International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Self-Driving Cars: Automation Testing Using Udacity Simulator

Shahzeb Ali[1]

AEE

Dept. Of Automotive Electronics Eng., Coventry University, Coventry, UK

---***---

Abstract - *Daily new and innovative cars with top features are being launched around the globe. But, the mention of self-driven cars still garners a lot of attention. The work on self-driven cars is going on for quite some time, but they are still not fully prepared to be launched in the market. One of the reasons can be, their inability to understand the common day-to-day etiquettes followed by the society on the road. Hence, this paper proposed a methodology to perform automated testing for Self-Driving Vehicles using Udacity Simulator*

Key Words: Machine Learning, Self-Driving, Behavioral Cloning, Simulator

1. INTRODUCTION

We live in an exciting period where each day more and more technology developments are under process and automation is one of the blessings of it. Automation is a way or method to deliver the output with minimal intervention from humans and in some case, there is no human intervention. These automations provide an accuracy level which is quite difficult to achieve manually and are highly beneficial in fields such as automotive industry, Aerospace, and military where the possibility to have an error in calculation or output is negligible.

Machine Learning has also grown widely in the field of Automotive Engineering and many vehicles today are driven based on the algorithms studied through machine learning. With the adequate help from machine learning we can develop models that can help the vehicle to take the decision automatically and perform the task with great precession. One of the tasks which is performed using this learning is developing a model for self-driving car and test it on a totally new track and environment for autonomous testing.

Following few interesting technologies provide a clear picture of the methodologies currently available in the market for simulating and testing the Self-Driving vehicles.

A deep learning algorithm which user virtual environment for self-driving vehicle [2]. The data is collected using an end to end method and Nvidia autonomous driving techniques.

After the successful collection of training data, AlexNet Convolutional Neural Network a pattern recognition method is used for the training. Once successfully trained on this virtual environment data the toy car is used for showing how the model performs in the real-world scenario.

Another method followed for implementing and testing self-driving cars is a simulator and algorithm based automated testing, it assesses the performance of the self-driving car using simulator. The simulation system will create different randomly, manually modified scenarios which are generated based on an algorithm. The genetic algorithm will be used for augmenting both manual and random scenes to identify the failures across the system.[4].

Developing a model which maps raw images can also be used. The focus is to develop a model that can map different raw images captured to perform training and testing to a correct steering angle using the deep learning algorithm. The data is collected using a vehicle platform built using 1/10 scale RC Car, Raspberry pi 3 Model B computer and Front facing camera [5].

Prototype based approach for testing Self-Driving cars includes creating a self-driving car prototype (using simulation) that uses cameras for navigation and generates a better output. For prototype purpose a 3D virtual city is designed which depicts a real environment with traffic cars, traffic signals and different type of obstacles. The sole target of our self-driving vehicle should navigate easily without violating any traffic regulations and hitting any unwanted obstacles. It should be quick efficient and comfortable so that fuel consumption is reduced, and minimum jerks are felt throughout the journey. [1]

In this paper, we are using the Udacity Simulator for training and the testing of the data. Udacity Simulator provides us with two different set of modes i.e. Training Mode and Autonomous Mode. The user drives the vehicle manually using the keyboard keys for generation of data i.e. images are captured using three different cameras attached on the front, left and right side of the vehicle and also the steering angle is captured for different turns and corners.

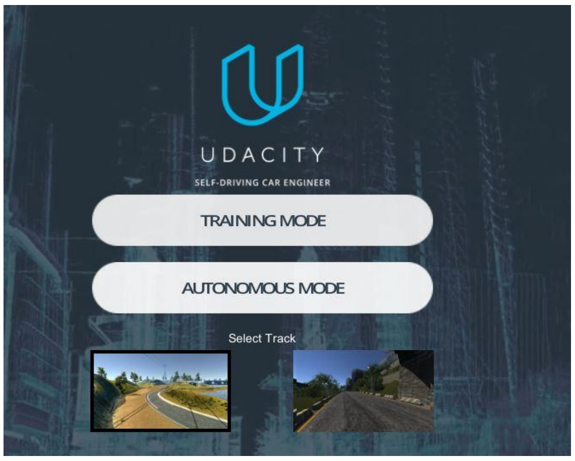


Figure -1: Udacity Simulator

Once the data is collected, it is processed and trained using a deep neural network and provides a trained model which is later used in the autonomous mode, where user uses Server-client mechanism to run the vehicle in autonomous mode.

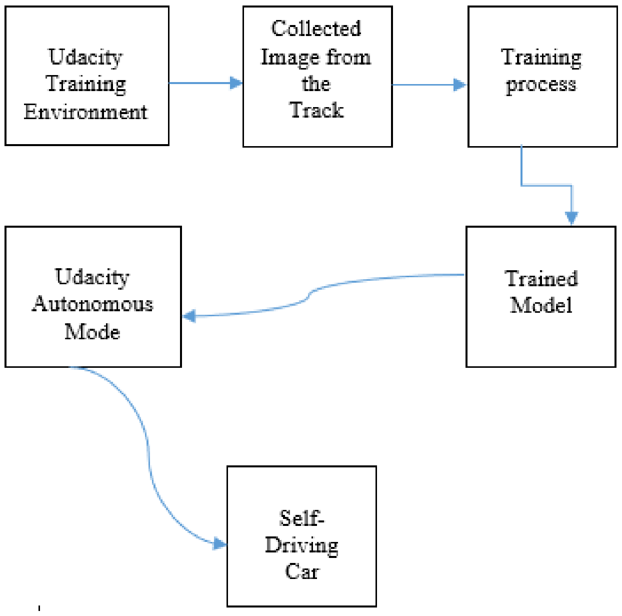


Figure -2: Block Diagram: Self Driving Cars

The main outcomes we can achieve from this paper are:

- Providing a method to shift the manual testing of vehicles to a safer and accurate automated method of testing.
- The image processing and training techniques helps in providing a high accuracy for the vehicle to be tested in an environment which is totally new and different from the track on which it is trained on.

2. DATA COLLECTION: TRAINING MODE

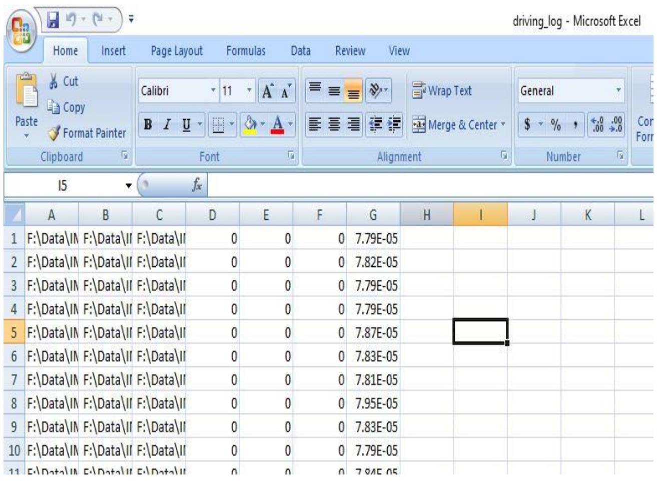
To perform the Training on a neural network the data needs to be collected for driving the vehicle on one of the tracks provided in the Udacity simulator.

The images from the three different cameras which are connected to the vehicle on the front hood, left mirror and right mirror in the virtual environment are collected continuously as we are recording in the training mode along with the steering angle for the different turns on the road.

The images which are collected while driving are stored in the desired path which can be set at the beginning of the training process. Along with the images, there is driving log .csv file generated which consist of following different columns:

.csv Column	Data description
Column 1,2,3	These columns contain the paths of dataset images for left, right and center camera images respectively.
Column 4	It contains the steering angle value, if set to 0, it means the vehicle is driven in straight direction
Column 5	It contains the throttle value
Column 6	It contains the deceleration value
Column 7	It contains the speed of the vehicle.

Figure -3: .csv file column Description



	A	B	C	D	E	F	G	H	I	J	K	L
1	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.79E-05					
2	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.82E-05					
3	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.79E-05					
4	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.79E-05					
5	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.87E-05					
6	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.83E-05					
7	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.81E-05					
8	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.95E-05					
9	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.83E-05					
10	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.79E-05					
11	F:\Data\IN	F:\Data\II	F:\Data\II	0	0	0	7.84E-05					

Figure -4: Driving Log.csv File

The vehicle is driven on the track using the following keys of the Keyboard i.e.

a. Upper-Arrow: Controls the straight direction

b. Lower-Arrow: Reverse Direction

c. Left-Arrow: Left Turn

d. Right Arrow: Right Turn

The following image depicts the different images which are captured for training of the vehicle in the virtual environment.

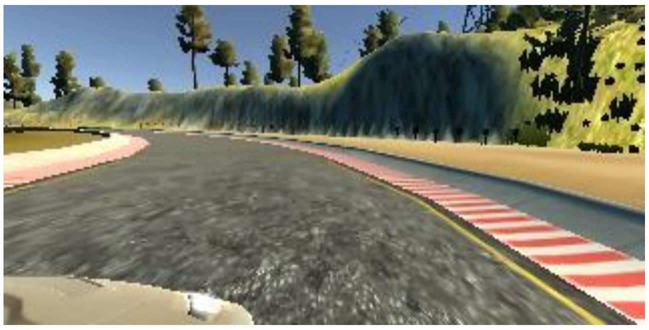


Figure -5: Image captured from Right Side Camera

Figure -6: Image captured from Left Side Camera

Figure -7: Image captured from Center Camera

3. IMAGE PROCESSING AND DATA AUGMENTATION

Once the data is collected successfully, the next step involves in training of data is the process of performing Image Augmentation and image processing. The images collected from recording of the driving of vehicle are not uniform, it is highly possible that center images are more or left images are more or vice-versa, there is a high possibility that images contain various pixels which are not useful for good training of data. Hence, it is better to perform some of the image Augmentation and Image processing Techniques. [14]

Following are some of the Image Augmentation techniques used:

- Image Zooming

  This method is used for magnifying image such that a better view of important image pixels is clear. In zooming the pixels are inserted into the image to increase the size of the image. It is made possible by pixel replication or interpolation.

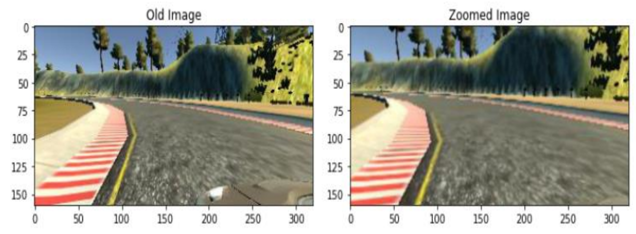


Figure -8: Image Zooming

- Image Panning

  The image panning is a augmentation technique which involves horizontal repositioning of the image. It helps in keeping the main object or subject in focus and blurring less essential background.

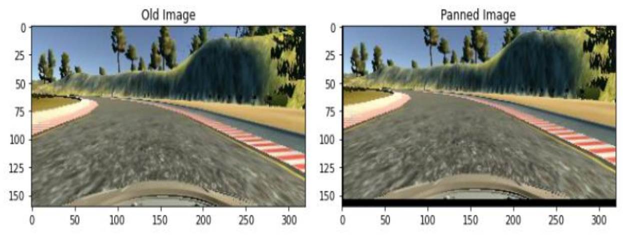


Figure -9: Image Panning

- Image Random Flip

  This technique is used to flip the images horizontally or vertically depending on the values specified. It is one of the methods used for uniformity.

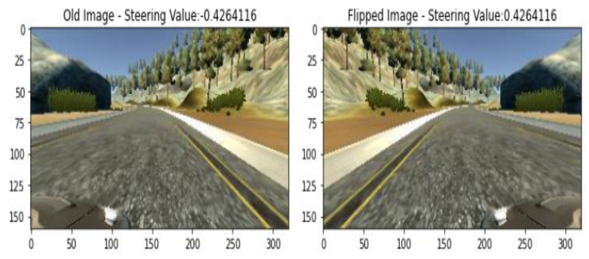

Old Image - Steering Value:-0.4264116
Flipped Image - Steering Value:0.4264116

Figure -10: Random Flipping

- Image Brightness

  The image brightness techniques are used to increase the intensity of the image acquired from the training dataset and making it clear for better training using CNN.

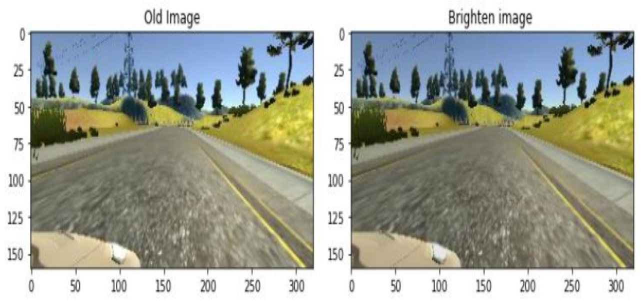


Figure -11: Image Zooming

Following are some of the Image Processing techniques used:

- Image Resizing
  The image resizing technique is used for varying the height and width of the image such that images are uniform and the main object required for training is clearly visible and unnecessary area is cropped.

- Image Gaussian Blur
  A noise reduction technique used for blurring the images. The images are blurred using a Gaussian function.

$$f(x) = ae\frac{-(x-b)^2}{2c^2}$$

Where,

a= height of the curve's peak
b= position of the center of the peak
c= Standard deviation
f(x)= function of x
e= Euler's number
x= Integer

- Image Color Conversion

  The images are converted from RGB to YUV, the YUV color spaces are more efficient and useful in the processing of images.

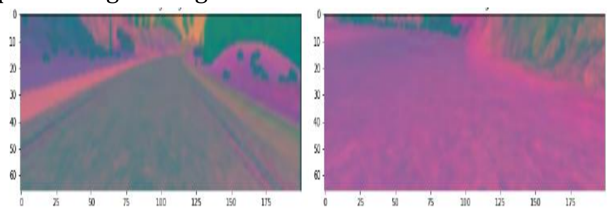


Figure -12: Image Color Conversion

4. LEARNING FROM TRAINING DATA

Once the data is successfully captured and images are processed and Augmented, next step involves the splitting of data into training set and validation set. The function train test split is used for splitting of the data. The data is split in such a way that each set contains normalized data, with somewhat equal number of images and varieties. Each set consists of following number of images:

- Training samples: 2954
- Validation samples: 1456

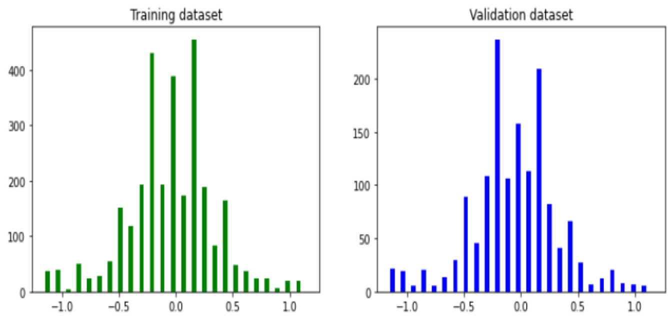


Figure -13: Graphical representation: Training set, Validation set

After splitting and processing of data, the most important step is learning from the training data using a Nvidia Convolutional Neural Network (CNN) model, this model helps in accurate mapping of the raw pixels from the images collected to the steering angle recorded in the driving log.

The CNN helps in learning the data automatically based on the images fed into the Convolutional Neural Network and model is tuned by varying the hyper parameters values and adjusting the weights.

The Nvidia network architecture used consist of 9 layers out of which 5 are Convolutional layers and 4 fully connected layers. The size of the input image is 66 x 200 x 3(Height x width x Depth).

Layer (Type)	Output shape	Parameters
Input	(66,200,3)	0
Conv1 (f=5, s=2)	(24,5,5)	1,824
Conv2 (f=5, s=2)	(36,5,5)	21,636
Conv3 (f=5, s=2)	(48,5,5)	43,248
Conv4 (f=3, s=1)	(64,3,3)	27,712
Conv5 (f=3, s=1)	(64,3,3)	36,928
Flatten	1152	0
FC6	100	115,300
FC7	50	5,050
FC8	10	510
FC9	1	11
Total		252,219

Figure -14: Network Parameters

The optimizer used in the network is 'Adam' and the learning rate is kept at $1e^{-4}$. The activation function used for this model is elu (exponential linear unit)

Activation Function	Elu
Learning Rate	$1e^{-4}$
Optimizer	Adam

Figure -15: Model Characteristic

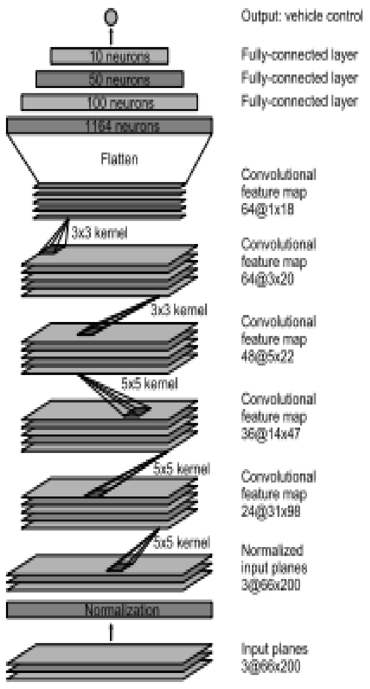


Figure -16: CNN Architecture (Mariusz Bojarsk, 2016)

5. TESTING IN AUTONOMOUS MODE

Once the model is completely tuned and has all hyper parameters set to a valid value. The model is executed for 592 epochs and each epoch consists of 400 samples. The final validation loss is approximately 0.035.

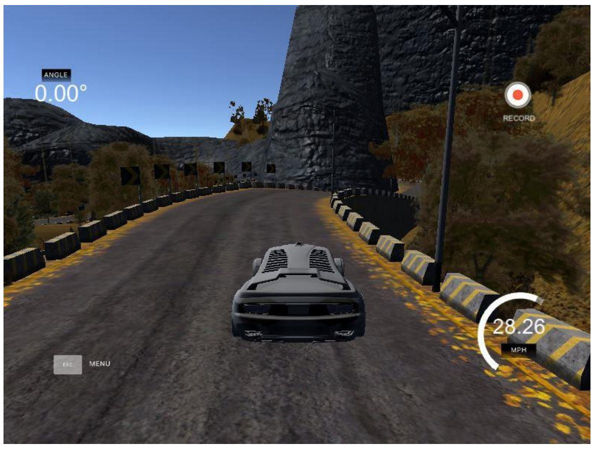


Figure -17: Testing on a new track-Image1

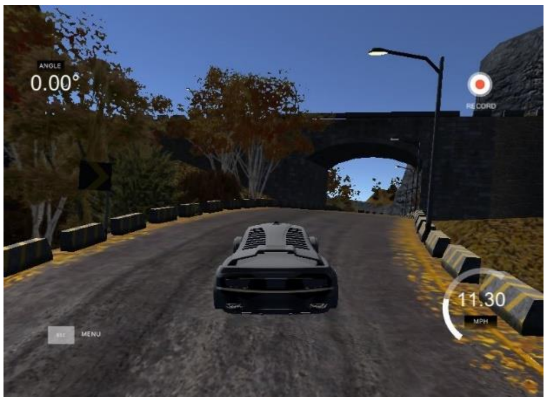


Figure -18: Testing on a new Track-Image 2

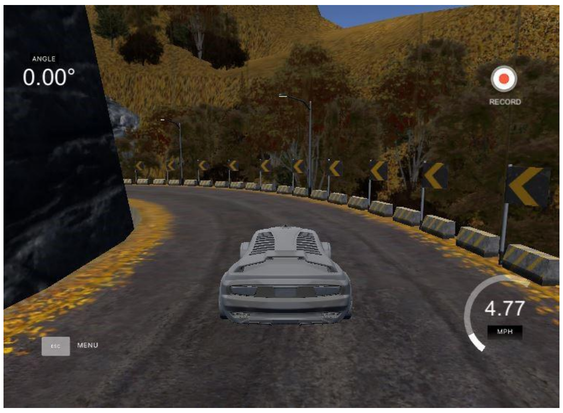


Figure -19: Testing on a new Track-Image 3

The model.h5 file is generated. H5 file is a types of file format which is used to store structure data, in our case this consist of the tuned model which will be feed back into the Udacity simulator using server-client script for driving the vehicle in autonomous mode. The vehicle is driven on the track on which it is trained and on a totally new track which is a testing track as the vehicle is totally unaware of the turns and corners of the track. The vehicle performs successfully on both the tracks with less deviation from the center of the lane and avoiding collision throughout the journey.

CONCLUSIONS

The Self-driving cars are the upcoming technology and will be the most successful in today's world with fuel prices hiking and pollution is increasing each day. The self-driving cars can help in not only cost-cutting but also reduce a lot of pollution from the surroundings as they are mostly hybrid or fully electric vehicles.

As a future work, we will be working on infusing object detecting features so that the vehicle can operate on the real-world environment with proper control and speed variation based on the objects and surroundings.

REFERENCES

[1] Bhaskar Barua, C. G. (2019). A Self-Driving Car Implementation Using Computer Vision for Detection and Navigation. IEEE.

[2] Juntae Kim, G. Y. (n.d.). Deep Learning Algorithm using virtual Environment data for Self-Driving Car. IEEE.

[3] Naoki.(2017).https://naokishibuya.medium.com/introduction-to-udacity-self-driving-car-simulator-4d78198d301d. Retrieved from https://naokishibuya.medium.com/.

[4] Straub, J. (2017). Automated Testing of a Self-Driving Vehicle System. IEEE.

[5] Truong-Dong do, M. T.-V.-H. (2018). Real-Time Self-Driving Car Navigation Using Deep Neural Network. IEEE.

[6] Mariusz Bojarsk, D. D. (2016). End to End Learning for Self-Driving Cars. arxiv.org

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation, "in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep Convolutional activation feature for generic visual recognition," in International conference on machine learning,2014, pp. 647-655.

[9] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," Computer vision Image understanding, vol. 106, no. 1, pp. 59-70, 2007.

[10] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, 2010, pp. 3485-3492.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep Convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.conference on, 2010, pp. 3485-3492.

[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep Convolutional activation feature for generic visual recognition," in International conference on machine learning, 2014, pp. 647-655.

[14] https://www.udemy.com/course/applied-deep-learningtm-the-complete-self-driving-car-course/