Writeup for Project 2

Files Submitted

1) This writeup
2) Traffic_Sign_Classifer.html
3) Five new test images
4) Five cropped images of the five new test images

Data Set Summary & Exploration

1. Dataset Summary

The code for this step is contained in the second code cell of the IPython notebook.

I used python len() function and the pandas library to calculate summary statistics of the traffic signs data set:

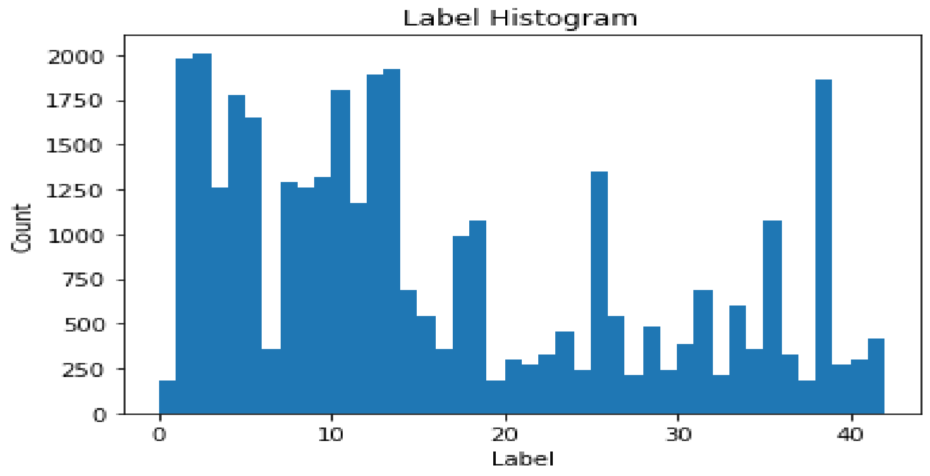
- The size of training set is 34799
- The size of test set is 12630
- The shape of a traffic sign image is (32, 32, 3)
- The number of unique classes/labels in the data set is 43

2. Exploratory Visualization

The code for this step is contained in the third and fourth code cell of the IPython notebook.

I first random plotted a traffic sign image, which is below:

Secondly, I tried to plot the label histogram of the train set, which is below:



Design and Test a Model Architecture

1. Preprocessing

I have used image normalization technique provided by opencv to preprocess the whole data set. The reason for me to do this is that I notice that some images are to dark, such as the image plotted above. Image normalization can reduce the discrepancy of the light condition of the images in the data set and therefore introduces less noise to the classifier.

The code in the fifth cell. All image normalization, the same image is brighter and more recognizable.

I didn't gray scale the dataset because it causes information loss and also CNN is able to process color image. As a last step, I shuffled the data. The code is in the sixth cell.

2. Model Architecture

The code for my final model is located in the seventh cell of the ipython notebook.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 5x5	1x1 stride, "VALID" padding, output 28x28x6
RELU	
Max Pooling	Stride 2x2, "VALID" padding, output 14x14x6
Convolution 5x5	1x1 stride, "VALID" padding, output 10x10x16
RELU	
Max Pooling	Stride 2x2, "VALID" padding, output 5x5x16
Flatten	output 400
Fully Connected	output 120
RELU	
Fully Connected	output 100
RELU	
Fully Connected	output 43

4. Model Training

The code for training the model is located in the 8th to 12th cells of the ipython notebook.

To train the model, I used Adam Optimizer. The batch size is 128 and the number of epochs is 10.

The learning rate is 0.001.

5. Solution Design
The code for calculating the accuracy of the model is located in the 12th and 13th cell of the Ipython notebook.

My final model results were:

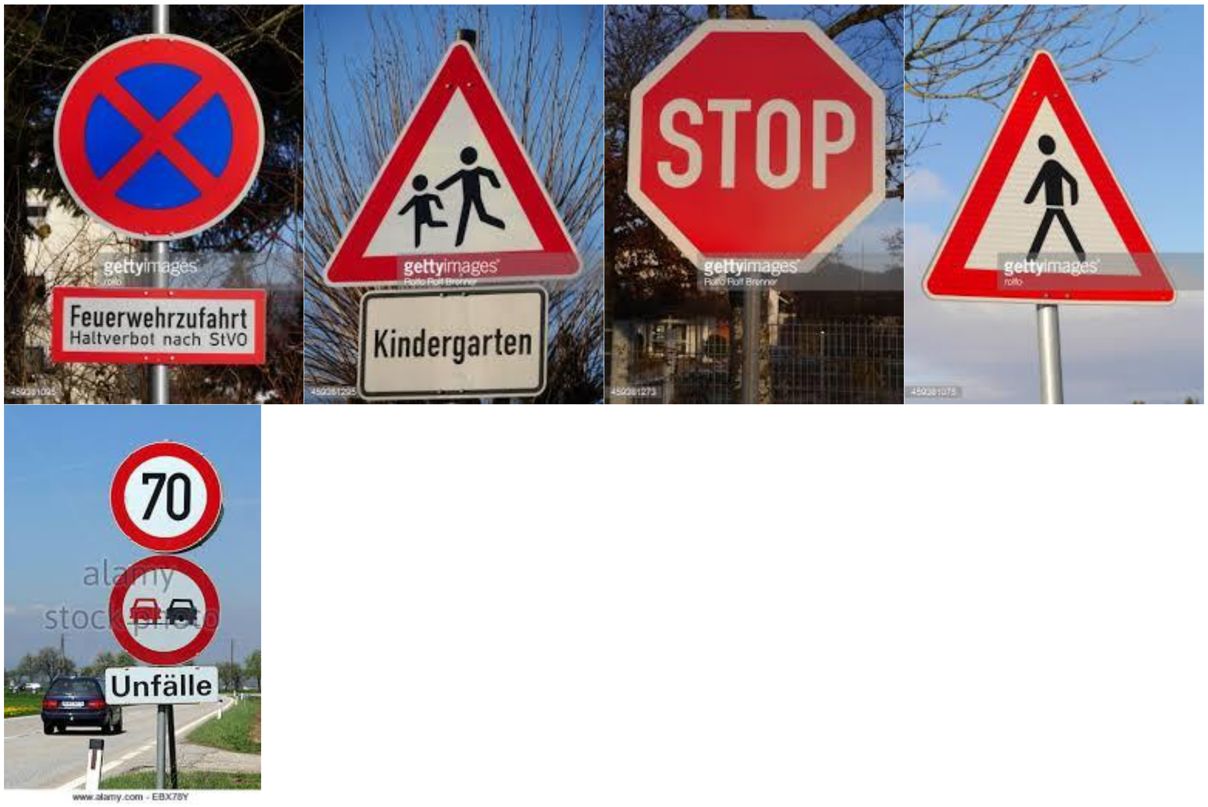
- validation set accuracy of 0.843 on average
- test set accuracy of 0.88

I am using LeNet which is taught in the lecture. Below is the process for me to choose current model and parameters.

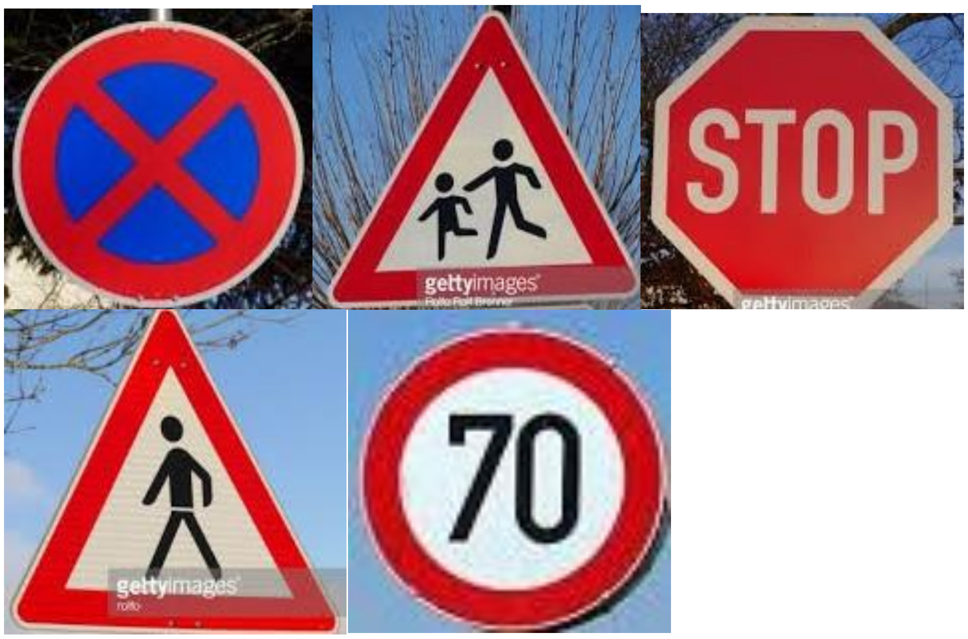
- LeNet works well for hand-written digit images. Though traffic sign dataset is more complicated than digit images in that it has more classes and the traffic sign objects are more complicated than digits, the underlying useful features are similar which are the edges constituting the whole object. Therefore, LeNet should also be effective for traffic sign images
- After picking up LeNet model, I used the same parameters as that taught in the lecture for training and testing. It turned out that the final testing accuracy is around 0.86 which is as high as that for LeNet on digit images.
- Therefore, I tried to tune the parameters such as the EPOCH, the batch size, convolution template size, convolution output layers and the output size of fully-connected layers. It turned out none of these tuning changed the decrease too much. Some tuning even decreased the accuracy. Only EPOCH seems to have the potential to increase accuracy, but not much. As a result, I used the original LeNet parameters with EPOCH of value 15. The final test accuracy is 0.88 which is shown in 13th Ipython cell.

Test a Model on New Images

1. Five German traffic signs
Here are five German traffic signs that I found on the web:



One common difficulty of these five images is that the images are larger than the dataset used for training the classifier. For the 2nd to 4th images, one additional difficulty is that there are letters in the area of the traffic sign. Therefore, I cropped the images manually and resized them to size 32x32. The results are below. The code is in 14th Ipython cell.



2. Prediction Results on these new traffic signs

The code for making predictions on my final model is located in the 15th

and 16th cell of the Ipython notebook.

Here are the results of the prediction:

Image	Prediction
No entry	General Caution
Child crossing	Child crossing
Stop sign	Yield
Pedestrians	General Caution
70 km/h	70 km/h

The model was able to correctly guess 2 of the 5 traffic signs, which gives an accuracy of 40%.

3. The softmax probabilities for each

The code for making predictions on my final model is located in the 17th

cell of the Ipython notebook.

For the first image, the top five soft max probabilities were below.

Unfortunately, the right label 9 is not within the top five.

Probability	Label	Prediction
.90	18	General Caution
.07	12	Priority Road
.01	33	Turn right ahead
.0017	25	Road work
.0014	14	Stop

For the second image, the top five soft max probabilities were below.

The classifier is very confident 28 is the correct answer.

Probability	Label	Prediction
.99	28	Children crossing
.006	22	Bumpy road
.002	18	General caution
.002	29	Bicycles crossing
.000001	20	Dangerous curve to the right

For the third image, the top five soft max probabilities were below.

The correct label 14 is in the top five.

Probability	Label	Prediction
.84	13	Yield
.14	14	Stop
.003	18	General caution
.0012	17	No entry
.0001	15	No vehicles

For the fourth image, the top five soft max probabilities were below.

The correct label 27 is also in the top five.

Probability	Label	Prediction
.65	18	General Caution
.34	27	Pedestrians
.00000004	24	Road narrows on the right
1.3e-11	11	Right-of-way at the next intersection
7.3e-15	21	Double curve

For the fifth image, the top five soft max probabilities were below.

The classifier is confident that 4 is the correct answer.

Probability	Label	Prediction
.99	4	Speed Limit 70 km/h
2.7e-03	0	Speed Limit 20 km/h
9.4e-07	1	Speed Limit 30 km/h
3.0e-10	40	Roundabout mandatory
2.2e-10	25	Road work