Eduardo Ribeiro de Campos - Udacity / SDCND - May./2019.

Self-Driving Car Engineer Nanodegree

Project4: Behavioral Cloning

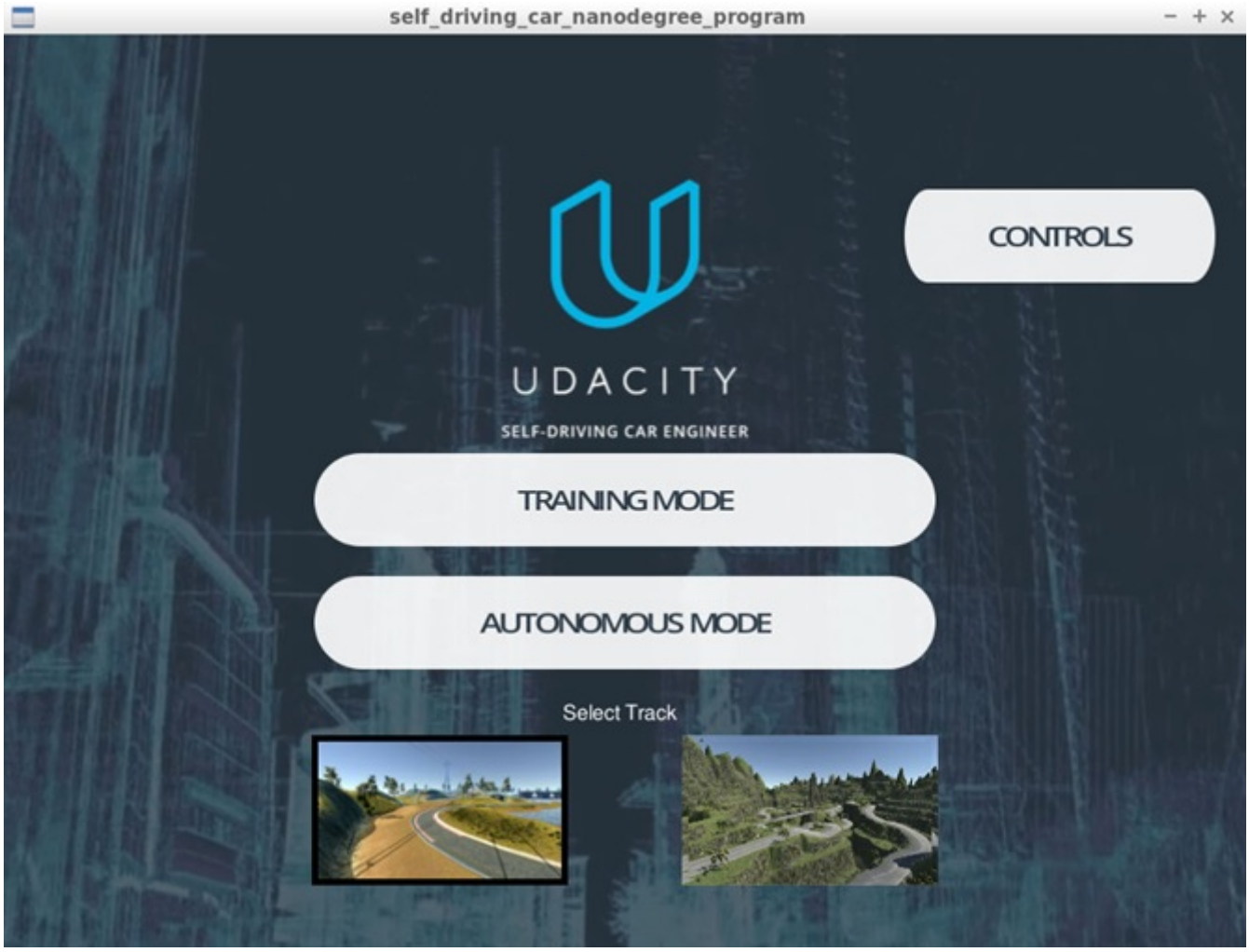
The goals / steps of this project are the following, more details see the rubric points (https://review.udacity.com/#!/rubrics/1968/view)

1. **Use the simulator to collect data of good driving behavior**
2. **Build, a convolution neural network in Keras that predicts steering angles from images**
3. **Train and validate the model with a training and validation set**
4. **Test that the model successfully drives around track one without leaving the road**

Here is a link to code file (./model.py)

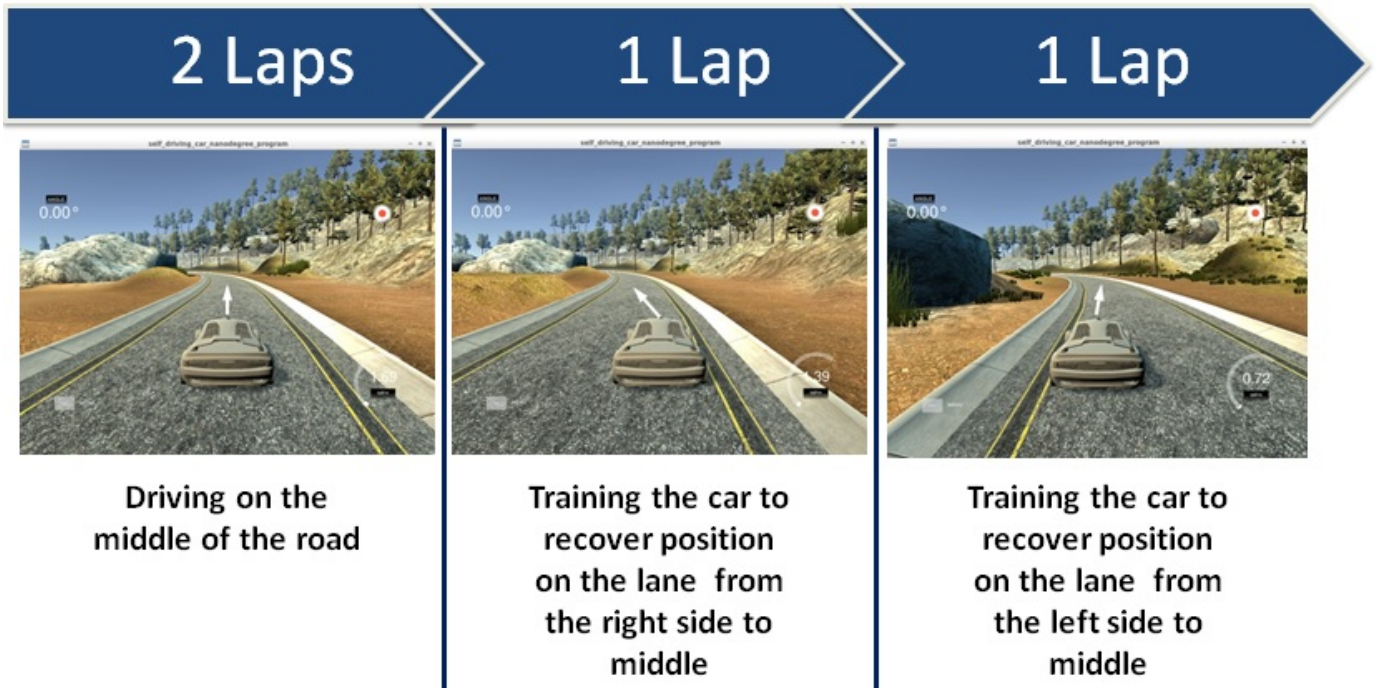
1 - Use the simulator to collect data of good driving behavior.

The simulator used was provided by Udacity and it is available in project workspace.



To accomplish the goals of the project it is necessary train and simulate the car driving autonomously only in the track 1 (the left one).

The strategy to getting data was offered in the project session by Udacity as a proposal. Below it is possible see in the schematic picture the 4 laps performed and the objective for each lap.

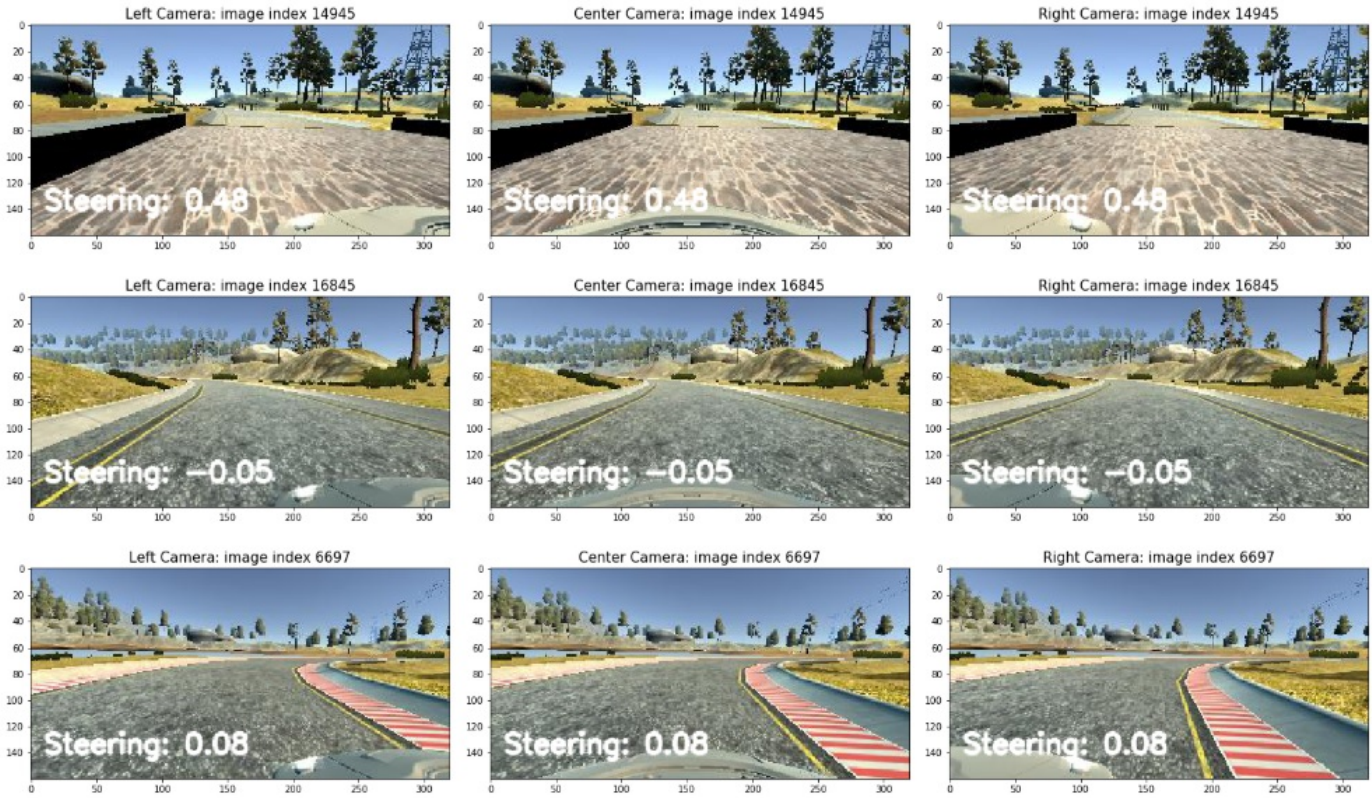


The output data was zipped and imported to my personal google drive account. Here is the link to download the file (https://drive.google.com/file/d/16wo6T1mjTcNS6QnCiLscyo8yJAqEPnOU/view?usp=sharing).

To import the data in the project workspace it was used a Script available on `stackoverflow` (https://stackoverflow.com/questions/25010369/wget-curl-large-file-from-google-drive), the answer was provided by Aatif Khan (https://stackoverflow.com/users/3292493/aatif-khan) on May 28 '18 at 21:11. The script file is available on the workspace, for more details see the file download_google_drive_data.sh (./download_google_drive_data.sh).

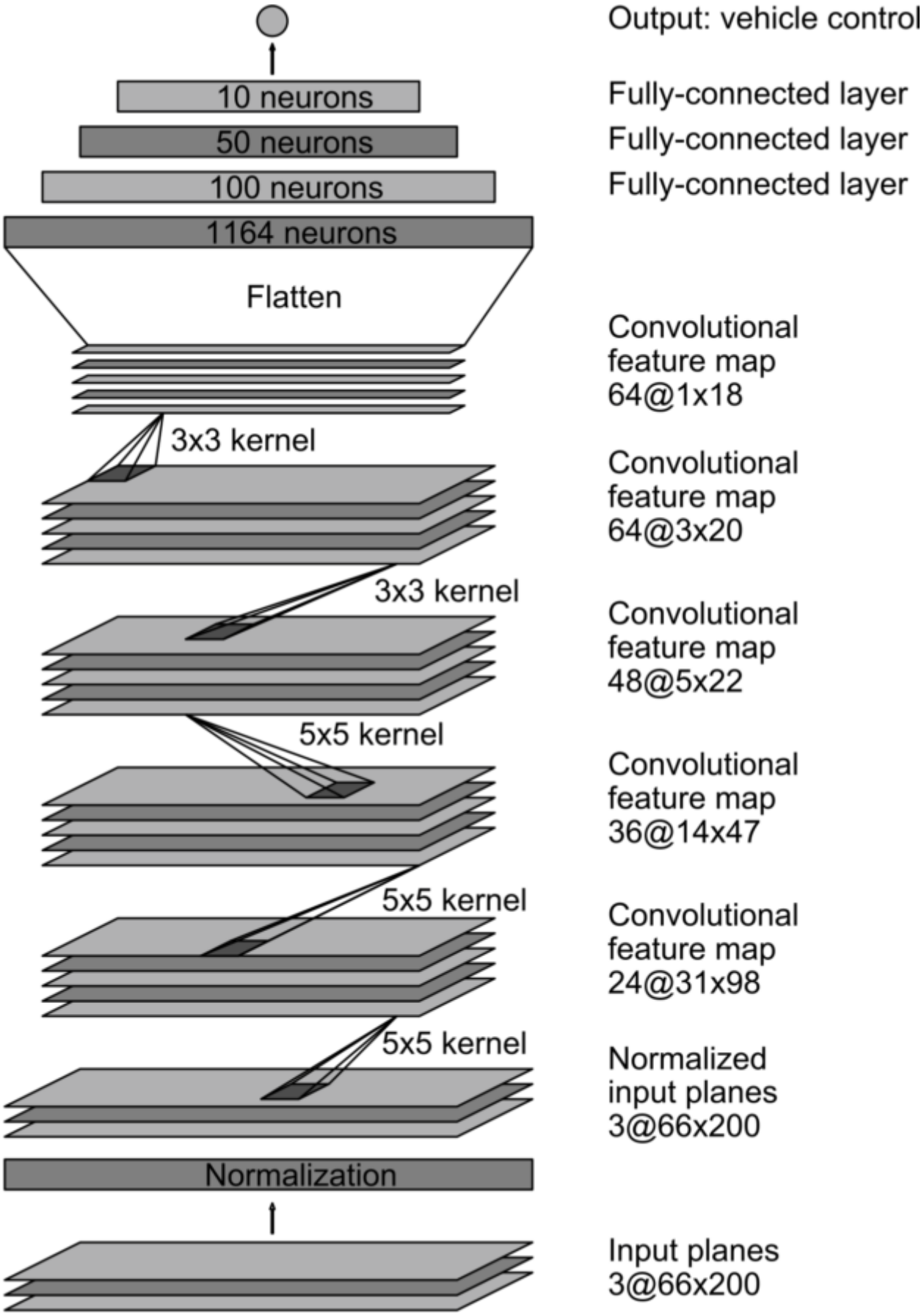
Before unzip the file is necessary delete the current folder named `data` .

The output data is a folder with images and a csv file containing in each rows the name of the images (3 images - center , left and right cameras) and the steering angle associated with the images. Below 3 examples:



2 - Build, a convolution neural network in Keras that predicts steering angles from images.

The reference used here was the convolutional neural network architeture provided by Nvidia (https://devblogs.nvidia.com/deep-learning-self-driving-cars/).



To construct the convolutional neral network was used [keras (https://www.tensorflow.org/guide/keras)](https://www.tensorflow.org/guide/keras) interface from `tensorflow`

Some parameter of the architeture were modified and the picture below show de results obtained from the function `model.summary()` (see line xxx).

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lambda_2 (Lambda)            (None, 160, 320, 3)       0
_________________________________________________________________
cropping2d_2 (Cropping2D)    (None, 65, 320, 3)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 31, 158, 24)       1824
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 14, 77, 36)        21636
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 5, 37, 48)         43248
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 3, 35, 64)         27712
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 1, 33, 64)         36928
_________________________________________________________________
flatten_2 (Flatten)          (None, 2112)              0
_________________________________________________________________
dense_5 (Dense)              (None, 100)               211300
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_6 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_7 (Dense)              (None, 10)                510
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 11
=================================================================
Total params: 348,219
Trainable params: 348,219
Non-trainable params: 0
_________________________________________________________________
```

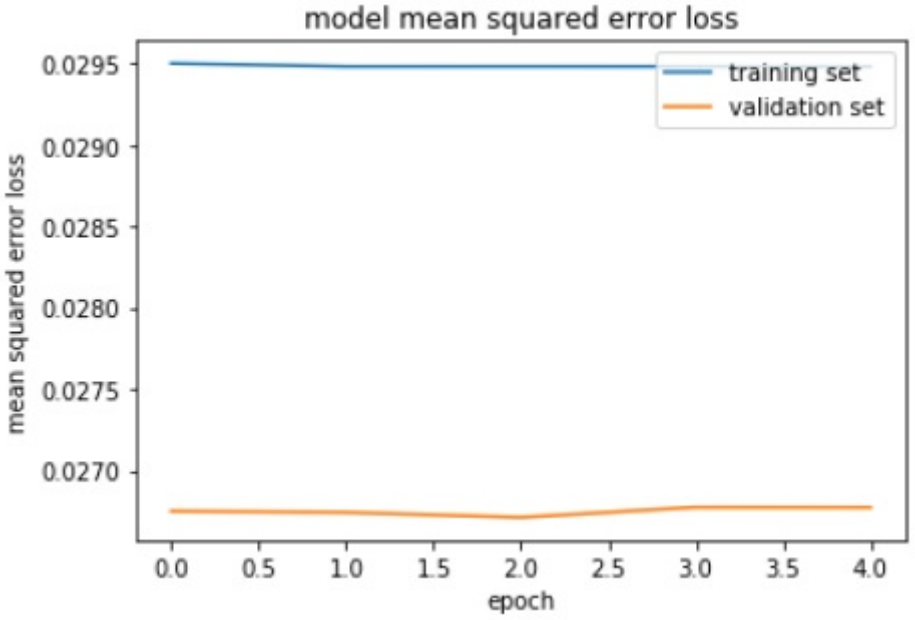
3 - Train and validate the model with a training and validation set.

The data set were created using a `generator` function following the guide provided by udacity in the project session. Using the python [scikit-learn (https://scikit-learn.org/stable/)](https://scikit-learn.org/stable/)library. the function `train_test_split` was applied to split the train and validation samples.

Model parameter tuning:

No of epochs= 5
Optimizer Used- Adam
Learning Rate- Default 0.001
Validation Data split- 0.2
Generator batch size= 32

Loss Function Used- MSE(Mean Squared Error as it is efficient for regression problem.



4 - Test that the model successfully drives around track one without leaving the road .

To simulate the algorithm result and see the car driving autonomously. it was used the script files available on the workspace and provided by Udacity. In the linux bash was inserted the command `python drive.py model.h5 run1` as first step and to converted the images generated in the run1 folder, was applied a second command `python video.py run1` .
see below the result !!!

0:00 / 1:19