# Deep Capsule Network Handwritten Digit Recognition

Yuxing Tan
School of Computer Science and Engineering Xi'an Technological University
Xi'an, China
E-mail: Yuxing_Tan@foxmail.com

Hongge Yao
School of Computer Science and Engineering Xi'an Technological University
Xi'an, China
E-mail: 835092445@qq.com

***Abstract*—Aiming at the weakness of CNN that is not sensitive to the changes of relative position and angle, a method of digital handwritten recognition based on deep capsule network is researched. The capsule network represents multiple attributes of an entity through a group of capsules composed of neurons, which effectively preserves the information about the position and posture of the entity. Dynamic routing algorithm makes the information interaction between capsules more clearly, and can determine the pose of the entity more accurately. While solving the shortcomings of convolutional neural networks, it also integrates the advantages of CNN and considers the relative position of it's lack, so that the recognition effect is improved. The design implements a deep capsule network, reduces the amount of trainable parameters by changing the size of the convolution kernel, expands on the original network structure, adds a convolution after the convolution layer, and a process of dynamic routing on the main dynamic routing is added, and the number of iterations is changed for experimentation, which makes the accuracy of network recognition higher on the MNIST data set.**

***Keywords-Component; Deeplearning; Nerve Capsule; Deep Capsule Network; Handwritten Digit Recognition***

## I. INTRODUCTION

In our daily life, handwritten numbers are very common, but in many areas of work, the part about numbers is sometimes very cumbersome, such as data collection, which is a time-consuming, large amount of work. At this time, the function of handwriting recognition technology is reflected, which brings convenience and efficiency to human.

The proposal of nerve capsule comes from a assumption of Hinton[1]: instead of using a group of coarse coding or single neurons to represent the relationship between the observer and the object similar to the object's posture information, a set of activated neurons is selected to represent it. This group of neurons is called nerve capsule. One of the advantages of capsule network is that it needs less training data than convolutional neural network, but the effect is not inferior to it.

For the traditional neural network, neurons can not represent multiple attributes at the same time, resulting in the activation of neurons can only represent a certain entity. In this way, the nature of the entity will be hidden in a large number of network parameters. When adjusting the network parameters, we can not guarantee the pure motivation. It must take into account the input of all kinds of samples, so it is inevitable to adjust the parameters in a troublesome and time-consuming manner. After the application of vector neurons, we will be able to determine the existence of all the properties wrapped in a capsule, in the adjustment of parameters, such constraints will be greatly reduced, the best parameters are easy to obtain.

The design and research of artificial neural network largely borrows from the structure of biological neural network. In the field of neurolysis, a conclusion has been drawn that there are a large number of cortical dominant structures in the cerebral cortex of most primates. There are hundreds of neurons in the cortex of most primates, and there are also hierarchical structures in it. These small units can handle different types of visual stimuli well. The researchers speculate that there is a mechanism in the brain that combines low-level visual features with some weight values to construct a colorful world in our eyes. Based on this discovery in biology, Hinton suggests that it is more appropriate to try to replace the relationship between the object and the observer with a series of active neurons instead of one. So, there is the nerve capsule mentioned earlier.

DOI: 10.21307/ijanmc-2020-031

In October 2017, Sabour, Hinton and others published the topic "Dynamic Routing Between Capsules "[10] at a top-level conference on machine learning called "NIPS" and proposed Capsule network (CapsNet). This is a deep learning method that shakes the whole field of artificial intelligence. It breaks the bottleneck of convolutional neural network (CNN) and pushes the field of artificial intelligence to a new level. This paper focuses on the recognition of MNIST data set based on capsule network. MNIST[7] is a data set composed of numbers handwritten by different people.

Although handwritten digits in BP neural network[9] and convolution neural network[2][5][6][11] have a certain good recognition effect, but the emergence of capsule network brings a new breakthrough to the recognition of data sets, and has a better recognition effect, and it's recognition accuracy greatly exceeds the convolutional neural network.

## II. RELATED WORK

The neural capsule proposed by Hinton is to implement ontology from the perspective of philosophy. The various properties of a particular entity are represented by the activity of nerve cells in an activated capsule. These attributes include the size, location, orientation and other information of the entity. From the existence of some special attributes, we can infer the existence of instances.

In the field of machine learning, the probability of entity existence is represented by the output size of independent logistic regression unit. In the neural capsule, the norm obtained by normalizing the output high-order vector represents the existence probability of the entity, and the attributes of the entity are represented by various "posture" of the vector. This reflects the essence of ontology, that is to define the existence of entity according to its various attributes.

In the research of capsule network, the working process of capsule network is closer to the behavior of human brain because of its less training data. In the aspect of white box adversarial attacks, capsule network shows strong resistance. Under the effect of the fast gradient symbol method, the accuracy can still be maintained above 70%. The accuracy of training and testing on MNIST is better than that of convolution neural network. In some practical applications, such as in specific text classification tasks, convolution capsule network can effectively improve the accuracy of feature extraction. [12]Chinese scholars have also applied the visual reconstruction method based on capsule network structure in the field of functional magnetic resonance imaging. In the intelligent traffic sign recognition, by introducing pooling layer into the main capsule layer, the super depth convolution model improves the feature extraction part of the original network structure, and uses the moving index average method to improve the dynamic routing algorithm, which improves the recognition accuracy of the network in the field of traffic sign recognition.

The capsule network first appeared in the article "Dynamic routing between capsules" published by Hinton et al. in October 2017. Based on the capsule network proposed by Sabour et al in 2017, an improved version of the new capsule system was proposed in the article "Matrix Capsules with EM Routing"[3] published in 2018.

In this system, each encapsulated capsule uses a logical unit to represent the presence or absence of an entity. A 4×4 pose matrix is used to represent the pose information of the entity. In this paper, the iterative routing method between capsule layers based on EM algorithm is mentioned. The output of the lower layer capsules reaches the higher level capsules through routing algorithm, so that the activated capsules get a group of similar pose voting. The new system is much more resistant to Lily attacks than baseline CNN. In the paper "Stacked Capsule Autoencoders"[4] published in 2019, an unsupervised capsule automatic encoder (SCAE) is introduced. By observing the neural encoders of all components, the existence and pose information of the target can be inferred, that is to say, the object can be inferred explicitly through the relationship between the components. The accuracy on SVHN[8] and MNIST datasets is 55% and 98.5%, respectively.

## III. Deep capsule network

### A. *Structure of deep capsule network*

#### 1) *Encoder structure of deep capsule network*

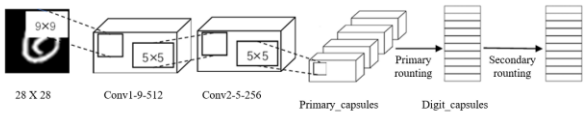


Figure 1. Network structure of deep capsule

**Conv1:** Standard convolution. It is dedicated to extract some low-level feature information from the input image. The preprocessing data layer of capsnet is to convert the brightness of pixels in the input layer into local feature output. The input image of this layer is 28×28, with 256 convolution kernels with step size of 1 and size of 9×9. After convolution, the output is a

three-dimensional array. By reshaping the array, the appropriate feature vector of position information is constructed for each dimension. The final output is a tensor of 20×20×256.

**Conv2:** Standard convolution layer, including 256 convolution cores with step size of 1 and size of 5×5, input tensor of 20×20×512 and output of tensor of 16 ×16×256.

**Primary capsule:** primary capsule layer, also known as the primary capsule layer, is in a low-level stage, multidimensional entities are described in capsnet from the perspective of "inverse graph". It is a reverse rendering process, that is, this layer can combine the low-level features detected by the previous layer. This layer is still committed to extracting feature information, so it still belongs to convolution layer. The object of convolution is changed from single neuron to capsule with larger granularity, which is the difference between convolution network and convolution network. The primary capsule layer is the convolution layer of "capsule version". This stage is also where the capsule really begins. This layer consists of 32 main capsules, each of which contains 8 convolution kernels of 9×9 ×256 with step size of 2. According to the above, the tensor of 6× 6×8×32 is obtained by inputting 20× 20×256 tensors in this layer.

**Digitcaps:** Digital capsule layer, also the full connection layer of capsule network. Using a fully connected topology, the capsules in this layer will connect all outputs of the previous primary capsule layer. Because this paper finally realizes the recognition of 0-9, so there are 10 capsules in this layer. The norm of each activation vector represents the probability of each classification and is used to calculate the classification loss. The input received by this layer is the tensor of 6×6 ×8×32 of the output of the previous layer, and the output is a matrix of 16×10.

Finally, the capsule network is compared with the improved deep capsule network as shown in the following table 1:

*2) Decoder structure*

The decoder structure in this paper is the same as capsnet, as shown in the figure. The goal of capsnet model optimization is to calculate the edge loss for each number to allow multiple numbers to exist at the same time. In addition, capsnet can reconstruct the input image based on the instantiation parameters obtained by previous processing. In the training process of image reconstruction, only the activated capsules are allowed to participate in the adjustment of three-level fully connected network at each time. The structure mainly responsible for reconstructing the image is the decoder, which receives a 16×10 matrix from the digital capsule layer, reconstructs a 28×28 image after three full connection layers.

TABLE I. STRUCTURE COMPARISON OF CAPSULE NETWORK AND DEEP CAPSULE NETWORK

| | Capsule network | Deep capsule network |
|---|---|---|
| Convolution layer | Conv1: 256*9*9 | Conv1: 512*9*9<br>Conv2: 256*5*5 |
| Primary Capsule | 9*9 | 5*5 |
| Digit Capsule | One time dynamic routing<br>Three iterations | Twice dynamic routing<br>The main route has three iterations, and the secondary route has three iterations |
| FC | | |

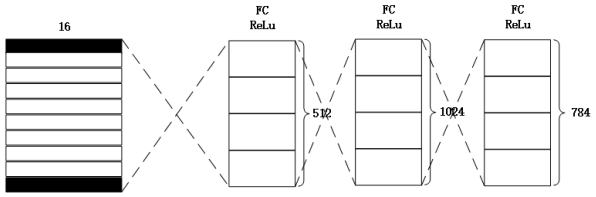


Figure 2. Decoder network

*B. Working mechanism of dynamic routing*

In this paper, there are two routes, the primary route and the secondary route, but both are the same dynamic path structure. It is used to ensure that the output of the capsule is only delivered to the appropriate parent node, which is similar to the idea of "focusing on Cultivation". It is necessary for the lower layer capsule i to know how to deliver its output vector to the higher-level capsule j. At this time, it is necessary to evaluate the coupling degree between the low-level capsules and the high-level capsules. This is represented by the scalar weight CIJ, which is the importance.

In this high-dimensional vector space, in order to describe the spatial relationship of different parts of the entity, each capsule is set with corresponding weight. An affine transformation matrix, which is composed of several weight vectors, an affine transformation matrix is generated. After transforming the matrix, we can get the **j** prediction vector $\widehat{\boldsymbol{u}}_{\mathbf{j}|\mathbf{i}}$ of each low-level capsule **I** to a high-level capsule. On the level of possible

advanced capsules, the prediction vector $\hat{\boldsymbol{u}}_{j|i}=\mathbf{W}_{ij}\mathbf{u}_i$ is obtained by multiplying the weight matrix $\mathbf{W}_{ij}$ with the output $\mathbf{u}_i$ of low-level capsules. The prediction vector provides instance parameters for the capsule of high-level, and the higher-level capsule will be activated when the information provided by multiple prediction results is consistent.$\hat{\boldsymbol{u}}_{j|i}=\mathbf{W}_{ij}\mathbf{u}_i$

The low-level neural capsule **I** is connected with any high-level capsule **J** which has a "coupling" relationship with it. By multiplying the corresponding coupling coefficient $\mathbf{C}_{ij}$ with the **j** prediction vector $\hat{\boldsymbol{u}}_{j|i}$ of each low-level capsule **I** for the high-level capsule, and then weighted sum operation, the output $\mathbf{S}_j$ can be obtained. The output of the capsule in the next round is a high-dimensional vector $\mathbf{v}_j$, which is obtained by squash extrusion function on $\mathbf{S}_j$. The calculation formula is as follows.

$$S_j = \sum_i C_{ij} \cdot \hat{u}_{j|i} \tag{1}$$

For an intermediate layer capsule, the input is a vector and the output is also a vector, but the input process for it is two stages:

*1) Linear combination: For a linear combination of neurons, the connection weights between capsules are represented by a matrix instead of a scalar value in the form of a vector.*

*2) Dynamic routing:The core work of this stage is to determine the close relationship between high level **j** and low level **I**, that is to find the most suitable coupling coefficient value $\boldsymbol{C_{ij}}$, which is determined in the repeated process of dynamic routing algorithm.*

### *C. Dynamic routing algorithm*

The process of dynamic routing algorithm is as follows:

*a.* Softmax processes data
*b.* Predict the output
*c.* Weighted sum
*d.* Compress the vector
*e.* Update coupling coefficient

The following figure 3 is a description of the dynamic routing algorithm.

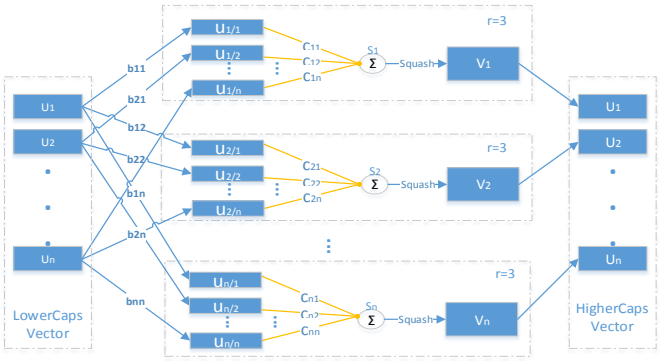


Figure 3. Dynamic routing algorithm

*1) The three input parameters are $\hat{\boldsymbol{u}}_{j|i}$ (prediction vector from **i** to **j**), **r** (number of iterations of routing algorithm) and **l** (number of layers of capsule).*

*2) For all layers of **l** capsules and (**l** + 1) capsules*

$$b_{ij} \leftarrow 0$$

For all layers of **l**capsules and (**l** + 1)capsules, the prior probability coefficient $\mathbf{b}_{ij}$ of two adjacent layers is initialized to 0, and its value will be used in the iterative update process of $\mathbf{b}_{ij}$. After the iteration, the value is stored in the corresponding $\mathbf{C}_{ij}$.

*3) Iteration with **r***

*4) The softmax rule is used to calculate the $\boldsymbol{C_{ij}}$ between the lower and higher layers.* In the beginning, because all $\mathbf{b}_{ij}$ are initialized to zero, so the obtained $\mathbf{C}_{ij}$ is also equal. That is to say, in this period, every node in the lower layer is equally important for the high-level capsule. The parent node at the higher level receives all the information from the lower level capsule. This kind of confusion in the initial stage of the algorithm will gradually become clearer in the later iterative calculation.

*5) The weighted sum of high-level capsules was calculated.* The weight of the combination used is the $\mathbf{C}_{ij}$ obtained in the previous step.

*6) $\boldsymbol{S_j}$ is a vector with a size and a direction.* However, if you want its length to be used as the probability of the existence of the entity, you need to normalize its size, and you need to use a nonlinear extrusion function to complete the normalization operation. This function retains the vector direction, and at the same time, the module length of the vector can be compressed within 1, so the output is the output of the high-level capsule

*7) The coupling between capsules is dynamic.* According to the formula, the larger the result value of $\hat{\boldsymbol{u}}_{j|i}\cdot\mathbf{V}_j$, that is, the more identical the pose information

of $\hat{\boldsymbol{u}}_{j|i}$ and $\mathbf{V_j}$,the greater the value of $\mathbf{b_{ij}}$, which indicates that the coupling degree between the previous layer capsule **i** and the high-level capsule **j** is higher

Dynamic routing algorithm focuses on clustering similar parts together, and then forms a larger granularity identification module. If the predicted vector $\hat{\boldsymbol{u}}_{j|i}$ and the output $\mathbf{V_j}$ of one of the high-level capsules are calculated by dot product and the result is very large, the relationship between nodes in this layer and high-level capsules will be strengthened after a reflection from front to back, that is to say, the coupling coefficient will be increased. At the same time, reduce the coupling coefficient with other high-level capsules. After **r** iterations, count the output of all high-level capsules, and determine the relevant routing parameters. The forward propagation will enter the next capsule layer of the capsule network.

### *D. loss function*

The traditional cross entropy function only supports the scenario of one classification, so this function is not suitable for capsule network. In order to distinguish multiple classifications in a picture, the edge loss function is used to achieve the objective function of model optimization for each digital capsule **k**. It is shown in the following formula.

$$L_k = T_k max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k)max\left(0, ||v_k|| - m^-\right)^2 \quad (2)$$

In the above formula, $\mathbf{T_k}$ is the function of classification, **k** is the classification, the value of $\mathbf{T_k}$ is related to the existence of the k-th classification, $\mathbf{L_k}$ is the calculated loss. If and only if the **k** classification exists, $\mathbf{T_k}$ is 1; if there is no **k** classification, $\mathbf{T_k}$ is 0.$\| \boldsymbol{v_k} \|$Represents the length of $\boldsymbol{v_k}$, which is the probability that the number **k** exists.$m^+$, $m^-$ are the threshold functions indicating the strength of the connection between the capsules .When it is lower than 0.1, it is considered that there is no connection relationship at all, and it is regarded as complete connection if it is higher than 0.9. In detail, $\boldsymbol{m}^+$ is the upper edge threshold, which is used to deal with the situation that the classification does not exist but exists in the prediction; $\boldsymbol{m}^-$ is the lower edge threshold, which deals with the situation that the classification does exist but is not predicted by the network.$\boldsymbol{\lambda}$ is called the sparsity coefficient and is used to adjust the weight between the two thresholds to adjust the parameters and steps. The values of $\boldsymbol{\lambda}$ are 0.9, 0.1 and 0.5. Add the loss $\mathbf{L_k}$ of each number to get the overall loss of the network.

## IV. EXPERIMENT

### *A. Experimental environment*

TABLE II. EXPERIMENTAL ENVIRONMENT

| | |
|---|---|
| Operating system | Windows10(RAM16.0GB） |
| CPU | Intel(R)Core(TM)i7-9750H |
| GPU | NVIDIA GeForce GTX 1660 Ti |
| Dataset | MNIST |
| Other | pytorch1.5.0+cu101 python 3.7.7 |

### *B. Experimental data analysis*

Handwritten digital machine vision database is widely used in image recognition and classification. The sample image in MNIST is 28 × 28 pixels, including four files: training set image, training set label, test set image and test set label. These files are binary files, each pixel of which is converted to a number between 0 and 255, where 0 is white and 255 is black. The training set has 60000 handwritten training samples. Its function is to fit model parameters, such as calculating offset and weight. The test set has 10000 samples, and its function is to test the final effect of the model.

*1) Precision of capsule network test*

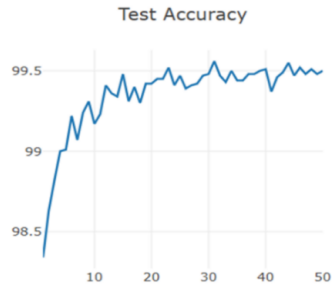


Figure 4. Test precision chart of capsule network under 50 epochs

As shown in Figure 4, the highest accuracy of this training is 99.55% in the 44th epoch.

*2) Test precision of deep capsule network*

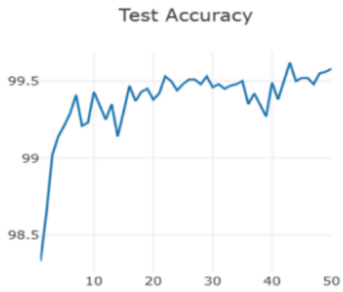


Figure 5. Test accuracy chart of deep capsule network under 50 epochs

As shown in Figure 5, the highest accuracy of this training is 99.62% in the 43rd epoch.

*3) When epoch = 30, the final test accuracy of capsule network is 99.46%*

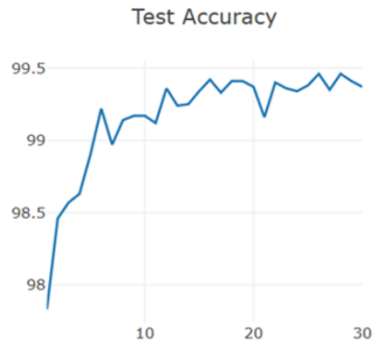


Figure 6. Test precision chart of capsule network under 30 epochs

*4) When epoch = 30, the final test accuracy of deep capsule network is 99.58%*

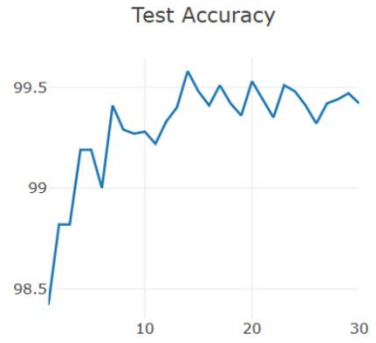


Figure 7. Test accuracy chart of deep capsule network under 30 epochs

*5) Accuracy comparison between deep capsule network and capsule network*

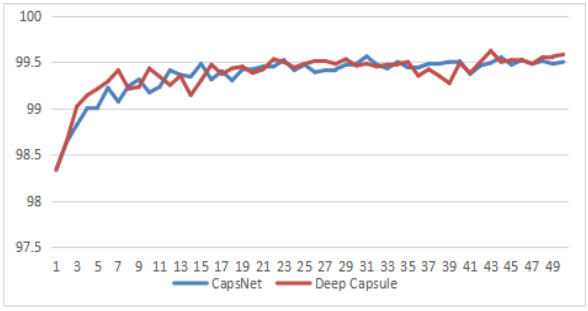


Figure 8. Comparison between the accuracy of capsule network and deep capsule network

As shown in Figure 8, it can be seen that the test accuracy of the deep capsule network in a short epoch increases faster than the accuracy of the capsule network and the recognition accuracy is also higher, under the same conditions.

*6) Performance of deep capsule networks with the same two routes: 1,2,3,4*

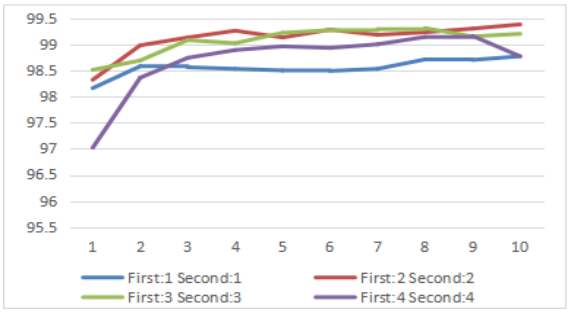


Figure 9. Impact of changing the number of routing iterations on the deep capsule network

As shown in the Figure 9, it shows that the number of route iterations is not the more the better, which should be obtained according to the specific experiment of network structure. In a smaller training period, It is more appropriate to select the number of iterations of the primary route 2 times and the number of iterations of the secondary route 3 times.

When the two routing iterations are different as to Figure 10.

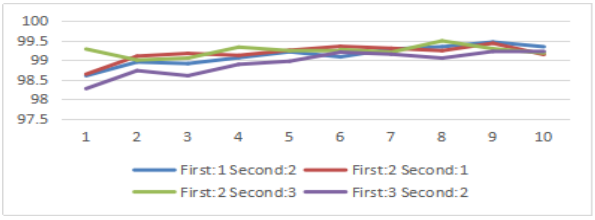


Figure 10. Influence of different iteration times of two routes on deep capsule network

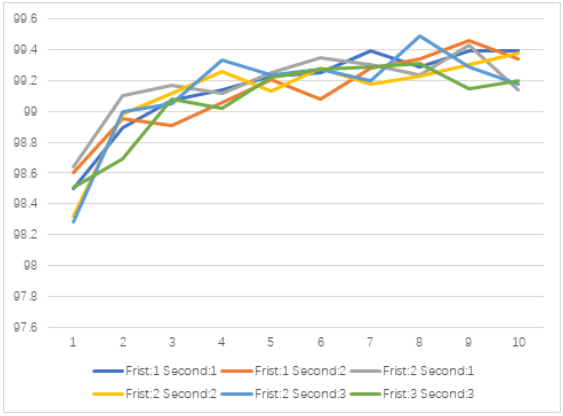


Figure 11. Influence of the same number of two routing iterations on deep capsule network

As shown in the Figure 11, the training time and classification accuracy of the network are compared under different collocation times of "primary route" and "secondary route". From the analysis of the data in the table, if only from the classification accuracy, the combination of "main route" iteration twice and "secondary route" iteration three times is the best, but the training time is long. If the training time and classification accuracy are considered comprehensively, the primary route is best to be iterated once and the secondary route is iterated twice.

*C. Reconstruction*

In order to understand the reconstructed picture, use the imshow function of matplotlib to draw and visualize, then the input picture 12 and the reconstructed picture 13 are shown in the following figure:

Figure 12. Schematic diagram of some pictures in MNIST database

Figure 13. Schematic diagram of reconstructed image

From the comparison, we can see that the reconstructed digital image is clearer and smoother than the input image. It can be inferred that the reconstructed image has the function of smoothing noise.

*D. Separation of overlapping handwritten numerals*

In the same way, we train the overlapped handwritten digital images with deep capsule network, and finally put the vectors into the decoder to decode the reconstructed images. Some of them are shown in figure 14, and the separation effect is basically accurate.


Figure 14. Comparison of input and output images of the network

Figure 15 shows the three separation effects' 0 'and' 1 ',' 3 'and' 4 ', ' 0 'and' 9 '. It is obvious that the network has been able to separate two completely coincident handwritten digits. Even if' 3 'and' 4 'overlap and it is difficult for human eyes to separate them, the network can still successfully separate them, with an accuracy rate of 93.53% .The accuracy rate of collaterals was only 88.10%.

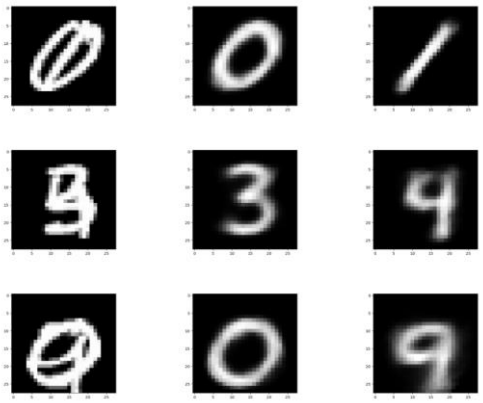

Figure 15. Partial reconstruction results of the improved network

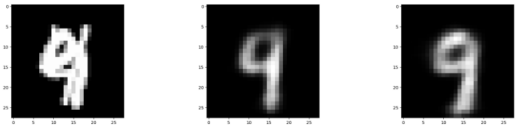

Figure 16. Improved partial error reconstruction results

However, the situation shown in Figure 16 still exists in the reconstructed picture. The original overlapping picture is the overlap of the numbers '9' and '4'. The two reconstructed images are like '9', without '4', the reconstruction is wrong, the error rate after the improvement is still 6.47%.

## V. CONCLUSION

The deep capsule network model in this paper is based on the characteristics and shortcomings of the capsule network. On the one hand, it retains the advantages of capsule network in understanding the attitude of objects; on the other hand, in view of the shortcomings of capsule network, the convolution kernel size of convolution layer is optimized, and the dynamic routing process is improved to twice routing. The final deep capsule network not only retains the advantages of traditional capsule network, but also improves the performance.

## REFERENCES


[1] Hinton G E, Krizhevsky A, Wang S D. Transforming auto-encoders[C]//International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2011: 44-51.

[2] 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.

[3] Hinton, Geoffrey E.; SABOUR, Sara; FROSST, Nicholas. Matrix capsules with EM routing. 2018.

[4] Kosiorek A, Sabour S, Teh Y W, et al. Stacked capsule autoencoders[C]//Advances in Neural Information Processing Systems. 2019: 15512-15522.

[5] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

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

[7] LeCun, Yann, Corinna Cortes, and Christopher JC Burges. "The MNIST database of handwritten digits, 1998." URL http://yann. lecun. com/exdb/mnist 10 (1998): 34.

[8] Netzer, Yuval, et al. "Reading digits in natural images with unsupervised feature learning." (2011).

[9] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.

[10] Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[C]//Advances in neural information processing systems. 2017: 3856-3866.

[11] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[12] Zhao W, Ye J, Yang M, et al. Investigating CapsuleNetworks with Dynamic Routing for Text Classification[J].arXiv preprint arXiv:1804.00538, 2018