IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 September 2022, accepted 22 October 2022, date of publication 1 November 2022, date of current version 9 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218890

RESEARCH ARTICLE

Progressive Network Grafting With Local Features Embedding for Few-Shot Knowledge Distillation

WEIDONG DU
School of Mechanical Engineering, Southeast University, Nanjing 210096, China
e-mail: davy1976@163.com

ABSTRACT Compared with traditional knowledge distillation, which relies on a large amount of data, few-shot knowledge distillation can distill student networks with good performance using only a small number of samples. Some recent studies treat the network as a combination of a series of network blocks, adopt a progressive graft strategy, and use the output of the teacher network to distill the student network. However, this strategy ignores the importance of the local feature information generated by the teacher block, which indicates what features should be learned by the corresponding student block. In this paper, we argue that using the features output from the teacher block can guide the student block to further learn more useful information from the teacher block. Therefore, we propose a joint learning framework for few-shot knowledge distillation that exploits both the output of the teacher network and the local features generated by the teacher block to optimize the student network. The local features will guide the student block to learn the output of the teacher block, and the output of the teacher network will allow the student network to take its learned local features to better contribute to the classification. In addition, further model compression was carried out to design a series of student networks with fewer number of parameters by reducing the number of network channels. Finally, extensive experiments using the model on CIFAR10 and CIFAR100 datasets show that our method outperforms SOTA, and our method has considerable advantages even with a very small number of parameters in further model compression experiments.

INDEX TERMS Knowledge distillation, few-shot learning, model compression, features embedding.

I. INTRODUCTION

DEEP neural networks are widely used in various computer vision tasks [1], [2], [3], [4] and have achieved remarkable results [5], [6]. However, the current state-of-the-art deep models suffer from huge energy consumption, high operating and storage costs, which greatly hinder their deployment in resource-efficient situations [7], [8], [9]. To solve this problem, a lot of works have been proposed to compress neural networks for obtaining more lightweight neural network models. These works are mainly divided into two technologies: network pruning [10], [11], [12] and knowledge distillation [13], [14].

Network pruning is usually performed on the trained network to remove unimportant channels or weights, and then the pruned network is retrained to restore the performance of the original network [15], [16]. These pruning methods usually require a large amount of labeled data, and the training process is very time-consuming [9], [17]. The knowledge distillation method transfers knowledge from the pre-trained teacher network to the student network, and trains the student network by making students imitate the output of the teacher network, so as to achieve the performance of the teacher network [18], [19], [20]. However, since the student network is usually set to be randomly initialized, it needs to rely on a large amount of data for knowledge transfer to train a model with good performance [9], [21]. Therefore, it is difficult for the existing methods to recover the lost accuracy with few training samples.

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

In order to solve the above problems, several few-shot knowledge distillation methods have been proposed to transfer knowledge from teacher to student. To reduce data and time costs, Li et al. [7] proposed a few-shot method to extract knowledge from unlabeled minority samples, combining network pruning and block distillation to compress the teacher model. Bai et al. [8] designed the cross-distillation model to reduce layer-wise accumulated errors in the setting of few-shot and realize the student network with stronger robustness and better performance. Zhou et al. [22] introduced a progressive training strategy to achieve knowledge transfer between student network and teacher network by matching the feature distribution between them. Shen et al. [9] proposed a progressive network grafting method, which trains the student network through block grafting and network grafting, reduces its parameter space, and enhances the robustness of knowledge distillation. However, these methods rarely take into account both the feature information of the teacher network and the classification result information to optimize the student network simultaneously. Since the structure of the student network is different from that of the teacher network, it is difficult for the student network to completely imitate the teacher's output when only the local features information is considered. Meanwhile, due to the scarcity of samples in the few-shot scenario, it is difficult to optimize the student network only by using the output from the teacher network. Naturally, the use of teacher feature information and classification result information at the same time can make up for their respective shortcomings and better optimize the student network in few-shot scenarios. Therefore, this paper proposes a progressive grafting network for the fusion of local features and classification results from the teacher network for few-shot knowledge distillation, so that the two can enhance each other in a complementary way, so that the teacher network can optimize the student network in terms of local features and global classification information. In addition, we make full use of local features information from the teacher block and output from the teacher network to optimize the knowledge transfer process of teacher network and student network. Moreover, this paper designs a student network with fewer parameters through a series of channel reduction settings to explore the effectiveness of our method.

The main contributions of this paper are concentrated in the following three parts.

(1) A progressive grafting network for the fusion of local features and output from student network for few-shot knowledge distillation is proposed. In the few-shot scenario, the method makes full use of the local features information of the teacher network and the classification result information from teacher network to optimize and improve the performance of the student network in a complementary manner. Among them, The local features will guide the student block to learn the useful local features of the teacher block, and the output of the teacher network will allow the student network to take its learned local features to better contribute to the classification.

(2) We further design a relatively lightweight network model to achieve model compression by reducing the number of student network channels. Thus, a series of student networks with fewer parameters can be obtained to achieve the few shot classification.

(3) Extensive experiments on CIFAR10 and CIFAR100 datasets show that our method outperforms SOTA. Notably, the designed lightweight network model has considerable advantages even with a very small number of parameters, which can effectively validate the effectiveness of the proposed method based on the learning strategy of knowledge distillation and model compression.

The rest of this paper is constructed as follows: Section II recalls some related knowledge of few-shot knowledge distillation. In Section III, the loss function of local feature distribution and global classification results distillation is developed. Then, a novel hybrid distillation method for feature distribution and classification results is designed. Section IV shows the experimental results. Finally, Section V summarizes the study and discusses future work.

II. RELATED WORK

Most of the previous studies on knowledge distillation rely on abundant labeled data to transfer the knowledge of teacher network to Student network. However, in real world, there may not be a large amount of data for model training, so knowledge distillation and Few-Shot Learning can be combined.

In order to make full use of existing training data, some existing works apply knowledge distillation by layer-wisely minimizing Euclidean distance [23], [24], [25], [26]. Layer-by-layer training is generally efficient because each layer of the student network is optimized separately, while requiring fewer parameters to optimize compared to back-propagation training for the entire student [14]. Aside from layer-wisely distillation of the model, data from different but related domains can assist pruning of the target domain [27]. In addition to labeled Few-Shot Learning, new methods to extract knowledge from a small number of unlabeled samples have also been investigated to improve data efficiency and training/processing efficiency [7]. Also based on the idea of unlabeled data, the Self-supervised Knowledge Distillation method for Few-shot Learning is proposed to learn the real output classification manifold through self-supervised learning. Once this structure is learned, the method trains a student model that preserves the original output manifold structure while collectively maximizing the discriminability of the learned representations [28]. Some studies have also improved layer-wisely distillation and proposed cross-distillation, which can effectively reduce the estimation error of layer-wisely distillation by cross-training the hidden layer network of teachers and students [8]. Besides the cross-distillation model, a principled dual-stage distillation scheme based on small samples has also been proposed, in which the student modules are grafted into the teacher network for training, then the trained student modules are spliced

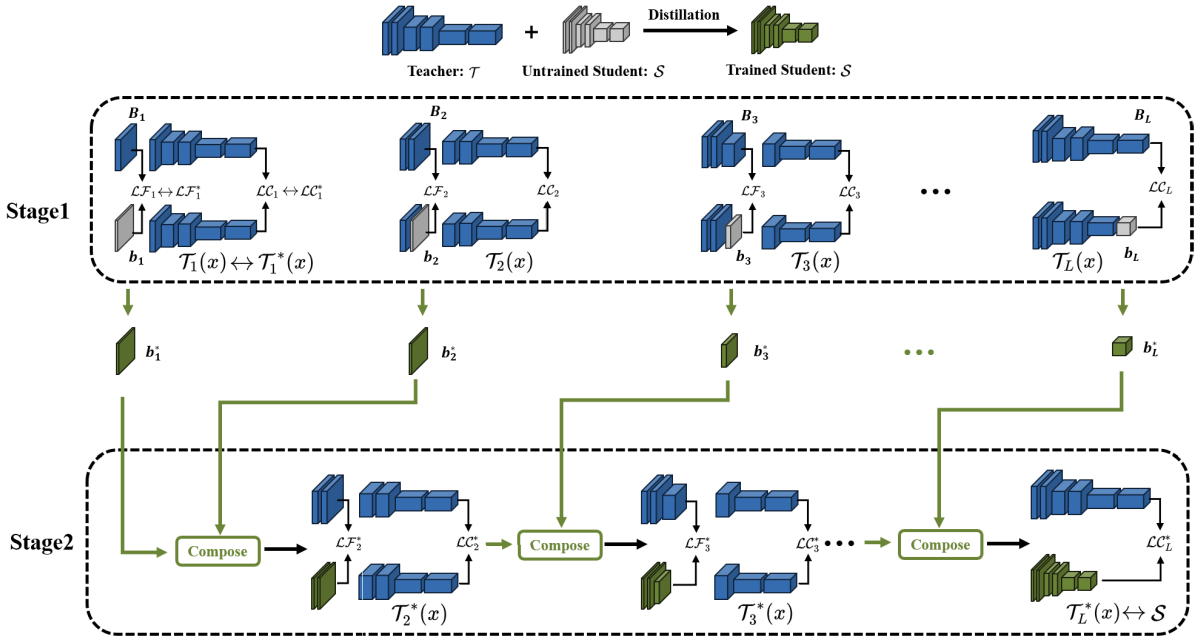


FIGURE 1. The dual-stage knowledge distillation strategy in our work for few-shot knowledge distillation. Firstly, student network $\mathcal{S}$ is decomposed into several blocks: $\{b_l\}_{l=1}^{L}$, each of which is grafted onto teacher as $\mathcal{T}_l(x)$ and then optimized by $\mathcal{LF}_l$ and $\mathcal{LC}_l$. Especially, the $\mathcal{LF}_l$ represents the feature distribution loss, while $\mathcal{LC}_l$ represents the classification loss. Secondly, trained block b_l^* in the first stage are sequentially composed into the trained student network $\mathcal{T}_l^*(x)$ and optimized by $\mathcal{LF}_l^*$ and $\mathcal{LC}_l^*$. Finally, the $\mathcal{T}_L^*(x)$ is the trained $\mathcal{S}$ that we want.

together and grafted into the teacher network, and finally the teacher network is replaced [9]. In some of the above methods, some will add additional convolutional layers to the compressed network during training, which increases the complexity of the network structure. Therefore, a progressive feature distribution distillation method without modifying the network structure is proposed, which can effectively match the feature distribution of the compressed network and the original network [22].Existing methods generally only focus on classification result alignment or feature distribution alignment to train student networks, and do not combine the two parts.

III. THE PROPOSED METHOD

Our goal is to distill the teacher network through a series of knowledge to obtain a compact student network with fewer parameters. In this paper, teacher network is denoted by $\mathcal{T}$, and teacher network can be regarded as $\mathcal{T}(x) = B_L \circ \cdots B_l \circ \cdots B_1(x)$. B_l represents the l-th teacher block. Denoting the student network by $\mathcal{S}$, the student network can be viewed as $\mathcal{S}(x) = b_L \circ \cdots b_l \circ \cdots b_1(x)$. b_l denotes the l-th student block. By simulating the feature distribution and classification results of the teacher block, the students gradually learn and master the knowledge in the teacher block.

In order to achieve the above goals, this paper proposes a feature distribution and logits hybrid distillation strategy, as shown in Figure 1. In the first stage, each student block is grafted and the corresponding teacher block is replaced, and the knowledge of the teacher block is learned by imitating the feature distribution of the corresponding teacher block and the classification result of the whole teacher network. In the second stage, all the trained student blocks are grafted into the teacher network, and the number of grafted student blocks is gradually increased to learn the feature distribution of the corresponding position of the teacher block sequence and the classification output information of the teacher network, and gradually replace the entire teacher network.

A. BLOCK GRAFTING AND NETWORK GRAFTING

Following the block grafting and network grafting strategies in paper [9], we divided the student network into a series of blocks with fewer parameters, and grafted each student block separately into the teacher network to learn the knowledge corresponding to the teacher block. The number of student blocks should be equal to the number of teacher blocks. The grafted teacher network can be expressed as:

$$\mathcal{T}_l(x) = B_L \circ \cdots B_{l+1} \circ b_l \circ B_{l-1} \circ \cdots B_1(x) \tag{1}$$

Among them, the l-th student block b_l replaces the teacher block B_L. To train and graft teacher network $\mathcal{T}_1$, only optimize the parameters of student block b_l.

Since there is a difference in the number of channels between the student block and the teacher block, an adaptive module is introduced in this paper to align the channel size differences between the block and network grafting. This module can be divided into two categories, namely, the self-adaptive module from teacher block to student block $a_{l-1}^{t\to s}(x^{l-1})$ and the self-adaptive module from student block to teacher block $a_{l-1}^{s\to t}(x^{l})$. Given the adaptive module, the

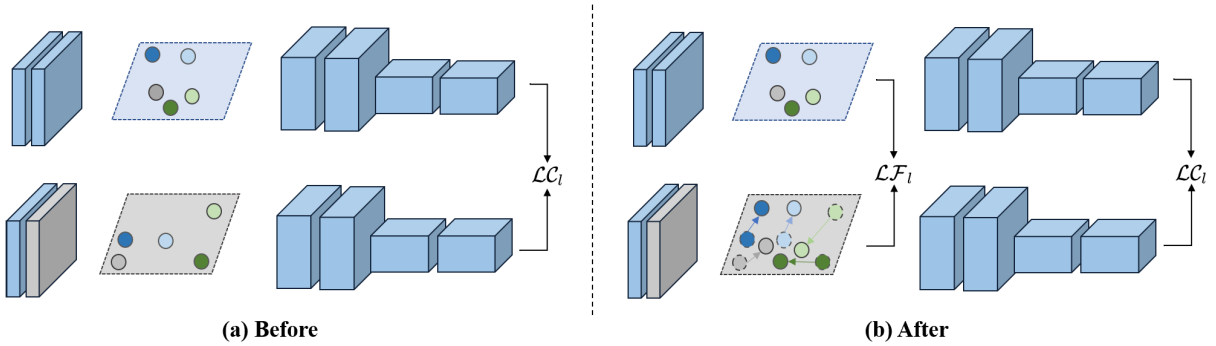


FIGURE 2. Comparison of effects before and after adding feature distribution loss. (a) Before adding the feature distribution loss $\mathcal{LF}_l^{(*)}$, although we have the classification loss $\mathcal{LC}_l^{(*)}$ to help us learn the overall effect of the teacher network $\mathcal{T}_l(x)^{(*)}$, the distribution of the student block $b_l^{(*)}$ are still different from the teacher block B_l. (b) After adding the $\mathcal{LF}_l^{(*)}$, each $b_l^{(*)}$ is aligned with the B_l from the feature level, which pushes the $b_l^{(*)}$ to learn more details.

wrapped $b_l^*(x^{l-1})$ can be expressed as:

$$b_l^*\left(x^{l-1}\right) = a_l^{s\to t} \circ b_l \circ a_{l-1}^{t\to s}\left(x^{l-1}\right) \tag{2}$$

Combined with Eq1 and Eq2, the final grafted teacher network can be expressed as:

$$\mathcal{T}_l(x) = B_L \circ \cdots B_{l+1} \circ b_l^* \circ B_{l-1} \circ \cdots B_1(x) \tag{3}$$

Each student block is trained separately through the strategy of block grafting. In order to realize the mutual cooperation of student blocks, the student blocks grafted into the teacher network are gradually increased, and the dependence on the original teacher is reduced. On the basis of $\mathcal{T}_1(x) = B_L \circ \cdots B_2 \circ b_1^*(x)$, training blocks $b_2^*(x)$, $b_3^*(x)$, $\ldots$,$b_l^*(x)$ is grafted to teacher network $\mathcal{T}$ in turn, as shown in Figure 1. The grafted teacher network in network grafting can be expressed as:

$$\mathcal{T}_l^*(x) = B_L \circ \cdots B_{l+1} \circ b_l^* \circ b_{l-1}^* \circ \cdots b_1^*(x) \tag{4}$$

In the process of network grafting, a series of models are optimized: $\left\{\mathcal{T}_l^*(x)\right\}_{l=1}^{L}$. Finally, all student blocks are connected and the complete network $\mathcal{T}_L^*$ is formed. However, $\mathcal{T}_L^*$ is still different from the original student network $\mathcal{S}(x)$. $\mathcal{S}_L^*$ consists of a series of b_l^*, and $\mathcal{S}(x)$ consists of b_l. Compared to b_l, b_l^* contains additional adaptive modules $a_l^{s\to t}$ or $a_{l-1}^{t\to s}$ which means that $\mathcal{T}_L^*(x)$ has more parameters than $\mathcal{S}(x)$. Since the adaptive module is linear, the module can be incorporated into the next convolution layer without adding any parameters. For $b_{l+1}^* \circ b_l^* = a_{l+1}^{s\to t} \circ b_{l+1} \circ a_l^{t\to s} \circ a_l^{s\to t} \circ b_l \circ a_{l-1}^{t\to s}$, $(a_l^{t\to s} \circ a_l^{s\to t})$ can be merged into b_{l+1}. We express it as Eq.(5). Then, $\mathcal{T}_L^*(x)$ can be converted to the following form:

$$\hat{b}_{l+1} = b_{l+1} \circ a_l^{t\to s} \circ a_l^{s\to t} \tag{5}$$

$$\mathcal{T}_L^*(x) = \hat{b}_L \circ \cdots \hat{b}_{l+1} \circ \hat{b}_l \circ \hat{b}_{l-1} \circ \cdots \hat{b}_1(x) \tag{6}$$

In other words, we realize knowledge transfer from teacher network T to student network S.

B. LOGITS DISTILLATION

Logits from different network architectures may vary greatly, which may lead to optimization difficulties. Therefore, in this paper, the l_2 loss function on the normalized logits of knowledge transfer between teacher block B_l and student block b_l is proposed to let the student block simulate and recover the output of the original teacher block. The calculation formula is as follows:

$$\mathcal{LC}_l(x) = \frac{1}{N}||\widetilde{\mathcal{T}}_l(x) - \tilde{\mathcal{T}}(x)||_2^2 \tag{7}$$

where $\tilde{\mathcal{T}}_l(x)$ and $\tilde{\mathcal{T}}(x)$ both express the normalized value. In block grafting optimization, only the encapsulated student block is learnable, and the parameters of the migrated student block are updated with a gradient. For network grafting, a similar distillation method is adopted, and the specific calculation formula is as follows.

$$\mathcal{LC}_l^*(x) = \frac{1}{N}||\widetilde{\mathcal{T}}_l^*(x) - \tilde{\mathcal{T}}(x)||_2^2 \tag{8}$$

The difference between network grafting and block grafting is that network grafting needs to optimize a sequence of wrapped student blocks rather than a single student block.

C. FEATURE DISTRIBUTION DISTILLATION

In this paper, we consider the feature distribution information and believe that it is beneficial to direct the student network to a configuration similar to the distribution of the teacher network([22]). To model these patterns among students, we use the minimum squared error MSE loss as a measure of knowledge distillation, referred to as characteristic distribution distillation.

Assuming that $\mathcal{H}_l(x)$ denotes the feature distillation of the *l-th* original teacher block, expressed by Eq.(7). $h_l(x)$ denotes the feature distillation of the *l-th* student block after the *l-th* student block is replaced to the teacher network, expressed by Eq.(8).Then the feature distribution loss between student block and teacher block is calculated as follows:

$$\mathcal{H}_l(x) = B_l \circ B_{l-1} \circ \cdots B_1(x) \tag{9}$$

$$h_l(x) = b_l^* \circ B_{l-1} \circ \cdots B_1(x) \tag{10}$$

$$\mathcal{LF}_l(x) = \frac{1}{N}||\tilde{h}_l(x) - \tilde{\mathcal{H}}_l(x)||_2^2 \tag{11}$$

where $\tilde{h}_l(x)$ and $\tilde{\mathcal{H}}_l(x)$ both express the normalized value.

In network grafting, it is no longer the feature distribution between the individual student block and teacher block, but the gradually accumulated student block and teacher block of the same length. The feature distribution loss of the two is calculated as follows:

$$h_l^*(x) = b_l^* \circ b_{l-1}^* \circ \cdots b_1^*(x) \tag{12}$$

$$\mathcal{LF}_l^*(x) = ||\tilde{h}_l^*(x) - \tilde{\mathcal{H}}_l(x)||_2^2 \tag{13}$$

where $\tilde{h}_l^*(x)$ represents the feature distillation of the sequence of student blocks grafted to the teacher network.

Algorithm 1 Progressive Network With Fusion of Feature and Logits for Few-Shot Knowledge Distillation

Input: Trained teacher model $\mathcal{T}$;Few unlabeled training data $\mathcal{D} = \{x_i\}_{i=1}^{N \cdot K}$

Output: The compact student model $\mathcal{S}$

Stage1: train every student block
for $l = 1 \rightarrow L$ **do**
  Pack b_l in $a_l^{s \rightarrow t}$ and $a_{l-1}^{t \rightarrow s}$ to get b_l^* by Eq.2;
  Graft b_l^* to get $\mathcal{T}_l$ by Eq.3;
  for training times **do**
    Compute $\mathcal{L}_l(x)$ by Eq.14;
    Update the parameters of b_l^*;
  end for
end for
Stage2: train student network
Initialize $\mathcal{T}_1^* = \mathcal{T}_1$
for $l = 2 \rightarrow L$ **do**
  Pack $\mathcal{T}_{1-1}^*$ and b_l^* to get $\mathcal{T}_1^*$ by Eq.4;
  for training times **do**
    Compute $\mathcal{L}_l^*(x)$ by Eq.15;
    Update the parameters of $\left\{b_j^*\right\}_{j=1}^{l}$;
  end for
end for
Merge and get student network $\mathcal{S}$

D. OPTIMIZATION

In this paper, we believe that there is a certain degree of complementarity between the feature distribution and the classification output. Aligning the feature distribution of the student block and the teacher block is beneficial to reduce the l_2 loss value of logits, while optimizing the logits loss can reduce the difference between the feature distribution of the student block and the teacher block. Therefore, this paper designs two loss functions in block grafting and network grafting respectively, and uses parameters to connect the feature distribution loss and logits loss. The formula for calculating the loss function in block grafting is as follows.

$$\mathcal{L}_l(x) = \lambda \mathcal{LC}_l(x) + \mathcal{LF}_l(x) \tag{14}$$

Because of the large difference between logits loss and feature distillation loss, the former is about 1000 times larger than the latter. Therefore, in block grafting, λ is set to 10^{-6} to achieve knowledge transfer between teacher block and student block by complementation of feature distribution and classification output. The calculation formula of loss function in network grafting is as follows.

$$\mathcal{L}_l^*(x) = \mathcal{LC}_l^*(x) + \beta \mathcal{LF}_l^*(x) \tag{15}$$

In the above formula, β is set to 10^{-3}. In the network grafting, the parameters of the student block sequence are optimized by the above loss function, so that the teacher network can realize the supervision of the student network from the feature level and the classification result level. Our proposed method can be summarized as Algorithm 1.

IV. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

Datasets and models. In this paper, comparative experiments were conducted on CIFAR10 and CIFAR100 datasets to verify the effectiveness of the proposed hybrid distillation method. Both CIFAR 10 and CIFAR 100 are Consists of 60,000 color images of 32×32 size. The specific information of the datasets is shown in Table 1.

TABLE 1. Desciption of the datasets.

No.	Datasets	Training	Test	Classes	Domains
1	CIFAR10	50000	10000	10	Image
2	CIFAR100	50000	10000	100	Image

In few-shot setting, K samples are randomly selected from each class of the CIFAR dataset as the training set, where the value of K is taken as 1,5,10. The training set is enhanced by random clipping and random horizontal flip, while the test set remains unchanged. In reference to the setting of paper [9], modified VGG16 Li et al. is adopted as the teacher model, and VGG16-half is used as the student model.

Experimental Details. The method presented in this article is implemented on a Nvidia TITANX Pascal 12GB GPU using PyTorch. Adam algorithm is used for network optimization in all experiments. When batch-size is 64, the following learning rate is effective; when batch-size is other values, the learning rate is scaled according to Batch-size/64 [9]. The learning rates of CIFAR10 dataset in block migration and network migration are set to $2.5*10^{-4}$, 10^{-4}, respectively. The learning rates of CIFAR100 dataset are respectively set as $2.5*10^{-4}$, 10^{-4}. Following [29], the weight decay is set to 0, the gradient and its square operating average are set to 0.9 and 0.999, respectively, and Kaiming initialization is used [30].

B. EXPERIMENTAL RESULTS

Intuitively, knowledge between homogeneous network structures tends to have clearer correlations than knowledge between heterogeneous networks, especially in the case of piecewise distillation. Therefore, in this section, we study

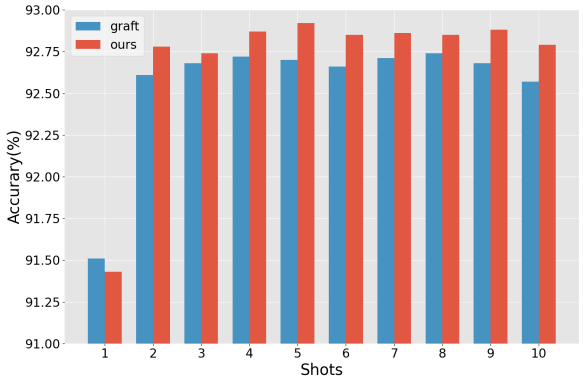
TABLE 2. The performance of few-shot distillation on CIFAR dataset.

Method	CIFAR10			CIFAR100		
	Samples	#Params	Accuracy(%)	Samples	#Params	Accuracy(%)
Teacher (VGG16)	5000	15.0M	92.83	5000	15.0M	69.82
KD (full)	5000	5.4M	92.06±0.17	5000	5.4M	68.31±0.15
FitNet (full)	5000	5.4M	92.75±0.11	5000	5.4M	70.12±0.12
KD(few)	1	5.4M	74.43±1.43	1	5.4M	41.67±1.23
	5		88.53±0.92	5		64.13±0.34
	10		88.76±0.14	10		65.64±0.15
FitNet(few)	1	5.4M	74.43±1.43	1	5.4M	41.67±1.23
	5		88.53±0.92	5		64.13±0.34
	10		88.76±0.14	10		65.64±0.15
FSKD	1	5.4M	87.42±0.84	1	5.4M	48.57±1.14
	5		88.83±0.63	5		65.47±0.18
	10		92.37±0.00	10		67.34±0.12
Cross Distillation	1	5.4M	69.57±1.39	1	5.4M	41.32±0.18
	5		84.91±0.98	5		63.81±0.15
	10		86.61±0.71	10		64.95±0.24
NetGraft(vgg16-half)	1	5.4M	90.74±0.49	1	5.4M	64.22±0.17
	5		92.88±0.07	5		68.16±0.20
	10		**92.89±0.06**	10		68.86±0.03
Ours(vgg16-half)	**1**	**5.4M**	**91.43±0.11**	**1**	**5.4M**	**65.28±0.14**
	5		**92.92±0.05**	**5**		**68.51±0.11**
	10		92.79±0.04	**10**		**69.02±0.06**

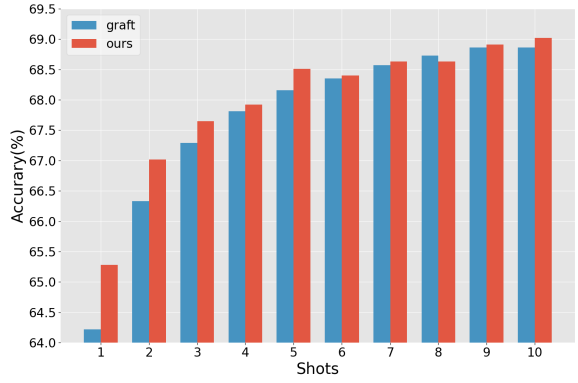
knowledge extraction between homogeneous networks to verify the effectiveness of the proposed method. Table 2 illustrates our experimental results, where we use vgg16-half to reduce the channels of the vgg16 corresponding layer.

- FSKD [26] improves upon KD [13] and FitNet [14].
- Cross Distillation can effectively reduce the estimation error of layer by layer distillation by cross training the hidden layer networks of teachers and students, but the performance of teacher networks may be reduced when students and teachers abandon each other.
- Netgraft does not consider the characteristic distribution information of teacher networks and student networks.

Therefore, as can be seen from the table, our proposed method generally outperforms the current research methods. And as we can seen, Netgraft is the best of the above methods. At 10-shot, our method outperforms Netgraft 0.69%,1.06% on the CIFAR10 and CIFAR100, At 5-shot, our method outperforms Netgraft 0.04%,0.35% on the CIFAR10 and CIFAR100, At 10-shot, our method outperforms Netgraft 0.16% on the CIFAR100.The above methods are all limited to aligning local features distribution from the teacher block or output from the teacher network, and do not consider the complementarity of the two parts. Because the structure of the student network is different from that of the teacher network, it is difficult for the student network to completely imitate the output of the teacher block when only local feature information is considered. At the same time, because of the scarcity of samples in the scene with few-shot distillation, it is difficult to optimize the student network using only the output of the teacher network. The improvement of our method is that it considers both the local features information from the teacher network and the output from the fully connected layer of the teacher network, which optimize and improve the performance of the student network in a complementary manner. In addition, this paper designs a student network with fewer parameters through a series of channel reduction settings to explore the effectiveness of our method.



(a) CIFAR10



(b) CIFAR100

FIGURE 3. Learning with different numbers of samples.

1) LEARNING WITH DIFFERENT NUMBERS OF SAMPLES

To further investigate the effect of the number of training samples on the model distillation effect, we conduct a series of few-shot distillation experiments using our method and

TABLE 3. Learning with different number of samples on CIFAR10 dataset.

Method	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot
NetGraft	**91.51**	92.61	92.68	92.71	92.70	92.66	92.71	92.74	92.68	92.57
Ours	91.43	**92.78**	**92.74**	**92.87**	**92.92**	**92.85**	**92.86**	**92.85**	**92.88**	**92.79**

TABLE 4. Learning with different number of samples on CIFAR100 dataset.

Method	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot
NetGraft	64.22	66.33	67.29	67.81	68.16	68.35	68.57	**68.73**	68.86	68.86
Ours	**65.28**	**67.02**	**67.65**	**67.92**	**68.51**	**68.40**	**68.63**	68.63	**68.91**	**69.02**

TABLE 5. The performance of various channels on CIFAR100 dataset.

Method	1-shot	3-shot	5-shot	7-shot	10-shot
NetGraft(vgg16-128)	62.89±0.36	65.94±0.29	67.66±0.08	67.77±0.20	68.63±0.20
Ours(vgg16-128)	**64.34±0.13**	**66.69±0.28**	**67.86±0.18**	**68.23±0.14**	**68.78±0.10**
NetGraft (vgg16-64)	60.76±0.34	64.65±0.18	66.80±0.12	67.24±0.10	67.98±0.17
Ours(vgg16-64)	**62.51±0.31**	**65.70±0.12**	**67.31±0.14**	**67.45±0.20**	**68.15±0.13**
NetGraft (vgg16-32)	56.40±0.50	62.76±0.35	65.44± 0.12	66.02±0.38	**67.13±0.16**
Ours(vgg16-32)	**58.70±0.59**	**64.78 0.24**	**66.06±0.13**	**66.61±0.13**	67.11±0.90
NetGraft (vgg16-16)	48.01±1.30	59.28±0.70	63.64±0.29	64.47±0.19	65.60±0.21
Ours(vgg16-16)	**49.82±1.89**	**62.21 0.43**	**63.81±0.17**	**64.77±0.14**	**65.66±0.13**

TABLE 6. The performance of various channels on CIFAR10 dataset.

Method	1-shot	3-shot	5-shot	7-shot	10-shot
NetGraft (vgg16-128)	89.86±0.44	92.52±0.15	92.69±0.10	92.58±0.08	**92.84±0.08**
Ours(vgg16-128)	**90.48±0.21**	**92.60±0.18**	**92.82±0.04**	**92.80±0.13**	92.82±0.06
NetGraft (vgg16-64)	87.85±0.74	91.63±0.11	92.46±0.05	**92.72±0.09**	92.69±0.04
Ours(vgg16-64)	**88.69±0.84**	**92.23±0.15**	**92.52±0.17**	92.53±0.05	**92.74±0.05**
NetGraft (vgg16-32)	82.12±1.99	**92.41±0.17**	92.03±0.07	**92.62±0.04**	**92.46±0.11**
Ours(vgg16-32)	**85.54±1.55**	91.36±0.18	**92.03±0.14**	92.29±0.17	92.42±0.15
NetGraft (vgg16-16)	69.85±2.23	**90.79±0.25**	**90.95±0.31**	91.44±0.10	91.65±0.08
Ours(vgg16-16)	**72.73±6.04**	87.66±1.71	90.86±1.63	**91.49±0.38**	**91.76±0.15**

NetGraft in the 1-shot to 10-shot situations, all experiments are implemented on the CIFAR10 and CIFAR100 datasets. As shown in the Table 3 and Table 4, the experimental results of our method are better than NetGraft in most cases. To better see the effect of shot number variation on experimental results, we plot the table as a graph. As shown in Figure 2, as the number of shots increases, the experimental results of our method and NetGraft also improve. At the same time, in most cases, the results of our method are higher than NetGraft, because we use the local feature information of the teacher block which is not used in NetGraft, which can greatly improve the experimental effect when the number of samples is small. In addition, under the 1-shot of the CIFAR10 dataset, our method is slightly worse than NetGraft, because in this case only 10 images can be used to train the model and it is difficult for the student network to learn the local features distribution from the teacher network, which may lead to underfitting of the model, and then the experimental results are lower than NetGraft, But under other experiments on the CIFAR10 dataset, our method outperforms NetGraft.

2) LEARNING WITH DIFFERENT NUMBERS OF CHANNELS

For further model compression, we reduce the number of channels in the vgg16-half network to obtain a student network with fewer parameters. Specifically, by setting the number of channels of vgg16-half to 128, 64, 32, 16, the corresponding networks are named vgg16-128, vgg16-64, vgg16-32, vgg16-16. Based on these four student networks, we do 4 sets of experiments on NetGraft and our method respectively, all experiments are implemented on CIFAR10 and CIFAR100 datasets. As shown in the Table 5 and Table 6, our method outperforms NetGraft in most experiments in most cases of various student networks with fewer parameters. Moreover, as shown in Figure 3, as the number of channels decreases, both our method and NetGraft show a drop in the experimental results, but our method is still better than NetGraft. Especially in the 1-shot case, the results of our method outperform NetGraft by a large margin, because we make full use of the local feature information of the teacher block, making the student network easier to optimize. As the number of samples increases, our method is closer to NetGraft, because after the number of samples increases, the

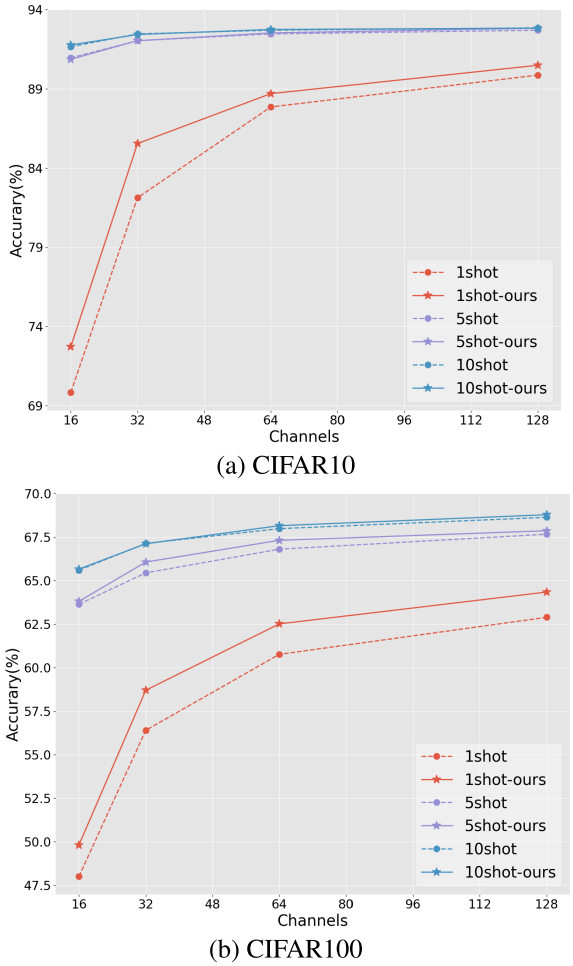


FIGURE 4. Learning with different numbers of channels.

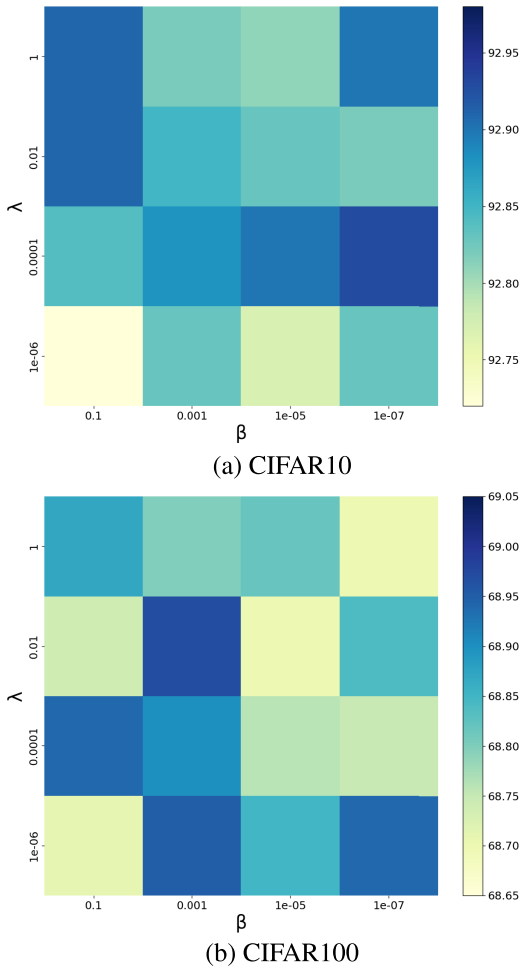


FIGURE 5. Analysis of λ and β. All experiments are under the 10-shot setting.

information contained in it only uses the output of the teacher network, which is enough to optimize the student network.

3) ABLATION STUDY

In this section, we show an ablative analysis to study the contribution of two main components of the learning framework, local features (LF) and network output (NO) on the dataset CIFAR10. The results are shown in Figure 6. From the results, we can obtain that the classification performances of local features are lower than that of the network output. By revealing the close relationship between local features and network output, the proposed method achieves the best performance in all 3-shot, 4-shot and 5-shot situations.

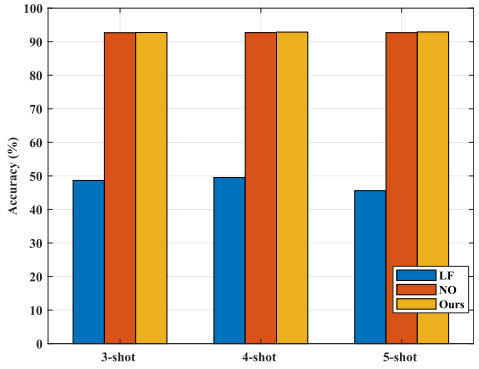


FIGURE 6. The ablative analysis of local features (LF), network output (NO) and our method on the CIFAR10 dataset in 3-shot, 4-shot and 5-shot situations.

4) ANALYSIS OF λ AND β

To balance the logits loss and feature distillation loss used in two training stages of our method, we introduced two superparameters: λ and β. And to determine these parameters, we performed a number of experiments in the 10-shot on the CIFAR10 and CIFAR100 datasets and used the grid search method to select the superparameters. In Figure 5, we can see that the choice of superparameter has only a slight effect on the experimental results in the 10-shot of CIFAR10 and CIFAR100 datasets, in other words, the superparameters we introduce are not sensitive to the performance of the experimental results. Lastly, to better show the effectiveness of our method, in all the other experiments in this paper, we only selected superparameter values with moderate performance in the grid search experiments: λ was set to 10^{-6} and β was set to 10^{-3}.

V. CONCLUSION

We propose a progressive network grafting with local features embedding for few-shot knowledge distillation, which grafts student blocks one by one to the corresponding trained teacher blocks for training. Using feature distillation can align the feature distribution of student network and teacher network, and using logits distillation can align the student network with the final prediction result of teacher network. The advantage of this method is that it not only considers the influence of the feature distribution of middle-level network on the output results, but also the influence of the output results on the feature distribution. By combining the two parts, the performance of student network can be improved. Several experimental results show that the proposed method achieves state-of-the-art performance.

REFERENCES

[1] H. Tang, H. Liu, W. Xiao, and N. Sebe, "When dictionary learning meets deep learning: Deep dictionary learning and coding network for image recognition with limited data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2129–2141, May 2021.
[2] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Comput. Vis. Image Understand.*, vol. 189, Dec. 2019, Art. no. 102805.
[3] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: A review through deep learning techniques for crowd analysis," *J. Big Data*, vol. 6, pp. 1–27, Jun. 2019.
[4] 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.
[5] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11962.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[7] T. Li, J. Li, Z. Liu, and C. Zhang, "Few sample knowledge distillation for efficient network compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14639–14647.
[8] H. Bai, J. Wu, I. King, and M. Lyu, "Few shot network compression via cross distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3203–3210.
[9] C. Shen, X. Wang, Y. Yin, J. Song, S. Luo, and M. Song, "Progressive network grafting for few-shot knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 3, pp. 2541–2549.
[10] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*.
[11] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[12] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[14] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.
[15] Y.-J. Zheng, S.-B. Chen, C. H. Q. Ding, and B. Luo, "Model compression based on differentiable network channel pruning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 15, 2022, doi: 10.1109/TNNLS.2022.3165123.
[16] X. Cai, J. Yi, F. Zhang, and S. Rajasekaran, "Adversarial structured neural network pruning," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, Nov. 2019, pp. 2433–2436.
[17] H. Zhang, L. Liu, H. Zhou, W. Hou, H. Sun, and N. Zheng, "AKECP: Adaptive knowledge extraction from feature maps for fast and efficient channel pruning," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 648–657.
[18] D. Nguyen, S. Gupta, K. Do, and S. Venkatesh, "Black-box few-shot knowledge distillation," 2022, *arXiv:2207.12106*.
[19] J. Song, Y. Chen, J. Ye, and M. Song, "Spot-adaptive knowledge distillation," *IEEE Trans. Image Process.*, vol. 31, pp. 3359–3370, 2022.
[20] X. Liang, X. Zhao, C. Zhao, N. Jiang, M. Tang, and J. Wang, "Task decoupled knowledge distillation for lightweight face detectors," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2184–2192.
[21] G. K. Nayak, K. R. Mopuri, V. Shaj, V. B. Radhakrishnan, and A. Chakraborty, "Zero-shot knowledge distillation in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4743–4751.
[22] Z. Zhou, Y. Zhou, Z. Jiang, A. Men, and H. Wang, "An efficient method for model pruning using knowledge distillation with few samples," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 2515–2519.
[23] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1389–1397.
[24] S. Chen, W. Wang, and S. J. Pan, "Deep neural network quantization via layer-wise optimization using limited training data," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 3329–3336.
[25] 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.
[26] T. Li, J. Li, Z. Liu, and C. Zhang, "Knowledge distillation from few samples," in *Proc. ICLR*, 2018, pp. 1–14.
[27] S. Chen, W. Wang, and S. J. Pan, "Cooperative pruning in cross-domain deep neural network compression," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2102–2108.
[28] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "Self-supervised knowledge distillation for few-shot learning," 2020, *arXiv:2006.09785*.
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

WEIDONG DU was born in Qinghai, in 1997. He received the Master of Business Administration degree from Shanghai University, in 2010. He is currently pursuing the Ph.D. degree with Southeast University. He joined Focusight, in 2014, as an Executive Vice President. As one of the earliest machine vision practitioners in China, he has nearly 20 years of industry experience in machine vision automation industry and has professional technical ability and rich experience in the field of machine vision measurement and testing. He holds 11 invention patents and 20 utility models.