CarND –Traffic Sign Classifier Write-up

Data Set Summary and Exploration

The German Traffic Sign Dataset used has:

A training set size of 34,799 .
A validation set size of 4,410.
A testing set size of 12,630.
The shape of each traffic sign image is 32x32x3.
And the number of unique classes (Amount of Traffic Signs) in the dataset is 43.

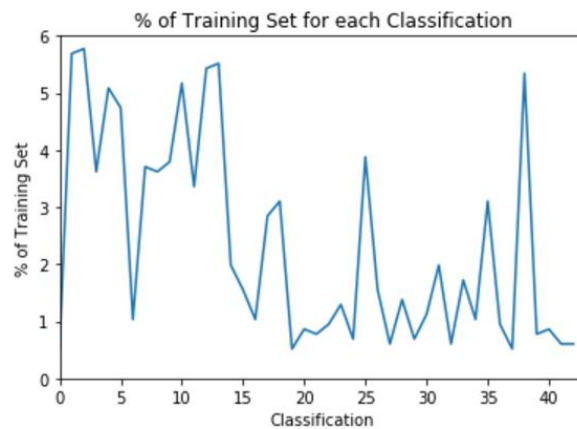The Amount of Images in each classification as a percentage of the whole training set is:

Figure 1: Percentage of Training Set for each Classification

The amount of images in each classification as a percentage of the whole validation set it:
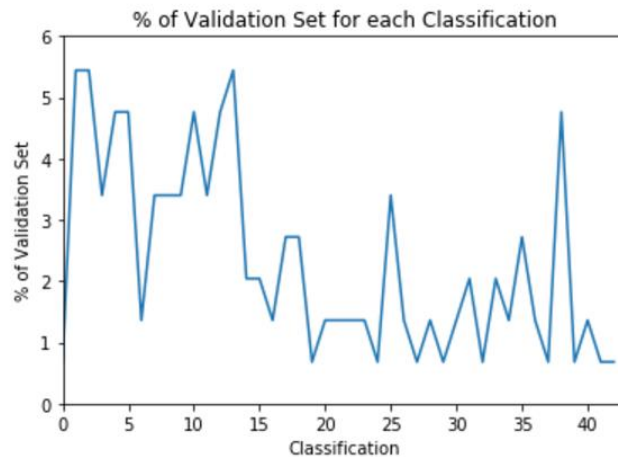

Figure 2: Percentage of Validation Set for each Classification

Both histograms look similar, making the validation set a good representation of the training set.

Design and Test Model Architecture:

Preprocessing:

Images where preprocessed with changing the images from RGB to greyscale, and then normalizing the mean and standard deviation of the image.
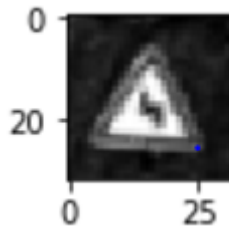

Figure 3: Before Processing Image

Figure 4: After Processing Image

The image was converted to greyscale in order to reduce the complexity of the Convolutional Neural Network. Traffic Signs do not share the same shape in only vary in color, which makes color data unnecessary when differentiating traffic signs.

The input data was normalized in order to aid the convergence of the optimization algorithm.

Model:

The Final Convolutional Neural Network Model used has the following layers:

Layer	Description
Input	32x32x1 image
Convolutional 5x5	1x1 Stride, valid padding, Output is 28x28x6
RELU	
Max Pooling	2x2 Stride, Output is 14x14x6
Convolutional 5x5	1x1 Stride, valid padding, Output is 10x10x16
RELU	

Max Pooling	2x2 Stride, Output is 5x5x16
Flatting	Output is 400
Fully Connected Layer 1	Input is 400, Output is 120
RELU	
Dropout Layer 1	
Fully Connected 2	Input is 120, Output is 84
RELU	
Dropout Layer 2	
Output Layer	Input is 84, Output is 43
Softmax	

Training:

Training was completed using an Epoch of 10, a Batch Size of 128, a learning rate of 0.001, and the Adam Optimizer.

I started with the LeNet Architecture and made adjustments based on the output validation set accuracy.

While experimenting with different architectures and parameter values, I found that over fitting was a major issue. This is when I decided to change add 2 layers of dropout after the fully connected layers. It also convinced me to normalize the input data in the preprocessing phase to help with the convergence of the optimizer.

My final model results were:

Validation set accuracy of 94.7% and a Test Set Accuracy of 93.3%.

Test a Model on New Images:

Five New Images used:

Figure 5: 20km/h

Figure 6: Children Crossing

Figure 7: No Entry

Figure 8: Right Turn

Figure 9: Stop

From looking at some of the images in the dataset, the size of the sign in the image is different. The cropping the perspective of the image might affect each image.

Predicted Outputs:

Image	Prediction
Speed Limit (20 km/h)	Children Crossing
Children Crossing	Right-of-way at the next intersection
No Entry	No Entry
Turn Right Ahead	Turn Right Ahead
Stop	Stop

The model guessed 3 of 5 correctly making the accuracy 60%. The testing sets accuracy is 93.3%, which means that there seems to be an overfitting issue with the model.

Softmax Visualization:

The first images second highest softmax value was the correct estimate, but there is a drastic difference between the two. The second images was completely incorrect. My assumption for this error is from the dataset having much less of a percentage of training examples that are Children crossing signs. It is a relatively complex sign so maybe having it more represented in the training set would improve the result. Overfitting in the model may have also led to these errors.
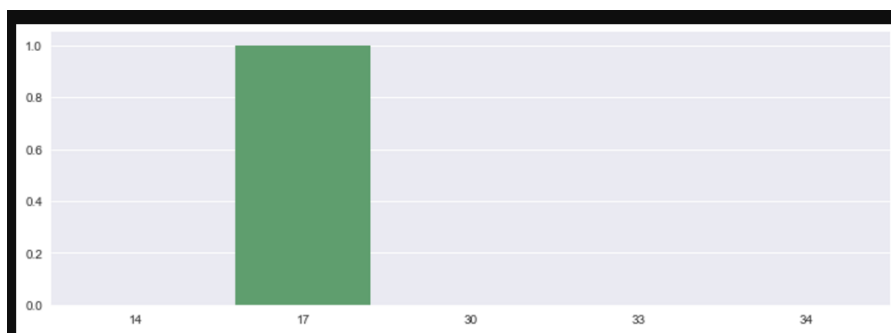
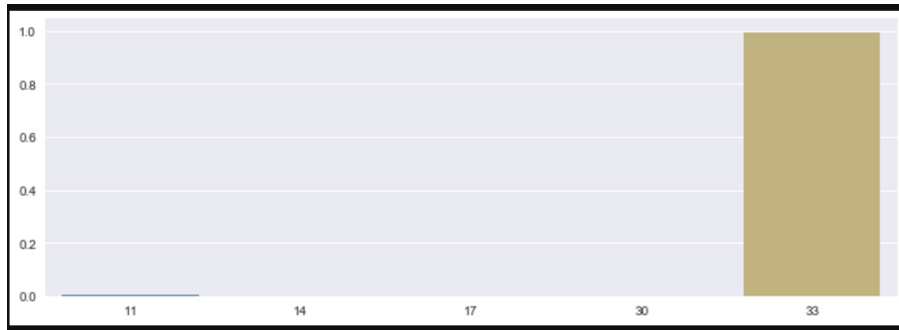Predicted Class: 28 | True Class: 0
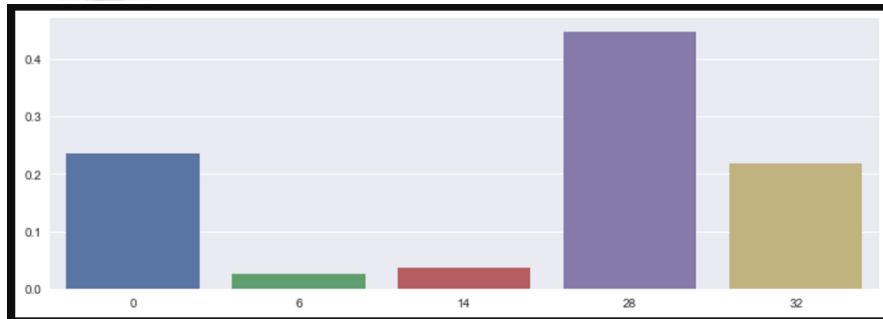
Predicted Class: 30 | True Class: 28
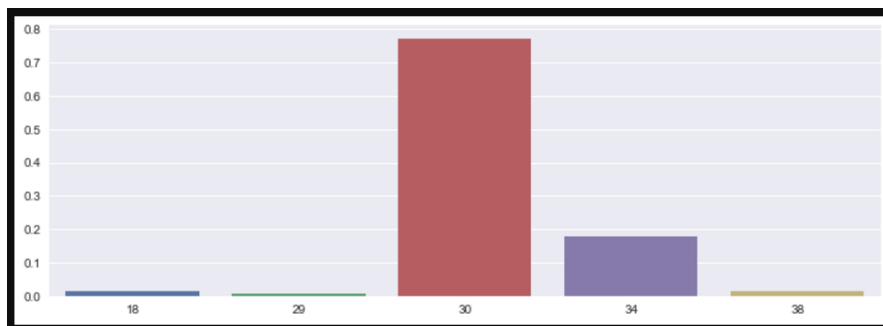
Predicted Class: 17 | True Class: 17

Predicted Class: 33 | True Class: 33

Predicted Class: 14 | True Class: 14