Image Processing Project

Self Driving Car AI

20th April, 2019

Submitted By :-

Abhay Gupta (17ucs003)

Arthak (17ucs036)

Honey Agrawal (17ucs070)

Jatin Kumar (17ucs186)

Submitted To :-

Dr. Puneet Jain

Dept. of CSE, LNMIIT.

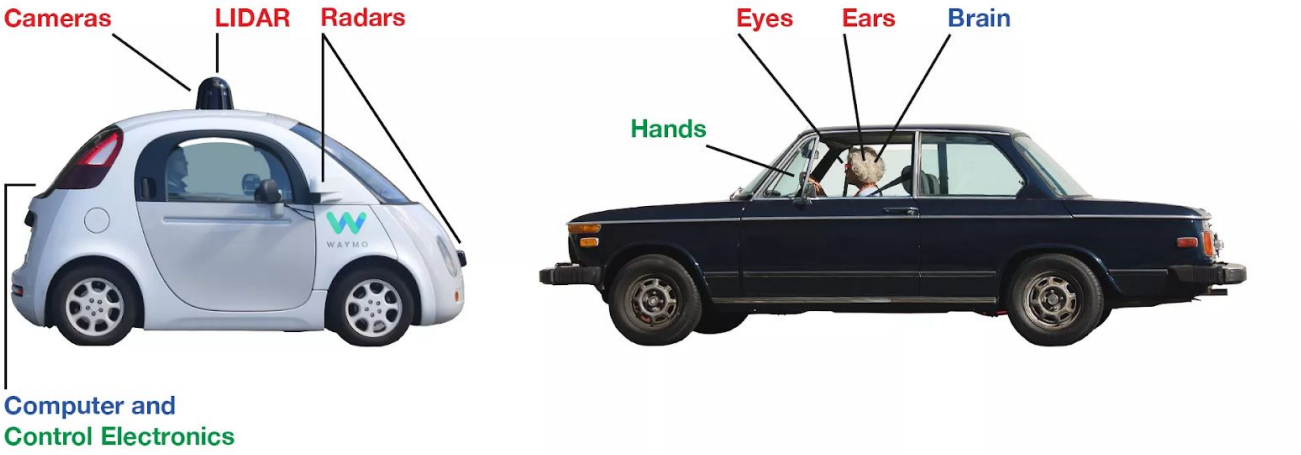
Introduction

Problem Statement

Self Driving Car.

The problem was the increasing number of accidents due to poor human driving behaviour. To solve this problem various experiments, project and research are being done for the self driving car so that number of accidents can be minimised and for the ease of life of course. In contribution to this sector, we have also carried our project in the same domain. Our statement or approach to this problem is as follow.

By using different techniques of Image Processing such Image Segmentation, Image Restoration, Thresholding, Morphology, Edge Detection, Hough transformation etc we tried to detect and identify real time traffic, lane, cars, surrounding etc and these were done with the help of Machine Learning that helped pre train our model to predict these things. Then using these information we have tried to give instruction to car whether which move to take(left or right) and achieve self driving. These things are shown using a game GRID Autosport so that our work can be made graphically and clearly visible and shown in working condition.

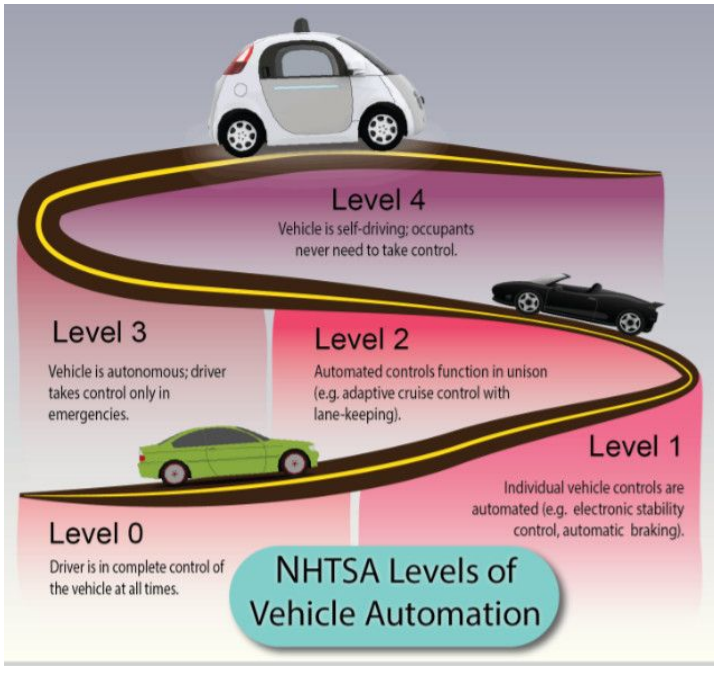


Motivation

A self-driving car (sometimes called an autonomous car or driverless car) is a vehicle that uses a combination of sensors, cameras, radar and artificial intelligence (AI) to travel between destinations without a human operator.
To qualify as fully autonomous, a vehicle must be able to navigate without human intervention to a predetermined destination over roads that have not been adapted for its use. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage.

Application

The top benefit touted by autonomous vehicle proponents is safety.
According to a research that 94% of serious crashes are due to human error or poor choices, such as drunk or distracted driving.
Autonomous cars remove those risk factors from the equation -- though self-driving cars are still vulnerable to other factors, such as mechanical issues, that cause crashes. If autonomous cars can significantly reduce the number of crashes, the economic benefits could be enormous.
In theory, if the roads were mostly occupied by autonomous cars, traffic would flow smoothly and there would be less traffic congestion. In cars that are fully automated, the occupants could do productive activities while commuting to work. People who aren't able to drive due to physical limitations could find new independence through autonomous vehicles and would have the opportunity to work in fields that require driving



Approach

There are following steps we did to approach towards solution :-

1. First of all we captured original frames from the game itself.
2. Then after getting the frames first of all we trained our machine by giving these as data sets.
3. These data sets contained images of car back and front and pictures of lanes.
4. Using techniques like thresholding and then image restoration we identified the required elements from the image and ignored all others.
5. Then we used Image segmentation techniques to form a complete line from the track or lanes (white colored) that was detected so that route for car can be traced.
6. Then after we got the data of the car and lanes we determined whether it should run in which lane.
7. We judged if the current lane is obstacle free. If not we gave car instruction to change the lane.
8. We did this by pressing ‘A’ for left move and ‘D’ for right move.
9. This way self driving was achieved as car can itself decide which direction to move.

Assumptions : The car is running on cruise mode and thus, we don’t manipulate the speed of car. The speed of the car is fixed. Further it is one way highway where it is running so there aren’t criss cross traffic as of now in our case.

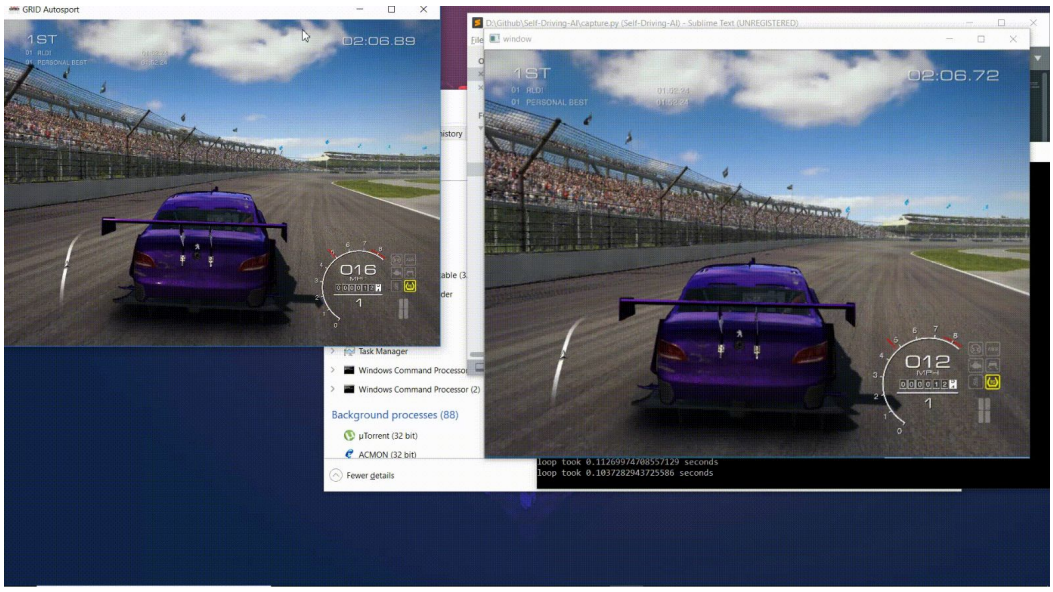
Experiments

Following experiments were performed during this project :-

1. We first experimented about lane detection. Our first experiment was just capturing the images of road and determining the lane from it by thresholding and then image segmentation technique.
2. Our next experiment was getting photographs from the front screen. Then we detected the cars from it. The cars were detected by training my machine with some data sets of cars.
3. Later, we experimented how we can simulate the GRID Autosport game. How we can take control over the car that is moving inside the game. This is done to depict our work on screen because doing the same on real car was not feasible.
4. Our final experiment was combining the results of all these experiments where we can detect lane and traffic and simulate our car in the game by using Image Processing and Machine Learning Techniques.

Algorithm

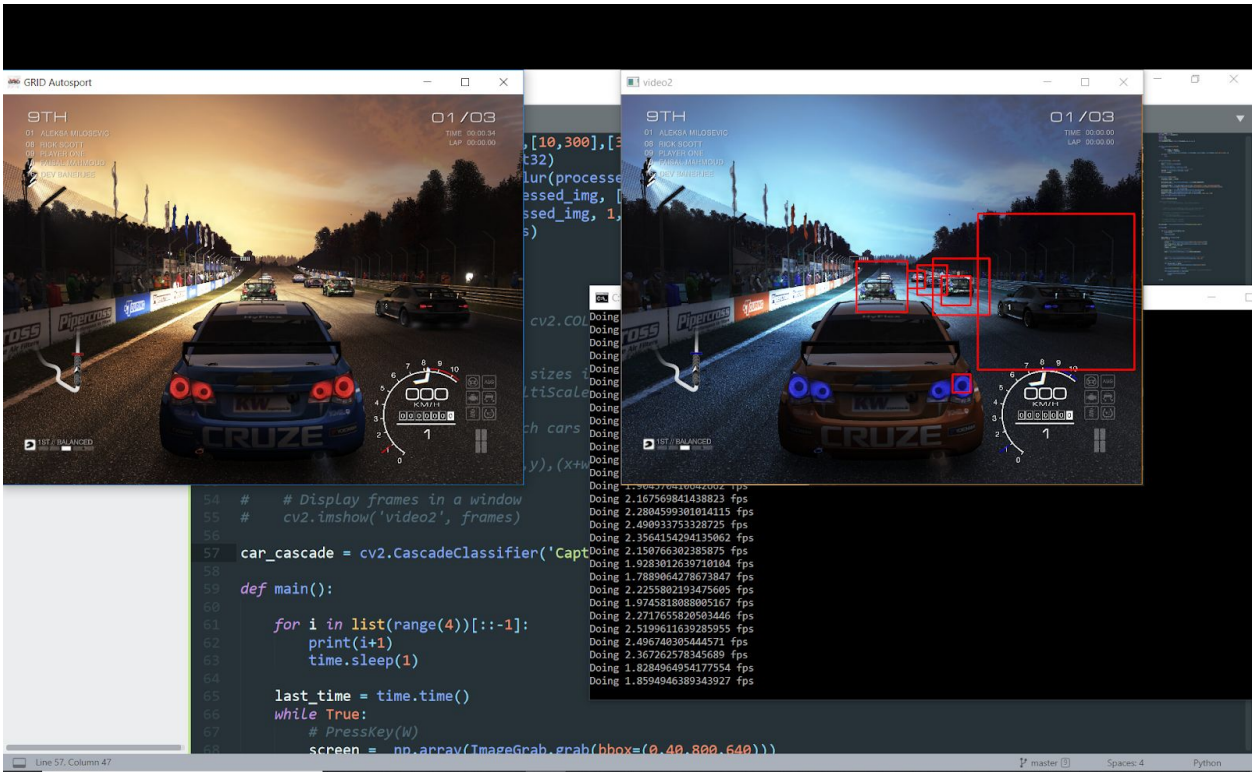
1. **IMAGE CAPTURING:**



Code link for capture and finding edges:
https://github.com/art-hack/Self-Driving-AI/blob/master/capture_and_find_edges.py

Using this algorithm, we obtained real time frames, captured them so that further processing can be done to identify lane and real time traffic. Similar images act as data set as well while training. Later, when real time frames was coming, this acted as situation to in-act for.

2. CAR DETECTION:



In order to control car in a suitable way, we can use a Haar Classifier to detect cars as we will need that data to use braking and collision avoidance, also the size of the formed rectangle can help us know the distance of the car in front from us and also can deem a car with rectangle bigger than certain size to be too close and apply braking.

Code link for car detection:

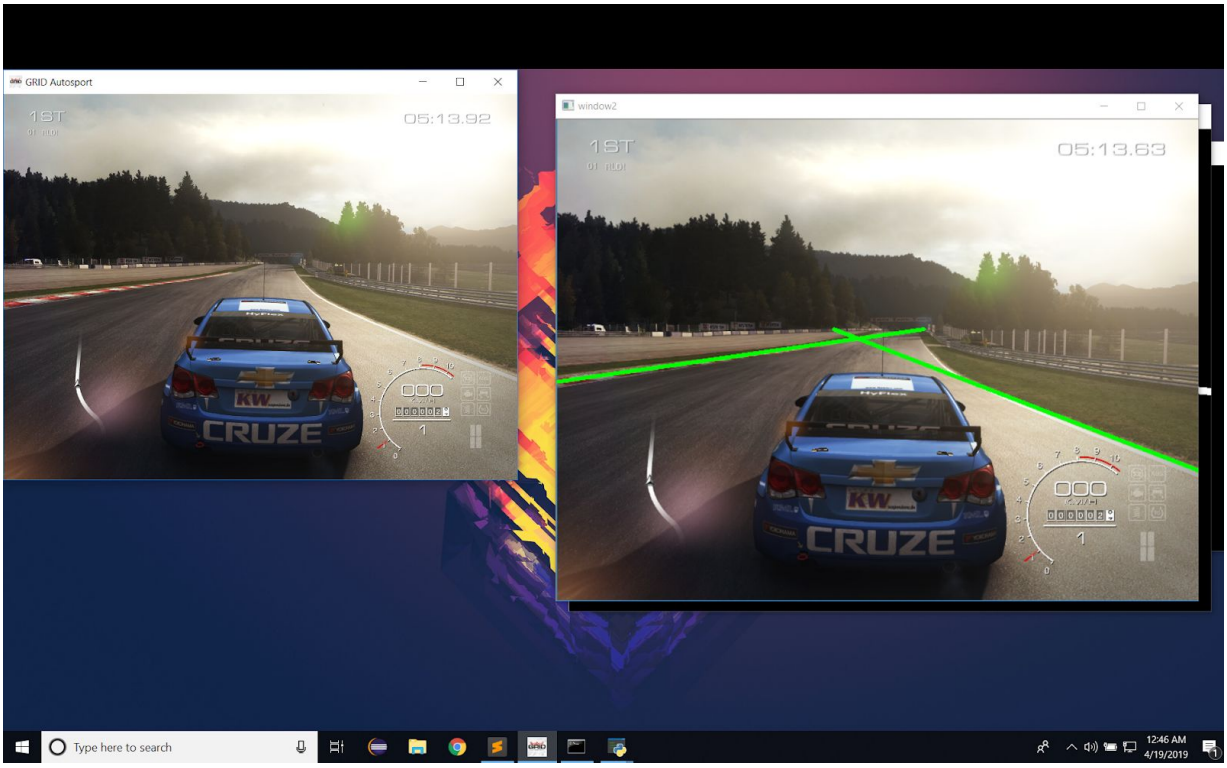
https://github.com/art-hack/Self-Driving-AI/blob/master/car_detect.py

Using this algorithm, we separated out and identified cars from the real time frames that was captured so that we can identify where the traffic is and what move should a car take.

3. HOUGH LINES (LANE DETECTION) :

From this image using Hough Transform we get these lines that detect the sides of the road and depending on the angle of these lines we can get the car to move in a basic algorithm.

- Left line have a positive angle and right line have a negative angle: Go straight (Key W)
- Both lines have positive angle : Go right (Key D)
- Both lines have negative angle : Go left (Key A)

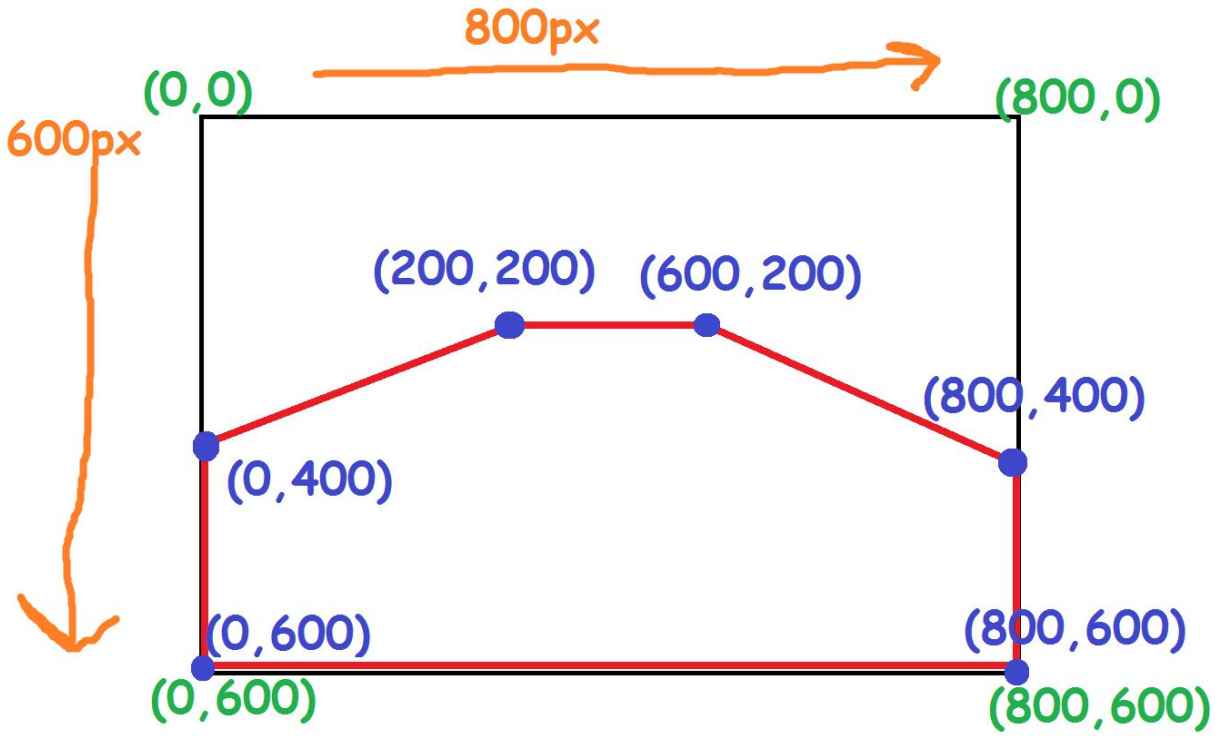


Code link for see Hough Line:

https://github.com/art-hack/Self-Driving-AI/blob/master/hough.py

Using this algorithm, we did thresholding to clearly separate out rest of background and our target i.e. lanes. Then by using image segmentation technique we made lines to make a track to clearly show car a way where it should move.

4. REGION OF INTEREST:

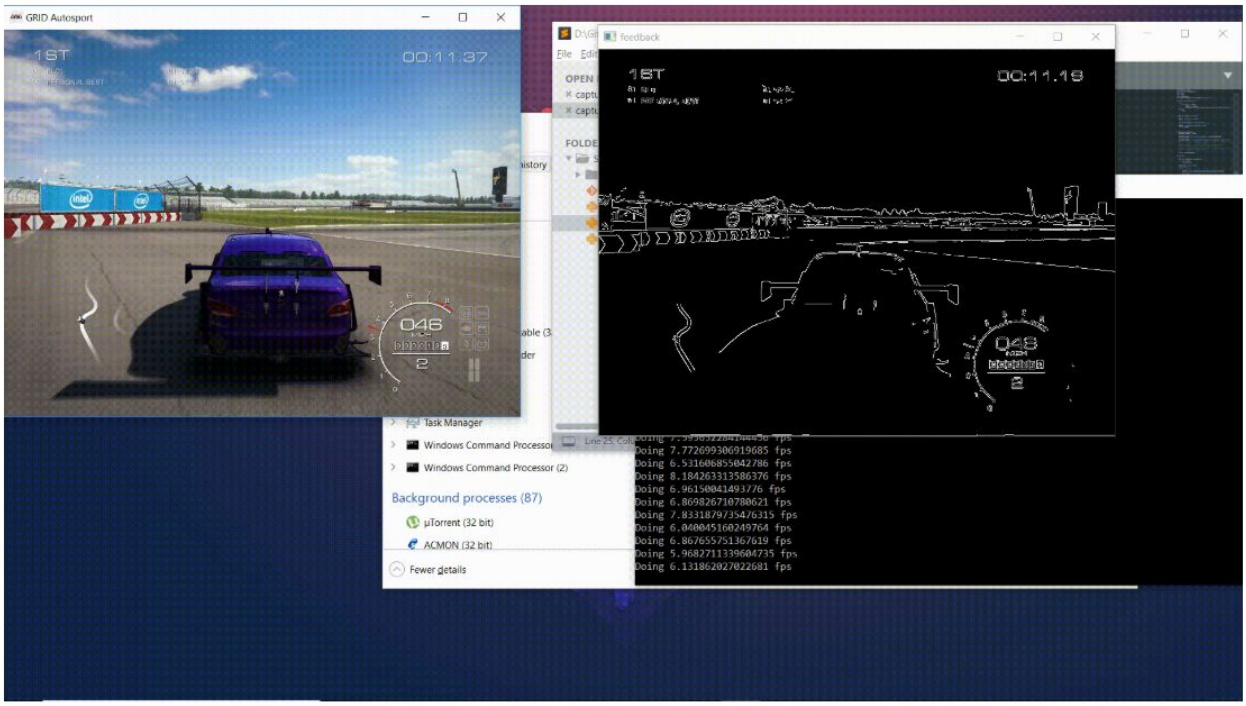


Consider vertices:
[[0,600],[0,400],[200,200],[600,200],[800,400],[800,600]]

We do not need all the frame so in order to remove the sky and unnecessary parts, we took out only the below half in shape of a trapezium like in this image.

Using this algorithm, we were able to identify the things that are of our help like car and lanes and ignore rest of the background.

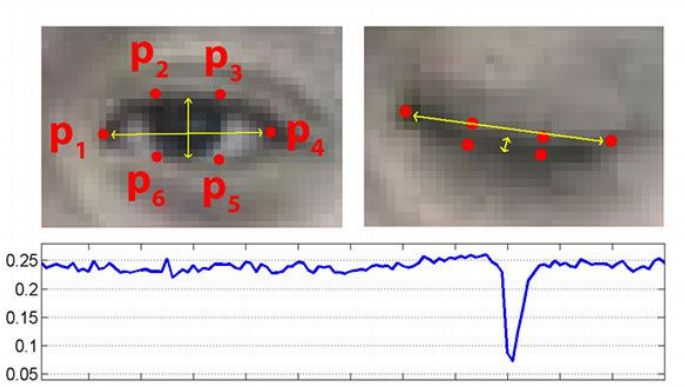
5. EDGE DETECTION:



From the extracted image we had to find the edges, so we implemented edge detection in the retrieved frame as well.

In this algorithm we used Canny Edge Detection techniques to get the shape of the object and thus, finally determine the object as car or something.

6. INDOOR DROWSINESS DETECTION:



We have also implemented a system in which we can check the attentivity of the person driving the car and we can alert the driver in case of a situation in which he/she is drowsy.

Results

In beginning we captured the image of car from the game GRID Autosport. We then corrected our region of interest as we don't need the whole surrounding. We then detected edges and other cars using our region of interest. Results are embedded above in the report.

1. We were able to detect lanes in which our car should move. We detected this using Hough Transform. It was important because we do not want our car to take wrong paths.
2. We the concentrated our efforts on traffic detection as we wanted to avoid collisions with other cars. You can see this result in attached photos. We were able to detect live traffic data. Then we used frames from the game to detect cars and other surrounding elements.
3. We were able to simulate our game.
4. Eventually we were able to run all the three above steps simultaneously to produce desired result.

Discussion

If the people's thought hasn't changed about the self-driving cars being safe, these cars are already safe and are becoming safer. Only if they believe and give a try to technology, they get to enjoy the luxury of computerized driving.

Driverless cars appear to be an important next step in transportation technology. They are a new all-media capsule- text to your heart's desire and it's safe (Wolff par. 10).

Developments in autonomous cars is continuing and the software in the car is continuing to be updated. Though it all started from a driverless thought to radio frequency, cameras, sensors, more semi-autonomous features will come up, thus reducing the congestion, increasing the safety with faster reactions and fewer errors.

People who currently reject self-driving cars would've said no to modern technology and automatic systems.

References

1. Wolff, Michael. “Self-driving revolution is set to begin now.” USA Today, 26 Sept. 2016, p. 01B. Opposing Viewpoints in Context.
2. Blog on how to simulate and control game using python. https://steamcommunity.com/app/255220/discussions/0/558754260200805027/
3. Wikipedia on Self Driving Car.

   https://en.wikipedia.org/wiki/Self-driving_car

4. Automatic Lane Detection. Mateusz Buczkowski*, Ryszard Stasinski´ *Poznan University of Technology

   http://pwt.et.put.poznan.pl/PWT_2012/PWT%202012_3349.pdf

5. Vehicle Detection Using Image Processing for Traffic Control.

   https://www.researchgate.net/publication/265481813_VEHICLE_DETECTION_USING_IMAGE_PROCESSING_FOR_TRAFFIC_CONTROL_AND_SURVEILLANCE_SYSTEM

Table of Contribution

Name	Roll Number	Contribution	Signature
Abhay Gupta	17ucs003	Initial Phase Experiment Performance on Image collection. Finding the game best fit for the purpose. Preparation of final PPT for presentation.	
Arthak	17ucs036	Application of Image Processing Techniques and using Machine Learning. Simulating the game GRID Autosport. Documenting final Report of the project.	
Honey Agrawal	17ucs070	Result compilation from different experiments and relating it to theoretical parts of Image Processing. Judging path for car in real time game. Documenting final Report of the project.	
Jatin Kumar	17ucs186	Collection of Data Sets. Indoor Drowsiness detection system using OpenCV. Control code for car for simulation in game. Preparation of final PPT for presentation.	