Project 2 Report – Traffic Sign Classifier

Rafael Barreto

1. DATA SET SUMMARY & EXPLORATION

Libraries that were used to visualize and plotted graphs, images from the dataset:

- PIL
- matplotlib.pyplot
- random
- numpy
- pandas

DESCRIPTION OF THE DATA

Total number of images used = 51839
Number of training examples = 34799 (67.13%)
Number of validation examples = 4410 (8.51%)
Number of testing examples = 12630 (24.36%)

Image data shape = (32, 32, 3)

Number of classes = 43

Table with all type of signs used in the dataset was generated in the project. Below an image with 11 signs:

	ClassId	SignName
0	0	Speed limit (20km/h)
1	1	Speed limit (30km/h)
2	2	Speed limit (50km/h)
3	3	Speed limit (60km/h)
4	4	Speed limit (70km/h)
5	5	Speed limit (80km/h)
6	6	End of speed limit (80km/h)
7	7	Speed limit (100km/h)
8	8	Speed limit (120km/h)
9	9	No passing
10	10	No passing for vehicles over 3.5 metric tons

2. DESIGN AND TEST A MODEL ARCHITECTURE

2.1 The dataset was preprocessed with only two techniques:

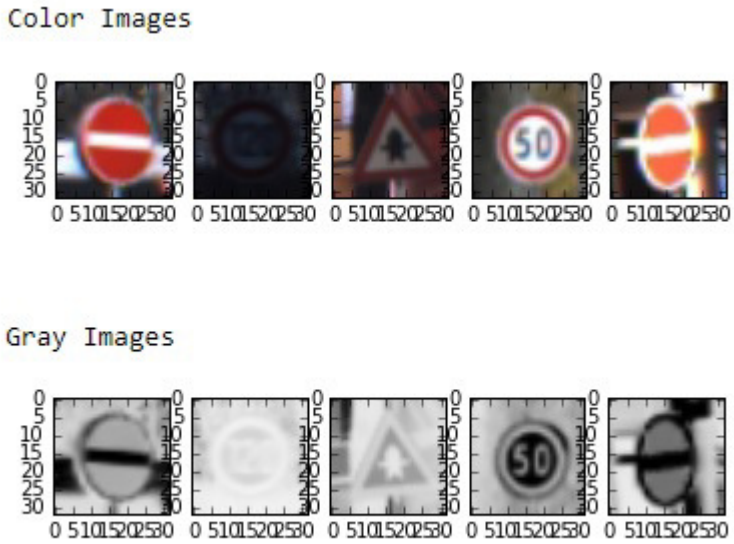
- Conversion to grayscale and reshape the array image.

This method takes the average of the three values of the pixel and build an image with shape (32,32,1). The colors are not important for classification task in this project and the system is going to perform better if the network does not have to learn colors. Therefore the conversion to grayscale is a good practice to image classification in this case.

- Feature scaling

This method rescales the range of features to in [0, 1] or [−1, 1]. In this project was chosen the range [-1,1]. The benefits of this technique vary according to the algorithm. In the case of Deep Leaning feature scaling makes the training faster and prevents getting stuck in local optima.

Below is showed the conversion of color images to grayscale using the method used:



2.1 Model Architecture:

Final model consisted of the following layers:

Layer	Description
Input	32x32x1 image Grayscale Normalized
Hyperparameters	mu = 0 sigma = 0.05
Convolutional 1	Input = 32x32x1 Output = 28x28x6 Strides Padding = 'VALID'
Activation 1	RELU
Pooling 1	Type: avg_pool Input = 10x10x16 Output = 5x5x16 Stride: 1x1 Padding: 'VALID'
Convolutional 2	Input =14x14x6 Output = 10x10x16
Activation 2	RELU
Pooling 2	Type: max_pool Input = 10x10x16 Output = 5x5x16 Stride: 1x1 Padding: 'VALID'
Flatten	Input = 5x5x16 Output = 400
Fully Connected 3	Input = 400 Output = 240.
Activation 3	RELU
Fully Connected 4	Input = 240 Output = 144
Activation 4	RELU
Fully Connected 5	Input = 144 Output = 86
Activation 5	RELU

Layer	Description
Fully Connected 6	Input = 86 Output = LOGITS = 43

3. TRAINED MODEL DESCRIPTION

- EPOCHS = 20
- BATCH_SIZE = 32
- Optimizer = Adam's algorithm (tf.train.AdamOptimizer)

Hyperparameters:

- mu = 0
- sigma = 0.05

4. APPROACH DESCRIPTION

The architecture chosen was the solution of the LaNet project provided by Udacity. This architecture achieved a validation accuracy of approximately 0.89 in my project. This architecture is relevant to the traffic sign application because it works very well in other projects of image classification.

After several attempts the architecture achieved 0.957 of validation accuracy. This was possible with the follow modifications:

- Adding one more fully connected layer
- The hiperparameters sigma = 0.05
- Learning rate = 0.0015
- First pooling type: avg_pool
- Changes of all fully connected inputs and outputs

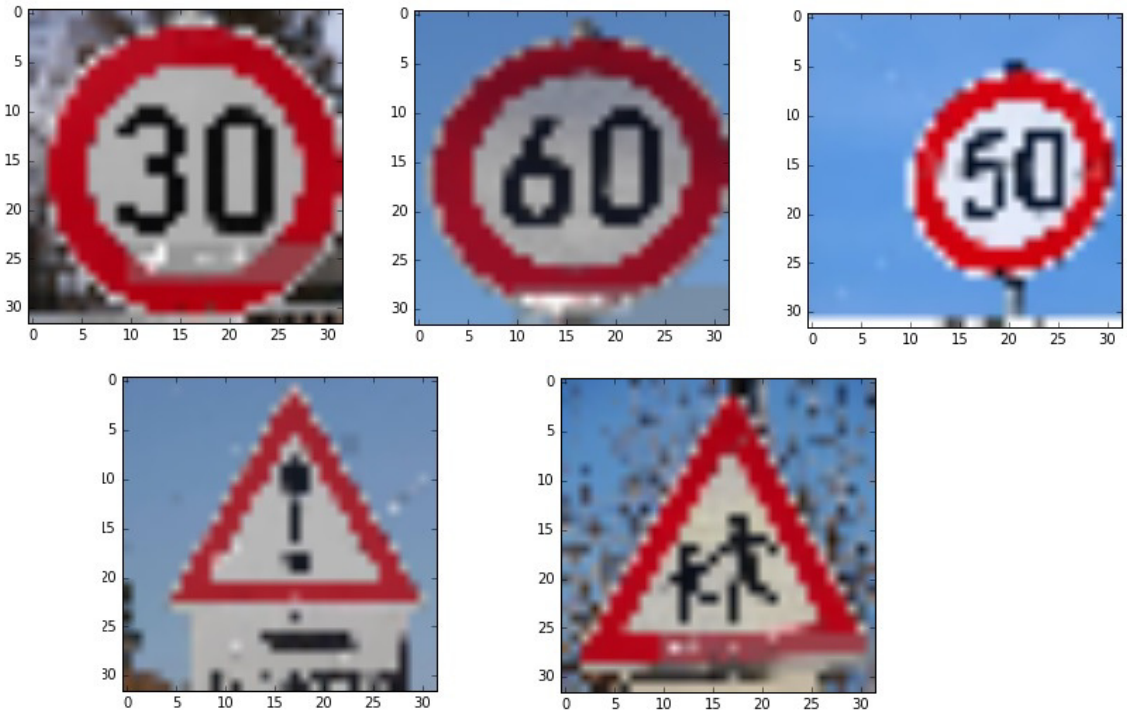
Final model results:

- Validation Set Accuracy: 0.957
- Test Set Accuracy: 0.91

5. TEST A MODEL ON NEW IMAGES

Here are five German traffic signs that downloaded from the web:

All five images were resized to 32x32 to be applied in the program:



Here are the results of the prediction:

Image	Prediction
Speed limit (30km/h)	Speed limit (30km/h)
Speed limit (60km/h)	Speed limit (60km/h)
Speed limit (50km/h)	Speed limit (60km/h)
General caution	Keep right
Children crossing	Children crossing

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%. More images are needed to make a correct conclusion if the result is favorable to the accuracy on the test set of 0.91. We noticed that after the resize the third image ("Speed limit (60km/h)"), the number five approached to the number 6. About the fourth image ("General Caution"), after resize it is much distorted.

Output Top 5 Softmax Probabilities:

For the **FIRST** image: a Speed limit (30km/h)

Probability	Prediction
1.00	Speed limit (30km/h)
0	End of speed limit (80km/h)
0	Speed limit (70km/h)
0	Speed limit (20km/h)
0	Speed limit (50km/h)

For the **SECOND** image: Speed limit (60km/h)

Probability	Prediction
0.9994856119	Speed limit (60km/h)
0.0002918579	Dangerous curve to the right

Probability	Prediction
0.0001507624	End of no passing
0.0000682464	End of speed limit (80km/h)
0.0000028231	Go straight or right

For the **THIRD** image: Speed limit (50km/h)

Probability	Prediction
0.9994856119	Speed limit (60km/h)
0.0002918579	Dangerous curve to the right
0.0001507624	End of no passing
0.0000682464	End of speed limit (80km/h)
0.0000028231	Go straight or right

For the **FOURTH** image: General caution

Probability	Prediction
0.7763769031	Keep right
0.2223720104	Speed limit (60km/h)
0.0012194272	Speed limit (30km/h)
0.0000163052	Priority road
0.0000078736	End of all speed and passing limits

For the **FIFFT** image: Children crossing

Probability	Prediction
≈1.000000000	Children crossing
0.0000000154	Turn left ahead
0	Bumpy road
0	General caution

Probability	Prediction
0	Ahead only