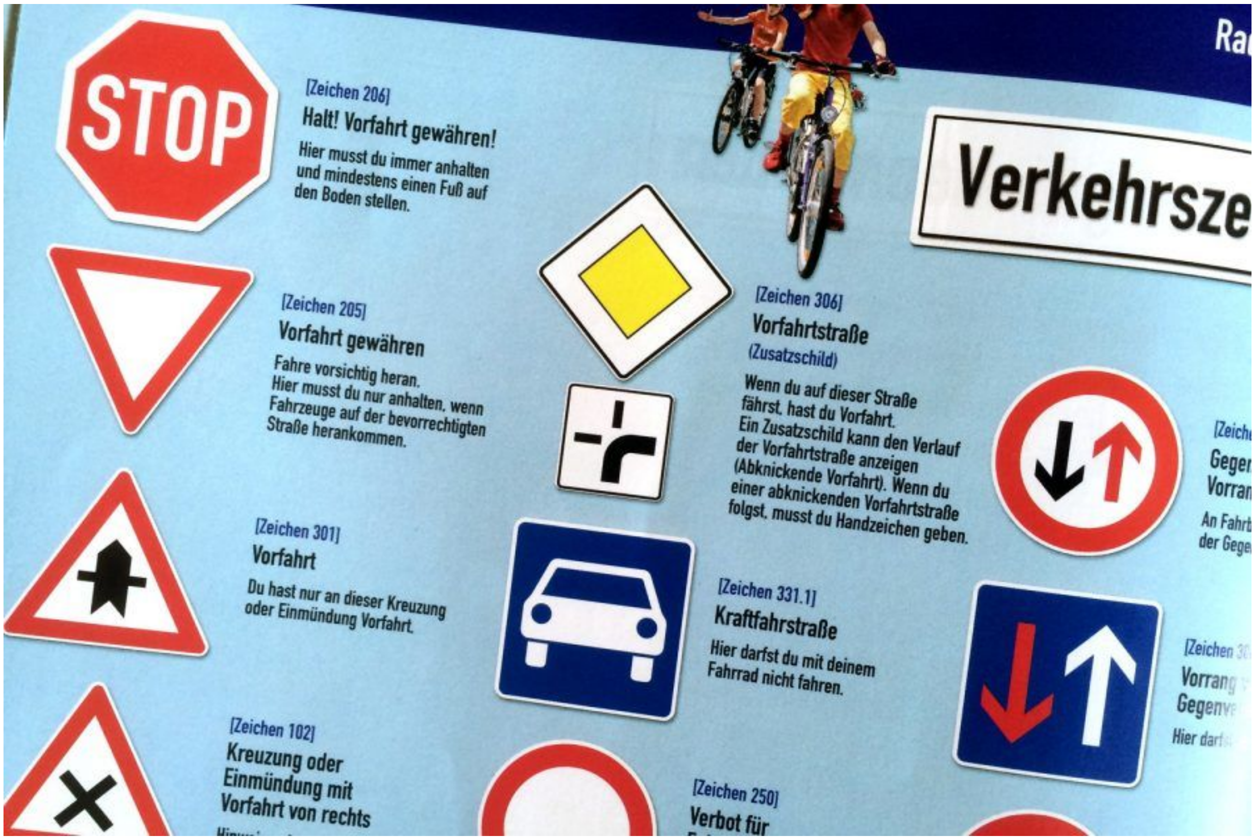
Self Driving Car Engineer Nanodegree
Udacity

Manikandan R
vasumaniram@gmail.com

German Traffic Sign Classifier - Project 3



Objective

The objective of the project is to train a CNN classifier based on LeNet Architecture to classify the German Traffic Signs and the following are the goals to achieve the same.

- Load the 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

Assumptions

- The training, validation and test datasets are available.
- Every image in training, validation and test datasets are in (32 X 32 X 3) dimensions.

1.Data Exploration:

1.1 Dataset Summary:

The following table summarizes the size of training,validation and test dataset,number of classes and the dimension of the image.

Number of training examples	34799
Number of validation examples	4410
Number of testing examples	12630
Image data shape	(32, 32, 3)
Number of classes	43

Table - 1.1

From the above summarization, we could infer the following,

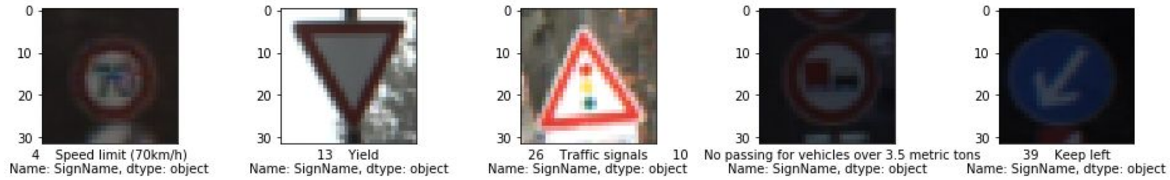
1. The size of the total dataset is 51839
2. The *train:validation:test* split percentage is *67:9:24*
3. The images are color images with 3 color channels.
4. There are 43 different traffic signs available.

1.2 Exploratory Visualization:

1.2.1 - Random Sample Images: The following section depicts the random traffic sign images plotted from training,validation and testing dataset distribution.

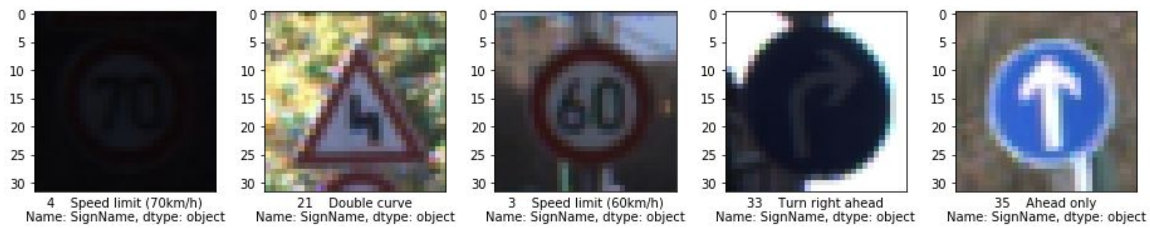
Random Sample Images from Training Dataset

```
visualize_images(X_train,y_train)
```



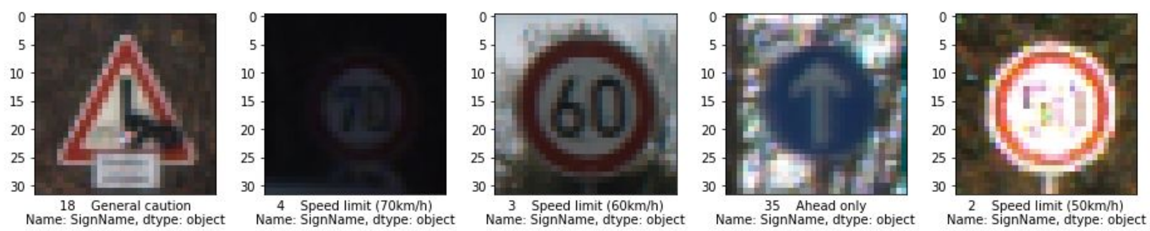
Random Sample Images from Validation Dataset

```
visualize_images(X_valid,y_valid)
```



Random Sample Images from Testing Dataset

```
visualize_images(X_test,y_test)
```

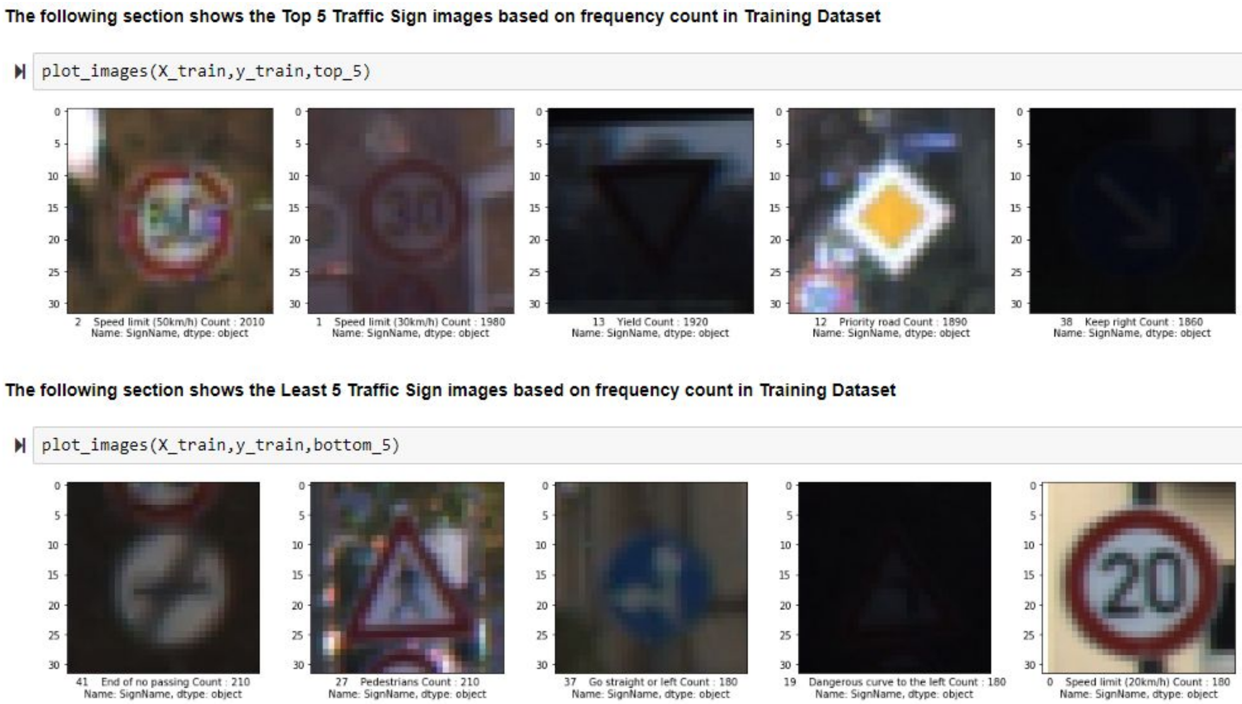


Picture - 1.2.1

There are images with less or zero clarity which will make it difficult for the CNN to train and predict.

1.2.2 - Top and Bottom Five Images:

The following section depicts the top 5 and bottom 5 traffic signs from training dataset based on the frequency of occurrences.



Picture - 1.2.2

Top 5	**Frequency**	**Bottom 5**	**Frequency**
2 - Speed limit(50km/hr)	2010	41 - End of no passing	210
1 - Speed limit(30km/hr)	1980	27 - Pedestrians	210
13 - Yield	1920	37- Go straight or left	180
12 - Priority road	1890	19 - Dangerous curve to the left	180
38 - Keep right	1860	0 - Speed limit(20km/hr)	180

Table- 1.2.2 - Training Dataset

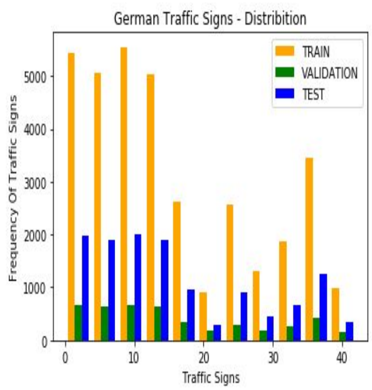
The traffic signs with more dataset will be trained by the model sufficiently where as the less dataset traffic signs will suffer from under training. Because Deep Learning is data hungry, the more data leads to performance of the model.

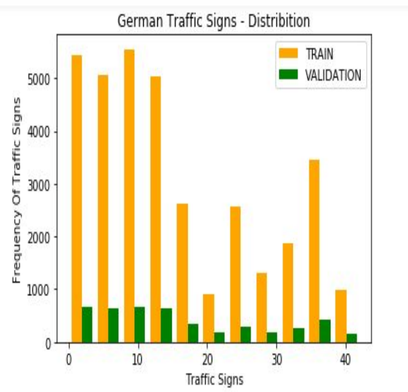
1.2.3 - Data Distribution Check:

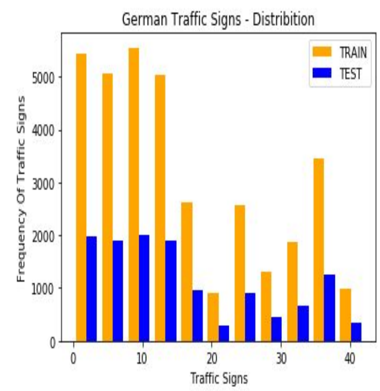
The training, validation and testing dataset should have the data from the same distribution. For example, if the validation or testing dataset has a traffic sign which does not exist in training data,then it will lead to poor performance of the model. Hence checked all the 43 classes are available in training, validation and testing datasets.

1.2.4 - Training,Validation, Testing Distribution:

The following section depicts the training, validation and testing distribution with respect to the frequency of each classes.







2.Design and Test a Model Architecture:

2.1 Preprocessing: Preprocessing is the technique used to tailor the input data for the algorithm to perform well. For the traffic sign classification, the input are of color images. There are multiple preprocessing techniques available for the images and out of which the following three have been considered,

1. Shuffling Dataset
2. Gray Scale
3. Normalization

Shuffling: It makes sure that the order of the dataset does not influence the model.

Grayscale: Most of the Computer Vision tasks can be done with the Gray Scale images alone. Color images have 3(RGB/HLS/HLV) channels where as grayscale images have a single channel which helps the algorithms to perform well in terms of time and space.

Normalization: It helps to scale down the image input data from 0..255 to 0..1. Deep Learning heavily dependent on Matrix Calculations and normalization also helps faster calculations. There are multiple ways to normalize and the applied normalization formula is,

$$Normalized\ Image\ =\ (Image\ -\ 128.0\ )/\ 128)$$

Note: 128.0, the decimal point was required to convert the operation into floating point operation.

2.2 Model Architecture:

The following table describes the Model Architecture,layers and parameters used to train the model.

Layer	**Description**
Input	32 X 32 X 3 RGB Image
Convolution 1 - 5 * 5	'VALID' Padding | 1 X 1 Stride | Filters - 16
RELU	
Max Pooling	"VALID' padding | 2 X 2 Stride
Convolution 2 - 5 * 5	'VALID' Padding | 1 X 1 Stride | Filters - 32
RELU	
Max Pooling	'VALID' padding | 2 X 2 Stride
Fully Connected 1	800 Inputs | 400 Outputs
Drop Out	Keep_prob : 0.7
Fully Connected 2	400 Inputs | 200 Outputs
Dropout	Kepp_prob: 0.5

Output - SoftMax	200 Inputs | 43 Output classes.

Table 2.2 - Model Architecture

For the German traffic sign classifier, there are quite a few more information for the model to learn and hence the filter sizes have been increased accordingly. Also the filter sizes have the impact on input nodes of fully connected layers for better training and generalization. Dropout regularization has been used in between the fully connected layers to make sure, the model does not overfit to the training examples and the accuracies of validation and test datasets proved the same,.

2.3 Model Training:

The following table summarizes the model Hyper Parameters used to train the model.

Hyper Param	**Value**
EPOCH	50
BATCH SIZE	128
LEARNING RATE	0.001

The objective is to achieve minimum of 93% validation accuracy. The model achieved the 93.3% at EPOCH 10 and the highest validation accuracy is 95.7%. BATCH_SIZE has been set to 128,so that the less memory machines can also able to train. The same model was able to train with a CPU machine in around 30 mins. As for as the learning rate is considered, the too aggressive learning rates like 0.01, started with very low validation accuracy and had tough time in generalizing further.

2.4 Solution Approach:

The model architecture was derived from the famous Lenet-5 CNN architecture for predicting the handwritten digits. The layering architecture is almost same except the few dropout layers for better generalization and there were changes in the parameters as there are different set of information to learn from the images for the model.

The following table summarizes the accuracy of validation and test datasets.

Highest Validation Accuracy	74.8
Highest Validation Accuracy	**95.7**
Test Accuracy	**94.4**

Table 2.4 - Validation and Test Accuracy

The less difference between the Validation Accuracy and Test Accuracy confirms that the model is not overfitted to training examples and generalizes well with the unseen examples.

3.Test a Model on New Images:

3.1 Acquiring New Images:

The following 5 German Traffic Sign images were downloaded from web to test the model. They are placed under the folder *./new_images*

Picture 3.1 - New German Traffic Sign Images

Extra Preprocessing: These downloaded images were in different sizes and all of them have been resized to 32 X 32 X 3 to fit the model.

3.2 Performance on New Images:

The following are the results of the prediction,

Image	Prediction	Result
Speed limit (50km/h)	Speed limit(80km/h)	FALSE
Speed limit(30km/h)	Speed limit(30km/h)	TRUE
Priority road	Priority road	TRUE
Speed limit (60km/h)	Speed limit (60km/h)	TRUE
Keep right	Keep right	TRUE

Picture 3.2 - New Images Prediction.

The model was able to predict 4 out of 5 new German Traffic Signs correctly with utmost certainty. It wrongly predicted 50km/h as 80km/h. The overall accuracy is 80%. With the less number of new test images(every image constitutes 20% of the accuracy), 80% is a good accuracy for this model even though the test accuracy was 94.4% with 12630 images.

3.3 Model Centainty - Softmax Probabilities:

The following table describes the Softmax probabilities obtained from the model for the new test images,

Image	Top 5 - SoftMax Probabilities
Speed limit (50km/h)	1.00000000e+00 1.80887902e-10 7.12835739e-15 5.55868760e-16 3.39245133e-17
Speed limit(30km/h)	1.00000000e+00 4.73600822e-38 0.00000000e+00

	0.00000000e+00 0.00000000e+00
Priority road	1. 0. 0. 0. 0.
Speed limit (60km/h)	1.00000000e+00 5.54046160e-08 1.41625289e-08 5.93922522e-10 9.95845073e-12
Keep right	1.00000000e+00 1.41335187e-27 0.00000000e+00 0.00000000e+00 0.00000000e+00

The model was utmost certain in predicting the results.

4.Further Enhancements:

- Deep Learning models like CNN require more data to generalize well.There a quite a few traffic signs, which lacks enough number of examples to train well. One of the image preprocessing techniques called 'Image Augmentation' could further be used to increase the number of examples to train. The same technique can be used for other traffic signs too for a better accuracy.
- Other model architectures like VGG Net, Alex Net could be implemented to test and compare the accuracy results.