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

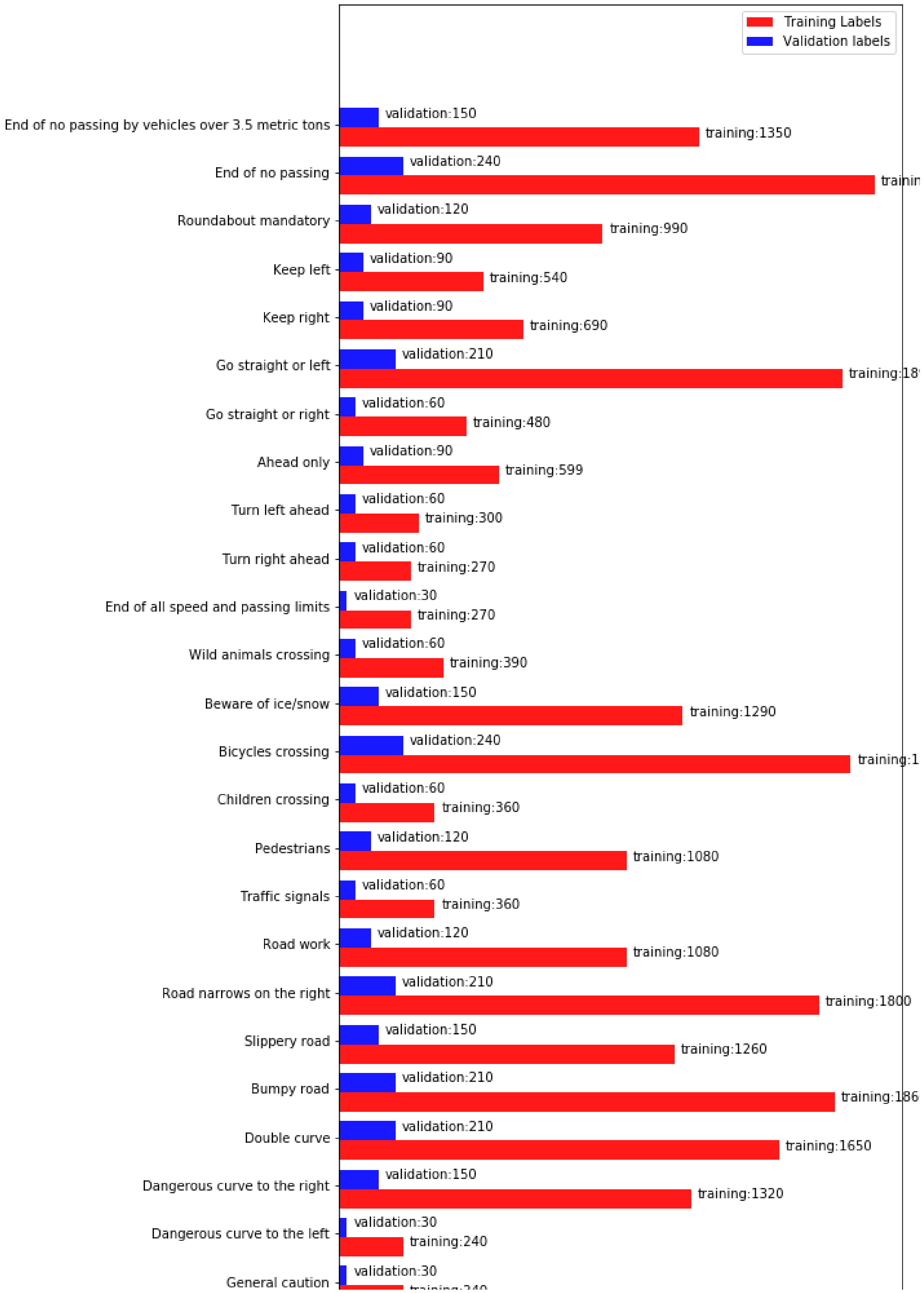
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

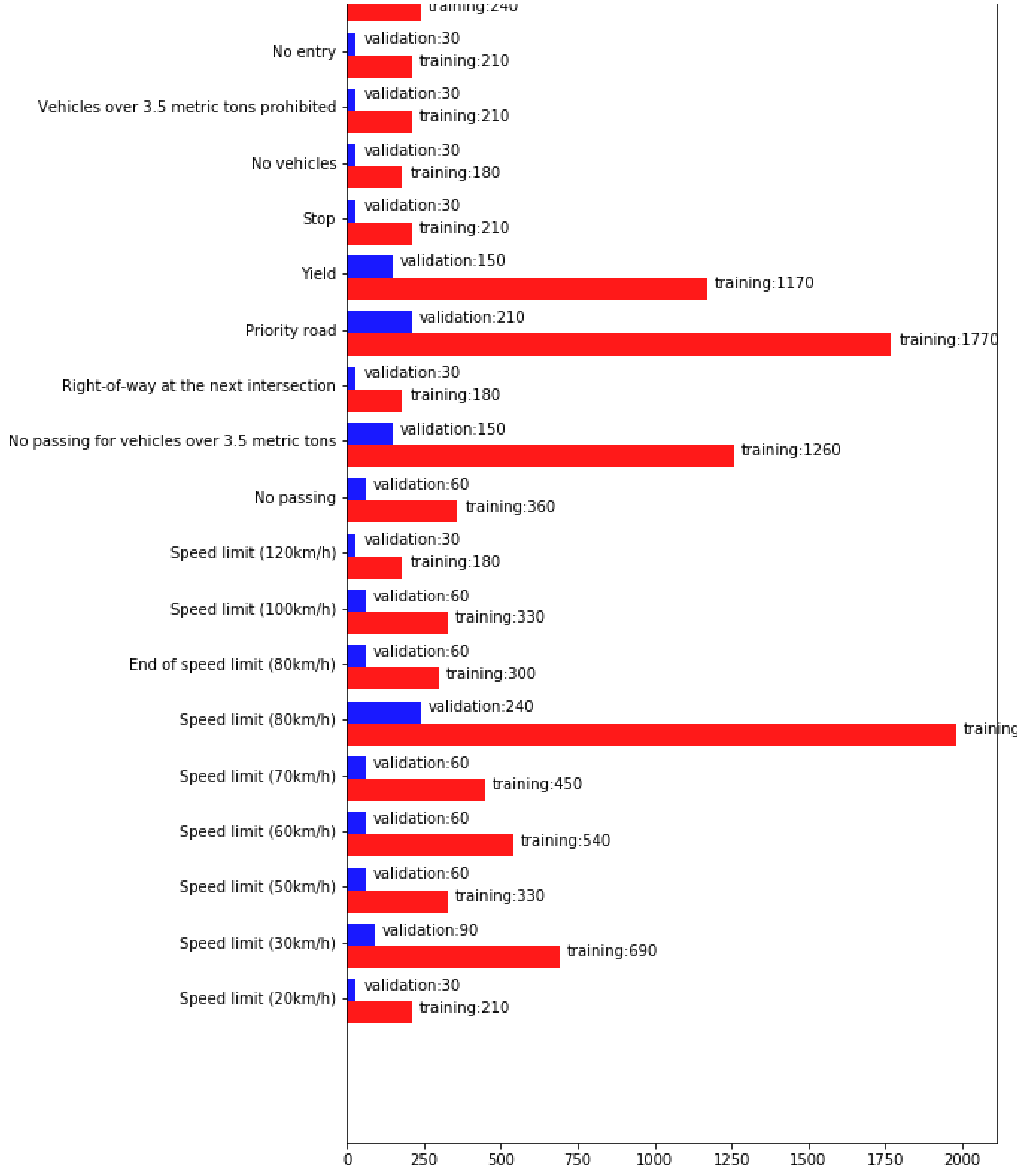
I used the pandas library to calculate summary statistics of the traffic signs data set:

- 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
training:240



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.)

I split the data using train_test_split() from sklearn.model_selection library. I have choosen 20% as test_size. Then,I normalized the image data because I want the image data to be in the range from 0 to 1. I have used ImageDataGenerator from Keras.preprocessing.image library to preprocess the data and augment it in real time. My ImageDataGenerator() argument values are given below

Argument	Value
featurewise_center	False
featurewise_std_normalization	False
shear_range	0.1
rotation_range	10.0
width_shift_range	0.1
height_shift_range	0.1
zoom_range	0.2

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 Keras.Sequential to design my model architecture. Sequential is a keras container fot linear stack of layers. Layers will do automatic shape inference expect the first layer, In the first layer I have provided the input shape as (32,32,3).

I have designed a model with 6 convolutional layers and a fully connected layer. I have added one dropout layer to prevent overfitting. My output contains a softmax activation function to return the probabilties.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 1	3x3 kernel, SAME padding, 32 Filters,relu activation
Convolution 2	3x3 kernel, SAME padding, 32 Filters,relu activation
Max pooling	2x2 stride, VALID padding
Convolution 3	3x3 kernel, SAME padding, 64 Filters,relu activation
Convolution 4	3x3 kernel, SAME padding, 64 Filters,relu activation
Max pooling	2x2 stride, VALID padding
Convolution 5	3x3 kernel, SAME padding, 128 Filters,relu activation
Convolution 6	3x3 kernel, SAME padding, 128 Filters,relu activation
Max pooling	2x2 stride, VALID padding
Flatten	
Dense 1	Outputs 512, relu activation

Layer	Description
dropout	Keep_prob=0.5
Dense 2	Outputs 43, softmax activation

To train the model, I have used 'categorical_crossentropy' as my loss fucntion, 'stochastic gradient descent(SGD)' as optimizer and metric is 'accuracy' and compiled the model using keras model.compile().

While Training the model I have generated more data on the fly by keras preprocessing techniques. I have used LearningRateScheduler from keras.callbacks library to select decaying learning rate over the epochs. I have used below function to decay the learning over the epochs. def learning_rate(epoch): return 0.001*(0.1**int(epoch/10))

I have trained the model using below function in keras

model.fit_generator(datagen.flow(X_train,y_train,batch_size=batch_size),steps_per_epoch=1500,epochs=epochs,(X_validation,y_validation) , callbacks=[LearningRateScheduler(lr_schedule), ModelCheckpoint('model.h5',save_best_only=True)])

Hyper parameters used in the model are given below

HyperParameter	Value
Epochs	10
Batch Size	32
Learning rate	0.001
Keep_Prob	0.5

First, my parameters are like- Constant learning rate=0.001, epochs=30,batch_size=32, steps_per_epoch=X_train.shape[0]//batch_size, and I had 3 dropout layers in my CNN. With this setup I have got a validation accuracy of 90.65%.

Later, I used decaying learning rate, epochs=10, batch_size=64, and removed two dropout layers, My validation accuarcy was 94.87%

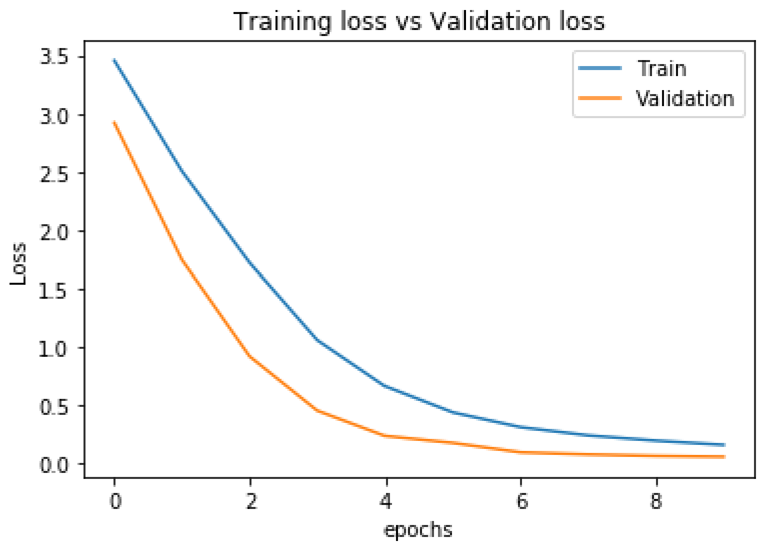
When I set the steps_per_epoch=X_train.shape[0] and It took a lot of time to train the model.

After trying different values for parameters, finally I have used decaying learning rate, epochs=10, Batch_size=32, steps_per_epoch=1500, I have got a validation accuracy of 98.38%

My final model results were:

- training set accuracy of 95.00%
- validation set accuracy of 98.38%

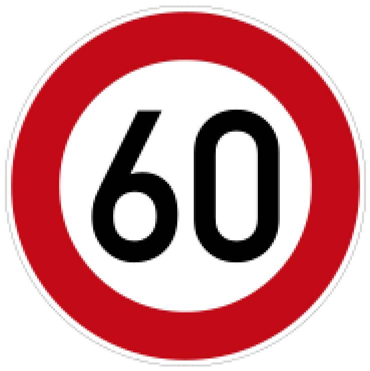
Below I have provided the Learning curves

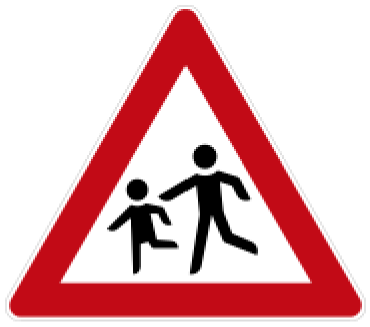


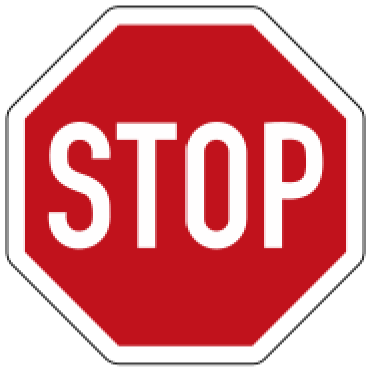
Test a Model on New Images

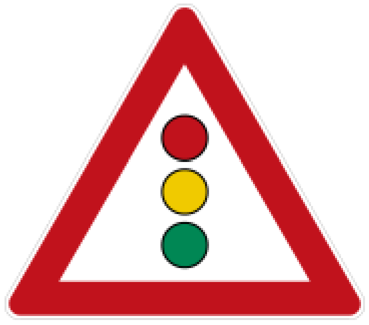
1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

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









The first four images are colored opaque images so I converted them to 3 channel(RGB) images. The last two images might be difficult to classify because the STOP sign is tilted at angle when it is resized to (32,32) it has become more noisy, And the Road work sign in not clear and also tilted.

Since I have added more image data by preprocessing the images while trianing my model is able to identify thes images correctly. Without data augmentation It may be difficult to identify the signs in these two images.

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).

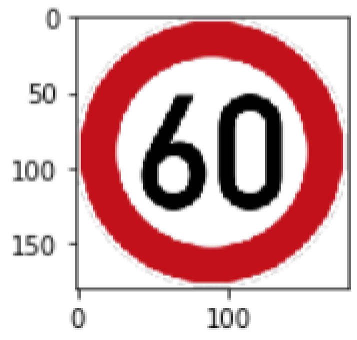
Here are the results of the prediction:

Image	Prediction
Speed limit (60km/h)	Speed limit (60km/h)
Children crossing	Children crossing
Traffic signals	Traffic signals
Stop	Stop
Road work	Roadwork
tilted STOP	STOP

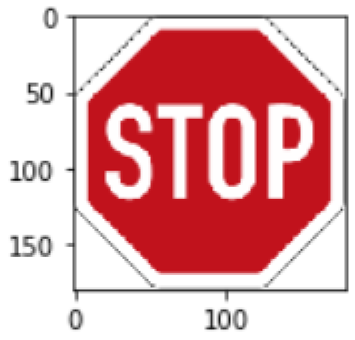
The model was able to correctly guess all the 7 traffic signs, which gives an accuracy of 100%.

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)

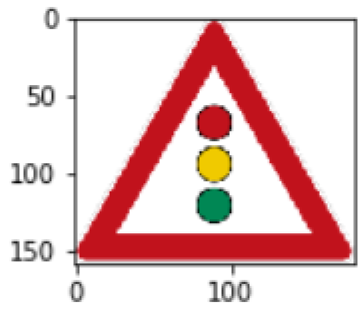
The top five probalities for each test image is given below. Probabilties are 1 for all the test images except for 60 speed limit the accuracy was 99.445%.



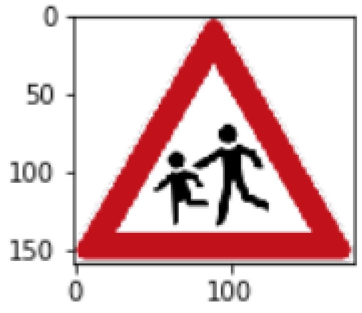
```
Speed limit (60km/h): 99.445%
Speed limit (50km/h): 0.485%
Speed limit (80km/h): 0.046%
Speed limit (120km/h): 0.013%
No vehicles: 0.009%
```



```
Stop: 100.000%
Bicycles crossing: 0.000%
Speed limit (30km/h): 0.000%
No vehicles: 0.000%
Road work: 0.000%
--------------------------------------------------
```

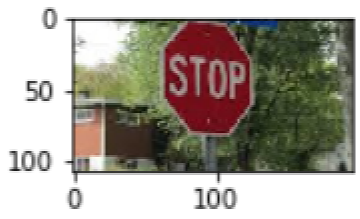


```
Traffic signals: 100.000%
Bicycles crossing: 0.000%
General caution: 0.000%
Road narrows on the right: 0.000%
Bumpy road: 0.000%
------------------------------------------------------
```



```
Children crossing: 100.000%
Bicycles crossing: 0.000%
Beware of ice/snow: 0.000%
Speed limit (60km/h): 0.000%
Right-of-way at the next intersection: 0.000%
```

![alt text][image17]



```
Stop: 100.000%
End of no passing by vehicles over 3.5 metric tons: 0.000%
No passing for vehicles over 3.5 metric tons: 0.000%
General caution: 0.000%
No entry: 0.000%
------------------------------------------------------------
```

```
Road work: 100.000%
End of no passing by vehicles over 3.5 metric tons: 0.000%
Dangerous curve to the right: 0.000%
General caution: 0.000%
```