Traffic Sign Recognition

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

Rubric Points

Here I will consider the rubric points (https://review.udacity.com/#!/rubrics/481/view) individually and describe how I addressed each point in my implementation.

Data Set Summary & Exploration

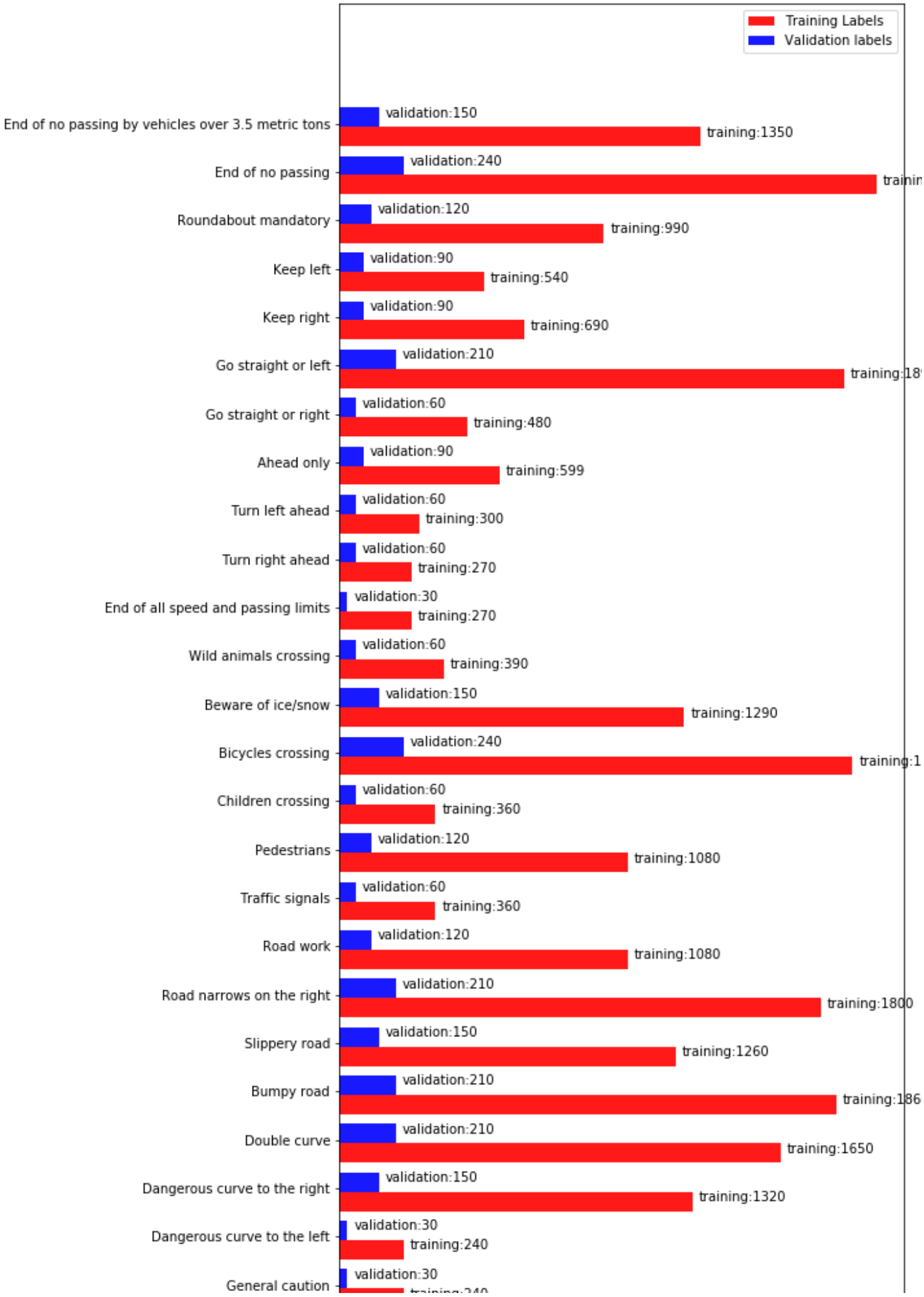
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

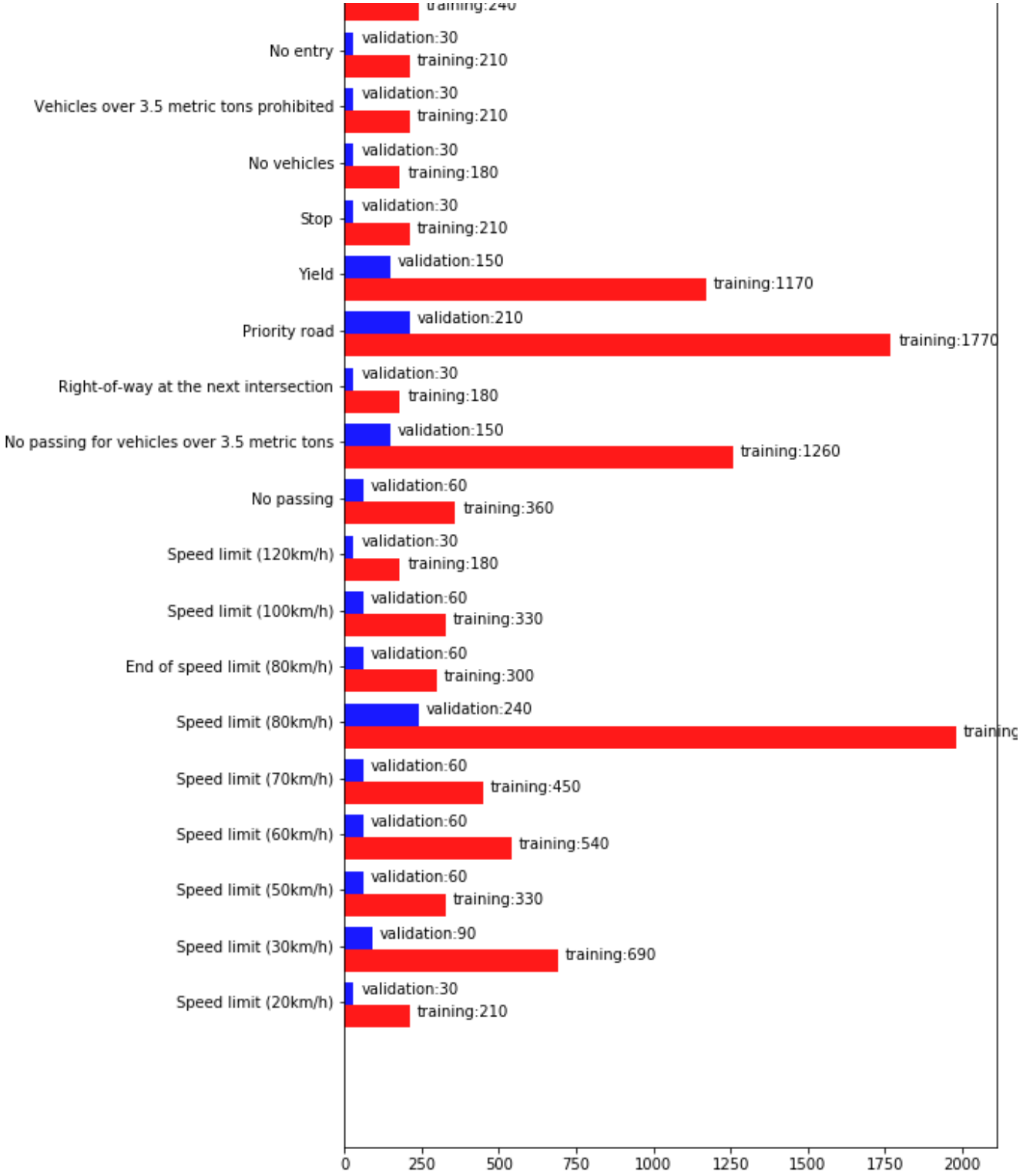
I used the pandas library to calculate summary statistics of the traffic signs data set:

- 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.

Here is an exploratory visualization of the data set. It is a bar chart showing how the data is divided as training and validation images. It shows the number of samples of each class on the traning set and the validation set.


Training Labels
Validation labels
End of no passing by vehicles over 3.5 metric tons
validation:150
training:1350
End of no passing
validation:240
trainin
Roundabout mandatory
validation:120
training:990
Keep left
validation:90
training:540
Keep right
validation:90
training:690
Go straight or left
validation:210
training:18
Go straight or right
validation:60
training:480
Ahead only
validation:90
training:599
Turn left ahead
validation:60
training:300
Turn right ahead
validation:60
training:270
End of all speed and passing limits
validation:30
training:270
Wild animals crossing
validation:60
training:390
Beware of ice/snow
validation:150
training:1290
Bicycles crossing
validation:240
training:1
Children crossing
validation:60
training:360
Pedestrians
validation:120
training:1080
Traffic signals
validation:60
training:360
Road work
validation:120
training:1080
Road narrows on the right
validation:210
training:1800
Slippery road
validation:150
training:1260
Bumpy road
validation:210
training:186
Double curve
validation:210
training:1650
Dangerous curve to the right
validation:150
training:1320
Dangerous curve to the left
validation:30
training:240
General caution
validation:30



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.)

Note: I tried to augment more images by rotating,scaling and transforming them. But it was messing up my model. That's why I haven't used the additional generated data for training my model. But I just included that pipeline in my submission.

First, I shuffled the data using sklearn kit. Then I split the data using train_test_split() from sklearn.model_selection library. I have choosen 20% as test_size.
As a last step I normalized the image data because I want the image data to be in the range from 0 to 1. I got better results after doing this step.

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.

I have used the existing LeNet model that is created in LeNet-Lab.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 1	1x1 stride, VALID padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, VALID padding, outputs 14x14x6
Convolution 2	1x1 stride, VALID padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, VALID padding, outputs 5x5x16
Flatten	Outputs 400
Fully Connected 1	Outputs 120
RELU	
Fully Connected 2	Outputs 84
RELU	
dropout	Keep_prob=0.5
Fully Connected 3	Outputs 43

To train the model, I used an AdamOptimizer with a learning rate of 0.0008. Hyper parameters used in the model are given below

HyperParameter	Value
Epochs	40
Batch Size	100
Mean(mu)	0
Variance(sigma)	0.1
Learning rate	0.0008

HyperParameter	Value
Keep_Prob	0.5

My final model results were:

- validation set accuracy of 99.0%
- test set accuracy of 94.0%

- I have chosen already existing LeNet model from LeNet lab but added a dropout layer in between.
- After so many trails with different hyperparameters I got 99% with the above given hyperparameters.
- My architecture is able to recognize all the test images(Downloaded from WiKi).

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 six German traffic signs that I found on the web:

The last image might be difficult to classify because it is not straight and tilted with some angle and in a different color space.

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
Speed limit (60km/h)	Speed limit (60km/h)
Slippery Road	Slippery Road

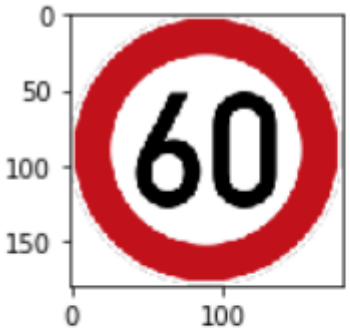
Image	Prediction
No entry	No entry
Children crossing	Children crossing
Traffic signals	Traffic signals
Stop	Stop
Road work	Dangerous curve to the right

The model was able to correctly guess 6 of the 7 traffic signs, which gives an accuracy of 85.7%.

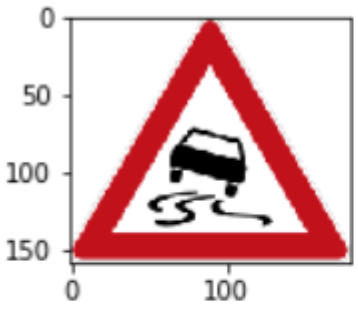
3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" 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 17th cell of the Ipython notebook.

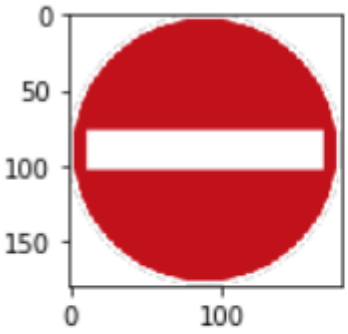
For Each image The top five soft max probabilities were given below



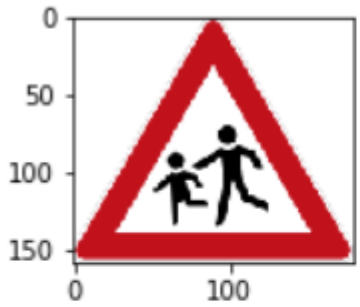

```
Speed limit (60km/h): 99.982%
Keep right: 0.018%
Road work: 0.000%
Speed limit (50km/h): 0.000%
Yield: 0.000%
```



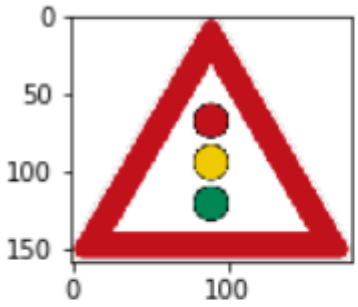

```
Slippery road: 100.000%
Dangerous curve to the right: 0.000%
Dangerous curve to the left: 0.000%
Speed limit (60km/h): 0.000%
Beware of ice/snow: 0.000%
```



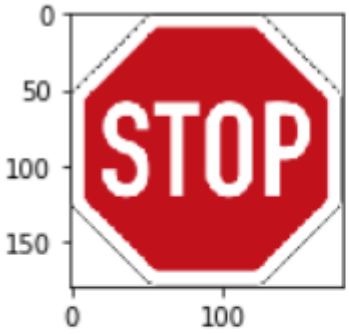

```
No entry: 100.000%
Traffic signals: 0.000%
Speed limit (20km/h): 0.000%
Stop: 0.000%
General caution: 0.000%
```




```
Children crossing: 100.000%
Dangerous curve to the right: 0.000%
General caution: 0.000%
Slippery road: 0.000%
Right-of-way at the next intersection: 0.000%
```




```
Traffic signals: 100.000%
General caution: 0.000%
Priority road: 0.000%
No vehicles: 0.000%
No entry: 0.000%
```




```
Stop: 100.000%
Speed limit (80km/h): 0.000%
Speed limit (120km/h): 0.000%
Speed limit (60km/h): 0.000%
No vehicles: 0.000%
```


```
Dangerous curve to the right: 99.907%
Slippery road: 0.078%
Keep right: 0.014%
Beware of ice/snow: 0.001%
Turn left ahead: 0.000%
```