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

Data Set Summary & Exploration

1. **Summary of the dataset**

The dataset is presented in such files:

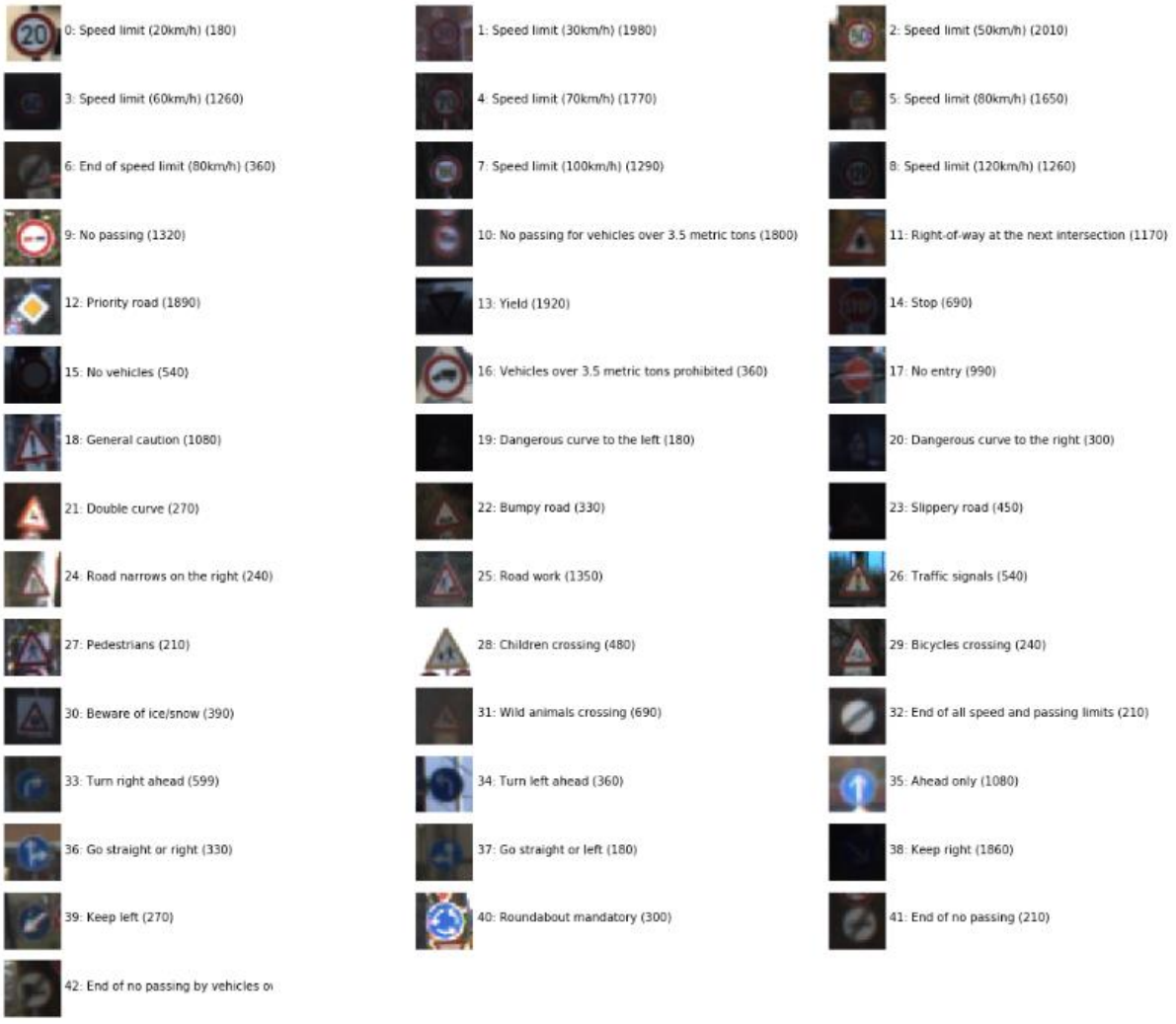
- train.p – input data for training dataset
- valid.p – input data for the validation dataset
- test.p – input data for the test dataset
- signnames.csv – possible output labels

Some statistics from the presented dataset:

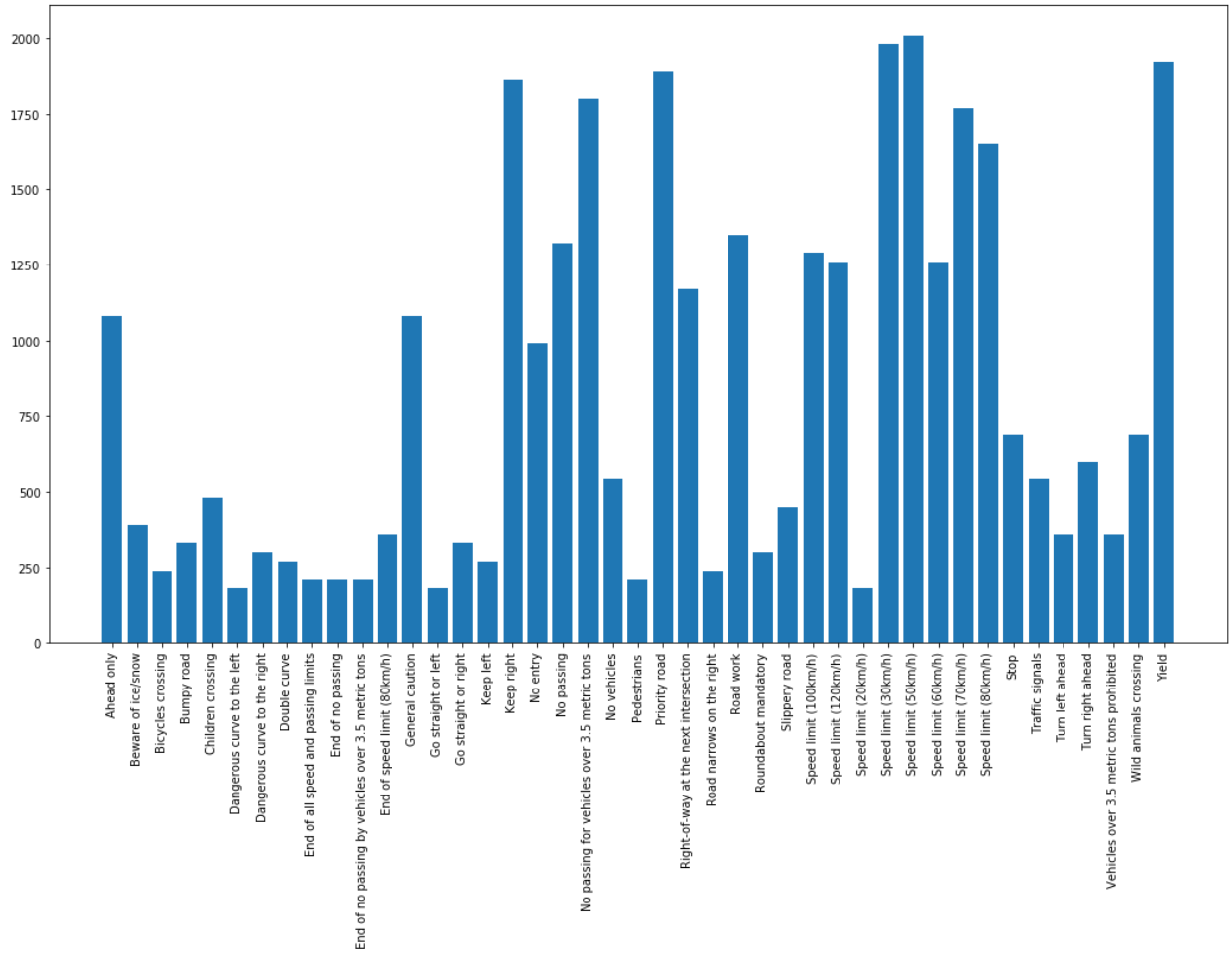
- The size of training set is 34799 examples
- The size of the validation set is 4410 examples
- The size of test set is 12630 examples
- The shape of a traffic sign image is 32x32px
- The number of unique classes/label in the data set is 43

2. Visualization of the dataset.

Labels which are presented in dataset (but not uniformly):



The distribution for each label in training dataset:



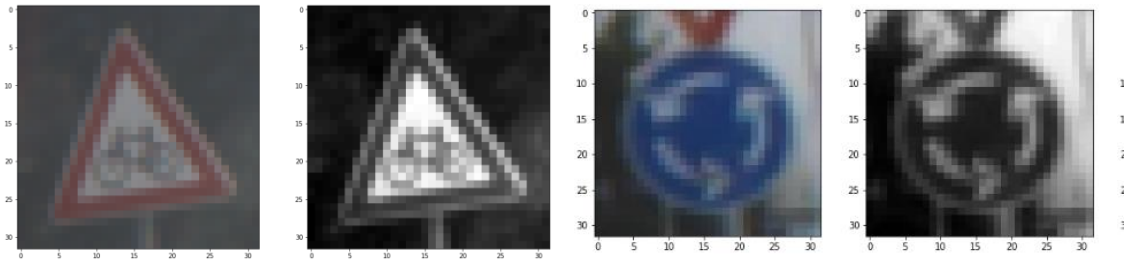
Design and Test a Model Architecture

1. Image preprocessing

At the first step all the images are converted to grayscale. In such case the number of input parameters (channel) is reduced, but in other case the main image contours which for sign is saved.

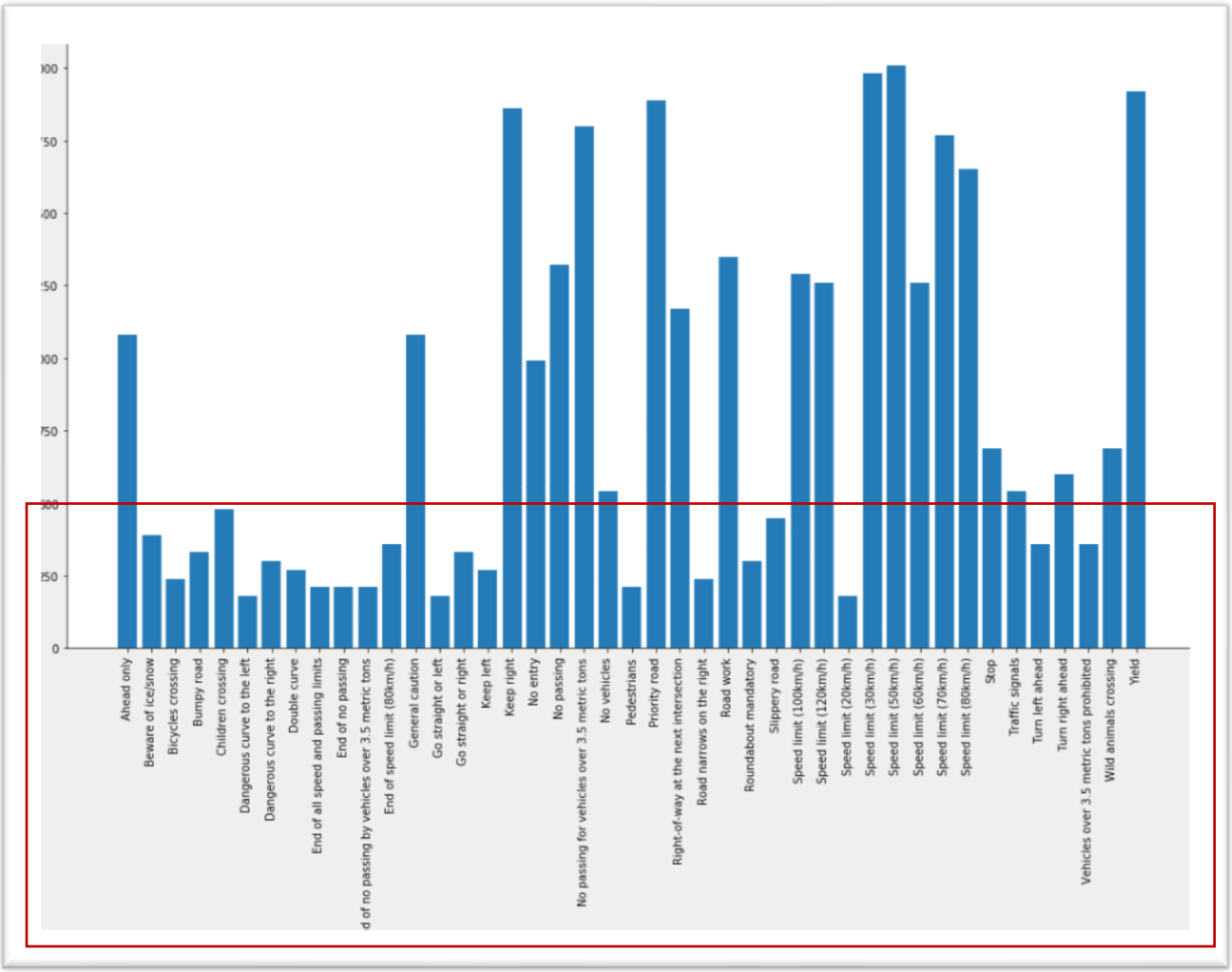
The other mandatory step for all the images is to normalize them

Here is an example of a traffic sign image before and after gray scaling and normalizing:



Normalized images make optimizer job much easier

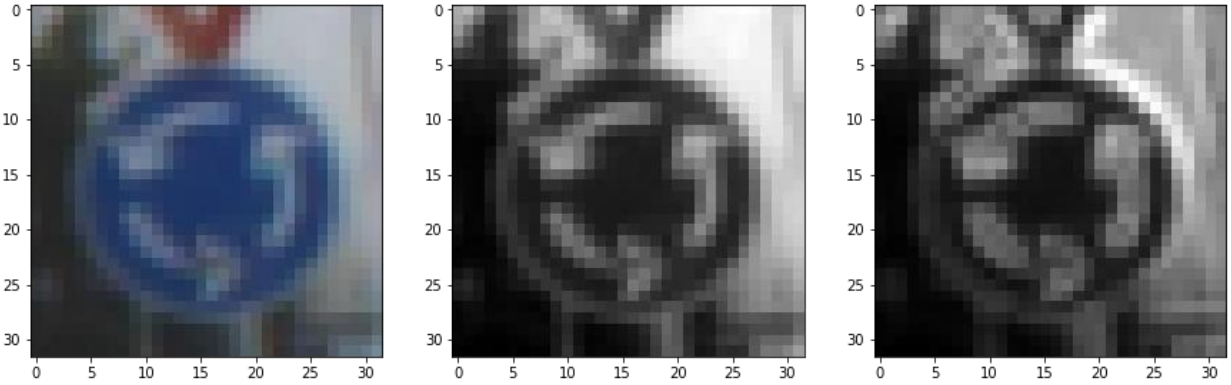
Another approach that was provided is to add fake images. Fake images were added only for images with small frequencies inside the training set. So the fake images were added for the labels with less than 500 images for it



The way data is generated in such way:

- Convert image to grayscaled
- Increase contrast for pixels where derivatives are changed. Sobel mechanism is used for this approach.

Example of fake image:



Original image, gray-scaled image, fake image

2. Model architecture.

My final model consisted mostly on initial model, but as the overfitting was discovered, so dropout to the third layer was added. Also channel sizes were changed a little bit

So the model contains the following layers:

Layer	Description
Input	32x32x1 GrayScale normalized image
Convolution 5x5	1x1 stride, same padding, outputs 28x28x8
RELU	Rectified linear method
Max pooling	2x2 stride, outputs 14x14x15
Convolution 5x5	1x1 stride, same padding, outputs 10x10x16
Fully connected	Input 300, Output 120
Dropout	With keep probability 80% (for trains)
Fully connected	Input 120, Output 84
Fully connected	Input 84, Output 43 (num of classes)

3. Model Training

To train the model, the proposed AdamOptimizer was used.

Stochastic model was used with the

BATCH_SIZE = 256

To increase the model accuracy, epoch mechanism was used, the number of epochs

EPOCHS = 15

Such learning rate was used:

rate = 0.005

I've tried to use rate = 0.01, the model trains faster, but overfitting is presented after ~ 6-7 epoch. With rate = 0.001 system trains too slow

4. Final model results

Final model results are:

1. training set accuracy is 0.990
2. validation set accuracy is 0.955
3. test set accuracy is 0.914

Test a Model on New Images

1. Traffic signs from the web

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



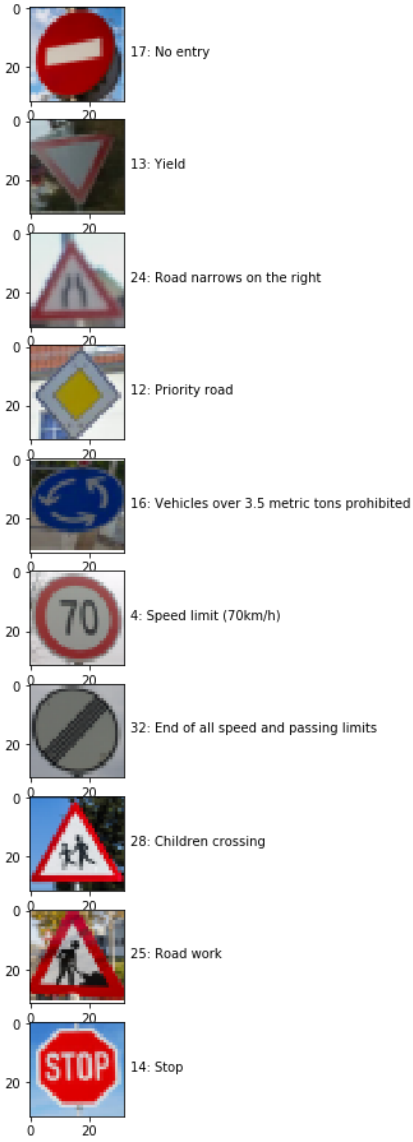






2. Results of the new traffic signs

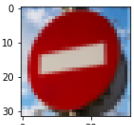
Here are the results of the prediction:



The model was able to correctly guess 9 of the 10 traffic signs, which gives an accuracy of 90%.

3. Top5 softmax probabilities for the each prediction

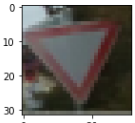
Here are the values of top5 probabilities for each image:



Real index 17: No entry

Top5 prob [[9.99997377e-01 2.53534404e-06 4.82898095e-08 3.32504655e-08
9.67228519e-09]]

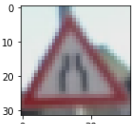
Top5 indices [[17 14 26 33 12]]



Real index 13: Yield

Top5 prob [[0.92005968 0.06373854 0.00773188 0.00303923 0.00196135]]

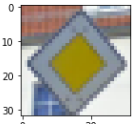
Top5 indices [[13 12 15 28 2]]



Real index 24: Road narrows on the right

Top5 prob [[0.86204737 0.12207108 0.01125188 0.00202885 0.00162516]]

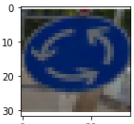
Top5 indices [[24 20 25 30 22]]



Real index 12: Priority road

Top5 prob [[0.86594665 0.07715287 0.02603776 0.00810745 0.00592352]]

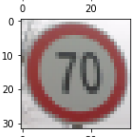
Top5 indices [[12 15 3 35 9]]



Real index 40: Roundabout mandatory

Top5 prob [[0.42445806 0.31118998 0.09625494 0.05509383 0.04070928]]

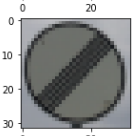
Top5 indices [[16 40 30 23 41]]



Real index 4: Speed limit (70km/h)

Top5 prob [[0.56124473 0.3337608 0.08823773 0.01543458 0.00099726]]

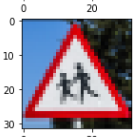
Top5 indices [[4 32 1 0 40]]



Real index 32: End of all speed and passing limits

Top5 prob [[9.99703467e-01 1.58465264e-04 1.34094924e-04 2.12376767e-06
6.92600395e-07]]

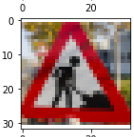
Top5 indices [[32 6 41 3 11]]



Real index 28: Children crossing

Top5 prob [[9.76905107e-01 1.20728919e-02 1.06443977e-02 1.49790445e-04
6.67664353e-05]]

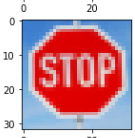
Top5 indices [[28 11 20 41 18]]



Real index 25: Road work

Top5 prob [[8.92981648e-01 6.21237196e-02 4.39382531e-02 6.56580320e-04
2.91180884e-04]]

Top5 indices [[25 22 36 20 26]]



Real index 14: Stop

Top5 prob [[1.00000000e+00 1.29737965e-09 5.77008927e-11 8.72566689e-12
7.50186458e-12]]

Top5 indices [[14 33 17 26 34]]

The probabilities visualization