Traffic Sign Recognition

python 3.6
Udacity CarND

In this project we built a traffic sign classifier based on the German Traffic Sign Dataset using a deep neural network trained in TensorFlow.

The goals / steps of this project are the following:

1) Learn how to Explore, summarise and visualise the data set

2) Learn to Design, train and test a model architecture

3) Use the model to make predictions on new images and analyse.

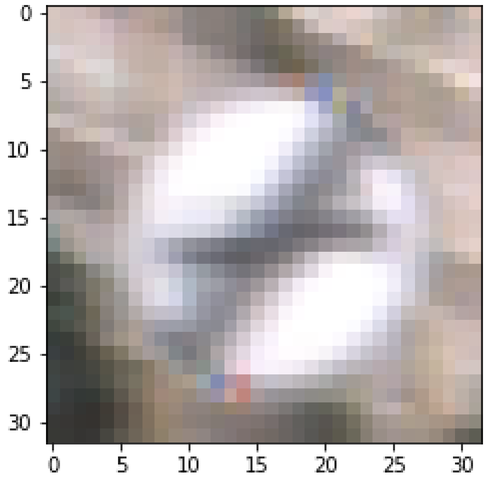
1. Examine the data set

The data set used for this project is German Traffic Sign Dataset

Data set	Number of samples
Training	34,799
Validation	4,410
Test	12,630
Unique classes	43

Every image has a dimension of **32 x 32 x 3** (width, height, channels).

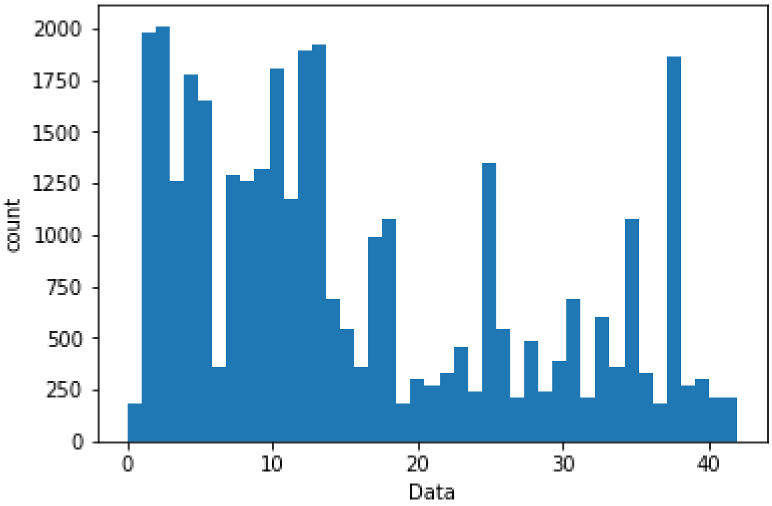
Sample traffic sign image looks like



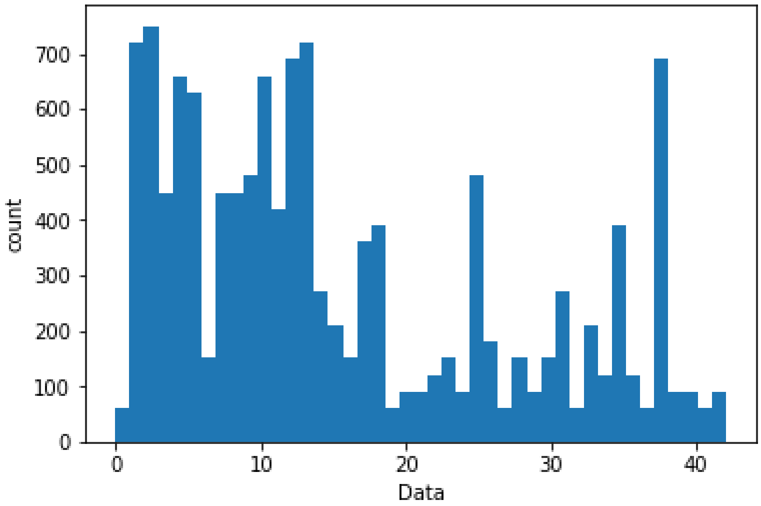
2. Data set distribution analysis

Data set distribution is not uniform,

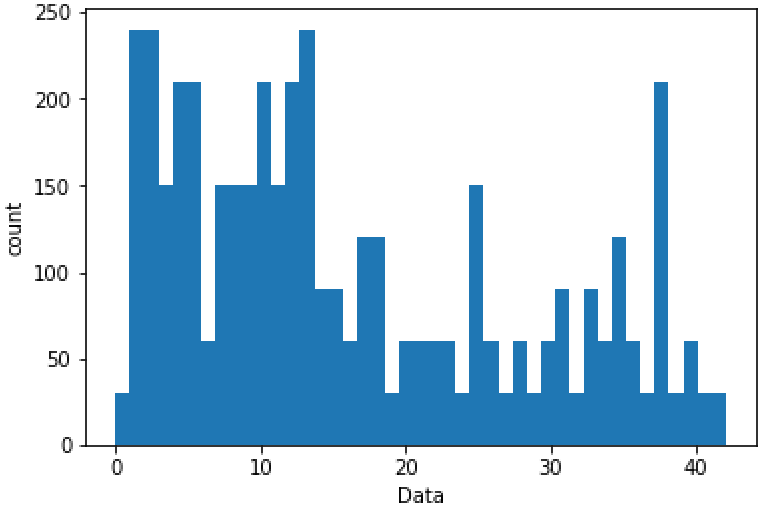
Train data set histogram



Test data set histogram



Valid data set histogram



3. Pre process

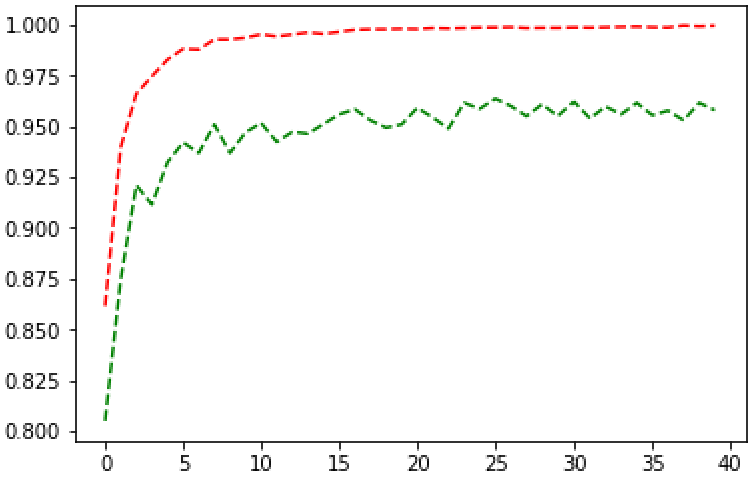
Converted the images to **grayscale** and **normalize** the image as this will help the model to converge faster.

4. Model

My final model consisted of the following layers:

Layer	Description
Input	32x32x1 grayscale image
Convolution	1x1 stride, valid padding, outputs 28x28x6
Activation	RELU
Pooling	2x2 stride, valid padding outputs 14x14x6
Convolution	1x1 stride, valid padding, outputs 10x10x16
Activation	RELU
Pooling	2x2 stride, valid padding outputs 5x5x16
Flatten	Output = 400
Fully Connected	Output = 120
Activation	RELU
Fully Connected	Output = 84
Activation	RELU
Fully Connected	Output = 43

5. Accuracy



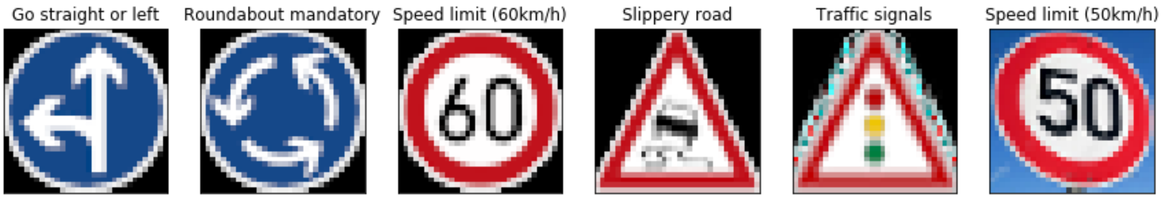
Parameter used for the training are mentioned below

```
training_keep_prob = 0.69
rate = 0.001
EPOCHS = 40
BATCH_SIZE = 30
optimizer = AdamOptimizer
```

My final model results were:

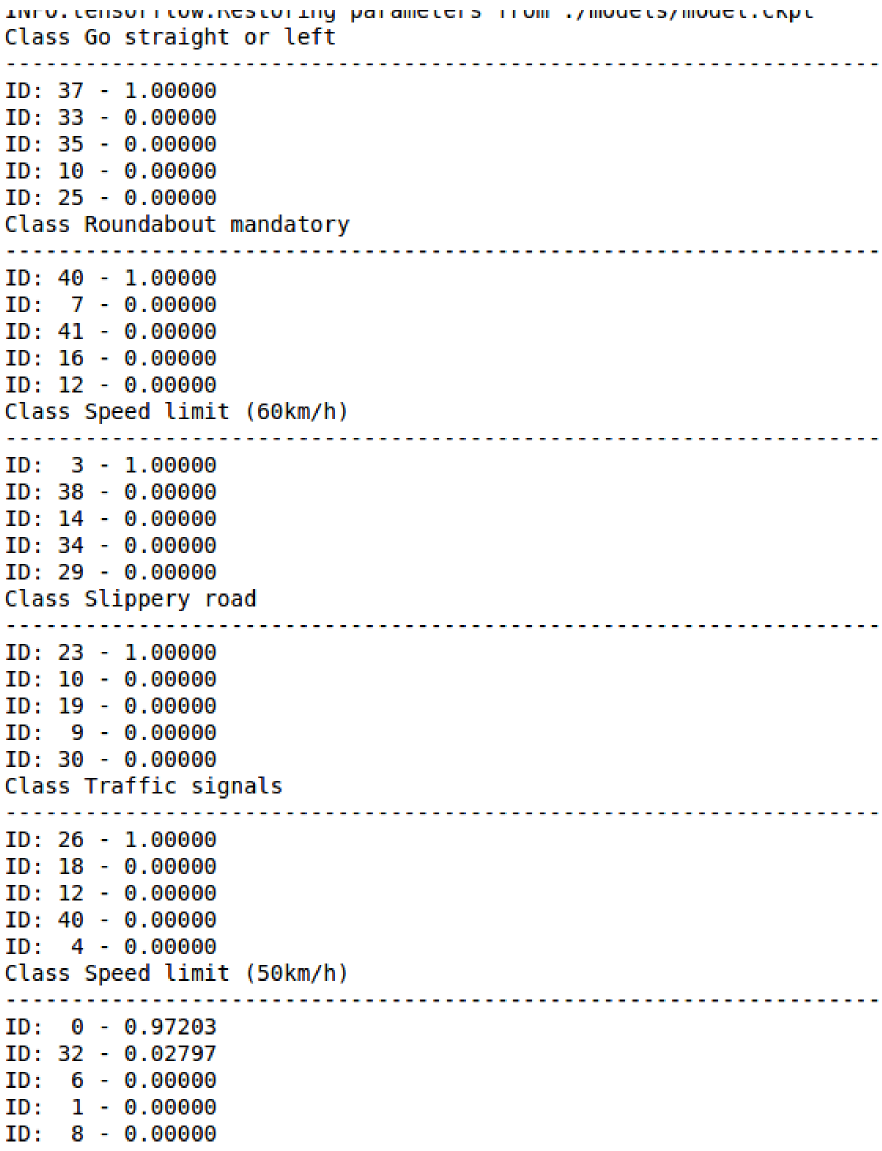
```
accuracy for Training set: 1.000
accuracy for Validation set: 0.958
accuracy for Test set: 0.943
```

Test images used



Models were able to correct 5 images out of 6 which gives an accuracy of 83.3%

Here are the top five soft max probabilities for these images:



```
INFO:tensorflow:Restoring parameters from ./models/model.ckpt
Class Go straight or left
------------------------------------------------------------------
ID: 37 - 1.00000
ID: 33 - 0.00000
ID: 35 - 0.00000
ID: 10 - 0.00000
ID: 25 - 0.00000
Class Roundabout mandatory
------------------------------------------------------------------
ID: 40 - 1.00000
ID:  7 - 0.00000
ID: 41 - 0.00000
ID: 16 - 0.00000
ID: 12 - 0.00000
Class Speed limit (60km/h)
------------------------------------------------------------------
ID:  3 - 1.00000
ID: 38 - 0.00000
ID: 14 - 0.00000
ID: 34 - 0.00000
ID: 29 - 0.00000
Class Slippery road
------------------------------------------------------------------
ID: 23 - 1.00000
ID: 10 - 0.00000
ID: 19 - 0.00000
ID:  9 - 0.00000
ID: 30 - 0.00000
Class Traffic signals
------------------------------------------------------------------
ID: 26 - 1.00000
ID: 18 - 0.00000
ID: 12 - 0.00000
ID: 40 - 0.00000
ID:  4 - 0.00000
Class Speed limit (50km/h)
------------------------------------------------------------------
ID:  0 - 0.97203
ID: 32 - 0.02797
ID:  6 - 0.00000
ID:  1 - 0.00000
ID:  8 - 0.00000
```

Reflection¶

The whole project was a great learning experience and I learned a lot about neural networks and deep learning; I have found it very interesting and bit difficult.. Looking forward to work on the next project.

There are different augmentation technique to be apply, and need to play around with more noisy data set like classifying the sign during rainfall or heavy snow.