REPORT: PROJECT 2. TRAFFIC SIGN CLASSIFIER

Author: Markus Buchholz

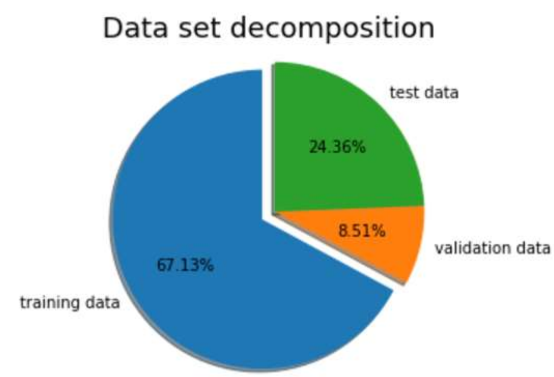
1. Introduction

This report discusses the solution for German traffic sign recognition. Recognition was performed by Convolutional Neural Network classifier. Prior to network development, the data exploratory work was performed. Learned network was verified both based on provided test data and 10 pictures founded on the web.

2. Exploration of data

Exploration of data set was done by use of Panda, Numpy and Matplotlib. Research data set can be represented in statistical form as follows:

Name	Value
Number of training examples	34799
Number of validation examples	4410
Number of test examples	12630
Image data shape	32,32,3
Number of classes	43



Data set is distributed among 43 classes and can be represented on below histograms:

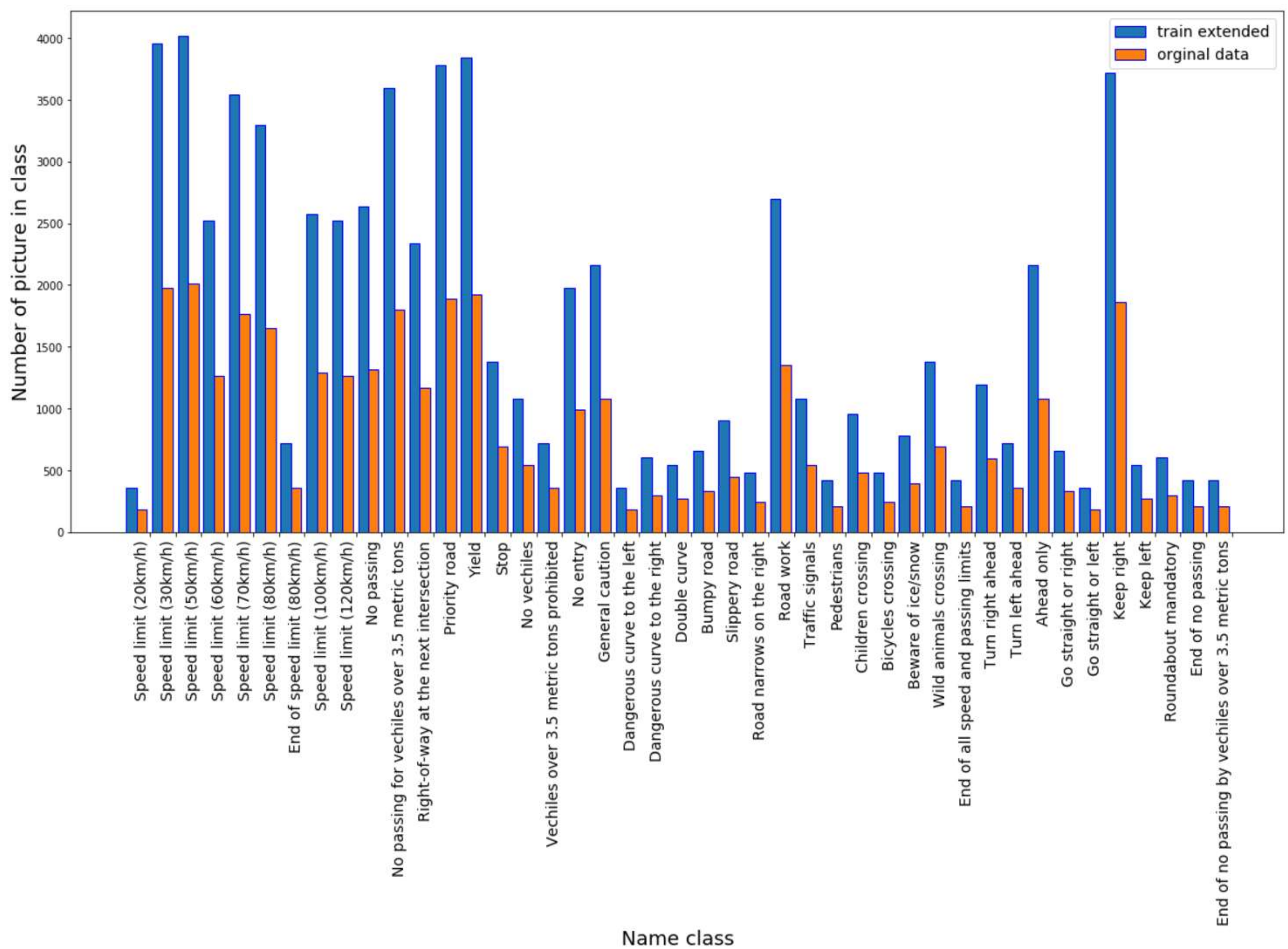
Class	Nr pictures	Sign
0	180	Speed limit (20km/h)
1	1980	Speed limit (30km/h)
2	2010	Speed limit (50km/h)
3	1260	Speed limit (60km/h)
4	1770	Speed limit (70km/h)
5	1650	Speed limit (80km/h)
6	360	End of speed limit (80km/h)
7	1290	Speed limit (100km/h)
8	1260	Speed limit (120km/h)
9	1320	No passing
10	1800	No passing for vechiles over 3.5 metric tons
11	1170	Right-of-way at the next intersection
12	1890	Priority road
13	1920	Yield
14	690	Stop
15	540	No vechiles
16	360	Vechiles over 3.5 metric tons prohibited
17	990	No entry
18	1080	General caution
19	180	Dangerous curve to the left
20	300	Dangerous curve to the right
21	270	Double curve
22	330	Bumpy road
23	450	Slippery road
24	240	Road narrows on the right
25	1350	Road work
26	540	Traffic signals
27	210	Pedestrians
28	480	Children crossing
29	240	Bicycles crossing
30	390	Beware of ice/snow
31	690	Wild animals crossing
32	210	End of all speed and passing limits
33	599	Turn right ahead
34	360	Turn left ahead
35	1080	Ahead only
36	330	Go straight or right
37	180	Go straight or left
38	1860	Keep right
39	270	Keep left
40	300	Roundabout mandatory
41	210	End of no passing
42	210	End of no passing by vechiles over 3.5 metric tons

3. Design and Test a Model Architecture

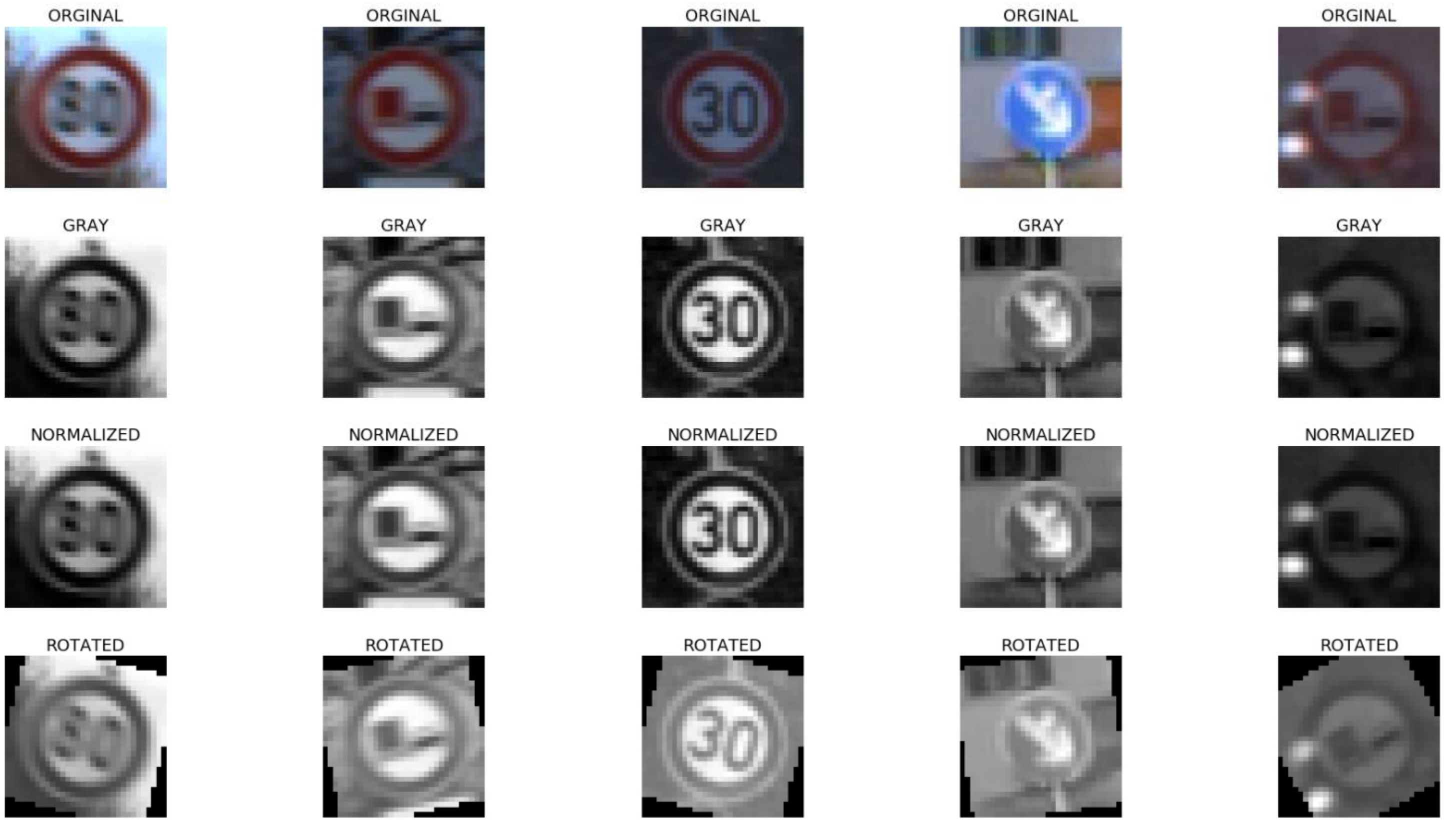
Tree channel data set was converted into one channel first (=> **gray** /by OpenCV). Following the recommended paper (Sermanent and LeCunn) conversion increases the accuracy of picture recognition. It can be understandable since traffic sign can be recognized by the shape and printed picture only. In this case, additional data like a color affects increase of data computation power. Secondly, to achieve consistency in dynamic range for a set of data, the picture **normalization** was performed.

Designed recognition systems was improved by iteration process. It was noticed that data set (here added as a modified training data set) plays significant role in recognition accuracy improvement. Number of picture spread among classes are not equal. It varies considerably (keep right vs. speed limit 20km/h). Therefore, it was reasonable to feed neural network with more data. In this project, it was decided to double the training data (for each class). Fake data (augmented) was created by **rotating** randomly (±30º) the provided class picture.

Final data set used for learning neural network can be presented as follows:



Described above process can be exemplified by following randomly taken pictures:



4. Final Model Architecture

Final model of classifier was figured out in the way of iteration approach. Values of hyperparameters were verified and adjusted. The implemented model was based on LenNet-5 architecture, which was presented during lectures. Laboratory investigations confirmed that this architecture was able to detect correctly the MNIST data. Base on that, it was assumed that for this project reason (recognition also not sophisticated pictures: traffic signs) LenNet-5 architecture would do similar good job. To achieve sufficient recognition accuracy, the core architecture was modified. Deployed architecture can be described as follows:

Layer	**Convolutional_1**	**Convolutional_2**	**Fully_Connected_1**	**Fully_Connected_2**	**Fully_Connected_3**
Input	32x32x1	14x14x32	1152	240	84
Filter	5x5x32	3x3x32	-	-	-
Output	28x28x32	12x12x32	240	84	43
Pooling (stride)	1,2x2,1	1,2x2,1	-	-	-
Output	14x14x32	6x6x32	-	-	-
Flatten	-	1152	-	-	-
Description	RELU applied MAX pooling $Conv1=W_1 \times X + b_1$	RELU applied MAX pooling $Conv2=W_2 \times Conv1 + b_2$	RELU applied DropOut = 0.7 $F1=W_3 \times Conv2 + b_3$	RELU applied DropOut = 0.7 $F2=W_4 \times F1 + b_4$	RELU applied $Logits=W_5 \times F2 + b_5$

5. Training and model approach

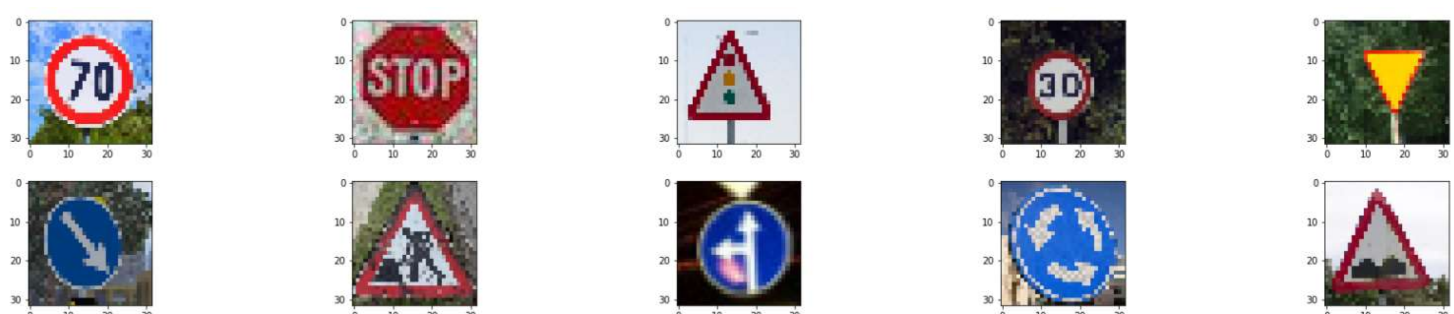
Bearing in mind project requirements (accuracy at least 0.93) several techniques to improve recognition quality was applied. At the beginning the standard Lab LenNet-5 was applied. Insufficient test result (on test dat) set accuracy (0.922) forced to first reorganize the network architecture. It was decided to increase the depth of the network by applying 5x5x32 filter. The dimension of the picture is rather small, therefore the more details should be recognized. Further, performing the max pooling (the averaging pooling was also checked but the result was not sufficient) at the size 2x2 received input for the next convolution. Here, in order to match correctly the size of the network the filter was reduced to 3x3. After the next max pooling 2x2 it was received the output with the dimension: 6x6x32. Finally, applying flatten, two times drop out and filters, it was reduced output to number of 43 classes.
Training was conducted by use of GPU with 4GB. The choice of learning rate of 0.001 was sufficient to achieve project requirement. At the beginning it was experimented with higher rate = 0.005 and 10 epochs but investigating more test scenarios it was noticed that the choice of 100 epochs and rate 0.001 were good enough to build sufficient classifier. It is worth mentioning that lowering learning rate to 0.0001 (0.0005 was also checked) did not improve the final accuracy. During training it was observed considerable rate fluctuations around 0.97 so the idea of reducing the rate below 0.001 was omitted.
Applying two times 0.7 drop out increased considerably the classifier accuracy. Value of the dropout was chosen by iteration approach. During the iteration process, it was checked value between 0.65 and 0.8 but only 0.7 sufficiently increased accuracy level. Model training is performed by use of Adam optimizer.

Accuracy for the final model setup:

Name	Accuracy
Training accuracy (after 100 epoch)	1
Validation accuracy (after 100 epoch)	0.971
Test accuracy	0.958

6. New images

It was decided to verify the quality of designed classifier by applying 10 new images which were found on the web. Following pictures can be seen below. The quality of original size of the pictures is well enough. Decreasing the size to 32x32 pixels makes that some pictures (mainly: 30km/hour speed limit sign and Road Work) can be misinterpreted (by human being). Finally, as expected trained machine influenced similar problem and classified the sign wrongly (Road work).



7. Predictions

Choice of 10 pictures gave better overview about the quality of designed classifier. Calculated final accuracy (on test set) was confronted with completely new pictures. Final prediction is included in below table. It seems that classifier predicts quiet well the rounded signs with white arrows. The most difficult for the classifier are the signs with unique signature (picture) like traffic light sign and road work, beside the large data training set for this class (for road work). As it was mentioned previously, the sign road work was predicted wrongly (human can meet the same problem). In addition, the traffic light was not classified proper also. As it will be discussed later the classifier does not recognize this sign completely.

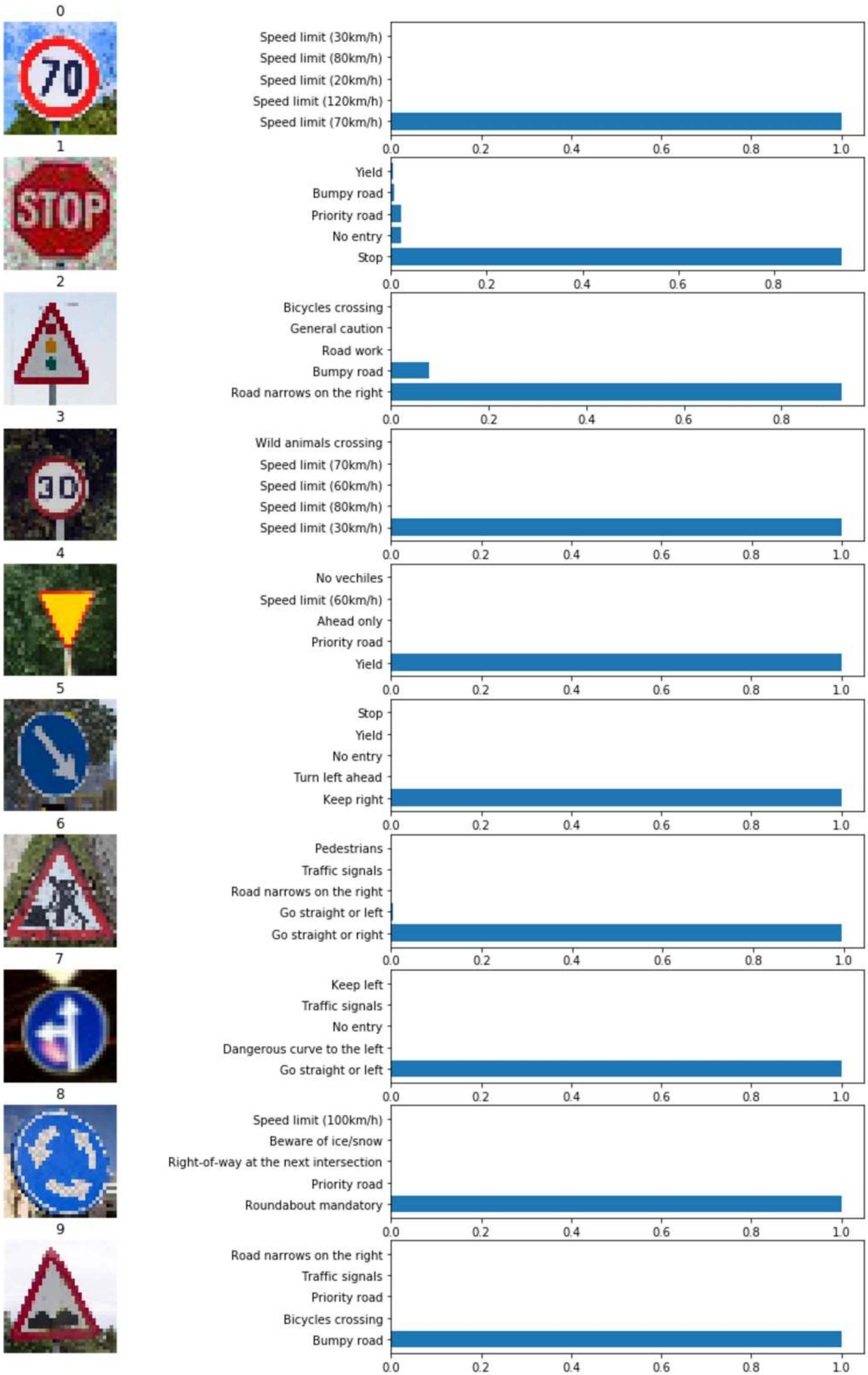
Calculated prediction for new pictures was estimated on 0.8 so it is below the computed accuracy for the test data set. It should be clarified that the accuracy for new pictures is computed on very small sample (10 pictures) while the model is tested on large data (12630 pictures), where well predicted classes contain most pictures.

Nr	Class of original image	Predicted class	Correctly predicted?
1	Speed limit (70km/h)	Speed limit (70km/h)	OK
2	Stop	Stop	OK
3	**Traffic signals**	**Road narrows on the right**	**NO**
4	Speed limit (30km/h)	Speed limit (30km/h)	OK
5	Yield	Yield	OK
6	Keep right	Keep right	OK
7	**Road work**	**Go straight or right**	**NO**
8	Go straight or left	Go straight or left	OK
9	Roundabout mandatory	Roundabout mandatory	OK
10	Bumpy road	Bumpy road	OK
		Prediction	**80%**

8. Softmax probabilities

For each 10 predicted pictures the top 5 softmax probability was calculated and depicted on below bar charts. Not surprisingly, the softmax prediction analysis depicts that round signs with white arrow is classified 100% correctly. There is not other assumption that this sign can belong to the other class then predicted. As it was discussed previously, the classifier meets the problem with Road Work. Softmax does not completely predict this sign (there is not perdition for correct label in top 5 calculated probabilities). The same happens for the traffic lights.

There is not fail assumption for the sig with 70km/hour speed limit. The same remark can be extrapolated for the yield sign and bumpy road. There are some other minor probabilities for the Stop sign but the probability over 0.9 for the correct label does this sign classified correctly.

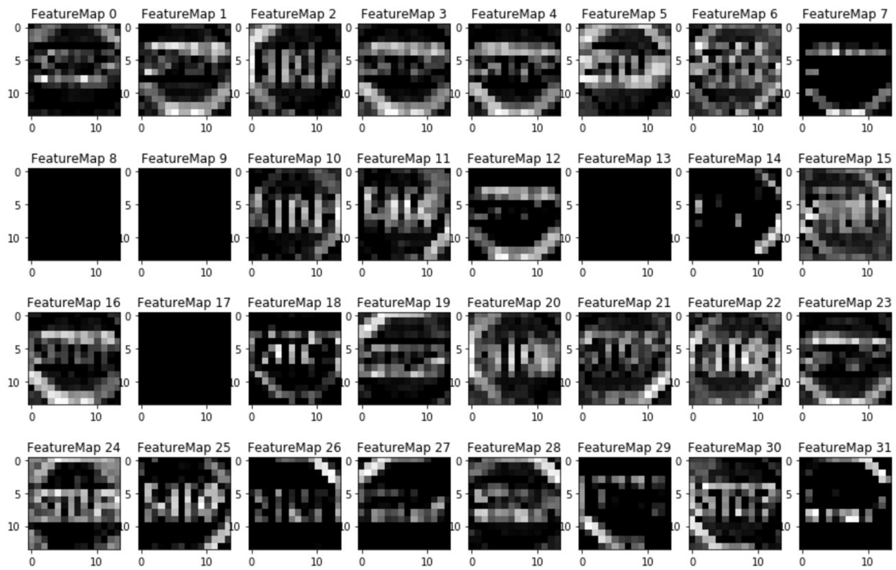


0
Speed limit (30km/h)
Speed limit (80km/h)
Speed limit (20km/h)
Speed limit (120km/h)
Speed limit (70km/h)
0.0 0.2 0.4 0.6 0.8 1.0
1
Yield
Bumpy road
Priority road
No entry
Stop
0.0 0.2 0.4 0.6 0.8
2
Bicycles crossing
General caution
Road work
Bumpy road
Road narrows on the right
0.0 0.2 0.4 0.6 0.8
3
Wild animals crossing
Speed limit (70km/h)
Speed limit (60km/h)
Speed limit (80km/h)
Speed limit (30km/h)
0.0 0.2 0.4 0.6 0.8 1.0
4
No vechiles
Speed limit (60km/h)
Ahead only
Priority road
Yield
0.0 0.2 0.4 0.6 0.8 1.0
5
Stop
Yield
No entry
Turn left ahead
Keep right
0.0 0.2 0.4 0.6 0.8 1.0
6
Pedestrians
Traffic signals
Road narrows on the right
Go straight or left
Go straight or right
0.0 0.2 0.4 0.6 0.8 1.0
7
Keep left
Traffic signals
No entry
Dangerous curve to the left
Go straight or left
0.0 0.2 0.4 0.6 0.8 1.0
8
Speed limit (100km/h)
Beware of ice/snow
Right-of-way at the next intersection
Priority road
Roundabout mandatory
0.0 0.2 0.4 0.6 0.8 1.0
9
Road narrows on the right
Traffic signals
Priority road
Bicycles crossing
Bumpy road
0.0 0.2 0.4 0.6 0.8 1.0

9. Neural Network's State

Making variable: conv_layer_1 and conv_layer_2 global allowed to capture the network stat after first and second convolution. In this case the feature maps was plotted in response to a Stop sign image.

conv_layer_1:



conv_layer_2