Transactions on Computer Science and Technology
Volume 9, 2022, PP. 1-7

CNN-based Traffic Sign Recognition

Qingkun Huang [1], Askar Mijiti [2]

1. Dali Nursing Vocational College, Dali 671000

2. Nanjing university of Finance & Economics, Nanjing 210000

Abstract

Background: The rapid development of the automobile industry has led to an increase in the output and holdings of automobiles year by year, which has brought huge challenges to the current traffic management. Method: This paper adopts a traffic sign recognition technology based on deep convolution neural network (CNN): step 1, preprocess the collected traffic sign images through gray processing and near interpolation; step 2, automatically extract image features through the convolutional layer and the pooling layer; step 3, recognize traffic signs through the fully connected layer and the Dropout technology. Purpose: Artificial intelligence technology is applied to traffic management to better realize intelligent traffic assisted driving. Results: This paper adopts an Adam optimization algorithm for calculating the loss value. The average accuracy of the experimental classification is 98.87%. Compared with the traditional gradient descent algorithm, the experimental model can quickly converge in a few iteration cycles.

Keywords: Traffic Sign Recognition; Convolution Neural Network (CNN); Adam Algorithm

PREFACE

Traffic signs use images and symbols to mark the lanes. On the one hand, it can standardize road traffic, on the other hand, it can reduce the driver's burden and avoid traffic accidents, especially in the promotion of driverless technology today, the safe and reliable traffic sign recognition system is an indispensable part of the vehicle system. However, in the real traffic environment, due to factors such as weather, light intensity and viewing angle, there are many difficulties in traffic sign recognition, and the practical application is not mature, so there is a broader research prospect in the field of computer vision. At present, most traffic sign recognition models are based on feature extraction and feature classification. For example, in literature [1], based on the color and shape attributes, the traffic signs were classified by neural network, and the recognition accuracy was up to 95%; in literature [2], image feature vectors was extracted through Hu matrix [3] and Zernike matrix [4], and was classified by neural network training features; in literature [5], feature vectors of pre-processed images were extracted through linear discriminant analysis, and then features were identified by the Bayesian classifier model; in literature [6], vector filter SVF was adopted to extract the features of specific colors on traffic signs for verification; in literature [7], CIECAM97 color features were used to extract corresponding features on traffic signs in the FOSTS (foveal system for traffic signs) model, and then classification experiments were performed; in literature [8], traffic signs were initially located through the three-component chromatic aberration of the RGB model, and then the shape information of the traffic signs was employed to achieve classification and recognition; in literature [9], the MSER algorithm was used to extract the region of interest of the traffic signs, and the SVM classifier was trained to detect the traffic signs based on HOG features.

Based on existing studies, this paper mainly adopts Convolution Neural Network (CNN) to recognize and classify traffic signs. CNN is based on the BP neural network. First, it automatically extracts image features through the convolutional layer and the pooling layer, which has the advantage of high invariance to two-dimensional image position translation, scaling, tilt, or other forms of deformation; then the extracted features are classified and output through the fully connected layer and the output layer. CNN is a multi-layer supervised learning neural network. Through CNN [10], the layer-by-layer processing mechanism of human brain perception of visual signals can be imitated to realize the automatic extraction and recognition of visual characteristic signals. Due to the deep structure

characteristics of the convolutional layer and the pooling layer, it has the advantage of sharing weights compared with the traditional BP neural network. But when the task has high time complexity and space complexity, this is a serious defect for the traffic sign recognition system that requires high real-time. Therefore, this paper proposes a CNN optimization algorithm to improve the Adam algorithm to solve the problem of traffic sign recognition. This method can significantly reduce the running time under the condition of extracting the same number of features, and the improved algorithm have good recognition rate and robustness.

1 OVERVIEW OF CNN

First, for image input, CNN employs a convolution kernel composed of a weight matrix to construct multi-dimensional features by convolution on several feature-maps in the horizontal and vertical directions, then the $Relu$ activation function and pooling are performed to achieve dimension reduction, and finally these features are input to the fully connected layer and the output layer. In the convolution process, the convolution kernel is used to perform convolution operations on the input $x \times x$ image along the horizontal x and vertical y directions with step size s. Relu is adopted as the activation function, and the convolution operation can be described as:

$$conv_i = Relu(input \cdot W_i + b_i) \tag{1}$$

$$Relu(x) = \begin{cases} 0 & if(x \le 0) \\ x & else \end{cases} \tag{2}$$

Where W_i represents the weight vector of the i-th layer of convolution kernel, and the operator "·" represents that the feature map is input for convolution operation. The output of the convolution is added to the bias vector b_i of the i-th layer, and the output feature map can be obtained through the $Relu$ activation function. In the process of convolution, the pixel matrix will become smaller after convolution, and there is a problem of information loss at the edge of the image, so pixel padding is necessary. There are two common types of padding, VALID and SAME. The pixel size of the image after VALID padding can be described as:

$$new\_height = \lceil (X_{height} - K_{height} + 1)/s \rceil \tag{3}$$

$$new\_weight = \lceil (X_{weight} - K_{weight} + 1)/s \rceil \tag{4}$$

Where, X_{height} and X_{weight} are the length and width of the original image input pixels; K_{height} and K_{weight} are the length and width of the patch size; and s is the stride of each movement of the convolution kernel. The pixel size of the image after SAME padding can be described as:

$$new\_height = \left\lceil \frac{X_{height}}{s} \right\rceil \tag{5}$$

$$new\_weight = \left\lceil \frac{X_{weight}}{s} \right\rceil \tag{6}$$

After SAME padding, a matrix with the same size as the original pixel can be obtained. Since there is over-fitting phenomenon for the CNN model during the training process, it is necessary to adopt the Dropout algorithm in the fully connected layer. The principle is to perform DNN training by reducing the number of hidden layer nodes (the activation value of a certain neuron is assigned a value of 0 under a certain probability), which is described as follows:

$$r_j^l \sim Bernoulli(p) \tag{7}$$

$$\tilde{y}^l = r^l \cdot y^l \tag{8}$$

$$z_i^{l+1} = w_i^{l+1} \cdot \tilde{y}^l + b_i^{l+1} \tag{9}$$

$$y_i^{l+1} = f(z_i^{l+1}) \tag{10}$$

In Eq. (7), the $Bernoulli$ function generates the probability vector r of 0 and 1 through the probability p. In forward propagation, the inner products of the hidden layer node value and the probability vector are calculated, and

the nonlinear transformation is performed according to Eqs. (9) and (10). It can be seen that the use of the Dropout algorithm, on the one hand, can reduce the model's dependence on the local features of the sample data, on the other hand, it can reduce training costs to improve the generalization ability of the model. Its essential idea is a sparse learning algorithm. In this experiment, the parameters of each layer of the constructed CNN are shown in Table 1.

TABLE 1 CNN PARAMETERS

Type	Patch Size/stride	Out Channels	Padding	Output Size	Dropout_Keep_Prob
Convolution1	[9×9]/4	64	VALID	54×54×64	
Max_pooling1	[3×3]/2	64	VALID	26×26×64	
Convolution2	[3×3]/1	128	SAME	26×26×128	
Max_pooling2	[3×3]/2	128	VALID	12×12×128	
Convolution3	[3×3]/1	256	SAME	12×12×256	
Max_pooling3	[3×3]/2	256	VALID	5×5×256	
Fully_conneted1	[5×5]/1	2048	VALID	1×1×2048	
Dropout1				1×1×2048	0.5
Fully_conneted2	[1×1]/1	2048	SAME	1×1×2048	
Dropout2				1×1×2048	0.5

2 ADAM OPTIMIZATION ALGORITHM

In the CNN model training process, a model can be analyzed through time complexity and space complexity, and the amount of calculation required by the algorithm and the required calculation space can be quantified. This paper assume that the pixel size of an image is $x \times x$, the size of a convolution kernel is $k \times k$, the number of convolution kernel channels is k (that is, the number of input channels in the previous layer), and the number of convolution kernels is k (that is, the output number of channels). In a CNN with a depth of D, the time complexity and space complexity of the corresponding i-th layer can be described as:

$$time \sim O_i\left(\sum_{i=1}^{D} x_i^2 \cdot k_i^2 \cdot m_{i-1} \cdot n_i\right) \tag{11}$$

$$space \sim O_i\left(\sum_{i=1}^{D} k_i^2 \cdot m_{i-1} \cdot n_i\right) \tag{12}$$

From Eqs. (11) and (12), it can be seen that CNN has high time complexity and space complexity in model training. Taking an image with a pixel size of 224×224 as an example, in the network model constructed in this experiment, the complexity required for each layer of training is shown in Table 2. Compared with BP neural network, the parameter complexity of CNN is greatly reduced by sharing weights, but each convolutional layer and pooling layer still requires higher training costs. Therefore, when calculating the loss value in back propagation, based on the Adam algorithm, an improved strategy is proposed (as shown in Table 3). By improving the training method, the loss function is minimized, the accuracy is improved, the training cost is reduced, and the purpose of rapid convergence is achieved.

TABLE 2 ALGORITHM COMPLEXITY

type	$time \sim O_i$	$space \sim O_i$
Convolution1	260112384	5184
Convolution2	49840128	73728
Convolution3	42467328	294912
Fully_conneted1	13107200	13107200

Fully_conneted2	4194304	4194304
Add	369721344	17675328

TABLE 3 ADAM OPTIMIZATION ALGORITHM

Initialization parameters: learning rate $\varepsilon: 0.01$; first-order moment estimation decay rate $\beta_1: 0.9$; second-order moment estimation decay rate $\beta_2: 0.99$, constant for maintaining numerical stability $\delta: 10^{-8}$; first-order moment variable $m_0: 0$; second-order moment variable $v_0: 0$; time $t: 0$; objective function $f(\theta)$ with training parameter θ; iteration function of time $\gamma(t)$; iteration times $i: 0$; task learning number n.

If (θ_t does not converge)

$t = t + 1$

$g_t = \nabla_\theta f_t(\theta_{t-1})$, calculate the gradient

$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$, update the biased first-order moment estimate, with momentum [11]

$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$, update the biased second-order moment estimation, with RMSprop [12]

$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$, correct the deviation of the first-order moment estimation

$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$, correct the deviation of the second-order moment estimation

When $i = \gamma(t)$

Update learning rate $\varepsilon = \frac{\varepsilon}{n}$

Update parameter $\theta_t = \theta_{t-1} - \varepsilon \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \delta}$

Return θ_t

3 EXPERIMENT

This experiment was based on tensorflow deep learning framework, cuda8.0 and win10 operating system. CPU was Intel(R)i7 processor, RMB was 8.0GB, and GPU was NVIDIA GTX1660ti. The experimental data set was collected by crawler technology, and a total of 62 common traffic signs were used as the experimental data set (as shown in Fig. 1). In the preprocessing stage, the color traffic sign image (with a pixel size of 224×224) was converted into a gray image, and then the near interpolation method was used to standardize these traffic signs, and the pixel size was adjusted to 32×32, as shown in Fig. 2.

Fig. 1 Traffic sign data set

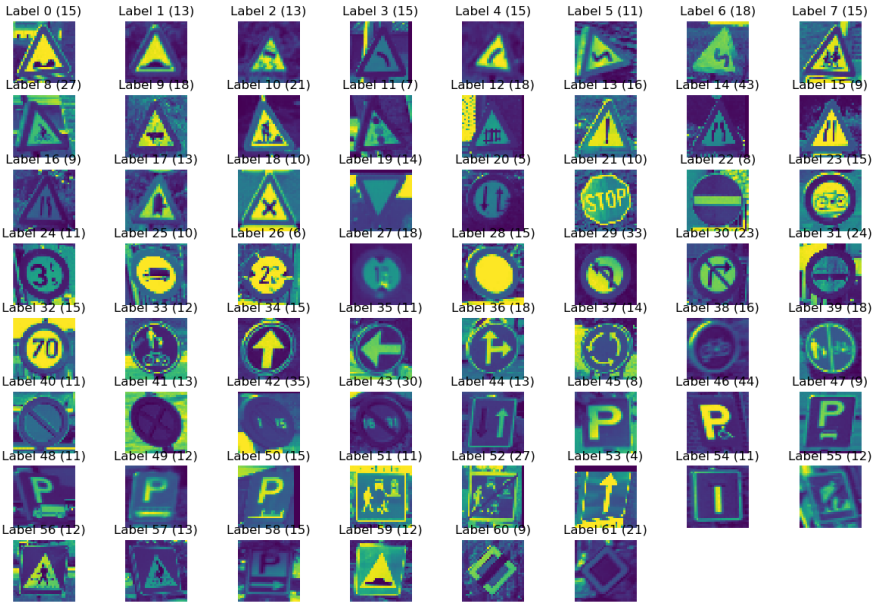

FIG. 2 PREPROCESSED IMAGE DATA SET

In the experiment, the CNN model was built on the tensorflow framework, and the optimized algorithm was adopted. The experimental flowchart was visualized by tensorboard, as shown in Fig. 3.

From bottom to top in Fig. 3, the experimental process is:

(1) Initialization of network weights and thresholds: The random distribution function is adopted to initialize the weight W to a random number between -1 and 1; the threshold b is initialized to 0; the initial value of the learning rate is 0.0001, and the conditions for changing the learning rate is set.

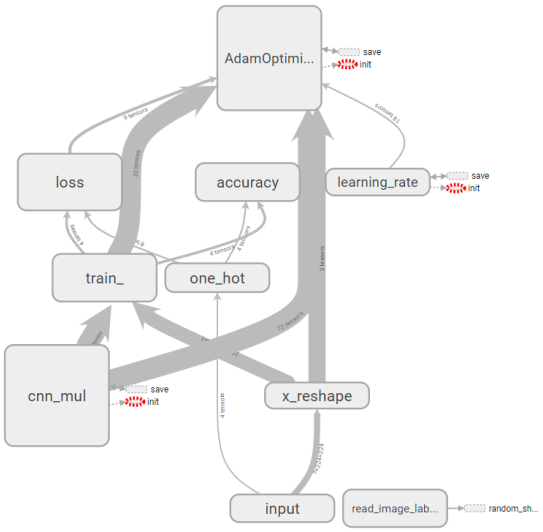

FIG. 3 TENSORBOARD VISUALIZATION EXPERIMENT FLOWCHART

(2) Division of the sample data set: The preprocessed traffic sign images are divided into training set and test set. In the experiment, the x_reshape operation is performed on the 32×32 image, and the correct classification label is one_hot coded, which is convenient to express their respective probability distributions.

(3) CNN model training: The data from x_reshape and one_hot are trained in the cnn_mul constructed in this experiment. In back propagation, the optimized Adam algorithm (AdamOptimizer) is adopted to calculate the loss value, and the accuracy and learning rate of each training are returned and stored until the model converges or reaches a preset number of times.

(4) Algorithm verification: The updated best error of the CNN model is compared with the original error, and the best loss value obtained in the training phase is evaluated. If the error value of the new model is lower than the best error value obtained in the previous iteration, the updated parameters of the model are saved, and finally the classification accuracy is checked by cross-validation.

4 Experimental Results and Analysis

In the experiment, by comparing the gradient descent algorithm, the Adam algorithm, and the algorithm adopted in this experiment, it is found that the optimization algorithm adopted in this experiment is better than the other two algorithms in terms of training time. During the data training process, 20 batches of samples are read each time, and through cross-validation, the average accuracy rate of 500 iterations is 98.85%. The accuracy histogram and part of the predicted images are shown in Figs. 4 and 5.

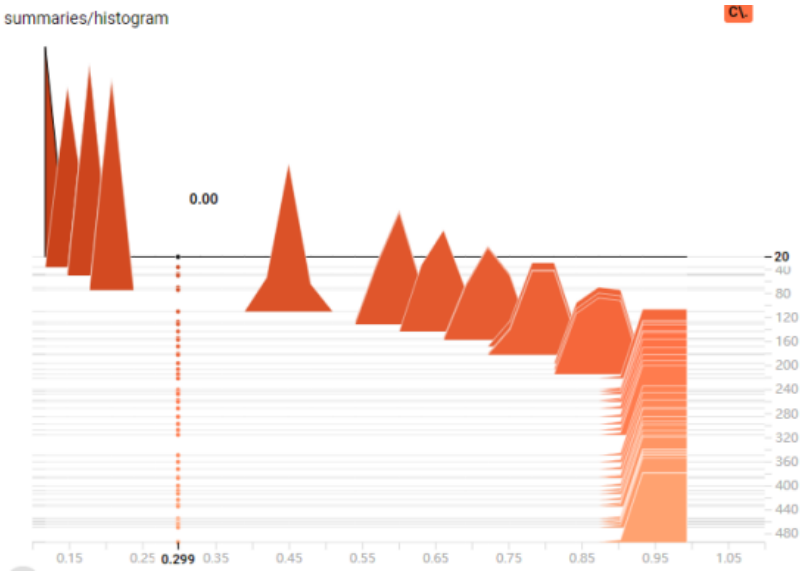

Fig. 4 Histogram of experimental accuracy

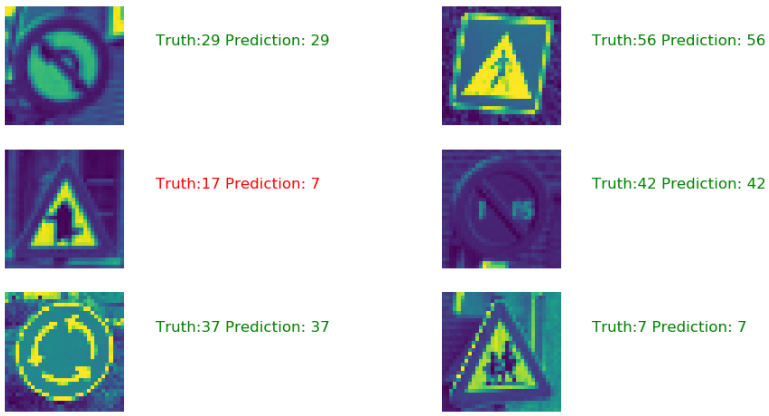

Fig. 5 Images of experimental prediction results

It can be seen from Fig. 4 that during the training and learning process of CNN, the accuracy curve of the entire network rises quickly and steadily, which reflects that the training and learning of CNN has good convergence. After layer-by-layer convolution and pool sampling, the extracted features have scaling and rotation invariance, so the rotated and scaled traffic signs can obtain a higher recognition rate. Compared with the traditional gradient descent algorithm and the Adam algorithm, the Adam optimization algorithm can achieve fast convergence. It can be seen from Fig. 5 that for the collected traffic sign images, the recognition rate of CNN has not yet reached a satisfactory result. The main reason is that there is serious background interference in the traffic sign images collected on the

road. Therefore, in future research, the traffic signs will be obtained under the condition of insufficient illumination or complex background, and a robust positioning algorithm need to be found for the positioning processing and classification.

5 CONCLUSION

In this paper, CNN is applied to road traffic sign recognition. The deep structure of CNN is adopted to simulate the mechanism of human brain perceiving visual signals, the visual features of traffic sign images are automatically extracted for classification and recognition. Experiments show that the recognition of traffic signs through CNN and the optimized Adam algorithm performs well. However, due to complex factors such as weather and light in practical applications, the research in this paper does not involve the recognition and classification of dynamic traffic signs and traffic signs in complex environments. Therefore, in future research, on the one hand, a large number of sample data sets will be supplemented; on the other hand, it is necessary to focus on the application of recurrent neural network and ensemble learning in the recognition and classification of dynamic traffic signs in complex environments.

REFERENCE

[1] Liu H, Ran B. Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicles[J]. Transportation Research Record Journal of the Transportation Research Board, 2001, 1748:161-166.

[2] Tian Qiuhong, Liu Chengxia, Du Xiao. Research on road traffic sign recognition method based on Zernike moments and BP network[J]. Journal of Zhejiang Sci-Tech University, 2012, 29(2): 235-239

[3] D Taubman. High performance scalable image compression with EBCOT[J]. IEEE Transactions on Image Processing, 2000, 9(7): P.1158-1170.

[4] Liao, Simon, X, et al. On the Accuracy of Zernike Moments for Image Analysis[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1998.

[5] Bahlmann C, Zhu Y, Ramesh V, et al. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information[C]// Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE, 2005.

[6] Asakura T, Aoyagi Y, Hirose O K. Real-time recognition of road traffic sign in moving scene image using new image filter[C]// Sice Sice Conference International Session Papers. IEEE Xplore, 2000.

[7] Wga X, Pb L, Sb D, et al. Recognition of traffic signs based on their colour and shape features extracted using human vision models[J]. Journal of Visual Communication and Image Representation, 2006, 17(4):675-685.

[8] Chen H L, Chen M S, Hu S H. An Efficient Embedded System for the Detection and Recognition of Speed-Limit Signs[M]. Springer Netherlands, 2014.

[9] Ahmed N, Rabbi S, Rahman M T, et al. Traffic Sign Detection and Recognition Model Using Support Vector Machine and Histogram of Oriented Gradient[J]. International Journal of Information Technology and Computer Science, 2021, 13(3):61-73.

[10] Arunabala C, Jwalitha P, Nuthalapati S. TEXT SENTIMENT ANALYSIS BASED ON CNNS AND SVM[J]. International Journal of Research -GRANTHAALAYAH, 2019, 7(6):77-83.

[11] Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[12] Krizhevsky A, et al. ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).