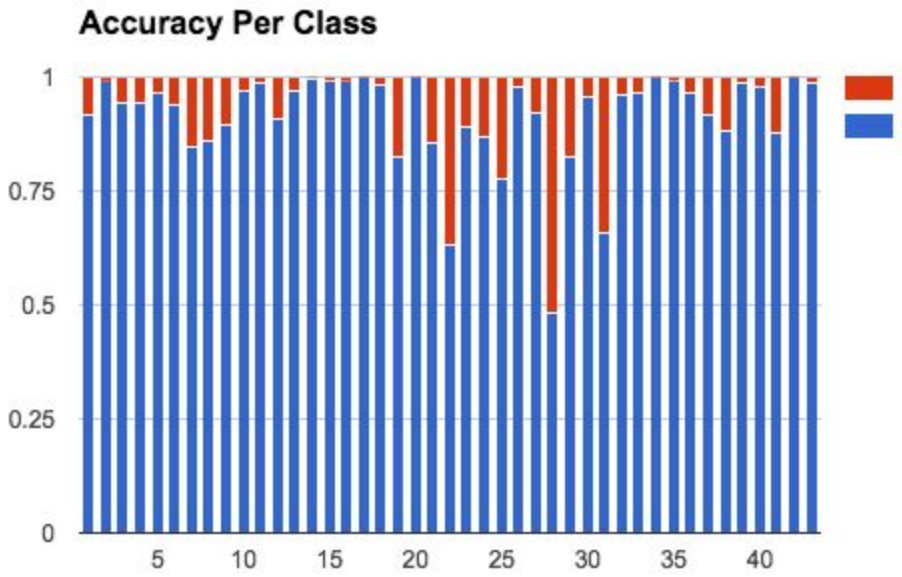
Traffic Sign Recognition

1)
   a) The size of training set is 34799 samples
   b) The size of the validation set is 4410
   c) The size of test set is 12630
   d) The shape of a traffic sign image is (32,32,3)
   e) The number of unique classes/labels in the data set is 43
2)
   Here we can visualize potential problem classes that are lowering our overall accuracy. Further work could be focused on these classes by either collecting more data, augmenting the data further or further looking into this class training data to gain clues as to its poor performance.



3) I didn't preprocess hardly at all. I simply one-hot encoded the labels and normalized the input values of the pixels to gain performance. I considered converting the images to grayscale or upsampling but figured the color data was too valuable and was able to get decent enough results without upsampling.

4) My final results were:
   Training: 99%
   Validation: 93%
   Test: 94%

   I started with LeNet but it was obvious that I wasn't going to get proper accuracy with that design. I had done something like this previously using AlexNet with very good results so I wanted a new challenge. I tried to use a small design while mimicking some choices from AlexNet such as stacking many of the same filter sized conv layers together. The major hyper parameters I tweaked were the # of epochs and batch size. I tried making my batch size large enough that it wouldn't cause excess noise while small enough to avoid overfitting the training data. I didn't manually select a learning rate but left it as the default Adam value: `0.001.` I found a strong enough accuracy with this default rate that I didn't investigate further. I chose 50 epochs as I hoped to train long enough to see results but not overfit. In retrospect I should have used early termination or at least checkpointing to find the optimal epoch.

   My model architecture ended up being:

Lambda	Normalize around (-1.0, 1.0)
Convolution	16 filters, size 3x3
Convolution	32 filters, size 5x5
Max Pooling	Strides 2x2
Convolution	64 filters, size 3x3
Max Pooling	Strides 2x2
Convolution	64 filters, size 5x5
Max Pooling	Strides 2x2
Dropout	Likelihood 20%
Max Pooling	Strides 2x2

Flatten	
Dense	Size: 512
Activation	Relu
Dense	Size 128
Activation	Relu
Dense	Size 128
Activation	Relu
Dense	43
Activation	Softmax

5) My 5 images from the web are:

Some of these images have significant amounts of sky in them which could have potentially tricked the net when looking for blue signs. As well, in image #3 there are actually two signs which could have caused issues. Furthermore, the images are all very different aspect ratios so when they are transformed to 32x32x3, they could have severe warping.

The net was able to correctly predict 3 of the 5 images and had the correct prediction in the top 5 predictions for 4 of the 5 images.

Label	Predicted	Probability
1	29	0.996809
2	2	1.0
13	13	0.984019
11	11	1.0
23	36	0.905292