Journal of Science and Technology
ISSN: 2456-5660 Volume 06,Special Issue 01, August 2021,

www.jst.org.in DOI: https://doi.org/10.46243/jst.2021.v6.i04.pp365-370

Decision System for a Self-Driven Car

Shardul Lokhande, Md Shehroz Khan , Gaurav Pandey

Dept. of Information Technology,
NBN Sinhgad School Of Engineering, Ambegaon

Shardul Lokhande : shardulokhande@gmail.com
Md Shehroz Khan : mdshehrozkhan@gmail.com
Gaurav Pandey: pandeygaurav850@gmail.com

To Cite this Article

Shardul Lokhande, Md Shehroz Khan , Gaurav Pandey,"Decision System for a Self-Driven Car", Journal of Science and Technology, Vol. 06, Special Issue 01, August 2021, pp365-370: .

Article Info
Received: 15.07.2021 Revised: 24.07.2021 Accepted:10.08.2021 Published: 16.08.2021

Abstract : *As the development in the sensor technology and as the accuracy of those sensors keep on upgrading, the perception of the surrounding environments and operational intricacy of the connected automated vehicles also enhances. In this paper, we propose an implementation of a decision-making system which is being developed for the control of autonomous vehicles. It focuses on the efficient usage of neural networks to realize the classification of traffic signs, detection of vehicles and the ultimate aim of a self-driven car that is to drive over any path without any human intervention. Unity based Udacity simulator can be modified to make variety of paths available to train and test the built autonomous car.*

IndexTerms - *Autonomous, robot, unity, machine learning, artificial intelligence.*

I. INTRODUCTION

Looking at the very recent studies and some developing technologies we can say that there is an assured advancement in the development of autonomous vehicles. The features and characteristics provided by both computing technologies and cutting edge communication along with the emergence of high-end cars offer the foundation of smart vehicles. The ultimate aim of our study is to develop a decision system for an autonomous vehicle which can operate safely with minimal human intervention. An autonomous vehicle, which is able to perceive its environment using sensors like cameras and radar and take a most correct decision. A tool in this area of research will be a connectivity and automation that combines internal sensors with external communications. Although, the decision system working will be demonstrated using a simulation.

Deep Learning is getting more and more enhanced in the research including computer vision. More and more research based on Convolutional Neural Networks(CNN) is proving to be significant. The efficiency of convolutional networks is one of the main reasons why the world has woken up to the efficiency of deep learning. The navigation of car and sensing of environment is carried out with minimal human intervention and the vehicle is generally equipped with an Inertial Navigation System and a GPS segment and sensors such a s remote radars, lidar laser sensors, optical cameras etc. In our paper, we use a GPU computing platform for realization and perception of the surrounding environment. Use of Unity platform in the design of road signs and signals and a perfect car environment so that the decision system is able to learn and train is being done.

II. RELATED WORK

We use unity platform for simulation of our autonomous vehicle in a well designed environment of cars and traffic signals. Unity is a cross-platform game engine developed by Unity technologies, which is used to develop video games for different platforms like desktop, consoles, mobile devices, etc. We have used OpenCV and various algorithms for lane detection, classifying traffic signals, controlling the trajectory of the vehicle, etc.

2.1 Lane Detection

For our lane detection module we are using OpenCV library of python. First and foremost the image is converted into the grayscale, then we apply Gaussian Blur[2] to the image. We use Gaussian Blur to smooth the image and reduce noise in the image, then we use the canny edge algorithm to detect and adjust the lower as well as higher threshold values. After the values are set we use an image mask to select the area of our interest. Then we apply Hough Transform[3] technique, to draw lines in the edges of the image. The parameters used in Hough Transform is not easy to adjust.

Table no 2.1 : Parameter Testing

Parameter	Value
Learning rate	0.001
Number of epochs	2000
Batch Size	2048
Training Time	10mins

There are four parameters to Hough Transform which are the binary image we have obtained by applying the canny algorithm, the second and the third arguments define the resolutions or accuracies of the Hough accumulator. The fourth argument is the threshold which means the minimum number of votes it should get to be accepted it as the line.

Figure 2.1 : Traffic Lane Detection

For obtaining a single line to the left and right lanes, we set the minimum length of the line and maximum gap between the pixels to obtain a single straight line in the image. After detecting the lane lines in the image we then apply a video to same technique and then detect the lane lines from the video.

3.2 Traffic Sign Classification and Detection

For this module we used convolutional neural network for classifying different traffic signal. Here we are using the German Traffic Sign dataset that consist about 50,000 images of 32x32 resolution each. Each image belongs to a traffic sign from one of the 43 classes. We first preprocess the image by converting each image to grayscale image. The dataset used is unbalanced and is some classes are represented to a lower extent than others. Here we have used deep neural network classifier module, it has 4 layers:- 3 convolutional layers for feature extraction and one fully connected layer as a classifier. After that we train our model t predict the

Journal of Science and Technology
ISSN: 2456-5660 Volume 06,Special Issue 01, August 2021,

www.jst.org.in DOI: https://doi.org/10.46243/jst.2021.v6.i04.pp365-370

signs which we will provide them while the testing phase. In testing phase more than 20000 images we feed to the model to predict the signs and we achieved an accuracy of about 97.24%.

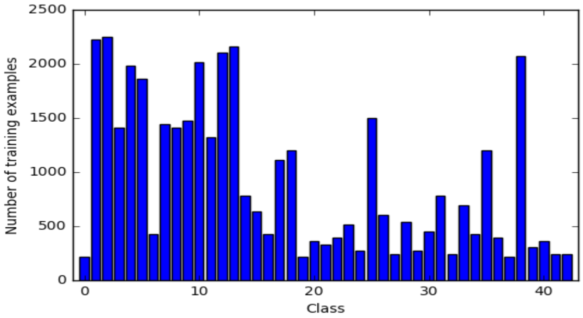
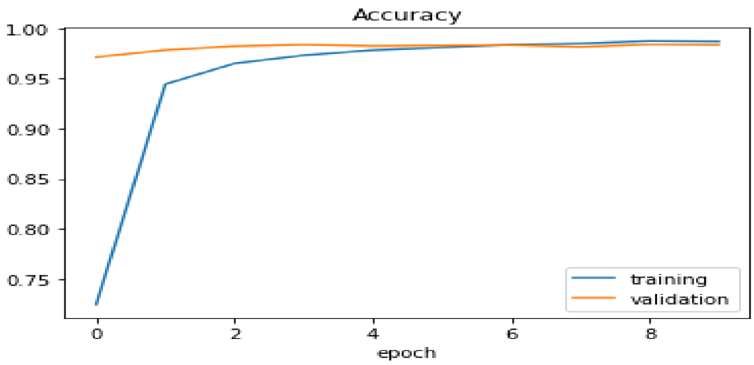


Figure 2.2: Dataset Classes Distribution




```
[27]

score = model.evaluate(X_test, y_test, verbose = 0)

print('Test Score : ', score[0])
print('Test Accuracy : ', score[1])

Test Score :  0.2579288363235143
Test Accuracy :  0.9724465608596802
```

Figure 2.3: Traffic Sign Classification Accuracy

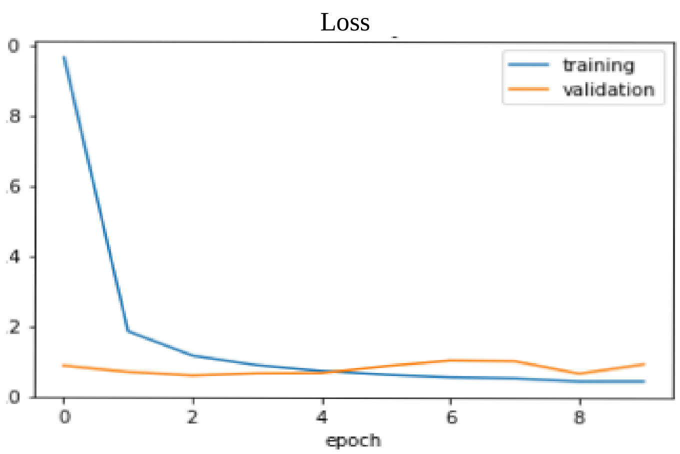


Figure 2.4: Reduction of loss

3.2 Vehicle Control

We used Behavioral Cloning, which is a technique by which the human skills can be captured and transformed in a computer program. The program results a skilled behavior which automatically controls systems for complex tasks such as steering control and lane detection. Here we used Udacity simulator for collecting our data, it has two modes, the training and the autonomous mode, and two tracks. training mode is used for logging the data to learn the behavior of driving[4]. The results are effective steering control. For logging the data we drove the car on track 1 by keeping the car to the center of the lane and for 3 laps. For collecting the data we followed the strategies such as :

1. Drove on track 1 for 3 laps by keeping the car at center of lanes.
2. Drove the car in opposite direction of lane for 1 lap so that the car doesn't become bias specifically towards either left turn only or right turn.
3. For multiple times drove the car to left of lane and center of the lane (tells the network if cars goes out of centre lane then how to come back to center lane), same for right lane to center lane.
4. Smooth turns data at each turn.

After the collection of data from the above strategies we had approximately 20000 images with steering angles as positive negative and 0 for right left and straight respectively, but most of the image had zero steering angles(car driving straightly). After this we removed most of the images with zero steering angle so that the car doesn't become biased for driving straight always and thus we were left with about 5000 images. The data was divided into training and testing set with standard splitting 20% data for testing and rest 80% for training. The data left after eliminating most of the zero angles was very less for training purpose; so, to increase the effectiveness of the model we used augmentation techniques for training data strictly such as zooming, panning and flipping.

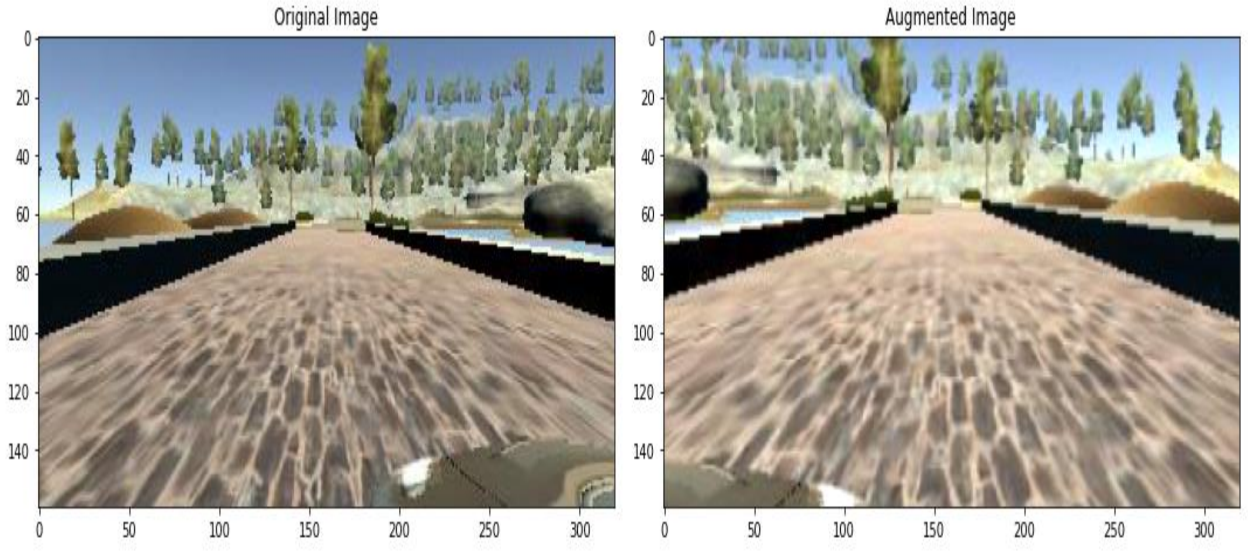


Figure 2.5: Augmentation (mirroring effect of image)

Mirroring creates data by flipping the images such that the selected image has its exact mirrored copy and the corresponding steering angle sign is also changed for every mirrored picture. Only few images should be augmented or model may turn out being over fit. Every preprocessed image has been converted to YUV format along with resolution of 200 x 66 as it suited most prominently to the training model used.

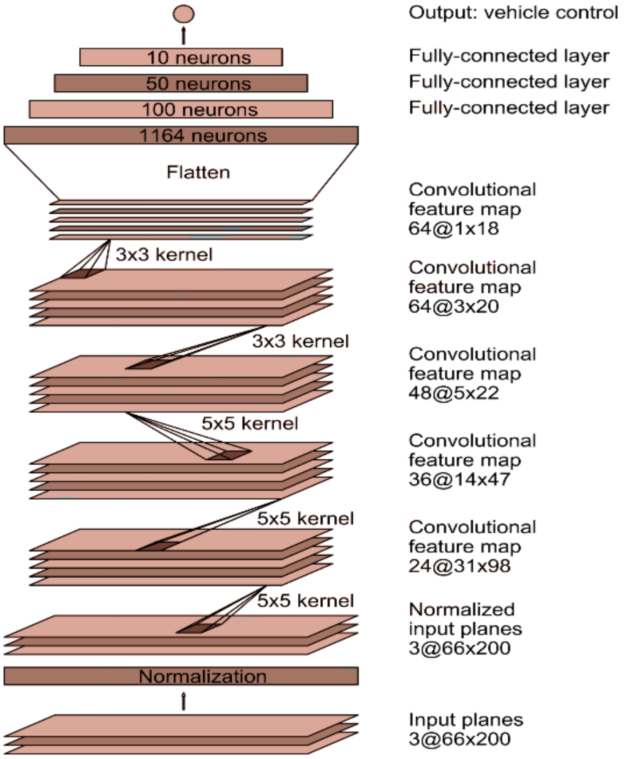


Fig: Nvidia model

Convolutional Neural Network has been used for training the autonomous car to guess accurate steering angle to drive along the road. CNN has widely helped to boost the process of pattern recognition. Nvidia has given once such training network that has 9 layers [4]. Weights of the network are adjusted to minimize the mean square error between the output given by neural network and validation data used .Network consists of 9 layers including a normalization layer, 5 convolutional layers followed by 3 fully connected layers. For the three convolutional layers among five, 5x5 kernel performs 2x2 stride then for further two layers, 3x3 kernel performs non-stride convolution. eLU activation function showed better results (decreases loss much significantly) in the model generation phase as compared to ReLU so eLU function was used because it tends to smooths slowly as compared to ReLU and produced more accurate results. Finally the model was trained with batch size of epochs as 300 and number of epochs set to 10 as supported by our GPU. The model showed accuracy of 99.32% and tested the vehicle on completely new track provided by Udacity simulator.

III. CONCLUSION

The increased glory of driver-less vehicles and the perception and planning systems require verifiability, safety and explain ability to allow the transition from systems suitable for showcases toward production-ready autonomous vehicles. Now-a-days most of the researchers are working in the field of autonomous vehicles, specially on the accuracy and safety of the driving vehicle. We know that autonomous systems that operate in complex, dynamic, and interactive environments require to predict and act according to the situations that may be unforeseen too. In this research, we have used various algorithm for lane detection, traffic sign classification and also vehicle control. It proved to be very helpful and reliable for the accuracy of the vehicle. Our study was mainly carried out on a GPU computing platform for realization and perception of the surrounding environment. A simulated environment is provided to the decision system designed for learning and training purpose and good results were achieved. We have also carried out runs of the vehicle on some different simulated environments and have achieved good results there too. The decision system was therefore successful on a completely different track.

REFERENCES

[1] D. Dong, X. Li and X. Sun, "A Vision-Based Method for Improving the Safety of Self-Driving," 2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS), Shanghai, China, 2018, pp. 167-171.
[2] Haar Romeny T B B, Florack L L, Swart M, et al.(2003). Deblurring Gaussian blur[M]// Front-End Vision and Multi-Scale Image Analysis. Springer Netherlands, 66-80.

[3] Kultanen P, Xu L, Oja E.(1990). Randomized Hough transform (RHT)[C]// International Conference on Pattern Recognition, Proceedings. IEEE, 2002:631-635 vol.1.

[4] Mariusz Bojarski , Davide Del Testa, Daniel Dworakowski , Bernhard Firner , Beat Flepp , Prasoon Goyal et al.(2016). End to End Learning for Self-Driving Cars. NVIDIA paper.