The overall strategy for deriving a good model was to use the [Nvidia](https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf) architecture since it has been proven to be very successful in self-driving car tasks. I would say that this was the easiest part since a lot of other students have found it successful, the architecture was recommended in the lessons and it's adapted for this use case.

In order to gauge how well the model was working, I split my image and steering angle data into a training and validation set. Since I was using data augmentation techniques, the mean squared error was low both on the training and validation steps.

The hardest part was getting the data augmentation to work. I had a working solution early on but because there were a lot of syntax errors and minor omissions, it took a while to piece everything together. One of the problems, for example, was that I was incorrectly applying the data augmentation techniques and until I created a visualization of what the augmented images looked like, I was not able to train a model that drives successfully around the track.