

# Traffic Sign Recognition

Udacity SDC Term 1, P2

By David Rose

2017-05-11

---

## Convolutional Neural Network in Base TensorFlow

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

- You're reading it! and here is a link to my code: [github.com/cipher982/Traffic-Sign-Conv-Net-Recognition](https://github.com/cipher982/Traffic-Sign-Conv-Net-Recognition)

## 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.

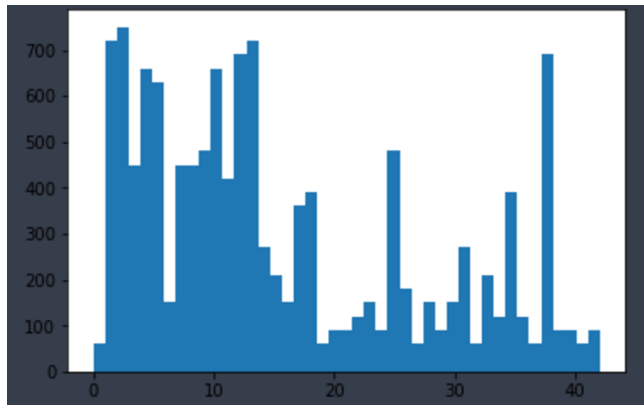
I used the pandas library to calculate summary statistics of the traffic signs data set:

- The size of training set is:
  - Before processing/augmentation: **34799**
- The size of the validation set is --
- The size of test set is **12630**
- The shape of a traffic sign image is **32x32** pixels, with **3** levels of color depth

- The number of unique classes/labels in the data set is **43**

## 2. Include an exploratory visualization of the dataset.

Here is a bar chart showing how the data

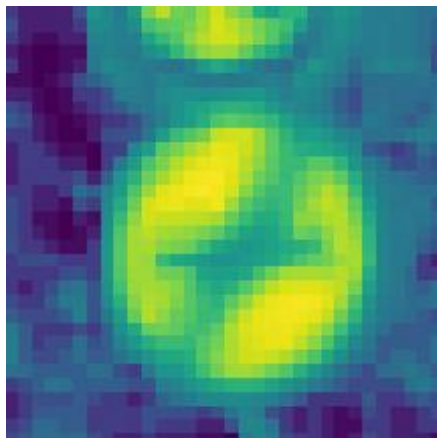
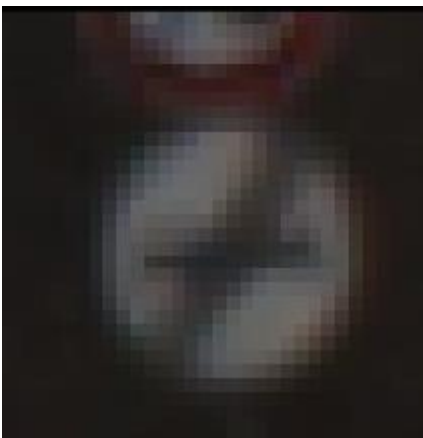


## Design and Test a Model Architecture

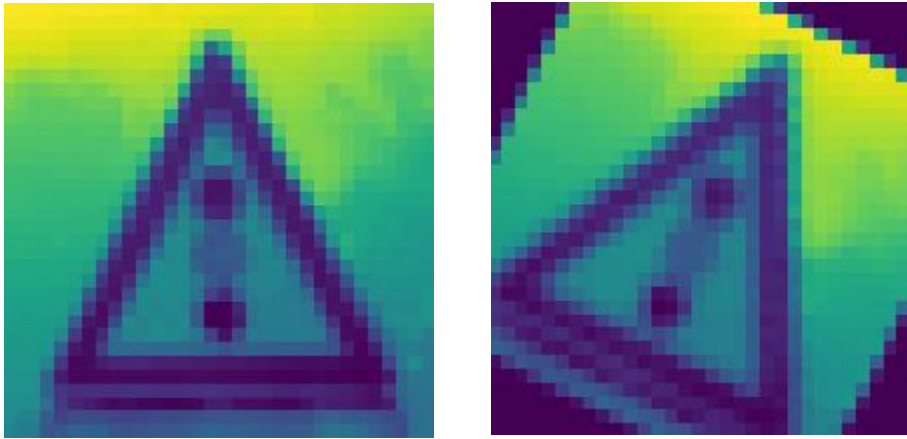
### 1. Describe how you preprocessed the image data.

I first converted to **one color channel** and **equalized the pixel brightness** average

- I did not want differences in lighting/exposure/brightness to bias or skew the training



Then I created additional training data from the original, as it is possible some signs may not be seen straight on. I did this by randomly rotating the images between **-30 and +30** degrees from original. The dataset ended up **3 times larger** than the original, at **104397**



## 2. Describe what your final model architecture looks like

My final model consisted of the following layers:

| Layer                      |
|----------------------------|
| Input, 32 input            |
| Convolution                |
| Convolution                |
| Convolution                |
| Drop Layer, keep 90%       |
| Convolution                |
| Flatten                    |
| Fully Connected            |
| Fully Connected            |
| Fully Connected, 43 output |

## 3. Describe how you trained your model.

- I used minibatches of size 150, with 15 total epochs. Trained over my Nvidia 960M GPU in laptop with tensorflow. Learning rate set at 0.001, sigma at 0.1, mu at 0.

## 4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93.

- My final results were the work of lots of random tweaking. Modifying parameters up/down until something started to work better. Switching from CPU to GPU allowed this process to go along much faster.
- I started out adding two extra fully connected layers, but this did not seem to help much at all. Then I read some others had better luck adding convolutional layers instead.
- I also came across a function for installing new layers, which took out a lot of the manual work of explicitly typing out layer by layer in TensorFlow.

|                         |      |
|-------------------------|------|
| training set accuracy   | .990 |
| validation set accuracy | .954 |
| test set accuracy       | .933 |

Dropout was also added, but I did not use as much as others do, only a single layer keeping 90%.

### Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report.



I tried to find relatively simple, straight on images. They vary a bit on backgrounds, but otherwise looks quite similar.

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set.

Here are the results of the prediction:

| Image         | Prediction         |
|---------------|--------------------|
| Yield         | Priority Road      |
| Double Turn   | Keep Left          |
| Workers Ahead | Beware of Ice/Snow |
| 70 km/h       | Slippery Road      |
| Slippery Road | Priority Road      |

The model was not able to guess a single correct sign.

**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.**

| <b>Sign</b>   | <b>Predictions</b>   |
|---------------|--|
| Yield         | 1 : Priority road 44.5974<br>2 : Beware of ice/snow 28.7431<br>3 : Road work 27.8022<br>4 : Slippery road 27.2703<br>5 : Right-of-way at the next intersection 26.5415       |
| Double Turn   | 1 : Keep left 28.2596<br>2 : Priority road 24.9101<br>3 : Slippery road 23.9862<br>4 : Beware of ice/snow 22.7613<br>5 : End of no passing 22.0854                           |
| Workers Ahead | 1 : Beware of ice/snow 43.4253<br>2 : Keep right 39.1332<br>3 : Priority road 34.3958<br>4 : No entry 26.0766<br>5 : End of no passing 24.5803                               |
| 70 km/h       | 1 : Slippery road 20.2756<br>2 : Priority road 19.9837<br>3 : No passing 19.7819<br>4 : Road work 19.7066<br>5 : General caution 18.6672                                     |
| Slippery Road | 1 : Priority road 41.8424<br>2 : Beware of ice/snow 32.9941<br>3 : Keep right 29.4746<br>4 : Speed limit (20km/h) 28.633<br>5 : Right-of-way at the next intersection 23.331 |