# Development of Self Driving Vehicle

[1]Shrujan BN, [2]Shreyas S Gaekwad, [3]Sagar HK, [4]S Vinaikumar, [5] Karthik S
[1,2,3,4]Student, [5]Assistant Professor
[1,2,3,4,5]School of Mechanical Engineering, REVA University, Bengaluru, India

***Abstract:*** Self-Driving cars are at the forefront of the automotive industry. Autonomous driving is the future. We are basing our vehicle on basic necessities such as identifying and classifying traffic signals, obstacles on the road. Lane switching and maneuvering is also one of the objectives to be accomplished. Artificial Intelligence, Deep Learning, machine Learning are the cornerstones of our development process. The implementation of this idea into the real world would result in a vast number of advantages. Prevention of accidents, safer transport and a huge revolution in transportation**.**

***Index Terms - Self Driving Cars, Artificial Intelligence, Deep Learning.***

## I. INTRODUCTION

A vehicle which can performs all driving task with least or without human intervention. It includes some level of Automation in it, such a vehicle can be claimed as Self-Driving vehicle.
Driving tasks consists of perceiving and analyzing the environment, planning and maneuvering and controlling the vehicle. As human beings the above tasks are like second nature to us and we carry it out with ease. But computers require large number of calculations to carry out similar tasks.
A self-driving vehicle must be able to detect and maneuver lanes with the least amount human interaction as possible.

Vehicle Automation is classified into the following:

**1) Level-0 Automation:**
In such vehicles there will be no automation the drivers have to concentrate on the driving task completely to avoid accidents.
Example – Indica, Tata Sumo etc.

**2) Level-1 Driving Assistance Automation:**
In such vehicles there will be driver assistance for certain task, means driver can ignore any one task which will be automatically controlled by vehicle itself.

- Lateral control - Driver has to control the steering, vehicle will control the speed
- Longitudinal control- Can help to stay in lane but driver has to control the speed.

Example- Mercedes, BMW etc.

**3) Level-2 Partial Driving Automation:**
In such vehicles both the lateral and longitudinal control are automated and vehicle itself manages both the things with itself. So, the driver can partially focus on another task.
Example- Audi, Skoda, Mercedes Benz etc.

**4) Level-3 Conditional Driving Automation:**
In such vehicles with Level-2 automation the vehicle includes automated object, event detection and response (OEDR) so it can able to analyse the environment also. Example-Uber, Volvo.

**5) Level-4 High Driving Automation:**
The system supports self-driving with no or minimal driver intervention, but primarily in mapped locations. Automated lane changing and other features available.
Example-Tesla

**6) Level -5 Full Automation:**
The system requires no human intervention, as in a robo-taxi or robo-truck.
Example-Waymo.

## II. LITERATURE REVIEW

Self-driving cars have been only left to the pages of science until recently when many automotive giants have made leaps and bounds with regards to the technological advancements of autonomous driving [2]. Consumers across the world are enthusiastic about the new era of autonomous vehicular transportation. It gives a glimpse into the various stages of the development of an autonomous vehicle [1]. Daily advancements in the field of sensors and various other equipment have led to less and less human interactions. Many features such as adaptive cruise-control, side-monitoring for safer lane changing, automatically slowing down or braking were implemented and have become the norm. With the help of the algorithm developed, vehicles can be set up to automatically navigate to the destination location by continuously receiving the direction from another vehicle moving ahead in the same direction [2]. Autonomous vehicles must be able to detect and classify objects as vehicles, people, traffic, signs or obstacles and then take appropriate actions all while abiding by traffic rules in all possible conditions. Only a radar is not enough to classify objects and hence a camera is used. Using Optical Character Recognition and Natural Language Processing it was able to detect and classify objects [4]. YOLO was the method chosen to train the camera to perform object detection. This radar-camera fusion provided a robust and accurate method of tracking, identifying and classifying the traffic signs and any other object [4]. A new system named Rider (Real-time Intelligent Driving Environment Recording) was created by MIT [3]. RIDER proved through extensive testing that it has adequate data collection abilities for naturalistic driving research. It seeks to break new grounds in demonstrating how autonomous vehicles and humans interact in the rapidly changing system of transportation [3]. Even with the inevitable rise in autonomous vehicles and the drastic decrease in accidents that will come with it, it is unanimously agreed that it will not completely risk free. It is accepted that the software will make some errors or the error maybe due to human drivers, programming crash algorithms of self-driving cars poses an ethical and legal challenge that is paramount [5]. Many systems of navigation such as location, **Electronic Map** (EM) and **Global Map Planning** are integral and must be worked upon. **Environmental Perception**, **Vehicle Control** are also important aspects.

**LIDAR** is another important technology that is immensely improving on a regular basis, It is a type of optical recognition device which helps classify and identify road sign, obstacles, civilians and many others [8]. The Google self-driving car also relies on a Light Detection and Ranging (LIDAR) system to provide it with a 360-degree view and is also equipped with external radar to provide direct velocity measurements [7]. Unmanned technology itself can have a positive impact on the urban environment and on the ways in which people move around the city only when combined with the development of shared services [6]. Some experts believe the self-driving cars are in for a boom, thanks to how the coronavirus pandemic has started to reshape how we purchase goods. When a top priority is to slow the spread of a contagious virus, having things delivered can help some people [9]. Currently on highways drivers usually keep between 40 to 50m (130 to 160ft) distance away from the car in front of their pathway. These increases in highway capacity sometime are one of the main significant reason for impact in traffic congestion, particularly in the urban areas and more affected in highway congestion in some places [15]. OpenCV and Deep Learning are vital and integral to the training of a robust self-driving car. "The different innovations we are using are obstacles and curb detection method, road vehicle tracker, and checking different traffic situations. This will make a robust autonomous self-driven car." [14]

## III. METHODOLOGY

### 1. Software and Hardware Requirements

- Raspberry Pi 3B+
- Arduino Uno
- Raspicam 2
- Raspbian OS
- Arduino IDE
- Open CV
- L298n H BRIDGE

- **RASPBERRY PI 3B+**

Raspberry Pi is small computer with a processing -power of 1.2 GHz and is a quad-core processor. It is developed and manufactured by the Raspberry Pi Foundation. It consists of both a Wi-Fi and a Bluetooth module. It is operated by using the Raspbian OS which is also designed and developed by the Raspberry Pi Foundation. The coding for the self-driving car is carried out using this OS.

Figure 1: Raspberry Pi 3B+

- **ARDUINO UNO**

The Arduino Uno is a small micro-controller with 20 input/output digital pins. Jumper cables are used to connect the Arduino Uno to various devices. It also comes with an USB adapter which is used to connect the Arduino UNO to the computer.

Figure 2: Arduino UNO

- **RASPICAM 2**

The Raspicam 2 is a camera module developed by the Raspberry Pi Foundation and is compatible with the Raspberry Pi. The v2 Camera Module has a Sony IMX219 8-megapixel sensor. The Camera Module can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners, but has plenty to offer advanced users. This piece of hardware is used of lane detection and object identification. The feed from the camera can be seen in the Raspbian OS.

Figure 3: Raspicam v2

- **RASPBIAN OS**

This is an operating system designed and developed by the Raspberry Pi Foundation. It is a Debian-based Operating System. It is optimized to a high degree specifically for the Raspberry Pi. It uses the ARM Architecture. This OS is where all the coding for the Vehicle takes place.

Figure 4: Raspbian OS

- **ARDUINO IDE**

It is an open-source software package designed and developed by Arduino. The code required for the movement of the motors of the vehicle are compiled in this environment and then uploaded on to the board.

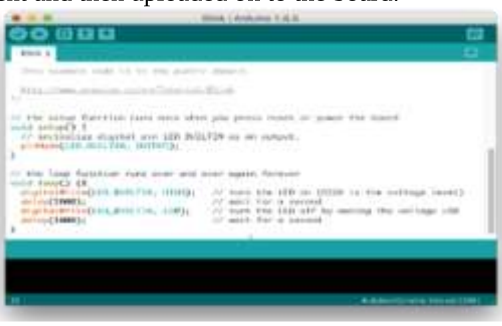

Figure 5: Arduino IDE

- **OPEN CV**

Open CV is an open-source programming package which deals with real time computer vision. It is used here for image processing. The leading need for Open is for Lane Detection and object detection.

Figure 6: OpenCV

- **L298N H BRIDGE**

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors. This is connected to the Arduino UNO which will operate the motors of the vehicle.

Figure 7: L298N H Bridge

- **Experimental Setup**

A metal frame was taken and the four wheels of the vehicle along with their motors are then attached to chassis. The wires from the DC motors are then cross-coupled to give uniform distribution of charge to both sets of motors and to ensure that they run in the same direction. These wires are then attached to the L298n H Bridge. The screws are tightened to ensure no loose connections. Using jumper cables, connections are made in the required way from the H Bridge and an Arduino UNO. The movement of these motors will be controlled by this Arduino UNO. A Raspberry Pi is then taken and fixed onto the chassis. To this, the Raspicam v2 is attached at a height such that a full view of the track can be seen by the camera module. All these components are then given power from two separate power sources. One source for all the motors and for the Arduino board and the H Bridge. Another source is then used to solely power both the Raspberry Pi and the Raspicam v2. A custom track is then built for testing purposes out of multiple cardboard sheets and has white strips of thin paper to represent lanes. This is the complete hardware setup of the vehicle.

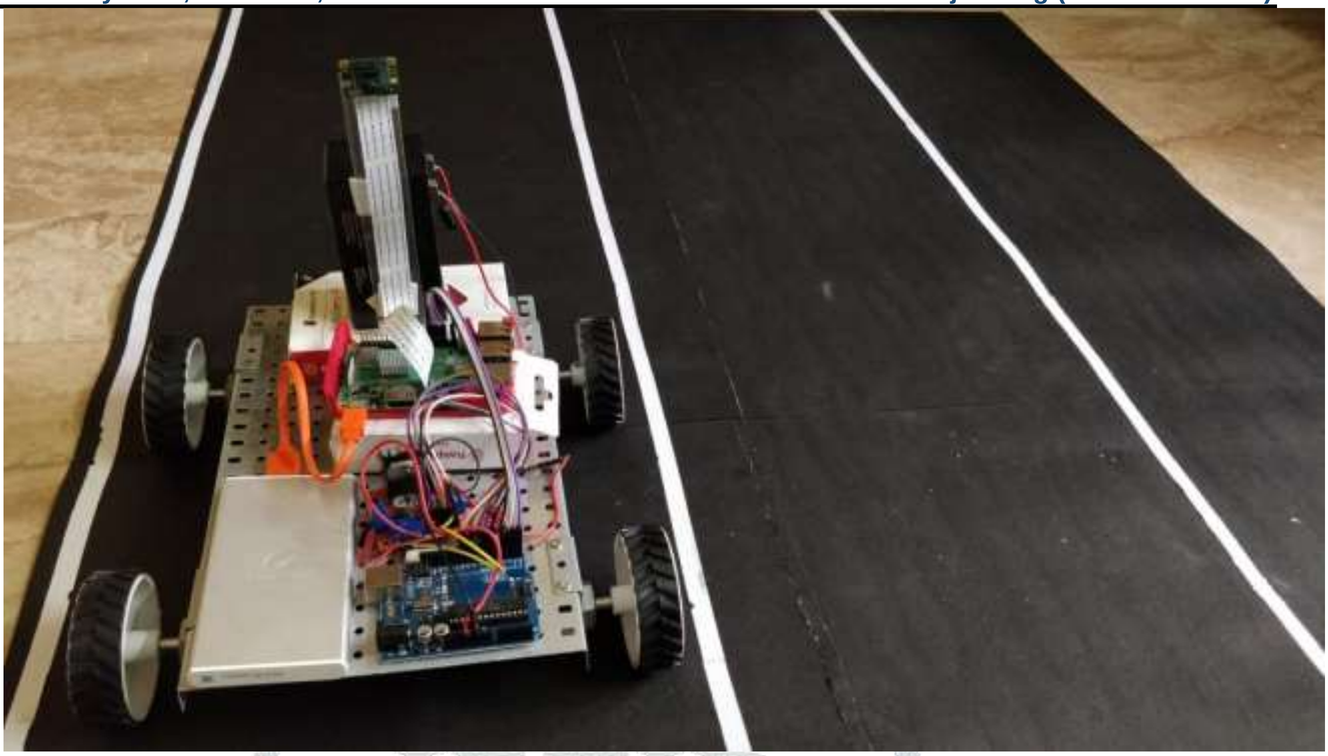

Figure 8: Experimental Setup

- **Software Implementation**

After setup of the hardware components are finalized, the next step is to carry out the software implementation. First step is to test the working condition of the motors. In order to do this a basic program is compiled and uploaded to the Arduino UNO by using the Arduino IDE. The motors should move upon supplying all the components with power from the battery source. After completion of this step, next is to download and install Raspbian OS onto the Pi unit.

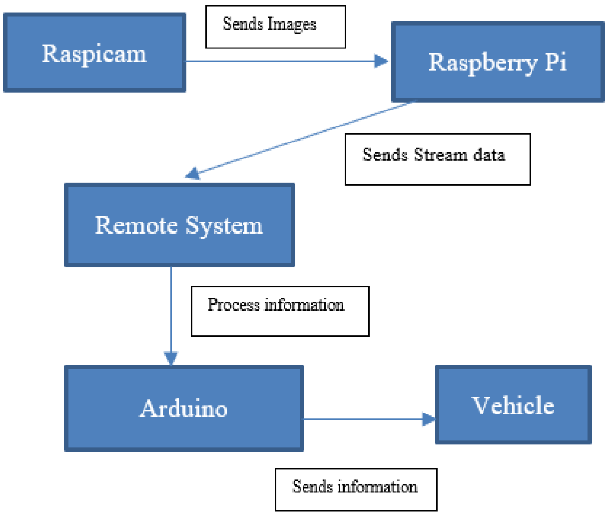


Figure 9: Block diagram for software implementation

- **Initializing the camera**

After installing the Raspbian OS onto the Raspberry Pi, the next major step is to install OpenCV. This is done so that image processing can take place. Then the camera module is setup and all its pre-requisites are installed. Then a few images are captured and the smoothness of the framerate are calibrated until the desired value is reached.

Figure 10: Camera Initialization

- **Lane Detection**

Programs are written using the Geany Editor. A program is first written for the identification of the lane. The feed received from the camera is in the BGR color space, so this must be converted into RGB color space. Next a Region of Interest or ROI is created so that the lane can be identified and followed by the vehicle. We use cv2.getperspective and cv2.wrapperspective to perform bird's eye view with given x and y points. We convert the given image into grey scale image as it will be helpful in canny edge detection. Gaussian filters are used to reduce the noise and remove any unnecessary details. With the help of histograms, we find the exact lane positions. The Raspberry Pi and the Arduino Uno is connected via cables and the link is between them on what action to perform according to the decimal numbers. With this successfully functioning, lane detection is successfully established with the help of OpenCV and image processing and recognition. Region of Interest and edge detection also play an integral role.

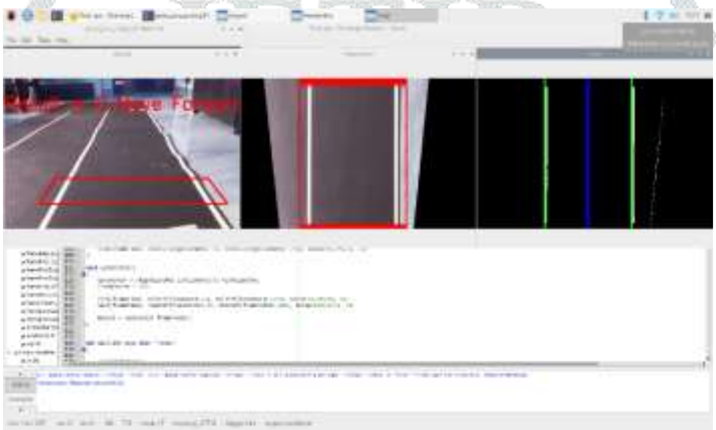

Figure 11: Lane Detection

- **Lane Maneuvering**

After lane detection and programming is completed, the next step is to write a program to interface between the Raspberry Pi and the Arduino UNO. When testing takes place, the vehicle is placed on the track. It is switched ON and senses the lanes which are represented by thin white strips. As the lane changes the vehicle must be able to detect these changes and maneuver accordingly. This means that if the lane changes to the left it will take a left turn and if changes to the right a right turn. The Raspicam module plays an integral part in lane maneuvering with the help of image processing. Thus, upon completion of this tasks, we can state that the vehicle has achieved lane maneuvering. Longitudinal speed and lateral lane change must be considered while conducting lane maneuvering.

## IV. Experimental results

The Raspberry Pi and the camera module attached will provide us with a video stream which is viewable on the desktop/laptop setup. From the image processing that takes place on the Raspberry Pi, instructions are sent in conjunction to the Arduino UNO and the whole setup operates to the required level. Hence while driving, the camera interfaced with the vehicle, observes some of the parameters of the lane detection with the help of image processing, if it detects these parameters then it will be recorded and the vehicle then acts according to this parameter and shifts by helping the vehicle follow the lane accordingly. On the test track it is placed and is switched ON. We are able to observe that as the vehicle moves along the track any change in direction in the track is met with and adjusted accordingly by the vehicle as it is able to change direction according to its lane orientation. Thus, reaching our objective of lane maneuvering. As more testing of the vehicle is done and on a larger scale, a more optimized usage of this vehicle can be found.

## V. FURTHER IMPROVEMENTS

Autonomous and electric vehicles are a hot topic in the automotive industry. The future of the automotive segment of the market lies in self-driving vehicles. The strides to be taken now are to achieve higher levels of automation. Achieving the highest level of automation that is level 5 automation is the end goal. Where there will be no human interaction required in the automation of the vehicle. These innovations may take nearly decades in order for it to reach public commonplace. The penultimate objective is to bring about more efficient use of infrastructure, improve the protection and luxury of users, and pave the way for the revolution in self-driving car. There should be made a sincere effort to reduce accident occurrences. With this we can truly say autonomous technologies and self-driving vehicles are a necessary innovation for the future.

## VI. CONCLUSION

This paper explains the various components and equipment which were required for the assembly of the self-driving vehicle. OpenCV and Raspberry PI play an integral role to help in the image processing and recognition in identifying the lanes and properly maneuvering them. The scope of this vehicle is wide and can have many applications in the industries of transportation of goods, military, agriculture, commercial etc. The algorithms and structure of the program was implemented on a small model of a vehicle. The Self-Driving vehicle has limitless potential for the future and will be a big part of the automotive sector.

## REFERENCES

**[1]** Keshav Bimbraw. Autonomous Cars: Past, Present and Future:3-5.
**[2]** Qudsia Memon, Muzamil Ahmed. Self Driving and Relaxing Vehicle:1-5.
**[3]** Daniel E Brown, Michael Glazer, Bryan Reimer. MIT Advanced Vehicle Technology Study. Large-Scale Naturalistic Driving Study of Driver Behaviour and Interaction with Automation:5-16.
**[4]** Shrey Shah. Accurate Identification of Traffic Signs Using Radar and Camera Fusion:6-19.
[**5**] Ivo Coca Villa. Self-Driving Cars in A Dilemmatic Situation.4-20.
[**6**] Aleksey Zomarev, Maria Rozhenko. Impact of Self-driving Cars for Urban Development:3-14.
[**7**] Michelle Birdsall. Google and ITE: The Road Ahead for Self-Driving Cars:4-8.
[**8**] Jianfeng Zhao, Boadong Liang, Qiuxia Chen. The Key Technology Toward the Self-Driving Car:3-7.
[**9**] Sean Szymkowski. “KFC rolls out self-driving food trucks in China and it's sort of wild”, November 2020:1-4
[**10**] Maarten Vinkhuyzen. Tesla Dojo Supercomputer Explained — How To Make Full Self-Driving AI, November 21, 2020:1-4.
[**11**] Harish Kumar. Check out This Self-Driving Tata Hexa from Europe, October 2020:1-4.
[**12**] Self Driving Cars – Coursera.
[**13**] Alex Davies. Self-driving trucks are now delivering refrigerators. *Wired* (Nov. 13, 2017)**:**1-4**.**
[**14**] MS. Sujetha, Chitrak Bari, Gareja Prdip, THE YAN: Self Driving Car using Deep Learning Technology:5-9
[**15**] Autonomous Self-Driving Cars using Raspberry Pi Model, International Journal of Engineering Research & Technology:4-9.