

[◀ Return to "Self-Driving Car Engineer" in the classroom](#)[DISCUSS ON STUDENT HUB](#)

# Traffic Sign Classification

## REVIEW

## HISTORY

### Meets Specifications

Great job! Congratulations and good luck on your next project/term!

### Files Submitted

The project submission includes all required files.

- Ipython notebook with code
- HTML output of the code
- A writeup report (either pdf or markdown)

All files are provided  +1:

### Dataset Exploration

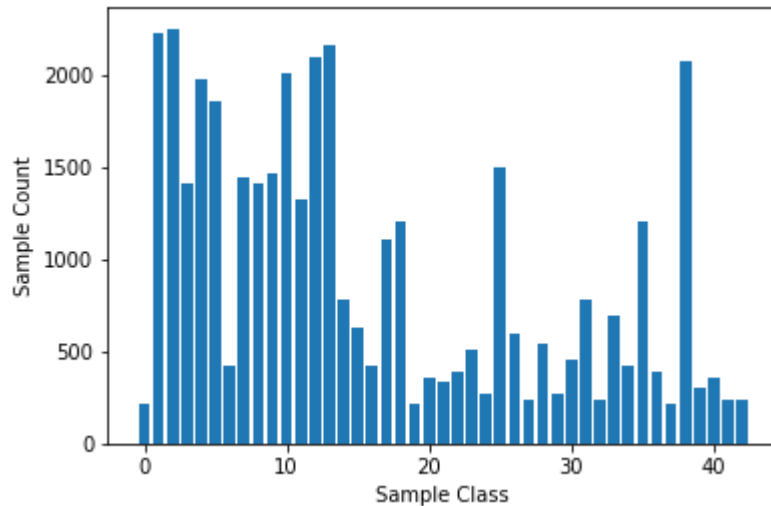
The submission includes a basic summary of the data set.

Good job with dataset exploration!

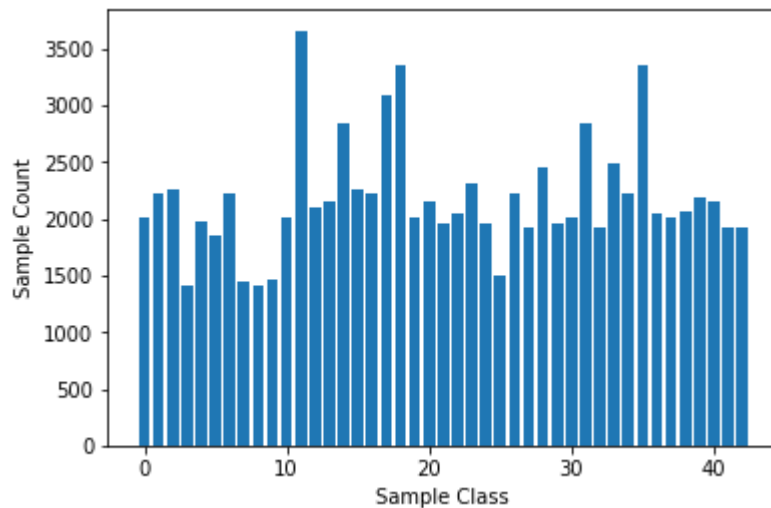
The submission includes an exploratory visualization on the dataset.

Well done outputting sample images of the data set, making a bar chart showing the number of images in each class! You can note that classes are unbalanced - some classes contain more than 1500, while some other classes contain less than 300 samples. You can balance them adding more images by augmenting.

Before balancing classes look like that:



After balancing the number of images in each class should be roughly equal. For example:



## Design and Test a Model Architecture

The submission describes the preprocessing techniques used and why these techniques were chosen.

Well done with normalization and converting images to greyscale. This will definitely improve the results. The reason for doing feature normalization is to improve the training conditions and speed up optimization.

Regarding greyscale there are some cons and pros.

Pros: more robust against variations in color saturation and lightness.

Cons: loss of valuable information. For example, turn signs are normally blue, while stop signs are red. This color difference would be useful in classification but it's lost after grayscale.

Here is an interesting paper about preprocessing:

<https://pdfs.semanticscholar.org/9052/2abb6613f083faf19ebc7d51a74a215cf344.pdf>

Some good guides:

- for beginners: <https://towardsdatascience.com/the-complete-beginners-guide-to-data-cleaning-and-preprocessing-2070b7d4c6d>
- overview: <https://towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6>

The submission provides details of the characteristics and qualities of the architecture, including the type of model used, the number of layers, and the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

- multiple 2D convolution layers are used
- nonlinearity applied using ReLU. Here is a good article about some activation layers if you are interested:  
<https://arxiv.org/pdf/1511.07289v1.pdf>  
[https://keras.io/layers/advanced\\_activations](https://keras.io/layers/advanced_activations)  
<http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>

You can also use Tensor Board for visualisation:

<https://www.datacamp.com/community/tutorials/tensorboard-tutorial>

The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyperparameters.

Well done with Adam optimizer and train/test/validation split!

Here is an excellent article about different gradient descent optimization algorithms:

<http://sebastianruder.com/optimizing-gradient-descent/index.html>

Here are more information about train/validation/test splits if you are interested:

<http://stats.stackexchange.com/questions/19048/what-is-the-difference-between-test-set-and-validation-set>

<http://stackoverflow.com/questions/13610074/is-there-a-rule-of-thumb-for-how-to-divide-a-dataset-into-training-and-validation>

The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

You can also try to use dropout to reduce overfitting and to help the model generalize better. Performing dropout for a layer will randomly dropout or deactivate a set number of nodes for the layer dropout is performed to try to reduce overfitting. Dropout can reduce overfitting and force the model to generalize better because the model cannot depend on any particular node for the dropout layer.

And here are also more info about dropout:

<http://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>

<http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf>

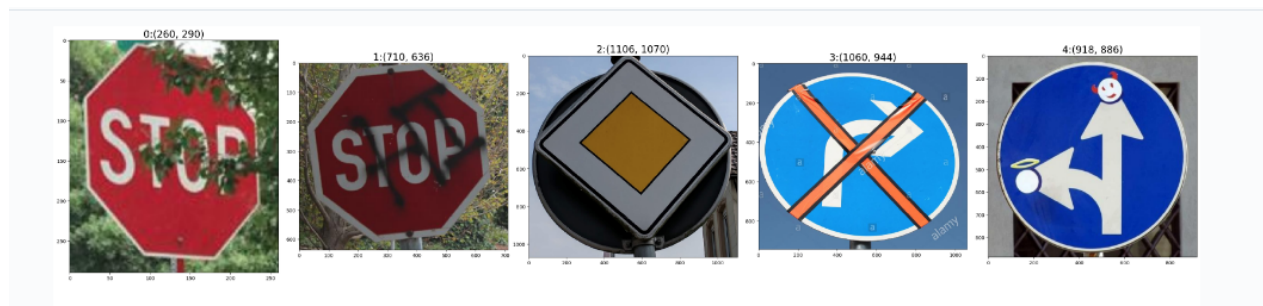
<https://pgaleone.eu/deep-learning/regularization/2017/01/10/analysis-of-dropout/>

## Test a Model on New Images

The submission includes five new German Traffic signs found on the web, and the images are visualized. Discussion is made as to particular qualities of the images or traffic signs in the images that are of interest, such as whether they would be difficult for the model to classify.

Good choice of images!

I can suggest you to try your model on the following 5 very interesting images:



It is interesting how your model will handle them as each of them contains some additional elements:

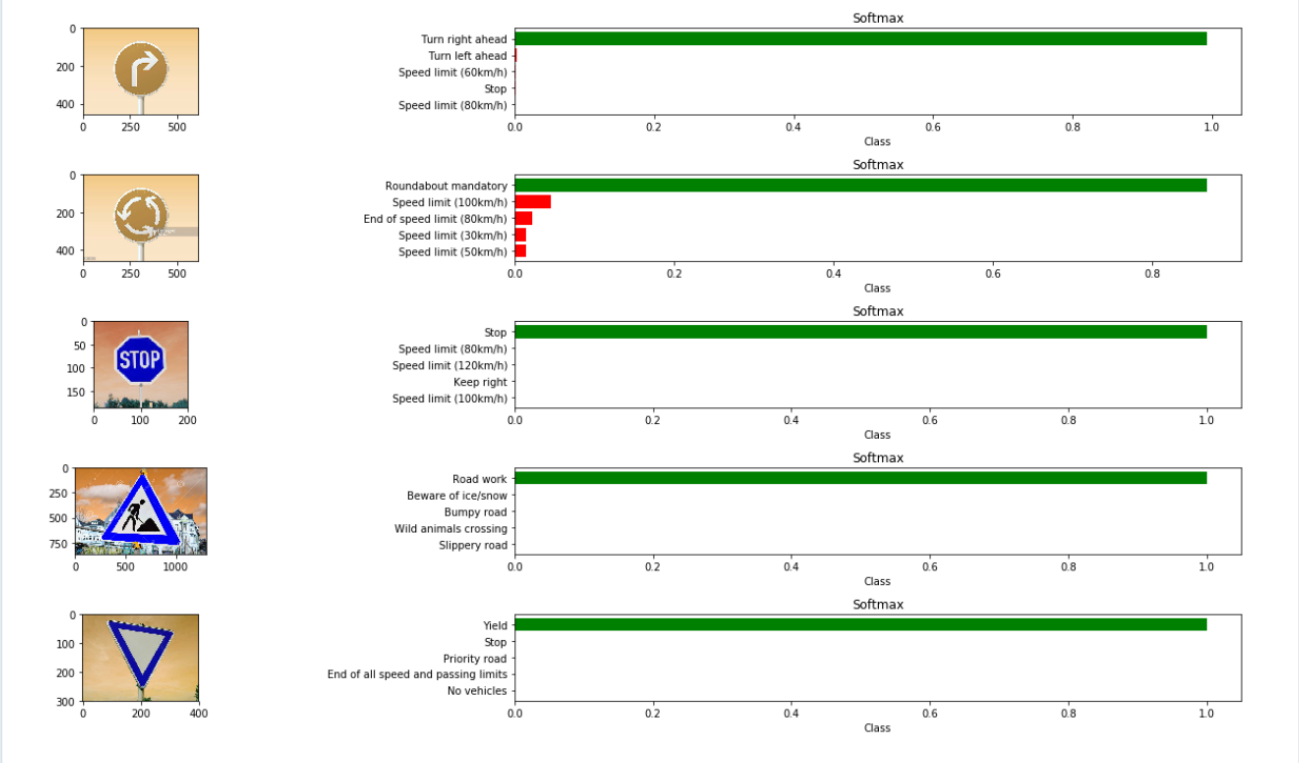
- #1 - a stop sign partially covered by tree leaves
- #2 - a stop sign with graffiti
- #3 - priority road sign with another round road sign on the back
- #4 - turn right ahead sign with orange tapes
- #5 - modified go straight or left sign

The submission documents the performance of the model when tested on the captured images. The performance on the new images is compared to the accuracy results of the test set.

The top five softmax probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

Good job explaining for which captured image example the model is certain or uncertain of its predictions.

You can also visualize top five probabilities using barchart like here:



[↓ DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)