SELF DRIVING CAR

Sanjay Revanna [1] Rohan A Mangsuli [1] A Saiteja [1]
Shashank S [1] Dr. Leena Giri G [1]

[1]Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore,560056
Corresponding Author: Sanjay Revanna(sanjayrevanna7@gmail.com)

***Abstract*— Since the introduction of the first car in 1886 by Carl Benz, the car industry has now grown into a major sector all around the world and is one of the highest in demand commodity across the globe and today there are over 50 different car manufacturers manufacturing cars of different shape size and purpose. In today's era cars are just a machine used to travel from point A to B with the need of human assistance, but also that can drive by itself and these cars are known as the self-driving cars. Here, we develop an autonomous car model that is capable of driving on its own with the help of the computer vision technology by detecting the lane marking of the road and automatically adjusting the steering angle, speed and the direction it shall travel towards. The car model shall also be able to detect the road end or road closure and take the proper U-turn and is able to detect the road signs such as stop signs and traffic signal using the machine learning technology.**

Keywords—Self driving car, Machine Learning, Computer Vision, Lane Marking, Road signs.

I.INTRODUCTION

In our day to day life we use cars as a mode of transport to travel from point A to point B and cars as such is a very complex machine with a lots of mechanical components such as engine, suspension, brakes which are required to make the car move and electrical components such as lights, battery, A/C, infotainment system etc. But in today's modern and technological advanced world, there has been a huge leap in development of the autonomous or the self -driving car which are capable of driving on its own without the need of the human intervention by following the road lane markings using technology such as Opencv software and machine learning for road traffic sign detection.

Making blunder while driving is one of the riskiest and causes loss of lives and leads to Accident. The common error that humans make like conversation on the phone while driving or by playing loud music in car the accidents are more probable. Apart from these mistakes mental and physical disabilities are also a cause in driving mistakes. These errors are accelerating day by day and have become more crucial to reduce them by modern technology. Self-driving cars are the resolution not just to minimize these errors but also to new prospect of our driving and efficient road administration systems. With the waterway change in technology, Automobile Engineers are coming up with new thought in the field of self-driving cars. These cars are sovereign cars that can drive by themselves without human intervention. This is basically a scale model of self-driving car in best available resources. In this model, we have used a Raspberry Pi controller and Arduino with LH-bridge to drive motors. The car is being powered by two 5000mah power bank.

The self-driving car would annihilate human participation in the field of driving making driving safer and in this self-driving car, we are using raspberry pi (4 model B) which has a built-in Wi-Fi and Bluetooth module, as the main processing chip. This raspberry pi module has capability to compute the result like a computer. We can even connect the mouse and keyboard to this component. We have used a 16GB micro-SD card as storage device to store booting files and installed the Raspbian legacy OS. The input to it is provided by uninterrupted flowing of images via Raspi Cam. This Raspi Cam is being connected to the raspberry pi component through the CSI cable. This input is sent to the processing algorithm over the private host. Here we use Computer Vision and Machine Learning to evaluate the output which is given to the Arduino

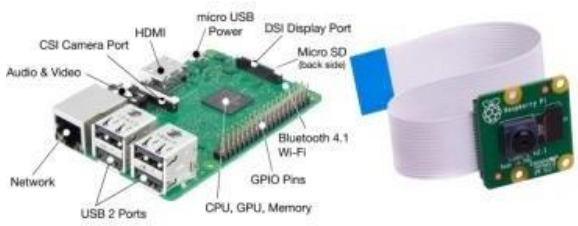


Fig -1: Raspberry Pi (4 Model B)

Fig -2: Car model

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). With a 14-pin configuration for input/output, a USB connection, a power jack, an ICSP header, and a reset button, it has everything requisite to support the microcontroller. This component can be well connected to any computer using a USB cable and can be approached by the environment called Arduino IDE. This IDE has a text editor with a toolbar for common utility and a number of menus. The raspberry pi component gives input to Arduino through digital pins (0,1,2,3) The Arduino feeds the proper signal to the LH Bridge through the residual of the pins to power the motors accordingly.

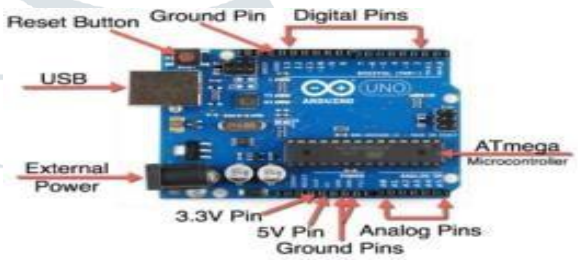


Fig -3: Arduino UNO

The L298 known as a dual LH-bridge is an integrated circuit in 15-lead Multi watt and PowerSO20 packages. It is mostly used to invert the polarity of the motors attached to it. This circuit has a high voltage and high current dual bridge driver which can judge TTL logic levels provided by Arduino. By using this constituent, the self-driving car can easily stop and run according to the information given.

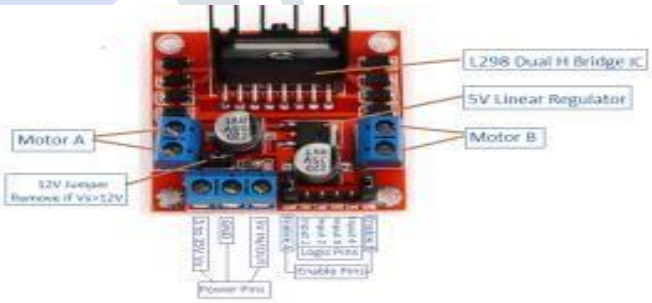


Fig -4: L289 H bridge

II. RELATED WORKS

In [1], Introduction to Open CV fundamentals, induction of Open CV and its libraries and how to process images and videos in open CV. Building GUI and primary filtering operations on the images. Object partition and detection using image pre-processing methods such as thresholding and contouring. Eventually, image processing using object recognition and machine learning algorithms.

In [2], Visual recognition work for example image categorization, localization, and detection, are the starring building blocks of many of independent applications, and recent developments in Convolutional Neural Networks (CNNs) have led to prominent execution in visual recognition work. This application is for aggregate object detection in a given video based on Open Computer Vision libraries.

In [3], Car piloting system, path planning, environment conceptualization and car control are the four main technologies concerned in Self driving car. An elaborate review about all four of them and how they are brought together in this project is proposed.

In [4], This paper declares a working example of self-driving car that is surefooted of driving safely or to say on various types of roads such as curved roads, straight roads and continuous followed by curved roads. A camera component is placed with proper altitude on one end of the car along with Raspberry Pi. This sends real time images to the Convolutional Neural Network which in turn anticipate the course to be locomote by the car i.e., right, left, forward, stop or reverse this instruction are sent from the Arduino to the motor controller as a outcome the car travels in the coveted direction without any human errors.

III. METHODOLOGY

Initial setup/design: The first step is to set up the self-driving car scale model which includes the car chassis on which the various components which are - Raspberry Pi, Arduino Uno, Raspicam, DC motors, L298H Bridge, Power bank are all stacked upon one another and the required connections are made between each other.

The next step includes setting up of the environment in which the car operates, for this we would be creating an artificial road with lane markings, traffic signals, stop sign, obstacles (such as other cars) using a black cardboard sheet on which the white lane markings would be done. The markings are required in order for the car to move in a particular lane.

We need to setup the Arduino Uno for controlling the speed and steering inputs of the car. The L298H bridge acts as a motor driver which connects and control all the 4 DC motors. We write the program code for motor speed control and direction using C++ in the Arduino IDE. Then we ensure the proper functioning of the motors through trail testing on the artificial road that we made.

The Raspberry Pi interface will be created through the laptop which enables us to connect with the Raspberry Pi through Wi-Fi. We install the Raspbian Legacy OS which is done by the Raspberry Pi Imager tool. The OS that is installed is then flashed in the Raspberry Pi. The various libraries such as the OpenCV library, WiringPi library and others that are required are then installed on to the Raspberry Pi.

The following block diagram gives us an idea of how exactly the self-driving car functions:

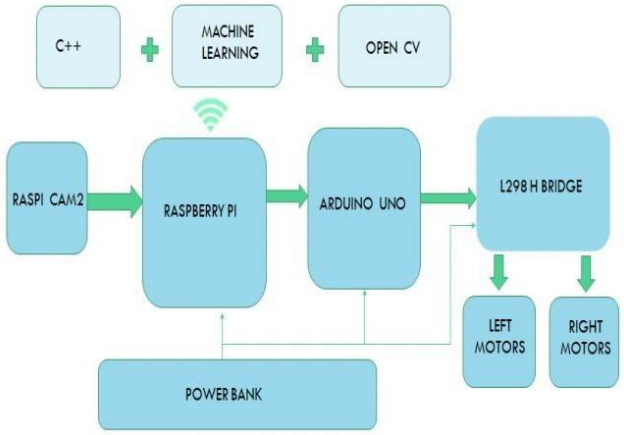


Fig -5: System Architecture

IV.IMPLEMENTATION

Image processing: Since the necessary image processing libraries are already installed in our initial step, we can directly jump into the image processing step. The input to the car is taken as a stream of images by the Raspicam. Since the OpenCV libraries function based on the BGR color scheme, the required changes from RGB to BGR are made to the given input using the appropriate functions. Then as per our environment, the region of interest (ROI) is created. ROI can be defined as the region the camera is currently focusing on for its navigation. The region of interest that is created undergoes perspective transformation for appropriate image analysis. Once that is done, image thresholding is done to convert the grayscale image captured by the Raspicam into the black and white image. On this threshold image, canny edge detection is done and then the threshold and the canny images are cascaded together.

Lane Detection: The next step after image processing is the detection of lanes by the car for its navigation. For this, we find the distance of the white pixels from the left side of the ROI. This distance is found to be 255 units. The pixels are then treated like vectors to find their positions, for which the histogram function is used. The appropriate commands are sent to the Arduino UNO for its navigation depending on this distance that is calculated. If the distance of the white pixel's increases, then the car is advised to take a right turn. If the distance decreases, then the car is advised to take a left turn. The steering angle (angle by which the car has to turn left or right) is calibrated depending on the distance calculated. A flowchart for how the lane detection is carried out can be shown as follows:

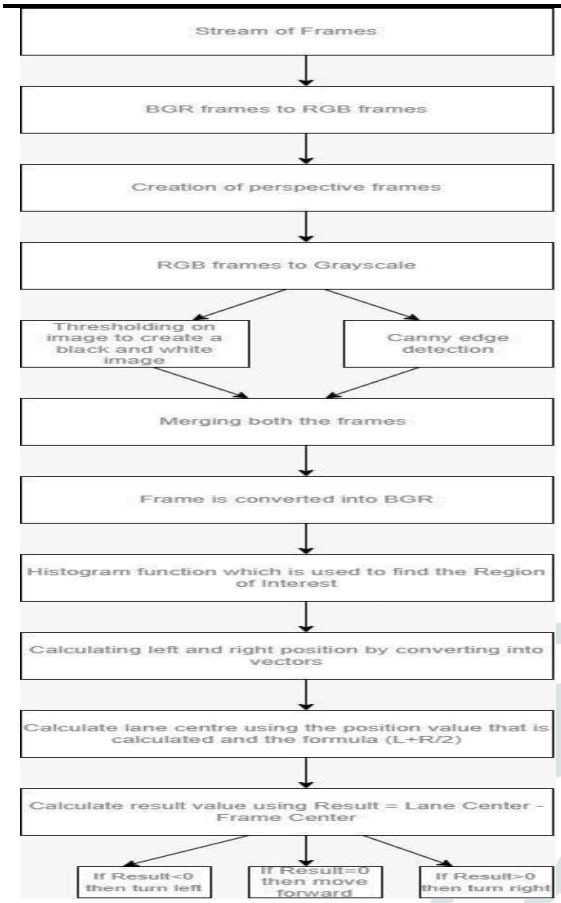


Fig -6: Flow chart for Image Processing and Lane Detection

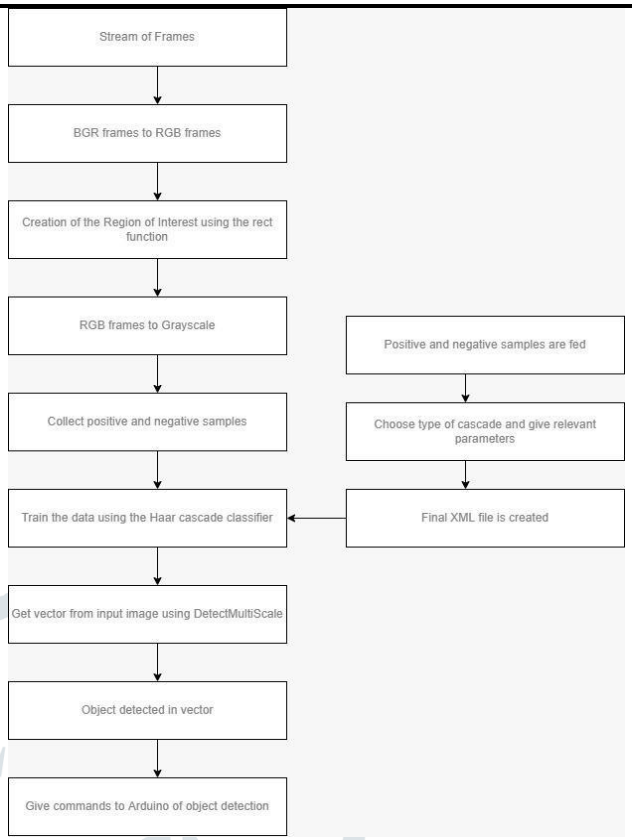


Fig -7: Flow chart for Object Detection

Lane end Detection: There is a white strip that is kept on the track at its end. This white strip indicates that the car has reached the end of the lane. For the car to detect that it has reached the end of the lane, the value of the intensity of the white strip is calculated. If the intensity is within a certain range of values, then the car realizes that it has reached the end of the lane and that it has to take a U turn.

Object detection: For the car to detect objects on the track, some concepts of machine learning are used. Three different types of objects are detected by the car - stop sign, traffic sign and a car that is kept in its way. There are two types of samples that are used for the training of the algorithm - positive samples and negative samples. Positive samples contain images of the object that has to be detected and negative samples contain all the samples of the image that should not be detected. For the training of the data, the Haar Cascade classifier is used. After this, in the function for detecting the target object, OpenCV methods are used for identifying the detected object and using the formula for a line equation: y=mx+c, the distance between the camera and the object. Once the distance is known, another function for halting the car as required is written.

Arduino UNO programming: There are three main parameters that are used for programming the Arduino, which are:
ENABLE - this is used for switching the respective motors ON.
HIGH and LOW - changing these parameters would result in making the motors move in forward and reverse direction as required. For the car to steer itself, the left and right turns have been divided into three different angles of degree of inclination. For e.g., if there is a sharp right turn the speed of the right motor is decreased close to the maximum extent while if there is a slight turn, the speed would not be decreased as much. The vice versa happens when the car has to take a left turn. If the speed of the motors is reduced to zero, then the car comes to a complete halt.

V. RESULTS

Lane Detection: While a self-driving cars integrated a wide array of technologies (deep learning, Image Processing and Neural Networks) supported by data from raspberry PI camera, one of the comparatively simpler tasks is to detect the lane markings on the road to find lanes while driving.

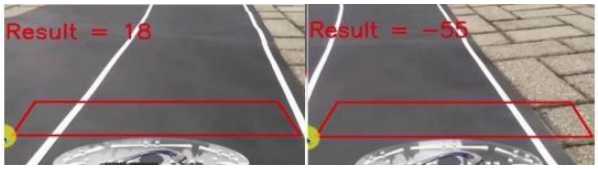
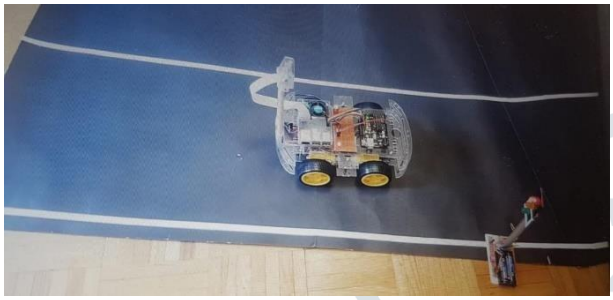


Fig -8: Lane Detection

Traffic signal Detection: when the car is made to run on road it receives data from the Raspicam. When there is a traffic signal detected, the color of the traffic signal is checked for. If red is detected, then the car comes to a complete halt. When the signal turns green, the car continues to move along its lane.



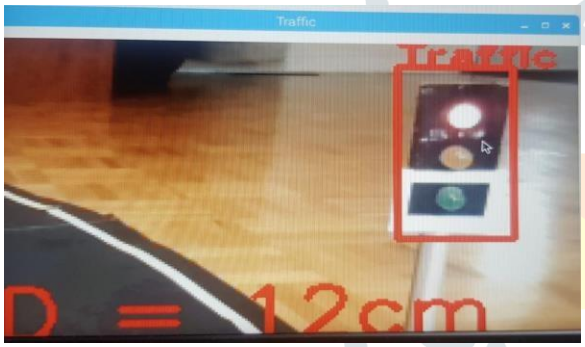


Fig -9: Traffic Signal Detection

Stop sign: When the car is moving along the road following the lanes it will detect the stop sign that is placed. Once detected it will stop for a few seconds and continues moving along the same lane.

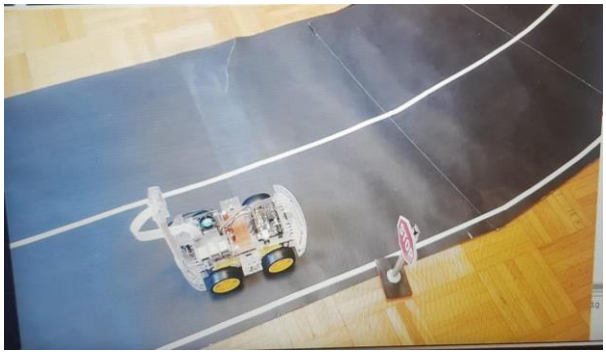


Fig -10: Traffic Signal Detection

Lane end detection: the unique and the safety feature of the self-driving car is that it can detect the dead end of the road and make a uturn is such a way that it again starts travelling along the adjacent lane in the opposite direction.

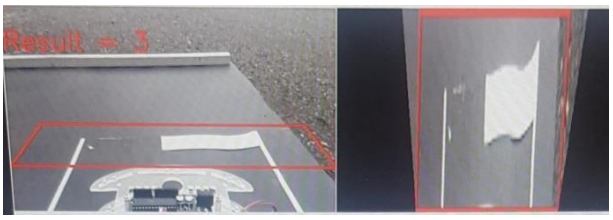


Fig -11: Traffic Signal Detection

VI. LIMITATIONS

For a self-driving car to become a success in India, the first thing which needs to be improved is infrastructure. Without proper road infrastructure, the cars would not be able to take veracious decisions on their own. One of the major issues which are present is the communication of self-driving cars and human drivers on roads. Human drivers drive their car intuitionally and their actions on roads become unreadable for self-driving cars in real-time which can prove dangerous for all. The working efficiency of equipment reduces while functioning in utmost weather conditions. Thus, the time taken in the process is significantly affected and is not dependable. Sometimes when the weather is cloudy or rainy wrong input signals also appear which may lead to unanticipated results. Correct and accurate signboards are necessary for the self-driving car to operate safely. One of the major threats to self-driving cars is that they are inclined to hacking because of Grey hat hackers

VII. FUTURE SCOPE

The current self-driving car can be enhanced if the car is equipped with the GPS system, it would reach the Respective location on its own thus making it fully involuntary. Furthermore, the car should be able to drive in dark and rainy conditions with the improvement of the camera and processor technology.

VIII. CONCLUSION

The project Self Driving Car is successfully enforced and tested. The car is trained with data set of images in different occasion. For experimentation we use high resolution raspberry pi camera. The Traffic signal and stop sign detection was successfully implemented using sensors which were placed at the front of the car cascate to capture the image and when a reflection is received then output is conveyed to the Arduino micro controller. This paper clearly describes the practical methodology of our self-driving car. Using detection sensor car detect the traffic signal. The traffic light detection system was very accurate for the given situation. The car could be successfully detect the placement and representation of traffic light Signal. The car is trained to detect the end of the road and automatically takes a Uturn and moves along adjacent lane of the opposite direction.

REFERENCES

[1] "Learning OpenCV 3: computer vision in C++ with the OpenCV library", A Kaehler, G Bradski.

[2] "Object Detection and Classification for Self-Driving Cars" Ruchita Sheri1, Niraj Jadhav2, Rahul Ravi3, Ankita Shikhare4, Sanjeev Sannakki5.

[3] "The key technology toward the self-driving car", J Zhao, B Liang, Q Chen.

[4] "Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino", Aditya Kumar Jain.

[5] "Build Your Own Self Driving Car Deep Learning, Opencv,C++,"https://www.udemy.com/course/selfdrivingcar/learn/lecture/15544112?start=15#overview