Traffic Sign Classification with Convolutional Neural Networks by TensorFlow Project Report

Zhaolin Zhang

SWE 645 Course Project

George Mason University

1. Introduction

The traffic signs classification problem is a classic problem in AI, computer vision, and machine learning field. Nowadays, more and more applications require a high accuracy result of traffic signs recognition and classification. As such, choosing traffic signs classification problem as a project could be an interesting topic for following up the most important and advanced technology.

2. Data Preparation

As in requirement, I use the German Traffic Sign Recognition Benchmark ("GTSRB - German Traffic Sign Recognition Benchmark.") from Kaggle as my dataset. It contains 39,209 color images which include one traffic sign per image for training, and 12,630 images for testing. There are 43 classes need to be classified in total.

For un-processed images, they are significantly differed in contrast and brightness, as well as the traffic sign's location. Therefore, in order to get a better result, I preprocessed all data images. First, I cropped the traffic signs from images only, and resized all images to 32*32 pixels. Then, according to Sermanet and Yann, using color channels didn't help a lot in improving accuracy (2011), so I turned all image into gray. Next, I sequentially smoothed by Gaussian filter, histogram equalized, and normalized the images.

Also, because the test data and the real word data image might be more shifted, scaled, or rotated, I processed the extend data images by randomly perturbing the preprocessed image in position (by [-2,2] pixels), in scale (by [.9,1.1] ratio) and rotation (by [-10,+10] degrees). Because each image would randomly generate 3 variations, the extend dataset has 117,627 images.

3. Model Architecture

According to Sermanet and Yann, #26 model -- 2 layers Convolutional Neural Network with 38 features and 64 features, theoretically could reach 98.89% accuracy. Therefore, I mocked that model. However, it has some differences. I add one more layer at the very beginning, and use a traditional feed-forward Convolutional Neural Network. Therefore, the model architecture is: 32*32 image as input; first convolutional layer with 16 filters and 5*5 kernel; first max pool with 2*2 pool size and 2*2 strides; a dropout layer with a keep rate 0.9; second convolutional layer with 32 filters and 5*5 kernel; second max pool with 2*2 pool size and 2*2 strides; a dropout layer with a keep rate 0.8; third convolutional layer with 64 filters and 5*5 kernel; third max pool with 2*2 pool size and 2*2 strides; a dropout layer with a keep rate 0.7; reshape it to fully connected layer with size 6*6*64; and dense it to output layer with size 43.

4. Result

The training basically contains 2 stages.

First, the model just trains the original non-extend data images (39,209 images). With the batch size is 500 images a time, the accuracy could reach 97% within 10 epochs, which is a really great result. Actually, for training dataset, the model has already reach 99%. So, we can see there is some features that the non-extend dataset doesn't cover. So, I add the extend image data in second stage to see what will happen.

Next, the model adds the extend data set in. So, the total training dataset size is 156,836 images. After 10 epochs, the accuracy can reach nearly 97%. That is a not bad result. Compared

with the non-extend training data, the mixed data seems like covers more features in and the accuracy difference between training set and test set is getting smaller. Also, we can clear see the loss is lower than it from non-extend set, which meaning the traffic sign variation is really happening in real world.

5. Final Discussion

In this project, a 3 layers Convolutional Neural Network has been implemented by TensorFlow with pre-processed data image by OpenCV from German Traffic Sign Recognition Benchmark. The model is simplified from Sermanet and Yann's model #26 (2011). Eventually the model reaches a test accuracy above 97%. The most difficult part for this project is, the output classes have more than 40 and it is hard to find a perfect fit model. Luckily, with the pre-processed data, the 3 layers Convolutional Neural Network has done well on this problem. For further improvement, on one hand, generating more training images which could largely mock test images might improve the accuracy from the source; on other hand, trying other model with a larger filter on convolutional layer might helpful from the model improvement.

Reference

"GTSRB - German Traffic Sign Recognition Benchmark." *Kaggle*, kaggle.com, December 2018, https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

Sermanet, Pierre, and Yann LeCun. "Traffic sign recognition with multi-scale Convolutional Networks." *IJCNN*. 2011.