Traffic Sign Classification

1. Submission Files: The project submission includes all required files.

The following files are included:

- Traffic_Sign_Classifier.ipynb
- Traffic_Sign_Classifier.pdf
- Traffic_Sign_Classifier_Report.ipynb
- road_signs.jpg
- WebImages: 12-1.jpg 32-1.jpg 36-1.jpg 38-1.jpg 9-1.jpg

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

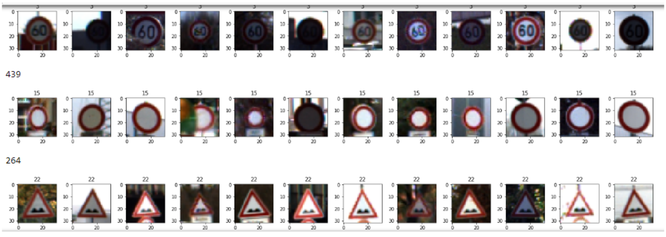
No of training examples: 34799 No of test examples: 12630

Amended Image Shape: (32, 32, 3)

Total number of unique labels: 43

Sort images in descending order of frequency of appearance in data set [(19, 141), (0, 142), (37, 143), (42, 160), (27, 162), (41, 174), (32, 175), (24, 194), (29, 197), (21, 210), (39, 215), (20, 231), (40, 238), (36, 262), (22, 264), (16, 282), (34, 304), (6, 307), (30, 323), (23, 352), (28, 382), (26, 435), (15, 439), (33, 483), (14, 544), (31, 556), (17, 799), (18, 861), (35, 869), (11, 948), (8, 996), (3, 1019), (7, 1026), (25, 1059), (9, 1072), (5, 1323), (4, 1377), (10, 1456), (38, 1496), (12, 1507), (13, 1515), (1, 1594), (2, 1607)]

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



4. Preprocessing: The submission describes the preprocessing techniques used and why these techniques were chosen.

Details of the preprocessing techniques employed are explained below. Rather simple techniques such as normalization and shuffling have been used and proved sufficient to achieve near-optimal results.

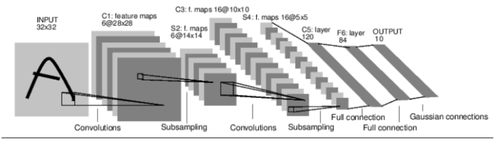
20% of the dataset has been put aside for validation and will be seen by the model only after training and testing has been finalised.

Normalize to zero mean

Normalization to zero mean and a small standard deviation will assist the model to do its job more efficiently. This is achieved by deducting 128 and then dividing the result by 128 for each channel of the image using OpenCV: cv2.normalize(image, zeros, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F) This normalizes the images for the optimizer.

The images are finally shuffled to remove inherent bias.

5. 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.



The LeNet framework appears to deliver surprising good results on this data set, achieving an accuracy of > 98% on the validation set after only 100 epochs.

Perhaps the LeNet's good fit for the task of classifying images arises from the fact that the framework was first used in classifying handwritten digits also contained in images with 32 by 32 pixels. However, in this case no normalization to grayscale has been undertaken.

LeNet-5 architecture

The LeNet architecture contains 5 layers as detailed below. ReLUs are used for activations and max pooling is employed. Drop-outs are added to limit overfitting.

- Convolution layer (Using 5x5 patch and the 3 filters)
- Subsample layer (Activation and max pool with a stride of 2x2)
- Convolution layer (Using a 5x5 patch with 6 filters)
- Subsample layer (Another activation using a 2x2 stride to reduce the dimensions)
- Fully connected layer to flatten the features and activation
- 4th layer fully connected
- Activation and relu
- 5th layer, we extract our features and apply softmax to convert them to probabilities that add to one.

Potential enhancements to LeNet framework

LeNet's advantage is its speed of computation and training, but the downside is its potential to overfitting.

The LeNet framework may be improved by using inception techniques to improve the CNN by using multiple patch sizes, 1x1 convolutions to increase the depth of the image, as well as pooling techniques. Running such a model on the traffic signs data sets may need deep GPU computaional resources.

6. Model Training: The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyperparameters.

Data augmentation was considered but deemed unecessary. The generation of new data by rotating the images and multiple brightness levels did not help and was also discarded.

The following hyperparameters were used:

Learning rate = 0.001

EPOCHS = 100

BATCH_SIZE = 128

The Adam Optimizer was used.

7. Solution Approach: The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

The LeNet-5 (LeCun) framework appeared to be fit-for-purpose for the problem at hand of traffic signs classification and provided surprisingly good results out of the box with little or no additional tweaks required.

The framework led to optimization within only 50 epochs and the model trained with computational efficiency.
The accuracy on the validation exceeded 0.98.

8. 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.

Five more images depicting German traffic signs were selected from the internet.
The images were selected based on their variety in terms of clarity, brightness, and focus.

9. 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.

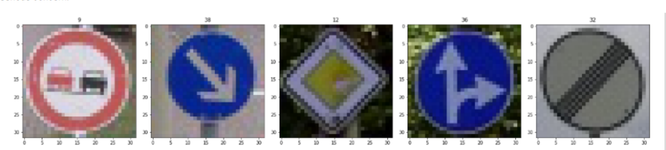
The model performed surprisingly well on the new images, scoring a 100% success rate at correct identification.

The probabilities returned by the model on the new images are 100% and this may call for further investigation tot understand if the results are biased.

Calculating the accuracy on these five German traffic sign images found on the web might not give a comprehensive overview of how well the model is performing. Consider ways to do a more detailed analysis of model performance by looking at predictions in more detail. For example, calculate the precision and recall for each traffic sign type from the test set and then compare performance on these five new images.

10. 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.

The model is very certain of its predictions which may indicate a degree of overfitting. Further investigation could be conducted to ascertain if the issue is of any serious concern.



```
0 : Label= 9 ,Predicted= 9 No passing = 1.0
1 : Label= 38 ,Predicted= 38 Keep right = 1.0
2 : Label= 12 ,Predicted= 12 Priority road = 1.0
3 : Label= 36 ,Predicted= 36 Go straight or right = 1.0
4 : Label= 32 ,Predicted= 32 End of all speed and passing limits = 1.0
Counted:  0 incorrect predictions
```

Print softmax probabilities

```
: print(probs[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.]
```

In [ ]: