Traffic Sign Recognition

Writeup

Build a Traffic Sign Recognition Project

The goals / steps of this project are the following:

- Load the data set (German Traffic Sign Recognition Benchmark(GTSRB))
- 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 / 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.

Data Set Summary & Exploration

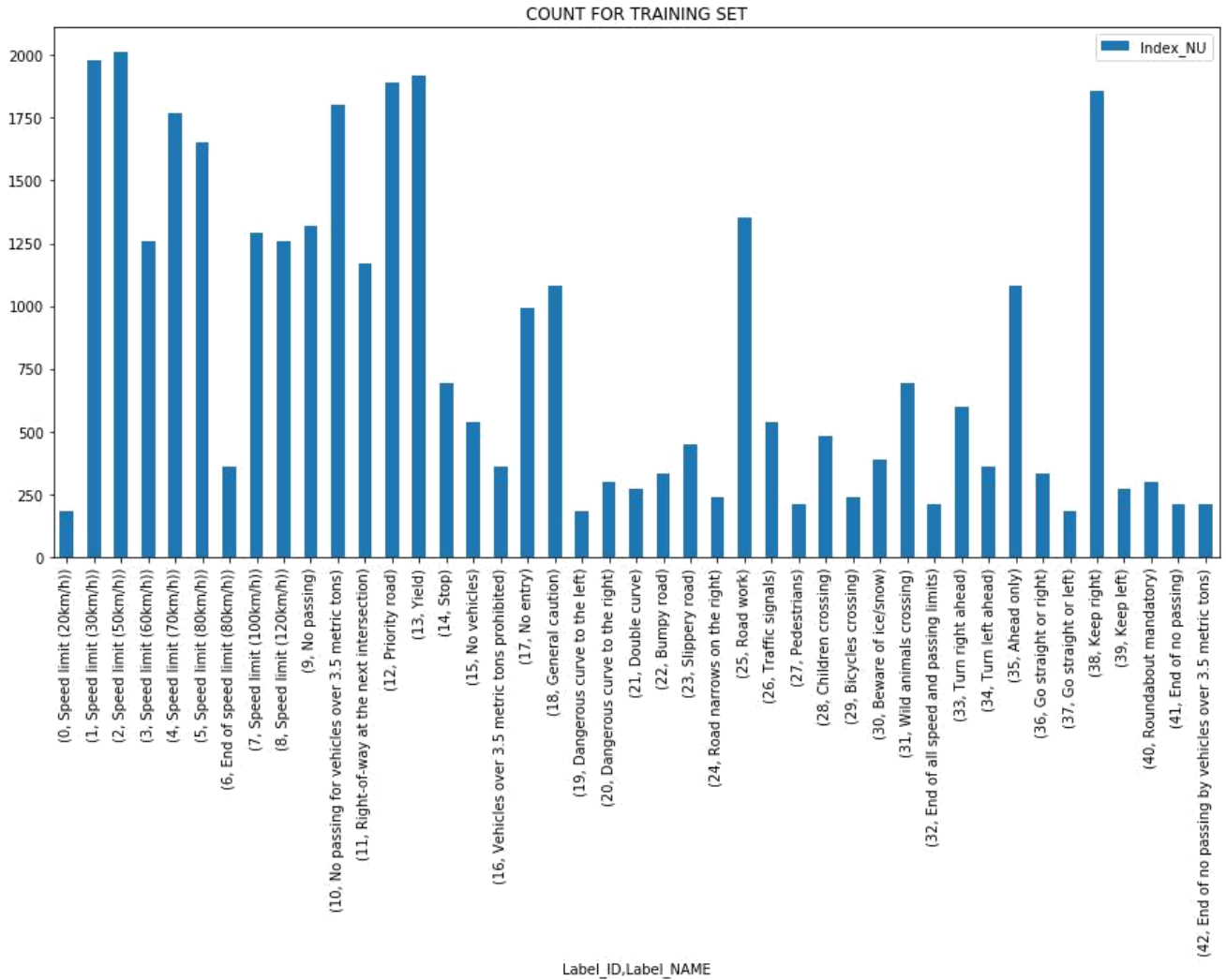
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

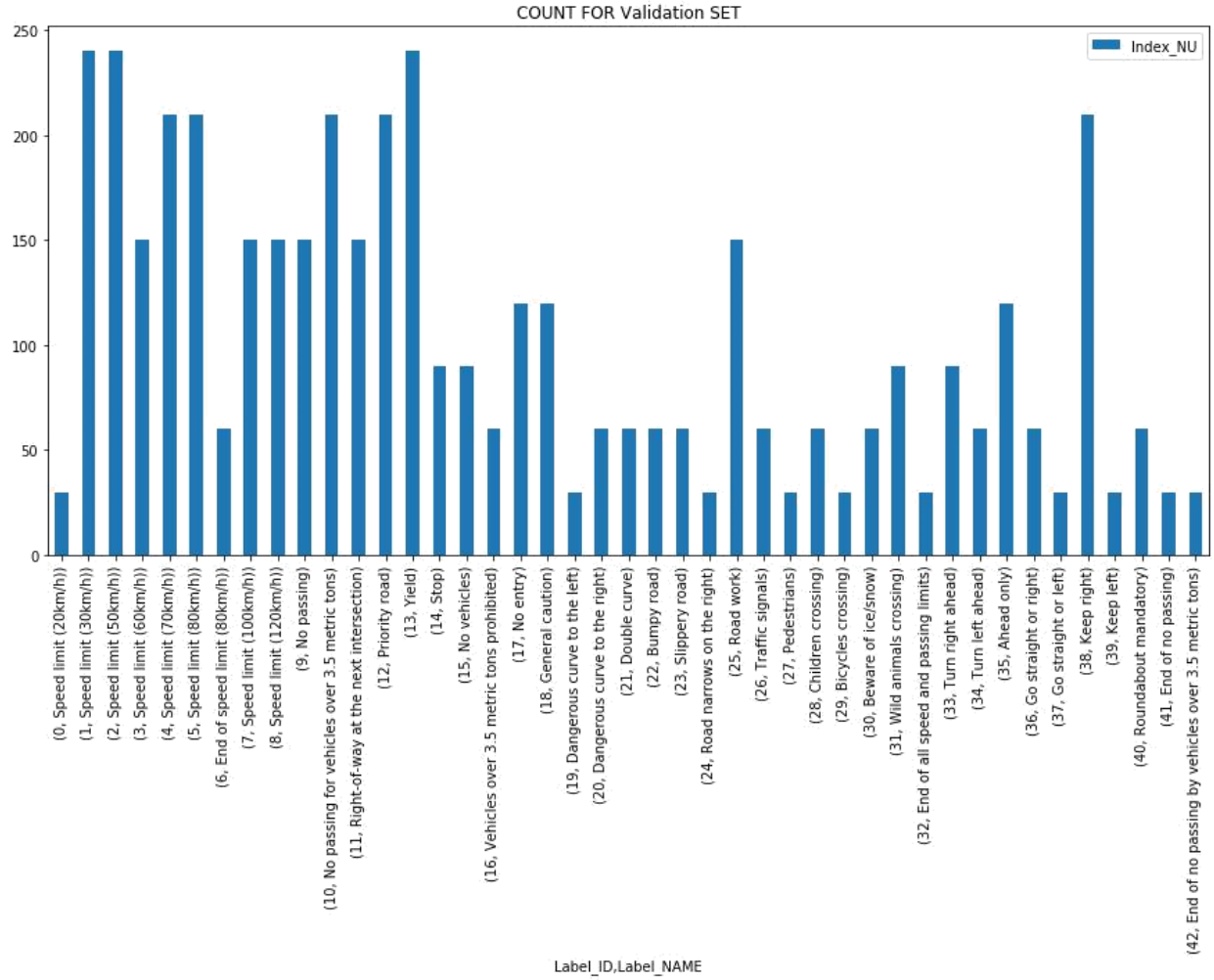
I used the pandas library to calculate summary statistics of the traffic signs data set:

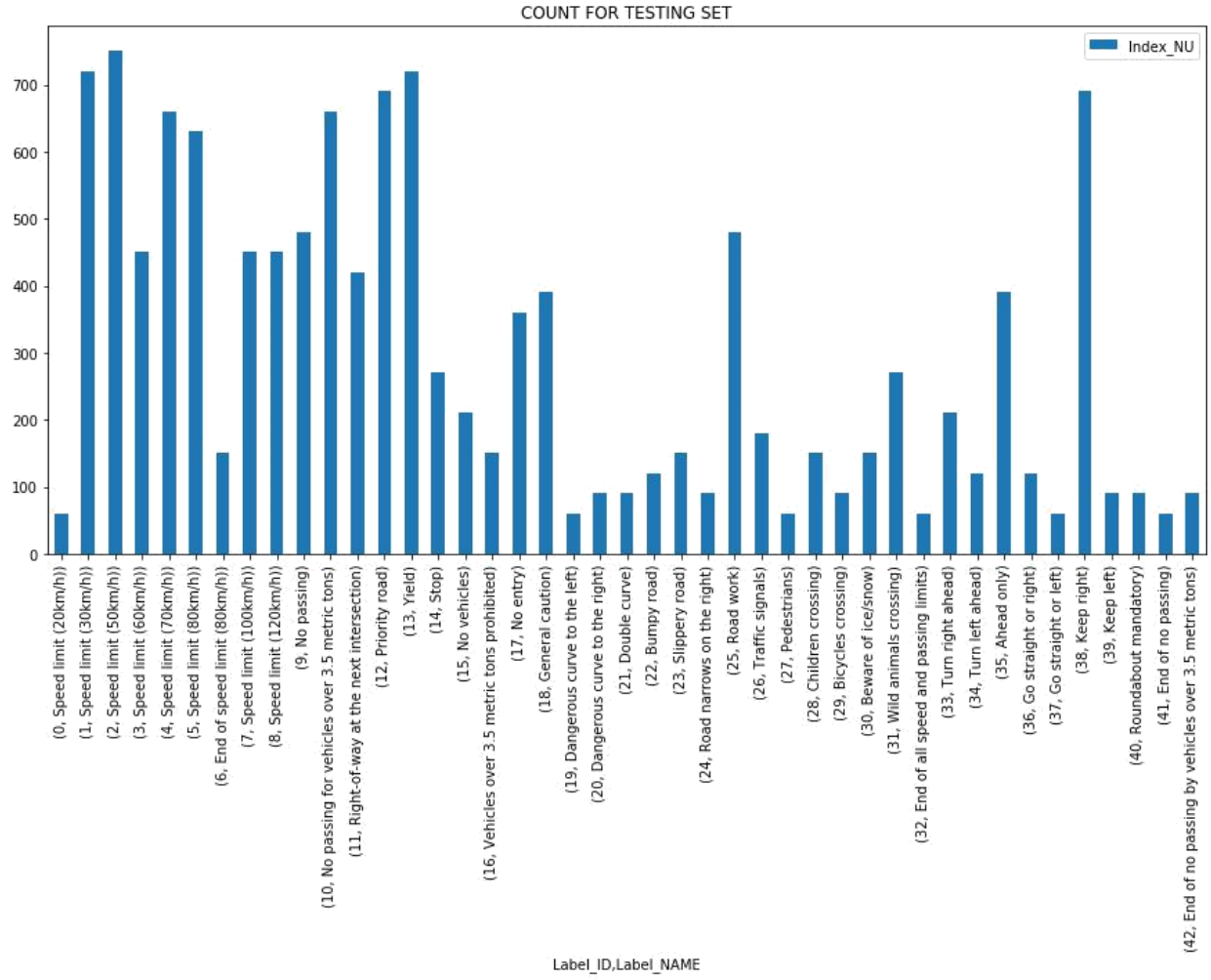
- The size of training set is 34799
- The size of test set is 12630
- The shape of a traffic sign image is (32, 32,3)
- The number of unique classes/labels in the data set is 43

2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set.




COUNT FOR Validation SET
Index_NU
250
200
150
100
50
0
(0, Speed limit (20km/h))
(1, Speed limit (30km/h))
(2, Speed limit (50km/h))
(3, Speed limit (60km/h))
(4, Speed limit (70km/h))
(5, Speed limit (80km/h))
(6, End of speed limit (80km/h))
(7, Speed limit (100km/h))
(8, Speed limit (120km/h))
(9, No passing)
(10, No passing for vehicles over 3.5 metric tons)
(11, Right-of-way at the next intersection)
(12, Priority road)
(13, Yield)
(14, Stop)
(15, No vehicles)
(16, Vehicles over 3.5 metric tons prohibited)
(17, No entry)
(18, General caution)
(19, Dangerous curve to the left)
(20, Dangerous curve to the right)
(21, Double curve)
(22, Bumpy road)
(23, Slippery road)
(24, Road narrows on the right)
(25, Road work)
(26, Traffic signals)
(27, Pedestrians)
(28, Children crossing)
(29, Bicycles crossing)
(30, Beware of ice/snow)
(31, Wild animals crossing)
(32, End of all speed and passing limits)
(33, Turn right ahead)
(34, Turn left ahead)
(35, Ahead only)
(36, Go straight or right)
(37, Go straight or left)
(38, Keep right)
(39, Keep left)
(40, Roundabout mandatory)
(41, End of no passing)
(42, End of no passing by vehicles over 3.5 metric tons)
Label_ID,Label_NAME

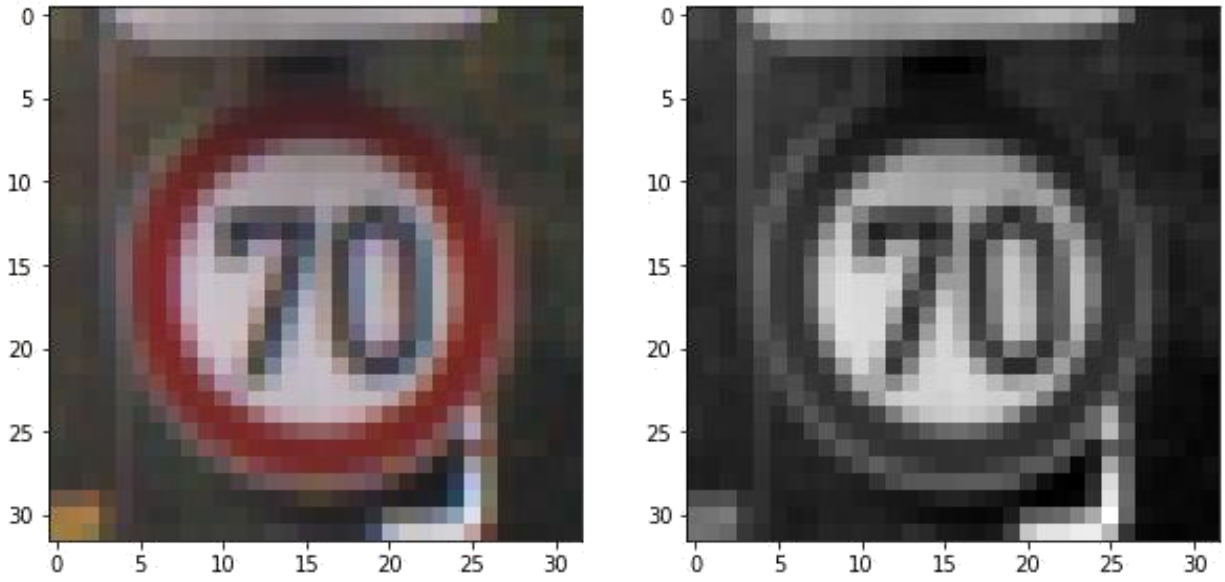


Design and Test a Model Architecture

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.)

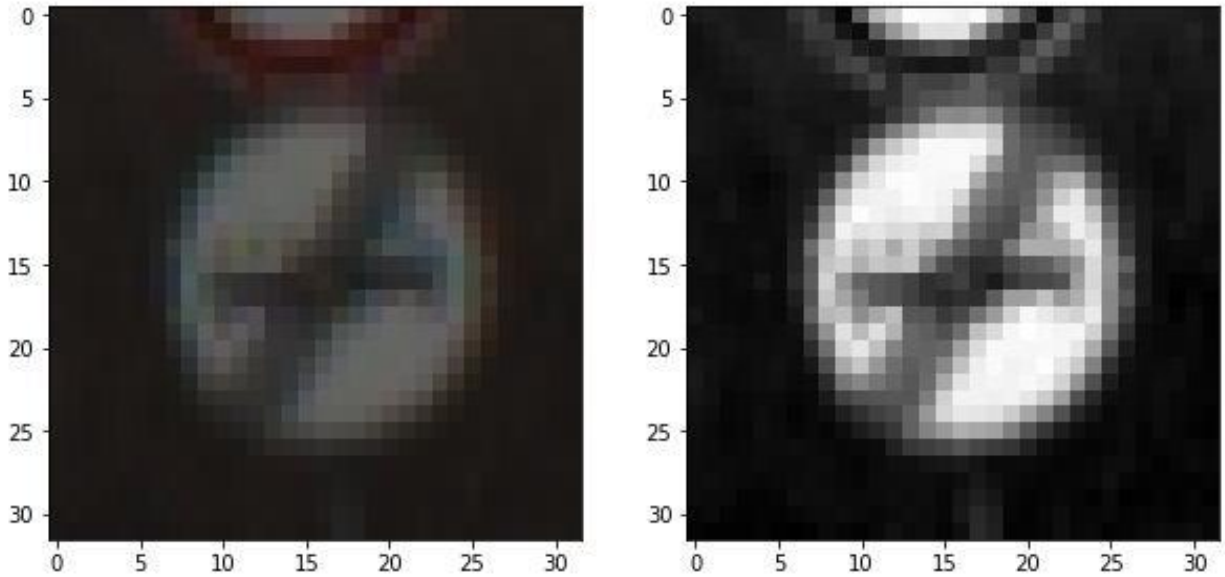
As a first step, I decided to convert the images to grayscale.

Here is an example of a traffic sign image before and after grayscaling.



As a last step, I normalized the image data to -1 to +1 range as the activations will be saturated with 0–255 range.

Here is an example of an original image and an augmented image:



The difference between the original data set and the augmented data set is that the images are gray scaled and then normalized.

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.

My final model consisted of the following layers:

Layer	Description
Input	32x32x1 RGB image
Convolution 5X5	1x1 stride, same padding, outputs 28X28X6
Activation	RELU
Max pooling	1x1 stride, outputs 14x14x6
Convolution 5x5	1x1 stride, same padding, outputs 10X10X16
Activation	RELU
Pooling	5X5X16
Flatten	400
Fully connected	Output 120
Activation	RELU
Fully Connected	Output 84
Activation	RELU
Fully Connected	Output 43

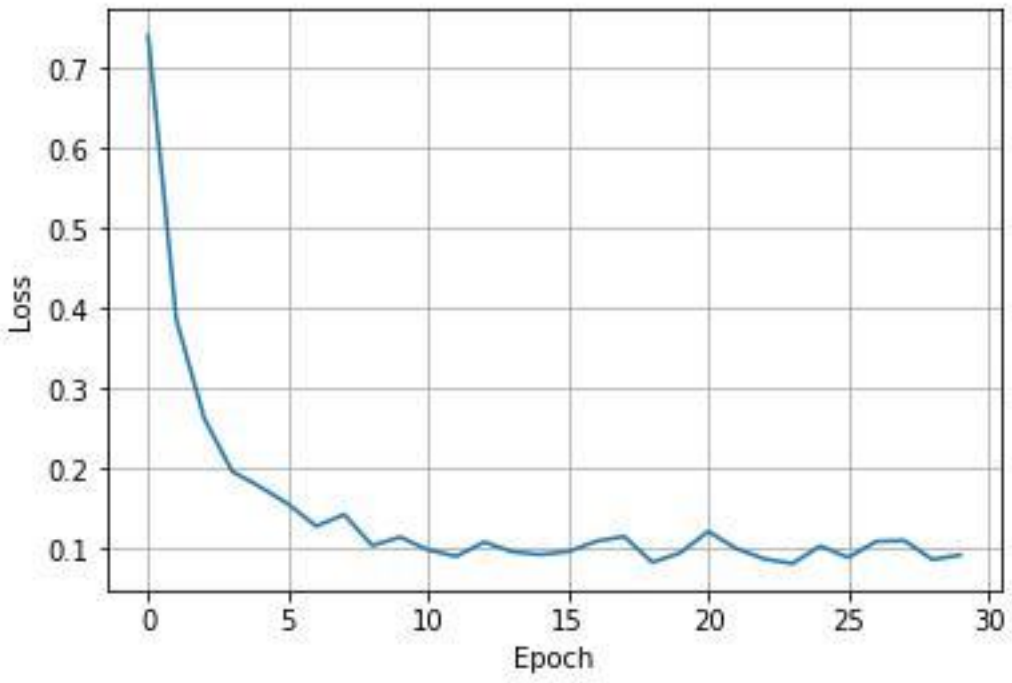
3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as l

I model was used Adam Optimizer along with 0.0009 learning rate, 30 Epochs and 100 Batch Size.

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:

- validation set accuracy of 0.982
- training set accuracy of



If a well known architecture was chosen:

- What architecture was chosen?

  I used the architecture referred b the instructors. LeNet by Lan Yacun. I preferred to stick with this architecture as it was giving good results.

- Why did you believe it would be relevant to the traffic sign application?

  Worked with it on the MNIST dataset and it gave good results after the training images are gray scaled and normalized and used with the LeNet architecture we can expect to have some promising results.

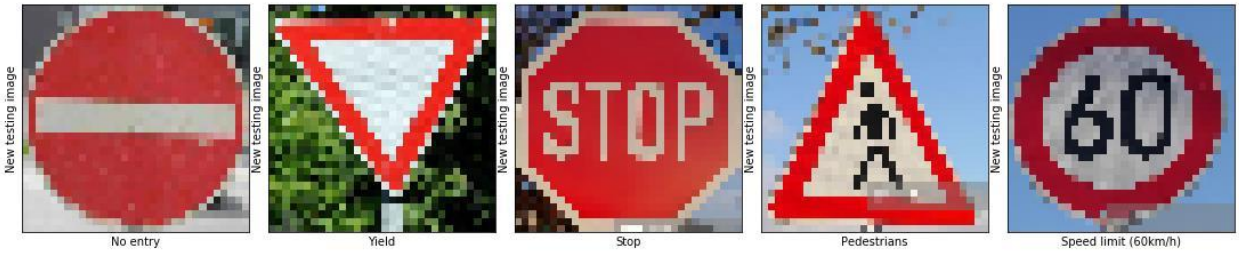
- How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

  The training set accuracy is 100% and the validation set accuracy is 0.988 thus it looks to overfit.

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.

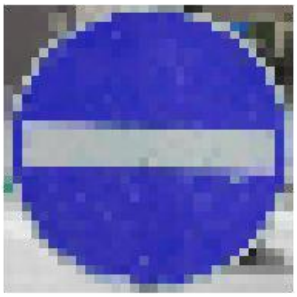
Here are five German traffic signs that I found on the web:

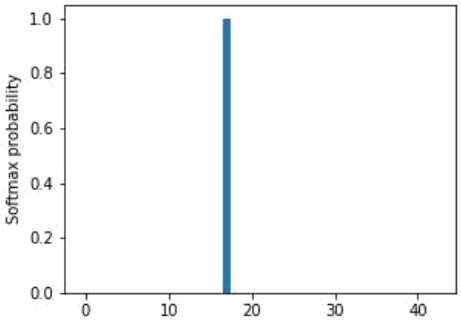


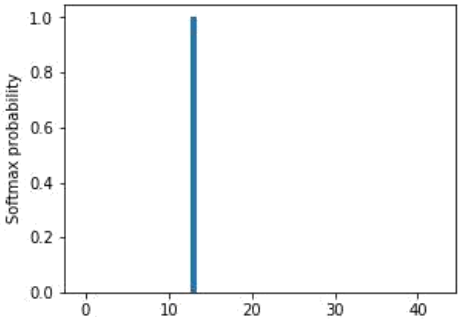
I used semi-easy images to classify and even modified them slightly. I grayscaled and normalized each image.

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).

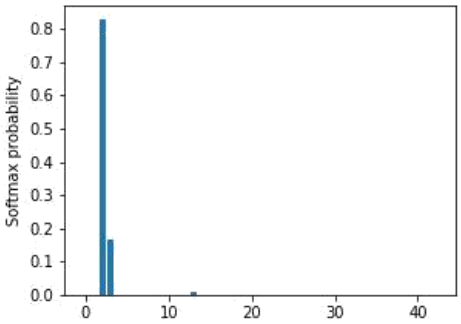
Here are the results of the prediction:

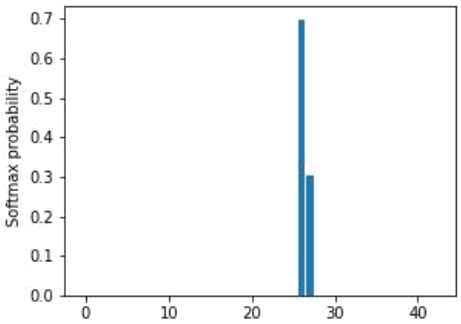



Softmax probability
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40

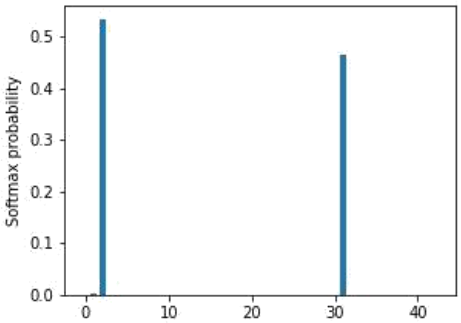

Softmax probability
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40


STOP


Softmax probability
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40


Softmax probability
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40


60


Softmax probability
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40