Self-Driving Car: Using OpenCV Library and Python

Astha Singh[1], Yash Bhadreshwara[1], Aditya Deshpande[1], Shivam Mane[1], Dhanashree Pannase[2]
[1]*Undergraduate Research Scholar, Dept. of Electronics and Telecommunication Engineering, Atharva College of Engineering, Malad , Maharashtra, India*
[2]*Professor, Dept. of Electronics and Telecommunication Engineering, Atharva College of Engineering, Malad , Maharashtra, India*

Abstract – The project uses the most recent OpenCV and Machine Learning technology to simulate a prototype of a self-driving car with monocular vision. Self-driving cars are autonomous vehicles that would reduce the need for human involvement, lowering the danger of accidents, and improving the safety, comfort, and accessibility of transportation. The car model will have the ability to recognise the lane path, sign boards, traffic light signals, and react to moving traffic in real time. The Raspberry Pi central processing unit, in conjunction with peripherals like the, L298 H-Bridge, and Raspberry Pi Cam2, is what gives our car the appropriate level of control. To give the car the necessary functionalities, computer vision is combined with algorithms like lane detection and object detection.

Key Words: Open CV, Machine Learning, Raspberry pi, Lane Detection, Object Detection, Canny Edge Detection.

1.INTRODUCTION

One of the most deadly driving errors results in fatalities and traffic. More likely are human blunders like talking on the phone while driving or listening to noisy entertainment systems in cars. In addition to these mistakes, mental and physical impairments contribute to failure in driving. It is more crucial than ever for today's technology to eliminate these errors because they are getting worse every day.

Self-driving cars are the answer to not just reducing these mistakes but also opening up new driving options and creating effective road management systems. Scientists are coming up with novel concepts in the realm of self-driving cars as a result of the rapid advancement of technology. These vehicles can travel by themselves and are autonomous. This is essentially a self-driving car model made using the best materials available. In this prototype, we've driven motors with an Arduino and a Raspberry Pi controller. A straightforward 10000mah power bank is used to power the vehicle. Driving would become safer and more comfortable thanks to the self-driving car's elimination of human intervention. Raspi Cam2's continuous image streaming serves as the device's input. The processing algorithm receives this input via a local host. Here, we analyze the output and in order to control the left and right motors appropriately, the Rpi feeds the H Bridge the required signals.

1.1Raspberry pi with Raspi cam.

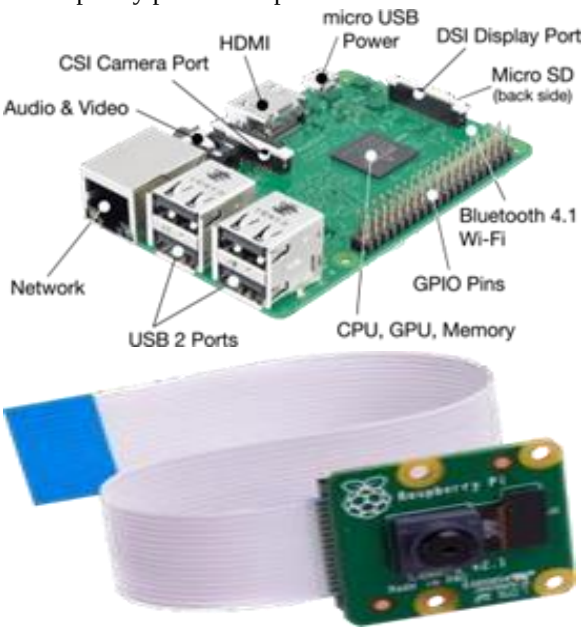

Fig -1: Raspberry Pi (3 Model B)

The main processing unit in this self-driving car is a Raspberry Pi 3 model B with an integrated Wi-Fi and Bluetooth module. This Raspberry Pi module can compute results similarly to a computer, but in a miniature credit card size. Even the mouse and keyboard can be connected to this module. For internal storage, we loaded the Raspbian OS and stored boot files on a 16GB micro-SD card. It receives input in the form of continuously streaming images from the Raspi Cam. The CSI port on the Raspberry Pi module is used to attach this Raspi Cam. This input is transmitted

through the local host to the processing algorithm. Here, we employ machine learning and computer vision to evaluate the output.

1.2L298 H bridge

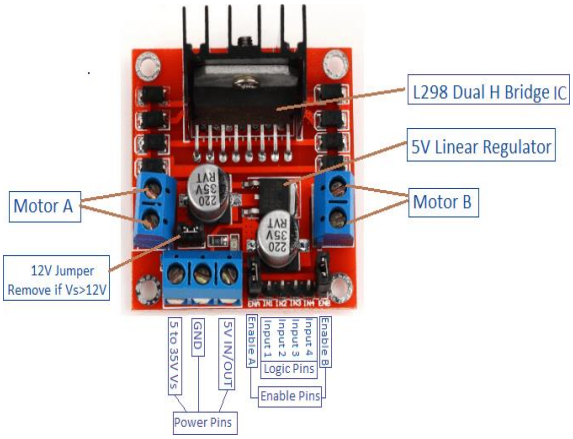

Fig -3: L298 H-Bridge

An integrated circuit in 15-lead Multi watt and PowerSO20 packages is the L298 (also known as a dual H-bridge). Typically, it is used to switch the polarity of the motors that are attached to it. High voltage and high current twin bridge drivers in this design can be fed via Arduino TTL logic levels. Our car can easily stop and run in accordance with the inputs supplied by employing this component.

2.RELATED WORK

The fundamentals of Open CV are introduced in [1], along with instructions on how to install Open CV and its libraries and read/write photos and videos in Open CV. creating a GUI and performing simple picture filtering. utilising thresholding and contouring as picture pre-processing techniques for object segmentation and detection. Lastly, image processing employing algorithms for object recognition and machine learning.

According to [2], visual recognition tasks including image classification, localization, and detection serve as the foundation for many autonomous applications, and convolutional neural networks (CNNs) have recently made significant strides towards achieving exceptional performance in these tasks. This programme uses Open Computer Vision (OpenCV) modules to recognise several objects in a given video.

According to [3], the four key technologies used in self-driving cars are the navigation system for cars, path planning, environment perception, and car control. There is a thorough discussion of all four of them and how they are combined in this project.

In [4], this research suggests a functioning self-driving automobile prototype that is capable of travelling safely on several sorts of tracks, including curved, straight, and straight followed by curved ones. Along with the Raspberry Pi, a camera module is attached on one end of the vehicle with the appropriate tilt. This transmits real-time images to the Convolutional Neural Network (CNN), which then forecasts the direction the car will travel in, such as right, left, forward, stop, or reverse. Commands are sent from the Arduino to the motor controller as a result, and the car moves in the intended direction without any human errors.

3.DESIGN ARCHITECTURE

We are using a raspberry pi as the primary processor chip in our self-driving vehicle. Since it contains a Wi-Fi module, the Raspberry Pi model 3 B is utilized to link our car via localhost. We installed the Raspbian Operating System on the Raspberry Pi after downloading it. Then, for the first time, it was connected to our PC via an RJ45 Ethernet connection. Through an Ethernet wire and the IP address that was given to the Raspberry Pi, Wi-Fi sharing was permitted. The OS was then updated to include OpenCV. Images from Raspi Cam2 are streamed continuously into the Raspberry Pi as input. The processing algorithm receives this input via localhost. Here, the result is evaluated using machine learning and computer vision.

3.1. IMAGE PROCESSING

The Raspi cam2 captures the input to the vehicle as a stream of photos. The OS came pre-built with the required camera libraries. There are algorithms for collecting still images, recording video, and calculating frames per second. The CV2 operates on the BGR image, converting the final image from BGR to RGB using the proper functions. Then, when genuine detection is necessary, we built Regions of Interest (ROI). The perspective of this ROI is altered for accurate picture analysis. The grayscale image taken by the Raspberry Pi Cam2 is then thresholded to create a black-and-white image. Canny edge detection is performed on this threshold image, and the threshold and canny images are combined.

3.2LANE DETECTION

After image processing, we calculated the distance of the white pixels (255) from the left side of the ROI for lane detection. To do this, the positions of the pixels are determined by treating them like vectors. For this, the histogram is employed. If the separation between white pixels widens, Rpi is given the proper instructions to turn in the proper direction. If the distance shrinks, it is told to turn left. Depending on the determined distance, the necessary degree of turn is calibrated.

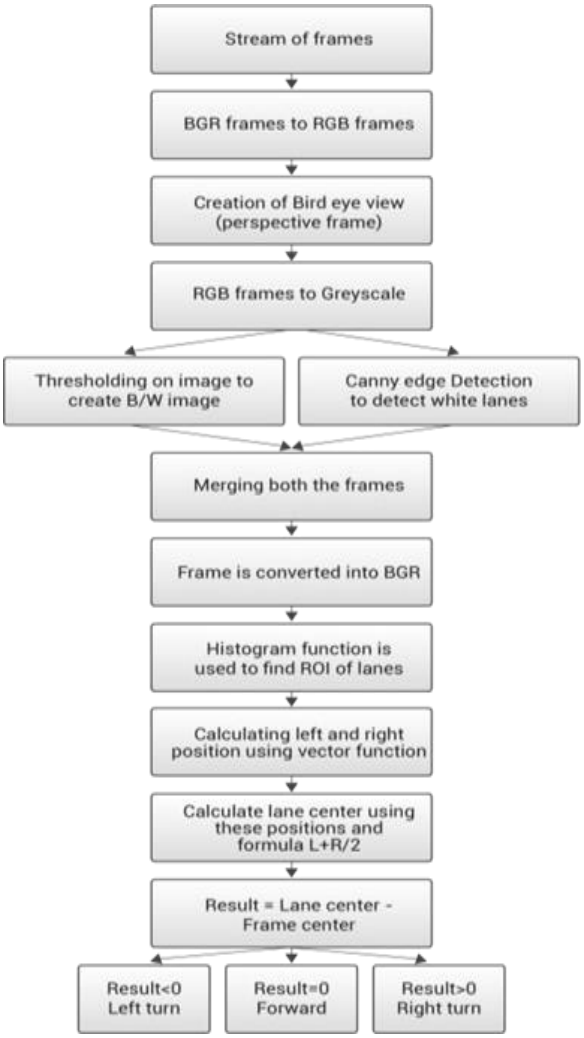

Fig -5: Flow chart for Image Processing and Lane Detection

3.3OBJECT DETECTION

When doing object detection, machine learning is involved. We start by collecting samples of the target object. The positive samples are what we refer to as. Then, we snap a number of pictures without the thing we're trying to detect. The negative samples are those. We used Cascade Classifier software for both training and detection. Then, using a variety of open CV methods, we were able to identify the target object in the function for object detection, and by using the distance formula (y=mx+c), we were able to determine the distance between the camera and the object by using both the distance and the pixels in the target object as parameters.

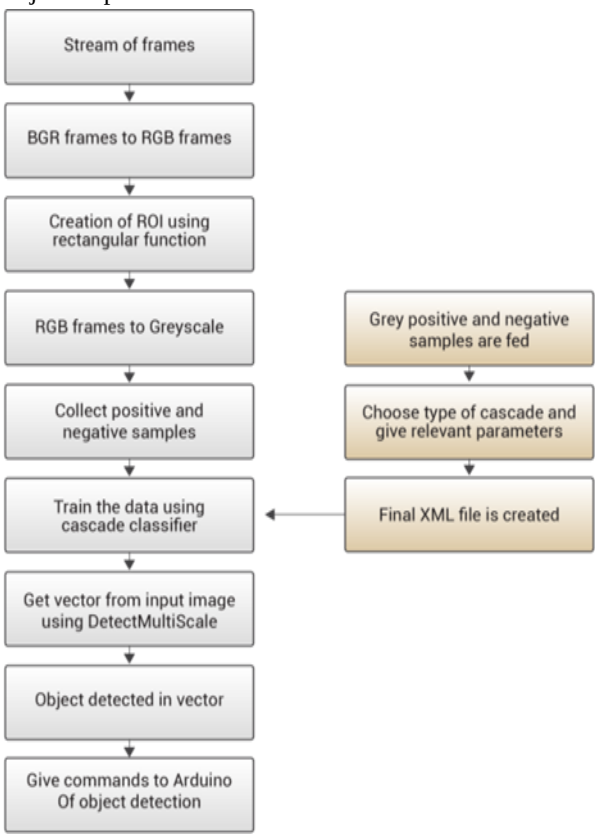

Fig -6: Flow chart for Object Detection

4.ADVANTAGES

- The movement of commodities into and out of cities will improve in safety and effectiveness. This can also be employed in vehicles that have self-driving capabilities so that the driver can focus on other vital tasks instead of on the road. This procedure will also result in less human interference.
- The elderly and the physically challenged will be able to travel independently when self-driving cars become a reality. There won't be a need for them to hire a driver. Children would also be permitted to move independently because the car

would be driving itself while being closely watched.

- Since the automobiles would operate on algorithms and would minimise avoidable mistakes, the length of traffic bottlenecks would be considerably reduced. In turn, this will enhance how the moving cars make decisions generally.

5.LIMITATIONS

- Infrastructure needs to be improved in India first if self-driving cars are to become commonplace. The automobiles wouldn't be able to make appropriate judgements on their own without a functioning road infrastructure.
- The interaction of self-driving cars and human drivers on the roads is one of the significant problems that are now present. Human drivers operate their vehicles instinctively, which makes it difficult for self-driving cars to read their movements in real time. This can be disastrous for all road users.
- When operating in harsh weather, machinery loses some of its working efficiency. As a result, it is unreliable to estimate how long the process will take. Incorrect input signals can occasionally arise when it's cloudy or raining, which can provide unexpected outcomes.
- For the self-driving car to function properly, appropriate and accurate sign boards are required.
- Self-driving cars have a number of risks, one of which is that they are vulnerable to hacking due to technological advancements.

6.OUTPUT

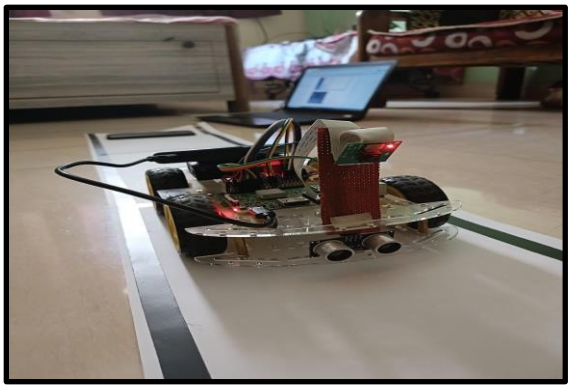

Fig -7: Side View of Hardware

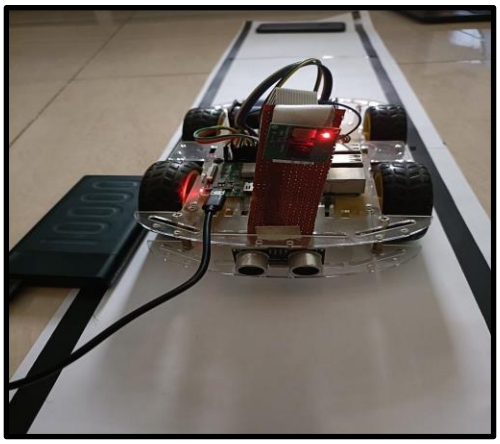

Fig -8: Front view

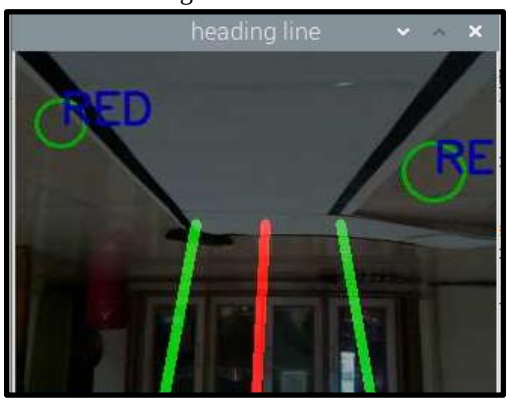

Fig -9: Heading lines

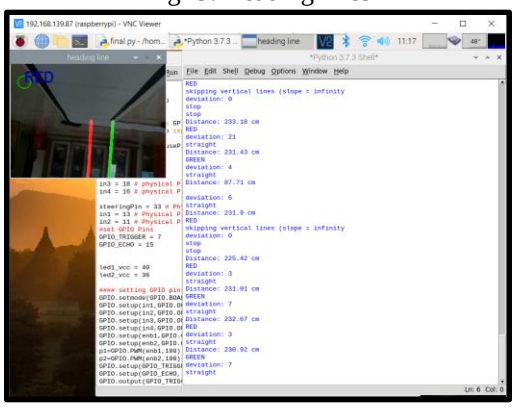

Fig -10: Run-Time

7.CONCLUSION

Our project's goal was to make transportation and travel safer and more enjoyable. On the given road, the car was able to travel automatically from one location to another. The car was successful in accomplishing this because it was able to see upcoming turns, traffic light signals, other vehicles, etc. and make decisions independently and in accordance with those observations. As a result, the vehicle was able to get

around every obstruction and travel safely along the road track.

8.ACKNOWLEDGEMENT

We acknowledge and immensely appreciate the constant guidance and support provided by our parents, friends and our professor Mrs. Dhanashree Pannase.

REFERENCES

[1] “Learning OpenCV 3: computer vision in C++ with the OpenCV library”, A Kaehler, G Bradski
[2] ”The key technology toward the self-driving car”, J Zhao, B Liang, Q Chen
[3] “Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino”, Aditya Kumar Jain