Automatic Robotic Arm For Chilli Plucker

Project Report

Submitted in partial fulfillment of the requirements for

the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

by

JAZEEL ANWAR

MAC19CS030

MANU P S

MAC19CS037

VIDYA V

MAC19CS059

Department of Computer Science & Engineering

Mar Athanasius College of Engineering

Kothamangalam

DECEMBER 2022

Automatic Robotic Arm For Chilli Plucker

Project Report

Submitted in partial fulfillment of the requirements for

the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

by

JAZEEL ANWAR

MAC19CS030

MANU P S

MAC19CS037

VIDYA V

MAC19CS059

Department of Computer Science & Engineering

Mar Athanasius College of Engineering

Kothamangalam

DECEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAR ATHANASIUS COLLEGE OF ENGINEERING

KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Automatic Robotic Arm For Chilli Plucker** *submitted by* **Mr. Jazeel Anwar, Reg. No. MAC19CS030, Mr. Manu P S, Reg. No. MAC19CS037 and Ms. Vidya V, Reg. No. MAC19CS059**, *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer Science and Engineering from APJ Abdul Kalam Technological University for December 2022 is a bonafide record of the seminar carried out by them under our supervision and guidance.*

....	....
Prof. Dr. Elizabeth Isaac	**Prof. Joby George**
Project Coordinator & Project Guide	***Head of the Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

We express our sincere gratitude to Dr. Bos Mathew Jos, Principal, and Prof. Joby George, Head of the Department for providing the necessary facilities and their encouragement and support.

We owe special thanks to our project guide and project coordinator Dr. Elizabeth Isaac for her corrections, suggestions, and sincere efforts to co-ordinate the project under a tight schedule.

We express our sincere thanks to staff members in the Department of Computer Science and Engineering who have made sincere efforts in guiding and correcting me in conducting this project.

Finally, we would like to thank god Almighty for helping us successfully complete our project. Also, we would like to acknowledge the sincere efforts, comments, criticisms, co-operation, and tremendous support given to me by my dear friends during the preparation of the project and also during the presentation without which this work would have been all the more difficult to accomplish.

ABSTRACT

Chilli is regarded as India's most valuable crop. Green chilli (Capsicum annuum) crops must be completely hand-harvested, which calls for a sizable workforce during a condensed window of time. Current harvesting robots have limited performance, due to the unstructured and dynamic nature of the target crops (variable shape, colour, size, texture, location) and their environment (changing illumination due to sun direction and clouds, obstructions such as branches and leaves). The main goal is to find a way to automatically harvest chilli in real-time with minimum human effort. The chilli on the plant can be recognized using the object detection algorithm and is harvested using the robotic arm. Object detection is done using YOLO (You Only Look Once) algorithm with a custom dataset. The 2D (x,y) coordinates can be identified from the object detection position and the 3D depth of the fruit is located using Intel RealSense Depth Camera D435i which can calculate the depth of the object from the camera. The robotic arm is designed in a way to take pictures from different angles to get an image of the fruit under the leaves. The robotic arm moves to the position of the chilli calculated and pluck it using the designed cutter blades.

Table of Contents

List of Figures

List of Abbreviations

DoF	Degrees of Freedom
YOLO	You Only Look Once
AI	Artificial Intelligence
DNN	Deep Neural Networks
CNN	Convolutional Neural Networks
1D-CNNs	One Dimensional Convolutional Neural Networks
RNNs	Recurrent Neural Network
GPU	Graphics Processing Unit
SoC	System on a Chip
CUDA	Compute Unified Device Architecture
OpenCV	Open Source Computer Vision Library
IOU	Intersection Over Union
API	Application Programming Interface
GPIO	General Purpose Input/Output
TP	True Positives
TN	True Negatives

FP	False Positives
FN	False Negatives
mAP	Mean Average Precision
IoT	Internet of Things

Chapter 1
Introduction

Chilli is a high-value vegetable item that is significant for both economic and consumer needs. Chilli consumption rises year after year as the population and income rise. Chilli output must be boosted in order to meet demand. Chilli is a popular vegetable spice that is cultivated all throughout the world, except in colder climates. It is also known as red pepper or hot pepper, and it is a well-known commercial crop that is used as a condiment, culinary additive, or vegetable. Chilli is mostly used in cooking to offer taste, colour, vitamin, and pungency. Chilli is almost always present in the kitchen. Vegetables, spices, condiments, sauces, and pickles are all cultivated in different types. Chilli is one of the most valuable crops, and it is planted all around the country. Tamil Nadu is one of the top chilli-producing states in India. Tamil Nadu has the most chilli area, output, and productivity. As a result, there is a lot of room to expand the chilli area.

Furthermore, the production of chilli decreases from year to year. One of the causes for the low output might be a technology gap, and chilli growers are also encountering production and marketing challenges. In every location, the green chilli crop is totally hand picked, necessitating a high number of agricultural employees during a very short harvest window.

The primary causes for diminished chile pepper output are high labour costs and restricted labour availability. According to research, automation or the use of robotic harvesting can lower costs by up to 10% [2]. Robotic systems have been deployed in farming systems to meet expanding worldwide agricultural concerns. Harvesting has been greatly impacted by a continually declining and increasingly costly human force; as a result, major research has been conducted on the design and development of harvesting field robots. Robotic arms (manipulators) are created or chosen from commercially available platforms with degrees of freedom (DoF) ranging from two to seven, with the majority having three. However, no previous study on harvesting robots has provided systematic analysis and design for the selection of the number of DoF.

Robotic end-effectors, on the other hand, are often custom-made and tailored for specific crop harvesting. Some of these robotic end-effectors use suction cups or a mix of suction cups and robotic fingers to grip and hold the fruit. Detaching the fruits at the stem is accomplished by heat cutting, scissor cutting, or manual twisting and tugging.

1.1 Chilli Plucker

A chilli plucker is essential equipment for farmers and gardeners that plant chiles. It enables growers to pick chilies from their plants fast and effortlessly without inflicting any damage to the fruit. This can significantly enhance crop productivity and help farmers raise their income. It is also a safe tool for youngsters to use since it reduces the chance of unintentional cuts or scratches that come with hand harvesting.

A chilli plucker is a machine that assists farmers in swiftly picking chiles. It's like a robot hand picking chilies from the vine and depositing them in a box. The equipment saves the farmer time since it can select a large number of chilies in a short period of time. It also reduces the amount of work the farmer has to do and prevents them from getting too tired.

1.2 Project Outline

The project intends to develop an automatic robotic arm for real-time detection of pepper using Machine learning and plucking using embedded systems. YOLOv5 is used for custom dataset training and object detection. Utilizing an object identification algorithm, the robotic arm can identify the pepper on the plant and pluck it. YOLO technique is used for object detection along with a unique dataset. Using Intel RealSense Depth Camera D435i, which can determine an object's depth from a camera, it is possible to identify the pepper's 3D depth and determine its 2D (x,y) coordinates. In order to capture a photo of the pepper hidden behind the leaves, the robotic arm is built to be able to snap.

Chapter 2
Background

In this chapter we discuss the necessary technologies and features that we require to build the project.

2.1 Machine Learning

Machine Learning is a subfield of Artificial Intelligence that deals with the design and development of algorithms that enable computers to learn from data and make predictions or decisions without explicit instructions. These algorithms are trained using a dataset, which is a collection of labeled data. There are different types of Machine Learning such as supervised, unsupervised, and reinforcement learning [9]. Machine Learning is used in a wide range of applications, including natural language processing, computer vision, speech recognition, and self-driving cars. With increasing data and advancements in technology, Machine Learning is being applied to more complex problems.

2.2 Deep Learning

Deep Learning is a subfield of Machine Learning that focuses on neural networks, which are algorithms inspired by the human brain. These networks are composed of layers of interconnected artificial neurons that process and transmit information. They are trained using large amounts of labeled data and can automatically learn features and representations from the data. Deep Learning is well-suited for tasks such as image and speech recognition, natural language processing, and self-driving cars. It requires significant amounts of data and computational resources to be implemented [10].

2.2.1 Advantages of Deep Learning

- The deep learning architecture is flexible to be fitted to new troubles in the future.
- It has the proficiency to develop new features from the limited convenient training data sets.

2.2.2 Disadvantages of Deep Learning

- Due to complicated data models, it is very expensive to train.
- It is not easy to comprehend output based on mere learning and requires classifiers to do so.

2.2.3 Applications of Deep Learning

There are several applications of deep learning across industries:

- Self driving cars
- Virtual assistants
- Natural Language Processing
- Virtual Recognition

2.3 Convolutional Neural Networks

A convolutional neural network (CNN) is a type of deep learning neural network that is mainly used for image and video recognition tasks [11]. It uses small filters to scan over the entire input and extract features such as edges, corners, and textures. The output is passed through non-linear activation functions and pooling layers for down-sampling, reducing computational costs and making the network more robust [10]. Finally, the output is passed through one or more fully connected layers for classification or regression. CNNs have been very successful in image recognition tasks such as object detection, face recognition, and image segmentation and also used in other areas such as natural language processing and speech recognition.

2.3.1 One Dimensional Convolutional Neural Networks

One Dimensional Convolutional Neural Networks (1D-CNNs) are a variant of CNNs that are designed to process one-dimensional data such as time series, audio signals, and text. They are similar to traditional CNNs in that they use convolutional layers to extract features from the input data, but the convolutional filters in 1D-CNNs are 1-dimensional, rather than 2-dimensional as in traditional CNNs.

The convolutional layers in 1D-CNNs slide a filter along the input data, computing dot products between the filter values and the input values at each location. The output of the convolutional layers is passed through activation functions and pooling layers, similar to traditional CNNs [10].

1D-CNNs are commonly used in audio processing tasks such as speech recognition, music transcription, and audio event detection. They are also used in natural language processing tasks such as text classification, sentiment analysis, and named entity recognition.

1D-CNNs have been found to be quite effective in these tasks because they are able to automatically learn time-frequency representations of the input data, which capture the important temporal patterns in the data. They also have been found to be more computationally efficient compared to traditional recurrent neural network (RNNs) and are able to handle large dataset, thanks to the pooling layers.

Overall, 1D-CNNs are a powerful tool for processing sequential data and have found applications in a wide range of fields.

2.4 Python

Python is a high-level, open-source programming language that is widely used for web development, scientific computing, data analysis, artificial intelligence and more. It is known for its simple and easy-to-read syntax, making it a popular choice for beginners and experienced programmers alike. Python has a large and active community, which has led to the development of many powerful libraries and frameworks such as NumPy, Pandas, and TensorFlow. It also supports multiple programming paradigms such as object-oriented, imperative and functional programming. Python is available on multiple platforms such as Windows, Linux, and macOS and can be used to build a wide range of applications from web apps to desktop apps, games, and more.

2.5 YOLO

YOLO is a real-time object detection system that is designed to be fast and accurate. It is a single neural network that is trained to predict bounding boxes and class probabilities for objects in images [12]. Unlike other object detection systems, YOLO divides the image into a grid of cells and predicts bounding boxes and class probabilities for each cell. This allows YOLO to make predictions for objects of different scales and aspect ratios, which is a significant advantage over other methods. Additionally, YOLO uses anchor boxes to predict multiple bounding boxes for the same object, which improves the accuracy of the predictions. YOLO is implemented using the Darknet framework, and it can be run on a variety of platforms including CPU, GPU and mobile devices. It has become very popular in recent years and is widely used in various applications such as self-driving cars, security systems, and robotics.

2.5.1 Advantages of YOLO

- Speed: This algorithm improves the speed of detection because it can predict objects in real-time.
- High accuracy: YOLO is a predictive technique that provides accurate results with minimal background errors.
- Learning capabilities: The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

2.5.2 Application of YOLO

- Autonomous driving: YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people, and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car.
- Wildlife: This algorithm is used to detect various types of animals in forests. This type of detection is used by wildlife rangers and journalists to identify animals in videos (both recorded and real-time) and images. Some of the animals that can be detected include giraffes, elephants, and bears.
- Security: YOLO can also be used in security systems to enforce security in an area. Let's assume that people have been restricted from passing through a certain area for security reasons. If someone passes through the

restricted area, the YOLO algorithm will detect him/her, which will require the security personnel to take further action.

2.6 Pandas

Pandas is a python library that provides easy-to-use data structures and data analysis tools. It is widely used for data manipulation and cleaning, data analysis and exploration, and data visualization. The two primary data structures in Pandas are the Series (1-dimensional) and Data Frame (2-dimensional)[13]. These structures are similar to the data structures in R. They are designed to handle large amounts of data with high performance. The library also provides powerful tools for handling missing data, working with time series data, and reading and writing data in various formats. Pandas also provides easy-to-use functions for data transformation and aggregation, as well as for handling and manipulating large datasets. It's a powerful tool for data scientists and analysts and is widely used in data science and machine learning projects.

2.7 Numpy

NumPy is a Python library that provides support for large multi-dimensional arrays and matrices of numerical data, as well as a large collection of mathematical functions to operate on these arrays [14]. It is the fundamental package for scientific computing with Python and is widely used in data science and machine learning projects. NumPy provides an efficient multi-dimensional array object called ndarray, which can perform mathematical operations on large data sets. It also provides functions for performing linear algebra, Fourier analysis, and other mathematical operations on these arrays. NumPy is highly optimized for performance and is designed to work seamlessly with other libraries such as Pandas and SciPy. It also provides an interface to C and Fortran libraries, allowing for easy integration with other languages.

2.8 Keras

Keras is an open-source deep learning library written in Python. It is designed to make building and experimenting with deep neural networks as fast and easy as possible. Keras provides a simple and consistent interface to various neural network libraries such as TensorFlow and Theano, enabling users to quickly prototype and test different architectures. It supports a wide range of neural network layers, activation functions, and optimizers, as well as advanced features such as recurrent networks, convolutional networks, and skip connections. Keras also provides a range of tools for data preprocessing, training, and evaluation, making it a popular choice among researchers and practitioners. Additionally, it can run on CPU and GPU (Graphics Processing Unit), making it a flexible tool for various use cases.

2.8.1 Embedded C

Embedded C is a variant of the C programming language that is specifically designed for use in embedded systems. Embedded systems are small, specialized computer systems that are used to control or monitor other devices or equipment. Embedded C is similar to standard C but is optimized for use in the limited memory and processing power environments of embedded systems. It is a procedural programming language, which means that the program is a set of instructions that tell the computer what to do. It is widely used in microcontroller based applications, such as in automotive systems, industrial automation, and consumer electronics. It is also used to control a variety of electronic devices, including cell phones, televisions, and washing machines. It is a powerful and efficient language that is well suited to the needs of embedded systems.

2.9 Raspberry Pi

Raspberry Pi is a small, low-cost, single-board computer that is designed for educational and hobbyist use. It is developed and manufactured by the Raspberry Pi Foundation, a UK-based charity organization. The device is built around a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and 512MB of RAM. The Raspberry Pi runs on the Linux operating system and is compatible with a wide range of software and programming languages, including Python, Scratch, and C++ [15]. It has a wide range of uses, including as a desktop computer, media center, game console, and as a platform for DIY electronics projects. It is widely used in education, computer science, and hobbyist projects, as it is a cheap and powerful tool that can be used to learn about programming and computer hardware.

2.10 PyTorch

PyTorch is an open-source machine learning library based on the Torch library. It is primarily developed by Facebook's AI Research lab and is widely used in research and production. PyTorch provides a simple and easy-to-use interface for building, training, and deploying deep learning models. It also provides a dynamic computational graph, which allows for fast and efficient model building and experimentation. Pytorch also has a large and active community, which provides a wide range of pre-trained models and resources for building, training and deploying ML models. Additionally, PyTorch has support for CUDA(Compute Unified Device Architecture) and other GPU acceleration libraries, which allows for fast training of deep learning models on a GPU. It also has a built-in support for data parallelism, making it easy to scale-up the training process across multiple GPUs.

2.10.1 TorchScript

TorchScript is a subset of the Python programming language that can be converted to a static computation graph, allowing models to be run in a variety of environments, such as on mobile devices or in a web browser. It provides a way to convert PyTorch models to a format that can be run without a Python interpreter, enabling faster inference and easier deployment. It also allows for the creation of custom C++ and CUDA extensions and the use of low-level control flow operations. In short, TorchScript is a way to make PyTorch models more portable and efficient.

2.11 OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. It provides a common infrastructure for computer vision applications and accelerates the use of machine learning in commercial products [16]. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. It includes more than 2500 optimized algorithms for image and video analysis, making it a popular choice for a wide range of applications, such as image processing, object detection, facial recognition, and more. It is written in C++ and has interfaces for C++, Python, and Java.

Chapter 3
Related Works

Currently, there are some approaches and methodologies for harvesting green chillies but most of them are either time consuming or expensive. The conventional approaches for chilli harvesting demands a lot of human effort . Below are the detailed explanations of different methods and their drawbacks.

3.1 Conventional Methods

There are two primary methods for collecting chillies from plants. The first is manual labor, where each chilli is picked individually by hand. The second is to use tools such as scissors to harvest the chillies in bulk. Both methods are time-consuming and not highly efficient, but are commonly used by farmers for their chilli crop[5].

Figure 3.1: Traditional Hand Picking

3.2 Braccio Robot Arm

The Braccio robotic arm is an affordable, 5-axis robot arm designed for the mechanized picking of chilli peppers. It utilizes DC Servo Motors that are controlled through internal circuitry based on input signals sent by a microcontroller[8].The robotic arm is controlled by an Arduino Due Development Board. The joints of the robot arm are referred to as base, shoulder, elbow, wrist pitch, and wrist roll, in that order from the base to the end effector[6].

All joints on the robotic arm, except for the shoulder joint (2nd joint), have a range of motion from 0 to 180 degrees. The shoulder joint is limited to a range of 15 to 165 degrees due to design and structural limitations rather than the motors' capabilities. The robot arm has a maximum reach of 400 mm and can carry a payload of 150 g at a distance of 320 mm and 400 g at a minimum distance of 400 mm.

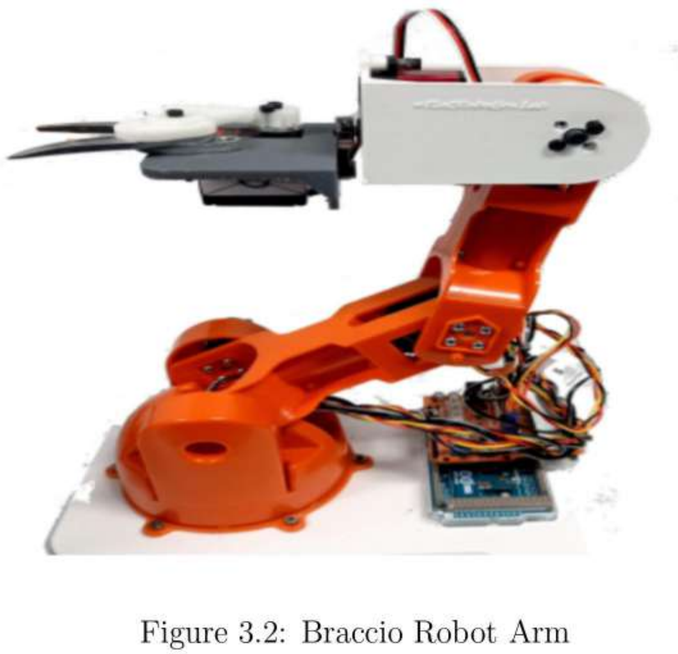

Figure 3.2: Braccio Robot Arm

3.3 An Innovative System for Plucking Mango Fruit

The design of the robot arm was focused on ensuring accurate cutting of the stem to minimize latex bleeding, protect the fruits from damage, and prevent them from falling. The cutting mechanism of the robot arm consisted of two circular saws that rotated in opposite directions towards the center, with a counter-sharp blade located at the center of the cutting discs[7].

The robot arm was tested on three types of mango trees: Keitt, Kent, and Tommy Atkins, which were low-stemmed trees with large fruit volume. The cross-section of the stem was also measured. The dimensions of the fruits were used to design the machine. The efficiency of cutting is low when overlapping occurs and may not detect all mangoes.

Figure 3.3: Image of the Manufactured Picker During Operation

3.4 Development of Sweet Pepper Harvesting Robot

The robotic system includes an industrial arm with six degrees of freedom, a custom-designed end effector, an RGB-D camera, a high-performance computer with a graphics processing unit, programmable logic controllers, other electronic equipment, and a small container to store the harvested fruits [4]. All of these components are mounted on a cart that can autonomously move on pipe rails and concrete floors in the end-user environment.

The average cycle time to harvest fruit was 24 seconds, with logistics taking up approximately 50% of this time (7.8 seconds for discharge of fruit and 4.7 seconds for platform movements). Laboratory experiments have shown that the cycle time can be reduced to 15 seconds by running the robot manipulator at a higher speed. The harvest success rates were 61% for the best-fit crop conditions and 18% in current crop conditions, highlighting the importance of finding the best-fit crop conditions and varieties for successful robotic harvesting. This sweeper robot is the first sweet pepper harvesting robot to demonstrate this level of performance in a commercial greenhouse. Its limitations include lower efficiency and the inability to detect all sweet peppers.

Figure 3.4: Sweet Pepper Harvesting Robot

Chapter 4

Design and Implementation

In the previous chapters we discussed the importance of chilli harvesting and its different methods. We also saw the advantages that machine learning and deep learning algorithms like YOLO can provide to accurate object detection.

In this chapter, we will discuss the architecture of the proposed product to speed up the real time harvest using robotic arm, and the functionalities implemented in phase 1 is also discussed below.

In the proposed system the chilli on the plant can be recognized using the object detection algorithm and is harvested using the robotic arm. The input images captured by a camera are processed using an object detection algorithm known as YOLOv5. YOLO, one of the most versatile and famous object detection algorithms, splits images into a grid system to find objects quickly and accurately. Each grid cell predicts all the bounding boxes and gives each of them a confidence score to determine the accuracy of each prediction. Data scientists and machine learning engineers always choose YOLO as their first choice for real-time object detection tasks. Improving productivity and reducing the amount of spoilt chilli from late harvesting will benefit chilli farmers.

A custom dataset is created by manual collection of images of chilli and labelling the chilli in the images. The custom dataset is used for training with weights of YOLOv5s and the set of new weights is used for detection.

The robotic arm moves around taking pictures of the plant. If the presence of the chilli in the image is detected 2D (x,y) coordinates can be identified from the object detection bounding box position and the 3D depth of the object is located using Intel RealSense Depth Camera D435i which can calculate the depth of the object from the camera. In case of detection of multiple chillies the chilli with the highest confidence is chosen for the plucking.

The robotic arm is designed in a way to take pictures from different angles to get an image of the fruit under the leaves. A mapping function is formulated to find the equivalent real-world 3D position of the calculated position from the image. The robotic arm moves to the position of the chilli calculated and plucks it using the designed cutter blades. The movement of the arm is controlled by microcontrollers programmed using embedded C.

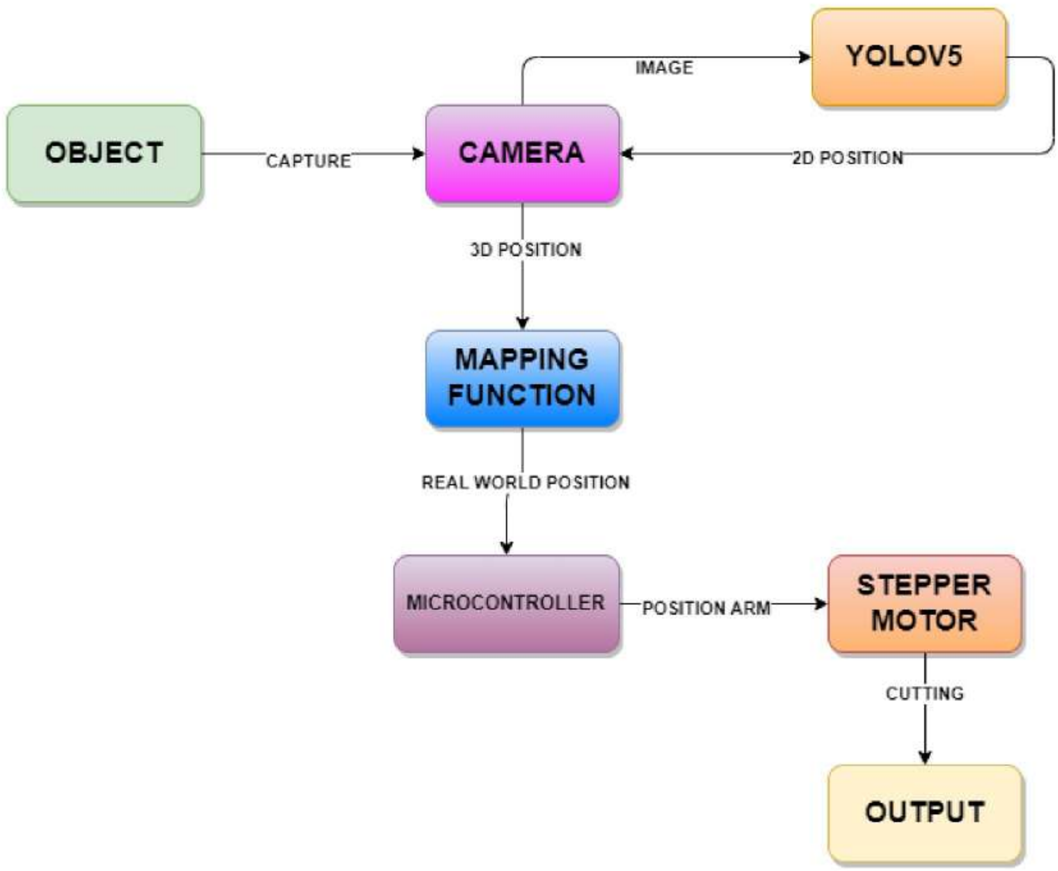

Figure 4.1: Workflow of the Proposed System

4.1 Understanding the Dataset and Creating a Custom Dataset

Phase 1 of the project included the collection of images of chilli on its plant from different angles and location. The images are made diverse so that it best represent chilli with all its features for identification.

4.1.1 Dataset

The project aimed to create a custom dataset for the training and validation of the YOLOv5 object detection algorithm for chilli plants. The dataset was created through manual collection of images of chilli on plants taken from different angles and from different plants. The goal was to gather as many diverse images as possible, so that the dataset best represents chilli with all its features for identification. Each image was scaled to a standard size and then each chilli in the image was labeled one after the other.

The dataset was then divided into training and validation sets. The dataset consists of folders for the images and their corresponding labels under training and validation. It was saved in a format compatible with the YOLO algorithm, so that it can be used to train the algorithm for detecting and identifying chilli plants in images.

4.2 Object Detection Using YOLOv5

YOLO is a family of single-stage deep learning-based object detectors. They are capable of more than real-time object detection with state-of-the-art accuracy. YOLOv5m is a medium-sized model with 21.2 million parameters. It is perhaps the best-suited model for many datasets and training as it provides a good balance between speed and accuracy. YOLOv5 uses a genetic algorithm to generate the anchor boxes.

They call this process auto anchor, which recomputes the anchor boxes to fit the data if the default ones are not good. This is used in conjunction with the k-Means algorithm to create k-Means evolved anchor boxes. This is one of the reasons why YOLOv5 works so well, even on varied datasets.

Another reason for such good training and detection results of the YOLOv5 model is mosaic augmentation. In simple words, it combines 4 different images into one so that the model can learn to deal with varied and difficult images. It uses other augmentation techniques also, along with mosaic augmentation.

The architecture of Yolov5 consists of three parts: (1) Backbone: CSPDarknet, (2) Neck: PANet, and (3) Head: Yolo Layer. The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Finally, Yolo Layer outputs detection results (class, score, location, size).

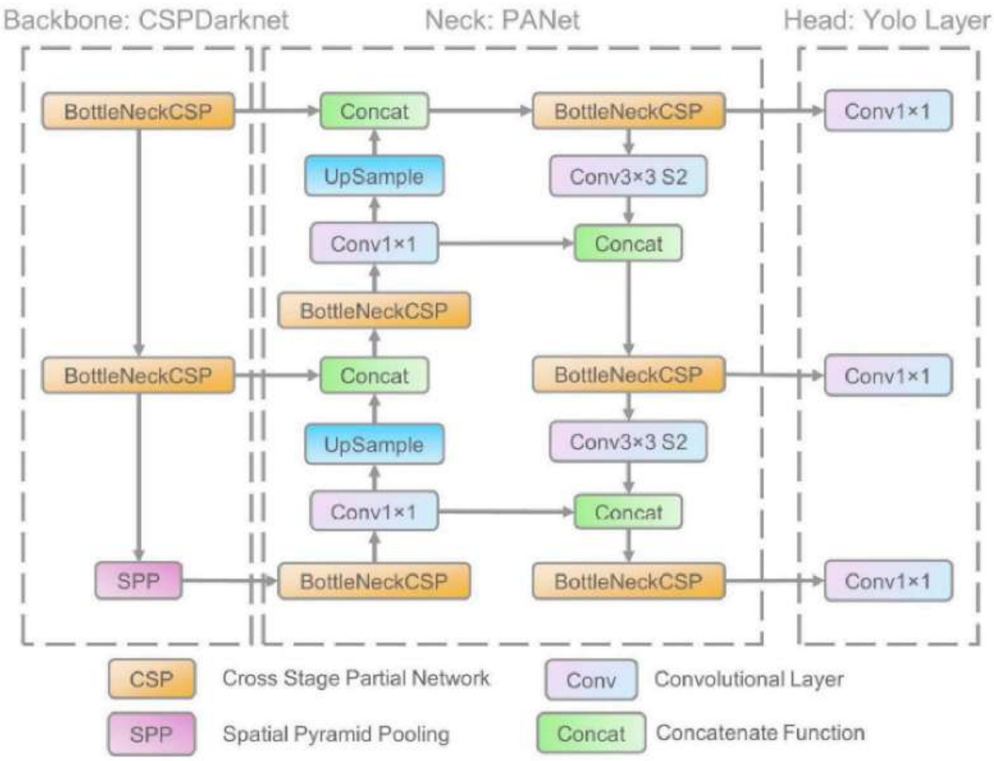

Figure 4.2: The Network Architecture of Yolov5

```
# YOLOv5m.yaml
```

```
# Parameters
nc: 80  # number of classes
depth_multiple: 0.67  # model depth multiple
width_multiple: 0.75  # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
```

```
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]],  # cat backbone P3
  [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]],  # cat head P4
  [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]],  # cat head P5
  [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
 ]
```

4.3 Design and Implementation of Robotic Arm

The parts of the robotic arm are decided, the placement of different parts and its configuration is designed along with the degree of freedom of the arm. The camera location relative to the arm position is dependent on the mapping function between the 3D position from the camera and the real-world position. Embedded C programming is used for programming the functionalities and movement of the arm.

Chapter 5

Evaluation and Analysis

5.1 Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. It is used to measure the performance of a classification model. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score. The following 4 are the basic terminology which will help us in determining the metrics we are looking for.

- True Positives (TP): when the actual value is Positive and predicted is also Positive.
- True negatives (TN): when the actual value is Negative and prediction is also Negative.
- False positives (FP): When the actual is negative but prediction is Positive. Also known as the Type 1 error.
- False negatives (FN): When the actual is Positive but the prediction is Negative. Also known as the Type 2 error.

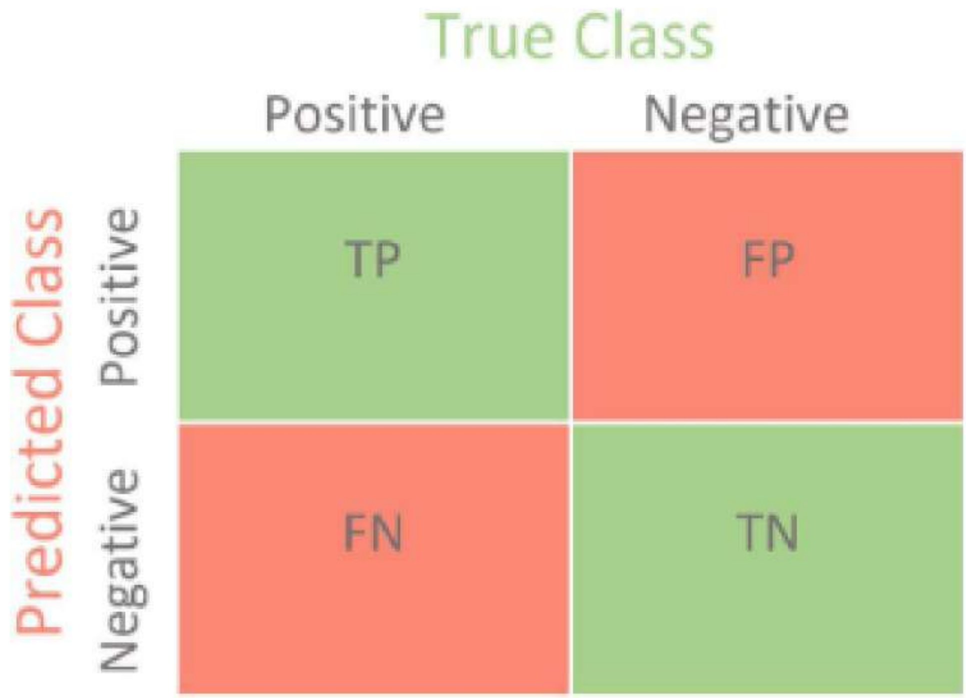

Figure 5.1: Confusion Matrix

5.1.1 Performance measures

- Accuracy:

  Accuracy is the number of correct predictions over all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

- Precision:

  Precision is a measure of how many of the positive predictions made are correct.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

- Recall:

  Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data.

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

- F1-score

  F1-Score is a measure combining both precision and recall. It is generally described as the harmonic mean of the two.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.4}$$

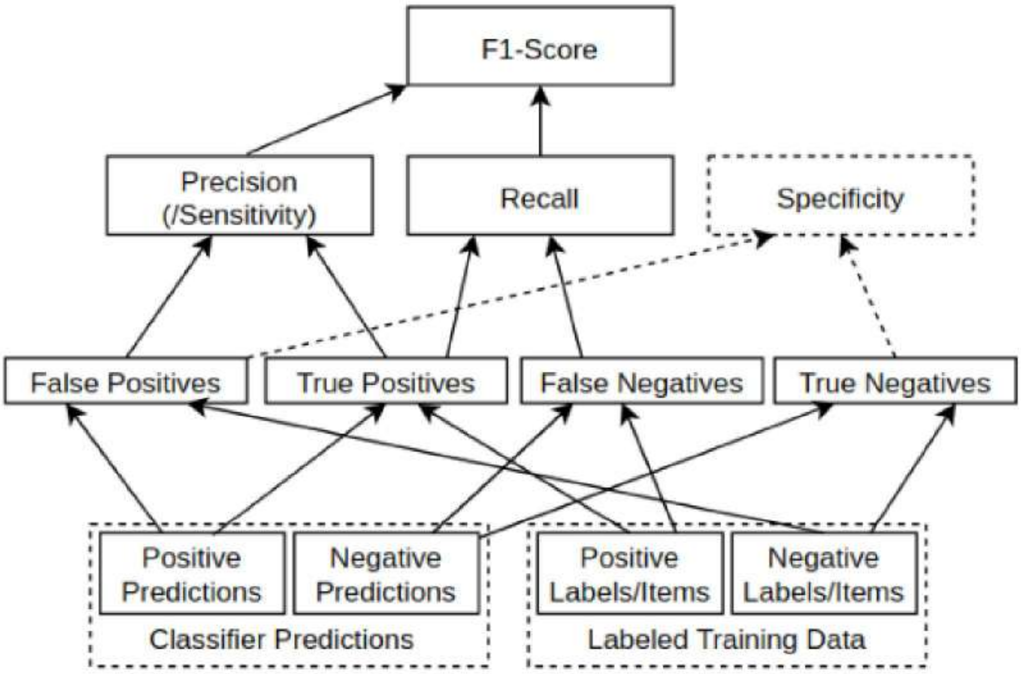

Figure 5.2: Hierarchy of Performance Metrics

5.2 mAP for Accuracy Evaluation

To evaluate object detection models like R-CNN and YOLO, the mean average precision (mAP) is used. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detections.

Confidence score: reflects the probability that an anchor box contains an object. It is usually predicted by a classifier.

Ground truth bounding box ($B_g t$): represents the desired output of an algorithm on an input, for example, the hand labelled bounding box from the testing set that specifies where the objects are in the image.

Predicted bounding box (B_p): represents a rectangle region generated from the model detector that indicates the location of the object predicted.

Intersection over union (IoU): an evaluation metric used to measure the area encompassed by both the ground-truth bounding box (B_gt) and the predicted bounding box (B_p).

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{5.5}$$

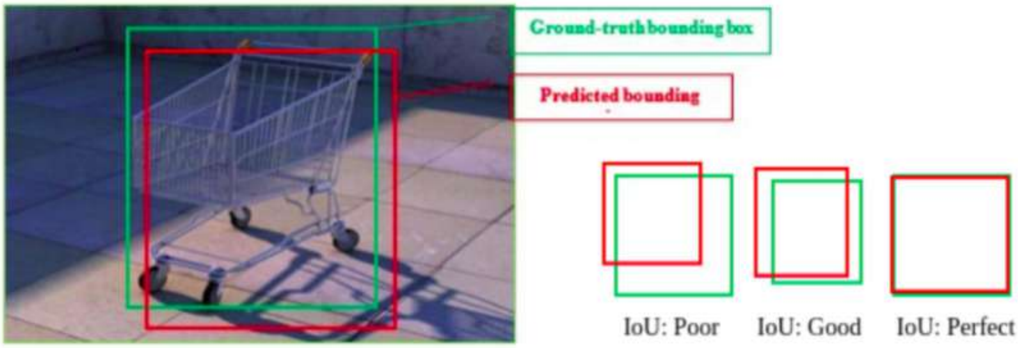

Figure 5.3: Bounding Box and IoU Analysis

Chapter 6

Future Scope

The product is a robotic arm that is capable of plucking green chilli. To expand the capabilities of the product, one potential modification would be to expand the detection labels to include more types of fruits and vegetables. This would require additional training data and algorithms to be implemented so that the robotic arm can identify and pluck a wider range of items. Another potential modification would be to add an additional collector to the product, which would be responsible for collecting the plucked items. This collector would need to be designed to work in coordination with the robotic arm so that it can collect the items without disrupting the plucking process.

Another possible modification would be to increase the height range of the robotic arm. This would allow the arm to reach higher or lower items, but it would depend on the design and capabilities of the current arm. Additionally, the product could be modified to distinguish between different ages of chili and determine its maturity rate. This would likely require additional sensors or imaging equipment, as well as additional training data and algorithms to make use of that data. However, these modifications will require more development, testing and validation to ensure that the product works effectively and efficiently.

Chapter 7

Conclusion

The project aims to create a device that can automate the harvesting process of chilli with minimal human effort. It has been developed to use the YOLOv5 algorithm, a real-time object detection algorithm that is able to identify chili peppers on the plant with a high accuracy of 0.9. This means that the algorithm is able to accurately locate the chilli peppers on the plant the majority of the time. By using the YOLOv5 algorithm, the device can locate the chilli with precision and provide a 3D location estimation system that has a minimal error.

Figure 7.1: Prediction Result of Chilli Identification

REFERENCES

[1] Bosland, P.W.; Walker, S. Growing Chiles in New Mexico [Guide H-230]; Guide H-New Mexico State University, Cooperative Extension Service (USA): Las Cruces, NM, USA, 2014.

[2] Taylor, J.E.; Charlton, D.; Yúnez-Naude, A. The end of farm labor abundance. Appl. Econ. Perspect. Policy 2012, 34, 587–598

[3] Funk, P.A.; Walker, S.J. Evaluation of Five Green Chile Cultivars Utilizing Five Different Harvest Mechanisms. Appl. Eng. Agric.2010, 26, 955–964

[4] Boaz Arad,Jos Balendonck,Ruud Barth, Development of a sweet pepper harvesting robot https://onlinelibrary.wiley.com/doi/epdf/10.1002/rob.21937

[5] Chilli Cultivation: Ideal Conditions, Varieties, Land Preparation, Sowing, Transplanting, Pest Management & Harvesting https://krishijagran.com/agripedia/chilli-cultivation-ideal-conditions-varieties-land-preparation-sowing-transplanting-pest-management-harvesting

[6] A Study on the Feasibility of Robotic Harvesting for Chile Pepper by Muhammad Umar Masood and Mahdi Haghshenas-Jaryani Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruses, NM 88003, USA

[7] An Innovative System for Picking Mango FruitMohamed, T. H. A.Agricultural Engineering Research Institute , Egypt

[8] A Study on the Feasibility of Robotic Harvesting for Chile Pepper by Muhammad Umar Masood and Mahdi Haghshenas-Jaryani

[9] Machine learning and its applications: A review Sheena Angra:Chitkara University, India,Sachin Ahuja: CURIN, Chitkara University, India

[10] Review of deep learning: concepts, CNN architectures, challenges, applications, future directions Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie & Laith Farhan

[11] Understanding of a convolutional neural network ,Saad Albawi: Department of Computer Engineering, Istanbul Kemerburgaz University, Istanbul, Turkey,Tareq Abed Mohammed: Altinbas Universitesi, Istanbul, TR,Saad Al-Zawi: University of Diyala, Diyala, IQ

[12] A comparative study of YOLOv5 models performance for image localization and classification, September 2022;Conference: 33rd Central European Conference on Information and Intelligent Systems (CECIIS) 2022At: Dubrovnik, Croatia;Marko HorvatMarko HorvatLjudevit JelečevićLjudevit JelečevićGordan GledecGordan Gledec

[13] Pandas: a Foundational Python Library for Data Analysis and Statistics;January 2011 ;Wes MckinneyWes Mckinney

[14] The NumPy Array: A Structure for Efficient Numerical Computation;May 2011Computing in Science and Engineering 13(2):22 - 30;SourceIEEE Xplore Stéfan Johann van der WaltStéfan Johann van der WaltS. Chris ColbertGael VaroquauxGael Varoquaux

[15] A Review Paper on Raspberry Pi and its Applications;January 2020;Hirak GhaelHirak Ghael

[16] A brief introduction to OpenCV;Ivan Culjak Faculty of electrical engineering and computing, University of Zagreb, Zagreb, Croatia,Tomislav Pribanic :Faculty of electrical engineering and computing, University of Zagreb, Zagreb, Croatia,Mario Cifrek:Faculty of electrical engineering and computing, University of Zagreb, Zagreb, Croatia