Frontiers in Signal Processing, Vol. 3, No. 4, October 2019
https://dx.doi.org/10.22606/fsp.2019.34005

Multi-Scale CapsNet: A Novel Traffic Sign Recognition Method

Gongbin Chen and Yansong Deng*

Key Laboratory of Electronic and Information Engineering (Southwest Minzu University), State Ethnic Affairs Commission, Chengdu, China
Email: 1067475412@qq.com

Abstract Convolutional Neural Networks (CNNs) have performed very well on image classification tasks, but CNNs is insensitive to detailed image information and requires a large amount of training data and time. Capsule Networks(CapsNets) can solve this problem very well, but the Baseline CapsNet model is very shallow, and the extraction of low-level features is not enough. We propose a Multi-Scale Capsule Network (Multi-Scale CapsNet), by extracting the low-level features of images with multi-channel convolution of multiple convolution kernels, so extracted features are more diverse, then passing from the bottom layer to the upper layer in the form of a "capsule", which encapsulats the multidimensional features of the image in the form of a vector, thus the features are saved in the network, rather than being recovered after being lost. In the German Traffic Sign Recognition Benchmark(GTSRB), we obtained competitive results with the accuracy of 99.4%, which is better than the human performance of 98.81% and the Multi-Scale Convolutional Neural Network(MS-CNN) of 97.33%.

Keywords: CapsNets, multi-scale CapsNet, traffic sign recognition, multi-scale convolution, CNN.

1 Introduction

The development of traffic sign detection and recognition began in the 1970s. Due to the limited computing power, the algorithm could not be verified experimentally, so related technology developed slowly. However, with the improvement of computer performance, more and more scholars and major automobile manufacturers have invested in the research of traffic sign detection and recognition.

1.1 Traditional Traffic Sign Recognition Method

The traditional traffic sign recognition classification method is mainly based on color and shape. These methods based on color often use a threshold to separate traffic signs from background[1]. Some researchers use HSI color space instead of RGB and have achieved good performance [2]. A novel color space Eigen color based on Karhunen-Loeve (KL), is used for traffic sign detection [3]. The main disadvantage of these color-based methods is that it is difficult to set the value of threshold because the color information is not invariant in real-world environment with different lightening conditions.

Methods based on shape of the traffic signs, have also been widely used. In [4] a method is proposed using smoothness and Laplacian filter to detect round signs. In [5] a method designed to detect triangle signs based on gradient and orientation information is proposed. In [6] a detection algorithm by using Hough transform is introduced. In order to speed up the detect algorithm, [7][8] use a fast detection method based on the symmetry on Radial direction of triangle, square, diamond, octagon and round signs. Most of the methodsabove rely on gradient features, which are really sensitive to noise. Because color information and shape information are both useful to traffic sign detection, it is natural to combine these two kinds of features. In [9] images are segmented in HSI color space, and template matching techniques are then used to find traffic signs.

1.2 Traffic Sign Recognition Method Based on Deep Learning

Recently, Convolutional Neural Network has been adopted in object recognition for its high accuracy [10] [11] [12] [13]. In [10], a multi-layer convolutional networks is proposed to boost traffic sign recognition,

using a combination of supervised and unsupervised learning. This model can learn multi stages of invariant features of image, with each layer containing a filter bank layer, a non-linear transform layer, and a spatial feature pooling layer. Feeding the responses of both two convolutional layers to the classifier can achieve an accuracy of recognition as high as 97.33%.

In December 2017, HINTON GE proposed the network structure of CapsNets[14], and the training precision on multiMINIST was 99.23%, achieving 79% accuracy on the affinist test set, far exceeding 66% of CNN, while CapsNets consume less time, and are the most accurate network at that moment. Usually, traffic signs on the road can be tilted, rotated, blurred, and so on. CNN has a poor learning effect on spatial location, so when identifying such traffic signs, it will lose detailed information such as position and posture. CapsNets[15][16][17] handles traffic sign images more than CNN in spatial position. The entire learning process of capsnets is transmitted from the bottom layer to the upper layer in the form of "capsules", which encapsulates multi-dimensional features, thus reducing the number of training samples while retaining the characteristics of traffic signs with less probability of occurrence. However, the structure of Baseline CapsNet is very shallow, and the extraction of low-level features is not enough. So, we proposes an improved capsule network-Multi-Scale CapsNet is applied to the traffic sign recognition, the results indicate that the recognition effect is better than the Baseline CapsNet and MS-CNN.

2 Capsule

2.1 Capsule

The original neural network relied on the use of a single scalar output to summarize the repetitive feature detector activity in a local pool. The CNN processed the single image after displacement, rotation, etc., as two image. Neural networks, however, should use multidimensional features, or "capsules", that perform some very complex internal calculations on their inputs and then encapsulate the results into a vector containing informative output. Each capsule learns to identify a visual entity implicitly defined within a local condition and effective deformation range, and outputs a probability and a set of entity parameters that exist within a finite range. The set of entity parameters will include lighting conditions relative to the visual entity, Accurate pose and deformation information. When the capsule is working properly, the probability of the presence of the visual entity is locally invariant, that is, the probability does not change when the entity moves over the apparent manifold within a limited range of capsule coverage. The entity parameter is " equivariant ". As the observation condition changes, the instance parameter will change correspondingly when the entity moves on the appearance manifold, because the instance parameter represents the internal coordinate of the entity on the appearance manifold, as shown in Fig.1.

Suppose that a capsule detects the traffic sign features in the image and outputs a three-dimensional vector of the fixed length. Then the capsule starts moving the traffic sign picture. At the same time, the vector is spatially rotated, indicating that the state of the detected traffic sign has changed, but its length will remain fixed because the capsule is still convinced that it detected the traffic sign. The neural activity will change as the object moves through the image, but the probability of detection remains constant, which is the invariance pursued by CapsNets, rather than the invariance based on maximum pooling provided by CNN.

$$U_{t|i} = W_{it} \cdot u_i \tag{1}$$

where $U_{t|i}$ indicates the high-level features of the underlying feature, W_{it} is the spatial relationship between low-level features and high-level features, u_i indicates low-level features.

2.2 Squash Function

The commonly used activation functions of CNN include ReLU, sigmoid, etc., which are used to compress the linearly superimposed values between 0 1 or -1 1. In CapsNets, because the vector is transported in the front layer network, the "capsule" needs to be processed in the direction of the activation function. The activation function of CapsNets is named Squash, and the expression is as shown in equation(2).

$$v_t = \frac{||s_t||}{1 + ||s_t||^2} \cdot \frac{s_t}{||s_t||} \tag{2}$$

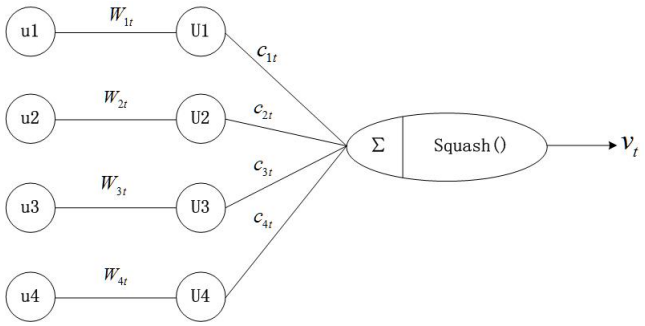

Figure 1. The transfer of Capsule from the bottom to the top.

where v_t is the vector output of capsule t and s_t is its total.

The effect of this function is mainly to make the v_t length not exceeding 1, and make the directions of v_t and s_t coincide. The first term of the formula is the compression function. When s_t is large, the first term approaches 1. If s_t is small, the first term is equal to 0; the second term of the formula is the unitized vector s_t, so the second The item length is 1. In this way, the length of the output vector v_t is between 0 and 1, so this length can be interpreted as the probability that the VN has a given VN feature.

2.3 Dynamic Routing

The dot product of the input and output of the capsule measures the similarity between the input and output, and then updates the routing factor. The optimal number of iterations in practice is one. The steps of dynamic routing are:

(1) Capturing the input images and outputting $U_{t|i}$, routing iteration times r ;

(2) The definition b_{it} is the probability that the l layer VN_i connects to the next layer VN_t, and the initial value is 0;

(3) Loop execution steps (4) to (7) r times;

(4) For VN_i of the l layer, use the softmax activation function to convert b_{it} to probability c_{it};

(5) For $l+1$ layer VN_t, weighted sum s_t;

(6) For VN_t of the $l+1$ layer, use the activation function to activate s_t to get v_t;

(7) Update b_{it} according to the relationship between $U_{t|i}$ and v_t.

The dot product of $U_{t|i}$ and v_t is used to update b_{it}. When the two items are similar, the dot product is large, b_{it} becomes larger, and the probability that the lower layer VN_i is connected to the upper layer VN_t becomes larger; on the contrary, when the difference between the them is large, the dot product is small, b_{it} becomes smaller, and the probability that the lower layer VN_i is connected to the upper layer VN_t becomes smaller.

2.4 Loss Function

The loss function of CapsNets is similar to the loss function of SVM, as shown in equation (3):

$$\begin{aligned} L_c = {} & T_c max(0, m^+ - ||v_c||)^2 \\ & + \lambda(1 - T_c) max(0, ||v_c|| - m^-)^2 \end{aligned} \tag{3}$$

Equation (3) also represents maximizing the distance from the positive and negative samples to the hyperplane. Where, v_c is output DigitCap. T_c =1 represents the correct DigitCap, T_c =0 represents the incorrect DigitCap. Here are two calibration points m^+=0.9 and m^- =0.1, and the loss expects the positive sample m^+ to be predicted at 0.9. If it exceeds 0.9, there is no need to continue to improve; the negative sample m^- is predicted to be 0.1, and if it is less than 0.1, there is no need to continue to fall. The value of λ is fixed at 0.5 and is used for numerical stability during training to prevent excessive loss at the beginning, resulting in shrinkage of all output values. Both terms in the formula have a square because the loss function has an L2 norm and the total loss is the sum of all class losses.

3 Architecture

3.1 Baseline CapsNet

Baseline CapsNet is a very shallow network with 2 convolutional layers and 1 fully connected layer. CNN performs very well on extracting low-level features. Conversely, CapsNets is used to represent an "instance" of an object, so it is more suitable for characterizing advanced instances. Therefore, at the bottom of CapsNets, the convolutional layer is added to extract the underlying feature.

As shown in Figure.2, from the low-level feature to the Primary Capsule, the dimension of the second convolutional layer is $6 \times 6 \times 8 \times 32$, and 8 convolution operations with 32 filters of 9×9 size of 2 strides. In the CNN, those layers of the dimension $6 \times 6 \times 1 \times 32$ have $6 \times 6 \times 32$ elements, and each element is a scalar. In the Capsule, those layers of the dimension $6 \times 6 \times 8 \times 32$ have $6 \times 6 \times 32$ elements, with each element being a 1×8 vector, mainly storing the vector of the level features.

From Primary Capsule to Digit Capsule, Primary Caps and Digit Caps are fully connected, but instead of the traditional CNN of scalar and scalar multiplication, this fully connected layer is a vector connected to a vector.

Digit Capsule to the final output, its length indicates the probability of the content of its representation, so when doing classification, we take the L2 norm of the output vector. CapsNets are different from the previous neural network, whose the output probability sum is 1. Because CapsNets has the ability to recognize multiple objects simultaneously.

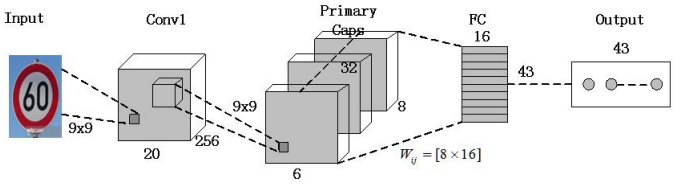

Figure 2. Baseline CapsNet structure.

3.2 Multi-Scale CapsNet

The structure of the Baseline CapsNet is very shallow, and only a convolution layer is used to extract low-level features, so the extraction of basic feature information is not enough. The Multi-Scale[18] CapsNet uses the Multi-channel convolution layer[19] to replace the single convolution layer of the Baseline CapsNet. The convolution of each channel is consisted of various convolution kernels, which include these sizes of 3×3, 5×5, 7×7, 9×9, then concating the outputs of the three channels as the input to the next layer.

As shown in Figure.3, there are three convolution channels in front of the Primary Capsule layer. The first channel uses the combination of 3×3 and 7×7 convolution kernels. Because small convolution kernels can reduce the network parameters and keep the accuracy, while large convolution kernels can expand the receptive field and obtain more information. The second channel uses the original 9×9 convolution kernel to maintain a large receptive field. The last channel uses the combination of 3×3 and 5×5 convolution kernels to replace the 7×7 convolution kernel of the first channel. The main purpose is to improve the depth of the network under the condition of ensuring the same perception field, and to improve the effect of the neural network in some extent. The output of the three channels is 20×20 feature maps, and the number of the feature maps by each channel is 128. Finally, the concating function is used to concating the output of the three channels in the last dimension, so that the feature map is richer and can express the basic features of the image better. After the Primary Capsule layer, capsules are used for route iterations from low-level features to advanced features.

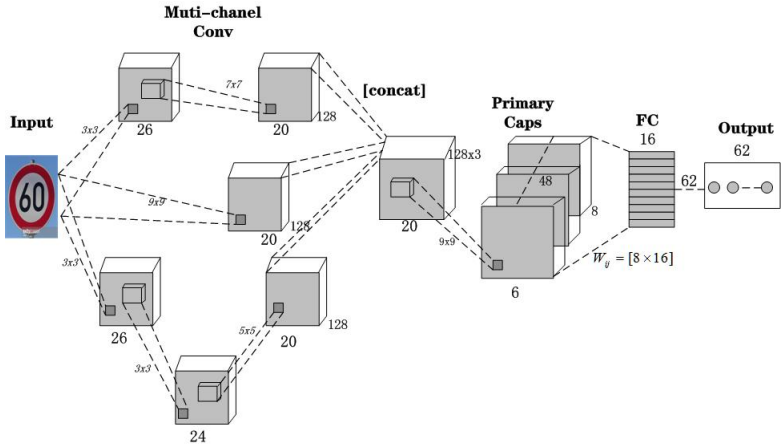

Figure 3. Multi-Scale CapsNet structure.

4 Experiments

4.1 Data Set

The data set used in this experiment is the traffic sign dataset provided by the GTSRB competition, which is divided into training set and test set. The training sets have 39210 inages and the test sets consist of 12569 images. There are a total of 43 categories in the images, and the images in each category are processed into a ppm format image by cropping, contrast, and brightness. During the experiment, the images needs to be uniformly cut into 28 × 28 size, and some images are shown in Figure.4.

Figure 4. Part of the 28 × 28 traffic sign images.

4.2 Experimental Result

This paper uses the Tensorflow open source framework to design and implement a capsule network. The entire experiment was conducted on a NVIDIA 1080Ti for 50 epochs of Baseline CapsNet and Multi-Scale CapsNet. Using the tensorboard visualization tool gets the total loss value and the train acc value of the two model as shown in Figure.5 and Figure.6.

From Figure.5, we can conclude that the loss trends of the two models are roughly the same, but eventually converge to different values. In (a), loss eventually converges to 0.00075, while in (b), it converges to 0.00062. From Figure.6, training accuracy also maintains a consistent trend, and eventually gradually approaching 100%. The curve in (a) does not converges as fast as that in (b), the former reaches 100% after 23k steps, while the latter reaches 100% after about 10k steps, then slightly shakes up and down, and finally approaches 100%.

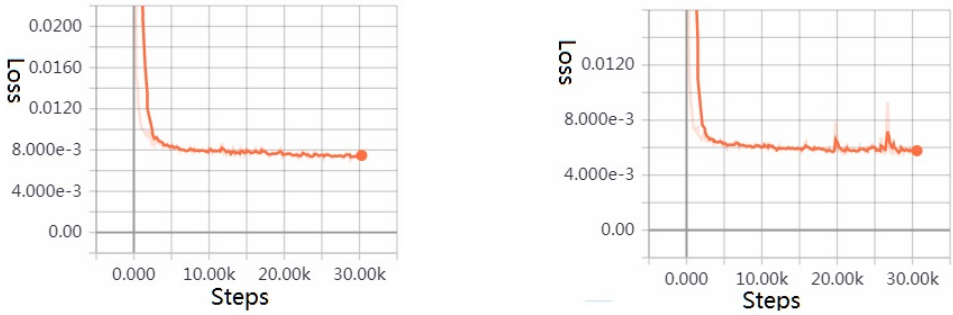

Figure 5. Loss of Base CapsNet(a) and Multi-Scale CapsNet(b).

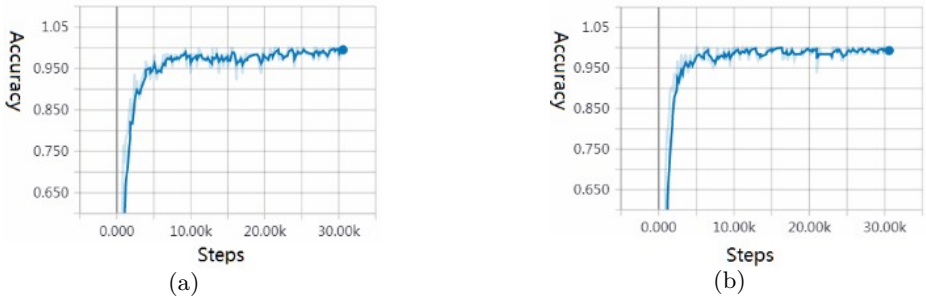

Figure 6. Train accuracy of Base CapsNet(a) and Multi-Scale CapsNet(b).

Comparing the loss and training accuracy of these two models, it can be concluded that the Multi-Scale CapsNet has faster training speed and lower loss compared with Baseline CapsNet.

Table 1. Comparison of different methods for traffic sign recognition.

Method	Routing iteration	Accuracy
$MS-CNN$[10]		97.33%
$Humanperformance$		98.81%
$BaselineCapsNet$	1	97.39%
$BaselineCapsNet$	3	98.63%
$Multi-ScaleCapsNet$	1	98.76%
$Multi-ScaleCapsNet$	3	**99.40%**

Table 1 lists the test accuracy of MS-CNN, Baseline CapsNet and Multi-Scale CapsNet under the same data set, and the comparison of the accuracy of the two capsule models under different routing iterations. In particular, it is better to use 3 routing iteration in our experiment.

By comparing the Capsule model and MS-CNN model, it can be concluded that the CapsNets is better than the MS-CNN in the recognition of the GTSRB dataset. However, by comparing the two capsule models, it can be concluded that the performance of Multi-Scale CapsNet proposed by us outperforms Baseline CapsNet.

5 Conclusion

Experiments have confirmed that Multi-Scale CapsNet, which use a multi-channel convolution technique with a small convolution kernel combination, improves the training speed. At the same time, the multi-

channel convolution is used to fully extract the low-level features of the image, which improves the accuracy of routing iteration between low-level features and advanced features, thereby increasing the recognition accuracy. In the follow-up work, we will also apply the model to more image classification tasks, and continue to study and optimize the performance of the model.

References

1. W. C. U. Ritter and D. B. AG, âĂIJTraffic sign recognition in color image sequences,âĂİ in Intelligent Vehicles '92 Symposium., Proceedings of the.IEEE, 1992, pp. 12âĂŞ17.
2. G. W. N. Ubong Lydia Jau, Chee Siong Teh, âĂIJA comparison of rgb and hsi color segmentation in real - time video images: A preliminary study on road sign detection,âĂİ in Information Technology, 2008. ITSim 2008. International Symposium on, vol. 4, 2008, pp. 1âĂŞ6.
3. L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, Y.-J. Tseng, K.-C. Fan, and C.-C. Lee, âĂIJRoad sign detection using eigen colour,âĂİ ComputerVision,IET, vol. 2, no. 3, pp. 164âĂŞ177, 2008.
4. Y. Aoyagi and T. Asakura, âĂIJA study on traffic sign recognition in scene image using genetic algorithms and neural networks,âĂİ in Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on, vol. 3, 1996, pp. 1838âĂŞ1843 vol.3.
5. R. Belaroussi and J. P. Tarel, âĂIJAngle vertex and bisector geometric model for triangular road sign detection,âĂİ in Applications of Computer Vision (WACV), 2009 Workshop on, 2009, pp. 1âĂŞ7.
6. B. Hoferlin and K. Zimmermann, âĂIJTowards reliable traffic sign recognition,âĂİ in Intelligent Vehicles Symposium, 2009 IEEE, 2009, pp. 324âĂŞ329.
7. G. Loy and N. Barnes, âĂIJFast shape-based road sign detection for a driver assistance system,âĂİ in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 1, 2004, pp. 70âĂŞ75 vol.1.
8. Y.-S. Huang, M.-Y. Fu, and H.-B. Ma, âĂIJA combined method for traffic sign detection and classification,âĂİ in Pattern Recognition (CCPR), 2010 Chinese Conference on, 2010, pp. 1âĂŞ5.
9. P. PaclÅĆk, J. Novovicov, P. Pudil, and P. Somol, âĂIJRoad sign classification using laplace kernel classifier,âĂİ Pattern Recognition Letters, pp. 1165âĂŞ1173, 2000.
10. Pierre Sermanet, Yann LeCun. Traffic Sign Recognition with Multi-Scale Convolutional Networks. In International Joint Conference on Neural Networks, 2011.
11. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, âĂIJWhat is the best multi-stage architecture for object recognition?âĂİ in Computer Vision, 2009 IEEE 12th International Conference on, 29 2009-oct. 22009, pp. 2146 âĂŞ2153.
12. D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, âĂIJA committee of neural networks for traffic sign classification,âĂİ in IJCNN'11, 2011, pp. 1918âĂŞ1921.
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, âĂIJImagenet classification with deep convolutional neural networks,âĂİ in NIPS'12, 2012, pp. 1106âĂŞ1114.
14. SABOUR S, FROSST N, HINTON G E. Dynamic routing between Capsules [J]. NIPS2017, 2017.
15. Geoffrey Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. ICLR 2018, 2018.
16. M. Amer and T. Maul, âĂIJPath capsule networks,âĂİ in preprint, arxiv., 2019.
17. R. Mukhometzianov and J. Carrillo, âĂIJCapsnet comparative performance evaluation for image classification,âĂİ arXiv preprint arXiv:1805.11195, 2018.
18. C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, âĂIJMs-capsnet: A novel multi-scale capsule network,âĂİ IEEE Signal Processing Letters, vol. 25, no. 12, pp. 1850âĂŞ1854, 2018.
19. Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.

