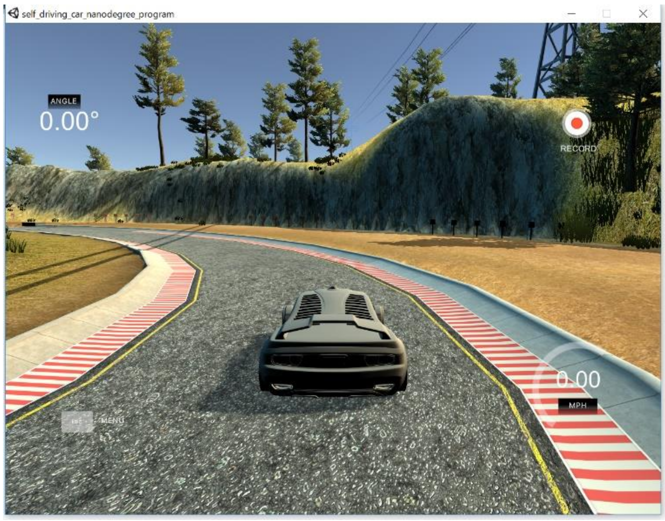
Project Name: Behavioral Cloning
Student Name: Kittipong Vadhanakom

Goals:

The goal of this project is to use the simulator to collect the data of driving behavior and build convolution network in Keras to predict the angles of steering from collected images and train, validate the model to have a car run autonomously without leaving the road.



Car Driving Simulator

Steps to provide this project:

1. **The following files are submitted for this project.**
   1.1 model.py contains the script with convolution network in Keras to train the model.
   1.2 drive.py is for driving the car in autonomous mode.
   1.3 model.h5 contains a trained convolution network model.
   1.4 writeup_report.pdf is a summarizing result and description of this project.
   1.5 video.mp4 is a captured video with autonomous driving.

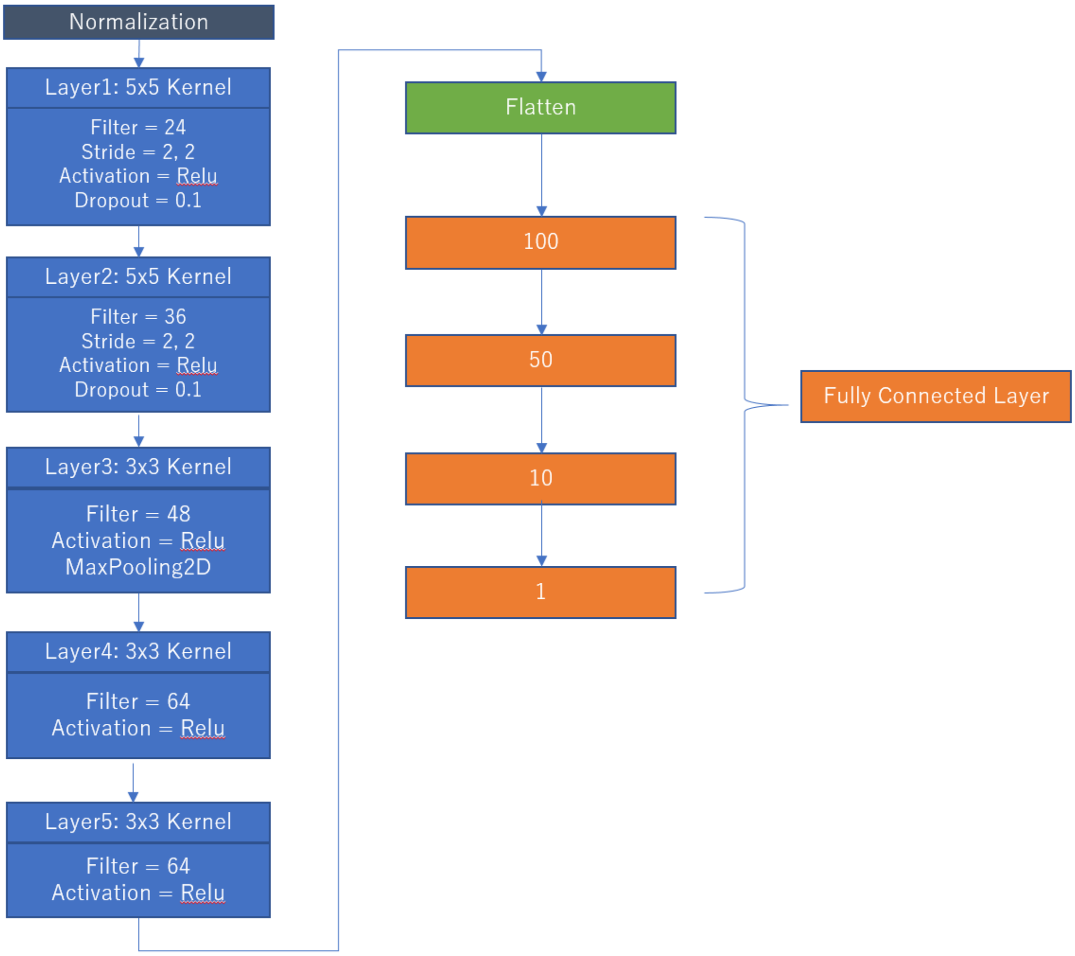
2. **Functional Code**

   Using provided simulator from Udacity and drive.py file, the car can be driven autonomously around the track by executing the following command:

Python drive.py model.h5

3. Model Architecture and Training Strategy

Model Architecture: In model.py file, it contains a model which consists of convolution neural network. This architecture is adopted from Nvidia team which I was noticed during learning in this lesson. The following is the detail of model structure:



Overfitting reduction: In this project, I used "Dropout" in the first and second layer to overcome overfitting which improve the result obviously. The vehicle drives within the road. (model.py, line 91-93)

<u>Parameter Tuning:</u> I used Adam optimizer, so the learning rate was not tuned manually (model.py line 105.)

<u>Training and validating data:</u> I have used the sample data that is provided in this lesson and combine with another two labs of driving data that I collected by myself.

- Left, right and center of image were used to train and validate.
- 20 % of training data was used for validation.

<u>Augmenting and preprocessing data:</u>

- The left and right of steering angle parameter were adjust. (Left = -0.27, Right = + 0.27)
- Flip images in horizontal to get more data to train and image with opposite steering angle. (model.py, line 20-37)
- Normalize data; using Lambda layer in Keras to normalizes on each image (model.py, line 110)
- Cropping images, to crop image helps in training data without unnecessary data.

4. Solution Design Approach

The overall strategy for deriving a model architecture was from the explanation during the lesson. The convolution architecture from NVidia team works fine.

First, I have tested with NVidia convolution network with sample data provided Udacity in this project. The car drives still out of track, but looks mostly fine.

Second, I applied dropout and the car drive stay on the lane, but has ran out at curve after passing the bridge.

Third, I collected two more labs driving data, and tested again, at this time, the car drives autonomously without leaving the road.

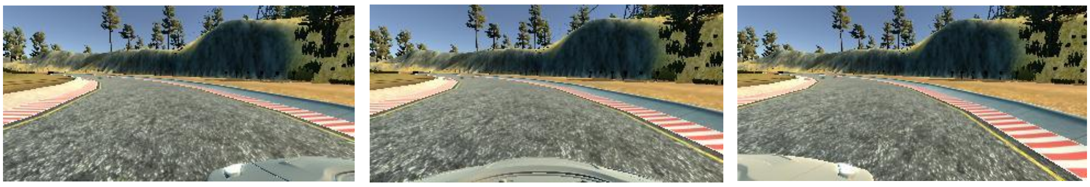
REVISED:

As per suggesting from reviewer, I have fixed the bug in file “drive.py” by adding the following code:

```
    image_array = image_array[:,:,::-1]
```

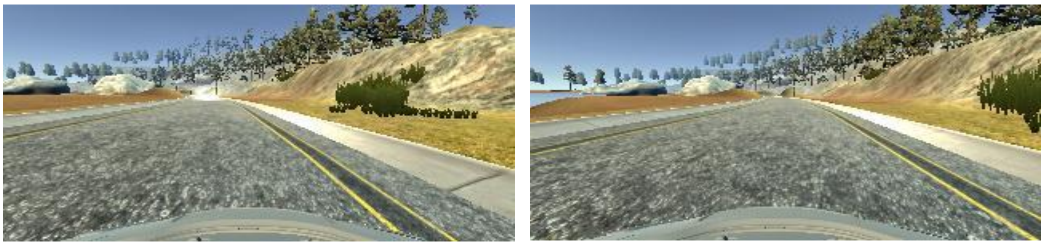
This code changes image from RGB to BGR since the code in file model.py uses cv2 to reads the image. The result was improved.

- **Here is the sample of the image of the lane from center, left and right camera.**



- **Below is sample of the image of the car recover from left and right to center, so that vehicle would learn to recover**

Right to Center



Left to Center