Traffic Sign Recognition

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 individually and describe how I addressed each point in my implementation.

Writeup / README

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

Here is a link to my project code

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.

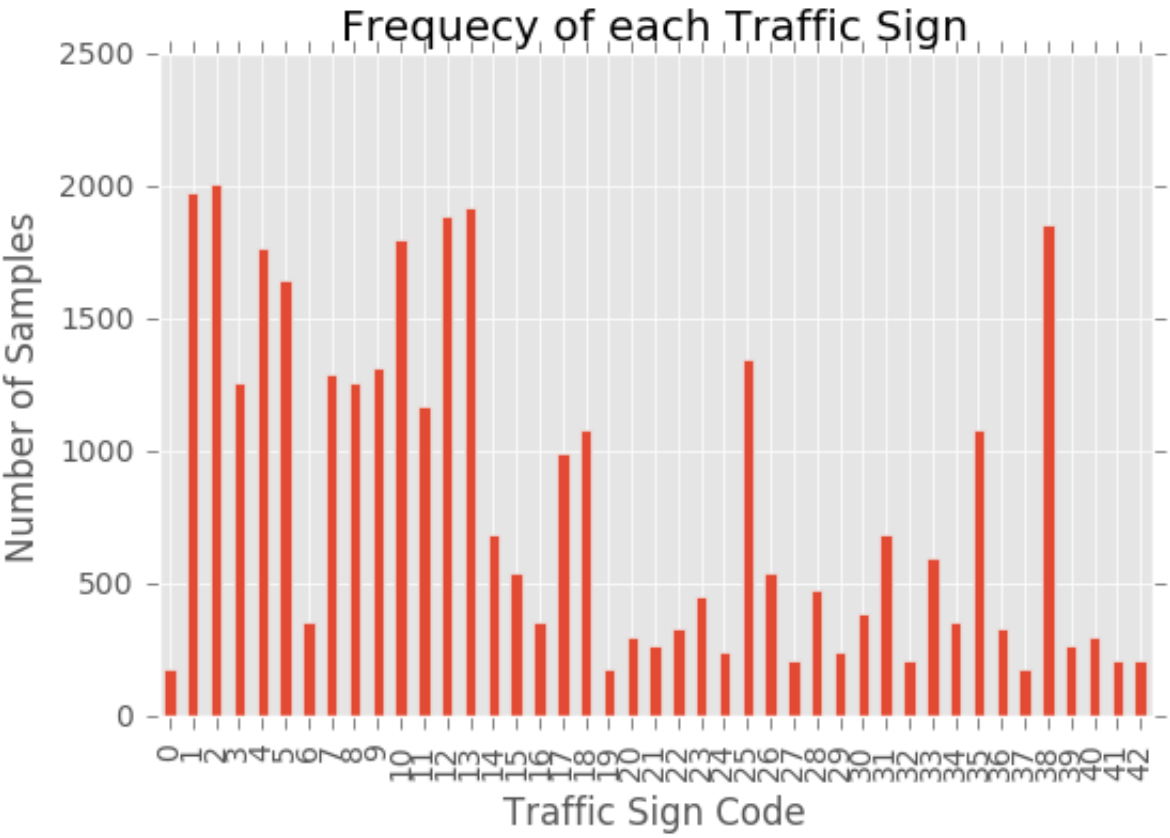
I used the pandas library to calculate summary statistics of the traffic signs data set:

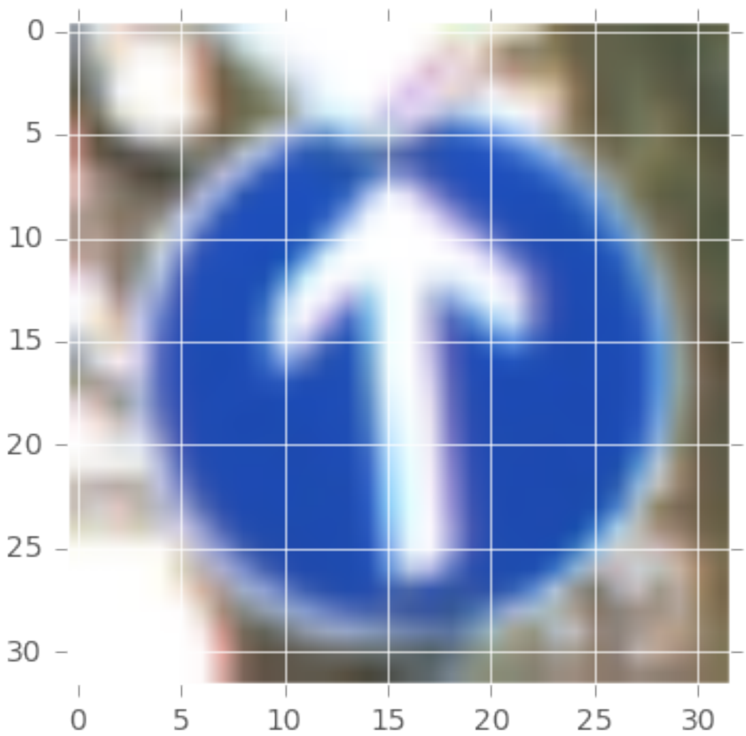
- Number of training examples = 34799
- The size of the validation set is ?
- Number of testing examples = 12630
- Image data shape = (32, 32)

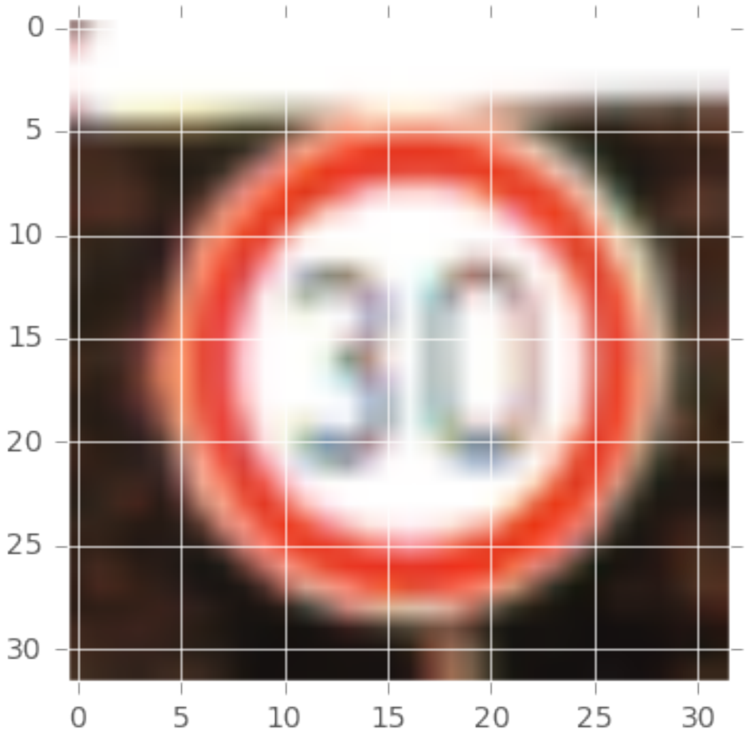
- Number of classes = 43

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 ...







Design and Test a Model Architecture

1. Describe how you preprocessed the image data.

- The data was shuffled to prevent the order of the images affecting the model
- The data was normalised to help the model converge
- Images were augmented with random changes including:
    - Flipping left and right
    - Random rotation
    - Random blur

The augmentation of the images helped the model generalise over the test set as the model was able to train on a larger variety of images than what was gathered in the training set.

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.

My final model was a modified LeNet that consisted of the standard layers plus a dropout after each maxpooling layer. The Model archecture is outlined in the table below:

Layer	Description
Input	32x32x3 RGB image
Convolution 5x5	1x1 stride, valid padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6, padding valid
Dropout	Dropout rate 0.2
Convolution 5x5	1x1 stride, valid padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, outputs 14x14x6, padding valid
Dropout	Dropout rate 0.2
Flatten	Flatten conv layer to 400 fully connected layer
Fully connected	400 nodes
RELU	
Dropout	Dropout rate 0.2
Fully connected	120 nodes
RELU	
Dropout	Dropout rate 0.2
Fully connected	84 nodes
Fully connected	43 nodes
Softmax	

3. Describe how you trained your model.

The model was trained using Adam optimiser. The standard batch size was increased to 256 to take advantage of the AWS g2.2Large instance using GPU acceleration. Dropout was included to prevent overfitting and a low learning rate was chosen to allow the model to better converge on the solution.

- EPOCHS = 100
- BATCH_SIZE = 256
- dropout = 0.2

- num_classes = 43
- Learning rate = 0.0001

- Validation Accuracy = 0.936

- Test Accuracy = 0.928

4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93.

My final model results were: * training set accuracy of 100% * validation set accuracy of 93.60% * test set accuracy of 92.80%

The approach to building this model was an iterative one to obtain the final result. This involved building a LeNet with standard tuning parameters to use as a baseline for validation set accuracy. This model was able to achieve around 90% accuracy on the validation set, but struggled to get any higher even with more epochs.

The next step was to try image augmentation as a preprocessing step to the LeNet. A random rotation, blur and flipping was done to the images before they were fit by the LeNet in TFLearn. The result was similar to the original LeNet's performance and it did not get above 91% accuracy even after 100 epochs.

After looking into this problem further, it became obvious that the LeNet architecture was performing well as within 9 epochs it already had an accuracy of 90% and that it could not converge beyond this point. The next experiment was to reduce the learning rate to 0.0001. This rationale behind this was to prevent overshooting the convergence point for model fit. This resulted in a validation accuracy of 93.60%.

Some modifications were made to the standard LeNet to improve model performance. This modifications was adding dropout of 20% after each convolution layer. The rationale here was to prevent overfitting the data by turning off some of the nodes which allows some redundacy and a class voting schema to be setup within the network to improve accuracy.

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report.

Below are the 7 German traffic signs that I found on the web. The third (general caution) and seventh image (turn right) might be difficult to classify because they contain a water mark which may interfere with the feature detection. The other images should be easily identified as they are similar to training images. The main difference between these images and the training set is that the training set's images are zoomed in on the sign whereas these images from the internet have been taken at a further distance. It will be interesting to see how the model performs under these conditions.


Achtung
Unfallschwerpunkt


50

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set.

Here are the results of the prediction:

Image	Prediction
Road Work	Road Work
Do Not Enter	No Entry
General Caution	Children Crossing
50 km/h	Slippery Road
Right of Way	Right of way at next intersection
Road Work 2	Road Work
Turn right	Turn right ahead

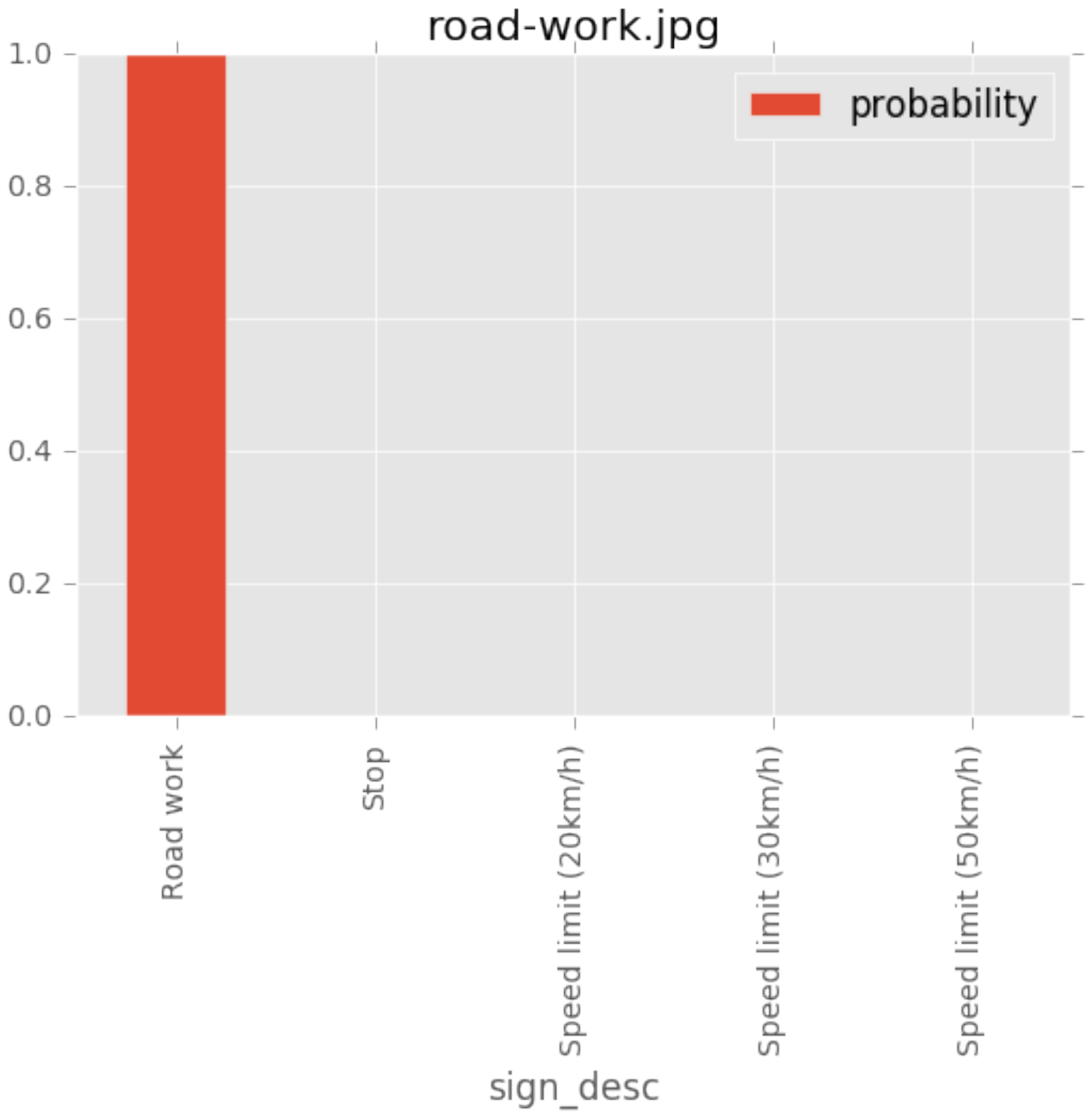
The algorithm correctly identified 5 out of 7 or 71.43% of the new images from the internet. This is poor compared to the test set accuracy of 92.80%. This could be due to the images being taken at a greater distance or including noise from other parts of the image such as other signs hanging on the same post as in the general caution image.

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.

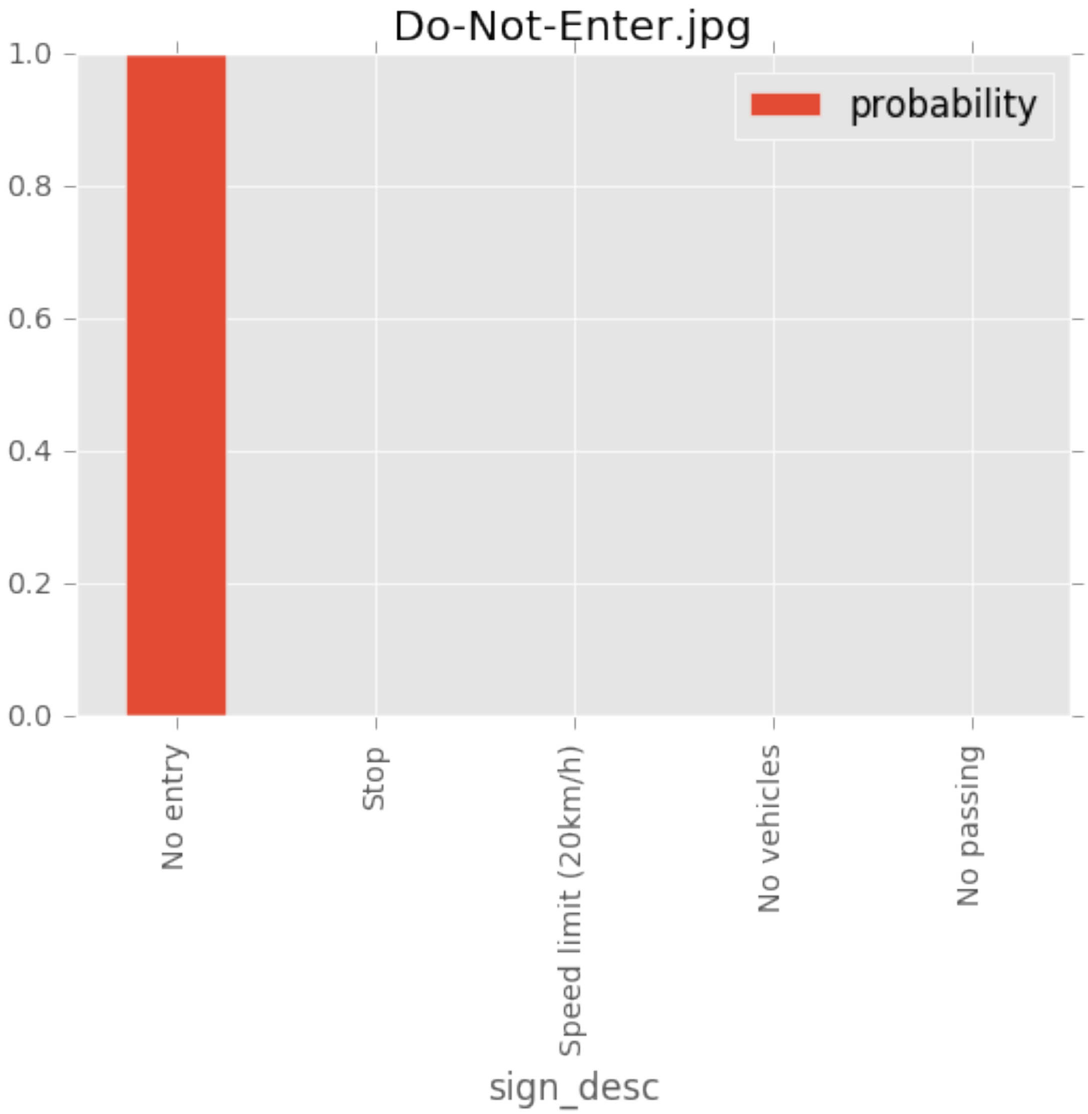
The code for making predictions on my final model is located in the 11th cell of the Ipython notebook.

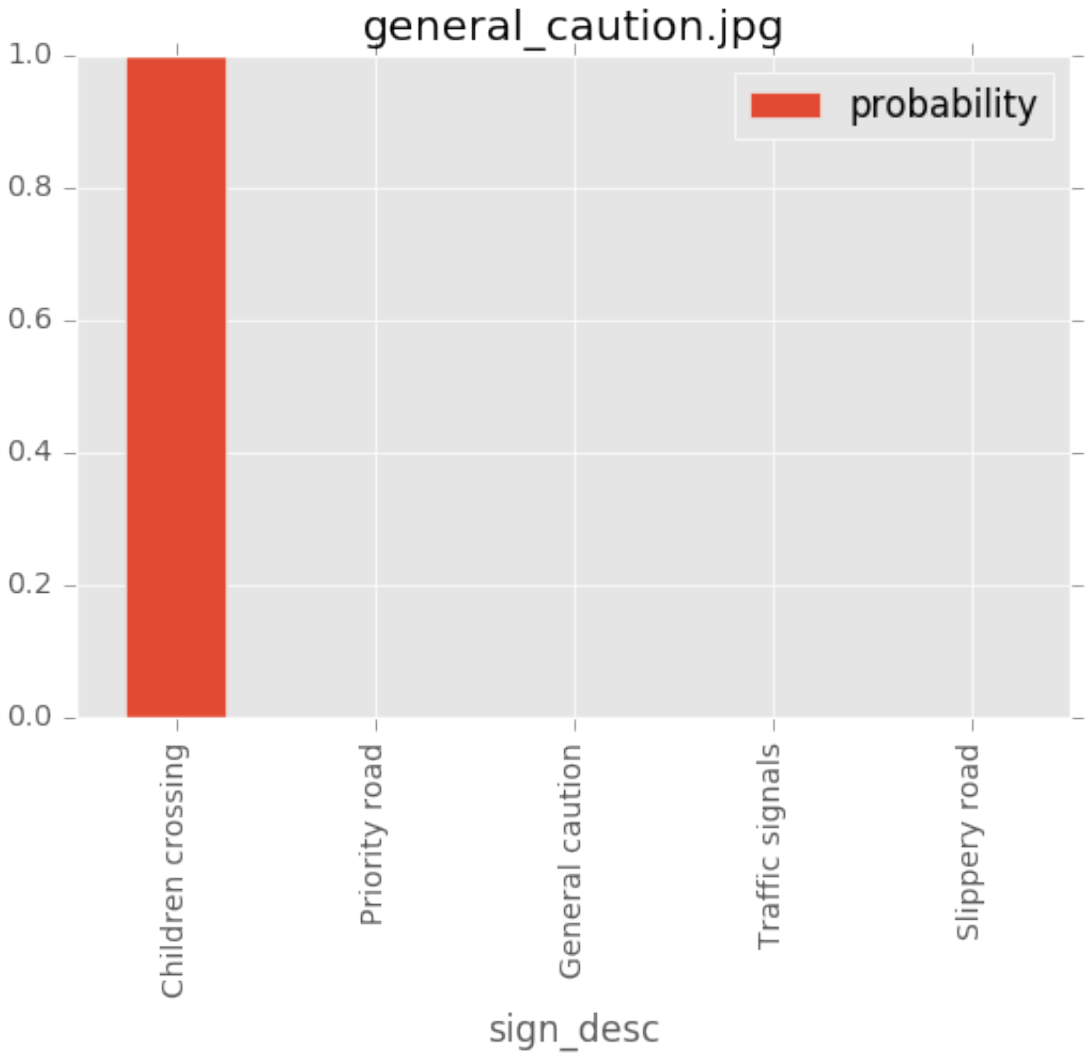
1. For the road work sign the model is 100% certain it is a road work signt
2. The model is 100% certain of the No Entry Sign
3. The model is 99.99% sure that the General Caution sign is a childrens crossing. This mistake is easy to make as both the true sign and the mistaken sign are red triangles with a black icon inside and as can be seen in the convolutional features at the end of this document, the algorithm looks for the outline of the signs to identify them
4. The model is 62% certain the 50km/h sign is a slippery road sign. This mistake is not easily excusable as the two signs have very different outlines (one triangle and one round)
5. The model is 100% certain of the right of way sign
6. The model is 100% certain of the roadwork 2 sign
7. The model is 100% certain of the turn right sign

The images below visually represent the softmax probabilities of the predictions and their likelihood.

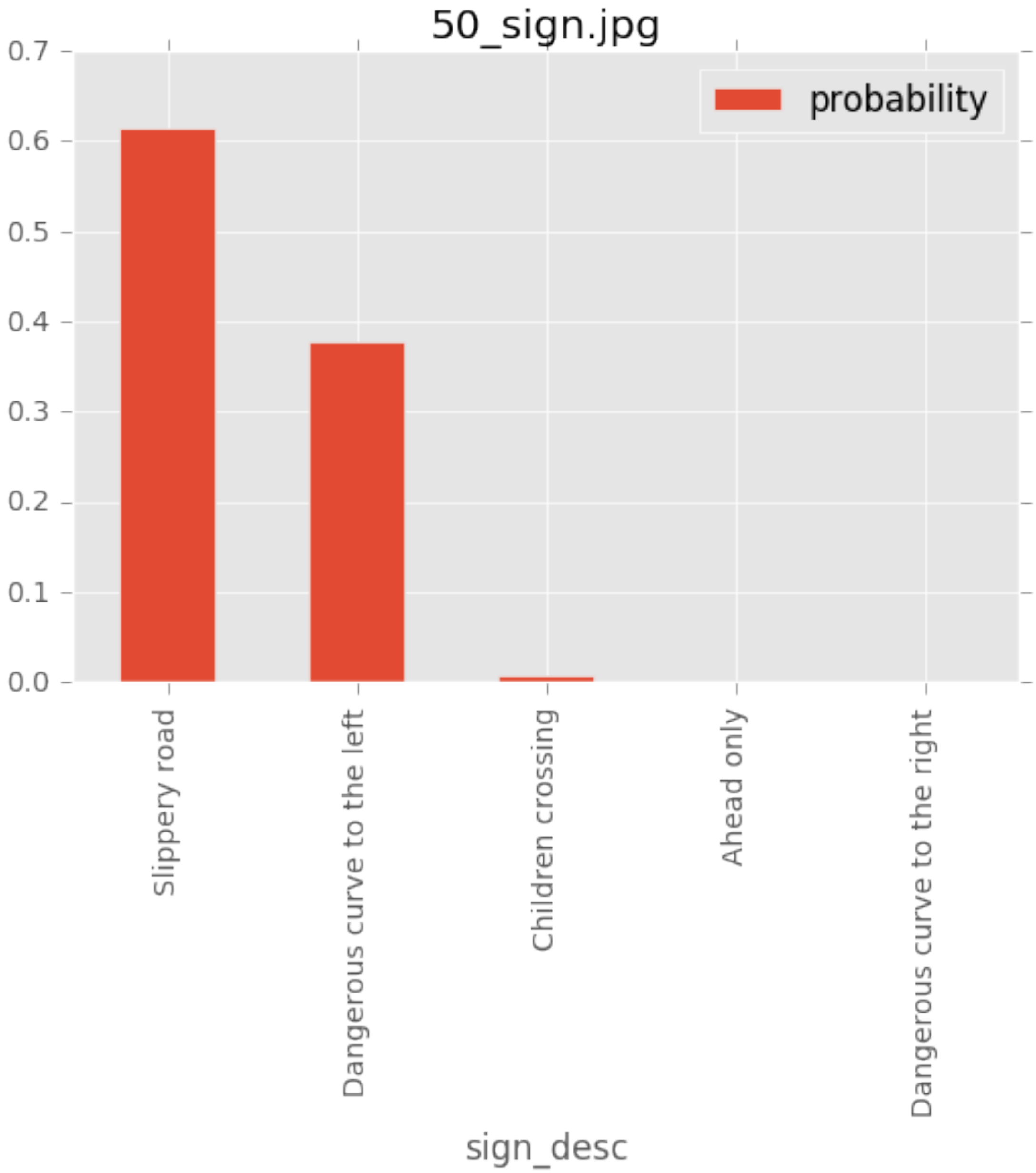


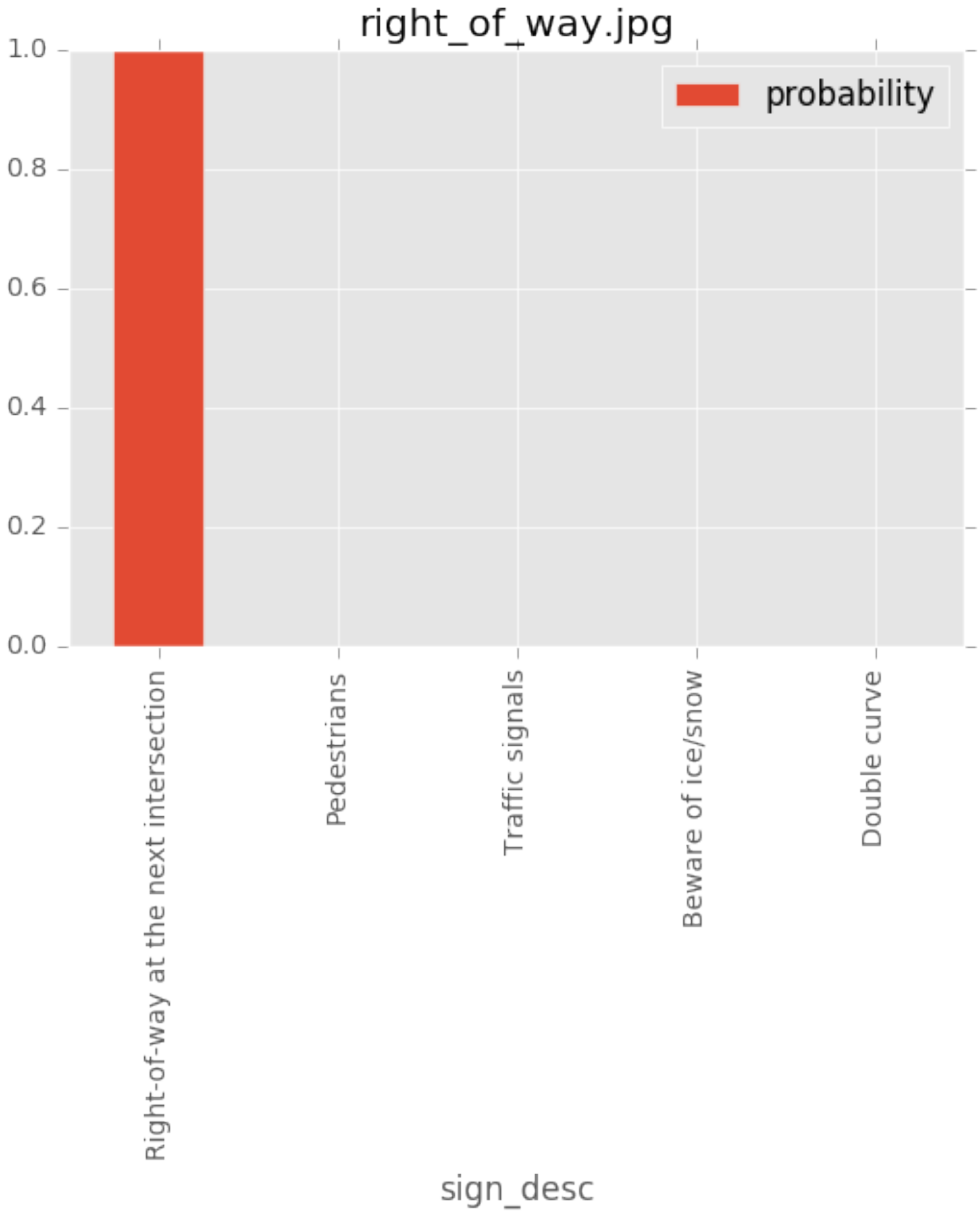
road-work.jpg
probability
1.0
0.8
0.6
0.4
0.2
0.0
Road work
Stop
Speed limit (20km/h)
Speed limit (30km/h)
Speed limit (50km/h)
sign_desc

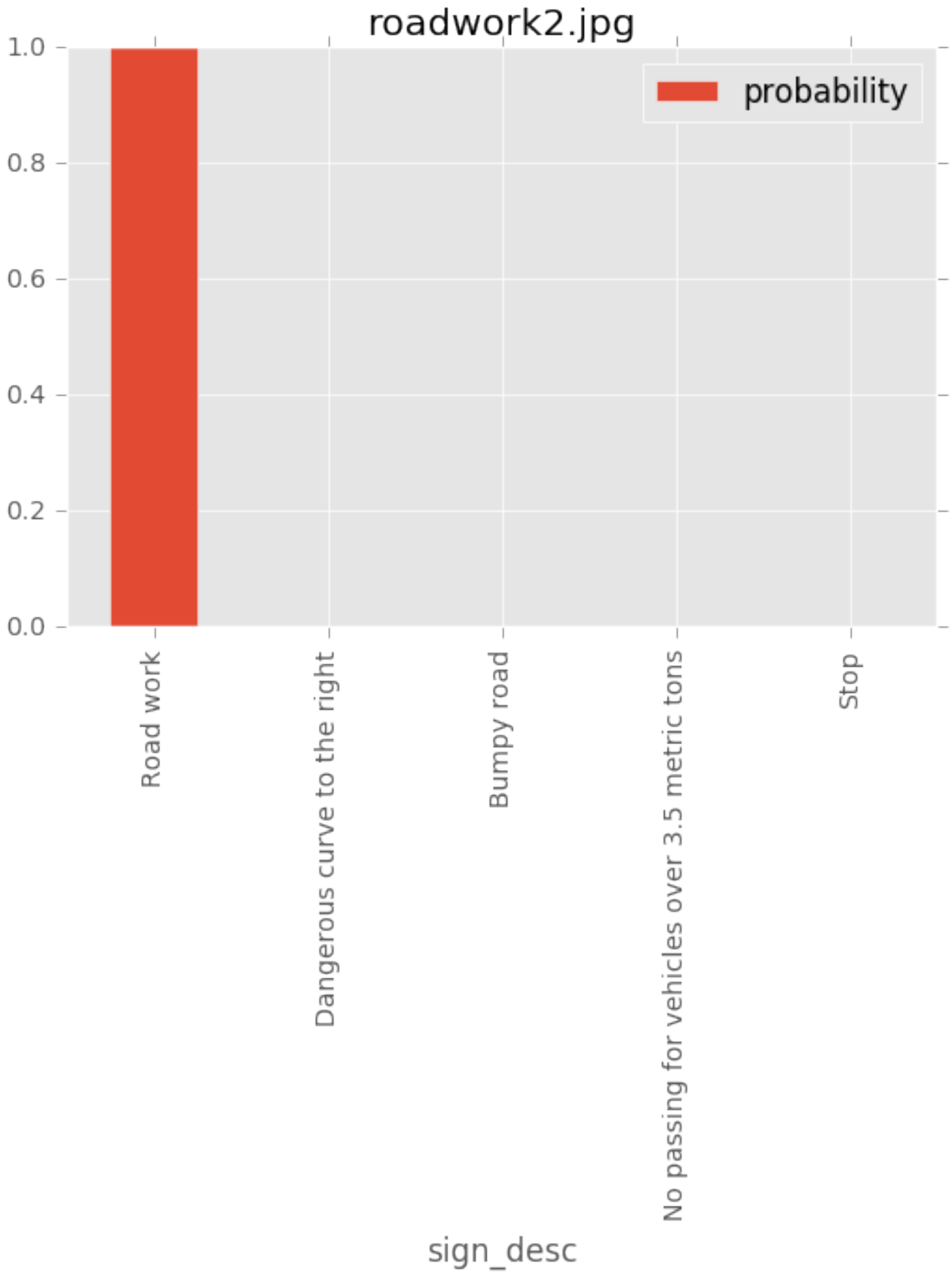

Do-Not-Enter.jpg
probability
1.0
0.8
0.6
0.4
0.2
0.0
No entry
Stop
Speed limit (20km/h)
No vehicles
No passing
sign_desc

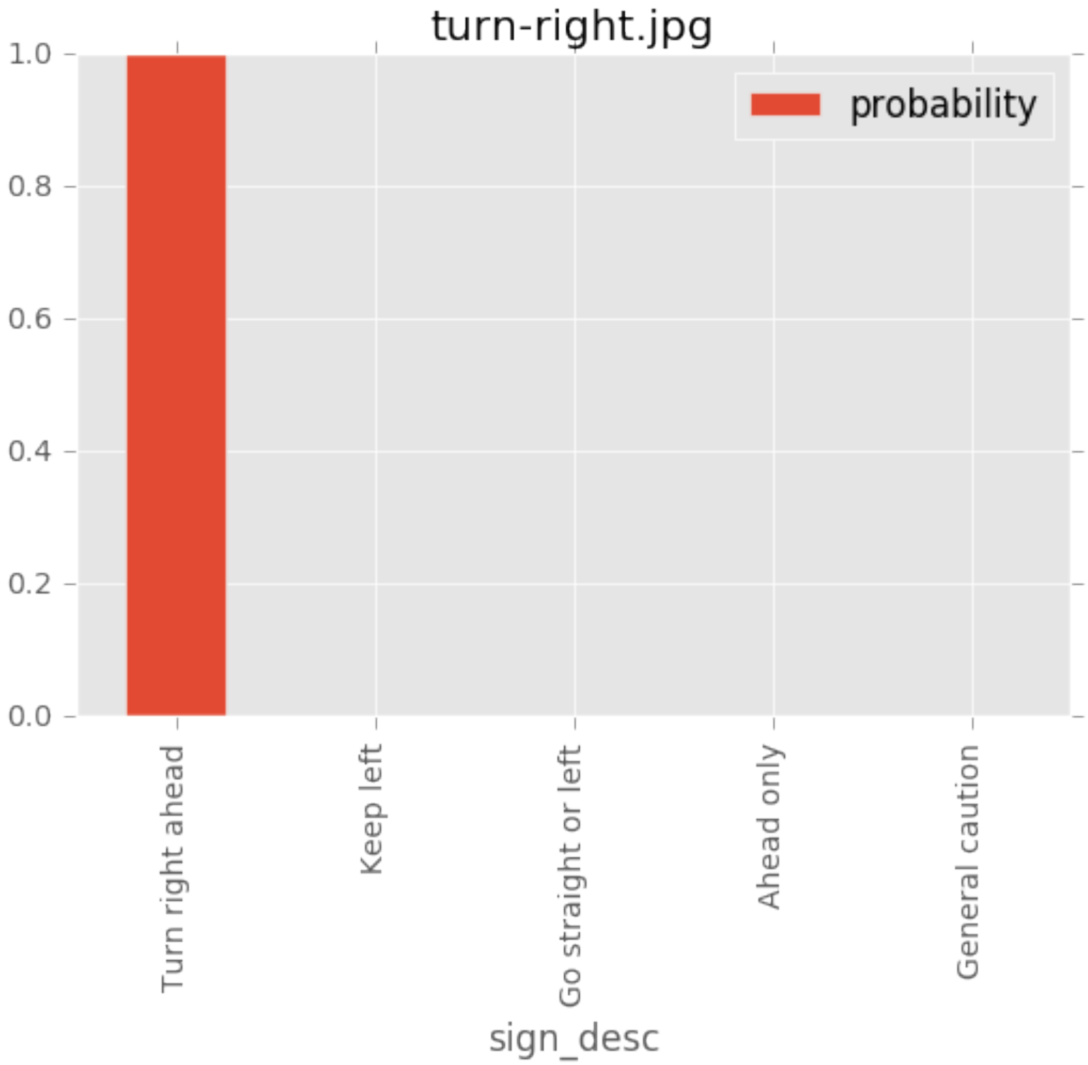


general_caution.jpg
1.0
0.8
0.6
0.4
0.2
0.0
probability
Children crossing
Priority road
General caution
Traffic signals
Slippery road
sign_desc


50_sign.jpg
probability
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Slippery road
Dangerous curve to the left
Children crossing
Ahead only
Dangerous curve to the right
sign_desc


right_of_way.jpg
probability
1.0
0.8
0.6
0.4
0.2
0.0
Right-of-way at the next intersection
Pedestrians
Traffic signals
Beware of ice/snow
Double curve
sign_desc


roadwork2.jpg
probability
1.0
0.8
0.6
0.4
0.2
0.0
Road work
Dangerous curve to the right
Bumpy road
No passing for vehicles over 3.5 metric tons
Stop
sign_desc



The table below has the softmax probabilities for the top 5 likely options for each sign downloaded from the internet.

image_desc	sign_desc	probability
Road Work	Road work	1
Road Work	Dangerous curve to the right	8.385069E-024
Road Work	Bumpy road	1.367577E-024
Road Work	No passing for vehicles over 3.5 metric tons	1.182849E-024
Road Work	Stop	2.093643E-025
Do Not Enter	No entry	1
Do Not Enter	Stop	1.478105E-018
Do Not Enter	Speed limit (20km/h)	3.042417E-022

Do Not Enter	No vehicles	1.33123E-028
Do Not Enter	No passing	3.703475E-032
General Caution	Children crossing	0.9999936
General Caution	Priority road	0.000006494914
General Caution	General caution	0.00000003432028
General Caution	Traffic signals	0.00000000192434
General Caution	Slippery road	0.000000001605059
50 km/h	Slippery road	0.6155943
50 km/h	Dangerous curve to the left	0.3777297
50 km/h	Children crossing	0.006646386
50 km/h	Ahead only	0.00002741164
50 km/h	Dangerous curve to the right	0.000002117236
Right of Way	Right-of-way at the next intersection	1
Right of Way	Pedestrians	9.537188E-024
Right of Way	Traffic signals	2.27464E-032
Right of Way	Beware of ice/snow	1.761734E-032
Right of Way	Double curve	3.451161E-033
Road Work 2	Road work	1
Road Work 2	Stop	2.480531E-018
Road Work 2	Speed limit (20km/h)	0
Road Work 2	Speed limit (30km/h)	0
Road Work 2	Speed limit (50km/h)	0
Turn right	Turn right ahead	1
Turn right	Keep left	2.835873E-017
Turn right	Go straight or left	4.532348E-026
Turn right	Ahead only	4.193974E-027

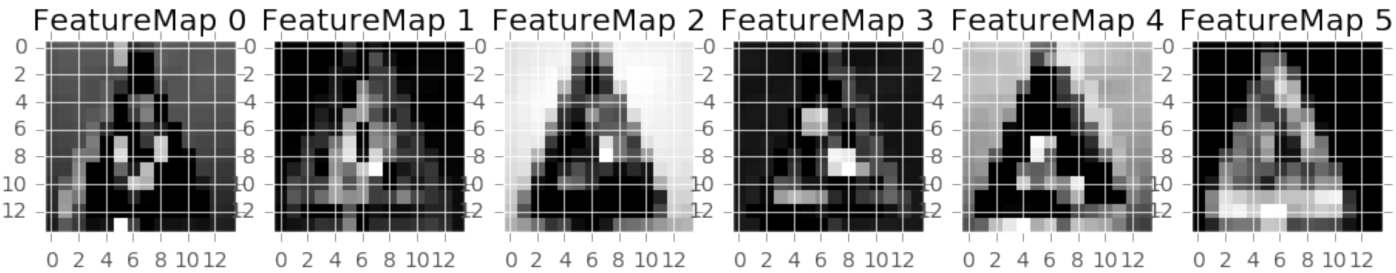
Turn right	General caution	3.570089E-030

(Optional) Visualizing the Neural Network (See Step 4 of the Ipython notebook for more details)

1. Discuss the visual output of your trained network's feature maps.

Looking at the feature map for the 1st convolutional layer the model is using the sign's outline shape to identify which sign it is. While the 2nd convolutional layer is looking for lower level features that relate to the icons inside the sign.

Feature Map from Conv Layer 1



Feature Map from Conv Layer 2