ISBN: 978-981-18-5852-9
WCSE 2022 Spring Event: 2022 9th International Conference on Industrial Engineering and Applications
doi: 10.18178/wcse.2022.04.013

An Improved Capsule Network for the Detection of X-ray Security Images

Hong Zhang [1,2], Baoyang Liu [1,2]

[1] School of Automation, Xi'an University of Posts and Telecommunications, Xi'an, China

[2] Automatic Sorting Technology Research Center, State Post Bureau of the People's Republic of China

Abstract. For X-ray security images inspection, the traditional convolutional neural networks require a large number of samples to train, but capsule network is not limited by training sample size, which contains the information of the position, size, and rotation angle of the object which is recognized through its unique capsule structure to achieve better training results with a small size data set. In this paper, we propose an improved capsule network named as DR-CapsNet for the X-ray security images inspection. DR-CapsNet uses DenseNet as the backbone network to extract image features more efficiently, and uses ResNet to connect the input and the output of DenseNet to prevent overfitting caused by deep network, then adds CBAM to the convolutional layer and the primary capsule layer, that can improve the learning ability, generalization ability and recognition accuracy. The experimental results show that the improved capsule network model DR-CapsNet has 99% recognition accuracy on the small size data set based on GDXray. Compared with the original capsule network, DR-CapsNet improves accuracy by 2%.

Keywords: X-ray security inspection, Capsule network, DenseNet, ResNet, CBAM, small size data set

1. Introduction

For the security inspection such as airports, railway stations, subway stations and public events, X-ray security inspection machines is an important way to detect dangerous goods such as knives, guns and radioactive materials to ensure the safety of passengers. At present, the most inspection and estimate are still processed by security inspectors according to the images captured by X-ray security machines. It results that the accuracy of inspection is limited to the experience of security inspectors, and the error estimation is easy to happen when inspectors face the challenges of a plenty of luggage. Therefore, for the accuracy and rapidity of security inspection, the development of automatic recognition technology for X-ray security images is the key to improve security inspection efficiency. At present, the algorithms are usually used for security screening images include ResNet, Inception, DenseNet, CapsNet and so on. Traditional network models such as ResNet and Inception require a large amount of training data to train for achieving high recognition accuracy. DenseNet can extract image feature information very well, but requires very high video memory and long training period, which is quite demanding for the equipment and cannot be used easily. CapsNet has low requirements for equipment and high accuracy for small size data set. Therefore, on the basis of CapsNet, this paper combines the unique network structure of DenseNet and ResNet to extract image features, which achieves low equipment requirements and higher accuracy for X-ray security images with small size training data set.

2. Capsule Network

2.1. Related Analysis

As one of the representative algorithms of deep learning, convolutional neural network has good feature extraction ability in image recognition. Compared with other conventional methods, the convolutional neural network reduces the number of training parameters and effectively reduces the feature dimension through pooling, so that improves the ability to generalize[1]. However, convolutional neural network still has some problems such as a large amount of information will be lost in the pooling layer of convolutional neural network. Since the scalar neurons of convolutional neural network cannot express the characteristic location, the network structure depends on a large number of sample training. But it is not possible to get a lot of

pictures of dangerous goods in many cases, so that the features of dangerous goods cannot be inadequate extracted under the background of complex image.

In view of the limitations of convolutional neural networks, Hinton had been searching for a better way to deal with it. In 2017, he published a paper entitled "Dynamic Routing Between Capsules" at the Conference on Neural Information Processing Systems[2]. Unlike the output of convolutional neural network which is a scalar, the output of capsule network is a vector with direction. Capsule network learn features in the capsule and retain key features to the greatest extent, so it can achieve the recognition accuracy of convolutional neural networks with less training data, which makes it more popular in the fields of face recognition, image recognition, character recognition, etc.

Each capsule contains a lot of information, such as the position, rotation angle, thickness, tilt, size and other information of the targeted object[3], so the capsule network can recognize the same type of objects at different angles. Therefore the capsule network has better recognition accuracy in the security inspection and identification of small size data set.

2.2. Network Structure

The capsule network is developed based on the convolutional neural network, but it is different in structure. The structure of capsule network is shown in Fig. 1, the first layer is the convolutional layer, and the second layer is the primary capsule layer, the third layer is the digital capsule layer. The primary capsule layer is also very similar to the convolutional layer, its function is to obtain the lowest-level multi-dimensional features and combine these features to form a capsule. The basic unit of the capsule network is the capsule, the capsule contains multiple neurons, which detect and learn some specific areas in the picture, and then output vectors. These vectors represent various attributes of the object in the input image, and these attributes include many different types of parameters, such as texture, position, size and so on.

The first convolution layer uses 9×9 convolution kernels with a number of 256, and uses relu activation function to output a tensor of 20×20×256. The primary capsule layer is one of the cores of the capsule network, which implement to convert scalars to vectors. The input of the primary capsule layer of the capsule network is a 20×20×256 feature map, it uses 9×9 convolution kernels with a number of 8×32, the step length is 2 to obtain 8 groups of 6×6×32 feature maps, and the corresponding positions of the 8 groups of feature maps are linearly combined to obtain 1152 capsules in the primary capsule layer.

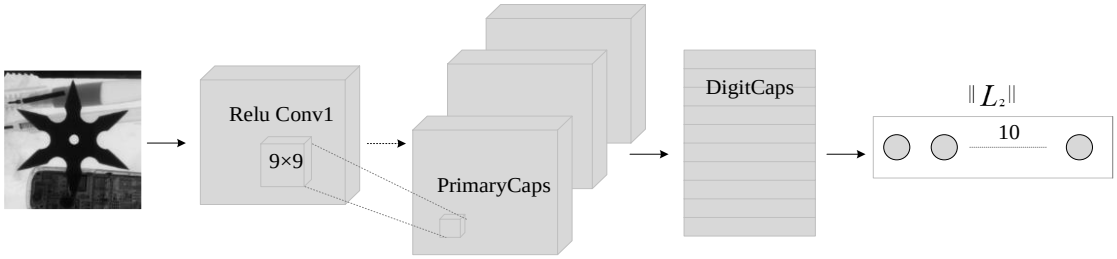

Fig. 1: The original capsule network.

A dynamic routing mechanism is added between the primary capsule layer and the digital capsule layer to find a set of coupling coefficients c. Then the vector v that best matches the output is obtained.

Given c is the coupling coefficient which is a special probability distribution. The greater the connection between the capsules, the greater the subordination between the lower capsules and the higher capsules. And the coupling coefficient has the following two characteristics:

(1) All coupling coefficients are non-negative scalar values, $c \geqslant 0$.

(2) For each low-level capsule, the sum of the coupling coefficients corresponding to all the high-level capsules connected to it should be 1.

The core work of the dynamic routing module is to find the best weight coefficient c. In order to get c, we usually need to initialize the variable $b_i^j = 0$, and then use b_i^j to initialize c.

$$c_i^j = \frac{\exp(b_i^j)}{\sum_k \exp(b_i^k)} \tag{1}$$

The output of the primary capsule layer is linearly combined with corresponding positions to form a vector u, and then the corresponding coupling coefficient which from formula(1) is weighted to sum to get vector s_j.

$$s_j = \sum c_i^j u^i \tag{2}$$

Squashing's nonlinear function is used as the activation function of the capsule to get the vector k_j:

$$k_j = \frac{\|s_j\|}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{3}$$

Then b_i^j is updated by the vector k_j which from formula(3). The formula for updating b_i^j is shown as follows:

$$b_i^j = b_i^{j-1} + k^j u^r \tag{4}$$

According to formula(1), we can use b_i^j to update c_i^j. In general, the best vector v is obtained after three iterations of dynamic routing which shown in Fig.2.

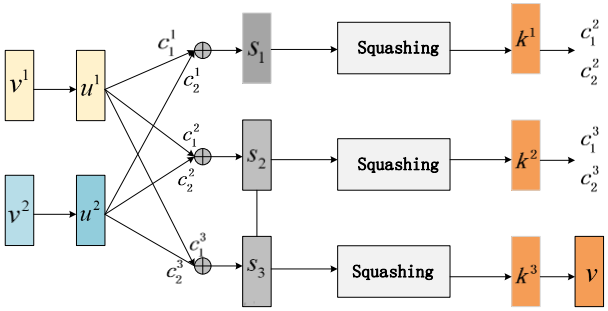

Fig. 2: Dynamic routing.

In Fig.1, the digital capsule layer takes output vectors from the base capsule layer as input. Inside the digital capsule layer, each output vector is mapped to the capsule output space through the weight matrix input control. The number of capsules in this layer is 10, and the capsule dimension is 16. When it is necessary to determine what type the output belongs to, the modulus lengths of 10 capsule vectors are respectively calculated[4]. The category corresponding to the capsule with the largest modulus is the judgment result of the network[5]. In other words, the length of the vector is taken as the measure of whether the object exists. The smaller the length is, the lower the possibility of its existence is, otherwise, the greater the possibility of its existence is. To allow for multiple digits, the loss function in the capsule network adopts Margin loss which commonly used in SVM, L_c for each digit capsule and the expression of the loss function is shown as follows:

$$L_c = T_c \max(0, m^+ - \|v_c\|^2) + r(1-T_c)\max(0, \|v_c\| - m^-)^2 \tag{5}$$

In the formula, C represents type. If class C exists, $T_c = 1$, on the contrary, $T_c = 1$. Such edge loss makes the corresponding vector length should be large if there are C class objects in the image.

3. Improved Capsule Network

3.1. DenseNet

In order to extract more adequate image features and to ensure maximum flow of feature information between the layers in the network, this paper use DenseNet[6] to feedforwardly connect the feature maps from all previous layers as input to the current layer, and the feature maps of the current layer are used as part of the input to all subsequent layers.

Instead of drawing representative features from extremely deep or wide architectures, DenseNet taps into the potential of the network through features reuse to generate model. It increases the efficiency by concatenating feature maps learned in different layers, improves the variability of inputs in subsequent layers without relearning redundant feature maps.

In order to avoid overfitting the network by too deep feature extraction network, this paper use a simplified DenseNet network with three Dense Blocks which named as Db in Fig.3. Each one individually contains 4, 8, and 4 bottlenecks. By using three transition layers to reduce the number of channels which named as Ts in Fig.3. The structure of DenseNet is as follows:

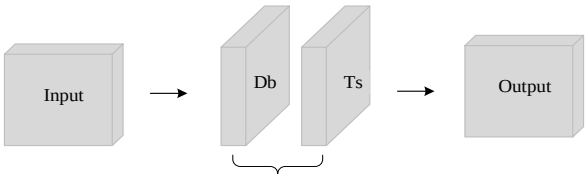

Fig. 3: DenseNet structure diagram.

The structure of bottleneck and transition are shown in Fig.4 and Fig.5, BN is also known as BatchNorm, this makes the input of each layer of neural network keep the same distribution in the training process of deep neural network, which can accelerate the convergence speed of the network. 1×1 convolution is used for dimensionality reduction, which greatly reduces the computation and the computational complexity of 3×3 convolution. The relu function is used to increase the nonlinear relationship between the layers of the neural network, and the dropout layer makes part of the hidden layer node value 0, which can be used to prevent overfitting.

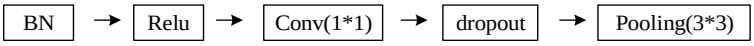

Fig. 4: The structure diagram of transition layer.

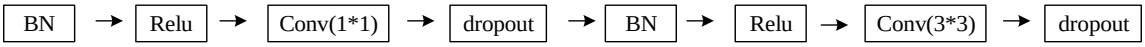

Fig. 5: The structure diagram of bottleneck layer.

3.2. ResNet

ResNet[7] was proposed by Kaiming He in 2015 to solve the network degradation problem caused by deepening the number of layers of the network. For small size data set, the simplified DenseNet still has a high complexity and there is still a risk of overfitting. Therefore, residual network is used to combine the initial input image information with features extracted by DenseNet to prevent overfitting and network degradation.

Its formula is as follows:

$$y = F(x) + x \tag{6}$$

$F(x)$ denotes the output of DenseNet, x denotes the input to the network and y denotes the output of DRNet.

We can name the new network which formed by the combination of DenseNet and Resnet as DRNet, its network structure is shown in Fig.6.

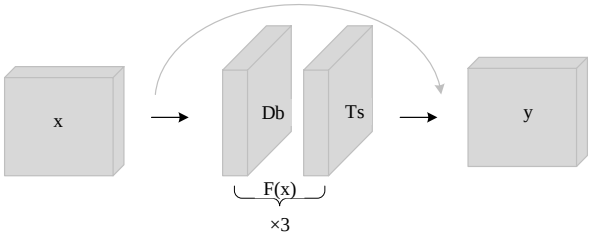

Fig. 6: The structure diagram of DRNet.

3.3. Cbam

Convolutional Block Attention Module(CBAM)[8] was proposed in 2018 by Sanghyun Woo et al. CBAM consists of two independent submodules, Channel Attention Module (CAM) and Spatial Attention Module (SAM), which perform attention in channel and space respectively. The attention mechanism in the channel was proposed in SENet[9]. Its structure diagram is shown in Fig.7.

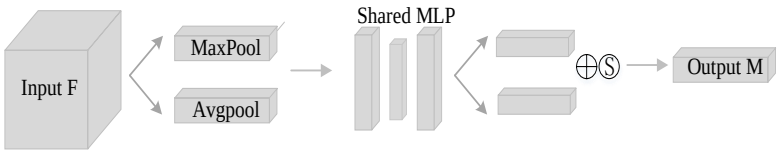

Fig. 7: The structure diagram of CAM.

The input feature maps F are subjected to global max pooling and global average pooling to obtain two feature maps, and then they are fed into a two-layer neural network. The activation function is relu and the two-layer neural network is shared. After that, the output features are subjected to element-wise summation and then use sigmoid activation to generate the final channel attention feature. Finally, the output M and the input feature map F are element-wise multiplied to generate the input features needed by the Spatial attention module.

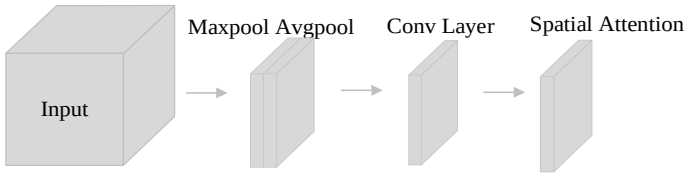

Fig. 8: The structure diagram of SAM.

The SAM takes the output of the CAM as the input feature maps of this module. Its structure diagram is shown in Fig.8. First, using a global max pooling and global average pooling which based on channel to get two feature maps, after that, we do channel splicing on these two feature maps.

Then, using a 7×7 convolution makes these feature maps into one channel and the spatial attention feature is generated by sigmoid. Finally, the feature is multiplied with the input of this module to get the final generated feature. The CAM is serialized with the SAM to get the CBAM, and its final structure is shown in Fig.9.

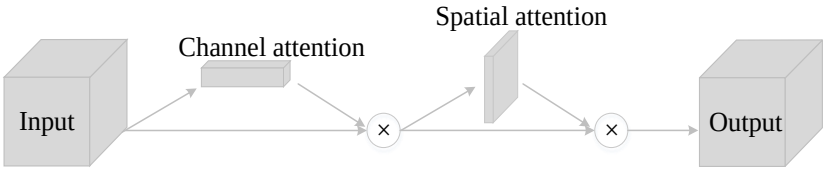

Fig. 9: The structure diagram of CBAM.

In this paper, the CBAM block is added between the convolution kernel and the primary capsule layer. In the CBAM block, the features extracted from the convolution are further processed by the attention mechanism, and then input into the primary capsule layer to form a capsule, so that the features information contained in each capsule is more rich. At the same time, because the dangerous objects in the process of security check and identification usually have different complex backgrounds, by adding the CBAM block, the network structure can focus more on the target object, so as to improve the generalization ability and the recognition rate when the background of the training set is significantly different from the background of the test picture. The new network which based on CapsNet use DRNet to extract image features and use CBAM to enhance recognition, we name the new network as DR-CapsNet, its structure diagram is shown as follows:

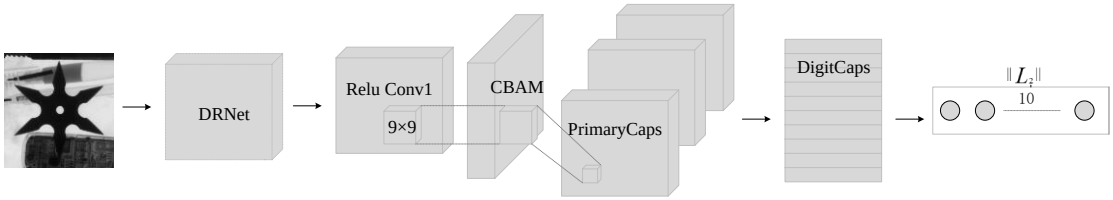

Fig. 10: The structure diagram of DR-CapsNet.

4. Experimental Results and Analysis

4.1. Experimental Environment

The computer processor used in this experiment is the Intel Core i7-11800H model, the graphics card is GTX3060, the running memory is 16GB, the operating system is Ubuntu20.04, the programming is tensorflow-estimator1.15.1 and python3.8.5.

4.2. Data Set and Preprocessing

In this paper, 1000 training samples and 200 test samples were selected from the GDXray[10] data set to form a small size data set in this project. GDXray Dataset consists of 19,407 X-ray images, including five sets of X-ray images such as: castings, welds, luggage, natural objects, and environment. The dataset is freely available, but only for research and educational purposes.

The training set and test set used in the experiment are divided into four categories: darts, blades, pistols and daggers, all of which need to be preprocessed. Data preprocessing first needs to gray out the sample image, and convert the image into a uniform size of 28*28, and then encode the image into binary format by code, such as train-images-idx3-ubyte. The preprocessed images are shown as follows:

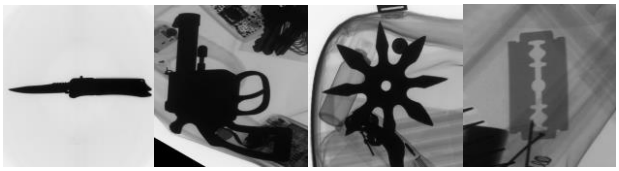

Fig. 11: Preprocessed images.

4.3. Result and Analysis

In this experiment, the network is initialized with the pre-trained weight file on MNIST. The original capsule neural network and DR-CapsNet are used to train and test respectively to obtain the recognition accuracy. For comparison, we also use ResNet18[11], InceptionV3[12], and MobileNet[13] to based on small size data set. The experimental results of loss function curves are shown as follows:

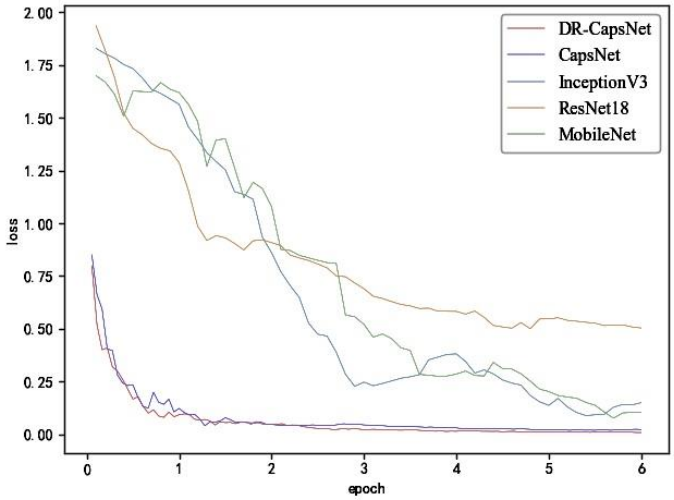

Fig. 12: Loss value comparison of five networks.

As shown in the Fig.12, the traditional neural network ResNet18, InceptionV3 and MobileNet obtain much higher loss value than CapsNet and proposed DR-CapsNet. CapsNet can get lower loss value due to its unique network structure and initial weight. The fluctuation of DR-CapsNet convergence curve is relatively small compared with the original CapsNet, and the final convergence value is also smaller, so DR-CapsNet can achieve higher identification accuracy.

The recognition accuracy value and the training time are listed in Table I.

Table 1: Comparison of metric values of networks

Network	Acc(%)	Training time(s)
ResNet18	91.00	236.78
InceptionV3	94.50	447.80
MobileNet	90.50	282.83
CapsNet	97.00	61.43
DR-CapsNet	99.00	70.27

Table 1 indicates the accuracy and the training time of small size data set for five neural network models, it shows the accuracy of proposed DR-CapsNet is the best, beause the DRNet and the CBAM block imporve the learning ability and generalization ability of system. Although the DRNet and the CBAM increase some calculation amount which leading to slightly higher training time than CapsNet, but much faster than the other three networks.

5. Conclusion

The unique capsule structure of the capsule neural network contains the location, size and direction information of the object to be recognized. For the dangerous objects that are usually disordered and sheltered in X-ray security inspection, the capsule neural network can have a high recognition accuracy for the small size data set of X-ray security inspection. In this paper, we propose DR-CapsNet which uses DenseNet as the feature extraction network in order to strengthen the reuse of features and applys ResNet to combine the initial input with the output of DenseNet to prevent the over-fitting caused by the deep network of DenseNet, moreover the CBAM block is added between the convolutional layer and the capsule layer to enhance the generalization ability of DR-CapsNet. Compared with original capsule network, ResNet18, InceptionV3 and MobileNet, our proposed model can obtain highest recognition accuracy with better object features extraction for the complex security image data set. So the DR-CapsNet can greatly improve security when assisting security inspection. In the future, we will improve the DR-CapsNet using self-attention mechanism to enchance the recognition ability for the security images with complex background.

6. Acknowledgements

This research is supported in part by Shaanxi Provincial Natural Science Foundation of China (Grant No.2021 SF-478).

7. References

[1] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354-377.

[2] Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[J]. arXiv preprint arXiv:1710.09829, 2017.

[3] Vijayakumar T. Comparative study of capsule neural network in various applications[J]. Journal of Artificial Intelligence, 2019, 1(01): 19-27.

[4] LaLonde R, Bagci U. Capsules for object segmentation[J]. arXiv preprint arXiv:1804.04241, 2018.

[5] Lin A, Li J, Ma Z. On learning and learned data representation by capsule networks[J]. arXiv preprint arXiv:1810.04041, 2018.

[6] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[7] Hwei Jen Lin, Yoshimasa Tokuyama, and Zi Jun Lin, "Residual Learning Based Convolutional Neural Network for Super Resolution," Journal of Image and Graphics, Vol. 7, No. 4, pp. 126-129, December 2019. doi: 10.18178/joig.7.4.126-129.

[8] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[9] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer

vision and pattern recognition. 2018: 7132-7141.

[10] Mery D, Riffo V, Zscherpel U, et al. GDXray: The database of X-ray images for nondestructive testing[J]. Journal of Nondestructive Evaluation, 2015, 34(4): 1-12.

[11] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[12] Xia X, Xu C, Nan B. Inception-v3 for flower classification[C]//2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2017: 783-787.

[13] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.