

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! and here is a link to my [project code](#)

Data Set Summary & Exploration

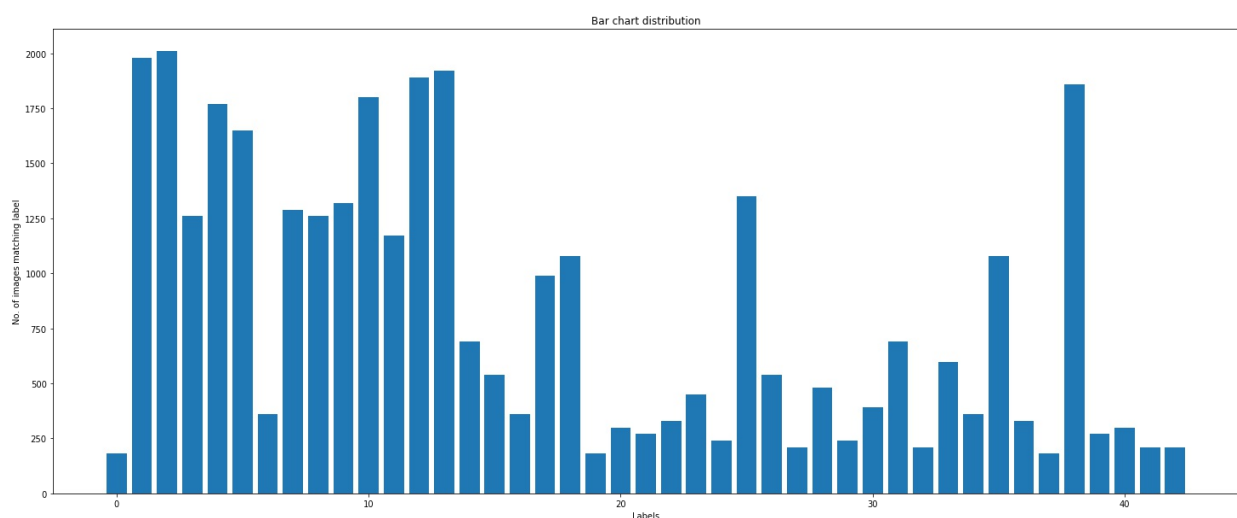
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

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

- The size of training set is 34799
- The size of the validation set is 4410
- The size of test set is 12630
- The shape of a traffic sign image is 32 x 32 x 3
- The number of unique classes/labels in the data set is 43

2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing distribution of 43 different types of traffic signs.



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. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data

for training, describe why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

I used normalization of image by subtracting 128 and dividing with 128. I also suffeled data in order to get best performance on model training.

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:

Layer	Description
Input	32x32x3 RGB image
Convolution	1x1 stride, same padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution	1x1 stride, same padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, outputs 5x5x16
Flattening	outputs 400
Fully connected	outputs 120
RELU	
Fully connected	outputs 84
RELU	
Fully connected	outputs 43

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 placeholders x and y for a batch of input images and output labels for that batch. I used Adam Optimizer passing to it cross entropy and minimizing the training_operation. That was further used to calculate the accuracy using argmax function.

The batch size used is 60 and epochs used is 20. Initially I used batch size of 120 and epoch of 10 and did not get good results. learning rate used is 0.001.

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 0.989
- validation set accuracy of 0.903
- test set accuracy of 0.910

Le-Net architecture was used to train the model. Traffic data is complex data and it justifies use of Le-Net architecture that has 13 layers. Looking at the final output, I was able to achieve training accuracy of 98.9%, validation accuracy of 90.3% and test accuracy of 91% that is very good.

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Here are six German traffic signs that I found on the web(I used three category, two images each):



I ran it few times and in one of the run, the first image was not recognized accurately. And it make sense as the image is very blurry. With subsequent runs, I got 100% accuracy on those images.

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).

Here are the results of the prediction:

Image	Prediction
Speed limit (70km/h)	Speed limit (70km/h)
No entry	No entry
Road narrows on the right	Road narrows on the right

The model was able to correctly guess 6 of the 6 traffic signs, which gives an accuracy of 100%.

3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

The code for making predictions on my final model is located in the 11th cell of the lpython notebook.

For the first image, the model is relatively sure that this is a speed limit sign (probability of 100%), and the image does contain a speed limit sign. The top five softmax probabilities were (two images were used in each category and probabilities are shown in pair here)

Probability	Prediction
.999 and 1.00	Speed limit (70km/h)
.974 and .999	No entry
.981 and .999	Road narrows on the right

For the second and third images as shown, the model predicted it accurately despite of them being blurred.

(Optional) Visualizing the Neural Network (See Step 4 of the lpython notebook for more details)

1. Discuss the visual output of your trained network's feature maps. What characteristics did the neural network use to make classifications?