# PROJECT 4 BEHAVIORAL CLONING: REPORT

PROJECT SCOPE:

The goals of this project are:

- Use the simulator to collect data to capture the driving behavior
- Build a Keras Convolution Neural Network that predicts steering angles based on camera input
- Train and validate the model using previously collected data
- Test that the model to drive within the road on Track1
- Summarize the project in a report

---

RUBRIC POINTS:

This section details how the rubric points were addressed.

*Required Files*:

The submission (https://github.com/ram-charan-m/CNN_Autonomous_Driver) includes a model.py file, drive.py, model.h5, a pdf report, and a video for each circuit, video_circuit1.mp4 and video_circuit2.mp4.

*Quality of Code*:

1. The model provided is functional, successfully generating the video.mp4 file by driving the car on the simulator.
2. The code is well commented for readability. It uses a Python generator for memory efficient operation.

*Model Architecture and Training Strategy*:

A Keras CNN model following the architecture proposed by NVIDIA mentioned in concept 15 of "Project: Behavioral Learning" module was deployed. The architecture is illustrated below. An additional fully connected layer with 1 output is added to directly output the steering angle to simulator.

The data used for validating and training the model is generated using the simulator. Seven different datasets were recorded.

| DATASET NUMBER | SUMMARY |
|---|---|
| 1 | Sample data provided by Udacity |
| 2 | 1 lap of Circuit 1 driving in lane center |
| 3 | 1 lap of Circuit 1 not driving in lane center |
| 4 | 1 lap of Circuit 1 in reverse |
| 5 | 1 lap of Circuit 2 |
| 6 | 1 lap of Circuit 2 in reverse |
| 7 | 1 dataset of recentering from lane edges from both circuits |

From these datasets, center, left and right camera images are used. A flipped version of center image is used to avoid steering bias. Steering correction for left and right images were iteratively computed.

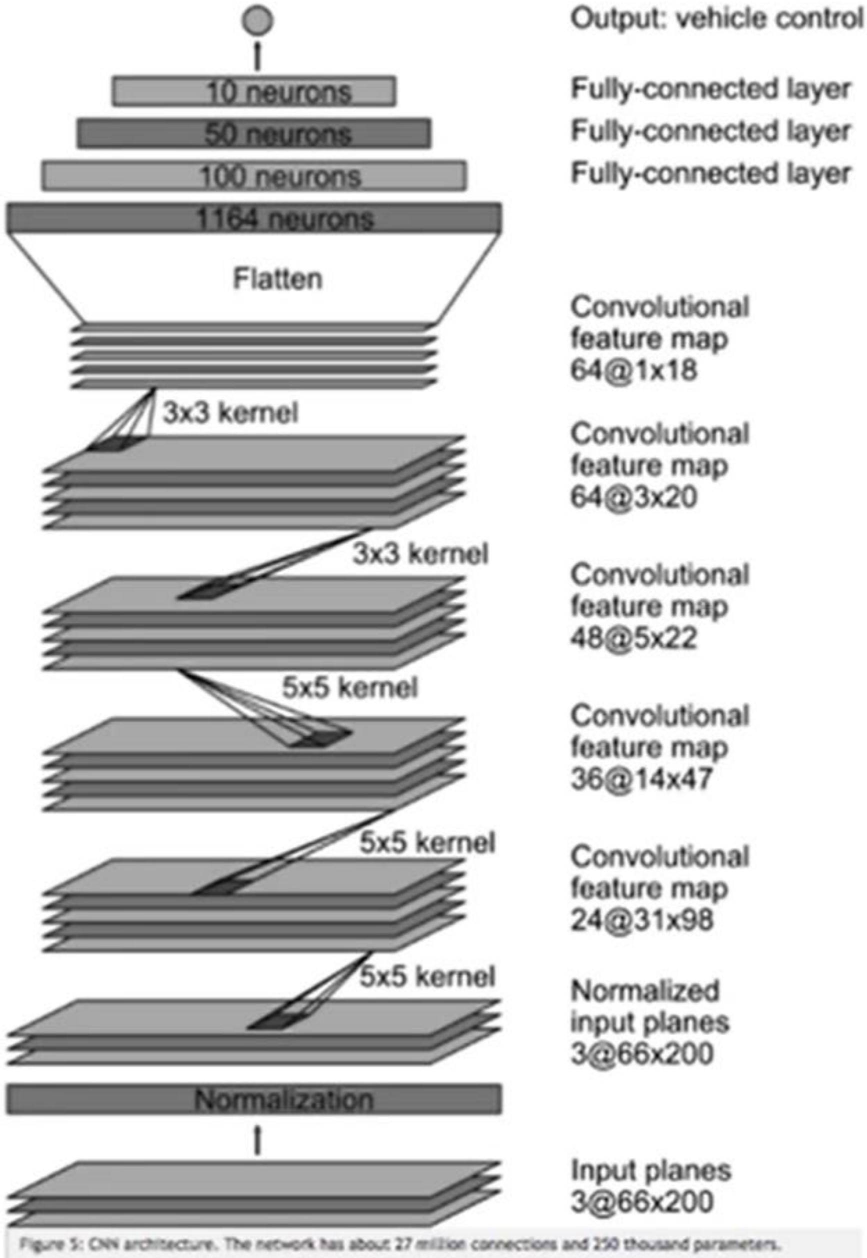



*Figure 1 NVIDIA CNN architecture (ref: Udacity CarND)*

These images are processed by mean centering them and adding a steering bias to prevent domination of no steering input. Flipping can also be considered as a quick and easy data processing technique. The images are also cropped to filter out unwanted information.

20% of the sample data is used for validation. The validation accuracy was quite low of about 0.06 mean squared error with 8 Epochs, after which the model is overtrained. To prevent overtraining, the training was aborted at 8 epochs. Two Dropout layers were added after the flatten and first fully connected layer respectively for generalization. The model was fit using an Adam optimizer to reduce RMSE of steering.

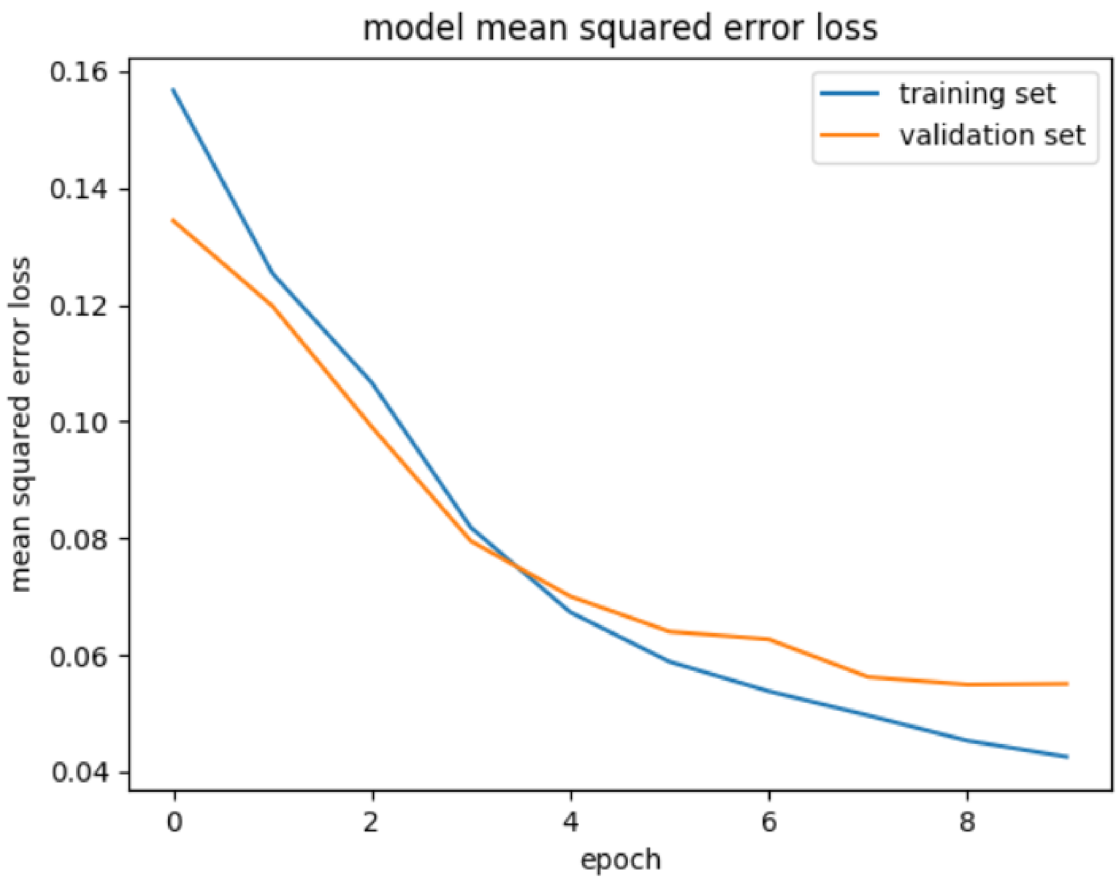
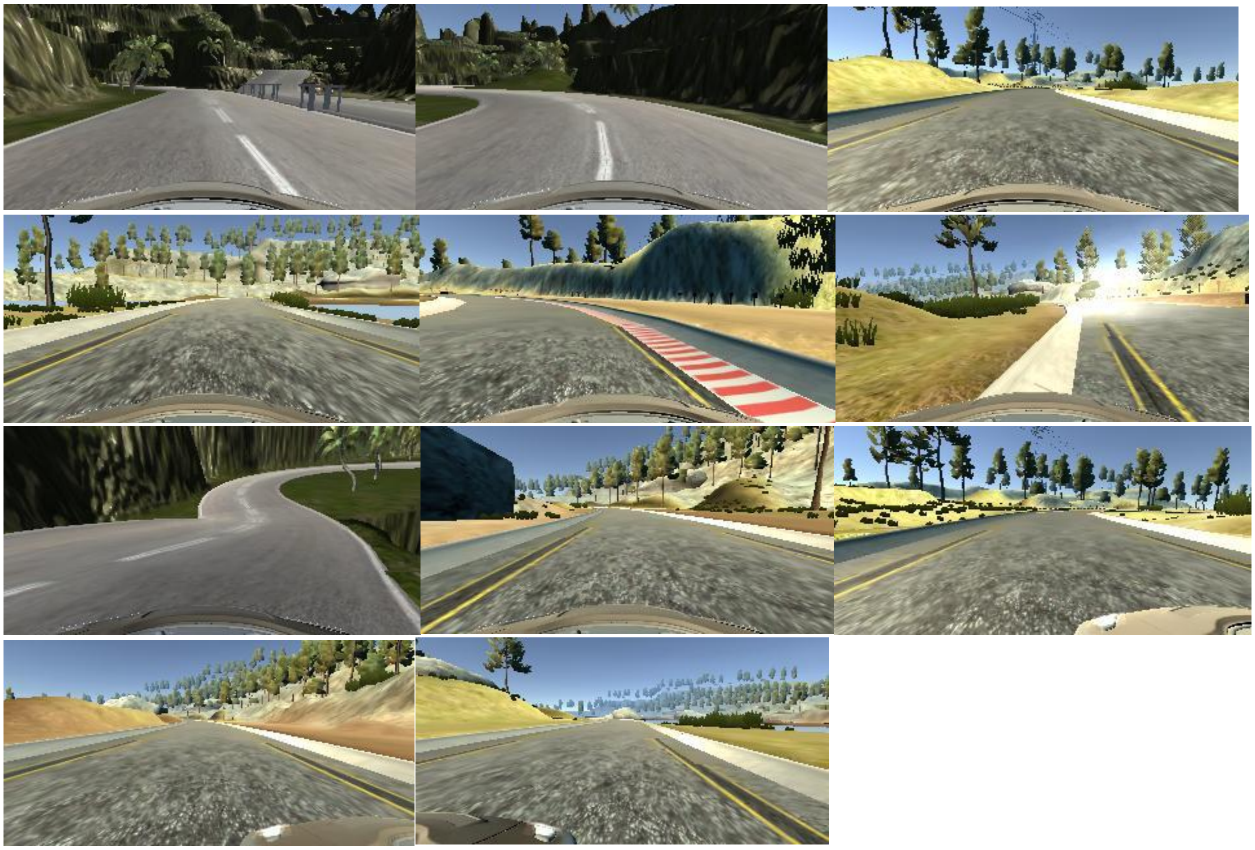



*Figure 2 RMSE for training and validation*

The training data was used to provide information on both tracks, prevent bias towards one direction of steering, prevent center driving bias and allow for recovery when the vehicle weaves off towards the sides.

*Architecture and Training strategy*:

The current report documents the architecture and design modifications along with image datasets. Few illustrations of the datasets are provided below.



*Simulation*:

The car was able to autonomously drive within the lane in ‘both’ the circuits. The results are provided in video_circuit1.mp4 and video_circuit2.mp4 respectively.