

Project: Build a Traffic Sign Recognition Classifier

-Vidit Shah

In this write up I will explain my code and Architecture which was used to develop this model.

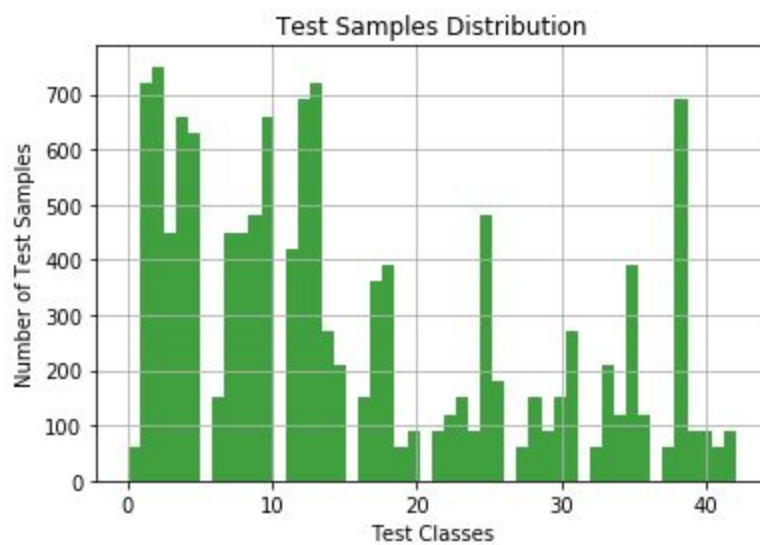
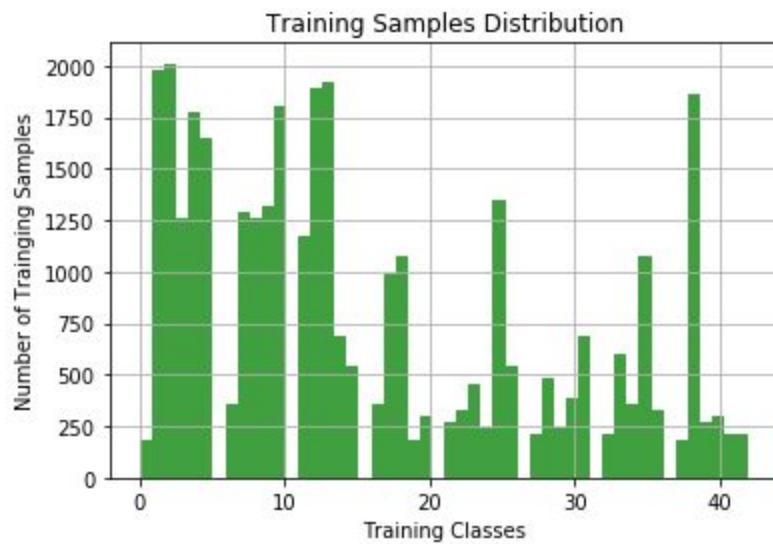
Data Set Summary & Exploration

Utilized Numpy Library to get the insights of the Data:

1. Size of training set-34799
2. Size of testing set -12630
3. Shape of image is - 32 by 32 by 3
4. The number of unique classes is : 43

Visualisation.

The image below Shows the distribution of training and testing samples



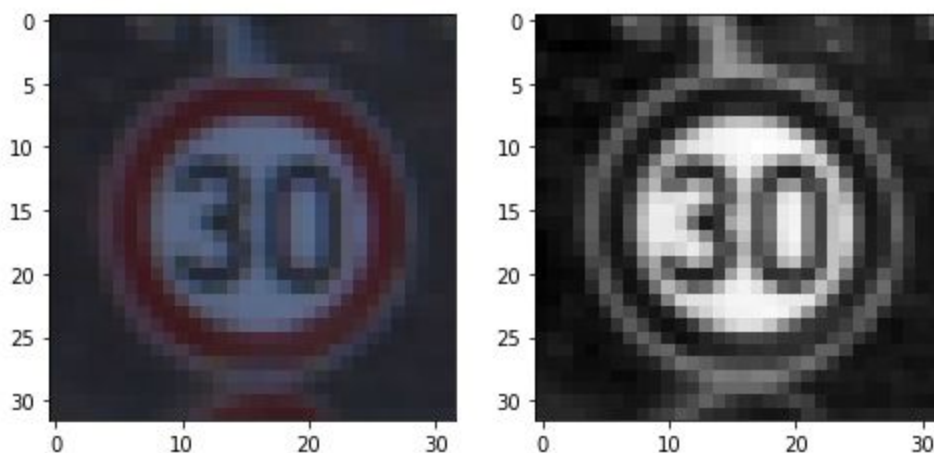
Preprocessing Technique:

The preprocessing technique used here is grayscale and normalization.

Converting into grayscale help us to deal with only 2 pixel values(0-255)

This reduces the Image depth from 3 to 1 which reduces the computation burden and also does not compromise the accuracy of the model.

Grey Scale Image shown below is:



Architecture.

I have used Convolutional Neural Network for the approach of this project.

My CNN is based on LENET Architecture.

The Values described Them is as follows:

Layer	Description
Input image	32 by 32 by 1 (1 because Gray scale)
Convolution Layer	1 by 1 stride with valid padding
relu	Activation
Max_pool	2 by 2 filter size

The following procedure is carried out twice till .ie 2 Convolutional layers and is passed on to the Fully connected layer.

This Fully connected layer gives us the softmax distribution of the layer.

TRAINING

I have used the following hyper parameters for training my model

EPOCHS = 150

BATCH_SIZE = 64

Learning rate =0.001

Optimizer used for minimizing the cost is AdamOptimizer

Accuracy

My Validation Accuracy is 0.94

My Initial approach was single convolutional layer which resulted in less accuracy thus I utilized Lenet Architecture for my project.

Problems on sign

Due to snow if the speed limit is 50 and zero gets covered Neural network may create false prediction.

Background objects also affect the judgement.

If signs are altered like they are fed with different angles in Network it may cause wrong prediction

Prediction:

The Prediction which was achieved was satisfying.

```
Validation Accuracy = 0.950
```

```
EPOCH 143 ...
```

```
Validation Accuracy = 0.938
```

```
EPOCH 144 ...
```

```
Validation Accuracy = 0.941
```

```
EPOCH 145 ...
```

```
Validation Accuracy = 0.950
```

```
EPOCH 146 ...
```

```
Validation Accuracy = 0.940
```

```
EPOCH 147 ...
```

```
Validation Accuracy = 0.944
```

```
EPOCH 148 ...
```

```
Validation Accuracy = 0.951
```