Report Behavior Cloning

I will list the rubric points and how I have fulfilled these points individually(below).

These files referenced are in fulfillment of the Project 4, Behavior Cloning in Udacity Self Driving Engineer.

Specifications:

- Are all required files submitted? The submission includes a model.py file, drive.py, model.h5 a writeup report and video.mp4.

The following files have been submitted model.py, drive .py model.h5 this report(writeup, “Report Behavior Cloning.pdf”), and “run1.mp4” (video.mp4), Readme.pdf

- Is the code functional? The model provided can be used to successfully operate the simulation.

Ran the car autonomously around the track in a terminal window in AWS system with the following command (run1 is the directory that the images are located from the autonomous mode):

```
python drive.py model.h5 run1
```

Then executing the simulator and selection Autonomous mode.

Then take images folder (run1) and made a video with the following command:

```
Python video.py run1
```

- Is the code usable and readable? The code in `model.py` uses a Python generator, if needed, to generate data for training rather than storing the training data in memory. The `model.py` code is clearly organized, and comments are included where needed.

The program is commented

The model.py python file contains code for training and the neural network, the neural network model saved as ‘model.h5’. The python file is well commented on how the code works.

- Has an appropriate model architecture been employed for the task? The neural network uses convolution layers with appropriate filter sizes. Layers exist to introduce nonlinearity into the model. The data is normalized in the model.

table 1 Model Architecture

Description	Output Shape	Param #
lambda_1 (Lambda)	(None, 160, 320, 3)	0
cropping2d_1 (Cropping2D)	(None, 65, 320, 3)	0
conv2d_1 (Conv2D)	(None, 31, 158, 24)	1824
dropout_1 (Dropout)	(None, 31, 158, 24)	0
activation_1 (Activation)	(None, 31, 158, 24)	0
conv2d_2 (Conv2D)	(None, 14, 77, 36)	21636
dropout_2 (Dropout)	(None, 14, 77, 36)	0
activation_2 (Activation)	(None, 14, 77, 36)	0
conv2d_3 (Conv2D)	(None, 5, 37, 48)	43248
dropout_3 (Dropout)	(None, 5, 37, 48)	0
activation_3 (Activation)	(None, 5, 37, 48)	0
conv2d_4 (Conv2D)	(None, 3, 35, 64)	27712
dropout_4 (Dropout)	(None, 3, 35, 64)	0
activation_4 (Activation)	(None, 3, 35, 64)	0
conv2d_5 (Conv2D)	(None, 1, 33, 24)	13848
dropout_5 (Dropout)	(None, 1, 33, 24)	0
activation_5 (Activation)	(None, 1, 33, 24)	0
flatten_1 (Flatten)	(None, 792)	0
dense_1 (Dense)	(None, 100)	79300
dense_2 (Dense)	(None, 50)	5050
dense_3 (Dense)	(None, 10)	510
dense_4 (Dense)	(None, 1)	11

Total params: 193,139

Trainable params: 193,139

Non-trainable params: 0

This model is the NVidia model.

- Has an attempt been made to reduce overfitting of the model? Train/validation/test splits have been used, and the model uses dropout layers or other methods to reduce overfitting.

To attack overfitting issue each convolution layer is followed by drop out layer. This improved the loss value.

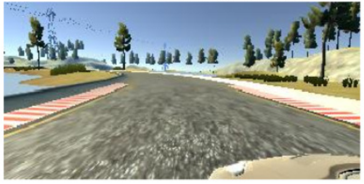
- **Have the model parameters been tuned appropriately?** Learning rate parameters are chosen with explanation, or an Adam optimizer is used.

An Adam optimizer was used.

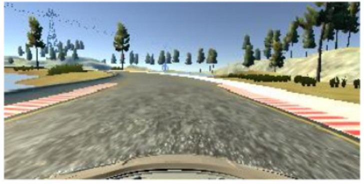
- **Is the training data chosen appropriately?** Training data has been chosen to induce the desired behavior in the simulation (i.e. keeping the car on the track).

Data was take in a CCW(counter clock wise), CW(clock wise), and CCW direction across the bridge, for a total of 37830 sample images. I added the bridge data set to improve the vehicle performance crossing the bridge. Example images below:

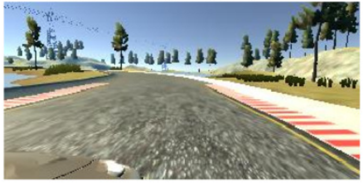
Left Cam: left_2019_03_05_16_31_31_542.jpg



Center Cam: center_2019_03_05_16_31_31_542.jpg



Right Cam: right_2019_03_05_16_31_31_542.jpg



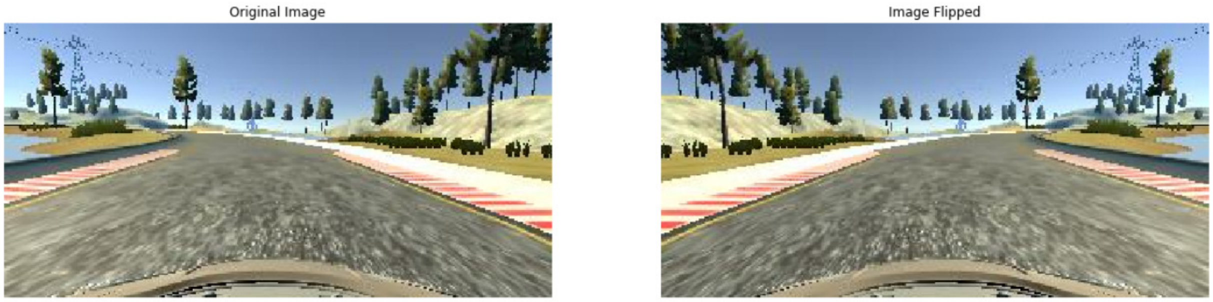
- **Is the solution design documented?** The README thoroughly discusses the approach taken for deriving and designing a model architecture fit for solving the given problem.

The images collected 37,830 these images were flipped using the following command and so was the steering measurement:

```
augment_images.append(cv2.flip(image,1))

augmented_measurements.append(measurement*-1.0)
```

Below is an example of a flipped image:



- Is the model architecture Is the creation of the training dataset and training process documented? The README provides sufficient details of the characteristics and qualities of the architecture, such as the type of model used, the number of layers, the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

Earlier in this document table 1 was presented with the model architecture.

- **Is the creation of the training dataset and training process documented?** The README provides sufficient details of the characteristics and qualities of the architecture, such as the type of model used, the number of layers, the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged. Here is one such tool for visualization.The README describes how the model was trained and what the characteristics of the dataset are. Information such as how the dataset was generated and examples of images from the dataset must be included. The README describes how the model was trained and what the characteristics of the dataset are. Information such as how the dataset was generated and examples of images from the dataset must be included.

The training set discussion was earlier

- **Is the car able to navigate correctly on test data?** No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe(if humans were in the vehicle).

The vehicle made it around the track successfully and did not leave the road surface.

run1.mp4

-