Traffic Sign Classifier

Data Set Summary & Exploration

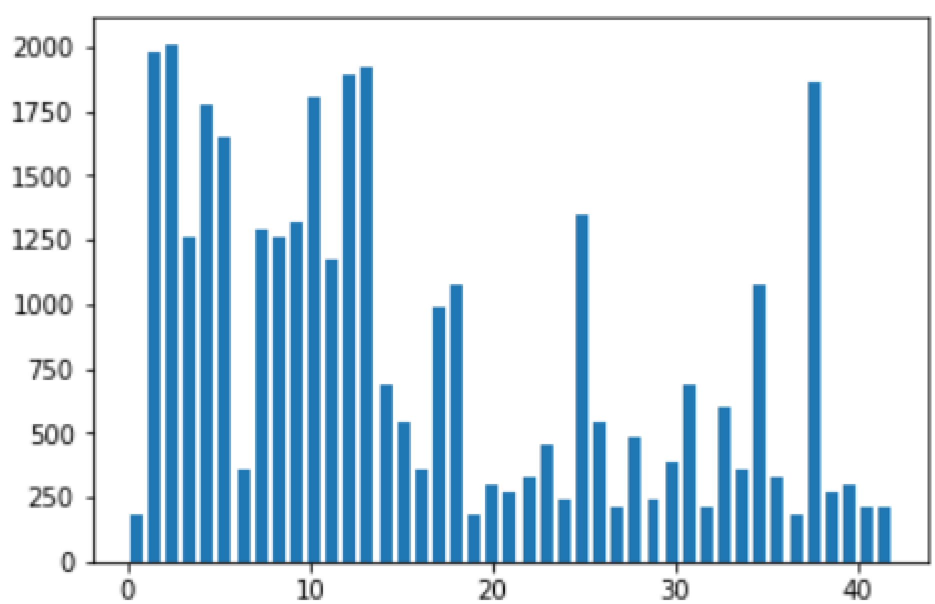
1. Basic summary of the data set.

I used the numpy library to calculate summary statistics of the traffic signs data set:

- Number of training examples = 34799
- Number of testing examples = 12630
- Image data shape = (32, 32, 3)
- Number of classes = 43

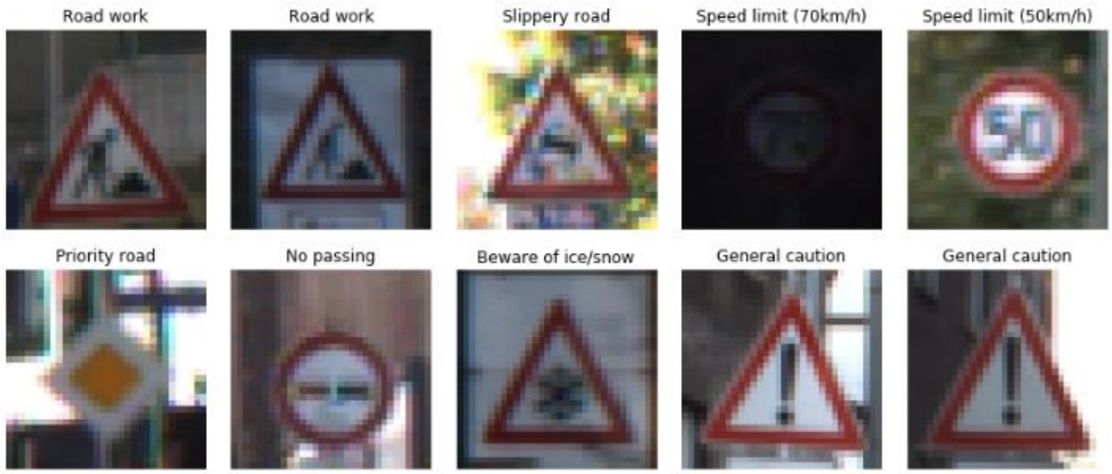
2. Exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing the frequencies of the various classes on the training set.



The X-axis is the 43 classes of the signals to be classified and the Y-axis is the frequency of their occurrences in the training data set.

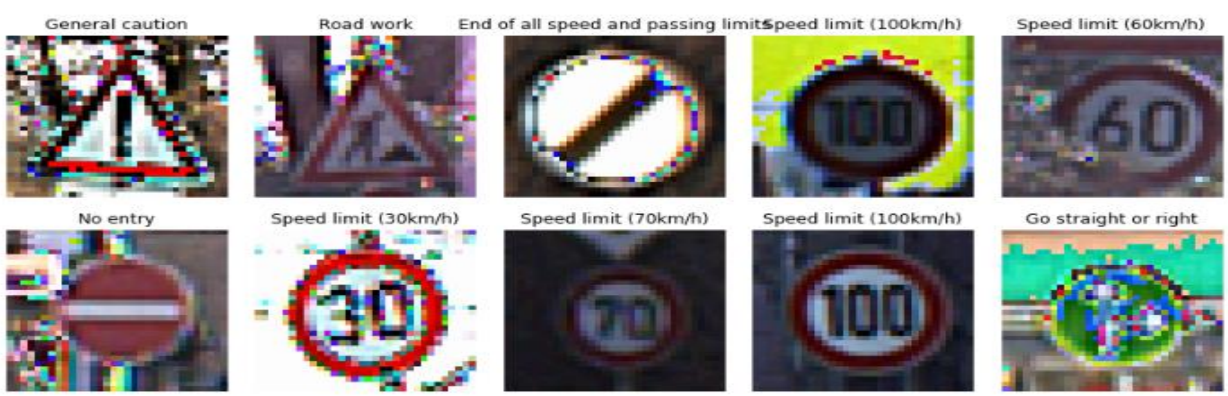
Below is the display of few random sings in the training set with their labels.



Design and Test a Model Architecture

1. **Preprocessed the image data:** Various techniques were chosen and experimented with for the preprocessing of the image file
   a) First step, the image was resized and cropped to focus on the important feature of the traffic signals instead of the world around it. But I found a decrease of validation accuracy on doing this. Thus I decided to focus only on sharpening the image,
   b) Second step, the images were converted to grayscale, but I realized that the color pays an important role in classifying the signals as I was getting better accuracy with colored images. Thus I left the images colored.
   c) Third step, the images were normalized.

Below is an example of random traffic signals images after preprocessing.



2. Final model architecture

Starting with the LeNet architecture and after experimentation the final model has the following layers:

Layer & Description	Input	Output
Layer 1: 5x5 Convolutional, followed by RELU activation and dropout.	32x32x3	28x28x6
Layer 2: 5x5 Convolutional, followed by RELU activation and dropout.	28x28x6	24x24x16
Layer 3: 1x1 Convolutional, followed by RELU activation and dropout.	24x24x16	24x24x32
Max-Pooling.	24x24x32	12x12x32
Layer 4: 3x3 Convolutional, followed by RELU activation and dropout.	12x12x32	10x10x64
Max-Pooling.	10x10x64	5x5x64
Layer 5: 2x2 Convolutional, followed by RELU activation and dropout.	5x5x64	4x4x128
Layer 6: 1x1 Convolutional, followed by RELU activation and dropout.	4x4x128	4x4x256
Flatten	4x4x256	4096
Fully Connected Layer followed by RELU activation and dropout.	4096	120
Fully Connected Layer followed by RELU activation and dropout.	120	84
Fully Connected layer	84	43

3. **Training the model.** The model is a modified LeNet with 4 added convolution layers out of which two are 1x1 convolution layers just to increase the depth. Following are the specification of the optimizer.

   a) **Type of optimizer:** Used an AdamOptimizer

   b) **The batch size:** 128

   c) **Number of epochs:** 20

   d) **Hyper-parameters** 1) ***learning rate*** **:** 0.001 **2)** ***mu*** = 0 **3)** ***sigma*** = 0.1

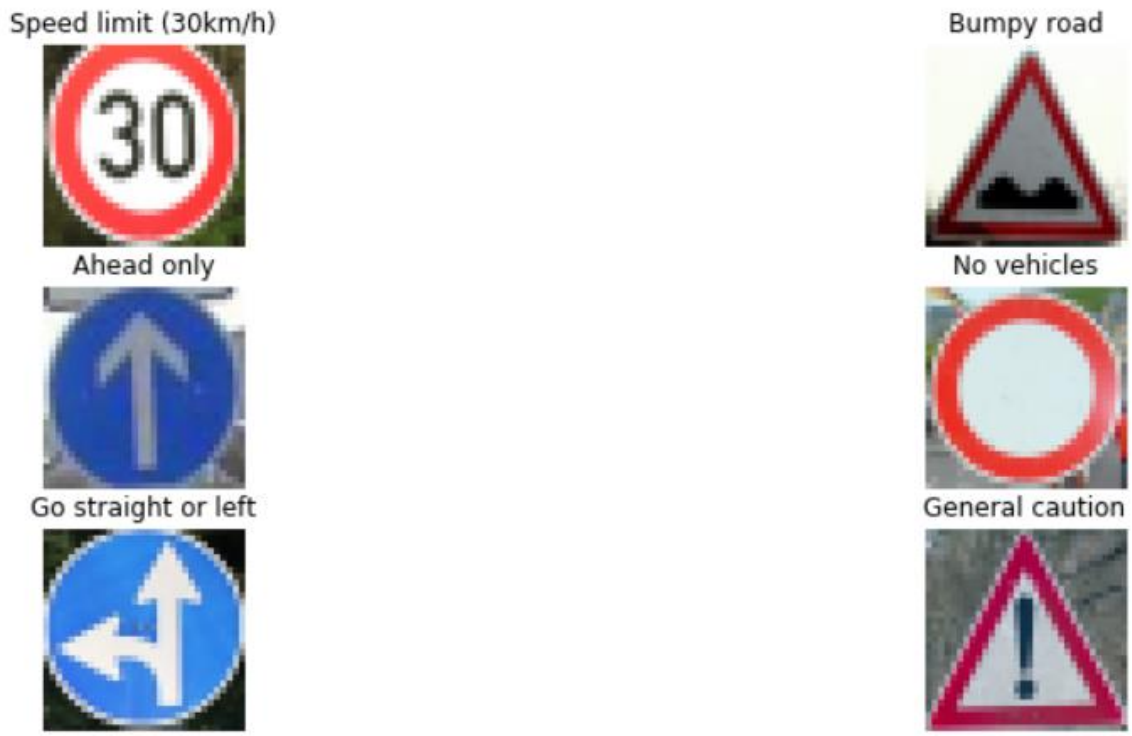
4. **Approach taken for finding a solution.** I started with 10 epochs and included grey-scale in the initial approach and got an accuracy of 85% in the initial attempts. After a) retaining the color b) increasing the depth of the model c) increasing the depth of the model I achieved the following accuracies.

   - Validation set accuracy of **96.60%**
   - Test set accuracy of **93.95%**

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Here are five German traffic signs that I found on the web:



All the images were made of the same size as the test and the validation set, both speed limit and no vehicle, bumpy road and general caution and ahead only and go straight or left are similar and would be difficult to classify.

2. Discuss the model's predictions on these new traffic signs

Here are the results of the prediction:

The model was able to correctly guess 4 of the 6 traffic signs, which gives an accuracy of 66.66 %. More training with larger training set can improve the accuracy. Also if there is more computing power the model can be made complex and even deeper.