BEHAVIORAL CLONING

1.0 Objective

The objective of this project is to use Convolutional Neural Network (CNN) to teach a car to drive itself around a track in the simulated environment provided by Udacity.

2.0 Project Background

Some human drivers are careful and scrupulous when driving while most of the human drivers are reckless and unpredictable. Some drivers drive while they are very tired, hence, fall asleep. Some drive while intoxicated. Some send text messages or even making phone calls while driving. Such flawed behaviors while driving have caused a lot of road accidents, thereby, maimed several people and cost a lot of lives and continue to pose a great danger to human lives.

Self-driving car could reduce such fatalities or accidents due human error or recklessness. Self-driving car have sensors that are ever vigilant. They are always looking for pedestrian, they are always watching for the car in front. They don't get distracted or drunk.

Though, Self-driving car may never be perfect, but it's very likely to drive safer than human.

3.0 Introduction

In this project, we are provided with simulator that comes in two modes

Training mode – We manually drive to collect data

Autonomous mode – The vehicle drives itself based on a model trained from the collected data.

The data log is saved in a csv file and it contains the path to the road images with their corresponding steering angles, throttle, speed and etc. We are only interested in the images and their corresponding steering angles. So we are going to use images from center, left and cameras as input to neural network (Supervised Regression) and expect it predict steering angles.

We will employ the following environment and tools to implement the project;

Python, Keras, Scikit-learn, Numpy, Opencv,

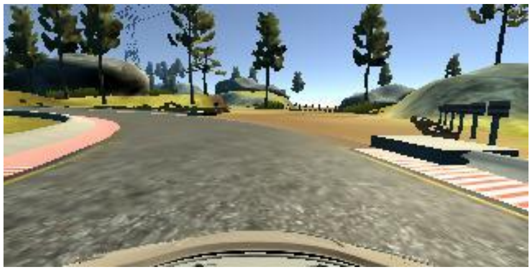
4.0 Design, Train and Test of the Model Architecture

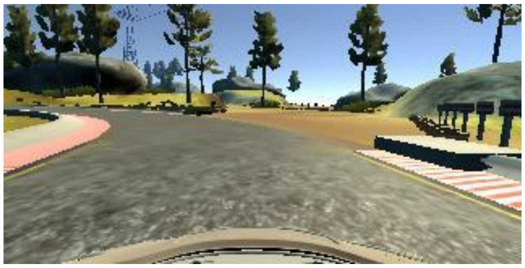
4.0.1 Datasets

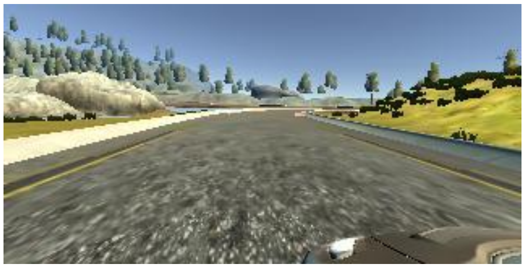
We used the sample dataset provided by udacity to train the model. We augmented the data in order to have more balanced dataset; we flipped the center images and inverted their corresponding steering angles. To add recovery data to the dataset we have taken left and right images add 0.2 and -0.2 to their steering angles respectively.

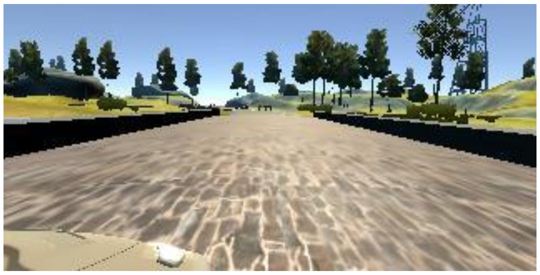
Below is the summary statistics:

Center images	6,428
Flipped Center images	6,428
Left images	6,428
Right images	6,428
Total	**25,712**







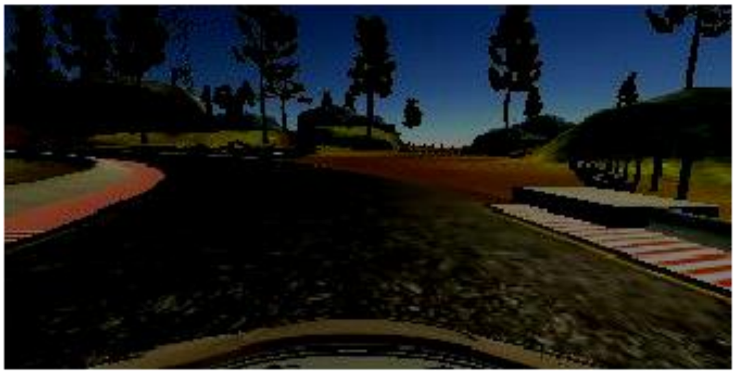


4.0.2 Preprocessing

Two preprocessing are applied to the images

- **Normalization**

  We normalize to ensure that the mean of our pixel distribution is 0. This helps our model treat the images uniformly.

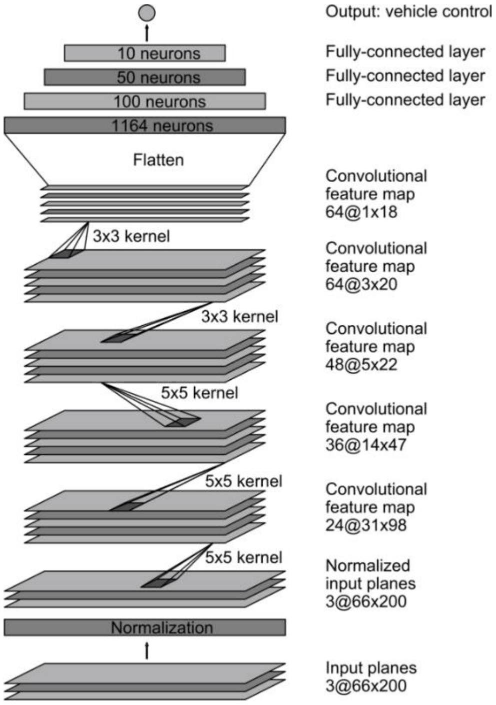


- **Cropping**

  We cropped the top and bottom of the images to remove the parts that play no role in determining the steering angle.

4.0.3 Model Architecture

The model used is based on architecture in the NVIDIA paper. We modified the model by adding dropout at first three convolutional layers and first four fully connected layer to prevent over fitting.



4.0.4 **Training and Result**

After we train our model with Adam optimizer with 1 epoch and 32 batch size. We ended up with the simulated car that drive autonomously on track 1 of the simulator.

5.0 **Conclusion**

In this project, we trained a model that enable a car drive itself in simulated environment. We started by loading the data and then preprocess the data. Finally we designed, trained and test the model.

In future, we will improve the model performance by incorporating with data augmentation as well as exploring some model architecture such as VGG, ResNet. GoogleNet and etc.