iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 11 **Issue:** XI **Month of publication:** November 2023

**DOI:** https://doi.org/10.22214/ijraset.2023.56780

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue XI Nov 2023- Available at www.ijraset.com*

# Autonomous Vehicle

Asst.Prof. Kavita Jain[1], Sean Pereira[2], Vrishab Shetty[3], Lysandra D'Souza[4], Akash Mathew[5]
[1]*Asst.Professor,* [2, 3, 4, 5]*Undergraduate Student, Department of Computer Science and Engineering, Xavier Institute of Engineering, Mumbai, India*

***Abstract:*** ***Self-driving cars are a rapidly advancing technology that is poised to revolutionize the way people travel. The purpose of the self-driving car project is to create a vehicle that can operate autonomously, without the need for a human driver. This technology is being developed to increase safety on the road, reduce traffic congestion, and provide a more efficient and convenient mode of transportation. The project involves a combination of hardware and software development, including sensors, cameras, and machine learning algorithms. Self-driving cars have the potential to reduce the number of accidents caused by human error, improve traffic flow, and provide greater mobility for individuals who are unable to drive. While there are still many technical and regulatory challenges to overcome, self-driving cars represent an exciting future for the automotive industry and society as a whole.***
***Keywords: Self Driving, Canny Edge, Raspberry Pi, Object Detection, HAAR Cascade, Signal Detection, Arduino Uno.***

## I. INTRODUCTION

Cars are now an integral part of our daily transport or commutation, be it public transportation or private. The concept of self-driving cars has been a topic of discussion for several years, and with the advancements in technology, it has become a reality. Self-driving cars are autonomous vehicles that can navigate and operate without human intervention. These vehicles use a combination of sensors, software, and communication technology to sense their environment, identify objects, and make decisions based on that information.

The development of self-driving cars has the potential to revolutionize the transportation industry, improve road safety, and reduce traffic congestion. The aim of our self-driving car project is to develop a fully autonomous vehicle that can operate on public roads safely and efficiently. Our project will involve designing and implementing a range of technologies, including object detection and tracking, path planning, and control systems. We will be using a variety of sensors to provide real-time data about the environment around the vehicle. Our self-driving car will also be equipped with a communication system that will allow it to interact with other vehicles on the road and communicate with the infrastructure. This will enable the car to make informed decisions about its route, speed, and other critical parameters. One of the significant challenges in developing a self-driving car is ensuring its safety and reliability. To address this, we will be using a range of testing and validation techniques to ensure that our system performs reliably and meets the required safety standards.

## II. LITERATURE SURVEY

### *A. AI Algorithm for road signs and traffic signal*

The model prototype which is developed aims to implement automation by handling tasks such as self-driving through lane line detection, stop sign and traffic signal detection and fore collision avoidance. The system design for implementing the same will consist of three units first is input unit containing a pi camera and ultrasonic sensor, the second is processing unit which is our laptop that will act as a server, the neural network will be running over here and third, is RC control unit which is Arduino. Firstly, in the input unit, the raspberry pi board of the B+ model will be connected with the raspberry pi camera and an ultrasonic sensor to stream input data[1].

There will be two client programs running on raspberry pi one to stream im- ages collected by the pi camera and another to send sensor data through a local wi-fi connection. Then the processing handles more than one tasks such as collecting data from raspberry pi, neural network training and steering prediction, stop sign and signal detection, distance measurement using monocular vision and sending commands to Arduino through a USB connection. A multithreaded TCP server program will execute on the raspberry pi to receive image frames and sensor data. The image frames will be converted to grayscale and decoded into NumPy arrays. The neural network which is the convolutional neural network will be trained to make steering predictions based on detected lane markings.

The lower half from the input image will be used for training and prediction purposes. There will be input, hidden and output layers, there will be four nodes in the output layer each corresponding to steering control instructions that are left, right, forward and reverse. The training data can be collected or the datasets already available can be used. For training, each frame will be cropped and transformed to a NumPy array. Then the train image will be paired with the train label. Then all the paired image data and labels will be stored in the npz file. OpenCV will be used to train the neural network After training the weights will be stored in an XML file and for generating predictions the same neural network will be created and loaded with the trained XML file.

### *B. Prototype of Autonomous Car Using Raspberry Pi*

There are two sub-system. That is the Image processing sub-system and obstacle detection subsystem. Camera attached to image processing subsystem which captures the image and provided system. System extracts the data from the image and generates the command about turn. Mainly image processing is used here to detect the road lane. Generated commands are for- ward to obstacle detection subsystems. Obstacle detection sub-system is to detect the obstacle in front of the car and also calculate the distance between the obstacle and the car. And if sufficient distance is available to move the car forward the command from Raspberry pi is forwarded to the motor driver else this command is rejected.

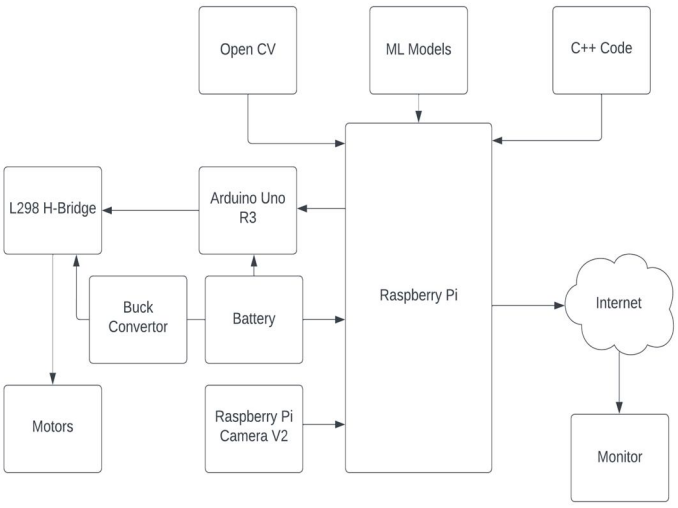
#### *1) Image processing sub-system algorithm*

Here in this algorithm Lane detection is done in the region of interest only as described below. Region of interest is marked red and the columns are helpful to decide the position of lane and also decision of turn is also carried out on the basis of this. Consider this box is the picture captured by the camera. Numbers of white pixels in each block with respect to dark pixels are calculated respectively. This number decides the command for the car. After the image analysis the image, raspberry pi generates the command. These generated commands are forwarded to the arduino board through GPIO pins. The work could be enhanced by improving the algorithm by adding advanced machine learning to it. Using advanced algorithms we can improve Image processing algorithms. Multilayered processors can be used for fast processing.

#### *2) Obstacle detection sub-system algorithm*

In this algorithm Obstacle detection is done using Computer Vision. The algorithm reads the commands from the raspberry pi and checks the ultrasonic distance. If the ultrasonic distance between car and the obstacle is less than 10cm, which is the threshold for halting. Arduino gives a stop command from the library to the driver, otherwise it gives a pass command from the same library to the driver. This process cycles through until the car ends its journey. The present obstacle detection algorithm just detects the obstacle and stops, but in future it can be improved by avoiding the obstacle, and go through another way using advanced obstacle detection algorithm.

## III. SYSTEM ARCHITECTURE




Block Diagram

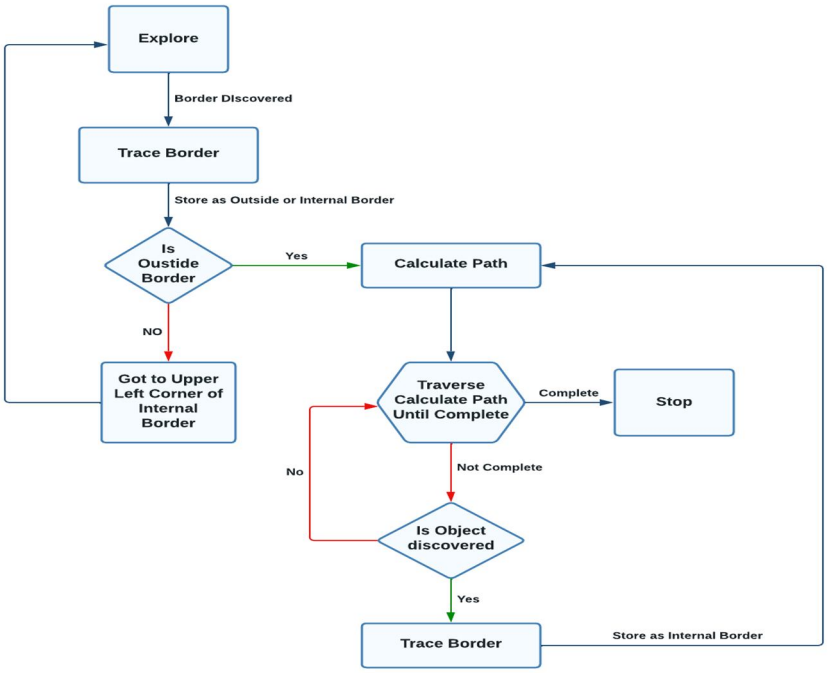
*A. Hardware*

*1)* Raspberry Pi 4b
*2)* RPi V2 Camera
*3)* Arduino UNO R3
*4)* Robot Chassis Kit
*5)* L238 H Bridge
*6)* Buck Convertor
*7)* Power Bank
*8)* Battery 9v
*9)* 6v DC Motors

*B. Software*

*1)* Open CV
*2)* Cascade Trainer
*3)* Raspbian OS
*4)* HAAR Cascade
*5)* Arduino IDE

*C. Flowchart*

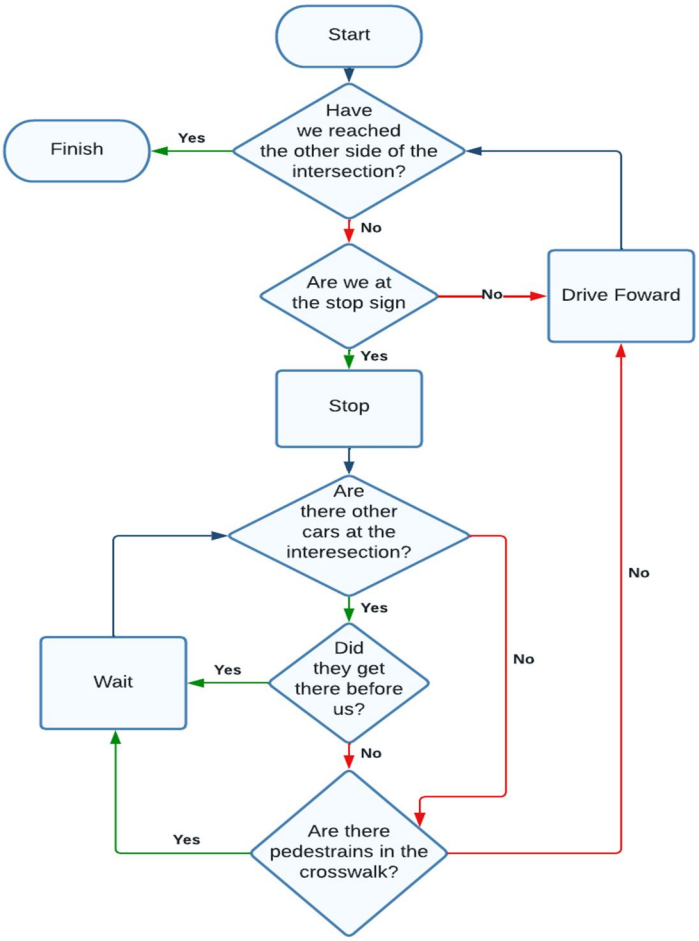



Flowchart for Lane Detection

The Flowchart above represents the algorithms used for lane detection.

*1)* Start
*2)* Capture image from camera
*3)* Convert image to grayscale
*4)* Apply Gaussian blur to reduce noise
*5)* Perform edge detection using the Canny algorithm
*6)* Select the region of interest using a polygon to define the area where lanes are expected to appear
*7)* Apply Hough transform to detect lines that represent the lanes
*8)* Check if lines are within a certain slope and distance range to filter out noise

*9)* Calculate the average position of the lanes
*10)* Mark the lanes on the original image using a different color and thickness to distinguish them from the rest of the image
*11)* Display the marked image
*12)* End




Flowchart for Object Detection

The following is a general outline of the flow chart:

*a)* Data is collected from the RPi camera module. This data includes images, point clouds, and signals.
*b)* The self-driving car's cameras capture an image or video feed of the surrounding environ-ment.
*c)* To simplify processing, the image is converted from RGB to grayscale.
*d)* The pre-trained HAAR Cascade models for object, signal, and stop sign detection are loaded into the program.
*e)* The object detection model is applied to the gray scale image. This model can identify potential objects of interest, such as cars, pedestrians, or buildings.
*f)* For each potential object identified in step 4, the program checks whether it matches the characteristics of a signal or stop sign using the signal and stop sign detection models. These models are trained to recognize specific patterns or shapes that correspond to signals or stop signs.
*g)* If a potential object matches the characteristics of a signal or stop sign, it is marked as such.
*h)* The detected objects, signals, and stop signs are overlaid onto the original image to generate a visual output. This output can be displayed to the car's passengers or used to inform the car's decision-making process.
*i)* The output is used to inform the self-driving car's decision-making process. For example, if a stop sign is detected, the car may slow down or come to a complete stop.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue XI Nov 2023- Available at www.ijraset.com*

*D. Working*

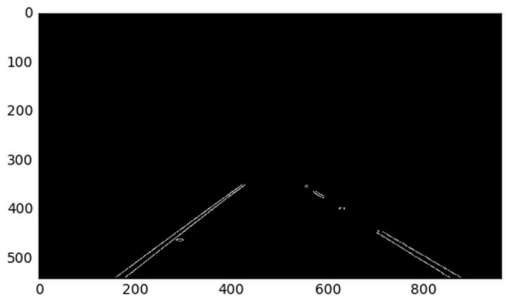
*1) Canny Edge Detection*

The Canny edge detection algorithm works by first smoothing the image using a Gaussian filter to reduce noise and remove small details. The magnitude of the gradient is then calculated, and non-maximum suppression is applied to thin the edges and reduce false positives.

```
void Threshold()
{
      cvtColor(framePers, frameGray, COLOR_RGB2GRAY);
      inRange(frameGray, 170, 255, frameThresh);
      Canny(frameGray,frameEdge, 400, 550, 3, false);
      add(frameThresh, frameEdge, frameFinal);
      cvtColor(frameFinal, frameFinal, COLOR_GRAY2RGB);
      cvtColor(frameFinal, frameFinalDuplicate, COLOR_RGB2BGR);
      cvtColor(frameFinal, frameFinalDuplicate1, COLOR_RGB2BGR);

}
```

Finally, a threshold is applied to the resulting image to identify strong edges. The algo-rithm can also perform hysteresis thresholding, which involves applying two thresholds: a high threshold to detect strong edges and a low threshold to detect weaker edges




*2) HAAR Cascade*

The Haar cascade algorithm works by training a machine learning model using positive and negative images of the object to be detected. The positive images are images that contain the object, while the negative images do not. The algorithm then extracts features from these images using Haar-like features, which are essentially rectangular patterns of pixels with different intensity values.


Figure 3.5: Positive and Negative Samples captured

After the features are extracted, the algorithm trains a classifier using a machine learning algorithm, such as the AdaBoost algorithm, to differentiate between positive and negative samples. Once the classifier is trained, it can be used to detect the object in new images by sliding a window across the image and applying the classifier to each window.

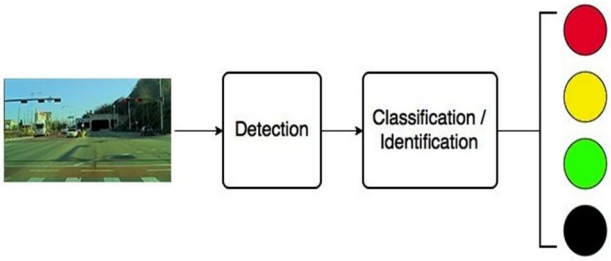
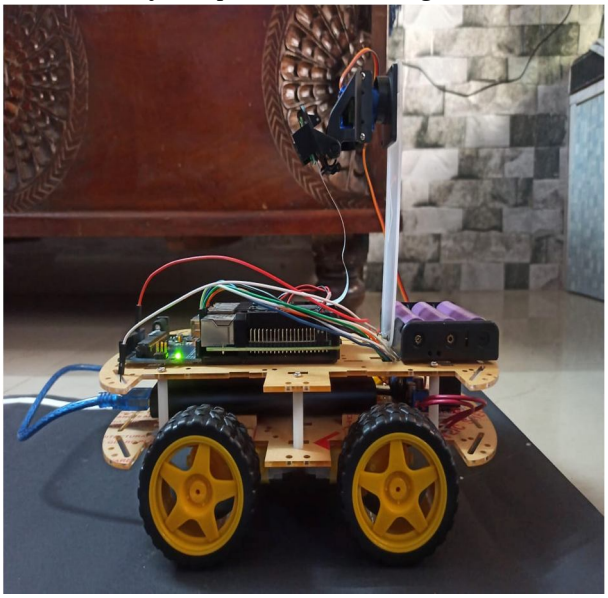



Figure 3.6: The flow of High Altitude Acute Response (HAAR) Cascade Classifier for traffic signals.

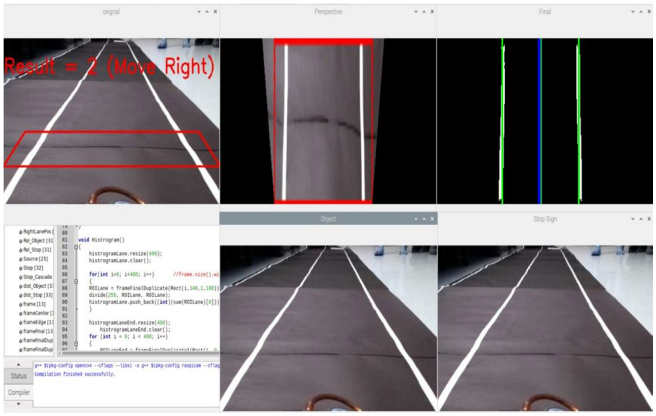
 

## IV. RESULT AND DISCUSSIONS

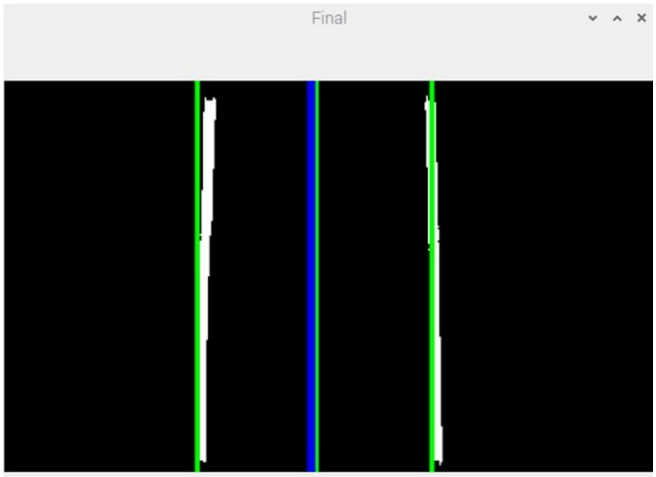
We have successfully created a vehicle that currently is capable of functioning in an autonomous manner.



Hardware Assembly




Lane Detection




Canny Edge Detection


*Stop Sign Detection*

## V. CONCLUSION

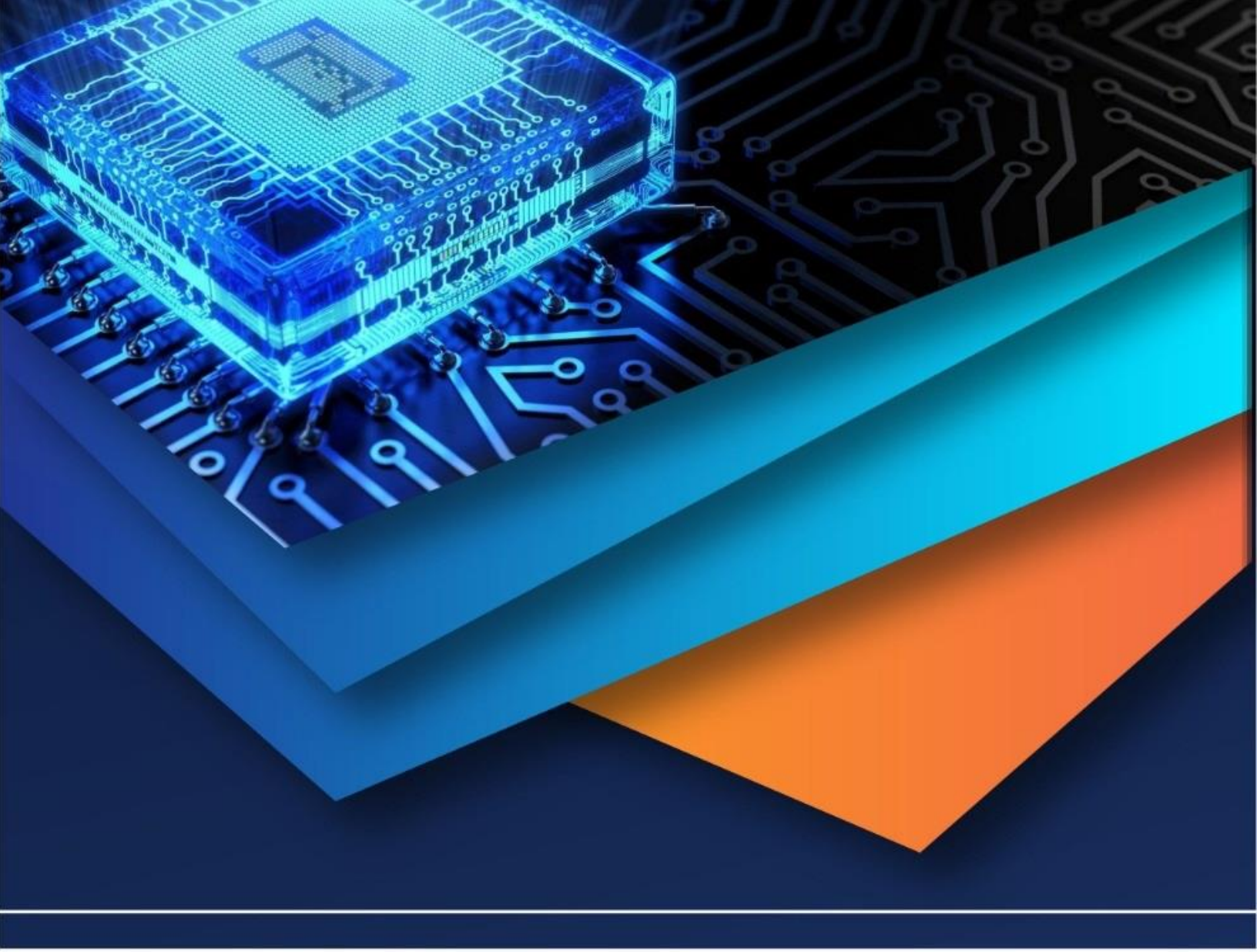
In conclusion, this self-driving car project has demonstrated promising results in detecting and avoiding obstacles on the road. By implementing haar cascade classifiers, we were able to identify and track objects such as pedestrians, vehicles, and traffic signs in real time. Combined with OpenCV's lane detection algorithms, we were able to achieve a more comprehensive understanding of the vehicle's surroundings, allowing it to make more informed decisions and navigate the environment safely. Furthermore, the implementation of PID control enabled the vehicle to regulate its speed.

## VI. FUTURE SCOPE

*1)* Self Parking System
*2)* Use of Data Analytics to Determine the Best Route.
*3)* Smart Anti Theft System.
*4)* Voice Automation.
*5)* GPS-Based Navigation

## REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587.
[2] Aditya Kumar Jain, "Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino", in Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018) IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1.
[3] U. Karni, S. S. Ramachandran, K. Sivaraman and A. K. Veeraraghavan, "Development Of Autonomous Downscaled Model Car Using Neural Networks And Machine Learning," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1089-1094.m
[4] D. Barnes, M. Gadd, P. Murcutt, P. Newman and I. Posner, "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 6433-6438,doi:10.1109/ICRA40945.2020.9196884.
[5] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.
[6] P. Sun et al., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 2443-2451, doi: 10.1109/CVPR42600.2020.00252..
[7] F. Yu et al., "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 2633-2642, doi: 10.1109/CVPR42600.2020.00271.
[8] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement", [Online]. Available:https://pjreddie.com/media/files/papers/YOLOv3.pdf [Accessed Sep. 20, 2020].
[9] Malay Shah, Prof. Rupal Kapdi, "Object Detection Using Deep Neural Networks", in International Conference on Intelligent Computing and Control Systems ICICCS 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY