

# SELF DRIVING CAR USING AI AND COMPUTER VISION

Submitted in partial fulfilment of the requirements for the award of  
Bachelor of Engineering degree in  
Electronics and Instrumentation Engineering

by

**C. Benoni Isaac Raj (Register Number: 38180001)**

**Bernard. A (Register Number: 38180002)**



**DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION  
ENGINEERING**

**SCHOOL OF ELECTRICAL AND ELECTRONICS**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI 600 119**

**MARCH-2022**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



## **DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of C. Benoni Isaac Raj (Register Number: 38180001) and Bernard. A (Register Number: 38180002) who carried out the project entitled "**SELF DRIVING CAR USING AI AND COMPUTER VISION**" under our supervision from October 2021 to March 2022.

#### **Internal Guide**

Dr.N.R Krishnamoorthy M.E., Ph.D.,

#### **Head of the Department**

Dr.S.LalithaKumari, M.E., Ph.D.,

**Submitted for Viva voce Examination held on**

Name:

Signature:

**Internal Examiner**

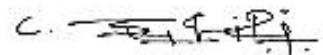
**External Examiner**

## DECLARATION

We C. Benoni Isaac Raj (Register Number: 38180001) and Bernard. A (Register Number: 38180002) declare that the project report entitled "**SELF DRIVING CAR USING AI AND COMPUTER VISION**" done by us under the guidance of Dr.N.R Krishnamoorthy M.E., Ph.D., at sathyabama institute of Science and Technology, Chennai-600119 is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Instrumentation Engineering.

DATE: 09.05.2022

PLACE: Chennai



**C. Benoni Isaac Raj (Register Number: 38180001)**



**Bernard. A (Register Number: 38180002)**

## **ACKNOWLEDGEMENT**

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. S. LALITHAKUMARI M.E.,Ph.D.**, Head of the Department, Department of Electronics and Instrumentation Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would also like to express our sincere and deep sense of gratitude to our internal guide **Dr. N. R Krishnamoorthy M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Electrical and Instrumentation Engineering who were helpful in many ways for the completion of the project.

## **ABSTRACT**

The goal of the proposed system is to represent the Autonomous model that uses the Raspberry Pi as the main working chip and our system uses Open CV with Artificial intelligence and deep learning technology. OpenCV (Open-Source Computer Vision Library) is an open-source computer vision software library. It was built to provide a common infrastructure for computer vision applications and to speed up the use of machine perception in commercial products. The proposed system will move on its own without human assistance to its destination. The model uses a central processing system as the Raspberry Pi, connected to the Pi camera module. The camera will give the input images for the processes. The whole process is completed using the Raspberry Pi and OpenCV with python language. The methods used to achieve the independent movement of the robot are canny edge detection algorithm. The robot with Raspberry Pi, Esp32, camera, LCD display, motors and wheels are being attached to the chassis and the required circuit connections are made. This system will be of assistance to the automotive industry as it will help to reduce the required surveillance and tension of the situation, thereby reducing the risk. This model is mainly used in industries, which helps in transportation and delivery of verities of goods from one point to another point. It also does the work faster than humans and ease the process.

# CONTENT

Chapter	Title	Page No.
	Abstract	v
	List of figures	ix
	List of abbreviation	xi
1	Introduction	1
2	Literature survey	2
3	Aim and scope of the present investigation	6
3.1	Aim	6
3.2	Existing system	6
3.3	Proposed system	7
4	Experimental or materials and methods, algorithms used	8
4.1	Methods and algorithm	8
4.2	Machine learning	8
4.3	Canny edge detector algorithm	9
4.4	Hardware requirements	9
4.4.1	Introduction to Raspberry Pi	10
4.4.2	Raspberry Board - Model A	10
4.4.2.1	Features	11
4.4.3	Raspberry Board - Model B	11
4.4.3.1	Features	11
4.4.4	Types of Raspberry Pi	12
4.4.5	Raspberry Pi 3 Model B+	12
4.4.6	Introduction to Raspberry Pi 3 B+	14

4.4.6.1	Improvements	15
4.4.6.2	Raspberry Pi 3 B+ Pinout	15
4.4.6.3	Hardware Specifications	16
4.4.6.4	Raspberry Pi 3 B+ Technical Specifications	17
4.4.6.5	Applications	18
4.4.7	Ultrasonic Sensor	18
4.4.7.1	HC-SR04 Ultrasonic Sensor Pin Configuration	20
4.4.7.2	Features	20
4.4.7.3	HC-SR04 Ultrasonic Sensor Working	21
4.4.8	Raspberry Pi Camera Module	22
4.4.9	L298N Motor Driver Board	23
4.4.9.1	Hardware Diagram	24
4.4.9.2	H-Bridge Dc Motor Control	26
4.4.10	DC Motors	26
4.4.10.1	Working Principle of A DC Motor	27
4.4.10.2	Fleming's Left Hand Rule	27
4.4.10.3	Back Emf	28
4.4.10.4	DC Gear Motor	28
4.4.11	Esp32-Wroom	29
4.4.11.1	Esp32 Peripherals Features	31
4.4.12	7-inch HDMI LCD	31
4.4.12.1	Specifications	32
4.5	Software Requirements	33

	4.5.1	OpenCV	33
	4.5.2	Python	34
	4.5.2.1	Advantages of the Python Programming Language	35
	4.5.3	Embedded C	36
5		Result and discussions, performance analysis	38
	5.1	Model training in machine learning	39
	5.2	Model in machine learning	39
	5.2.1	Defining the Problem	39
	5.2.2	Data collection	39
	5.2.3	Preparing the Data	39
	5.2.4	Assigning appropriate model / protocols	40
	5.2.5	Training the Machine Model	40
	5.2.6	Evaluating and defining measure of success	40
	5.2.7	Parameter tuning	40
	5.3	Decision tree in AI	41
	5.4	Program flow in OpenCV	41
6		Conclusion	50
		References	51

## LIST OF FIGURES

<b>FIGURE No.</b>	<b>DESCRIPTION</b>	<b>PAGE No.</b>
3.1	Existing System	6
3.2	Proposed System	7
4.1	Raspberry Pi	13
4.2	Raspberry Pi 3 B+ description	14
4.3	Raspberry Pi pin diagram	16
4.4	Ultrasonic sensor	19
4.5	Ultrasonic sensor working	19
4.6	Ultrasonic sensor description	22
4.7	Raspberry Pi Camera Module	23
4.8	L298N Motor Driver Board	24
4.9	Details of Motor Driver- L298N	25
4.10	H-Bridge Circuit	26
4.11	DC motor	27
4.12	DC gear motor	29
4.13	Esp32-Wroom	29
4.14	Esp32-Wroom pin diagram	31
4.15	7-INCH HDMI LCD	32
5.1	Assembled model	39
5.2	Detecting the track through camera	41
5.3	Raspberry Pi camera captured image	43
5.4	RGB into grayscale converted image	44
5.5	Blurred image	45

5.6	Edge detection	45
5.7	Theta calculation	46
5.8	Line detection	47
5.9	Deciding whether to turn right or left or straight	48
5.10	Model moving through the path to destination	48
5.11	Flowchart of Program flow	49

## LIST OF ABBREVIATIONS

AI	-	Artificial intelligence
ML	-	Machine learning
RGB	-	Red, green, blue
IDE	-	Integrated development environment
CUDA	-	Compute Unified Device Architecture
GND	-	Ground
MP	-	Mega Pixel
CSI	-	Camera Serial Interface
TTL	-	Transistor–transistor logic
UART	-	Universal asynchronous receiver-transmitter
ADC	-	Analog to digital converter
DAC	-	Digital to analog converter
LCD	-	Liquid Crystal Display
LED	-	Light Emitting Diode
PCB	-	Printed Circuit Board
RAM	-	Random Access Memory
GPIO	-	General - Purpose Input/Output
EEPROM	-	Electrically Erasable Programmable Read-Only Memory
PoE	-	Power Over Ethernet
BLE	-	Bluetooth Low Energy
LAN	-	Local area network
GPU	-	Graphical processing unit
MMC	-	Modular multilevel converter
SD	-	Secure Digital
HDMI	-	High-Definition Multimedia Interface
HD	-	High definition
SDRAM	-	Synchronous Dynamic Random Access Memory

- DC - Direct current
- IOT - Internet of things
- RNN - Recurrent neural network
- CNN - Convolutional neural network
- OpenCV - Open computer vision

# CHAPTER 1

## INTRODUCTION

Mechanical engineering, electronic engineering, information engineering, computer science, and other engineering and science disciplines are all part of robotics. Design, manufacture, operation, and use of robots, as well as computer systems for their perception, control, sensory feedback, and information processing, are all covered under robotics. Robotics' purpose is to create intelligent machines that can help and assist humans in their daily lives while also keeping everyone safe.

Robotics is the science of developing machines that can act like humans and copy their movements. Robots can be employed in a variety of scenarios and for a variety of objectives, but many are now used in hazardous areas (such as inspection of radioactive items, bomb detection and deactivation), manufacturing operations, or other situations where humans are unable to live. A robot that executes behaviors or tasks with a high degree of autonomy is known as an autonomous robot (without external influence). Artificial intelligence, robotics, and information engineering are all considered subfields of autonomous robotics. Spaceflight, household maintenance (such as cleaning), wastewater treatment, and delivering goods and services are just a few of the applications for autonomous robots.

Within the narrow limitations of their immediate surroundings, some modern industry robots are "independent." Although their surroundings may not provide every degree of freedom, the manufacturing robot's working is demanding and frequently contains unpredictable, unpredicted circumstances. The exact orientation and position of the next work object, as well as the type of object and the required task (in more complex factories), must be defined. This can change unexpectedly (at least from the robot's perspective). The ability of a robot to cope with its environment, whether on land, underwater, in the air, underground, or in space, is an important topic in robotics research.

## CHAPTER 2

### LITERATURE SURVEY

[1]Aninditya Anggari Nuryono in 2020 discussed that robotic vision is a robot that can obtain information through image processing by the camera. The camera installed on the line following robot aims to identify image-based lines and to navigate the robot to follow the path. The method of image preprocessing along with robot action for line-following robots is used. The image preprocessing includes dilation, erosion, Gaussian filtering, contour search, and centerline definition to detect path lines and to decide the robot's action. The robot is implemented using Webots simulator. OpenCV and Python are used to create line detection systems and robot movements. The simulation displays that the method is implemented properly, and the robot can follow a few path lines such as zigzag, dotted, and curved line.

[2]Sagar Shetty in 2019 discussed image processing-based line following robot which is made so that it can follow a predetermined path and is used for transportation. A camera is utilized to gain an image of track and then it is converted into bitmap image. Least Square Method is used to understand the predefined path. By calculating the slope of line and angle the robot determines where it should make a turn. Thus, by using image processing technique the line following robot is directed along the desired path.

[3]Chun-Fei Hsu in 2018 discussed a vision-based two-wheel self-balancing (TWSB) robot to follow a black line using visual feedback. MATLAB software is used to connect the URL of ip camera and uses image processing toolbox to process the image from the ip camera. After image processing technique, it sets 10 coordinates to detect if the black line is straight or the black line is in different position. Then it considers the black line together with straight line, curve line, intersection, and inconsecutive line. Thus, a cascade intelligent motion control system is used to control the balance and moment of the vision-based two-wheel self-balancing (TWSB) robot with tracking the position and direction commands from MATLAB. In the end, it

indicates that the vision-based two-wheel self-balancing (TWSB) robot can trace the black line on the map perfectly from the real-time experimental results.

[4]Reza Javanmard Alitappeh presented a deep neural network for autonomous line following robot learning in 2020. The robot travels automatically and follow the line placed on the ground. By training the network with images and videos a sufficient amount of data is collected. In this application, the autonomous driving robot is viewed from two perspectives: one based on image instances, and the other on temporal video data. In this experiment, CNN (convolutional neural network) and RNN (recurrent neural network) networks are used. Several sensors are also put on the robot to provide it a local perception of its surroundings. A camera is mounted on the top of the robot with 45-degree facedown. Other sensors such as infrared sensor is mounted on the front side to percept obstacle in front of the robot. The camera and sensors gives the input for data collection and for further processing. A Webot simulator equipped with several sensors is designed for data collection.

[5]Rahul P. Kharapkar in 2020 discussed a mini version of self-driving car using IOT with raspberry pi and Arduino UNO running as a main processor chip. The 8-megapixel high resolution pi camera will give the necessary data, and the raspberry pi will analyze the data (samples) and train the neural network and machine learning algorithm on the raspberry pi. As a result, road lanes and traffic lights can be detected. More than 200 photos in various lighting conditions are used to train the vehicle. The robot can also make appropriate turns. In addition to these characteristics, if the automobile encounters a barrier, it will overtake with appropriate LED warnings.

[6]Ming-Han Lee presented a four-wheel independent steering and four-wheel independent driving (4WIS4WID) car with an autonomous and manual driving system in 2016. The autonomous driving system is based on machine vision and fuzzy control theory and includes three applications: lane following, reverse parking, and parallel parking. The vehicle does these tasks by detecting lines on the ground with four webcams and controlling all the server motors with related control technologies. The 4WIS4WID vehicle has webcams on all sides for all-around viewing, however there are still many blind spots, and the view isn't wide enough. Furthermore, a parking place

cannot be seen in its entirety in a single image. As a result, integrating judgements of vision with fuzzy control methods is required to ensure that the 4WIS4WID vehicle can conduct the proper motions. The 4WIS4WID vehicle can adjust its velocity in a timely manner under any condition and successfully navigate a confined and curved lane utilizing the proposed fuzzy rules. Furthermore, the manual driving system is based on the classic driving technique, allowing individuals to quickly adapt. An indoor real-time experiment is done to verify the viability of these uses for the 4WIS4WID vehicle.

[7]Daping Jin presented a method in 2020 for identifying, tracking, and following a target person in dynamic surroundings that combines a Line-of-Sight sensor (a 2D laser range finder) and a Nonline-of-Sight sensor (a low-cost Angle-of-Arrival (AOA) tag). To begin, a Kalman Filter is utilised to fuse person tracking information from AOA data and laser data in order to obtain a smooth and continuous moving trajectory of the target person. The laser data is then used to create a real-time robot-centric rolling grid map. A target potential field is built on top of the rolling grid map using the fast-marching approach, and then a direction gradient field is created based on the target potential field. An improved dynamic window algorithm, FMM-DWA, is suggested to drive the robot to move towards the target using the rolling grid map, the target potential field, and the direction gradient field. To prevent falling into the local optimum, this method analyzes not only the distance between the robot and the objective, but also the difference between the robot's heading and the reference direction provided by the direction gradient field. A series of person following experiments are carried out in difficult dynamic circumstances to validate the performance of the suggested technique. The results of the experiments reveal that the suggested following algorithm can successfully deal with the occlusion problem and reliably follow the target individual while avoiding static and dynamic obstacles.

[8]Aiman Salim presented an autonomous automobile prototype in 2019 that employs the Raspberry Pi as the main processor, and our system makes use of OpenCV and machine learning technology. The system will proceed to the destination on its own, without any assistance from humans. The car's main processing system is a Raspberry Pi, which is connected to a Pi camera module and streams video to a

monitor acting as the local host. Pedestrians, cars, traffic signs, and signals are detected, and relevant orders are delivered serially to the Arduino to control the car. The Raspberry Pi includes features such as traffic signal detection, vehicle detection, pedestrian detection, and road sign detection, all of which help the suggested system arrive at the proposed or defined location safely and on time. Every step is carried out on the Raspberry Pi, which is programmed in C++. Gaussian Blur, CED, and Region of Interest are the methods utilized to achieve autonomous car movement. Assembled the robot body by putting together the robot car's chassis and wheels, soldering the motors, and mounting them in the chassis, and connecting the Raspberry Pi, Arduino Uno, and motor driver.

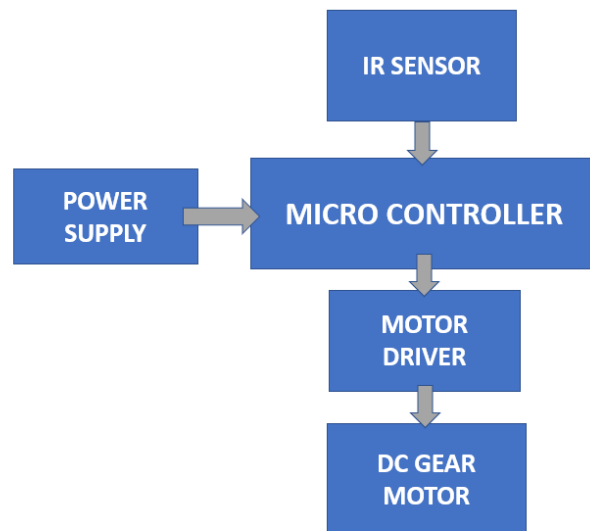
## CHAPTER 3

### AIM AND SCOPE OF THE PRESENT INVESTIGATION

#### 3.1 AIM

This vehicle is a self-driving car that employs artificial intelligence and computer vision. In this project, we create a car with a camera that detects the track or path around it. If the automobile is about to steer into one, it should respond. The "eye" of self-driving vehicles is computer vision with an AI-based algorithm. The primary goal of computer vision is to provide a smooth self-driving experience. The foundation of Artificial Intelligence technology is computer vision. AI assists computers in decoding and comprehending visual input obtained from diverse sources. It entails employing AI algorithms to do autonomous visual understanding. The facial recognition technology is the best example of computer vision. This model employs the Canny edge detection technique. This variant is mostly utilized in industries to assist in moving and transferring things from one end to the other. As a result, it reduces the need for human intervention and makes the work easier.

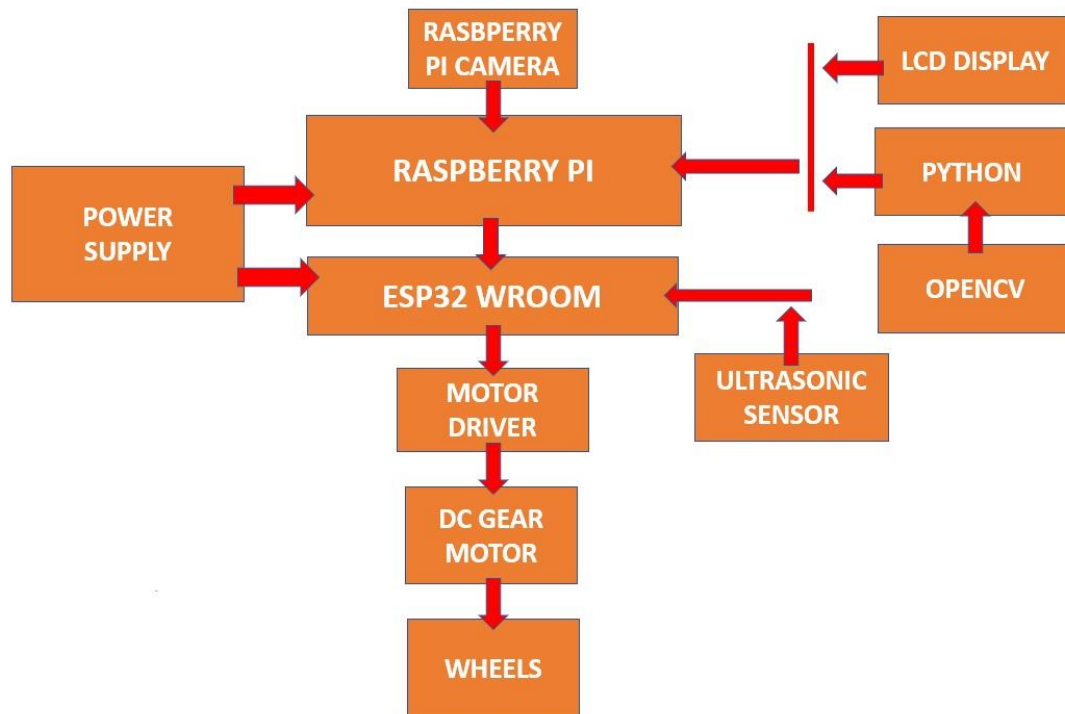
#### 3.2 EXISTING SYSTEM



*Fig: 3.1 Existing System*

The system uses a microcontroller with an IR sensor to reach the destination. It follows the desired path or line and moves automatically. The IR sensor senses the and the information is given to the microcontroller. Microcontroller gives commands to the motor driver and then the motor works. This system is a basic system which has low features, and it also has various disadvantages.

### 3.3 PROPOSED SYSTEM



**Fig: 3.2 Proposed System**

The proposed system will move on its own without human assistance to its destination using Artificial intelligence and Machine learning technology. The model uses a central processing system as the Raspberry Pi, connected to the Pi camera module. The camera will give the input images for the processes. The whole process is completed using the Raspberry Pi and OpenCV with python language. The methods used to achieve the independent movement of the robot are canny edge detection algorithm. The robot with Raspberry Pi, Esp32, camera, LCD display, motors and wheels are being attached to the chassis and the required circuit connections are made.

## **CHAPTER 4**

### **EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED**

#### **4.1 METHODS AND ALGORITHM**

Artificial intelligence with Canny edge detection algorithm and machine learning technology is used. Artificial intelligence (AI) is the simulation of human intelligence in robots that have been trained to think and act like humans. The phrase can also refer to any machine that demonstrates human-like characteristics like learning and problem-solving. The ability of artificial intelligence to rationalize and execute actions that have the best likelihood of reaching a certain goal is its ideal feature. Machine learning is a subset of artificial intelligence that refers to the idea that computer systems can learn from and adapt to new data without the need for human intervention. Deep learning techniques allow for this autonomous learning by absorbing large volumes of unstructured data including text, photos, and video.

#### **4.2 MACHINE LEARNING**

Machine learning is a sort of artificial intelligence that allows software applications to improve their accuracy at predicting outcomes without having to be explicitly programmed to do so. To forecast new output values, a machine learning system uses historical data as input. Machine learning is frequently used in recommendation engines. Fraud detection, spam filtering, malware threat detection, corporate process automation, and predictive maintenance are all common applications. Machine learning is significant because it allows businesses to see trends in customer behavior and business operating patterns while also assisting in the development of new goods. Machine learning is at the heart of many of today's most successful businesses, like Facebook, Google, and Uber. For many businesses, machine learning has become a crucial competitive differentiation. Machine Learning comes in a variety of forms. The way an algorithm learns to become more accurate in its predictions is how traditional machine learning is often classified. Supervised

learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four primary methodologies. The algorithm that data scientists use is determined by the sort of data they wish to predict.

### **4.3 CANNY EDGE DETECTOR ALGORITHM**

The Canny edge detector is an edge detection operator that detects a wide range of edges in images using a multi-stage approach. It was created in 1986 by John F. Canny. Canny also developed a computational theory of edge detection that explains how the method works. A multi-stage edge detector is the Canny filter. The intensity of the gradients is computed using a filter based on the derivative of a Gaussian. The Gaussian filter decreases the impact of image noise. Then, by deleting non-maximum pixels of the gradient magnitude, possible edges are reduced down to 1-pixel curves. Finally, applying hysteresis thresholding on the gradient magnitude, edge pixels are maintained or eliminated. The width of the Gaussian (the noisier the image, the wider the width), as well as the low and high thresholds for hysteresis thresholding, are all customizable in the Canny.

The following are some general criteria for edge detection:

1. Edge detection has a low error rate, which indicates that the detection should catch as many of the image's edges as possible.
2. The edge point recognized by the operator should be accurate in locating the edge's center.
3. Image noise should not cause spurious edges, and a specific edge in the image should only be marked once.

### **4.4 HARDWARE REQUIREMENTS**

1. Raspberry Pi 3 Model B+
2. Raspberry Pi Camera Module
3. Ultrasonic Sensor
4. L298N Motor Driver
5. DC Gear Motor

6. Esp32-Wroom
7. 7 - inch HDMI LCD

#### ***4.4.1 Introduction to Raspberry Pi***

The Raspberry Pi is a credit card-sized computing board. It can do many of the same things as a desktop PC, such as spreadsheets, word processing, gaming, and watching high-definition video. The Raspberry Pi Foundation in the United Kingdom created it. Since 2012, the Raspberry Pi has been available for public purchase with the goal of creating a low-cost educational microcomputer for students and children. The Raspberry Pi was designed to inspire learning, experimentation, and invention. The Raspberry Pi computer is both portable and affordable. The majority of Raspberry Pi boards are used in Raspberry Pi projects, mobile phones, and solar streetlights.

The beginning of the twenty-first century saw a massive increase in mobile computer technology, with the mobile phone industry driving a substantial portion of this expansion. ARM technology was used in 98 percent of mobile phones. The ARM CPU core would subsequently be used on the Raspberry Pi, allowing the ARM technology to be exploited.

Model A and Model B are the two different models of the system. The key distinction between these two is the presence of USB ports. The model A board will use less power because it does not have an Ethernet port. The type B, which comes with an Ethernet interface and is made in China. The Raspberry Pi includes a collection of open-source technologies, including web connectivity and multimedia, as well as the ability to interface with a microcontroller.

#### ***4.4.2 Raspberry Board - Model A***

The Raspberry Pi is based on the Broadcom BCM2835 chip and features a 700 MHz ARM1176JZF-S core CPU and 256 MB of SDRAM. The USB 2.0 ports employ only external data communication methods. A micro-USB converter with a 500 MA minimum range powers the Raspberry Pi (2.5 watts). The graphics specialist chip is designed to reduce the time it takes to calculate and manipulate images. It has a

Broadcom video core IV cable, which is helpful if you want to utilise your Raspberry Pi to play games and watch videos.

#### *4.4.2.1 Features*

- 256 MB SDRAM memory
- Broadcom BCM2835 SoC full HD multimedia processor
- Dual Core Video Core IV Multimedia coprocessor
- Single 2.0 USB connector
- HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC) Video Out
- 3.5 mm Jack, HDMI Audio Out
- SD, MMC, SDIO Card slot on board storage
- Linux Operating system
- 8.6cm\*5.4cm\*1.5cm dimensions

#### **4.4.3 Raspberry Board - Model B**

The Raspberry Pi is a system based on the Broadcom BCM2835 chip, with a 700 MHz ARM1176JZF-S core CPU. SDRAM is integrated inside the Raspberry Pi. It has 512 MB. The USB 2.0 ports are utilised only for external data communication. The Raspberry Pi's principal form of connection with other devices and the internet in model B is Ethernet. A micro-USB adaptor with a minimum range of 500 MA is used to power the Raspberry Pi (2.5 watts). The graphics specialist chip is designed to speed up the processing of image calculations. If you wish to use your Raspberry Pi to play games or watch videos, it comes with a Broadcom video core IV cable.

#### *4.4.3.1 Features*

- 512 MB SDRAM memory
- Broadcom BCM2835 SoC full HD multimedia processor

- Dual Core Video Core IV Multimedia coprocessor
- Single 2.0 USB connector
- HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC) Video Out
- 3.5 MM Jack, HDMI Audio Out
- SD, MMC, SDIO Card slot on board storage
- Linux Operating system
- 8.6cm\*5.4cm\*1.7cm dimensions
- On board 10/100 Ethernet RJ45 jack

#### ***4.4.4 Types of Raspberry Pi***

- Raspberry Pi 1 model B
- Raspberry Pi 1 model A
- Raspberry Pi 1 model B+
- Raspberry Pi 1 model A+
- Raspberry Pi Zero
- Raspberry Pi 2
- Raspberry Pi 3 model B
- Raspberry Pi Zero W

#### ***4.4.5 Raspberry Pi 3 Model B+***



**Fig: 4.1 Raspberry Pi**

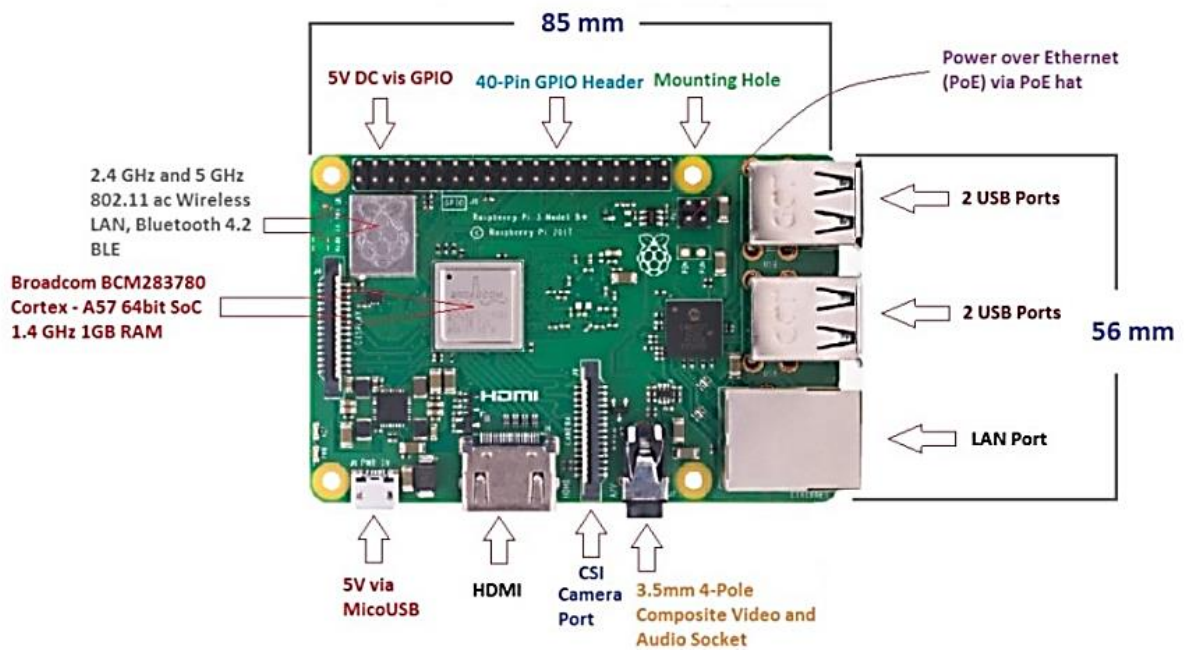
It is a customized version of the Raspberry Pi 3 B, which was released in 2016 and featured a CPU, GPU, USB ports, and I/O pins. Both models are nearly identical in terms of functionality and technical specs; however, the B+ model has USB boot, network boot, and Power over Ethernet, which are not available on the B model.

Technology has progressed over time with the goal of making people's lives easier and more convenient. This device was a significant advancement in technology that made computer learning so simple that anyone could get their feet wet in the process with minimal effort. The BCM2837B0 system-on-chip (SoC) of the Raspberry Pi 3 Model B+ contains a 1.4 GHz quad-core ARMv8 64bit processor and a powerful Video Core IV GPU. Snappy Ubuntu Core, Raspbian, Fedora, and Arch Linux, as well as Microsoft Windows 10 IoT Core, are among the ARM GNU/Linux distributions that can be run on the Raspberry Pi. The Raspberry Pi 3 Model B+ boasts significant performance improvements over the Model B including a faster CPU clock speed (1.4 GHz versus 1.2 GHz), greater Ethernet throughput, and dual-band Wi-Fi. With a Power over Ethernet HAT, it also enables Power over Ethernet (not included).

The dual-band wireless LAN has modular compliance certification, which allows the board to be integrated into end devices with much reduced wireless LAN compliance testing, lowering both cost and time to market. The Raspberry Pi

Foundation created the Raspberry Pi to provide a low-cost platform for computer programming exploration and teaching. The Raspberry Pi can perform many of the same tasks as a traditional desktop computer, such as word processing, spreadsheets, high-definition video, games, and programming. The board's four USB ports can be used to connect USB devices such as keyboards and mice.

#### 4.4.6 Introduction to Raspberry Pi 3 B+



**Fig: 4.2 Raspberry Pi 3 B+ description**

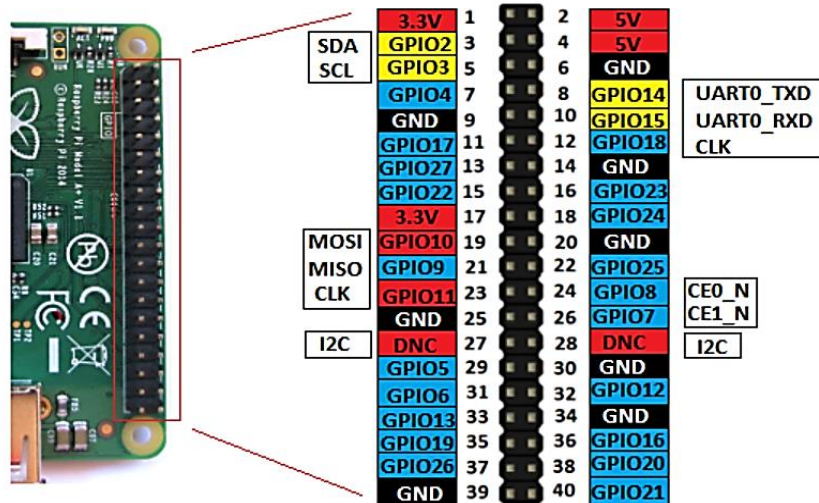
The Raspberry Pi Foundation announced the Raspberry Pi 3 B+ on March 14, 2018. It's an advanced version of the Raspberry Pi 3 B model, which was released in 2016. It's a small computer board featuring a CPU, GPU, USB ports, I/O pins, Wi-Fi, Bluetooth, USB, and network boot capabilities, and it can do some functions similar to a standard computer. The features of the B+ model are nearly identical to those of the B model; however, USB and Network Boot, as well as Power over Ethernet, are only available on the B+ model. This device also has two additional USB ports. The SoC (system on chip) is quicker than the Pi 2 and 3 models because it combines the CPU and GPU into a single device.

#### 4.4.6.1 Improvements

In terms of processing speed, the model B+ maintains its lead and adds increased wireless capabilities. The dual-band Wi-Fi 802.11ac operates at 2.4GHz and 5GHz for improved range in wirelessly difficult areas, while Bluetooth 4.2 with BLE support is available. Instead of plastic on earlier models, the top side is coated with metal shielding, which works as a heat sink and drains excess heat if the board is exposed to high temperatures or pressure. This B+ model is three times quicker than the Pi 2 and 3, which is a significant advancement in terms of performance, allowing it to perform a variety of tasks at a reasonable speed. The ethernet port has a speed of 300 Mbit/s, which is significantly faster than the previous version's speed of 100 Mbit/s. It's called gigabit ethernet since it uses a USB 2.0 interface. On the board near the 40 pin headers, a four-pin header is added. This enables Power over Ethernet (PoE), which delivers the required electrical current to the device using data cables rather than power cords. It's really beneficial and cuts down on the amount of cords needed to install a gadget in a given project. Only in the presence of a PoE hat does PoE work.

#### 4.4.6.2 Raspberry Pi 3 B+ Pinout

- A 40-pin header is utilised to connect the electronic gadget to the outside world. This is identical to prior versions, making it compatible with any devices that support past versions.
- Of the 40 pins, 26 are used as digital I/O pins, and 9 of the remaining 14 are designated as dedicated I/O pins, indicating that they have no other function.
- Pins 3 and 5 have an inbuilt 1.8 k $\Omega$  resistor, while Pins 27 and 28 are allocated to the ID EEPROM.
- The GPIO header has been significantly moved in the B+ variant to make room for the additional mounting hole.
- Devices that function with the B model may work with the B+ version, however they may not sit exactly the same as the prior version.



**Fig: 4.3 Raspberry Pi pin diagram**

#### 4.4.6.3 Hardware Specifications

**CPU:** The CPU is the little computer's brain, assisting in the execution of a variety of instructions based on mathematical and logical principles. It has a 64-bit storage capacity.

**Clock Speed and RAM:** It has a 1.4 GHz Broadcom BCM2837B0 processor with quad-core ARM Cortex-A53 processor and 1GB of RAM memory (identical to the previous version)

**GPU:** GPU stands for graphics processing unit, which is used to calculate images. The gadget that is primarily used for playing video games has a Broadcom video core cable installed.

**HDMI and Composite Connection:** Both the audio output socket and the video composite socket have been combined into a single 4-pole 3.5mm socket near HDMI. In the new B+ model, the power connector has been moved adjacent to the HDMI plug. All of the power and audio video composite sockets are now on one side of the PCB, giving it a neat and tidy appearance.

**USB Hard Drive:** The USB hard drive is accessible from the board that is used to boot the device. It's the same as a typical computer's hard disc, except that Windows is used to boot the computer's hard drive.

**USB Ports:** This new version adds two more USB ports, removing the need for an external USB hub for connecting multiple peripherals to the device.

**Micro-USB Power Source Connector:** This connector is used to supply the board with 5V power. It consumes 170 to 200 milliamps more than the B model.

**PoE:** The B+ model has a Power over Ethernet (PoE) function, which is a novel feature in this device that allows the necessary electrical current to be delivered via data lines.

**Other Changes:** The B+ version offers little in the way of new functionality and has a slightly altered layout in terms of component placement. A tiny SD memory card slot replaces the SD memory slot (works similar to the previous version). The status LEDs have been relocated to the opposite end of the PCB and are now simply red and green in colour.

#### *4.4.6.4 Raspberry Pi 3 B+ Technical Specifications*

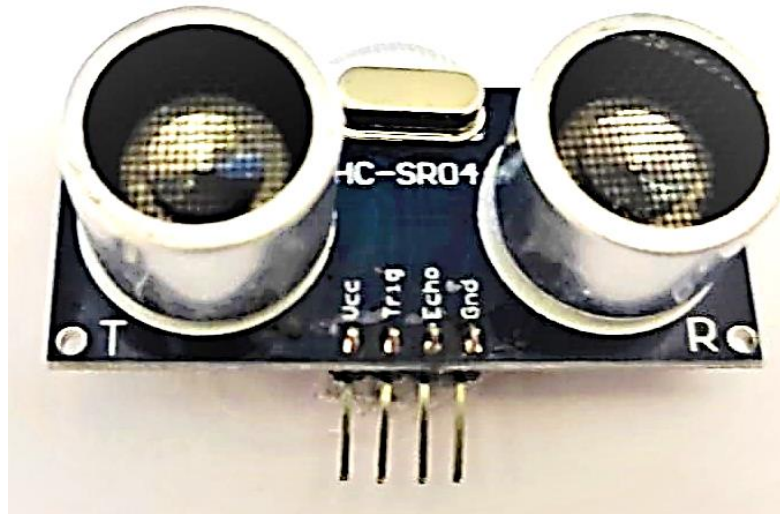
- Has a 64-bit CPU with 1GB RAM (random access memory)
- Has a Broadcom BCM2837B0 chipset
- Has a 1.4GHz quad-core ARM Cortex-A53 processor with four cores
- Contains a 40-pin header (26 GPIOs)
- 3.5mm jack connector supports stereo audio and composite video
- 4 USB 2.0 ports
- Gigabit Ethernet
- This device has a crucial feature called PoE (power over Ethernet) that the B model does not have.
- 2-pin reset header

- Micro SD socket for expanding the board's memory capacity
- MicroUSB power connector for charging the device
- HDMI
- CSI camera interface
- WiFi and Bluetooth capabilities not found in prior Raspberry Pi 1 and 2 models
- DSI connector for official screen display

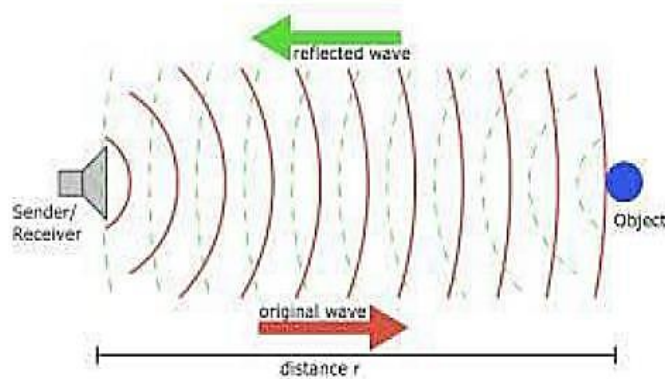
#### *4.4.6.5 Applications*

- Tablet Computer
- Home Automation
- Controlling Robots
- Coffee Projects
- Arcade Machine
- Media Streamer
- Internet Radio
- Cosmic Computer

#### **4.4.7 Ultrasonic Sensor**



**Fig: 4.4 Ultrasonic sensor**



**Fig: 4.5 Ultrasonic sensor working**

A transmitter and receiver are included with the HC-SR04 ultrasonic sensor. This sensor is used to determine the distance between the target and the sensor. The distance between the sensor and an object is determined by the amount of time it takes to transmit and receive the waves. Using non-contact technology, this sensor detects sound waves. The required distance for the target may be measured without causing damage using this sensor, and accurate details can be obtained. This sensor has a range of 2 to 400 centimeters. The HC-SR04 is an ultrasonic sensor that uses sonar

to determine the distance between the sensor and the object it detects. It has a wide range of non-contact detecting capabilities, as well as high accuracy and consistent readings. Ultrasonic transmitter and receiver are two of the modules included. This sensor can be found in a wide range of applications, including direction and speed measurement, intruder alarms, medical, sonar, humidifiers, wireless charging, non-destructive testing, and ultrasonography.

#### *4.4.7.1 HC-SR04 Ultrasonic Sensor Pin Configuration*

- Pin1 (Vcc): This pin provides the sensor with a +5V power source.
- Pin2 (Trigger): This is an input pin that is utilised to start the measurement by emitting ultrasonic waves for 10 seconds.
- Pin3 (Echo): This is an output pin that goes high for a set amount of time, which is equal to the time it takes for the wave to return to the sensor.
- Pin4 (Ground): This is a GND pin that connects to the system's GND.

#### *4.4.7.2 Features*

- This sensor is powered by a +5V DC power supply.
- Dimensions: 45mm x 20mm x 15mm
- Quiescent current: less than 2mA
- Input pulse width: 10uS
- Operating current: 15mA
- Measuring angle: 30°
- Distance range: 2cm to 800 cm
- Resolution: 0.3 cm
- Effective Angle: less than 15°
- Operating frequency range: 40Hz
- Accuracy: 3mm

#### 4.4.7.3 HC-SR04 Ultrasonic Sensor Working

HC-SR04 has 4 pins: Vcc, Trigger, Echo, and Ground. This sensor is used to determine the precise distance between the sensor and the target. This sensor is mostly used to detect sound waves.

When this module is given power, it emits sound waves that travel through the air and strike the required object. These waves strike and return from the object, which the receiver module catches. Because the time required to travel a greater distance is longer, both the distance and the time spent are directly related. When the trigger pin is held high for 10 $\mu$ s, ultrasonic waves are generated that travel at the speed of sound. As a result, it generates eight cycles of sound bursts, which are collected within the Echo pin. This ultrasonic sensor is connected to an Arduino board to determine the required distance between sensor and object. The following formula can be used to calculate the distance.

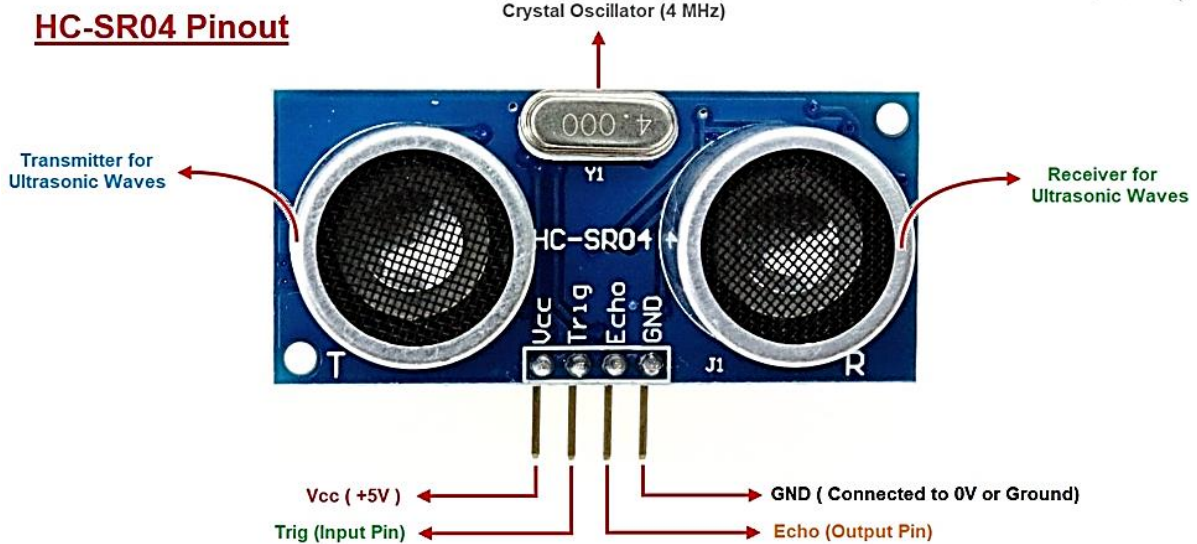
$$S = (V \times t)/2$$

S - required distance

V - sound's speed

t - time taken for sound waves to return back after striking the object.

Because the time it takes for the waves to travel and return from the sensor is twice as long, the real distance can be estimated by dividing the value by two.



**Fig: 4.6 Ultrasonic sensor description**

#### **4.4.8 Raspberry Pi Camera Module**

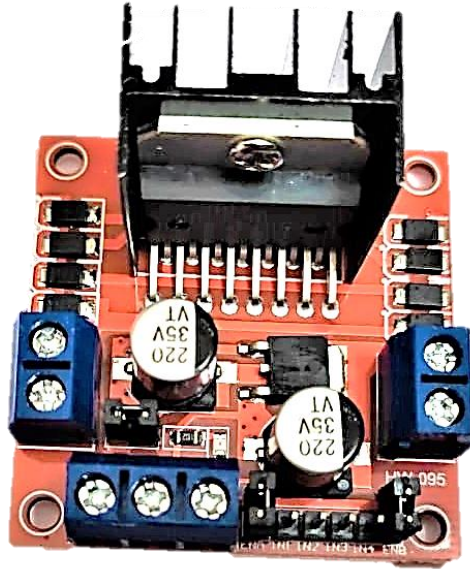
Prior to the camera module, an appropriate webcam could be used to access a camera feed on the Raspberry Pi. Camera technology had previously been used into a number of applications, ranging from live weather monitoring to robots. The Raspberry Pi Foundation, on the other hand, knew that if the Raspberry Pi community repurposed their existing webcams, an add-on camera module had a good chance of succeeding. The camera module, which was created exclusively for Raspberry Pi boards, included a variety of characteristics that set it apart from many other advanced cameras on the market (see the box titled "Camera Specs"). The Raspberry Pi camera module can capture both video and still images in high definition. It's simple to use for beginners, but it has a lot to offer advanced users who want to learn more. People have used it for time-lapse, slow-motion, and other video tricks, and there are plenty of examples online. To create effects, you can also use the libraries that come with the camera. A five-megapixel fixed-focus camera on the module captures video in 1080p30, 720p60, and VGA90 formats, as well as stills. It connects to the Raspberry Pi's CSI port via a 15cm ribbon wire. The MMAL and V4L APIs can be used to access it, and there are various third-party libraries for it, including the Pi camera Python library. The Camera Board for the Raspberry Pi connects directly to the Raspberry Pi's

CSI connection. It can record 1080p HD video at 30 frames per second or give a crystal clear 5MP quality photograph. The Raspberry Pi Camera Board has a 5MP (2592\*1944 pixels) Omnivision 5647 sensor in a fixed focus module that was custom designed and produced by the Raspberry Pi Foundation. The module connects to the Raspberry Pi via a 15-pin ribbon cable to the unique 15-pin MIPI Camera Serial Interface (CSI), which was built specifically for camera interface. The CSI bus, which is capable of exceptionally high data speeds, is used to carry pixel data to the BCM2835 processor entirely. The board is small, measuring roughly 25mm x 20mm x 9mm and weighing just over 3g, making it ideal for mobile and other applications where size and weight are critical. The sensor itself features a fixed focus lens and a native resolution of 5 megapixels. The camera can capture static photographs with a resolution of 2592 x 1944 pixels and can record video in 1080p @ 30fps, 720p @ 60fps, and 640x480p 60/90. Raspbian, the Raspberry Pi's chosen operating system, now includes support for the camera.



***Fig: 4.7 Raspberry Pi Camera Module***

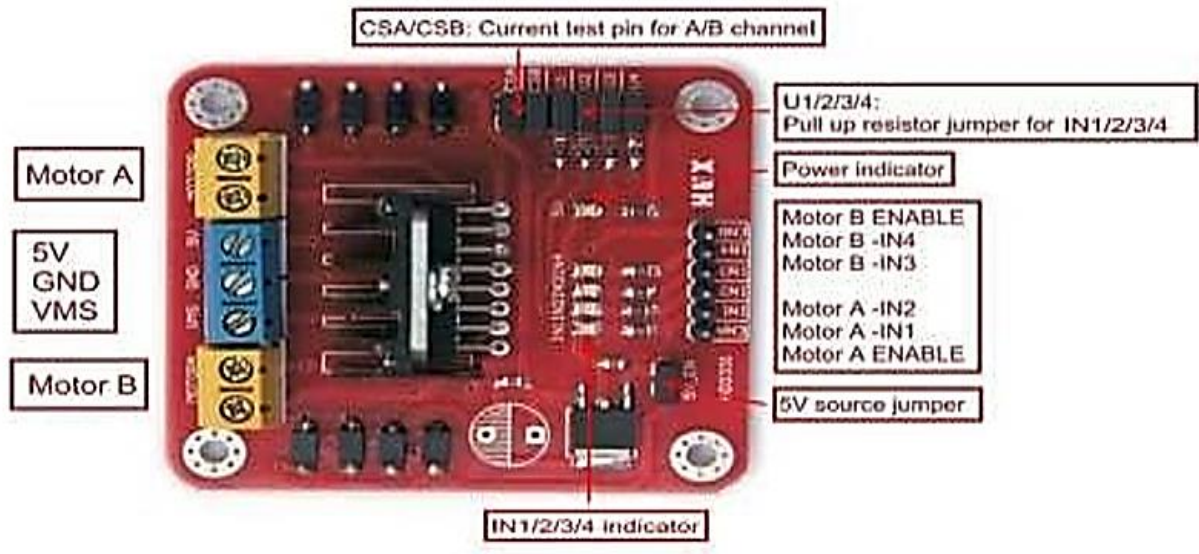
#### ***4.4.9 L298N Motor Driver Board***



***Fig: 4.8 L298N Motor Driver Board***

The ST L298N dual full-bridge driver, an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packaging, is used in the Double H driver module. It's a high-voltage, high-current dual full-bridge driver that accepts conventional TTL logic levels and drives inductive loads such relays, solenoids, DC motors, and stepping motors. The device can be enabled or disabled independently of the input signals using two enable inputs. Each bridge's lower transistors' emitters are connected together, and an external sensing resistor can be connected to the corresponding external terminal. To allow the logic to operate at a lower voltage, an additional supply input is provided.

#### *4.4.9.1 Hardware Diagram*



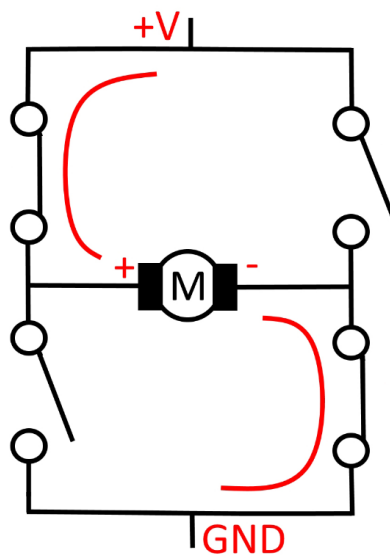
**Fig: 4.9 Details of Motor Driver- L298N**

- CSA: This is the current test pin for motor A; it can be wired with a resistor for current testing or disabled with a jumper.
- CSB: This is the current test pin for motor B; it can be linked with a resistor for current testing or disabled with a jumper.
- VMS: VMS is the motor's 5V-35V power supply. GND is negative, while VMS is positive.
- 5V: The logic circuit on the board receives power from this source.
- 5V-EN: Jumper for a 5V source. When the jumper is set, the 78M05 draws 5V from the VMS port to power the logic circuit on the board ( $7V < VMS < 18V$ ). When this jumper is turned off, the 5V port supplies power to the board's logic circuit.
- U1/2/3/4: IN1/2/3/4 pull-up resistor When the jumper is on, the microcontroller's pull up resistor is enabled; when it is off, the microcontroller's draw up resistor is disabled.
- EA/EB: EA/EB is the enable pin for the two motors A/B, and the PWM of this pin can also regulate the motor speed.

- IN1/2/3/4: IN1/2/3/4 is the motor control pin. Motor B is identical to Motor A.

#### 4.4.9.2 H-Bridge Dc Motor Control

To regulate the rotation direction, however, we just need to reverse the direction of the current flow through the motor, which is most commonly accomplished using an H-Bridge. An H-Bridge circuit is made up of four switching devices, such as transistors or MOSFETs, with the motor in the middle, producing a H shape. We can change the direction of the current flow and consequently the rotation direction of the motor by activating two specific switches at the same time. We can have complete control over the DC motor if we combine these two technologies, PWM and H-Bridge. Many DC motor drivers have these characteristics, and the L298N is one of them.



**Fig: 4.10 H-Bridge Circuit**

#### 4.4.10 DC Motors

A Direct Current motor is a device that transforms DC electrical energy into mechanical energy. When a current-carrying conductor is subjected to a magnetic, it exerts a mechanical force.



**Fig: 4.11 DC motor**

#### *4.4.10.1 Working Principle of A DC Motor*

An electric motor is a machine that transforms electrical energy into mechanical. "When a current carrying conductor is put in a magnetic field, it exerts a mechanical force," says the basic operating principle of a DC motor. Fleming's left-hand rule determines the force's direction, and  $F = BIL$  determines its magnitude. Where B denotes the magnetic flux density, I denote the current, and L is the length of the conductor within the magnetic field.

#### *4.4.10.1 Fleming's Left Hand Rule*

If we stretch out the first, second, and thumb of our left hand perpendicular to each other, the first finger signifies the magnetic field, the second finger signifies the current, and the thumb signifies the force experienced by the current carrying conductor. When armature windings are coupled to a DC supply, they generate an electric current. Field winding (electromagnetism) or permanent magnets can be used to generate a magnetic field. As per the above-mentioned principle, current-carrying armature conductors are subjected to a force by the magnetic field. To achieve unidirectional torque, the commutator is segmented. Or else, the direction of the force would have reversed every time the conductor's movement in the magnetic field was reversed.

#### *4.4.10.2 Back Emf*

No energy conversion is conceivable until there is something to oppose the conversion, according to natural rules. This opposition is given by magnetic drag in generators, but back emf is present in dc motors. When the armature of a motor rotates, the conductors cut the magnetic flux lines, causing an emf to induce in the armature conductors, according to Faraday's law of electromagnetic induction. The induced emf is directed in the opposite direction as the armature current. The direction of the back emf and armature current is shown in the circuit diagram below. The emf equation of a DC generator can be used to calculate the magnitude of the back emf.

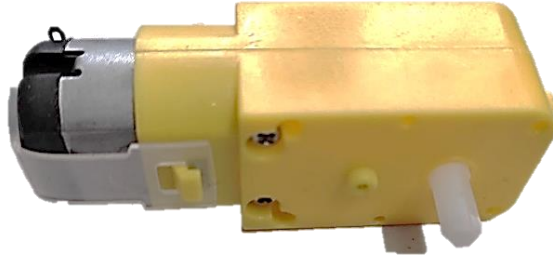
#### *4.4.10.3 Significance of Back Emf*

The magnitude of the back emf is related to the motor's speed. Consider a dc motor whose load is abruptly lowered. The required torque will be little in comparison to the current torque in this instance. Due to the additional torque, the motor's speed will begin to increase. As a result, because the back emf is related to the speed, the magnitude of the back emf will likewise grow. The armature current will start to decrease as the back emf increases. Because torque is related to armature current, it will decrease until it is adequate for the load. As a result, the motor's speed will be regulated. When a dc motor is quickly loaded, however, the load causes the speed to decrease. Back emf will drop as speed decreases, allowing greater armature current. To meet the load demand, increased armature current will increase torque. As a result, the presence of the back emf causes a dc motor to self-regulate.

#### *4.4.10.4 DC Gear Motor*

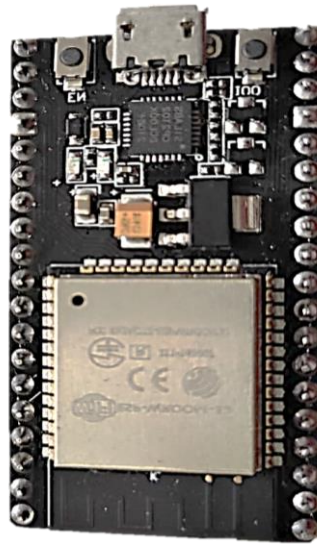
A gear motor combines a motor and a gearbox into one unit. When a gear head is added to a motor, the speed is reduced but the torque output is increased. In terms of gear motors, the most significant criteria are speed (rpm), torque (lb-in), and efficiency (percent). To choose the best gear motor for your application, you must first calculate the load, speed, and torque requirements for your application. ISL Products has a wide range of Spur Gear Motors, Planetary Gear Motors, and Worm Gear Motors

to satisfy your needs. Most of our DC motors can be combined with one of our one-of-a-kind gearheads to create a highly efficient gear motor solution.



**Fig: 4.12 DC gear motor**

#### **4.4.11 Esp32-Wroom**

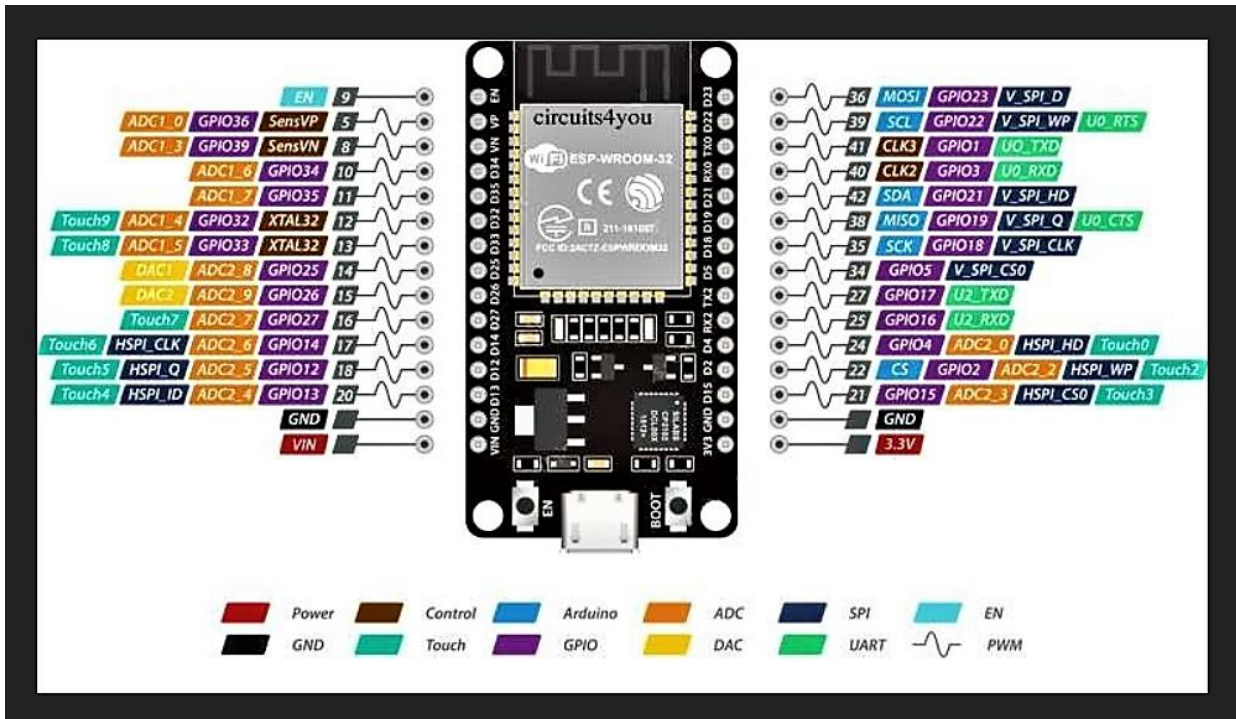


**Fig: 4.13 Esp32-Wroom**

The ESP32 is a family of low-cost, low-power system-on-a-chip microcontrollers that include built-in Wi-Fi and dual-mode Bluetooth. The ESP32 series uses a Tensilica Xtensa LX6 dual-core or single-core microprocessor, a Tensilica Xtensa LX7 dual-core microprocessor, or a single-core RISC-V microprocessor with built-in antenna switches, RF baluns, power amplifiers, low-noise receive amplifiers, filters, and power-

management modules. Espressif Systems, a Shanghai-based Chinese firm, designed and developed the ESP32, which is produced by TSMC using their 40 nm technology. It is the ESP8266 microcontroller's replacement.

The ESP32-WROOM-32 is a versatile Wi-Fi+BT+BLE MCU module that can handle everything from low-power sensor networks to the most demanding operations like voice encoding, music streaming, and MP3 decoding. The ESP32-D0WDQ6 chip is at the heart of this device. The integrated chip is made to be scalable and adaptable. The CPU clock frequency can be adjusted from 80 MHz to 240 MHz, and there are two CPU cores that can be separately regulated. The user can alternatively disable the CPU and utilise the low-power co-processor to continuously monitor the peripherals for changes or threshold crossings. Capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S, and I2C are all included in the ESP32. The module's integration of Bluetooth, Bluetooth LE, and Wi-Fi ensures that it can be used for a wide range of applications and that it is future-proof: Wi-Fi allows for a large physical range and direct connection to the internet via a Wi-Fi router, while Bluetooth allows the user to connect to the phone or broadcast low-energy beacons for its detection. The ESP32 chip has a low sleep current of less than 5  $\mu$ A, making it ideal for battery-powered and wearable electronics applications. The ESP32 has a data throughput of up to 150 Mbps and a 20.5 dBm antenna output power for the best physical range. As a result, the chip has industry-leading specs and provides the finest performance in terms of electronic integration, range, power consumption, and connection. The operating system used for the ESP32 is free RTOS with LwIP, as well as TLS 1.2 with hardware acceleration. Developers can also use secure (encrypted) over-the-air (OTA) upgrades to keep their goods updated after they've been released.

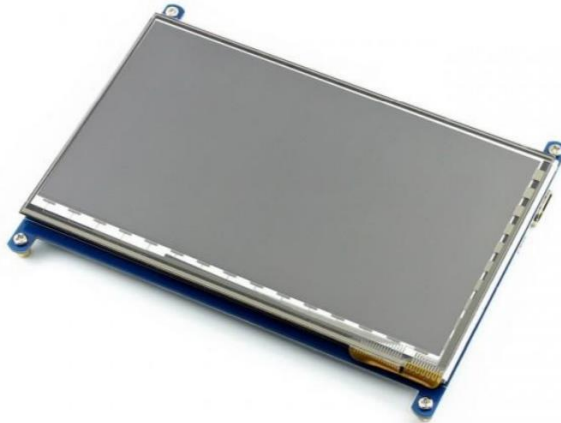


**Fig: 4.14 Esp32-Wroom pin diagram**

#### 4.4.11.1 Esp32 Peripherals Features

- analog-to-Digital Converter (ADC) channels
- capacitive sensing GPIOs
- UART interfaces
- SPI interfaces
- I2C interfaces
- WM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces

#### 4.4.12 7-INCH HDMI LCD



**Fig: 4.15 7-INCH HDMI LCD**

The 7-inch HDMI LCD supports a variety of systems, including Raspberry Pi, Banana Pi, Banana Pro, and BB Black, and can display Ubuntu, Raspbian, and Angstrom images with a high resolution of 1024x600 pixels and a capacitive Touch Screen. It can be used on a Raspberry Pi, an HDMI display screen, or any other tiny PC or computer display. No input and outputs are required for displaying and the driver is included (works with custom Raspbian directly). Backlight control to reduce power consumption while maintaining good immersion quality surface coated in gold. It has HDMI output and is software configurable up to 1920x1080. It works with Raspberry Pi, Jetson Nano, BB Black, and Banana Pi tiny PCs, as well as standard desktop computers.

Raspberry Pi supports Raspbian /Ubuntu /Kali /RetroPie /WIN10 IoT without the need for drivers. Supports Windows 10/8.1/8/7, five-point touch, and is driver-free in a computer monitor. It is compatible with gaming consoles such as Microsoft's Xbox 360, Sony's PlayStation 4, and Nintendo's Switch, among others. OSD menu with multi-language options for power management, brightness/contrast adjustments, and more.

#### *4.4.12.1 Specifications*

- Hardware resolution:1024 x 600
- Frequency: 60Hz

- Vcc: 4.7-5v
- Total Current: 400mA
- Close Backlight current: 120mA
- Backlight Brightness: 300cd/m<sup>2</sup>
- Audio Frequency: 44.1Khz
- Size: 165mm x 122mm

## **4.5 SOFTWARE REQUIREMENTS**

1. OpenCV
2. Python
3. Embedded C

### **4.5.1 OpenCV**

OpenCV (Open-Source Computer Vision Library) is a free software library for computer vision and machine learning. OpenCV was created to provide a common infrastructure for computer vision applications and to help commercial goods incorporate machine perception more quickly. Because OpenCV is a BSD-licensed product, it is simple for businesses to use and alter the code. More than 2500 optimized algorithms are included in the collection, which covers both classic and cutting-edge computer vision and machine learning techniques. These algorithms can be used to detect and recognize faces, identify objects, classify people's actions in videos, track camera motion, track moving objects, collect 3D models of objects, generate 3D point clouds from stereo camera systems, sew images with each other to create a high-resolution picture of an entire thing, identify similar images from the database, remove red eyes from pictures captured with flash, pursue eye movements, recognize scenery, and overlay markers. OpenCV has a user group of about 47 thousand members and an estimated number of downloads of over 18 million. The library is widely used by businesses, research groups, and government agencies. Along with well-known corporations like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, and Toyota that use the library, several startups like Applied Minds, VideoSurf, and Zeitera use it

extensively. OpenCV's applications range from stitching street view images together to detecting intrusions in surveillance tape in Israel, tracking mine machinery in China, assisting Willow Garage robots in navigating and picking up objects, designed to detect swimming pool drowning collisions in Europe, running engaging art in Spain and New York, inspecting airstrips for debris in Turkey, and quick facial recognition in Japan. It works with windows, Linux, Android, and Mac OS and offers C++, Python, Java, and MATLAB. When MMX and SSE instructions are accessible, OpenCV tends to rely toward real-time vision applications. CUDA and OpenCL interfaces with full functionality are currently being developed. Over 500 algorithms exist, with roughly ten times as many functions that compose or support them. OpenCV is written in C++ and provides a templated interface that integrates with STL containers elegantly.

#### **4.5.2 Python**

Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its increased built in data structures, coupled with dynamic typing and dynamic binding, make it quite interesting for faster Application Development, including for use as a scripting or glue language to connect existing elements together. Python's concise, easy-to-learn syntax prioritizes readability, which lowers software maintenance costs. Modules and packages are supported by Python, which fosters programme modularity and code reuse. The Python interpreter and its substantial standard library are free to download and publish in source or binary form for all major platforms. Python is popular among programmers because of the enhanced productivity it offers. The edit-test-debug cycle is extraordinarily rapid because there is no compilation step. Python scripts are simple to debug: a bug or improper input will never result in a segmentation fault. Instead, when the interpreter finds a mistake, it throws an exception. The interpreter prints a stack trace if the application fails to catch the exception. Inspection of local and global variables, execution of arbitrary expressions, setting breakpoints, stepping through the code one line at a time, and so on are all possible with a source level debugger. The debugger is written in Python, demonstrating Python's introspective capabilities. On the other hand, adding a few print statements to the source code is frequently the quickest

method to debug a programme: the fast edit-test-debug cycle makes this simple approach quite successful.

#### *4.5.2.1 Advantages of the Python Programming Language*

Python's clear and basic syntax attracts newcomers who desire to learn this scripting language. From some angles, it may appear logical and predetermined that Python will become the de facto coding language, rendering all of its competitors obsolete. Its code is simple to understand, distribute, and update. There is no overuse of words, and the language is simple to pick up.

Python programmes are text files that provide instructions for the interpreter and are written in a text editor or integrated development environment (IDE). Text editors do not generally include IDE functions, but they can be altered. IDEs are full-featured and include in-built tools such as syntax checks, debuggers, and code browsers. Python also comes with a plethora of third-party packages, libraries, and frameworks to help with development. Python is therefore ideal for large-scale applications due to its optimization capabilities.

Here, we're talking about company speed and the time-to-market statistic. Python is a dynamic scripting language, which means it's not designed for developing applications from the ground up, but rather for connecting components. The interfaces between components and scripts are well-defined, and components are designed to be reusable. It all helps to speed up software development by making Python a highly concise and productive language. Despite the fact that Python prioritizes code simplicity and readability over flexibility, the language nonetheless possesses it. Python may be used in a variety of projects. It gives programmers the option of using object-oriented or procedural programming techniques. Python is also versatile when it comes to data types. Number, String, List, Tuple, and Dictionary are the five root types, and each sub-data type corresponds to one of them. As a result of Python's versatility, conducting exploratory data analysis becomes easier.

Python was created with portability in mind. Its programmes run on any modern computer operating system. Python scripts are interpreted because of the high-level nature of the language, therefore they may be written for further interpretation on Linux, Windows, Mac OS, and UNIX without requiring any changes. Python applications can also be used to create portable user interfaces. Python has a constantly rising user base and exemplifies what a strong community looks like. Python's sophisticated toolkit has hundreds of contributors — Pythonists. Users have already contributed around 200,000 custom-built software programmes to an online repository. All of this means that the large supportive community is both the cause and the result of the language's popularity.

### **4.5.3 Embedded C**

Embedded C is the most popular programming language for constructing electronic gadgets in the software system industry. Every processor in an electronic system is linked to a software system called embedded software. Embedded C programming is essential for the CPU to accomplish specified tasks. We have a tendency to utilise a variety of electronic gadgets in everyday life, such as a portable, washer, photographic camera, and so on. This multi-device operation is based on a microcontroller programmed in embedded C. For blinking the LED attached to Port0 of the microcontroller, the embedded C code shown in the diagram is used. C code is preferred over other languages in embedded system programming. As a result of the following factors:

- Simple to comprehend
- Exceptional Reliability
- Portability
- Scalability

The majority of customers are used to software packages that provide functionality on a computer. Embedded package is less visible, but not any less complex. Unlike an application package, an embedded package has certain hardware requirements and capabilities, and the insertion of third-party hardware or software is

rigorously regulated. All essential device drivers must be included in the embedded package at the time of manufacture; therefore device drivers are created specifically for the hardware. The mainframe and individual chips chosen are very important to the package. Most embedded package engineers have at least a passing knowledge of reading schematics and knowledge sheets for components to see how registers and communication systems are used. Conversion among decimal, hex, and binary is also advantageous, as is bit manipulation. The ability to communicate between CPUs and other components is critical. I2C, SPI, serial ports, and USB are all typical protocols in addition to direct memory access. Firms such as Inter Niche Technologies and CMX Systems offer closed source communications protocols for embedded systems. uIP, lwip, and other open-source protocols are examples. A keyword is a word that has a particular meaning for the compiler (a C Compiler for example, is a software that is used to convert programme written in C to Machine Code). For example, some of the keywords in the Keil's Cx51 Compiler (a popular C compiler for 8051 based Microcontrollers) are as follows:

- bit
- sbit
- sfr
- small
- large

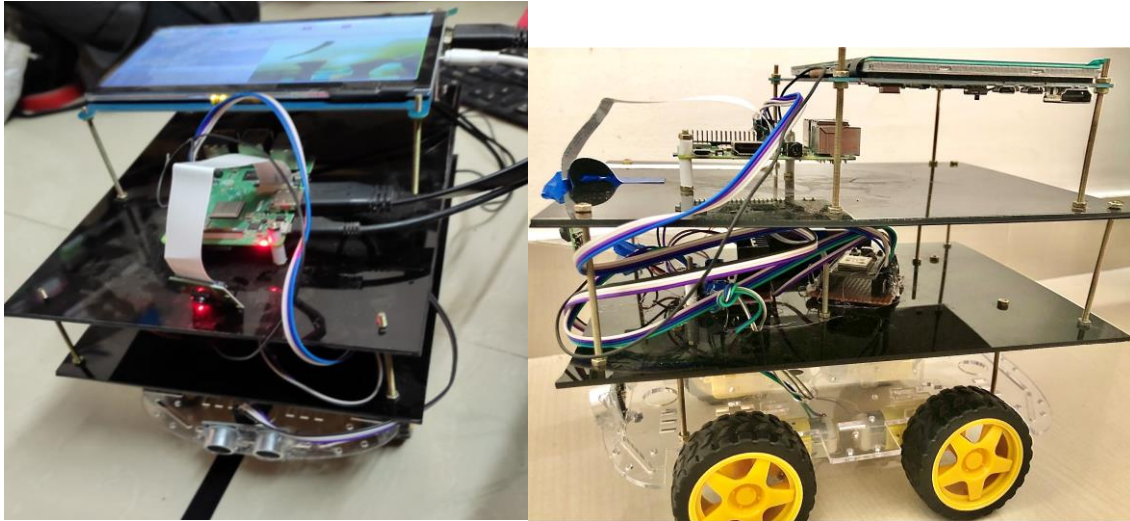
Along with the conventional C Keywords, these are only a few of the many keywords linked with the Cx51 C Compiler.

## **CHAPTER 5**

### **RESULT AND DISCUSSIONS, PERFORMANCE ANALYSIS**

#### **5.1 MODEL TRAINING IN MACHINE LEARNING**

A training model is a dataset used to train a machine learning algorithm. It is made up of sample output data as well as the equivalent sets of input data that have an impact on the outcome. The training model is used to process the input data via the algorithm in order to compare the processed output to the sample output. The model is modified based on the results of this association. "Model fitting" is the term for this iterative procedure. The precision of the model is dependent on the correctness of the training or validation dataset. In machine language, model training is the act of supplying data to an ML algorithm to help it find and learn suitable values for all of the variables involved. There are various types of machine learning models, with supervised and unsupervised learning being the most common. When the training data comprises both the input and output values, supervised learning is possible. A supervisory signal is a set of data that contains both the inputs and the intended outcome. When the inputs are fed into the model, the training is based on the deviation of the processed output from the documented result. The process of unsupervised learning entails identifying patterns in data. After then, further data is used to fit patterns or clusters. This is likewise an iterative procedure that improves accuracy by comparing the observed patterns or clusters to the expected patterns or clusters. This method does not have a reference output dataset.



***Fig: 5.1 Assembled model***

## **5.2 MODEL IN MACHINE LEARNING**

### ***5.2.1 Defining the Problem***

The first step in determining what an ML model should do is to define the problem statement. This step also allows for the identification of relevant inputs and outputs; questions such as "what is the main objective?" "what is the input data?" and "what is the model trying to predict?" must be addressed at this point.

### ***5.2.2 Data Collection***

It is required to explore and acquire facts that can be utilised to feed the machine after creating the problem statement. This is a crucial step in the process of building an ML model because the amount and quality of data used will determine the model's effectiveness. Data can be extracted from existing databases or created from scratch.

### ***5.2.3 Preparing the Data***

The data preparation stage involves profiling, formatting, and structuring data in order to prepare it for model training. This is the point at which the proper data traits and attributes are chosen. This stage will very certainly have a direct impact on the execution time and outcomes. This is also the point at which data is divided into two

groups: one for training the machine learning model and the other for evaluating it. At this stage, data is pre-processed by normalising, removing duplicates, and correcting errors.

#### ***5.2.4 Assigning Appropriate Model / Protocols***

The choice and assignment of a model or protocol must be made in accordance with the ML model's goal. You can choose from a variety of models, including linear regression, k-means, and Bayesian. The type of data being used has a big impact on the models that are used. For example, for image processing, convolutional neural networks would be the greatest choice, and for segmentation, k-means would be the best option.

#### ***5.2.5 Training the Machine Model Or “The Model Training”***

The ML algorithm is taught by feeding datasets to it at this stage. This is the stage during which the student learns. Consistent training can increase the ML model's prediction rate greatly. The model's weights must be initialised at random. The algorithm will learn to alter the weights accordingly as a result of this.

#### ***5.2.6 Evaluating and Defining Measure of Success***

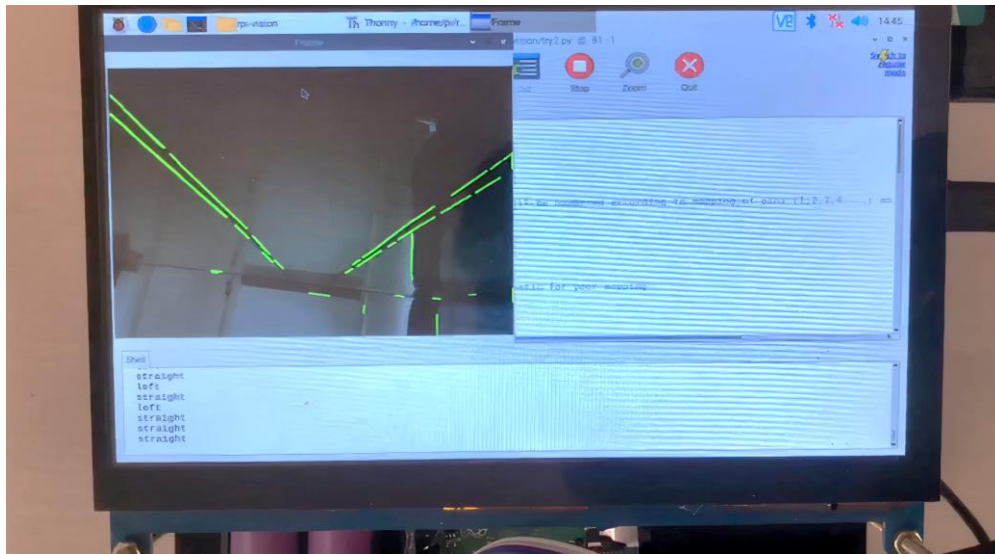
The "validation dataset" will be used to test the machine model. This aids in determining the model's correctness. For justifying correlation, identifying the measurements of success based on what the model is designed to achieve is crucial.

#### ***5.2.7 Parameter Tuning***

The key to achieving accurate correlation is choosing the right parameter to change to influence the ML model. Hyperparameters are a set of parameters that are chosen depending on their impact on the model architecture. Parameter tuning is the process of identifying hyperparameters through tuning the model. The correlation parameters should be properly set such that the validation point of decreasing returns is as close to 100 percent accuracy as possible.

### 5.3 DECISION TREE IN AI

The denotative representation of a decision-making process is a decision tree. In artificial intelligence, decision trees are used to reach conclusions based on the evidence available from previous judgments. Furthermore, these conclusions are given numerical values, which are then used to forecast the course of action that will be done in the future. Decision trees are statistical, algorithmic machine learning models that interpret and develop responses to a variety of problems and their potential outcomes. As a result, based on the data available, decision trees understand the rules of decision-making in certain settings. The learning process is ongoing, and feedback driven. This enhances the learning outcome over time. Supervised learning is the term for this type of learning. As a result, decision tree models serve as aids to supervised learning. Thus, rather than relying on intuition, decision trees give a scientific decision-making process based on facts and values. This technique is used by businesses to make important business choices.



*Fig: 5.2 Detecting the track through camera*

### 5.4 PROGRAM FLOW IN OPENCV

- ❖ Step 1: Importing libraries
- ❖ Step 2: Read image from cam

- ❖ Step 3: Color conversion
- ❖ Step 4: Image smoothing
- ❖ Step 5: Edge detection
- ❖ Step 6: Find lines
- ❖ Step 7: Decision-making part
- ❖ Step 8: Integrate

### **Step 1: Importing libraries**

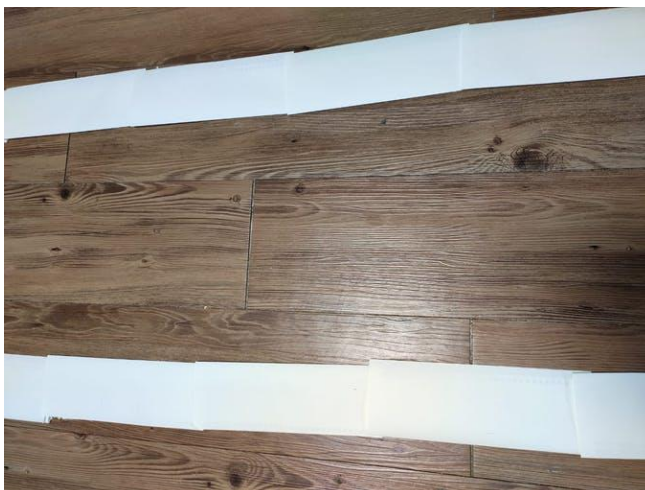
In this step, OpenCV and numpy is imported, then math library is imported for mathematical calculations. Serial library is also imported for serial communications between Raspberry Pi and Esp32.

```
import cv2  
import numpy as np  
import math  
import serial
```

### **Step 2: Read image from cam**

The Raspberry Pi camera module gives input to the Raspberry Pi. The given input will be in the form of video, then this video is converted to images and saved.

```
image = cv2.imread(r'C:\Users\****\Desktop\new1.jpeg')
```

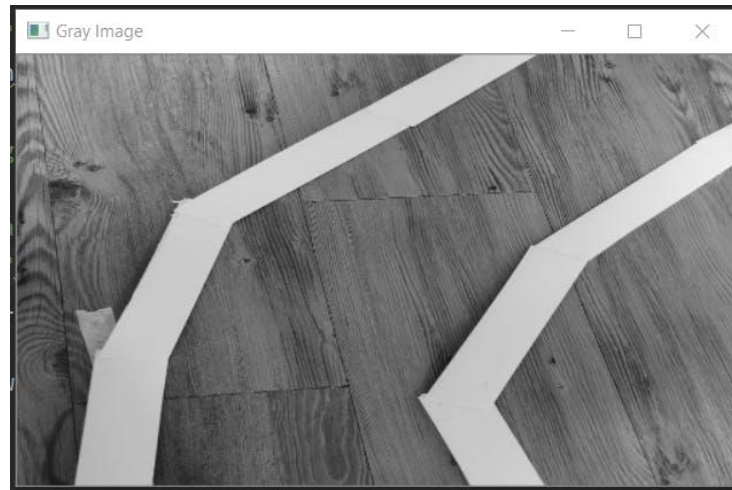


***Fig: 5.3 Raspberry Pi camera captured image***

### **Step 3: Color conversion**

The image is in RGB form. This RGB is converted into grayscale image. Because for a small image it will have high set of values if not converted to Grayscale image from RGB. There are three color channels and three dimensions in RGB images, whereas grayscale images are one-dimensional. The RGB photos of 10x10x3 pixels were used to train the neural article. There will be 300 input nodes in the input layer. For grayscale photos, however, the same neural network will only require 100 input nodes. Many algorithms have been tweaked to only function with grayscale photos. In RGB we will have to compare it for R, G and B. In grayscale we have to compare it only one time.

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```



**Fig: 5.4 RGB into grayscale converted image**

#### **Step 4: Image smoothing**

One of the most significant pre-processing phases in computer vision and image processing is smoothing and blurring. We can reduce the quantity of high-frequency material, such as noise and edges (i.e., the "detail" of an image), by smoothing a picture before applying techniques like edge detection. While it may seem counterintuitive, lowering the information in an image allows us to more readily locate objects of interest. This also permits us to concentrate on the image's bigger structural elements. Image smoothing is done for reducing noise in the image. Low pass and high pass filters are used for image smoothing which is inbuilt in OpenCV.

*# given input image, kernel width =5 height = 5, Gaussian kernel standard deviation*

*k\_width = 5*

*k\_height = 5*

*blurred = cv2.GaussianBlur(gray, (k\_width, k\_height), 0)*



**Fig: 5.5 Blurred image**

### Step 5: Edge detection

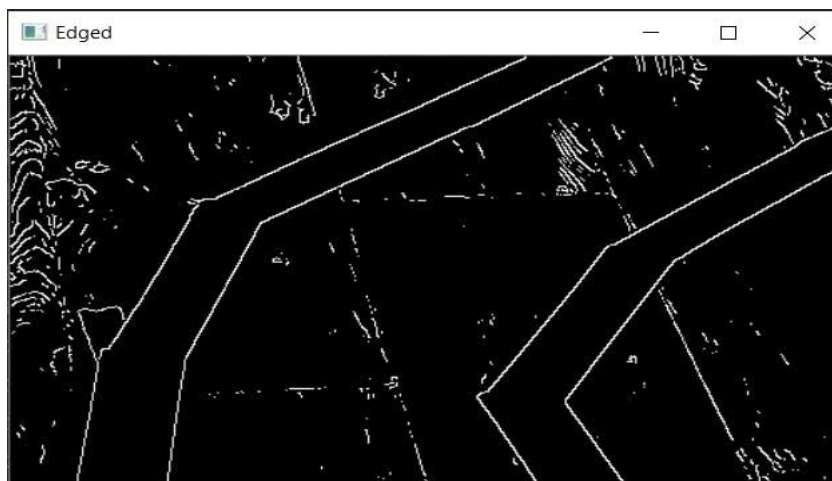
Edge detection is an image processing technique for finding the boundaries of objects within images. The purpose of edge detection is to find whether the path is curve or straight. We use AI here to identify the edges of the image. Image processing with Canny edge detection algorithm is used. To preserve high gradient values in image, we select low and high threshold values. This will help us to preserve the strong edges in image.

*# Find the edges in the image using canny detector*

*threshold1 = 80*

*threshold2 = 80*

*edged = cv2.Canny(blurred, threshold1, threshold2)*



**Fig: 5.6 Edge detection**

## Step 6: Find lines

In this step it determines whether the robot need to turn left or right according to the detections made in previous steps. After edge detection and line detection the robot makes a decision to turn left or right using machine learning technology.

Hough transform is a feature extraction method for detecting simple shapes such as circles, lines etc in an image.

*# it will return line coordinates,it will return 3darray.*

```
lines = cv2.HoughLinesP(edged,1,np.pi/180,max_slider,minLineLength,maxLine  
Gap)
```

```
for x in range(0, len(lines)):
```

```
    for x1,y1,x2,y2 in lines[x]:
```

```
        # draw line in image using cv2.line function.
```

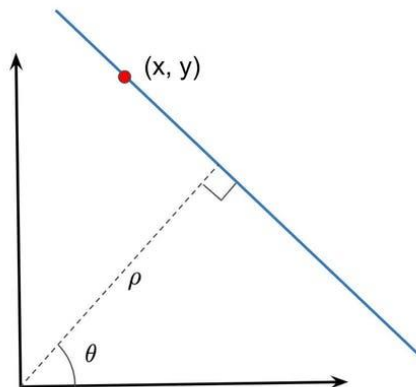
```
        cv2.line(image,(x1,y1),(x2,y2),(255,0,0),3)
```

```
        theta=theta+math.atan2((y2-y1),(x2-x1))
```

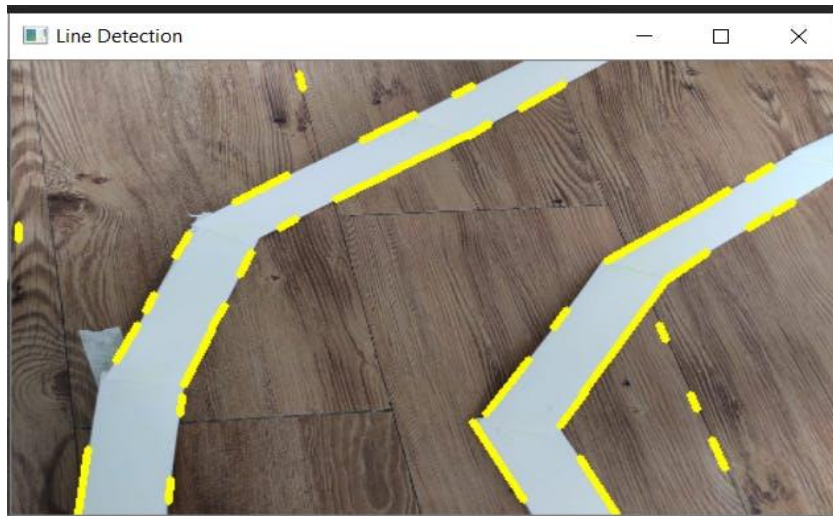
```
        print(theta)
```

*About theta calculation:*

atan2() function return value in radians, which means value between  $-\pi$  and  $\pi$  representing the angle theta of a (x, y) point and positive x-axis.



**Fig: 5.7 Theta calculation**



**Fig: 5.8 Line detection**

For taking the coordinate of first line this line of code can be used

*for x1,y1,x2,y2 in lines[0]:*

### **Step 7: Decision-making part**

Now, set the threshold value, based on difference between threshold and theta value, command is sent to ESP32 for drive the motor in particular direction.

*threshold=5*

*if(theta>threshold):*

*print("Go left")*

*if(theta<-threshold):*

*print("Go right")*

*if(abs(theta)<threshold):*

*print("Go straight")*



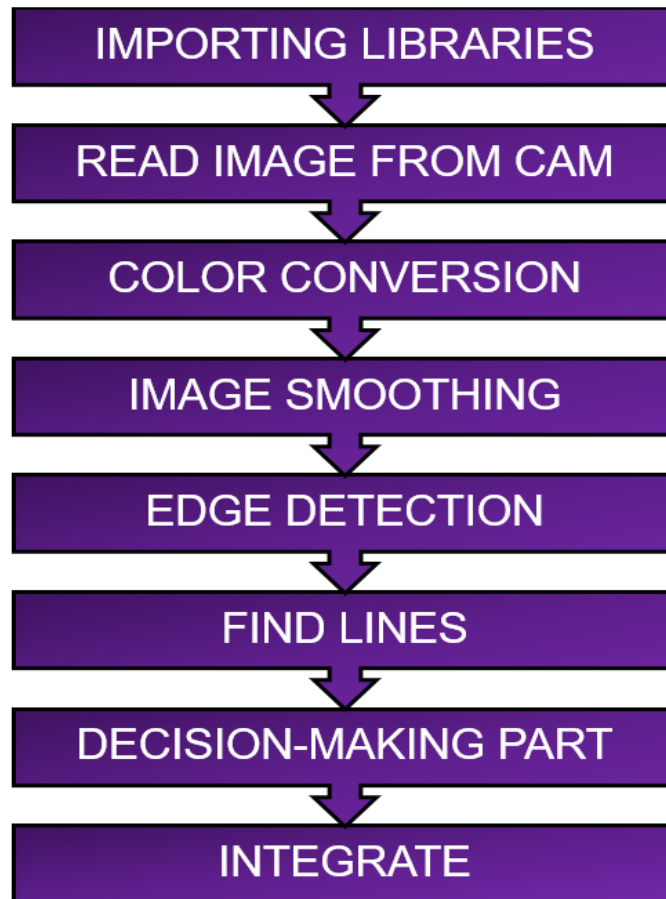
**Fig: 5.9 Deciding whether to turn right or left or straight**

### **Step 8: Integrate**

Then the robot moves to the desired destination by these algorithms. It also detects for any obstacles in its path.



**Fig: 5.10 Model moving through the path to destination**



*Fig: 5.11 Flowchart of Program flow*

## **CHAPTER 6**

### **CONCLUSION**

The proposed system gives the desired output through AI and ML. The scope of the proposed methodology lies in achieving an efficient robot that completes the task effectively. The use of this robot improves workplace safety in a number of ways. For starters, they can accomplish tasks that humans might consider unsafe. Second, they run in a smooth, regulated manner, reducing the chances of human-operated vehicles damaging other workers. Furthermore, improved workplace safety can lower costs and save money by minimizing the amount of harm that human operators can cause. This technology is designed to be precise and accurate. Of course, robots can make mistakes, but when compared to human errors, the chances of making a mistake are quite low.

## REFERENCES

- [1]Alfian Ma'arif, Aninditya Anggari Nuryono, Iswanto, "Vision-Based Line Following Robot in Webots", International Conference on Electrical Engineering (FORTEI-ICEE), September 2020.
- [2]Sagar Shetty, Aman Bhavsar, Mahesh Mergu, Ajay Talekar, "Line Following Robot Using Image Processing", Third International Conference on Computing Methodologies and Communication ICCMC, March 2019.
- [3]Chun-fei hsu, Chien-ting su, Wei-fu kao, Bore-kuen lee, "Vision-based line-following control of a two-wheel self-balancing robot", International Conference on Machine learning and Cybernetics, China, July 2018, doi:10.1109/icmlc.2018.8526952.
- [4]Reza Javanmard Alitappeh, Amir Hossein Zabbah, Mohammad Karimi, Abdolmaleki, Kossar Jeddisaravi, "Line Following Autonomous Driving Robot using Deep Learning", 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), December 2020. doi:10.1109/icspis51611.2020.9349547.
- [5]Rahul P. Kharapkar, Abhishek S. Khandare, Zeeshaan W. Siddiqui, "IoT based Self Driving Car", International Research Journal of Engineering and Technology (IRJET), Vol. 07, No. 3, pp: 5177-5181, March 2020.
- [6]Tzue-Hseng S. LI, Ming-Han Lee, Chia-Wei Lin, Guan-Hong Liou, Wei-Chung Chen, "Design of Autonomous and Manual Driving System for 4WIS4WID Vehicle", IEEE Access, Vol. 4, pp: 2256 – 2271, March 2016.
- [7]Daping Jin , Zheng Fang, Jiexin Zeng, "A Robust Autonomous Following Method for Mobile Robots in Dynamic Environments", IEEE Access, Vol. 8, pp: 150311 - 150325, August 2020.
- [8]Yasir A, Aiman Salim, Arya Dileep, Anjana S, "Autonomous Car using Raspberry PI and ML", International Journal of Recent Technology and Engineering (IJRTE), Vol. 9 No. 2, July 2020.