Build a Traffic Sign Recognition Project

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

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

You're reading it! and here is a link to my project code

Data Set Summary & Exploration

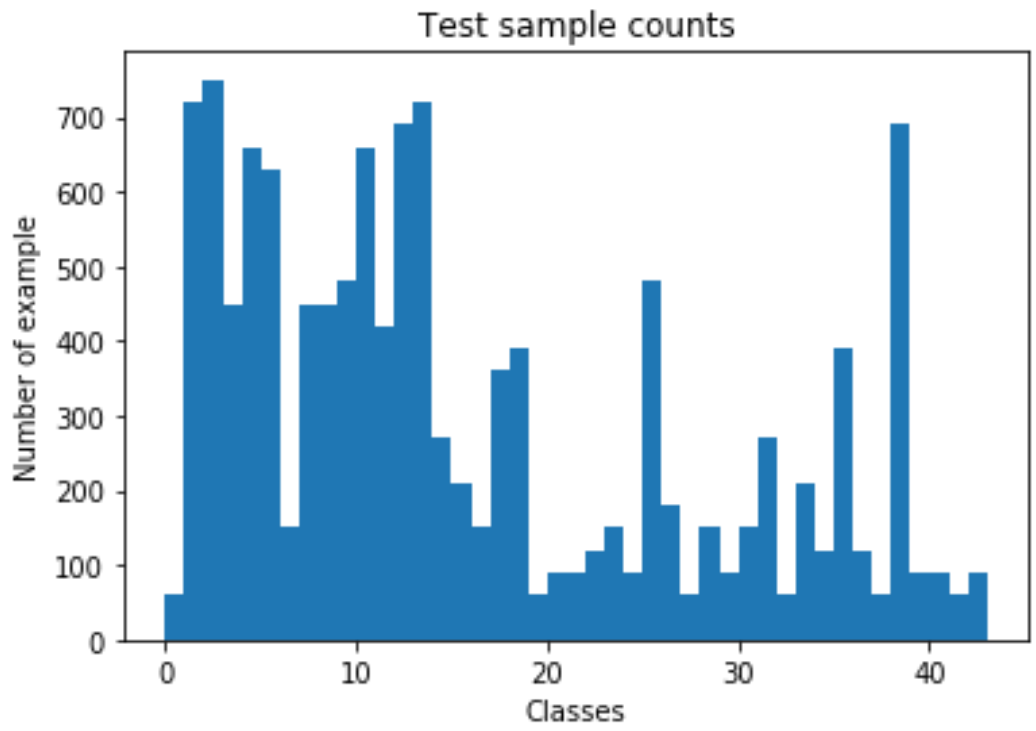
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

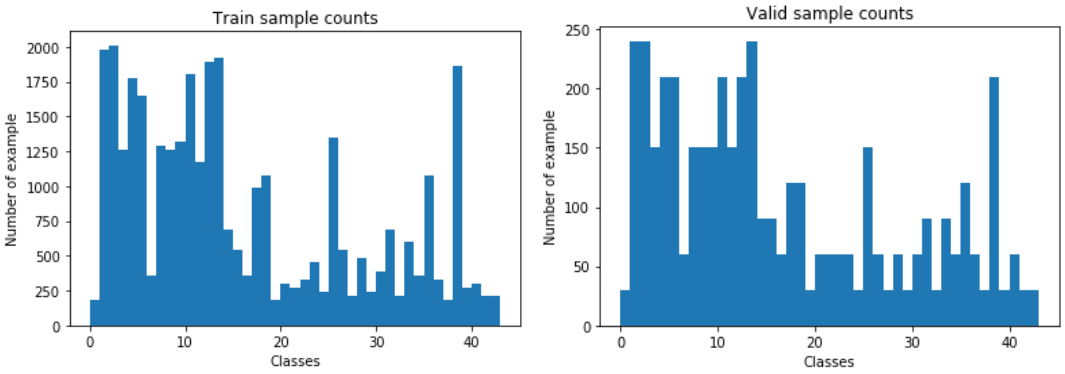
I used the pandas library to calculate summary statistics of the traffic signs data set:

- The size of training set is 34799
- The size of the validation set is 12630
- The size of test set is 4410
- 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. It is a bar chart showing how the data ...





We can see in the data set some of class have lower data and some of higher, this may affect the training for some class.

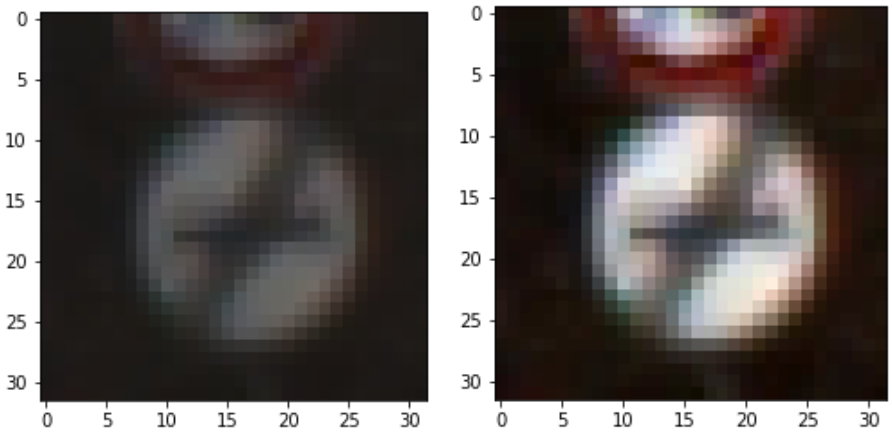
Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques?

First step Normalization

- We'd like in this process for each feature to have a similar range so that our gradients don't go out of control.

- Below the the image before and after normalization



Next step, I decided to convert the images to grayscale because the traffic sign is not dependent on color of image. So, we can ignore the color of image and it save some amount of processing time.

Last Step additional of data (Note - tried but not in final training)

- Some of the classes have less number of data as compare to other. I add additional data by rotating the and blurring the image. When I try to train the model somehow accuracy goes down. So in final training i avoid the additional data and train the model by given data

My final model consisted of the following layers:

Input	(32,32,1) gray image
Convolutional (5,5)	1x1 stride, valid padding, outputs 28x28x6
RELU	
Max_ Pooling	Stride (2,2), valid padding, outputs 14x14x6
Convolutional	1x1 stride, same padding, outputs 10x10x16
RELU	
Pooling	Stride (2,2), valid padding, outputs 5x5x16
Flatter	Ouptputs – 400
Fully_Connected 1	Outputs 120

Relu	
Fully Connected 2	Outputs – 84
Relu	
drop_out 2	Drop unit – 0.5
Fully Connected 3	Outputs - 43
Softmax	

My final model results were:

- validation set accuracy of 94.6%
- test set accuracy of 91.6 %

To train the model, I used an ....

- I used the le-net model with some modification on it. First I get 86-89% accuracy in validation set. Then I try some modification …

- I changes the learning rate (increase and decrease) but 0.001 is the best fit.

- Next I normalize the data, (get accuracy improve 88-90%)

- I generate some date from given training set but rotating the image and adding bightness but some how I get low accuracy(I think I made same mistake in code, that's why accuracy not improve) So I drop this idea.

- Next I made modification in lenet model, I added dropout layer before last layer

  Get accuracy improvement. Then I add 2 more drop_out layer but the accuracy level not improve. I think 1st layer remove over fitting the data again adding more dropout layer may increase the data loss that's why accuracy is not improving by adding more drop_out layer.

- I again modified the data rgb to gray scale. It works accuracy improve to 91-93%

- Then last I play with number of epocs I find that 50 in best case accuracy 95%

Increasing epoces means more training with same date, so model learn more and accuracy improves.

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:

   these are five image I choose.

   1. **Simple Image**
   2. **Lower data in training set**
   3. **Rotated image**
   4. **Image taken from Left side**
   5. **Simple image sample**

2. 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. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

Image	**Prediction**
Stop Sign	Stop Sign
Yield	Yield
Pedestrian	General caution(wrong prediction)

Road_Work	Road_Work
Speed Limit 30	Speed Limit 30

Stop Sign **Road Work** **Speed limit 30**

Probability	Prediction	Probability	Prediction	Probability	Prediction
.95	Stop sign	1	Road work	.99	Speed 30
.02	Speed limit (60km/h)	Very low	Road narrows on the right	.001	Speed 20
.015	Priority road	Very low	Beware of ice/snow	.0002	Speed 80
.004	No vehicles	Very low	Bumpy road	Very low	Keep right
.0004	Turn right ahead	Very low	Bicycles crossing	Very low	End of speed 80

Yield **Pedestrian**

Probability	Prediction	Probability	Prediction
1.0	Yield	.99	General caution
Very low	Road work	Very low	Dangerous curve to the right
Very low	Road work	Very low	Road work
Very low	Speed limit (50km/h)	Very low	Keep right
Very low	Speed limit (30km/h)	Very low	Bumpy road

```
array([[ 1,  0,  5, 38,  6], class-type of image - 1
       [18, 20, 25, 38, 22], class-type of image – 27 (wrong prediction)
       [25, 24, 30, 22, 29], class-type of image - 25
       [14,  3, 12, 15, 33], class-type of image - 14
       [13, 25, 38,  2,  1]])) class-type of image – 13
Accuracy on five image is 80%
```

All Done
Thank You,
Sandip Kumar