Traffic Sign Classifier

<u>Basic summary of the data set</u>

###Data Set Summary & Exploration
####1. Provide a basic summary of the data set

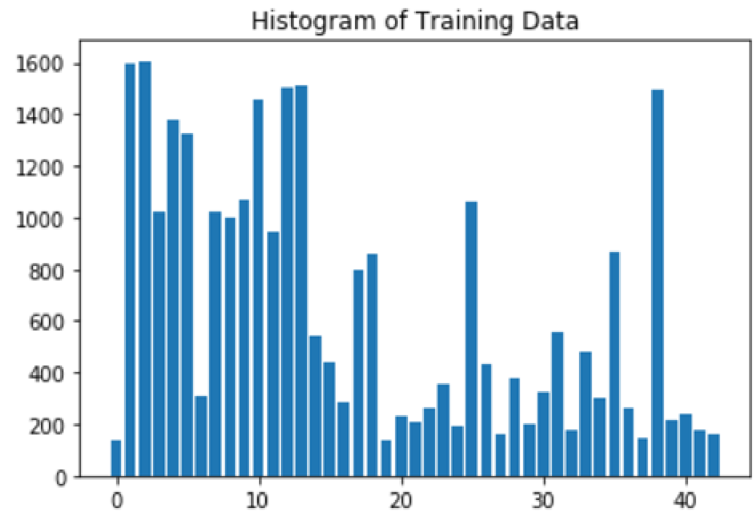
I used python to calculate the summary statistics of the traffic signs data set (Initial Summary) *However, this is not the final set for training and validating the model. The actual set is available after Augmenting the image set.*

- The size of training set is **34799**
- The size of the validation set is **4410**
- The size of test set is **12630**
- The shape of a traffic sign image is **(32, 32, 3)**
- The number of unique classes/labels in the data set is **43**

<u>Exploratory visualization of the dataset</u>

####2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing how the data is distributed for each class. The x-axis represents the class and y-axis the number of training data for each class.



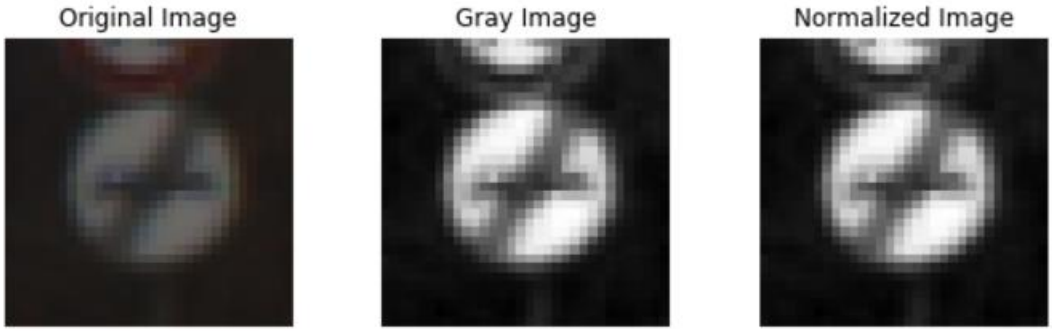
<u>Design and Test a Model Architecture</u>

####1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc.

As a first step, I decided to convert the images to grayscale because gray scale images have only intensity information and it is an important feature for distinguishing visual features.

Next, I normalized the image so that it the pixel intensities are within a certain range. This shall avoid poor contrast within the image

Here is an example of an original traffic sign image after gray scaling and normalization.



Normalization is done using the formula (pixel - 128)/ 128 as given in the project description.

I decided to generate additional data because from the histogram it is evident that few classes have very less training data and others have pretty high data.

Augmented data is generated by following steps

a. Randomly flip an image horizontally (random_flip_left_right)
b. Random Saturation (random_saturation)
c. Adjust the brightness of images by a random factor (random_brightness)
d. Adjust the contrast of an image by a random factor (random_contrast)
e. Adjust the hue randomly (random_hue)
f. Randomly crop an image to a given size (random_crop)
g. Pad the cropped image to have same input shape before cropping (pad)

I randomly chose the above steps to generate augmented image. Every class with training data less than 200 shall be augmented with new images.

Here is an example of an original image and an augmented image

Summary statistics of the traffic signs data after Augmenting. This is the final data with which model is trained, validated and tested.

- The size of training set is **27887**
- The size of the validation set is **6972**
- The size of test set is **12630**
- The shape of a traffic sign image is **(32, 32, 1)**
- The number of unique classes/labels in the data set is **43**

The difference between the original data set and the augmented data set are Brightness, Contrast, Saturation, Hue, Cropped and Padded and Flipped Horizontally.

<u>Final model architecture</u>

####2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model.

My final model consisted of the following layers

Layer	Description
Input	32x32x1 Gray Scale Image
Convolution 5x5	1x1 stride, VALID Padding, outputs 28x28x6
RELU	Rectified Linear Unit
Max pooling	2x2 stride, outputs 14x14x6
Convolution 5x5	1x1 stride, VALID Padding, outputs 10x10x16
RELU	Rectified Linear Unit
Max pooling	2x2 stride, outputs 5x5x16
Flatten	Input = 5x5x16. Output = 400
Fully Connected	Input = 400. Output = 120
RELU	Rectified Linear Unit
Fully Connected	Input = 120. Output = 84
RELU	Rectified Linear Unit
Fully Connected	Input = 84. Output = 43

<u>Training the model</u>

####3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

The model was trained on Adam Optimizer instead of most common approach of stochastic gradient descent optimizer algorithm with a batch size of 128 and total epochs of 10. The optimizer updates the network weights during the learning process. The learning rate was chosen as 0.001. Softmax cross entropy is used to calculate the entropy between logits and labels. The traffic sign labels are encoded as one-hot encoding.

The approach taken for finding a solution

####4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the architecture is suitable for the current problem.

My final model results are:

- Training set accuracy of **98.5%**
- Validation set accuracy of **97.2%**
- Test set accuracy of **87.4%**

The LeNet architecture approach was chosen to train model to classify traffic signs. The Convolution neural network are used for image classification problem and LeNet is basically a combination of several CNN with subsampling between each layer. With the validation accuracy of 97.2% it is a pretty good model.

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:

####2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set

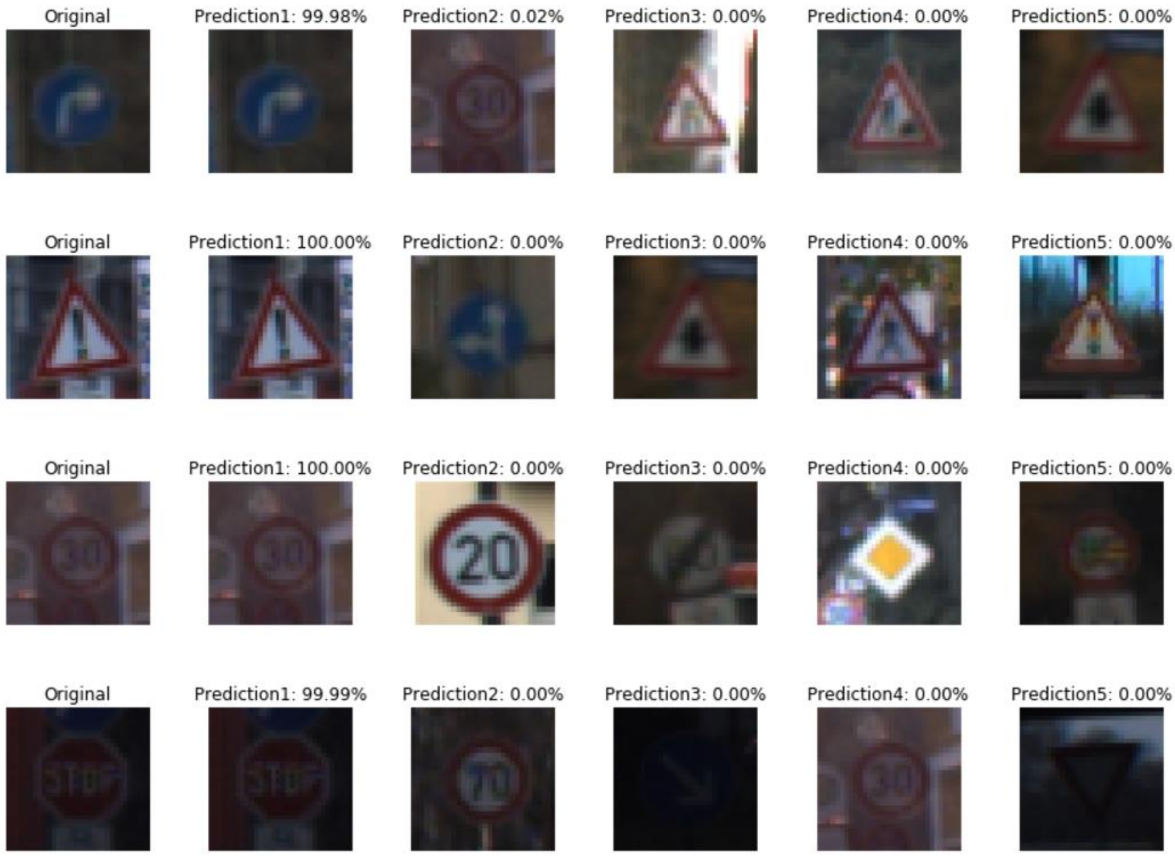
Here are the results of the prediction

Image	Prediction
Turn Right ahead	Turn Right ahead
General caution	General caution
Speed limit (30km/h)	Speed limit (30km/h)
Stop	Stop
Speed limit (70km/h)	Speed limit (70km/h)
Road Work	Road Work

The model could correctly guess 6 of the 6 traffic signs, which gives an accuracy of 100%.

####3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability

The top five soft max probabilities for all the new images are as shown below



The sign board of speed limit 70km/h is the toughest one. The model predicts it with only 58% accuracy. It predicts it as sign board for speed limit of 30km/h and 20km/h with percentage of 32 % and 9% respectively.