Traffic Sign Classification

Chenxu Zhuang[1], Lei Mu[2], Yangjian Xiao[1]

[1]*Dept. of Civil Engineering, New Mexico State Univ., Las Cruces, NM 88003*
[2]*Dept. of Mechanical and Aerospace Engineering, New Mexico State Univ., Las Cruces, NM 88003.*

Motivation

Traffic signs are an integral part of our road infrastructure. They provide critical information, sometimes compelling recommendations, for road users, which in turn requires them to adjust their driving behavior to make sure they adhere with whatever road regulation currently enforced. Without such useful signs, we would most likely be faced with more accidents, as drivers would not be given critical feedback on how fast they could safely go, or informed about road works, sharp turn, or school crossings ahead. In our modern age, around 1.3M people die on roads each year. This number would be much higher without our road signs.

To save people's life in vehicles, many strategies have been applied in the automobile industry, among which advanced driver assistance system and autonomous driving system are the most popular ones. For example, automobile industry has introduced vision systems in their high-end cars. Examples are the BMW 7er, Mercedes S-Class, Audi A8, Opel Insignia and VW Phaeton [1]. To achieve these systems, the so-called computer vision has been considered as a necessary gradient. Among other tasks solved with computer vision, the traffic sign recognition (TSR) problem is one of the most well-known and widely discussed by lots of researchers. However, the main problems of such systems are low detection accuracy and high demand for hardware computational performance, as well as the inability of some systems classify the traffic signs from different countries [2].

Traditionally, standard computer vision methods, such as artificial neural networks (ANNs) [3] and support vector machines (SVMs) [4], were employed to detect and classify traffic signs. Other methods have been reported in [5], [6], and [7]. However, these traditional methods require considerable and time-consuming manual work to handcraft important features in images. Recently deep convolutional networks have surpassed traditional learning methods in traffic signs classification. With the rapid advances of deep learning algorithm structures and feasibility of its high performance implementation with graphical processing units (GPU), it is advantageous to relook the traffic signs classification problems from the efficient deep learning perspective. In the last decade, the deep learning-based classifiers have been increasingly developed in automobile industry. Recently deep convolutional networks have surpassed traditional learning methods in traffic signs classification, as reported in [8-10]. In the current project, we create a deep convolutional network model that reliably classifies traffic signs, learning to identify the most appropriate features for this problem by *itself*.

Problem Definition

Classification of traffic signs is not so simple task, images are affected to adverse variation due to illumination, orientation, color variation the speed variation of vehicles etc. 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

Solution Explanation

While neural networks and other pattern detection methods have been around for the past 50 years, there has been significant development in the area of convolutional neural networks in the recent past**.** The major advantages of using CNN for image recognition include:

(a) Ruggedness to shifts and distortion in the image
Detection using CNN is rugged to distortions such as change in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc. However, CNNs are shift invariant since the same weight

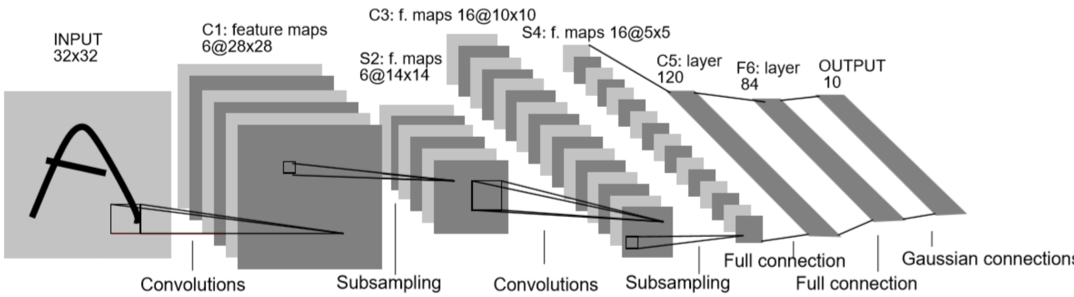


Figure 2 A schematic of convolutional neural network framework

configuration is used across space. In theory, we also can achieve shift invariantness using fully connected layers. But the outcome of training in this case is multiple units with identical weight patterns at different locations of the input. To learn these weight configurations, a large number of training instances would be required to cover the space of possible variations.

(b) Fewer memory requirements

In this same hypothetical case where we use a fully connected layer to extract the features, the input image of size 32x32 and a hidden layer having 1000 features will require an order of 106 coefficients, a huge memory requirement. In the convolutional layer, the same coefficients are used across different locations in the space, so the memory requirement is drastically reduced.

(c) Easier and better training

Again using the standard neural network that would be equivalent to a CNN, because the number of parameters would be much higher, the training time would also increase proportionately. In a CNN, since the number of parameters is drastically reduced, training time is proportionately reduced. Also, assuming perfect training, we can design a standard neural network whose performance would be same as a CNN. But in practical training, a standard neural network equivalent to CNN would have more parameters, which would lead to more noise addition during the training process. Hence, the performance of a standard neural network equivalent to a CNN will always be poorer.

On the basis of above advantages, the CNNs method becomes our prior option. As inspired by the good performance of convolutional neural network in recognition of image recognition and 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 3. The overall idea for this CNNs architecture contains 2 convolutions, 2 subsampling, and 2 full connections. The convolution is used to extract feature information from 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.

Data Description

The dataset collected 51839 pieces of traffic sigh photos, which are composed of 43 different classifications (shown in figure 1). During data preprocessing, the dataset were divided into three groups, namely, train, valid, and test datasets. the details of these datasets are tabulated in table 1. Each instance is a 32*32*3 image.

Table 1 Summary of datasets

Datasets	Categories	Shape
Train	X_train	(34799, 32, 32, 3)
	y_train	(34799,)
Validation	X_validation	(4410, 32, 32, 3)
	y_validation	(4410,)
Test	X_test	(12630, 32, 32, 3)
	y_test	(12630,)

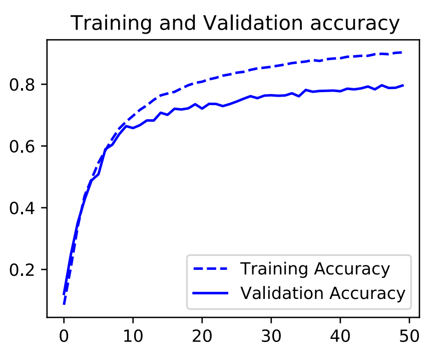
Before training model, the dataset must be pre-processed for future use. The most essential features of traffic signs are the outlines and shapes displayed on them instead of the colors. Meanwhile, this not so important feature would make computation costs increased dramatically. Therefore, we removed the color from the images during dataset preprocessing. 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 3 Transformation of image to a gray scale

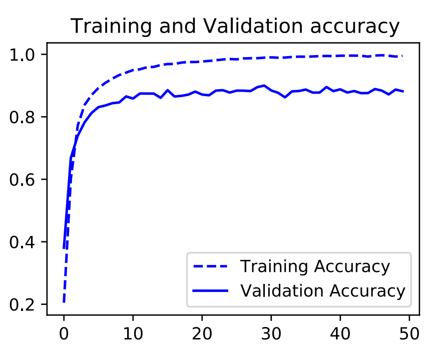
Result Analysis

A preliminary analysis was conducted with the CNN framework in Figure 4. All training dataset in Table 1 is utilized in the training process. The original images are color image with RBG layers. With the consideration of reducing running time, the images are also transformed to grey images with only one layer. As we can see, after 50 iterations, the overall prediction accuracy has already exceeded 0.9. However, the training accuracy is much higher than validation accuracy, which is a typical overfitting issue.

As compared with the accuracy curves for both color and grey images, the running time of grey images is 35 s, which is much faster than that of color images of 48 s. One more interesting finding is that the testing accuracy of grey images (0.8683) is higher than inputting color images (0.7949).



(a) Color images



(b) Grey images

Figure 4 Training and validation accuracy (without dropout layer)

In order to prevent the overfitting issue, the dropout layer was added in the training process. Such approach is a much effective method to prevent overfitting issue in practical deep neural networks. The dropout is considered as an ensemble method, which can increase the generality of CNN.

Then a follow-up training was undertaken. The overfitting issue is solved successfully. As shown in Figure 5, the training and validation accuracies reached to a higher level, especially for the validation accuracy. These two curves seem overlapping each other, which indicates no overfitting as compared with the accuracy in Figure 4. The corresponding training and validation losses are relatively small. The final testing accuracy is 0.9236, which is much higher than the previous one.

In addition, several traffic sign images with both predicted and true labels are shown in Figure 7, which indicates that all 9 images are correctly predicted.

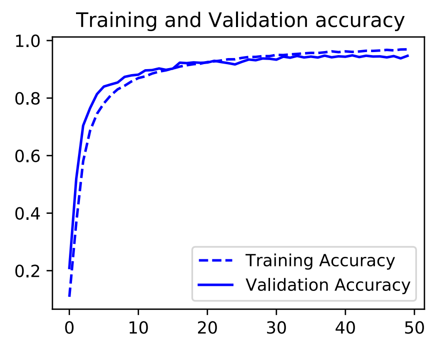


Figure 5 Training and validation accuracy (with dropout layer)

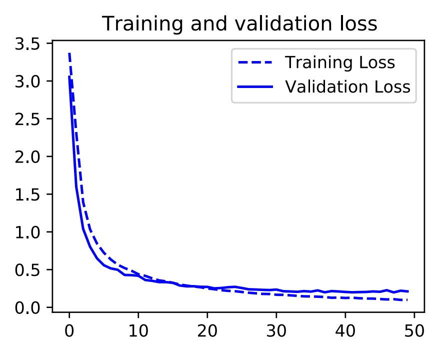


Figure 6 Training and validation loss

Figure 7 Prediction results

Conclusions

In this project, a typical CNN is implemented to recognize 43 types of traffic signs. A lot of work in this project focused on data preprocessing, including image conversion (color to gray), normalization, and tensor transformation. Based on the implemented CNN and preprocessed dataset, 43 types of traffic signs can be successfully predicted with a testing accuracy of 0.9236. Some conclusions are summarized as follows.

- Converting images from color to grey scale is a significant step, which can keep the image features and reduce the dataset size. It is concluded that transformed grey images can achieve a higher accuracy than original color images.
- The overfitting issue can be successfully prevented with the consideration of dropout layer in the training process.
- During the parameter tuning process, the number of output channel in the convolution layer is considered as the most important parameter. However, more output channels lead to higher computational cost.

Reference

[1] Peemen, M., Mesman, B., & Corporaal, H. (2011, May). Speed sign detection and recognition by convolutional neural networks. In Proceedings of the 8th international automotive congress (pp. 162-170). sn.

[2] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. Procedia engineering, 201, 718-725.

[3] Ghica, D., Lu, S. W., & Yuan, X. (1995, November). Recognition of traffic signs by artificial neural network. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 3, pp. 1444-1449). IEEE.

[4] Maldonado-Bascón, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gómez-Moreno, H., & López-Ferreras, F. (2007). Road-sign detection and recognition based on support vector machines. IEEE transactions on intelligent transportation systems, 8(2), 264-278.

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

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

[7] 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.

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

[9] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. Procedia engineering, 201, 718-725.

[10] Mao, X., Hijazi, S., Casas, R., Kaul, P., Kumar, R., & Rowen, C. (2016, June). Hierarchical CNN for traffic sign recognition. In 2016 IEEE Intelligent Vehicles Symposium (IV) (pp. 130-135). IEEE.

Response to comments:

Comment 1: The motivation could be expanded with more information and literature support. (Mu) **Response 1:** A more comprehensive literature study has been performed, which has been incorporated in the current report.
Comment 2: The document does not mention the challenges of doing this project. **Response 2:** Classification of traffic signs is not so simple task, images are affected to adverse variation due to illumination, orientation, color variation the speed variation of vehicles etc.
Comment 3: What methods are usually used to solve this type of problem? **Response 3:** Convolutional neural networks (CNNs) are widely used in pattern- and image-recognition problems as they have a number of advantages compared to other techniques. Therefore, CNNs are utilized in this problem.
Comment 4: The report does not mention if you want to classify all 43 categories or just a subset. **Response 4:** In this project, all of 43 categories (different signal signs) are classified by the composed CNNs.
Comment 5: It is not clear, or well supported, why are you removing the color from the images. **Response 5:** The most essential features of traffic signs are the outlines and shapes displayed on them instead of the colors. Meanwhile, this not so important feature would make computation costs increased dramatically. Therefore, we removed the color from the images during dataset preprocessing.
Comment 6: Don't you think that this color could help in making better predictions? For example, if color red is present on the edge of some figures, whereas color blue is on the edge of others. (Xiao) **Response 6:** Intuitively, the color may improve the prediction accuracy. Thereupon, both situations input with color and input without color were analyzed. The results, however, shows that removing color makes prediction better. One advantage of gray figures, in this specific project, is that their labels are easier to be recognized than colorful figures.
Comment 7: *From your results, it is clear that the NN is overfitting the data.* **Response 7:** the overfitting issue has been solved by adding the dropout layer in the training process.