IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 9, 2020, accepted August 19, 2020, date of publication August 25, 2020, date of current version September 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019432

A Neural Network Model Compression Approach Based on Deep Feature Map Transfer

ZHIBO GUO[1], XIN YAO[1], YIXUAN XU[2], YING ZHANG[1], AND LINGHAO WANG[1]
[1]School of Information Engineering, Yangzhou University, Yangzhou 225009, China
[2]School of Computer Science, University of Nottingham, Nottingham NG8 1BB, U.K.

Corresponding author: Zhibo Guo (zbguo@yzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, and in part by the Science asnd Technology Project of Jiangsu Provincial Archives Bureau under Grant 2018-10.

ABSTRACT Neural network is widely used in computer vision. However, with the continuous expansion of the application field, high-precision large parameter neural network model is difficult to deploy on small equipment with limited resources. In order to obtain a small but efficient network, the soft output of the teacher network was used to train students through the teacher-student structure. A new method of neural network model compression based on deep feature map transfer (DFMT) is proposed in this paper, which uses visual system characteristics adequately. A small decoder is designed in the network to generate a deep feature map from the features extracted by the network, and the feature map is used to transfer knowledge. In addition, cosine similarity is used as the evaluation index of knowledge transfer. A smaller model with better precision can be obtained by the proposed method. Experiments on benchmark datasets prove the validity and advancement of the proposed approach.

INDEX TERMS Knowledge distillation, machine learning, model compression, neural networks, pattern recognition, transfer learning.

I. INTRODUCTION

As a typical realization of brain-like intelligence, neural network simulate the information processing patterns in human brain by employing broadly interconnected method and effective training mechanisms, which is becoming the preferred approach for artificial intelligence. From the M-P neuron and Hebb learning method in the 1940s, to the BP (Back Propagation) algorithm and Fully-Connected Neural Network in the 1980s, to the Deep Belief Networks (DBN) [1] and Convolutional Neural Network (CNN) [2], during the course of development, neural networks once received ignorance and doubts, but also achieved many brilliant achievements.

With the continuous improvement of neural network technology, the development of artificial intelligence has reached a new peak. As the main branch of artificial intelligence, computer vision [3], [4] has been widely concerned. By virtue of its perfect network structure, neural networks such as VGG [5], GoogLenet [6], ResNet [7],and DenseNet [8] have achieved outstanding results in many fields such as pattern recognition and computer vision. However, top-performing networks always have wide and deep network structures with many parameters in the models [9]. Therefore, much memory and time are spent to perform complex matrix operations at the inference time of the models.

The memory and calculation requirements of complex neural networks during training and prediction are different [10]. The memory footprint of parameter operation exceeds that the memory footprint of the model when the network is trained by large batch-sizes on big datasets. Training procedure of networks can be implemented in GPU cluster in a distributed calculation environment. Furthermore, the memory footprint of the model dominates the runtime memory requirements when the network is predicted by small batch-sizes. This method based on integration training to improve the accuracy of prediction have become inapplicable. Therefore, one of the biggest challenges is how to deploy the trained high-performance network model in the application system with limited memory and time.

Model compression has become a popular solution to the above challenges [11]. With compressing model, the training of a smaller low memory footprint neural network was guided by an advanced and complex deep neural network. Bucila *et al.* [12] used the integration outputs of the advanced

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

networks to label the data and trained the small network by the labeled data. Sujith [13] designed a joint optimization framework to train lighter networks. Hashing [14] was used to reduce the dimension of the extracted features, and the lighter network could learns knowledge from the full network. As the main framework of model compression, knowledge distillation (KD) [15] has been widely concerned recently. The knowledge extracted from the high-performance teacher network was transferred to the student network through the teacher-student structure. Student network was trained by learning true labels and the soft target labeled by teacher network. This method can simplify the deep complex neural network, and compress the deep and wide network into a small network with similar depth.

Much work has been done to improve the framework of KD, Zhou *et al.* [16] trained teacher network and student network at the same time. The teacher network could constantly provide the latest soft target to train the student network. This method of cooperative training greatly shortened the total training time. Anil *et al.* [17] proposed an online distillation method. The extra parallelism was used to train large datasets in the method, which increased the training speed and made the accurate prediction of the network repeatable. Wei *et al.* [18] effectively combines knowledge distillation with quantification. Quantization [19] can reduce the parameter search space and bring regularization effect, and KD is responsible for transferring knowledge from complex teacher network to student network. In the method of FitNet [20], the hint layer and the guided layer were defined as the teacher's middle layer and the student's middle layer respectively. The hint layer instructs the guided layer to train the student's initial parameters, and then the KD was used to train the whole network. Yim *et al.* [21] proposed an information measurement method, in which teachers used information measurement to transfer the extracted knowledge to students. Zagoruyko and Komodakis [22] proposed a method denoted as ''AT'', in which the spatial attention of the teacher network is regarded as knowledge and transferred to the student network.

For the same number of parameters, the deeper network always has better performance. Depth is very important for feature learning of neural networks. Neural networks encourage the reuse of features, and the deep layer learns features from the shallow layer to extract more abstract and complex features during the training phase. The expression of deep features is exponentially higher than that of shallow features in the function [23].

The network architecture of student network usually far deeper than the teacher network in the framework of knowledge distillation. KD achieves the effect of model compression by compressing the width of the teacher network. However, with the increase of network depth, the student network has the difficulty of optimization in knowledge transfer. Furthermore, KD does not make full use of the deep features of the teacher network. FitNet defined the teacher's middle layer as the hint layer and used the high-dimensional vector from the hint layer to train the initial parameters of the student network. Since using a high-dimensional vector to transfer knowledge, the optimization of student network is difficult and the training time excessively is long. In addition, FitNet require two-stage learning scheme, the training process is complicated.

A new method of neural network model compression based on deep feature map transfer (DFMT) is proposed in this paper. The proposed method only requires one-stage learning scheme. This method uses teacher-student structure, and makes the best of the depth information of the teacher network and the knowledge extracted from the deep hidden layer to train the student network. Small decoders are designed in teacher network and student network respectively, which generate a feature map with the most representative information from the high-dimensional vector extracted from the convolutional layers. Cosine similarity is used as the measurement function of knowledge transfer.

II. RELATED WORK

This section details the concept of knowledge distillation and FitNet, and how to guide the training process of the student network by using the knowledge extracted from the teacher network.

A. KNOWLEDGE DISTILLATION

Aiming to effectively compress the complex model, the KD method is proposed by Hinton, which trains a student network (simple, low-complexity) from the softened output of a teacher network (complex, efficient). The training of the student network is guided from soft targets and true labels in KD. The student network not only captures the accurate knowledge learned by the teacher network, but also acquires the category information by the true labels.

For simple classification tasks, complex networks can almost always complete tasks with high confidence. However, much of the valuable information about the complex network exists in the ratios of very small probabilities in the outputs. When directly using the class probabilities outputted by the teacher network as the soft targets, it has little effects on the cross-entropy cost function during the stage of knowledge transfer because the probabilities of valuable information are so close to zero.

Hinton solves this problem by using the inputs of the softmax [24] rather than the outputs of the softmax as the soft targets during the stage of knowledge transfer. In (1), the complex teacher network produces appropriate soft targets by increasing the temperature of the softmax output layer. The same temperature is used by the student network to mimic soft targets during the training stage.

$$\boldsymbol{q}_i = softmax\left(\frac{z_i}{t}\right) = \frac{exp\,(z_i/t)}{\sum_j exp(z_j/t)} \tag{1}$$

Temperature (t) is a tunable parameter was used by the teacher network to generate soft targets. z_i is the input of the softmax. The z_i of each class is converted into a probability

$\boldsymbol{q}_i$ by softmax. Using a higher value for the temperature can produces a softer distribution over soft targets. The cost function of KD is:

$$L_{\mathrm{KD}}(\boldsymbol{x}) = (1-\lambda)H\left(\boldsymbol{Q}_{\mathrm{T}}^{t}, \boldsymbol{Q}_{\mathrm{S}}^{t}\right) + \lambda H\left(\boldsymbol{y}_{\mathrm{ture}}, \boldsymbol{Q}_{\mathrm{S}}\right) \quad (2)$$

where λ is a parameter to balance the ratio of soft targets and true labels. H refers to the cross-entropy cost function. $\boldsymbol{Q}_{\mathrm{S}} = \left(\boldsymbol{q}_{\mathrm{S1}}, \boldsymbol{q}_{\mathrm{S2}}, \ldots . \boldsymbol{q}_{\mathrm{Sn}}\right)$ is the output of the teacher network; $\boldsymbol{Q}_{\mathrm{S}}^{t} = \left(\boldsymbol{q}_{\mathrm{S1}}^{t}, \boldsymbol{q}_{\mathrm{S2}}^{t}, \ldots . \boldsymbol{q}_{\mathrm{Sn}}^{t}\right)$ is the soft output of the student network by increasing the temperature of the softmax output layer; Similarly, $Q_{\mathrm{T}}^{t} = \left(q_{\mathrm{T1}}^{t}, q_{\mathrm{T2}}^{t}, \ldots . q_{\mathrm{Tn}}^{t}\right)$ is the soft output of the teacher network;

In (2), the first team refers to the cross-entropy between the soft targets and the class probabilities obtained by the student network raising the temperature. The second term is the cross-entropy between true labels and the outputs of the student network. The student network is forced to learn knowledge from true labels and the softened targets of the teacher network by minimizing (2).

B. FitNet

Adriana Romero trains a deeper, thinner student network than the teacher network by taking advantage of depth on the basis of knowledge distillation. The hint layer and the guided layer are defined as the teacher's middle layer and the student's middle layer respectively. The hint layer is responsible for the learning process of the student network.

Since the student network usually be thinner than the teacher network, the guided layer can't directly learn knowledge from the hint layer because it has fewer outputs than the hint layer. On this account, a regressor is added on top of the guided layer to match the dimension of the hint layer. Then, training the parameters up to the guided layer by minimizing the following loss function $L_{\mathrm{HT}}(\boldsymbol{W}_{\mathrm{G}}, \boldsymbol{W}_{\mathrm{r}})$:

$$L_{\mathrm{HT}}(\boldsymbol{W}_{\mathrm{G}}, \boldsymbol{W}_{\mathrm{r}}) = \frac{1}{2}\left\| u_{\mathrm{h}}\left(\boldsymbol{x}, \boldsymbol{W}_{\mathrm{hint}}\right) - r\left(v_{\mathrm{g}}\left(\boldsymbol{x}, \boldsymbol{W}_{\mathrm{guided}}\right), \boldsymbol{W}_{\mathrm{r}}\right)\right\| \quad (3)$$

where u_h and v_g are the activation function of the teacher and the student up to their hint layer and guided layer with parameters $\boldsymbol{W}_{\mathrm{hint}}$ and $\boldsymbol{W}_{\mathrm{guided}}$. r is the function of the regressor with parameters $\boldsymbol{W}_{\mathrm{r}}$.

The training of FitNet has two stages. The first stage is training the parameters up to the guided layer by minimizing (3), and the second stage is using KD to train the parameters of whole student network.

III. PROPOSED METHOD

In this section, a new method of neural network model compression based on deep feature map transfer (DFMT) is described, which makes the best of the knowledge of the hidden layer and the deep features of the teacher network to train the student network.

The complex teacher network produces appropriate soft targets by increasing the temperature of the softmax layer during the stage of knowledge distillation, and then the student network is trained to match the soft targets by the same temperature. For simple classification, the dimension of the soft targets is the number of categories, so that the soft targets can't fully reflect the knowledge extracted from the teacher network.

In order to train a thinner and lighter student network by making full use of the knowledge extracted from the hidden layer of deep teacher network, the last convolutional layer of teacher network is defined as the hint layer. Analogously, the guided layer is defined as the last convolutional layer of student network. It is hoped that the guided layer can mimic the output of the hint layer, and the output of the hint layer responsible for guiding the training process of the student network.

Given that teacher network usually adopts a wider network architecture to achieve better performance, the guided layer has fewer outputs than the hint layer when the last convolutional layer in teacher network is chosen as the hint layer. Beyond that, the output of the hint layer consists of deep features, which obtained from the last convolutional layer. Because the hint layer contains a large number of neurons, its outputs may have many redundant features. It's difficult to optimize student network when training the guided layer to mimic the output of the hint layer.

It has been demonstrated that different positions of the human retina have different information sensitivity and information processing abilities. In order to effectively use limited visual processing resources, human quickly select high-value details of visual information for observation and learning, and ignore other visible information when using vision to perceive external things [25]. This is a visual perception formed in the process of human evolution, which improves the efficiency and accuracy of human visual information processing. Therefore, DFMT is inspired by visual perception, and select the most valuable information from the output of the hint layer to generate a depth feature map. This depth feature map is used to replace the output of the hint layer for knowledge transfer.

A decoder is designed on top of the hint layer to select the most valuable information from the output of the hint layer, which can remove the redundant part of the hint layer and generate a high-value feature map from the features extracted from convolutional layers. Feature maps of different sizes are shown in Fig. 1. The same decoder is designed on top of the student's guided layer to matches the size of the hint layer.

Features $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2 \ldots, \boldsymbol{v}_n), \boldsymbol{v}_i \in R^c$ obtained by spatial convolution in the neural network over c channels with filter weights $\boldsymbol{W}^{c' \times c \times s \times s}$:

$$\boldsymbol{v}^{l}{}_{m,n} = f\left(\sum_{i}\sum_{j} \boldsymbol{W}_{i,j} \cdot \boldsymbol{x}^{l}{}_{i,m,n} + \boldsymbol{b}^{l}\right) \quad (4)$$

where $s \times s$ is defined as convolution window, l is the number of convolutional layers, $\boldsymbol{b}$ denotes biases, and f refers to nested function.

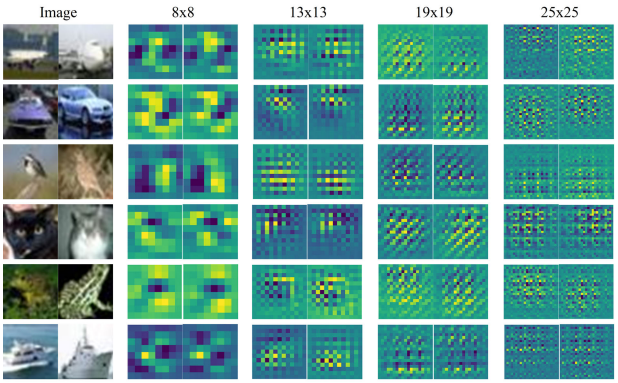

FIGURE 1. Feature maps of different sizes extracted by hint layer.

The purpose of decoder is to transform high-dimensional features into a feature map. The process can be described by the function:

$$\boldsymbol{P} = \sum_{k=1} \boldsymbol{F}_k \otimes \boldsymbol{Z}_k = \boldsymbol{F} \otimes \boldsymbol{Z} \tag{5}$$

where $\boldsymbol{F}(\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3 \ldots, \boldsymbol{F}_n)$ are the filters of decoder.

The parameters from input up to the decoder are trained by minimizing the loss function $L(\boldsymbol{x})$, and the update of parameters by standard backpropagation can be written as:

$$\boldsymbol{W}^e = \boldsymbol{W}^{e-1} - \alpha \frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{W}^{e-1}} \tag{6}$$

where α is the learning rate of backpropagation, e is the number of current iterations.

Fig. 2 represents the general structure of DFMT. When the student network completes the training from the input layer to the guided layer, the guided layer can predict the outputs of the hint layer. Because the structure is designed consistently, the parameters from the decoder to the output layer of the student network directly migrates the parameters of the teacher network. In this way, there is unnecessary to use KD to train the whole network and shortens the training time and cost.

As shown in Fig. 2, $\boldsymbol{x}$ and $\boldsymbol{y}$ denote input and true labels of the teacher/student network; $\boldsymbol{W}_{\mathrm{T1}}(\boldsymbol{x})$ and $\boldsymbol{W}_{\mathrm{S1}}(\boldsymbol{x})$ are the weightS from input layer to the decoder in the teacher network and student network respectively; $\boldsymbol{Z}_{\mathrm{T}}(\boldsymbol{x})$ and $\boldsymbol{Z}_{\mathrm{S}}(\boldsymbol{x})$ refer to the high-dimensional features before the decoder in the teacher network and student network; $\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})$ denotes the outputs of the hint layer. Analogously, $\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x})$ denotes the outputs of the guided layer to predict $\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})$.

In the proposed approach, since the parameters from input up to the decoder are trained by minimizing the loss function $L(\boldsymbol{x}) = f\,(\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x}), \boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x}))$, it's important to select an appropriate loss function in this work.

- knowledge distillation: $L_{KD}(\boldsymbol{x})$ as shown in (2).
- MSE of the hint layer and the guided layer:

$$L_{\mathrm{MSE}}(\boldsymbol{x}) = \|\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x}) - \boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x})\|_2^2 \tag{7}$$

- Cosine distance of the hint layer and the guided layer:

$$L(\boldsymbol{x}) = C(\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x}), \boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})) = \frac{\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x}) \cdot \boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})}{\|\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x})\| \cdot \|\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})\|} \tag{8}$$

In the method of KD, the complex teacher network produces appropriate soft targets by increasing the temperature of the softmax output layer, and then the training stage of the student network uses the same temperature to mimic soft targets. Since cross entropy is a measure of the similarity between two probability distributions, $L_{\mathrm{KD}}(\boldsymbol{x})$ doesn't apply to our approach.

The loss of MSE got by calculating the actual distance between two features in high dimension space. The contribution of each feature to euclidean distance is the same in the feature map. When the features have random fluctuations of

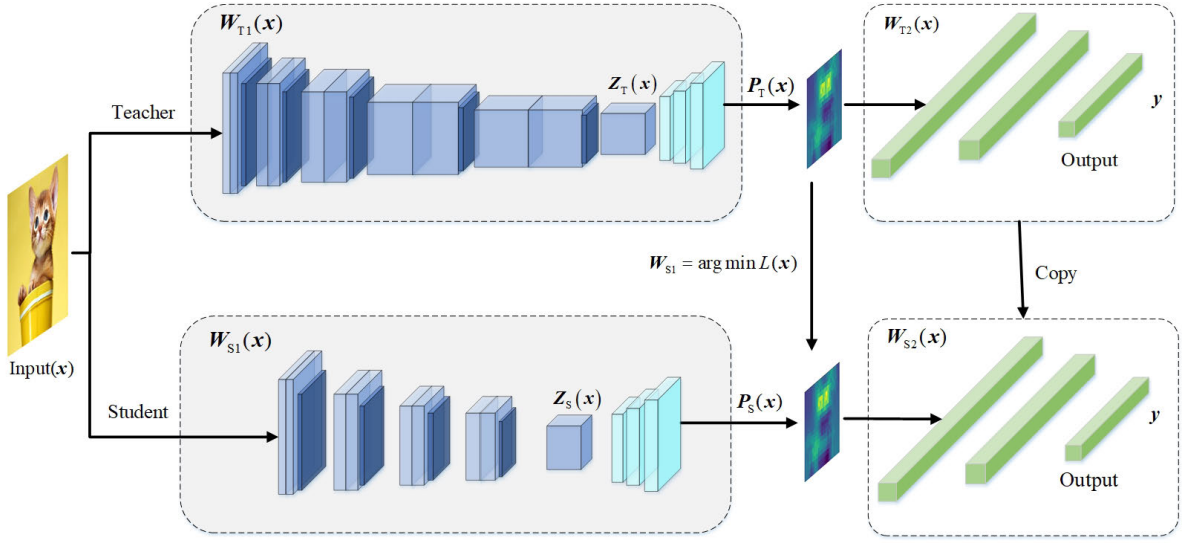

FIGURE 2. The architecture of the proposed approach.

different sizes, MSE can't reflect the influence of changes on distance.

Cosine similarity measures the similarity between two feature maps by measuring their cosine values. The two feature maps point in the same direction when the angle is close to zero, which proves that the two feature maps are similar.

To visualize the depth features in the process of knowledge transfer, an example on MNIST dataset is used to intuitively show the distributions of the depth features. The output of the last hidden layer is modified to 2. Since the dimension of the deep features is 2, the depth feature is directly plotted on 2-D surface. As shown in Fig. 3, the points of different colors refer to features of different classes. The features of different classes present radioactive distribution, which can be distinguished by decision boundaries [26]. Therefore, the knowledge extracted by the teacher network also has cosine characteristics, and the features of different classes are distinguishable.

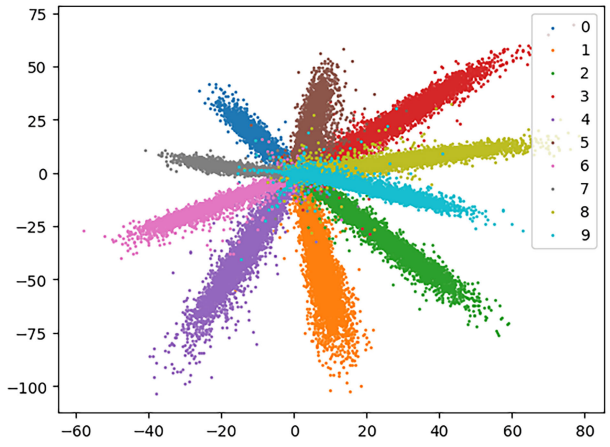

FIGURE 3. The distribution of deep features in training set of MNIST.

The difference between the categories can be distinguished from the direction, so that the output of the guide layer conforms to the characteristic distribution. Experiments demonstrate that cosine similarity is more effective than MSE to transfer knowledge in the proposed approach.

The approach makes the best of the depth information of the teacher network and the knowledge extracted from the deep hidden layer to train the student network. Fig. 2 represents the general structure of the proposed approach. Starting from a randomly initialized student network and a high-performance teacher network, the last convolutional layer of the teacher network is defined as the hint layer. Analogously, the guided layer is defined as the last convolutional layer of student network. Afterword, a decoder is designed on top of the hint layer and the guided layer, which can remove the redundant part of the hint layer to generate a high-value feature map. Training the student network up to the guided layer by minimizing loss function (8). Finally, the parameters from the decoder to the output layer of the student network directly migrates the parameters of the teacher network. Algorithm 1 details the training process of the proposed approach.

Algorithm 1 Training Process of the Proposed Approach

Input: The parameters up to the hint layer h: $\boldsymbol{W}_{\mathrm{T1}}$; the parameters from the hint layer to the output layer: $\boldsymbol{W}_{\mathrm{T2}}$; the randomly initialized parameters up to the guided layer k: $\boldsymbol{W}_{\mathrm{S1}}$.

Output: The parameters of the whole student network:$\boldsymbol{W}_{\mathrm{S}}$.

$\boldsymbol{W}_{\mathrm{T1}} = \left\{{\boldsymbol{W}_{\mathrm{T1}}}^{1}, {\boldsymbol{W}_{\mathrm{T1}}}^{2}, \cdots {\boldsymbol{W}_{\mathrm{T1}}}^{h}\right\}$

$\boldsymbol{W}_{\mathrm{S1}} = \left\{{\boldsymbol{W}_{\mathrm{S1}}}^{1}, {\boldsymbol{W}_{\mathrm{S1}}}^{2}, \cdots {\boldsymbol{W}_{\mathrm{S1}}}^{k}\right\}$

while not done **do**

  for each batch with size N **do**

    $\boldsymbol{Z}^{l}{}_{m,n} = f(\sum_{i}\sum_{j} \boldsymbol{W}^{l}{}_{i,j} \cdot \boldsymbol{x}^{l}{}_{i,m,n} + \boldsymbol{b}^{l})$

    $\boldsymbol{P}_{\mathrm{S}}(x) = \boldsymbol{F} \otimes \boldsymbol{Z}^{l}{}_{m,n}$

    Calculate $L(\boldsymbol{x})$: $L(\boldsymbol{x}) = \frac{\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x}) \cdot \boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})}{\|\boldsymbol{P}_{\mathrm{S}}(\boldsymbol{x})\| \cdot \|\boldsymbol{P}_{\mathrm{T}}(\boldsymbol{x})\|}$

    Update $\boldsymbol{W}_{\mathrm{S1}}^{e} = {\boldsymbol{W}_{\mathrm{S1}}}^{e-1} - \alpha \frac{\partial L(\boldsymbol{x})}{\partial {\boldsymbol{W}_{\mathrm{S1}}}^{e-1}}$

  end for

end while

get $\boldsymbol{W}_{\mathrm{S2}}$: $\boldsymbol{W}_{\mathrm{S2}} \leftarrow \boldsymbol{W}_{\mathrm{T2}}$

$W_{\mathrm{S}} = \{W_{\mathrm{S1}}, W_{\mathrm{S2}}\}$

IV. EXPERIMENTS

In this section, DFMT is validated on benchmark datasets and compared with knowledge distillation (KD), FitNet, and AT. In addition, the generality of DFMT is verified in two typical network structures: VGGNet and ResNet. The excellent performance of the teacher network is very important in teacher-student structure. Therefore, VGG-16 Net and ResNet50 are chosen as teacher network. Table 1 describes four different student network architectures used for the teacher network of VGG-16 Net. ResNet14 is chosen as the student network to learn knowledge from ResNet50. All the experiments are repeated 3 times with same environment, and the average of error rates or accuracy is taken as the final results. These experimental results present that the proposed approach can effectively improve the performance of student network and outperforms other methods.

A. EXPERIMENTS ON CIFAR-10

Aiming to validate the proposed approach, an experiment is done on the CIFAR-10 [27]. CIFAR-10 dataset consists of 60,000 images with 50,000 training images and 10,000 test images from 10 classes. Each picture is a color picture with a resolution of 32×32.

When VGG-16 Net is used as the basic structure of the teacher network, a small decoder is designed behind the last convolutional layer of VGG-16 Net. The decoder can remove the redundant part of the hint layer and generate a high-value feature map from the features extracted from convolutional layer. The feature map is responsible for the learning process of the student network.

TABLE 1. Different architectures of student network based on VGGNet structures.

Student-V1	Student-V2	Student-V3	Student-V4
Conv $3 \times 3 \times 8$	Conv $3 \times 3 \times 16$	Conv $3 \times 3 \times 16$	Conv $3 \times 3 \times 32$
Conv $3 \times 3 \times 16$	Conv $3 \times 3 \times 16$	Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 32$
Conv $3 \times 3 \times 16$	Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 48$
Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 48$
Maxpool 2×2	Maxpool 2×2	Maxpool 2×2	Maxpool 2×2
Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 64$
Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 64$
Conv $3 \times 3 \times 48$	Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 80$
Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 80$
Maxpool 2×2	Maxpool 2×2	Maxpool 2×2	Maxpool 2×2
Conv $3 \times 3 \times 64$	Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 96$
Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 96$
Conv $3 \times 3 \times 80$	Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 112$
Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 112$	Conv $3 \times 3 \times 112$
Maxpool 2×2	Maxpool 2×2	Maxpool 2×2	Maxpool 2×2
Conv $3 \times 3 \times 96$	Conv $3 \times 3 \times 112$	Conv $3 \times 3 \times 112$	Conv $3 \times 3 \times 128$
Conv $3 \times 3 \times 112$	Conv $3 \times 3 \times 112$	Conv $3 \times 3 \times 128$	Conv $3 \times 3 \times 128$
Maxpool 2×2	Maxpool 2×2	Maxpool 2×2	Conv $3 \times 3 \times 144$
			Conv $3 \times 3 \times 144$
			Maxpool 2×2
Decoder	Decoder	Decoder	Decoder
FC	FC	FC	FC
Softmax	Softmax	Softmax	Softmax

VGG-16 Net is composed of 13 convolution layers, 5 pooling layers, 2 full connection layers and a softmax layer. A large number of 3×3 convolution kernels are used to reduce the parameter calculation. It achieved excellent performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 competition [28].

Since VGG-16 Net has a deep network architecture, it takes a lot of cost to train a teacher network with excellent performance. Deep neural network obtains hierarchical feature information of input through pre-training method. The shallow layers of networks obtain the input low-level semantic information (edge, color, etc.) of input, and this information is common in the classification task. Therefore, transfer learning [29] is used to save time when training the teacher network. The weights of the VGG16 network on the Imagenet dataset are used as the initial parameter, and then the parameters of the whole network are trained by standard backpropagation. Adam [30] is used for training the teacher network with a learning rate of 0.001. After 1k iterations, the teacher network achieves an accuracy of 92.64%.

In order to verify the effectiveness of the proposed approach, a thinner and lighter student network than the teacher network is designed. The student-4 is shown in Table 1. This student network contains four convolutional blocks, each of which has four convolutional layers of kernel size 3×3 and a non-overlapping 2×2 max-pooling. After the final convolution block, a small decoder with the same architecture as the decoder in the teacher network is designed, which can generate a high-value feature map from the features extracted from convolutional layers. Finally, two full connection layers and a softmax layer are behind the decoder.

Similarly, adam is also used for training the guided layer to mimic the output of the hint layer. These training samples of the training set are divided into 40,000 training samples and 10,000 validation samples. The student network up to the guided layer is trained by minimizing (8) through stochastic gradient descent. The Data is enhanced during the training stage by random translation and flipping. The mini-batch is set at 256, and the size of the feature map is set at 19×19. The training is stopped when the training error of the validation set done not reduce after 1,000 epochs. For the parameters from the decoder to the output layer of the student network, which directly migrates the parameters of the teacher network.

Firstly, the student network is trained by standard backpropagation to obtain the original performance of the student network, and compared with the proposed approach to verify the effectiveness of DFMT. After that, KD, FitNet, and AT are used to train the same student network. The advancement of DFMT can be verified by comparing with these two methods.

Using a higher value for temperature can produce softer targets during the process of KD. When different temperature parameters are selected, the effect of knowledge transfer are also different. To improve the credibility of the comparison, the of 20 and 30 are chosen respectively when KD is used to train the student network.

Table 2 presents recognition rates for different methods. The student network is thinner and lighter than the teacher network and as roughly 1/12 of the teacher parameters. Student-original denotes student network is

TABLE 2. Accuracy on CIFAR-10 using VGGNet structures.

Algorithm	# Params	Accuracy	# Iter
Teacher(VGG-16)	16.1M	92.64%	1k
Student-V4-original	1.3M	88.95%	4k
KD-T=20	1.3M	90.12%	4k
KD-T=30	1.3M	90.30%	4k
FitNet [16]	2.5M	91.61%	6k
AT	1.3M	91.92%	6k
Proposed Method(DFMT)	1.3M	**93.32%**	1k

trained by using standard backpropagation and achieves an accuracy of 88.95%. Compared with using the standard backpropagation, the student network obtains a great performance improvement through DFMT, and achieves an accuracy of 93.32% after 1000 iterations. Finally, compared with other network compression methods, DFMT cost the least but provides the best performance.

As shown in Table 1, different architectures of the student network are designed to verify the applicability of our approach. Each student network consists of successive zero-padded convolutional layers of kernel size 3×3. A non-overlapping max-pooling takes place after some of the convolution layers. After the final convolution block, a small decoder with the same architecture as the decoder in the teacher network is designed.

The same size of the feature map is used to transfer knowledge when training these student networks to mimic the output of the hint layer. Adam is used to training these student networks with a learning rate of 0.001.

The features of different classes present radioactive distribution, which can be distinguished by decision boundaries. The knowledge extracted by the teacher network also has cosine characteristics, and the features of different classes are distinguishable. Therefore, cosine distance is chosen as the loss function when training the student network. Aiming to evaluate the validity of our choice, MSE and cosine similarity are respectively used as loss functions to compare in the same student model.

TABLE 3. Contrast accuracy on CIFAR-10 using different architectures student networks and different loss functions.

Model	# Params	Accuracy		
		Student-original	DFMT In MSE	DFMT In COS
Student-V1	0.6M	87.86%	90.35%	**91.62%**
Student-V2	0.7M	88.15%	90.47%	**91.96%**
Student-V3	0.8M	88.37%	90.84%	**92.51%**
Student-V4	1.3M	88.95%	91.26%	**93.32%**

Table 3 presents the obtained results. VGG-16 Net is used as the basic structure of the teacher network. The proposed approach is verified on four different student networks, and two loss functions are respectively used to compare in the same student model. Student-original denotes that the student network is trained by using standard backpropagation. As shown in Table 3, the student-1 network with fewer parameters can also achieve good performance. The performance of student networks improves with the increase in their parameters. For the selection of loss function, cosine similarity is superior to MSE in different student networks.

DFMT is inspired by visual perception. A decoder is designed on top of the hint layer, which can remove the redundant part of the hint layer and generate a high-value feature map from the features extracted from convolutional layers. The depth feature map is used to replace the output of the hint layer for knowledge transfer. The decoder consists of several deconvolutional layers, and different sizes of feature maps can be got by setting different strides of filters.

The feature map contains the important knowledge of teacher network, which plays an important role in the proposed approach. Therefore, aiming to verify the influence of different sizes of feature maps on knowledge transfer, different sizes of feature maps are used to train student networks with different sizes respectively.

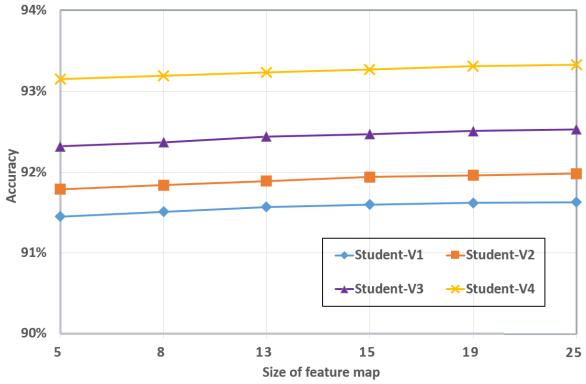

FIGURE 4. DFMT performance on CIFAR-10 using different sizes of the feature map.

Fig. 4 reports the obtained results. The student network is trained with 5×5, 8×8, 13×13, 15×15, 19×19 and 25×25 size feature maps respectively. Cosine similarity is used as loss function, and adam is used for training student networks with a learning rate of 0.001. From Fig. 4, it can be seen that student networks can obtain advanced performance in different sizes of feature maps.

When VGG-16 Net is used as the basic structure of the teacher network, the validity and advancement of the proposed approach are empirically verified by comparing with student networks trained by standard backpropagation, knowledge distillation, and FitNet. In order to verify the generality of the algorithm in the typical network, the residual network structure is used to experiment.

It has been demonstrated that a very deep network is difficult to train because of the problems of gradient disappearance and gradient explosion. ResNet is composed of several residual blocks and skip connection is used in the residual block, which solved the problem of performance degradation of deep convolution neural networks.

A residual network with 50 layers is used as the basic structure of the teacher network and a small decoder is designed behind the last convolutional layer. Similarly, transfer learning is used to training the teacher network. The weights of the ResNet50 on the Imagenet dataset are used as the initial parameter, and then standard backpropagation is used to training the parameters of the whole network. The teacher network achieves an accuracy of 92.86% after transfer learning. A residual network with 14 layers is chosen as the student network during the learning process, which has the same network structure as the teacher network. A small decoder with the same architecture as the decoder in the teacher network is designed behind the last convolutional layer.

In the experiment, The batch size is set at 256, and the size of the feature map is set at 11×11. The Data is enhanced during the training stage by random translation and flipping. Adam is used for training the student network with a start learning rate of 0.001, and the learning rate is changed to 0.0005 and 0.0001 when iterations of 300 and 600. The student network is trained to learn knowledge from the teacher network by minimizing (8). The training stage is stopped when the training error of the validation set done not reduce after 1000 epochs. Because the structure is designed consistently, the parameters from the decoder to the output layer of the student network directly migrate the parameters of the teacher network.

Recognition rates as indicated in Table 4. 50 layers and 14 layers of the residual network are used as the teacher network and the student network respectively. The recognition rates of the student network increased by 2.8% when compared with using the standard backpropagation. When compared with other network compression methods, DFMT provides the best performance.

TABLE 4. Accuracy on CIFAR-10 using ResNet structures.

Algorithm	# Params	Accuracy
Teacher(ResNet50)	2.4M	92.86%
Student(ResNet14)-original	0.7M	89.65%
KD-T=20	0.7M	90.31%
KD-T=30	0.7M	90.52%
Proposed Method(DFMT)	0.7M	**92.45%**

B. EXPERIMENTS ON CIFAR-100

An experiment is done on the CIFAR-100 to validate the effectiveness of the proposed approach. Similar to the CIFAR-10 dataset, CIFAR-100 consists of 60,000 images with 50,000 training images and 10,000 test images from 100 classes. Each picture is a color picture with a resolution of 32×32.

Since the CIFAR-100 dataset contains 100 classes and there are fewer images per class, it is more difficult to train an excellent deep network on the CIFAR-100 dataset. When VGG-16 Net is used as the basic structure of the teacher network and transfer learning is used to train the teacher network. This training method greatly reduces the training time of teacher network and saves the training cost.

The student network uses the same architecture as in CIFAR-10. The student-4 is shown in Table 1. Due to the increase in classes of datasets and the decrease of the number of pictures in each class, the student network needs more iterations to learn knowledge from the teacher network. The size of the feature map is set at 19×19, and the Data is enhanced during the training stage by random translation and flipping. Similarly, cosine similarity is used as loss function, and adam is used for training student networks.

Table 5 summarizes the results of the student network through different methods. The teacher network achieves a 69.93% recognition rate, and The student network achieves 57.74% accuracy by using the standard backpropagation. Surprisingly, compared with using the standard backpropagation, the student obtains a great performance improvement through DFMT, and achieves an accuracy of 68.73%. when compared with the performance of other knowledge Transfer methods, DFMT provides better performance than existing methods.

TABLE 5. Accuracy on CIFAR-100 using VGGNet structures.

Algorithm	# Params	Accuracy
Teacher(VGG-16)	16.1M	69.93%
Student-V4-original	1.3M	57.74%
KD-T=20	1.3M	63.15%
KD-T=30	1.3M	63.47%
FitNet [16]	2.5M	65.96%
AT	1.3M	66.21%
Proposed Method(DFMT)	1.3M	**68.73%**

In addition, the generality of DFMT is verified on the CIFAR-100 dataset. Similarly, residual network with 50 layers and 14 layers are used as the teacher network and the student network respectively. The batch size is set at 256, and the size of the feature map is set at 15×15. The Data is enhanced during the training stage by random translation and flipping. Adam is used for training the student network and the learning rate of 0.001, 0.0005 and 0.0001 until iterations of 300,600, and 800.

Recognition rates of different methods for student networks on CIFAR-100 dataset as shown in Table 6. The student network achieves 62.43% accuracy by using the standard backpropagation. Compared with using the standard

TABLE 6. Accuracy on CIFAR-100 using ResNet structures.

Algorithm	# Params	Accuracy
Teacher(ResNet50)	2.4M	70.26%
Student(ResNet14)-original	0.7M	62.43%
KD-T=20	0.7M	66.15%
KD-T=30	0.7M	66.47%
Proposed Method(DFMT)	0.7M	**69.14%**

backpropagation, the student obtains a great performance improvement through DFMT and achieves an accuracy of 69.14%. DFMT provides the best performance compared with other network compression methods.

C. EXPERIMENTS ON SVHN

As a sanity check for the proposed approach, an experiment is made on the SVHN [31]. The SVHN dataset consists of 32 × 32 pixel house numbers pictures collected in GoogleStreet View, of which there are 73,257 pictures in the training set, 26,032 pictures in the test set and 531, 131 pictures in extra set.

VGG-16 Net is chosen as the basic structure of the teacher network, and transfer learning is used when training the teacher network to save time. The student-V4 network specified in Table 1 is chosen as the student network. The Data is enhanced during the training stage by random translation and flipping. The mini-batch is set at 256, and the size of the feature map is set at 19 × 19. Similarly, cosine similarity is used as loss function, and adam is used for training student networks.

The error rate in SVHN dataset is shown in Table 7. When compared with using the standard backpropagation, DFMT gets excellent performance. The student network obtains an error rate of 2.19%, which is lower than the error rate 6.45% of standard backpropagation. Furthermore, compared with KD and FitNet, DFMT provides the best performance.

TABLE 7. Accuracy on SVHN using VGGNet structures.

Algorithm	# Params	Error
Teacher(VGG-16)	16.1M	1.92%
Student-V4-original	1.3M	6.45%
KD-T=20	1.3M	5.43%
KD-T=30	1.3M	5.08%
FitNet [16]	1.5M	2.38%
AT	1.3M	2.25%
Proposed Method(DFMT)	1.3M	**2.19%**

In order to evaluate DFMT on a different kind of network architecture, an experiment is done on the SVHN dataset using ResNet architecture. A residual network with 50 layers is used as the basic structure of the teacher network and a residual network with 14 layers is used as the student network. The batch size is set at 256, and the size of the feature map is set at 11 × 11. Cosine similarity is used as the loss function, and adam is used for training student networks.

The error rate of the student network trained by standard backpropagation, KD, and DFMT on SVHN is shown in Table 8. Compared with using the standard backpropagation, the student obtains a great performance improvement through DFMT, and achieves an error rate of 2.15% after1000 iterations. Finally, compared with knowledge Distillation, DFMT also provides the best performance.

TABLE 8. Accuracy on SVHN using ResNet structures.

Algorithm	# Params	Error
Teacher(ResNet50)	2.4M	1.85%
Student(ResNet14)-original	0.7M	6.24%
KD-T=20	0.7M	5.37%
KD-T=30	0.7M	5.16%
Proposed Method(DFMT)	0.7M	**2.15%**

V. CONCLUSION

A new method of neural network model compression inspired by visual perception is proposed in this paper. It generates a high-value feature map from the features extracted from convolutional layers of the teacher network, and the feature map is used to guide the training of the student network. The student network with fewer parameters can obtain outstanding performance by this learning method. DFMT is universal and suitable for different kinds of network architectures. These experiments on benchmark datasets evaluate the effectiveness and advancement of the proposed approach. Furtherly, in future work, we would like to research other more effective knowledge transfer methods, so that the network with fewer parameters can get better performance.

REFERENCES

[1] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[3] R. Szeliski, *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer, 2010.
[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.
[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[6] 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)*, Oct. 2015, pp. 1–9.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 770–778.
[8] 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. 2261–2269.
[9] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Sep. 2016, pp. 7–10.
[10] A. Mishra, E. Nurvitadhi, J. J. Cook, and D. Marr, "WRPN: Wide reduced-precision networks," Sep. 2017, *arXiv:1709.01134*. [Online]. Available: http://arxiv.org/abs/1709.01134
[11] J. Cheng, P.-S. Wang, G. Li, Q.-H. Hu, and H.-Q. Lu, "Recent advances in efficient computation of deep convolutional neural networks," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 64–77, Jan. 2018.
[12] C. Bucilă, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 535–541.
[13] S. Ravi, "ProjectionNet: Learning efficient on-device deep networks using neural projections," 2017, *arXiv:1708.00630*. [Online]. Available: http://arxiv.org/abs/1708.00630

[14] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4032–4044, Aug. 2019.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[16] G. R. Zhou, Y. Fan, R. P. Cui, and W. J. Bian, "Rocket launching: A universal and efficient framework for training well-performing light net," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4580–4587.

[17] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," 2018, *arXiv:1804.03235*. [Online]. Available: http://arxiv.org/abs/1804.03235

[18] Y. Wei, X. Pan, H. Qin, W. Ouyang, and J. Yan, "Quantization mimic towards very tiny CNN for object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 274–290.

[19] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5292–5303, Nov. 2018.

[20] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*. [Online]. Available: http://arxiv.org/abs/1412.6550

[21] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7130–7138.

[22] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2016, *arXiv:1612.03928*. [Online]. Available: http://arxiv.org/abs/1612.03928

[23] G. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2014, pp. 2924–2932.

[24] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 5528–5531.

[25] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 2048–2057.

[26] Y. D. Wen, K. P. Zhang, Z. F. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Nov. 2015, pp. 499–515.

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

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[29] M. S. Long, Y. Cao, J. M. Wang, and M. L. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 97–105.

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

[31] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, p. 4.

ZHIBO GUO received the M.S. degree from the Nanjing University of Technology, Nanjing, China, in 2003, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, in 2007. He is currently an Associate Professor with the School of Information Engineering, Yangzhou University. He has authored over 60 scientific articles and patents. His research interests include pattern recognition, computer vision, machine learning, and artificial intelligence.

XIN YAO received the B.S. degree from Yangzhou University, Yangzhou, China, in 2018, where he is currently pursuing the M.S. degree in science in control engineering with the School of Information Engineering. His current research interests include deep learning and pattern recognition.

YIXUAN XU graduated from the Department of Foreign Languages and Literatures Foundation Program, Tsinghua University, Beijing, China, in 2018. He is currently pursuing the B.S. degree in computer science (artificial intelligence) with the University of Nottingham, Nottingham, U.K. His current research interests include programming and computer vision.

YING ZHANG received the B.S. degree from Yangzhou University, Yangzhou, China, in 2017, where she is currently pursuing the M.S. degree in science in control engineering with the School of Information Engineering. Her current research interests include machine learning and pattern recognition.

LINGHAO WANG received the B.S. degree from Shenyang Urban Construction University, Shenyang, China, in 2018. He is currently pursuing the M.S. degree in science in control engineering with the School of Information Engineering, Yangzhou University, Yangzhou, China. His current research interests include deep learning and pattern recognition.