Our project aims to design and implement an autonomous tomato harvesting robot with a 6DOF robotic arm and a machine learning-based recognition system. The robot's 6DOF robotic hand enables precise and gentle fruit detachment from tomato plants, minimizing crop damage and wastage. The machine learning-based recognition system, employing Histograms of Oriented Gradients (HOG) features and a Support Vector Machine (SVM) classifier, accurately identifies and locates ripe tomatoes in a tomato field.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATION

HOG: Histogram of Oriented Gradient
Li-ion: Lithium-Ion battery
SVM: Support Vector Machine
CLAHE: Contrast Limited Adaptive Histogram Equalization
NB: Naive Bayes
FCR: False Color Removal
NMS: Non-Maximum Suppression
ROC: Receiver Operating Characteristic
AUC: Area Under the Curve

CHAPTER 1: INTRODUCTION

1.1 Background Theory

The agricultural industry is continually seeking innovative solutions to address labor shortages, enhance productivity, and improve harvesting efficiency. In recent years, autonomous harvesting robots have emerged as a promising technology to revolutionize the way crops are harvested. This report presents the design and implementation of an autonomous tomato harvesting robot equipped with a 6 Degrees of Freedom (6DOF) robotic hand and a machine learning-based recognition system. The primary objectives of this research are to optimize the Harvesting Mechanism, develop an efficient Recognition System for ripe tomato identification, and design robust and precise movement capabilities for the robot. There are three function we need to consider for the design of robot. They are:

1.1.1 Harvesting Mechanism:

The Harvesting Mechanism is a crucial component of the autonomous tomato harvesting robot. The primary goal is to ensure a gentle yet efficient method of detaching ripe tomatoes from the plant to minimize damage and preserve the crop's quality. This advanced robotic hand offers greater dexterity and versatility, allowing the robot to handle delicate tomatoes with precision and efficiency.

1.1.2 Recognition System:

Accurate recognition of ripe tomatoes is essential for the efficient operation of the harvesting robot. To achieve this, we implement a machine learning-based Recognition System using advanced computer vision techniques. The system employs Histograms of Oriented Gradients (HOG) features and a Support Vector Machine (SVM) classifier to identify and locate ripe fruits accurately. By training the machine learning model on a vast dataset of tomato images, the robot can intelligently differentiate between ripe and unripe tomatoes, enhancing its harvesting capabilities.

1.1.3 Movement System:

The movement of a harvesting robot on wheels involves the integration of sophisticated control systems and motion planning algorithms. These elements allow the robot to traverse uneven terrains within the agricultural fields, seamlessly maneuvering around crop rows and obstacles. The agility and speed of the robot's movement are crucial in optimizing harvesting efficiency and reducing time-consuming manual labor.

Additionally, the design and implementation of wheel movement mechanisms must ensure minimal soil compaction to preserve the soil's health and fertility. The capability to move smoothly and with precision is paramount to minimize crop damage during the harvesting process.

1.2 Problem Statement

The traditional manual harvesting of tomatoes in agricultural practices is labor-intensive and time-consuming, leading to increased production costs and inefficiencies in crop harvesting. The lack of an automated and efficient harvesting solution poses a significant challenge to the agricultural industry. Moreover, accurately identifying ripe tomatoes amidst foliage can be challenging, resulting in potential crop damage and wastage. To address these issues, there is a pressing need for the design and implementation of an autonomous tomato harvesting robot with advanced recognition capabilities and precise harvesting mechanisms to improve productivity, reduce labor dependence, and ensure optimal crop quality.

1.3 Objectives

- Develop an autonomous tomato harvesting robot with a 6DOF robotic hand and optimized movement capabilities for precise and gentle fruit detachment from the plant, efficient navigation through tomato plants, and avoidance of obstacles.

- To Implement a machine learning-based Recognition System using HOG features and SVM classifier to accurately identify and locate ripe tomatoes.

1.4 Scope and Applications

- Revolutionizing agricultural practices by enhancing harvesting efficiency and reducing manual labor dependency.
- Improving crop productivity and quality by minimizing crop damage and wastage during harvesting.
- Adapting the autonomous harvesting robot for various greenhouse and field cultivation setups.
- Enhancing sustainability in agriculture through the efficient use of resources and reduced environmental impact.
- Potential application in other fruit and vegetable harvesting tasks to address labor shortages and improve production efficiency in the agricultural industry.

1.5 Organization of Project Report

The material presented in this report is organized into five chapters: Chapter 1 consists of the introduction, objective, and background of the project. The scope and application of the project are also discussed. Chapter 2 deals with the literature review that describes the past works and research that were done related to this project and the methodology that was used in those projects. Chapter 3 discusses the conceptual theories about various related aspects, and components used. Chapter 4 describes the methodology, basic design, outline, and process of the project, and Chapter 5 consists of the expected outputs/results. Finally, Chapter 6 consists of the epilogue of the project.

CHAPTER 2: LITERATURE REVIEW

In recent years, the agricultural sector has faced the growing challenge of labor shortages and the need to enhance productivity. Autonomous harvesting robots have emerged as a promising solution to address these issues, revolutionizing traditional crop harvesting methods. This literature review provides insights into the design and development of autonomous tomato harvesting robots, focusing on their key components: harvesting mechanisms, recognition systems, and precise movement capabilities. These advancements hold the potential to not only optimize tomato harvesting efficiency but also minimize crop damage, ultimately contributing to more sustainable and productive agricultural practices.

On this paper author focuses on the design and development of an autonomous tomato harvesting robot, which aims to address labor shortages and boost crop productivity in agriculture. The robot's key elements include harvesting, recognition, and movement. It utilizes a plucking mechanism with a rotational gripper for efficient tomato harvesting and integrates a color camera for fruit recognition. The review highlights the importance of improving recognition methods to increase the harvesting rate and discusses challenges, such as grasp state estimation and simultaneous recognition of fruit and stem positions. The robot's evaluation through extensive harvesting experiments in diverse environments is also explored. [1]

This research article introduces a method for detecting mature tomatoes using computer vision techniques, with a particular focus on applying machine learning and color analysis to achieve precise tomato detection. The authors utilize Histograms of Oriented Gradients (HOG) features and a Support Vector Machine (SVM) classifier for the detection process. The article provides an in-depth description of the HOG feature extraction process, SVM classifier training, and the adoption of a sliding window approach to identify tomatoes. [2]

In this document the author focuses on design and testing of a tomato harvesting robot equipped with a vision positioning system and a picking gripper for efficient tomato harvesting. The robot's control system comprises an information acquisition unit, IPC operation control unit, and motion execution unit. Tested in a greenhouse, the robot achieved a successful tomato harvesting rate of 83.9%. Its primary goal is to enhance efficiency and reduce labor in tomato harvesting processes. [3]

This document focuses on the design and research of the end actuator of a tomato picking robot The end effector, comprising hardware components such as fingers, vacuum corrugated suckers, bidirectional screws, DC servo motors, and more, plays a vital role in achieving successful fruit and vegetable picking. [4]

The paper focuses on the design of a chassis system for a tomato picking robot in a greenhouse. It provides detailed information on the mechanical structure design, girder assembly, and drive line, as well as the overall layout of the chassis. Additionally, it discusses the components' functions, such as the mounting frame for the picking system and the control system installation area. The paper also covers the design of the power supply circuit and control program. [5]

CHAPTER 3: RELATED THEORY

3.1 Hardware

3.1.1 Raspberry Pi 4 Model B with 4GB RAM:

The Raspberry Pi 4 Model B with 4GB RAM is a single-board computer that offers improved processing power and memory compared to its predecessors. It is part of the Raspberry Pi series, which has gained popularity as an affordable and versatile platform for various projects, including home automation, robotics, IoT (Internet of Things) applications, and even lightweight machine learning tasks.

The key features of it are as follows:

1. **Processor:** The Raspberry Pi 4 is powered by a Broadcom BCM2711 quad-core ARM Cortex-A72 processor running at 1.5 GHz. This provides a significant performance boost compared to earlier models.
2. **RAM:** This version of the Raspberry Pi 4 comes with 4GB of LPDDR4 SDRAM. The increased RAM capacity allows for smoother multitasking and better performance in memory-intensive applications.
3. **Video Output:** The board supports dual micro-HDMI ports with 4K video output at 60 frames per second. This makes it suitable for multimedia and display-intensive projects.
4. **USB Ports:** The Raspberry Pi 4 has two USB 3.0 ports and two USB 2.0 ports, allowing for high-speed data transfer and connecting various peripherals.
5. **Gigabit Ethernet**: The board includes a Gigabit Ethernet port for fast wired network connectivity.
6. **Wireless Connectivity:** It has built-in 802.11ac Wi-Fi and Bluetooth 5.0, enabling wireless communication and connectivity with other devices.
7. **GPIO Pins:** Like other Raspberry Pi models, the Raspberry Pi 4 also has GPIO (General Purpose Input/Output) pins for interfacing with external components and devices.
8. **Power Supply:** The board can be powered using a USB Type-C power supply.

Figure 1: Raspberry Pi 4 Model B with 4GB

3.1.2 Servo Motor – MG966R Series:

The MG966R is a popular series of servo motors commonly known for its high torque and reliability, making it suitable for a wide range of applications, including robotics, RC (remote control) vehicles, drones, and automation projects.

The key features of the MG966R series servo motor:

1. **High Torque:** The MG966R series servo motor provides high torque, making it capable of handling tasks that require significant force and precision.

2. **Metal Gear Train:** It is equipped with a metal gear train, which enhances the durability and strength of the servo motor, making it suitable for heavy-duty applications.

3. **Operating Voltage:** The typical operating voltage for the MG966R series is 4.8V to 6V. It is essential to supply the servo motor with a voltage within this range to ensure proper operation.

4. **Control Interface:** The servo motor can be controlled using PWM (Pulse Width Modulation) signals. Most microcontrollers, including Raspberry Pi and Arduino boards, can generate PWM signals to control the position and movement of the servo motor.

5. **Rotation Range:** The typical rotation range of the MG966R series is approximately 180 degrees, allowing it to move within a wide angular range.

6. **Operating Speed:** The operating speed of the servo motor is specified in seconds per 60 degrees of rotation. This parameter indicates how fast the servo can move to a specific position.

7. **Dimensions:** The MG966R series servo motor comes in various sizes and dimensions, so it's important to choose the appropriate size for your specific project requirements.

Figure 2: Servo motor MG966R Series

3.1.3 6-Channel 12-bit PWM/Servo Driver - I2C interface - PCA9685:

The 6-Channel 12-bit PWM/Servo Driver with an I2C interface based on the PCA9685 is a popular integrated circuit (IC) commonly used to control multiple servos or LEDs with precise PWM (Pulse Width Modulation) signals. It is often used in robotics, automation, RC vehicles, lighting control, and other projects that require precise and synchronized control of multiple servos or LEDs.

The key features of the PCA9685 PWM/Servo Driver:

1. **PWM Channels:** The PCA9685 offers 6 independent PWM channels, allowing you to control up to 6 servos or LEDs individually.

2. **12-bit Resolution:** It provides a high-resolution 12-bit PWM output, allowing for fine-grained control of the servo's position or the brightness of the LEDs.

3. **I2C Interface:** The PCA9685 communicates with the host microcontroller or single-board computer (such as Arduino or Raspberry Pi) using the I2C (Inter-Integrated Circuit) interface. I2C is a popular serial communication protocol that allows multiple devices to share the same bus.

4. **External Power Supply:** The PCA9685 can operate using an external power supply to provide sufficient power to the servos or LEDs, independent of the voltage supplied to the control board.

5. **Prescaler:** The PCA9685 has a prescaler feature that allows you to adjust the PWM frequency. This feature is useful for matching the PWM frequency with the requirements of the servos or other controlled devices.

6. **Hardware Reset:** It includes a hardware reset pin to reset the IC if necessary.

7. **Address Select Pins:** The PCA9685 has address select pins that allow you to change the I2C address of the device, making it possible to use multiple PCA9685 boards on the same I2C bus.
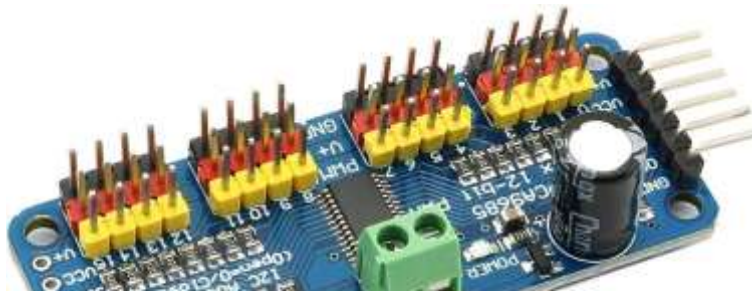
Figure 3: 6 -Channel 12-bit PWM/Servo Driver - I2C interface - PCA968

3.1.4 Buck Converter:

A Buck Converter, also known as a Step-Down Converter, is a type of DC-DC converter used to step down or reduce the voltage level from a higher input voltage to a lower output voltage. It is widely used in electronic circuits and power supply designs to efficiently convert power from a higher voltage source to a lower voltage level required by various electronic components.

The key features and characteristics of a Buck Converter:

1. **Voltage Conversion:** The primary function of a Buck Converter is to convert a higher input voltage to a lower output voltage. It is designed to deliver a stable output voltage, regardless of fluctuations in the input voltage.

2. **Switching Regulator:** A Buck Converter operates as a switching regulator, using semiconductor switches (usually MOSFETs) to control the flow of current and efficiently regulate the output voltage.

3. **Efficiency:** Buck Converters are known for their high efficiency, which is achieved through the switching action that minimizes power loss compared to linear regulators.

4. **Inductor:** The Buck Converter uses an inductor to store energy during the switch-on period and release it to the load during the switch-off period. The inductor helps smooth the output voltage and current.

5. **Control Circuitry:** A control circuit is used to monitor the output voltage and adjust the duty cycle (on/off time) of the switching components to maintain the desired output voltage level.

6. **Duty Cycle:** The duty cycle of a Buck Converter represents the ratio of time the switch is ON to the total switching period. It determines the output voltage level.

7. **Continuous and Discontinuous Mode:** Buck Converters can operate in continuous conduction mode (CCM) or discontinuous conduction mode (DCM) depending on the current flow through the inductor.

Figure 4: Buck Converter

3.1.5 Ultrasonic sensor:

An ultrasonic sensor is a type of sensor that uses ultrasonic waves to measure distances, detect objects, or perform other proximity-related tasks. It is commonly used in various applications, including robotics, automation, distance measurement, and object detection.

Here's how an ultrasonic sensor typically works:

1. **Ultrasonic Waves:** The sensor emits ultrasonic waves, which are sound waves with frequencies higher than the human audible range (usually above 20 kHz). The most common frequency used in ultrasonic sensors is around 40 kHz.

2. **Echo Reception:** After emitting the ultrasonic waves, the sensor waits for the waves to bounce off an object or surface in its vicinity. The time taken for the waves to travel to the object and back is recorded.

3. **Time-of-Flight Measurement:** By measuring the time it takes for the ultrasonic waves to return, the sensor can calculate the distance to the object based on the speed of sound in the medium (typically air). The distance can be determined using the formula: Distance = (Speed of Sound × Time-of-Flight) / 2.

4. **Analog or Digital Output:** Ultrasonic sensors can have either analog or digital output. Analog sensors provide a continuous voltage or current signal that corresponds to the measured distance, while digital sensors provide a discrete signal indicating whether an object is detected within a certain range.

The key features of ultrasonic sensors:

1. **Distance Range:** Ultrasonic sensors can measure distances ranging from a few centimeters to several meters, depending on the sensor's design and capabilities.

2. **Non-Contact Measurement:** Ultrasonic sensors are non-contact sensors, meaning they can measure distances without physical contact with the object being measured.

3. **Wide Field of View:** Most ultrasonic sensors have a wide beam angle, which allows them to detect objects over a relatively large area.

4. **Accuracy:** The accuracy of ultrasonic sensors can vary based on the sensor's design, the environment, and the distance being measured.

Figure 5: Ultrasonic Sensor

3.1.6 HC-05 Bluetooth Module:

The HC-05 Bluetooth module is a widely used and popular Bluetooth communication module commonly found in hobbyist electronics projects and commercial products. It is based on the Bluetooth 2.0 specification and is commonly used for wireless serial communication between devices.

The key features of the HC-05 Bluetooth module:

1. **Bluetooth Version:** The HC-05 module is based on Bluetooth 2.0+EDR (Enhanced Data Rate) specification, which provides support for basic data communication.

2. **Communication Interface:** The HC-05 module uses UART (Universal Asynchronous Receiver/Transmitter) serial communication for interfacing with other devices, such as microcontrollers like Arduino or Raspberry Pi.

3. **Operating Modes**: The module can operate in either Slave mode or Master/Slave mode, allowing it to function as either a Bluetooth master or slave device.

4. **AT Commands:** The HC-05 module can be configured and controlled using AT commands sent over the UART interface. This enables users to set parameters such as device name, pairing code, and operating mode.

5. **Range:** The Bluetooth range of the HC-05 module is typically around 10 meters (approximately 33 feet) in open spaces, but it can vary based on the environment and obstacles.

6. **Power Supply:** The module operates on a 3.3V power supply and typically requires an external 3.3V voltage regulator if used with 5V systems like Arduino.

7. **Status Indicators:** Most HC-05 modules have status indicator LEDs to show the connection status and communication activity.

Figure 6: HC-05 Bluetooth Module

3.1.7 Dc motor(9v):

DC motors are widely used in various applications due to their simplicity, reliability, and ease of control. They can be found in toys, robotics, electronic devices, fans, pumps, and many other products and systems.

The key features of a DC motor (9V):

1. **Voltage Rating:** The motor is designed to operate at a specific voltage, in this case, 9 volts. Applying a voltage higher or lower than the rated voltage may affect its performance or even damage the motor.

2. **Speed and Torque:** The rotational speed (rpm) and torque output of the motor depend on the specific design and construction. Different types of DC motors (e.g., brushed, brushless) have varying speed-torque characteristics.

3. **Direction of Rotation:** DC motors can rotate in both clockwise and counterclockwise directions, depending on the polarity of the applied voltage.

4. **Operating Current:** The motor's operating current will vary based on its load and the supply voltage. It is essential to consider the motor's current rating to avoid overloading and overheating.

5. **Controlling Speed:** The speed of a DC motor can be controlled using various methods, such as pulse width modulation (PWM) or adjusting the supply voltage.

6. **Mechanical Load:** The motor's performance may be influenced by the mechanical load it drives. A heavy load may require a motor with higher torque output.

7. **Voltage Regulation:** If you are using a 9V battery to power the motor, consider using a voltage regulator or motor driver circuit to maintain a constant voltage and prevent voltage spikes that could damage the motor.

Figure 7: DC- Motor

3.1.8 Robotic Arm:

A robotic arm is a mechanical device designed to mimic the structure and functionality of a human arm. It is a fundamental component of robotic systems used in various industries and applications, from manufacturing and assembly lines to research, healthcare, and exploration.

The key components and features of a robotic arm:

1. **Links and Joints:** A robotic arm typically consists of multiple links connected by joints. The links represent the segments of the arm, and the joints allow the arm to move and articulate in various ways.

2. **Degrees of Freedom (DOF):** The number of independent movements a robotic arm can make is referred to as its degrees of freedom. Common robotic arms have anywhere from 3 to 7 DOF, which enables them to perform complex movements and reach different positions and orientations.

3. **End Effector:** The end effector is the "hand" of the robotic arm, responsible for carrying out specific tasks. It can be a gripper, a tool, a camera, or any other device needed to interact with the environment.

4. **Actuators:** Actuators are the components responsible for moving the robotic arm's joints. They can be electric motors, pneumatic cylinders, hydraulic actuators, or other types of actuators, depending on the application and requirements.

5. **Controller:** The robotic arm is controlled by a specialized controller, which interprets commands and coordinates the movement of the actuators to achieve the desired tasks.

6. **Sensors:** Robotic arms often incorporate various sensors, such as encoders, force sensors, proximity sensors, and cameras, to provide feedback on the arm's position, orientation, and interactions with the environment.

7. **Programming Interface:** To operate the robotic arm, users can interact with a programming interface, which may include software applications, graphical user interfaces, or programming languages.

12

Figure 8:Robot arm

3.1.9 Li po battery (3200 mah):

A LiPo (Lithium Polymer) battery with a capacity of 3200mAh refers to a rechargeable battery that uses lithium-ion technology with a polymer electrolyte. LiPo batteries are known for their high energy density, lightweight, and ability to deliver high current, making them popular choices for various portable electronic devices and robotics projects.

The key features of a LiPo battery (3200mAh):

1. **Capacity:** The capacity of a battery is measured in milliampere-hours (mAh), representing the amount of charge the battery can store. A 3200mAh LiPo battery can deliver a current of 3200mA (or 3.2 Amperes) for one hour or proportionally lower currents for a more extended period.

2. **Voltage:** LiPo batteries usually have a nominal voltage of 3.7 volts per cell. Multiple cells can be connected in series to achieve higher voltages (e.g., 7.4V for a 2S pack, 11.1V for a 3S pack).

3. **Rechargeable:** LiPo batteries are rechargeable and can be recharged using a compatible LiPo battery charger.

4. **Connector:** LiPo batteries come with different types of connectors, such as JST connectors, XT60, XT30, or Deans connectors, depending on the manufacturer and application.

5. **C-rating:** The C-rating of a LiPo battery indicates its maximum discharge rate relative to its capacity. For example, a 3200mAh LiPo battery with a 20C rating can deliver a maximum continuous current of 3200mAh x 20C = 64 Amperes.

6. **Care and Safety:** LiPo batteries require proper care and handling to ensure their safe and reliable operation. Overcharging, over-discharging, and physical damage can lead to fire or damage to the battery. It is essential to use a LiPo-safe charging bag, follow charging guidelines, and store and handle the batteries with care.

3.1.10 Raspberry Pi Camera Module V2 (8MP):

The Raspberry Pi Camera Module V2 is an official camera module specifically designed for use with Raspberry Pi boards. It features an 8-megapixel Sony IMX219 sensor, making it capable of capturing high-quality images and videos. The camera module is a popular choice for various applications, including photography, video recording, surveillance, and computer vision projects.

The key features of Raspberry Pi Camera Module V2 are as follows:

1. **Image Sensor:** The camera module is equipped with an 8-megapixel Sony IMX219 image sensor. It can capture still images with a maximum resolution of 3280 x 2464 pixels.

2. **Video Recording:** The camera can record videos at various resolutions, including 1080p (Full HD) at 30fps, 720p at 60fps, and VGA resolution at 90fps.

3. **Lens:** The camera module has a fixed-focus lens with an aperture of f/2.0. It provides a field of view (FOV) of approximately 62.2 degrees.

4. **Interface:** The camera module connects to the Raspberry Pi board via a ribbon cable. It uses the CSI (Camera Serial Interface) port on the Raspberry Pi for data transfer.

5. **Compatibility:** The camera module is compatible with all Raspberry Pi models that have the CSI connector, including Raspberry Pi 3, Raspberry Pi 4, and their respective variations.

6. **Software Support:** The camera module is well-supported by the official Raspberry Pi OS (previously known as Raspbian) and various third-party software libraries and tools for capturing and processing images and videos.

7. **Camera Commands:** Raspberry Pi OS provides several command-line utilities and Python libraries to control the camera module, adjust settings, and capture images and videos programmatically.

8. **NoIR Version:** There is also a NoIR (No Infrared) version of the camera module available, which lacks the infrared (IR) filter. The NoIR version is suitable for applications involving night vision or a low-light condition, as it allows the use of IR illuminators.

Figure 9:Pi camera

3.2 Software

3.2.1 Raspberry pi software:

Linux is a free open-source operating system and it belongs to the Unix operating systems. Actually, Linux means the kernel itself which is the heart of the operating system and handles the communication between the user and hardware. Normally Linux is used to refer to the whole Linux distribution. Linux distribution is a collection of software based on the Linux Kernel. It consists of the GNU-project's components and applications. Because Linux is an opensource project, anyone can modify and distribute it. That is the reason why there are many variations of Linux distributions. Most popular distributions are Ubuntu, Red Hat Linux, Debian GNU/Linux and SuSe Linux.

3.2.2 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• Python is interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

• Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

• Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

3.2.3 Python's standard library

- OpenCV: OpenCV is an open-source computer vision library used for image processing and computer vision tasks. It can be used to load and preprocess images, as well as to draw bounding boxes around objects.

- NumPy: NumPy is a library for scientific computing with Python. It provides a high-performance array object and tools for working with arrays.

- Matplotlib: Matplotlib is a library for plotting graphs in Python. It can be used to visualize the results of your object detection experiments.

CHAPTER 4: METHODOLOGY

4.1 Block Diagram

Harvesting

Servo motor 1
Servo motor 2
Servo motor 3
Servo motor 4
Servo motor 5
Servo motor 6

Raspberry Pi

Recognition

Camera

Movement

Accelerometer

IR sensor

DC motor 1
DC motor 2
DC motor 3
DC motor 4
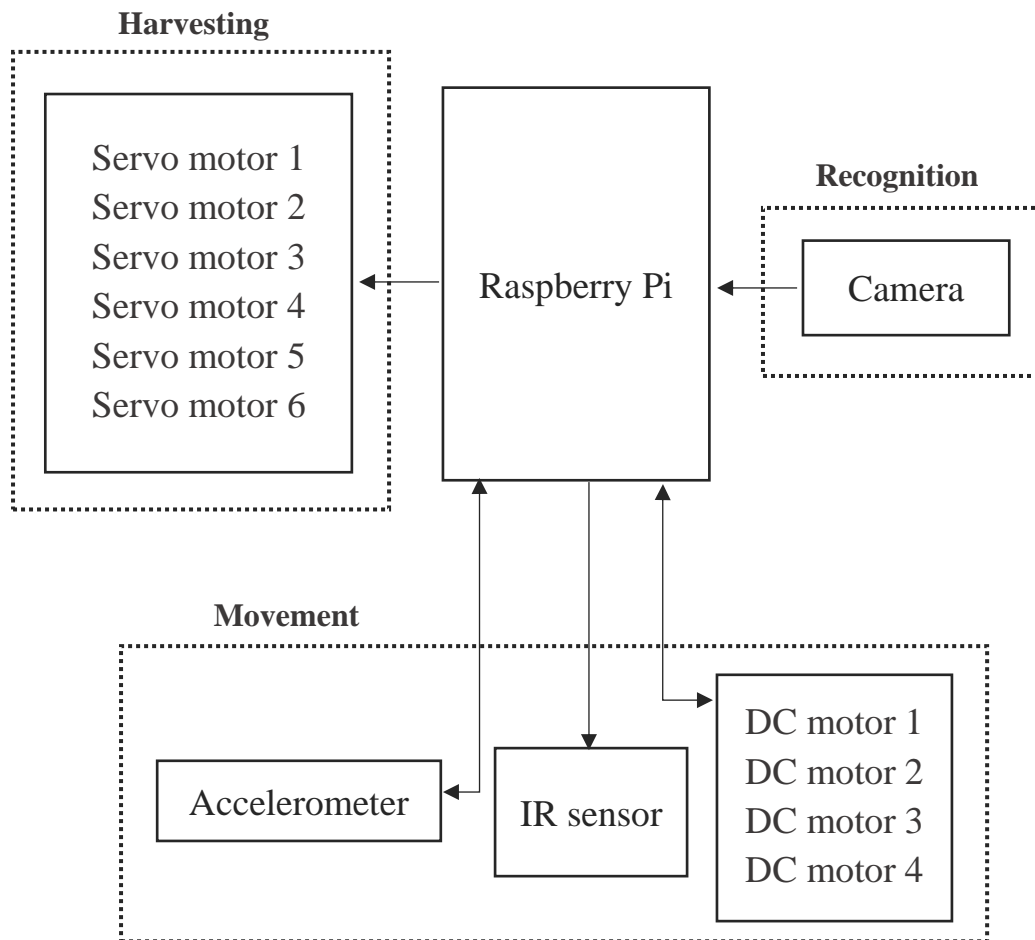
Figure 4. 1 Block Diagram

4.2 Algorithm

General algorithm
1. Move to initial position
2. Detect the fruits
3. Is tomato found, if yes move to step 5 otherwise move to step 4
4. Move to next tree and follow step 2
5. Calculate the distance
6. Sorting
7. Can robot reach if yes move to step 10 otherwise move to 8.
8. Adjust the position
9. Detect the fruits move to 7
10. Catch the fruit
11. Collect into the basket
12. Check the basket if reached to its capacity. If yes move to step 13 otherwise move to step 2.
13. Return to common tomato collection
14. Return to resume the work move to step 1

4.3 Flowchart

Figure 4. 2 General Flowchart of wall climbing robot

Training Process   **Detection Process**

```
┌─────────────────┐          ┌─────────────────┐
│ Training images /          / Test images     │
└─────────────────┘          └─────────────────┘
        │                            │
        ▼                            ▼
┌─────────────────┐          ┌──────────────────────┐
│   Cropped       │          │  ROI extraction  (3) │
│   samples       │          └──────────────────────┘
└─────────────────┘                  │
        │              ┌ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ┐
        │                            ▼
        │              │  ┌──────────────────────┐       │
        │                 │ Apply next sub-window│   (4)
        │              │  │    on the ROI        │◄──┐   │
        │                 └──────────────────────┘   │
        │              └ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─│─ ┘
        ▼                            ▼               │
┌─────────────────┐          ┌──────────────────────┐│
│ (1) HOG feature │          │ Sub-window HOG       ││
│     extraction  │          │ feature extraction (5)│
└─────────────────┘          └──────────────────────┘│
        │                            │               │
        ▼                            ▼               │
┌─────────────────┐          ┌──────────────────────┐│
│(2) SVM training │─────────▶│ Trained classifier (6)│
└─────────────────┘          └──────────────────────┘│
                                     │               │
                                     ▼               │
                              ╱ Is sub-window ╲  N    │
                           ◄─╲    fruit?     ╱       │
                           │   ╲           ╱         │
                           │        │ Y              │
                           │        ▼               │
                           │  ┌──────────────┐       │
                           │  │ Apply FCR (7)│       │
                           │  └──────────────┘       │
                           │        │               │
                           │        ▼               │
                           │   ╱ Exceed  ╲   N       │
                           │  ╲ threshold?╱──┐       │
                           │   ╲        ╱    │       │
                           │     │ Y        │       │
                           │     ▼          │       │
                           │ ┌────────────┐ │       │
                           │ │Fruit marking│ │       │
                           │ └────────────┘ │       │
                           └────────┼───────┘       │
                                    ▼               │
                             ╱ Are all sub- ╲   N    │
                            ╲ windows scanned?╱──────┤
                             ╲             ╱         │
                                  │ Y               │
                                  ▼                 │
                           ┌──────────────┐         │
                           │Image downscaling│       │
                           └──────────────┘         │
                                  │                 │
                                  ▼                 │
                           ╱ Does image reach ╲  N   │
                          ╲  minimal size?    ╱──────┘
                           ╲               ╱
                                │ Y
                                ▼
                         ┌──────────────┐
                         │   NMS    (8) │
                         └──────────────┘
                                │
                                ▼
                         / Detection results /
                                │
                                ▼
                             ( End )
```
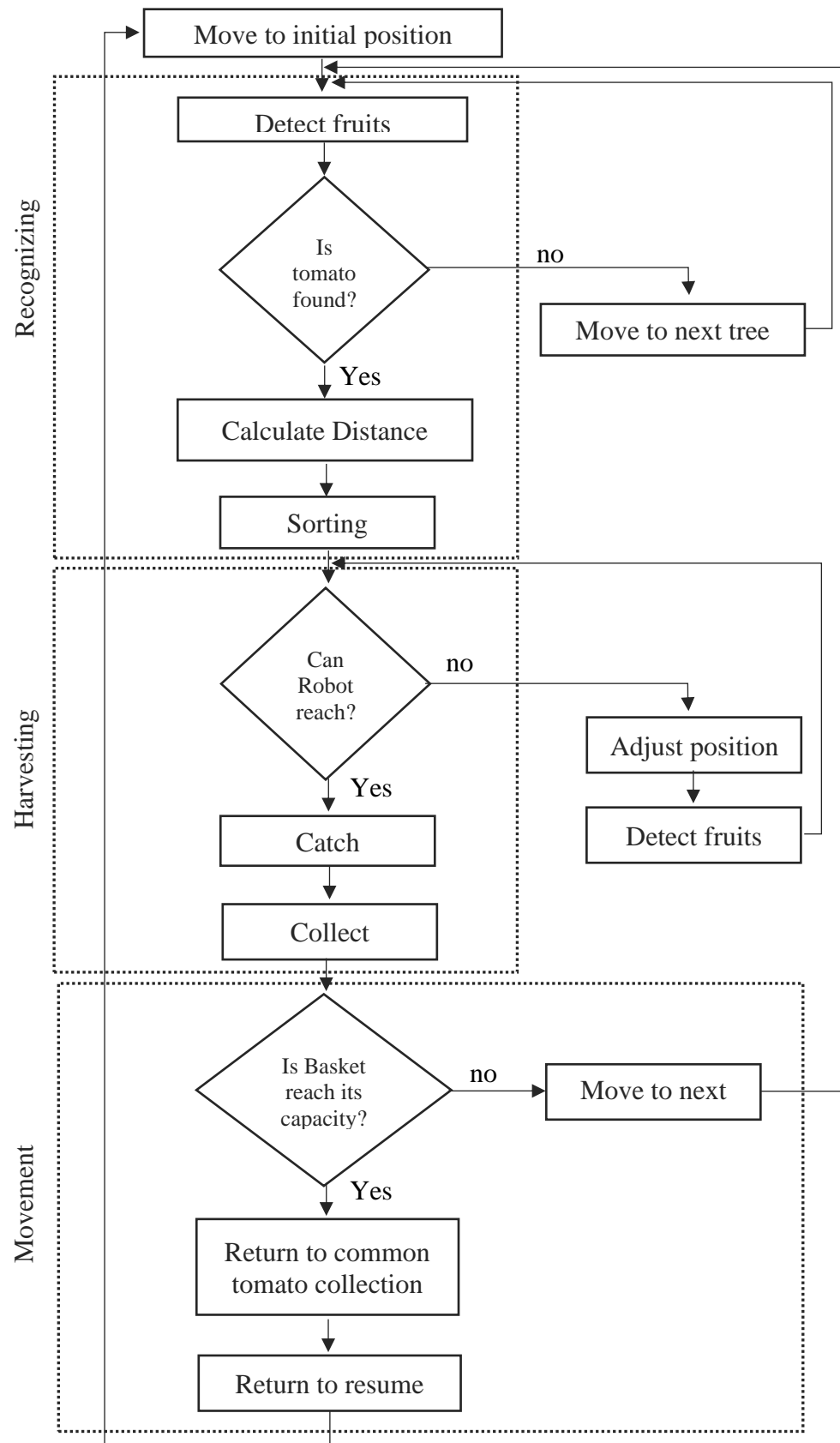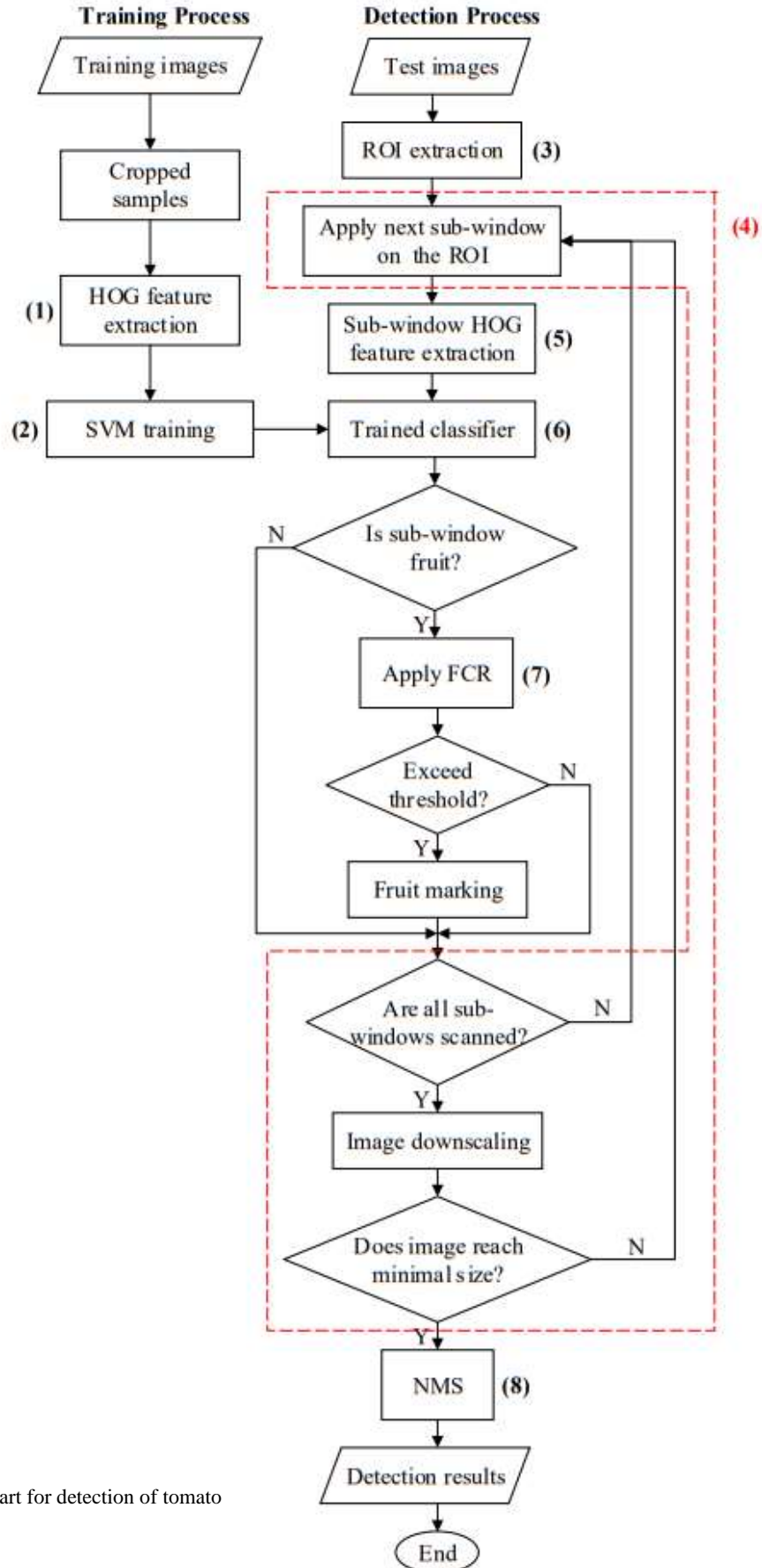
Figure 4. 3 Flowchart for detection of tomato

19

4.3.1 Flowchart explanation

Image Acquisition and Preprocessing

1.Image acquisition: A raw capture image of tomatoes of different situations is acquired from a public source.
2.Image resizing: The image is resized to 360 x 202 pixels to speed up the image processing. This is proceeded through the use of bicubic interpolation algorithm. It works by interpolating the pixels in the original image to create new pixels in the resized image. The bicubic interpolation algorithm works by fitting a cubic polynomial to the four nearest pixels in the original image. The cubic polynomial is then used to calculate the value of the new pixel in the resized image.
3.Image conversion to HSI space: The RGB color space is converted to the HSI color space. The HSI color space is a more objective color space than the RGB color space and it is better suited for image processing tasks such as image segmentation and object detection.
4.Image enhancement: An illumination enhancement method is used to reduce the effect of uneven illuminations. The image is first converted from RGB space to HSI space. The HSI image is then split into the hue (H), saturation (S), and intensity (I) channels. The I channel is then enhanced using the Contrast Limited Adaptive Histogram Equalization (CLAHE) method. The enhanced I channel is then combined with the H and S channels to obtain the final enhanced image.

The Dataset

A total of 247 images were used for the experiment. To train the SVM classifier, 100 images were randomly selected from the captured images, 72 images were used for validation set, and the remaining 75 images were used for the test. From the training images, 207 tomato samples and 621 background samples were manually cropped to construct a training set. The training samples were augmented with random rotations of 0°–360°. This doubles the size of the training set (1656 samples in all). All of the cropped samples were resized to 64×64 pixels to unify the size. The tomato samples contained a margin of about 5 pixels on all the sides. The background samples were randomly cropped to contain leaves, stems, strings, and other objects, and all the samples were separately labeled, 1 for the tomatoes and −1 for the backgrounds.

Overview of the Detection Algorithm

(1) Extracting the HOG features of the training samples
(2) Training an SVM classifier using the extracted features and corresponding labels
(3) Extracting the Region-of-Interest (ROI) on the test image using a pretrained Naive Bayes classifier
(4) Sliding a sub-window on the ROI of the image with different resolutions using an image pyramid
(5) Extracting the HOG features of each sub-window
(6) Recognizing tomatoes within the pretrained classifier
(7) Performing FCR to remove any false positive detections
(8) Merging the detection results using the NMS method

1.Extracting the HOG features of the training samples: The first step is to extract the HOG features of the training samples. HOG features are a type of feature that can be used to represent the appearance of an object. They are extracted by dividing the image into small cells and then computing the gradients of the intensity values in each cell. The HOG features are then used to train a classifier.

2.Training an SVM classifier using the extracted features and corresponding labels: The second step is to train an SVM classifier using the extracted features and corresponding labels. The labels are the ground truth, which means that they indicate whether each training sample is a tomato or not. The SVM classifier learns to distinguish between tomatoes and non-tomatoes based on the HOG features.

3.Extracting the Region-of-Interest (ROI) on the test image using a pretrained Naive Bayes classifier: The third step is to extract the Region-of-Interest (ROI) on the test image using a pretrained Naive Bayes classifier. The ROI is the part of the image that is likely to contain a tomato. The Naive Bayes classifier is used to identify the ROI based on the color and texture of the image.

4.Sliding a sub-window on the ROI of the image with different resolutions using an image pyramid: The fourth step is to slide a sub-window on the ROI of the image with different resolutions using an image pyramid. An image pyramid is a set of images with different resolutions. The sub-windows are slid on the ROI of the image at different resolutions to ensure that tomatoes of all sizes can be detected.

5.Extracting the HOG features of each sub-window: The fifth step is to extract the HOG features of each sub-window. The HOG features are extracted from each sub-window to represent the appearance of the sub-window.

6.Recognizing tomatoes within the pretrained classifier: The sixth step is to recognize tomatoes within the pretrained classifier. The pretrained classifier is used to classify each sub-window as a tomato or not a tomato.

7.Performing FCR to remove any false positive detections: The seventh step is to perform FCR to remove any false positive detections. FCR is a method for removing false positive detections. It works by merging the detection results of overlapping sub-windows.

8.Merging the detection results using the NMS method: The eighth step is to merge the detection results using the NMS method. NMS is a method for merging overlapping detections. It works by keeping the detection with the highest score and removing the detections with lower scores.

CHAPTER 5: EXPECTED RESULT

Our project aims to develop an autonomous tomato harvesting robot with advanced capabilities for efficient and precise harvesting. The key expected results include:

1. Tomato Recognition: Equipped with a sophisticated vision system, the robot will accurately recognize ripe tomatoes using image processing and machine learning algorithms.

2. Distance Calculation and Sorting: The robot will prioritize harvesting the nearest ripe tomatoes for optimized efficiency.

3. Automated Robotic Arm: A 6-degree-of-freedom robotic arm will delicately detach ripe tomatoes to minimize crop damage.

4. Collection Mechanism: Harvested tomatoes will be securely placed in a collecting basket, ensuring smooth and efficient harvesting.

5. Continuous Harvesting: The robot's intelligent navigation system will enable seamless detection and harvesting of ripe tomatoes in the field.

6. Intelligent Basket Management: The robot will autonomously navigate back to a designated common place when the collecting basket reaches its maximum capacity.

7. Resuming Harvesting: After clearing the basket, the robot will resume harvesting from the last point, eliminating duplicate harvesting.

Our project aims to revolutionize tomato harvesting, offering a cost-effective and labor-saving solution that enhances productivity and quality in agriculture. The autonomous robot will contribute to a sustainable and efficient future for tomato harvesting, benefiting farmers and the agricultural community.

CHAPTER 6: TIME SCHEDULE

6.1 Gantt chart

GANTT CHART (TIME SCHEDULE)

Tabel 1Gant chart

Task ID	Task Title	Effort (hours)	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	DELIVERABLES
1	Project Initiation	110	38	18	35	8	11								Selection of a feasible project
1.1	Define project objectives and scope	18	18												Identifying and writing basic objectives and scope of project
1.2	Conduct initial research and feasibility analysis	92	20	18	35	8	11								Literature review, scope of the project, objectives, methodology and expected output
2	Requirement Gathering	60		15	20	10	15								Essential hardware and software materials
2.1	Study of existing Robot (Design)	60		15	20	10	15								Information about the existing robot and things to change and modify
3	Hardware	85						20	30	20	15				
3.1	Hardware Implementation	50						10	20	10	10				Hardware collection and building a system
3.2	Programming with Pie	35						10	10	10	5				Programming the system to perform accordingly
4	Software	378				30	35	55	25	45	45	50	45	48	
4.1	Develop Software	89				10	8	23		10	8	10	10	10	A software to support the model
4.2	Compiling	197				20	22	17	15	20	25	30	20	28	Training of various data sets in ML and testing them accordingly to get accurate results
4.3	Debugging and Testing	92					5	15	10	15	12	10	15	10	Testing the system and checking either any error is present or not and if present debugging those errors and solving it
5	Report	123	10	20	20	10	10		5	10	8	10	10	10	Report according to the progress of project
	Total Hours	756	48	53	75	58	71	75	60	75	68	60	55	58	

CHAPTER 7: COST ESTIMATION

Tabel 2 Cost estimation

S.N	Components	Quantity	Market Cost (Rs.)	Total (Rs.)	Remarks
1	Raspberry Pi 4 4 GB	1	12000	12000	Available in College
2	Raspberry Pi 500W infrared Camera module	2	1500	3000	Himalayan Solution
3	**Arduino Mega 2560**	1	800	800	Available in College
4	Servo Motor – MG966R Series	6	400	2400	Available in College
5	6-Channel 12-bit PWM/Servo Driver - I2C interface - PCA9685	1	1000	1000	Available in College
6	Buck Converter	1	250	250	Available in College
7	Ultrasonic sensor	2	200	200	Available in College
8	H3-05 Bluetooth Module	1	200	200	Available in College
9	Dc motor(9v)	4	500	2000	Himalayan Solution
10	Robotic Arm	1	_	_	Himalayan Solution
11	Wheel	4	50	200	Available in College
12	Li po battery (3200 mah)	1	1200	1200	Available in College
13	Chassis	1	_	_	Himalayan solution
14	Accelerometer Sensor (Gy-45 MMA845X)	1	390	390	""
15	L293D Dc motor Driver shield	1	650	650	""
16	IR sensor	4	275	1100	Himalayan Solution
17	Weight Sensor 20kg	1	225	225	Himalayan Solution
15	Total			26615	

REFERENCES

[1] K. N. T. H. a. M. I. Hiroaki Yaguchi, "Development of An Autonomous Tomato Harvesting Robot with," *International Conference on Intelligent Robots and Systems (IROS),* 2016.

[2] 2. ,. S. M. 2. a. J. H. K. 2. Guoxu Liu 1, "A Mature-Tomato Detection Algorithm Using Machine," *International Conference onMachine Learning and Computing (ICMLC),* 2019.

[3] X. W. G. W. Z. L. Qingchun Feng, "Design and Test of Tomatoes Harvesting Robot," *International Conference on Information and Automation,* 2015.

[4] F. z. Chen Huangfei, "Design and research on the End Actuator of Tomato Picking," *IOP Conf. Series,* 2019.

[5] J. X. Tao Zhu*, "Chassis Design of Tomato Picking Robot in the Greenh," *Journal of Physics: Conference Series,* 2021.