Behavioral Cloning

Files Submitted & Code Quality

1. Code Structure

My project includes the following files:

- model.py containing the script to create and train the model
- final_model.ipynb containing preprocessing of the data, data visualization and model summary
- drive.py for driving the car in autonomous mode
- model.h5 containing a trained convolution neural network
- writeup_report.md or writeup_report.pdf summarizing the results

2. Submission includes functional code

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

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

3. Submission code is usable and readable

The model.py/final_model.ipynb files both contain the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

Model Architecture and Training Strategy

1. Model Architecture

Layer (type)	Output Shape	Param #
cropping2d_1 (Cropping2D)	(None, 90, 320, 3)	0
lambda_1 (Lambda)	(None, 90, 320, 3)	0
conv2d_1 (Conv2D)	(None, 45, 160, 24)	1824
spatial_dropout2d_1 (Spatial	(None, 45, 160, 24)	0
conv2d_2 (Conv2D)	(None, 23, 80, 36)	21636
spatial_dropout2d_2 (Spatial	(None, 23, 80, 36)	0
conv2d_3 (Conv2D)	(None, 10, 38, 48)	43248
spatial_dropout2d_3 (Spatial	(None, 10, 38, 48)	0
conv2d_4 (Conv2D)	(None, 8, 36, 64)	27712
spatial_dropout2d_4 (Spatial	(None, 8, 36, 64)	0
conv2d_5 (Conv2D)	(None, 6, 34, 64)	36928
spatial_dropout2d_5 (Spatial	(None, 6, 34, 64)	0
flatten_1 (Flatten)	(None, 13056)	0
dropout_1 (Dropout)	(None, 13056)	0
dense_1 (Dense)	(None, 100)	1305700
dense_2 (Dense)	(None, 50)	5050
dense_3 (Dense)	(None, 10)	510
dropout_2 (Dropout)	(None, 10)	0
dense_4 (Dense)	(None, 1)	11

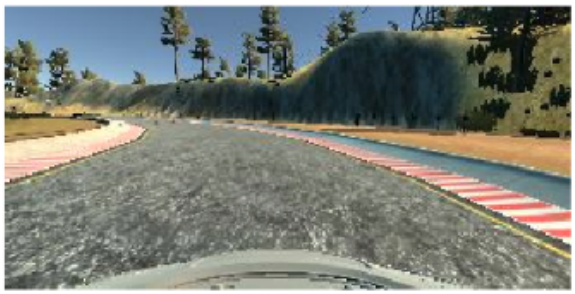
Total params: 1,442,619
Trainable params: 1,442,619
Non-trainable params: 0

Training Strategy

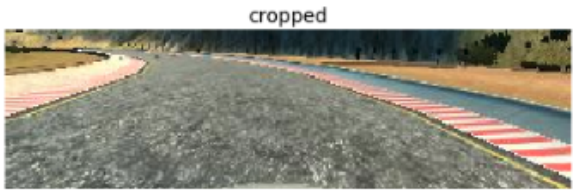
1. Data Preprocessing

The original image was cropped to remove redundant top portion (sky and other details which is not required to decide steering angle). Also the bottom of the image displaying car hood was cropped out.

raw



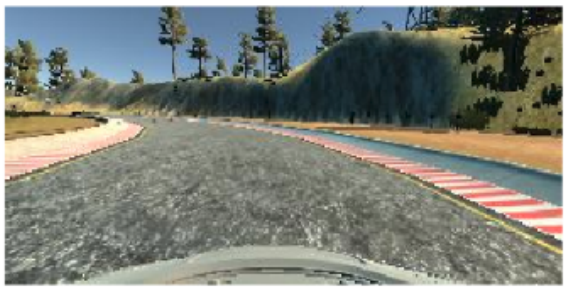
cropped



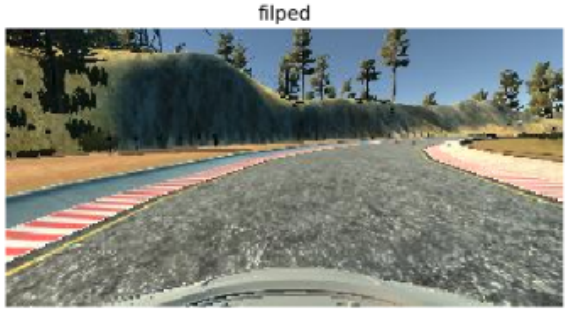
2. Data Augmentation

Mirroring the image and reversing the steering angle gives equally valid image for training.

raw



filped



The code can be found in final_model.ipynb file

What more can be done?

- Using the right and left images
- Adding more augmentation
- Trying more complex network