Traffic Sign Recognition

Writeup

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

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.

You're reading it! and 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.

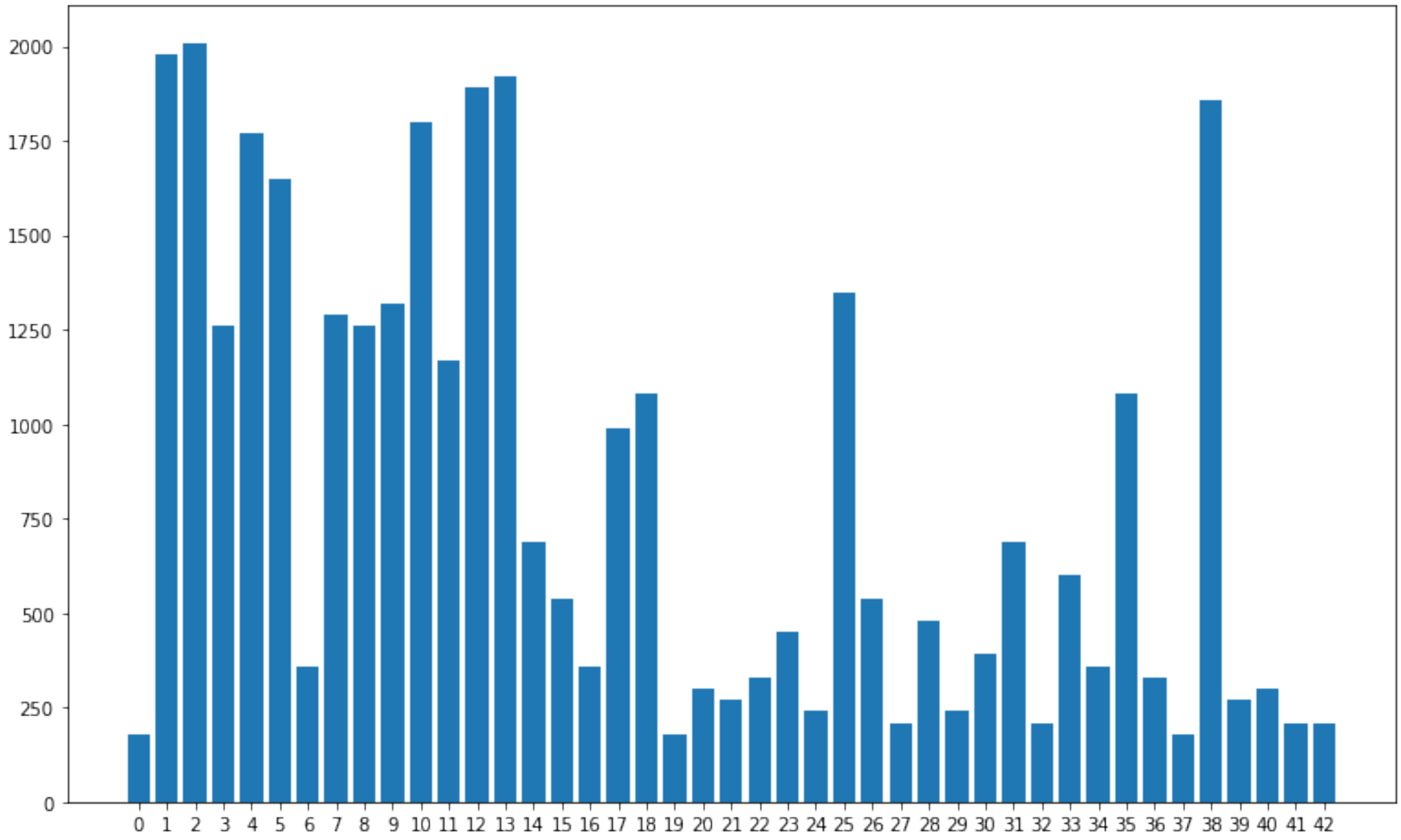
I used the collections to analyze and provide the summary statistics of the data.

signs data set:

- The size of training set is 34799
- The size of the validation set is 4410
- 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. 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 …

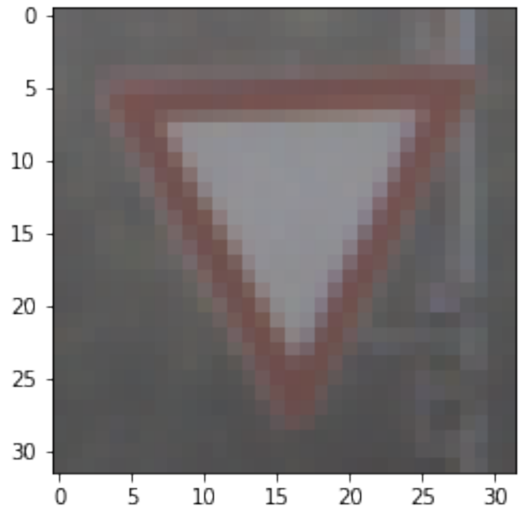
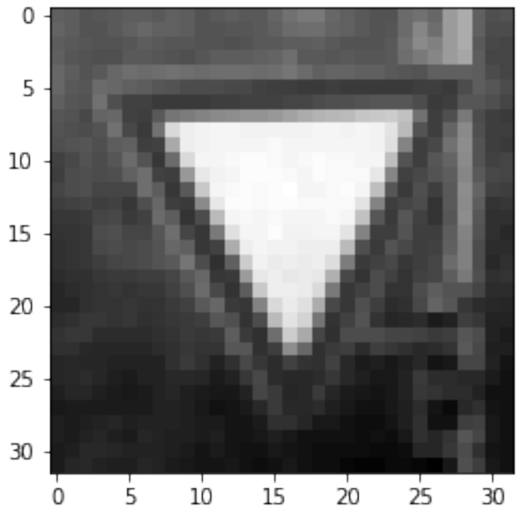


As the result indicates there is significant unbalance between the classes that might cause problem for scaling and generalizing the model. This unbalance could be solved by generating some artificial data out of the original data. Since for this project we could achieve our goal, and due to the lack of time I skipped this part for now and I used the original data set.

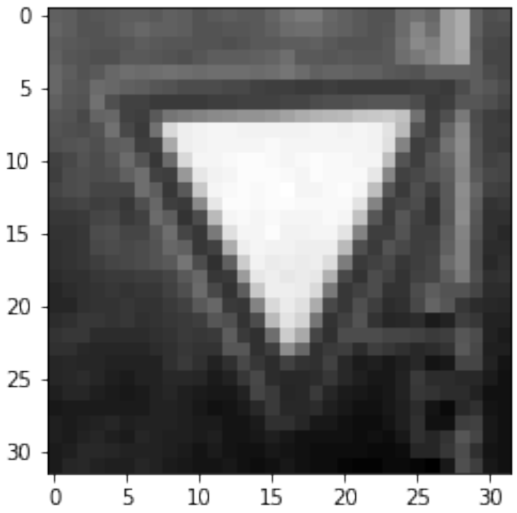
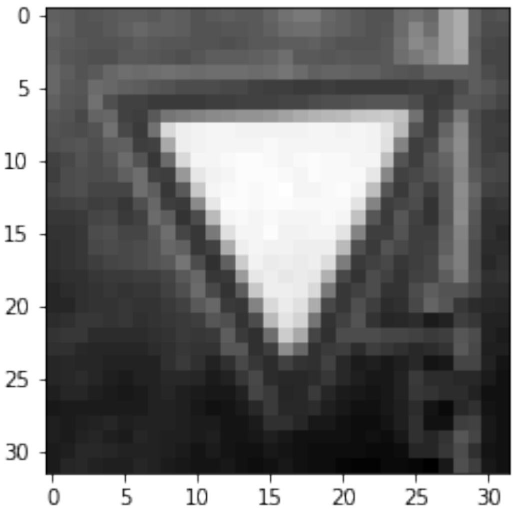
Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data for training, describe why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

First all the image converted to grayscale to reduce the effect of the shadowing and color variation.

Before	After
	

Then all images color values were normalized to reduce the range of the pixel variation and make the optimizer faster.

Grayscaled	Normalized
	

As results indicate, normalizing does not have any effect on the image and only reduced the range of the pixel variation.

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 consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 1	1x1 stride, VALID padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution 2	2x2 stride, VALID padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, output = 5x5x16
Flatten	output = 400
Fully connected	input = 400, output = 120
RELU	
Fully connected	input = 120, output = 84
RELU	
Fully connected	input = 84, output = 10
Softmax	

For training the model I used 60 epochs with the training rate of 0.002, and the batch size of 32.

3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

An Adam optimizer was used in the model, and the model was trained as mini-batches of 32 with the training rate of 0.002 in 60 epochs.

4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the

architecture is suitable for the current problem.

My final model results were:

- training set accuracy of 0.999
- validation set accuracy of 0.947
- test set accuracy of ? 0.932

I used the LeNet architecture for building and training my model, and only by tuning the batch size and learning rate, it could achieve the accuracy goal.

I chose this model because previously I implemented this model for the MNIST data and I knew that it would be beneficial for this project as well.

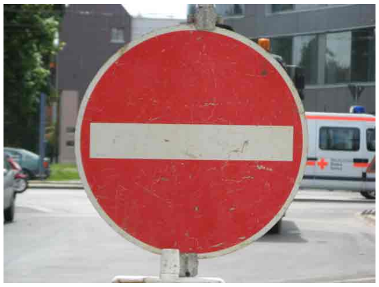
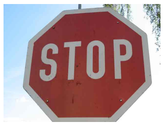
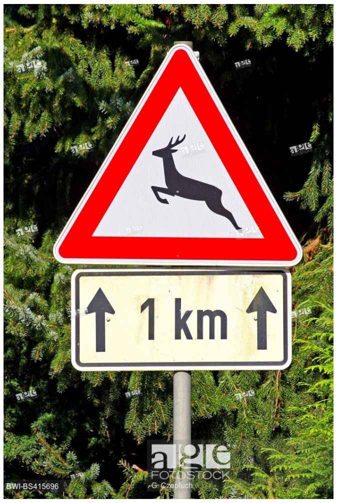
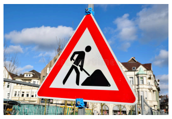
Also, base don’t he results it could be said that there was no significant over-fitting in the model, because all the scores are close to each other. I also, tried to add drop out layers but the accuracy decreased significantly and that’s why I kept the original model.

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

I found these signs from internet:

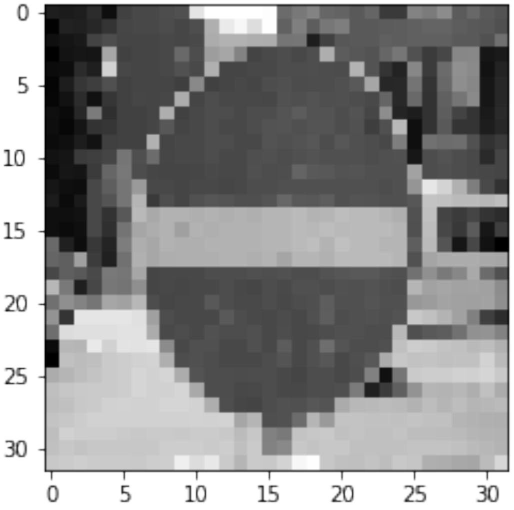
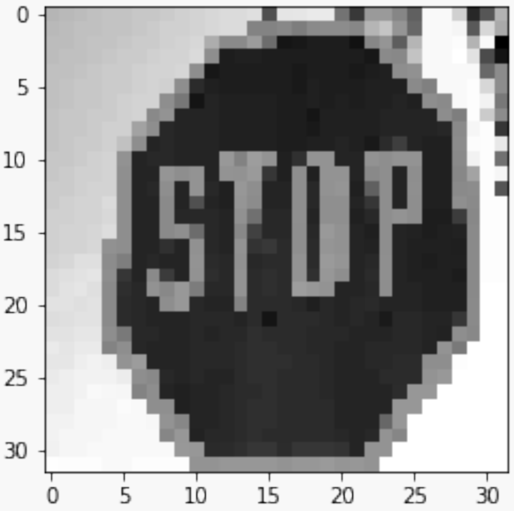
$$1^{st}\ Table : Random\ images$$

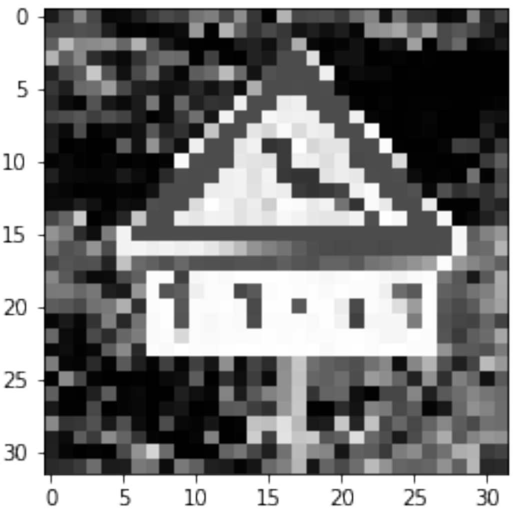
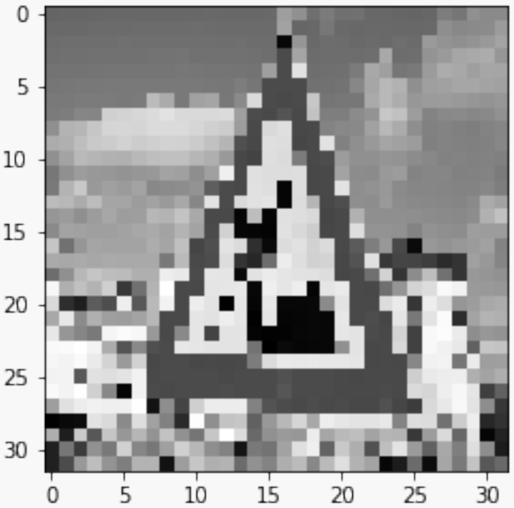
Signs				
Challenges	This image should be classified correctly since it is a big picture with suitable lighting. Only the background might cause a little bit of difficulty for the classifier. Also, it should be noticed that the image is rectangular and changing its size to 32x32 might cause problem and make it unrecognizable.	This image has a little bit of orientation as well as background but the lighting is good and image was taken from a close distance	This sign was taken fro a suitable distance in a good lighting condition but it is not the sign alone and it has an attachment. Also, the background as well as watermarking might cause problem for recognizing the image. Another issue it is the fact that the image is not square	This image has orientation as well as background. Also, it was taken from relatively far distance

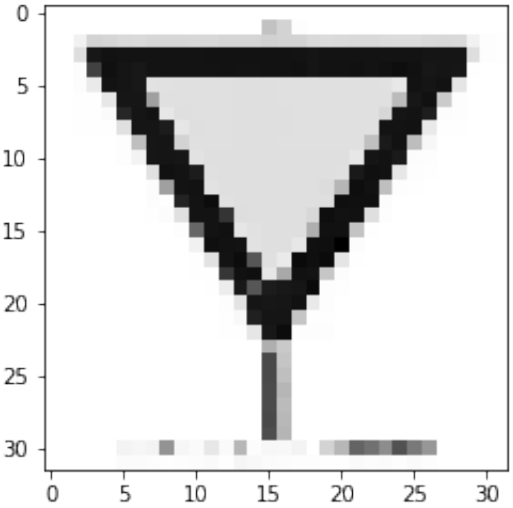
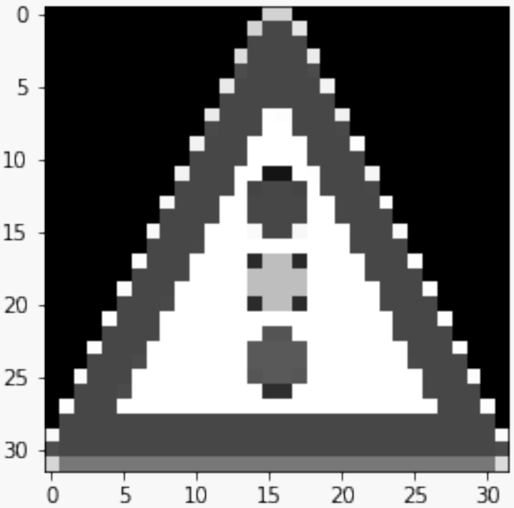
2nd Table : Clean images

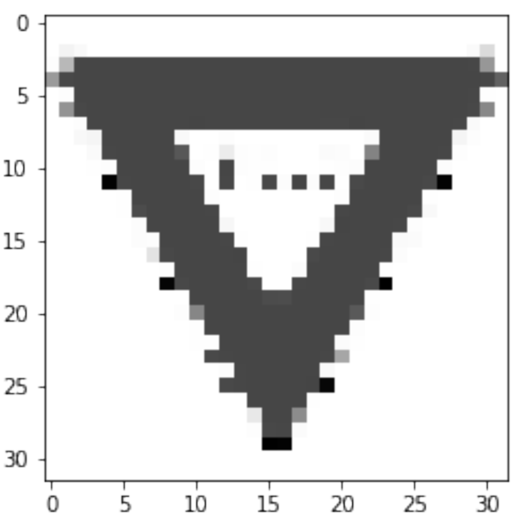
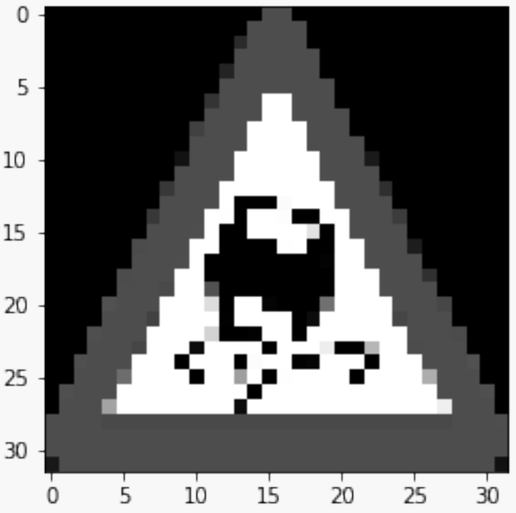
As the above tables show, these images are in different sizes and half of them including background. Also, some of them have orientation in the picture that might affect the prediction results. It should be noticed that all the image in the second table are very clean and suitable for the classifier.

2. Discuss the model’s predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).

Images after pre-processing	Results
No entry  0 5 10 15 20 25 30 0 5 10 15 20 25 30	1. No entry: 100.0% 2. Speed limit (20km/h): 0.0% 3. Speed limit (30km/h): 0.0% 4. Speed limit (50km/h): 0.0% 5. Speed limit (60km/h): 0.0%
Stop  0 5 10 15 20 25 30 0 5 10 15 20 25 30	1. Stop: 100.0% 2. Go straight or right: 4.181570844116354e-13% 3. Road work: 1.467370364019094e-14% 4. Yield: 1.3145126487219661e-17% 5. Turn right ahead: 2.718254110225337e-18%

Wild animals crossing  0 5 10 15 20 25 30 0 5 10 15 20 25 30	1. Road narrows on the right: 93.84663105010986% 2. Traffic signals: 6.153365224599838% 3. Road work: 5.414727555219656e-08% 4. Pedestrians: 3.373701566685664e-19% 5. Dangerous curve to the left: 2.2217763054870752e-30%
Road work  0 5 10 15 20 25 30 0 5 10 15 20 25 30	1. Bumpy road: 94.01511549949646% 2. Road work: 5.984881520271301% 3. Dangerous curve to the right: 6.325669957797482e-27% 4. Stop: 1.314043475264442e-28% 5. Slippery road: 6.029408630734881e-29%
Yield	1. Yield: 100.0% Speed limit (20km/h): 0.0% Speed limit (30km/h): 0.0% Speed limit (50km/h): 0.0% Speed limit (60km/h): 0.0%

	
Traffic signals 	1. Traffic signals: 100.0% 2. General caution: 1.2879885057764365e-14% 3. Speed limit (20km/h): 0.0% 4. Speed limit (30km/h): 0.0% 5. Speed limit (50km/h): 0.0%
Yield	1. Yield: 100.0% 2. Speed limit (20km/h): 0.0% 3. Speed limit (30km/h): 0.0% 4. Speed limit (50km/h): 0.0% 5. Speed limit (60km/h): 0.0%

 0 5 10 15 20 25 30 0 5 10 15 20 25 30	
Slippery road  0 5 10 15 20 25 30 0 5 10 15 20 25 30	1. Slippery road: 100.0% 2. Speed limit (20km/h): 0.0% 3. Speed limit (30km/h): 0.0% 4. Speed limit (50km/h): 0.0% 5. Speed limit (60km/h): 0.0%

As the results indicates, 6 out of 8 images (75%) were predicted correctly. Although the model could not predict the "Wild animal crossing" and "Road work", it can not be told that the model has a low recall score for these two signs, since one of them was rectangular and was deformed after transformation, and the sign in other one has orientation. The results indicate that the image with no background and closer images can be predicted with higher accuracy.