

# **Traffic Sign Classifier (TSC) Project Submission Writeup Report**

**For the Udacity Self-Driving Car Program – Term 1**

**Submitted by:**

**Stewart Teaze**

**May 12, 2017**

## Project Goals

The goals / steps of this project are:

- Load the provided German Traffic Signs 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

The following sections of this document relate to each of the project's rubric points (project requirements), and provide details of how each requirement was addressed in the project implementation.

---

## Writeup

1. This file is the Writeup itself, and is provided in .pdf format. The project code can be found at: <https://github.com/blunderbuss9/CarND-TSC-StewartTeaze.html>

## Data Set Summary & Exploration

### 1. Following are basic summary statistics of the provided German Traffic Signs data set:

- The size of (original) training set is 34799
- The size of the validation set is 8699 ( $1/4 * 34799$ )
- The size of test set is 26100 ( $3/4 * 34799$ )
- The shape of a traffic sign image is (32,32,3)
- The number of unique classes/labels in the data set is 43

### 2. Exploratory visualization of the dataset.

Table 1 below provides an exploratory visualization of the provided German Traffic Signs data set, and includes additional summary statistics. In the project code itself, this analysis and visualization was performed and rendered using python, with the numpy and matplotlib libraries. The table provides one representative image from the training set file, corresponding to each Traffic Sign type that was found in the training set file.

The title text above each image is X:Y, where X is the Traffic Sign Class label/type, & Y is the number of occurrences of that label/type that were encountered in the training set file:

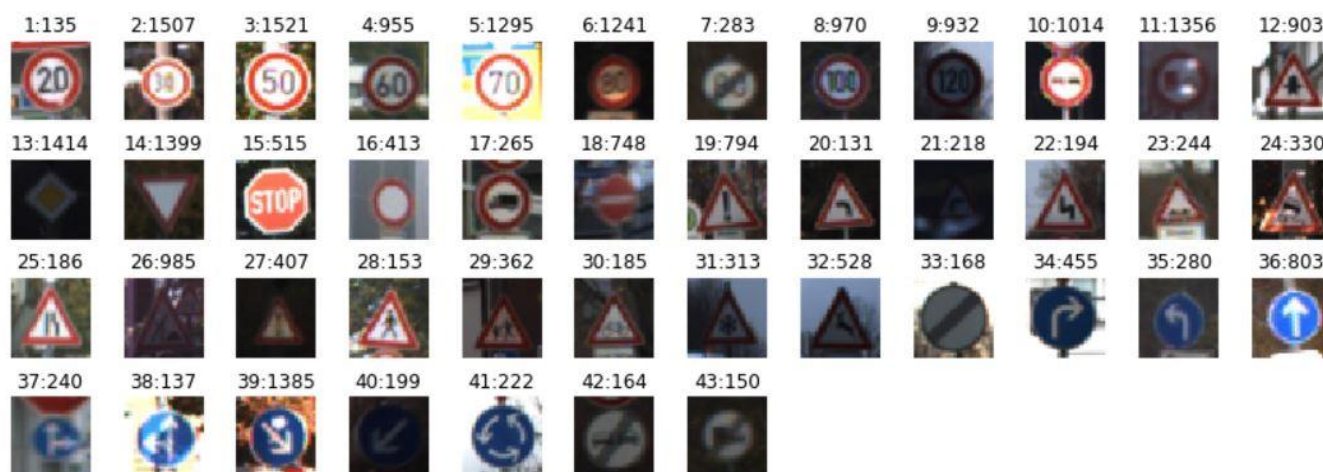


TABLE 1

## Design and Test of the Model Architecture

### 1. Image Data Preprocessing.

The images that were provided in the German Traffic Sign dataset file train.p (a subset of the German Traffic Sign Recognition Benchmark dataset of more than 50,000 traffic sign images [Stallkamp]), are read in, and then split into two parts: 75% training data, and 25% validation data; by doing this, validation set accuracy measurements of between 0.944 and 0.961 were achieved, during various test runs against the created validation set - all of these measurements were above the 0.93 minimum requirement for a successful project submission, and a significant increase over the roughly .88 accuracy achieved during initial runs made with the provided validation dataset (whose size was only roughly 15% of the training set).

The training data is then shuffled randomly for each run – it is important to randomly shuffle the training data, otherwise the order of the data could have a huge effect on how well the network model trains.

### 2. Model Architecture Description.

The machine learning network model architecture implementation is a multi-layer (Deep Learning) Convolutional Neural Network (CNN), that is based on and adapted from the LeNet-5 5-Layer architecture [LeCun98]. The following table provides the details for the stages (steps/techniques) implemented, within each layer of this project's machine learning network model's architecture implementation:

	STAGES / LAYER			
LAYER 1	<u>Convolutional</u> Stride: 1x1 Padding: VALID Input: 32x32x3(RGB) Output: 28x28x6	<u>Activation</u> Type: ReLU	<u>Max Pooling</u> Stride: 2x2 Padding: VALID Input: 28x28x6 Output: 14x14x6	
LAYER 2	<u>Convolutional</u> Stride: 1x1 Padding: VALID Input: 14x14x6 Output: 10x10x16	<u>Activation</u> Type: ReLU	<u>Max Pooling</u> Stride: 2x2 Padding: VALID Input: 10x10x16 Output: 5x5x16	<u>Flatten</u> Input: 5x5x16 Output: 400
LAYER 3	<u>Fully Connected</u> Input: 400 Output: 120	<u>Activation</u> Type: ReLU		
LAYER 4	<u>Fully Connected</u> Input: 120 Output: 84	<u>Activation</u> Type: ReLU		
LAYER 5	<u>Fully Connected</u> Input: 84 Output: 43 (Logits)			

TABLE 2

### 3. Model Training Description

The last output (logits), described in the table above, is then softmax cross-entropy one-hot encoded, for later use by the AdamOptimizer loss minimization functional step in each epoch pass in the training pipeline.

A batch size of 128 per epoch was used; and the training pipeline used 10 epochs (passes). The learning rate hyperparameter was set to .001. Mean and standard deviation parameters within all layers of the model were set to 0 and .1, respectively.

### 4. Solution Approach used for Achieving Validation Accuracy > 0.93

The well-known LeNet-5 5-Layer CNN architecture was used, as it has been shown to yield good results on image recognition type problems, and has been well-studied, with numerous support resources readily available on The Internet; for example, the LeNet and CNN approaches are the core of the Stanford Computer Vision course CS231n, which has generated huge interest throughout the Artificial Intelligence machine learning community, and whose resources are accessible via The Internet to those interested [\[CS231n17\]](#).

The main approach used, to achieve validation accuracy was to split the provided German Traffic Sign dataset file train.p, into two parts: 75% training data, and 25% validation data; by doing this, validation set accuracy measurements of between 0.944 and 0.961 were achieved, during numerous test runs against this increased-size validation set. These measurements were a significant improvement over the roughly .88 accuracy achieved during initial runs I had made with the provided validation dataset file valid.p (whose size was only roughly 15% of the training set). This approach was used, because numerous research results have shown the validation dataset should be a “substantial fraction” of the training dataset size, to achieve optimal results... “In most cases a single validation set of respectable size substantially simplifies the code base, without the need for cross-validation with multiple folds. You’ll hear people say they “cross-validated” a parameter, but many times it is assumed that they still only used a single validation set” [\[CS231n17\]](#).

My final model results were:

- validation set accuracies of between .0944-.0961
- test set accuracy of .866

## Model Test Results using Web-based Images

Following are five sample German traffic signs that I found on the web, and used for the test evaluation run.

### 1. Sample Traffic Sign Images

- a. This image, found at <http://media.istockphoto.com/vectors/german-road-sign-speed-limit-100-kmh-vector-id472360055> was cropped to 512x512 pixels using MS Paint; while it has some watermark artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (NOTE: for comparison purposes, the 32x32 re-sized images, produced and output during the project code evaluation section run, are included next to each of the original-sized images presented here).



- b. This image, found at [https://ak2.picdn.net/shutterstock/videos/19542826/thumb/9.jpg?i10c=img.resize\(height:160\)](https://ak2.picdn.net/shutterstock/videos/19542826/thumb/9.jpg?i10c=img.resize(height:160)) was originally cropped to 372x353 pixels, and resized square to 352x352 pixels using MS Paint; while there are some aspect ratio artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run(see 2<sup>nd</sup> image below).



- c. This image, found at

<http://www.gettyimages.com/detail/photo/sign-clapiers-herault-languedoc-roussillon-france-royalty-free-image/154336385>

was originally cropped and resized square to 212x212 pixels using MS Paint; while there are some aspect ratio artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (see 2<sup>nd</sup> image below).



d. This image, found at

<http://www.gettyimages.com/detail/photo/germany-berlin-two-direction-signs-at-potsdam-royalty-free-image/522934973>

was originally cropped and resized square to 352x352 pixels using MS Paint; while there are some aspect ratio artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (see 2<sup>nd</sup> image below). Perhaps more of a concern is the slight angle of the sign (skewed perhaps 5% counter-clockwise).





d. This image, found at

<https://previews.123rf.com/images/bwylezich/bwylezich1602/bwylezich160200111/51786163-children-crossing-German-road-sign-Stock-Photo.jpg>

was originally cropped and resized square to 640x640 pixels using MS Paint; due to the upward angle, there are some aspect ratio artifacts, and there are some watermark artifacts, but it can be seen that the effects on classification of these artifacts should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run(see 2<sup>nd</sup> image below).



## 2. Trained Model's Prediction Accuracy on New Dataset.

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
Stop Sign	Stop sign
U-turn	U-turn
Yield	Yield
100 km/h	Bumpy Road
Slippery Road	Slippery Road

The model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%. This compares favorably to the accuracy on the test set of ...

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 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
.60	Stop sign
.20	U-turn
.05	Yield
.04	Bumpy Road
.01	Slippery Road

For the second image ...

## **(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?**

## REFERENCES

- [Stallkamp] Stallkamp, I., Schlipsing, M., Salmen, J., Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition; The 2011 International Joint Conference on Neural Networks (IJCNN). <http://ieeexplore.ieee.org/document/6033395/>
- [LeCun98] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- [CS231n17] Karpathy, A., Li, F. (2017) <http://cs231n.stanford.edu/> Convolutional Neural Networks for Visual Recognition; Stanford University, Spring 2017.