**Traffic Sign Classification**

--- Stage 4

4.13.2020

BY

Chenxu Zhuang

Lei Mu

Yangjian Xiao

Table of Contents

[1. Motivation 3](#_Toc37694351)

[2. Problem definitions 3](#_Toc37694352)

[3. Dataset introduction 4](#_Toc37694353)

[4. Sample data visualization 7](#_Toc37694354)

[5. Data preparation 8](#_Toc37694355)

[6. Approach selected: a convolution neural network framework 8](#_Toc37694356)

[7. Reference 11](#_Toc37694357)

# Motivation

Image recognition is of great significance and has already been utilized in different areas. For example, the traffic sign classification is one of the most significant skills, which must be considered in designing self-driving cars. Such ability is the first step to make a right decision during the driving, such as turning right, stop, speeding up, etc. Therefore, developing a reliable framework to handle real-time traffic sign detection and recognition is a real traffic situation is needed.

Traditional computer vision and machine learning based methods were widely used for traffic signs classification, but those methods were soon replaced by deep learning-based classifiers [3]. Recently deep convolutional networks have surpassed traditional learning methods in traffic signs classification. Classification of traffic signs is not so simple task, images are affected to adverse variation due to illumination, orientation, the speed variation of vehicles etc.

# Problem definitions

In real situations, a self-driving car will encounter a number of traffic signs with different background colors and symbols. Making the right classification for such photos and extracting the accurate information ensure the safety during self-driving. A dataset is available online, including 43 typical types of traffic signs, such as speed limit, animal crossing, stop sign, etc. A photo of these signs is shown in Figure 1.

In our project, a deep learning-based algorithm will be developed to make traffic sign classification. A total number of 43 traffic sign labels will be predicted with a convolutional neural network.



Figure 1 Various traffic signs in the project

# Dataset introduction

The dataset contains 43 different classes of images.

|  |  |
| --- | --- |
| ( 0, b'Speed limit (20km/h)')  ( 1, b'Speed limit (30km/h)')  ( 2, b'Speed limit (50km/h)')  ( 3, b'Speed limit (60km/h)')  ( 4, b'Speed limit (70km/h)')  ( 5, b'Speed limit (80km/h)')  ( 6, b'End of speed limit (80km/h)')  ( 7, b'Speed limit (100km/h)')  ( 8, b'Speed limit (120km/h)')  ( 9, b'No passing')  (10, b'No passing for vehicles over 3.5 metric tons')  (11, b'Right-of-way at the next intersection')  (12, b'Priority road')  (13, b'Yield')  (14, b'Stop')  (15, b'No vehicles')  (16, b'Vehicles over 3.5 metric tons prohibited')  (17, b'No entry')  (18, b'General caution')  (19, b'Dangerous curve to the left')  (20, b'Dangerous curve to the right')  (21, b'Double curve')  (22, b'Bumpy road') | (23, b'Slippery road')  (24, b'Road narrows on the right')  (25, b'Road work')  (26, b'Traffic signals')  (27, b'Pedestrians')  (28, b'Children crossing')  (29, b'Bicycles crossing')  (30, b'Beware of ice/snow')  (31, b'Wild animals crossing')  (32, b'End of all speed and passing limits')  (33, b'Turn right ahead')  (34, b'Turn left ahead')  (35, b'Ahead only')  (36, b'Go straight or right')  (37, b'Go straight or left')  (38, b'Keep right')  (39, b'Keep left')  (40, b'Roundabout mandatory')  (41, b'End of no passing')  (42, b'End of no passing by vehicles over 3.5 metric tons') |

Three groups of datasets are available, including train, valid, and test datasets.

Table 1 Summary of datasets

|  |  |  |
| --- | --- | --- |
| Datasets Name | Categories | Shape |
| Train dataset | X\_train | (34799, 32, 32, 3) |
| y\_train | (34799, ) |
| Validation dataset | X\_validation | (4410, 32, 32, 3) |
| y\_validation | (4410, ) |
| Test dataset | X\_test | (12630, 32, 32, 3) |
| y\_test | (12630, ) |



Figure 2 Statistic result of train dataset



Figure 3 Statistic result of test dataset



Figure 4 Statistic result of validation dataset

According to Table 1, the number of instances for train, test, and validation datasets are 34799, 4410, and 12630, respectively. Each instance is a 32\*32\*3 image.

As shown in Figures 2 to 4, it is obvious that each dataset contains all 43 types of images. The number of each label can be observed from the bar plot above.

# Sample data visualization

A picture containing different, old

Description automatically generated

Figure 5 Image samples with label on the top

# Data preparation

Before training model, the dataset must be pre-processed for future use. The original image is a color 32\*32\*3 image, which means it is a 32\*32 with 3 layers (RGB). The first step is to transform all images to a gray scale to 32\*32\*1. Then a typical standardized process is applied to data. Such procedure can make the size of each figure smaller. In Figure 6, one traffic sign image with its gray color, original color, gray color with normalization is shown. The image of gray color with normalization will be used in the training process, which includes enough information as compared with original image.



Figure 6 Transformation of image to a gray scale

# Approach selected: a convolution neural network framework

As inspired by the good performance of convolutional neural network in recognition of handwritten characters, such approach is utilized to solve traffic sign recognitions with more labels considered. In this project, a total number of 43 different signal signs will be classified with this convolutional neural network, as show in Figure 6. The overall idea for this approach contains 2 convolutions, 2 subsampling, and 2 full connections. The convolution is used to extract feature information form the original image. The subsampling process makes the image smaller, which can improve the computational efficiency significantly. The full connection is utilized to make final predictions.

A screenshot of a cell phone

Description automatically generated

Figure 7 A convolutional neural network framework selected

The abovementioned approach is implemented with keras and tensorboard libraries, including 6 steps. In step 1, each input is a 32\*32\*1 image and the output is 28\*28\*6, mapping the original image to 6 layers. In step, only half of feature is selected for next convolution.

Corresponding codes:

|  |
| --- |
| **STEP 1: THE FIRST CONVOLUTIONAL LAYER #1**   * + Input = 32x32x1   + Output = 28x28x6   + Output = (Input-filter+1)/Stride\* => (32-5+1)/1=28   + Used a 5x5 Filter with input depth of 3 and output depth of 6   + Apply a RELU Activation function to the output   + pooling for input, Input = 28x28x6 and Output = 14x14x6   **STEP 2: THE SECOND CONVOLUTIONAL LAYER #2**   * + Input = 14x14x6   + Output = 10x10x16   + Layer 2: Convolutional layer with Output = 10x10x16   + Output = (Input-filter+1)/strides => 10 = 14-5+1/1   + Apply a RELU Activation function to the output   + Pooling with Input = 10x10x16 and Output = 5x5x16   **STEP 3: FLATTENING THE NETWORK**   * + Flatten the network with Input = 5x5x16 and Output = 400   **STEP 4: FULLY CONNECTED LAYER**   * + Layer 3: Fully Connected layer with Input = 400 and Output = 120   + Apply a RELU Activation function to the output   **STEP 5: ANOTHER FULLY CONNECTED LAYER**   * + Layer 4: Fully Connected Layer with Input = 120 and Output = 84   + Apply a RELU Activation function to the output   **STEP 6: FULLY CONNECTED LAYER**   * + Layer 5: Fully Connected layer with Input = 84 and Output = 43 |

A preliminary model training is conducted with the approached selected, the whole training dataset is used. In each iteration, the batch size is set as 500 instances. As we can see, after 10 iterations, the prediction accuracy has already exceeded 0.9. For validation data set, the accuracy is higher than 0.8. In Figure 9, both training and validation accuracy is increasing as the increase of epochs.

A screenshot of text

Description automatically generated

Figure 8 Running results of training



Figure 9 Training and validation accuracy

# Reference

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[2] CireşAn, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural networks*, *32*, 333-338.

[3] Haloi, M. (2015). Traffic sign classification using deep inception based convolutional networks. *arXiv preprint arXiv:1511.02992*.

[4] Saadna, Y., & Behloul, A. (2017). An overview of traffic sign detection and classification methods. *International journal of multimedia information retrieval*, *6*(3), 193-210.

[5] Wang, G., Ren, G., Wu, Z., Zhao, Y., & Jiang, L. (2013, August). A hierarchical method for traffic sign classification with support vector machines. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-6). IEEE.