

Traffic Sign Detection

Udacity Term - 1

Period October to January

1. System & Software Specification

OS - Windows 7

Hardware: Intel i7 core CPU

Programming language: Python 3.x

Python Libraries used:

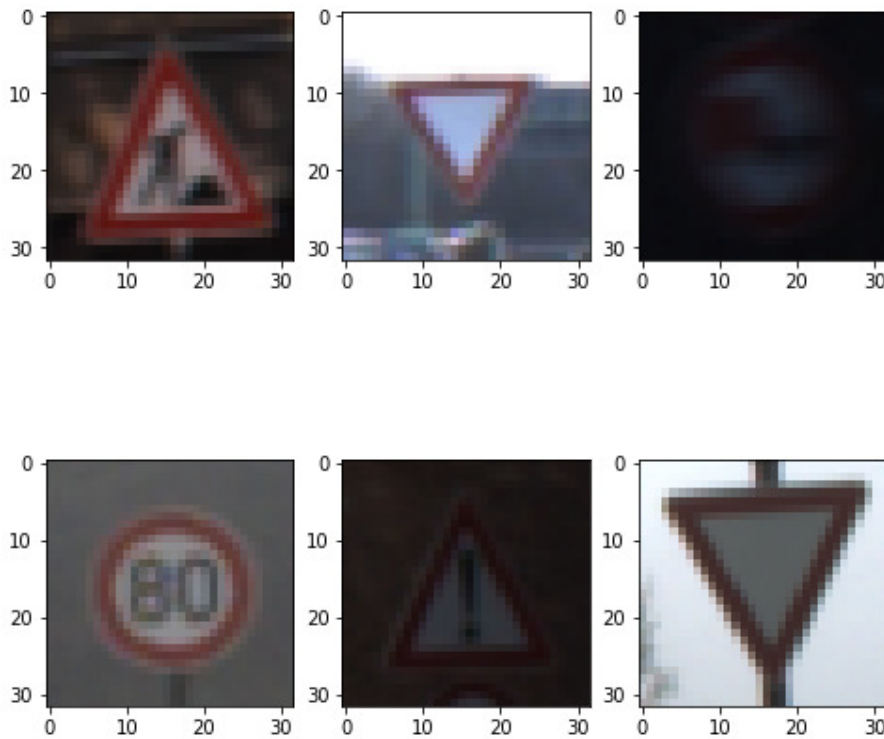
- OpenCV: library name "cv2" . Used for image processing
- Numpy: Array related functionality
- Matplotlib: used for plotting images
- pickle: Used to load test, validation and test data

2. Data Set Summary & Exploration

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

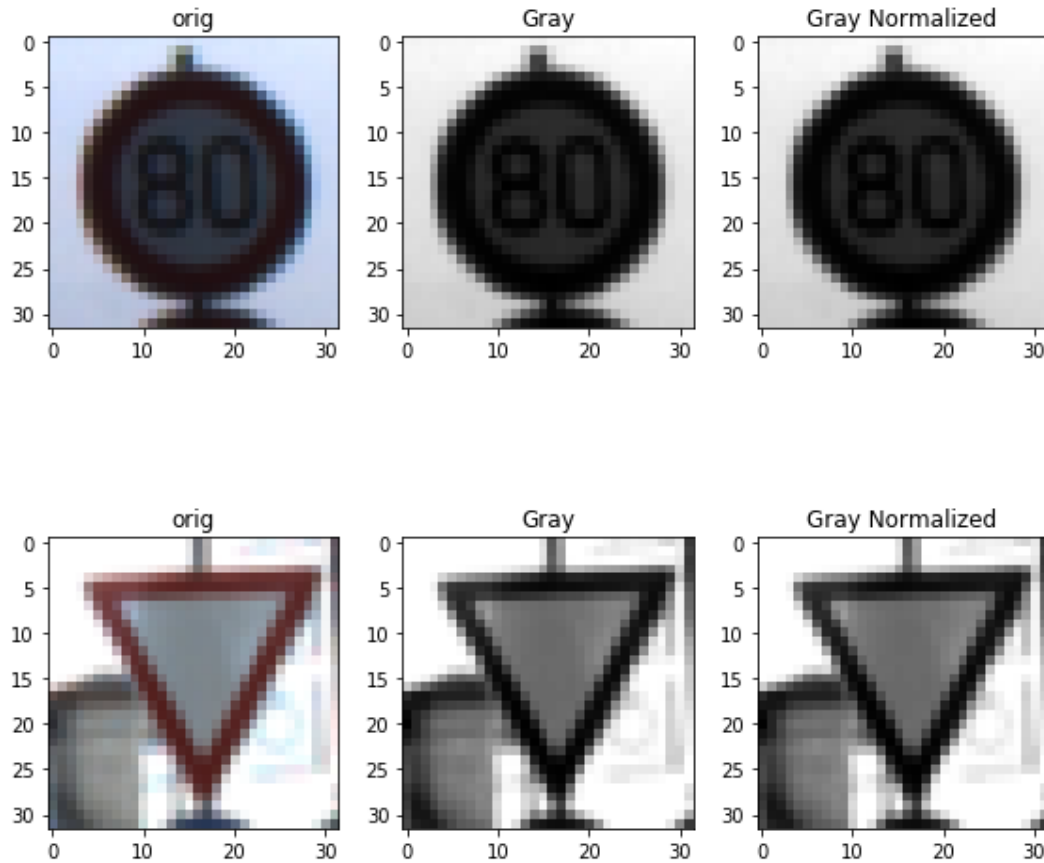
- The size of training set is 34799
- The size of the validation set is 4410
- The size of test set is 12630
- The shape of a traffic sign image is (32, 32, 3)
- The number of unique classes/labels in the data set is 43

Some examples of the training data set is as shown below.



3. Preprocessing

The training images are color images (3 channel, R, G, B). These are converted to grey scale images using OpenCV. And further on, normalized. Sample images are shown below



4. Model Details

.The model details are as shown below.

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 1	Description
Input	32x32x1 gray normalized image
Convolution	Input = 32x32x1. Output = 28x28x32., Kernel 5X5, Stride 1X1, Padding : valid
Activation	RELU
Max pooling	2x2 stride, Input = 28x28x32, outputs 14x14x32

Layer 2	Description
Input	14X14X32
Convolution	Input = 14X14X32. Output = 10x10x64., Kernel 5X5, Stride 1X1, Padding : valid
Activation	RELU
Max pooling	2x2 stride, Input = 10x10x64, outputs 5x5x64
Flatten	Input = 5x5x64, output is 1600

The next layer is a fully connected layer with 1600 inputs and 120 outputs

Layer 3	Description
Input	1600
Convolution	Input = 1600 output 120
Activation	RELU

Layer 4	Description
Input	120
Convolution	Input = 120 output 84
Activation	RELU

Layer 5	Description
Input	84
Convolution	Input = 84 output 43
Activation	RELU

5. Approach

Initially I used the Lenet model for the training. I noticed that, the accuracy wasn't good enough.

To improve the accuracy the following changes were done

- 1) Convert to gray scale
- 2) Normalize
- 3) Change the number of filters to 32 in layer 1 and 64 in layer 2

The initial settings for the training parameters are set as below.

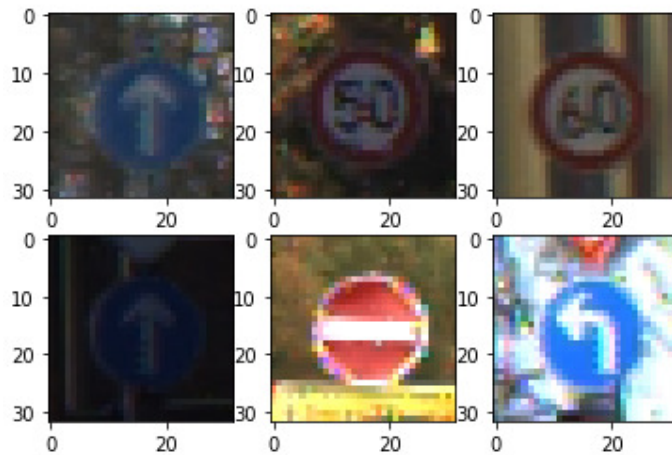
- 1) EPOCH = 50
- 2) Batch Size = 128
- 3) Rate = 0.001
- 4) Sigma = 0.1

6. Training, validation and test

- 1) The Validation accuracy is : 95.5%
- 2) The Test accuracy is : 93.6%

7. Testing with Web images

Few German traffic signs were downloaded from web. Details are as below



The output accuracy and probabilities are as shown below.

