Traffic Sign Recognition

The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Rubric Points

Data Set Summary & Exploration

1. Provide a basic summary of the data set.

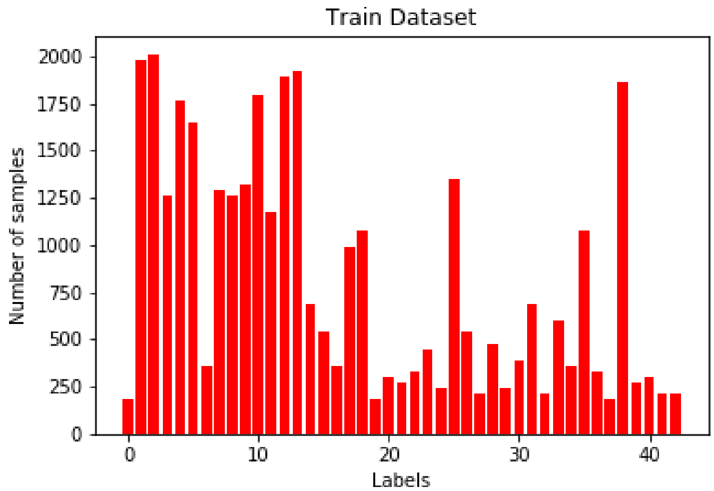
   I used the Numpy library to calculate summary statistics of the traffic signs data set:

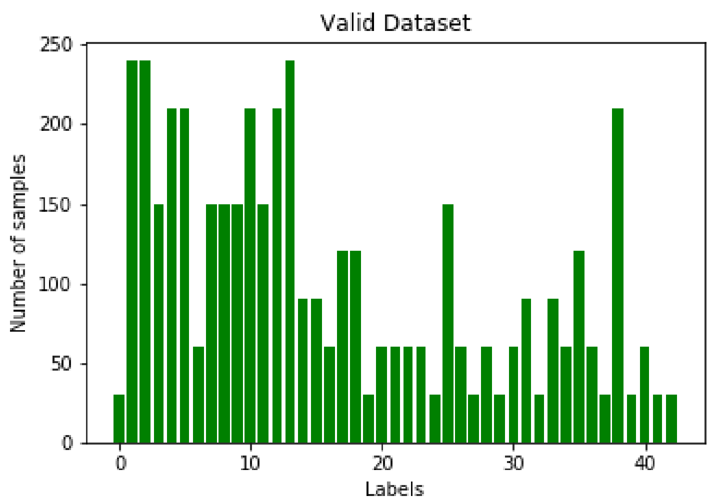
- The size of training set is: (34799, 32, 32, 3)
- The size of the validation set is: (4410, 32, 32, 3)
- The size of test set is: (12630, 32, 32, 3)
- The shape of a traffic sign image is: (32, 32, 3)
- The number of unique classes/labels in the data set is: 43

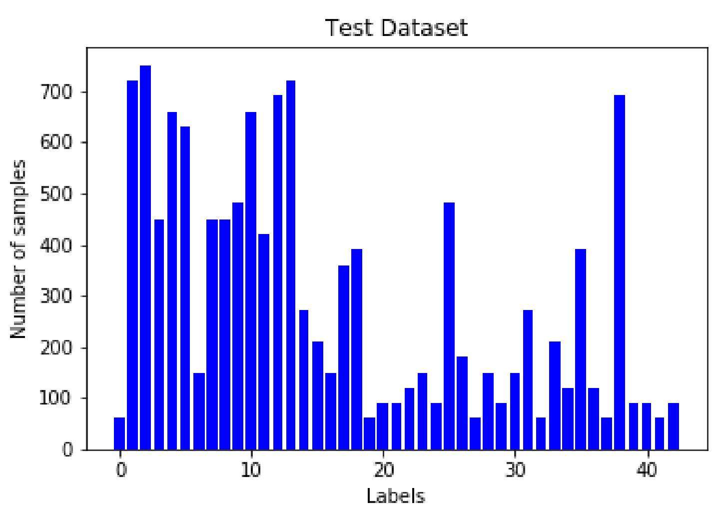
2. Include an exploratory visualization of the dataset.

- Here is an exploratory visualization of the data set.

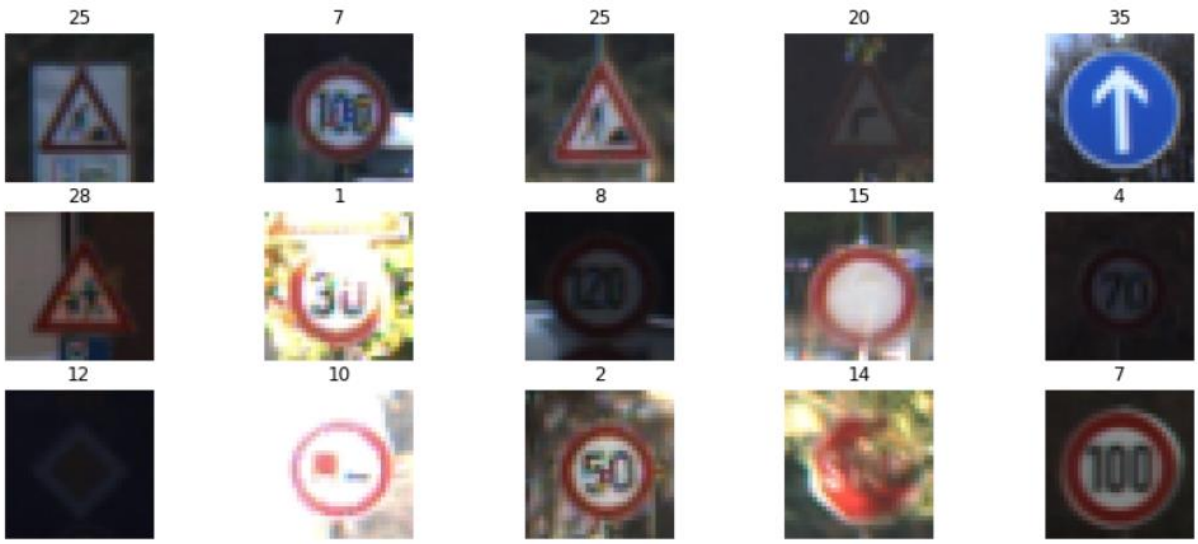
  There are 3 bar graphs showing the distribution of the data in 43 different labels of the dataset. The 1st, 2nd and the 3rd graphs show the training, validation and test dataset respectively.







Here are 15 random image from the training dataset along with their labels.



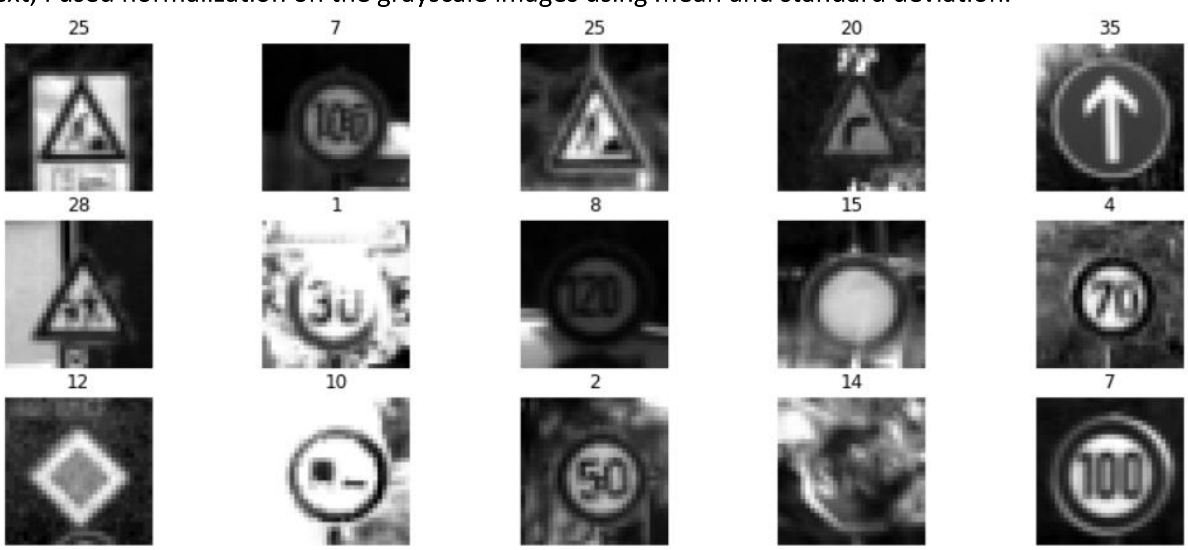
Design and Test a Model Architecture

1. Preprocessing the Data:

Initially, I tried to run the model directly on the RGB 3 channel images directly from the dataset. However, the accuracy did not come up to the mark.

So, I converted the images from RGB format to grayscale format. This decreased the loss and improved the accuracy a lot. The possible reason of that could be because the excess information leads to extra confusion in the learning.

Next, I used normalization on the grayscale images using mean and standard deviation.



2. Model Architecture:

My final model consisted of the following layers:

Layer	Description	Output
Input	32x32x1	
Convolution 3x3 Filters = 10	1x1 stride, Valid padding	30x30x10
RELU		
Convolution 3x3 Filters = 20	1x1 stride, Valid padding	28x28x20
ReLU		
Max Pooling	2x2 stride	14x14x20
Dropout	50% keep probability	
Convolution 3x3 Filters = 40	1x1 stride, Valid padding	12x12x40
ReLU		
Dropout	50% keep probability	
Convolution 3x3 Filters = 80	1x1 stride, Valid padding	10x10x80
ReLU		
Max Pooling	2x2 stride	5x5x80
Dropout	50% keep probability	
Flatten		2000
Fully Connected	Input = 2000	400
ReLU		
Dropout	50% keep probability	
Fully Connected	Input = 400	120
ReLU		
Fully Connected	Input = 120	43

3. Training the Data:

At first, I started with the LeNet architecture as given, but then I changed it with adding an extra layer of convolution layer. Also, instead of using the 5x5 convolution filter, I used 3x3 filter in each layer of the convolution. The details of the model are shown above in the table.

For the optimization process, I used the Adam optimizer and it worked fine for me.

Batch Size: 128

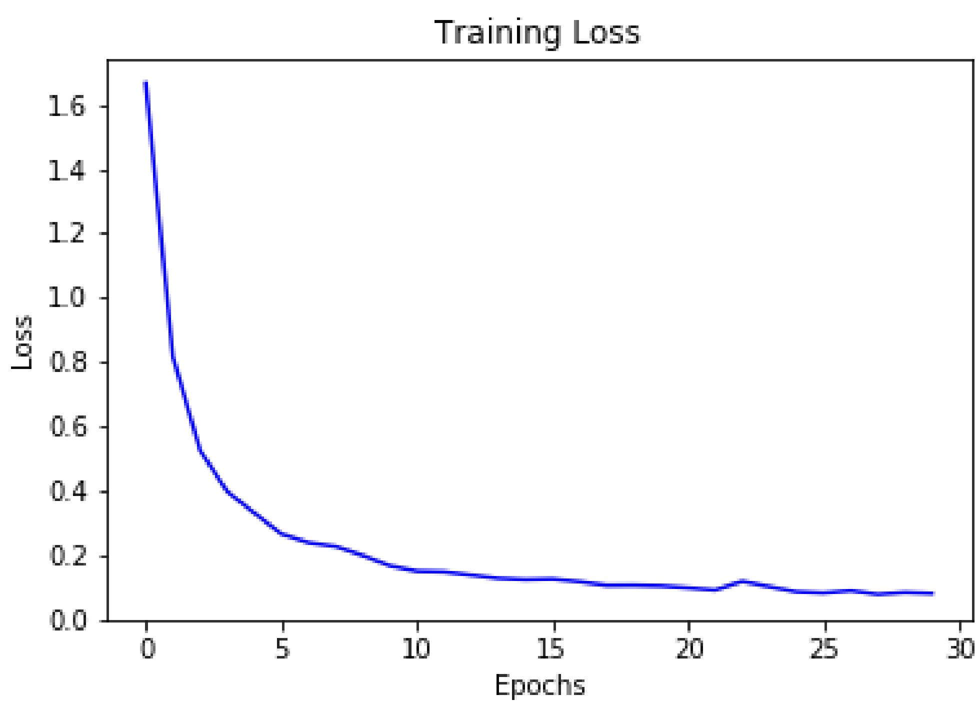
EPOCHS: 30

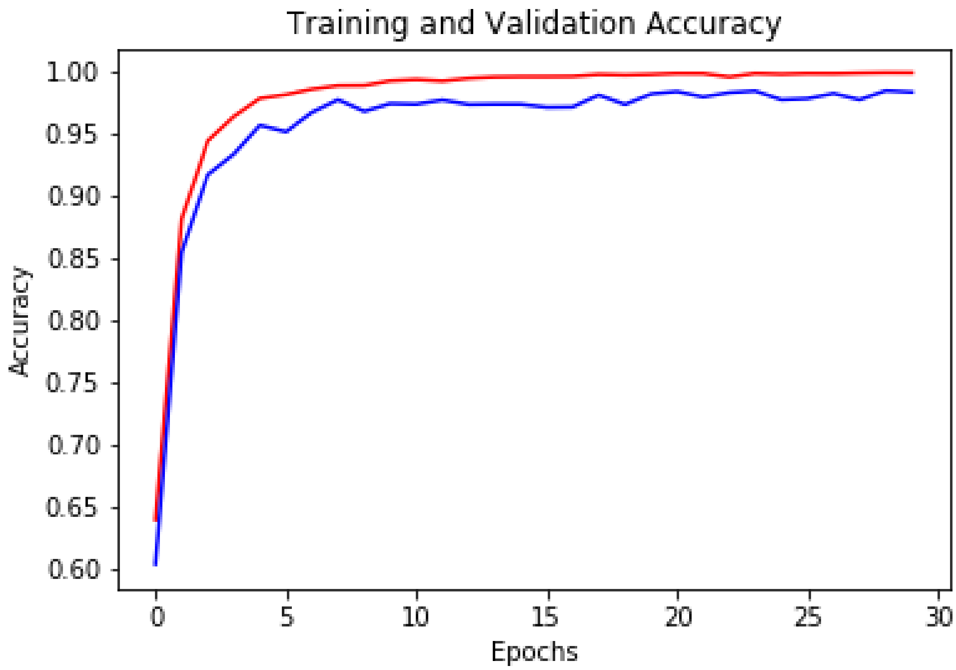
Learning Rate: 0.0011

4. Approach taken for finding a solution and getting validation set accuracy to be at least 0.93:

My final model results are:

- training set accuracy of: 0.999
- validation set accuracy of: 0.983
- test set accuracy of: 0.972





As mentioned earlier, I began with the LeNet architecture. It was all trial and error process. I tried training the data using the LeNet, however, the accuracy did not reach the expectations. So, I started using tweaking the architecture of LeNet and added another convolutional layer. I also added dropout layers in the convolution as well as fully connected layers. The major change which I did was to use 3x3 filters in the convolutions. Using this model, I was able to get the validation accuracy to approximately 97%. Epochs, learning rate and batch size and dropout probability were all set by trial and error by visualizing the training loss curves.

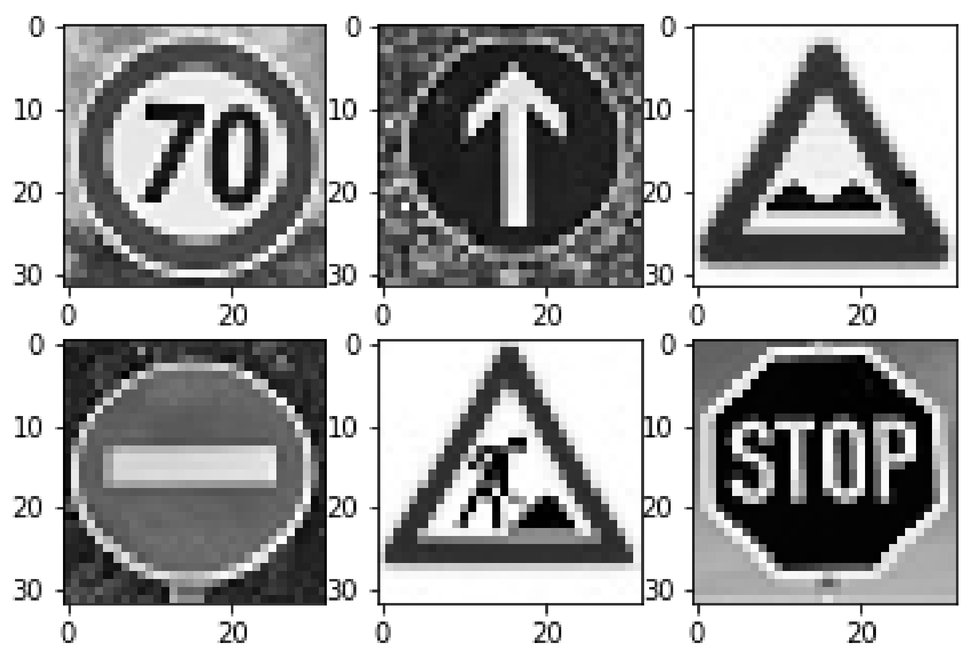
Test a Model on New Images

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 six German traffic signs that I found on the web:

The 1st 2nd 4th and 6th image might be a little difficult to classify because they have background noise.

Preprocessing of the image:



2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set.

Here are the results of the prediction:

Image	**Prediction**
Speed limit (70km/h)	Speed limit (70km/h)
Ahead only	Ahead only
Bumpy Road	Bumpy Road
No entry	No entry
Road work	Road work
Stop	Stop

The model was able to give accurate predictions for all the newly 6 images as shown above, 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.

For the first image, the top five soft max probabilities were:

Probability	**Prediction**
0.999677062	4
1.92583830e-04	1
1.12182512e-04	0
1.51593067e-05	2
7.79633126e-07	15

For the 2nd image, the top five soft max probabilities were:

Probability	Prediction
0.999997616	35
1.95363805e-06	36
2.09032137e-07	34
1.17683904e-07	3
7.79265719e-08	12

For the 3rd image, the top five soft max probabilities were:

Probability	Prediction
0.999965429	22
3.45447297e-05	29
7.02063616e-08	28
1.30251250e-08	31
4.19084456e-09	18

For the 4th image, the top five soft max probabilities were:

Probability	Prediction
0.999998689	17
9.96259587e-07	9
2.45354130e-07	40
5.18857917e-08	34
2.02538786e-08	12

For the 5th image, the top five soft max probabilities were:

Probability	Prediction
0.999891639	25
3.78689547e-05	30
2.65996550e-05	24
2.02968313e-05	31
1.00123170e-05	29

For the 6th image, the top five soft max probabilities were:

Probability	Prediction
0.999930501	14
2.82507954e-05	3
1.22250431e-05	38
6.47725938e-06	5
4.38485267e-06	34