Writeup

April 7, 2020

1 Traffic Sign Recognition

1.1 Writeup

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

1.2 Rubric Points

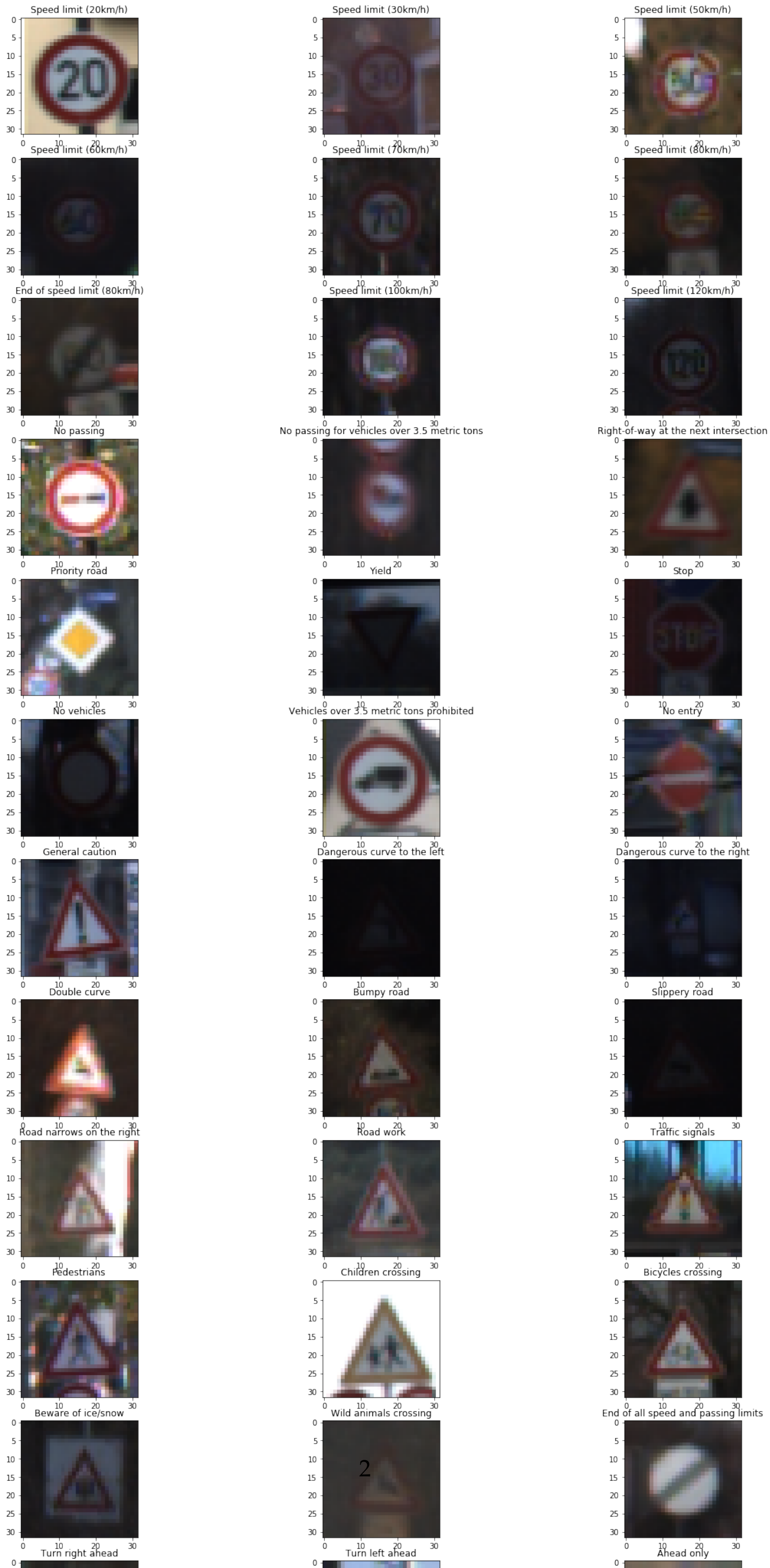
1.2.1 Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

1.2.2 Data Set Summary & Exploration

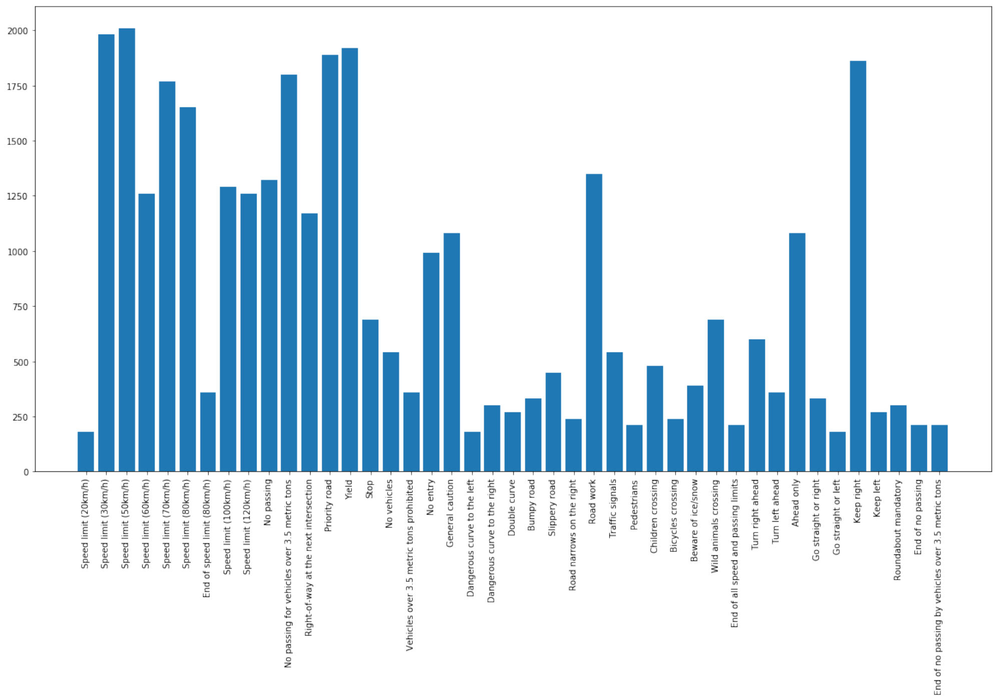
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually. I used the pandas library to calculate summary statistics of the traffic signs data set:

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

2. Include an exploratory visualization of the dataset. As an inital overview here is an impression of example images for each class:


Speed limit (20km/h)
Speed limit (30km/h)
Speed limit (50km/h)
Speed limit (60km/h)
Speed limit (70km/h)
Speed limit (80km/h)
End of speed limit (80km/h)
Speed limit (100km/h)
Speed limit (120km/h)
No passing
No passing for vehicles over 3.5 metric tons
Right-of-way at the next intersection
Priority road
Yield
Stop
No vehicles
Vehicles over 3.5 metric tons prohibited
No entry
General caution
Dangerous curve to the left
Dangerous curve to the right
Double curve
Bumpy road
Slippery road
Road narrows on the right
Road work
Traffic signals
Pedestrians
Children crossing
Bicycles crossing
Beware of ice/snow
Wild animals crossing
End of all speed and passing limits
Turn right ahead
Turn left ahead
Ahead only

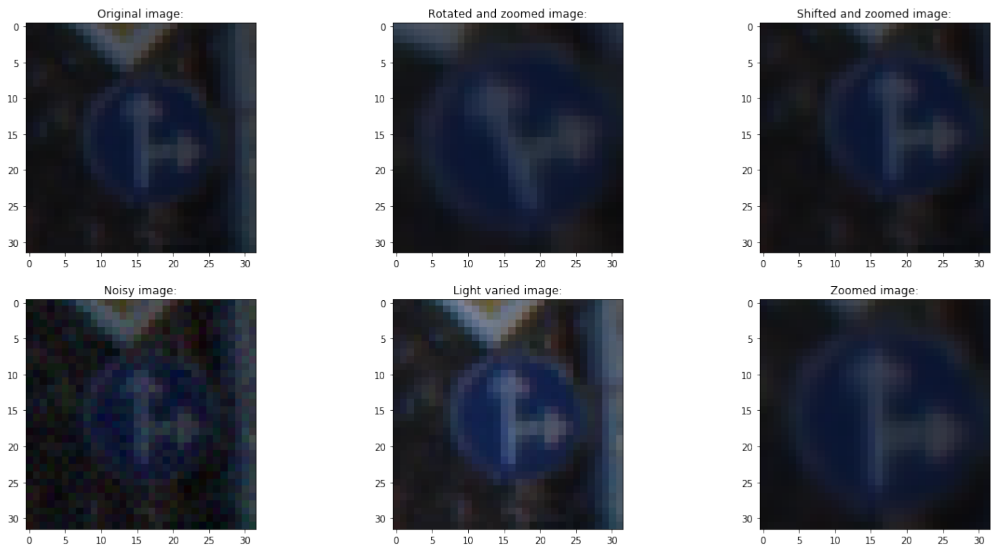
Here is an exploratory visualization of the data set. It is a bar chart showing how the data is distributed in over each label



alt text

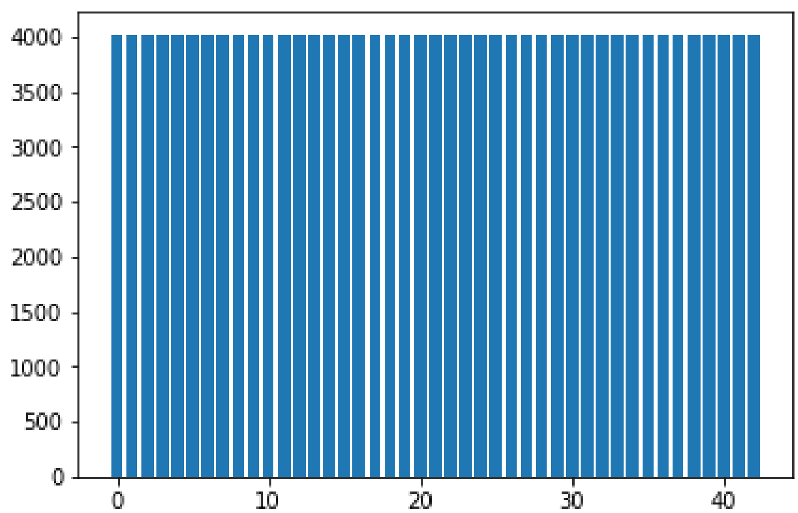
1.2.3 Design and Test a Model Architecture

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 equalized the training dataset so that each class of image has the same number of datapoints, so the dataset now has even distribution. This is done by adding augmented images on the basis of the already existing ones of each class. Here is an example:



alt text

I not only filled up classes with less images I basically extended the amount of data even for the class with the most original images by factor two. So the new distribution with about 175000 images looks like:



alt text

Finally I converted the images to the YUV colorspace for further processing as recommended by Lecun et. al.. Then the Y-Channel is equalized so the maximum Y-Value (Grayscale) for each image is 255 and the lowest 0:

alt text

At last all channels of the images are converted to a range -1 to 1 for better convergency properties of the fitting algorithm of the neural network, as given in the lessons.

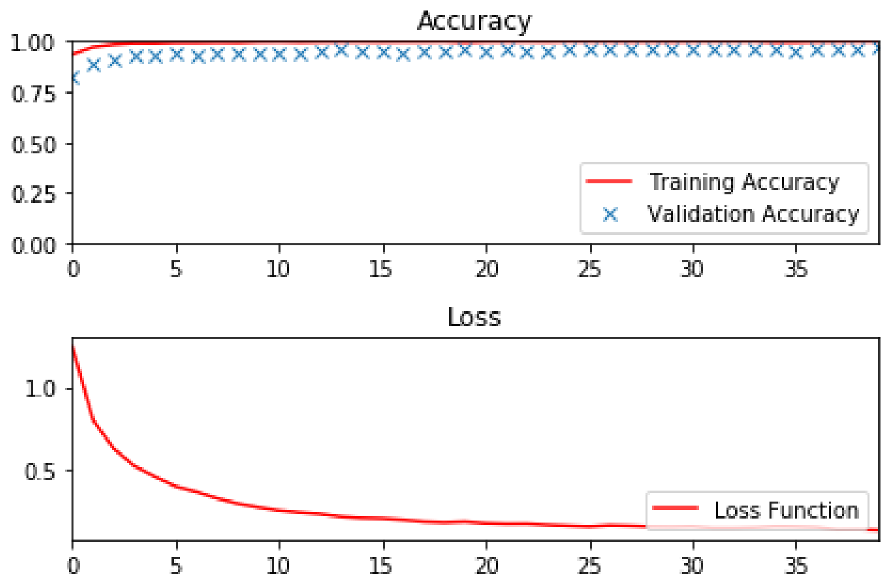
2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model. As given in Lecun et. al. a simplified variant of the given architectures is used here. This network is chosen as it already gave pretty good results for a traffic sign detection usecase.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 YUV image
Convolution 1 5x5	1x1 stride, same padding, outputs 28x28x30 for Y Channel (Input: 32x32x1) and 28x28x8 for the UV Channel (Input: 32x32x2)
70% Dropout 1	
RELU 1	
Max pooling 1	2x2 stride, outputs 14x14x38
Convolution 2 5x5	Output 10x10x64
70% Dropout 2	
Max pooling 2	2x2 stride, Input from Convolution 2, outputs 5x5x64
Fully connected 1	Input: 1600; Output: 200
Fully connected 2	Input: 200; Output: 43
Softmax	End of Graph

For instance due to the limited data set, the number of convolutions are set to a mimimum. However, a similar combination gave valid results in the mentioned paper. Additionally a dropout in the second layer is introduced to improve robustness. For the sake of computational performance, the parallel feedtrough of the first layer into the fully-connected hidden layer is ommit as it would introduce plenty of additional parameter.

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 used the adam-optimization as suggested in the lessons. A batch size of 512 proved to be useful as it allowed me to train the model at my local machine. The optimization of the network for the augemented training data is done using almmost an infinite number of epoches. The termination criteria here was the accuracy of the augmentated data, which should be less than 0.5%. The results of the accuracy and loss can be seen here:



alt text

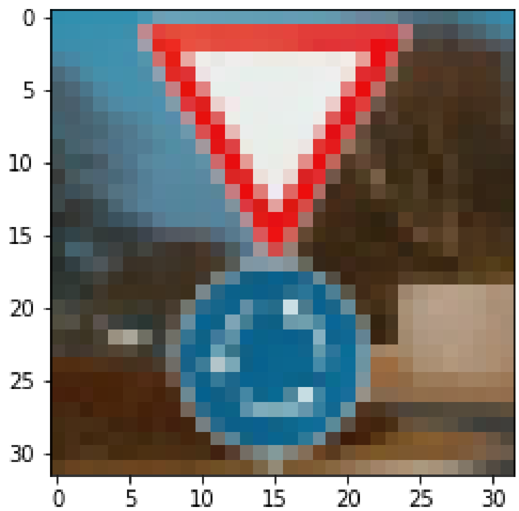
After training the augmented data, i combined the validation data in order to get better results for the test data. From my point of view this is reasonable, as the would throw valuable data away. The retraining results are visible here:

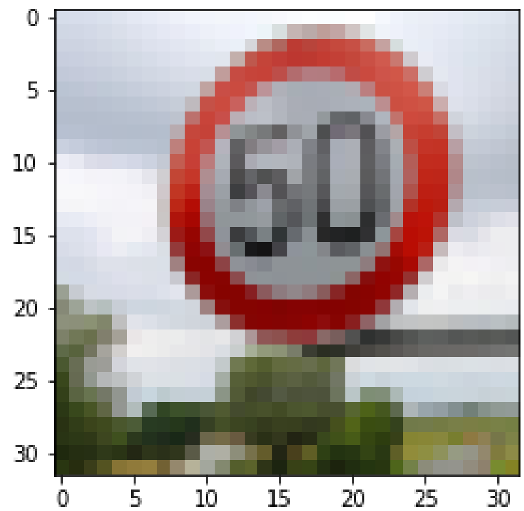
alt text

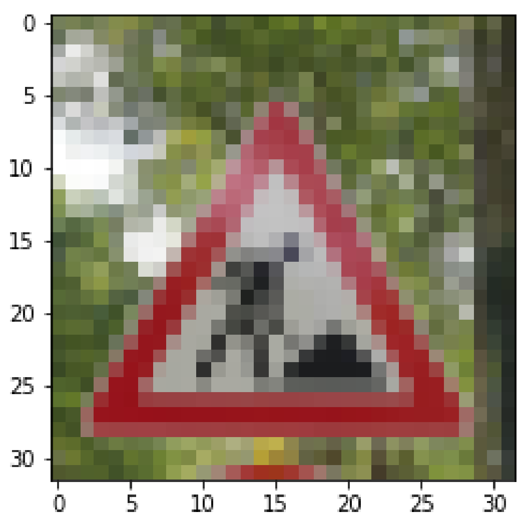
4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, 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 0.999 * training set augmented accuracy of 0.996 * validation set accuracy of 0.972 * training set augmented + validation set accuracy of 0.993 * test set accuracy of 0.963

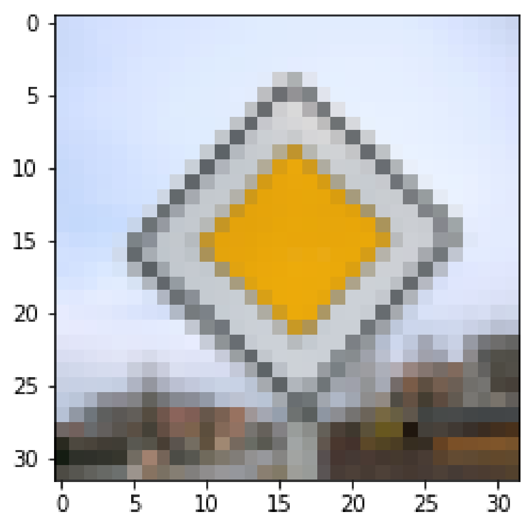
1.2.4 Test a Model on New Images

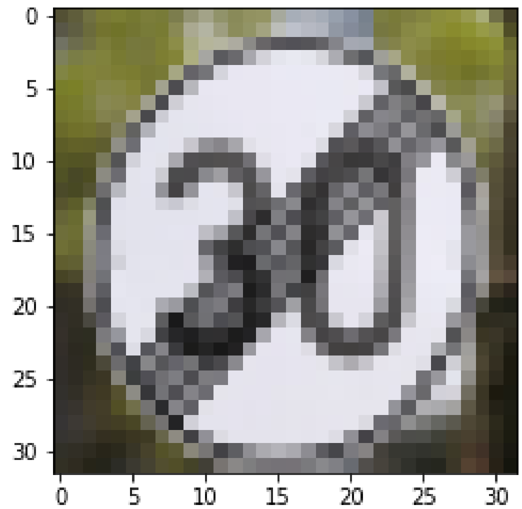
1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify. Here are five German traffic signs that I found on the web:











The selected images are meant to be challenging. One of the five images containts two traffic signs in one. This is, because I wanted to see, if the algorithm detects either one of them of fails completely. The some of the other images are view from a different perspective, which was not covered as augmentation in the enlargment procedure of the dataset.

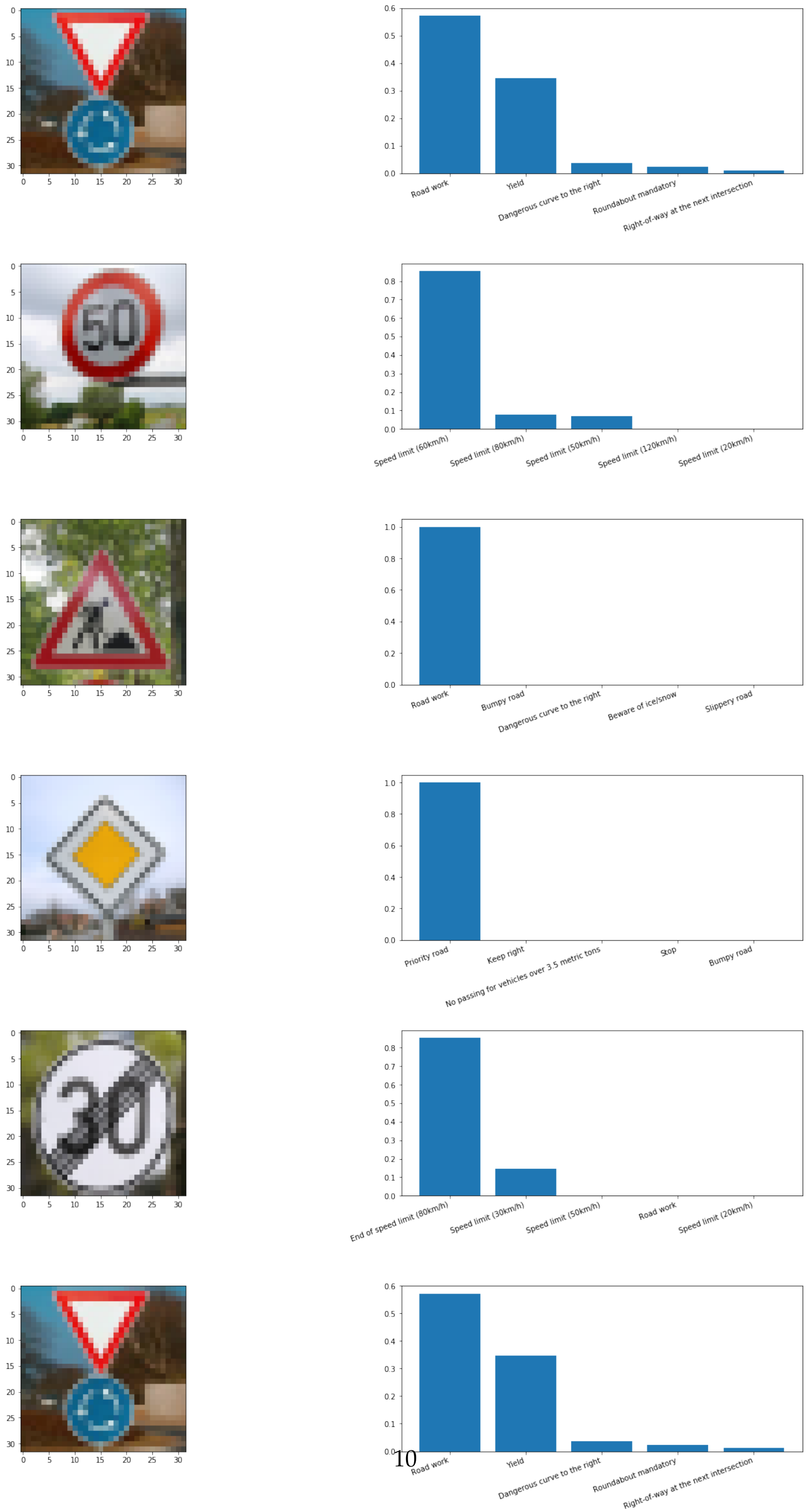
2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric). Here are the results of the prediction:

Image	Prediction
Yield + Roundabout	Road work
Speed-Limit 50km/h	Speed-Limit 60km/h
Road work	Road work
Priority road	Priority road
End of speed limit (30 km/h) (not included in labels)	End of speed limit (80 km/h)

The model was able to correctly guess 2 of the 5 traffic signs, and got it very close at the speed-limit and end-of-speed-limit signs while disagreeing with the numbers. The two-sign case was from this point of view not very satifying. This gives an accuracy of 80%, when looking from an optimistic point of view.

Probability	Prediction
0.57	Road work
0.83	Speed-Limit 60km/h
0.99	Road work
0.99	Priority road
0.84	End of speed limit (80 km/h)

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 returned probabilities pretty distinct for the correct detected images. For the miss-predicted 50km/h sign the net returned at least a low probability, that it could be a 50km/h sign. However, it detected that it actually is a speed limit sign. For the two-signs-in-one-image case the net did not find the correct answer, but the yield-sign was estimated with the second best probability and the roundabout-sign with the 4th best probability. As this case was not trained before. I was pretty satisfied with the results.


Road work
Yield
Dangerous curve to the right
Roundabout mandatory
Right-of-way at the next intersection
Speed limit (60km/h)
Speed limit (80km/h)
Speed limit (50km/h)
Speed limit (120km/h)
Speed limit (20km/h)
Road work
Bumpy road
Dangerous curve to the right
Beware of ice/snow
Slippery road
Priority road
Keep right
No passing for vehicles over 3.5 metric tons
Stop
Bumpy road
End of speed limit (80km/h)
Speed limit (30km/h)
Speed limit (50km/h)
Road work
Speed limit (20km/h)
Road work
Yield
Dangerous curve to the right
Roundabout mandatory
Right-of-way at the next intersection

alt text