Cross-Layer Fusion for Feature Distillation

Honglin Zhu[1], Ning Jiang[1,✉], Jialiang Tang[1], Xinlei Huang[1], Haifeng Qing[1], Wenqing Wu[2], and Peng Zhang[2]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, Sichuan, 621000, China
✉Corresponding author:jiangning@swust.edu.cn
[2] School of Mathematics and Physics, Southwest University of Science and Technology, Mianyang, Sichuan, 621000, China

Abstract. Knowledge distillation is a model compression technology, which can effectively improve the performance of a small student network by learning knowledge from a large pre-trained teacher network. In most previous works of feature distillation, the performance of the student is still lower than the teacher network due to it only being supervised by the teacher's features and labels. In this paper, we novelly propose **C**ross-layer **F**usion for **K**nowledge **D**istillation named CFKD. Specifically, instead of only using the features of the teacher network, we aggregate the features of the teacher network and student network together by a dynamic feature fusion strategy (DFFS) and a fusion module. The fused features are informative, which not only contain expressive knowledge of teacher network but also have the useful knowledge learned by previous student network. Therefore, the student network learning from the fused features can achieve comparable performance with the teacher network. Our experiments demonstrate that the performance of the student network can be trained by our method, which can be closer to the teacher network or even better.

Keywords: Model Compression · Knowledge Distillation · Cross-Layer Fusion.

1 Introduction

With the development of deep learning, neural networks have obtained satisfactory performance in various computer vision tasks [4, 10, 27]. However, the required parameters and calculations for a superior network are huge on the whole. In practice, these cumbersome networks with high performance are difficult to deploy on resource-limited devices, such as mobile phones, which hinders the practice application of neural networks. To solve this problem, a variety of research methods have been proposed, including network pruning [16, 17], quantization [12], lightweight network design [9, 18, 22] and knowledge distillation [6, 21, 26, 29].

Among these methods, we are concerned with knowledge distillation (KD), which is mainly to transfer knowledge from the large teacher network to the

compact student network. KD aims to train a small student network with similar performance to the large teacher network, thus the student network can be applied to resource-limited devices for practice application. The original distillation approach is introduced by Hinton et al. [6], which makes a student learn the logit soft output after the linear layer from a teacher network. To improve the efficiency of knowledge transfer, Romero et al. [21] proposed a feature distillation framework to utilize the teacher's intermediate features and ground truth labels to supervise the student training. Some works [3,20,25,26] have further improved the feature distillation method. In general, features from a certain layer of the teacher are transferred to the student. After filtering, a part of useful information extracted by a certain layer of the teacher may be discarded. Therefore, the improvement brought by transferring the knowledge of the teacher alone is limited. Some studies [7, 14, 15] have found that the aggregated feature maps from two parallel networks are expected to generate better prediction results than those of the feature maps of a single network. It is believed that the prediction results can be improved because the fused features are rich and discriminative [7].

In this study, we propose cross-layer fusion for knowledge distillation (CFKD). In our method, instead of only using teacher features, the student is supervised by expressive knowledge combined with the features of the teacher network and student network, as shown in Fig. 1. We introduce the fused features for two reasons. The first is that the thin and small student network could provide shallow texture information for fused features. The second is that the student could effectively improve their performance through self-distillation of their deeper stage features [8, 13, 28]. We believe that the fused features that aggregate the deeper stage features of the student are beneficial to the student network training. Furthermore, we explore a dynamic feature fusion strategy, which can get valuable fused features by controlling the role of the student's features in the fused features. In some teacher/student networks such as 'ResNet50/ResNet20' and 'WRN-40-2/WRN-16-2', the student network outperforms the teacher network in our experiments, which validates the effectiveness of our method.

The details of our proposed distillation method are presented in Sec. 3.3. The specific content of the fusion module and the dynamic feature fusion strategy are shown in Sec. 3.4. Extensive experiments validate the effectiveness of our method in Sec. 4.

2 Related work

In this section, we introduce the related work in detail. Related works on knowledge distillation and feature distillation are discussed in Sec. 2.1 and Sec. 2.2, respectively. Related works on the feature fusion method are discussed in Sec. 2.3.

2.1 Knowledge Distillation

Reducing model parameters and speeding up network inference are the main purposes of model compression. Knowledge distillation is a simple and convenient

approach among model compression methods, which improves the performance of the student network by learning the output of the well-trained teacher network. This idea is first introduced by Hinton et al. [6], the student network is not only supervised by ground-truth labels but also mimics the teacher's predicted probabilities called soft targets. In [2, 29], they explore an online distillation method to improve network performance by training multiple student networks and encouraging each network to learn soft targets from other networks.

2.2 Feature-Map Distillation

The goal of feature distillation is to promote the student to learn the teacher's features. Romero et al. [21] first introduced intermediate layer feature distillation. They proposed that the teacher network transfers the intermediate layer features as knowledge to the student network, which can further improve the prediction ability of the student network. The attention mechanism is introduced by Zagoruyko et al. [26] to extract expressive information from the teacher's middle layers for knowledge distillation. Heo et al. [5] proposed a margin ReLU function, which advances the position of feature distillation before ReLU. Furthermore, they used a partial L_2 loss to reduce the transfer of useless information.

To the best of our knowledge, previous work has not considered the role of the student's deeper features in the teacher-student learning paradigm. In this study, we propose the combination of the student's deeper features and teacher features as knowledge for student learning.

2.3 Feature fusion method

The feature fusion method can combine different features through fusion operations. Lin et al. [15] employed matrix product to aggregate two feature maps produced by two parallel convolutional networks for image classification. They think that fused features can get higher local features. The method of Hou et al. [7] fused features by introducing the 'SUM' operation. They argue that richer and more accurate images produced by feature fusion methods are beneficial for recognition. Kim et al. [14] apply fused features for online distillation, to boost each untrained sub-network. Shen et al. [23] aggregated features from multiple teachers to guide the learning of the student network by amalgamation module.

3 Method

This section introduces the cross-layer fusion knowledge distillation (CFKD). The notations are in Sec. 3.1. Section 3.2 briefly introduces logit-based distillation. Figure 1 shows an overview of our distillation method. The details of the proposed method are described in Sec. 3.3. Section 3.4 discusses the fusion method and dynamic feature fusion strategy in detail.

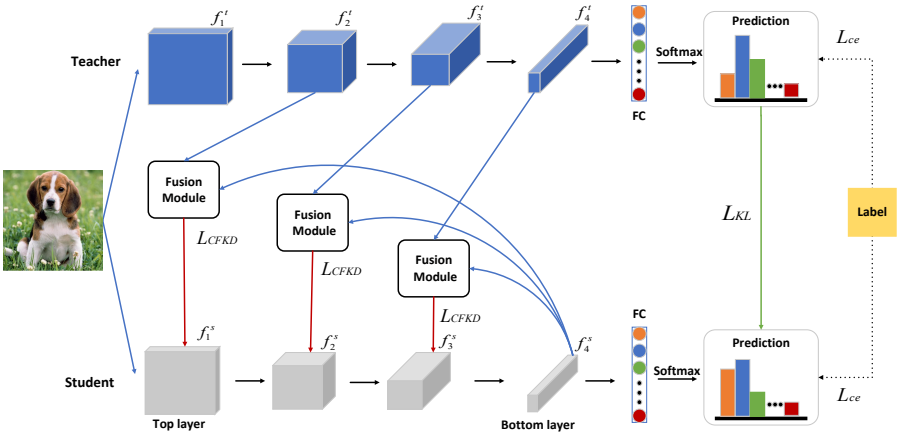


Fig. 1: The overall framework of cross-layer fusion for feature distillation. The process of distillation consists of two stages. In the first stage, the fusion module aggregates the features from the teacher and the student to generate fused features. In the second stage, rich fused features are learned by the student to improve performance.

3.1 Notations

Given a set of input data $\mathbf{X} = \{x_1, x_2.., x_k\}$ including $\mathbf{k}$ examples, the label corresponding to each example is denoted as $\mathbf{Y} = \{y_1, y_2.., y_k\}$. We define a pre-trained and fixed teacher network as $\mathcal{N}_T$ and a student network as $\mathcal{N}_S$. Let $\mathbf{g}^s$, $\mathbf{g}^t$ denote the logits of the student network and teacher network, respectively, where the variables with superscripts s and t represent them are outputted by the student network and teacher network, respectively, throughout this paper. $\mathbf{f}_j^s$ represents the j-th layer's features of the student network, $j \in \{1, ..., n\}$. $\mathbf{f}_i^t$ represents the i-th layer's features of the teacher network, $i \in \{1, ..., n\}$, where n represents the maximum number of blocks in the network.

3.2 Logit-based Knowledge Distillation

The purpose of knowledge distillation is to train a student network by imitating the output of a teacher network. In original knowledge distillation [6], the student network is expected to imitate the teacher's soft targets. For image classification tasks, the traditional distillation loss uses the cross-entropy loss and Kullback-Leibler (KL) divergence. The loss function L_{logit} can be expressed as:

$$L_{logit} = (1-\lambda)L_{ce}(\mathbf{y}, \sigma(\mathbf{g}^s)) + \lambda\tau^2 L_{KL}(\sigma(\mathbf{g}^t/\tau), \sigma(\mathbf{g}^s/\tau)), \tag{1}$$

where $L_{ce}(\cdot,\cdot)$ denotes the cross-entropy loss, $L_{KL}(\cdot,\cdot)$ denotes KL divergence. $\tau > 0$ is a temperature parameter that controls the level of smoothness between categories. We use τ^2 times L_{KL} because we need to scale the gradient of the soft targets by $1/\tau^2$. λ is a balancing hyperparameter.

3.3 Teacher-student Feature Fusion

In our method, the key point is to enhance the performance of the student model by learning fused features of the teacher-student model. As shown in Fig. 1, the overall distillation process is divided into two stages. In the first stage, the same data is inputted into a student network and teacher network to obtain features of each layer of the teacher network and features of the bottom layer of the student network. The features from the bottom layer of the student network and the features from the different layers of the teacher network are aggregated into fused features through a fusion module, respectively. In the second stage, the fused features are seen as rich knowledge, which is transferred to different stages of the student network.

First, we obtain the features $\widetilde{\mathbf{f}}_i$ aggregated from the features $\mathbf{f}_i^t$ of the i-th layer of the teacher network and the features $\mathbf{f}_n^s$ of the student network. The fused features $\widetilde{\mathbf{f}}_i$ of a single layer can be defined as:

$$\widetilde{\mathbf{f}}_i = \mathcal{F}(W \cdot \mathbf{f}_i^t + b, W \cdot \mathbf{f}_n^s + b) \tag{2}$$

where $\mathcal{F}(\cdot,\cdot)$ function is the fusion operation described in detail in Sec. 3.4. $W(\cdot)$ is a linear function for matrix transformation, b is the bias matrix.

After getting the fused features, we can get the distillation loss of CFKD through mean squared error (MSE). The single-layer distillation loss $L_{CFKD}^{'}$ can be written as:

$$L_{CFKD}^{'} = MSE\left(\widetilde{\mathbf{f}}_i, \mathcal{R}(\mathbf{f}_{n-1}^s)\right), \tag{3}$$

where $\mathcal{R}(\cdot)$ is a regression function matching the dimensions of the fused features which is accomplished by [21].

In multi-layer knowledge distillation, the knowledge of the teacher is transferred to multiple layers of the student network. It is worth noting that multi-layer knowledge distillation can get higher efficiency for transferring knowledge. We further generalize our method to multi-layer knowledge distillation. More specifically, each layer of the student network learns the corresponding fused features. The multi-layer distillation loss L_{CFKD} is calculated by the following:

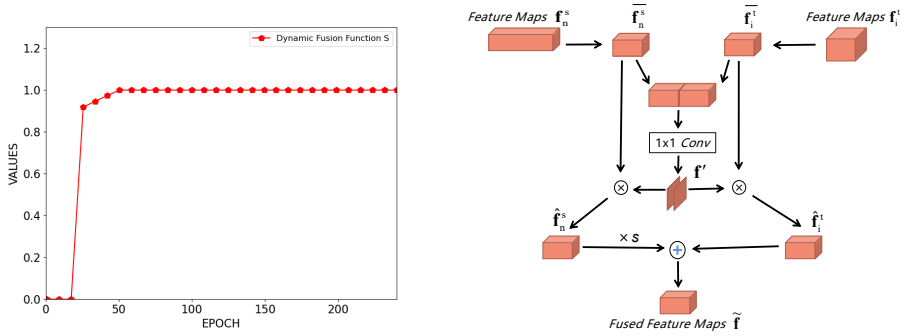
$$L_{CFKD} = \sum_{j=1,i=j+1}^{n-1} MSE\left(\widetilde{\mathbf{f}}_i, \mathcal{R}(\mathbf{f}_j^s)\right). \tag{4}$$

In our method, the original cross-entropy loss and KL divergence are added to the total loss function. We introduce hyper-parameters α, β, and γ to tune the relationship of several loss functions. Here is our overall loss function L_{total}:

$$L_{total} = \alpha L_{ce} + \beta\tau^2 L_{KL} + \gamma L_{CFKD}. \tag{5}$$

3.4 Fusion Module and Dynamic Feature Fusion Strategy

The details of the fusion module are shown in Fig. 2b. First, two feature maps $\mathbf{f}_n^s$ and $\mathbf{f}_i^t$ should be input into the fusion module. The channel dimensions and sizes



(a) Dynamic Fusion Function S (b) Fusion Module

Fig. 2: An overview of the proposed dynamic fusion function and fusion module. (a)The dynamic fusion function is a piecewise function. When the training time increases, the dynamic fusion function obtains different function values. (b)Both features are compressed. The compressed features are aggregated into fused features through 1x1 convolution, multiplication, and addition.

of the feature maps $\mathbf{f}_n^s$ and $\mathbf{f}_i^t$ are compressed to a relatively small size. Second, The two compressed feature maps $\overline{f_n^s}$ and $\overline{f_i^t}$ are aggregated into the feature maps $\mathbf{f}^{'} \in R^{b\times 2\times h\times w}$. Then, the concatenated features $\mathbf{f}^{'}$ are multiplied by the compressed features to produce two feature maps $\hat{f}_n^s$ and $\hat{f}_i^t$. Finally, the fused features $\widetilde{\mathbf{f}}_i$ are obtained by adding the two features $\hat{f}_n^s$ and $\hat{f}_i^t$. The structure is inspired by Chen et al. [3]. Different from them, we add the operation of compressing the feature maps in the fusion module.

The purpose of the dynamic feature fusion strategy (DFFS) is to obtain high-quality fused features by controlling the role of student features in fused features. In the early stage of training, the student has poor performance, whose bottom layer features may bring negative effects to the fused features. Therefore, only the transferred knowledge from the features of the teacher is used in the beginning. With the improvement of the student's performance, features from the student are more beneficial to the generated fused features. The features of the two models are actively combined into fused features. At this point, we introduce a dynamic fusion function S to control the effect of student features on fused features. The dynamic fusion function is shown in Fig. 2a. In the fusion process, the features of the student are multiplied by S and combined with the features of the teacher. When the student's performance is similar to the teacher's performance, the features of the two models are fused in the same proportion. The equation of dynamic fused function S can be written as:

$$S(EPOCH) = \begin{cases} \dfrac{EPOCH}{300} + 0.83334 & 20 =< EPOCH <= 50 \\ 1 & 50 < EPOCH <= 240 \end{cases} \tag{6}$$

4 Experiments

The training details for all experiments are in Sec. 4.1. The effectiveness of the proposed distillation method has been verified on different datasets. In Sec. 4.2, we provide the results on CIFAR-10 dataset. In Sec. 4.3, we show experimental results of teacher-student networks with the same structures and different structures for distillation on CIFAR-100. To further verify the effectiveness of the method, we provide the results of the comparison of different distillation methods. Finally, the results of ablation experiments are presented in Sec. 4.4.

4.1 Experimental setup

Dataset (1) CIFAR-10 contains 60,000 examples, which have 50,000 images for training and 10,000 images for testing. It includes 10 categories and all examples are 32x32 RGB images. (2) CIFAR-100 has 100 classes for image classification with 50K training examples and 10K test examples, where the resolution of each example is 32x32.

Implementation Details All experiments use the gradient descent algorithm [1]. Horizontal flip and random crop are used for data augmentation in the experiments. The batch size is set to 128 and the weight decay value is $5e^{-4}$ on CIFAR-100 and CIFAR-10 datasets. The temperature hyperparameter is set to 4. For CIFAR-100, all networks are trained for 240 epochs. The initial learning rate for the experiments we set is 0.05. We set the learning rate to drop at 150, 180, and 210 epochs. Our training setup is consistent with CRD [24]. All our results are obtained from training four times. For CIFAR-10, all networks are trained for 200 epochs. During training, the initial learning rate is 0.1 and is decayed at 60, 120, and 160 epochs. The overall experimental setup follows AT [26].

Table 1: Top-1 test accuracy of the same architectures of the teacher-student networks on CIFAR-10. Ours is CFKD.

Teacher Student	WRN40-2 WRN16-2	WRN40-1 WRN16-1
Teacher	94.73%	93.42%
Student(base)	93.69%	91.14%
KD [6]	93.92%	91.30%
Ours	**94.23%**	**91.56%**

4.2 Experiments of CIFAR-10 dataset

For CIFAR-10, the WRN model [27] was used as teacher-student combinations for evaluation. Table 1 shows the experimental results of our CFKD on CIFAR-

10. The student network using our method achieves 94.23% and 91.56% results, respectively. Compared to the KD [6], our method achieves better accuracy.

4.3 Experiments of CIFAR-100 dataset

Table 2 shows the experimental results of the same teacher-student architecture on CIFAR-100. We set up multiple groups of networks with the same architectures, for example, WRN-40-2/WRN-16-2, ResNet56/ResNet20. From Table 2, our proposed fusion feature distillation outperforms other distillation methods. Especially, the best example is in the setting of 'WRN-40-2/WRN-16-2' where the student's performance is even better than the teacher's performance.

As shown in Table 3, we set up multiple groups of teacher-student networks with different architectures, such as ResNet32x4/ShuffleNetV2 [18]. This experiment shows that our method achieves great performance in different teacher-student architectures. The results in Table 2 and Table 3 show that our proposed cross-layer fusion feature distillation successfully improves student network performance.

Table 2: Test accuracy of the same architectures of teacher-student networks on CIFAR100 (ours is CFKD). The accuracy of the comparison method comes from the papers of other authors. In this experiment, CFKD outperforms all other methods.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet56 ResNet20	ResNet110 ResNet20	ResNet110 ResNet32
Teacher	75.61%	75.61%	72.34%	74.31%	74.31%
Student(base)	73.26%	71.98%	69.06%	69.06%	71.14%
KD [6]	74.92%	73.54%	70.66%	70.67%	73.08%
FitNet [21]	73.58%	72.24%	69.21%	68.99%	71.06%
AT [26]	74.08%	72.77%	70.55%	70.22%	72.31%
PKT [19]	74.54%	73.45%	70.34%	70.25%	72.61%
FSP [25]	72.91%	N/A	69.95%	70.11%	71.89%
NST [11]	73.68%	72.24%	69.60%	69.53%	71.96%
CRD [24]	75.48%	74.14%	71.16%	71.46%	73.48%
Ours	**75.70%**	**74.75%**	**72.39%**	**71.60%**	**74.11%**

4.4 Ablation Study

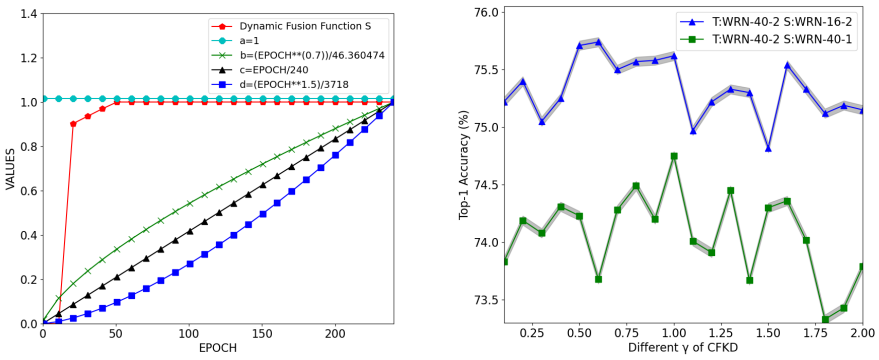
In this section, we verify the effectiveness of the method through different ablation experiments. We mainly use the WRN network as the base model for validation experiments. We mainly conduct experiments by setting different loss hyperparameter weights, the number of fused features, and the dynamic fusion function.

Table 3: Test accuracy of the different architectures of teacher-student networks on CIFAR100 (ours is CFKD). The accuracy of the comparison method comes from the papers of other authors.

Teacher Student	ResNet32x4 ShuffleNetV1	ResNet32x4 ShuffleNetV2
Teacher	79.42%	79.42%
Student(base)	70.50%	71.82%
KD [6]	74.07%	74.45%
AT [26]	71.73%	72.73%
FitNet [21]	74.12%	73.54%
NST [11]	74.12%	74.68%
PKT [19]	74.10%	74.69%
CRD [24]	75.11%	75.56%
Ours	**75.61%**	**75.86%**

Table 4: The effect of the different number of features on distillation efficiency.

Teacher Student	WRN40-2 WRN16-2	WRN40-2 WRN40-1
Teacher	79.42%	79.42%
Student(base)	70.50%	71.82%
num=1	75.58%	73.95%
num=2	**75.71%**	**74.75%**
num=3	75.33%	74.52%



(a) Different dynamic fusion functions. (b) Effects of varying γ.

Fig. 3: Effects of different dynamic fusion functions and hyperparameter γ on distillation performance. (a) Multiple dynamic fusion functions are set to verify the effectiveness of the method. (b) We verify the effect of varying γ on model training.

Table 5: Ablation with different dynamic fusion functions on CIFAR100.

Teacher Student	WRN40-2 WRN16-2	WRN40-2 WRN40-1
Constant Function a	75.29%	73.80%
Convex Function b	75.47%	74.26%
Linear Function c	75.19%	73.91%
Concave Function d	75.54%	73.64%
Dynamic Fusion Function S	**75.71%**	**74.75%**

The number of fused features Learnable structural knowledge has a positive effect on the student's network training. To some extent, the number of fused features can directly affect the structure of knowledge. Therefore, it is necessary to explore the number of fused features for training. We set up several experiments. In the first group, we only use the teacher's features to transfer knowledge. Because the fused features only have one type, we mark this group as 'num=1'. Fused features of the second group are composed of the last layer of the student's features and a certain layer of the teacher's features. We label this group as 'num = 2'. In the last group, the multi-layer features from the teacher are aggregated with the student features to obtain the aggregated features. 'num = 3' represents the last group. Table 4 shows the validation results under different experimental settings. The accuracy of the second group significantly outperformed the other groups.

Dynamic fusion function The purpose of the dynamic fusion function is mainly to control the role of features of the student in the fused features. To verify the effectiveness of the proposed method, we set different dynamic fusion functions including constant function a with constant one, increasing convex function b, increasing linear function c, increasing concave function d and proposed dynamic fusion function S. As shown in Fig. 3a, the label in the figure shows the expression of the functions. Table 5 provides experimental results for different functions. From Table 5, we can see that the dynamic fusion function S outperforms other functions in this experiment.

Weight of loss hyperparameter The hyperparameters of the loss function can affect the performance of the network because the hyperparameters affect the gradient propagation. Figure 3b shows our experimental results using different hyperparameter values. The range of the hyperparameter γ is 0.1 to 2.0. The interval between two adjacent values is 0.1. From Fig. 3b, we can see that parameter values between 0.5 and 1.0 are more favorable for network training.

5 Conclusion

In this study, we propose cross-layer feature fusion for knowledge distillation. The purpose of our method is to improve the performance of the compact student

network by learning fused features aggregated from the features of the teacher and the deeper stage (bottom layer) features of the student. The method consists of two stages. Firstly, the fusion module aggregates two different features to obtain fused features. Then, the fused features are viewed as knowledge and then transferred to the student network. To obtain high-quality fused features, a dynamic fusion function is proposed. Extensive experiments demonstrate that our method can improve the student's performance. Compared to previous methods, it is shown that the methods we propose can achieve better accuracy.

Acknowledgement. This research is supported by Sichuan Science and Technology Program (No. 2022YFG0324), SWUST Doctoral Research Foundation under Grant 19zx7102.

References

1. Bottou, L.: Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade, pp. 421–436. Springer (2012)
2. Chen, D., Mei, J.P., Wang, C., Feng, Y., Chen, C.: Online knowledge distillation with diverse peers. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3430–3437 (2020)
3. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling knowledge via knowledge review. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5008–5017 (2021)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1921–1930 (2019)
6. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)
7. Hou, S., Liu, X., Wang, Z.: Dualnet: Learn complementary features for image recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 502–510 (2017)
8. Hou, Y., Ma, Z., Liu, C., Loy, C.C.: Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1013–1021 (2019)
9. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019)
10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
11. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2704–2713 (2018)

13. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10664–10673 (2021)
14. Kim, J., Hyun, M., Chung, I., Kwak, N.: Feature fusion for online mutual knowledge distillation. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 4619–4625. IEEE (2021)
15. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1449–1457 (2015)
16. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270 (2018)
17. Luo, J.H., Wu, J., Lin, W.: Thinet: A filter level pruning method for deep neural network compression. In: Proceedings of the IEEE international conference on computer vision. pp. 5058–5066 (2017)
18. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
19. Passalis, N., Tefas, A.: Probabilistic knowledge transfer for deep representation learning. CoRR, abs/1803.10837 **1**(2), 5 (2018)
20. Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5007–5016 (2019)
21. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
23. Shen, C., Wang, X., Song, J., Sun, L., Song, M.: Amalgamating knowledge towards comprehensive classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3068–3075 (2019)
24. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. arXiv preprint arXiv:1910.10699 (2019)
25. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017)
26. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
27. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
28. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3713–3722 (2019)
29. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4320–4328 (2018)