CHAPTER 4

MECHANISM IMPLEMENTATIONS & TESTING

4.1 Mechanism

The project work is considered incomplete without having a considerable working mechanism and its testing. In this chapter we are going to describe the different mechanism involved in the project and how they function in the project. It has different mechanisms involved and are mentioned below.

Various mechanism in the humanoid robot -

- Eye
- Hand
- Motion
- Line Following
- Machine Vision

The mechanisms mentioned plays a crucial role because to perform a particular task given, making the robot versatile, adaptable and more helpful. The mechanisms involved are explained below.

4.1.1 Eye Mechanism

The eye mechanism creates a motion similar to that of a human eye by using a series of links for the “X” and “Y” motion of the eyes. Two “SG90” Servo motors are used to accomplish the whole “X” and “Y” motion. The servo motor is as shown in the Fig. 4.1 below and the rear view of eyes with servo is shown in Fig 4.2.

Fig 4.1 SG90 Servomotor [44]

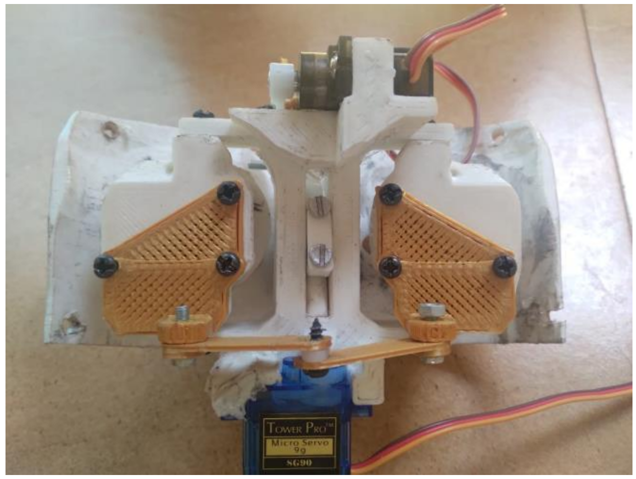


Fig 4.2 Rear view of eyes with Servo

The home position of each servo is initially set to 90° and each side has a buffer of 30° of motion; that is, the range of motion for each servo is defined as 60° to 120°. With the use of a camera in its left eye, the eye can follow the person in front of it. It detects the presence of a face using a face detection module, and it provides a global coordinate that is mapped to the servo between 60° and 120°. The angle provided is fed to a microcontroller via serial communication. It uses a set of different parts which are given in the Fig 4.3 below.

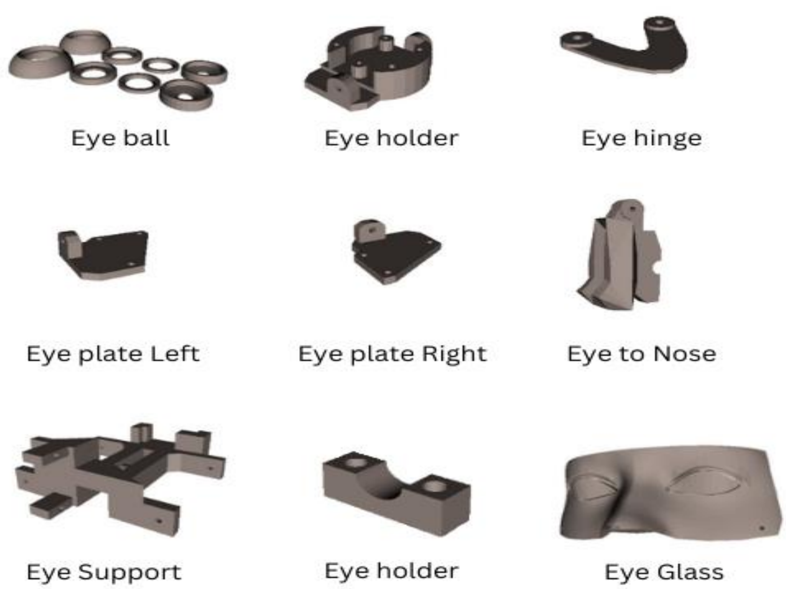


Fig 4.3 Eye mechanism parts used

The main purpose of this mechanism is to resemble animatronic look and mimic the human eye in facial and mechanism. With features like facial recognition and face tracking using OpenCV and python.

4.1.1.1 Calibration:

For calibration the servo first is set at at 90̊, so that a neutral position is achieved which can be then assembled in the slot. The below Fig 4.4 shows the interface of arduino with servomotor.

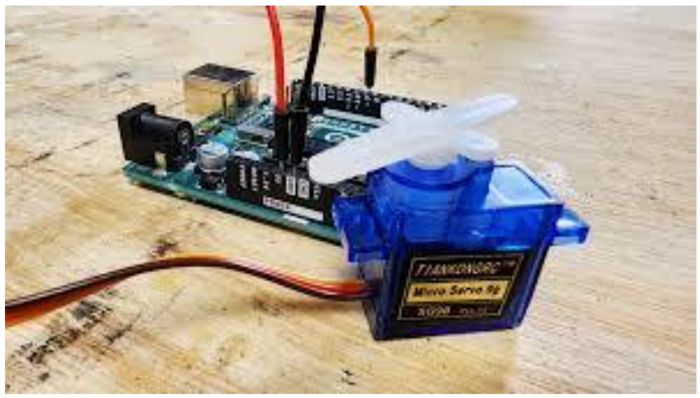


Fig 4.4 Arduino interface with servo

Manual coordinates are entered in the servo to check the maximum range of motion possible for the eye motion.

4.1.1.2 Eye mechanism program testing

To test the servo initially a joystick was used to move the servo in "X" and "Y" direction. The face tracking algorithm data is interpreted and mapped to the coordinates of the servo and tested.

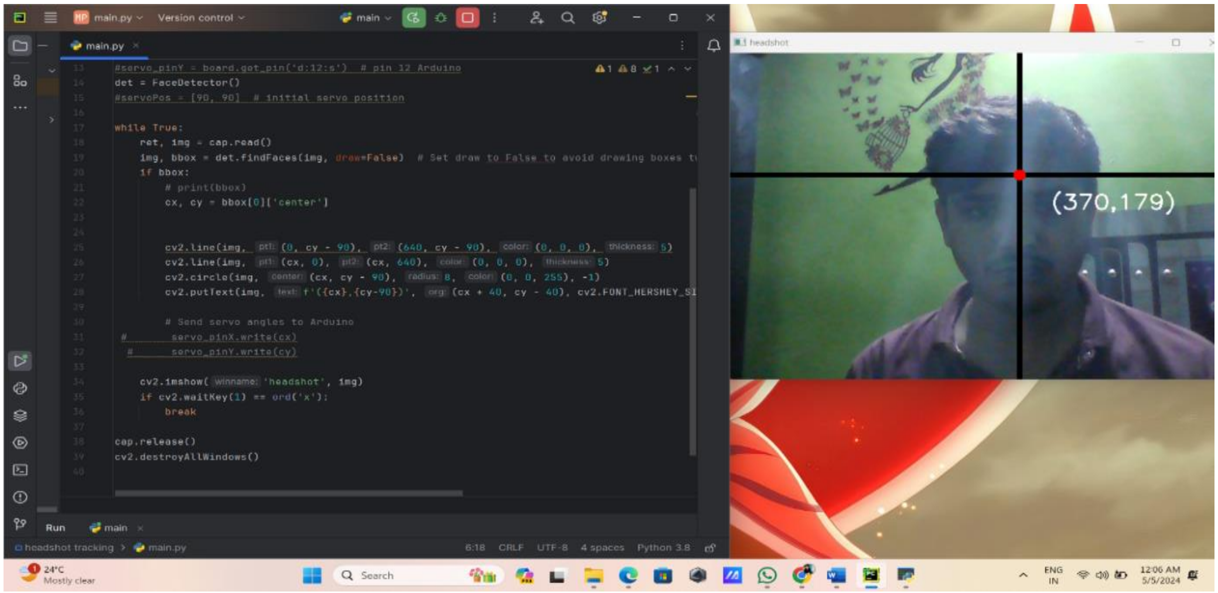
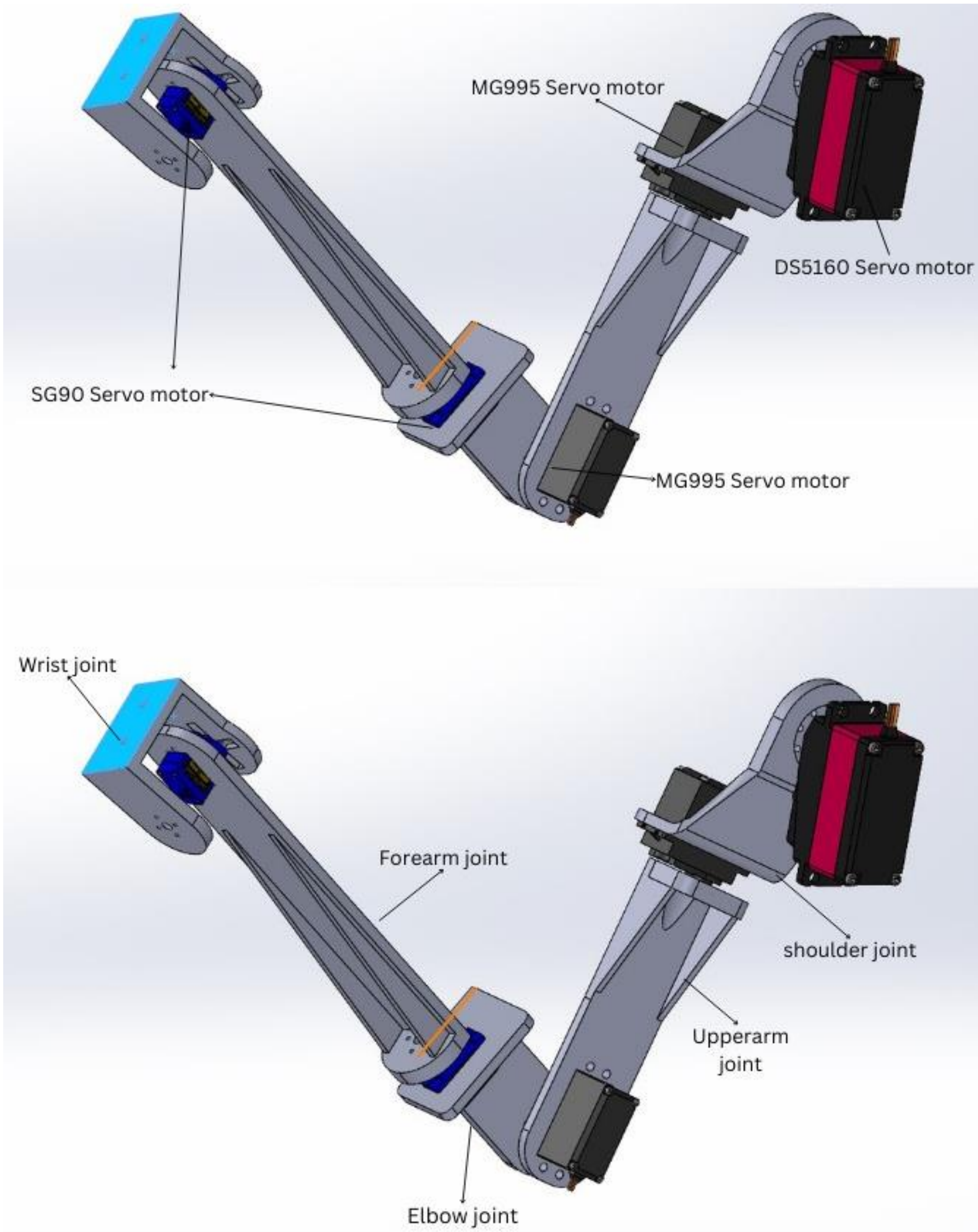


Fig 4.5 Python algorithm for face tracking

4.1.2 Hand Mechanism

The arm serves as the primary mechanism for hand motion and manipulation tasks within the robotic system the project. It provides necessary dexterity and range of motion to grasp objects, perform gestures and interact with the environment.



Fig

4.6 CAD image of hand

4.1.2.1 Technical specification of components used in the hand mechanism.

Number of Servo motors: 5

Types of joints: Exclusively revolute

Total Degrees of Freedom (DOF): 5

Table 4.1 Servo Motor Specification.

Types of servo motors used	Torque (in kgcm)	Type of motion	Joint connected	No of servo used(per arm)
1.) Ds6150 Servo motor 	60	Rotary	Shoulder joint	1
2.) MG995 Tower pro servo motor 	11	Rotary	Upper arm joint, elbow joint	2
3.) SG90 Servo motor 	1.6	Rotary	Forearm joint, wrist	2

4.1.2.2 Discussion of DOF

The DOF of the arm is known by calculating the DOF associated with each type of motion. The type of motions are -

- A rotary joint: It provides 1 DOF around 1 axis.
- A linear joint: It can provide up to 3 DOF depending on its configuration, enabling movement along 3 different axes.
- A Spherical Joint: It offers 3 DOF enabling movement along 3 perpendicular axes.

The total DOF can be calculated by summing the DOF provided by each joint type and multiplying by the number of joints of that type. In our case it can be considered as below

Number of Rotary joints: 5

Number of Linear joints: 0

Number of Spherical joints: 0

Therefore, the total DOF = (Number of Rotary Joints × DOF per Rotary Joint) + (Number of Linear Joints × DOF per Linear Joint) + (Number of Spherical Joints × DOF per Spherical Joint)

Total DOF = (5 X 1) + (0 X 0) + (0 X 0)

= 5+ 0 + 0

= 5

Therefore, the total degrees of freedom for the robotic arm is 5.

4.1.2.3 Controller: Arduino nano

To test the range of motion possible for each motor the test is conducted for each joint with a certain motion and assembled it which gives the most smooth and easiest motion.

Role of Arduino nano in the project:

The hand mechanism is a specially made part intended to enable five rotary motions necessary for the manipulation activities in the project. It is made up of a skillfully designed system with five linked links that work together to produce the desired motion sequences. The mechanism uses servo motors to provide precise control and dependability throughout its operation.

Fig 4.7 Arduino nano [45]

4.1.3 Forward Motion

The project's forward mobility mechanism is essential to the robot's ability to move around and navigate its surroundings effectively. It makes use of two wiper motors that are mounted at one end of the robot's chassis and are upheld by two front-facing caster wheels. Forward motion has been made possible by this system, and it additionally offers stability and maneuverability.

4.1.3.1 Motor’s used:

The project uses two wiper motors running at a rpm of 45 at 12V which are rated at 50W. They produce a No-load current at 1.8amp and 5.5 amp at load they have a working speed of 40 RPM.

They are attached with tyres of dia 100mm and 30mm thickness for better traction with the surface. With the help of these data collected there are certain factors which can be calculated which are given below

1. Running Speed (m/s):

To calculate the running speed of the robot we need the circumference of the tyres used

Circumference of the tyres = π X diameter = π X 0.1 meters **= 0.1π meters** -------**Eqn 1**

To calculate the speed in Kmph we need the distance it can travel in 1 hour.

Distance traveled per hour = Distance per minute $\times$ 60 = 75.39meters ≈ **75 meters** ------**Eqn 2**

Converting meters to kilometers: Distance travelled per hour / 1000

Running speed in km/h ≈ 75 / 1000 km/h = 0.075kmph = **0.02083m/s**

2. **Power consumption:**

The need to measure the power of each motor requires for it to perform without any issue which can be done as follows -

Power used when there is no load = Voltage $\times$ Current = 12V $\times$1.8A = **21.6W** ----**Eqn 3**

Power usage when we apply the load = Voltage $\times$ Current = 12V $\times$ 5.5A = **66W** -----**Eqn 4**

3. **Torque:**

Torque is a measure of the rotational force applied to an object, causing it to rotate around an axis. To calculate the torque of the motor in order to determine the maximum load carrying capacity -

The formula for torque is -

Torque (τ) = Power (P) / Angular Velocity (ω) ----- **Eqn 5**

Therefore, Angular Velocity (ω) = 2π $\times$ RPM / 60 (since RPM is in minutes) -----**Eqn 6**

Angular Velocity (ω) = (2π $\times$ 40) / 60 **≈ 4.188 rad/s** -----**Eqn 7**

Hence Torque (τ) = 50W / 4.188 rad/s ≈ **11.94 Nm** -----**Eqn 8**

4. Efficiency:

To calculate the effenciency of the motors we need the ratio of output power to the input power

Therefore, Efficiency = (Output Power / Input Power) × 100% **-----Eqn 9**

But Output Power is the mechanical power output by the motor, which is equal to the torque produced multiplied by the angular velocity.

i.e., Output Power = Torque × Angular Velocity **-----Eqn 10**

≈ 11.94 Nm × 4.188 rad/s

≈ **50W**

Input Power is the electrical power input to the motor.

i.e., input power= voltage × current **-----Eqn 11**

input power = 12×5.5

input power = **66 W**

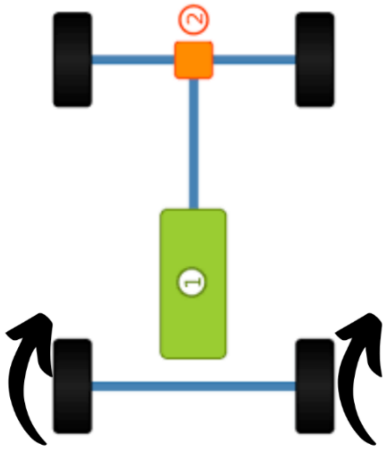
Hence, Efficiency = (50W / 66W) ×100%

≈ 75.76%

4.1.4 Direction control of the robot

To control the robot the principle used is 2-wheel drive. A 2-wheel drive or 2WD system that uses all the 4 wheels in which 2 wheels provide the motion and the other 2 wheels provides stability during the motion.

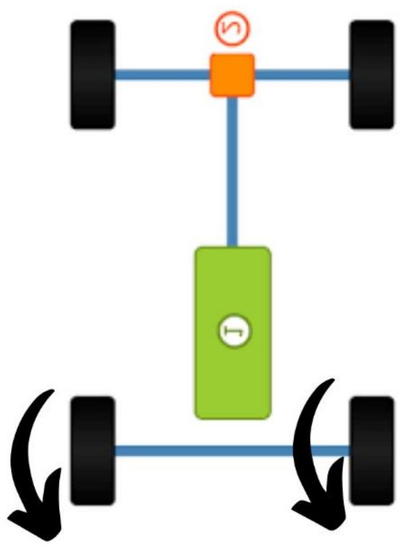
4.1.4.1 Forward direction motion: To achieve the robot's trajectory in forward direction the rotation of the motor in clockwise direction at constant speed is necessary. And is achieved with the help of the motor driver and a microcontroller.



Clockwise rotation

Fig 4.8 Forward direction motion [46]

4.1.4.2 Backward direction motion: To achieve the robot's trajectory in backward direction the rotation of the motor in anticlockwise direction at constant speed is necessary.



counterclockwise rotation

Fig 4.9 Backward direction motion [46]

4.1.4.3 Left direction motion: To achieve the robots trajectory in left direction the rotation of the right motor in clockwise direction and left motor in anticlockwise direction at constant speed is necessary.

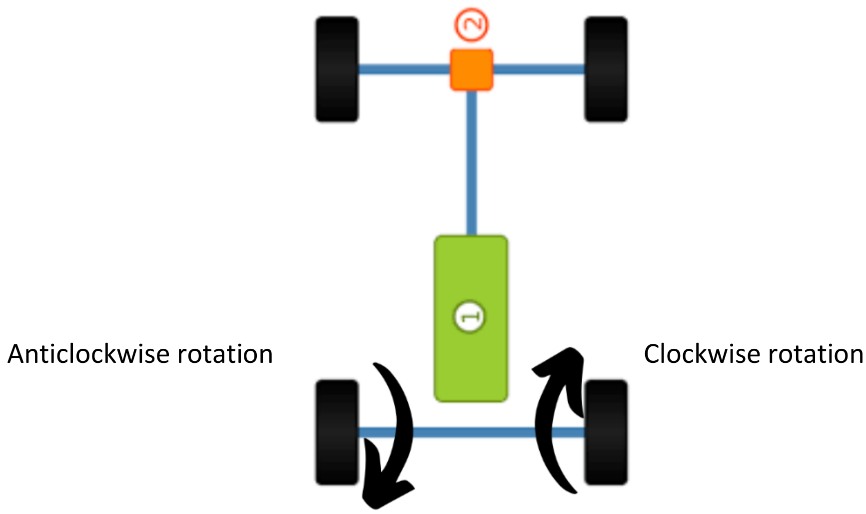


Fig 4.10 Left direction motion [46]

4.1.4.4 Right direction motion: To achieve the robot's trajectory in left direction the rotation of the left motor in clockwise direction and right motor in anticlockwise direction at constant speed is necessary.

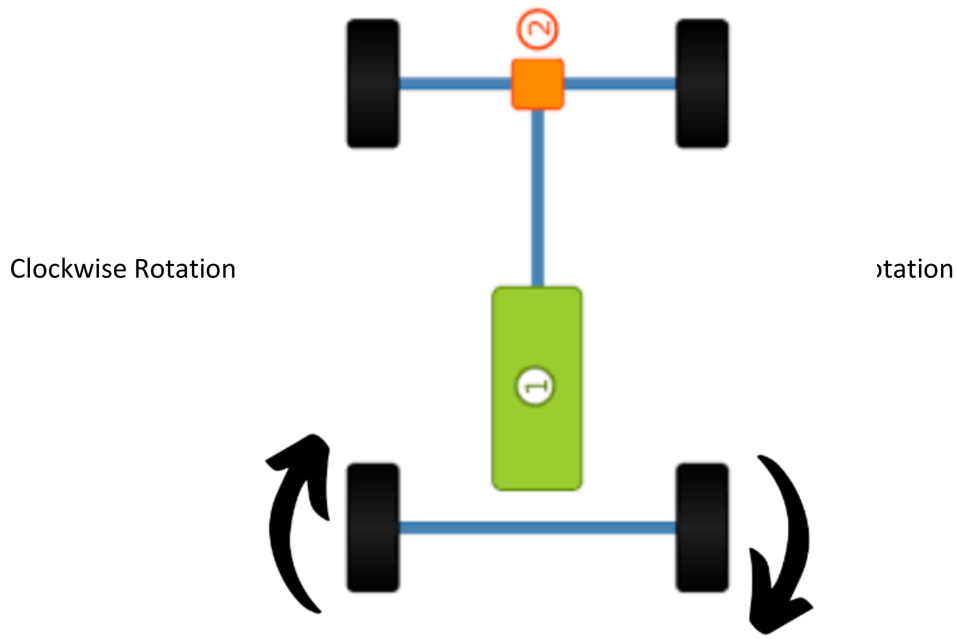


Fig 4.11 Right direction motion [46]

4.1.5 Face Recognition

The Face recognition is a biometric technique that uses face feature analysis to identify the people. There are multiple steps are involving in the procedure. First, facial landmark-identifying algorithms are used to detect faces within picture or video frames. Subsequently, distinct features are identified from the identified faces, frequently expressed as numerical information. Neural networks and machine learning algorithms are used to compare these traits against a database of recognized faces. The below fig 4. is show the flow chart of the face recognition. The likelihood of a match is determined based on how similar the extracted features are to the features in the database. Lastly, the comparison results are used to identify or validate the person's identity. Applications for face recognition include surveillance, access control, security systems, and personal device identification.

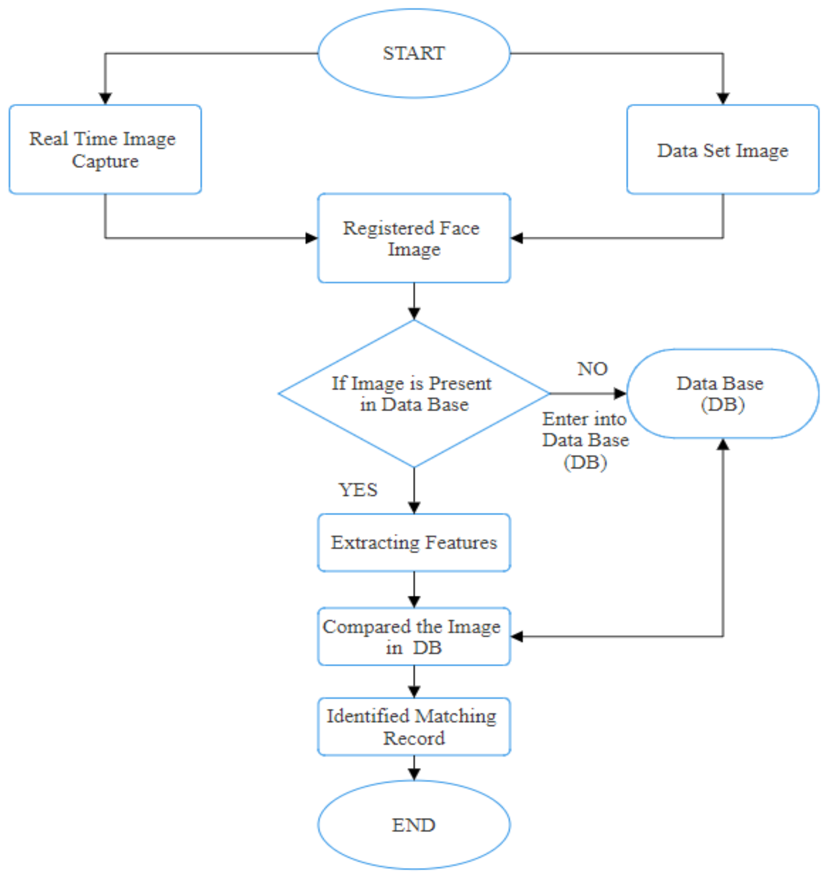


Fig 4.12 Flowchart of the facial recognition

The above Fig shows that how the face recognition is worked in order to create, practice, and implement face recognition systems. A number of tools and libraries are frequently used in face recognition processes.

Open CV (OpenSource Computer Vision Library): The OpenCV is a most popular open-source computer vision library that offers the number of features for processing images and videos, in this library's such as face recognition, feature extraction, and face detection.

Fig.4.13 OpenCV Library [47]

- **For python3:** Using this command, the OpenCV library for Python 3 is installed. It contains the OpenCV main package and all of its dependencies.

Copy code

```
pip install opencv-python
```

Fig.4.14 Code to install to OpenCV(Pycharm)

- **For additional OpenCV modules (optional):** You can install extra modules or features that go beyond the basic OpenCV library by using the following method.

Copy code

```
pip install opencv-contrib-python
```

Fig.4.15 Install extra modules (PyCharm).

TensorFlow: The TensorFlow it is created by Google for the open-source machine learning framework. The neural networks utilized in facial recognition systems are among the machine learning models that can be created, trained, and implemented using its tools.

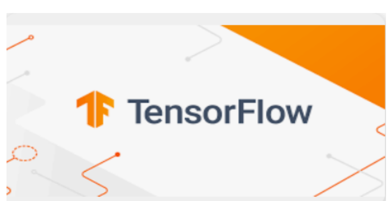


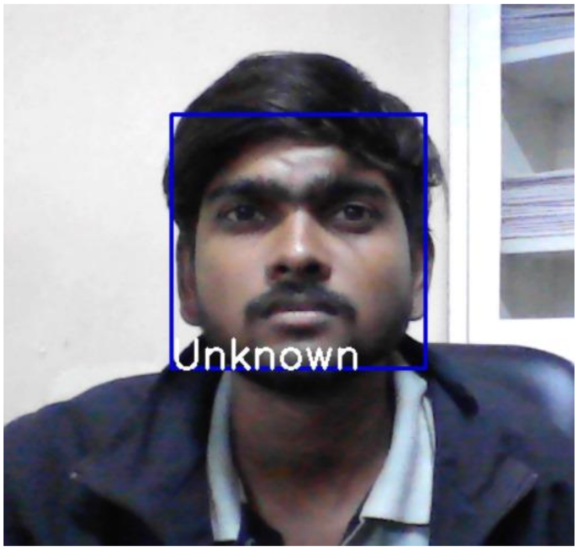
Fig.4.16 Machine learning model [48]

Face Recognition Library: There are several related face recognition's libraries are available, such as face recognition using Python library, which provides simple APIs for face detection, and face recognition tasks.

Fig.4.17 Face recognition [49]

Importance of facial recognition in the project:

1. User Identification: In user identification using facial recognition technology, the humanoid service robots can identify and recognize people through their data base. This feature increases user involvement by adjusting responses and services based on individual preferences and previous interactions.



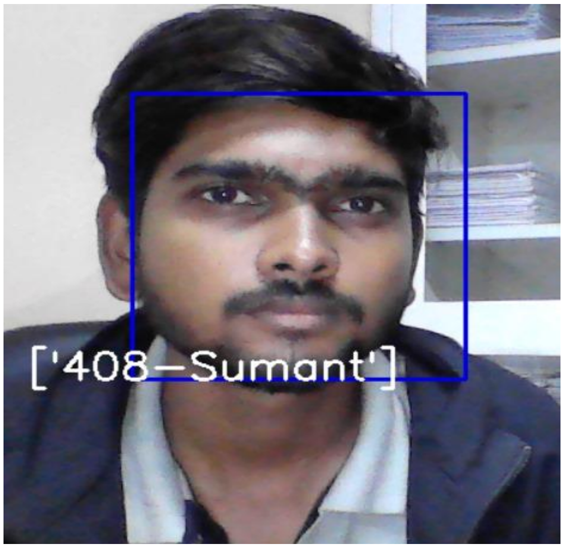


Fig 4.18 Face recognition before

Fig 4.19 Face recognition after

2.Customized Services: By utilizing facial recognition technology, humanoid service robots may provide customized services according to personal preferences or previous exchanges.

3.Testing techniques: In testing techniques to identifying and measure the facial features in the image or videos. In testing techniques there are many types of testing are used few of them are in the below.

4.Performance Testing: Performance testing checks how it is quickly and efficiently working with the system and uses its resources. This may be achieved by timing how long it takes the system to identify faces and examining how much memory it uses. Performance standards are set to make sure the system satisfies performance requirements.

5.Recognition accuracy: The capacity of a face recognition system to accurately identify or validate people based just on their facial traits is known as recognition accuracy. Usually, it is expressed as the proportion of accurate identifications or verifications among all efforts. When it comes to face recognition, the speed of recognition is the amount of time the system takes to recognize or authenticate a person based just on their facial features. It is an essential component of system performance, especially for real-time applications where prompt action is needed.

4.1.6 Line Following

The line-following mechanism detects the existence of a black line on the ground by means of three infrared (IR) sensors that are positioned strategically along the robot's chassis. The intensity of light reflected back from the surface is measured by these sensors, which emit infrared light. The mechanism identifies where the line is in relation to the robot's trajectory by assessing changes in the intensity of reflected light, and then modifies the robot's motion accordingly. Alignment with the line is maintained by the robot's movement thanks to control algorithms on the ESP32 microcontroller that analyze sensor data.

4.1.6.1 Turning in line following mode:

The turning of the robot in line following mode is controlled by the data received by from the IR sensors. If the right sensor is on the black line and left sensor is on the bright surface then it detects and understands that it needs to turn right and

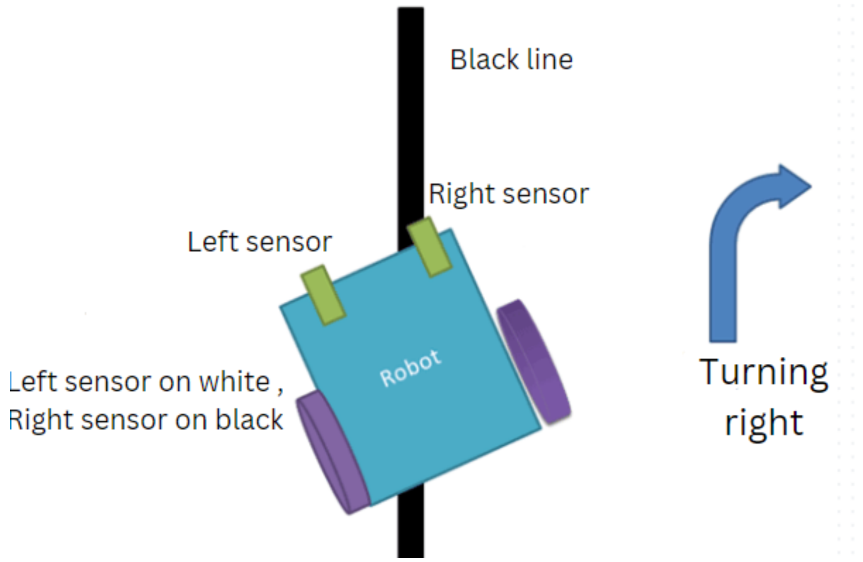


Fig 4.20 Robot turning right [50]

,Similarly if the left sensor is on the black line and right sensor is on the bright surface then it detects and understands that it needs to turn left and

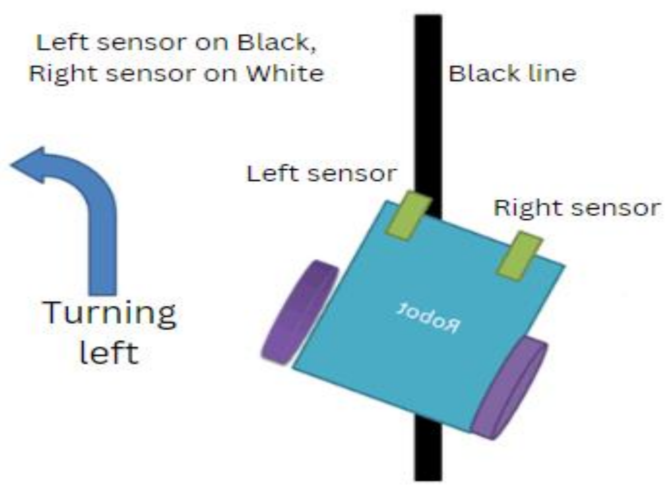


Fig 4.21 Robot turning left [50]

And if both the sensor are on the bright surface then there is no need for turning and it follows a straight path.

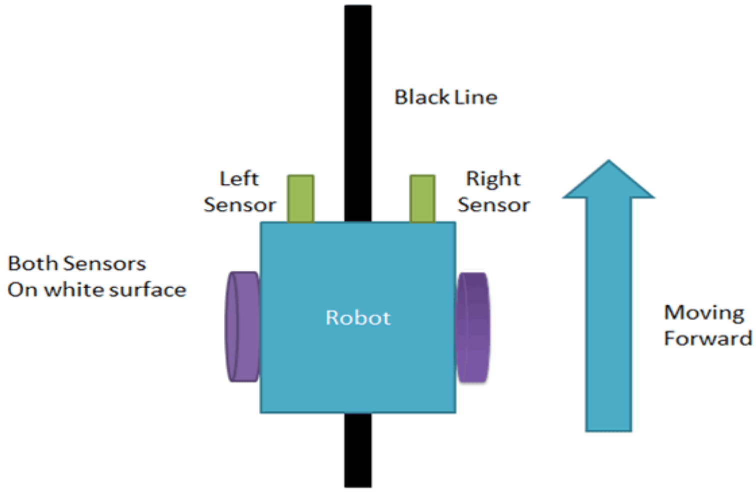


Fig 4.22 Robot moving forward [50]

It was crucial to change the sensor's height above the ground and ensure that each one was set to active low in order to test the line following mechanism's accuracy. When a black line is detected, the sensor sends a high pulse to the controller, which uses the PID algorithm to steer the robot's trajectory and keep the center sensor on the black line while the other sensors stay off of it.