NATURALISTIC DRIVING SIMULATION USING AUTOMATION

1st Dr. J.SUBHASHINI
dept. electronics and communication(of Aff.)
SRM institute of science and technology (of Aff.)
chennai,india
subhashj@srmist.edu.in

2nd Gaurav Khanna
dept. electronics and communication (of Aff.)
SRM institute of science and technology (of Aff.)
chennai,india
gaurakhanna1997@gmail.com

3rd Rohan Arya
dept. electronics and communication (of Aff.)
SRM institute of science and technology (of Aff.)
chennai,india
rohan.arya237@gmail.com

4th Leela Guna Krishna Kompalli dept.
electronics and communication (of Aff.) SRM
institute of science and technology (of Aff.)
chennai,india
kompalligk@gmail.com

5th Rahul Gupta
dept. electronics and communication (of Aff.)
SRM institute of science and technology (of Aff.)
chennai,india
rahulgpt823@gmail.com

Abstract—The automobile industry has been finding ways to make driving safer and more convenient. We have reached a point where we need to automate the process of driving and make it more affordable. The present implementations are very robotic and cannot run among human driven vehicles, this made us realise the need of having a more naturalistic driving style for the autonomous vehicles This research paper focuses on making a level 3 autonomous vehicle with the help of computer vision and deep learning. The main objective is to achieve the most ideal results in lane detection and object detection which can be worked upon to improve results. The reason that this project is of such great value is the fact that 80%of road accidents happen due to human error and negligence and we believe that providing a driver aid of some sorts could be really helpful in such scenarios With the help of this paper we will be able to obtain high values with great accuracy with the help of which this can be implemented in real time vehicles by just making a few alterations and by simple calibrations. We believe this will be a step towards a complete autonomous future.

Index Terms—driving, naturalistic, autonomous, computer vi- sion, deep learning, lane detection, object detection

I. Introduction

Naturalistic driving is the way we are able to train the vehicle to drive in a human-like fashion as any vehicle being built has to be driven in a sea of other vehicles. Self-driving vehicles have been a constantly evolving subject and have 5 levels depending on the dependability and human inputs needed. The 1st level is when the car can either control the speed or the steering angle, an example of this is the cruise controls present in vehicles. The 2nd level of autonomy is meant to handle both steering angles and braking for the vehicle. The 3rd level allows the user to take his eyes off the road but needs to be mentally aware of the surroundings. The 4th level is a self-dependent system where the user input isn't needed for most of the journey. Level 5 is a completely automated system and doesn't need any human even be present in the vehicle. The history of autonomous cars dates back to 1939 which was an electric vehicle guided with the help of radio-controlled electromagnetic field which was guided with the help of magnetized metal spikes which was later improved upon with a camera based system in 1977 by the Japanese but was flawed due to the lack of resources at that time. The mainstream market today provides level 1 autonomy in the help of assistive parking and braking systems or adaptive cruise control. Companies like tesla were able to achieve level 2 of autonomy and are on the way to achieve level 3. Level 4 and 5 is yet to be introduced by manufacturers as the data set needed to train the vehicle is insufficient at this moment and people are not very supportive of the technology as we end up giving too much control to technology, it is very easy to differentiate between a car driven by a human compared to a car be autonomously driven and this is the biggest flaw due to which we haven't reached a fully autonomous system. The project can be simulated on various Open Source Simulators such as udacity autonomous simulator, Carla simulator, lgslv simulator, etc. In our project we have used Udacity autonomous simulator for lane detection and path following, and Carla for object detection (vehicles, pedestrians and traffic lights). We have used the OpenCV library in order to perform the desired tasks. We have used

Canny Edge Detection and Hough Transformation to detect the edges and the boundary lines. Carla uses the same feature as mentioned above. The object detection part is done in Carla sim. We have made a dataset consisting of required objects to detect and trained it (including traffic lights). For a much faster detecter, "YOLO" can be used which requires a huge amount of processing power

II. Literature Survey

1) Lane Detection: Lane Detection [1]upholds the most vital role in the field of Autonomous Driving as it deals with organizing the speed of the vehicle with the response to traffic congestion. Lane Detection in a practical Scenario is very challenging as it totally depends upon the quality of light from the surrounding and the light coming from other vehicles. Yuenan Hou [2]in his work used a concept of Self-attention distillation(SAD) for lane detection. SAD works on a simple approach that allows the networks that deal with lane detection to make it learn of itself without the need for extra supervision or additional labels. Briefly, SAD can be Understood as an observation involving the training of a lane detection to a reasonable level and the maps which have been brought up at different layers gives the rough outline of scene and this learning is added to the half-trained model which strengthen the representation of lane. The major drawback this paper holds is that it's for boosting the performance of small lane detection network.

The method that we used involves the conversion of an image into a multi-dimensional NumPy array (an image converted into the matrix in which pixel intensity is defined from 0 to 255)which is later converted into grayscale which makes it easier to differentiate between the lanes and the background because usually, lanes have the high intensity so if we converted into grayscale there will be only to the variable which we need to worry about for black 0 and for white as 255 [3].

0 0 255 255
0 0 255 255
0 0 255 255
0 0 255 255
0 0 255 255

this kind of the matrix is formed o represents black white represent high-intensity post-processing eliminates the outliers which result in smooth image .now edge detection which identifies the sharp changes in the intensity in adjacent pixels. changes in the brightness known as gradient change there are strong gradient change and small gradient change gradient. the gradient should be strong which detects the edges.

2) Object Detection: This stage is used to create an algorithm to detect the various objects in the surroundings of the vehicle and classify them for the autonomous vehicle. This is possible due to the camera array present on the vehicle which captures images and processes them at the desired rate depending on the hardware.

a) To Detect an Object: This is mainly used to expand the data sets for the vehicle, by expanding the dataset we can help understand the environment. The dataset is collected from the support of the lidars and cameras present on the vehicle. Then these collected images are distributed based on the object it represents [4].

b) Object Comparison: : The captured image of the object is compared with the already present data set with its assistance we can divide the images into classes this can be done by comparing the features present in one of the images with the immediate next images captured. This is done by a process called ORB which stands for oriented fast and rotated brief, this process helps us in object tracking in the various images captured [5]. ORB uses key points from each image captured and compares them with an algorithm known as FAST(Features acceleration and segments test) this compares the movement of those key points and depending on the intensity changes in the pixels it can understand the movement of the overall object and the shape and dimensions of the image. The endmost step is done with the help of the OpenCV library, with the help of its Brute Force Matcher the captured image is compared with the pre-existing data set to identify the image. This is not very accurate but can be increased with the increase in the data set to improve accuracy [6].

c) CNN (Convolutional Neural Networks): Convolutional Neural Network (CNN) is a kind of Artificial neural network, made to train with a set of images which are known previously and learn from the provided information. The image is represented in a 3-dimensional matrix format. This consists of length, width, and depth. Depth holds the information about the three color channel RGB(Red,Green,Blue). If conventional artificial neural networks (ANNs) are used for training of an image, the size of the initial layer would be represented by a huge set of each input which represent a single color channel of a single pixel. It can be understood by taking an example of an image whose resolution is 50X50X3 ending up giving 7500 input data and such a large amount of database make neural networking complex.CNNs at this point plays an important part as it uses depth of an image and model it in the form of volume and trains it. While doing so, the initial layer studies the edges and lines present in an image. The other layer deeper to the initial one gains the information about faces or shapes [7].

d) YOLO (You Look Only Once): YOLO expanding to "you only look once" is a method to achieve high levels of accuracy and give us results in real time. The algorithm refers each image only once and sense the predictions needed to be made by the neural network and hence giving us a forward propagation. After moving through the algorithm the outputs are placed together in the form of hopping boxes.The CNN is able to simultaneously predict the various hopping boxes and classify the probability of those boxes with the help of YOLO [8].This algorithm trains on full dimension images and advances with time. This model has a number of benefits over other object detection methods:

- YOLO is a more reliable process for object detection as it is extremely fast.

- Instead of seeing parts of an image is able to detect by seeing the whole image during the training and examining of the neural network and is able to encode the information of the particular image into its particular class with their appearance.
- YOLO is able to learn of how generalise a cluster of images and use it to its advantage. This method is able to outperform any other algorithm of object detection.

III. Methodology

For Lane detection, the techniques we have used are Hough Transform, Canny Edge Detection, Bilateral Filter.

1) Hough Transform: Analyzing an image and Processing an image, have the features which are retrieved by using Hough Transform. Considering an image space, the edge detection in the primary stage (pre-processing) is used to figure out the points in the desired curve. But images may or may not be perfect and the edge detector may have some imperfections too. So, because of that, few pixels may miss on the curve. To avoid this problem, the Hough Transform has been introduced [9].

2) Edge Detection: Depending upon the differences in the brightness of an image, Edges are estimated at those points. The main benefit of edge detection is that it does not acknowledge the data which is insignificant as the data that has to be processed gets reduced by the algorithm. In this project, we have used Canny Edge detection as the algorithm for detecting edges which is a multiple-stage algorithm [10].

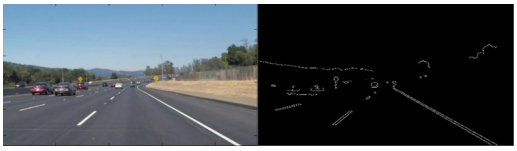

Fig. 1. **Edge Detection using Canny Algorithm**

3) Bilateral Filter: The basic scheme of Bilateral filter is similar to the anti-aliasing technique, but leaving or preserving the edges alone without changing and smooths the image if the two pixels are close to each other. This is a non-iterative scheme and used to make sure whether the vehicle follows the right path or not [11].

A. System Design

1) Sensor interface modules: This section indicates, all the communication between the sensors on the vehicle and the vehicle is done in this block. The section enables us to provide the various kinds of data from the sensors to all other blocks. The main sensors include camera, radar, GPS, IMU, wheel encoders and LIDARS.

2) Perception modules: These modules perform processing on perception data from sensors such as LIDAR, camera, and radar and divides the data to find moving and static objects. They also help direct the self-driving vehicles relative to its digital map of the environment.

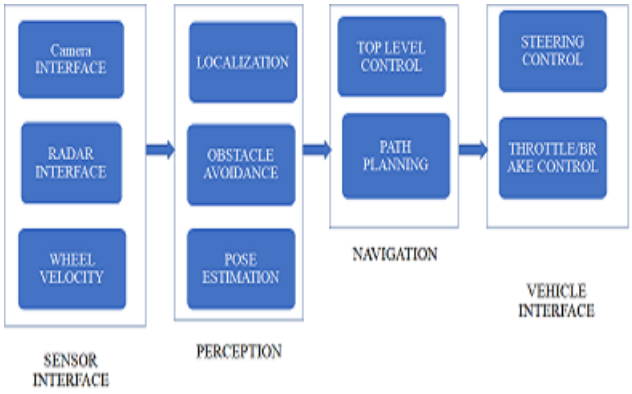

Fig. 2. System Architecture

3) Navigation modules: This module determines the behaviour of the autonomous car. It has motion planners and finite state machines for different behaviours in the robot.

4) Vehicle interface: After planning the path, the controller commands the steering, throttle, and brake control, are sent to the vehicle through electronic signals to control the vehicle.

B. Hardware and Software

1) Raspberry Pi v3: It is a pocket-size computer which can be operated by a single board. The pi3 B+ model has a 4xCortex-A53 1.4 GHz processing unit.It has a memory size of 1GB which is shared along with GPU, it has 4 USB 2.0 port [12].The video and images are accessed using the MIPI camera interface (CSI) connector, used with the Raspberry Pi camera. It has a micro SDHC slot for onboard storage with Wifi and Bluetooth module [13].

2) Arduino uno: It is a microcontroller board based on an open-source platform, the board consists of multiple analog and digital pins which can be linked with various shields and circuits.The board consists of 14 digital I/O pins out of which 6 pins are capable of PWM output, 6 analog I/O pins, and is programmable with the Arduino IDE, via a type B USB cable [14]. The power supply can be given using a USB cable or by an external 9-volt battery as it accepts voltages from a range of 7-20 volts.

3) Motor Driver-L298n: L298n is a motor driver operating on a paired H bridge circuit whose work is to switch the polarity whenever there is a change in voltage. The H bridge is controlled by PWM. It can drive a motor between 5V-35V and a maximum current of 2A. It uses an L298 IC and has the onboard 5V regulator which it can provide to an external circuit. It can manage a maximum of 4 DC with directional and speed control. The two motors are given a PWM signal using Enable pins present on the board.

4) Boost Converter: A Boost converter increases a DC voltage from the input to the output. The circuit operation depends on the conduction state of the MOSFET: During the MOSFET is in on condition than the current through the inductor increases linearly and the diode blocks. and when it is in Off-state then the current passing through the inductor

can not change rapidly, the diode must lift the current so it can pass and begins to conduct. Energy is transferred from the inductor to the capacitor resulting in a decreasing inductor current. During steady-state, the circuit is said to operate in two modes i.e. n discontinuous conduction mode if the inductor current reaches zero and in continuous conduction mode if the inductor current never reaches zero.

5) Arduino IDE: It's a multiple platform application for an operating system like Windows, UNIX system and many more which are written using functions from C and C++ language. The basic work is to build and compile the programs to the boards which are compatible with Arduino, but with the help of third part cores, it can be compatible with different boards. The Prototype is trained using Arduino IDE which involves the movement of the vehicle over the lane.

6) OpenCV: Open Source Computer Library is a software library used in the field of computer vision and machine learning. OpenCV is drafted in C++ programming language and the binding is present in Python, Java and MATLAB. It was developed to provide a common foundation for applications in Computer Vision and to advance the use of machine recognition in commercial products. OpenCV is inclusive of more than 2500 optimized algorithm consisting of Computer Vision and machine learning algorithms.

7) VNC: It is known as Virtual Network Computing, which enables the user to share the graphic interface of the desktop remotely. It follows the frame buffer protocol which gives access to the user to remotely control another computer over a network connection. It works on the model of Client/Server. VNC shares the Keystrokes, movement of the mouse and the clicks and other data with the server.

8) CARLA: It is an open-source Simulation Environment for the research of Autonomous Vehicle.CARLA involves the development of the Support System, Training of Vehicle and authentication of Autonomous vehicles [15]. Other than being open-source, CARLA provides the assets involving buildings, pedestrians, a vehicle that can be accessed freely. The Simulation environment provides many features as Autonomous Vehicle sensors suite, Flexible API, map generation and many more.CARLA is used in this project to show how the vehicle is detecting the object and respond to it.

9) Udacity: Udacity has developed its self-driving Car simulator Source code which can be used to train your learning vehicle model to follow the driving behavior. It operates in two modes i.e. Training mode and Autonomous mode. In Training mode user manually drives the vehicle along the path where your vehicle captures the road image which can be processed in Autonomous mode. In Autonomous mode, the vehicle uses the machine learning model to predict the best driving instructions [16]. The driving instruction includes mechanics such as Steering angle, acceleration, brakes and throttle which are the major element for bringing change in speed and direction of the car.

IV. Results

The use of udacity simulator we are able to train our model to give an accuracy of 90% on a left biased road (vehicles driving on the left side of the road) and we are able to understand the importance of training the vehicles ourselves instead of using only algorithms as there are a few cases where the algorithm with cause a very robotic movement of the vehicle and is certain to get off track or bang into some other vehicle.

Fig. 3. **Simulation on Udacity**

The above images depict how the vehicle follows the track without going off track on two different kind of terrain.In Udacity the vehicle is manually trained to follow the path.While training, The vehicle captures the images of the path and generates a database which makes the vehicle to Autonomously follow the generated path.In addition, It also keep the track of speed,Steering angle,acceleration and brake.

Using the Carla simulator, we were able to create a real time environment for validating our code. We are able to create a small world we real time objects, we were able to understand that our system worked perfectly in various weather conditions as well as various situations. We were able to detect traffic signals and stop signs in real time and were able to teach our vehicle how to react to the various situations that a vehicle would face on the road .To validate both the above software simulations, we were able to build a prototype and train the model using basic image processing and get the desired output.

Fig. 4. **Simualtion on CARLA**

The image represents the vehicle detecting the objects in its surrounding to ensure it doesn't collide with anything in it's path or checks if any incoming vehicle is coming in it's path.

Fig. 5. **Detection of Green Signal**

The above image represents our trained vehicle detecting the signal turn green with the help of object detection understanding that the vehicle can now move. It can detect the vehicles around it to see there is no issue or incoming vehicle before following its original path.

Fig. 6. **Detection of Red Signal**

The trained vehicle with the help of object detection sees that the vehicle turns red indicating that the vehicle should stop and stays stationery until the signal turns green. When the Vehicle analyses the Signal colour and it display red its acceleration value automates to zero and vehicle stops at that point.

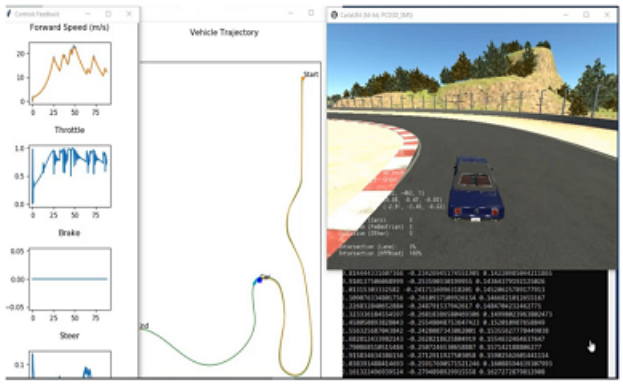

Fig. 7. **Data obtained from vehicle while training**

This is a real-time representation of the Udacity simulator used, we can see that the location of the trained vehicle shown in terms of its coordinates at the bottom right telling, a 2D way-point created (in the vehicle trajectory file) by the vehicle to better understand the path to be followed under which our vehicle can vary the speed of the vehicle by changing the throttle input in a way as we can see this is done without breaking, in this case, reducing the losses caused because of braking saving fuel. The steering angle is calculated by measuring the change in the road's angle. This method helps the vehicle in following its path without going off track, giving high accuracy. This result verifies our code, giving us the desired accuracy. The vehicle can travel from a point and point, a set on the waypoint without going off the track and by following the lane whenever needed.

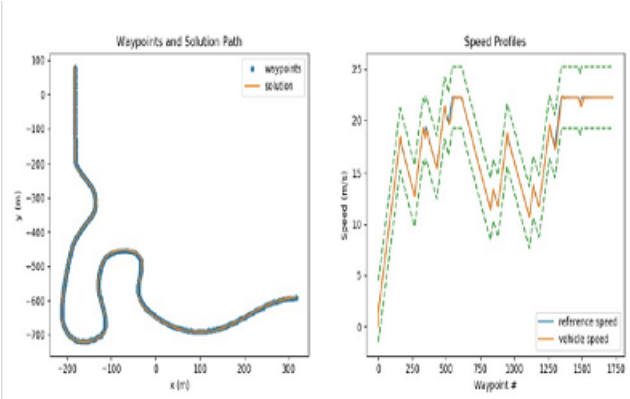

Fig. 8. **Generated waypaths and speed profile**

For the graph depicting way point and solution path, we created a 2D path for the autonomous vehicle to follow by providing necessary way point to find out the precision and accuracy of the vehicle.The reference speeds of the vehicle are

given by us during training and the vehicle speed is a graphical representation on how it lies between the reference speeds all the time.

V. CONCLUSION

This paper deals with a different approach to level 3 autonomous driving by using relatively inexpensive hardware compared to the real time vehicles present on the roads. The use of CARLA and udacity simulator helped us understand the constraints that autonomous vehicles faced and how using naturalistic driving solutions we are able to obtain a more interactive driving experience. Using the above results we can understand that by increasing the possible data sets and with continuous training of the above code we can take this project to higher levels of autonomy. The only constraint in the paper is acceptance of autonomous vehicles in a sea full of drivers. But this is a step towards a more accepting approach as the results have proven to give a really high level of accuracy and is only bound to improve with time.

REFERENCES

[1] M.Bertozzi and A.Broggi.Gold:Aparallel real-time stereo vision system for generic obstacle and lane detection. IEEE Transactions on Image Processing, 7(1):62–81, 1998

[2] Yuenan Hou, Zheng Ma, Chunxiao Liu,and Chen Change Loy. Learning Lightweight Lane Detection CNNs by Self Attention Distillation.International Conference on Computer Vision (ICCV), At COEX Convention Center, Seoul, Korea,2019

[3] 3- L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: scale-aware semantic image segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3640–3649, 2016

[4] D. J. T. M. J. Girshick R., "Rich feature hierarchies for accurate object detection and semantic segmentation," in Computer Vision and Pattern Recognition, 2014.

[5] D. N. a. T. B., "Histograms of oriented gradients for," in Computer Vision and Pattern Recognition, 2005.

[6] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in International Conference on Computer Vision, Barcelona, 2011.

[7] "Convolutional Neural Networks for Visual Recognition," [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[8] Redmond, "YOLO: Real-Time Object Detection," [Online]. Available: https://pjreddie.com/darknet/yolo/.

[9] Tseng, Chien-Cheng, Hsu-Yung Cheng, and Bor-Shenn Jeng. "A lane detection algorithm using geometry information and modified Hough transform." 2005 IPPR.

[10] Mallick, Rajendra Nath. "Implementation of Canny Edge Detection Algorithm." PhD diss., 2012

[11]] "Bilateral Filter", [online available]: wikipedia.org

[12] "Introducing Raspberry Pi Model B+". Raspberry Pi Foundation. Retrieved 14 July 2014

[13] "diagram of Raspberry Pi with CSI camera connector". Elinux.org. 2 March 2012. Retrieved 22 June 2012.

[14] "Introduction to Arduino" (PDF). priceton.edu. Archived from the original (PDF) on 3 April 2018. Retrieved 4 February 2018

[15] Alexey Dosovitski,German Ros,Felipe Codevilla,Antonio Lopez and Vladlen Koltun. CARLA: An Open Urban Driving Simulator, IN 1st Conference on Robot Learning (CoRL),2017

[16] https://medium.com/activating-robotic-minds/introduction-to-udacity-self-driving-car-simulator-4d78198d30d