

# Traffic Sign Recognition

---

## 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

---

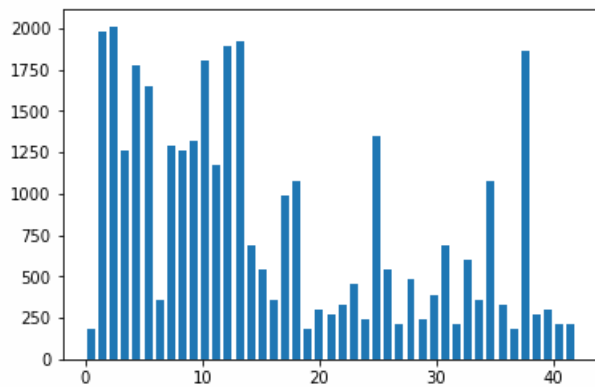
**1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.**

I used the numpy 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 16230
- The shape of a traffic sign image is 32,32,3
- 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. I visualized random image and its label(ClassId) in the dataset and made a statistic figure of labels.



## 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 only normalized the image data because I believed that color is also important features, especially for traffic sign. But it turns out my assumption is wrong which the accuracy is not very well. Normalizing the image can reduce the number of shades which can improve the performance. To reduce time for future use, I saved the preprocessed

data into harddrive.

**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 5x5	1x1 stride, same padding, outputs 28x28x64
RELU	
Max pooling	2x2 stride, outputs 14x14x64
Convolution 5x5	1x1 stride, same padding, outputs 10x10x256
RELU	
Max pooling	2x2 stride, outputs 5x5x256
Flatten	outputs 6400
Fully connected	outputs 128
Dropout	keep probability = 0.5
Fully connected	outputs 43
Softmax	

**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 the follow global parameters:

epochs = 20 (actually 15 is enough)

batch\_size = 512

keep\_probability = 0.5

Optimizer - Adam Optimizer

**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.**

If an iterative approach was chosen:

\* What was the first architecture that was tried and why was it chosen?

I tried a model based on simplified VGG16 architecture at first due to it is very famous in image classifaication.

\* What were some problems with the initial architecture?

However, the VGG16 is too complicated for this project (which the image size is only 32x32x3), very slow and without good performance.

\* How was the architecture adjusted and why was it adjusted? Typical adjustments could include choosing a different model architecture, adding or taking away layers (pooling, dropout, convolution, etc), using an activation function or changing the activation function. One common justification for adjusting an architecture would be due to overfitting or underfitting. A high accuracy on the training set but low accuracy on the validation set indicates over fitting; a low accuracy on both sets indicates under fitting. I tried a model based on simplified VGG16 architecture at first, but is only 0.88 even after I reduced block number, then I reduceed more cov layers, make it as a classic LeNet model but with more output depth(more details), the accuracy start more than 0.9.

\* Which parameters were tuned? How were they adjusted and why?

Mainly I tuned Convolution layer output depth, if it is too small, it could only get 0.88 accuracy, but too large may be the trainning would be too slow and the performance didn't increase.

\* What are some of the important design choices and why were they chosen? For example, why might a convolution layer work well with this problem? How might a dropout layer help with creating a successful model?

The architecture is very important, it should choose according to dataset.

If a well known architecture was chosen:

\* What architecture was chosen?

VGG16 and LeNet-5.

\* Why did you believe it would be relevant to the traffic sign application?

They are both famous CNN image classification model, however VGG16 is too large for this application.

\* How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

My final model results is:

\* training set accuracy of 0.999

\* validation set accuracy of 0.959

\* test set accuracy of 0.945

## 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:



The 1,4,5,6 image might be difficult to classify because the background is complicated.  
The 2,3 image might be difficult to classify because the light is not enough

**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
Turn left	Turn left
60km/h	60km/h
Yield	Yield
Ahead only	Ahead only
Stop	Stop
No entry	No entry

The model was able to correctly guess 6 of the 6 traffic signs, which gives an accuracy of 100%. This compares favorably to the accuracy on the test set of 94.5%. I am considering the pics might be too easy for this network.

**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 Ipython notebook.

For the 1st image, the model is sure that this is a Turn left sign (probability of 0.99). The top five softmax probabilities were Turn left, Turn left ahead, Keep right, Go straight or right, Speed limit (80km/h). The first four prediction is reasonable.

Probability	Prediction
.99	Turn left ahead
.001	Ahead only

Probability	Prediction
.001	Go straight or right
.001	peed limit (80km/h)

For the 2nd image, the model is quite sure that this is a Speed limit (60km/h) sign (probability of 1). The top five soft max probabilities were Speed limit (60km/h),Speed limit (80km/h), Speed limit (50km/h), No vehicles, Ahead only. The first three prediction is reasonable.

Probability	Prediction
1.	Speed limit (60km/h)
.0	Speed limit (80km/h)
.0	Speed limit (50km/h)
.0	No vehicles
.0	Ahead only

For the 3rd image, the model is quite sure that this is a Yield Sign (probability of 1). The top five soft max probabilities were Yield,No vehicles, Speed limit (60km/h), No passing, Speed limit (80km/h). All the five prediction is reasonable.

Probability	Prediction
1.	Yield
.0	No vehicles
.0	Speed limit (60km/h)
.0	No passing
.0	Speed limit (80km/h)

For the 4th image, the model is quite sure that this is a Ahead only Sign (probability of 1). The top five soft max probabilities were Ahead only, Turn left ahead, End of speed limit (80km/h), Turn right ahead, Keep left. All the five prediction is reasonable.

Probability	Prediction
1.	Ahead only
.0	Turn left ahead
.0	End of speed limit (80km/h)
.0	Turn right ahead
.0	Keep left

For the 5th image, the model is quite sure that this is a Stop Sign (probability of 1). The top five soft max probabilities were Stop, No entry, Yield, Road work,Speed limit (60km/h). The first three prediction is reasonable.

Probability	Prediction
1.	Stop
.0	No entry
.0	Yield
.0	Road work
.0	Speed limit (60km/h)

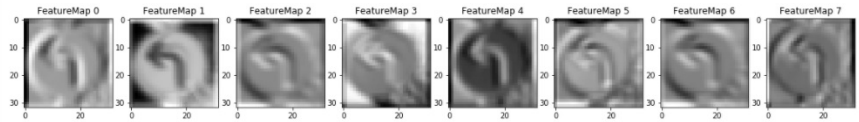
For the 6th image, the model is quite sure that this is a No entry Sign (probability of 1). The top five soft max probabilities were No entry, Stop, Speed limit (20km/h), Speed limit (30km/h), Speed limit (50km/h). All the predictions are reasonable.

Probability	Prediction
1.	No entry
.0	Stop
.0	Speed limit (20km/h)
.0	Speed limit (30km/h)
.0	Speed limit (50km/h)

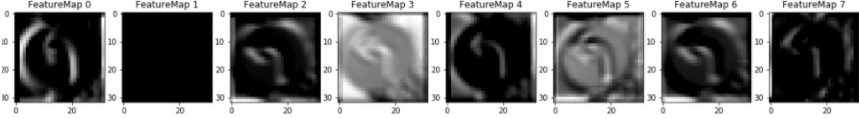
### **(Optional) Visualizing the Neural Network (See Step 4 of the Ipython notebook for more details)**

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

Convolution (layer 1) It's possible to see shape of sign, but with many background noises.



ReLU-activation (layer 1 ) Noise from background reduced.



Max-pooling (layer 1) Image size reduced, but still with important features.

