Sebastian Nietardt

Udacity Self-Driving Car Nanodegree

Project 2

February 20, 2017

Traffic Sign Recognition

Data Set Summary & Exploration

Dataset Summary

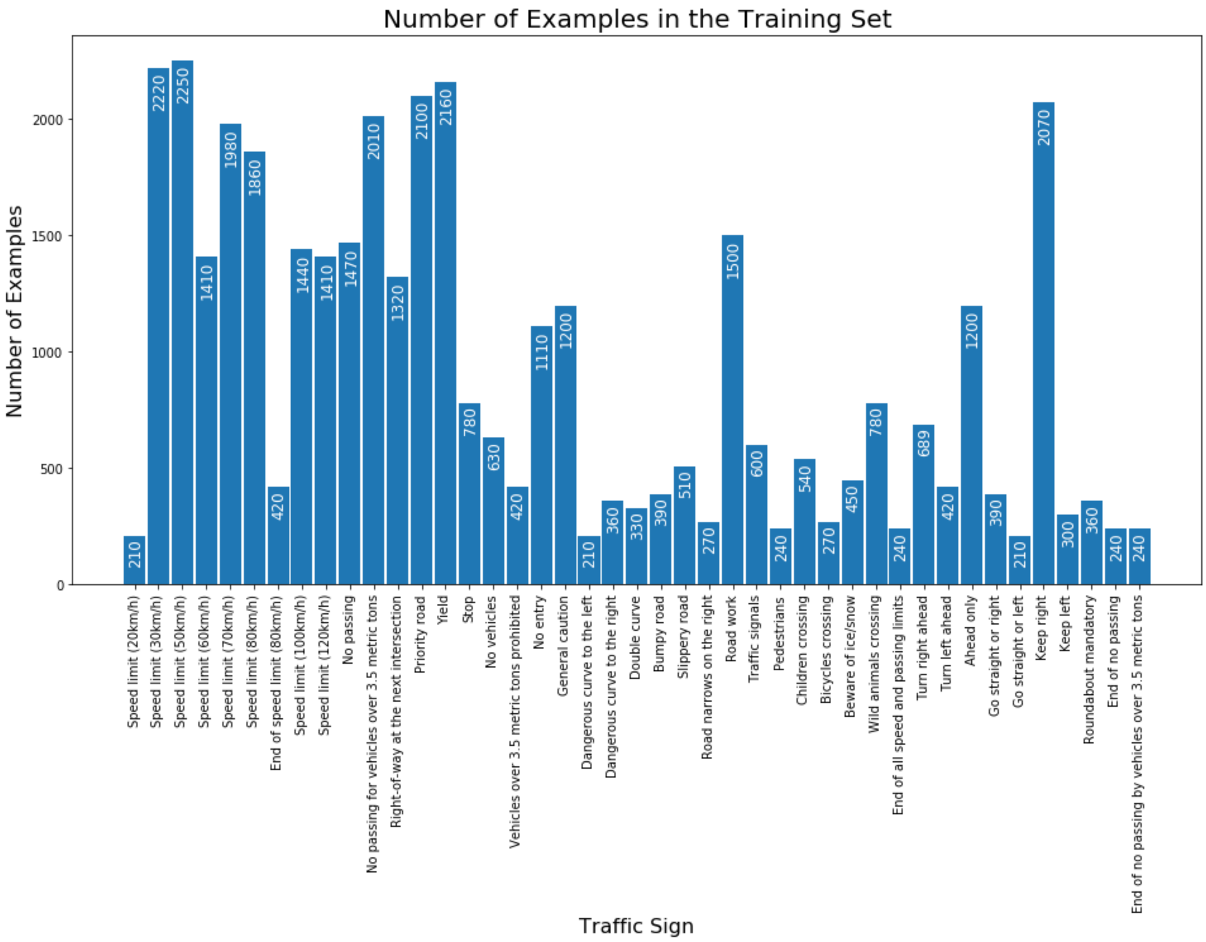
The code for this step is contained in the second code cell of the jupyter notebook.

The data set contains 39209 training examples and 12630 testing examples. Each image is in color, 32 pixels high and 32 pixels wide. The total number of different traffic signs in the data set is 43.

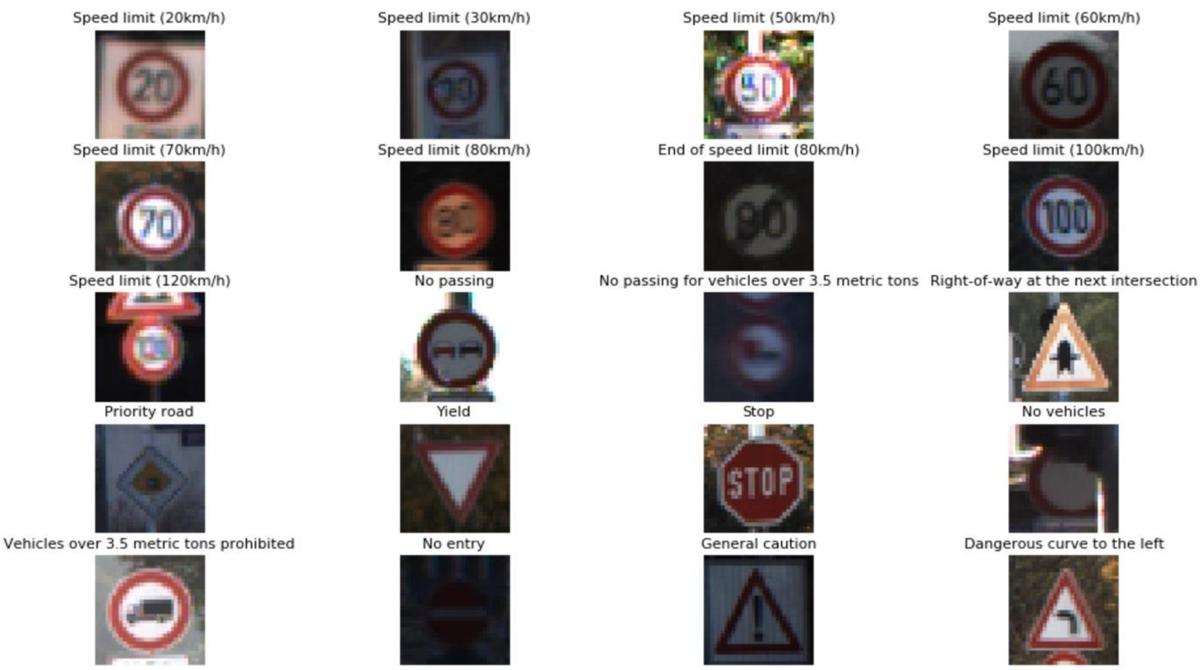
Exploratory Visualization

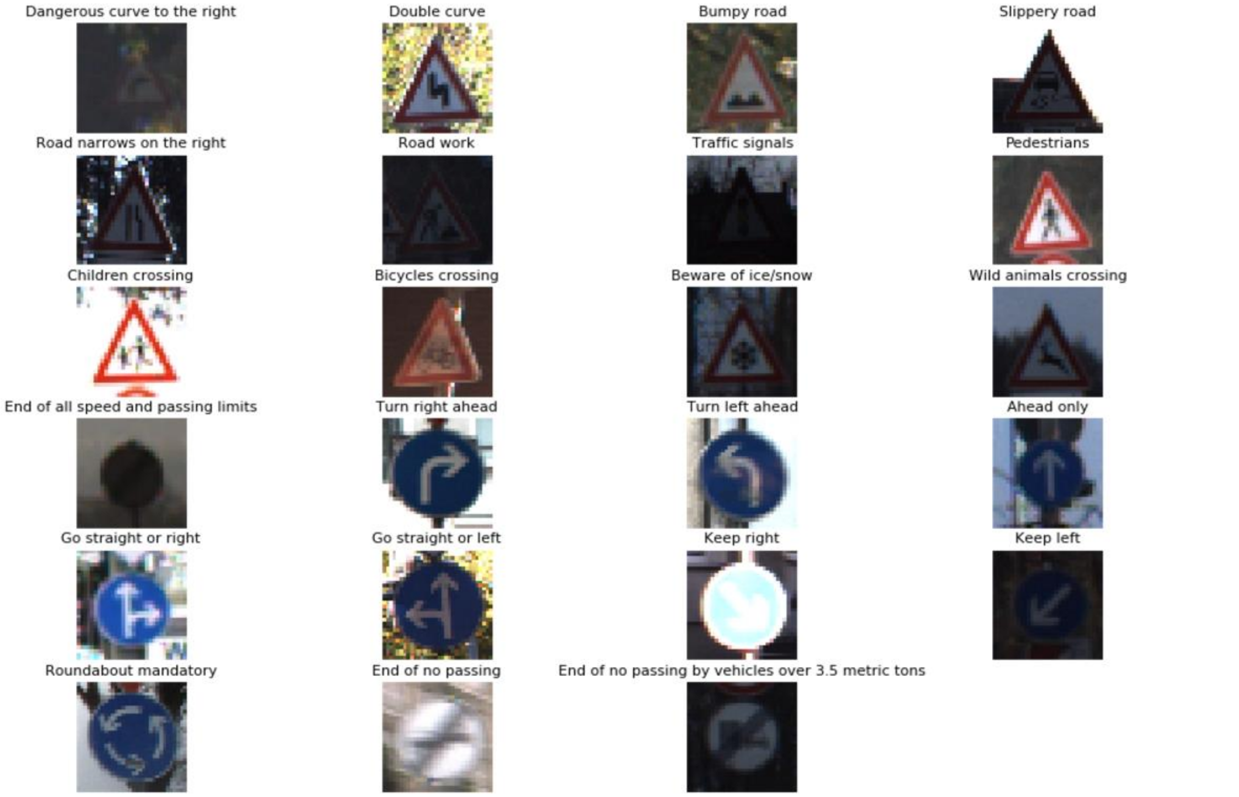
The code for this step is contained in the third and fourth code cell of the jupyter notebook.

The following bar chart shows the number of examples for each traffic sign within the data set. The number of examples for each traffic sign is very unbalanced. The traffic sign 'Speed limit (50km/h)' has the most examples with an amount of 2250. The smallest number of examples have the three traffic signs 'Speed limit (20km/h)', 'Dangerous curve to the left' and 'Go straight or left' with an amount of 210. This is more than ten times less than 'Speed limit (50km/h)'.



To get an impression of the quality of the pictures within the data set, one example of each traffic sign is displayed in the following. The picture quality differs a lot. Some pictures are clear and the traffic signs are easy to recognize. Other pictures are blurred or dark. Some traffic signs are skewed, have shadows on them, reflect the sun or are partly overlapped by something else.





Design and Test a Model Architecture

Preprocessing

The code for this step is contained in the fifth and sixth code cell of the jupyter notebook.

The pictures had been already preprocessed before the project. They are in the correct size of 32x32 pixels, the traffic signs are centered and zoomed.

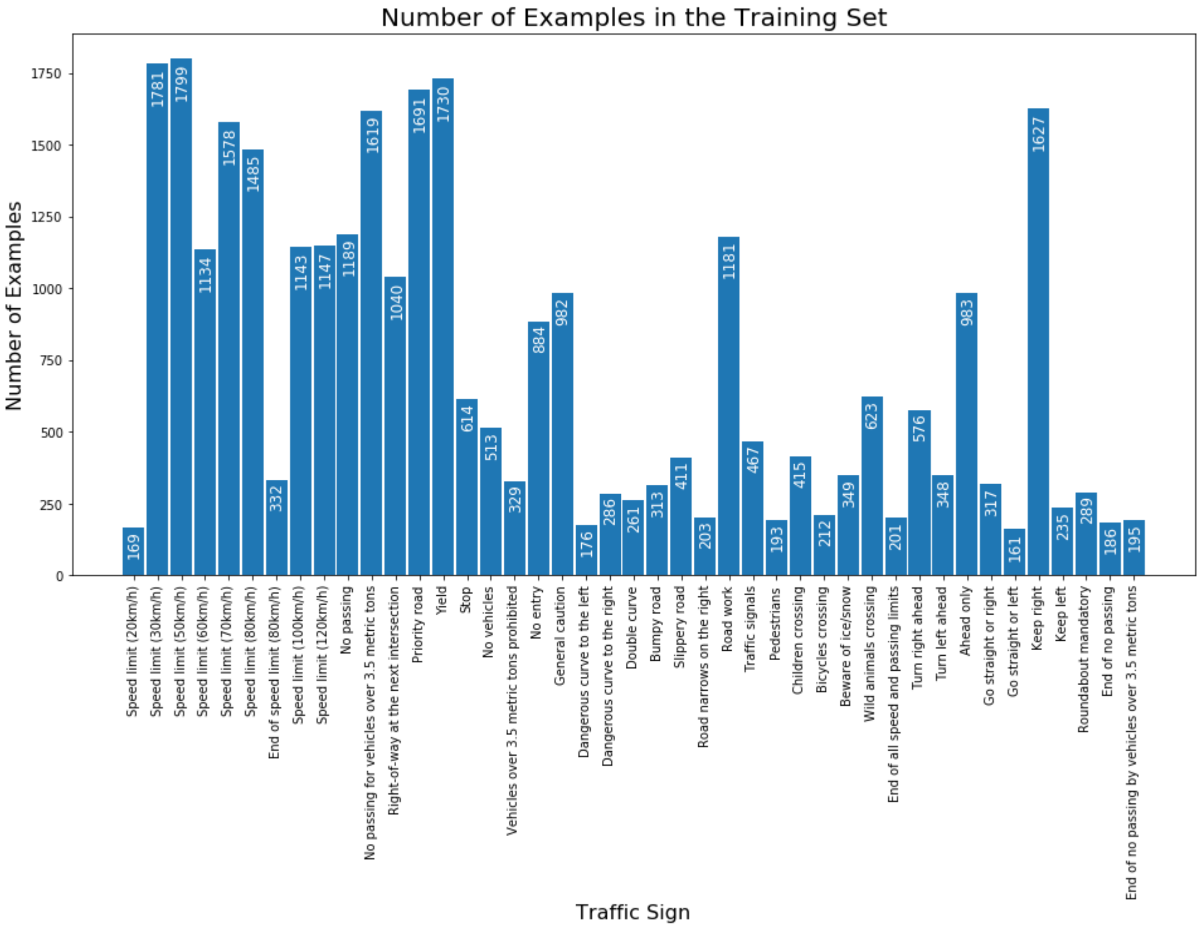
In this project, the data set had been gray scaled and normalized. Gray scaling simplifies the image, makes calculations faster and makes it easier for algorithms to recognize edges because it can focus on one value instead of three. The gray scaling had been done with OpenCV which puts more weight on green and less weight on blue than the $\frac{(red+green+blue)}{3}$ gray scaling by using the following formula:

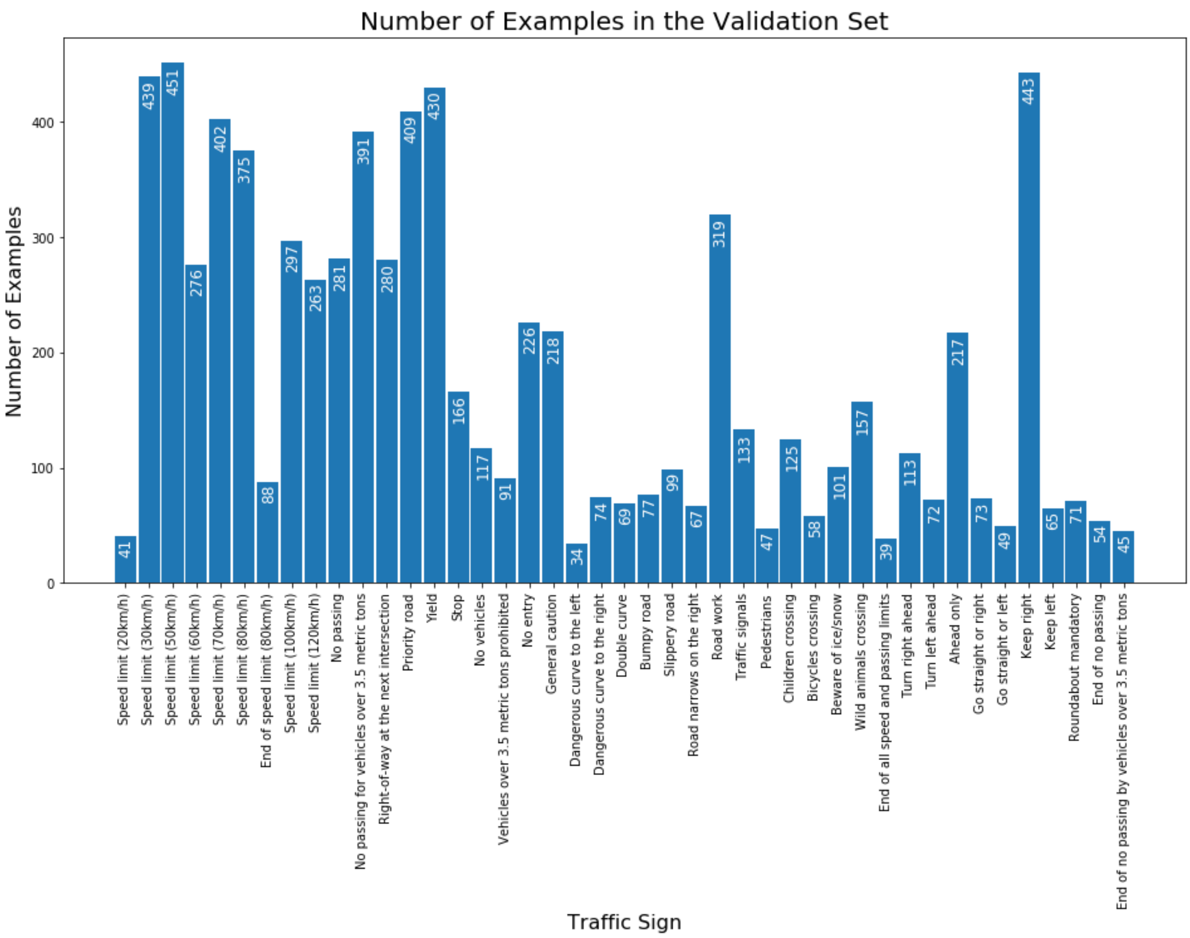
$$0.299 * red + 0.587 * green + 0.114 * blue$$

After gray scaling the resulted gray value was normalized to a float value between -1 and 1. It was done my dividing the gray scaled data (which is a value between 0 and 255) by 127.5 and subtracting 1. A value between -1 and 1 is usually good for several activation functions. Very high or very low values could lead to side effects in gradient calculations or forces the model to massively adapt its weights.

Because of the lack of a validation set, the training set needed to be split. 20% of the training data was randomly cut off the training set and put into a validation set by using the train_test_split function from scikit-learn.

The following bar chart shows the number of examples in the training and validation set after the split. The code for this step is contained in the seventh and eights code cell of the jupyter notebook.


Number of Examples in the Training Set
Number of Examples
Traffic Sign


Number of Examples in the Validation Set
Number of Examples
Traffic Sign

Model Architecture

The code for my final model is located in the ninth cell of the jupyter notebook.

My final model consisted of the following layers:

Layer	Description
Input	32x32x1 gray image
Convolution 3x3	1x1 stride, valid padding, outputs 28x28x12
RELU	
Max pooling	2x2 stride, outputs 14x14x12
Dropout	50% keep probability on training
Convolution 3x3	1x1 stride, valid padding, outputs 10x10x32
RELU	
Max pooling	2x2 stride, outputs 5x5x32
Dropout	50% keep probability on training
Flatten	Output 800
Fully connected	Output 300
RELU	
Dropout	50% keep probability on training
Fully connected	Output 200
RELU	
Dropout	50% keep probability on training
Fully connected	Output 43

Model Training

The code for training the model is located in the tenth, eleventh and thirteenth cell of the jupyter notebook.

For training the model a learning rate of 0.001 has been used. The epochs where set to 100 and the batch size was set to 128. The Adam algorithm was used as an optimizer.

Solution Design

The code for calculating the accuracy of the model is located in the twelves, thirteenth and fourteenth cell of the jupyter notebook.

The final model results were 99.6% accuracy on the training set, 99.1% accuracy on the validation set and 95.6% accuracy on the testing set. Since the accuracy on the training set is much higher than the accuracy on the testing set, it looks like the model is still overfitting.

The first architecture which was tried in the data set was the LeNet architecture with color images and an output of 43 logits. The LeNet architecture was originally used to recognize handwritten numbers from images but works also good on other image recognition problems. Gray scaling the images improved the accuracy as well as the normalization of the data. Since the traffic sign data set contains images of 43 different traffic signs and not just 10, the depth of all the layers was increased in order to store more important features of the images. After trying different depth sizes, the depth of the first convolution layer had been increased to 12 and the depth of the second convolution layer had been increased to 32. After the second convolution layer, the data is flattened because three fully connected layer follow. The first fully connected layer was increased to an output of 300, the second fully connected layer was increased to an output of 200 and the last fully connected layer was increased to an output of the 43 logits which represents the number of different traffic signs in the data set. After that the model seemed to overfit. In order to prevent overfitting, a dropout layer had been added after each activation function. A probability of 50% to keep the inputs turned out to work best. Several values for the training rate, epochs and batch sizes had also been tried. 100 epochs, a batch size of 128 and a learning rate of 0.001 turned out to be good and performant. A validation accuracy of 95.6% is a good result for a start and seeing the quality of the images, the model should work on other real world data. It just needs to be preprocessed like the date in the given dataset.

Test a Model on New Images

Acquiring New Images

The code for displaying the new images is located in the fifteenth and sixteenth cell of the jupyter notebook.

The following images had been acquired from the internet and resized to 32x32 pixels.

The new images should be a mixture of images which are clear to see and some images where the traffic sign is harder to recognize.

Let us have a look at the first row. The first and the fourth image should be easy to recognize. The traffic signs are clear to see. The second image is difficult to read because it is covered with snow. I myself do not think that the model is able to recognize it. The third sign is clear to read but the traffic sign has a sticker on it and it is skew. This image could also be hard to classify.

The second row contains just images which are clear to read. The first three should be easy for the model, even though the traffic sign in the second image is partly overlapped by a traffic light. The fourth image is a little skew. This could also be hard to classify.

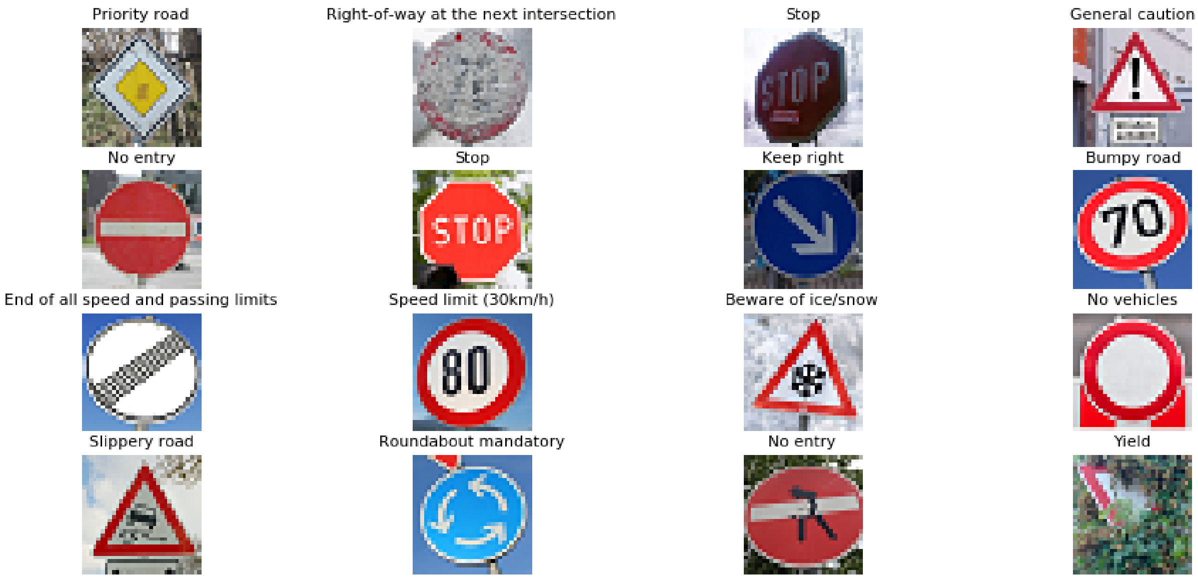
The images in the third row are also easy to recognize. The second and the third picture are a little skew. The background of the third image is gray/white, like the traffic sign itself. This might make it a little harder for the model.

The first image in the fourth row is clear to recognize. The second one is a little skew, but since it is a round sign with similar arrows on it, the rotation should not be an issue. The third sign has a funny sticker on it, like one can see it in some cities. This sticker might make it a little harder for the model to recognize the traffic sign. The last image is the most interesting one in my opinion. It shows a 'yield' sign which is covered half by a plant. This should be hard for the model to recognize as well.

Performance on New Images

The code for making predictions on the final model is located in the eighteenth and nineteenth cell of the jupyter notebook.

The following picture shows the prediction of the model:



In the first row 'Speed limit (70km/h)' sign was wrongly classified as 'Right-of-way at the next intersection'. That is not surprising because of all the snow. The other three traffic signs ('Priority road', 'Stop' and 'General caution') are correctly classified. Even the skew stop sign with the sticker on it.

In the second row the 'Speed limit (70km/h)' sign was wrongly classified as 'Bumpy road'. This is kind of surprising, because it is easy to read. The other three ('No entry', 'Stop' and 'Keep right') are classified correctly.

In the third row the 'Speed limit (80km/h)' was mistaken by a 'Speed limit (30km/h)' sign. The three others ('End of all speed and passing limits', 'Beware of ice/snow' and 'No vehicles') have been guessed correctly.

The four traffic signs ('Slippery road', 'Roundabout mandatory', 'No entry' and 'Yield') in the fourth row were all correctly classified. Even the last one which is covered with plants.

The model seems to have a problem with speed limits since all three speed limit traffic signs were interpret as something else. 13 of 16 new images had been classified correctly which means the accuracy for the new images is 81.2%. Compared to the accuracy on the testing set, the accuracy on the new images is way lower. The two accuracies are not exactly comparable, though. There are just 16 new images, while the testing set contains of 12,630 images. With just 16 images, the accuracy can vary a lot depending on which images where chosen. In addition, the 16 images where not chosen randomly. They were chosen manually and by the students' curiosity.

Model Certainty - Softmax Probabilities

The code for making predictions on the final model is located in the twentieth cell of the jupyter notebook.

For the 'Priority road' sign the model is sure of its prediction. The probability is 100%.

Original traffic sign was: 12 (Priority road)
Predicted with a probability of 100.000% to be 12 (Priority road)
Predicted with a probability of 0.000% to be 40 (Roundabout mandatory)
Predicted with a probability of 0.000% to be 35 (Ahead only)
Predicted with a probability of 0.000% to be 7 (Speed limit (100km/h))
Predicted with a probability of 0.000% to be 13 (Yield)

For the first 'Speed limit (70km/h)' sign the model made a wrong prediction. The correct sign is not even among the top 5. The top 5 probabilities are:

Original traffic sign was: 4 (Speed limit (70km/h))
Predicted with a probability of 24.321% to be 11 (Right-of-way at the next intersection)
Predicted with a probability of 23.936% to be 33 (Turn right ahead)
Predicted with a probability of 10.588% to be 40 (Roundabout mandatory)
Predicted with a probability of 7.994% to be 35 (Ahead only)
Predicted with a probability of 7.181% to be 12 (Priority road)

For the first 'Stop' sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 14 (Stop)
Predicted with a probability of 84.130% to be 14 (Stop)
Predicted with a probability of 4.274% to be 3 (Speed limit (60km/h))
Predicted with a probability of 2.866% to be 33 (Turn right ahead)
Predicted with a probability of 2.400% to be 34 (Turn left ahead)
Predicted with a probability of 0.826% to be 35 (Ahead only)

For the 'General caution' sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 18 (General caution)
Predicted with a probability of 99.954% to be 18 (General caution)
Predicted with a probability of 0.046% to be 26 (Traffic signals)
Predicted with a probability of 0.000% to be 22 (Bumpy road)
Predicted with a probability of 0.000% to be 15 (No vehicles)
Predicted with a probability of 0.000% to be 31 (Wild animals crossing)

For the first 'No entry' sign the model is sure of its prediction. The probability is 100%.

Original traffic sign was: 17 (No entry)
Predicted with a probability of 100.000% to be 17 (No entry)
Predicted with a probability of 0.000% to be 33 (Turn right ahead)
Predicted with a probability of 0.000% to be 34 (Turn left ahead)
Predicted with a probability of 0.000% to be 14 (Stop)
Predicted with a probability of 0.000% to be 9 (No passing)

For the second ‘Stop’ sign the model is sure of its prediction. The probability is 100%.

Original traffic sign was: 14 (Stop)
Predicted with a probability of 100.000% to be 14 (Stop)
Predicted with a probability of 0.000% to be 3 (Speed limit (60km/h))
Predicted with a probability of 0.000% to be 33 (Turn right ahead)
Predicted with a probability of 0.000% to be 1 (Speed limit (30km/h))
Predicted with a probability of 0.000% to be 2 (Speed limit (50km/h))

For the ‘Keep right’ sign the model is sure of its prediction. The probability is 100%.

Original traffic sign was: 38 (Keep right)
Predicted with a probability of 100.000% to be 38 (Keep right)
Predicted with a probability of 0.000% to be 10 (No passing for vehicles over 3.5 metric tons)
Predicted with a probability of 0.000% to be 34 (Turn left ahead)
Predicted with a probability of 0.000% to be 13 (Yield)
Predicted with a probability of 0.000% to be 5 (Speed limit (80km/h))

The second ‘Speed limit (70km/h)’ sign was incorrectly classified. The correct sign has, with a value of 43.801%, the second highest probability. The top 5 probabilities are:

Original traffic sign was: 4 (Speed limit (70km/h))
Predicted with a probability of 49.937% to be 22 (Bumpy road)
Predicted with a probability of 43.801% to be 4 (Speed limit (70km/h))
Predicted with a probability of 4.556% to be 15 (No vehicles)
Predicted with a probability of 0.739% to be 1 (Speed limit (30km/h))
Predicted with a probability of 0.711% to be 2 (Speed limit (50km/h))

For the ‘End of all speed and passing limits’ sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 32 (End of all speed and passing limits)
Predicted with a probability of 99.908% to be 32 (End of all speed and passing limits)
Predicted with a probability of 0.081% to be 41 (End of no passing)
Predicted with a probability of 0.011% to be 6 (End of speed limit (80km/h))
Predicted with a probability of 0.000% to be 36 (Go straight or right)
Predicted with a probability of 0.000% to be 12 (Priority road)

The ‘Speed limit (80km/h)’ sign was incorrectly classified. The model was, with a probability of 83.153%, relatively sure is was a ‘Speed limit (30km/h)’ sign. The correct sign has, with a value of 0.904%, the third highest probability. The top 5 probabilities are:

Original traffic sign was: 5 (Speed limit (80km/h))
Predicted with a probability of 83.153% to be 1 (Speed limit (30km/h))
Predicted with a probability of 14.709% to be 2 (Speed limit (50km/h))
Predicted with a probability of 0.904% to be 5 (Speed limit (80km/h))
Predicted with a probability of 0.473% to be 8 (Speed limit (120km/h))
Predicted with a probability of 0.386% to be 0 (Speed limit (20km/h))

The ‘Beware of ice/snow’ sign was correctly classified, but the model was not very sure about its prediction. The probability was just 49.217%. The top 5 probabilities are:

Original traffic sign was: 30 (Beware of ice/snow)
Predicted with a probability of 49.217% to be 30 (Beware of ice/snow)
Predicted with a probability of 29.365% to be 28 (Children crossing)
Predicted with a probability of 21.037% to be 11 (Right-of-way at the next intersection)
Predicted with a probability of 0.255% to be 29 (Bicycles crossing)
Predicted with a probability of 0.092% to be 23 (Slippery road)

For the 'No vehicles' sign the model is sure of its prediction. The probability is 100%.

Original traffic sign was: 15 (No vehicles)
Predicted with a probability of 100.000% to be 15 (No vehicles)
Predicted with a probability of 0.000% to be 9 (No passing)
Predicted with a probability of 0.000% to be 2 (Speed limit (50km/h))
Predicted with a probability of 0.000% to be 4 (Speed limit (70km/h))
Predicted with a probability of 0.000% to be 32 (End of all speed and passing limits)

For the 'Slippery road' sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 23 (Slippery road)
Predicted with a probability of 83.981% to be 23 (Slippery road)
Predicted with a probability of 11.043% to be 30 (Beware of ice/snow)
Predicted with a probability of 3.373% to be 20 (Dangerous curve to the right)
Predicted with a probability of 0.730% to be 28 (Children crossing)
Predicted with a probability of 0.579% to be 29 (Bicycles crossing)

For the 'Roundabout mandatory' sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 40 (Roundabout mandatory)
Predicted with a probability of 98.821% to be 40 (Roundabout mandatory)
Predicted with a probability of 0.953% to be 12 (Priority road)
Predicted with a probability of 0.064% to be 7 (Speed limit (100km/h))
Predicted with a probability of 0.039% to be 35 (Ahead only)
Predicted with a probability of 0.038% to be 38 (Keep right)

For the 'No entry' sign, the model is relatively sure of its prediction. The top 5 probabilities are:

Original traffic sign was: 17 (No entry)
Predicted with a probability of 99.939% to be 17 (No entry)
Predicted with a probability of 0.045% to be 33 (Turn right ahead)
Predicted with a probability of 0.014% to be 9 (No passing)
Predicted with a probability of 0.001% to be 34 (Turn left ahead)
Predicted with a probability of 0.000% to be 16 (Vehicles over 3.5 metric tons prohibited)

The plant covered 'Yield' sign was correctly classified and the model was even relatively certain about it. The probability of a yield sign was 84.552%. The top 5 probabilities are:

Original traffic sign was: 13 (Yield)
Predicted with a probability of 84.552% to be 13 (Yield)
Predicted with a probability of 9.763% to be 35 (Ahead only)
Predicted with a probability of 1.290% to be 10 (No passing for vehicles over 3.5 metric tons)
Predicted with a probability of 1.001% to be 9 (No passing)
Predicted with a probability of 0.615% to be 33 (Turn right ahead)