Writeup

1. Files Submitted

The following files have been submitted:

1. Traffic_Sign_Classifier.ipynb = jupyter notebook (code)
2. Traffic_Sign_Classifier_org.html = jupyter notebook in html running the original training data-set
3. Traffic_Sign_Classifier_aug.html = jupyter notebook in html running an augmented training data-set
4. Traffic_Sign_writeup.pdf = this file

The files are located at the GitHub: https://github.com/carstenMIELENZ/TrafficSignClassifier.git

2. Data Set Summary & Exploration

2.1. Basic Summary of the Data-set

I used the *sklearn.model_selection library* to split the original training data-set into training and validation data-set by the *function train_test_split*.

X_train, X_validation, y_train, y_validation = train_test_split(X_train,y_train,test_size=0.2,random_state=0)

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

- The size of training set is 31367
- The size of the validation set is 7842
- 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

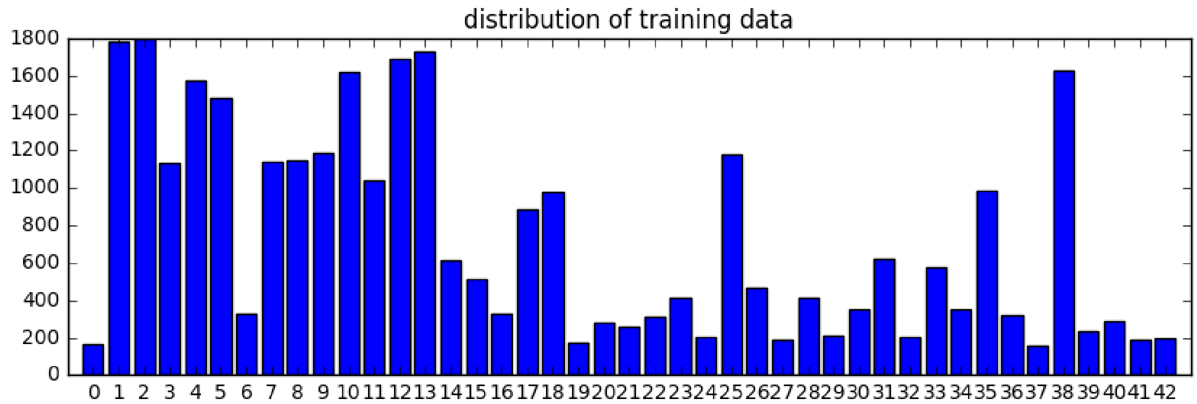
2.2 Exploratory Visualization of the Data-set

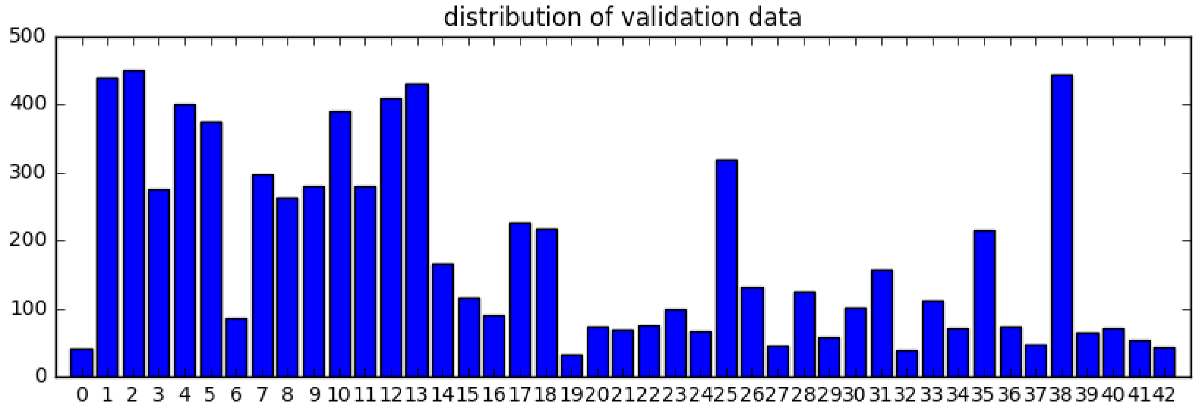
I visualized the data-set into to two ways.

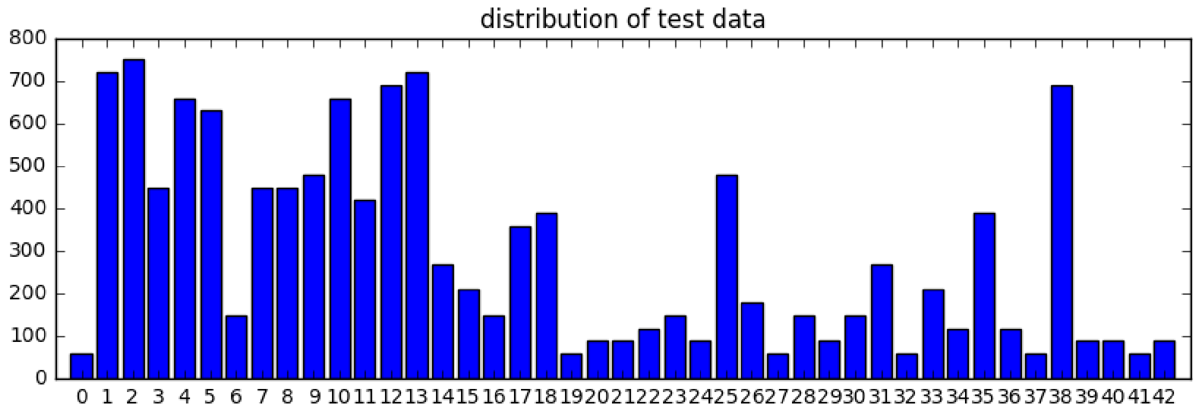
1.Data-set Distribution

I analysed the distribution of the traffic signs for the training, validation and test data-sets, see also bar-chart diagrams below (the x-axis of the diagrams presents the traffic sign type, according the provided *signnames.csv* file, the y-axis the number of types).

It seems that the traffic sign types are differently distributed within the data-sets but the distribution between the training, validation and test data-sets seems similar. My guess is that these distributions represent the traffic sign distribution close to reality.

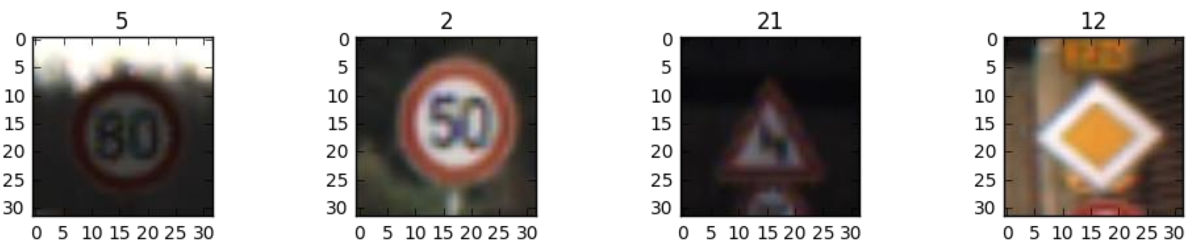






2.Sign Image Visualization

I visualized 4 traffic signs of the training data-set which were randomly selected. The title of each image presents the traffic sign type according the provided *signnames.csv* file, see images below.



3. Design and Test a Model Architecture

3.1 Pre-processing

I pre-processed the data-sets in the following way:

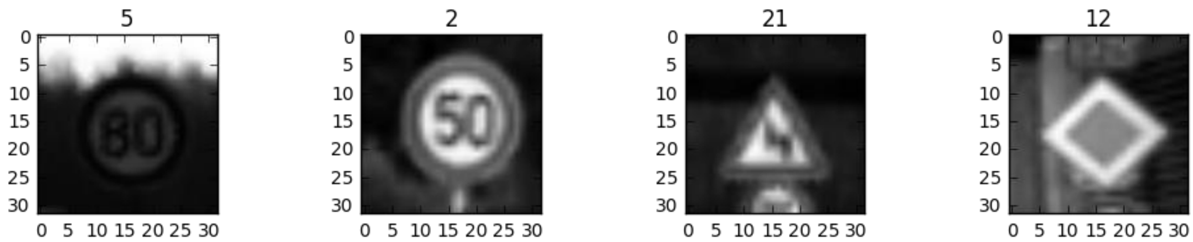
1.Grayscale Conversion

I converted the colour images into grayscale images. **Rationale** for this step is that the signs are easier to recognize in grayscale plus the data complexity of the image is decreased moving from colour to grayscale by removing 2 colour channels.

2.Normalization

I normalized all input images by the pixel modification *(pixel - 128)/ 128* which represents a *[-1.0,+1.0]* normalization. I normalized all output labels using an one-hot normalization. **Rationale** for the normalization step is to reduce data value dependencies.

The figures below present the 4 images pre-processed by the grayscale modification and normalization. The images have the same reference as the images shown in step 2.2.2 (Image Visualization).



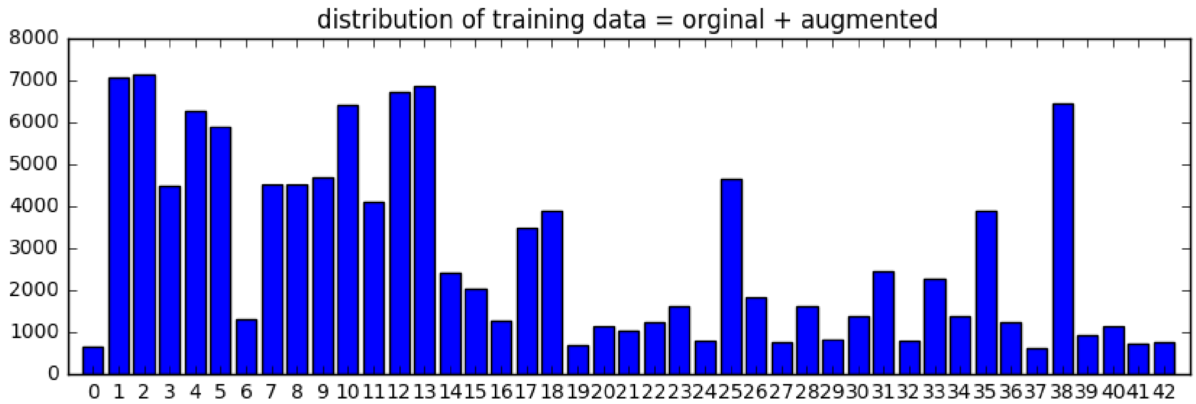
3.2 Augmented Training Data-set

I created an extra training data-set by augmenting the original training data-set. I used for this step the *ImageDataGenerator* of the *keras library*. The augmentation was performed on colour images. After this step the images were pre-processed with grayscale and normalization as described in the previous section. **Rationale** for the augmentation of the training data-set is to make the model more robust by providing more training samples.

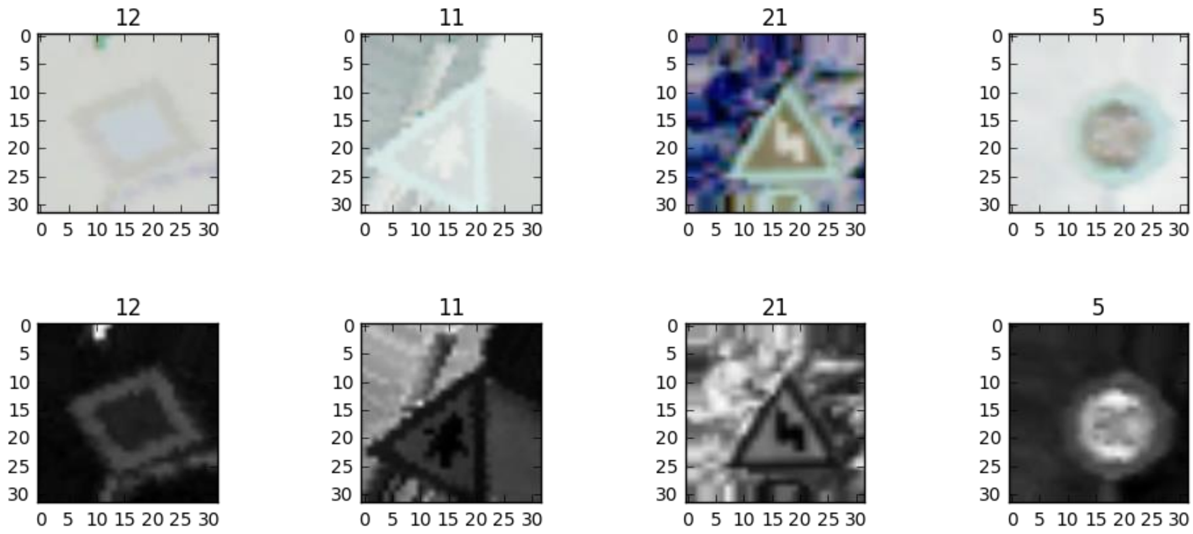
I used the following image modifications for the augmentation of the data-set:

- width & height movements by +/-10%
- rotation +/-15 degrees
- zoom out/in by 90%/110%

I generated extra **93000** augmented images for the training data-set. The training data-set with the augmented images had the size of **124367 images** finally. The figure before shows that the distribution of the augmented training data-set (the x-axis of the diagram presents the traffic sign type, according the provided *signnames.csv* file, the y-axis the number of types).



The figures below show colour images after the augmentation in the first row. The second row shows the augmented colour images after the pre-processing step (grayscaling and normalization), as described in chapter 3.1. Note, the images were randomly selected.



3.3 Final Model Architecture

1.Trade-off Model Architectures

I used the *LeNet* model architecture of the *CarND-LeNet-Lab* project as starting point for the model evaluation. I did trade-off 4 different architectures based on *LeNet*, see the table below. All trade-off model architectures used a 32x32 grayscale and normalized traffic sign image.

Trade-off Model Architectures	Description
LeNet original of CarND-LeNet-Lab	LeNet architecture of CarND-LeNet-Lab project
LeNet with multi-scale & 5x5x6 conv1	LeNet architecture with a multi-scale* input towards fully connected FC0 layer and a 5x5x6 convolution in conv1 layer
LeNet with multi-scale & 3x3x8 conv1	LeNet architecture with a multi-scale* input towards fully connected FC0 layer and a 3x3x8 convolution in conv1 layer
LeNet with multi-scale & 3x3x15 conv1	LeNet architecture with a multi-scale* input towards fully connected FC0 and a 3x3x15 convolution in conv1 layer

* Traffic Sign Recognition with Multi-Scale Convolutional Networks
https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwiUt-Lq1cfTAhWCrRoKHaCWCToQFggsMAE&url=http%3A%2F%2Fyann.lecun.org%2Fexdb%2Fpublis%2Fpsgz%2Fsermanet-ijcnn-11.ps.gz&usg=AFQjCNGTHlNOHKmIxaKYw3_h-VYrsgpCag

The *LeNet* architecture with multi-scale and a 3x3x15 convolutional input performed best versus the other model architecture options using the original training data-set. *i.e. I achieved almost 100% training and 98.58% validation accuracy by this model running 30 epochs*.

The **rationale** for using the multi-scale feature was to provide additional low level information to the fully connected layers in order to better recognize patterns. The **rationale** for using smaller convolutional inputs was to recognize details of images.

2.Final Model Architecture

As described in the previous section the *LeNet* architecture with multi-scale and a 3x3x15 convolutional input was selected as the final model architecture. This model consists of the following layers:

ID	Layer	Description
0	Input	32x32x1 grayscale image
1	Conv1	Input: 0
1.1	Convolution 3x3x15	1x1 stride, valid, outputs 30x30x15
1.2	RELU	
1.3	Max pooling	2x2 ksize, 2x2 stride, outputs 15x15x15
1.4	Max pooling	2x2 ksize, 2x2 stride, outputs 7x7x15
2	Conv2	Input: 1.3
2.1	Convolution 5x5x16	1x1 stride, valid, outputs 11x11x16
2.2	RELU	
2.3	Max pooling	2x2 ksize, 2x2 stride, outputs 5x5x16
3	Multi-scale	Inputs: 1.4, 2.3
3.1	Join & Flatten	flatten (7x7x15, 5x5x16) = 1x1135,
4	Fully Connected FC0	Input: 3.1
4.1	Linear Mul, Add	1x1135 * 1135x120 + 1x120, outputs 1x120
4.2	RELU	
5	Fully Connected FC1	Input: 4.1
5.1	Linear Mul, Add	1x120 * 120x84 + 1x84, outputs 1x84
5.2	RELU	
6	Fully Connected FC2	Input: 5.1
6.1	Linear Mul, Add	1x84 * 84x43 + 1x43, outputs 1x43
6.2	Logits	1x43
7	Softmax	Input: 6.2, outputs softmax of logists
8	Cross Entropy	Inputs: 7, one_hot(y), outputs cross_entropy
9	Loss	Input: 8, outputs Loss
10	Training Optimization	Input: const. learning rate, opt. model vs. Loss using adam algo.

3.Model Training

I have trained the final model using two training data-sets (for comparison)

1. the **original** training data-set
2. an **augmented** training data-set (size = 4 x original), see also chapter 3.2

In both cases I used the same parameters for:

- learning rate of **0.001**
- batch size of **128**
- loss optimizer using ***adam*** algorithm
- initialization: **zeros** for bias, gauss **normal distribution** (mean 0, std. 0.1) for weights

In terms of number of epochs achieving similar training-, validation- and test accuracy, I used

1. for the final model trained with the **original** training data-set **30 epochs**
2. for the final model trained with the **augmented** training data-set **90 epochs**

Note: In the training process using the original and augmented data-set, the final modem did not show any under- or overfitting.

4. Solution Finding

As descripted in chapter 3.3.1 different model architectures were used for trade-off. All architectures fulfilled the target of a validation **accuracy > 93%** using the original training data-set. The final model architecture solution was outperforming over all other model architectures achieving 99.9% training-, 98.7% validation- and 93.2% test accuracy.

As described in the previous section I trained the final model solution by two different training data-sets:

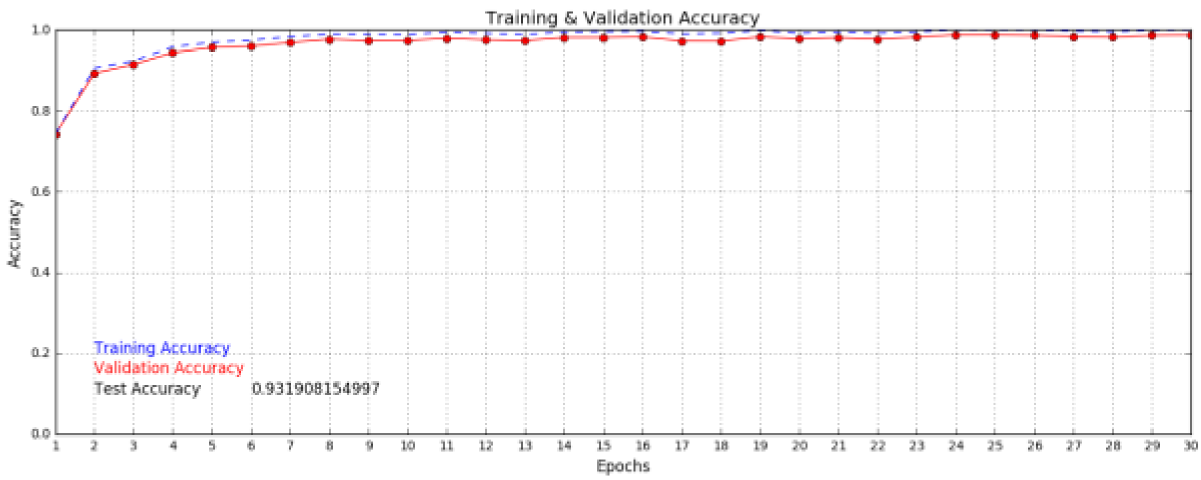
1. the **original** training data-set
2. an **augmented** training data-set in order to get a better model robustness

Both training data-sets show similar training-, validation- and test accuracies on the final model using different number of epoch runs. The **rationale** for the different epoch runs is that the augmented data-set needs more runs on the model for learning because the training data-set is much bigger.

Final model with original training data-set

The model trained with the original training data-set did run **30 epochs** and show the following accuracies:

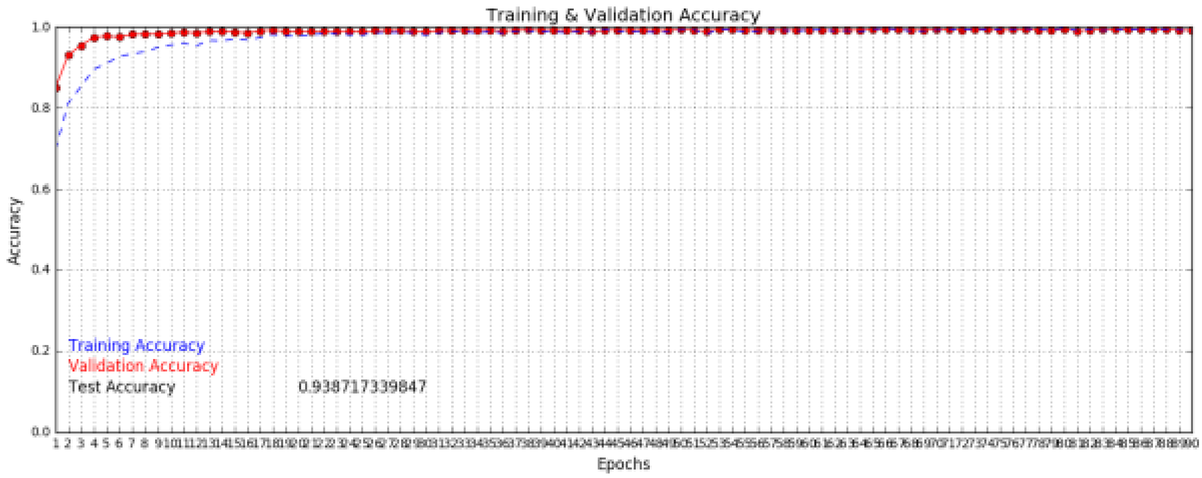
- training accuracy of **99.9%**
- validation accuracy of **98.7%**
- test accuracy of **93.2%**



Final model with augmented training data-set

The model trained with the augmented training data-set did run **90 epochs** and shows the following accuracies:

- training accuracy of **99,1%**
- validation accuracy of **99.2%**
- test accuracy of **93.9%**



Additional Model Architecture Optimizations

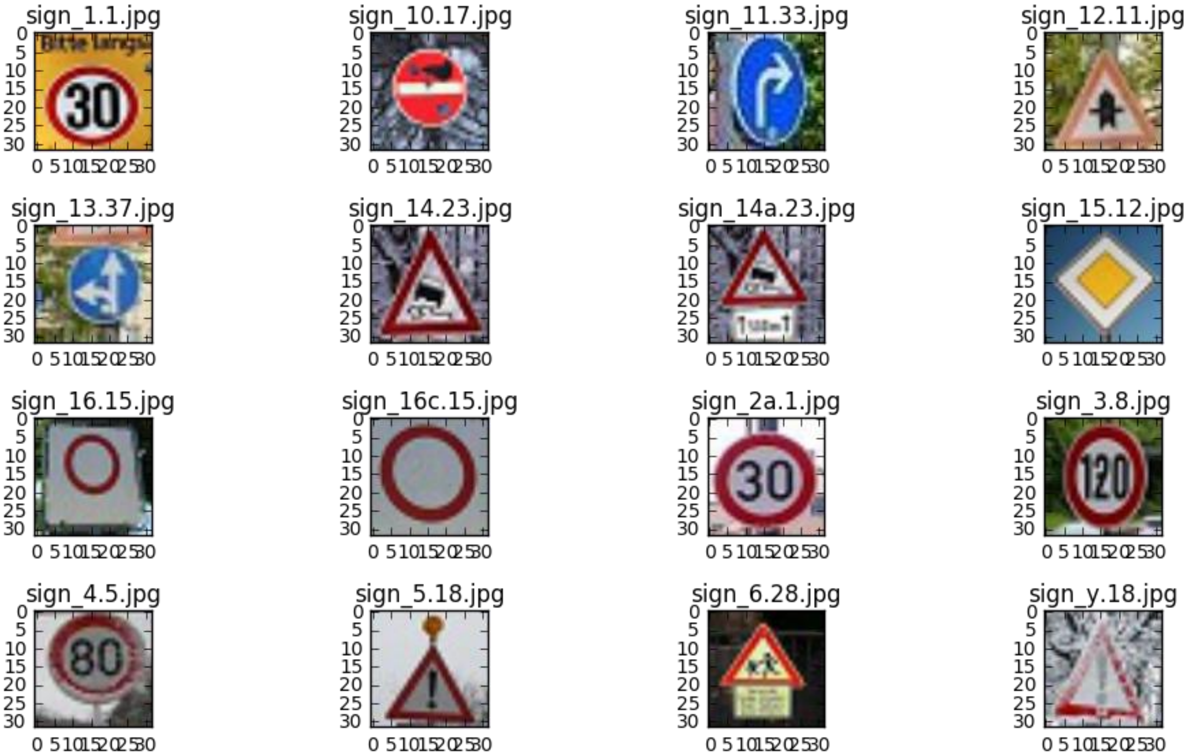
Dropout
I tried to use dropout function for the fully connected layers FC0 and FC1 of the final model. The result was a **drop** of validation accuracy by almost **10%** for both training data-sets. Therefore I have decided not to use this feature. I think the **rationale** for this observation is that the model was not overfitted and therefore the dropout feature did not improve the model performance.

L2 Regularization - not analysed

4.Test a Model on New Images

4.1 German Traffic Signs found on the WWW

I found 16 different German traffic signs on the WWW. All signs were scoped and down scaled to the 32x32 colour format. The down scaling process of the images may stretch and compressed the width and height of the images.

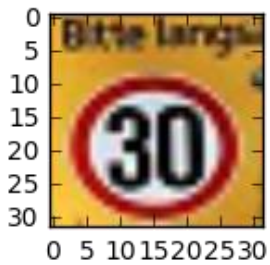


Note: There are two traffic sign pairs with different image sizes: *sign_14.23.jpg* & *sign_14a.23.jpg* and sign_16.15jpg & sign_16c.jpg. The **rationale** for these pairs is to test the size sensitivity of the model.

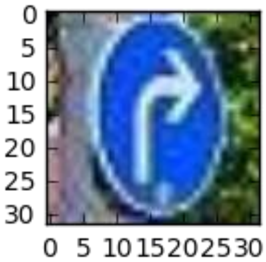
Difficult Traffic Signs

Out of the 16 traffic signs from the WWW I selected 8 traffic signs in order to discuss why these images may be difficult to classify.

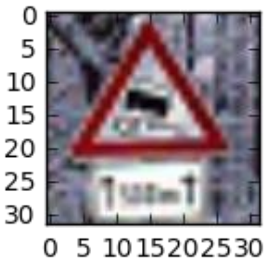
1. The sign **Speed Limit (30km/h)** may be difficult to classify because the sign is integrated into bigger one which complexes the detection process.



2. The sign **Turn Right Ahead** may be difficult to classify because the shape is stretched.

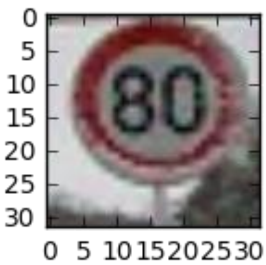


3. The sign **Slippery road** may be difficult to classify because there are two signs which complexes the detection process.

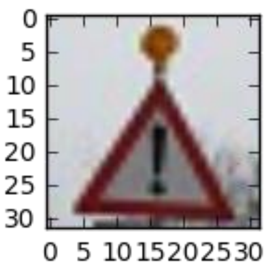


4. The sign **No vehicles** may be difficult to classify because the sign shape is small and inside bigger sign which complexes the detection process. Additionally, there are over-painted characters in this sign.

5. The sign **Speed limit (80 km/h)** may be difficult to classify because the sign is partly covered by snow.

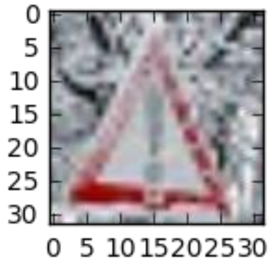


6. The sign **General caution** may be difficult to classify because it is small and has a light on top which complexes the detection process.



7. The sign **Children crossing** may be difficult to classify because the sign is small and connected with another sign which complexes the detection process.

8. The sign **General caution** may be difficult to classify because it is partly covered by snow.



4.2 Model Predictions

The following presents the prediction for the 16 traffic signs from the WWW using

1. the final model trained with the **original** training data-set
2. the final model trained with **augmented** training data-set

1. Predictions with the original training data-set

The final model trained with the original training data-set achieved the following accuracies:

- WWW Traffic Signs Accuracy: **52.6%**
- Test Accuracy: **93.2%**

The figures and the table below summarize which sign was detected (**True**) and which was not (**False**).

Recognized	Traffic	Prediction	Observation
False	Speed limit (30km/h)	Road Work	Inside wrongly interpreted
True	No entry	No entry	
True	Right Ahead	Right Ahead	
True	Right-of-way at the next intersection	Right-of-way at the next intersection	
True	Go straight or left	Go straight or left	
True	Slippery road	Slippery road	
False	Slippery road	Speed limit (20km/h)	Sign wrongly interpreted
True	Priority road	Priority road	
False	No vehicles	End of no passing	Inside wrongly interpreted
True	Speed limit (30km/h)	Speed limit (30km/h)	
True	Speed limit (120km/h)	Speed limit (120km/h)	

False	Speed limit (80km/h)	Slippery road	Sign wrongly interpreted
False	General caution	Right-of-way at the next intersection	Inside wrongly interpreted
False	Children crossing	Turn left ahead	Sign wrongly interpreted
False	General caution	Road narrows on the right	Inside wrongly interpreted

2. Predictions with augmented training data-set

The final model trained with the augmented training data-set achieved the following accuracies:

- WWW Traffic Signs Accuracy: **87.5%**
- Test Accuracy: **93.9%**

The figure and table below summarizes which sign was detected (**True**) and which was not (**False**)

Recognized	Traffic	Prediction	Observation
False	Speed limit (30km/h)	Speed limit (20km/h)	Inside of sign wrongly interpreted
True	No entry	No entry	
True	Right Ahead	Right Ahead	
True	Right-of-way at the next intersection	Right-of-way at the next intersection	
True	Go straight or left	Go straight or left	
True	Slippery road	Slippery road	
True	Slippery road	Slippery road	
True	Priority road	Priority road	
False	No vehicles	No passing	Inside of sign wrongly interpreted
True	Speed limit (30km/h)	Speed limit (30km/h)	
True	Speed limit (120km/h)	Speed limit (120km/h)	
True	Speed limit (80km/h)	Speed limit (80km/h)	
True	General caution	General caution	
True	Children crossing	Children crossing	
True	General caution	General caution	

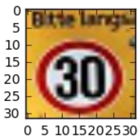
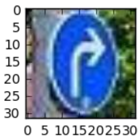
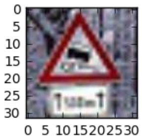
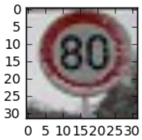
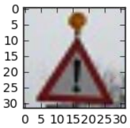
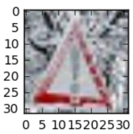
Observations:

1. IF the traffic sign size is small THEN there is higher probability that the sign will not be recognized.
2. Training data-set augmentation makes the more robust and increases the model accuracy.

4.3 Softmax Probabilities

1. Probability of Predictions with original training data-set

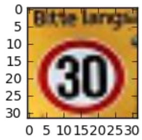
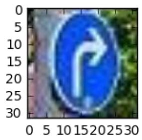
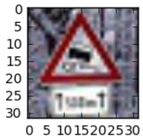
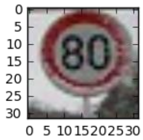
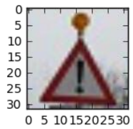
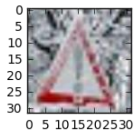
The table below lists the probabilities of the predictions of the model using the original training data-set:

Traffic Sign	Probability of Predictions
	SIGN: Speed limit (30km/h) RECOGNIZED: False top (1) 0.99992144107818603516 Speed limit (70km/h) top (2) 0.00007696625107200816 Speed limit (50km/h) top (3) 0.00000083608955492309 Stop top (4) 0.00000068090128024778 Speed limit (30km/h) top (5) 0.00000000001560197631 Speed limit (20km/h)
	SIGN: Turn right ahead RECOGNIZED: True top (1) 0.99999988079071044922 Turn right ahead top (2) 0.00000014091803279825 Ahead only top (3) 0.00000001456393050603 Road narrows on the right top (4) 0.00000000000232284352 Road work top (5) 0.00000000000195126792 Go straight or left
	SIGN: Slippery road RECOGNIZED: False top (1) 0.99993944168090820312 Speed limit (20km/h) top (2) 0.00005689891622751020 Roundabout mandatory top (3) 0.00000371043438462948 Speed limit (30km/h) top (4) 0.00000000165818181408 Traffic signals top (5) 0.00000000052337578715 Wild animals crossing
	SIGN: No vehicles RECOGNIZED: False top (1) 0.99864798784255981445 No passing top (2) 0.00134783785324543715 Slippery road top (3) 0.00000398034626414301 Yield top (4) 0.00000024362469730477 Speed limit (50km/h) top (5) 0.00000000007727519319 End of no passing
	SIGN: Speed limit (80km/h) RECOGNIZED: False top (1) 0.99982637166976928711 Slippery road top (2) 0.00016014919674489647 Roundabout mandatory top (3) 0.00001339767368335743 Go straight or left top (4) 0.00000014848212970264 Dangerous curve to the right top (5) 0.00000000292749557929 Keep right
	SIGN: General caution RECOGNIZED: False top (1) 0.97561353445053100586 Right-of-way at the next intersection top (2) 0.02226061373949050903 General caution top (3) 0.00175580696668475866 Pedestrians top (4) 0.00035783444764092565 End of all speed and passing limits top (5) 0.00000950620142248226 Traffic signals
	SIGN: Children crossing RECOGNIZED: False top (1) 0.80761754512786865234 Turn left ahead top (2) 0.09815347194671630859 End of no passing top (3) 0.08341362327337265015 Slippery road top (4) 0.01062205899506807327 Speed limit (60km/h) top (5) 0.00018462585285305977 Bicycles crossing
	SIGN: General caution RECOGNIZED: False top (1) 0.99996972084045410156 Road narrows on the right top (2) 0.00002439473792037461 General caution top (3) 0.00000381597510568099 Road work top (4) 0.00000200285057871952 Pedestrians top (5) 0.00000001963087647994 Traffic signals

Observation: Only 3 of 7 non-recognized traffic signs contain a correct prediction in the top 5.

2. Probability of Predictions with augmented training data-set

The table below lists the probabilities of the predictions of the model using the augmented training data-set:

Traffic Sign	Probability of Predictions
	SIGN: Speed limit (30km/h) RECOGNIZED: False top (1) 1.00000000000000000000 Speed limit (20km/h) top (2) 0.00000000000030439743 Speed limit (30km/h) top (3) 0.00000000000000034673 Speed limit (70km/h) top (4) 0.00000000000000000000 Speed limit (100km/h) top (5) 0.00000000000000000000 Children crossing
	SIGN: Turn right ahead RECOGNIZED: True top (1) 1.00000000000000000000 Turn right ahead top (2) 0.00000000000000000000 Stop top (3) 0.00000000000000000000 Speed limit (20km/h) top (4) 0.00000000000000000000 Speed limit (30km/h) top (5) 0.00000000000000000000 Speed limit (50km/h)
	SIGN: Slippery road RECOGNIZED: True top (1) 1.00000000000000000000 Slippery road top (2) 0.00000000000000000000 Children crossing top (3) 0.00000000000000000000 Dangerous curve to the right top (4) 0.00000000000000000000 Road work top (5) 0.00000000000000000000 Speed limit (20km/h)
	SIGN: No vehicles RECOGNIZED: False top (1) 0.99998617172241210938 No passing top (2) 0.00001382658138027182 No vehicles top (3) 0.00000000000000000000 Yield top (4) 0.00000000000000000000 End of all speed and passing limits top (5) 0.00000000000000000000 Speed limit (20km/h)
	SIGN: Speed limit (80km/h) RECOGNIZED: True top (1) 1.00000000000000000000 Speed limit (80km/h) top (2) 0.00000000103185282541 Speed limit (50km/h) top (3) 0.00000000000000000294 Speed limit (30km/h) top (4) 0.00000000000000000011 Speed limit (100km/h) top (5) 0.00000000000000000000 Speed limit (60km/h)
	SIGN: General caution RECOGNIZED: True top (1) 1.00000000000000000000 General caution top (2) 0.00000000000000000728 Road work top (3) 0.00000000000000000000 Wild animals crossing top (4) 0.00000000000000000000 Traffic signals top (5) 0.00000000000000000000 Road narrows on the right
	SIGN: Children crossing RECOGNIZED: True top (1) 1.00000000000000000000 Children crossing top (2) 0.00000000000000000000 Bicycles crossing top (3) 0.00000000000000000000 Slippery road top (4) 0.00000000000000000000 Speed limit (20km/h) top (5) 0.00000000000000000000 Pedestrians
	SIGN: General caution RECOGNIZED: True top (1) 0.99999296665191650391 General caution top (2) 0.00000699463998898864 Traffic signals top (3) 0.00000000547620615521 Road narrows on the right top (4) 0.00000000011272024980 Bicycles crossing top (5) 0.00000000010384513388 Double curve

Observation: Both non-predicted traffic signs contain the correct prediction at the 2nd rang (top 2)

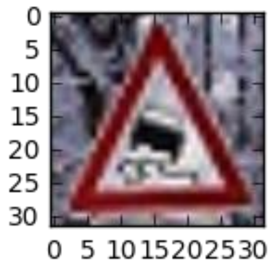
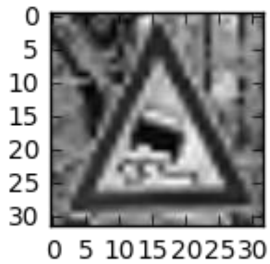
5. Visualizing the Neural Network

I visualized the features maps of the traffic sign **Slippery road** found on the WWW using the final model trained with the original training data-set and with pre-processed images (grayscaling and normalization).

The following feature maps were visualized:

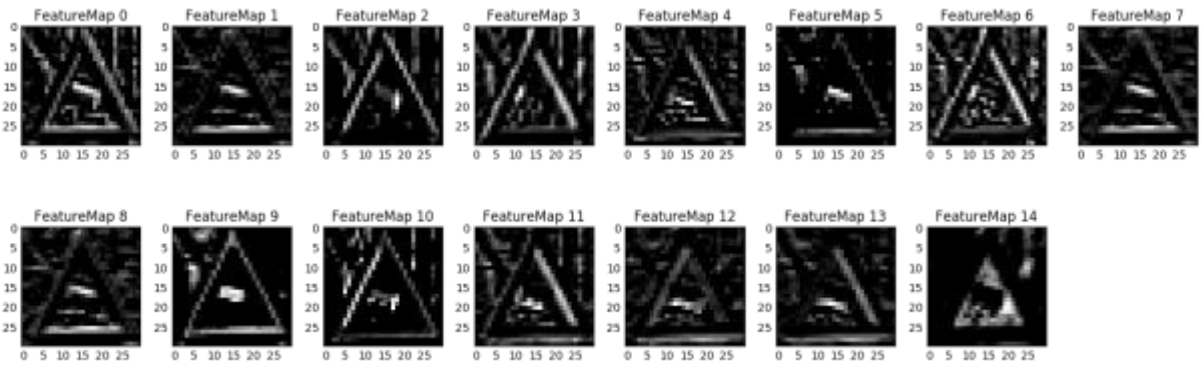
1. Conv1 activation function (ID 1.2 of final model architecture – table of chapter 3.3.2)
2. Conv1 max pooling (ID 1.3 of final model architecture – table of chapter 3.3.2)
3. Conv2 activation function (ID 2.2 of final model architecture – table of chapter 3.3.2)
4. Conv2 max pooling (ID 2.3 of final model architecture – table of chapter 3.3.2)
5. Conv1 max pooling (ID 1.4 of final model architecture – table of chapter 3.3.2)

Test Traffic Sign: **Slippery road** (in colour and pre-processed used by the model):

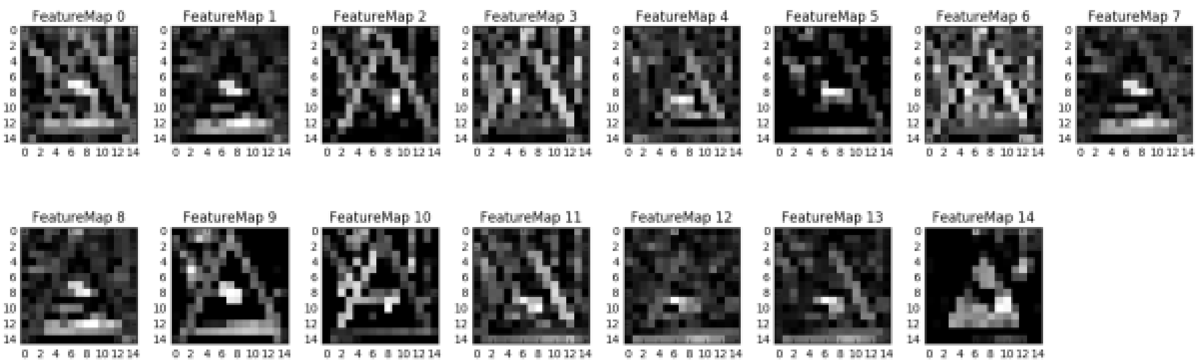
1. Conv1 activation function 30x30x15

The feature maps act as different filters versus the image.



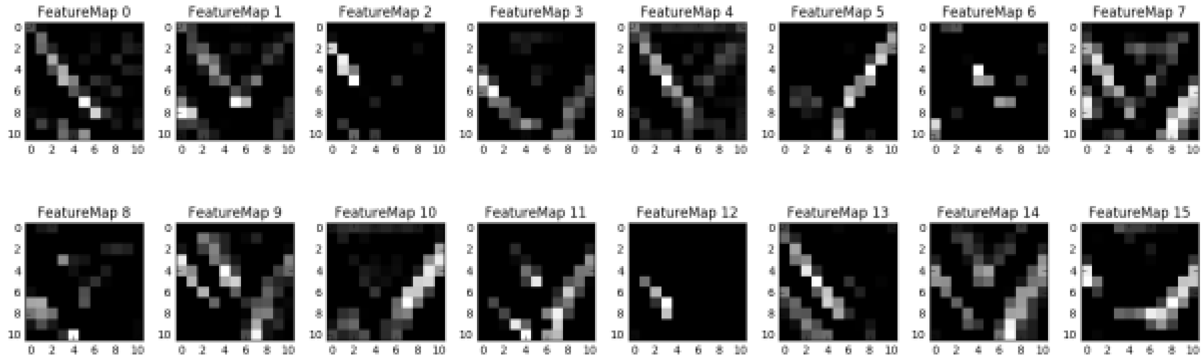
2. Conv1 max. pooling 15x15x15

The feature map filters get more rough-grain by 2x2 pooling.



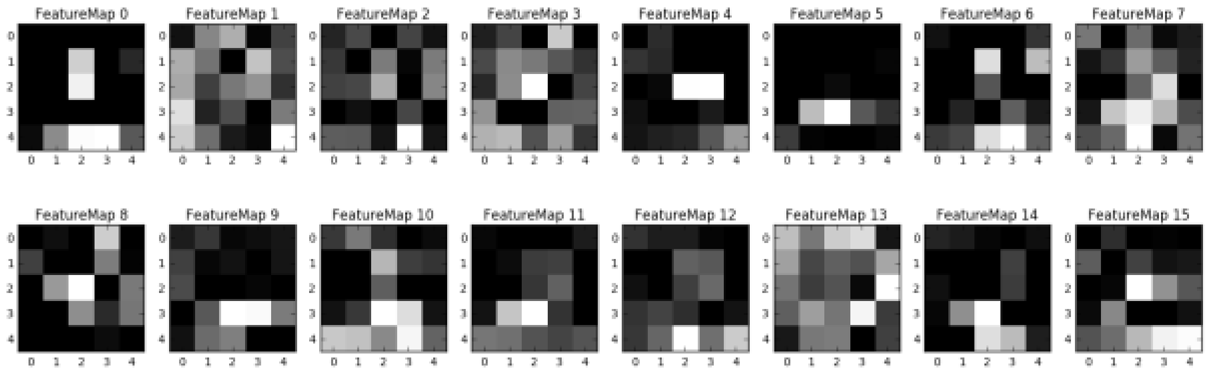
3. Conv2 activation 11x11x16

The convolution of 5x5x16 used for the conv1 max pooling output. Filters focus on different patterns.



4. Conv2 max. pooling 5x5x16

Features maps get a more rough-grained patterns by pooling 2x2



5. Conv1 max. pooling 7x7x15

Two times 2x2 pooled feature map of conv1. Feature map filter high-level patterns.