iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.42074

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 10 Issue IV Apr 2022- Available at www.ijraset.com

Miniature Model of Autonomous Vehicle Using Arduino UNO and Open CV

Tejas Walke[1], Akshada Agnihotri[2], Reshma Gohate[3], Shweta Mane[4], Suraj Pande[5], Kalyani Pendke[6]
[1, 2, 3, 4, 5, 6]*Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Research, Nagpur, Maharashtra, India*

Abstract: ***The one thing that makes Tesla stand out from the crowd is its full automation Self-Driving feature. Not just the name itself but the technology behind it is even more interesting.***
It is not only about the luxury but also has a lot of advantages which backs the term. And what blows our mind is that not a lot of Indian car companies are focused on this technology on wide scale.
So, inspired by that this paper presents to build a Self-Driving Autonomous car model but on minimalistic basis which basically will be focused on three main features which are to operate in accordance with the surrounding depending on the direction of the road, to detect stop sign and halt for 5-10 seconds and detect traffic signs and make decisions accordingly. The miniature self-driving car will detect the two-lane path and perform the above functions.
Keywords: Autonomous Vehicles, Self-driving car, Computer Vision, Neural Network, Image processing, wireless sensor networks, control systems, path planning.

I. INTRODUCTION

A self-driving automobile (also known as an autonomous car or a driverless car) operates without human intervention and can perceive its surroundings.

To build a self-driving car, technologies from various disciplines are combined, which comprises technologies from Computer Science, Electronics, and Mechanical Engineering. It has a range of sensors that are related to electrical technologies, and it requires the usage of computer software to program those sensors. The mechanical development technologies underpin the entire automobile concept.

Approximately 1.3 million people die each year as a result of traffic accidents, the majority of which are caused by human error. In fact, according to a study conducted by the National Highway Traffic Safety Administration (NHTSA), drivers are responsible for 94 percent of all incidents. Self-driving cars are not only luxurious, but they can also result in fewer accidents due to the lack of human intervention.

It's difficult to accept that a car driven by computers may be safer at first, but consider this: how many car accidents have been caused by human mistakes, whether it's speeding, driving recklessly, inattentiveness, or, worse, drunk driving? It turns out that people are at blame for the vast majority of mishaps. According to predictions, by 2030 Self-driving or autonomous vehicles will be reliable, affordable, and will provide huge benefits and savings.

Self-driving cars, on the other hand, are entirely analytical, navigating with the help of cameras, radar, and other sensors. There are no distractions such as cell phones, and no impairing variables such as alcohol to impact driving performance. A smart car's computers react faster than human minds and aren't prone to the many potential blunders we might make on the road. As a result, a self-driving automobile future will be a safer one. [10] Machine learning has a probability of being more powerful for building an exhaustive system that helps in decision-making.

Machine learning involves many types such as supervised learning, unsupervised learning, deep learning, semi-supervised learning, and active and inductive learning. For the purpose of detection of objects, the classifiers of machine learning algorithms need to be trained on large amounts of data.

Autonomous systems require comprehensive testing, because the system is complex, and any decision made by the software affects human lives directly. The traditional validation and testing techniques are not feasible. Thus, an alternative approach is needed. Autonomous vehicles will fully sculpt all three levels - public communication, human-machine interaction, and technical feasibility for transportation.

©IJRASET: All Rights are Reserved

There are 5 different levels of Driving Automation. They are as follows in TABLE I:

TABLE I
Levels of Automations and it's Characteristics

Level	Defining Characteristics
Level 0 -- No automation	The driver is responsible for all core driving tasks. However, Level 0 vehicles may still include features like automatic emergency breaking, blind-spot warnings, and lane-departure warnings.
Level 1 -- Driver assistance Hands on/ share	Vehicle navigation is controlled by the driver, but driving-assist features like lane centring or adaptive cruise control are included.
Level 2 -- Partial automation Hands off	Core vehicle is still controlled by the driver, but the vehicle is capable of using assisted-driving features like lane centring and adaptive cruise control simultaneously.
Level	Defining Characteristics
Level 4 -- High automation Steering wheel optional	The vehicle can carry out all driving functions and does not require that the driver remain ready to take control of navigation. However, the quality of the ADS navigation may decline under certain conditions such as off-road driving or other types of abnormal or hazardous situations. The driver may have the option to control the vehicle.
Level 5 -- Full Automation Mind off	The ADS system is advanced enough that the vehicle can carry out all driving functions no matter the conditions. The driver may have the option to control the vehicle.

This paper proposes a working miniature prototype of a self-driving car using Raspberry pie, Arduino, and some open-source software and is based on Level 5 automation of driving. Raspberry Pi collects inputs from a camera module and an ultrasonic sensor. It then sends the data to the computer wirelessly. The Raspberry Pi processes input images and sensor data for object detection (stop signs and traffic lights) and collision avoidance respectively. A neural network model runs on a Raspberry Pi and makes predictions for steering based on input images. Prediction is then transferred to the Arduino for controlling the car.

II. RELATED WORK

In [3], The article explains how autonomous cars have evolved through time and how they have changed in each model. The history of self-driving cars dates back to the early 1500s when Leonardo da Vinci created a cart that did not require human assistance. It was mostly owing to the force generated by high tension, as well as a predetermined path for the steering. In the year 1925, an automobile was seen crossing streets in Manhattan without a driver. The car was radio-controlled and could execute operations such as shifting gears, honking the horn, and activating the engines. In 1958, General Motors developed a radio-controlled self-driving automobile model where the steering wheel was moved by current flowing through wire embedded in the road. In 1961, It was the first time when cameras were used in an autonomous vehicle to autonomously detect and follow the track and that too to be used on the moon. The model was called Stanford Cart created and prototyped by James Adam. From that time cameras were used to process images of roads. The first-ever self-driving passenger vehicle was tested in 1977 that could reach up to 20 miles per hour. In 1995 an autonomous car travelled 2,797 miles which were created by Carnegie Mellon researchers but here the car's speed and the braking were controlled by the user. Then, when we arrived in 2000, automation was in full swing, with various researches and challenges underway to automate the autonomous sectors such as the DARPA challenge and the US research arm. Many major automotive makers including Mercedes-Benz, BMW, and Volvo, produced Parma, Mercedes-Benz-s, and Volvo s60, which come in level 3 to level 4 automation levels. The closest was TESLA which understood the assignment and made a full self-driving hands-free package but it also has a driving feature.

In [4], The key parts or features of an autonomous vehicle are as follows-

Use of sensors and various image processing algorithms in the navigation system and path planning- The car should be able to automatically and intelligently determine which path to follow from the source to the destination, utilizing methods such as map mapping, GPS, and so on, and choose the best driving route using its intelligence. Environment perception - The car's control system must be able to perceive the surrounding environment to make the necessary decisions.

It includes radar navigation, as well as visual navigation employing various sensors such as laser sensors and radar sensors. The data collected from the sensors is used to create perceptions about the environment, such as barriers, stop signs, and so on.

Vehicle control — This section mostly includes controlling the vehicle's speed and direction. The perception module receives data such as environment perception, vehicle status, driving target, traffic regulations, and driving knowledge, and the vehicle control algorithm calculates the control target, which is subsequently sent to the vehicle control system. Finally, the vehicle control system puts those instructions into action to manage the vehicle's direction, speed, light, horn, etc.

In [5], The paper explains a prototype of a simulation model for a collision-avoidance system that can send alerts before a collision occurs and apply brakes to avoid the collision. The device can also show the distance between the car and oncoming vehicles in a visual depiction. In [8], The paper explains a prototype model of a robot that can detect the object and does actions that depend upon the object's movement, and maintains a constant distance between the object and the robot. The Linux operating system was used for the prototype presented in the paper. In [9], The deployment of Machine learning for a higher level of driving assistance can enable an automobile to acquire data from cameras and other sensors about its surroundings, understand it and decide what actions to take. Cameras for a perfect view of their surroundings, self-driving cars have a lot of cameras at every angle. While some cameras offer a 120-degree field of view, others have a restricted field of view for long-range vision.In [12], The model proposes a system for a self-driving car that can reach a predetermined destination using voice instructions or web-based controls. Any obstructions in its path are detected and avoided. The system offers a generic model that may be used on any device regardless of the size of the vehicle. Both the model has the same functioning and the factors/features is what make them different. Both models use one technique for lane detection and that is the Hough transform. In ([13], [14]) the paper proposes a molecular vision autonomous car using Raspberry pie as a chip for processing. The car is controlled using a remote-control interface which may be using the web interface or even a mobile interface.

III. DESIGN ARCHITECTURE

A. *Hardware/Software Requirements*

1) Robo Car Chassis
2) L298 H Bridge
3) Power Bank
4) Raspberry Pi 3 B+
5) Raspberry Pi Camera
6) Raspberry Pi Clear Case
7) Camera CSI Cable 12"
8) Arduino Uno
9) USB Cables for Power Bank
10) Micro USB Cable
11) 16GB SD Card
12) OpenCV
13) C++ Programming Language
14) Neural Network
15) Machine Learning and Artificial Intelligence

B. *Modules*

1) Hardware Assembly
2) Arduino Setup
3) Raspberry Pie Setup
4) Image Processing using C++ and OpenCV
5) Lane detection
6) Obstacle/Object detection System in Real-Time
7) Stop sign detection System using Neural Network
8) Traffic Light Detection System
9) Final Testing on Track

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

C. Proposed Methodology

1) *Assembly:* The DC motor is soldered using copper wires. Ref to fig.3 The two dc motors are placed parallel to each other. 4 dc motors are then mounted on the chassis using screws. The dc motors are placed facing each other. The power bank and H bridge are also mounted on the chassis. A USB cable connects the H bridge to the power bridge. Another USB is connected to make a connection and to provide a power supply to other components. Arduino Uno is placed above another plate of chassis. Another circuit (custom build pc) is placed below Arduino for connecting to Raspberry pie. To complete the build tires are connected to the chassis for the purpose to move. A set of sensors like R sensors and the L298 H-bridge motor driver are connected to the Raspberry Pi 3 controller through its General-Purpose Input Output (GPIO) Pins. The Raspberry Pi board's USB port is used to attach the web camera. Two dc motors spinning at 30 RPM drive the front wheels. After receiving control signals from the Raspberry Pi controller, an H-bridge driver circuit regulates the movement of these motors in either a clockwise or anticlockwise orientation.

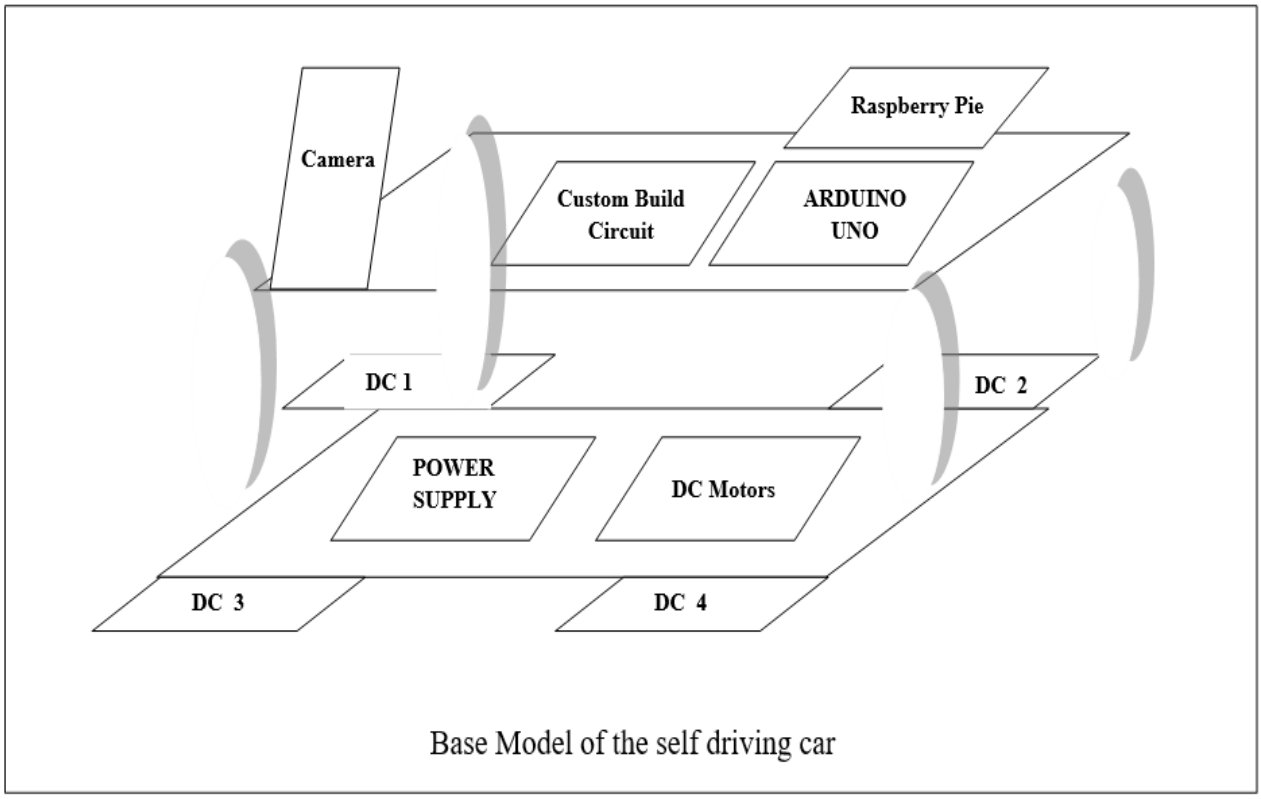


Fig. 1 Base Model of the vehicle

2) *Arduino Setup:* The most crucial and basic operations of a vehicle are forward and backward movements, as well as left and right movements. The programming of functions like forward and backward are coded in one file using Arduino ID. Separate files are used to code both operations. The output is encoded into Arduino which is placed on the base model to perform given actions.
3) *Raspberry Pie Setup:* The installation of Raspberry OS is required for a Raspberry Pie setup. The Raspberry Pi is connected to the computer through an Ethernet wire. The computer is then booted from the SD card. For image processing, OpenCV must be installed on the Raspberry Pi. Raspicam is used for image processing and the libraries for the same need to be linked with the Raspberry Pi. The camera is initialized and photos and movies are captured using C++ code.
4) *Image Processing using C++ and OpenCV:* OpenCV library is used to change the color space of the image/frames and to create a region of interest. A perspective wrap is to be created around the region of interest as in fig.2 to get a bird's eye view of the region of interest. To find lanes from tracks the perspective wrap is to be fed to Raspberry pie for image processing. The yellow line shown in the below diagram explains the region of interest for the following model. To enhance the image, basic image processing is done.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

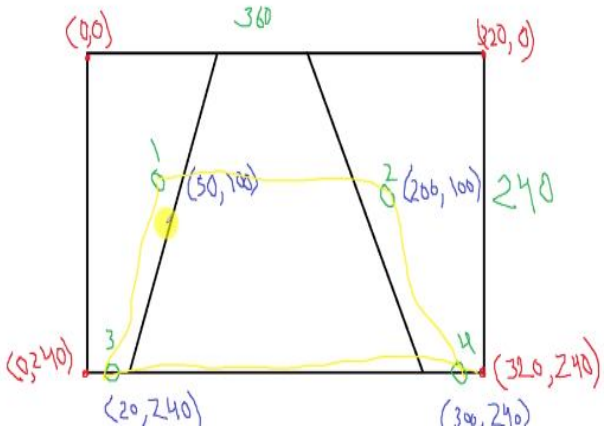


Fig. 2 Region of Interest for proposed model

5) *Lane Detection and Following System:* To achieve the fundamental goal of lane detection and tracking, a self-driving automobile must be able to identify, track, and distinguish multiple routes for optimal road movement. Web-cam mounted on the self-driving car is connected to the Raspberry Pi controller to detect the position of the car relative to the White line marked at the border of the road. In the proposed self-driving car, a lane on the road is designed with a White line drawn at the borders of the road and the middle of the road. When we switch on the power supply all components start working and the webcam detects the white line with the help of image processing and the car starts its motion. Whenever a car gets to the right-hand border of the road web cam detects that with Edge Detection and by taking a slight right and starting its motion between the border white lines. Ref fig. 3.

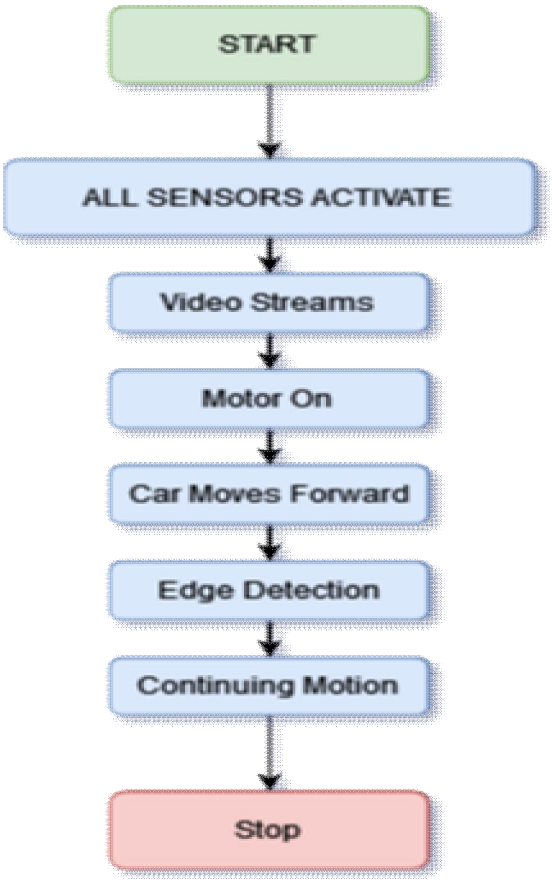


Fig. 3 Lane Detection and Following System

6) *Obstacle/Object detection system in Real Time:* Many applications, such as driverless vehicles, security, surveillance, and industrial applications, rely on object detection. Depending on the nature of the problem to be solved, picking the correct object detection method is crucial. The Single Shot Detector (SSD) is a suitable option because it can run on video and achieves a fair balance of speed and accuracy. The Real-Time Object Detection System (RTODS), which incorporates all sensors and the camera, is activated when the power is turned on. The Raspberry Pi controller receives captured images from the webcam as input.

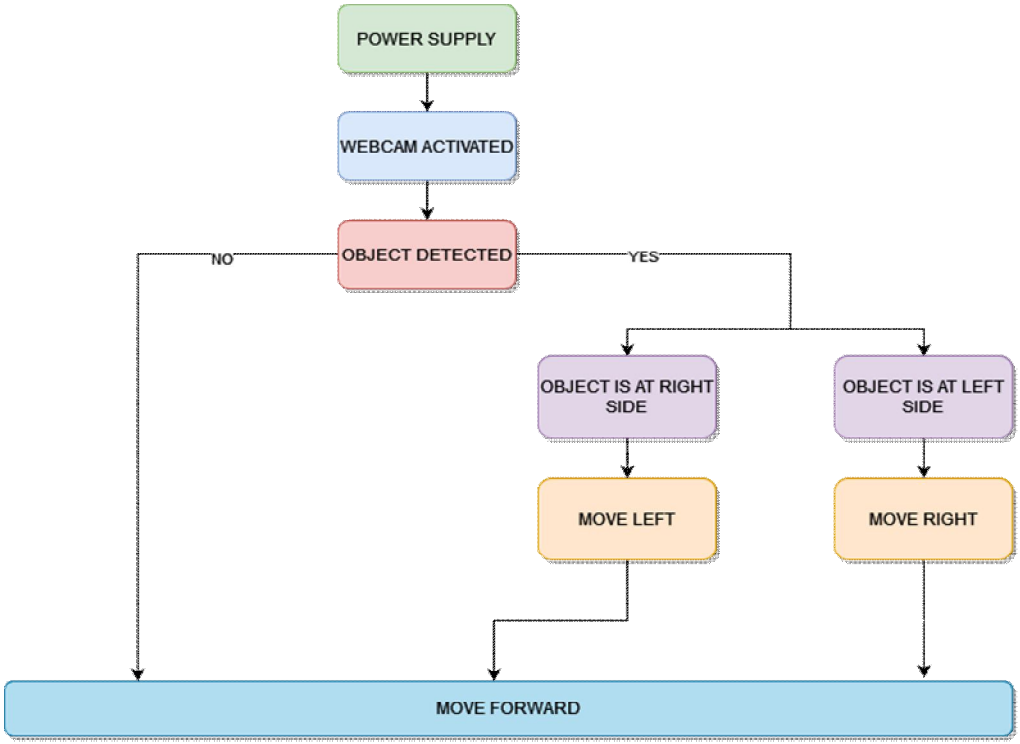


Fig. 4 Object Detection Algorithm Flowchart

The Raspberry Pi controller uses the received image to conduct a real-time object detection algorithm and delivers a control signal to the H-bridge, which operates the motor. If any animal, human, or object is spotted, the automobile will stop for 10 seconds and check for the presence of objects; if nothing is detected, the car will continue forward. Blob, a pre-processed image serves as input to feed the image into the network. Cropping, scaling, and color switching are among the pre-processing processes used. At various sizes, feature maps represent the image's most prominent features. As a result, using multi-Box on several feature maps increases the size of any object you want to detect. To guarantee that the network understands what constitutes an erroneous detection, a 3:1 ratio of negative to positive predictions is employed during training instead of all negative predictions. The non-maximum suppression technique is applied to all bounding boxes, and the top predictions are maintained. This ensures that the network's most likely event predictions are kept. Ref Fig. 4.

7) *Stop sign detection system using Neural Network:* By turning on the system, including all of the sensors and the camera, the Raspberry Pi controller receives captured images from the webcam as input. The Raspberry Pi controller runs a real-time object detection algorithm on the received image. It then sends the signal to H-bridge which drives the motor. If any animal, human, or item is spotted, the automobile will stop for 10 seconds and check for the presence of objects; if nothing is detected, the car will continue forward. Ref Fig. 5. Various images of the stop signal are fed in the training and the testing model and also by using raspberry pi and Arduino. The input to the car is taken as a stream of images by the RaspiCam. The necessary camera libraries are installed and built into the OS. Frames are calculated per second using algorithms. Using the proper routines, the final photos are transformed from BGR to RGB. Then, where the real detection is necessary, we constructed a Region of Interest (ROI). For proper image analysis, this ROI is subjected to perspective transformation. The grayscale image acquired by the Raspberry Pi Cam2 is then converted to a black and white image using image thresholding.

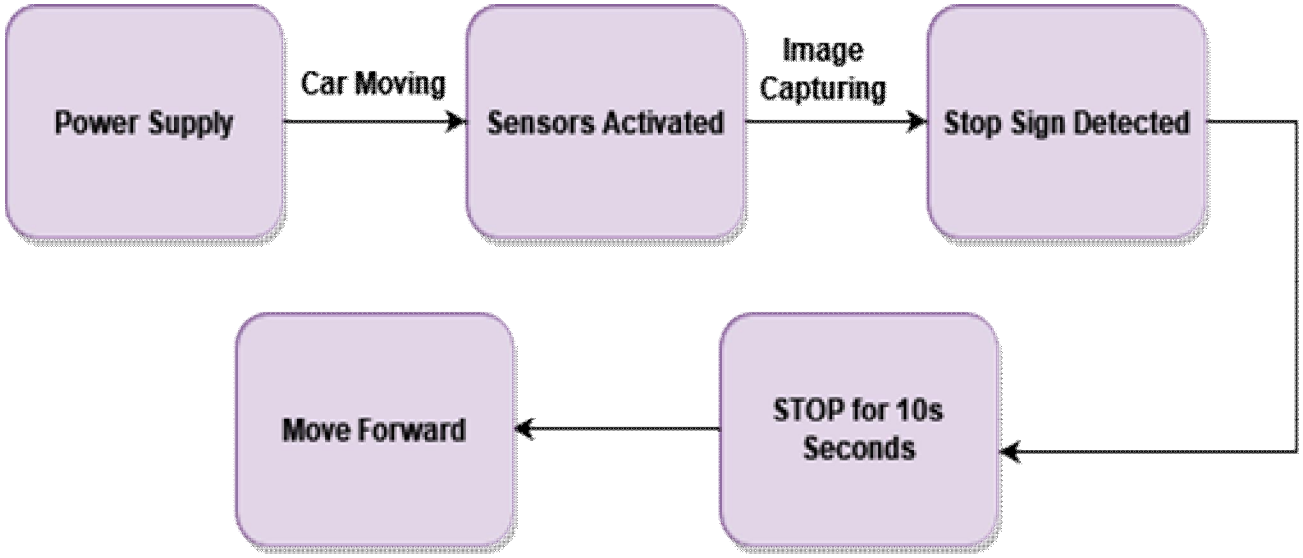


Fig. 5 Stop Sign Detection Algorithm

8) *Traffic Light Detection System:* To detect and track the red, yellow, and green colors of the traffic light system, fundamentals of computer vision are used. The flow chart of the Traffic Light Detection System (TLDS) algorithm, shown in Fig. 6, illustrates how self-driving car recognizes and responds to traffic light. The input frames obtained from the camera are in BGR format which is converted from BGR color space to corresponding Hue Saturation Value (HSV). In OpenCV, the range of values for Hue (representing color) is 0 - 179, the range of values for Saturation (representing intensity/purity of color) is 0 - 255, range of numbers for value (representing brightness of color) is 0 - 255. The color range to be monitored is defined according to the requirements, and then morphological changes in the color are applied to reduce noise from the binary image. Then, the contouring of the colors is done with a rectangular bounded line called 'contour' to differentiate between each color.

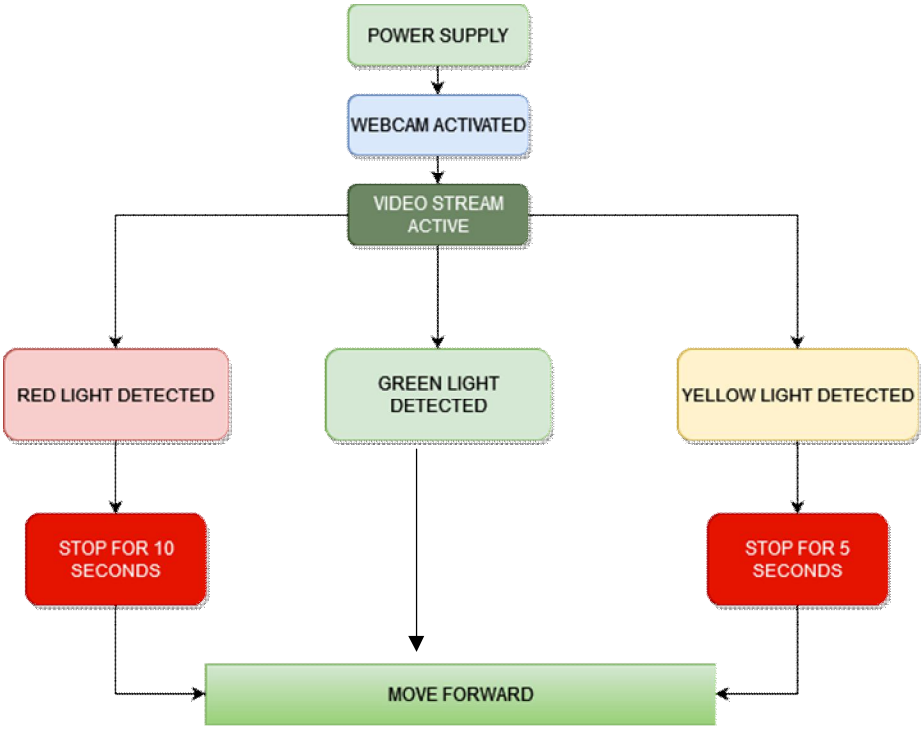


Fig. 6 Traffic Light Detection Algorithm

IV. APPLICATIONS

Automated Vehicle Path Planning - The vehicle can plan a path with any shape that the vehicle's model can confirm. This is therefore called Automated Path Planning. For successful autonomous driving, automated vehicles must navigate a variety of path alterations. If an obstacle is moving, it could be stationary or on the verge of colliding. To avoid the barrier, the automated vehicle must analyze the vehicle model as well as the environment map to determine the shortest path back to the original path with the least amount of clearance to surrounding objects. Vehicle automation is a much bigger market than commercial vehicle automation. Freelance Robotics' work on a range of vehicles has emphasized automation. Civil engineering is another area of application, where successful robots for pipe inspection have been developed. These applications are just a few instances of the automated industry in general. Automated vehicles are a clear development market due to the potential variety and usability of applications.

GPS Acquisition and Processing - When operating in an open environment, automated vehicles frequently employ GPS technology to determine their exact location. To go around, the vehicle will have speed and bearing commands.

V. RESULT

Below are the actual images of the proposed model. The components described in Fig. 1 i.e., Base Model of Vehicle can be clearly seen in the images below which are mounted on a chassis.

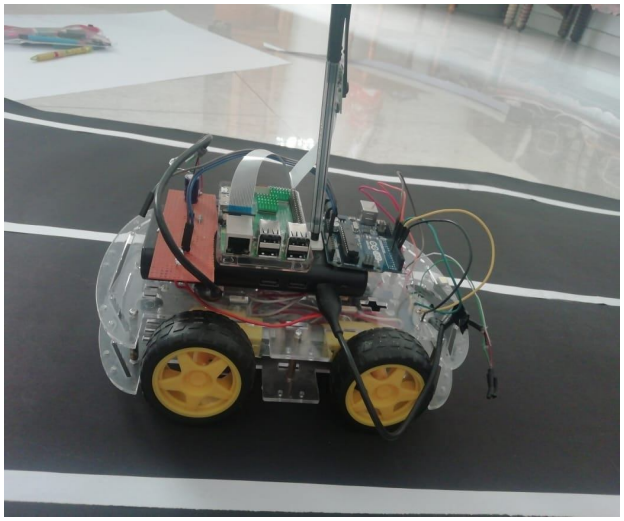


Fig. 7 Side View of the Proposed Vehicle

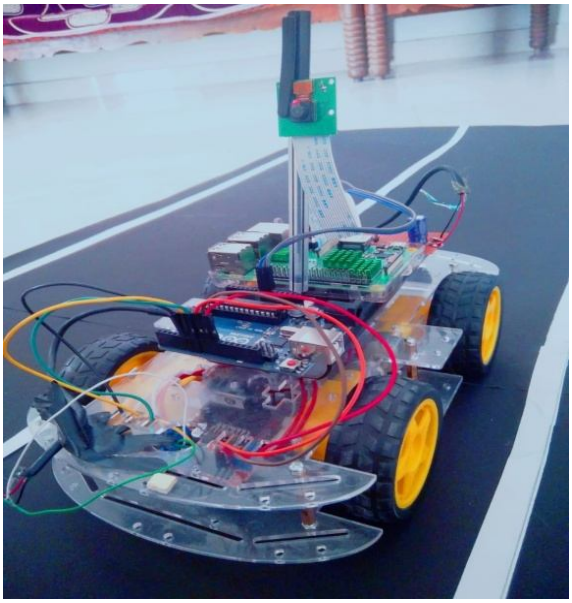


Fig. 8 Front View of the Proposed Vehicle

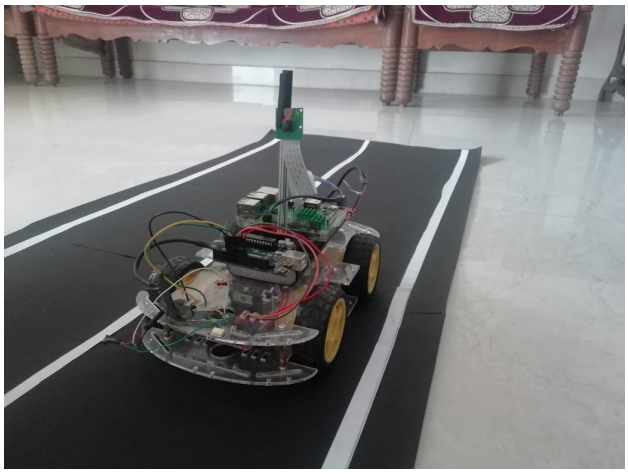


Fig. 9 Image of the Proposed Vehicle Running on Track

The Camera sends images for image processing and then it's processed through Open CV and by understanding the surrounding the control instructions are then sent to Arduino to further control the car and follow the path that's safe such as after processing the images shown below it has been detected that it has to move straight as there is a straight lane, these images are sent to master device i.e. (Raspberry pie for processing) which decides and sends instructions to slave device i.e.(Arduino) which controls the wheels. The lane detected is straight that's why the car is moving forward and not in the left or right direction. The Road is being detected by the camera mounted on the cars. The Region of Interest is clearly visible in the below screenshots for the proposed model.

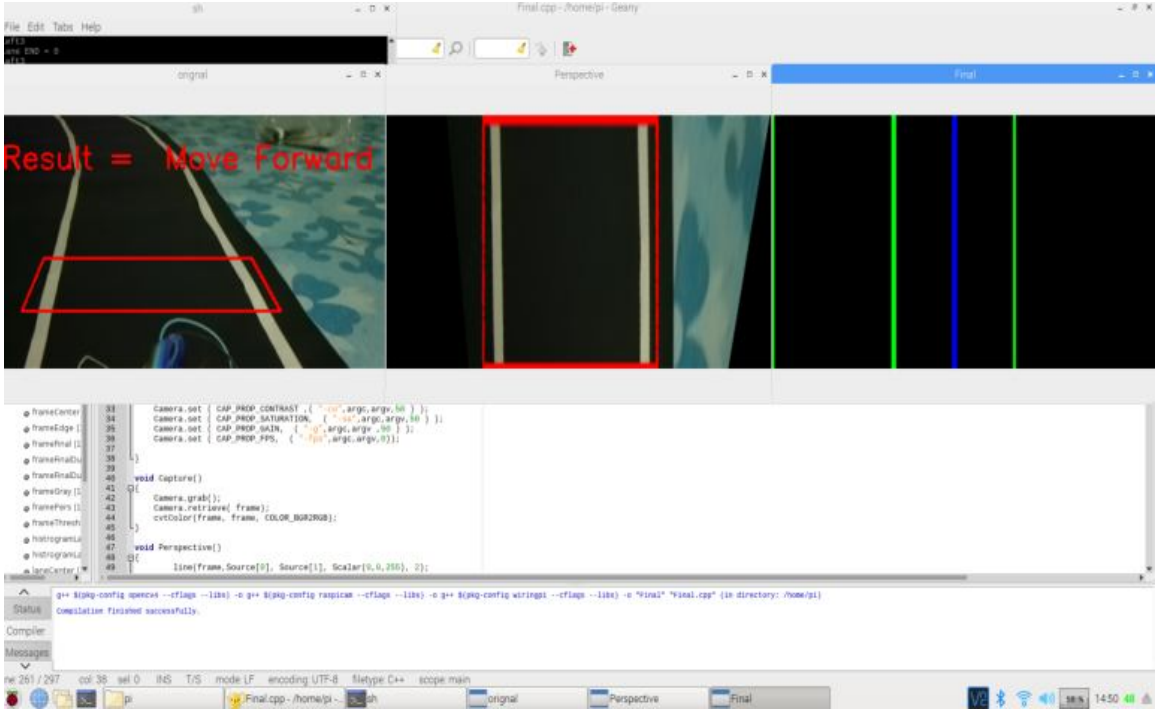


Fig. 10 Car following a Straight Lane Path and moving Forward

The Camera sends images for image processing and then it's processed through Open CV and by understanding the surrounding the control instructions are then sent to Arduino to further control the car and follow the path that's safe such as after processing the images shown in Fig 11 and Fig 12, the Guidelines of the road are being detected and sent to OpenCV and it has been processed and found that they are tilting and thus instructions to turn as per the guidelines are then sent to the Arduino UNO that it has to move slightly right for taking a turn towards the right direction. This can be very well seen in Fig 11 and Fig 12.

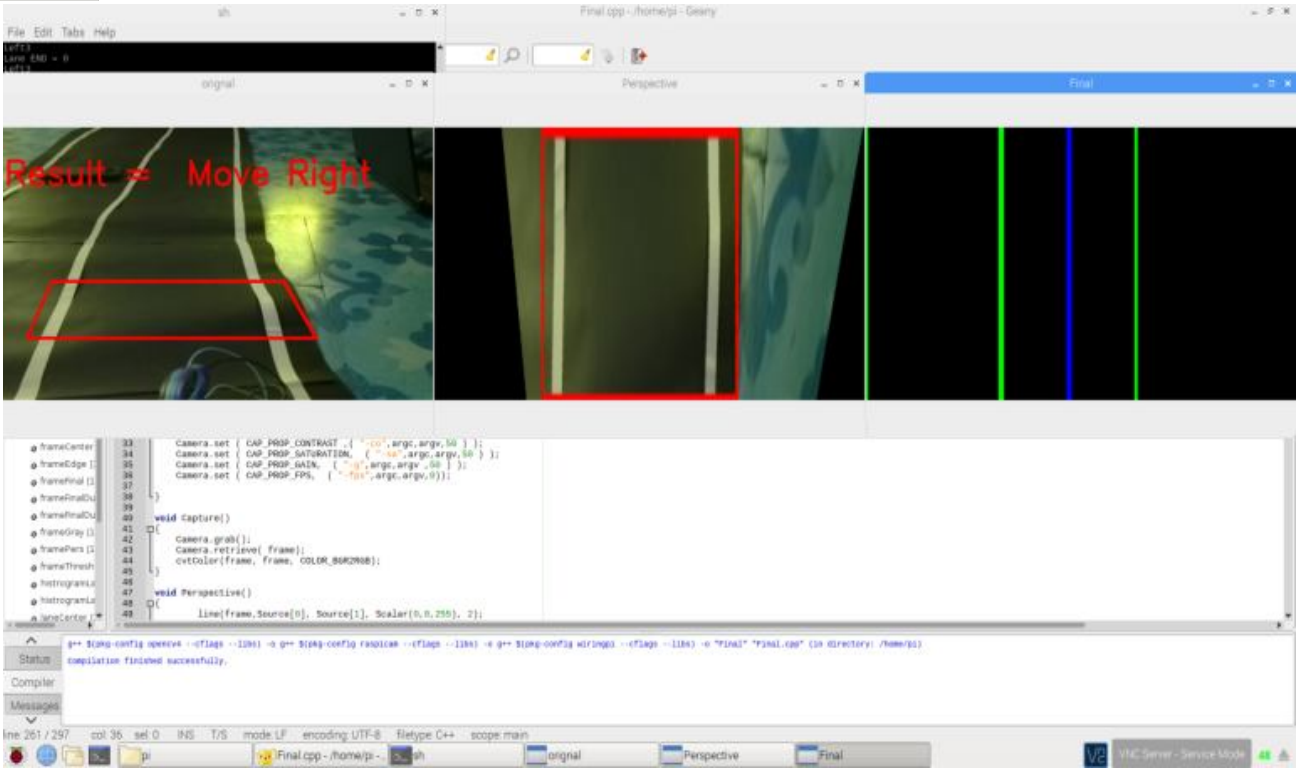


Fig. 11 Change of Direction according to Guidelines (Right)

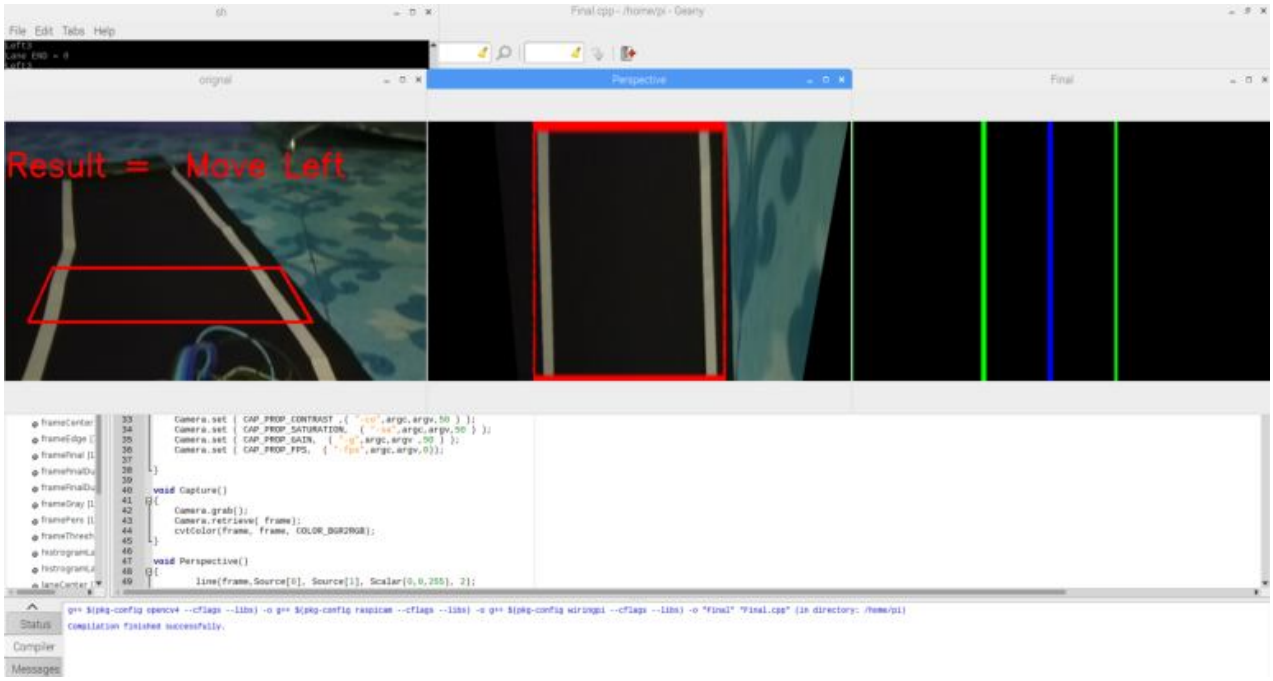


Fig. 12 Change of Direction according to Guidelines (Left)

The Camera sends Stop Sign Images for Image Processing and then its processed through Open CV and after understanding the surrounding, instructions are sent to control the car, such as in the Fig 13 where the camera has identified the Stop Sign. Instruction to stop are then sent to slave device i.e. (Arduino UNO) which controls the wheels, then the car will stop for specific period of time and then again will start moving forward. This can be very well seen in Fig 13.

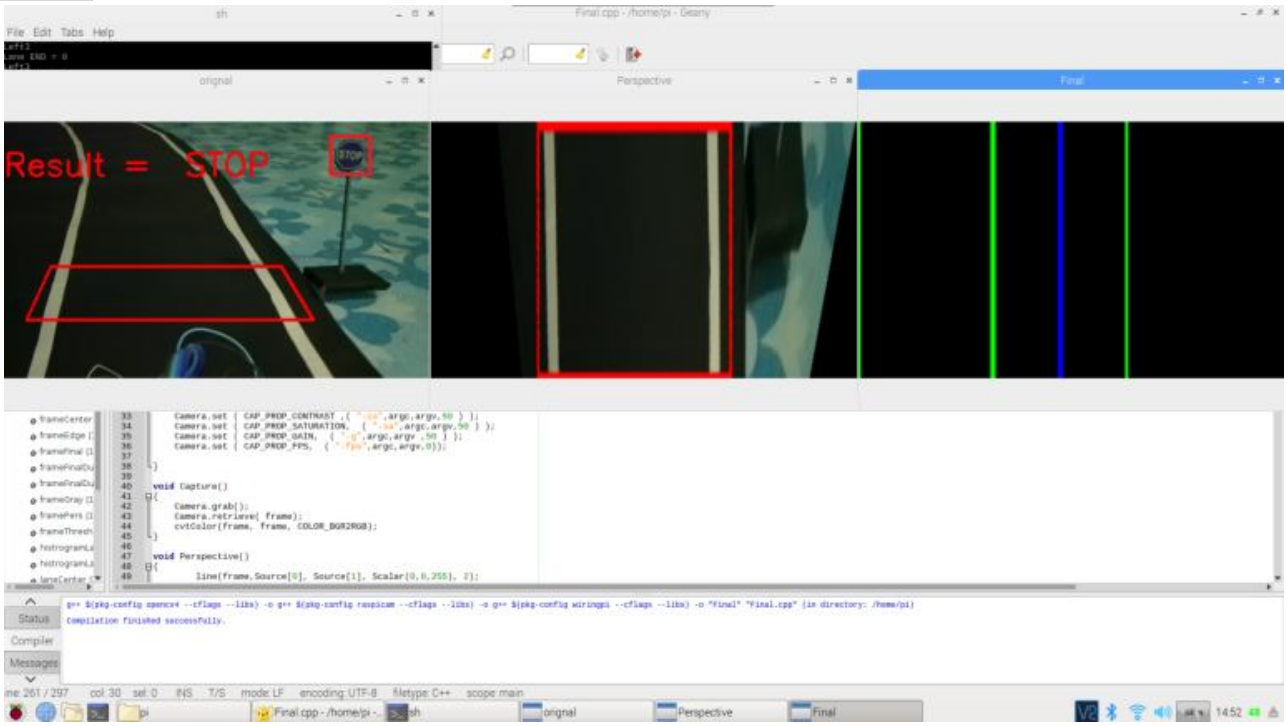


Fig 13 STOP Sign Detection

VI. FUTURE SCOPE

This technology can be further used in different fields right from small household appliances to big industries. A lot of mishaps happen when transporting via sea but it eventually results in human accidents. The technology can be utilized in sailboats and cargo ships to move items without the need for human participation, although certain automation and features additionally are required. Why the hassle of the movie the shopping trolley especially when it contains loads of items. This technology can be used so that the trolley can follow the user by itself. The farmers work in sun and the rain but cannot just stop because they need to feed themselves. This technology can be used while ploughing the land. The plough can be attached to the vehicle with the proposed technology and environmental planning is necessary.

VII. CONCLUSION

Self-driving car results in comparatively fewer accidents because of no human input. Autonomous vehicles may see a boost in the upcoming years. Our roads will be safer. We will be more productive as the time will be saved. They'll cut down on any accident-induced costs, can be fuel-efficient, and eventually, money can be saved. It's feasible that in the future, we'll see a completely efficient highway with only autonomous intersections, where the automobile never has to stop until it arrives at its destination. The favourable environmental impact could be even higher. Embracing the future of self-driving automobiles is worth it even for the environmental benefits.

REFERENCES

[1] Todd Litman, Autonomous Vehicle Implementation Predictions Implications for Transport Planning, Victoria Transport Policy Institute
[2] What Are the Levels of Automated Driving https://www.aptiv.com/en/insights/article/what-are-the-levels-of-automated-driving?
[3] History of Autonomous Cars https://www.tomorrowsworldtoday.com/2021/08/09/history-of-autonomous-cars/
[4] Jianfeng Zhao, Bodong Liang and Qiuxia Chen, "The key technology toward the self-driving car", International Journal of Intelligent UnmannedSystems, Vol. 6, No. 1, 2018 pp. 2-20, DOI 10.1108/IJIUS-08-2017-0008
[5] Manas Metar, Harihar Attal, "Designing a Vehicle Collison-Avoidance Safety System using Arduino", International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 9 Issue XII Dec 2021
[6] Margarita Martínez-Díaza, Francesc Soriguerab, "Autonomous vehicles: theoretical and practical challenges", ScienceDirect Transportation Research Procedia 33 (2018) 275–282, 10.1016/j.trpro.2018.10.103
[7] P. A. Hancocka, Illah Nourbakhshc, and Jack Stewartd, "On the future of transportation in an era of automated and autonomous vehicles", 7684–7691, PNAS , April 16, 2019 , vol. 116 , no. 16

[8] Lokesh M. Giripunje , Manjot Singh, Surabhi Wandhare, Ankita Yallawar, Object Tracking Robot by Using Raspberry PI with open Computer Vision (CV), International Journal of Trend in Research and Development, Volume 3(3), ISSN: 2394-9333 www.ijtrd.com IJTRD | May - Jun 2016 Available Online@www.ijtrd.com 31

[9] How Machine Learning in Automotive Makes Self-Driving Cars a Reality https://mindy-support.com/news-post/how-machine-learning-in-automotive-makes-self-driving-cars-a-reality/

[10] The Role of Machine Learning in Autonomous Vehicles https://www.electronicdesign.com/markets/automotive/article/21147200/nxp-semiconductors-the-role-of-machine-learning-in-autonomous-vehicles

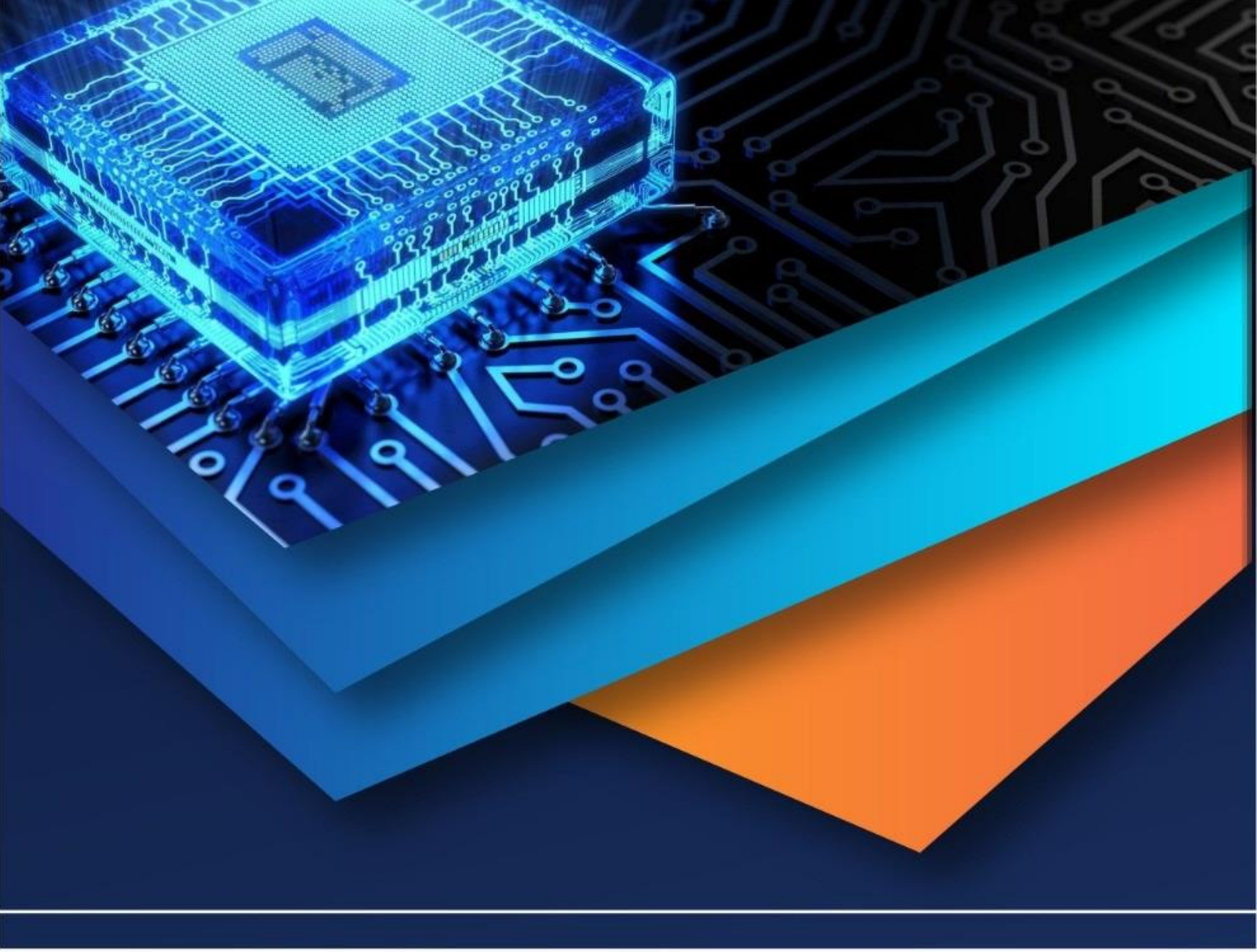
[11] Sehajbir Singh and Baljit Singh Saini, Autonomous cars: Recent developments, challenges, and possible solutions, 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012028

[12] Raj Shirolkar, Anushka Dhongade, Rohan Datar, Gayatri Behere, SelfDriving Autonomous Car using Raspberry Pi, India International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181 IJERTV8IS050100 Vol. 8 Issue 05, May-2019

[13] Gurjashan Singh Pannu, Mohammad Ansari and Pritha Gupta, Designand Implementation of Autonomous Car using Raspberry Pi, International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 9, March 2015

[14] Mr. Nihal A Shetty, Mr. Mohan k , Mr. Kaushik k, Autonomous Self-Driving Car using Raspberry Pi Model, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org R TESIT - 2019 Conference Proceedings , Volume 7, Issue 08

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)