Traffic Sign Classifier

1. **Purpose :**
   The purpose of the project is to use the Neural network architecture for Traffic sign classification.

2. **Project Inputs :**

   i. Following inputs are provided by Udacity for development and analysis of Neural network architecture –
      a) Training data (train.p)
      b) Test data (test.p)
      c) Validation data (valid.p)

   ii. Following inputs are taken from internet -
      a) Test images in folders (traffic-signs-new*)

3. **Project Outputs :**

   i. To satisfy project rubric following artifacts are expected –
      a) Ipython notebook with code
      b) HTML output of the code
      c) A writeup report (either pdf or markdown)

   ii. To satisfy project rubric following outcome is expected –
      a) Validation set accuracy of 0.93 or greater

   iii. To satisfy project rubric following explanation is expected –

a) Dataset Exploration

- Dataset Summary - The submission includes a basic summary of the data set.

- Exploratory Visualization - The submission includes an exploratory visualization on the dataset.

b) Design and Test a Model Architecture

- Preprocessing - The submission describes the preprocessing techniques used and why these techniques were chosen.
- Model Architecture - The submission provides details of the characteristics and qualities of the architecture, including the type of model used, the number of layers, and the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.
- Model Training - The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyper parameters.
- Solution Approach - The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

C) Test a Model on New Images

- Acquiring New Images – The submission includes five new German Traffic signs found on the web, and the images are visualized. Discussion is made as to particular qualities of the images or traffic signs in the images that are of interest, such as whether they would be difficult for the model to classify.
- Performance on New Images - The submission documents the performance of the model when tested on the captured images. The performance on the new images is compared to the accuracy results of the test set.
- Model Certainty - Softmax Probabilities - The top five Softmax Probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

4. Description :

i. Dataset Exploration –

The Dataset consisted of a Training file, a Validation file and a Testing file. The content of data set consisted of German Road Traffic signs which was divided in to “**43 classes”** of **32x32x3** RGB image.

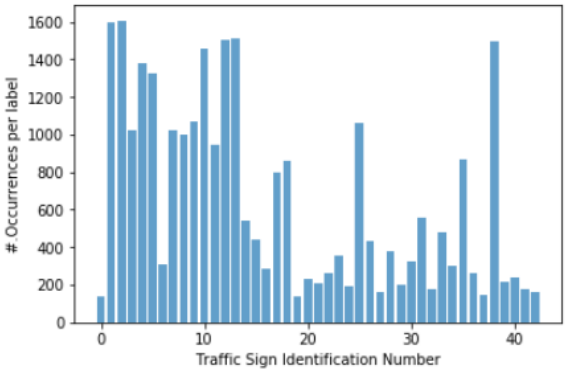
Following is the summary of dataset:

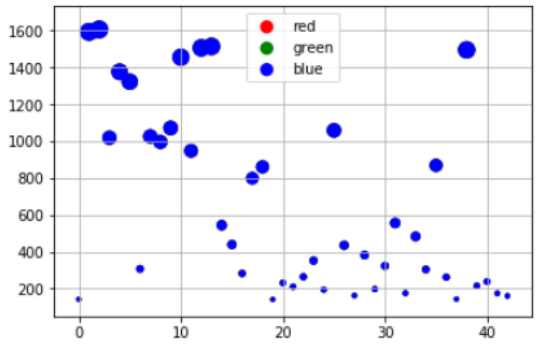
```
Number of training examples = 27839
Number of validation examples = 4410
Number of testing examples = 12630
Image data shape = (32, 32, 3)
Number of classes = 43
```

Following visualization provides information of the Traffic signs :

```
Traffic Sign count:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
33, 34, 35,36, 37, 38, 39, 40, 41, 42]
Traffic Sign ID:  [3, 15, 22, 41, 17, 40, 35, 2, 38, 11, 29, 7, 40, 34,
3, 10, 38, 2, 0, 12, 35, 18, 26, 9, 2, 12, 3, 23, 15, 6, 13, 10, 38, 22,
13, 20, 34, 5, 25, 9, 4, 25, 12]
```

The count of each observation is plotted with the help of Bar chart and Scatter chart. The difference between the Bar chart and Scatter chart is that the Bar chart provides a very clear correlation between the classes and the number of samples. But Scatter chart is little bit difficult in this regard.





ii. Preprocessing –

Grayscaling – For preprocessing I followed two steps. First one Is the Grayscaling. I used the same CV function *cv2.cvtColor()* was also used in Advanced Lane Finding project. The Grayscale images are used to reduce the complexity of model by focusing on the shape of the traffic sign rather than on other attributes like color, contrast etc.

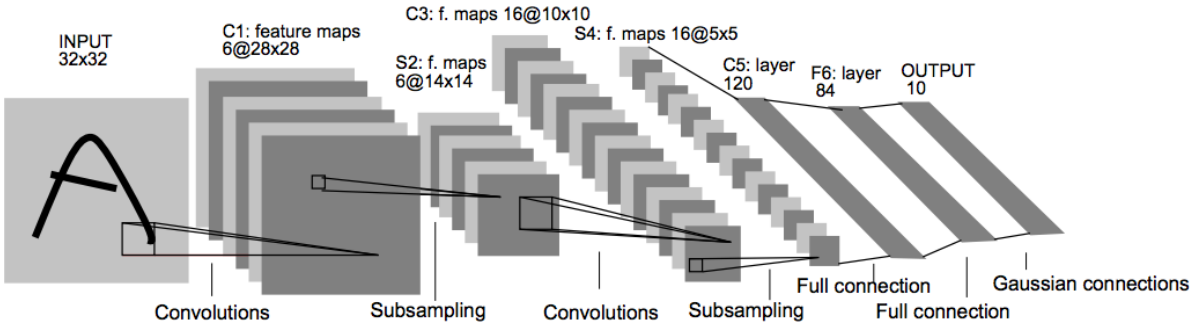
Normalization - After Grayscale conversion image data is normalized With Equation *(pixel - 128)/ 128*. The image data should be normalized so that the data has mean zero and equal variance.

Resize – The images are resized to *32x32* Grayscale images.

All the above steps are performed by calling the function *normalize_grayscale().*

iii. Model Architecture -

I used *5 Layer LeNet* for Model Architecture as shown below –



Source: Yann LeCun

The LeNet architecture is an excellent “first architecture” for Convolutional Neural Networks (especially when trained on the MNIST dataset, an image dataset for handwritten digit recognition).

LeNet is small and easy to understand — yet large enough to provide interesting results. Furthermore, the combination of LeNet + MNIST is able to run on the CPU, making it easy for beginners to take their first step in Deep Learning and Convolutional Neural Networks.(Reference from net)

Following are the specifications of architecture :

Layer	Description
Input	32x32x1 Grayscale image
Convolution	5x5 filter with 1x1 stride, valid padding, outputs 28x28x6
RELU	
Convolution	5x5 filter with 1x1 stride, valid padding, outputs 14x14x6
RELU	
Convolution	5x5 filter with 1x1 stride, valid padding, outputs 10x10x16
RELU	

Max Pooling	2x2 ksize with 2x2 stride, valid padding, outputs 5x5x16
Flatten	outputs 400
Fully Connected	Input 400 and outputs 120
RELU	
Dropout	keep_prob=0.5
Fully Connected	Inputs 120 and outputs 84
RELU	
Fully Connected	Inputs 84 and outputs 43

iv. Model Training -

I trained the model on the Training data set of German traffic signs. I tried with different training rates of 0.001, 0.0009 and 0.00085. Also I tried with three different EPOCH 10, 50 and 75. The selected batch size is 128. The best Validation accuracy I received from training rate of 0.00085 for 75 EPOCH. I used the default mu =0 and Sigma = 0.1 as used in tutorials. However I also observed that sometimes the Validation accuracy reached to maximum around 65 EPOCH and then not much improvement later on.So there is a case of Overfitting for large number of EPOCH. But I was able to attain the Validation Accuracy of more than 0.93 as mentioned in project rubric.

Parameter	Value
EPOCH	75
BATCH_SIZE	128
Training rate	0.00085
mu	0
sigma	0.1
OUTPUT	
Validation Accuracy	0.940

```
EPOCH 64 ...
Validation Accuracy = 0.930

EPOCH 65 ...
Validation Accuracy = 0.940

EPOCH 66 ...
Validation Accuracy = 0.936

EPOCH 67 ...
Validation Accuracy = 0.936

EPOCH 68 ...
Validation Accuracy = 0.935

EPOCH 69 ...
Validation Accuracy = 0.934

EPOCH 70 ...
Validation Accuracy = 0.935
```

```
EPOCH 71 ...
Validation Accuracy = 0.935

EPOCH 72 ...
Validation Accuracy = 0.934

EPOCH 73 ...
Validation Accuracy = 0.933

EPOCH 74 ...
Validation Accuracy = 0.935

EPOCH 75 ...
Validation Accuracy = 0.933

Model saved
```

v. Solution Approach –
As mentioned in preprocessing section, I converted the RGB Images to Gray scale. To adapt to this I made changes in the architecture to accept images with single channel instead of three channels.
I started with the number of EPOCH = 10 and training rate of 0.001.But with this I was not able to get the Validation accuracy greater than 0.93. Then I tried with increasing the EPOCH and changing the training rates. Finally I settled with EPOCH = 75 and training rate of 0.00085 to get Validation accuracy more than 0.93.

vi. Acquiring New Images and Performance evaluation –
I downloaded four sets of 5 images. First two sets of images are taken from the German Traffic sign database.

- First set of images -

For this data set I got a Test accuracy of 1.0.

- Second set of images –

The second set of 6 images are also taken from the German Traffic sign database.

All the images are identified accurately with Test accuracy of 1.0.

- Third set of images –

For this set the accuracy was reduced very much. The model Identifies the first two images correctly but for next four images It makes wrong predictions. The Test accuracy for this set is 0.333. The interesting thing I observed here is that the Traffic signal(#26) is identified as General Caution(#18). And the Bicycle crossing(#29) is identified as Round about compulsory (#40).

- Fourth set of images –

The fourth set of images I collected randomly from internet. The Purpose is to analyze if the General caution(#18), Traffic signs (#26) and Round about(#40) are identified correctly by the model or not. Unfortunately the model failed completely for this set of images.

vii. Softmax Probabilities –

For the first set of images the Softmax probabilities are accurate. The model recognized the listed images with probability of ‘1’.

```
img_folder = 'traffic-signs-new/'
#my_images = np.zeros((5,32,32,3))
my_labels = [35, 20, 18, 25, 1]   #Defining Labels
```

```
Actual Label: 20
Output Label: 20
Softmax Logits: [[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]]
```

For the fourth set of images, the probabilities are incorrect. The Model predicted the wrong images with confidence of ‘1’.

```
img_folder = 'traffic-signs-new4/'
#my_images = np.zeros((5,32,32,3))
my_labels = [18, 18, 18, 18, 18, 26, 26, 26, 26, 26, 40, 40]    #Defining Labels
```

```
Actual Label: 18
Output Label: 12
Softmax Logits: [[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
   0.  0.  0.  0.  0.  0.  0.]
```

5. Conclusion :

The model is working well with the Test images from German Traffic sign database and some random images. But it needs Improvement to identify traffic signs from different sources.