IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 2, 2021, accepted July 26, 2021, date of publication August 3, 2021, date of current version August 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102489

An Improved Capsule Network Based on Capsule Filter Routing

WEI WANG[1,2], FEIFEI LEE[1,2], (Member, IEEE), SHUAI YANG[1,2], AND QIU CHEN[3], (Member, IEEE)

[1]Shanghai Engineering Research Center of Assistive Devices, School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]Rehabilitation Engineering and Technology Institute, University of Shanghai for Science and Technology, Shanghai 200093, China
[3]Major of Electrical Engineering and Electronics, Graduate School of Engineering, Kogakuin University, Tokyo 163-8677, Japan

Corresponding authors: Feifei Lee (feifeilee@ ieee.org) and Qiu Chen (q.chen@ieee.org)

This work was supported in part by the Programme for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and in part by the JSPS KAKENHI under Grant 15K00159.

ABSTRACT Capsule network (CapsNet) is a novel type of network that can retain spatial information, because each capsule can integrate more information than scalar-output features. However, the CapsNet learns all the features in the input image due to the lack of pooling operation, and there is no connection between different layers in the multi-layer network structure. In this paper, we propose an improved capsule network (CapsNet) based on Capsule Filter Routing (CFR) to address this problem, namely CFR-CapsNet. Firstly, we propose a new routing method called CFR for filtering capsules based on capsule activation values, which can speed up the operation of the model, and then introduce a self-attention mechanism to improve the performance of the primary capsule in the capsule space. Furthermore, in the multi-layer network structure, we transmit the information of the classified capsule with the largest activation value in the previous capsule layer to the primary capsules of the next layer, which improves the relevance of the overall structure. We conduct experiments on Fashion-MNIST, SVHN, and CIFAR-10/100, and the experimental results show that our method can improve the performance of the CapsNet more effectively than other state-of-the-arts.

INDEX TERMS Capsule network, capsule filter, self-attention, multi-layer network structure.

I. INTRODUCTION

With the emergence of a large number of images, deep learning has been widely applied in the field of computer vision, and there are many tasks in different directions, such as image classification [1]–[3], scene recognition [4], [5], object detection [6], [7], image segmentation [8], etc. Convolutional neural network (CNN) is one of the most important structures to solve these tasks. From the initial deep learning network model LeNet [9] to now, researchers have proposed many new CNN structures [10], and more variants are being proposed to achieve emerging visual tasks.

In a typical CNN, pooling can greatly reduce the computation, however, it is costly because pooling extracts the maximum or average value of the pixel value, which makes the network lose the specific location information of the target. Moreover, the network only learns to check whether the target exists in the input image, it does not pay attention to the specific spatial information of the target in the image. When facing different angle samples of the same object, humans form coordinate system to recognize the images and memory the pictures they learn, but the CNN needs to learn new parameters again to recognize. Although the structure of CNNs can complete the existing image tasks, convolution, as the basic architecture of deep learning image processing, still needs to be improved. Hinton proposed the concept of a capsule for the first time in 2011 [11]. It is suggested that in some special tasks similar to facial recognition, the networks need more attention to the spatial information of the target. The original capsule network (CapsNet) [12] uses a vector-output capsule instead of scalar-output feature detectors and uses dynamic routing instead of max-pooling, which solves the problem of lack of spatial information in a CNN.

The dynamic routing method for capsule information processing is similar to the max-pooling that a way to deal with the feature maps, it proves that routing method can be used as an effective way to transfer low-level capsule information although it has some disadvantage about that the training time and efficiency when dealing with a large number of capsules. Self-routing [13] is a novel routing that uses an extra matrix

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

to avoid iterative methods. For the input capsule, it also uses the convolution slide window method, which greatly speeds up the routing. But it ignores some edge capsule information and processes some information repeatedly.

The original CapsNet performs well on simple datasets, such as MNIST [14]. However, the convolution layer in the network uses two 9×9 convolution kernels, resulting in insufficient feature extraction. Also, many capsules represent an entity of background information, which mislead the results in the process of routing and affect the speed of dynamic routing. A multi-level dense CapsNet [15] proposes a new structure, which consists of three layers of dense blocks [16]. By connecting the blocks of each layer, it can extract richer features for complicated datasets and reduce a lot of parameters compared with ResNet [3].

In summary, the original CapsNet has the following shortcomings. First, the features extracted by the shallow convolutional network weaken the performance of the capsule. Secondly, the iterative calculation of the coupling coefficient of the routing algorithm makes the algorithm unable to converge, and the number of iterations artificially determined needs to be improved. Among the improvement research for the CapsNet, improving the processing efficiency of primary capsules is few, but it is crucial.

In order to better use the advantage of keeping space information of capsules, we propose two methods in the capsule layer to enhance the presentation ability of the primary capsule based on the multi-level dense CapsNet. Firstly, an Enhanced Capsule Attention Module (ECAM) is introduced when the feature maps reshaped into primary capsules. It assigns weight to different capsules based on the size of capsule activation value, for example, the capsules with larger activation value have greater weight. At the same time, we reassign the weight of each dimension of the capsule and scale the weight to prevent a change in the direction of the capsule vector. Previous attention methods [17]–[19] highlight more effective feature maps, while ECAM can show more active capsules in capsule space. Also, we introduce a mechanism about capsule information supplement between different capsule layer. In particular, we transfer the classified capsule from the upper layer to the primary capsules layer of the next layer instead of calculating the loss directly, and adjust the weight of other capsules in the routing process which can guide the routing direction of other capsules.

In the processing methods of many features in the network, most of the pruning methods are proposed to reduce the complexity of networks [20], [21]. Besides, there are some pruning methods to select features, such as dropout [22], which has big randomness. Our modified algorithm is a capsule filtering routing based on the pruning method. Specifically, the capsule filter is a way to delete the capsules with lower activation values adaptively. We combine the capsule filter with the self-routing [13] to avoid the repetition of capsule information transmission, and the routing uses more useful capsules on the basis of global capsule information processing without iterative calculation.

Based on the combination of the previous methods, we propose a routing method based on activation value filter, which deletes some fewer effective capsules. The enhanced capsule attention module (ECAM) is introduced into capsule space to improve the performance of the primary capsule. Meanwhile, a new capsule supplement mechanism (CSM) is proposed to establish the relationship between capsule layers. The major contributions of this work are as follows:

1) We propose an improved capsule network (CapsNet) based on Capsule Filter Routing (CFR), namely CFR-CapsNet.
2) We propose a Capsule Filter Routing (CFR) based on activation value, and the filtering proportion can be adjusted automatically for different input images. The combination of the filtering method and routing improves the efficiency of the network.
3) The self-attention mechanism is introduced in the capsule space to improve the performance of the primary capsule. And the weight distribution method is based on the size of the activation value.
4) We propose a new Capsule Supplement Mechanism (CSM), which combines the classified capsule formed by the integration of low-level information with the primary capsule of high-level information to enhance the connection between different capsule layers.
5) We combine the proposed methods and new routing in the multi-layer network structure to experiment on different classification datasets, and the results show that our routing method is more efficient than other routing methods, and the two methods proposed in the network also improve the accuracy.

The rest of the paper is organized as follows. Section II introduces related works. In Section III, we describe our structure and routing methods in detail. Section IV shows the results of the experiments. Finally, Section V concludes the paper.

II. RELATED WORKS

A. CAPSULE NETWORKS

Capsule network (CapsNet) with dynamic routing is firstly introduced by Sabour *et al.* [12]. This makes many researchers realize the shortcomings of CNNs and have a new research direction.

A complete CapsNet mainly includes feature extraction of convolution layer and capsule information transmission of capsule layer. However, in recent years, there are different promotion ideas for the original CapsNet in order to adapt to more complex tasks. In convolutional layer, the capsule is obtained based on the feature maps that formed by the convolution operation of the input images. Many improved CapsNets add new convolution modules, such as the introduction of DenseNet [15], Res2Net [23] and U-Net [24], to enhance the feature extraction. Besides, the design of multi-scale method can also increase the effect of the feature extraction, Xiang *et al.* [25] use multi-scale feature extraction to transform the feature maps into different dimensions of the primary capsule.

In the capsule layer, the primary capsules are processed by some methods, such as pruning [26] or merging [27], to reduce the calculation cost of routing and accelerate capsule processing [28]. In addition, the representation ability of capsules also promoted by the attention modules [29], the space promotion module [30], and several methods of interaction between capsules [31]. What's more, the definition and the equivariance of capsules are reconsidered [32], [33].

Dynamic routing is an iterative algorithm that constantly correct the weight distribution of each capsule, it can process the information of primary capsules and generate classification capsule, but it is not efficient enough. Based on the iteration methods to train a coupling coefficient, Rajasegaran *et al.* [34] propose a new dynamic routing based on 3D convolution, which predicts each capsule 3D convolution operation. Mandal *et al.* [35] divide dynamic routing into two steps to calculate the consistency between different levels of capsules. Also, the attention mechanism [36], [37] is introduced into the routing to deal with the primary capsules which have more information. Some details in dynamic routing, for example, the calculation method of coupling coefficient [28], [38], the consistency calculation between parent and child capsules are improved to handle capsules more effectively [39], [40].

The reconstruction part plays a regularizing role in the original CapsNet, but the parameters introduced by fully connected layers are too much. Rajasegaran *et al.* [34] reconstruct the classification capsule with the largest module length, which greatly reduces the number of full connection parameters. In [41], they introduce reconstruction into routing as a constraint.

CapsNet also has many applications in other directions [42]–[44]. Such as few-shot learning [45], unsupervised learning [46], GAN [47] and so on. In [48], they use improved dynamic routing for legal judgment of predicting charges. Yang *et al.* [49] use CapsNet to help graph network better extract hierarchical graph features.

B. DYNAMIC ROUTING AND SELF-ROUTING

Hinton uses dynamic routing [12] to replace the max-pooling for delivering the information of capsules. The vector-output capsule, where the direction of the vector represents the properties of the entity, and the length of the vector represents the probability of the entity. In the initial stage of dynamic routing, all child capsules in the lower layer i have equal connection probabilities b_{ij} with the next layer j. The coupling coefficient c_{ij} is the normalization of the connection probability in the low-level capsule layer. Then, the parent capsule is multiplied by the predicted capsule and the coupling coefficient. Also, the b_{ij} and c_{ij} are updated by the agreement between the parent capsules and the prediction capsules for k iterations. Multiple iterations can find a more accurate coupling coefficient, but it also increases the computation and reduces the efficiency.

Self-routing [13] uses capsules that separately defines the activation scalar. At the same time, it abandons the iteration way and uses an additional weight matrix to train the coupling coefficient. The obtained coupling coefficient is used to calculate the direction, size, and corresponding activation value of the high-layer capsule. The method about slide window is used for partial selection to the generated convolution block. Each convolution block with corresponding depth is regarded as multiple capsule blocks, and each capsule block get own classified capsule. Finally, the high-level capsule and the corresponding activation value are generated. The convolution slide window speeds up the routing, but it performs locally, and lacks of connection between capsules.

III. PROPOSED METHODS

A. NETWORK ARCHITECTURE

The structure of our proposed multi-layer network CFR-CapsNet is shown in Figure 1. The whole structure is based on a three-layer dense CNN structure [15]. The feature maps obtained by each layer are reshaped to get the primary capsule, and the classified capsules are obtained by routing. After getting the primary capsule, we firstly use the Enhanced Capsule Attention Module (ECAM) that will be described in Section III-A-(2)-a to operate on the primary capsule, the purpose is to adjust the weight of the capsule number dimension and the direction dimension. Then we filter the promoted primary capsules and use the Capsule Supplement Mechanism (CSM) that will be described in Section III-A-(2)-b to concatenate the classified capsules obtained from the upper layer with the primary capsules to obtain the classified capsules of this layer. After ECAM, it is more helpful for the implementation of Capsule Filter Routing (CFR) described in Section III-B-(2) and reducing the wrong deletion of some capsules. The sum of the margin losses of the last three layers is used as the total loss of the network for backpropagation.

1) CONVOLUTIONAL LAYERS

In our model, we use the same way as in [15] to extract the features in the input image, the difference is that the dimensionality of the capsules increases with the deepening of the convolutional layer. The specific convolution process and parameter setting are shown in Figure 2.

2) CAPSULE LAYERS

a: ENHANCED CAPSULE ATTENTION MODULE (ECAM)

Inspired by the self-attention and channel attention mechanisms in computer vision tasks, we introduce a new self-attention module to enhance the performance of capsules in the primary capsule layer.

Most attention mechanisms focus on the relationship between feature maps. Through training, we can get a feature map with a larger weight, so as to find the important feature

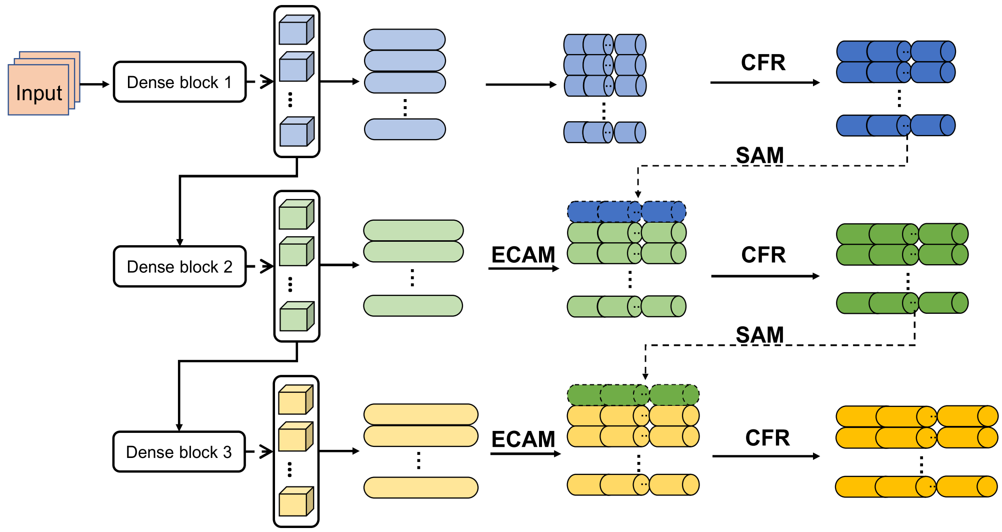

FIGURE 1. An overview of our CFR-CapsNet structure.

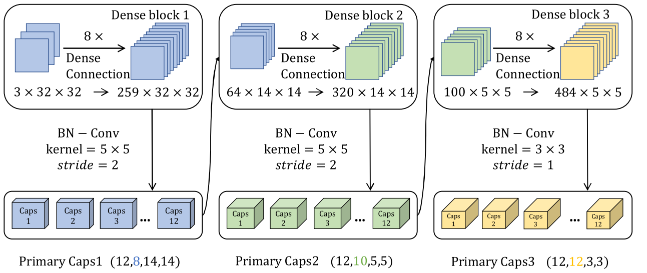

FIGURE 2. The convolutional network structure for CFR-CapsNet. The dimensions of the generated capsule blocks are expressed respectively (number of capsule blocks, dimension of capsules, height of capsule blocks, width of capsule blocks), for example, Primary Caps1 (12,8,14,14).

map. Similarly, the weight of capsule is different during routing. And each dimension in each capsule also has different weights to influence the direction of the capsule vector (the entity represented). So, we use a self-attention module based on capsule activation value to adjust the weight of different primary capsules, and introduce a new weight value to adjust the weight of different dimensions in the capsule for correcting the direction of the vector-output capsule. The structure diagram of ECAM is shown in Figure 3.

The primary capsule reshaped by feature maps is carried on matrix transformation for new capsule spaces Q, K. In capsule space Q, we compress its own dimension to 1 for each capsule.

$$u_Q = W_Q^{1\times d} u_i \tag{1}$$

And the capsules transformed into capsule space K are divided into two parts: one is used to calculate attention with the capsule in capsule space Q, and the other is used to adjust the weight of each dimension of the capsule.

$$u_{K1} = W_{K1}^{d\times d} u_i, \quad u_{K2} = W_{K2}^{d\times d} u_i \tag{2}$$

Then, we multiply the activation value (a_{K1}) of the capsule corresponding to u_{K1} by the capsules in the capsule space Q, and use SoftMax function to assign weight between the inter-capsules with a larger activation value. At the same time, we directly use the SoftMax function for u_{K2} to distribute the weight of each dimension of the intra-capsules.

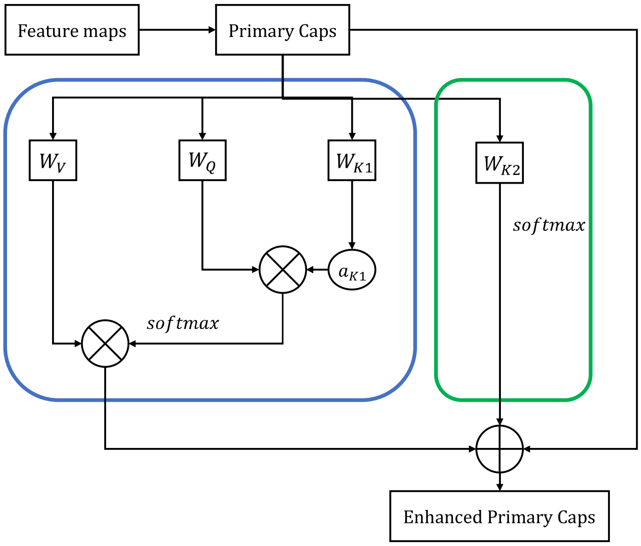

FIGURE 3. The proposed enhanced capsule attention module (ECAM). ⊗ and ⊕ represent matrix multiplication and addition of multiple items respectively. a_{k1} is calculated from the modules of the capsule transformed by W_{k1}. The blue box indicates that the inter-capsules attention mechanism for weight distribution, and the green box indicates that the intra-capsules methods for weight distribution.

$$att1 = SoftMax\left(u_Q \cdot a_{K1}\right) \tag{3}$$

$$att2 = SoftMax\left(u_{K2}\right) \tag{4}$$

After that, we can obtain enhanced primary capsules by combining the weight coefficient with the primary capsules.

$$u_V = W_V^{d\times d} u_i \tag{5}$$

$$\bar{u}_i = \rho \cdot u_V \cdot att1 + \sigma \cdot att2 + u_i \tag{6}$$

where ρ and σ are learnable scalar and they are initialized as 0. By using the ECAM for the primary capsules, the weight of the capsules in each layer can be reallocated to improve the effect of primary capsules.

b: CAPSULE SUPPLEMENT MECHANISM (CSM)

In the multi-layer network structure, the features that can be represented by the feature maps gradually change from partial to global. Although the capsules formed by the transformation of the deep feature maps have a stronger ability to express more information, the details in the shallow feature maps can also play a role in the classification. At present, many fusion mechanisms are proposed to represent the combination of the deep feature map and the shallow feature map, such as the feature pyramid network structure [50].

In a multi-level CapsNet, the capsules obtained from the shallow feature map can also play a considerable role in the classification, even if the final result is mainly based on the high-level capsule to show the global feature of the entity. So, in our model, the dimension of the capsule gradually increases because the feature represented by the feature map is more global with the deepening of convolution operation, and the dimension of the classified capsule of routing result is consistent with the dimension of the primary capsule in the next layer. In other words, the dimension of classified capsule in the previous layer is the same as the primary capsule of the next layer.

We propose a way to supply the capsule information, which combines the capsule with the largest module value in the classified capsule of the previous layer with the primary capsules of the next layer to implement routing together. The specific combination of CSM is shown in Figure 4. In the process of combination, we train a weight coefficient α to enlarge or reduce the classified capsule of the previous layer l, so as to improve the effect of combination with the primary capsule.

$$u_{\mathrm{n}}^{l+1} = u_m^{l+1} + \alpha \cdot v^l \tag{7}$$

where $(l+1)$ refers to the higher layer, u and v are the primary capsules and classified capsules, m and n are the number of capsules in the layer.

B. ROUTING ALGORITHM

1) CAPSULE FILTER

The primary capsules are reshaped by the feature map obtained by the convolution operation of the input image. We use the feature maps of different channels at the same position to obtain a capsule. It is proved that such capsule formation method is more effective than other methods of capsule expression in [27]. The length of the capsule determines the probability of the appearance of the entity. By reshaping all the primary capsules and screening the activation value of all the primary capsules, a certain proportion of the primary capsules with higher activation value can be selected. We call the way dealing with capsules as CapsFilter.

It has been proven that only a part of the primary capsules works in routing [26]. However, most of the current filter ratios are determined by experimental results, which have great uncertainty in different datasets. So, we propose a new method to choose an appropriate proportion. For example, in our model for CIFAR-10, the number of capsules in the three-layer capsule is 2352, 300, and 108, respectively. Due to the large difference in the activation value of capsules, the average value of the activation value is very small. In a group of data, median is the middle representative value of all data, and has low sensitivity which means that is not affected by the maximum or minimum value of data distribution. Therefore, we sort the activation value of each capsule and take the median of the activation value as the basis of screening, and we divide it by the maximum activation value to ensure that it is a proportional value within (0, 1). The proportion of screening increases when the activation value represented by the median is closer to the maximum activation value. The filtering ratio is adjusted adaptively according to each input image, so that the computational pressure of the model is not increased. Although most of the proportion can be calculated to a specific range, we set the scope of application of the ratio artificially in order to prevent too few capsules from affecting the accuracy of the training process. The specific steps of our CapsFilter are shown in Figure 5.

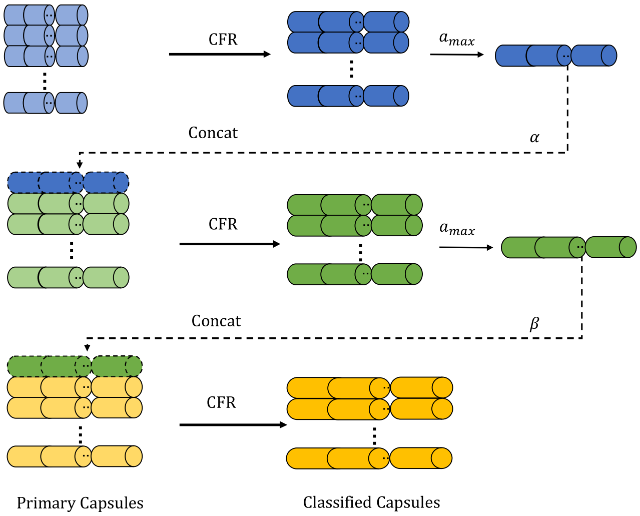

FIGURE 4. Supplement of upper layer classified capsule to next layer primary capsule. a_{max} is the maximum activation values in classified capsules. α and β are trainable parameters used to expand or scale the size of the capsule.

$$r = \begin{cases} 0.1, & r < 0.1 \\ \dfrac{a_{med}}{a_{max}}, & 0.1 < r < 0.8 \\ 0.8, & r > 0.8 \end{cases} \tag{8}$$

where a_{med} and a_{max} represent the median and maximum activation values of capsules respectively.

2) CAPSULE FILTER ROUTING (CFR)

Inspired by the CapsFilter and the additional weights which are used to train the coupling coefficient, a new routing method CFR is proposed. The module value (a) of the capsule represents the size of the vector. In our routing, we use the activation value as the length of the capsule. At the same time, it can also represent the probability of the existence of the instance, which is more interpretable.

Algorithm 1 Capsule Filter Routing Algorithm

```
Input: $u_i$
Output: $v_j$
for all capsule $i$ in layer $l$:
    $u_k = CapsFilter(u_i)$
end for
for all capsule $k$ in layer $l$:
    $\hat{u}_{\mathrm{j}|k} = W^1_{kj} u_k$
    $c_{kj} = softmax(W^2_{kj} u_k)_j$
end for
$v_j = \frac{\sum_{k \in l} c_{kj} a_k \hat{u}_{\mathrm{j}|k}}{\sum_{k \in l} c_{kj} a_k}$
return $v_j$
```

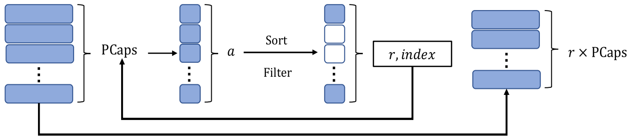

FIGURE 5. The process of screening a certain proportion of capsules. The "PCaps" means primary capsules.

For the input low-level capsules, as indicated in Algorithm 1, the proposed capsule filter method is used to screen the capsules, and some capsules with low activation value are deleted, which can eliminate some possible interference and also improve the speed of the routing process. we discard the square function in order to show the performance of capsule filter. Compared with the operation of the convolution slide window, the capsule with high activation value is more useful in routing process.

$$u_k = CapsFilter\ (u_i) \tag{9}$$

Then we divide the routing process into two steps, one step is the prediction: the primary capsule is predicted to get the high-dimensional capsule, and the coupling coefficient normalized by SoftMax is predicted by another matrix; the other step is routing: the high-level classified capsule is obtained from the generated predicted capsule, the module value of the original primary capsule and the predicted coupling coefficient.

$$v_j = \frac{\sum_{k \in \Omega_l} c_{kj} a_k \hat{u}_{j|k}}{\sum_{k \in \Omega_l} c_{kj} a_k} \tag{10}$$

Two transformation matrices are updated by backpropagation.

C. LOSS FUNCTION

We regard the digital capsule with the largest activation value as the final result of classification. The margin loss [12] can train the capsules which have a length for the true class. In our network, we use three margin losses for the existence of digit capsules.

$$L_k = T_k max\left(0, m^+ - \|v_k\|\right)^2 + \lambda\left(1 - T_k\right) max\left(0, \|v_k\| - m^-\right)^2 \tag{11}$$

$T_k = 1$ if the class k is present. We use $m^+ = 0.9$ and $m^- = 0.1$, and the λ is set to 0.5 to decrease the influence of false class.

IV. EXPERIMENTS

A. DATASETS

We conduct various experiments in four datasets to verify the performance of our model. We use Fashion-MNIST (FMNIST) [51] instead of MNIST, where the image size, the division of training set, and test set are consistent with MNIST. There are 60,000 training images and 10,000 test images, with an image size of 28 × 28. FMNIST is more complicated than MNIST because the content of the pictures are different clothing products. SVHN is also a digital dataset, which includes 73,257 training images and 26,032 test images. The numbers in the dataset images come from the number of street view house numbers. The realness of the image is stronger, which increases the difficulty of model recognition. CIFAR-10 [52] is a standard dataset used to test the effect of the model. It is composed of 60,000 images with a size of 32 × 32 × 3, of which 50,000 are used for training and 10,000 for testing. CIFAR-100 [52] is a more complex dataset than CIFAR-10. It has 100 classes, but each class corresponds to 600 images, 500 for training and 100 for testing. The number of images in each category is relatively small.

B. EXPERIMENTAL SETUP

In our model, we set the total number of epochs to 40, batch size to 50, and the initial learning rate to 0.001. Our experimental environment is PyTorch, and the whole network model is implemented on Tesla T4 with 16GB RAM in Google Colaboratory. The Adam [53] is used as our optimizer. We use the warm-up [54] method for the first five epochs. With the progress of training, the learning rate becomes 0.1 times the original after every 20 epochs. A data enhancement method is adopted for the input image, which is mainly to randomly flip the image horizontally. For each experiment, we conduct several experiments, and remove the maximum and minimum values in the results, then take the average value as the final result. In our model, we use the total loss of the three layers is trained as the loss of the model. When the loss is basically unchanged for 10 epochs, we stop training the model.

C. EXPERIMENTS RESULTS

1) CAPSULE FILTER

We compare the results before and after using CapsFilter in different datasets and find that when the accuracy reached a certain degree, the proportion of network screening capsules tends to be stable. The comparison of the corresponding scale, speed, and accuracy is shown in Table 2. In the Filter Ratio section, we also show the proportion of the first two layers, which can more intuitive to see the proportion of the selected capsules in different capsule layers.

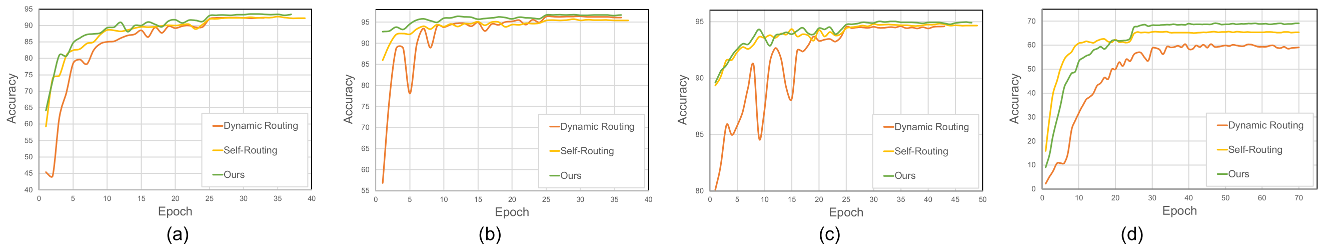

FIGURE 6. Convergence curves of CFR-CapsNet with different routing on CIFAR-10, SVHN, FMNIST and CIFAR-100. (a) CFR-CapsNet on CIFAR-10. (b) CFR-CapsNet on SVHN. (c) CFR-CapsNet on FMNIST. (d) CFR-CapsNet on CIFAR-100.

TABLE 1. Speed and effect comparison of different routing on different datasets. (s/E means second/epoch).

Routing Method	Time(s/E)/ CIFAR-10	Time(s/E)/ SVHN	Time(s/E)/ FMNIST	Time(s/E)/ CIFAR-100
Dynamic Routing	444/92.42%	608/96.36%	303/94.58%	1920/60.42%
Self-routing	330/92.69%	504/95.68%	245/94.81%	333/65.59%
CF-routing	**227/93.49%**	**335/96.75%**	**185/95.03%**	**280/69.16%**

TABLE 2. Comparison of capsule filter method on CIFAR-10, SVHN, FMNIST and CIFAR-100. (s/E means second/epoch).

Datasets	Filter Ratio	Time(s/E)	Test Accuracy
CIFAR-10	1/1/1	331	93.36%
	0.18/0.17/0.3	227	93.49%
SVHN	1/1/1	504	96.33%
	0.13/0.15/0.26	341	96.75%
FMNIST	1/1/1	261	94.97%
	0.15/0.16/0.28	185	95.03%
CIFAR-100	1/1/1	509	65.03%
	0.18/0.27/0.37	270	69.16%

We delete some capsules with low activation values, which represents low probability existence of an entity, such as some capsules describing the background in the input image. On four different datasets, it can improve the running speed by 40% - 50%, and the classification accuracy can reach the same or even higher accuracy that without this method.

Table 1 shows the classification accuracy of three different routing methods and the running speed of the model under the same network structure in different datasets. It can be found that our routing method reduces the training time by 20% - 40%. On the CIFAR-10, the accuracy is improved by 1.07% compared with dynamic routing, and 0.8% higher than self-routing.

Our routing method has been significantly improved on four datasets. Dynamic routing needs multiple iterations to get a more accurate coupling coefficient, so the training time is longer. Self-routing removes the iterative method, and the convolution slide window method speeds up the training speed. However, the global feature representation still needs to be strengthened because the way of extracted features is still local. Our routing method takes advantage of self-routing and operates all input capsules at the same time. It can not only operate based on all capsules but also reduce iterative training and speed up training.

Next, we show the results of the third capsule layer of the three routing methods on different datasets. In Figure 6, it can more intuitively reflect the effects of different routing methods under the same network structure and the same training method. It can be seen from the figure that the dynamic routing has low starting accuracy and large oscillation. In contrast, our routing method is more stable, which is improved based on self-routing.

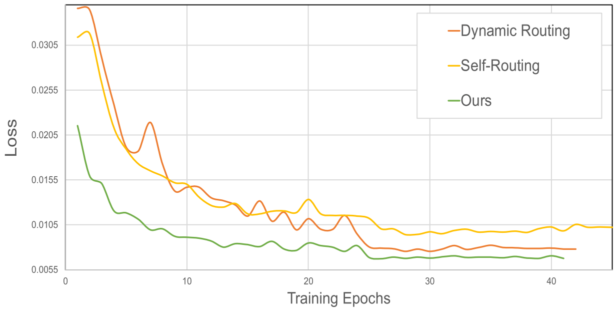

FIGURE 7. The loss curves of the three routing methods on CIFAR-10.

The training loss curves of the three routes in CIFAR-10 are also shown in Figure 7. It can be seen that our routing method has a smaller training loss and a more stable decline rate compared with the other two routing methods.

2) EVALUATION ON CIFAR-10

a: RESULTS OF ECAM

We apply the Self-Attention mechanism in our model and add a weight to adjust the weight of the internal dimensions of the capsule. Our ECAM redistributes the weight of capsule quantity and direction dimension so that the network can pay more attention to the capsules with more information.

In Table 4, we compare the difference between the network with and without ECAM modules. Since 2,352 capsules are generated in the capsule layer of the first layer, the module significantly increases the training time. We also conduct a comparative experiment that uses the module only in the second and third layer, and the results show that the accuracy is improved when the training time is basically the same as without ECAM.

b: RESULTS OF CSM

The proposed CSM supplies the information from the classification capsule with the largest activation value in the upper layer to the capsule in the next layer. At the same time, the classification capsule with the largest module value can

TABLE 3. Effect comparison of our proposed methods on CIFAR-10. "✓" means that we use this method. "Para" is the parameter of the network. The "C10" represents the performance in the CIFAR-10, and the "C10+" adds data enhancement method that randomly flip the image horizontally, the "Warm-up" represents the warm-up training method, the "All-TM" represents all the training methods includes the data enhancement and the warm-up method. (M means millions).

Proposed Methods			Para.	C10	C10+	Warm-up	All-TM
CFR	ECAM	CSM					
✓			3.63M	90.66%	92.81%	91.19%	93.49%
✓	✓			90.87%	93.1%	91.46%	93.7%
✓		✓		90.75%	92.8%	91.23%	93.6%
✓	✓	✓		91.05%	93.15%	91.53%	93.88%

TABLE 4. Comparison of our ECAM on CIFAR-10. The number in () denotes the layer where the ECAM module used. (s/E means second/epoch).

Method	Time(s/E)	Test Accuracy
CapsNet	227	93.49%
CapsNet + ECAM (2/3)	226	**93.7%**
CapsNet + ECAM (1/2/3)	389	93.58%

TABLE 5. Effect comparison of different CSM on CIFAR-10. The numbers in the table indicate the number of capsule layers, and 1→2 indicate that the subcategory capsule of the first layer is connected with the primary capsule of the second layer. (s/E means second/epoch).

Method		Time(s/E)	Test Accuracy
CapsNet		227	93.49%
CapsNet + Add		235	93.29%
CapsNet + Concat	1→2, 2→3	236	**93.6%**
	1→ 2, 2→3, 1→3	267	93.43%

be backpropagated to update the parameters in the training process. In Table 5, the experimental results show that CSM improves the performance of the capsule after supplement. In addition, we test that concatenating the classified capsule in the first layer with the primary capsules in the third layer as a comparative experiment, the results show that the accuracy is basically unchanged, there may be problems in parameter updating after two repeated supplements of the third capsule layer. We also try to change the operation mode of concatenating to add operation, and the experiment results show that it weakens the performance of the network.

c: EFFECTIVENESS OF TRAINING METHODS

Through the above experiments, we select the method of corresponding structure, and do experiments on CIFAR-10 to compare the effect of our proposed methods in different training methods condition. In Table 3, the data enhancement method has a significant improvement for our network, and the average accuracy improvement is 2%. The warm-up method also increases the potential capacity of the network. The improvement of CSM is limited, and the reason may be that some supplementary capsules do not have enough information compared to high-level primary capsules.

d: COMPARISON WITH DIFFERENT CAPSULE LAYERS

We apply our proposed method to a multi-layer network structure. The precision curve of each layer is shown in Figure 8. On the CIFAR-10, the accuracies of the L1, L2, L3 layers are 88.69%, 93.16%, 93.88%, respectively. The network achieves the highest accuracy in approximately 35 epochs. Compared with the previous multi-layer CapsNet [15], the capsule performance of each layer is improved. Specifically, the accuracy of each layer increases by 0.6%, 2.3% and 1.7%, respectively. Furthermore, the training of our model is also more stable under the same CNN structure.

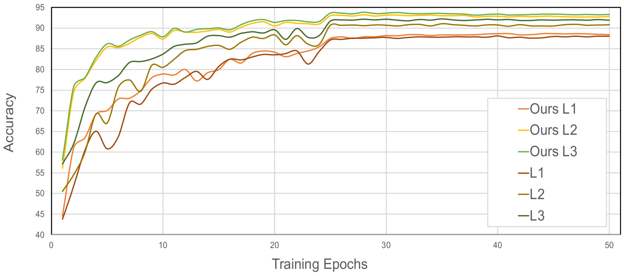

FIGURE 8. Statistics of the loss for three-level classified capsule layers on CIFAR-10.

e: METHODS WITH DYNAMIC ROUTING

We propose two methods in network structure, ECAM and CSM. Under the same convolution conditions, we combine two methods with the dynamic routing method in our network. In Table 7, the experimental results show that the two methods can not only improve the effect of classification, but also speed up the running speed of the model. Dynamic routing can allocate weight to the primary capsule according to the update of the coupling coefficient in iteration. The use of ECAM improves the effect of weight allocation and the CSM also improves the routing performance in the next capsule layer.

3) COMPARISON WITH STATE-OF-THE ARTS

We integrate all the proposed methods into the model for training. To show the effectiveness of our module, the original CapsNet, three-level CapsNet, self-routing CapsNet, and some recent improvements of the CapsNet are shown in Table 6. We unify the three convolutional layers of networks, proving that our method is independent of the improvement of the convolution effect. The reason for the difference in parameters is mainly that weights are not shared in dynamic routing. In the table, "-" means data not given in the original paper. Our CFR-CapsNet achieves 95.03%, 96.88% and 93.88% accuracy on FMNIST, SVHN and CIFAR-10, respectively. Obviously, our routing method is more efficient than dynamic routing and self-routing. Due to the increase of CIFAR-100 image categories, we appropriately increase the extraction of the convolutional layer and finally achieve 71.18% accuracy on CIFAR-100. The Group Feedback CapsNet has higher accuracy than our network, but it uses more parameters.

TABLE 6. Classification performance comparison on FMNIST, SVHN, and CIFAR-10/100. (M means millions).

MODEL	PARAMS	FMNIST	SVHN	CIFAR-10	CIFAR-100
CAPSNET	7.99M/31.5M	91.47%	95.7%	75.20%	49.8%
MULTI-LEVEL CAPSNET	13.4M/37.05M	94.65%	96.9%	89.71%	58.75%
SELF-ROUTING	3.2M	-	96.88%	92.14%	65.05%
CV-CAPSNET [55]	2.69M	94.4%	-	86.7%	-
DEEPCAPS [34]	8.5M	92.24%	97.16%	91.01%	-
MASK DR [24]	-	93.68%	-	90.01%	59.95%
DE-CAPSNET [30]	0.95M/**0.99M**	94.25%	-	92.96%	-
INVERTED DOT-PRODUCT AR [56]	**0.56M**/1.46M	-	-	85.17%	57.32%
GROUP FEEDBACK CAPSNET [27]	8.1M		**98.32%**	**95.44%**	**79.99%**
CPR-CAPSNET (OURS)	3.63M/5.56M	**95.03%**	96.88%	93.88%	71.18%

TABLE 7. Accuracy comparison of two methods combined with dynamic routing. "✓" means that we use this method. (s/E means second/epoch).

Method			Time(s/E)	Test
DR	ECAM	CSM		Accuracy
✓			444	92.42%
✓	✓		388	92.73%
✓		✓	422	92.66%
✓	✓	✓	433	92.79%

V. CONCLUSION

In this paper, we propose an improved capsule network (CapsNet) namely CFR-CapsNet, which is based on a new routing method called Capsule Filter Routing (CFR) and two methods designed for capsules in a three-layer capsule network. Our CFR deletes some capsules with lower activation values and improves the efficiency of the network. The proposed Enhanced Capsule Attention Module (ECAM) allocates the weight of each capsule between and inside the capsule dimension, which helps to distinguish the importance of capsules; and the Capsule Supplement Mechanism (CSM) connects different capsule layers, which is helpful for the routing efficiency of the next capsule layer. By analyzing the multi-layer convolution structure and the capsule layer, we carry out a lot of experiments on FMNIST, SVHN, and CIFAR-10/100. Our methods significantly reduce the training time of the model and improve the accuracy. The experimental results show that our routing method is better than dynamic routing and self-routing, and the two proposed modules can also perform well in other network structures. In future work, we will design own convolution structure to ensure more efficient feature extraction with fewer parameters.

ACKNOWLEDGMENT

(Wei Wang and Feifei Lee contributed equally to this work.)

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[2] S. Huang, F. Lee, R. Miao, Q. Si, C. Lu, and Q. Chen, "A deep convolutional neural network architecture for interstitial lung disease pattern classification," *Med. Biol. Eng. Comput.*, vol. 58, no. 4, pp. 725–737, Jan. 2020.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] L. Xie, F. Lee, L. Liu, K. Kotani, and Q. Chen, "Scene recognition: A comprehensive survey," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107205.

[5] L. Xie, F. Lee, L. Liu, Z. Yin, and Q. Chen, "Hierarchical coding of convolutional features for scene recognition," *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1182–1192, May 2020.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[7] J. Cai, F. Lee, S. Yang, C. Lin, H. Chen, K. Kotani, and Q. Chen, "Pedestrian as points: An improved anchor-free method for center-based pedestrian detection," *IEEE Access*, vol. 8, pp. 179666–179677, Sep. 2020.

[8] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[11] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming autoencoders," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 44–51.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[13] T. Hahn, M. Pyeon, and G. Kim, "Self-routing capsule networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 7656–7665.

[14] Y. LeCun, C. Cortes, and C. J. Burges. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[15] S. S. R. Phaye, A. Sikka, A. Dhall, and D. R. Bathula, "Multi-level dense capsule networks," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2018, pp. 577–592.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[17] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[18] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[20] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, pp. 1–18, 2017.

[21] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5058–5066.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] S. Yang, F. Lee, R. Miao, J. Cai, L. Chen, W. Yao, K. Kotani, and Q. Chen, "RS-CapsNet: An advanced capsule network," *IEEE Access*, vol. 8, pp. 85007–85018, 2020.

[24] J. Chen and Z. Liu, "Mask dynamic routing to combined model of deep capsule network and U-Net," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2653–2664, Jul. 2020.

[25] C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, "MS-CapsNet: A novel multi-scale capsule network," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1850–1854, Dec. 2018.

[26] T. Jeong, Y. Lee, and H. Kim, "Ladder capsule network," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 3071–3079.

[27] X. Ding, N. Wang, X. Gao, J. Li, X. Wang, and T. Liu, "Group feedback capsule network," *IEEE Trans. Image Process.*, vol. 29, pp. 6789–6799, 2020.

[28] V. M. do Rosario, E. Borin, and M. Breternitz, "The multi-lane capsule network," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1006–1010, Jul. 2019.

[29] W. Huang and F. Zhou, "DA-CapsNet: Dual attention mechanism capsule network," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, Dec. 2020.

[30] B. Jia and Q. Huang, "DE-CapsNet: A diverse enhanced capsule network with disperse dynamic routing," *Appl. Sci.*, vol. 10, no. 3, p. 884, 2020.

[31] R. Pucci, C. Micheloni, G. L. Foresti, and N. Martinel, "Deep interactive encoding with capsule networks for image classification," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 32243–32258, Nov. 2020.

[32] S. Venkatraman, S. Balasubramanian, and R. R. Sarma, "Building deep, equivariant capsule networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–19.

[33] F. D. S. Ribeiro, G. Leontidis, and S. Kollias, "Introducing routing uncertainty in capsule networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–14.

[34] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10725–10733.

[35] B. Mandal, R. Sarkhel, S. Ghosh, N. Das, and M. Nasipuri, "Two-phase dynamic routing for micro and macro-level equivariance in multi-column capsule networks," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107595.

[36] J. Choi, H. Seo, S. Im, and M. Kang, "Attention routing between capsules," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–9.

[37] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-CapsNet: Capsule network with self-attention routing," 2021, *arXiv:2101.12491*. [Online]. Available: http://arxiv.org/abs/2101.12491

[38] Z. Yang and X. Wang, "Reducing the dilution: An analysis of the information sensitiveness of capsule network with a practical improvement method," 2019, *arXiv:1903.10588*. [Online]. Available: http://arxiv.org/abs/1903.10588

[39] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2018, pp. 1–29.

[40] L. Zhao, X. Wang, and L. Huang, "An efficient agreement mechanism in CapsNets by pairwise product," in *Proc. 24th Eur. Conf. Artif. Intell.*, 2020, pp. 1722–1729.

[41] S. Zhang, W. Fan, and X. Wu, "Towards capsule routing as reconstruction with sparsity constraints," *Pattern Recognit. Lett.*, vol. 140, pp. 193–199, Dec. 2020.

[42] T. Liu, X. Lin, W. Jia, M. Zhou, and W. Zhao, "Regularized attentive capsule network for overlapped relation extraction," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6388–6398.

[43] H. Lin, F. Meng, J. Su, Y. Yin, Z. Yang, Y. Ge, J. Zhou, and J. Luo, "Dynamic context-guided capsule network for multimodal machine translation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1320–1329.

[44] M. Edraki, N. Rahnavard, and M. Shah, "Subspace capsule network," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 10745–10753.

[45] F. Wu, J. S. Smith, W. Lu, C. Pang, and B. Zhang, "Attentive prototype few-shot learning with capsule network-based embedding," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 237–253.

[46] S. Sabour, A. Tagliasacchi, S. Yazdani, G. E. Hinton, and D. J. Fleet, "Unsupervised part representation by flow capsules," 2020, *arXiv:2011.13920*. [Online]. Available: http://arxiv.org/abs/2011.13920

[47] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, "CapsuleGAN: Generative adversarial capsule network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 526–535.

[48] Y. Le, C. He, M. Chen, Y. Wu, X. He, and B. Zhou, "Learning to predict charges for legal judgment via self-attentive capsule network," *Frontiers Artif. Intell. Appl.*, vol. 325, pp. 1802–1809, May 2020.

[49] J. Yang, P. Zhao, Y. Rong, C. Yan, C. Li, H. Ma, and J. Huang, "Hierarchical graph capsule network," 2020, *arXiv:2012.08734*. [Online]. Available: http://arxiv.org/abs/2012.08734

[50] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[51] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: http://arxiv.org/abs/1708.07747

[52] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[54] A. Marchisio, B. Bussolino, A. Colucci, M. A. Hanif, M. Martina, G. Masera, and M. Shafique, "FasTrCaps: An integrated framework for fast yet accurate training of capsule networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[55] X. Cheng, J. He, J. He, and H. Xu, "Cv-CapsNet: Complex-valued capsule network," *IEEE Access*, vol. 7, pp. 85492–85499, 2019.

[56] Y. H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, "Capsules with inverted dot-product attention routing," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–15.

WEI WANG received the B.S. degree in weapon system and engineering from North University of China, Shaanxi, China, in 2019. He is currently pursuing the master's degree in control science and engineering with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include computer vision and deep learning.

FEIFEI LEE (Member, IEEE) received the Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2007. She is currently a Professor with the University of Shanghai for Science and Technology. Her research interests include pattern recognition, video indexing, and image processing.

SHUAI YANG received the M.S. degree in control science and engineering from the University of Shanghai for Science and Technology, in 2021. His research interests include computer vision and deep learning.

QIU CHEN (Member, IEEE) received the Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2004. Since then, he has been an Assistant Professor and an Associate Professor with Tohoku University. He is currently a Professor with Kogakuin University. His research interests include pattern recognition, computer vision, information retrieval, and their applications. He serves on the editorial boards for several journals, as well as committees for a number of international conferences.