Neural Processing Letters (2023) 55:6165–6180
https://doi.org/10.1007/s11063-022-11132-w

Knowledge Fusion Distillation: Improving Distillation with Multi-scale Attention Mechanisms

Linfeng Li[1] · Weixing Su[2] · Fang Liu[3] · Maowei He[2] · Xiaodan Liang[2]

Accepted: 12 December 2022 / Published online: 3 January 2023

Abstract
The success of deep learning has brought breakthroughs in many fields. However, the increased performance of deep learning models is often accompanied by an increase in their depth and width, which conflicts with the storage, energy consumption, and computational power of edge devices. Knowledge distillation, as an effective model compression method, can transfer knowledge from complex teacher models to student models. Self-distillation is a special type of knowledge distillation, which does not to require a pre-trained teacher model. However, existing self-distillation methods rarely consider how to effectively use the early features of the model. Furthermore, most self-distillation methods use features from the deepest layers of the network to guide the training of the branches of the network, which we find is not the optimal choice. In this paper, we found that the feature maps obtained by early feature fusion do not serve as a good teacher to guide their own training. Based on this, we propose a selective feature fusion module and further obtain a new self-distillation method, knowledge fusion distillation. Extensive experiments on three datasets have demonstrated that our method has comparable performance to state-of-the-art distillation methods. In addition, the performance of the network can be further enhanced when fused features are integrated into the network.

Keywords Multi-scale attention mechanism · Knowledge distillation · Selective dense feature connections · Model compression

✉ Fang Liu
lf@tiangong.edu.cn

Linfeng Li
llftjpu@163.com

Weixing Su
suweixing@tiangong.edu.cn

Maowei He
hemaowei@hotmail.com

Xiaodan Liang
lxdtjpu@163.com

[1] School of Artificial Intelligence, Tiangong University, Tianjin 300387, China

[2] School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

[3] School of Software, Tiangong University, Tianjin 300387, China

1 Introduction

Deep neural networks [5] have been widely used for object detection [20, 30], wind speed prediction [3], cyber-attacks detection [19], etc. However, in some real-time applications such as autonomous driving and video analysis, the accuracy and response time of deep neural networks are strictly required. In addition, large networks are difficult to deploy on edge devices due to the storage and computational burden. Many efforts have been proposed to solve this problem. For example, Bhosale and Patnaik [2] designed a lightweight neural network, LDC-Net, to detect COVID-19 cases, with encouraging results on the Raspberry Pi. In addition to this, various techniques have been proposed to solve this problem, including pruning [41], quantization and [38] knowledge distillation [8, 25, 40]

Self-distillation, as a special form of knowledge distillation, uses the network's own knowledge to guide its own training, thereby boosting the performance of the model. When the performance of a small network reaches the same level as that of a large network, the large network can be replaced by the small network. In the deployment phase, small models are used for inference, thus model compression and inference acceleration are achieved [2]. Self-distillation can also be seen as a means of significantly improving network performance [35]. Zhang et al. [35] first proposed a self-distillation framework, BYOT (be your own teacher), which uses the knowledge provided by the deepest layer of networks to guide itself for training. It solves the slow training problem of traditional knowledge distillation. Further, Zhang et al. [36] introduced an attention mechanism, which significantly improves the accuracy of branches. Zhang et al. [37] redesigned the branches and the attention mechanism to further improve the accuracy of branches and reduce the computational complexity. Ji et al. [15] proposed a new self-distillation framework which exploits the BiFPN [26] to fuse features. The fused features are used for self-distillation. However, there are some slight downsides to these efforts: (1) [35–37] all utilized the features from the deepest layer to guide students in training. We found that the deepest layer of the network does not provide the best distillation knowledge. Although the feature of the deepest layer contains rich semantic information, it lacks the spatial information that is equally important. (2) [15] used BiFPN for self-distillation. However, BiFPN is a late feature fusion (LFF) method. The network cannot directly benefit from LFF. In contrast, early feature fusion (EFF) can integrate the fused feature maps into the network to further improve the performance of the network. However, since features from early stages contain weak semantic information [18], it is challenging to efficiently fuse features from different stages.

Based on the above mentioned, in this paper, we have investigated how to provide better knowledge for branches through EFF and proposed a new distillation method, KFD. Specifically, we found that dense feature connections conflict with the self-distillation framework as features in the early stages contain weak semantic information (as shown in Table 1). To solve this problem, we have proposed a Selective Dense Feature Connection Module (SDFC). Based on SDFC, a new self-distillation framework, KFD is proposed. The advantages of KFD are as follows: (1) The combination of dense feature connections and SDFC provides better knowledge for self-distillation; (2) the fused feature maps can be integrated into the network to further enhance the performance of the network. Extensive experiments have shown that KFD has comparable performance to the state-of-the-art distillation methods. In addition, the performance of the network can be further enhanced when fused features are integrated into the network.

The contributions of this paper are summarized as follows:

- We use early feature fusion to provide better knowledge for self-distillation. To the best of our knowledge, this is the first work to combine early feature fusion with self-distillation.
- We have found that early feature fusion degrades the performance of self-distillation and propose SDFC to solve this problem. Based on SDFC, a new self-distillation framework, KFD, is proposed, which can provide rich knowledge for branches.
- We have conducted extensive experiments to demonstrate the ability of KFD to improve model performance and compress models. In addition, the performance of the network can be further enhanced when fused features are integrated into the network.

2 Related Work

Knowledge distillation As a common method of model compression, knowledge distillation allows the knowledge of large models to be transferred to small models. Knowledge distillation was first proposed by Hinton et al. [8], who they trained student networks using logits from teacher networks. Since then, various efforts have defined different distillation knowledge. Romero et al. [23] used intermediate features of the teacher network to guide the training of the student network. Zagoruyko and Komodakis [34] used attention maps as distillation knowledge. Yim et al. [32] used the inter-layer relationship of the feature map for knowledge distillation. However, there are obvious limitations of conventional knowledge distillation: (1) the training of small models depends on a pre-trained large model; and (2) the model capacity gap between the student and the teacher affects the training of the small model.

Some efforts have been done to improve the latter. For example, Mirzadeh et al. [21] introduced a teaching assistant network to compensate for the impact of the gap in model capacity between teachers and students on training. In addition, an early-stop strategy was proposed to mitigate this problem [4]. Zhang et al. [35] made improvements on both problems. They proposed a self-distillation framework, BYOT, which allows the deep structure of the network to guide the shallow structure of the network for training. Since there is only one network in the training process, the training time is significantly reduced. Zhang et al. [36] combined self-distillation and attention mechanisms to significantly improve the performance of branches. Zhang et al. [37] redesigned the branches and the attention mechanism to further improve the performance of branches and reduce the computational complexity.

Multi-scale attention mechanisms The significance of attention mechanisms has been extensively studied. It draws on human attention thinking and can enhance important feature information and suppress less important ones. Wang et al. [27] constructed an end-to-end convolutional neural network with a bottom-up top-down attention mechanism. SE-Net [12] enhances the channel-level feature using global average pooling and fully connected layers. BAM [22] and CBAM [28] further emphasize feature information in the channel and spatial dimensions. SKNet [17] uses a channel-level attention mechanism to select the appropriate convolutional kernel for different samples. CA-Block [9] further encodes the spatial information along the horizontal and vertical directions separately, thus capturing the positional relationships of the features. These efforts integrate attention mechanisms into the backbone to improve network performance. In this paper, we use the attention mechanism to selectively fuse features, which suppresses the noise of features from the shallow layers of the network.

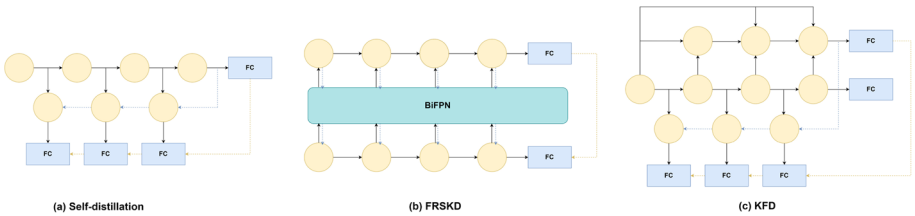


Fig. 1 Comparison of various self-distillation methods. The black line is the forward path; the green line is response-based distillation; the blue line is feature-based distillation. **a** Using the feature and the prediction from the deepest layer for self-distillation [35–37]; **b** Feature maps of the network are fed into BiFPN to generate refined feature maps. The refined feature maps are then used for classification to obtain predictions. Finally, features and the prediction are used to self-distillation [15]; **c** Our proposed method. Feature maps from different stages are fused by dense feature connections as well as SDFC. The fused feature maps are used for classification to obtain predictions. Finally, the fused feature maps and predictions are used for self-distillation

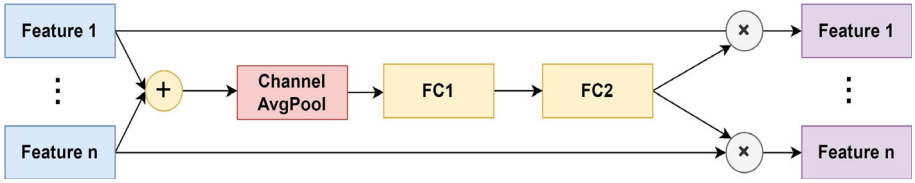


Fig. 2 Selective dense feature connection module

3 Method

3.1 Selective Dense Feature Connection Module

It is well known that the features extracted from deep layers of the network contain rich semantic information, which facilitates the determination of what the samples are. The features extracted from the early stages of the network contain rich spatial information, which is also beneficial for classification. However, the features from the early stages of the network contain weak semantic information [18], which is not favorable for classification. To effectively fuse the feature maps from different stages, we propose a selective dense feature connection module. Some details about SDFC are as follows.

As shown in Fig. 2, we first simply fuse n feature maps at different scales by an element-level summation to obtain the feature map M:

$$M = \sum_{i=1}^{n} F_i \tag{1}$$

Then, channel-level global average pooling is used to embed the channel information into the spatial dimension to obtain a spatial-level attention map Fs. Specifically, the attention map Fs is calculated through the channel dimension:

$$Fs = Avg(M) = \frac{1}{C} \sum_{i=1}^{C} M(i, h, w) \tag{2}$$

Finally, two fully connected layers and a softmax operation are used to generate the respective weights of the features at different scales:

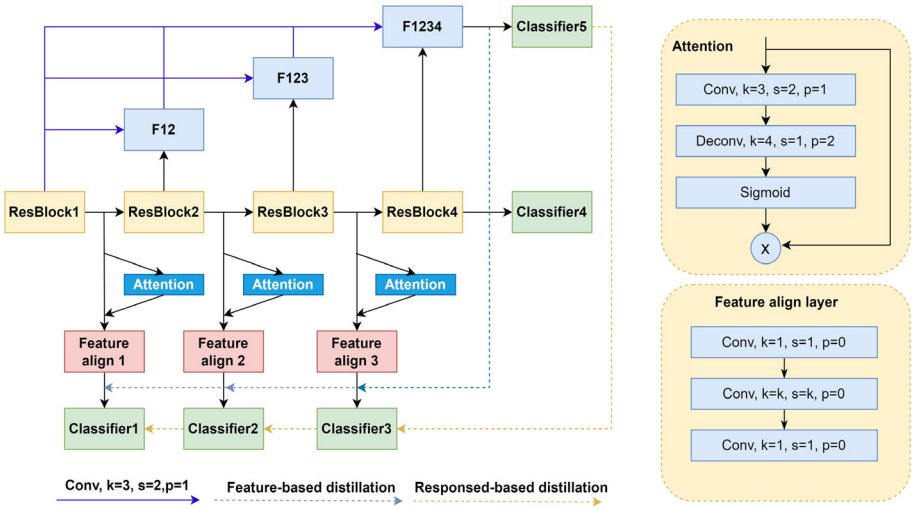


Fig. 3 Knowledge fusion distillation (F1234 is the final multi-scale feature map, which is used to guide the training of branches; the blue line represents feature-based distillation; the green line represents response-based distillation; k refers to kernel, s refers to stride, and p refers to padding.)

$$Weight = Softmax(FC_2(\delta(Bn(FC_1(Fs))))) \tag{3}$$

where δ for ReLU [6], Bn for Batch Normalization [14], and FC for the fully connected layer. In addition, to reduce the computational complexity, the first fully connected layer maps the feature Fs to dimension d:

$$d = max(H * W/r, L) \tag{4}$$

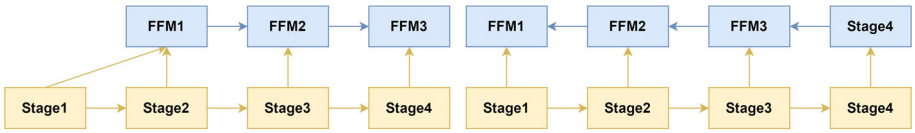
where L denotes the minimum value of d and r is the reduction ratio (L=16, r=16 in our experimental setup).

In summary, SDFC reweights the feature maps to be fused using the two fully connected layers and the spatial information of the feature maps. It emphasizes the important feature maps and suppresses the less important ones. The feature maps obtained by SDFC can be considered as a fusion of effective features from different stages. Therefore, the fused feature map can provide better guidance to the network.

3.2 Knowledge Fusion Distillation

3.2.1 KFD Framework

The main purpose of KFD is to obtain a feature map containing multi-scale features through EFF, which can provide better knowledge for the network itself. As shown in Fig. 4, EFF fuses the feature maps of different stages in the forward propagation of the network; LFF fuses the feature maps from different stages by upsampling the feature map of the deepest layer of the network. Compared with LFF, the feature maps obtained with EFF can be integrated into the network to further improve the performance of the network. As shown in Fig. 3, extra branches are added in different stages of the network, which include an attention mechanism, a feature alignment layer, and a classifier. Dense feature connections and SDFC are used to fuse features from different stages. Eventually, the richest feature map is transferred to a



(a) early feature fusion (b) late feature fusion

Fig. 4 Early feature fusion and late feature fusion. (FFM refers to fused feature maps)

fully connected layer to obtain the probability distribution. Then, the feature map and the probability distribution are used to guide the training of branches. All branches can be dropped without introducing additional computational complexity.

In the KFD framework, each branch is subject to three supervised information except for the deepest branch and the additional feature fusion branch.

The first supervised information is the cross-entropy loss, which is calculated with q^i and Y. Note that q^i is the the softmax layer's output of the ith classifier and Y is the ground truth labels. In addition, α is used to adjust the supervised information.

The second supervised information is the Kullback–Leibler divergence between q^i and q^k. As shown in Fig. 3, q^k is the softmax layer's output of classifier 5.

The last supervised information is a hint from the feature map containing multi-scale features. Hint refers to the output of the hidden layer in the teacher model, which guides the student network to generate more accurate feature maps. Note that β is used to adjust the supervised information, F_i refers to the feature map of the ith branch, and F_k is the richest feature map.

In summary, the loss function of the whole network consists of the loss functions of all classifiers, which can be written as (the recommended hyper-parameters α and β are 0.1 and 5e−5):

$$loss = \sum_{i}^{n}(1-\alpha)\cdot CrossEntropy(q^i, Y) + \sum_{i}^{n-2}(\alpha \cdot KL(q^i, q^k) + \beta \cdot \|F_i - F_k\|_2^2) \quad (5)$$

$$F_k = SDFC(F_1, ..., F_n) \quad (6)$$

3.2.2 Implementation Details

Dense feature connections In order to maintain the consistency of the feature map, it is necessary to adjust the channel and size of the feature map. Although average pooling has low computational complexity, it can corrupt feature information (as shown in Sect. 4.2). Therefore, we use 3×3 convolution to adjust both channel and the feature map size.

Attention mechanism As shown in Fig. 3, we use the bottom-up top-down attention mechanism [27], which consists of convolution, deconvolution, and a sigmoid operation.

Feature align layer As shown in Fig. 3, to reduce the computational complexity, we refer to the Bottleneck structure [7], which uses 1×1 convolution to reduce the computational complexity.

Table 1 Comparison of self-distillation in three cases (SD-DFC refers to self-distillation with dense feature connections; SD refers to self-distillation; SD-SDFC refers to self-distillation with SDFC; The numbers in the table are the accuracy for classifier4)

Distillation	SD	SD-DFC	SD-SDFC
Resnet18	78.99	**79.34** (+**0.35**)	**79.48** (+**0.49**)
Resnet34	79.91	79.76 (−0.15)	**80.31** (+**0.55**)
Resnet50	80.37	80.54 (−0.27)	**81.07** (+**0.53**)

Bolded numbers are the accuracy that higher than the SD

4 Experiments

We have evaluated the KFD framework on five convolutional neural networks (ResNet [7], WideResNet [33], MobileNet [10], ResNeXt [29], VGG [24]) and three datasets (CIFAR10 [16], CIFAR100 [16], Tiny-ImageNet). These networks include deep networks, wide networks, and lightweight networks. Learning rate decay and some methods to prevent models overfitting (data augmentation, l2 regularization and early stopping) were used in the training process. All experiments were implemented by Pytorch on Tesla P100.

SGD with learning rate decay and momentum are utilized to optimize neural networks. On the CIFAR10 and CIFAR100 datasets, neural networks were trained for 200 epochs and the learning rate was divided by 10 at the 75th, 130th, and 180th epoch. Batch size and initial learning rate are set to 128 and 0.1, respectively. On the Tiny-ImageNet dataset, neural networks were trained for 100 epochs, and the learning rate was divided by 10 at the 30th, 60th, and 90th epoch. Batch size and initial learning rate are set to 64 and 0.1, respectively.

The hyper-parameters used in the experiments are as follows. (1) The hyper-parameters α, β and T for this paper are set to 0.1, 5e−5 and 3 respectively; (2) The hyper-parameters L and r for SDFC are set to 16 and 16 respectively; (3) The hyper-parameters in the other distillation methods are set to the numbers recommended in the corresponding papers. For example, α, β and T are set to 1, 8 and 4 in DKD [39]; α, β and T are set to 0.1, 1e-6 and 3 in BYOT [35].

The following metrics are used in the experiments: (1) Accuracy: probability of correct prediction in all samples, which indicates the performance of the model; (2) FLOPs (floating point operations per second): it measures the complexity of the model; (3) Storage: the size of the model, which measures how much storage the model takes up; (4) Inference speed: the time taken to infer the test dataset of CIFAR100 when models are deployed on Tesla P100; (5) Noise level: The noise level refers to the value of σ in Gaussian noise (the larger the σ, the higher the noise level). Please note that all numbers from Table 1 to Table 9 refer to accuracy. The figures in Tables 10 and 11 include Accuracy, FLOPs, Storage and Inference speed. The numbers in Tables 12 and 13 indicate the accuracy and the noise level.

4.1 Why Dense Feature Connections Lower the Performance of the Self-Distillation?

As shown in Table 1, as the depth of the model increases, the improvement of self-distillation with dense feature connections based on self-distillation gradually decreases from 0.35% to −0.27%. In contrast, self-distillation with SDFC performs well for models with different depths. This illustrates that the feature maps from the early stages of the network contain noise. This makes sense because the feature maps from the early stages of the network contain weak semantic information according to Lin et al. [18], which is not conducive to

Table 2 Comparison of different downsampling method (The numbers in the table are the accuracy for different classifiers)

Model	Downsample	Classifier1	Classifier2	Classifier3	Classifier4
Resnet18	1×1 Conv + Avgpool	70.37	75.88	78.48	78.94
	3×3 Conv	**70.63**	**76.43**	**79.07**	**79.48**
Resnet34	1×1 Conv + Avgpool	**71.96**	77.95	79.97	80.12
	3×3 Conv	71.61	**78.09**	**80.17**	**80.31**
Resnet50	1×1 Conv + Avgpool	70.56	76.64	79.93	80.53
	3×3 Conv	**71.96**	**77.65**	**80.63**	**81.07**

Bolded numbers are the higher accuracy obtained with distillation when different downsampling methods are used

Table 3 Comparison of integrate or not (The numbers in the table are the accuracy for different classifiers)

Model	Integrate	Classifier1	Classifier2	Classifier3	Classifier4	Ensemble
Resnet18	✗	70.63	76.43	79.07	79.48	79.95
	✓	**71.05**	**77.64**	**79.31**	**79.71**	**80.39**
Resnet50	✗	71.96	77.65	80.63	81.07	81.29
	✓	**72.66**	**78.82**	**81.36**	**81.58**	**82**
Resnet101	✗	71.82	78.35	81.41	81.63	81.91
	✓	**72.37**	**78.54**	**81.77**	**82.08**	**82.4**

Bolded numbers are the higher accuracy

classification. Therefore, SDFC is necessary to emphasize the important features and suppress the unimportant ones.

4.2 The choice of Downsampling Method

In this subsection, we compare different downsampling methods. In method 1, the 1x1 conv is used to adjust the channels and the average pooling is used to reduce the size of the feature map. In method 2, a 3x3 conv is used to adjust both the channel and the size of the feature map. As shown in Table 2, the method 2 performes better on 3 Resnet models. It makes sense, since average pooling can destroy spatial features [1, 13].

4.3 What if We Integrate the Fused Features Into the Backbone?

One of the most significant advantages of early feature fusion is that the fused features can be integrated into the backbone. In Table 3, we show the distillation effect when fused features are integrated into the network (to reduce the number of parameters and FLOPs, the 3×3 DWConv [11] is used to replace the 3×3 Conv). It can be observed that the performance of all classifiers is further improved after the fused feature maps are integrated into the backbone. This is due to two reasons: (1) efficient feature reuse and (2) feature connections enhance gradient propagation.

Table 4 Comparison of SDFC and SKNet on CIFAR100 (The numbers in the table are the accuracy for different classifiers or ensemble; Ensemble means that all classifiers are used to predict)

Model	Fusion method	Classifier1	Classifier2	Classifier3	Classifier4	Ensemble
Resnet18	SDFC	**70.63**	**76.43**	**79.07**	**79.48**	**79.95**
	SKNet	70.29	76.29	79.01	79.32	79.78
Resnet34	SDFC	**71.61**	**78.09**	**80.17**	**80.31**	**81.27**
	SKNet	71.6	77.56	79.99	80.25	80.8
Resnet50	SDFC	**71.96**	**77.65**	**80.63**	**81.07**	**81.29**
	SKNet	71.08	76.87	80.03	80.06	80.77

Bolded numbers are the higher accuracy

4.4 Comparison of SDFC with SKNet for Distillation on CIFAR100

SKNet [17] uses channel-level information to fuse different features adaptively. However, as shown in Table 4, the distillation becomes less effective as the network becomes deeper. We believe that this is due to the fact that features from the early stages contain weak semantic information [18]. Therefore, what really matters for features of the early stages is the spatial information. So we propose SDFC (as shown in Fig. 2). We summarize the experiments on 3 Resnet models and demonstrate the excellence of SDFC at Table 4.

4.5 What has Attenion Learned When Different Features are Used for Distillation?

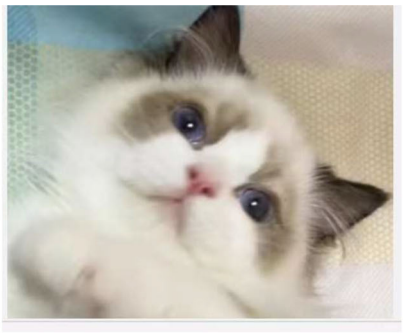
To demonstrate more intuitively that the feature map obtained by SDFC is better knowledge, we summarize the visualization experiments. As shown in Fig. 5, the images on the first row are the feature maps for different stages of WRN-38-2. It can be observed that feature maps from the shallow layer are rich in detailed features and feature maps from the deep layer are rich in abstract features. Next, we use different feature maps for distillation (the feature map from the deepest layer and the feature map obtained by SDFC) and compare the results. Specifically, we have visualized the output of attention on different branches. It can be observed that the results of distillation with the fused feature map contain richer feature.

4.6 Results on CIFAR10

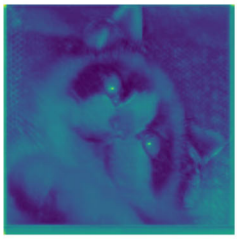
Table 5 shows results of the experiments on CIFAR10. The numbers highlighted in bold are the highest accuracy. It can be observed that (1) the average accuracy is increased by 1.29%, and (2) all the second branches of the neural networks have higher accuracy than the baseline, which can achieve model compression as well as inference acceleration to some extent.

4.7 Results on CIFAR100

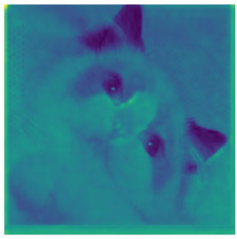
Table 6 shows the results of the experiments on CIFAR100. The numbers highlighted in bold are the highest accuracy. It can be observed that (1) the average accuracy is increased by 4.17%; (2) the accuracy of the third branch of all the neural networks is higher than the baseline and (3) KFD is valid for lightweight networks, deep networks and wide networks.



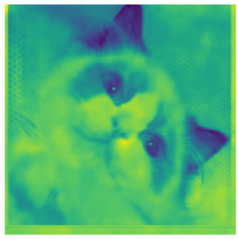
(a) Original image



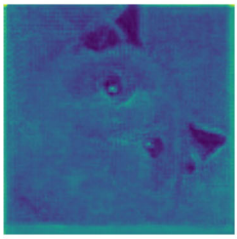
(b) baseline-stage1



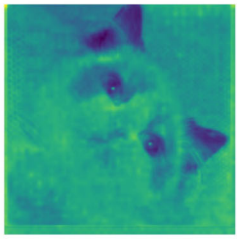
(c) baseline-stage2



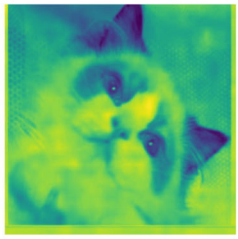
(d) baseline-stage3



(e) Attention1-DL



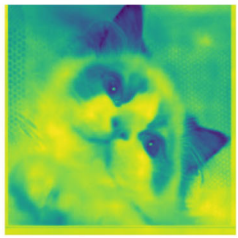
(f) Attention2-DL



(g) Attention3-DL

(h) Attention1-SDFC

(i) Attention2-SDFC



(j) Attention3-SDFC

Fig. 5 Feature maps for different stages (Attention-DL is the feature map obtained by distillation with the feature map from the deepest layer; Attention-SDFC is the feature map obtained by distillation with the fused feature map; Baseline refers to the effect of a network that has not been distilled)

Table 5 Results on CIFAR10 (Baseline refers to the effect of a network that has not been distilled; Ensemble refers to all classifiers being used for prediction; The numbers in the table are the accuracy for different classifiers or ensemble)

model	Baseline	Classifier1	Classifier2	Classifier3	Classifier4	Ensemble
Resnet18	94.25	92.21	94.55	95.39	**95.56**	95.54
Resnet50	94.53	92.61	95.1	95.64	95.76	**95.91**
WRN-38-2	94.54	92.48	94.87	95.63	**95.71**	95.66
WRN-26-2	94.16	92.35	94.28	95.12	**95.35**	95.25
VGG	93	93.27	94.22	94.18	94.15	**94.4**

Bolded numbers are the highest accuracy

Table 6 Results on CIFAR100 (Baseline refers to the effect of a network that has not been distilled; Ensemble refers to all classifiers being used for prediction; The numbers in the table are the accuracy for different classifiers or ensemble)

Model	Baseline	Classifier1	Classifier2	Classifier3	Classifier4	Ensemble
Resnet18	77.09	70.63	76.43	79.07	79.48	**79.95**
Resnet50	77.68	71.96	77.65	80.63	81.07	**81.29**
Resnet101	77.98	71.82	78.35	81.41	81.63	**81.91**
WRN38-2	78.29	71.76	78.24	81.11	81.44	**81.62**
WRN50-2	78.48	73.13	78.28	80.92	81.64	**82.43**
MobilenetV3	60.69	58.67	66.29	68.12	65.8	**69.7**
ResNeXt50-32-4	79.42	72.83	78.81	80.97	81.14	**81.73**
VGG-19	72.25	72.68	75.54	75.45	75.04	**76.65**

Bolded numbers are the highest accuracy

Table 7 Results on Tiny-ImageNet (Baseline refers to the effect of a network that has not been distilled; Ensemble refers to all classifiers being used for prediction; The numbers in the table are the accuracy for different classifiers or ensemble)

Model	Baseline	Classifier1	Classifier2	Classifier3	Classifier4	Ensemble
Resnet18	60.59	52.19	58.69	61.76	62.77	**64.75**
Resnet34	62.01	53.77	61.24	63.85	64.45	**66.47**
Resnet50	64.32	55.69	62.24	65.55	66.48	**68.04**
WRN-26-2	63.88	55.43	61.33	63.97	66.36	**66.71**
WRN38-2	65.46	55.15	62.01	66	68.02	**68.26**
ResNeXt26-32-4	63.93	55.84	61.59	63.79	65.93	**67.05**
VGG-19	57.75	56.2	62.37	63.43	61.77	**65.36**

Bolded numbers are the highest accuracy

4.8 Results on Tiny-ImageNet

Table 7 shows the results of the experiments on Tiny-ImageNet. The numbers highlighted in bold are the highest accuracy. It can be observed that (1) the average accuracy is increased by 4.1% and (2) the accuracy of the third branch of almost all the neural networks is higher than the baseline.

Table 8 Comparison on CIFAR10 (KFI refers to the effect of the fused features being integrated into the backbone; Baseline refers to the effect of a student network that has not been distilled; KFI† refers to the result of the ensemble, i.e. all classifiers are used to predict; The numbers in the table are the accuracy for different distillation methods)

Teacher	Student	Baseline	BYOT	SCAN	AdaIN	SD	DKD	KFD	KFI	KFI†
Resnet34	Resnet18	94.25	95.33	95.15	95.16	95.21	95.22	**95.56**	**95.84**	**95.86**
Resnet101	Resnet50	94.53	95.15	95.63	95.5	95.53	95.09	**95.76**	**96.02**	**96.08**
WRN-50-2	WRN-38-2	94.54	94.98	95.62	95.41	95.49	95.29	**95.71**	**95.92**	**96.3**
WRN-50-2	WRN-26-2	94.16	95	95.2	95.15	95.04	95.03	**95.35**	**95.73**	**95.84**
Resnet34	VGG	93	93.83	93.98	93.86	93.85	93.12	**94.15**	**95.3**	**95.44**

Bolded numbers are the accuray proposed in this paper that higher than other distillation methods

Table 9 Comparison on CIFAR100 (KFI refers to the effect of the fused features being integrated into the backbone; Baseline refers to the effect of a student network that has not been distilled; KFI† refers to the result of the ensemble, i.e. all classifiers are used to predict; The numbers in the table are the accuracy for different distillation methods)

Teacher	Student	Baseline	BYOT	SCAN	AdaIN	SD	DKD	KFD	KFI	KFI†
Resnet34	Resnet18	77.09	78.64	78.99	78.63	78.71	80.09	79.48	79.71	**80.39**
Resnet101	Resnet50	77.24	80.56	80.45	80.1	79.57	79.66	**81.07**	**81.58**	**82**
WRN-50-2	WRN-38-2	78.29	80.98	80.58	80.53	79.72	80.5	**81.44**	**82**	**82.35**
WRN-50-2	WRN-26-2	77.96	78.83	80.2	80.03	79.44	79.84	**80.34**	**81.48**	**81.83**
Resnet34	VGG	72.25	74.34	74.43	74.38	73.4	72.74	**75.04**	**76.25**	**77.63**

Bolded numbers are the accuray proposed in this paper that higher than other distillation methods

4.9 Comparison with Other Distillation Methods

In this subsection, we have compared KFD with other distillation methods over the last few years on the CIFAR10 and CIFAR100 datasets, which include BYOT [35], SCAN (scalable neural networks) [36], AdaIN (adaptive instance normalization) [31], SD (self-distillation) [37] and DKD (decoupled knowledge distillation) [39]. Among them, BYOT, SCAN and SD are self-distillation methods, the teacher and student configurations are for AdaIN and DKD. Of these, DKD is the state-of-the-art distillation method. As shown in Tables 8 and 9, it can be observed that (1) KFD becomes more effective as the depth of the network increases; (2) For very deep networks, KFD outperforms the state-of-the-art distillation method currently available, DKD; (3) even in networks that are not very deep (e.g., Resnet18, VGG), KFD has comparable performance with the current state-of-the-art methods.

4.10 Model Compression and Inference Acceleration

In addition to improving the performance of the network, KFD can be used for model compression. This can be achieved in two ways: (1) using KFD to improve the performance of branches and then dropping the rest of the network (as shown in Table 10); (2) Using KFD to enhance the performance of small networks, then replacing large networks with small networks (as shown in Table 11).

As shown in Table 10, we have highlighted the figures of the branch that with the least number of parameters for a higher accuracy than the baseline. It can be observed that in the

Table 10 Model compression and inference acceleration on CIFAR100

	Indicator	Baseline	Classifier1	Classifier2	Classifier3	Classifier4
Resnet18	Accuracy	77.06	70.63	76.43	**79.07**	79.48
	FLOPs	0.55 G	0.45×	0.67×	**0.9**×	1×
	Storage	11 M	0.12×	0.13×	**0.42**×	1×
VGG	Accuracy	72.25	**72.68**	75.54	75.45	75.04
	FLOPs	0.4 G	**0.27**×	0.6x	0.95×	1×
	Storage	21.72 M	**0.08**×	0.2x	0.79×	1×
Resnet101	Accuracy	77.98	71.82	**78.35**	81.41	81.63
	FLOPs	2.52 G	0.59×	**0.71**×	1.38×	1×
	Storage	42 M	0.08×	**0.2**×	1.29×	1×

Bold numbers are indicators of the first branch with higher accuracy than baseline

Table 11 Model compression and inference acceleration on CIFAR100

Dataset	Indicators	Distilled resnet18	Resnet34	Resnet50
CIFAR100	Accuracy	**79.48**	77.48	77.68
	Inference Speed	**3.99 s**	5.59 s	9.24 s
	Storage	**11.22 M**	21.32 M	23.7 M

Bolded numbers are the highest accuray, the fastest inference speed and the smallest storage

Table 12 Robustness analysis (the model is resnet18 distilled with KFD; the distilled resnet18 is trained and tested on CIFAR100 with different noise levels)

Noise-level	0.003	0.005	0.007	0.009
Accuracy	79.41	79.24	78.93	78.91 (−**0.5**)

Bolded numbers are the decreased numbers when the noise level is increased

average case, the accuracy is improved by 0.93%, the storage is reduced by 76% and the FLOPs are reduced by 37%. Among them, the FLOPs and storage of VGG are reduced by 73% and 92%, respectively, without loss of accuracy.

Table 11 shows how much inference costs can be reduced when Resnet34 and Resnet50 are replaced by distilled Resnet18. Inference speed is the speed of inference on the test dataset of CIFAR100 when the model is deployed on the Tesla P100. It can be observed that: (1) the accuracy of the distilled Resnet18 is 2% and 1.8% higher than Resnet34 and Resnet50 respectively; (2) the inference speed of Resnet18 on Tesla P100 is 1.4x and 2.31x faster than Resnet34 and Resnet50 respectively; (3) the storage occupancy of Resnet18 is 0.52x and 0.47x of Resnet34 and Resnet50 respectively.

4.11 Robustness Analysis

To demonstrate the robustness of KFD, we have conducted two sets of experiments: (1) We add Gaussian noise to CIFAR100 and train Resnet18 with KFD. Then, the noise level is gradually increased from 0.003 to 0.009. As shown in Table 12, the accuracy of Resnet18 distilled with KFD drops by no more than 0.6% when the noise level is increased from 0.003 to 0.009; (2) We have trained Resnet18 with KFD on CIFAR100 with a noise level of 0.003.

Table 13 Robustness analysis (the model is resnet18 distilled with KFD; it is trained on CIFAR100 with a noise level of 0.003 and tested on CIFAR100 with different noise levels)

Noise-level	0.003	0.004	0.005	0.006	0.007	0.008	0.009
Accuracy	79.41	79.22	79.15	79	78.7	78.45	78.05 (−**1.33**)

Bolded numbers are the decreased numbers when the noise level is increased

Then, the noise level of the test dataset of CIFAR100 is gradually increased. As shown in Table 13, the accuracy of the distilled Resnet18 decreases by no more than 1.5% when the noise level is increased from 0.003 to 0.009.

5 Conclusion

In this paper, we have found that dense feature connections conflict with the self-distillation framework and proposed the SDFC module to solve this problem. Based on this, we further propose a distillation method, KFD. We have demonstrated the effectiveness of SDFC through comparative experiments as well as visualization experiments. In addition, we have conducted extensive experiments on three datasets and five models to demonstrate that KFD has comparable performance to the state-of-the-art distillation methods.

Limitations The noticeable limitation is discussed as follows. In KFD, the accuracy is further improved when all classifiers are used for prediction. However, this brings additional storage occupation and computational complexity.

Future work Future research directions are as follows: (1) In KFD, only the final fused feature map (F1234 in Fig. 3) is used for self-distillation, we will next investigate whether the intermediate fused feature maps (F12, F123 in Fig. 3) can further improve the performance of self-distillation; (2) The performance of the network is constantly improving during the training process. Thus, how to dynamically adjust the hyper-parameters is also a future research direction for us.

Acknowledgements This work was supported in part by National Key R&D Program of China (2021YFB2501800); Tianjin Technology Innovation Guide Special (21YDTPJC00130).

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

References

1. Bao L, Ma B, Chang H, et al. (2019) Preserving structural relationships for person re-identification. In: IEEE International conference on multimedia and expo workshops, ICME workshops 2019, Shanghai, China, July 8-12, 2019. IEEE, pp 120–125, https://doi.org/10.1109/ICMEW.2019.00028
2. Bhosale YH, Patnaik KS (2022) Iot deployable lightweight deep learning application for covid-19 detection with lung diseases using raspberrypi. In: 2022 International conference on IoT and blockchain technology (ICIBT), pp 1–6, https://doi.org/10.1109/ICIBT52874.2022.9807725
3. Chen M, Zeng G, Lu K et al (2019) A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN, and LSTM. IEEE Internet Things J 6(4):6997–7010. https://doi.org/10.1109/JIOT.2019.2913176

4. Cho JH, Hariharan B (2019) On the efficacy of knowledge distillation. In: 2019 IEEE/CVF International conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, pp 4793–4801, https://doi.org/10.1109/ICCV.2019.00489
5. Du G, Zhang J, Jiang M et al (2021) Graph-based class-imbalance learning with label enhancement. Trans Neural Netw Learn Syst Early Access. https://doi.org/10.1109/TNNLS.2021.3133262
6. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Gordon GJ, Dunson DB, Dudík M (eds) Proceedings of the fourteenth international conference on artificial intelligence and statistics, AISTATS 2011, Fort Lauderdale, JMLR Proceedings, vol 15. JMLR.org, pp 315–323, URL http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf
7. He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on computer vision and pattern recognition, CVPR 2016, Las Vegas. IEEE Computer Society. https://doi.org/10.1109/CVPR.2016.90
8. Hinton GE, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. CoRR arXiv:1503.02531
9. Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. In: IEEE Conference on computer vision and pattern recognition, CVPR 2021. Computer vision foundation / IEEE, pp 13,713–13,722, URL https://openaccess.thecvf.com/content/CVPR2021/html/Hou_Coordinate_Attention_for_Efficient_Mobile_Network_Design_CVPR_2021_paper.html
10. Howard A, Pang R, Adam H, et al. (2019) Searching for mobilenetv3. In: 2019 IEEE/CVF International conference on computer vision. ICCV 2019, Seoul, Korea. IEEE, pp 1314–1324, https://doi.org/10.1109/ICCV.2019.00140
11. Howard AG, Zhu M, Chen B, et al. (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR arXiv:1704.04861
12. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: 2018 IEEE Conference on computer vision and pattern recognition. CVPR 2018, Salt Lake City. Computer vision foundation / IEEE computer society, pp 7132–7141, https://doi.org/10.1109/CVPR.2018.00745
13. Hu X, Xu X, Xiao Y et al (2019) Sinet: a scale-insensitive convolutional neural network for fast vehicle detection. IEEE Trans Intell Transp Syst 20(3):1010–1019. https://doi.org/10.1109/TITS.2018.2838132
14. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach FR, Blei DM (eds) Proceedings of the 32nd international conference on machine learning. ICML 2015, Lille, JMLR workshop and conference proceedings, vol 37. JMLR.org, pp 448–456, URL http://proceedings.mlr.press/v37/ioffe15.html
15. Ji M, Shin S, Hwang S, et al. (2021) Refine myself by teaching myself: feature refinement via self-knowledge distillation. In: IEEE Conference on computer vision and pattern recognition, CVPR 2021. Computer vision foundation / IEEE, pp 10,664–10,673, https://doi.org/10.1109/CVPR46437.2021.01052, URL https://openaccess.thecvf.com/content/CVPR2021/html/Ji_Refine_Myself_by_Teaching_Myself_Feature_Refinement_via_Self-Knowledge_Distillation_CVPR_2021_paper.html
16. Krizhevsky A, Hinton G, et al. (2009) Learning multiple layers of features from tiny images
17. Li X, Wang W, Hu X, et al. (2019) Selective kernel networks. In: IEEE Conference on computer vision and pattern recognition, CVPR 2019, Long Beach. Computer vision foundation/IEEE, pp 510–519, https://doi.org/10.1109/CVPR.2019.00060
18. Lin T, Dollár P, Girshick RB, et al. (2016) Feature pyramid networks for object detection. CoRR arXiv:1612.03144
19. Lu K, Zeng G, Luo X et al (2021) Evolutionary deep belief network for cyber-attack detection in industrial automation and control system. IEEE Trans Ind Inform 17(11):7618–7627. https://doi.org/10.1109/TII.2021.3053304
20. Mao L, Li X, Yang D et al (2021) Convolutional feature frequency adaptive fusion object detection network. Neural Process Lett 53(5):3545–3560. https://doi.org/10.1007/s11063-021-10560-4
21. Mirzadeh S, Farajtabar M, Li A, et al. (2020) Improved knowledge distillation via teacher assistant. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020. The thirty-second innovative applications of artificial intelligence conference, IAAI 2020. The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York. AAAI Press, pp 5191–5198, URL https://aaai.org/ojs/index.php/AAAI/article/view/5963
22. Park J, Woo S, Lee J, et al. (2018) BAM: bottleneck attention module. In: British Machine Vision Conference 2018, BMVC 2018, Newcastle. BMVA Press, p 147, URL http://bmvc2018.org/contents/papers/0092.pdf
23. Romero A, Ballas N, Kahou SE, et al. (2015) Fitnets: Hints for thin deep nets. In: Bengio Y, LeCun Y (eds) 3rd International conference on learning representations, ICLR 2015, San Diego. Conference track proceedings, arxiv:1412.6550

24. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Bengio Y, LeCun Y (eds) 3rd International conference on learning representations, ICLR 2015, San Diego. Conference track proceedings. arxiv:1409.1556
25. Tan C, Liu J, Zhang X (2021) Improving knowledge distillation via an expressive teacher. Knowl Based Syst 218(106):837. https://doi.org/10.1016/j.knosys.2021.106837
26. Tan M, Pang R, Le QV (2020) Efficientdet: Scalable and efficient object detection. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition, CVPR 2020, Seattle. Computer vision foundation / IEEE, pp 10,778–10,787, https://doi.org/10.1109/CVPR42600.2020.01079, URL https://openaccess.thecvf.com/content_CVPR_2020/html/Tan_EfficientDet_Scalable_and_Efficient_Object_Detection_CVPR_2020_paper.html
27. Wang F, Jiang M, Qian C, et al. (2017) Residual attention network for image classification. In: 2017 IEEE Conference on computer vision and pattern recognition, CVPR 2017, Honolulu. IEEE Computer society, pp 6450–6458, https://doi.org/10.1109/CVPR.2017.683
28. Woo S, Park J, Lee J, et al. (2018) CBAM: convolutional block attention module. In: Ferrari V, Hebert M, Sminchisescu C, et al (eds) Computer vision - ECCV 2018 - 15th European conference, Munich. Proceedings, part VII, Lecture notes in computer science, vol 11211. Springer, pp 3–19, https://doi.org/10.1007/978-3-030-01234-2_1
29. Xie S, Girshick RB, Dollár P, et al. (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on computer vision and pattern recognition, CVPR 2017, Honolulu. IEEE Computer Society, pp 5987–5995, https://doi.org/10.1109/CVPR.2017.634
30. Yan Z, Zheng H, Li Y et al (2021) Detection-oriented backbone trained from near scratch and local feature refinement for small object detection. Neural Process Lett 53(3):1921–1943. https://doi.org/10.1007/s11063-021-10493-y
31. Yang J, Martínez B, Bulat A, et al. (2020) Knowledge distillation via adaptive instance normalization. CoRR arXiv:2003.04289
32. Yim J, Joo D, Bae J, et al. (2017) A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In: 2017 IEEE Conference on computer vision and pattern recognition, CVPR 2017, Honolulu. IEEE Computer Society, pp 7130–7138, https://doi.org/10.1109/CVPR.2017.754
33. Zagoruyko S, Komodakis N (2016) Wide residual networks. In: Wilson RC, Hancock ER, Smith WAP (eds) Proceedings of the British machine vision conference 2016, BMVC 2016, York. BMVA Press. URL http://www.bmva.org/bmvc/2016/papers/paper087/index.html
34. Zagoruyko S, Komodakis N (2017) Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: 5th International conference on learning representations, ICLR 2017, Toulon. Conference track proceedings. OpenReview.net, URL https://openreview.net/forum?id=Sks9_ajex
35. Zhang L, Song J, Gao A, et al. (2019a) Be your own teacher: improve the performance of convolutional neural networks via self distillation. In: 2019 IEEE/CVF International conference on computer vision, ICCV 2019, Seoul. IEEE, pp 3712–3721, https://doi.org/10.1109/ICCV.2019.00381
36. Zhang L, Tan Z, Song J, et al. (2019b) SCAN: a scalable neural networks framework towards compact and efficient models. In: Wallach HM, Larochelle H, Beygelzimer A, et al (eds) Advances in neural information processing systems 32: annual conference on neural information processing systems 2019. NeurIPS 2019, Vancouver, pp 4029–4038. URL https://proceedings.neurips.cc/paper/2019/hash/934b535800b1cba8f96a5d72f72f1611-Abstract.html
37. Zhang L, Bao C, Ma K (2021) Self-distillation: towards efficient and compact neural networks. Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2021.3067100
38. Zhang R, Jiang X, An J et al (2022) Data-free low-bit quantization for remote sensing object detection. IEEE Geosci Remote Sens Lett 19:1–5. https://doi.org/10.1109/LGRS.2021.3122875
39. Zhao B, Cui Q, Song R, et al. (2022) Decoupled knowledge distillation. CoRR arXiv:2203.08679
40. Zhao H, Sun X, Dong J et al (2021) Knowledge distillation via instance-level sequence learning. Knowl Based Syst 233(107):519. https://doi.org/10.1016/j.knosys.2021.107519
41. Zheng YJ, Chen SB, Ding CH et al (2022) Model compression based on differentiable network channel pruning. Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3165123