TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Kalimati, Kathmandu

Department of Electronics, Communication
And Information Engineering

A Major Project Report On

"Automatic tomato harvesting with Machine learning"

Submitted By:

Mausam Gurung (KAT076BEI014)
Shalon Maharjan (KAT076BEI027)
Shubham Bista (KAT076BEI028)
Symon Shrestha (KAT076BEI030)

Kathmandu, Nepal

March -2024

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Kalimati, Kathmandu

A Major Project Report On

“Automatic tomato harvesting with Machine learning”

Submitted By:

Mausam Gurung (KAT076BEI014)
Shalon Maharjan (KAT076BEI027)
Shubham Bista (KAT076BEI028)
Symon Shrestha (KAT076BEI030)

Submitted To:

Department of Electronics, Communication
and Information Engineering In Partial Fulfillment Of The Requirement For The Bachelor Of Engineering

Kathmandu, Nepal

March -2024

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Kalimati, Kathmandu

Department of Electronics, Communication
And Information Engineering

CERTIFICATE

The undersigned certify that they have read and recommended to the Department of Electronics, Communication and Information Engineering a major work entitled "Automatic Tomato Harvesting with Machine Learning" submitted by Mausam Gurung (KAT076BEI014), Shalon Maharjan (KAT076BEI027), Shubham Bista (KAT076BEI028) and Symon Shrestha (KAT076BEI030) in partial fulfillment of the requirements for the degree of Bachelor of Engineering.

Associate Prof. Sagun Manandhar (Project Supervisor) Department of Electronics, Communication and Information Engineering (Kathmandu Engineering College)	Prof. Ram Krishna Maharjan (External Examiner) Department of Electronics and Computer Engineering Pulchowk Campus Institute of Engineering (IOE)
Er. Pushpa Dhamala (Project Supervisor) Department of Electronics, Communication and Information Engineering (Kathmandu Engineering College)	Associate Prof. Rajan Lama (Head of Department) Department of Electronics, Communication and Information Engineering (Kathmandu Engineering College)

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to **Er. Sagun Manandhar** and **Er. Pushpa Dhamala,** our Major Project supervisors, for their unwavering support and guidance throughout the project, helping us turn our ideas into reality. Their invaluable experience served as a guide, instilling us with the confidence and curiosity to tackle problems and see the bigger picture.

We are also deeply grateful to **Er. Rajan Lama**, The Head of The Department of Electronics, Communication, and Information Engineering, for providing us with the opportunity to work on the project. Our heartfelt thanks go out to our project coordinator, **Er. Dipen Manandhar**, for effectively managing and coordinating the project, and for his timely support and guidance.

We would like to extend our gratitude to all the teaching staff of The Department of Electronics, Communication, and Information Engineering for their constant encouragement, support, constructive criticism, and guidance, and for their keen interest in our project. We are also fortunate to have received support from research paper authors whose interest in our work helped to clear up any blind spots we encountered and refine our ideas. Without their support, our project would not have progressed as smoothly.

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 6-degrees-of-freedom (6DOF) robotic arm, enabling precise and gentle fruit detachment to minimize damage and waste. At its core lies a sophisticated deep learning neural network, a lightweight model tailored to meet the real-time application requirements of CNNs in low-cost embedded systems such as IoT applications. Employing the state-of-the-art YOLOv4 tiny object detection with the Darknet framework, the system accurately identifies and locates unripe, semi-ripe, and ripe tomatoes in the field, surpassing traditional methods and ensuring efficient and accurate harvesting. By combining the dexterous 6DOF arm with vision from an Insta360 camera and YOLOv4's tiny yet robust detection capabilities in real-time, along with integrating a TOF sensor for depth calculation, our project paves the way for a future of autonomous and intelligent tomato harvesting. This innovative approach holds immense potential to boost agricultural efficiency and contribute to a more sustainable food production system.

Keywords: Darknet, 6DOF's, Robot arm, Raspberry Pi , deep learning neural network, Machine learning, computer vision, YOLOv4-tiny, CNN's, object detection, un-ripe, semi-ripe and ripe tomato detection, TOF sensor, 360 Insta web camera.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATION

LIST OF ABBREVATION

1. AUC: Area Under the Curve
2. Li-ion: Lithium-Ion battery
3. YOLO: You Look Only ONCE
4. IoU: Intersection over Union
5. GIoU: Generalized Intersection over Union
6. DIoU: Distance Intersection over Union
7. CIoU: Complete Intersection over Unio

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 arm 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 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.

1.1.2 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 recognition system employs a combination of state-of-the-art technologies to efficiently identify tomatoes. Utilizing YOLO V4 Tiny for object detection, it ensures precise and reliable tomato recognition. Integrated with a high-resolution webcam, the system enables real-time video processing, providing visual data for YOLO V4 Tiny analysis, acting as the system's eyes, the webcam facilitates accurate tomato detection. To enhance precision in tomato harvesting, the system incorporates the VL53L0X Time of Flight (TOF) Distance Sensor. This laser-based sensor measures distances, enabling

the system to determine the exact proximity of tomatoes within the plant. This additional layer of information enables optimal harvesting conditions providing x, y, z real time co-ordinate value ensuring tomatoes are picked at their peak ripeness.

1.1.2.1 YOLOv4-Tiny Architecture:

YOLOv4-Tiny utilizes a couple of different changes from the original YOLOv4 network to help it achieve these fast speeds making a real time framework. First and foremost, the number of convolutional layers in the CSP backbone are compressed with a total of 29 retrained convolutional layers. Additionally, the number of YOLO layers has been reduced to two instead of three and there are fewer anchor boxes for prediction.

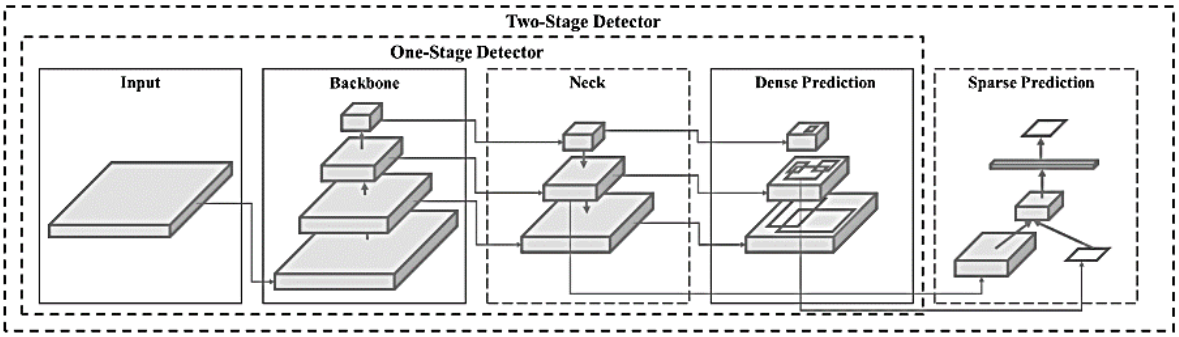


Figure 1. 1: YOLO*v4*- Architecture

The YOLOv4-tiny architecture is a lightweight version of the YOLOv4 object detection model, designed to balance computational efficiency with high performance. Here's an overview of its architecture:

Input: YOLOv4-tiny takes an input image of fixed dimensions, typically 416x416 pixels.

Backbone: The backbone of YOLOv4-tiny consists of a series of convolutional layers, which extract features from the input image. This backbone network is usually a variant of Darknet, a convolutional neural network architecture optimized for object detection tasks.

Neck: Following the backbone, YOLOv4-tiny includes a "neck" component, which further refines the features extracted by the backbone. This may involve additional convolutional layers or other operations to enhance feature representation.

Head: The head of the YOLOv4-tiny architecture comprises several convolutional layers responsible for predicting bounding boxes, objectness scores, and class probabilities. YOLOv4-tiny utilizes anchor boxes, predetermined shapes that are used to predict object locations and sizes within the image.

Output: The final output of YOLOv4-tiny consists of a grid of cells, each associated with a set of bounding boxes and corresponding confidence scores for different classes. These bounding boxes represent the predicted locations of objects within the input image.

1.1.2.2 Performance Optimizations:

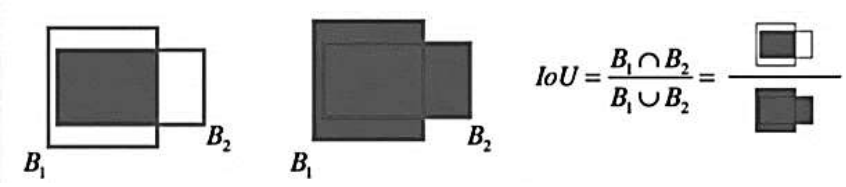
- Bag of Freebies: A "bag of freebies" approach in machine learning refers to using various optimization methods during training and inference without increasing inference time. These methods, which can include techniques like data augmentation, learning rate schedules, and weight decay, are applied during training to improve model performance without adding complexity to the inference process. While these techniques can increase the training time due to additional computations and adjustments, they ultimately result in a more robust and efficient model that performs better in real-world scenarios.

- Bag of Specials: The "bag of specials" approach in machine learning involves incorporating different specialized modules into a model to significantly improve its accuracy. These modules can include advanced architectural features, novel activation functions, or specialized layers tailored to the specific task. While these additions can enhance the model's performance, they may also lead to a slight increase in inference time due to the additional computations required by the specialized modules. Despite this, the trade-off is often considered worthwhile as the improved accuracy can lead to better overall performance in practical applications

1.1.2.3 Metrics used to evaluate the quality of object detection bounding boxes:

Intersection over Union (IoU) loss, also known as Jaccard loss, is a popular loss function used in object detection tasks, particularly in the training of neural networks. IoU loss is a crucial component in training object detection models, as it guides the optimization process towards generating more accurate bounding box predictions.

Intersection Union Intersection over Union



Equation 1: IoU

Figure 1. 2 Intersection over Union

GIoU(Generalized Intersection over Union): GIoU measures the extent of overlap between two bounding boxes while considering their geometric properties. It provides a more accurate assessment of the spatial agreement between the predicted and ground truth bounding boxes compared to traditional Intersection over Union (IoU) metrics.

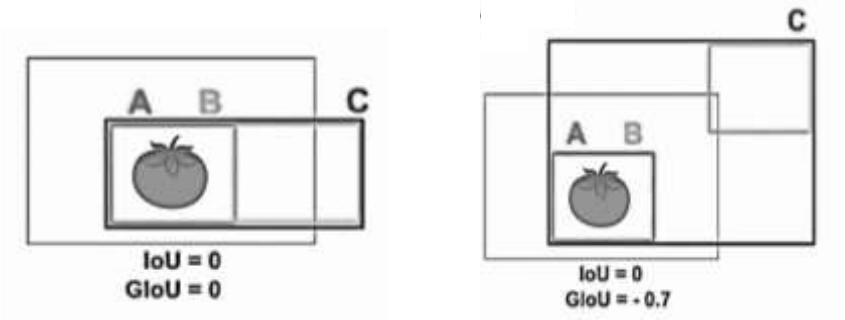


Figure 1. 3 Generalized Intersection over Union

$$L_{GIoU} = 1 - IOU + \frac{|C - B \cup B^{gt})|}{|C|}$$

Equation 2: GIoU

DIoU (Distance IoU): DIoU extends GIoU by incorporating distance information between bounding boxes' centers. This additional measure penalizes predictions that are far away from ground truth bounding boxes, leading to improved localization accuracy.

$$L_{DIoU} = 1 - IOU + \frac{\sigma^2(b, b^{gt})}{c^2}$$

Equation 3: DIoU

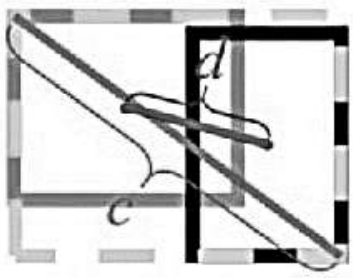


Figure 1. 4 Distance Intersection over Union

DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized. *c* is the diagonal length of the smallest enclosing box covering two boxes, and $d = \sigma(b,b^{9t})$ is the distance of central points of two boxes.

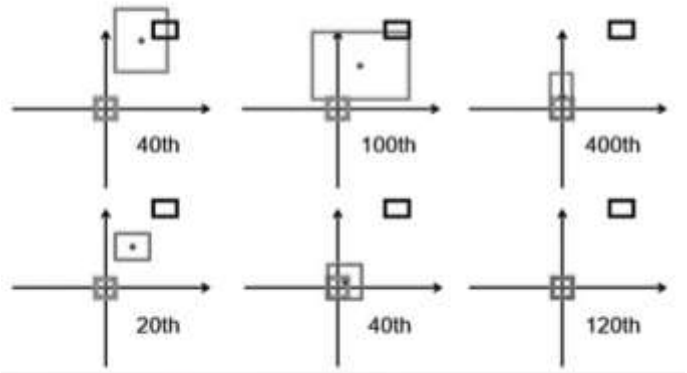


Figure 1. 5 DIoU over different epoch

CIoU (Complete IoU): CIoU further enhances DIoU by introducing additional terms to account for aspect ratio discrepancies and box overlap. It addresses the limitations of previous metrics by considering both spatial and structural information, resulting in more precise bounding box evaluations. It considers Overlapping Area, Distance between centers and Aspect ratios.

$$L_{DIoU} = 1 - IOU + \frac{\sigma^2(b,b^{gt})}{c^2} + \alpha v$$

Equation 4: CIoU

Where, $v = \frac{4}{\pi^2}(arc\ tan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})$

$$\alpha = \frac{v}{(1-IOU)+v}$$

1.1.2.4 Working diagram of YOLO V4-TINY

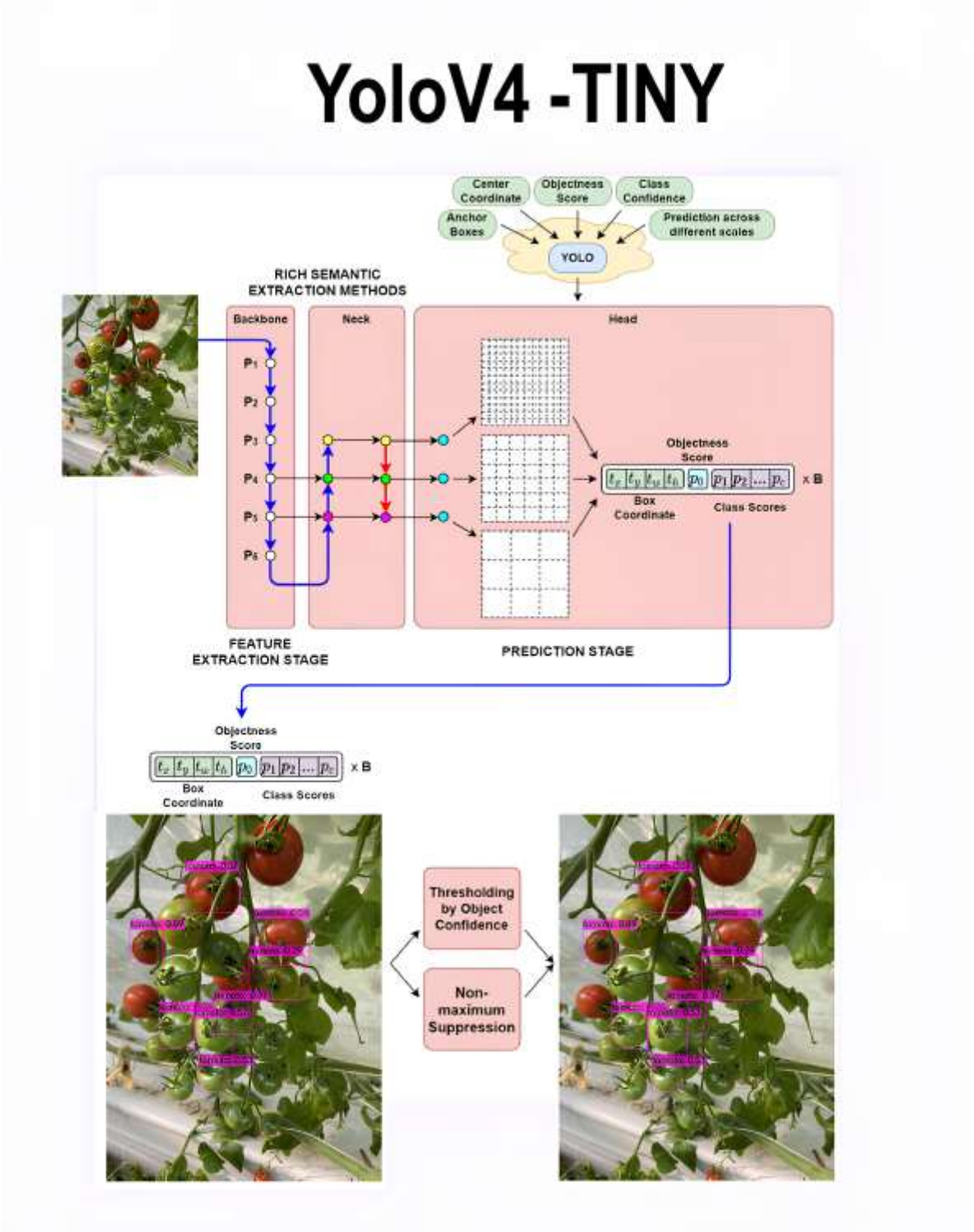


Figure 1. 6 Working diagram of YOLO V4-TINY

1.1.2.5 The NVIDIA Tesla T4 GPU:

The NVIDIA Tesla T4 GPU is a robust choice for accelerating AI inference, offering up to 40 times faster performance for deep learning inference with its Turing Tensor Cores. It features RT Cores for ray tracing, 15GB of GDDR6 memory, and a 70W power consumption, making it efficient for various tasks. With multi-user support, it's suitable for applications like image classification, NLP, and recommender systems. While not optimized for training, its balance of performance, efficiency, and affordability makes it an attractive option for cloud, edge, and on-premise deployments.

1.1.3 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. Following the detection of tomatoes, the robotic arm is adjusted to reach the tomatoes by applying kinematics.

1.1.3.1 Kinematics:

There are two main types of kinematic models: forward kinematics and inverse kinematics. In forward kinematics, the length of each link and angle of each joint is given, and through that, position of any point (x, y, z) can be found. In inverse kinematics, the length of each link and position of some points (x, y, z) is given, and the angle of each joint is given to obtain that position.

Transformation matrix of each joint

$$A_i = \begin{bmatrix} \cos\theta i & -\sin\theta i \cos\alpha i & \sin\theta i \sin\alpha i & \alpha i \cos\theta i \\ \sin\theta i & \cos\theta i \cos\alpha i & -\cos\theta i \sin\alpha i & \alpha i \sin\theta i \\ 0 & \sin\alpha i & \cos\alpha i & di \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We get A1, A2, A3, A4, A5 and A6 for each respective joints. Multiplying all the matrices we can calculate the final transformation matrix that is T_0^6 . But we also multiply A1, A2 & A3 and A4, A5 & A6 to get T_0^3 and T_3^6 respectively. For matrices R and P:

The final transformation matrix, which can be written in the form:

$$T_6^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \times T_5^4 \times T_6^5,$$

$$T_6^0 = \begin{bmatrix} r11 & r12 & r13 & Px \\ r21 & r22 & r13 & Py \\ r31 & r32 & r33 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^0 = \begin{bmatrix} R_6^0 & P_6^0 \\ 0 & 1 \end{bmatrix}.$$

From where we can extract R and P matrices.

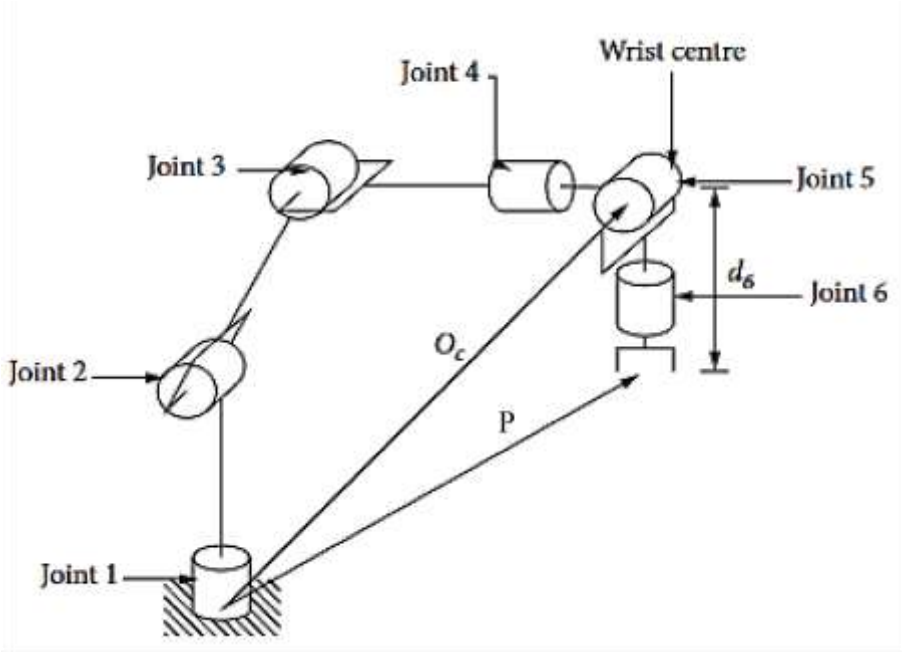


Figure 1. 7 Kinematics *decoupling*

$$P = O_c + d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

Where O is the position P_6^0 and R is orientation which is R. The above equation can be written in terms of O_c as follow

$$O_c = P - d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

The first three joints can be found in the following steps. They will determine the position of the manipulator:

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix},$$

$$O_c = \begin{bmatrix} x_c \\ y_c \\ Z_c \end{bmatrix},$$

$$\begin{bmatrix} x_c \\ y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} P_x - d_6 r_{13} \\ P_y - d_6 r_{23} \\ P_z - d_6 r_{33} \end{bmatrix}.$$

Solution:

Θ_1= Atan2(x_c , y_c).

Θ_3 = atan2(D, $\pm\sqrt{1 - D^2}$ where cos Θ_3 = D

Θ_2 = a tan 2$\left(\sqrt{x_c^2 + y_c^2}\, , z_c - d_1\right)$ – a tan2(a_2+a_3 cosΘ_3,a_3 sin Θ_3)

To solve for Θ_4, Θ_5 and Θ_6, which are Euler angles, we use Euler Transformation and

$R = R_3^0 R_6^3$

We get:

Θ_5 =atan2($r_{33,}$ $\sqrt{1 - r33\ \hat{}2}$

Θ_4 =atan2($r_{13,}$ r_{23})

Θ_6,= atan2(-$r_{31,}$ r_{32}) or Θ_6,= atan2($r_{31,}$ -r_{32})

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

- To 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.

1.4 Scope and Application

- 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, Chapter 5 consists of the Epilogue where we have results, conclusion and further enhancement and finally Chapter 6 consist Bibliography.

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.

The paper introduces an autonomous tomato harvesting robot equipped with a rotational plucking gripper, a stereo camera, and an omnidirectional mover. It proposes a tomato fruit detection algorithm based on color extraction, Euclidean clustering, and sphere fitting. Additionally, an inverse kinematics-based harvesting algorithm is presented. The robot system is evaluated through experiments in a tomato robot competition and a real farm, measuring harvesting time, success rate, and failure cases. Challenges and future improvements include grasp state estimation, tomato stem recognition, and multiple view measurement. [1]

The paper proposes an innovative algorithm for automatic tomato detection in regular color images. It combines Histograms of Oriented Gradients (HOG) features with a Support Vector Machine (SVM) classifier to achieve accurate tomato recognition. The algorithm employs a coarse-to-fine scanning method, False Color Removal (FCR), and Non-Maximum Suppression (NMS) to enhance detection accuracy and efficiency. Evaluated on a dataset of 247 tomato images under various conditions, the algorithm achieves impressive performance metrics, outperforming existing methods. Researchers can leverage these findings for agricultural automation and harvesting robot development. [2]

The paper introduces YOLO-Tomato, a modified YOLOv3 framework designed for tomato detection in complex environments using harvesting robots. By incorporating a densely connected architecture and a spatial pyramid pooling module, YOLO-Tomato enhances feature extraction and detection accuracy for small and occluded tomatoes. Evaluation against other state-of-the-art methods demonstrates YOLO-Tomato’s superior performance in terms of average precision, F1-score, and real-time detection speed. [3]

In their research, the authors have developed an efficient tomato-detection method that significantly improves the performance of the YOLOv4-tiny model in complex environments. Challenges in fruit detection arise due to factors such as occlusion by leaves or branches, illumination variations, and fruit overlap or clustering. The proposed solution involves modifications to the backbone network, integration of a tiny CSP-SPP module, and the use of a CARAFE module for high-resolution feature maps. Experimental results demonstrate superior accuracy and speed compared to the original YOLOv4-tiny and other state-of-the-art methods. The proposed method achieves impressive precision, recall, F1 score, and mAP, making it a promising solution for real-world applications in agriculture. [4]

The paper introduces YOLOv4, an efficient and accurate object detector. YOLOv4 leverages techniques such as Bag-of-Freebies and Bag-of-Specials to enhance training and inference without increasing model complexity. The backbone network, CSPDarknet53, optimizes object detection with a large receptive field and various modules. Experimental results demonstrate YOLOv4's state-of-the-art performance on the MS COCO dataset, outperforming other real-time detectors. [5]

This study presents novel loss functions, Distance-IoU (DIoU) and Complete IoU (CIoU), enhancing object detection models. Integrating them during custom object detector training improves bounding box regression accuracy. Adopting DIoU for non-maximum suppression (NMS) during inference enhances the efficiency of eliminating redundant bounding boxes. Experimental evaluations on benchmark datasets demonstrate effectiveness, showcasing improvements in IoU and GIoU. [6]

The paper presents a comprehensive study on the design and evaluation of an intelligent tomato harvesting robot. It covers aspects such as the robot's vision positioning system, flexible gripper, and performance testing in a greenhouse environment. The proposed system demonstrates promising results in terms of picking efficiency and success rate, with potential for further enhancements. The relevant literature supports the research, providing a solid foundation for this innovative agricultural application. [7]

In this paper, the authors present a comprehensive literature review on fruit and vegetable picking robots and their end-effectors. They analyze existing methods, discuss advantages and disadvantages, and propose a novel design for a tomato-picking robot's end-effector. The study includes kinematic analysis, workspace evaluation, and control system design.

Picking tests validate the end-effector's performance, while the paper also highlights areas for improvement and future research directions. [8]

The authors propose a versatile and expandable chassis system for a greenhouse tomato picking robot. This system integrates mechanical components, driving transmission, chassis shell, and electrical control. Through performance analysis, simulation, and prototype experiments, the feasibility of the proposed chassis system is demonstrated. [9]

The study aimed to develop and validate kinematic models for a 6 DOF serial robot arm and analyze its workspace. Using Denavit-Hartenberg parameters and an analytical approach, the authors derived both forward and inverse kinematics. MATLAB was employed to verify forward kinematics and plot the robot arm's workspace. In real-world testing, the inverse kinematics successfully provided accurate joint angles for precise end-effector positioning within the workspace, achieving ±0.5cm precision. The study concludes that these accurate and versatile kinematic models have potential applications in industrial and educational contexts. [10]

The paper details the creation of a 6-DOF robot arm, modeled on the AR3. Using Solid works for 3D modeling and CNC technology for manufacturing, a cost-effective prototype was developed. This robot arm can perform basic tasks and serves as a reference for future low-cost robot arm developmen. [11]

This paper reviews the significant advancements in fruit harvesting robots over the past three decades, with a primary focus on recognition and picking systems. Challenges include improving detection accuracy, minimizing fruit damage, increasing harvesting speed, and adapting to diverse environmental conditions. Researchers have explored various types of fruits (such as tomatoes, cucumbers, apples, and strawberries) and employed sensors like color cameras and range sensors for fruit recognition. Additionally, end-effectors such as grippers and cutters, along with methods like thermal cutting and laser cutting, have been used for efficient fruit picking. This review provides valuable insights for agricultural robotics researchers and practitioners [12]

The study investigates the kinematics of a 6-DoF articulated robot with a spherical wrist, using the Comeau NM45 manipulator as an example. Employing the modified Denavit–Hartenberg convention, the paper derives forward and inverse kinematics equations for the robot. The kinematics decoupling technique simplifies the inverse kinematics problem. Verification through MATLAB simulations demonstrates that the robot can follow smooth trajectories. Overall, this approach is applicable to other articulated robots with a spherical

wrist, offering accurate and efficient kinematics solutions for practical implementation in robotics research and development. [13]

This paper describes the design and implementation of a line following robot, which is an autonomous robot that can follow a black or white line on a contrasting surface. The robot uses infrared sensors to detect the line and a motor driver to control the movement of the wheels. [14]

The paper proposes a method for a line follower robot that computes the radius of curvature of the line using infrared line sensors. The sensor layout and method selection significantly impact the robot's response accuracy and speed. Additionally, the robot is equipped with an anti-collision system using an ultrasonic distance sensor to detect and avoid obstacles, particularly during line crossovers when other robots share a common complex line. [15]

CHAPTER 3: RELATED THEORY

For the implementation of any system, hardware and software components are the essential parts. Thus, the integral parts of the hardware to be used in our major project are mentioned below:

3.1 Arduino Mega

The Arduino Mega is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Figure 3. 1 Arduino Mega

3.2 Servo Motor – MG996R Series:

The MG996R Series servo motor is a versatile and reliable component widely used in robotics and RC vehicles. Its high-torque, metal-geared design enables precise control over angular position, making it ideal for applications requiring accurate and repeatable movement. The MG996R is valued for its durability and performance, making it a popular choice among hobbyists and professionals alike.

Figure 3. 2 Servo Motor – MG996R Series

3.3 Infrared (IR) sensors

Infrared (IR) sensors detect or emit infrared radiation, used for proximity sensing, object detection, and communication. They work by detecting the infrared radiation emitted by objects, converting it into an electrical signal. IR sensors come in various types and find applications in robotics, automation, security systems, and industrial processes. They offer advantages such as working in various lighting conditions, non-contact sensing, and affordability, but are susceptible to interference and have limited range compared to other sensing technologies. Overall, IR sensors are versatile tools for detecting objects based on their infrared radiation.

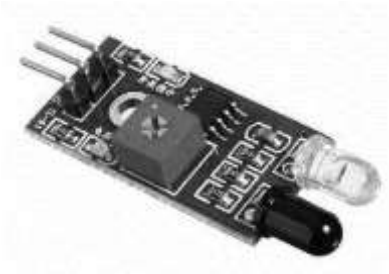


Figure 3. 3 Infrared (IR) sensors

3.4 L293D MOTOR Driver

The L293D is a popular motor driver IC (integrated circuit) that can control two DC motors in both directions. It is commonly used in robotics and other projects to control the speed and direction of motors. The L293D can handle a wide range of voltages (up to 36V) and currents (up to 600mA per channel), making it versatile for various motor types. It is easy to use and requires minimal external components, making it a popular choice for hobbyists and professionals alike.

Figure 3. 4 L293D MOTOR Driver

3.5 VL53L0X TOF Distance Sensor:

The VL53L0X is a Time-of-Flight (ToF) distance sensor that provides accurate distance measurement in a compact module. It uses a laser to measure the time taken for the light to reflect back from a target, allowing it to calculate the distance. The sensor has a range of up to 2 meters and offers high accuracy and fast response times. It is commonly used in robotics, drones, and other applications requiring precise distance sensing.

Figure 3. 5 VL53L0X TOF Distance Sensor

3.6 Insta 360 Webcam:

The Insta360 webcam is a compact and versatile camera designed for capturing immersive 360-degree video and photos. It features dual lenses that capture a full 360-degree field of view, allowing you to capture everything around you in stunning detail. The camera is equipped with advanced image stabilization technology to ensure smooth and steady footage, even in challenging conditions. It can be easily connected to your computer via USB for use as a webcam, allowing you to stream live 360-degree video or video conference in a unique and engaging way. The Insta360 webcam is compatible with a wide range of video conferencing and streaming platforms, making it a versatile tool for content creators, educators, and professionals alike.

Figure 3. 6 Insta 360 Webcam

3.7 GY-45 (MMA845x) Accelerometer:

The GY-45 (MMA845x) accelerometer is a small sensor module that measures acceleration in three axes: X, Y, and Z. It is based on the MMA845x series of accelerometers from NXP Semiconductors. The sensor can measure acceleration in the range of ±2g, ±4g, or ±8g, depending on the specific model. It communicates with a microcontroller or other devices via I2C or SPI interface, providing digital acceleration data for further processing. The GY-45 accelerometer module is commonly used in applications such as motion detection, tilt sensing, and vibration monitoring in devices like smartphones, tablets, and wearable devices.

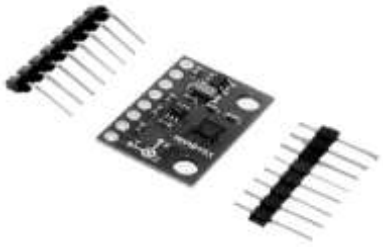


Figure 3. 7 GY-45 (MMA845x) Accelerometer

3.8 6DOF Robotic Arm:

A 6DOF (Degrees of Freedom) robotic arm is a mechanical arm-like structure capable of moving in six different ways in 3D space. It can rotate horizontally at the base, move up and down vertically, bend at the elbow, tilt the wrist up and down, rotate the wrist horizontally, and twist the wrist from side to side. These movements allow the arm to reach and manipulate objects from various angles and orientations, making it ideal for tasks such as manufacturing, assembly, and research. The arm is controlled using motors or actuators at each joint for precise movement and positioning of the end effector.

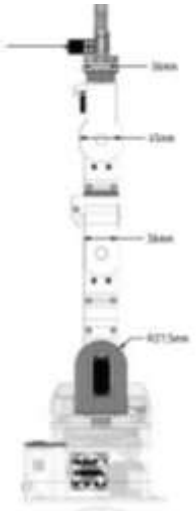


Figure 3. 8 6DOF Robotic Arm

3.9 Li po battery (3200 mAh):

A 3200mAh LiPo (Lithium Polymer) battery is a rechargeable power source commonly used in drones, RC vehicles, and portable electronics. It offers a high energy density, providing longer runtimes in a compact package. LiPo batteries have a nominal voltage of 3.7 volts per cell and should be handled with care to prevent damage or safety hazards. They are versatile and widely used in various electronic devices.

Figure 3. 9 Li po battery (3200 mAh)

3.10 Load cell:

A load cell is a transducer that converts force or load acting on it into an electrical signal. Load cells are used in various applications for measuring force, weight, or torque. They are commonly used in industrial scales, material testing machines, and force measurement devices. Load cells come in different types, such as strain gauge load cells, hydraulic load cells, and piezoelectric load cells, each suitable for different applications and load ranges. Strain gauge load cells, for example, use a strain gauge mounted on a flexible element that deforms under load, causing a change in resistance that is proportional to the applied force. Load cells provide accurate and reliable measurements and are essential in many industries for ensuring safety, quality control, and efficiency.

Figure 3. 10 Load cell

3.11 Nema 17 Stepper motor:

The Nema 17 stepper motor is a compact and versatile motor widely used in 3D printers, CNC machines, and automation equipment. Named after its standard NEMA 17 size specification of 1.7 x 1.7 inches (43.2 x 43.2 mm), this motor offers precise control over motion by dividing a full rotation into a large number of steps, typically 200 steps per revolution (1.8 degrees per step). Despite its small size, the Nema 17 stepper motor can deliver high torque, making it suitable for applications requiring accurate positioning and control. Its compatibility with various driver systems and controllers enhances its versatility, making it a popular choice among makers and engineers.

Figure 3. 11 Nema 17 Stepper motor

3.12 DC motors (TMSN5-6F):

The TMSN5-6F is a type of DC motor known for its compact size and high efficiency. It operates at a voltage range of 3V to 6V DC, making it suitable for a variety of applications. This motor offers a high speed and torque output, making it ideal for tasks requiring both power and precision. Its compact size makes it suitable for use in robotics, small appliances, and handheld devices where space is limited. The TMSN5-6F DC motor is designed to be highly efficient, helping to conserve battery power and reduce heat generation during operation. Overall, it is a reliable and versatile motor suitable for a wide range of applications.

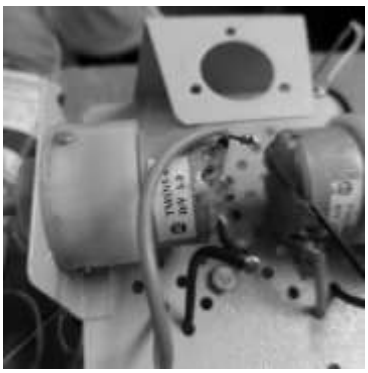


Figure 3. 12 DC motors (TMSN5-6F)

3.13 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software application that provides a platform for programming Arduino microcontroller boards. It offers a simple and user-friendly interface for writing, compiling, and uploading code to Arduino boards, making it accessible for beginners and advanced users alike. The IDE includes a text editor with features like syntax highlighting and auto-completion, as well as a serial monitor for debugging and communication with the Arduino board. Overall, the Arduino IDE is an essential tool for developing projects with Arduino boards, offering a seamless experience for programming and testing various applications.

CHAPTER 4: METHODOLOGY

4.1 System Block Diagram

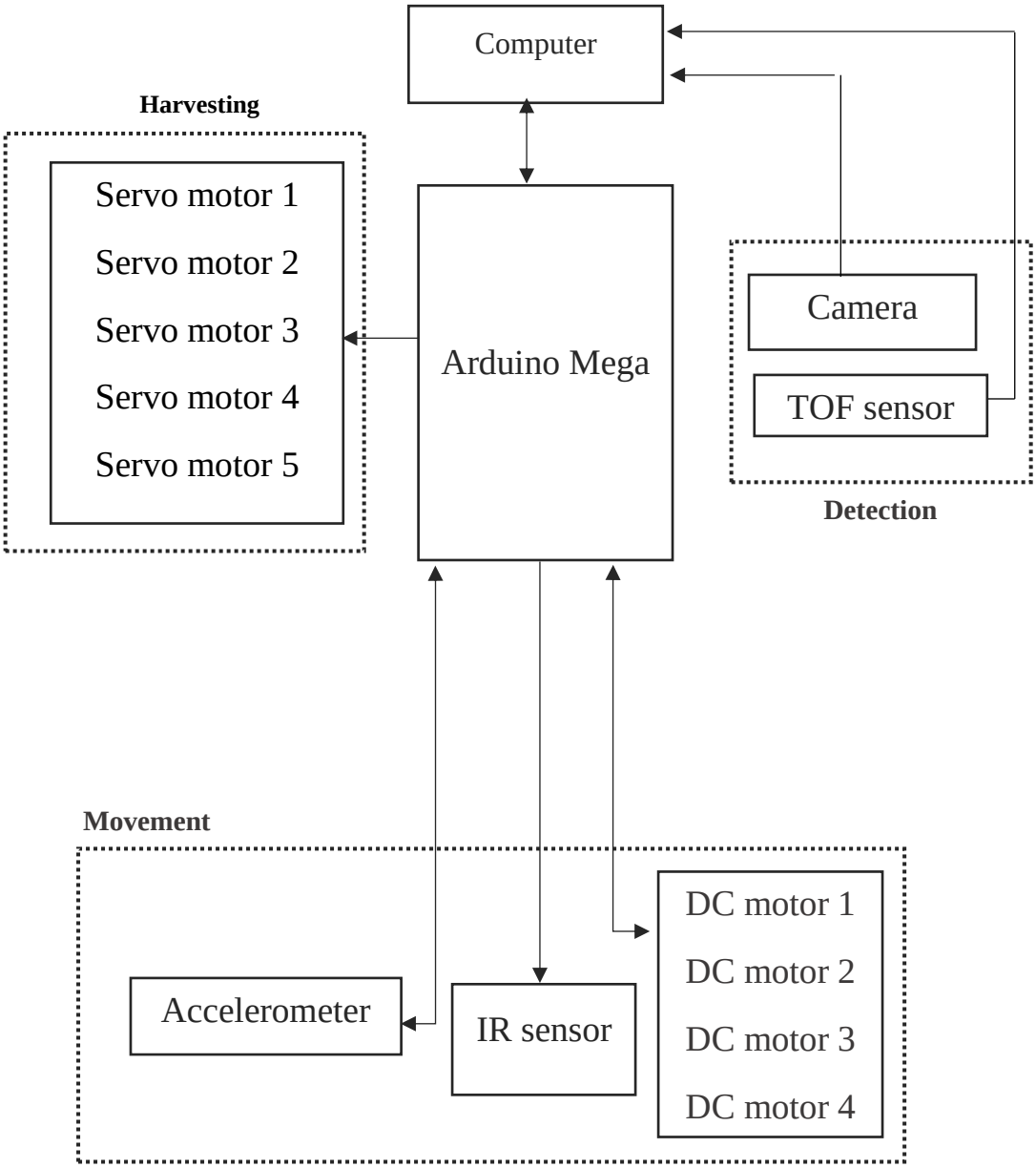


Figure 4. 1 System Block Diagram

The final location; the tomato is hanging in midair. Our four-wheeled vehicle has a camera that is adjusted on a surface to detect tomatoes. After that, the car follows the predetermined route until it finds a tomato. Following detection, the tomato's position is determined using the time of flight sensor and Yolov4 TINY. Yolov4 TINY provides the x and y coordinates, whereas the latter provides the z (depth). Inverse kinematics is used to move the arm; the arm's end-effector is positioned so that it reaches the tomato that has been detected. The ripe tomato is grab with the help of a stepper motor at the end effector.

4.2 Algorithm

4.2.1 Algorithm for Movement System (line following)

Step 1. Initialize:

- Set up sensor pins and motor controllers.
- Set the motor speed.
- Set the robot in the initial state.

Step 2. Read Sensor Values:

- Read values from left and right sensors.

Step 3. Detect Starting Line:

- Check if both left and right sensors detect a black line.
- If true, continue to the next step.
- If false, go back to step 2.
- Move Forward on Starting Line:

Step 4. Move Forward on Starting Line:

- Activate motors to move the robot forward.
- Continue moving until a change in sensor readings is detected.

Step 5. Follow Line Logic:

- Once the robot leaves the starting line, transition to regular line-following logic.
- Implement line-following behavior based on sensor readings:
- If both sensors on the line, move forward.
- If left sensor off the line, turn left.
- If right sensor off the line, turn right.
- Adjust motor speeds and directions based on the requirements.

Step 6. Check for Junction or Turn:

- Periodically check for signs of junctions or turns using additional sensors or algorithms.
- If a junction or turn is detected, transition to the appropriate behavior or decision-making logic.

Step 7. Repeat:

- Go back to step 2 and continue the loop.

4.2.2 Algorithm for YOLOv4 for object detection:

Step 1. Dataset Preparation:

- Collect a dataset of tomato images and annotate them with bounding boxes.
- Ensure annotations are in the YOLO format (text files containing class index and bounding box coordinates).

Step 2. Download YOLO Model:

- Download the pre-trained weights for YOLOv3 or YOLOv4 from the official YOLO website or GitHub.

Step 3. Configuration:

- Modify the YOLO configuration file (e.g., yolov3.cfg or yolov4.cfg) to set the number of classes to 1 (for tomatoes).

Step 4. Training:

- Use the YOLO model's architecture and load the pre-trained weights.
- Replace the output layer with a new layer for one class.
- Fine-tune the model on the tomato dataset.

Step 5. Inference:

- Load the trained model.
- Input an image to the model for inference.

Step 6. Post-processing:

- Filter out predictions with low confidence scores (e.g., below 0.5).
- Extract bounding box coordinates for the remaining predictions.

Step 7. Visualization:

- Draw bounding boxes on the original image to visualize tomato detections.

4.2.3 Algorithm for Kinematics (6DOF Robot Arm):

Step 1: Measurement of links and joints

Step 2: Transformation matrix for each joint ${}^{1}_{0}T$, ${}^{2}_{1}T$, ${}^{3}_{2}T$, ${}^{4}_{3}T$, ${}^{5}_{4}T$ and ${}^{6}_{5}T$

Step 3: Final transformation matrix ${}^{6}_{0}T$, ${}^{3}_{0}T$, ${}^{6}_{3}T$

Step 4: Form matrices R and P

Step 5: Calculation of Oc (x_c, y_c, z_c)

Step 6: Calculate ${}^{3}_{0}R$

Step 7: Calculate θ_1, θ_2, θ_3

Step 8: Calculate ${}^{6}_{3}R$

Step 9: Calculate θ_1, θ_2, θ_3

4.3 Flowchart

4.3.1 System General Flowchart

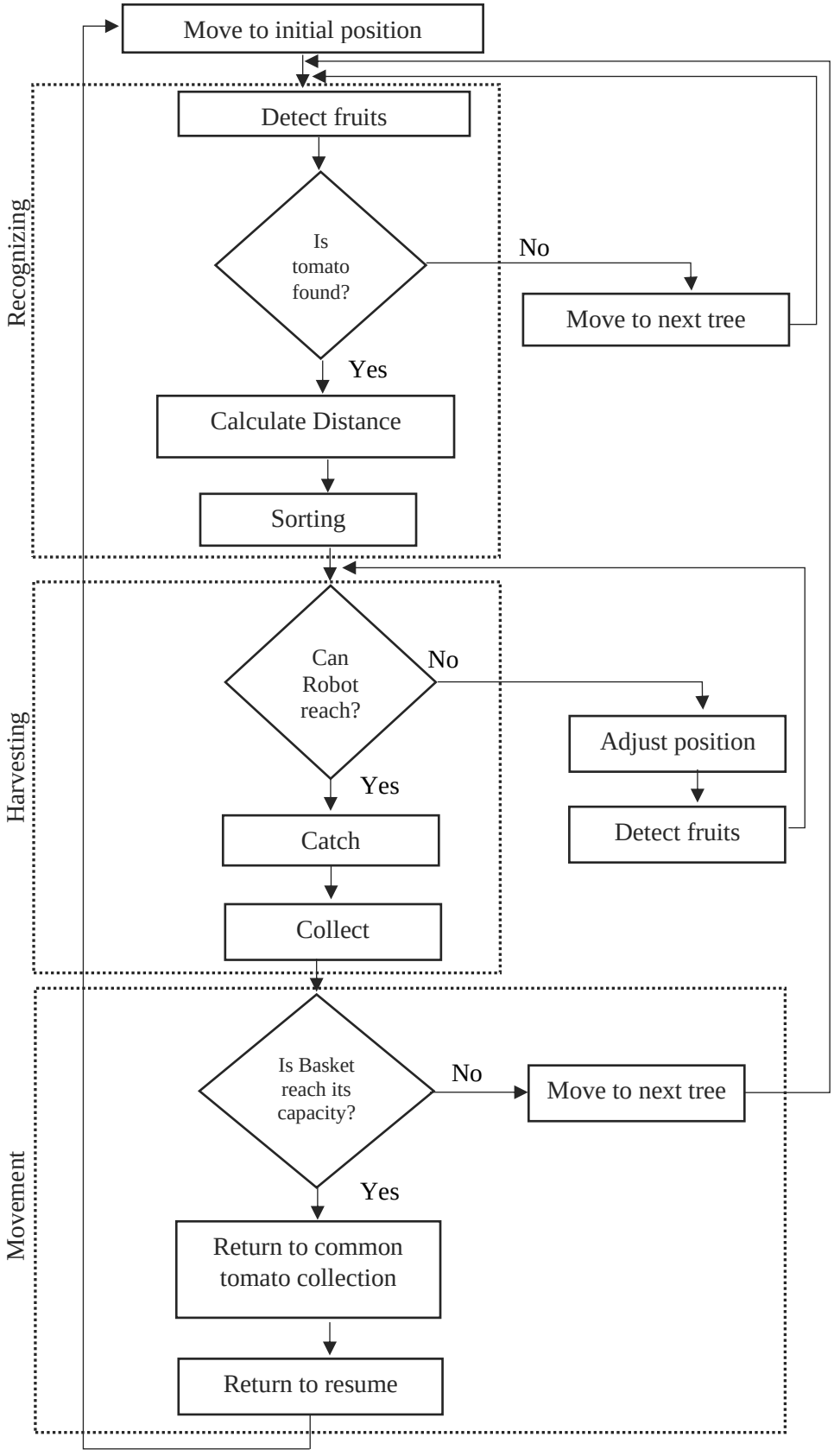


Figure 4. 2 System General Flowchart

4.3.2 Movement Flowchart (Line follower)

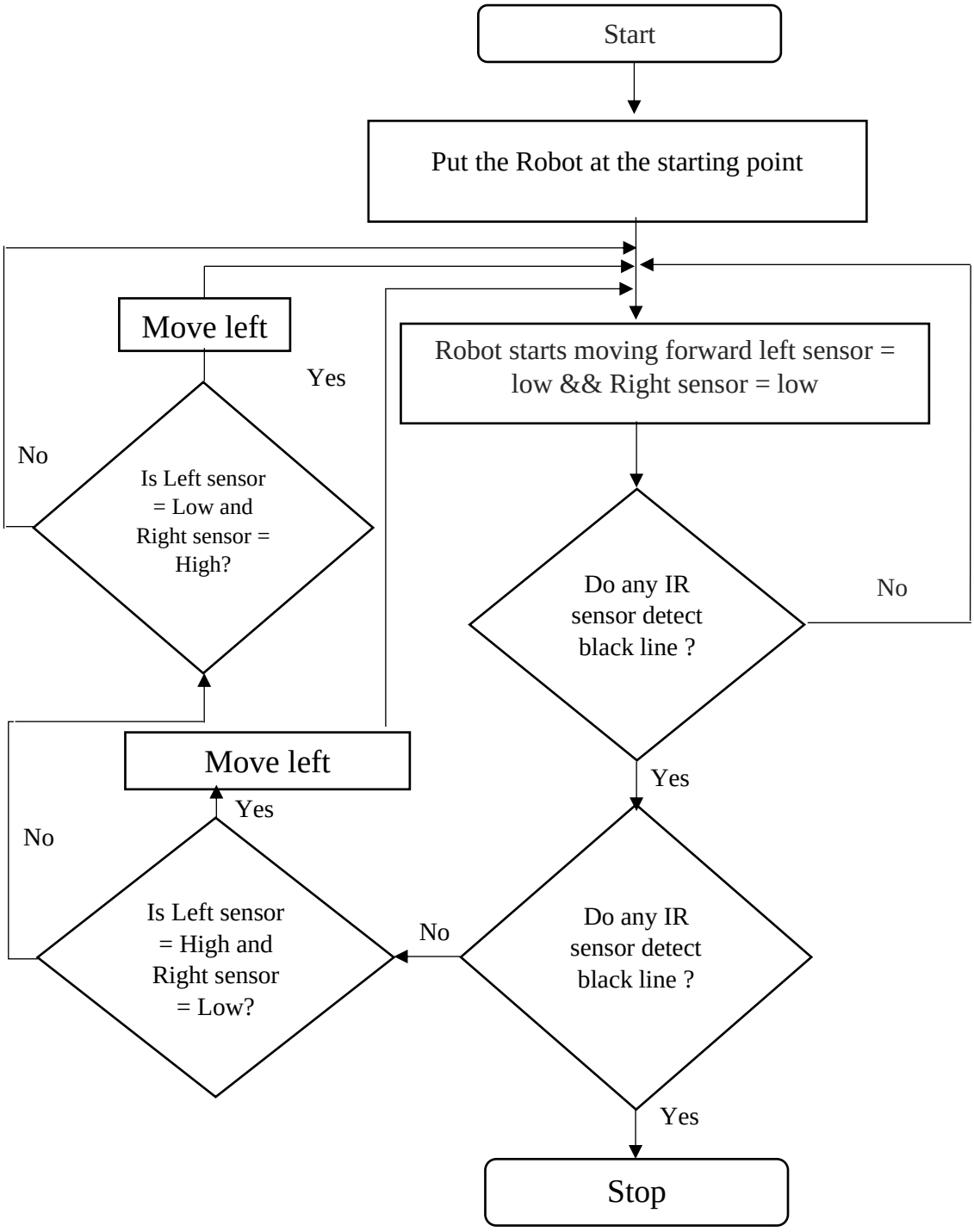


Figure 4. 3 Movement Flowchart (Line follower)

4.3.3 Robotic Arm Flowchart

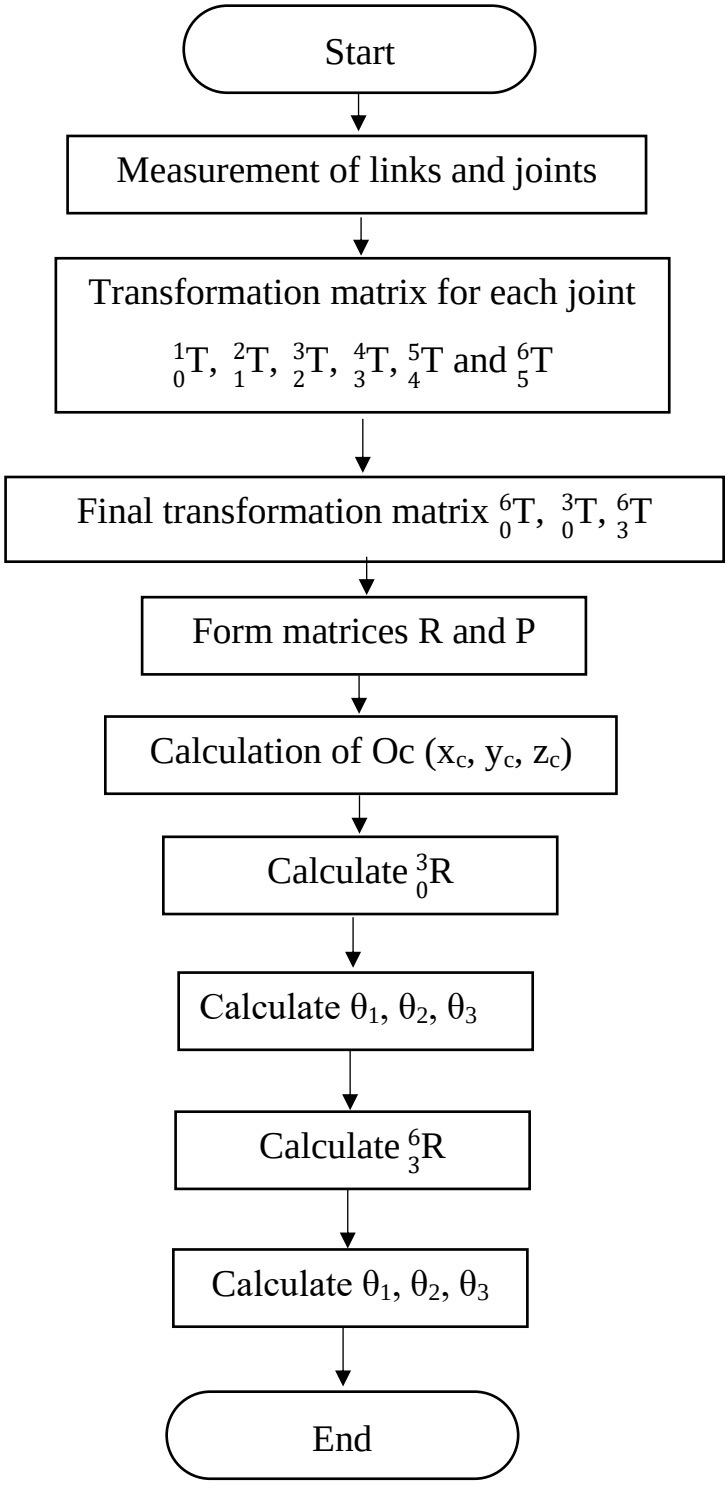


Figure 4. 4 Robotic Arm Flowchart

4.3.4 Model Training Flowchart

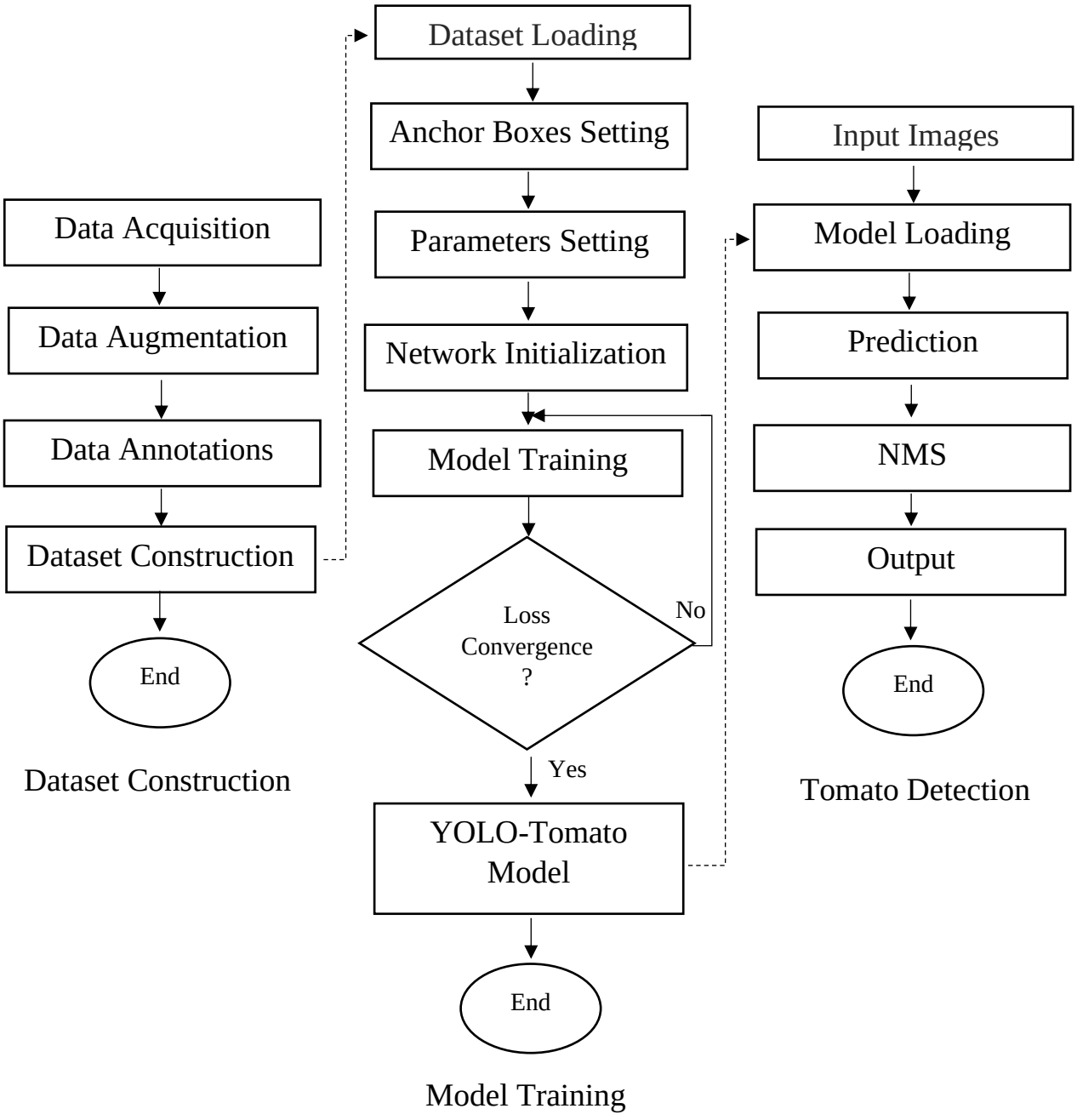


Figure 4. 5 Model Training Flowchart

CHAPTER 5: EPILOGUE

5.1 Results

We have divided the results into two major part Hardware Integration Part and Software work part details are as below:

5.1.1 Hardware Integration:

In the hardware integration phase of our tomato harvesting system, significant strides have been made with the successful assembly of a line-following robot and an integrated robotic arm designed for tomato picking. This key development represents a pivotal achievement in our project, providing the essential mobility and dexterity needed for effective harvesting. The remaining hardware components, a webcam for real-time visual data and a VL53L0X Time of Flight (ToF) Distance Sensor for precision in proximity measurements, are poised for seamless integration. The robotic arm, attached at the forefront of the line-following robot, stands ready to delicately harvest tomatoes. As we approach the final stages of hardware integration, the collective system, comprising the line-following robot, robotic arm, webcam, and distance sensor, promises a comprehensive solution for automated and optimized tomato harvesting in agricultural settings.

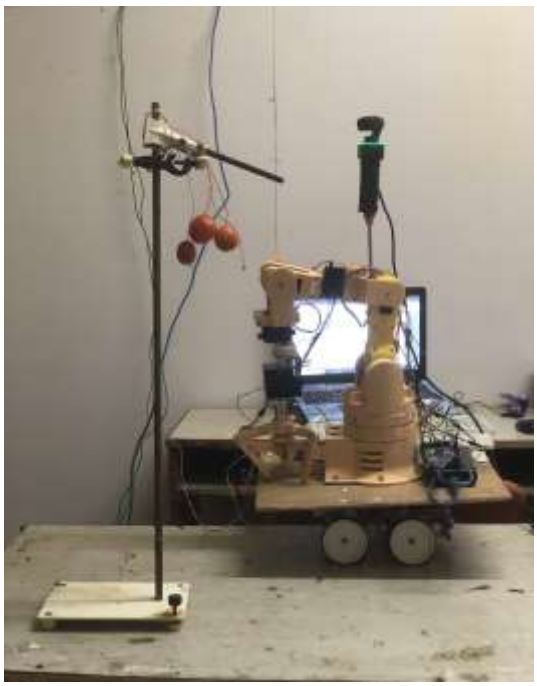


Figure 5. 2 Overall System Image

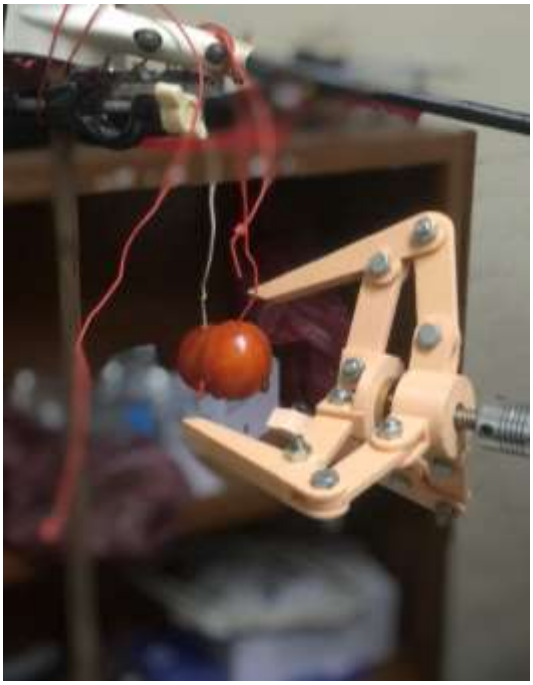


Figure 5. 1 Gripper trying to harvest tomato

5.1.2 Software works:

5.1.2.1 Dataset Creation:

In our dataset it consists of 1878 images with 5634 annotated tomato fruit samples of unripe, semi-ripe and fully-ripe classes.

Sample images with tomato fruit annotations are shown below:

5.1.2.2 Data organization:

The dataset was split into train and test set according to 80%/20% train-test split ratio.

	Train	Test
Images	1502	376
Annotated Boxes	4507	1127

Table 1 Data organization

5.1.2.3 Data format:

Statistics and data analysis:

Tomato classes:

The table below shows the number of annotated objects for each class of the dataset.

Unripe	Semi-Ripe	Fully-Ripe
3710	920	1004

Table 2 Data format

Additionally, the following figure illustrates the relative appearance frequencies of those three classes of the dataset. The classes of the dataset are clearly not balanced, however their relative proportion is in line with the actual appearance frequency of each class in a realistic scenario.

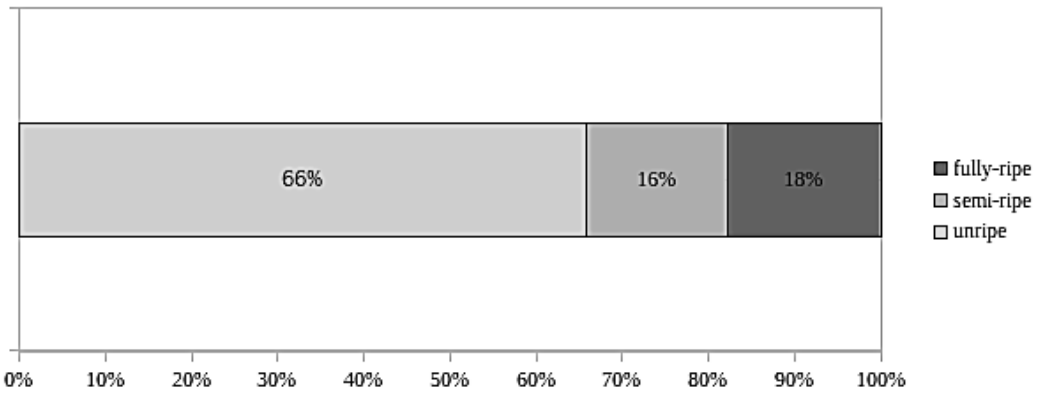


Figure 5. 3 Percentage of Fully-ripe, Semi-ripe and Unripe tomatoes

Size distribution of bounding boxes:

The percentile relative size of each bounding box is calculated, which indicates the proportion of the diagonal length of each box over the diagonal length of the image. In the image below, the histogram of the percentile relative size distribution of the bounding boxes is presented. Most of the bounding boxes have a size of 3% to 15% relative to the image size.

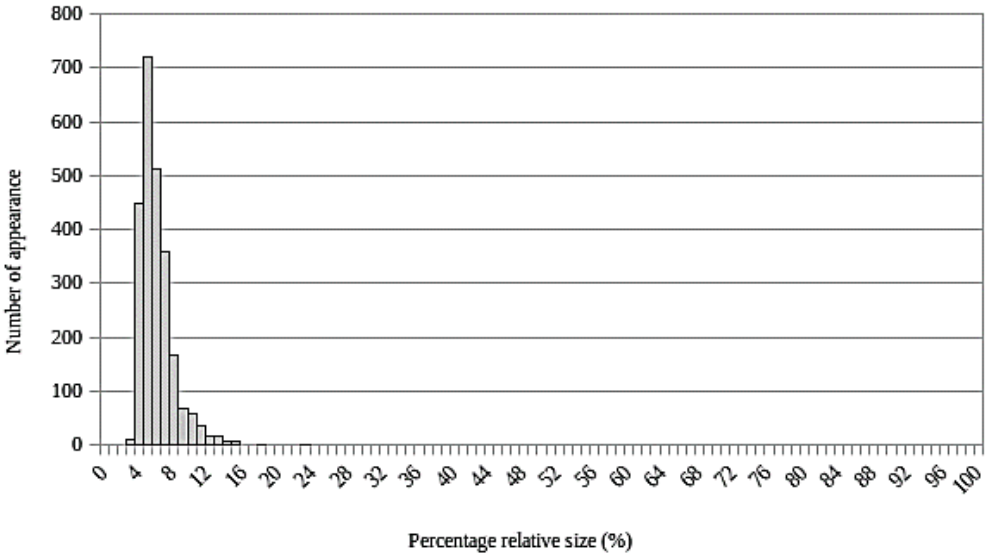


Figure 5. 4 Size distribution of bounding boxes

Number of labelled instances per image:

Only 1% of images have one category per image and 11% of images include 8 instances, while the maximum number of instances per image, which is 20, is found only in 0.72% of the images. The dataset has an average of 8.7 instances per image. The image displays the histogram of the number of annotation instances per image.

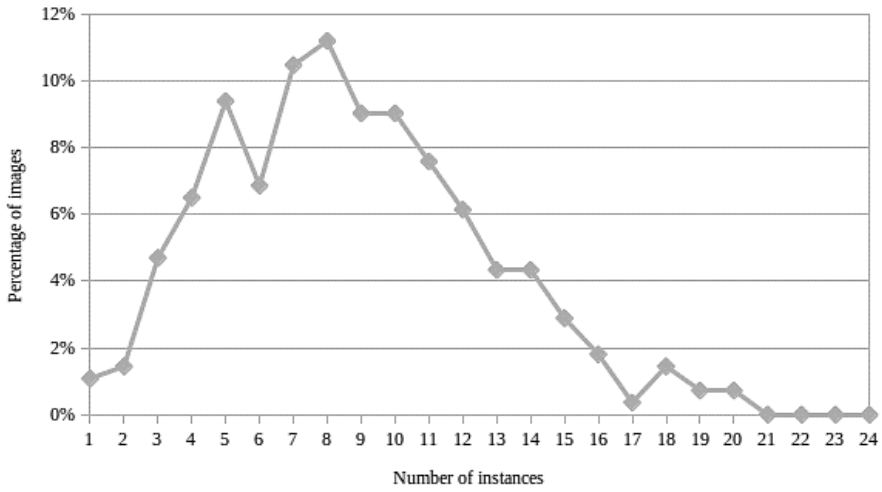


Figure 5. 5 Number of labelled instances per image

Number of categories in images:

As the next figure shows, more than 50% of the images contain objects of all 3 categories, while less than 8% of the images have objects of a single category.

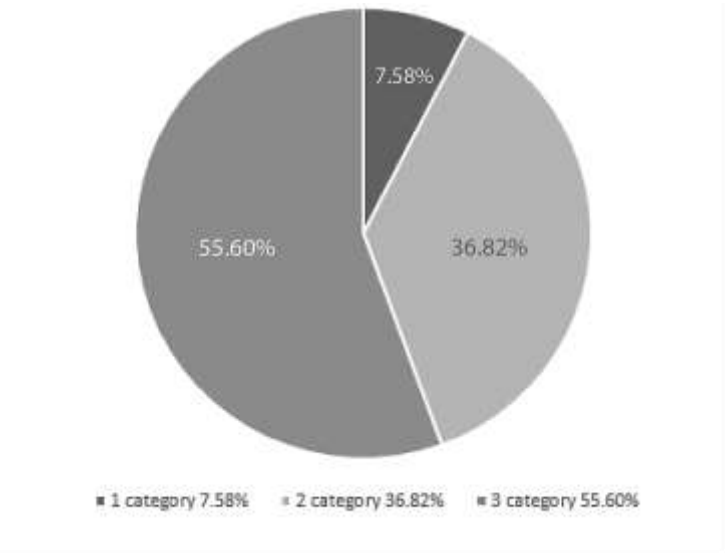


Figure 5. 6 Pie chart of number of categories in images

Visualize:

Dataset:

IMG20211226160636_00_jpg.rf.7adde6409176dbb1a0c2c234a1bd8761.txt:

2 0.32572115384615385 0.09254807692307693 0.12379807692307693 0.18509615384615385

1 0.5036057692307693 0.39783653846153844 0.2439903846153846 0.47716346153846156

IMG20211226160654_00_jpg.rf.0c58ddbff8132f44be41c90ce692c69d.txt:

1 0.5264423076923077 0.6706730769230769 0.1141826923076923 0.3076923076923077

IMG20211226160630_00_jpg.rf.68dbd7973fd19aaceb4d6cdb4bd1af49.txt:

1 0.4567307692307692 0.46634615384615385 0.12379807692307693 0.26322115384615385

2 0.6021634615384616 0.41947115384615385 0.10576923076923077 0.234375

IMG20211226160630_00_jpg.rf.2d53183c5ed429743d71314d173309e8.txt:

1 0.4639423076923077 0.4050480769230769 0.14543269230769232 0.3088942307692308

2 0.6346153846153846 0.34975961538461536 0.125 0.27524038461538464

Figure 5. 7 Image dataset

Figure 5. 8 Testing digital tomato

Hyper parameters:

Batch Size:

Batch size during training: batch=64

Batch = 1 on test

Grid Size:

From the last layer(s) of the network. In YOLOv4 Tiny, we have two [yolo] layers, each associated with a specific grid size.

For the first [yolo] layer:

- Anchor Mask: mask = 3,4,5

- Anchors: anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319

This layer predicts objects on a grid with a certain size, determined by the anchors.

For the second [yolo] layer:

- Anchor Mask: mask = 0,1,2
- Anchors: anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319

Similar to the first layer, it predicts objects on another grid with the same size.

Training Epochs and Learning Rate:

- Max Batches: max_batches = 6000
- Learning Rate: learning_rate=0.00261
- Burn-in Period: burn_in=1000

Policy and Scales for learning rate adjustment: policy=steps, steps=4800,5400, scales=.1,.1

Input Image Size:

- Width: width=416
- Height: height=416

Epoch's value:

To calculate the number of epochs for training with our batch size and dataset size, we need to first determine the total number of iterations (batches) needed to cover our entire dataset. Then, we can use this information to find the number of epochs.

First, let's calculate the total number of batches needed to cover our dataset:

Total number of samples in our dataset: 1878

Batch size during training: 64

Total number of batches = (Total number of samples)/(Batch size)

Total number of batches = 1878/64

Total number of batches ≈ 29.34375

We round up to ensure we cover the entire dataset, so we need 30 batches.

Next, we calculate the number of epochs:

Max batches: 6000

Epochs = (Max Batches)/(Total number of Batches)

Epochs = 6000/30

Epochs = 200

So, we trained for approximately 200 epochs to cover our dataset of 1878 images with a batch size of 64 and a maximum of 6000 batches.

Training results:

Loss chart during training:

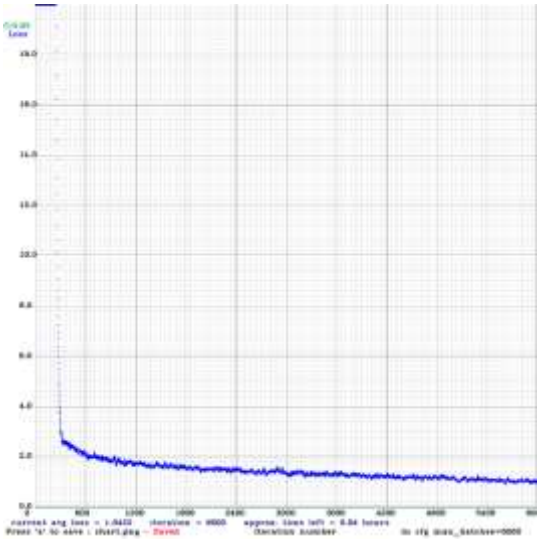


Figure 5. 9 Loss convergence chart during training

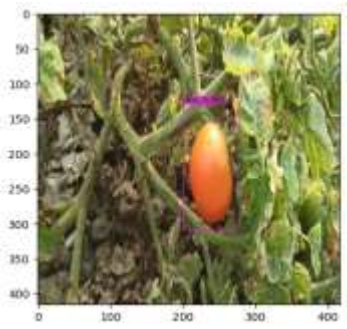
Performance:

mAP

- 3000_weights: 98.03%
- Final_weights:98.52%

```
Loading weights from /mydrive/yolov4-tiny/training/yolov4-tiny-custom_3000.weights...
 seen 64, trained: 192 K-images (3 Kilo-batches_64)
Done! Loaded 38 layers from weights-file

 calculation mAP (mean average precision)...
 Detection layer: 30 - type = 28
 Detection layer: 37 - type = 28
100
 detections_count = 4740, unique_truth_count = 650
class_id = 0, name = ripe, ap = 100.00%          (TP = 27, FP = 0)
class_id = 1, name = semiripe, ap = 99.05%       (TP = 37, FP = 7)
class_id = 2, name = unripe, ap = 96.09%         (TP = 563, FP = 146)

 for conf_thresh = 0.25, precision = 0.80, recall = 0.96, F1-score = 0.88
 for conf_thresh = 0.25, TP = 627, FP = 153, FN = 23, average IoU = 62.77 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.983804, or 98.38 %
Total Detection Time: 220 Seconds

Set -points flag:
 `-points 101` for MS COCO
 `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
 `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset


   calculation mAP (mean average precision)...
   Detection layer: 30 - type = 28
   Detection layer: 37 - type = 28
  100
   detections_count = 3009, unique_truth_count = 709
  class_id = 0, name = ripe, ap = 100.00%          (TP = 13, FP = 2)
  class_id = 1, name = semiripe, ap = 99.62%       (TP = 47, FP = 3)
  class_id = 2, name = unripe, ap = 95.95%         (TP = 617, FP = 111)

   for conf_thresh = 0.25, precision = 0.85, recall = 0.95, F1-score = 0.90
   for conf_thresh = 0.25, TP = 677, FP = 116, FN = 32, average IoU = 68.42 %

   IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
   mean average precision (mAP@0.50) = 0.985210, or 98.52 %
  Total Detection Time: 1 Seconds

  Set -points flag:
   `-points 101` for MS COCO
   `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
   `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Detection and its bounding value:



Test image Output image

Figure 5. 10 Tomato Detection

{[
"frame_id":1,
"filename":"/mydrive/tomato_test_image/test.jpg",
"objects": [
{"class_id":0, "name":"ripe", "relative_coordinates":{"center_x":0.574722, "center_y":0.527996, "width":0.182953, "height":0.433232}, "confidence":0.984806}] }

5.2 Conclusion

In conclusion, our project has successfully integrated a line-following robot with a specialized robotic arm for efficient tomato picking, employing YOLOv4 tiny for precise tomato detection across different ripeness stages and incorporating a Time of Flight (TOF) sensor for real-time 3D coordinate mapping. Through in-depth exploration of the kinematics of a 6-DOF robot arm, we've gained insights into motion planning, control, and optimized trajectory planning, addressing critical factors such as workspace and joint limitations. By contributing to the advancement of robotics and automation, our project not only offers practical solutions but also holds significant potential for applications in automated harvesting and agricultural quality control.

5.3 Future Enhancement

Vision System Enhancement: Elevate the robot's vision system to accurately detect ripe/unripe tomatoes and identify defects, employing advanced image processing techniques and potentially integrating machine learning algorithms for heightened precision.

Deep Reinforcement Learning (DRL): Embed DRL methodologies into the robot arm to amplify its capabilities in grasping, navigation, and tomato detection. This empowers the arm to adjust actions based on environmental cues, fostering more efficient harvesting practices.

Scaling: Opt for industrial-grade hardware components (motors, grippers, etc.) to render the project suitable for industry deployment. Leverage superior software and high-memory GPU computers to expedite computations, thus augmenting overall efficiency.

Diversification: Expand automated harvesting functionalities beyond tomatoes by detecting diverse plant components like fruits, leaves, and more, enabling deployment across various agricultural domains.

Tailored Algorithms: Customize the line-following algorithm to accommodate specific tomato farm layouts, ensuring precise navigation and targeted tomato production zones. Adapt alternative algorithms for farms with different configurations to optimize performance.

CHAPTER 6: BIBLIOGRAPHY

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

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

[3] M. O. Lawa, "Tomato detection based," Scientific Reports, 2021.

[4] G. L. J. S. S. C. K. a. J. H. K. hilippe Lyonel Touko Mbouembe, "An efficient tomato-detection method based on improved YOLOv4-tiny model in complex environment," *Frontiers*, 2023.

[5] C.-Y. W. a. H.-Y. M. L. Alexey Bochkovskiy, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[6] ,. P. W. W. L. J. L. ,. R. Y. a. Zhaohui Zheng, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression Dongwei Ren," 2019.

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

[8] F. z. Chen Huangfei, "Design and research on the End Actuator of Tomato Picking," 2019.

[9] J. X. Tao Zhu, "Chassis Design of Tomato Picking Robot in the Greenhouse," *ICEEIE*, 2021.

[10] M. R. U. I. ,. H. K. amshed Iqbal, "Modeling and analysis of a 6 DOF robotic arm manipulato," *Canadian Journal on Electrical and Electronics Engineering*, vol. 3, 2012.

[11] N. V. T. D. T. P. T. a. T. V. M. Tran Thanh Tung, "Development of a prototype 6 degree of freedom robot arm," *Result in Engineering*, 2023.

[12] S. Bachche, "Deliberation on Design Strategies of Automatic Harvesting," *robotics*, 2015.

[13] S. A. a. P. Webb, "Kinematics Analysis of 6-DoF Articulated Robot with," *HIndawi*, 2021.

[14] S. Pathak, "LINE FOLLOWING ROBOT," *IJIREEICE* , 2021.

[15] A. E. A. B. E. A. hmed Bendimrad, "Design and implementation of line follower and obstacle," *International Journal of Power Electronics and Drive System (IJPEDS)*, 2020.