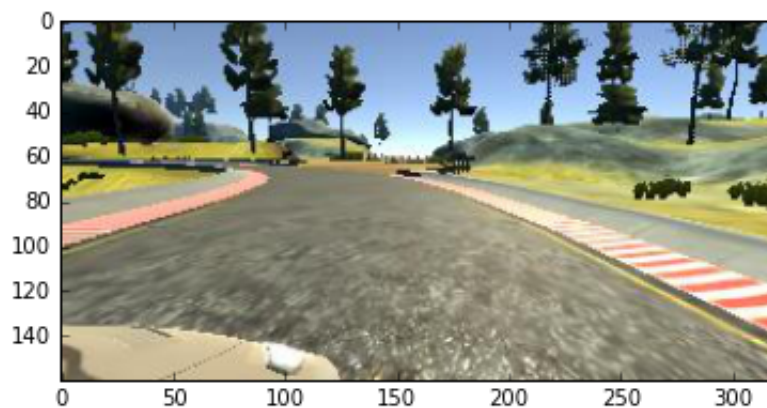


Image Selection

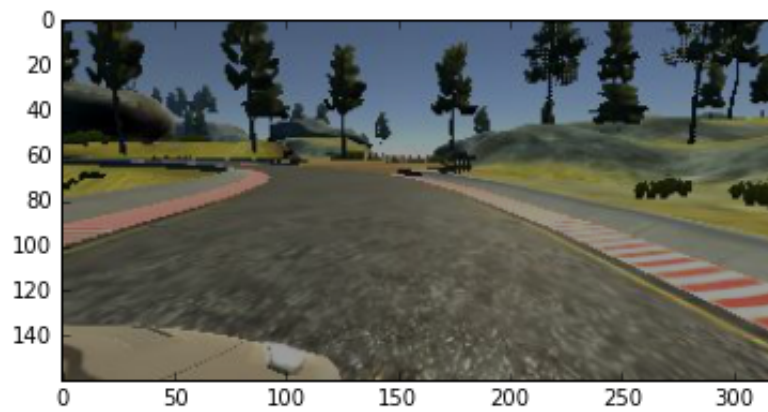
I tried to use my keyboard to control the car, but it cannot turn smoothly, so i just choose the default udacity data.

Image Preprocess

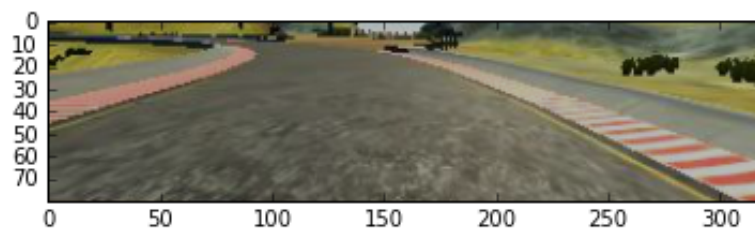
- If we only have normal driving data, we cannot get the car back on track when the car is off track, so I use images from left/right camera for training, pretend to be off track.
- 2 tracks have more left turn than right turn, so I flip the image to make neural network see the more right turn cases.
- Random change the lightness of image to prevent neural network overfitting.
- Crop images. The sky and headstock is not informative and may cause overfitting.
- Resize images. Decrease the image size without lossing information can make the neural network feel comfortable and train faster.



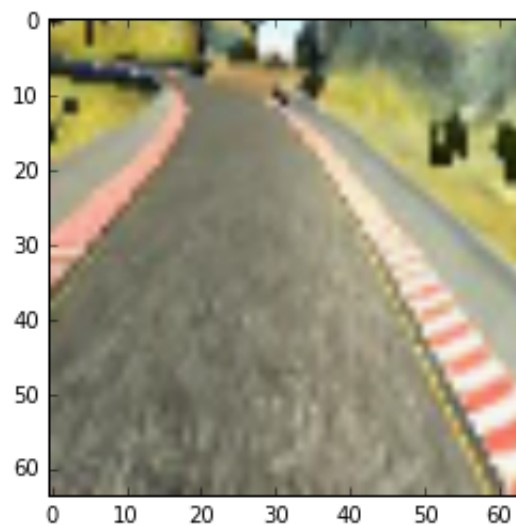
original



brightness



crop



resize

Network Architecture

- Because the image from this task is more complex than the traffic sign task, So I think NN need more conv layer and full-connected layer. And I

think the architecture from the paper that udacity suggested is a good starting point.

- I use 5 conv/maxpooling layer and 5 full-connected layer. The feature map depth is from 24 to 64 because I think the low level feature such as edges, lines is limited and the high level features are more diversiform because it combines the low level features.
- I didn't use dropout because it trains slower and no dropout is just work well.
- Learning rate 0.001, 0.0001, 0.00001 all works, but i found 0.00001 works better.