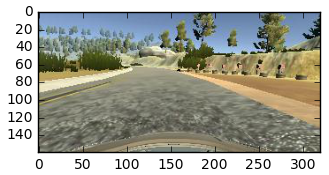
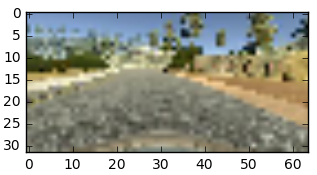
**Image Processing:**

The image size from the simulator was 320 x 160. Working with large size image had 2 issues, one loading and working with such a large size data was not feasible unnecessary overhead and second the image had lot of information not needed for the car to drive in autonomous mode.

So the image was reduced to 32 by 64. And cropped 13 px from top and 4 pixels from bottom using the Keras cropping function

Cropping2D (cropping= ((13, 4), (0, 0))

  
Image before processing

  
Image after resized

**Image Shift – Left and right camera images:**

This was the most challenging to find the right shift for the images. Tried different shifts for both left and right cameras ranging from 0.1 – 0.5 and based on the model and driving results the best shift was set at 0.2.

**Image Color:**

Since CV2.imread was used to read the images the default was BGR format so the images were transformed from BGR to RGB.

**Model Design:**

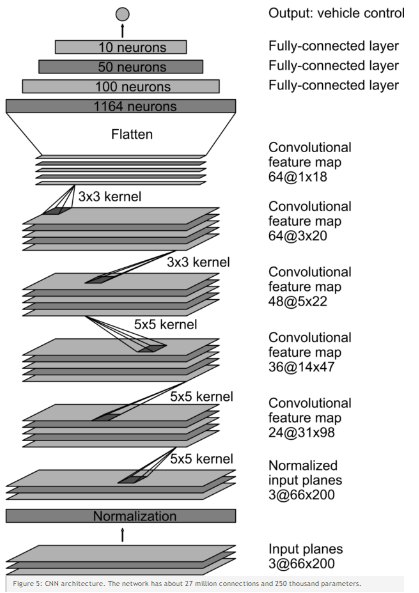
Followed the following steps to come up with a running model for Deep learning network.

**LENET:**

Started with the LENET architecture and was not completely successful in driving the car around the track, the car was failing at the turns around the bridge. So went on to use the below design

**End-to-End Deep Learning for Self-Driving Cars**

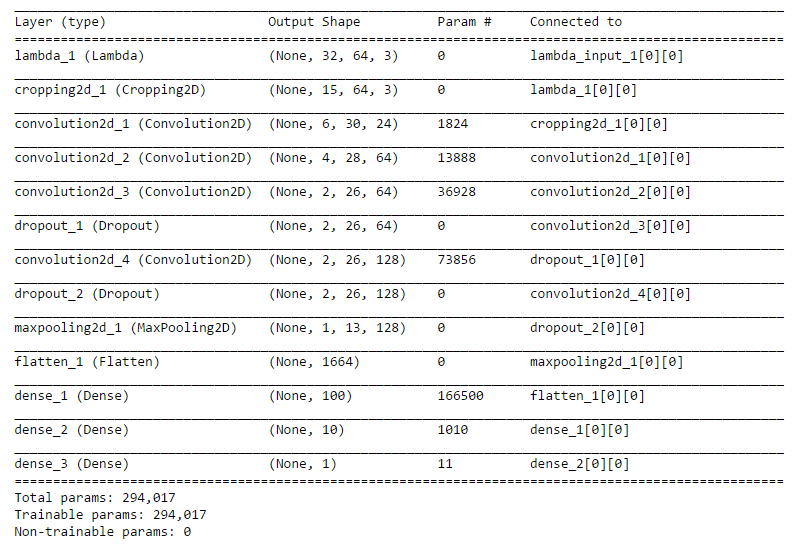
<https://devblogs.nvidia.com/parallelforall/deep-learning-self-driving-cars/>

Implemented the NVDIA architecture the only change I made in this architecture was the input size of the image.

The issues I ran into this architecture is the training time on my system. It seemed over for the project requirement I had and training this deep network was not feasible so I stated modifying this architecture and came up the below design as my final model.

**Modified Architecture:**

Below image shows the final design of the model.



**Validation and training data split**:

The training data and validation data was split 80/20.

**Epchos:**

For the model training the initial epoch used were 10 that showed the model was overfitting with the validation set jumping back up significantly as the epoch increased. So based on the first run with 10 the best number came at 4. Even though even with just 2 epoch the model was working fine but had more zig zag movement on turns, after 4 epochs the car was driving much smoother.