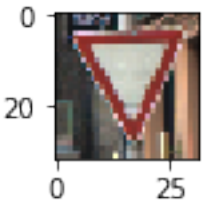
TRAFFIC SIGNAL CLASSIFIER PROJECT

Data Set Summary & Exploration:

* The size of training set : 34799
* The size of the validation set : 4400
* The size of test set is : 12630
* The shape of a traffic sign image is : 32x32 (Colored Image)
* The number of unique classes/labels in the data set is : 43

Visualisation of the Image:



Design and Test a Model Architecture

PREPROCESSING THE IMAGE:

1. Conversion to gray scale using opencv :

The images in the training set is converted from RGB to single channeled Grayscale image.So that we can reduce the input data size.

2. Normalisation:

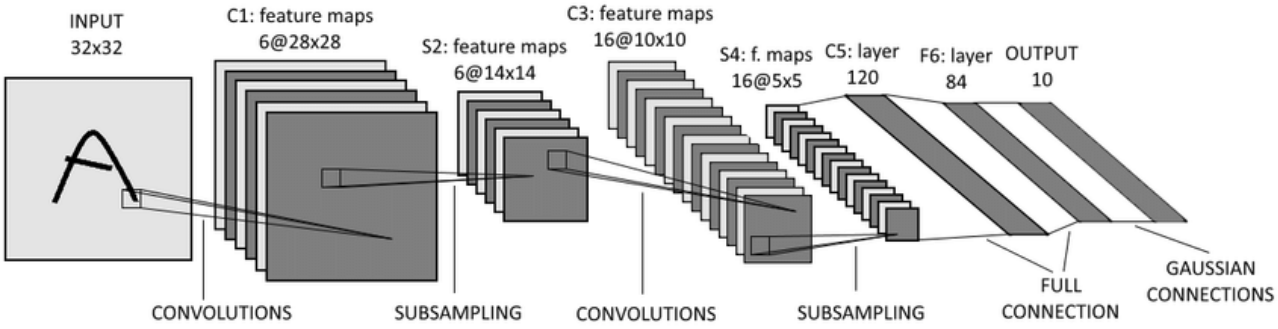
The image are normalized using the formula

pixel = (pixel/225)-0.5.

Normalisation was done to convert the input in the range of (-0.5 to 0.5) so that gradient descent flow will be easier.

Architecture of the model:

LeNet Architecture



The LeNet architecture is straght forward and small — it can even run on the CPU .

It consists of 5 layers two convolutional layers , two fully connected layers and a final classifier.

Layer 1: Convolutional. The output shape 32x32x6.

Pooling. The output shape 14x14x6.

Layer 2: Convolutional. The output shape 10x10x16.

Activation. ELU activation.

Pooling. The output shape should be 5x5x16.

Flatten. Flatten the output shape of the final pooling layer .

Layer 3: Fully Connected. 120 outputs.

Activation. ELU activation.

Dropout : Keep_prob =0.4

Layer 4: Fully Connected. 84 outputs.

Activation. ELU activation

Dropout : Keep_prob =0.4

Layer 5: Fully Connected (Logits). 43 outputs.

Training Parameters :

Epoch	: 20
Batch Size	: 128
Learning Rate	: 0.001

Optimizer Used : AdamOptimizer (tf.train.AdamOptimizer).
minimize() object in the AdamOptimizer is used to apply gradient descent and update the architecture variables.

Weights : Weights are initialized as a normal distribution of mean 0 and standard deviation that depends on the input of the each layer.

Biases : Biases are initialised as zeros

Training :

1. First the training epoch was set to 10 and the batch size was set to 64 and the validation accuracy is monitored for every epoch.

2. Based on the training accuracy of each epoch dropout keep_prob and other hyperparameters are adjusted .

3. The validation accuracy is tested for the batch sizes of 128,64,32 and learning rates 0f 0.001,0.001 and 0.0001 and the optimum value was selected.

4. Epoch size is choosen as 30 to get the maximum validation accuracy and the batch size is chosen as 128.

5. Finally i reached the validation accuracy of 0.952.

Test Accuracy of the Model:

Test Accuracy = 0.932.

Testing of Images taken from the web:

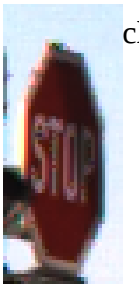
Label of the image : Speed limit (70km/h)
Size of the image : 36x36

This image would be difficult to classify because its brightness is too low.

Label of the image : Turn left ahead
Size of the image : 135x120

This is blurred so it would to difficult to classify.

Label of the image : Yield
Size of the image : 59x54



The brightness of the image is too high. So it may be difficult to classify.

Label of the image : No entry
Size of the image : 34x80

In this image the STOP board is tilted at some angle a part of the STOP board is shadowed.

Label of the image : Speed Limit (50km/h)
Size of the image : 88x82

As this image is not clear it would be difficult to classify.

Prediction of the Model:

If the model predicts all the five images taken from the web correctly then the accuracy is 100% .

No of images predicted correctly	Accuracy in percentage
5	100
4	80
3	60
2	40
1	20

The Prediction is

(4, b'Speed limit (70km/h)')
(28, b'Children crossing')
(25, b'Road work')
(8, b'Speed limit (120km/h')
(2, b'Speed limit (50km/h)')

The Accuracy of the model tested at random images from web is :**80**