###Preprocessing and Model Architecture
I started with Lenet architecture, car did not stay in lane. Changed to Alex Net, the car was able to stay better than Lenet but was still was going off road.

Added flipped images technique, that did improve the performance.

Also, did some image cropping from top and bottom to get rid of irrelevant information. This definitely improved the car behavior and went further on the track without going off road.
Till now I was using just the center camera image to train the model. Added data from left and right camera images adjusting the steering appropriately. This made the car stay in lane better.

Next modified the model to look similar to Nvidia net as described below. This improved the performance quite a lot.

Convolution network used for final training:
Layer 1: Depth=25 , filter size = 5x5 , stride = 2x2
Activation = Relu
Layer 2: Depth=36 , filter size = 5x5, stride = 2x2
Activation = Relu
Layer 3: Depth=64 , filter size = 5x5, stride = 2x2
Activation = Relu
Layer 4: Depth=64 , filter size = 3x3, stride = 1x1
Activation = Relu
Layer 5: Depth=64 , filter size = 3x3, stride = 1x1
Activation = Relu

Fully Connected Network used:
Layer 6: Number of neurons= 120
Layer 7: Number of neurons= 84
Layer 8: Number of neurons= 16
Layer 9: Number of neurons= 1

Also played around with dropout, pooling but that didn't help much.

###Gathering Training data

I split the sample data provided in the project into training data (80%), validation data (20%). With above model the car was performing well on straight roads but reared off the road on curves. It would get off track and fail to get back

Tried to improve the training data by driving the car in training mode on track 1. Not much improvement in car performance.

Adjusted the batch_size=256. The car was able to reach the bridge but was going off road on the second curve.

Made some more training data by:

1. Driving the car clockwise.
2. Added data for curves by driving slowly and staying in the center.
3. Drove an extra lap both clockwise and anticlockwise.

Data Size:
#Frames total (including center, left, right, flipped) = 30324
Train on 24259 samples, validate on 6065 samples

Image shape: (160, 320, 3)

###Training Parameters
BATCH_SIZE = 256
Optimizer= Adam
Loss = MSE
validation_split=0.2

Experimented with #epochs 1-5 and more but it didn't help much, so kept it at 2 to avoid overfitting.

###Model Evaluation Metrics:

With above model, below is the metrics for 2 epochs:
Train on 24259 samples, validate on 6065 samples
Epoch 1/2
24259/24259 [==============================] - 297s - loss: 0.0619 - val_loss: 0.0520
Epoch 2/2
24259/24259 [==============================] - 276s - loss: 0.0546 - val_loss: 0.0512

###Autonomous driving performance
The car was able to stay on road for full lap and stayed same for multiple laps. The driving seems to be pretty smooth on track 1. The car does not go onto ledges or roll over any surfaces and drives pretty safe. It also recovers towards the center if it was going towards the edge (attached video)

On track 2, performance is pretty bad. The car goes off the road pretty easily and frequently. I tried training on more data gathered by driving on track 2, but that didn't help much. Few ideas to address this:

1. Gather data by driving multiple laps (its not really easy to drive on this track, so data collection is not trivial)

2. Train for more epochs
3. Try lesser resolution images
4. Add more filters in the network