# 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

## 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

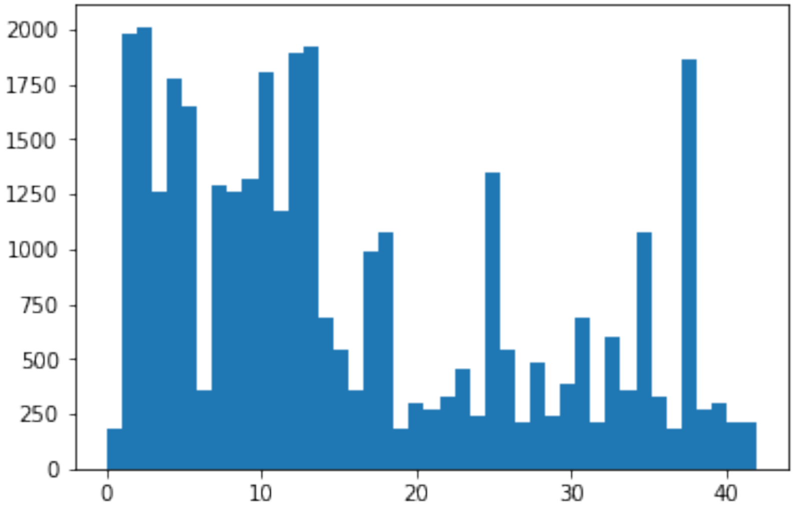
### 1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

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 ?  12630
- The size of test set is ?  4410
- The shape of a traffic sign image is ? (32,32)
- 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 how the data ...



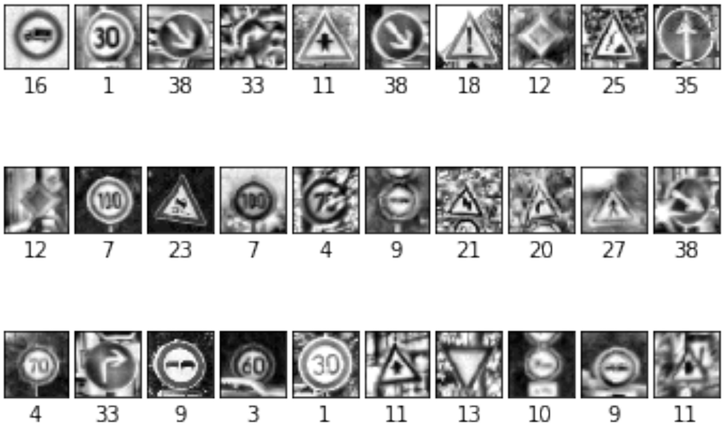
## Design and Test a Model Architecture

### 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.)

As a first step, I decided to convert the images to grayscale because only the shape of the sign matters, we don't care much about the color .

Secondly, I applied CLAHE on the images to adjust the relatively constraction, because I found out that some of the pictures are too dark to be seen.

Finally, I applied normalization of the images to further normalize the constraction. The result is shown in below.



### 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 | 32x32x1 Grayscale image |
| Convolution 5X5 | 1x1 stride, valid padding, outputs 28x28x16 |
| RELU | |
| Max pooling | 2x2 stride, outputs 14x14x16 |
| Convolution 5X5 | 1x1 stride, valid padding, outputs 10x10x36 |
| RELU | |
| Max pooling | 2x2 stride, outputs 5x5x36 |
| Flatten | Outputs 900 |
| Dropout | 0.7 |
| Fully Connected | Input 900, output 200 |
| RELU | |
| Fully Connected | Input 200, output 80 |
| RELU | |
| Fully Connected | Input 80, output 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.

Optimizer : AdamOptimizer

Epochs: 10

BatchSize: 128

Learning rate : 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 ?   0.973
- validation set accuracy of ?   0.951
- test set accuracy of ?  0.935

A well known architecture was chosen:

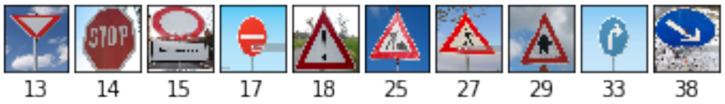
- What architecture was chosen?   LeNet
- Why did you believe it would be relevant to the traffic sign application?  Because the LeNet focuses on the classification, it has been studied for years.
- How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

  The final result of the training model is pretty 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 ten German traffic signs that I found on the web:



The sixth and seventh images are hard to be classified, because it has some noises.

### 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 |
|---|---|
| Yield | Right-of-way at the next intersection |
| Stop | Stop |
| No Vehicle | Wild animals crossing |
| No Entry | No entry |
| General Caution | General caution |
| Road work | Road work |
| Pedestrians | General caution |
| Bicycles Crossing | Right-of-way at the next intersection |
| Turn righ ahead | General caution |
| Keep right | Yield |

The model was able to correctly guess 4 of the 10 traffic signs, which gives an accuracy of 40%. This is very different from the test  accuracy. It is possible that the skewness of the images afftect the result.

### 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)

For the first image, the model is relatively sure that this is a stop sign (probability of 0.6), and the image does contain a stop sign. The top five soft max probabilities were

| Probability | Prediction |
|---|---|
| .6599 | Right-of-way at the next intersection |
| .2782 | Turn right ahead |
| .02414 | End of all speed and passing limits |
| .01834 | Speed limit (30km/h) |
| .0065 | General caution |

Second Image :

| Probability | Prediction |
|---|---|
| .38826 | Stop |
| .2294 | Yield |
| .065 | Speed limit (50km/h) |
| .063 | General caution |
| .036 | Speed limit (30km/h) |

Third Image:

| Probability | Prediction |
|---|---|
| 0.90995872 | Wild animals crossing |
| 0.03560625 | Bumpy road |
| 0.02832666 | Stop |
| 0.00663636 | Road work |
| 0.00529973 | Speed limit (80km/h) |

Forth Image:

| Probability | Prediction |
|---|---|
| 9.88100052e-01 | No entry |
| 1.07322214e-02 | Turn right ahead |
| 6.67361019e-04 | Stop |
| 1.58636904e-04 | Turn left ahead |
| 1.37301351e-04 | Ahead only |

Fifth Image:

| Probability | Prediction |
|---|---|
| 9.90048528e-01 | General caution |
| 9.83966794e-03 | Right-of-way at the next intersection |
| 8.42810477e-05 | Pedestrians |
| 2.41885737e-05 | Traffic signals |
| 3.01798991e-06 | Road narrows on the right |

We can see from the fifth image is detected very well, and the possible reason is that fifth image has the best image quality for detection.

## (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?