Temporal Classification Error Compensation of Convolutional Neural Network for Traffic Sign Recognition

Seungjong Yoon, Eungtae Kim

Dept. Electronics Engineering, Korea Polytechnic University, 237, Sangidaehak-ro, Siheung-si, Gyeonggi-do, Korea

Abstract. In this paper, we propose the method that classifies the traffic signs by using Convolutional Neural Network(CNN) and compensates the error rate of CNN using the temporal correlation between adjacent successive frames. Instead of applying a conventional CNN architecture with more layers, Temporal Classification Error Compensation(TCEC) is proposed to improve the error rate in the architecture which has less nodes and layers than a conventional CNN. Experimental results show that the complexity of the proposed method could be reduced by 50% compared with that of the conventional CNN with same layers, and the error rate could be improved by about 3%.

1. Introduction

Recognizing traffic signs is a crucial technology for either drivers or automotive vehicles because it can provide lots of necessary information for drivers. Compared to the conventional location-based traffic sign information, recognition through video frames enables drivers to respond more flexibly by collecting real-time information in fast-changing environments. Recognizing traffic signs can be classified using the Convolutional Neural Network (CNN) [1, 2, 3], which is known to perform well in image data among the Artificial Neural Networks (ANN). Considering the characteristics of ANN, the higher number of nodes and layers can help enhance performance.

Beyond certain point, however, more layers and nodes are required in order to enhance the accuracy rate. As a result, as the number of nodes and layers increases, the computational complexity of CNN methods increases exponentially and the real-time processing becomes difficult. In this paper, we propose the Temporal Classification Error Compensation (TCEC), which can enhance the error rate while restricting the number of nodes and layers. This paper proposes two methods: a) classifying traffic signs using CNN b) compensating the error rate by referencing the adjacent successive frames.

First, in Chapter 2 of this paper, we present the related work about conventional method. In Chapter 3, we look into the CNN architecture with the proposed TCEC, and Chapter 4 will introduce the result of a comparison experiment between the proposed method and the conventional CNN methods. Chapter 5 will present the final conclusion.

2. Related Works

Traffic Sign Classification aims to classify the detected traffic signs into specific classes. Traffic signs are designed to be easily recognizable. For example, colors, shapes, and icons are obvious. These features are also used for learning the networks and classification. Feature extraction algorithms such as Histogram of Oriented Gradients(HOG) and Speed-Up Robust features(SURF) are used to extract the features from the images. For classification, Support Vector Machine (SVM), Random Forest, and

Deep Neural Network (DNN) could be used as the classifier. Particularly, CNN, which is one kind of DNN, is attracting attention because it has good performance in object recognition recently.

The Committee of CNNs [4], which performed best in the IJCNN2011 competition [5], learned HOG and HAAR features through CNN. Among the methods other than CNN, there were Random Forest method [6] and hierarchical SVM method [7]. The CNN method has performed better than the human [8]. However, in the architecture, the number of filters used in the three convolutional layers was 100, 150, and 250, respectively, and 200 neurons were used in the fully connected layer. Because of the large number of the nodes and layer, it had large time complexity. For real-time implementation, more researches are needed to maintain good detection performance while adjusting to have appropriate nodes and layers.

3. Proposed Method

3.1. Classification

In the classification stage, the traffic signs extracted from the frames are classified into 43 classes. Figure 1. shows the CNN model that has two convolution layers. The extracted traffic signs are converted to 28 X 28 size. They are then sent to the C_1 Convolution Layer of Figure 1. C_1 consists of 16 of different 5 X 5 size filters. Each filter is convoluted to the input image, and creates the feature maps same as the number of filters. Then, they go through one sub-sampling and ReLu[9]. ReLu Layer helps to solve several issues that can occur in network learning, and can enhance the learning speed by saving the conventional Sigmoid calculation time.

In order to prevent overfitting problems in the learned network, Dropout Layer [10] is added. Dropout is a method of deleting arbitrary nodes that were used in learning in order to prevent overfitting when the data are propagated while they are learned. When a network is overfitted, it can have good performance for the data set that have been used for training, but it shows very poor performance for the test real data set. Dropout can be carried out in a relatively simple manner, and can reduce the network complexity because the Dropout Layer reduces the number of nodes. The Dropout Layer deletes same number of arbitrarily nodes as the dropout rate. The speed of learning is delayed a little depending on deleting of nodes.

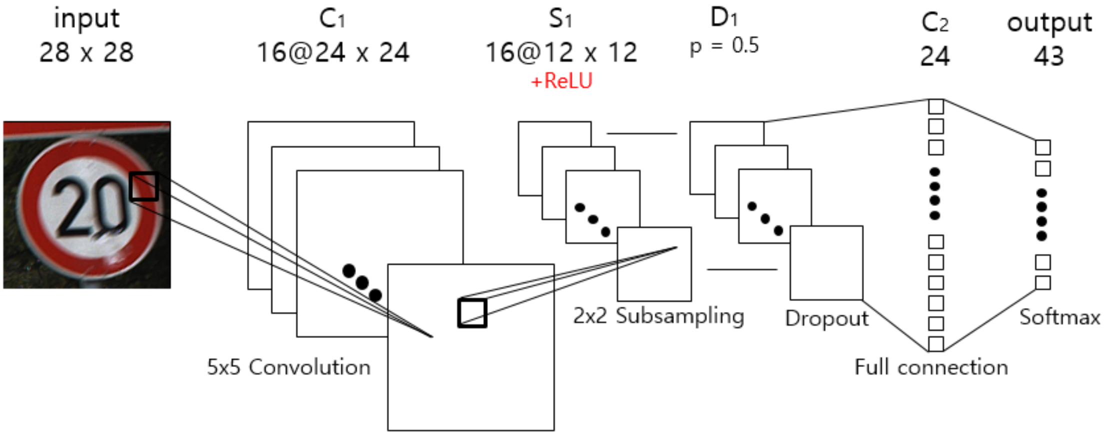

Figure 1. Architecture of the proposed network.

After going through the Dropout Layer, the classification is completed through the Full connection and Softmax stages. As shown in Figure 1, the final output value of CNN equals the number of classes, which is 43. The predicted image class, which has been the input source, has the largest class value.

3.2. Temporal Classification Error Compensation (TCEC)

The error rate could be compensated by making adjustment to the results of current frame, referring to the expected results of the N number of adjacent frames that has processed the classification stage.

Making reference and holding the results of N can be done in two ways: a) referring to the past or future frames only and b) referring to the past and future frames simultaneously.

When a burst error occurs in method a), the error rate rather often increases because the adjustment can be made to the results of current frame with the error rate (that has been failed to be predicted) in the boundary surface of the cluster error. On the other hand, when using method b), the error rate is not high while adjustment can be made to other random errors because of the non-dominant errors in the burst error boundary surface. In other words, TCEC is more effective for local random errors than block errors.

When N number of frames are referred in order to compensate the error rate, the predicted results of CNN class are sequentially saved in the buffer with the size of N. After the buffer is filled with N number of results, the predicted result, placed in the center of buffer as the dominant predicted cost, is sent as feedback of the CNN result in order to adjust to the predicted results. As shown in **Figure 2**, expected results of the five classes are placed sequentially, and after all of the buffers are filled, the most dominant predicted results are used for the adjustment. There are total of two results predicted as Class 11, and total of three results predicted as Class 13. Therefore, the class result is compensated using 13, which is the most dominant result among the five different frames. In case there is no dominant result or it overlaps, the current predicted results are maintained. When applying TCEC, it should be noted that the object within the referred frame needs to be the exact same object. This is because referring to different object can lead to an error propagation.

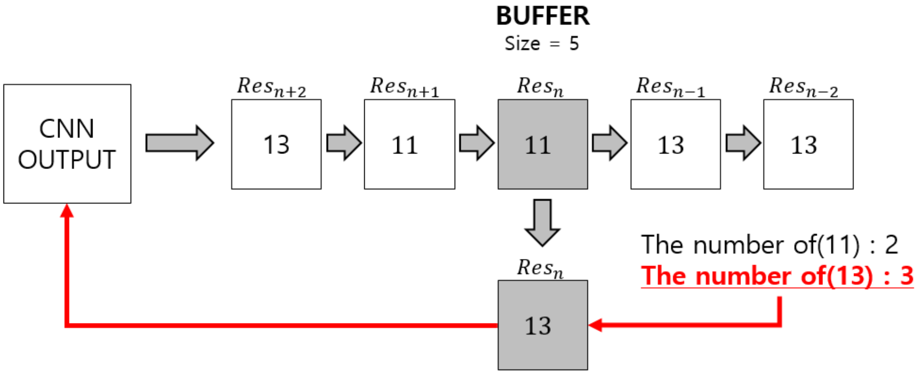

Figure 2. Example of Temporal Classification Error Compensation.

When the (N=5) of buffer of size is used, if there are too many referencing frames, the possibilities of error increases and the temporal correlation with the frame decreases. On the other hand, if very few frames are used as a reference, it can be said that the predicted results of CNN are not dominant enough. Therefore, the number of referred frames must be set considering the error rate and environment where errors occur.

4. Experiment

For the learning of CNN in this experiment, the German Traffic Sign Recognition Benchmark(GTSRB) dataset provided by the Institute for Neural Computation (Institut für Neuroinformatik) in Germany [5]. This includes approximately 40,000 traffic sign learning data and the dataset to test the performance. The dataset includes the images of various sizes and environments.

The 40,000 GTSRB training data were learned after being converted into 28 X 28 grey-scaled image. The number of layers and filters (f_n) were altered for two models to conduct a comparison experiment, and the data were learned for 30 epochs, with an added dropout layer that will prevent overfitting. In this paper, the dropout rate has been set as 0.5.

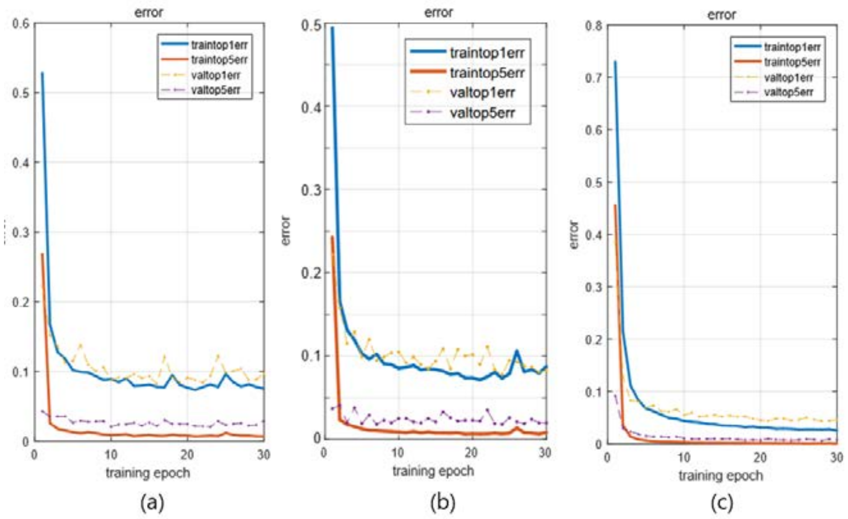

Figure 3. Result of the Learning (a) proposed method without TCEC, (b) 2 layer Conventional CNN Model 1, (c) 3 layer Conventional CNN Model 2

In the experiment, the conventional two models with different numbers of nodes and layers were compared with the proposed method. In Figure 3**,** (a) shows the learning result of proposed method with 2 layers and no TCEC, (b) is the result of Conventional CNN Model 1 with two layers, and (c) shows the learning result of the Conventional CNN Model 2 with three layers. The solid lines are the error rate for the training data, and the dotted lines are the error rate for the validation data. As it is shown in the graph, the error rate of (c) is almost half of (a). While performance enhances when there are three layers, the complexity almost triples compared to the proposed method. In this experiment, A PC with 3.4 GHz Intel Core i7 CPU, 16 GB memory was used, and the experiment was conducted on MATLAB using matConvnet library.

The video images that were used for the actual test are driving car clip images in Germany and edited for 1~4 seconds. A total of 42 clip sequences were used for the experiment. Figure 4 shows some examples of the footages that were used. They were 1280 × 720 HD resolution videos. While they were 25fps videos, the experiment was done after sampling them into 5fps videos in order to prevent the data from becoming massive. The total number of frames of test video were 749, and there were also the frames including partially captured traffic signs as shown in Figure 5**.** Among the videos, 296 images were captured under hard condition such as rainy.

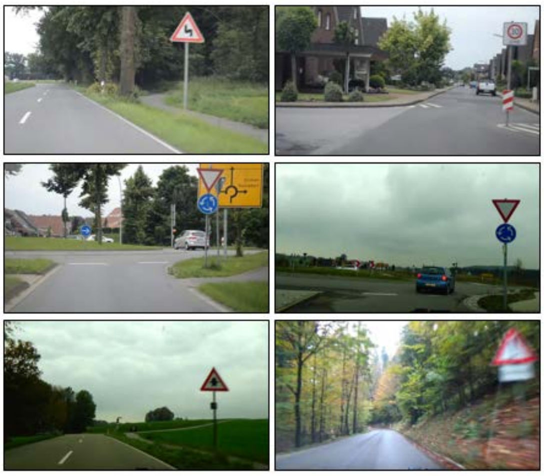

Figure 4. Examples of Test data samples

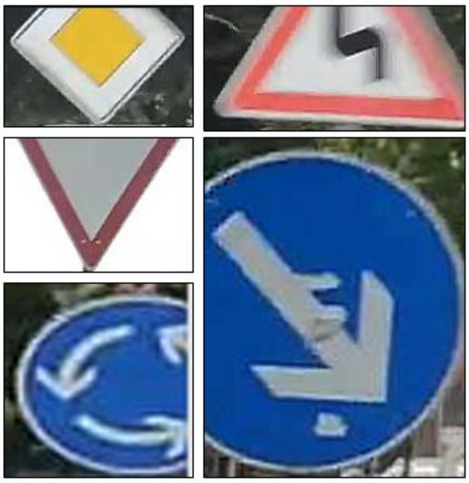

Figure 5. Examples of incomplete signs

Each frame has information on the location of traffic signs and classes, RoI areas are extracted within the frames. The images were input after being converted into grey-scaled images of 28 X 28 size, which is the same as the learned images. The information on classes is used to calculate the error rate. In TCEC, compensation was done after referring two past and future frames, respectively.

Table 1. Results of the experiment.

	Conventional CNN Model 1 with 2 layer $(\mathbf{f_1 = 32, f_2 = 64})$	Conventional CNN Model 2 with 3 layer $(\mathbf{f_1 = 16, f_2 = 32, f_3 = 64})$	Proposed Method without TCEC $(\mathbf{f_1 = 16, f_2 = 24})$	**Proposed Method with TCEC** $(\mathbf{f_1 = 16, f_2 = 24})$
Number of Errors	119	81	122	**89**
Error Rate	0.1589	0.1081	0.1629	**0.1188**

As the results of an experiment are shown in Table 1, while the proposed method without TCEC had higher error rate than the conventional CNN model 1 with two layers, the proposed method with TCEC had lower error rate than it despite the fact that it had more number of nodes. In addition, the error rate of the proposed method with TCEC was similar with that of the conventional CNN model 2 with 3 layer, which has three times more complexity than the proposed method.

$$O\left(\sum_{l=1}^{d} n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2\right) \qquad (1)$$

Equation (1) can be used to calculate the time complexity of all layers of CNN. In the equation, l refers to an index number of convolution layer, d is a depth of the convolution layer. n_l is a width of filter, and s_l refers to the filter size. m_l is a size of feature map [11]. The higher the time complexity, the slower the learning speed of the network.

The time complexity of conventional CNN model 1 is 755,712, and that of conventional CNN model 2 is 1,082,368. The time complexity of proposed method is 285,696. As the proposed method uses classified results of each frame in the conventional method, the complexity does not increase. While the results are output after frame 2, the proposed method improved the error rate even with using less number of nodes and layers compared to models 1 and 2, with more than half their complexities.

5. Conclusion

In this paper, a method of Convolutional Neural Network(CNN) with Temporal Classification Error Compensation(TCEC) for recognizing traffic signs in video sequences has been proposed. It was able to enhance the performance of CNN while not even requiring additional system resources. The proposed method is suitable to be used at sections where traffic signs sequentially appear within a frame. Using the proposed TCEC, we were able to decrease the complexity more than halve compared to conventional CNN models while enhancing the error rate by an additional 3%. It is expected that the proposed method can be effectively applied to autonomous vehicle system or advanced driver assistance systems (ADAS).

Acknowledgement

This paper was supported by Gyeonggi-do and Siheung-city under Industry-University Convergence Campus Outcome Utilization Business, provincial government financial support program, Korea.

References

[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] LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In *ISCAS* (pp. 253-256).

[3] Sermanet, P., Chintala, S., & LeCun, Y. (2012, November). Convolutional neural networks applied to house numbers digit classification. In *Pattern Recognition (ICPR), 2012 21st International Conference* on (pp. 3288-3291). IEEE.

[4] Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011, July). A committee of neural networks for traffic sign classification. *In Neural Networks (IJCNN), The 2011 International Joint Conference* on (pp. 1918-1921). IEEE.

[5] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In *Neural Networks (IJCNN), The 2011 International Joint Conference* on (pp. 1453-1460). IEEE.

[6] Zaklouta, F., Stanciulescu, B., & Hamdoun, O. (2011, July). Traffic sign classification using kd trees and random forests. *In Neural Networks (IJCNN), The 2011 International Joint Conference* on (pp. 2151-2155). IEEE.

[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 Neural Networks (IJCNN), The 2013 International Joint Conference* on (pp. 1-6). IEEE.

[8] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, **32**, 323-332.

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[10] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting.*Journal of Machine Learning Research*, **15**(1), 1929-1958.Another reference

[11] He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5353-5360).