Self-Driving Car Project #2

Traffic Sign Recognition Classifier

Complete on Feb 23rd, 2017

Haowei Zhang

Part 1 Data Set Summary & Exploration

Task 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.

Response:
I used *shape* attribute of numpy.ndarray to find out the following statistics of the dataset.

(1) Number of training examples: 34799

(2) Number of validation examples: 4410

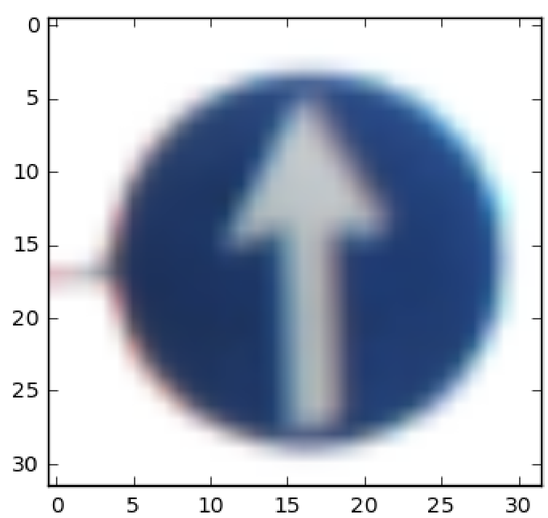
(3) Number of testing examples: 12630

(4) Image data shape: (32, 32, 3)

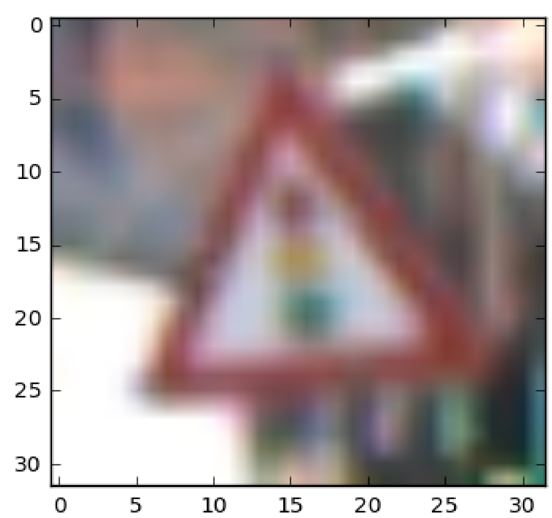
(5) Number of unique classes/labels in the data set: 43

Task 2:
Include an exploratory visualization of the dataset.

Response:
I use matplotlib to randomly plot 2 images for visualization. All images in the dataset are 3-channeled RGB images with a shape of (32, 32, 3).



(a) Visualization Image No.1



(b) Visualization Image No.2

Figure 1: Visualization of the training dataset

Part 2 Design and Test a Model Architecture

Task 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 pre-processing 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.)

Response:
Following are techniques I used to pre-process training dataset images.

(1) Convert images in dataset from 3-channeled RGB images to single-channel greyscale images. The conversion formula is as follows.

$$Greyscale\_image = R\_image \times 0.299 + G\_image \times 0.587 + B\_image \times 0.114 \tag{1}$$

(2) Next I normalize all images after converting to greyscale so the images would have zero-mean and a standard deviation of 1. This brings all images in the same dataset to the same signal level. The normalization technique is described as follows.

$$Image\_NORM = \frac{(Greyscale\_image - \mu)}{\sigma} \tag{2}$$

where μ and σ are mean and standard deviation of the greyscale image

Task 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.

Response:
Following are my model architectures.

(1) Layer: Input
Description: 32 × 32 × 3 RGB images

(2) Layer: Convolutional Layer
Description: takes input of 32 × 32 × 1 RGB images; uses 5 × 5 × 1 kernel with depth of 6; produces an output of shape 28 × 28 × 6 using RELU activation function

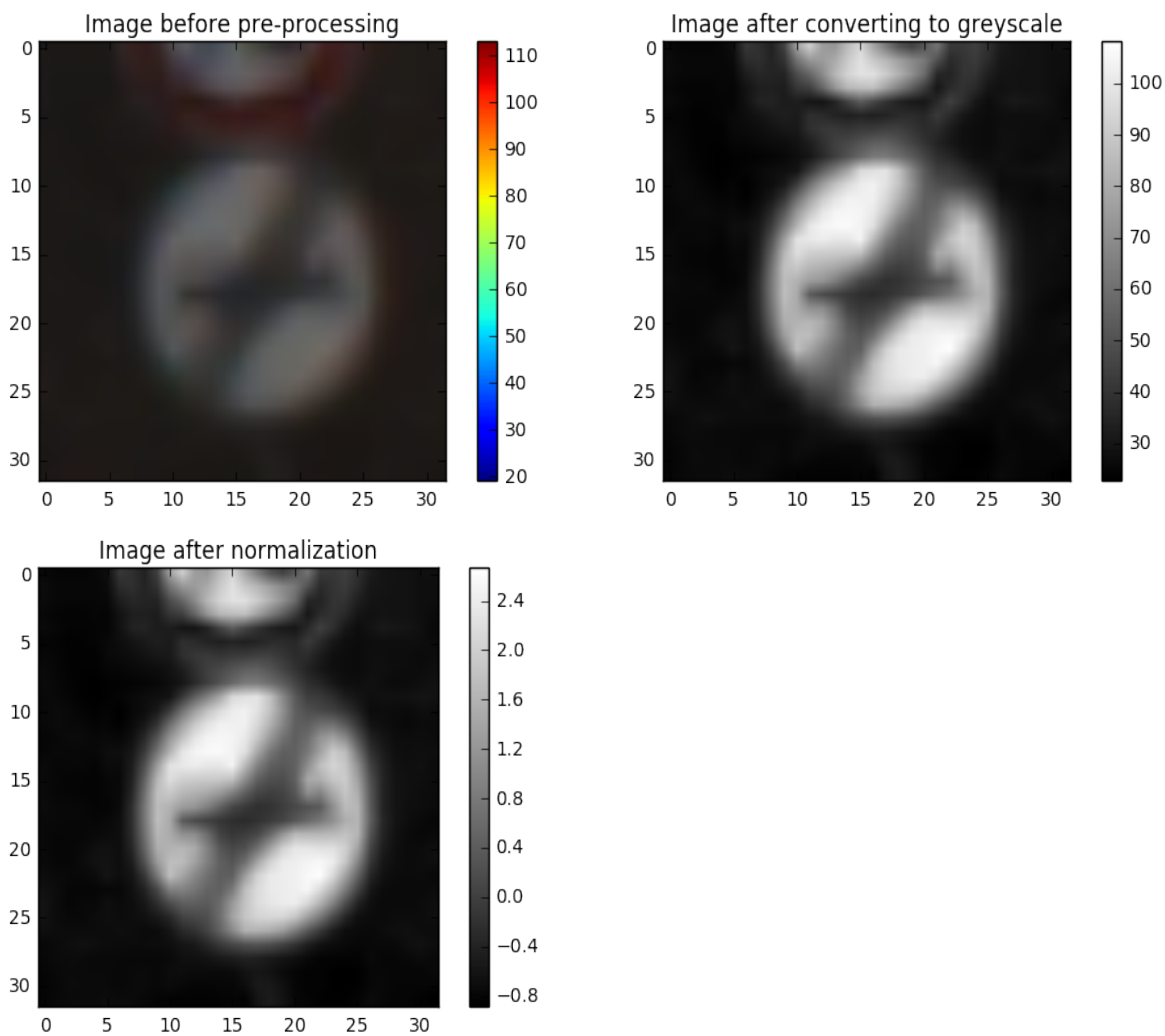


Figure 2: Visualization of the pre-processed dataset

(3) Layer: Max Pooling Layer
Description: takes input of shape 28 × 28 × 6; max pool with a kernel of shape 2 × 2 × 1; produces an output of shape 14 × 14 × 6

(4) Layer: Convolutional Layer
Description: takes input of shape 14 × 14 × 6; uses 5 × 5 × 6 kernel with depth of 16; produces an output of shape 10 × 10 × 16 using RELU activation function

(5) Layer: Max Pooling Layer
Description: takes input of shape 10 × 10 × 16; max pool with a kernel of shape 2 × 2 × 1; produces an output of shape 5 × 5 × 16

(6) Layer: Fully Connected Layer
Description: takes input of shape 5 × 5 × 16; flatten the input to a 1D array of size

400; produces an output of a 1D array of size 120 using RELU activation function

(7) Layer: Fully Connected Layer
Description: takes input of a 1D array of size 120; produces an output of a 1D array of size 43 using RELU activation function

Task 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.

Response:
To train the model, I use the Adam Optimizer with learning rate 0.001 together with a batch size of 128. The total training is finished within 10 epoches.

Task 3:
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.

Response:
After training for 10 epoches, the training, validation and test accuracies are 99.6%, 93.5% and 91.2% respectively. I tried to use LeNet at the very beginning to train and I changed the final fully connected layer in LeNet model to work with this data set. The validation accuracy is over 93% when I trained the network for the first time so I didn't have to tune any of the hyperparameters to make it work. I chose to use LeNet because in the previous lessons I have learnt that LeNet is a well-performed CNN model to classify images.

Part 3 Test a Model on New Images

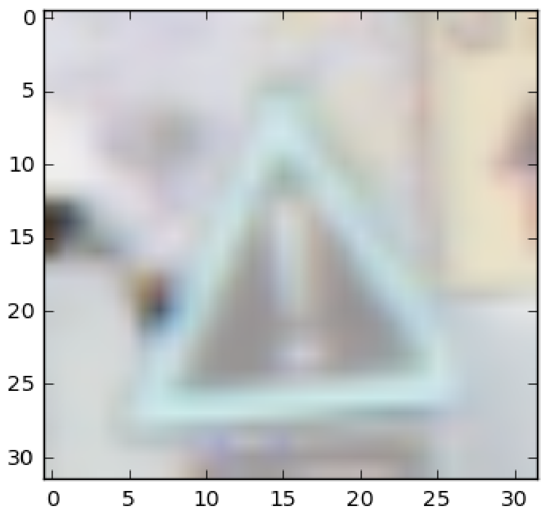
Task 1:
Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

Response:
Following are 5 images I selected from German traffice sign dataset.

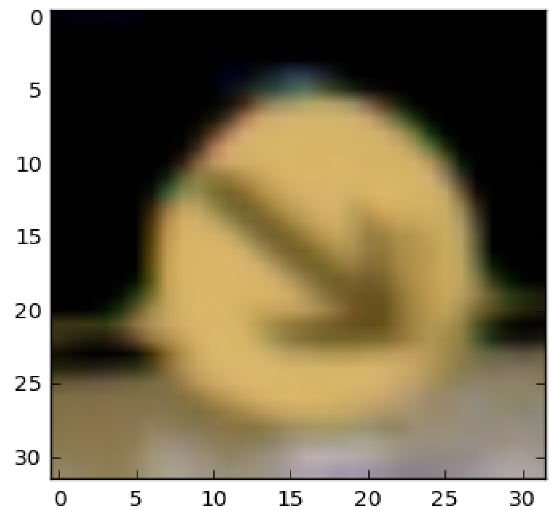
Since the images selected have width and height of 32 pixels, the resolution of images could be a problem for the classifier to do a correct prediction.
Task 2:
Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what 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).

Response:
The predictions to 5 images are $General caution$, $Keep right$, $Keep right$, $Speed limit (70 km per h)$, and $General caution$. The model predicted 5 out of 5 correct.

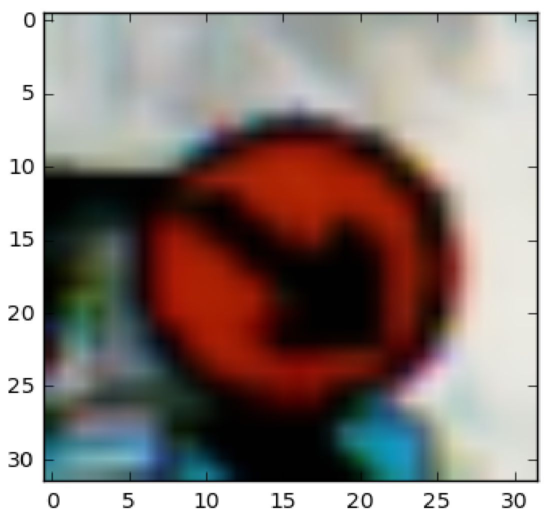
Task 3:
Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)
Response:
The model is really certain of its prediction and the probability of the model assigning to the correct prediction is always greater than 99%. In this sense, I would not print the probabilities. At least this means the model is really robust and does not get confused.



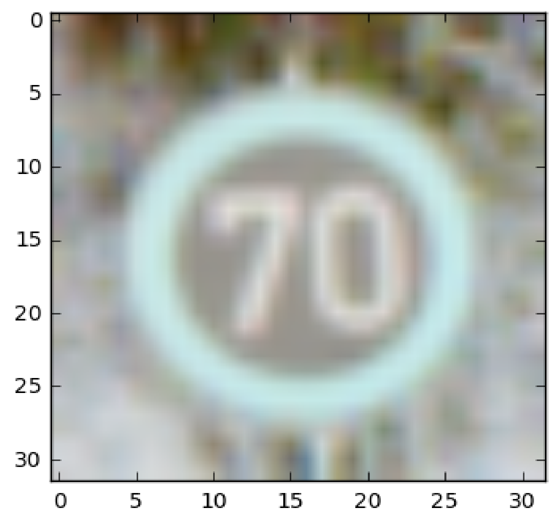
(a) Image No.1



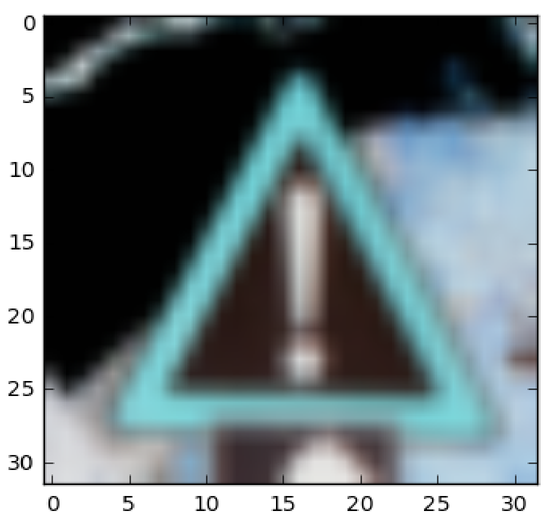
(b) Image No.2



(c) Image No.3



(d) Image No.4



(e) Image No.5

Figure 3: Visualization of the pre-processed dataset