e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:02/Issue:05/May-2020 www.irjmets.com

AUTONOMOUS DRIVING SIMULATION USING COMPUTER VISION

Abhishek Hegde[*1], Yash Gupta[*2], Swapnil Shikhar[*3], Hrishikesh Verma[*4], Raghav S[*5]
[*1,2,3,4]Student, VTU, Department of Information Science and Engineering,
Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India
[*5]Associate Professor, Department of Information Science and Engineering,
Sir M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

ABSTRACT

Autonomous driving has been a topic of discussion in companies from many years. It has always been promised that this innovation can bring life-changing safety and ease to the otherwise prosaic life of the people.Although something truly driverless is practically nowhere in sightof being available to consumers, itdefinitely is closer than many think. Current estimates speculate thatthe world in 2025 will see over a hundred thousand self-driving cars on road, and by 2035 that number would have sky-rocketed to almost 21 million. The objective of our project is to use a Convolutional Neural Network (CNN) to learn safe driving behavior and smooth steering maneuversso as to provide a virtual and cost-efficient testing environment forimplementing autonomous driving.The training data will be collected from 3 cameras (front-facing) and the steering commands, torque and speed will also be collected corresponding to the images. This data will be used to train the proposed CNN to implement AutonomousDriving. Section III explains the detailed methodology implemented. An appropriate neural network is constructed and the hyper parameters are adjusted to achieve maximum performance in metrics of accuracy, the detailsof which are provided in Section IV. Finally, Section V illustrates the results obtained.

KEYWORDS: Computer Vision, Autonomous Driving,Artificial Intelligence, Simulation, Road Safety.

I. INTRODUCTION

Scrutiny of developments till date in the world of automated driving systems reveals that with the usage of tortuous systems such as that of cameras, lasers, radar, GPS, and interconnected communication between vehicles, some car models can now offer functionalities such as, but not limiting to, the following:

- Provide parking assistance;
- Estimate the speed of vehicles in front, and monitor speed of vehicle thereupon;
- Avoid accidents by braking, steering and change of acceleration;
- Vehicle centering by lane width and speed maintenance;
- Collision avoidance by blind spots checks before lane changes and alerting you if another car is in yourway.

New add-ons are being introduced almost on a daily-basis in the market on these features. For an instance, companies are working towards cars that can provide door to door pickup and drop, meanwhileautonomously parking themselves, then coming back to retrieve you as and when summoned by your smart phone. Each autonomous feature added, makes the dream of having truly driverless vehicles seem more and more feasible and imminent.While the benefits of driverless cars are numerous – think increased safety, transportation outreach to new population segments, a considerable increase in the time of productivityof drivers – there are also many issues that will need to be addressed as this technology becomes more advanced. Who will take upon the responsibility if anautonomous vehicle gets in an accident? Who decides the code formats and criterions? How will the insurance department address the concept of driverless cars? What kinds of new risks will immerge and will need to be insured against? How will the federal and state laws and regulations be affected and adjusted as autonomous cars become commonplace? And how will litigation be affected by this innovation? Although answers to many such questions may remain hazy or unclear at this point, it is of prime importanceto start considering how driverless cars will accord to and revolutionizeour society. As futuristic as self-driving cars may seem, extensive autonomous features can already be found in many cars on the road today, and a truly driverless car may not be an immediate event but definitely is an eventuality.The world of autonomous cars is evolvingwith each passing day, with more and more tech beingadded each month. Thus, we thought of doing our bit to the scientific community by proposing our take on this in-demand topic.

II. LITERATURE SURVEY

Over the years, the research area of making cars self-driven has seen much work put into it, but due to advancement in technology and the increasing population,this dream is far from fruition. The year 2009 sawthe launch of Google's self-driving car venture, started by colleagues who had effectively devoted years to the innovation. 2012 was when the first Google car hit the road for testing. The upcoming years saw major developments and innovations to the primary model and the car got equipped with multiple sensors, radars, lasers, Global Positioning System (GPS), heavily detailed maps, and many other things to better the ride safety drive and navigation without human intervention. The car could not only achieved autonomy in driving but alsoparking. Driving on freewayswas made possible now as cameras were used to find and detect objects that are then processed by the computer within the car. A new concept for their driverless car with neither a steering wheel nor pedals was presented in May 2014 by Google.A fully functioning prototype was unveiled in December of that. In summer 2015, different features were launched and tested, some where each model's speed is limited at a neighborhood-friendly 25mph. Moving over to some other significant researches carried out, in apaper titled:'Self-Driving and relaxing driver vehicle' [2] two applications of an automated car are focused, one in which two vehicles have the same destination and one knows the route, whereas the other doesn't. The 2ndvehicle tails the target (i.e. Front) vehicle automatically. Automated driving during a heavy traffic jam wasthe other application, which relaxed the driver from continuously pushing brake, accelerator or clutch. The Google car was again the main inspiration for the idea described in this paper. Dynamic decision making was the one aspect under consideration here. A vehicle automatically following the destination of another vehicle was the solution proposed. The aspect of taking intelligent decisions in the traffic, being also an issue for the automated vehicle, has been taken under consideration in this paper.In another paper titled:'Towards Behavioral Cloning for Autonomous Driving'[3]an off-policy imitation learning methodology for autonomous driving using a doubly-deep recurrent convolutional architecture ,been referred to as NAVNet (Navigation Network) and being end-to-end trainable,that learns compositional representations in both space and time domains has been proposed. The recurrent long-term models are directly connected with the visual convolutional models. This approach is non-data driven and the system learns a regression-based mapping function between input images and steering angle. Even though there are a lot of ways and techniques for the implementation of the concept of autonomously driven vehicle, yet there is a huge room for improvement. In this paper we discuss one such methodology which proposes the use of artificial intelligence using a Neural Network based architectureproposing a way using which the objective of autonomous driving can be achieved and tested on any simulator or hardware using the appropriate API for connection.

III. METHODOLOGY

The process will mainly consist of building the Virtual Autonomous Driving Car model using CNN

Architecture:

a) Using Computer Vision (OpenCV) to find lane lines:

Loading Image: imread() method of the OpenCV library reads a multidimensional numpy array ofpixel intensities.

Grayscale Conversion: cvtColor(img, cv2.COLOR_RGB2GRAY) converts color of 'img' fromRGB2GRAY. This is done to make processing easy and faster as gray contains only one channel ofintensities as compared to three in RGB.

Smoothening Image: A 5x5 kernel will be created of set weighted averages acc. to gaussiandistribution (the farther the pixel, the lesser the weight assigned). cv2.GaussianBlur(gray, (kernel,kernel), 0). Blur helps rule out fake boundaries and reduces noise in the native image.

Canny Edge Detection: cv2.Canny(blur, 50, 150), this function ensures that pixel differences morethan 150 be turned to ->255, less than 50 ->0, and in between ones be 0->255 only if they touch theboundary. Thus, highlighting the required boundary regions.

Region of Interest & BitwiseAND: Return masked canny image of only the triangular area underconsideration. An array of zeros with dimensions same as that of canny image is created. Usingmatplotlib, the area of interest's co-ordinates are retrieved and in a sea of zeros, every pixel enclosed intriangle will be made 1. All irrelevant areas of canny will be deleted after the bitwise and operation onthe original image. (and operation).Lane Detection & Optimizing: Now, we call the averaging method to get two lines, right averagedand left averaged lines using the

Hough Transform technique. Then, we overlay 100% of line imageover 80% intensified lane image to makes lines visible clearly. (Figure-1)

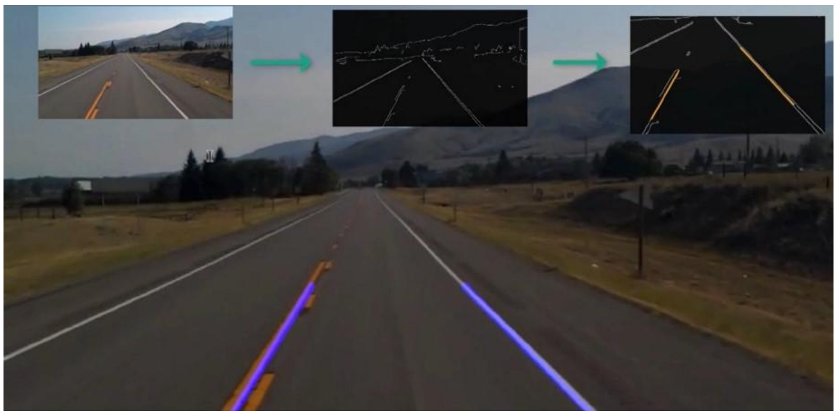


Figure-1: Finding Lanes**.**

b) Getting training data from open source simulator: Udacity Self-Driving Simulator:

The 'train' option of the Udacity self-driving car simulator lets us capture dashboard images from an imaginary right, left and center dashboard cam mounted on the simulated car. These images are named accordingly and stored as PNG files. We have used approximately 16k total labeled images. This means that each image has its associated labeled steering angle stored as a table in an excel file.

c) Building a learning model using CNN:

The NVIDIA's model was used as a basis to improve upon for prediction of steering angle, on input of a preprocessed image. The steps involved in a CNN pipeline such as convolution, pooling and flattening were carried out and the flattened output was passed through multiple fully connected layers to form a Deep CNN architecture. The layers in between were appropriately given activation functions such as 'relu' or 'elu', finally one outputwas generated as the steering angle.

d) Setting appropriate hyperparameters:

In order to decrease our target metric: the mean squared error loss, we had to adjust the values of the hyperparameters such as number of layers, number of nodes in each layer, order of layers, activation function, optimizers, number of epochs, learning rate, batch size, data split ratio, kernel size, strides, padding etc.

e) Using Image Augmentation to avoid over-fitting.

In order to improve the generalization of the model, we built our own image augmentation functions to augment more images into the training dataset by introduction of random pan, zoom, brightness, flip and rotation.(Figure- 2 and Figure-3) This would greatly reduce the validation error along with having a much better performance on a variety of test tracks.

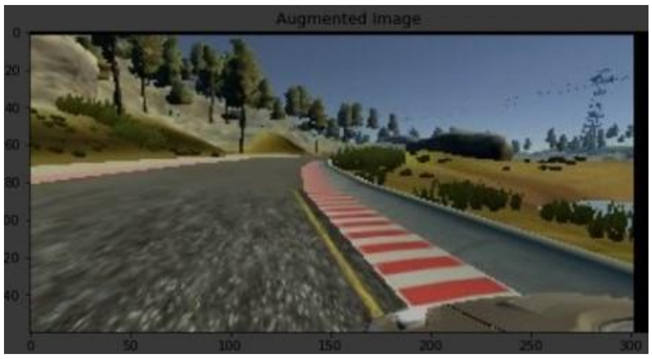


Figure-2: Image Augmentation- Brightness.

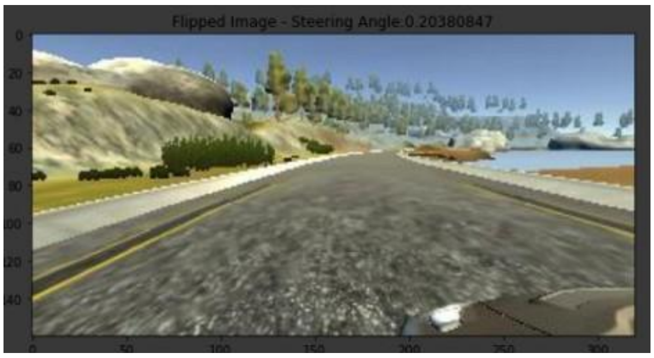


Figure-3: Image Augmentation- Flipping.

f) Building an API to communicate with the simulator:

We had to build a Flask API to communicate with the simulator. The API would implement pre-declared functions of the simulator involving telemetry and connection and other aspects of the simulator. This was the main interface connecting our AI model to the simulator, where the actual output would be displayed.

g) Testing out the model on the test track of the simulator:

Using the 'autonomous' mode option of the Udacity self-driving car open-source simulator, connected the simulator to our prediction model via the Flask API. Every frame had its corresponding output displayed as well as rendered on the simulator as a turn made of that predicted angle. In this way, the car was able to drive perfectly, without any collisions on both the train as well as the test track maintaining a mean speed of around 20 miles per hour.

h) Evaluating and plotting the results, and further improving the test metric scores.

The final results were documented involving different losses and accuracies for different combination of hyperparameters used and their corresponding performance on the test tracks were observed. Effect of speed on the test track results was also observed.

IV. MODELING AND ANALYSIS

The Self-Driving System is based on the concepts of building an artificialintelligenceby the use ofDeep Convolutional Neural Networks. The ultimate aim for the model is to learn to predict different steering angles as the output based on the training lane-images asinput. The training is being done for various scenarios by collecting the data in the form of images from the Udacity Simulator. The model details are described below:

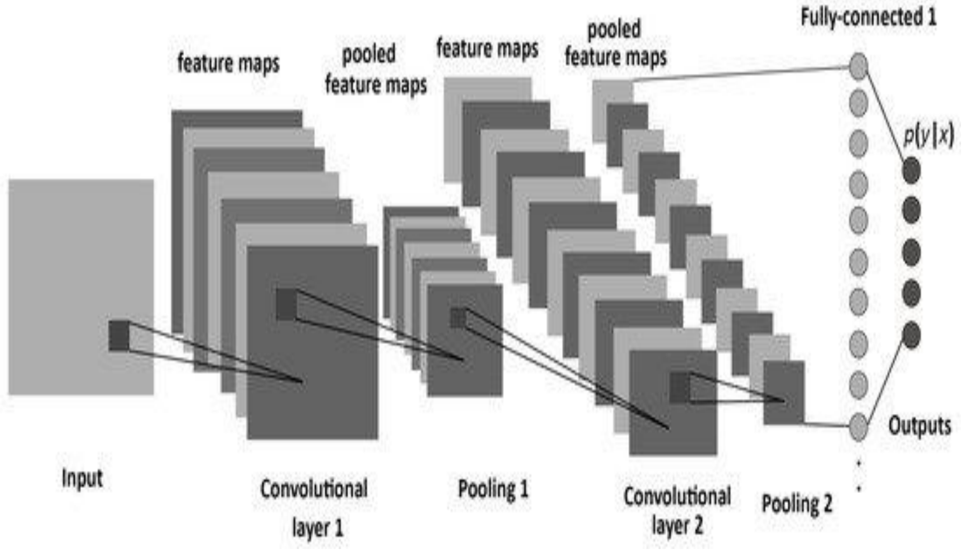
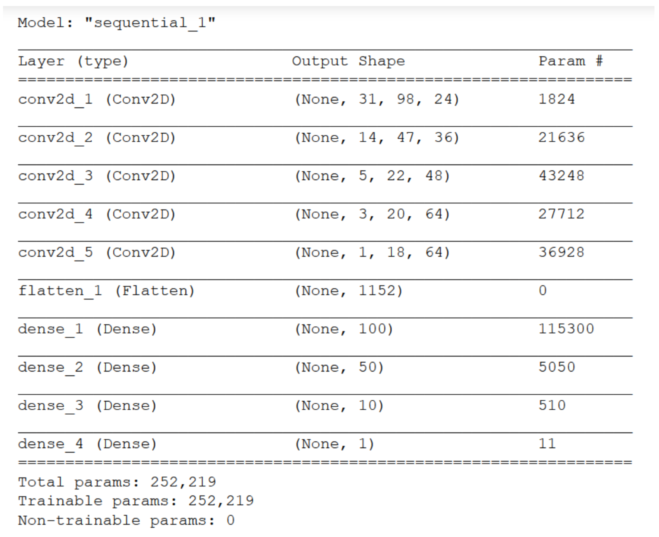


Figure-4: General Architecture of a CNN



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 31, 98, 24)        1824
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 47, 36)        21636
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 5, 22, 48)         43248
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 20, 64)         27712
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 18, 64)         36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               115300
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 11
=================================================================
Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0
```

Figure-5: CNN Model Details

The figure above describes the architecture of proposed CNN used for training the model. It contains 5 convolutional layers which are then fed to a network of dense layers by flattening them into a 1-D array using the ‘flatten_1’ layer. As seen, the final output contains only 1 value, which is our required value for ‘steering angle’. The ‘Elu’ activation function is being used as it tend to converge to zero faster and produce more accurate outputs.

Furthermore, the optimizer being used to update weights is Adam, which gave better results upon experimentation.

The batch size is kept to 100 i.e. - 100 images are being used to train over an epoch/iteration of cycle.

The general flow of steps for the system is shown by the following figure:

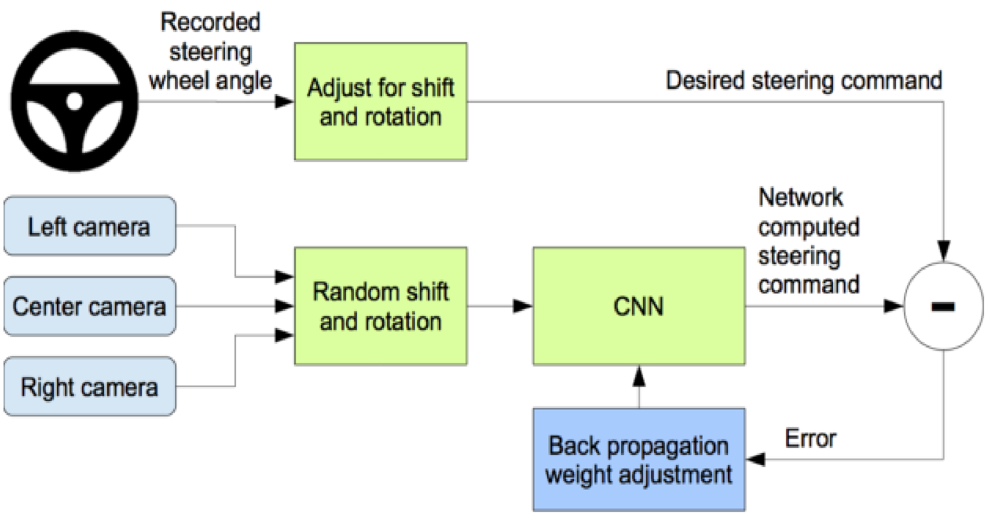

Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
-
Back propagation weight adjustment
Error

Figure-6: Training the network.

V. RESULTS AND DISCUSSION

Upon various experimentation, analysis and setting different parameters for the model, we compare different models and hyperparameters for the model (Table 1.). The best possible model which is ‘Model-I’ is the case with the least training (T.L. - 0.1667) and validation loss (V.L. - 0.1382). Hence, the model is generalized and open to new circumstances as well as suited good enough to perform on the existing environment. Further, since the training has been done only one track in the simulator. The testing is done on a uniquely different track the model has never encountered before. The Autonomous vehicle performs well on both the tracks and thus the objective of the project is fulfilled. The graph denoting the variance of training and validation losses with epochs is given in Figure-7.

Table 1. Comparison of different hyper parameters and losses

S No	Model Name	Epochs	No. of CNN Layers	No. of Dense Layers	Activation Function	Optimizer	Batch Size	Training Loss	Validation Loss
1	Model-A	5	2	1	ReLU	SGD	30	0.3599	0.2978
2	Model-B	5	3	2	ReLU	Adam	50	0.2927	0.2537
3	Model-C	10	4	3	ReLU	Adam	100	0.2349	0.1967
4	Model-D	5	2	1	Elu	SGD	30	0.3045	0.2880
5	Model-E	5	3	2	Elu	SGD	30	0.2998	0.2040
6	Model-F	5	3	3	Elu	SGD	50	0.2946	0.1989
7	Model-G	5	4	3	Elu	Adam	50	0.2045	0.1823
8	Model-H	10	4	3	Elu	Adam	50	0.1891	0.1692
9	Model-I	10	5	4	Elu	Adam	100	0.1667	0.1382

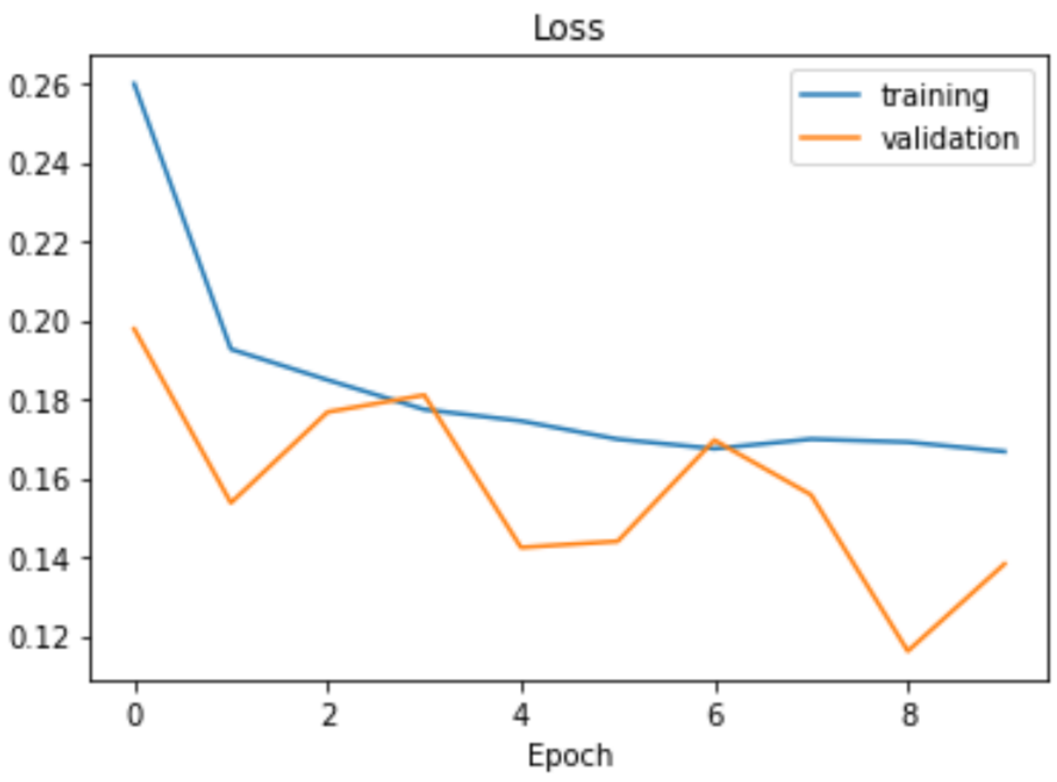
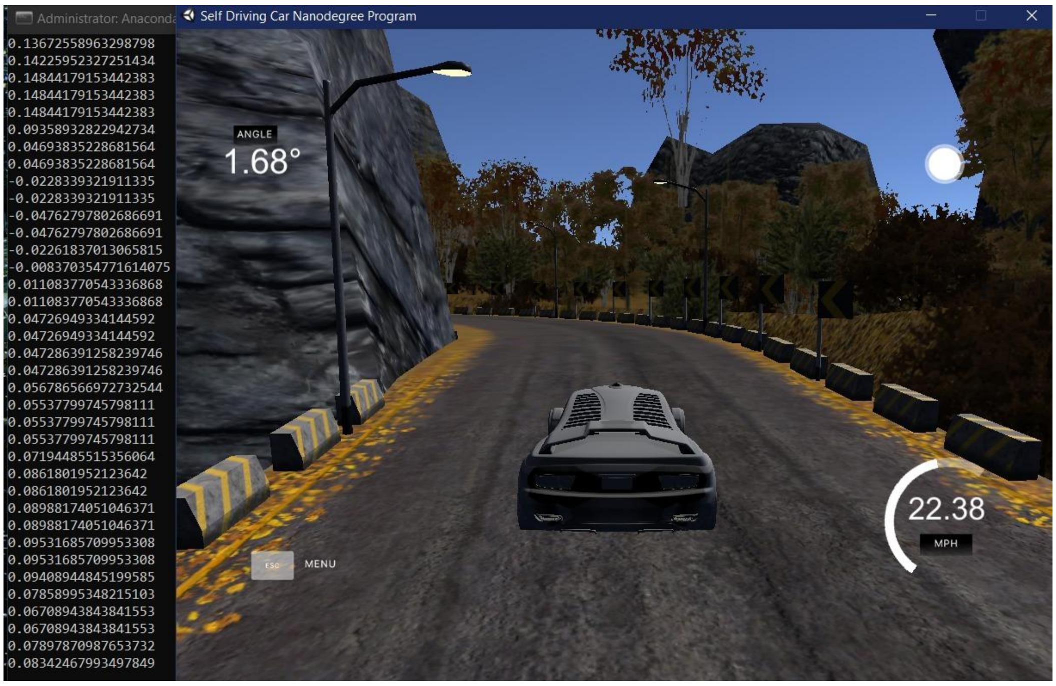


Figure-7: Training and validation loss for Model-I

The translation of model output on the simulator is depicted in the image below. It shows a standstill from the output in Autonomous mode: Test Track (right), corresponding real-time steering angle predictions (left).



```
0.13672558963298798
0.14225952327251434
0.14844179153442383
0.14844179153442383
0.14844179153442383
0.09358932822942734
0.04693835228681564
0.04693835228681564
-0.0228339321911335
-0.0228339321911335
-0.04762797802686691
-0.04762797802686691
-0.02261837013065815
-0.008370354771614075
0.011083770543336868
0.011083770543336868
0.04726949334144592
0.04726949334144592
0.047286391258239746
0.047286391258239746
0.056786566972732544
0.05537799745798111
0.05537799745798111
0.05537799745798111
0.07194485515356064
0.0861801952123642
0.0861801952123642
0.08988174051046371
0.08988174051046371
0.09531685709953308
0.09531685709953308
0.09408944845199585
0.07858995348215103
0.06708943843841553
0.06708943843841553
0.07897870987653732
0.08342467993497849
```

Figure 8: Output in Autonomous Mode.

VI. CONCLUSION

This paper proposes a CNN-based modelfor achieving safe driving autonomously. The detailed architecture of the CNN is presented in Section IV. The structure of the complete training, validation, and test data are described. The computer vision concepts and algorithms used for image processing have been described as well and their effects are analyzed. It has been shown that our model is able to learn the wholeprocedure of lane and road following without manual markings or human intervention.A small amount of labelled training data images from training on one or two tracks was sufficient to train the model to drive safely in multiple tracks. Avery sparse training signal (steering alone)is enough for the CNN to learn meaningful road features. It has been established that the data quality (much more than quantity) is primarily of significant importance for this application. Hence, a comprehensivedata pre-processing pipeline of training has been carefully implemented. Moreover, the proposed course of improvement actions for betterment of the work has been elaborated next. The presented model presents a centerpiece in facilitating the existence of fully autonomous cars in the near future.

Suggested Improvements:

- The whole purpose of the model proposed is to be finally taken into test on-road by implementation of a suitable API to plug-in the tested intelligence into a hardware machine. So, appropriate hardware can be designed to supplement the needs of putting the AI into action. Rigorous testing and performance improvements on various simulators will ensure good result when tested in the real world.
- Features such as traffic signal recognition, which also uses computer vison can be merged with this model, to make the system even more realistic.
- Variables such as aerodynamic features, road friction, inertia of the car and terrain can also be accounted for, which would make the real-life implementation of the model more feasible.
- Predicting more parameters such as acceleration/deceleration and torque in addition to steering angle, will add an extra layer of autonomy to the system.

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my effort with success.We are indebted to our Guide, Mr. Raghav S., Asst. Professor, Department of Information Science and Engineering, Sir MVIT, Bangalore, who has not only coordinated our work but also given suggestions from time to time.We are greatly thankful to Dr. P.Vijayakarthik, Prof and HOD, Department of Information Science and Engineering, Sir MVIT, Bengaluru, for his able guidance, regular source of encouragement and assistance throughout this project.We would like to express our immense gratitude to Dr. V.R. Manjunath, Principal In charge, Sir MVIT, Bengaluru, for providing us with excellent infrastructure to complete our project work.We gratefully acknowledge the help lent out to us by all faculty members of the Department of Information Science and Engineering, Sir MVIT, Bengaluru, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided during the course of the project. Furthermore, we acknowledge the support and feedback of my parents and friends.

VII. REFERENCES

[1] Rassõlkin, T. Vaimann, A. Kallaste, R. Sell, "Propulsion Motor Drive Topology Selection for Further Development of ISEAUTO Self-Driving Car", Power and Electrical Engineering of Riga Technical University (RTUCON) 2018 IEEE 59th International Scientific Conference on, pp. 1-5, 2018.

[2] Qudsia Memon, Muzamil Ahmed, Shahzeb Ali, Azam Rafique Memon and Wjiha Shah "Self-driving and driver relaxing vehicle", 2016.

[3] Saumya Kumaar, Navneethkrishnan B, Sinchana Hegde, Pragadesh Raja and Ravi M. Vishwanath "Towards Behavioural Cloning for Autonomous Driving", 2019 Third IEE International Conference on Robotic Computing (IRC).

[4] Sunil Kumar Vishwakarma, Akash, Divakar Singh Yadav "Analysis of Lane Detection Techniques using OpenCV, IEEE INDICON 2015.

[5] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5079–5087.

[6] K. L. Campbell, "The shrp 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety," TR News, no. 282, 2012.

[7] WaelFarag, Zakaria Saleh, "Traffic Signs Identification by Deep. Learning for Autonomous Driving", Smart Cities Symposium (SCS'18),Bahrain, 22-23 April 2018.

[8] Udacity Sample Training Data,https://d17h27t6h515a5.cloudfront.net/topher/2016/December/584f6edddata/data. zip.

[9] Léon Bottou, "Online Algorithms and Stochastic Approximations",Online Learning and Neural Nets, Cambridge Univ. Press, ISBN 978-0-521-65263-6, (1998).

[10] Udacity Simulator, https://github.com/udacity/self-driving-car-sim

[11] Keras Documentation, "https://keras.io/".

[12] TensorFlow, "https://www.tensorflow.org/".

[13] Python, "https://www.python.org/"