Traffic Sign Recognition
Report by Cristian Cucchiella

Objective

In this project I built a convolutional neural network using tensorflow with the goal to classify traffic signs from images. The model was trained using the German Traffic Sign Dataset.

- The project consisted of the following steps:
- Dataset exploration
- Preprocessing and model architecture and design
- Training and testing of the model
- Usage of the model to make predictions on new images
- Analysis of results on the new images
- Visual exploration of the model activations

Rubric Points

Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

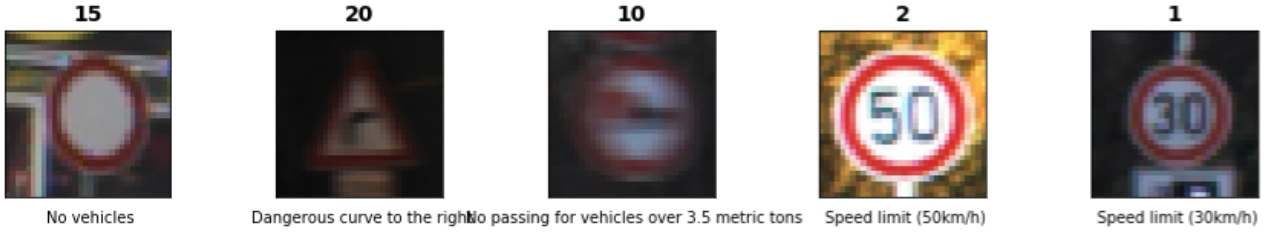
Writeup / README

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.

The size of training set is 34799
The size of the validation set is 4410
The size of test set is 12630
The shape of a traffic sign image is (32, 32, 3)
The number of unique classes in the data set is 43

2. Include an exploratory visualization of the dataset.

Training dataset

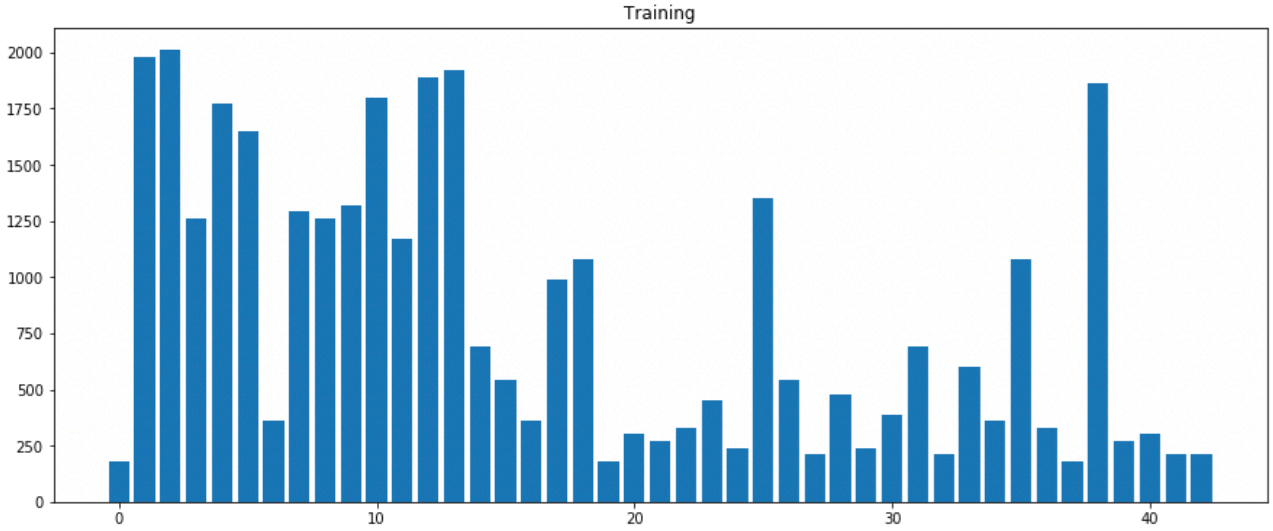


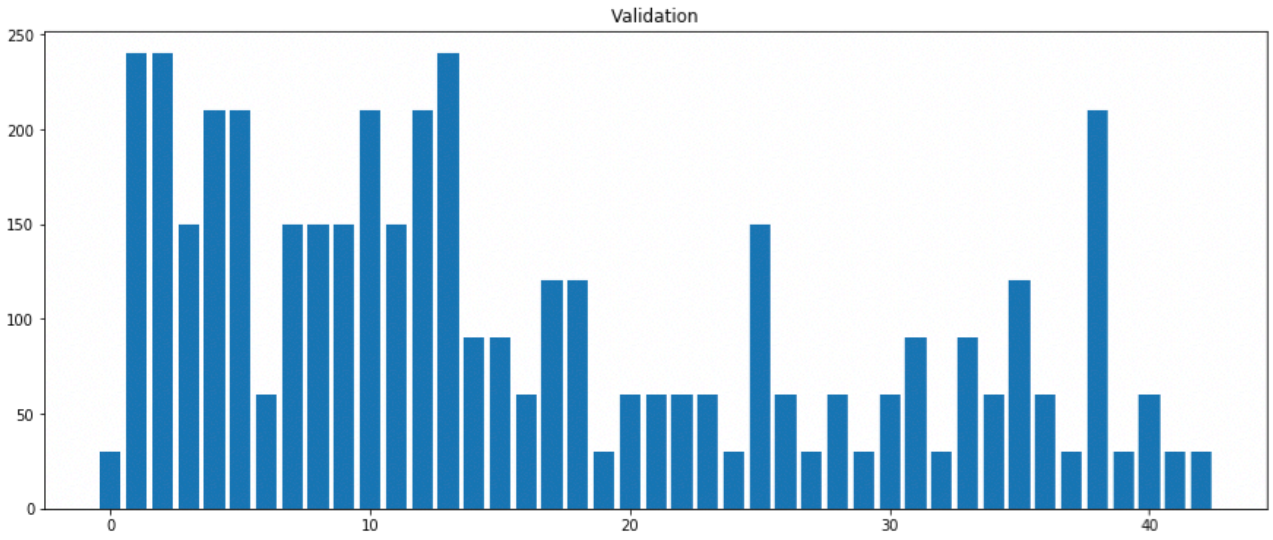
Validation dataset

Test dataset

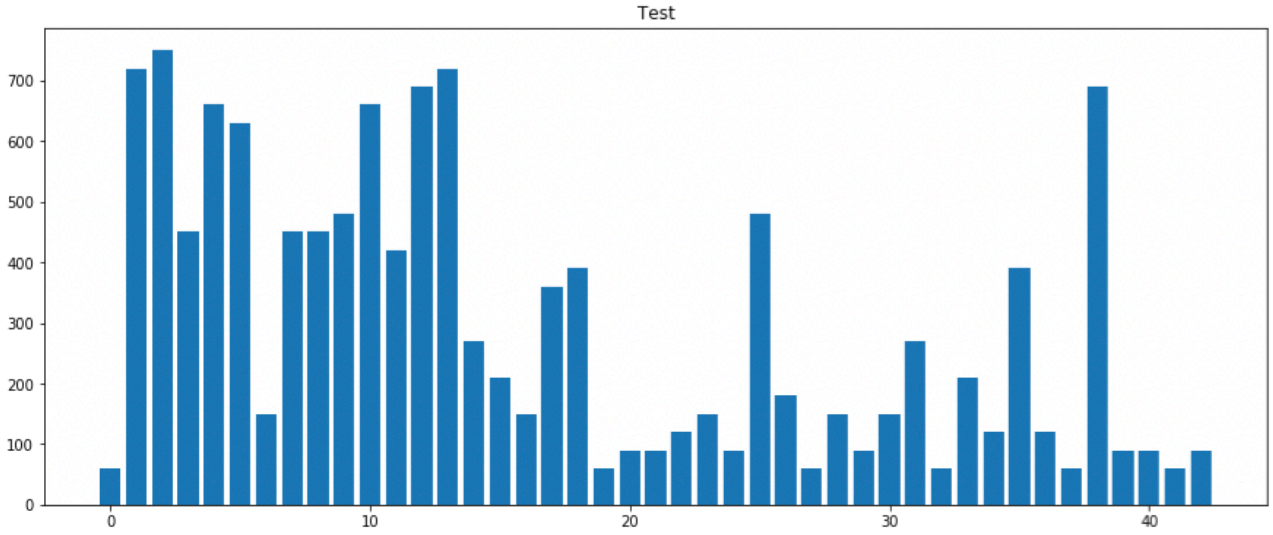
Because of a brightness shift, we need to normalize the images or play around with the different color channels in a pre-processing step.

It follow plots of the distribution on a bar chart





The distribution of the classes is not uniform and contains very few examples, this may lead to poor performance on any model. This is why it is important to augment the dataset.



Samples' distribution is the same among the various datasets.

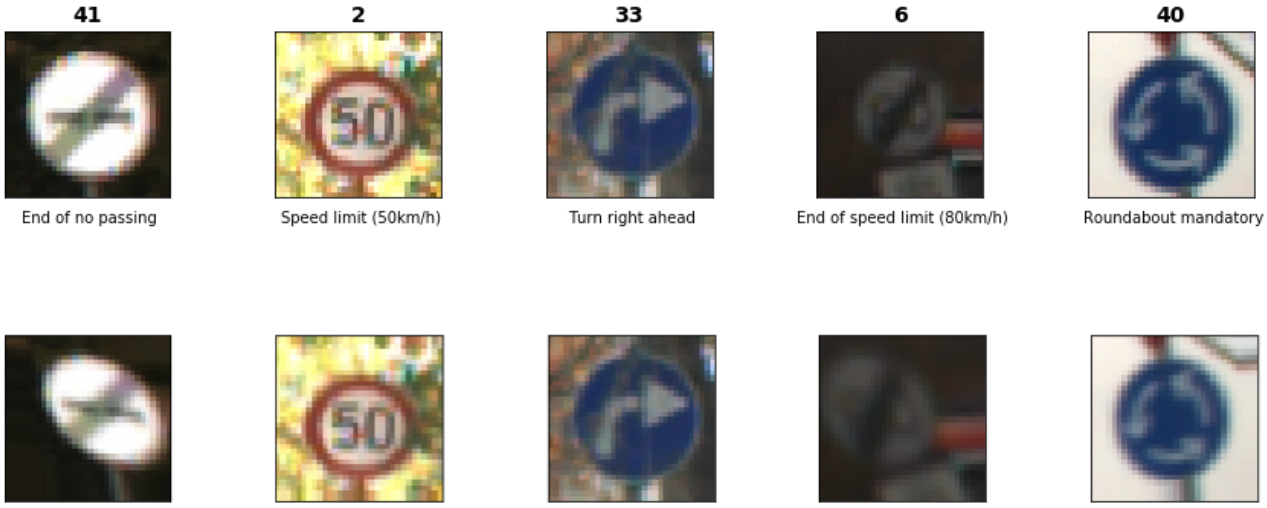
Design and Test a Model Architecture

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.)

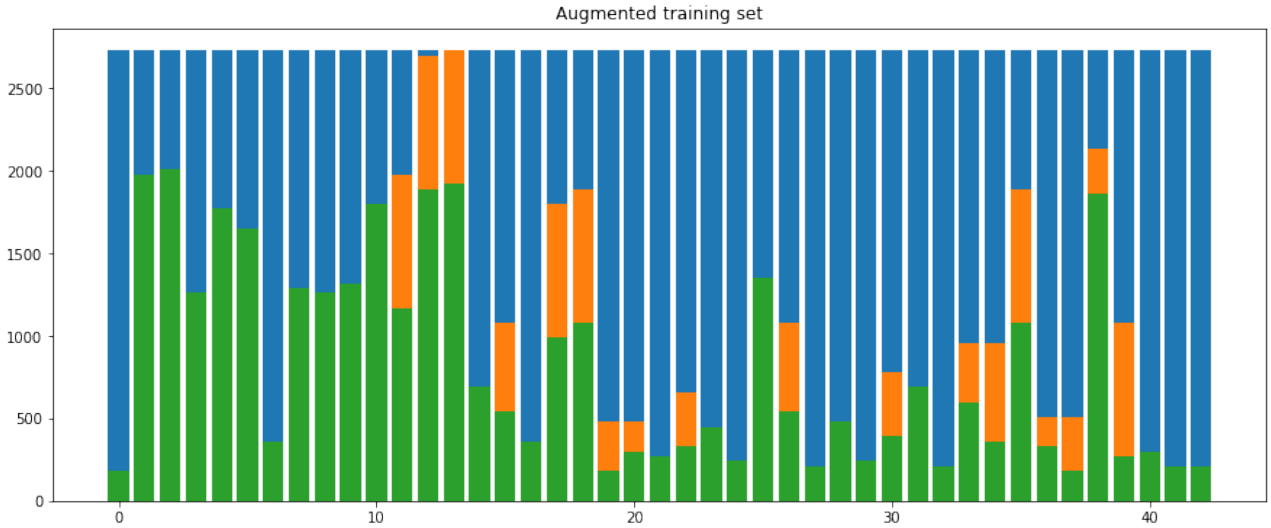
The images have a strong shift in brightness. In order to fix the contrast we can equalization, we try using Contrast Limited Adaptive Histogram Equalization (CLAHE): https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html

Data Augmentation
The distribution of classes is not uniform and there are classes with very few examples. Some images can be mirrored retaining the same class (yield, stop, ...), while others can be mirrored but their class is swaped (keep left vs keep right). First of all, mirror horizontally those classes and add it back.

After mirror, I used the following techniques: scaling, translation, perspective transformation, image rotation, gamma variation.



New distribution after dataset augmentation



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.

Layer	Description
Input	32x32x3 RGB image
Convolution 3x3	1x1 stride, same padding, outputs 32x32x32
RELU	
Convolution 3x3	1x1 stride, same padding, outputs 32x32x32
RELU	
Max pooling	2x2 stride, outputs 16x16x32
Convolution 3x3	1x1 stride, same padding, outputs 16x16x64
RELU	
Convolution 3x3	1x1 stride, same padding, outputs 16x16x64
RELU	
Max pooling	2x2 stride, outputs 8x8x64
Convolution 3x3	1x1 stride, same padding, outputs 8x8x128
RELU	
Convolution 3x3	1x1 stride, same padding, outputs 8x8x128
RELU	
Max pooling	2x2 stride, outputs 4x4x128
Fully Connected	outputs 2048x512
Dropout	
Logits	outputs 512x43

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.

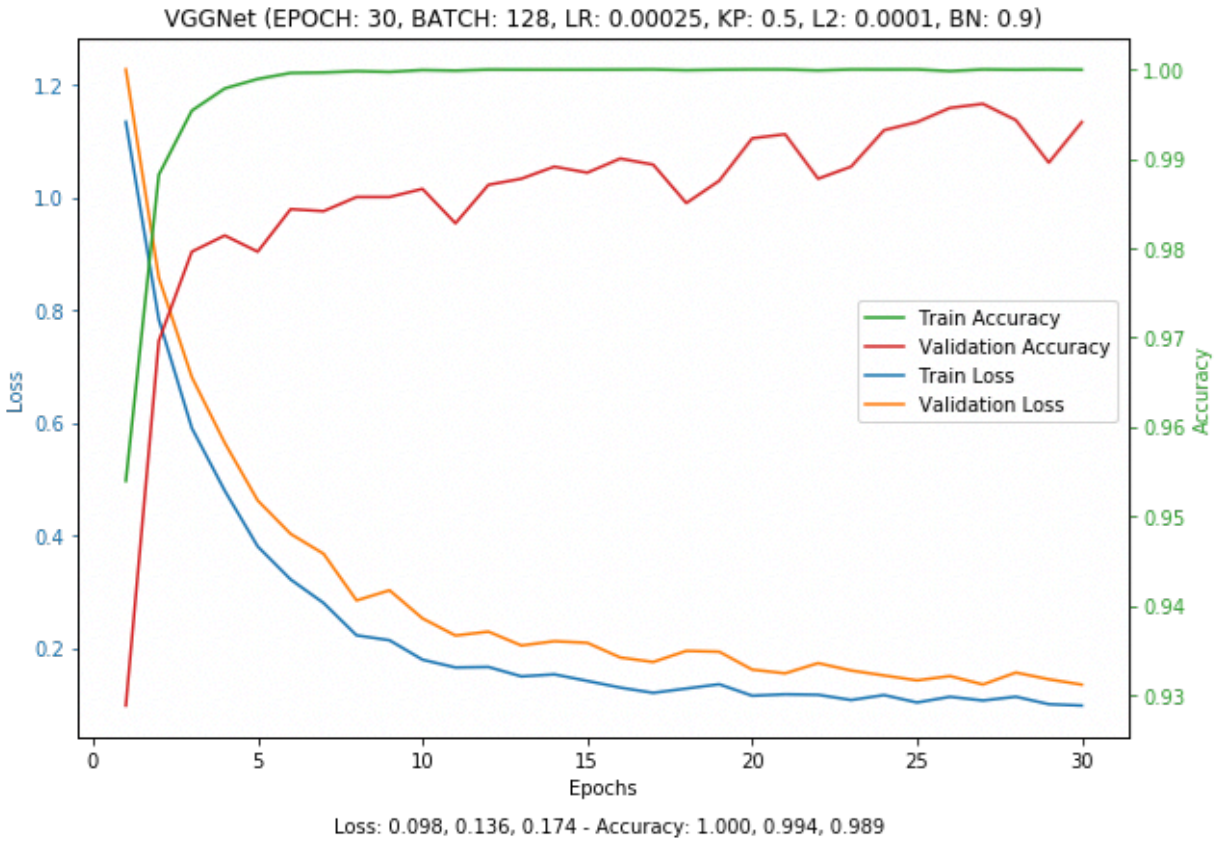
I use a class that allowed me to be flexible in trying out several different techniques for improving the model. After the initial experimentation with the basic LeNet architecture I added L2 regularization, batch normalization and dropout for the fully connected layers. The model uses gradient descent with Adam optimization as it seems to provide good results and the hope was that it would need less tweaking for batch size and the learning rate. For the loss we use cross entropy that for the given dataset is a good choice since it is designed to computed the probability error in discrete classification tasks when the classes are mutually exclusive.

The model parameters:
Epochs: 30
Batch size: 128
Learning rate: 0.00025

Dropout probability (Fully connected): 0.5
L2 Beta: 0.0001
Batch Norm Decay: 0.9

Model results:
Loss: 0.098, 0.136, 0.174
Accuracy: 1.000, 0.994, 0.989

It follows a plot of the loss and accuracy recorder at each epoch

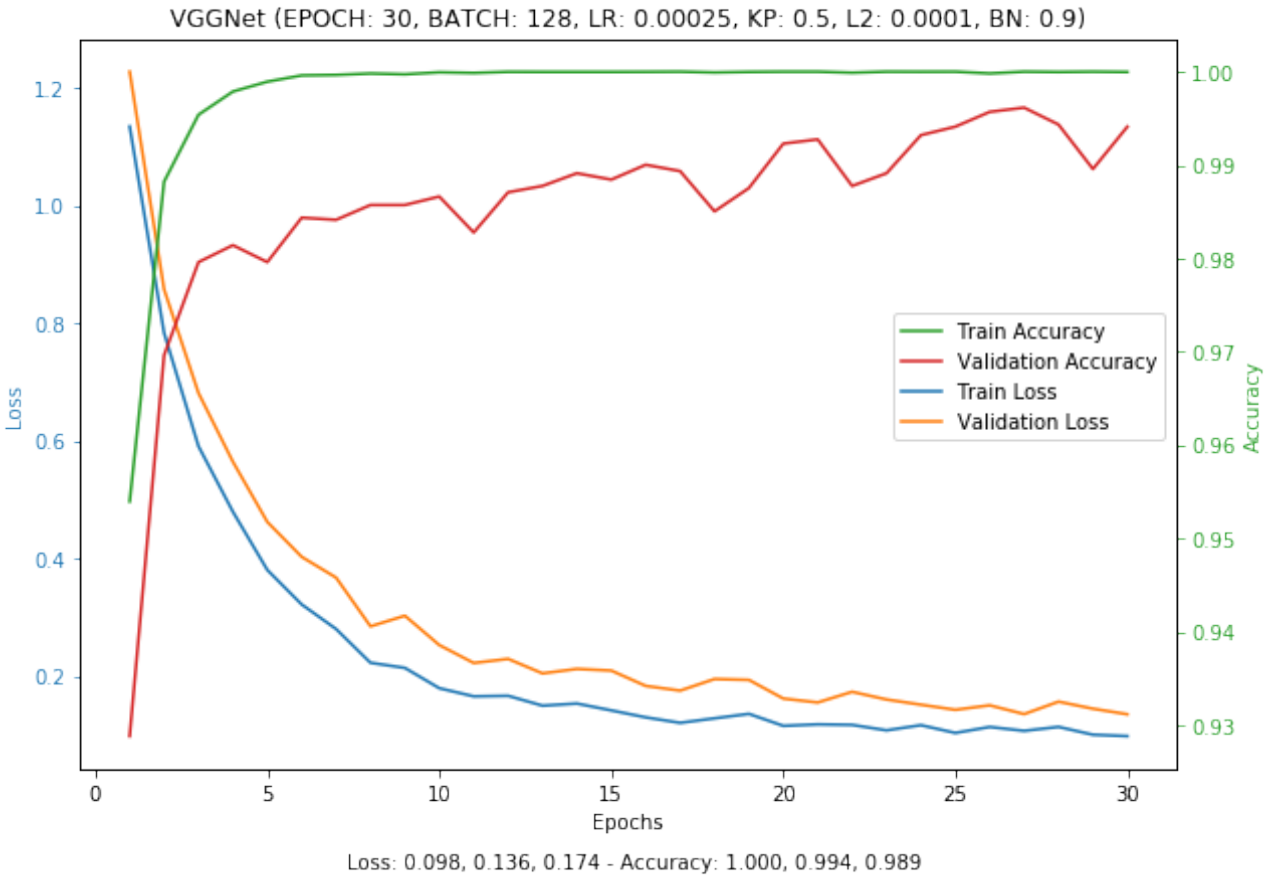


The model could have been trained longer because the trend of the accuracy is increasing and the loss trend is decreasing.

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.

After several experiments with different hyper parameters, I have been able to get the following results:

- training set accuracy of 1.0
- validation set accuracy of 0.994
- test set accuracy of 0.989



The model could have been trained longer because the trend of the accuracy is increasing and the loss trend is decreasing.

Test a Model on New Images

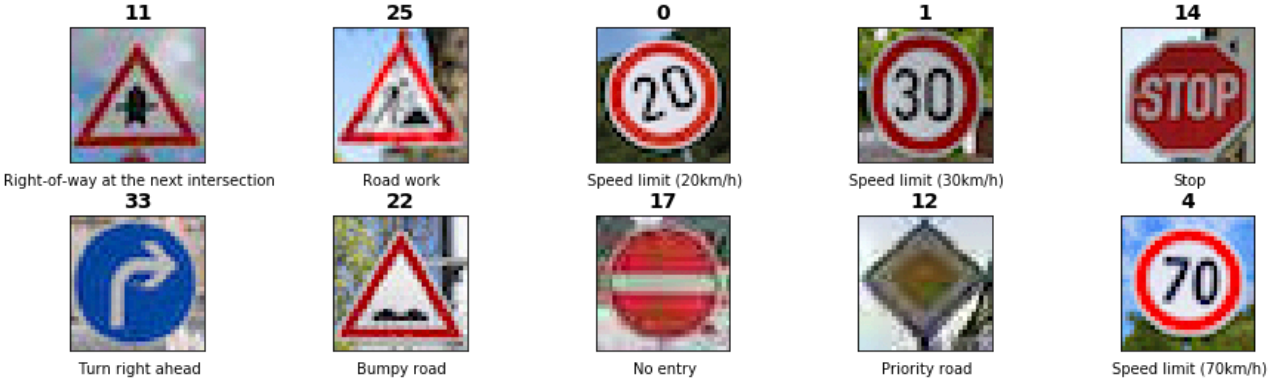
1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Here are some German traffic signs that I found on the web:

All the images have been cropped to 32x32.

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).

As expected the model classified all the images correctly:



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)

It follows the summary of the top 5 probabilities for each sign:

Prediction: 11 (Right-of-way at the next intersection) - Confidence: 0.947
Prediction: 27 (Pedestrians) - Confidence: 0.001
Prediction: 29 (Bicycles crossing) - Confidence: 0.001
Prediction: 28 (Children crossing) - Confidence: 0.001
Prediction: 40 (Roundabout mandatory) - Confidence: 0.001

Prediction: 25 (Road work) - Confidence: 0.931
Prediction: 22 (Bumpy road) - Confidence: 0.003
Prediction: 24 (Road narrows on the right) - Confidence: 0.003
Prediction: 27 (Pedestrians) - Confidence: 0.002
Prediction: 29 (Bicycles crossing) - Confidence: 0.002

Prediction: 0 (Speed limit (20km/h)) - Confidence: 0.334
Prediction: 8 (Speed limit (120km/h)) - Confidence: 0.123
Prediction: 4 (Speed limit (70km/h)) - Confidence: 0.041
Prediction: 24 (Road narrows on the right) - Confidence: 0.027
Prediction: 5 (Speed limit (80km/h)) - Confidence: 0.020

Prediction: 1 (Speed limit (30km/h)) - Confidence: 0.965
Prediction: 3 (Speed limit (60km/h)) - Confidence: 0.001
Prediction: 34 (Turn left ahead) - Confidence: 0.001
Prediction: 2 (Speed limit (50km/h)) - Confidence: 0.001
Prediction: 29 (Bicycles crossing) - Confidence: 0.001

Prediction: 14 (Stop) - Confidence: 0.917
Prediction: 8 (Speed limit (120km/h)) - Confidence: 0.002
Prediction: 5 (Speed limit (80km/h)) - Confidence: 0.002
Prediction: 29 (Bicycles crossing) - Confidence: 0.002
Prediction: 18 (General caution) - Confidence: 0.002

Prediction: 33 (Turn right ahead) - Confidence: 0.891
Prediction: 35 (Ahead only) - Confidence: 0.003
Prediction: 26 (Traffic signals) - Confidence: 0.003
Prediction: 11 (Right-of-way at the next intersection) - Confidence: 0.003
Prediction: 24 (Road narrows on the right) - Confidence: 0.003

Prediction: 22 (Bumpy road) - Confidence: 0.842
Prediction: 15 (No vehicles) - Confidence: 0.008
Prediction: 26 (Traffic signals) - Confidence: 0.006
Prediction: 24 (Road narrows on the right) - Confidence: 0.005
Prediction: 30 (Beware of ice/snow) - Confidence: 0.004

Prediction: 17 (No entry) - Confidence: 0.820
Prediction: 14 (Stop) - Confidence: 0.006
Prediction: 8 (Speed limit (120km/h)) - Confidence: 0.005
Prediction: 29 (Bicycles crossing) - Confidence: 0.005
Prediction: 25 (Road work) - Confidence: 0.005

Prediction: 12 (Priority road) - Confidence: 0.871
Prediction: 11 (Right-of-way at the next intersection) - Confidence: 0.003
Prediction: 29 (Bicycles crossing) - Confidence: 0.003
Prediction: 28 (Children crossing) - Confidence: 0.003
Prediction: 15 (No vehicles) - Confidence: 0.003

Prediction: 4 (Speed limit (70km/h)) - Confidence: 0.892
Prediction: 0 (Speed limit (20km/h)) - Confidence: 0.003
Prediction: 7 (Speed limit (100km/h)) - Confidence: 0.003
Prediction: 11 (Right-of-way at the next intersection) - Confidence: 0.003
Prediction: 37 (Go straight or left) - Confidence: 0.003

The confidence of the predictions is relatively high for the correct prediction and very low for the ones that are not choosen. The model is generalizing on the subject

Resources

https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html
https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html
https://github.com/marcomarasca/CarND-Traffic-Sign-Classifier