Autonomous Car using Computer Vision with Raspberry Pi

Aditya Date[1], Amey Parab[1], Ishita Kheria[2], Jay Doshi[2], Jay Jani[2], Jazib Dawre[2], Amit A. Deshmukh[3]
[1] U. G. Student, Department of Mechanical Engineering, D.J. Sanghvi College of Engineering, Vile Parle (W), Mumbai- 400056
[2] U. G. Student, Department of Computer Engineering, D.J. Sanghvi College of Engineering, Vile Parle (W), Mumbai- 400056
[3] Professor & Head, Dept. of Electronics and Telecommunication, D. J. Sanghvi College of Engineering, Vile Parle (W), Mumbai- 400056
E-mail: [1]adityadate0@gmail.com, [1]ameyparab45@gmail.com, [2]Kheriaishita@gmail.com, [2]jjjayjani@gmail.com, [2]Jayutrdoshi@gmail.com, [2]jazib980@gmail.com, [3]amit.deshmukh@djsce.ac.in

***Abstract*—The automobile industry has been reshaping itself since decades with new innovations taking birth every now and then. However, with the industry's focus on making the upcoming technologies smarter, the type of advancements made in the automobile sector have evolved drastically. Along with the attempt to get the maximum output from cars, we equally research on making them smarter thus evolving them into artificially intelligent machines. In order to maximize safety and homogenous traffic control, both the public and industry's interest in autonomous cars has increased dramatically. The aim of this project was to equip the team with hands-on experience in building an autonomous car born with self-driving capabilities. Most importantly, learning and understanding the workflow along with handling the challenges on the way to creating a fully functional autonomous car were the major takeaways from this project. This paper proposes a prototype built on such futuristic concept. Such concepts are being planned to be used in various other sectors too but the paper limits the project scope to commercial automobiles only. Although the paper revolves around a simple model, it deals with a prototype which tries to mimic the working of an actual autonomous car on the streets and its responses to constantly changing conditions in the environment the car runs in dealing with road tracking, traffic sign detection and collision avoidance.**

Keywords—Automation, Artificial Intelligence, Raspberry Pi, Arduino, Path following, Obstacle detection, Python, App development, Infrared vision.

I. INTRODUCTION

Human perception and Computer Vision – two mirroring approaches of the fundamental process of seeing. While one could argue that Computer Vision is nothing but a crude replication of eyesight with significant constraints, another argument can be made that human eyesight in itself suffers from innumerable constraints which leads us to a belief that computer vision is the future. Computer Vision is the overall identification of information from images or moving images which is nothing but the changing environment around a computer. A camera recording the walkthrough on a street captures street lights, pedestrians, other cars and so on. These objects are perceived by the machine from the frames of the video. The information could be anything related to 3D objects, model detection, compilation and processing of information and lot more. It is the process of employing an image sensor to read images and then using a computer processor to evaluate these images to decode the necessary information.

When the word artificial comes up, it always means it has an edge over the part but a biological one. Comparison between a camera and a human eye forms a quick example. A camera is nothing but the eyes of the computer but better in every way. Thanks to ever advancing optical technology, making stuff visible in a dark room has become possible, which is obviously something a human eye cannot. Moving through a foreign dark environment becomes a tough job when we can't rely on our eyes. Object Recognition involves detecting various image data, understanding it using numerous different algorithms and displaying it using the dedicated hardware. This process uses the previously stored data of image datasets and after comparing with the target image, makes a decision accordingly. A machine trained on a car dataset will decide the object it sees is a car if it understands four wheels and a similar chassis shape. Object recognition as a task is used in a variety of operations where identifying various objects and their tracking is dealt with. To detect and recognize an object correctly, we need to first find out if an object is present. After knowing this, we need to examine the location of the object. The correct location of the object is immensely important so as to avoid any errors. Finding the number of objects in a given camera frame is important and distinguishing them clearly helps our cause. Lastly its size, classification and real time tracking helps us prepare the final output. Object detection systems usually work on sublime processes and understanding methodologies to plot points of an object set. It is widely applied in image recovery, defence and monitoring.

Any object can have different shape and size and its identification becomes a tedious and sometimes complicated process depending on the number and density. Different lighting and distance from the locator determine the kind of tools be used to define the object outlines. There are various techniques used to detect objects based on different input parameters sometimes called hyperparameters in machine learning. Apart from the appearance-based methods, there are some feature-based processes like interpretation trees, SIFT, SURF and pose clustering which are very widely used. The detection of objects in real time requires a fast optimization of the datasets and total control. Naturally for quick and lag free object detection, equally powerful hardware becomes a necessity. The recognition of 3-D objects requires the proper detection of edges and curves. For creating such a system there are two major sub-systems which are instance level and category level. The instance level recognition includes the feature process and model fitting while category level recognition uses the sliding window method. Segmentation process makes it easier to access image regions so that working on them becomes simpler. After segmentation, a learned classifier can be used which helps in further progress. In most of the cases the object is taken as a multiple sample set that has some usual objective for the user. The samples should have segments of distinguishable objects or portray different scenarios of interest shown in the examined scenes. The objective of image mapping is sorting of samples which are a part of specific classes.

In detection systems, the images that are a part of the object being examined are processed by some analytical procedure. For example, in motion detection, the objects are usually analysed using some image mapping process: successive frames of a video are processed and the average image evaluation is then used for the required examination. In the object recognition process, it is required that the application of the dedicated hardware will lessen the total run-time needed to understand the structure of objects and give better evaluation to the virtual environment. Hence the need of a powerful hardware is desired when the computing time is to be kept as low as possible. A deep learning model for object detection is done in the following steps – feeding the training data, training the machine on the datasets and prediction of the final output.

II. OBJECTIVE

The aim of this paper was to understand and apply the concepts discussed in the beginning to build a prototype which detects objects in different scenarios, decides and controls the movement automatically, all powered by computer vision to capture an image, analyse it using some pre-defined logic and give a relatable output using dedicated hardware. The specific objectives would be to analyse and evaluate how to program the Arduino when the car is operated manually, to constraint the sensors to effectively control the vehicle in case of obstacles sensed on the path and programming the Raspberry-PI module using python, to interface the module with its camera to take images and to create an image processing technique to recognize images correctly and also enable the vehicle to be controlled over the voice. For example, when a red light on the signal is detected, the vehicle should be able to stop at the pre-programmed distance from the light or detect the varying direction of the road ahead and give intermediate inputs to the steering. After recognizing the object, the conversion is done using a specific methodology. Matching the object with the dataset results to ensure the vehicle is able to understand it that comes into the way is the most important task. In short, the main objective was to fabricate a mini prototype having the potential to demonstrate the actual behaviour of self-driving cars which might be running on the streets in a few years. Also, the proper integration between hardware and software to carry out smooth functioning of the model was one of the most important tasks in the project timeline.

III. BACKGROUND OF HARDWARE

A. Raspberry Pi 4 Model B

Raspberry Pi 4 is a micro, credit card computer but powerful enough to process and compute complex tasks owing to its powerful Cortex processor and 4 Gigabytes of Ram. It has also the capability to connect up to two 4K displays and supports multiple hardware inputs from a mouse to ethernet ports and shipped with an on-board 802.11n Wireless LAN adapter for fast and simple internet connectivity. This is the brain of the car as it handles everything from interpreting the camera inputs to controlling the motors and quick integration between the drive modes.

Fig.1 *Raspberry 4 Model B*

B. Arduino Mega 2560 Microcontroller

Arduino is an open-source electronics platform based on a microcontroller which can be programmed in C++ language for interfacing wide range of simple to complex electronic like LEDS, light sensors, potentiometers and so on. Arduino boards are cheap and easy to use ranging from nano to mega variants depending on the number of output and input pins on it. It also supports Mac and Linux and has its own software IDE for simple operation. It has been used extensively in projects involving artificial intelligence, remote operation and in many more such projects. It can also be customized using various connectivity adapters called shields to extend the capabilities of the board. For example, a Wi-fi shield can be used to control motors or LEDs using a wireless network.

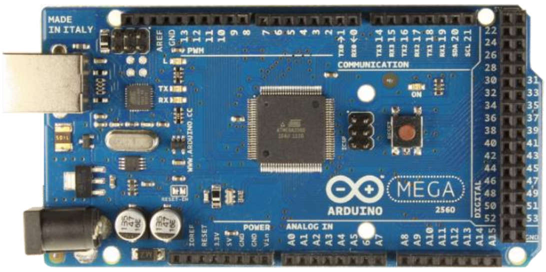

Fig.2 *Arduino Mega microcontroller*

C. Ultrasonic Sensors

Ultrasonic sensors are employed in distance measurements by transmitting ultrasonic sound waves and converting the reflected waves into an electrical signal captured by the receiver (Echo). Distance is calculated by the wave speed and the obstacle distance. It has been used in the car for determining the distance to the obstacle and hence for steering the car accordingly avoiding them.

Fig.3 *Ultrasonic Sensor*

D. Infrared Sensors

Infrared (IR) sensors transmit infrared light radiation and based on its reflection, the type of surface or its colour can be judged accordingly. This principle was applied for bringing about the line follow mode by detecting white path line as the radiation reflects from the white surface compared to no reflection from a black path. The car can hence treat a white strip as the road path and change the vehicle direction when the sensors detect a black part to bring the car back on track.

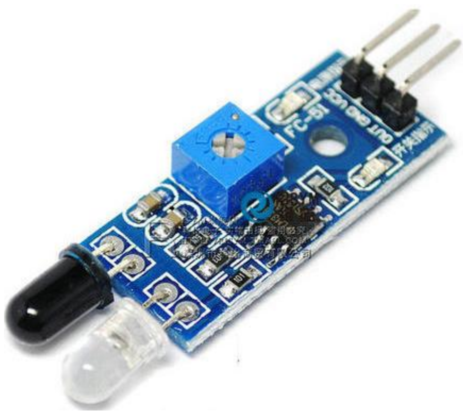

Fig.4 *IR Sensor*

E. L298 Motor Driver

The Motor Driver is a motor module which enables us to successfully control the working speed and direction of two motors simultaneously. L298 Motor Driver has been designed and developed based on the L293D IC. The L293D is the 16 pin Motor Driver IC. This module was designed to generate bidirectional drive currents at voltages ranging from 5 V to 36 V.

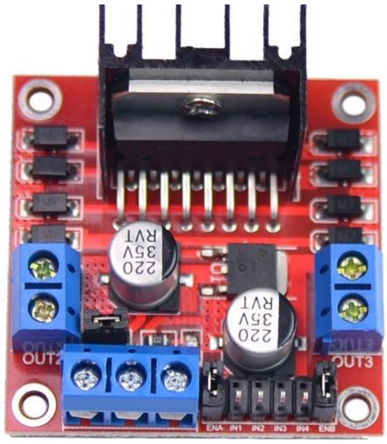

Fig.5 L298 Motor Driver

F. Raspberry Pi 5MP Camera Module

The camera module are the eyes of the car detecting various kinds of objects like an actual self-driving car on the street. The module comes with a flexible 15 pin Serial interface cable which is easy to attach to the raspberry pi board which eliminates the requirement of any adapters reducing costs and making it easier to operate. It can capture videos at 1080p, 30 fps and also at 720p, 60 and 30 fps.

Fig.6 Raspberry Camera Module

IV. PROPOSED DESIGN

For bringing computer vision to life, we have used the Open-CV which is an open-source platform for the overall implementation of the project. It is a powerful framework which has many algorithms and techniques for identifying and classifying images. It helps in pre-processing data and is much faster than most of the other libraries with a very high operational speed with minimum glitches. With in-built trained models, it reduces the setup time for training models for comparison. As Raspberry-PI is based on Linux kernel, it can be conveniently programmed in python to make the recognition task easier. It is widely used for real time applications which have a high accuracy requirement. The convolutional neural network helps in defining surfaces and boundaries for the correct allocation of objects and thus their identification, which has been used to follow the road surface the vehicle drives on. The ImageAI python library helps us integrate various functions for user-specific applications.

The proposed prototype is built on 7 different operating modes which include:

- Manual Mode (Radio Controlled Operation)
- Autonomous Mode (Powered by Computer Vision)
- Line Follower Mode (Using Infrared Sensors)
- Voice Controlled Mode (Over an Android Application)
- Obstacle Avoidance Mode (Using Ultrasonic Sensors)
- Internet Controlled Mode (Over a website using onscreen GUI buttons)
- Virtual Joystick Operation (Over an onscreen joystick)

We have used the Raspberry-PI 4 Model B that acts as a portable computer which is plugged in a monitor device and makes use of a regular mouse and keyboard. It allows the user to pre-define the code in the python programming language and provides other extensive functionality which is of utmost importance for object recognition. The module which we have used has a 1.5GHz 64-quad-core processor with a dual band

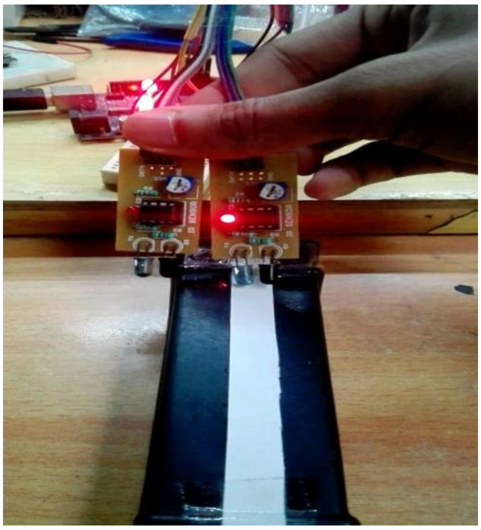

Fig.7 *Sensor detecting a white strip on road*

wireless functionality, fast Ethernet and has the capacity to connect up to 2 displays with 4K(HDMI). We have used a 16 GB micro-SD card and New Out-Of-Box Software (NOOBS) operating system for its connection establishment. It is equipped with 2GB Ram and the charging of the module can be done by a 2.5A USB cable and 2000 MAh battery backup can ensure uninterrupted working for at least 6-8 hours.

We have also made use of the Raspberry-PI interfacing camera. This camera module has to be carefully attached to the Raspberry-PI module so that we can capture various images on it. This camera is installed with the module which would give the vehicle the ability to see things around it.

We have used the Raspberry-PI Camera which is a 5 megapixel which can capture 2592 x 1944 static image pixels and supports the 1080p30 video formats. It is connected to the Camera Serial Interface (CSI) of the module using a 15-pin ribbon cable.

For the manual mode, obstacle avoiding and voice-controlled mode, we have employed Arduino Uno board to which the

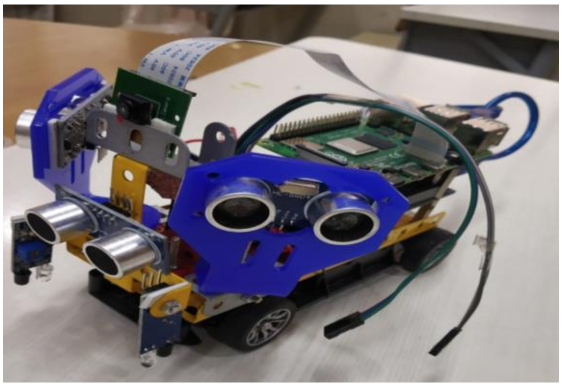

Fig.8 *Camera module on top of the Ultrasonic Sensors in front of Raspberry Pi module*

necessary sensors would be pinned on. An infrared sensor was used to bring about the line tracking ability. The Board and the remote are connected to the same host to produce the necessary manoeuvres. Obstacle detection was produced by using an Ultrasonic sensor placed at the hood of the car. It was pre-programmed to produce a left turn when the distance to the obstacle falls below 5 cm. The left motor rotates in the opposite direction to turn the vehicle and hence avoid the obstacle.

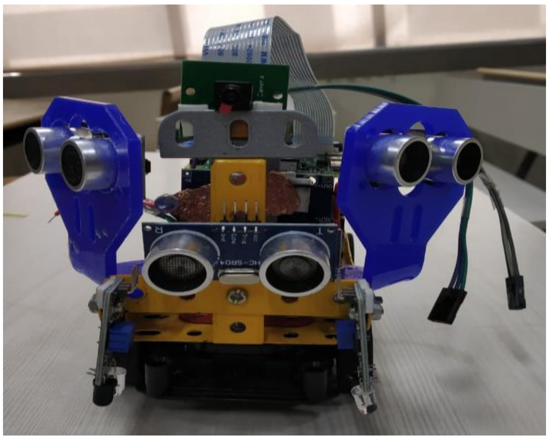

Fig. 9 *Ultrasonic Sensors mounted at the front*

When the car switches to line following mode, an infrared sensor gets to work by transmitting and receiving IR rays. When the transmitter transmits a ray, it is reflected if the surface is white and detected by the receiver. In case of black surface, no light reflects and hence the receiver fails to receive the light ray. Using this simple working, we programmed the module to produce the turns when the sensors catch the light accordingly. For example, when the right sensor is above the black line, the motor driver will drive the motors accordingly to turn right and similarly for the left sensor.

The vehicle is also equipped with the capability to detect the road edges and transform into an actual self-driving automobile. We have trained the module on a road edge detecting computer vision model and then use the test set to evaluate the car performance for further improvement and upgrade. The model was constructed by converting still images from videos into grayscales and then detecting the road lines, masking out the unimportant points from the image and finally understanding the road ahead, all powered by OpenCV and deep learning.

V. IMPLEMENTATION

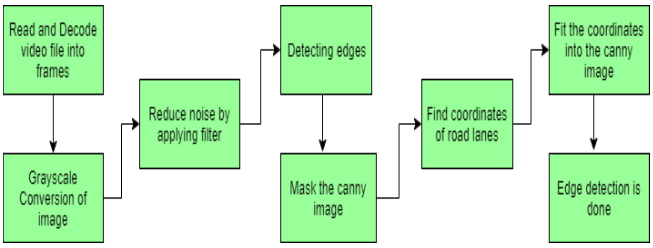

Fig.10 *Flowchart of lane detection*

We have taken numerous input datasets which are stored extensively. We then imported these images into the required process so as to keep as a reference model for distinguishing them. The input datasets contain all the usual image models which could be faced by a visually impaired person on a daily basis. We took our input image using a Raspberry Pi camera module and interfaced it with the Raspberry Pi hardware. Taking many input images of a wide range of objects gives a choice over the objects we would encounter and increases the possibility of recognition. Running the entire process for a higher number of epoch cycles enhances the precision and identifies even the smallest of objects with utmost ease. We have used the HAAR cascade classifier which uses rectangular and fast computation features to identify objects. It uses the moving window technique to define boundaries and identify the object in that given frame. The image datasets were then transformed in a numpy array. We start our code implementation by importing all the packages and parsing the arguments. After working on the frames on the bases of color and resizing it is passed through the neural network so as to obtain the required identifications. If it is above the required level then we get an approximate direction to the object we are looking for. Pre-defining input and output queues helps us threading different processes and makes out task easier. This process gives us a high processing speed. Thus, this multi-processing technique can be easily used for real-time detection of various static and dynamic objects. Thus, this process correctly identifies the object. We have used the NOOBS operating system installer which has the in-built Raspbian and LibreELEC features which helps our cause.

VI. WORKING

All the embedded drive modes of the car can be accessed and switched easily using an online dashboard giving a comprehensive information and current status of the car. On the webpage the user can choose between multiple modes and the information about all of them can be viewed. The Preview-Mode gives a short insight of all the various features

Fig.11 *Project Dashboard*

of the car which is essential to know before starting the operation. Once the sub-menus are selected, the appropriate controls then appear on the screen for easy usage.

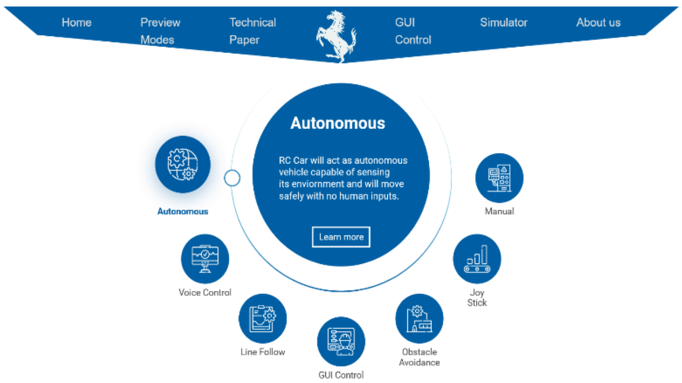

Fig.12 *Dashboard giving an idea about various modes*

The Simulator gives a quick switch between various modes as shown in the below above. First the mode desired needs to be clicked on and on pressing Connect, the mode will be switched successfully.

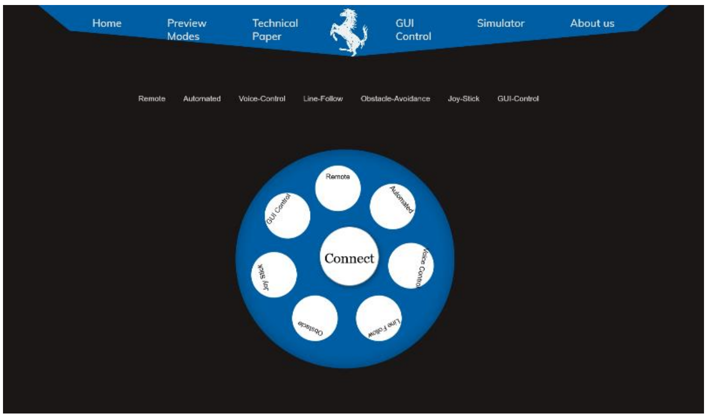

Fig.13 *Drive Mode Selection*

Switch to manual control mode can be made when the user desires to operate the car manually using onscreen GUI powered movement buttons like that in a radio-controlled car. Simple but vital five movements are supported in this mode.

- Forward

- Backward
- Right and Left
- Stop

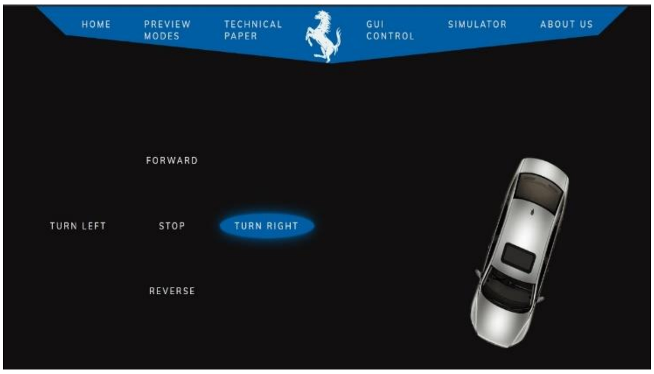

Fig.14 *Manual Mode using GUI*

When the autonomous mode is initiated, quick mode switch is performed and the camera turns on to start live video coverage with AI programmed on Raspberry Pi kicking in. All the frames are processed, type of turn is determined, objects are detected and accordingly all the control systems work together to bring about the automatic vehicle movement.

Fig.15 *System detecting a Stop sign*

The figure below shows the readings taken by the system to control the steering according to the road curvature.

Fig.16 *Road Surface Lane and Curve detection*

All these movements and readings taken by the car can be monitored on the main base device and practically we can observe what the car is literally seeing and performing the calculations automatically on its own after getting trained on the datasets before.

VII. RESULTS AND CONCLUSION

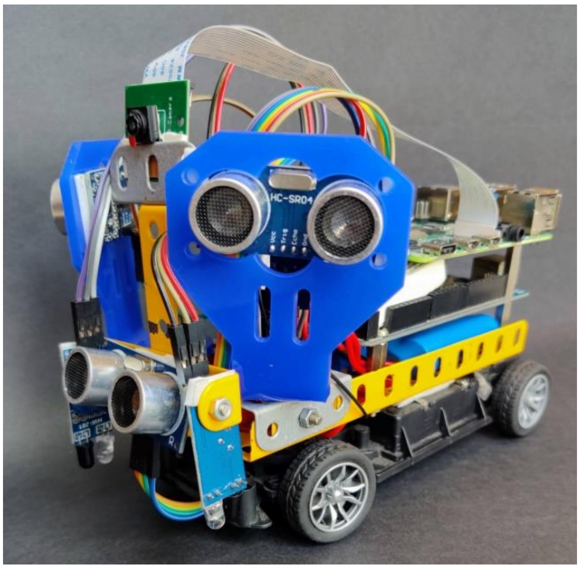

Fig.17 *Final Prototype*

As seen the vehicle successfully detected the road edges, street light and any miscellaneous obstacle on the path and so we noted the behavior towards the dynamically changing environment. The Raspberry-PI camera module captured all the relative objects in the frame and after comparison with the dataset; it recognized the object in the image frame and accordingly drove the vehicle. The exact location between the locator and object was calculated using depth calculation and the speed was controlled depending on the value. This helped in determining if the object was actually present of if there was some error. The vehicle was also able to detect the obstacles on the way using IR sensors and avoid them by reversing the left motor when the distance fell below the threshold value. Quick and hassle-free switching between multiple modes using a webpage was achieved for easy operation of the vehicle. The webpage also provided a comprehensive overview of the project and the team which worked tirelessly and consistently to achieve the objective and the goal of the project.

VIII. SOCIAL IMPACT

The introduction of autonomous cars is getting people more and more interested in promoting transportation means which would allow them to travel wherever they need as fast as possible and while being harmless for the environment and not dangerous for the people. Truly, Artificial Intelligence is the key to achieve this as it will reduce the accidents and the journey time will also be reduced, since they know the best path as instructed by the GPS. It will also give the handicaps, a chance to get the feeling of the driver's seat since they do not need to drive it. Thus, a completely different society might emerge with confidence of moving freely anywhere without the fear of getting killed on the roads. It would also be cheaper along with saving the time spent in concentration on the wheel and instead can be used in doing the things we might like.

ACKNOWLEDGMENT

This project and technical paper were greatly supported by Prof. Aarti G. Ambekar. Her appropriate mentoring and total guidance helped us at every stage of this project and this paper would not have been possible without her valuable inputs and contributions. We would like to thank Prof. Amit A. Deshmukh for giving us the consent to work in the classrooms of EXTC department even in the Covid-19 pandemic. We would also like to thank our colleagues who provided insight and expertise that greatly made an important contribution to the research and helped us with the entire process. We are thankful to all the competition co-ordinators who made it possible to conduct the contest completely online throughout the academic year. We thank the institution for following all the safety and health protocols and maintaining frequent sanitization which enabled us to work freely without any health concerns and risks.

REFERENCES

[1] Pratibha I Golabhavi , B. P. Harish.," Self-Driving Car Model using Raspberry Pi-Issue 2," in International Journal Of Engineering Research & Technology, in volume 9 February 2020. [Online]. Available: https://www.ijert.org/self-driving-car-model-using-raspberry-pi

[2] Prof. Z.V. Thorat, Sujit Mahadik, Satyawan Mane, Saurabh Mohite, Aniket Udugade.," Self Driving Car using Raspberry-Pi and Machine Learning-Issue 3," in International Research Journal of Engineering and Technology, in volume 6 March 2019. [Online]. Available: https://www.irjet.net/archives/V6/i3/IRJET-V6I3178.pdf

[3] Sertap Kamçı, Dogukan Aksu*, Muhammed Ali Aydin," Lane Detection for Prototype Autonomous Vehicle," Computer Engineering Department, Istanbul University-Cerrahpasa, Istanbul, Turkey, Rep. FBA-2019-33004.

[4] Sana Liaquat,Umar S. Khan,Ata-Ur-Rehman,"Object detection and depth estimation of real world objects using single camera" 2015 Fourth International Conference on Aerospace Science and Engineering (ICASE),September 2015

[5] Tiple Anjali Hemant, Tiple Hemant P, Gurav Sagar Hanumant.," Prototype of Autonomous Car Using Raspberry Pi-Issue 4," in International Journal of Engineering Research in Electronics and Communication Engineering, in volume 5 April 2018. [Online]. Available:https://www.technoarete.org/common_abstract/pdf/IJERECE/v5/i4/Ext_56874.pdf

[6] K.N.V.Satyanarayana, B.Tapasvi, P.KanakaRaju, G.RameshBabu.,"Implementation Of Artificial Intelligence for Autonomous Cars-Issue 12," in International Journal of Engineering Research and Application, in volume 7 December 2017. [Online]. Available: https://www.ijera.com/papers/Vol7_issue12/Part-5/L0712057682.pdf

[7] M. M. Farhad,S. M. Nafiul Hossain,Md. Imtiaz Hossain ,Md. Sohorab Hossain,Mohiuddin Ahmad," An efficient moving object detection and distance measurement algorithm using correlation window"2014 9th International Forum on Strategic Technology (IFOST),October 2014

[8] N.Saranya,M.Nandinipriya,U.Priya,"Real Time Object Detection for Blind People,"First International Conference on NextGen Technologies (ISBN- 978-93-86171-90-0)

[9] Chi-Chia Sun,Yi-Hua Wang,Ming-Hwa Sheu,"Fast Motion Object Detection Algorithm Using Complementary Depth Image on an RGB-D Camera " IEEE Sensors Journal (Volume: 17 , Issue: 17 , Sept.1, 1 2017)

[10] "OpenCV | Real Time Road Lane Detection", Dec. 27, 2019. Accessed on: Feb. 7, 2021. [Online]. Available: https://www.geeksforgeeks.org/opencv-real-time-road-lane-detection

[11] "Simple Lane Detection", May. 21, 2018. Accessed on: Feb. 7, 2021. [Online]. Available: https://www.hackster.io/kemfic/simple-lane-detection-c3db2f

[12] Viren Pereira,Vandyk Amsdem Fernandes,Junieta Sequeira,"Low cost object sorting robotic arm using Raspberry Pi" 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS),December 2014