MACHINE VISION BASED AUTOMATED 3-DOF ARTICULATED ROBOT FOR FRUIT DEFECT IDENTIFICATION AND SEGREGATION

A PROJECT REPORT

Submitted by

ARAVIND S	**181201004**
ARIHARAN K	**181201005**
VENKATESHWARAN T K	**181201051**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

MECHATRONICS ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO ANNA UNVERSITY, CHENNAI)

DEPARTMENT OF MECHATRONICS ENGINEERING

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2022

RAJALAKSHMI ENGINEERING COLLEGE

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO ANNA UNVERSITY, CHENNAI)

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this Report titled "**MACHINE VISION BASED AUTOMATED 3-DOF ARTICULATED ROBOT FOR FRUIT DEFECT IDENTIFICATION AND SEGREGATION**" is the bonafide work of **ARAVIND S (181201004), ARIHARAN K (181201005), VENKATESHWARAN T K (181201051)** who carried out the work under my supervision.

SIGNATURE
(Dr. V. SANTHANAM)
PROFESSOR AND HEAD

Department of Mechatronics Engineering
Rajalakshmi Engineering College
(Autonomous)
Rajalakshmi Nagar, Thandalam.
Chennai - 602 105

SIGNATURE
(Mr. S .Ramkumar)
SUPERVISOR
Assistant Professor

Department of Mechatronics Engineering
Rajalakshmi Engineering College
(Autonomous)
Rajalakshmi Nagar, Thandalam.
Chennai - 602 105

This **MT17811-Project Work Phase-II** report is submitted for the **VIVA-VOCE** Examination held on **............................** at Rajalakshmi Engineering College (Autonomous), Chennai.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The automation scenario in the current industrial as well as domestic applications has seen an exponential growth over the decade. Robot plays an important role in industrial automation but in some cases, it needs some extent of human intervention in quality inspection. This paper focuses on solving problems related to defect identification and segregation of fruits using 3-DOF articulated robot configuration with the integration of machine vision system and deep learning algorithms. To perceive the features of object, USB based camera is used. Machine vision system will then employ various digital image processing techniques to extract the required information and make the decision of what to do with the product, usually by passing it on to the next part of the process, diverting it to another process, or discarding the product from the production or distribution system. In recent days, the advancement in deep learning leads to the phenomenal growth in computer vision. There are many state-of-the-art object detection algorithms available in deep learning technology and we are using Inception-V3 algorithm that can be used for the machine vision operation in our solution. Based on the results from the algorithm, segregation operation is carried out successfully.

Keywords: Industrial automation, Computer vision, Machine vision, Defect detection, Articulated Robot, InceptionV3 Algorithm.

ACKNOWLEDGEMENT

Initially we thank the almighty for being with us through every walk of life it is our privilege to express our sincerest thanks to our respected Chairman **Mr. S. Meganathan, B.E., F.I.E**. and sincerest thanks to our beloved chairperson **Dr. Mrs Thangam Meganathan, M.A., M.Phil., Ph.D.** and beloved Vice-chairman **Mr. M. Abhay Shankar Meganathan, B.E., M.S** for providing us with the requisite infrastructure and extending support in all endeavors. Our heartfelt thanks to **Dr. S. N. Murugesan, M.E., Ph.D**. our principal for his kind support and resources provided to complete our work in time. We deeply express our sincere thanks to **Dr. V. Santhanam, M.E., Ph.D**. Head of our Department, for his encouragement and continues support to complete the project in time.

We are glad to express our sincere indebtedness to our department project coordinator **Mr. M. Sridharan, M.E.,** Assistant Professor, Department of Mechatronics Engineering for their constructive criticism throughout the duration of our project. We are glad to express our sincere thanks and regards to our guide & supervisor **Mr. S. Ramkumar, M.E** Assistant Professor, Department of Mechatronics Engineering for his guidance and suggestion throughout the course of the project. Finally, we express our thanks for all teaching, non- teaching faculty of our Mechatronics Engineering department and our parents for helping us with the necessary suggestions and guidance during the time of project.

ARAVIND S

ARIHARAN K

VENKATESHWARAN T.K

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THE WORK

To implement the deep learning model and integrate it with the articulated robot and enhance the process of defect detection by minimalizing the manual labour workforce.

1.2 GENERAL

The first use of the word "robot" occurred in a play about mechanical men that are built to work on factory assembly lines and that rebel against their human masters. Asimov had a much brighter and more optimistic opinion of the robot's role in human society than did Karel Capek. He generally characterized the robots in his short stories as helpful servants of man and viewed robots as a better, cleaner race. A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of task. These robots are subjected to work under Asimov's robotics laws.

1.1.1 Industrial Revolution

Manufacturers were turning away from hand production methods towards machines to increase productivity. The First Industrial Revolution began in the 18th century through the use of steam power and mechanisation of production. Steam power was already known. The use of it for industrial purposes was the greatest breakthrough for increasing human productivity. The Second Industrial Revolution began in the 19th century through the discovery of electricity and assembly line production. Henry Ford (1863-1947) took the idea of mass production from a slaughterhouse in Chicago. Henry Ford carried over these principles into automobile production and drastically altered it in the process. The Third Industrial Revolution began in the ’70s in the 20th century through partial automation using memory programmable controls and computers. Since the introduction of these technologies, we are now able to automate an entire production process without human assistance.

We are currently implementing the Fourth Industrial Revolution. This is characterised by the application of information and communication technologies to industry and is also known as "Industry 4.0". It builds on the developments of the Third Industrial Revolution. Industry 4.0 is the digital transformation of manufacturing/production and related industries and value creation processes. They use modern control systems, have embedded software systems and dispose of an Internet address to connect and be addressed via IoT (the Internet of Things). Industries 4.0 refers to the intelligent networking of machines and processes for industry with the help of information and communication technology.

1.3 TYPES OF ROBOTS

There are various fundamental types of robot configurations used in numerous applications. An articulated robot is a robot which is fitted with rotary joints. Rotary joints allow a full range of motion, as they rotate through multiple planes, and they increase the capabilities of the robot considerably. A Selective Compliance Articulated Robot Arm (SCARA) is a good and cost-effective choice for performing operations between two parallel planes. SCARA robots excel at vertical assembly tasks such as inserting pins without binding due to their vertical rigidity. Delta robots, also referred to as “spider robots,” use three base-mounted motors to actuate control arms that position the wrist. Cartesian robots typically consist of three or more linear actuators assembled to fit a particular application. Cartesian robots typically use standard linear actuators and mounting brackets, minimizing the cost and complexity of any "custom cartesian system.

1.3.1 Use of Articulated Robots

Articulated robots provide more degrees of freedom than any other robot type, which is why they are commonly utilized amongst manufacturers. Their enhanced range of motion closely mimics that of a human's, making them ideal solutions for production lines. They also provide more flexibility within production operations. Their ability to cover a number of movements makes them more adaptable to changes to the production process or workpieces. Enhanced motion provides a greater work envelope for the robot, allowing for the handling of a variety of workpieces from small to large. They also provide versatility through the numerous applications they can complete. These applications include arc welding, material handling, assembly, part transfer, pick and place, packaging, machine loading, and palletizing along with many others

1.4 MACHINE VISION

Machine Vision Systems see a major development over the period since automation showed its head in the industrial revolution. These systems are used to monitor, inspect and notice the parameters that are set to the particular system used in an industrial application. Even these systems are now integrated with Artificial Intelligence and Machine Learning Algorithms to perform as cognitive as a human does. As a result of this, automation process became much more meaningful and reasonable to be used in any field nowadays. This increases the process efficiency to a great extent. Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain. Henceforth, this increases the chance of opportunities of usage of the same in the near future.

1.4.1 Object Detection

Object recognition is an application of dee KI p learning and it is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs. Image classification involves predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. Object detection combines these two tasks and localizes and classifies one or more objects in an image.

1.5 KINEMATICS

Kinematics is the study of motion of points, objects and systems of group of objects without reference to the force that causes the motion. The word “kinesis” comes from a Greek word meaning motion. Kinematic analysis is the process of measuring the kinematic quantities used to describe motion. Kinematics aims to provide a description of the spatial position of bodies or systems of material particles, the rate at which the particles are moving and the rate at which their velocity is changing.

1.5.1 Forward Kinematics

The kinematics that deals with the analytical study of the geometry of motion of a robot arm with respect to a fixed reference coordinate system as a function of time without regard to the forces that causes the motion. The relations between the joint-variable space and the position and orientation of the end-effector of a robot arm. Vector and matrix algebra are utilized to develop a systematic and generalized approach to describe and represent of links of a robot arm.

1.5.2 Inverse Kinematics

The kinematics that deals with the analytical study of angles of the joints with respect to the position and orientation of the gripper. There are various approaches and methods that can be used to solve the inverse kinematic problem. Some of them are inverse transform, screw algebra, dual matrices, dual quaternion, iterative and geometric approaches.

1.6 ROBOTIC MANIPULATOR

Manipulators are composed of an assembly of links and joints. Links are defined as the rigid sections that make up the mechanism and joints are defined as the connection between two links. The device attached to the manipulator which interacts with its environment to perform tasks is called the end-effector. A robot manipulator is an electronically controlled mechanism, consisting of multiple segments, that performs tasks by interacting with its environment. They are also commonly referred to as robotic arms. Robot manipulators are extensively used in the industrial manufacturing sector and also have many other specialized applications (for example, the Canadarm was used on space shuttles to manipulate payloads). The study of robot manipulators involves dealing with the positions and orientations of the several segments that make up the manipulators. This module introduces the basic concepts that are required to describe these positions and orientations of rigid bodies in space and perform coordinate transformations.

1.6.1 Robot Workspace

The robot workspace is also known as reachable space. The places that end effector can reach. The workspace is dependent on the DOF angle/translation limitations, the arm link lengths, the angle at which something must be picked up A robot's workspace is the total volume swept out by the end effector as the manipulator executes all possible motions. The shape of the workspace dictates the applications for which each design can be used.

1.6.2 Degrees of Freedom

Degrees of freedom refers to the maximum number of logically independent values, which are values that have the freedom to vary, in the data sample. Degrees of freedom are commonly discussed in relation to various forms of hypothesis testing in statistics, such as a chi-square. Calculating degrees of freedom is key when trying to understand the importance of a chi-square statistic and the validity of the null hypothesis.

1.7 DEEP LEARNING

Deep learning is a type of machine learning; a subset of artificial intelligence (AI) that allows machines to learn from data. Deep learning involves the use of computer systems known as neural networks. In neural networks, the input filters through hidden layers of nodes. These nodes each process the input and communicate their results to the next layer of nodes. This repeats until it reaches an output layer, and the machine provides its answer. There are different types of neural networks based on how the hidden layers work. Image classification with deep learning most often involves convolutional neural networks, or CNNs. In CNNs, the nodes in the hidden layers don’t always share their output with every node in the next layer (known as convolutional layers). Deep learning allows machines to identify and extract features from images. This means they can learn the features to look for in images by analysing lots of pictures. So, programmers don’t need to enter these filters by hand.

1.7.1 Image Classification

Image classification is where a computer can analyse an image and identify the ‘class’ the image falls under. (Or a probability of the image being part of a ‘class’.) A class is essentially a label, for instance, ‘car’, ‘animal’, ‘building’ and so on. Early image classification relied on raw pixel data. This meant that computers would break down images into individual pixels. The problem is that two pictures of the same thing can look very different. They can have different backgrounds, angles, poses, etcetera. This made it quite the challenge for computers to correctly ‘see’ and categorise images.

1.7.2 Inception-V3

The Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014. As the name suggests it was developed by a team at Google.

The inception V3 is just the advanced and optimized version of the inception V1 model. The Inception V3 model used several techniques for optimizing the network for better model adaptation. It has higher efficiency. It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised. It is computationally less expensive. It uses auxiliary Classifiers as regularizes.

The inception v3 model was released in the year 2015, it has a total of 42 layers and a lower error rate than its predecessors. Let's look at what are the different optimizations that make the inception V3 model better. The major modifications done on the Inception V3 model are Factorization into smaller convolutions, Spatial factorization into asymmetric convolutions, Utility of auxiliary classifiers, Efficient grid size reduction.

1.8 PROBLEM STATEMENT

In the area of defect detection of fruits and vegetables there is demand for labour workforce. The market value of fruits and vegetables depend upon the consistency of external features like colour, shape, size. Deep learning models are yet to integrate with articulated robots and used in these areas. Hence, we are about to provide a solution for the same.

1.9 OUTLINE OF THE PROJECT

The objective of our project is to identify the defective and non-defective fruits and segregate it accordingly. Technology used are Robotics, Computer Vision, Deep Learning. Tools used are Pick and Place Robot Kit, Arduino microcontroller, laptop, python, Deep Learning Algorithm (Inception V3).

This project setup has the camera faced down towards the fruit and the pick and place robot place near the fruit to pick. The camera capture the image of the fruit when it is placed in the respective position and the image is given into the deep learning algorithm for further processing. After the processing the deep learning algorithm classifies the fruit as defective or non-defective based upon the texture of the fruit. If it is defective, then the robot picks the fruit and places it in the left slot and if it is non-defective it place the fruit in the right slot. Then the camera checks if there is any fruit is place in the pickup slot and if there is any fruit placed then the robot starts it operation. Otherwise, the robot remains idle. This operation continues until the user stop the process using the program manually.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In the modern world, it is difficult to apply knowledge into automation field and expect changes at a faster rate. It takes years of dedication, persistence and creativity to make an automated system to serve a purpose. We also do realize that other people are also working in the same field in order to develop a better tomorrow. So, it necessitates us to learn about other technologies being developed simultaneously, thereby helping us to adapt and connect to the changing world. In this section, we will discuss about the technologies developed so far, individually and the technologies incorporated in our system along with the modifications.

2.2 PREVIOUS WORK

In this section, similar projects have been studied to enable better implementation option.

2.3 BASED ON DESIGN

Sapan Naik et al. (2017) The quality of the fruit is determined by the factors like the outside appearance and internal quality. The appearances of the fruits are examined based on color, texture, size, shape. The manual process of controlling the external quality of the fruits is time consuming and requires a large labor force. To overcome this , machine has been integrated with computer vision to automate the task of controlling the external quality of fruits. The approaches that are used in extraction of features are Speeded Up Robust Features (SURF), Histogram of Oriented Gradient (HOG), Local Binary Pattern

(LBP). After the features extraction the data is processed by the machine learning algorithms like K-nearest neighbor (KNN), Support Vector Machine (SVM) , Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN).

A.Anushya (2020) The quality inspection of the banana can be done by checking the ripeness of the banana. In this research, the banana is classified into three groups - ripe, mid-ripe, and overripe. The ripeness of the banana can be found by using computer vision techniques like Gray Level Co-occurrence matrix and edge detection algorithm like Canny Edge detector to extract the texture features of the banana and Machine Learning algorithms like decision tree or deep learning algorithms to classify the bananas into the three groups i.e., ripe, mid-ripe, over-ripe. The neural networks give more accuracy than decision tree in this research. So, the quality inspection of the banana can be done without the help of skilled labor and time consumption can be reduced.

Madiha Farman et al.(2018) This paper concerns with the design of a three degrees of freedom robotic arm, which is intended to pick and place lightweight objects based on a color sorting mechanism. It is mainly made of three joints, a gripper, two rectangular shaped links, a rotary table and a rectangular platform. The angular rotation of each joint is powered by a servomotor. Furthermore, the angular position of each servomotor shaft is controlled by a signal from an Arduino microcontroller which executes a Matlab code. The Matlab code includes the inverse kinematics equations which are necessary for the determination of the target joint angles for a certain Cartesian position of the end-effector. The robotic arm’s design process included several static and dynamic calculations, mechanical properties calculations and prototype testing in order to provide a final product with well-established structure and functionalities.

Hamza Alzarok et al.(2020) Industrial robots have been more and more involved in the automation industry due to their capability to perform precise tasks, with an accuracy in sub-millimeters, tasks such as welding and drilling where a successful cooperation between the robots and the machine vision is necessary to end tasks within a demanded accuracy and in less execution time. The feedback from the machine visionsis used for enhancing the efficiency of detection, tracking and control of the robot motion by utilizing their visual information. The feedback, therefore, improves the safety of the system by preventing the robots from being damaged and operators from being injured which, in turn, saves the production time.

2.4 BASED ON COMPUTING TECHNIQUES

Narendra V.G et al.(2020) In this paper various algorithms proposed for quality inspection, including external fruit defects (i.e., RGB to L*a*b * color conversion and defective area calculation methods, are used to recognize errors in both Apple and Orange) and vegetables (i.e., K-means cluster and defective area calculation methods are used to identify defective tomatoes in color) and other image techniques are also used. The image of the vegetables and fruits are seen as the RGB image. In order to enhance image quality, the pre-processing approach has been used. The segmentation process focuses on the desired fruit and vegetable portion. Feature extraction is carried out using the Segmentation-based Fractal Texture Analysis (SFTA) method and is categorized according to the data collection. The data collection was trained and tested in the Naïve Bayes classifier.

Sergey Soltan et al.(2020) Accurate object classification and position estimation is a crucial part of executing autonomous pick-and-place operations by a robot and can be realized using RGB-D sensors becoming increasingly available for use in industrial applications. In this paper, we present a novel unified framework for object detection and classification using a combination of point cloud processing and deep learning techniques. The proposed model uses two streams that recognize objects on RGB and depth data separately and combines the two in later stages to classify objects. Experimental evaluation of the proposed model including classification accuracy compared with previous works demonstrates its effectiveness and efficiency, making the model suitable for real-time applications.

2.5 INFERENCE FROM LITERATURE SURVEY

We learned about the deep learning algorithms and computer vision techniques to do the image processing and then the image classification. Image classification is the main task in our project. We learned how to achieve that using the deep learning algorithms. And we also learned about the kinematic analysis, structural study, workspace of the robots.

CHAPTER 3

METHODOLOGY

3.1 SELECTION OF COMPONENTS:

3.1.1. Arduino Uno – Our Microcontroller

For the robot to be engaged in the operation of segregating the fruit (lemon in our case), we need a microcontroller that can provide sufficient supply of current to the servo motors that rotate the joints of the robots. We found Arduino UNO as our apt microcontroller because it can provide a maximum output current of 1A. This is more than sufficient for each servo motor to be served with. The 14 digital Pins in Arduino UNO had known to be useful with providing enough power to signal the servo motors. Thus, we selected and procured Arduino UNO ATMega328P to be the microcontroller of our solution.

3.1.2. Mg995 Servo Motors

For robot to perform its operation, we need a driving force for each joint in the same. These joints are powered by MG995 Servo Motors as we require a maximum of 0.89 Nm Torque at 1A of maximum current for our operation as calculated. These servos were able to satisfy our requirements along with the Plastic mount horns to fix to individual links of the robotic arm respectively.

3.1.3. 12V Li-Ion Battery Pack

For the servo motors to work well, an external power source was required as the Arduino can't satisfy the current requirement to all the three servo motors. Hence a 12V Li-ion based Battery Pack with 6 cells of 18650 type with BMS is used in our solution as our main power source.

Alternatively, we can also use 12V 1.5A DC Adapter to either charge the battery or to directly power the circuit.

3.1.4. Step Down Dc-Dc Voltage Converter 5a (Xl4015)

As we already know, Arduino can provide 5V as its maximum voltage to its digital pins which is not sufficient for the servo motors when the current is to be provided. Hence the above used 12V power supply is reduced approximately to 6V using this step-down DC-DC Voltage Converter to power the servo motors to perform at enough power and torque required for the operation intended. This is an XL4015 Based converter and it can easily sustain current up to 5A from the power source and provide it to the servo motors without any discrepancies.

3.1.5 Computer/Laptop

As we are integrating deep learning in our solution, Arduino based Embedded C programming cannot be used. Hence, we use PyFirmata to instruct the microcontroller with Python packages including the deep learning algorithm as mentioned in the method. For this, we need a computer with Python pre-installed and can run Deep learning packages so as to be serially connected with the Arduino to perform the operation using it. Serial Communication must be established to instruct the Arduino using Python for the robot to perform its operation.

3.1.6 Base Board

As our project is intended to concentrate on segregation, we require a workspace to move forward. We used a Ply Wood that was available with us and cut it to a dimension of 700×415×5 mm. This was enough for us to implement our prototype to segregate on both the sides as shown in the picture (Fig 3.1).

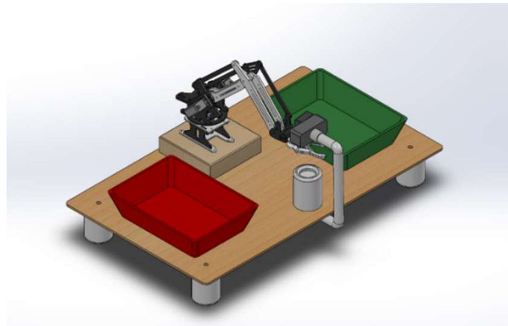


Fig 3.1 Baseboard of the robot

3.1.7 Wooden Block

Our robot being a basic model required a sturdier platform to be supported on. Hence, we fixed the base to a wooden block of dimension 150×150×40 mm. This is nailed to the base board in order to hold the robot at its operative position.

3.1.8 Plastic Bush

Our setup is now required to be held from the ground by certain height. We used 5 plastic bushes to be attached to the base with 50mm height and 25.4mm diameter. They were fastened in rectangular pattern and one in the middle of the base. Another plastic bush that was available was used as a staging location to withstand our subject which is the fruit. This enabled the robot to pick the fruit during the operation without any disturbances.

3.1.9 Lenovo Webcam 1080p

For our Deep learning algorithm to perform with good accuracy, we required a good imaging sensor with favourable resolution to be used. Hence, we chose Lenovo's Webcam that has a resolution of 1920×1080. This enables the subject to be trained, tested and sensed in real time along with capturing entire features of the subject that is required to be handled. Hence, we selected this camera to our imaging sensor in our solution.

3.1.10 Other Components

PVC Pipes - We needed a good camera mount to fix the camera in top view of the subject. We used simple pipes of required lengths and elbows to make a mount that brings out from the base.

Breadboard and Jumper Wires - For connecting the servo motors to the power supply and bringing the signal from the microcontroller, we used jumper wires and breadboard in our circuit. Thus, these were the components that were selected during our project requirement phase and also implemented in our solution.

3.2 DEEP LEARNING TRAINING AND IMPLEMENTATION

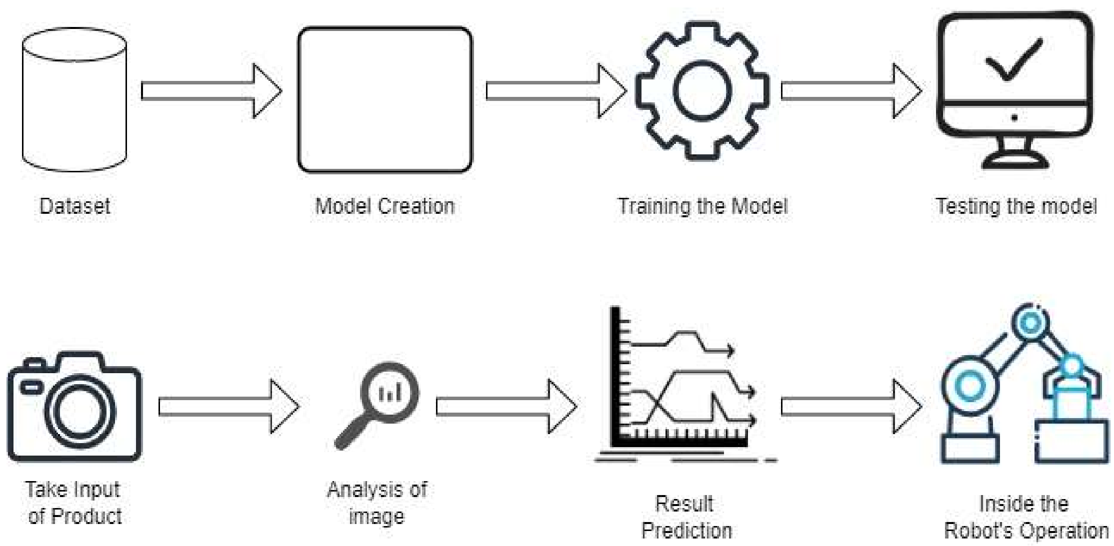


Fig 3.2 Flow Diagram of Deep Learning Testing

The operational flowchart (Fig 3.2) shows the working of the deep learning model that is employed for the fruit image classification. The first stage is where the dataset is gathered by capturing the images of the fruit. A model is created with the gathered dataset. The model is then trained to predict whether the fruit is defective or not by extracting the external features like colour, size, appearance. Then the model is tested to verify whether the output is as per the actual output. The output percentage may vary depending upon the methods that are used for the prediction. The second stage is where the implementation of the deep learning model takes place. The image from the camera is sent as input to the deep learning model it then uses Inception V3 algorithm to identify the type of fruit and then sends signals to initiate the robot operation.

3.3 CIRCUIT DIAGRAM

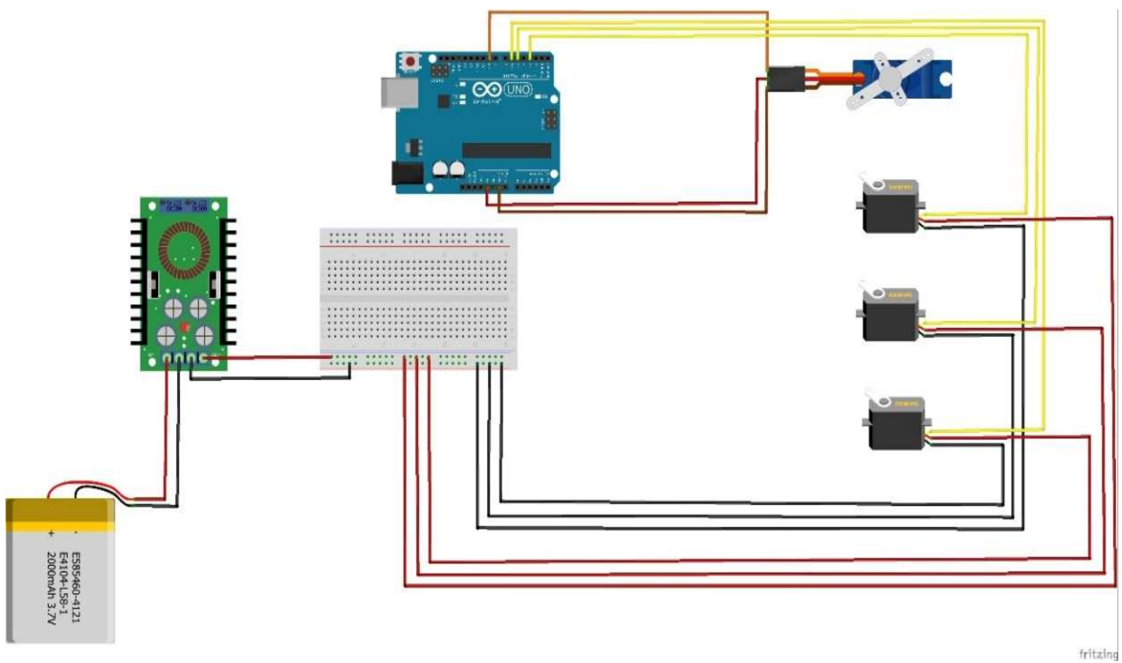


Fig 3.3 Circuit Diagram for 3DOF Robot operation

The circuit diagram (Fig 3.3) shows the connections for the robot operation. The components that are used for this connection are Li-ion battery, Arduino Uno Borad, MG955 metal servo motor, Breadboard, X4015 Step-down DC converter, jumper wires. The supply for the servo motors are from the Li-ion battery except the end effector servo which get its supply from the arduino board itself. The Step-down DC converter is used, as the output from the battery is more than the required supply for the servo motors. The base servo, left servo, right servo, end effector servo motors are connected to the pin no 3,5,6,9 respectively. These pins are used to send the actuating signals to the respective servo motors

3.4 FREE BODY DIAGRAM

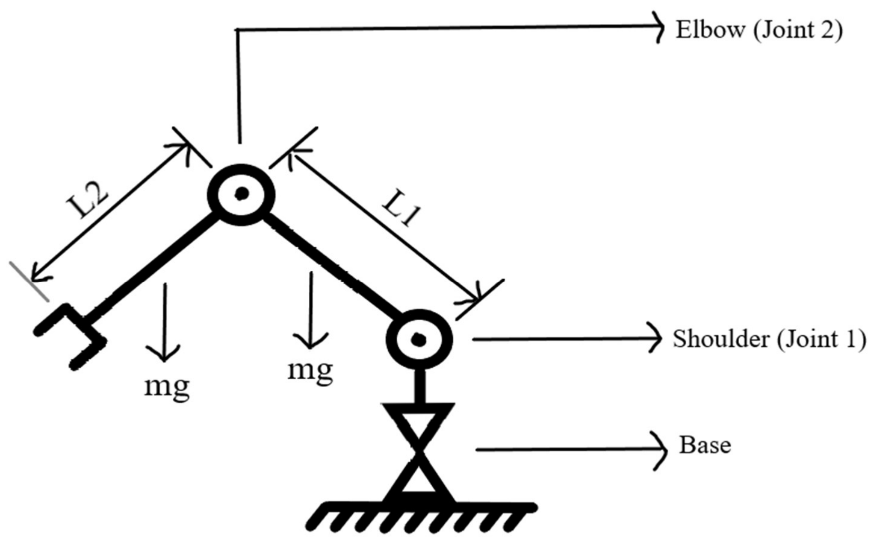


Fig 3.4 Free Body diagram of Robot Arm

Let

L_1 – Length of the link1 from shoulder to elbow,

L_2 – Length of the link2 from elbow to end effector,

mg – Force of gravity acting upon the link

The free body diagram of the robot arm (Fig 3.4) shows the force that are acting upon the link. The robot consists of a base part where the robot is fixed to ground and a servo motor is attached to the base of the robot. The base part will rotate about the axis that is fixed in an range from 0° to 180°. The shoulder part of the robot is where the servo motors are attached for the actuation of the shoulder part. The elbow part of the robot is attached with a ternary link so as facilitate the movement of the robot. The end effector attached here is a gripper as the fruit is needed to be picked.

3.4.1 SPECIFICATIONS OF THE ROBOT

Table 3.1 Robotic Arm Specifications

Type	Articulated
Degrees Of Freedom	3
Configuration	RRR
Maximum Reach	400mm approx.
Maximum Payload	40 grams
Maximum height	250 mm

The specification of the robot is mentioned in (Table 3.1). As every industrial project, we also used the articulated robot in our project. This articulated robot is also termed as pick and place robot in layman terms. This can be easily done with three degrees of freedom configuration. We need a revolute joint at the end of each frame to optimize our pick and place operation and so we choose RRR configuration for our robot. The maximum reach of the end effector of our robot is upto 400 mm. This maximum reach is fixed for the optimal workspace that we designed. We designed this robot to pick the fruits that weigh upto 40 grams and the maximum height that the robot’s end effector reaches upto 250 mm.

3.5 ROBOT OPERATIONAL FLOW

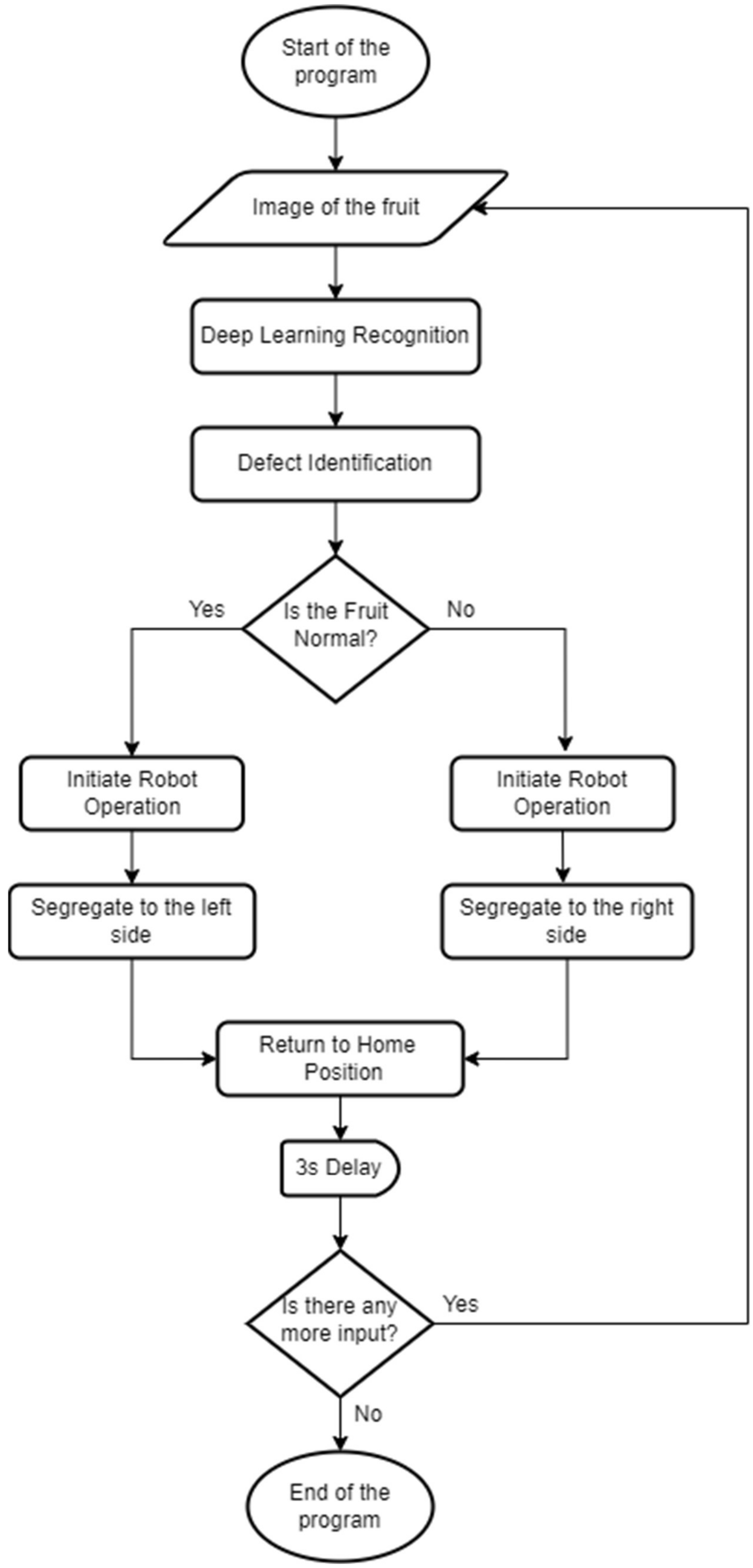


Fig 3.5 Workflow of the Robot

3.6 CALCULATIONS

3.6.1 Torque Calculations

$$T = F \times sin\theta \times r \tag{3.1}$$

We know that,

$$F = m \times g \tag{3.2}$$

By substituting (3.2) in (3.1) we get,

$$T = m \times g \times sin\theta \times r \tag{3.3}$$

Torque developed at Joint 1:

$$T_1 = F_{load} \times (L_1 + L_2) + W_2 \times (L_2 \div 2) + L_1 + W_3 \times L_1 + W_1 \times (L_1 \div 2) \tag{3.4}$$

Torque developed at Joint 2:

$$T_2 = F_{load} \times L_2 + W_2 \times L_2 \div 2 \tag{3.5}$$

By Calculating based on the above given formulas, let us consider 200g payload at maximum we obtain approximate torque value of

$$T_1 = 0.9\ Nm$$

$$T_2 = 0.3\ Nm$$

3.6.2 DH Parameters

The DH parameters of the robot is given in the table (Table 3.2). We have taken the DH parameters of the robot with one orientation of the robot. The twist angle in the base servo, shoulder, elbow are 0°, 90°, 0°. The link length of link 1,link 2,link 3 are 0, 0.15m (a_1), 0.16m (a_2). The offset in shoulder is 0.01m (d_2)

Table 3.2 DH Parameters of 3-DOF Robot

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	0	θ_1
2	90°	a_1	d_2	θ_2
3	0°	a_2	0	θ_3

The matrix for obtaining the homogeneous matrix is given below,

$$^{i-1}T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix for each joint are listed below,

$$^0T_1 = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

$$^1T_2 = \begin{bmatrix} cos\theta_2 & 0 & sin\theta_2 & 0.15cos\theta_2 \\ sin\theta_2 & 0 & -cos\theta_2 & 0.15sin\theta_2 \\ 0 & 1 & 0 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

$$^2T_3 = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & 0.16cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & 0.16sin\theta_3 \\ 0 & 0 & 1 & 0.02 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Multiplying all the transformation matrices,

$$T = {}^0T_1 \times {}^1T_2 \times {}^2T_3$$

We arrive at the final homogeneous transformation matrix,

$$T = \begin{bmatrix} acos\theta_3 & -asin\theta_3 & b & 0.16acos\theta_3 + 0.02b + c \\ sin\theta_3 & -dsin\theta_3 & e & 0.16dcos\theta_3 + 0.02e + f \\ sin\theta_3 & cos\theta_3 & 0 & 0.16sin\theta_3 + 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Here,

$$a = cos\theta_1\theta_2 - sin\theta_1\theta_2$$

$$b = cos\theta_1 sin\theta_2 + sin\theta_1 cos\theta_2$$

$$c = 0.15cos\theta_1 cos\theta_2 - 0.15sin\theta_1 sin\theta_2$$

$$d = sin\theta_1 cos_2 + cos\theta_1 cos\theta_2$$

$$e = sin\theta_1 sin\theta_2 - cos\theta_1 cos\theta_2$$

$$f = 0.15sin\theta_1 cos\theta_2 + 0.15cos\theta_1 sin\theta_2$$

The position values (x, y ,z) are derived from the above matrix T,

$$x = 0.16 \times a \times cos\theta_3 + 0.02 \times b + c \quad (3.10)$$

$$y = 0.16 \times d \times cos\theta_3 + 0.02 \times e + f \quad (3.11)$$

$$z = 0.16 \times sin\theta_3 + 0.01 \quad (3.12)$$

Let $\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 30°$

By substituting the values of θ_1, θ_2, θ_3 in the equations (3.10), (3.11), (3.12) , we arrive at the position values,

$$x = 0.02m$$

$$y = 0.15m$$

$$z = 0.15m$$

Hence, we have achieved the forward kinematics of the robot with the angles specified.

3.7 SOFTWARES AND CODES USED

3.7.1 pyfirmata

Firmata is a protocol for communicating with microcontrollers from software on a host computer. The protocol can be implemented in firmware on any microcontroller architecture as well as software on any host computer software package. The Arduino repository described here is a Firmata library for Arduino and Arduino-compatible devices. pyFirmata is a Python interface for the Firmata protocol. There are two main models of usage of Firmata. In one model, the author of the Arduino sketch uses the various methods provided by the Firmata library to selectively send and receive data between the Arduino device and the software running on the host computer. For example, a user can send analog data to the host using Firmata.sendAnalog(analogPin, analogRead(analogPin)) or send data packed in a string using Firmata.sendString(stringToSend).

3.7.2 Algorithm Used

3.7.2.1 Inception V3 architecture implementation

```
#importing the required libraries
import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import cv2

#Loading the trained model to the variable named
model

model =
keras.models.load_model("F:\\Project\\Deep_Learning\
\model.h5")
```

```
path = "F:\\Project\\Deep_Learning\\Lemon fruit
disease dataset\\Unhealthy_Lemons\\Test.jpg"

vid = cv2.VideoCapture(0)

#This predict() function gets the live image from
the camera and returns the result as normal lemon or
defective lemon

def predict(input_img):

img = image.load_img(input_img, color_mode="rgb",
target_size=(150, 150), interpolation="nearest")
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
img = img/255
images = np.vstack([img])
classes = model.predict(images, batch_size=1)
#predict_result = []
 predict_result = ""
 max = np.amax(classes[0])
    if np.where(classes[0] == max)[0] == 0:
      predict_result = "Normal Lemon"
    elif np.where(classes[0] == max)[0] == 1:
      predict_result = "Defective Lemon"
    #print(predict_result)

    plt.figure(figsize=(5, 5))
    plt.imshow(image.load_img(path,
color_mode="rgb", target_size=(1080, 1080),
interpolation="nearest"))
    title = f"predict: {predict_result}
({round(float(max)*100, 2)}%)"
    plt.title(title, color='black')
    plt.axis('off')
    plt.show()

    return predict_result

#This start() function is used to initiate the
camera operation and storing the images to the given
path
```

```
def start():
    while(vid.isOpened()):
        ret, frame = vid.read();
        if(ret == True):
            cv2.imwrite(path, frame);
            result = predict(path);
            cv2.imshow(result, frame)

            if cv2.waitKey(25) & 0xFF == ord('q'):
                break
            return result

if __name__=="__main__":
    start()
```

CHAPTER 4

MODELLING AND ASSEMBLY

4.1 ROBOT MODEL

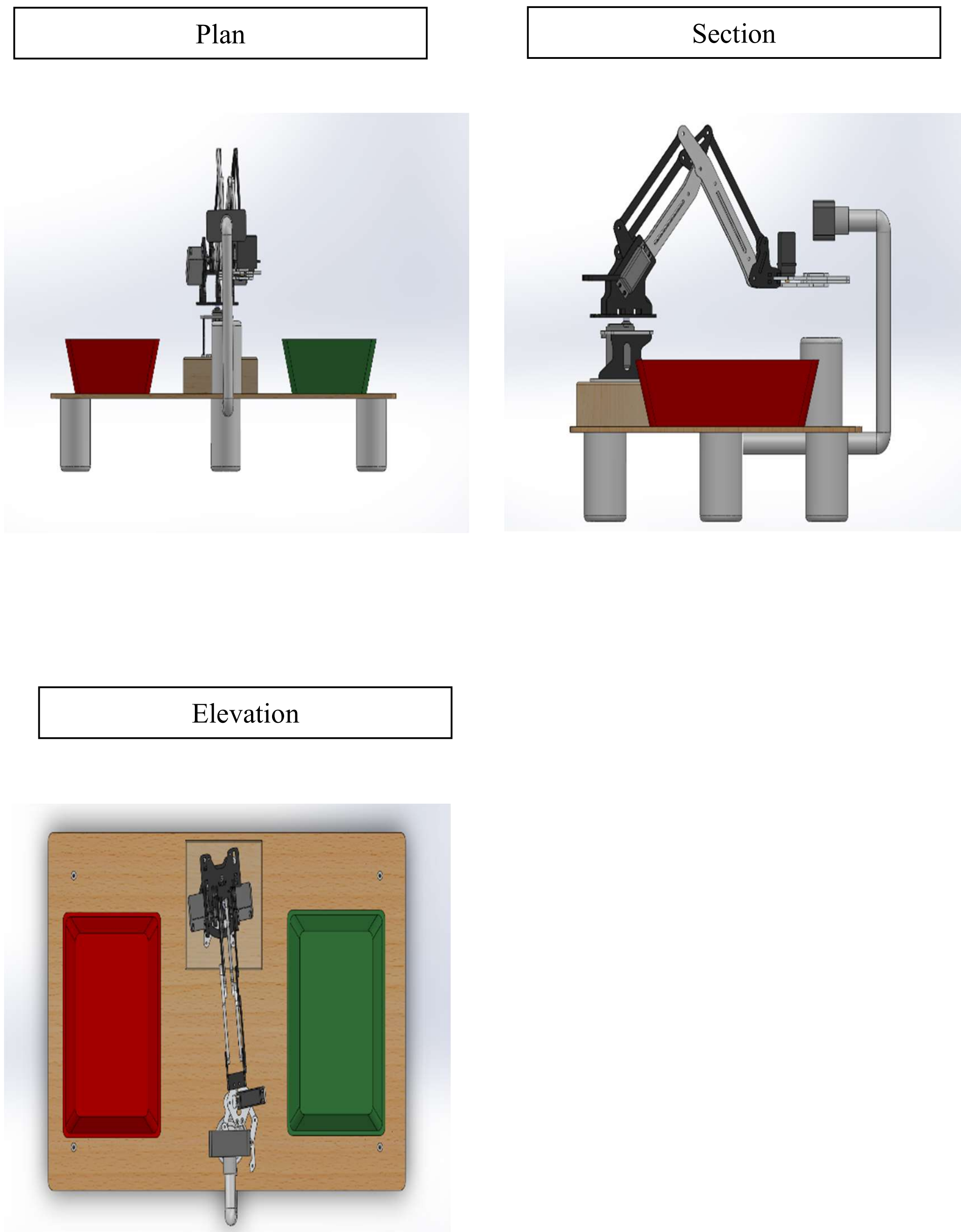


Fig 4.1 Robot Model

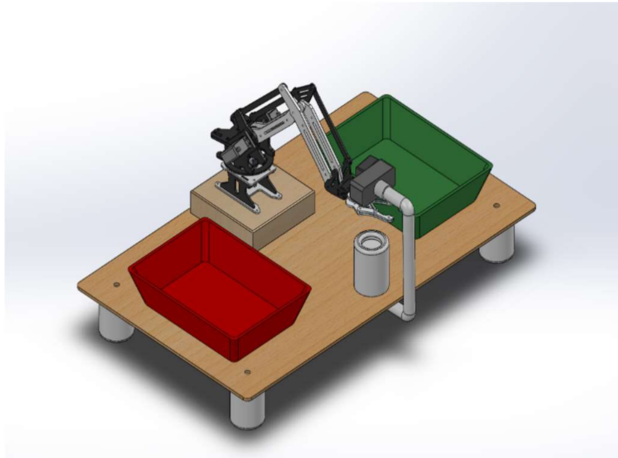


Fig 4.2 Isometric projection

4.2 ASSEMBLY OF THE ROBOT

The robot assembly is divided into three stages. The first stage is where the supporting components like the bushes, wooden block, PVC pipes are attached to the base board. Four bushes are attached to the back side of the base board and one bush is given for the fruit. Wooden block provides a support for the robot to minimize the moment while rotating. PVC pipes are attached as a supporting structure for the camera. The second stage is where the robot, camera, trays are attached to the supporting structures. The third stage is where the supply and electrical components like Arduino board, Step Down DC-DC converter, battery are attached to the base board and the connections are given.

Fig 4.3 Robot Assembly

CHAPTER 5

RESULTS AND DISCUSSION

5.1 PROJECT RESULT

We have achieved the process of detecting the defective lemon by using the deep learning algorithm named Inception V3. We have assembled the pick and place robot to do the cyclic process of collecting the defective lemon and segregate it. To achieve the desired results we followed the stage-by-stage process as mentioned in Fig 5.1

Assembly: Components have been attached as per the requirement

⇩

Integration: A link between Deep Learning and Arduino is established using pyFirmata

⇩

Operation: Robot is operated based on the condition specified from DL Result value

⇩

Testing & Results: Real time simulation and testing is carried out successfully based on Normal or Defective Fruit.

Fig 5.1 Work phases of project

5.2 FRUIT IDENTIFICATION RESULT

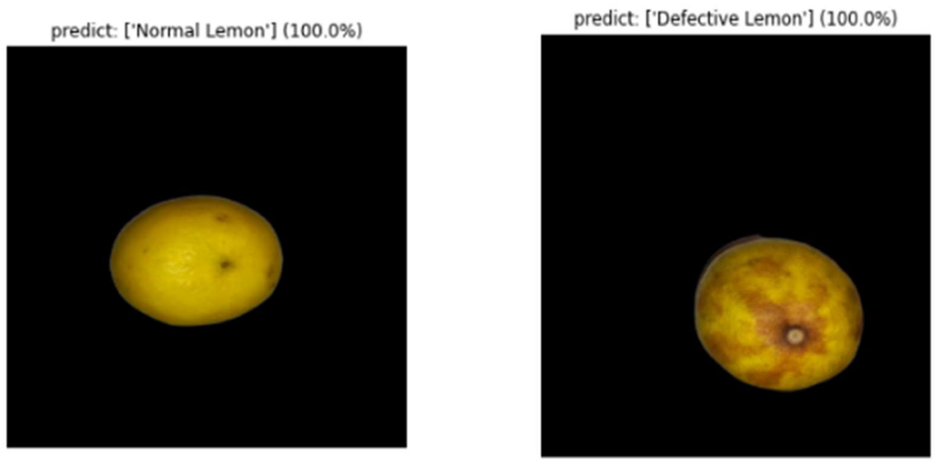


Fig 5.2 Testing result

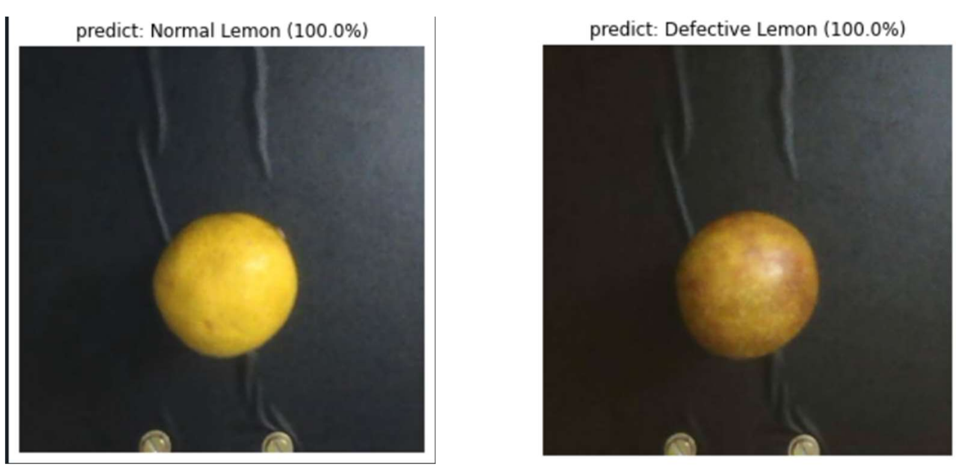


Fig 5.3 Real time result

The Deep Learning Algorithm that we trained retrieve the features from the whole image and not the specific area of interest (defective texture part of product).The results are shown in Fig 5.2 and Fig 5.3.

5.3 ROBOT CYCLE TIME

We have calculated the cycle time of the robot as and the value is taken as mean of many operation cycles of the robotic arm as mentioned in Table 5.1.

Table 5.1 Cycle Time of Robot

Total Cycle Time	40s
Time Taken for Normal Fruit Segregation	17s
Time Taken for Defective Fruit Segregation	17s

5.4 DISCUSSION

The dataset contains nearly 150 sample images and we increased the size of dataset by using data augmentation techniques. In this project the Inception-V3 algorithm is used to acquire the better accuracy in image classification. We achieved low accuracy at first when we train the algorithm with the dataset available from the internet and not worked well in real time. To overcome this challenge we took the images in real time using the camera and included the real time captured images also. Then the model reached the high accuracy.

CHAPTER 6

CONCLUSION

6.1 EXPECTED OUTCOME OF THE PROJECT

First the lemon is subjected to identification whether the lemon is defective or not. If the lemon is found to be defective it will be picked up by the articulated robot from the location specified and releases the lemon at the location specified without dropping it.

6.2 ACTUAL OUTCOME OF THE PROJECT

As per the expected outcome of the project the lemon is segregated according to the type of the lemon (defective or normal). The robot works as expected and the deep learning algorithm is deployed successfully.

6.3 SCOPE OF THE PROJECT

The future plans are to upgrade the robot to the next level where the lemon is moving in the conveyor and we will be trying to segregate from it. It will be deployable in a large scale.

REFERENCES

1. Chikhaoui, M.T., Rabenorosoa, K., Andreff, N.,2016, ‘Kinematics and performance analysis of a novel concentric tube robotic structure with embedded soft micro-actuation’Mechanism and Machine Theory, 104, pp. 234-254

2. Design and Control of 4-DOF Robotic-Arm Simultaneously using Matlab and Arduino(2019), Design and Control of 4-DOF Robotic-Arm Simultaneously using Matlab and Arduino’- The 4th International under-Graduate Research Conference, IUGRC 2019, July 29th -August 1st, 2019

3. Dr. A.Anushya (2020), ‘Quality Recognition of Banana Images using Classifiers’- Journal of Computer Science and Information Technology

4. Hamza Alzarok, Simon Fletcher, Naeem S. Mian(2020), ‘Review for the current performances for Machine Visions in Industrial Robotic welding and drilling tasks’- Journal of Electrical and Electronics Engineering

5. K. E. Clothier and Y. Shang, ‘A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation’, J. Robot., 2010

6. Mohammad M. Aref, Jouni Mattila(2018), ‘Deep Learning of Robotic Manipulator Structures by Convolutional Neural Network’- 2018 Ninth International Conference on Intelligent Control and Information Processing

7. Narendra V G(2021), ‘Defects Detection in Fruits and Vegetables Using Image Processing and Soft Computing Techniques’- 6th International Conference on Harmony Search

8. Shahriar Rahman Fahim, Md. Rashiduzzaman, Sanjay Dey, Md. Sakhawat Hosen, Subrata K. Sarker, and Sajal K. Das(2019), 'Modeling and Development of a Five DoF Vision Based Remote Operated Robotic Arm'-

9. Tatiana Victorovna Zudilova, Sergei Evgenievich Ivanov(2016), 'Mathematical Modeling of the Robot Manipulator with Four Degrees of Freedom'- Global Journal of Pure and Applied Mathematics

APPENDIX 1

CODES:

 # Robot Code

```
from pyfirmata import Arduino, SERVO
import time
base_servo = 3
left_servo = 5
right_servo = 6
end_effector = 9
board = Arduino("com7")
board.digital[base_servo].mode=SERVO
board.digital[left_servo].mode=SERVO
board.digital[right_servo].mode=SERVO
board.digital[end_effector].mode=SERVO
def servo1_rotate(base_servo,angle):
    board.digital[base_servo].write(angle)
    time.sleep(0.01)
def servo2_rotate(left_servo,angle):
    board.digital[left_servo].write(angle)
    time.sleep(0.01)
def servo3_rotate(right_servo,angle):
    board.digital[right_servo].write(angle)
    time.sleep(0.25)
def end_rotate(end_effector,angle):
    board.digital[end_effector].write(angle)
    time.sleep(0.25)
'''Here in the below code snippet, we define a function for the robot to
operate. We provide angles to each of the joints based on the sequence of
```

operation needed to be done. This includes, traversing to mean position which in our case is 55 degrees. Here the subject is picked by the robot and held firmly by the end effector’’’

```
def operation():
    servo2_rotate(left_servo, 90)
    time.sleep(2)
    servo3_rotate(right_servo,160)
    time.sleep(2)
    servo1_rotate(base_servo,55)
    time.sleep(1)
    end_rotate(end_effector, 160)
    time.sleep(1)
    servo3_rotate(right_servo,120)
    time.sleep(1)
    servo2_rotate(left_servo, 70)
    time.sleep(1)
    end_rotate(end_effector, 80)
    time.sleep(1)
    servo2_rotate(left_servo, 90)
    time.sleep(1)
    servo3_rotate(right_servo,160)
    time.sleep(1)
def input_robot(x):
    if x=='S':
        operation()
    elif x=='C':
        for i in range(60,120, 1):
```

```
        servo1_rotate(base_servo, i)
    time.sleep(2)
    servo3_rotate(right_servo,130)
    time.sleep(1)
    end_rotate(end_effector, 150)
    time.sleep(1)
elif x=='A':
    for i in range(60,0, -1):
        servo1_rotate(base_servo, i)
    time.sleep(2)
    servo3_rotate(right_servo,130)
    time.sleep(1)
    end_rotate(end_effector, 150)
    time.sleep(1)
elif x=='B':
    servo1_rotate(base_servo, 0)
    time.sleep(1)
    servo3_rotate(right_servo,170)
    time.sleep(1)
elif x=="Normal Lemon":
    operation()
    for i in range(55,120, 1):
        servo1_rotate(base_servo, i)
    time.sleep(2)
    servo3_rotate(right_servo,130)
    time.sleep(1)
    end_rotate(end_effector, 150)
    time.sleep(1)
```

```
    elif x=="Defective Lemon":
        operation()
        for i in range(55,0, -1):
            servo1_rotate(base_servo, i)
        time.sleep(2)
        servo3_rotate(right_servo,130)
        time.sleep(1)
        end_rotate(end_effector, 150)
        time.sleep(1)
print('---------Done-----------')
```

 # Main File

```
import os
os.chdir("D:\E drive copy\REC\Projects\Final Year Project\Phase 2\Python code")
from Deep_Learning_Testing import *
from Robot_Code_1 import *
import time
normal_count=2
defective_count=2
while True:
    result = start()
    if result == 'Normal Lemon':
        normal_count-=1
        input_robot('Normal Lemon')
        time.sleep(3)
    elif result == 'Defective Lemon':
        defective_count-=1
        input_robot('Defective Lemon')
```

```
        time.sleep(3)
    if normal_count==0 or defective_count==0:
        break
print("----Operation Done Successfully!------")
```

Deep Learning Training Results:


```
from google.colab import drive
drive.mount('/content/drive')

     Mounted at /content/drive


import os
import zipfile
import random
import shutil
import numpy as np
from shutil import copyfile

%matplotlib inline
import matplotlib.image as mpimg
import matplotlib.pyplot as plt

import tensorflow as tf
from google.colab import files
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator


def make_dir(PATH):
  if not os.path.exists(PATH):
      os.mkdir(PATH)
      return PATH
  else:
    shutil.rmtree(PATH)
    os.mkdir(PATH)
    return PATH

train_datagen = ImageDataGenerator(
    rescale=1./255,
    width_shift_range=0.2, #0.2, 0.5
    height_shift_range=0.2, #0.2, 0.5
    shear_range=0.2,
    zoom_range=[0.5, 1.0], # 0.2, 0.5, [0.5,1.0]
    rotation_range=90, #20, 40, 60, 90
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='reflect' #nearest, reflect, wrap
)

img_dir = '/content/drive/MyDrive/Deep_Learning/Deep_Learning/Colab Notebooks/Lemon_dataset'

train_generator = train_datagen.flow_from_directory(img_dir,
                                                    batch_size=32,
                                                    color_mode="rgb",
                                                    # shuffle = False,
```

```
 2D)

 flatten (Flatten)          (None, 36992)             0

 dense (Dense)              (None, 512)               18940416

 dense_1 (Dense)            (None, 6)                 3078

=================================================================
Total params: 19,036,742
Trainable params: 19,036,742
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras import layers
from tensorflow.keras import Model
import pandas as pd

pre_trained_model = InceptionV3(input_shape=(150,150,3),
                                include_top=False)

for layer in pre_trained_model.layers:
    layer.trainable = False

x = layers.Flatten()(pre_trained_model.output)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.2)(x)
x = layers.Dense(2, activation='softmax')(x)

model = Model(pre_trained_model.input, x)
model.compile(optimizer='adam', #RMSprop(lr=0.0001), adam
              loss='categorical_crossentropy',
              metrics=['accuracy'])

pd.set_option('max_colwidth', None)
layers = [(layer, layer.name, layer.trainable) for layer in pre_trained_model.layers]
pd.DataFrame(layers, columns=['Layer Type', 'Layer Name', 'Layer Trainable'])
```

```
                                                target_size=(150,150), #?
                                                class_mode='categorical')
```

```
Found 117 images belonging to 2 classes.
```

```
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('accuracy') > 0.98):
      print("\nReached 98% accuracy. Stop Training")
      self.model.stop_training = True

callbacks = myCallback()


model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),
    # tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    # tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(6, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-appl
87916544/87910968 [==============================] - 1s 0us/step
87924736/87910968 [==============================] - 1s 0us/step
```

	Layer Type	Laye
0	<keras.engine.input_layer.InputLayer object at 0x7ff5bfcaff50>	
1	<keras.layers.convolutional.Conv2D object at 0x7ff5bec48210>	co
2	<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7ff5be8ba0d0>	batch_norma
3	[illegible]	[illegible]

```
train_len = 0
for foldername in os.listdir(img_dir):
    train_len = train_len + len(os.listdir(os.path.join(img_dir,foldername)))

print(train_len)
```

```
117
```

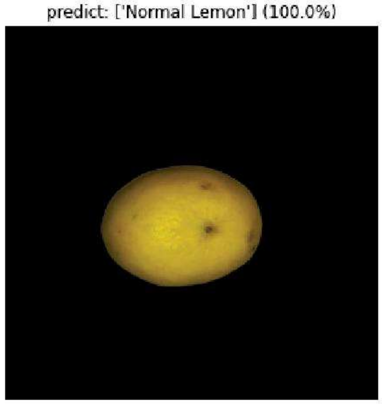
308	<keras.layers.merge.Concatenate object at 0x7ff5b0876e10>	concate

```
history = model.fit(
    train_generator,
    steps_per_epoch=(train_len/32),
    epochs=50,
    verbose=1,
    callbacks=[callbacks],

)
```

```
Epoch 1/50
3/3 [==============================] - 36s 5s/step - loss: 32.6654 - accuracy: 0.5470
Epoch 2/50
3/3 [==============================] - 1s 396ms/step - loss: 4.3558 - accuracy: 0.7607
Epoch 3/50
3/3 [==============================] - 2s 458ms/step - loss: 2.1040 - accuracy: 0.8889
Epoch 4/50
3/3 [==============================] - 1s 387ms/step - loss: 1.6552 - accuracy: 0.9402
Epoch 5/50
3/3 [==============================] - 2s 445ms/step - loss: 0.6766 - accuracy: 0.9145
Epoch 6/50
3/3 [==============================] - 2s 452ms/step - loss: 0.4252 - accuracy: 0.9573
Epoch 7/50
3/3 [==============================] - 2s 403ms/step - loss: 0.5757 - accuracy: 0.9744
Epoch 8/50
3/3 [==============================] - 2s 398ms/step - loss: 0.2186 - accuracy: 0.9658
Epoch 9/50
3/3 [==============================] - 2s 446ms/step - loss: 0.5283 - accuracy: 0.9402
Epoch 10/50
4/3 [========================================] - ETA: 0s - loss: 0.1500 - accuracy: 0.9915
Reached 98% accuracy. Stop Training
3/3 [==============================] - 2s 396ms/step - loss: 0.1500 - accuracy: 0.9915
```

```
model.save('model.h5')

path = '/content/drive/MyDrive/Deep_Learning/Deep_Learning/Colab Notebooks/Lemon_dataset/Heal

img = image.load_img(path, color_mode="rgb", target_size=(150, 150), interpolation="nearest")
# imgplot = plt.imshow(img)
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
img = img/255

images = np.vstack([img])
classes = model.predict(images, batch_size=1)
predict_result = []

max = np.amax(classes[0])
if np.where(classes[0] == max)[0] == 0:
  predict_result.append('Normal Lemon')
elif np.where(classes[0] == max)[0] == 1:
  predict_result.append('Defective Lemon')

plt.figure(figsize=(5, 5))
plt.imshow(image.load_img(path, color_mode="rgb", target_size=(1080, 1080), interpolation="ne
title = f"predict: {predict_result} ({round(float(max)*100, 2)}%)"
plt.title(title, color='black')
plt.axis('off')
plt.show()
```

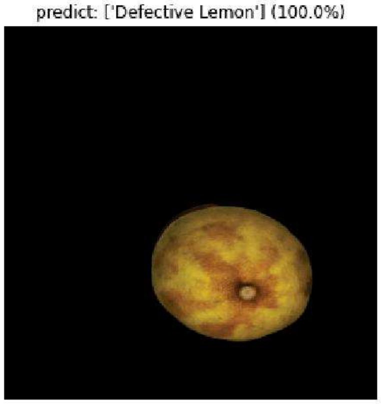
predict: ['Normal Lemon'] (100.0%)



```
path = '/content/drive/MyDrive/Deep_Learning/Deep_Learning/Colab Notebooks/Lemon_dataset/Unhe

img = image.load_img(path, color_mode="rgb", target_size=(150, 150), interpolation="nearest")
# imgplot = plt.imshow(img)
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
img = img/255
```

```
img = img/255

images = np.vstack([img])
classes = model.predict(images, batch_size=1)
predict_result = []

max = np.amax(classes[0])
if np.where(classes[0] == max)[0] == 0:
  predict_result.append('Normal Lemon')
elif np.where(classes[0] == max)[0] == 1:
  predict_result.append('Defective Lemon')

plt.figure(figsize=(5, 5))
plt.imshow(image.load_img(path, color_mode="rgb", target_size=(1080, 1080), interpolation="ne
title = f"predict: {predict_result} ({round(float(max)*100, 2)}%)"
plt.title(title, color='black')
plt.axis('off')
plt.show()
```

predict: ['Defective Lemon'] (100.0%)



RAJALAKSHMI ENGINEERING COLLEGE
(An Autonomous Institution, Affiliated to Anna University Chennai)
DEPARTMENT OF MECHATRONICS ENGINEERING
CURRICULUM AND SYLLABUS REGULATIONS – 2017
B.E.MECHATRONICS ENGINEERING

VISION:

To attain excellence in academics, research, and technological advancement in Mechatronics Engineering with a concern for society.

MISSION:

- To impart high-quality professional education and produce Mechatronics Engineers with all-around knowledge of multi-disciplinary branches of engineering and technology.
- To foster skill sets required to be a global professional in the areas of automation, intelligent systems, robotics, and research for technology management and to fulfill the expectations of industry and the needs of society.
- To inculcate entrepreneurial qualities for creating, developing, and managing global engineering ventures.

Programme Educational Objectives (PEOs):

PEO I

Graduates will have comprehensive knowledge in the analytical, scientific, and engineering fundamentals necessary to model, analyse and solve engineering problems and to prepare them for graduate studies and for successful careers in the industry.

PEO II

Graduates will effectively design and develop products in the areas such as automation, manufacturing, Internet of Things, machine vision, system simulation, intelligent systems and robotics.

PEO III

Graduates will acquire Technical expertise, Leadership skills, Ethical practices and Team spirit with a concern towards greener society.

PROGRAM OUTCOMES (POs):

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

Engineering Graduates will be able:

PSO 1: To innovate a Mechatronics system to meet the requirements and specifications.

PSO 2: To analyse and improve the performance of a Mechatronics system and enhance the intellectual capabilities of the system

PSO 3: To lead professional career in industries or an entrepreneur by applying Engineering and Management principles and practices.

MT17811 **PROJECT WORK PHASE II** **L T P C**
0 0 16 8

OBJECTIVES:

- To develop the ability to solve a specific problem right from its identification and literature review till the successful solution of the same.
- To train the students in preparing project reports and to face reviews and viva-voce examination

OUTCOMES:

Upon completion of this course, the students will be able to

CO1: Ability to fabricate any components using appropriate manufacturing techniques
CO2: Use of design principles and develop conceptual and engineering design of any mechatronics component.
CO3: Demonstrating the function of the fabricated model
CO4: Ability to prepare the project as a technical report and deliver it in oral presentation
CO5: Ability to show their teamwork and technical Skills

CO - PO – PSO Matrices of Course

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)
If there is no correlation, put "-"

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1															
CO2															
CO3															
CO4															
CO5															
Average															