Traffic Sign Recognition

Writeup

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

Build a Traffic Sign Recognition Project

The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Rubric Points

Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

Writeup / README

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

You're reading it!

Data Set Summary & Exploration

1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

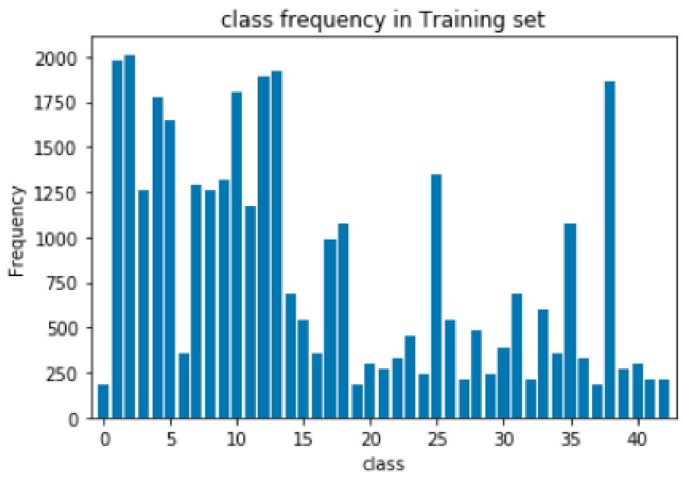
I used the pandas library to calculate summary statistics of the traffic signs data set:

```
Total number of classes:43
Number of training examples = 34799
Number of validation examples = 4410
Number of testing examples = 12630
Image data shape= (32, 32, 3)
Number of classes = 43
```

2. Include an exploratory visualization of the dataset.

Here is a sample of the images from the dataset:

Here is the bar diagram showing the frequency of class in the Training Set:

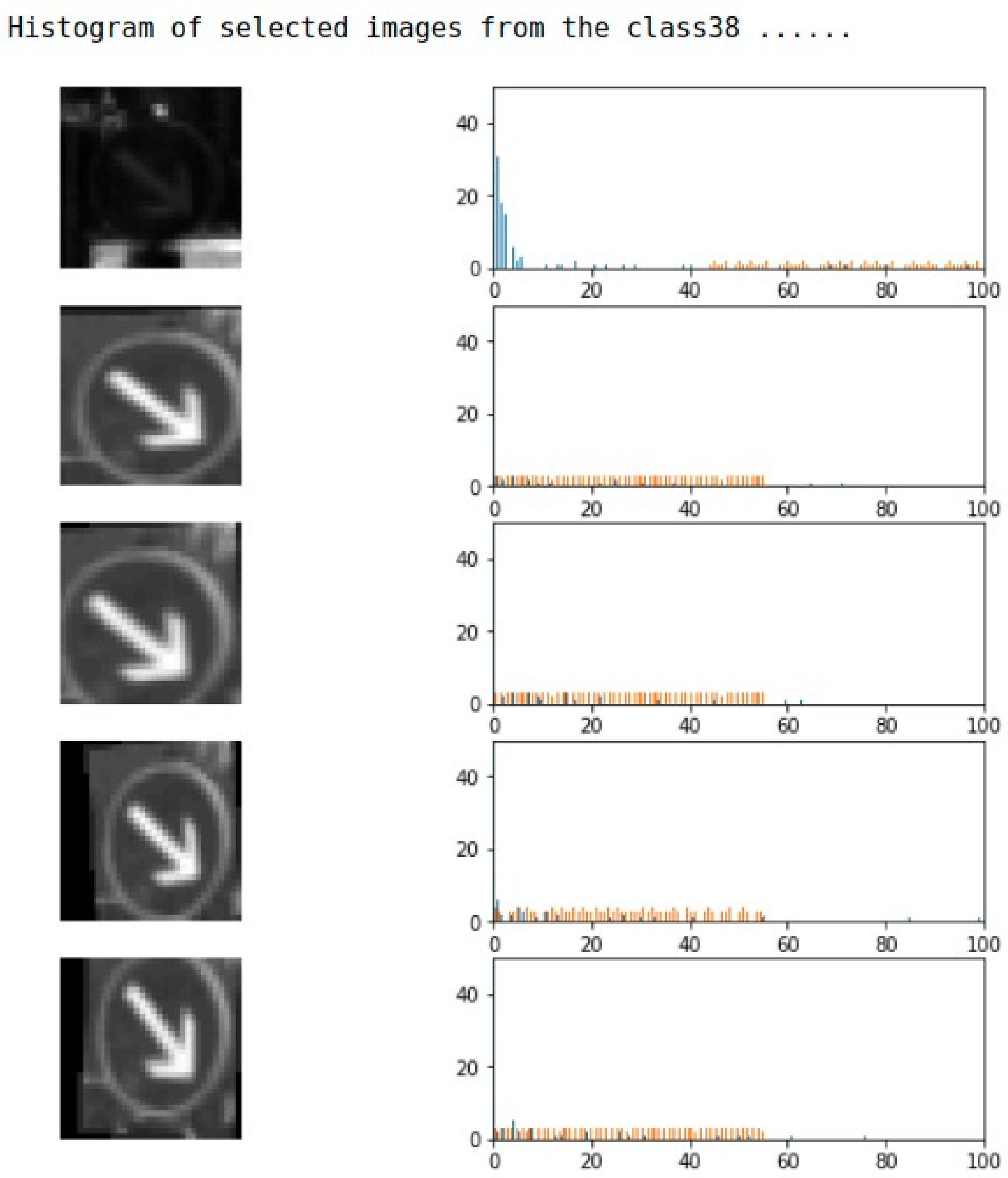


```
+-------------+----------------------------------------------------+
| class value |                Name of Traffic sign                |
+-------------+----------------------------------------------------+
|      0      |                Speed limit (20km/h)                |
|      1      |                Speed limit (30km/h)                |
|      2      |                Speed limit (50km/h)                |
|      3      |                Speed limit (60km/h)                |
|      4      |                Speed limit (70km/h)                |
|      5      |                Speed limit (80km/h)                |
|      6      |            End of speed limit (80km/h)             |
|      7      |               Speed limit (100km/h)                |
|      8      |               Speed limit (120km/h)                |
|      9      |                     No passing                     |
|      10     |    No passing for vechiles over 3.5 metric tons    |
|      11     |       Right-of-way at the next intersection        |
|      12     |                   Priority road                    |
|      13     |                       Yield                        |
|      14     |                        Stop                        |
|      15     |                    No vechiles                     |
|      16     |      Vechiles over 3.5 metric tons prohibited      |
|      17     |                      No entry                      |
|      18     |                  General caution                   |
|      19     |            Dangerous curve to the left             |
|      20     |            Dangerous curve to the right            |
|      21     |                    Double curve                    |
|      22     |                     Bumpy road                     |
|      23     |                   Slippery road                    |
|      24     |             Road narrows on the right              |
|      25     |                     Road work                      |
|      26     |                  Traffic signals                   |
|      27     |                    Pedestrians                     |
|      28     |                 Children crossing                  |
|      29     |                 Bicycles crossing                  |
|      30     |                 Beware of ice/snow                 |
|      31     |               Wild animals crossing                |
|      32     |        End of all speed and passing limits         |
|      33     |                  Turn right ahead                  |
|      34     |                  Turn left ahead                   |
|      35     |                     Ahead only                     |
|      36     |                Go straight or right                |
|      37     |                Go straight or left                 |
|      38     |                     Keep right                     |
|      39     |                     Keep left                      |
|      40     |                Roundabout mandatory                |
|      41     |                 End of no passing                  |
|      42     | End of no passing by vechiles over 3.5 metric tons |
+-------------+----------------------------------------------------+
```

Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc.

As a first step, I decided to convert the images to grayscale. We convert our 3 channel image to a single grayscale image (we do the same thing in project 1 - Lane Line Detection)



```
Histogram of selected images from the class38 ......
```

Then we used functions as

- random_transform()

  To take an image and performs a set of random affine transformation

- data augmentation()

  To achieve data augmentation.

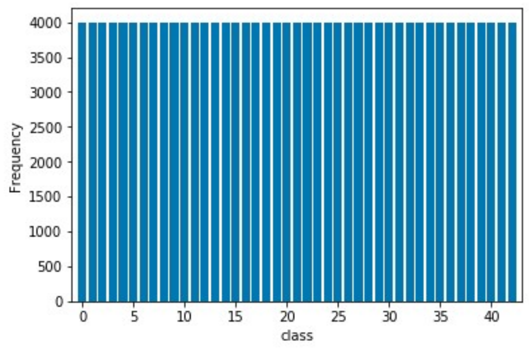
Here the reason we need to process the dataset is to

1. increase the size of the training set.

2. adjust the distribution of class for a better test result.

After dataset augmentation we have the following sample visualization and bar diagram:

```
*** Train dataset after augmentation
        Total Number of images in Train dataset:172000
```



Comparing the difference based on previous result we can see significant improvement.

2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model.

My final model consisted of the following layers:

Block	Layer	Description	Output
	Input	32x32x1 RGB image	32x32x1
Block 1	Convolution	1x1 stride, valid padding	30x30x80
	RELU + Dropout		
	Convolution	1x1 stride, same padding	28x28x120
	Max pooling	2x2 stride	14x14x120
	RELU + Dropout		
Block 2	Convolution	1x1 stride, valid padding	11x11x180
	RELU + Dropout		
	Convolution	1x1 stride, same padding	9x9x200
	Max pooling	2x2 stride	4x4x200
	RELU + Dropout		
Block 3	Convolution	1x1 stride, valid padding	2x2x200
	RELU		
	Max pooling	2x2 stride	1x1x200
	Dropout		
Fully connected layers	Fully connected 1	RELU + dropout	
	Fully connected 2	RELU + dropout	
	Softmax		

The convolution layers are formed into 3 blocks, each one has 2 convolution layers and a max pooling layer for the first 2 blocks. The 3rd block has a minor order change for the convolution layer and max pooling. Then it's applied to 2 FC layers and then softmax for final output.

3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

To train the model, I used an Adam Optimizer, 150 epochs and batch size of 200, with the learning rate starting from 0.001 until epoch 30 and then reduced to 1/5 to epoch 50.

4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the architecture is suitable for the current problem.

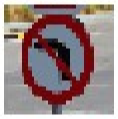
My final model results were:

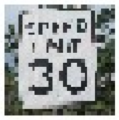
- training set accuracy of 1.0000
- validation set accuracy of 0.9991
- test set accuracy of 0.9825

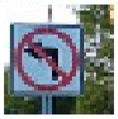
I used the LeNet-5 architecture as reference because it has a well-known performance on gray scaled pictures recognization, I changed the pooling type and some filter size to improve training speed of the network. We can see from the result that the accuracy of the final test result is very promising.

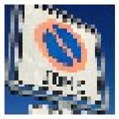
Test a Model on New Images

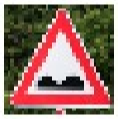
I used some random images I found online and here are the prediction result from my model:

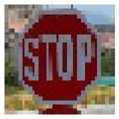


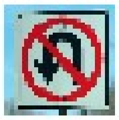


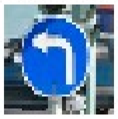


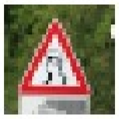


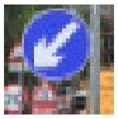




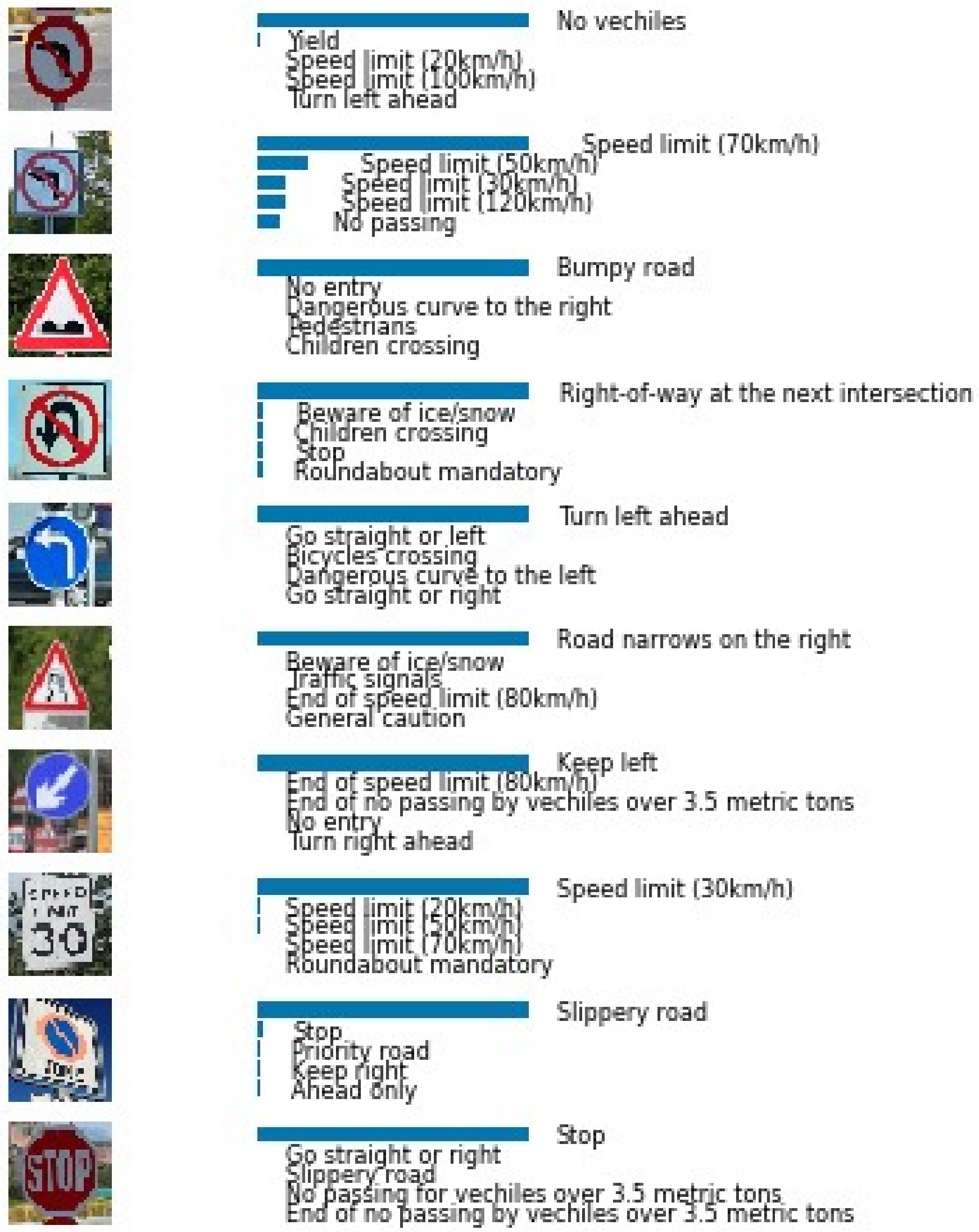








We can see that some of them have the wrong prediction result, I'm listing out the top 5 possibilities for all of them:


No vechiles
Yield
Speed limit (20km/h)
Speed limit (100km/h)
Turn left ahead
Speed limit (70km/h)
Speed limit (50km/h)
Speed limit (30km/h)
Speed limit (120km/h)
No passing
Bumpy road
No entry
Dangerous curve to the right
Pedestrians
Children crossing
Right-of-way at the next intersection
Beware of ice/snow
Children crossing
Stop
Roundabout mandatory
Turn left ahead
Go straight or left
Bicycles crossing
Dangerous curve to the left
Go straight or right
Road narrows on the right
Beware of ice/snow
Traffic signals
End of speed limit (80km/h)
General caution
Keep left
End of speed limit (80km/h)
End of no passing by vechiles over 3.5 metric tons
No entry
Turn right ahead
Speed limit (30km/h)
Speed limit (20km/h)
Speed limit (50km/h)
Speed limit (70km/h)
Roundabout mandatory
Slippery road
Stop
Priority road
Keep right
Ahead only
Stop
Go straight or right
Slippery road
No passing for vechiles over 3.5 metric tons
End of no passing by vechiles over 3.5 metric tons