

Knowledge Fusion Mutual Self-distillation: Each Branch is a Good Teacher

Yue Jia¹, Chuanqi Ma^{2*}, Shuiping Ni¹, and Mingfu Zhu¹

¹ School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China
jiayue@home.hpu.edu.cn, {nishuiping, mfzhu}@hpu.edu.cn

² Research and Development Department, Chuitian (Xinjiang) Energy Technology Company Limited,
Shihezi 832000, China
17399933008@163.com

Received 19 March 2025; Revised 24 July 2025; Accepted 4 August 2025

Abstract. Deep learning has achieved significant success in image classification. However, training deep learning models solely based on datasets often leads to limited improvement in their performance. Self-distillation, as a novel technique in knowledge distillation, utilizes the model's structure to construct multiple branches and improves performance through distillation. However, existing self-distillation methods often utilize deep-level features to guide shallow-level features, neglecting the value of shallow-level features. Moreover, most methods rely on branches with deeper layers as the source of knowledge, which limits the utilization of knowledge from branches with shallower layers. Based on this, we propose a knowledge fusion mutual self-distillation (KFMSD) framework. KFMSD constructs multiple branches at different depths based on the structure of the original network, encouraging mutual learning between the original network and each branch. The framework includes a knowledge fusion module (KFM) that can fuse features from deep and shallow layers, enabling effective knowledge transfer between them. We validate the proposed method on three datasets, CIFAR-10, CIFAR-100, and Tiny-ImageNet, and compare it with other knowledge distillation methods. Experimental results demonstrate that KFMSD can further enhance model performance.

Keywords: deep learning, knowledge distillation, self-distillation, knowledge fusion

1 Introduction

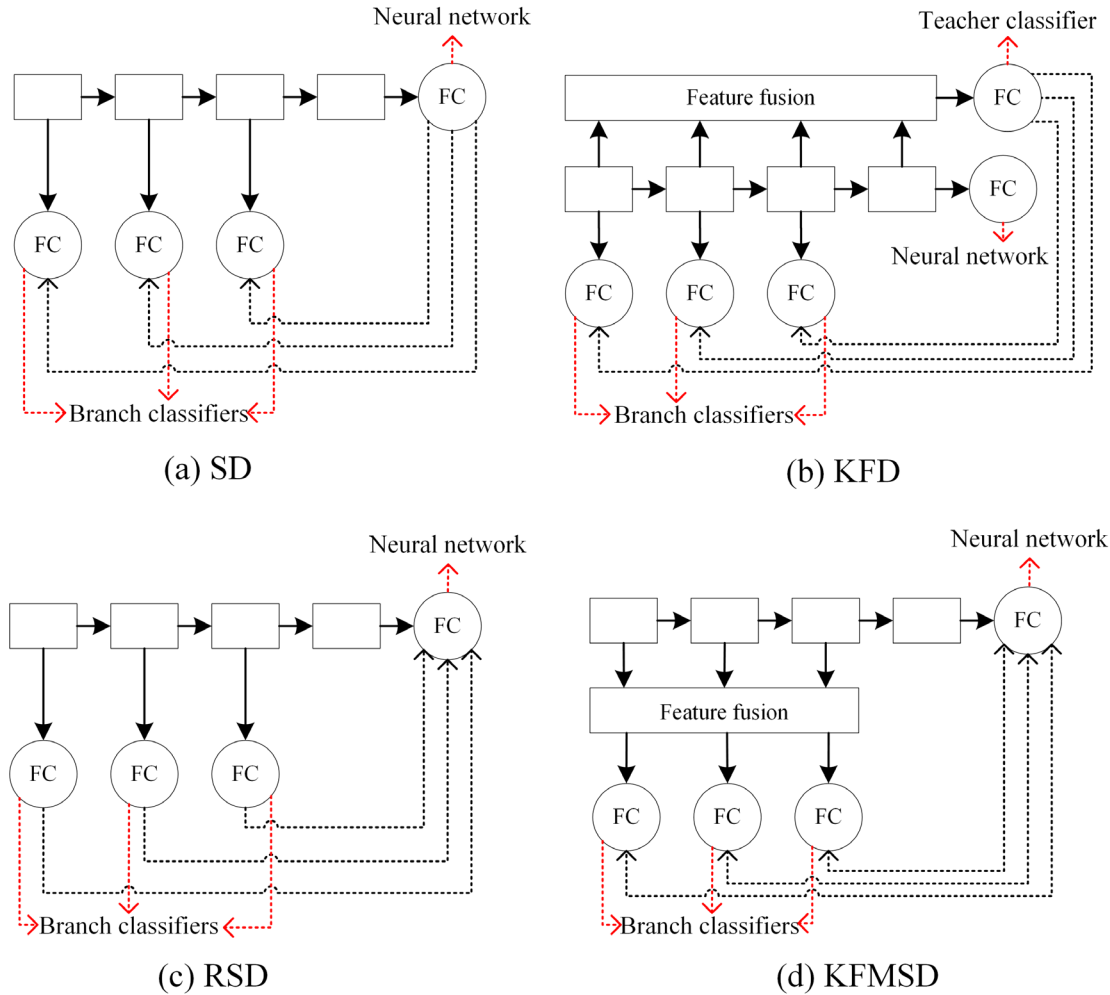
Deep neural networks [1, 2] have gained significant attention and rapid development in recent years, especially in image classification [3, 4], medical image processing [5, 6], object detection [7, 8], image generation [9, 10], and image segmentation [11, 12]. However, deploying better-performing neural networks on low-resource devices can be challenging due to their large number of parameters. It is essential to maintain model performance while also achieving a lightweight design. Based on this, various model compression methods [13] have been proposed, including knowledge distillation, lightweight neural network design, and quantization. Knowledge distillation [14] aims to transfer the knowledge from a high-capacity, well-performing teacher model to a low-capacity, less-performing student model, thereby enhancing the student model's performance without altering its parameter count.

However, traditional knowledge distillation relies on a pretrained teacher model, which increases the training cost. In contrast, self-distillation (SD) [15] eliminates this dependency, thereby attracting considerable attention. SD treats the original network as the teacher and constructs branch structures for distillation within the same model. To further improve SD, Li et al. [16] proposed knowledge fusion distillation (KFD), which optimizes self-distillation by fusing the network's features and constructing a new teacher model to guide all branches. However, both SD and KFD overlook the significance of all branches, each of which can provide valuable knowledge. Moreover, compared to SD, KFD introduces an additional classifier as the teacher model, increasing the training cost. Ni et al. [17] proposed reverse self-distillation (RSD) to utilize knowledge from branches. Each branch acts as a teacher for the original network. However, RSD neglects the knowledge from the original network and the features of branches can be further optimized.

Based on this, we propose a knowledge fusion module (KFM). The KFM can integrate the feature maps from the original network and the branches, providing high-quality target features for learning. It is designed based

* Corresponding Author

on attention mechanisms, using mathematical operations to fuse features without generating additional storage. Based on KFM, Knowledge Fusion Mutual Self-distillation (KFMSD) is proposed to enable mutual learning between the original network and each branch, allowing both the network and the branches to become good teachers. The KFMSD framework fully leverages the knowledge of each branch and enables efficient knowledge transfer within the framework without introducing additional teacher models. The four distillation approaches are compared in Fig. 1.



(a) The original network is the sole knowledge source (b) The teacher classifier constructed through feature fusion is the sole knowledge source (c) All branches serve as knowledge sources (d) The original network and the branches serve as knowledge sources for each other

Fig. 1. Comparison of the four distillation approaches (The red dashed line indicates the classifier type, the black solid line represents the forward path, and the black dashed line denotes the distillation path.)

The contributions of this paper are as follows:

1. We design the KFM that effectively fuses feature maps from the original network and branches, providing higher-quality learning targets for the original network.
2. We introduce a new self-distillation approach called KFMSD based on the proposed KFM. KFMSD enhances knowledge diversity within the framework by utilizing both original network and branches.
3. We conduct numerous experiments on various datasets to demonstrate the effectiveness of KFMSD. Furthermore, KFMSD has excellent abilities in compressing models effectively.

In the remainder of this paper, we first review related works and highlights the key research problem. We then present the proposed KFMSD model architecture, including the KFM module, the design logic of the distillation framework, and the construction principles of the loss function. Next, we describe the experimental design and empirical results to demonstrate the effectiveness and advantages of the proposed method. Finally, we summarize the conclusions of this study and discuss potential directions for future research.

2 Related Works

2.1 Attention Mechanisms

Attention mechanism [18] is a fundamental component of deep learning, significantly enhancing the performance and computational efficiency of deep neural networks by selectively focusing on critical information while suppressing irrelevant details. By dynamically emphasizing important features, attention mechanisms facilitate superior feature representation learning, thereby improving model generalization and robustness.

Squeeze-and-Excitation Networks [19] (SENet), one of the most influential early attention mechanisms, introduced the concept of channel attention. Its “squeeze-and-excitation” operation adaptively recalibrates the importance of feature channels, enhancing the representational capacity of convolutional neural networks without incurring substantial computational overhead. Subsequently, Efficient Channel Attention Network [20] (ECA-Net) further optimized channel attention by employing a lightweight strategy to capture cross-channel dependencies, maintaining high accuracy while ensuring computational efficiency. Frequency Channel Attention Network [21] (FCA-Net) enhances feature representation by incorporating frequency components, leveraging spectral information to improve feature extraction capabilities, particularly in tasks requiring fine-grained feature discrimination. In the domain of spatial attention, Convolutional Block Attention Module [22] (CBAM) and Bottleneck Attention Module [23] (BAM) integrate channel and spatial attention through sequential and parallel approaches, respectively, providing a comprehensive feature enhancement mechanism. Similarly, Spatial Group-wise Enhance [24] (SGE) optimizes the spatial distribution of sub-features within groups, strengthening the network’s ability to process structured information. SimAM [25] introduces an innovative approach by directly estimating 3D attention weights for each feature, circumventing the need for handcrafted pooling operations or complex transformations. This method enables precise assessment of feature importance, effectively capturing intricate spatial relationships and dependencies, thereby improving the efficacy of feature representation learning. These advancements in attention mechanisms have significantly contributed to the performance enhancement of deep learning models across various tasks. In this paper, we employ the attention mechanism to mitigate feature noises from the network and branches. Subsequently, these features are fused to generate high-quality distilled targets.

2.2 Knowledge Distillation

Knowledge distillation plays a crucial role across various domains, serving as a universal framework for both model compression and transfer learning. By facilitating the transfer of knowledge from a larger, more complex model (teacher) to a smaller, more efficient one (student), knowledge distillation significantly enhances model performance while reducing computational costs. The knowledge distillation framework proposed by Hinton et al. [26] employs softened probability distributions, where the output logits are adjusted using a temperature parameter to retain richer information, making it easier for the student model to learn. However, relying solely on logits may not be sufficient, as they provide limited insight into the model’s internal representations. To address this limitation, researchers have explored various alternative approaches to extract more representative knowledge from intermediate layers for improved knowledge transfer.

For instance, Zagoruyko et al. [27] introduced an attention-based distillation method, which utilizes attention maps derived from the teacher model’s intermediate layers. These attention maps highlight essential regions of the input data, guiding the student model to focus on relevant features. Similarly, Guan et al. [28] proposed an approach that aggregates channel-wise features from multiple layers of the teacher network, allowing the student model to benefit from a more comprehensive set of feature representations. Beyond individual feature-based distillation, some advanced methods [29, 30] have explored the use of inter-sample relationships to enhance knowledge transfer. By capturing structural similarities between data samples, these techniques allow the student model to learn richer feature relationships, leading to improved generalization.

Despite their effectiveness, traditional knowledge distillation methods often require pre-training a teacher model, which adds substantial computational overhead. Additionally, the discrepancy between the teacher and student architectures—often referred to as the generation gap—can hinder the effectiveness of knowledge transfer. To overcome these challenges, self-distillation techniques have emerged as a promising alternative. Unlike conventional approaches that rely on an external teacher model, self-distillation enables a single neural network to generate and transfer knowledge within itself, effectively bypassing traditional limitations.

Based on the structure of a single network, Zhang et al. [31] introduced a multi-branch design, where each branch of the network learns and refines knowledge extracted from the original model, leading to improved performance. However, the knowledge within the branches was not fully utilized, limiting the potential benefits of the design. Ni et al. [17] further enhanced this concept by encouraging different branches to actively share knowledge with the entire network. Nevertheless, their method did not facilitate effective knowledge transfer from the main network back to the branches. Li et al. [16] employed attention mechanisms to fuse features from different network layers and constructed an additional classifier acting as the teacher for branches, which increased the training cost due to the introduction of an extra teacher model. Additionally, Zhang et al. [32] proposed a multi-model architecture method that encourages continuous knowledge exchange among multiple identical models. However, the homogeneity introduced by using identical models may reduce the efficiency of knowledge transfer.

To address these challenges, we propose a Knowledge Fusion Mutual Self-Distillation (KFMSD) method that enables efficient knowledge flow within a single network architecture. This enables the original network and its branches to engage in mutual learning, allowing both to act as effective teachers. To further enhance the representational capability of each branch, we introduce a Knowledge Fusion Module (KFM), which integrates feature maps from the original network and its branches to provide high-quality target features for learning. Moreover, to alleviate the loss imbalance caused by differences in the parameter scales of various branches, we incorporate the parameter-free SimAM attention mechanism within KFM to select effective branch features for fusion. This ensures efficient and lightweight knowledge transfer within the framework without introducing additional parameters or storage overhead.

3 Proposed Method

3.1 Knowledge Fusion Module

KFM, based on SimAM [25], can fuse n feature maps of the same scale into a fusion feature map. The specific details of KFM are illustrated in Fig. 2. There are n feature maps $\{f_p\}_{p=1}^n \in \mathbb{R}^{C \times H \times W}$. C , H , and W represent the number of channels, height, and width of the feature maps, respectively.

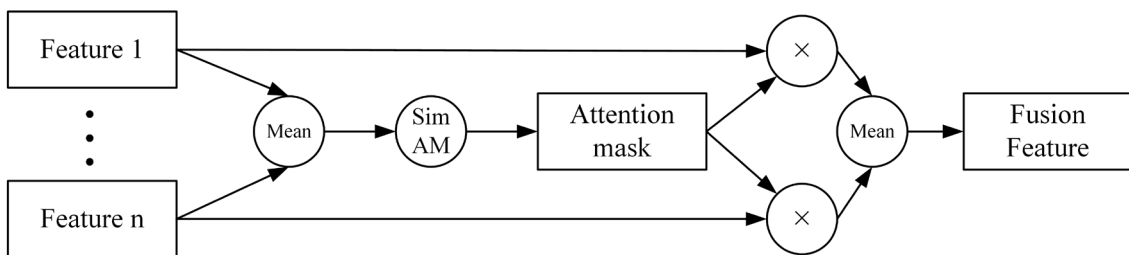


Fig. 2. Knowledge fusion module

Firstly, we compute the element-wise mean of n feature maps, resulting in an average feature map $A \in \mathbb{R}^{C \times H \times W}$:

$$A(i, j, k) = \frac{1}{n} \sum_{p=1}^n f_p(i, j, k) \quad (1)$$

Here i, j , and k represent the indices for channels, height, and width, respectively.

Secondly, the intermediate feature map $M \in \mathbb{R}^{C \times H \times W}$ is obtained by squaring the difference between each element value on each channel and the mean of all elements in that channel in feature map A :

$$M(i, j, k) = \left(A(i, j, k) - \frac{1}{H \times W} \sum_{j=1}^H \sum_{k=1}^W A(i, j, k) \right)^2 \quad (2)$$

Then, the M and a sigmoid operation are used to generate the attention mask $AM \in \mathbb{R}^{C \times H \times W}$:

$$AM(i, j, k) = \text{sigmoid} \left(\frac{M(i, j, k)}{4 \cdot v + 4 \times 10^{-4}} + 0.5 \right) \quad (3)$$

Here v is generated from the feature map M :

$$v = \frac{\sum_{j=1}^H \sum_{k=1}^W M(i, j, k)}{H \times W - 1} \quad (4)$$

Finally, the n feature maps are element-wise multiplied with the AM to obtain n output feature maps. The fusion feature map $FM \in \mathbb{R}^{C \times H \times W}$ is the average of all the output feature maps:

$$FM(i, j, k) = \frac{1}{n} \sum_{p=1}^n (AM(i, j, k) \cdot f_p(i, j, k)) \quad (5)$$

3.2 Knowledge Fusion Mutual Self-distillation Framework

Based on the KFM, we introduce an innovative knowledge distillation method named KFMSD. This method is designed to enhance the knowledge transfer process by decomposing the original network into three distinct segments according to its structural characteristics. Each of these segments is then extended by integrating additional modules to form independent branches. Subsequently, a bidirectional knowledge exchange is established between the original network and each of these newly created branches, allowing them to function as mutual teachers. During the distillation process, KFM plays a crucial role by effectively fusing features derived from both the original network and its branches. This fusion process contributes to improving the overall performance of the branches. Additionally, the fused features generated by KFM serve as high-quality learning targets for refining the performance of the original network itself.

As shown in Fig. 3, we implement the KFMSD framework using ResNet18 as the original network. Specifically, we construct Branch1 by extending ResBlock1 with three additional modules: a downsampling module, module1, and module2.

The detailed architectural design of these three modules is illustrated in Fig. 4. Similarly, we build Branch2 by enhancing ResBlock2 through the addition of the downsampling module and module2. Lastly, Branch3 is formulated based on ResBlock3 by incorporating module2. Each of these branches is further equipped with a fully connected layer to facilitate the learning process. The downsampling module is particularly important as it ensures that the internal feature scales of the branches are aligned with those of the original network. This alignment significantly benefits the fusion process performed by KFM, enhancing the overall knowledge distillation effectiveness. The convolution operations within the downsampling module utilize depthwise separable convolutions, which are instrumental in reducing the storage and computational overhead of the branches. Notably, the internal convolutional structures of both module1 and module2 are identical. However, module2 is uniquely designed to include an adaptive average pooling layer, which can resize the feature map to 1×1 .

Within the KFMSD framework, the feature fusion process is executed in two distinct stages, ensuring effective knowledge exchange between the original network and its branches. The first stage occurs when the output feature B from ResBlock2 is integrated with the output feature A from the downsampling module in Branch1. This fusion process results in a new feature representation, denoted as E. In the second stage, a similar fusion

operation takes place where the output feature D from ResBlock3 is combined with the output feature C from the downsampling module in Branch2, producing the fused feature F. All fusion processes implemented by KFM are performed solely within the branches without affecting the inference of ResNet18. To facilitate the discussion on the roles of different branches in the distillation framework, we refer to the original network as Branch4. During the model deployment phase, Branch1, Branch2, and Branch3 can be entirely removed without modifying the core architecture of the original network. This means that the trained ResNet18 (i.e., Branch4) retains its original structure and inference capabilities without additional computational burden.

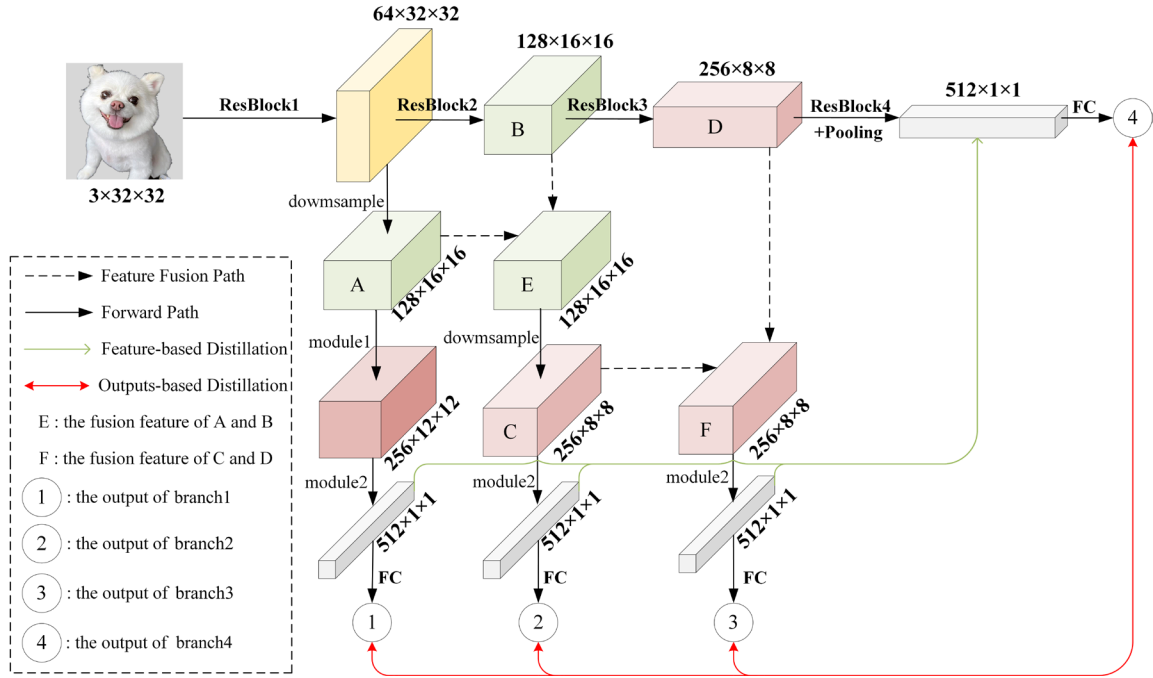


Fig. 3. Details of ResNet18 equipped with proposed KFMSD

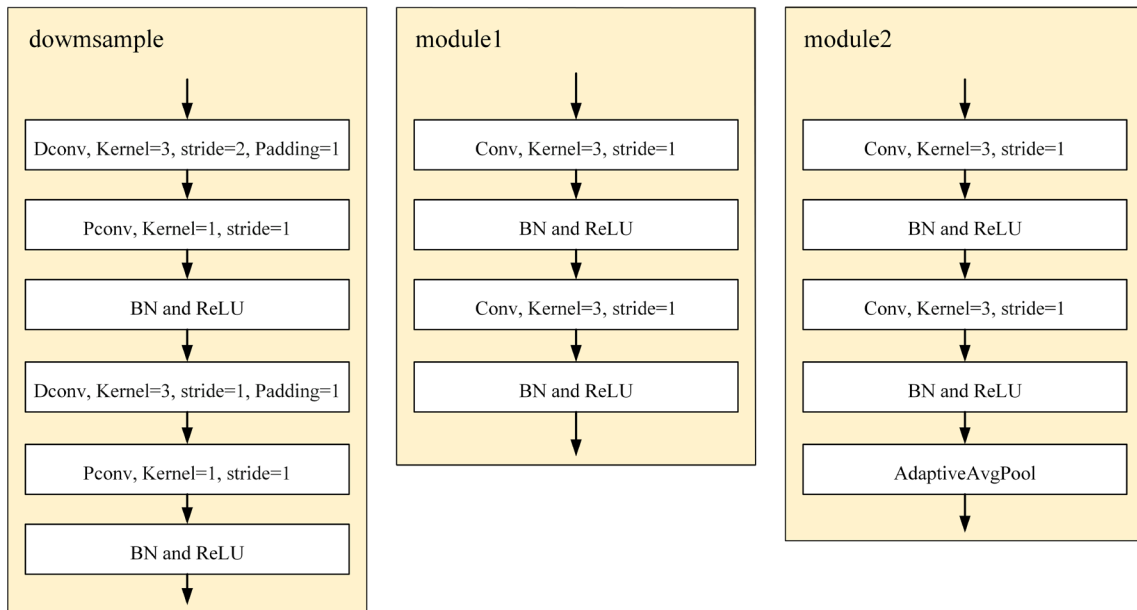


Fig. 4. Structures of branches modules (Conv represents convolution, and BN is batch normalization. Dconv and Pconv refer to depthwise convolution and pointwise convolution.)

3.3 Loss Function

Given that the KFMSD framework has N samples $X = \{x_i\}_{i=1}^N$, G classes $Y = \{y_i\}_{i=1}^G$, and a network with M branches. The outputs of the fully connected layers in the network and each branch must be processed using the softmax with a temperature:

$$p_i = \frac{\exp(k_i/t)}{\sum_j \exp(k_j/t)}. \quad (6)$$

Here p_i represents the output probability of the class i . t and k_i refer to the temperature and the output of the class i after the fully connected layer.

In KFMSD, the total loss is composed of three parts: the dataset's true labels loss ($Loss_{TL}$), the mutual self-distillation loss ($Loss_{MS}$), and the feature loss ($Loss_F$).

$Loss_{TL}$: The original network and branches are supervised with the true labels from the dataset. This approach enables them to acquire knowledge directly from the dataset. The sum of the cross-entropy losses between the four outputs and the true labels:

$$\sum_{i=1}^4 CE(p^i, y) \quad (7)$$

is the first part of the total loss. Here, y and p^i represent the true labels and the softmax output of the original network with $t = 1$. p^i ($i = 1, 2, 3$) represents the softmax output of Branch i with $t = 1$.

$Loss_{MS}$: The original network and branches act as teachers for each other. This approach enables both sides to learn from the accumulated knowledge of each other, collectively improving their performance. The second part of the total loss is the Kullback-Leibler divergence between the original network and the three branches:

$$\sum_{i=1}^3 KLD(q^i, q^4) + \sum_{i=1}^3 KLD(q^4, q^i), \quad (8)$$

where KLD and q^4 refer to the Kullback-Leibler divergence and the softmax output of the original network with $t = 3$. q^i ($i = 1, 2, 3$) represents the softmax output of Branch i with $t = 3$.

$Loss_F$: Since the fusion process is confined within the branches, we use the features before the fully connected layers of branches as learning targets for corresponding original network features. The L2 distance between the network features and the features of all the branches:

$$\sum_{i=1}^3 \|F_4 - F_i\|_2^2 \quad (9)$$

is the third part of the total loss. Here, F_i and F^4 represent the features before the fully connected layers of Branch i and the corresponding original network features.

Additionally, we introduce two hyperparameters, γ and δ , to balance the above three losses. The total loss can be written as:

$$Loss = (1 - \gamma) \cdot Loss_{TL} + \gamma \cdot Loss_{MS} + \delta \cdot Loss_F. \quad (10)$$

4 Experiments

4.1 Experiments Setting

We validate the effectiveness of KFMSD on three datasets (CIFAR-10 [33], CIFAR-100 [33], Tiny-ImageNet [34]) using seven different networks (ResNet [35], WRN [36], ResNeXt [37], VGG [38], SqueezeNet [39], MobileNetV1 [40]). WRN belongs to a wide neural network. ResNet, ResNeXt, and VGG belong to deep neural networks. SqueezeNet and MobileNetV1 are lightweight networks. The CIFAR-10 dataset consists of 60,000 color images, each of size 32x32 pixels, categorized into 10 classes. Each class contains 6,000 images, and they are evenly distributed, making it a balanced dataset. The CIFAR-100 dataset consists of 60,000 color images, each of size 32x32 pixels, which contains 100 classes. These classes are grouped into 20 superclasses, each containing five distinct subclasses. Each subclass has 600 images, making the dataset balanced and evenly distributed. The Tiny-ImageNet dataset consists of 200 classes, each representing a specific object or category. Each class has 600 images. All images are RGB and have a fixed resolution of 64x64 pixels.

Table 1. Experiments setting

Setting	CIFAR-10/100	Tiny-ImageNet
Optimization algorithm	SGD	SGD
Initial learning rate	0.1	0.1
Image size	32x32	64x64
Batchsize	128	64
γ	0.7	0.7
δ	0.03	0.03
t	3.0	3.0

All experiments are conducted on Tesla P100 using PyTorch. Experimental setting is shown in Table 1. We use SGD with a weight decay of $5e-4$ and a momentum of 0.9 to optimize all networks. The initial learning rate is set to 0.1. The recommended values for γ , δ are 0.7 and 0.03. All networks are trained for 200 epochs on CIFAR-10 and CIFAR-100 datasets. The batch size is set to 128. The learning rate is divided by ten at the 66th, 133rd, and 190th epochs. On the Tiny-ImageNet dataset, the batch size is set to 64. All networks are trained for 100 epochs. The learning rate is divided by ten at the 33rd, 66th, and 90th epochs.

4.2 Experimental Results on CIFAR-10

The experimental results of KFMSD on CIFAR-10 are shown in Table 2. Compared to the baseline, the average accuracy improvement of the five networks is 0.58%, and the average ensemble accuracy improvement is 0.90%. In different network architectures, KFMSD shows varied improvements. For instance, on ResNet18, Branch4 achieved the highest accuracy of 95.73%, an improvement of 1.06% over the baseline, while the ensemble model reached 95.69%, closely approaching the highest branch accuracy. On ResNet50, Branch3 achieved the highest accuracy among all branches at 95.70%, with the ensemble model further improving to 96.02%, which is 0.85% higher than the baseline.

Table 2. Experiment results of accuracy (%) on CIFAR-10

Model	Baseline	Branch1	Branch2	Branch3	Branch4	Ensemble
ResNet18	94.67	94.72	94.91	95.56	95.73	95.69
ResNet50	95.17	95.14	95.36	95.70	95.51	96.02
WRN-50-2	95.13	95.44	95.58	95.75	95.78	96.19
ResNeXt50-32-4	95.17	95.34	95.68	95.71	95.77	96.18
VGG19	93.94	93.09	94.28	94.30	94.30	94.53

In more complex networks like WRN-50-2 and ResNeXt50-32-4, Branch4 achieved peak accuracies of 95.78% and 95.77% among their branches respectively, with the ensemble model pushing accuracy to 96.19%

and 96.18%, improving by 1.06% and 1.01% over the baseline. This demonstrates that the KFMSD method exhibits strong adaptability and generalization across networks of varying depth and width.

In contrast, VGG19 showed more noticeable performance degradation at Branch1 (93.09%), but Branch2 and higher exceeded baseline accuracy, resulting in an ensemble model achieving 94.53%, a 0.59% improvement. This indicates that KFMSD remains effective even on standard convolutional networks like VGG.

4.3 Experimental Results on CIFAR-100

The experimental results of KFMSD on CIFAR-100 are shown in Table 3. Compared to the baseline, all networks have an average accuracy improvement of 3.06%, and the average ensemble accuracy improvement is 4.59%. After distillation, the accuracy shows a maximum improvement of 3.71% and a minimum of 2.36%.

Specifically, in the ResNet18 architecture, Branch3 and Branch4 achieved accuracy rates of 78.95% and 80.45%, respectively, representing improvements of 1.31% and 2.81% over the baseline model (77.64%). Furthermore, by incorporating an ensemble learning strategy, the model performance was further improved to 81.02%. These results strongly validate the effectiveness of the KFMSD method in enhancing the representational capacity of ResNet18. Notably, this performance improvement trend is also observed in deeper network architectures such as ResNet34 and ResNet50.

In the experiments on the VGG series, the KFMSD method exhibited distinct performance characteristics. Taking VGG11 as an example, while the accuracy of Branch1 (68.06%) was 1.98% lower than the baseline (70.04%), all subsequent branches outperformed the baseline. Notably, Branch2 achieved an accuracy of 71.32%, surpassing the baseline by 1.28%.

A particularly noteworthy aspect is the outstanding performance of the KFMSD method in lightweight networks. For instance, in MobileNetV1, Branch1 achieved an accuracy of 78.26%, outperforming the baseline by 4.94%. This result indicates that the KFMSD method is also applicable to resource-constrained device scenarios.

Table 3. Experiment results of accuracy (%) on CIFAR-100

Model	Baseline	Branch1	Branch2	Branch3	Branch4	Ensemble
ResNet18	77.64	74.59	76.40	78.95	80.45	81.02
ResNet34	78.41	74.37	76.92	80.46	81.22	81.70
ResNet50	77.14	78.11	78.88	80.84	80.52	82.25
WRN-50-2	77.61	77.89	79.18	81.00	81.32	82.16
ResNeXt50-32-4	78.75	78.39	79.23	80.95	81.75	82.64
VGG11	70.04	68.06	71.32	73.38	73.39	74.57
VGG16	73.07	72.24	75.39	76.60	76.76	77.56
VGG19	72.68	72.23	74.99	75.06	75.04	75.94
SqueezeNet	71.49	74.86	75.95	76.64	73.88	78.06
MobilenetV1	73.32	78.26	77.61	78.18	76.40	80.19

4.4 Experimental Results on Tiny-ImageNet

KFMSD continues to demonstrate outstanding performance when applied to complex datasets. Compared to the CIFAR-100 dataset, KFMSD achieves even greater performance improvements on the more challenging Tiny-ImageNet dataset, further highlighting its effectiveness.

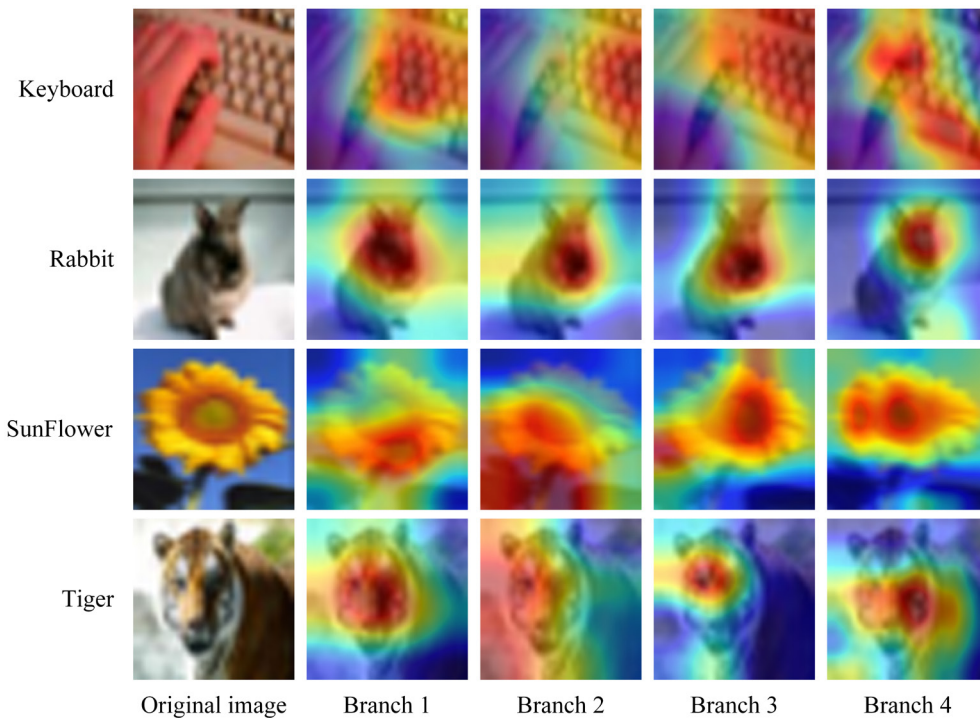
The experimental results, as detailed in Table 4, provide a comprehensive comparison. In contrast to the baseline models, all tested networks exhibit a significant improvement in average accuracy, achieving an overall increase of 4.26%. Furthermore, when utilizing the ensemble strategy, the average accuracy improvement reaches an even higher margin of 6.68%, demonstrating the robustness of the proposed method. Notably, all models incorporating the Branch2 and Branch3 structures outperform their respective baseline counterparts, further substantiating the advantage of the proposed approach. Additionally, aside from ResNet18 and ResNet34, all other models integrated with the Branch1 structure also achieve higher accuracy than their baseline versions, reinforcing the superiority and adaptability of KFMSD across different network architectures. These results collectively validate the efficacy of KFMSD in enhancing classification performance on large-scale datasets.

Table 4. Experiment results of accuracy (%) on Tiny-ImageNet

Model	Baseline	Branch1	Branch2	Branch3	Branch4	Ensemble
ResNet18	57.86	56.39	58.60	59.78	62.18	63.65
ResNet34	58.57	56.72	59.80	61.91	64.88	65.48
ResNet50	60.85	60.96	62.62	64.43	64.04	66.84
VGG16	58.16	58.40	58.30	62.38	63.05	64.86
SqueezeNet	56.25	60.34	60.59	61.62	58.16	62.96
MobilenetV1	58.12	60.91	61.97	65.32	63.08	66.10

4.5 Visualization of Experimental Results

To investigate the image regions attended to by different branches during training and the differences in the knowledge they contribute, this study conducts a visualization analysis on the CIFAR-100 training set using the Grad-CAM technique based on the ResNet18 architecture. We select the last convolutional layer of four branches as the target layer for gradient propagation. As shown in Fig. 5, from Branch 1 to Branch 4, the regions attended to by the model exhibit a clear trend of progressive narrowing. This indicates that as the depth of the position where the branch is constructed based on the original model increases, the model tends to focus more on fine-grained features of the image.

**Fig. 5.** Grad-CAM visualization of the last convolutional layer in each branch

Specifically, taking the visualization results of the Rabbit image as an example, the shallow branch, Branch 1, primarily attends to the overall head region of the rabbit, whereas the deep branch, Branch 4, significantly focuses on the eye region. Due to the substantial differences in the attended regions across branches, the knowledge representations provided during the knowledge distillation process also vary. The shallow branches, acting as teacher networks, can offer more generalized knowledge representations to the deep branches, effectively regularizing their learning process. Conversely, the deep branches, as teacher networks, provide more discriminative knowledge to the shallow branches, thereby supervising their feature learning. Through this bidirectional knowl-

edge transfer mechanism, each branch serves both as a teacher and a student, achieving mutual enhancement and collaborative optimization, ultimately improving the overall performance of the base model.

4.6 Ablation Experiment

To evaluate the effectiveness of each loss term in the proposed approach for improving the performance of different network branches, we conducted comprehensive ablation experiments using three loss terms, $Loss_{TL}$, $Loss_{MS}$, and $Loss_F$, on two baseline networks: ResNet18 and ResNet34. Here, $Loss_{TL}$ represents the cross-entropy loss between the model output and the ground truth labels, which serves as the fundamental loss term in the knowledge distillation algorithm. The experimental results are presented in Table 5, clearly illustrating the impact of different loss combinations on model performance.

The results, as shown in Table 5, demonstrate that, in both the ResNet18 and ResNet34 architectures, enabling all three loss terms ($Loss_{TL} + Loss_{MS} + Loss_F$) simultaneously leads to the optimal performance across all branches. This finding confirms that each loss term contributes positively to model performance improvement. Through an in-depth analysis of the influence of each loss term, we observed that $Loss_{MS}$ has the most significant impact on performance enhancement, whereas the effect of $Loss_F$ is relatively weaker. Specifically, in the ResNet18 network, when using only $Loss_{TL} + Loss_F$ (excluding $Loss_{MS}$), the accuracy of Branch1 and Branch2 showed a decline compared to using only L1. Moreover, the accuracy of all branches was significantly lower than when using $Loss_{TL} + Loss_{MS}$. This trend was further validated in the ResNet34 network: compared to using only $Loss_{TL}$, incorporating $Loss_{MS}$ ($Loss_{TL} + Loss_{MS}$) consistently resulted in greater performance improvement than incorporating $Loss_F$ ($Loss_{TL} + Loss_F$). These findings strongly support the critical role of the $Loss_{MS}$ loss term in enhancing model performance.

Table 5. Performance comparison of different traditional distillation approaches

Model	Branch	$Loss_{TL}$	$Loss_{TL} + Loss_{MS}$	$Loss_{TL} + Loss_F$	$Loss_{TL} + Loss_{MS} + Loss_F$
ResNet18	1	73.27	74.11	73.05	74.59
	2	74.56	76.74	74.65	76.40
	3	78.00	78.38	77.33	78.95
	4	78.60	80.06	79.53	80.45
ResNet34	1	73.34	74.17	73.92	74.37
	2	75.19	76.67	76.11	76.92
	3	79.12	80.20	79.51	80.46
	4	79.70	80.86	80.53	81.22

Table 6. Performance comparison of different traditional distillation approaches

Teacher models	ResNet50	ResNet34	VGG16	ResNet50
Student models	ResNet18	ResNet18	VGG11	MobileNetV1
Teacher baselines	77.14	78.41	73.07	77.14
Student baselines	77.64	77.64	70.04	73.32
KD	79.09	79.38	70.77	73.87
DKD	79.18	79.43	71.35	75.28
LSKD	79.07	79.73	72.40	74.27
NormKD	79.19	79.36	71.85	75.08
FKD	79.24	79.24	71.55	75.45
RSD	79.84	79.84	72.38	74.00
Ours	80.45	80.45	73.39	76.40

*FKD, RSD, and our method do not require a teacher model.

4.7 Comparison of Different Distillation Approaches

To validate the effectiveness of the proposed method, we compared it with traditional knowledge distillation methods (including KD [26], DKD [41], LSKD [42], and NormKD [43]) as well as self-distillation methods (including FKD [16] and RSD [17]). Considering that traditional distillation methods typically employ teacher networks with a large number of parameters and high accuracy, we selected ResNet50 and ResNet34 as teacher

models and relatively smaller models as student networks to comprehensively evaluate the knowledge transfer performance across different architectures.

Table 6 presents the performance comparison of various knowledge distillation methods under different teacher-student architectures. The experimental results demonstrate that KFMSD consistently improves the accuracy of student models in both homogeneous architectures (e.g., ResNet50-ResNet18) and heterogeneous architectures (e.g., ResNet50-MobileNetV1), outperforming all baseline methods. Specifically, when using ResNet18 as the student model, KFMSD boosts its accuracy to 80.45%, achieving a 0.72 percentage point improvement over the second-best method, LSKD (79.73% under ResNet34-ResNet18). Notably, KFMSD exhibits outstanding performance on lightweight models, achieving 76.40% accuracy on MobileNetV1, which surpasses the baseline by 3.08 percentage points and outperforms DKD by 1.12 percentage points. Furthermore, our study reveals that DKD and LSKD exhibit inconsistent performance improvements due to variations in teacher-student architectures. Specifically, DKD performs poorly in the VGG16-VGG11 setting, while LSKD demonstrates suboptimal performance in the ResNet50-MobileNetV1 configuration.

Benefiting from efficient knowledge fusion and mutual learning strategies, KFMSD achieves higher accuracy improvements compared to self-distillation methods such as FKD and RSD. In particular, KFMSD outperforms RSD on MobileNetV1 by 2.40 percentage points.

5 Conclusions

We designed a knowledge fusion mutual self-distillation (KFMSD) framework, which constructs multiple branches based on the architecture of the original network to enable effective mutual learning between the original network and its branches. KFMSD includes a knowledge fusion module (KFM) that enables efficient knowledge transfer between deep and shallow features. Experimental results on CIFAR-10, CIFAR-100, and Tiny-ImageNet demonstrate that KFMSD improves the accuracy of networks such as ResNet18, VGG11, and MobileNetV1 without modifying their architectures. For instance, ResNet18 with KFMSD achieved 80.45% accuracy on CIFAR-100, outperforming the original by 2.81%. The ensemble model further enhances performance, with the KFMSD-based ResNeXt50-32x4d reaching 82.64% accuracy on CIFAR-100. Furthermore, visualization analysis using Grad-CAM reveals that different branches focus on distinct image regions, with deeper branches attending to finer details. This complementary attention pattern supports effective bidirectional knowledge transfer, where shallow branches provide generalized knowledge, and deep branches offer more discriminative supervision. Ablation experiment on each loss term indicates that every loss in the proposed scheme effectively enhances the model’s performance. Additionally, compared to other knowledge distillation methods, this approach significantly enhances the performance of each network with only a single training session.

6 Acknowledgement

This study is supported by Key Research Project of Henan Province (231111210500).

References

- [1] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, *Proceedings of the IEEE* 109(3)(2021) 247-278.
<https://doi.org/10.1109/JPROC.2021.3060483>
- [2] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, *Digital Signal Processing* 73(2018) 1-15.
<https://doi.org/10.1016/j.dsp.2017.10.011>
- [3] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, Y. Miao, Review of image classification algorithms based on convolutional neural networks, *Remote Sensing* 13(22)(2021) 4712.
<https://doi.org/10.3390/rs13224712>
- [4] Y. Sun, G. Shi, W. Dong, X. Xie, MADPL-net: Multi-layer attention dictionary pair learning network for image classification, *Journal of Visual Communication and Image Representation* 90(2023) 103728.
<https://doi.org/10.1016/j.jvcir.2022.103728>

- [5] M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: Overview, challenges and the future, in: N. Dey, A. Ashour, S. Borra (Eds.), *Classification in BioApps*, Springer, Cham, 2018 (pp. 323-350).
https://doi.org/10.1007/978-3-319-65981-7_12
- [6] V. Narayan, P.K. Mall, S. Awasthi, S. Srivastava, A. Gupta, FuzzyNet: Medical image classification based on GLCM texture feature, in: *Proc. 2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, 2023.
<https://doi.org/10.1109/AISC56616.2023.10085348>
- [7] T. Diwan, G. Anirudh, J.V. Tembhurne, Object detection using YOLO: Challenges, architectural successors, datasets and applications, *Multimedia Tools and Applications* 82(6)(2023) 9243-9275.
<https://doi.org/10.1007/s11042-022-13644-y>
- [8] Z.-Q. Zhao, P. Zheng, S.-T. Xu, X. Wu, Object detection with deep learning: A review, *IEEE Transactions on Neural Networks and Learning Systems* 30(11)(2019) 3212-3232.
<https://doi.org/10.1109/TNNLS.2018.2876865>
- [9] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, in: *Proc. 2019 International Conference on Machine Learning*, 2019.
<https://proceedings.mlr.press/v97/zhang19d.html>
- [10] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, Y.J. Lee, GLIGEN: Open-set grounded text-to-image generation, in: *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
<https://doi.org/10.1109/CVPR52729.2023.02156>
- [11] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(7)(2022) 3523-3542.
<https://doi.org/10.1109/TPAMI.2021.3059968>
- [12] F. Yuan, Z. Zhang, Z. Fang, An effective CNN and Transformer complementary network for medical image segmentation, *Pattern Recognition* 136(2023) 109228.
<https://doi.org/10.1016/j.patcog.2022.109228>
- [13] Z. Li, H. Li, L. Meng, Model compression for deep neural networks: A survey, *Computers* 12(3)(2023) 60.
<https://doi.org/10.3390/computers12030060>
- [14] J. Gou, B. Yu, S.J. Maybank, D. Tao, Knowledge distillation: A survey, *International Journal of Computer Vision* 129(6) (2021) 1789-1819.
<https://doi.org/10.1007/s11263-021-01453-z>
- [15] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, K. Ma, Be your own teacher: Improve the performance of convolutional neural networks via self distillation, in: *Proc. 2019 IEEE/CVF International Conference on Computer Vision*, 2019.
<https://doi.org/10.1109/ICCV.2019.00381>
- [16] L. Li, W. Su, F. Liu, M. He, X. Liang, Knowledge fusion distillation: Improving distillation with multi-scale attention mechanisms, *Neural Processing Letters* 55(5)(2023) 6165-6180.
<https://doi.org/10.1007/s11063-022-11132-w>
- [17] S. Ni, X. Ma, M. Zhu, X. Li, Y.-D. Zhang, Reverse self-distillation overcoming the self-distillation barrier, *IEEE Open Journal of the Computer Society* 4(2023) 195-205.
<https://doi.org/10.1109/OJCS.2023.3288227>
- [18] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R.R. Martin, M.-M. Cheng, S.-M. Hu, Attention mechanisms in computer vision: A survey, *Computational Visual Media* 8(3)(2022) 331-368.
<https://doi.org/10.1007/s41095-022-0271-y>
- [19] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: *Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
<https://doi.org/10.1109/CVPR.2018.00745>
- [20] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in: *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
<https://doi.org/10.1109/CVPR42600.2020.01155>
- [21] Z. Qin, P. Zhang, F. Wu, X. Li, FcaNet: Frequency channel attention networks, in: *Proc. 2021 IEEE/CVF International Conference on Computer Vision*, 2021.
<https://doi.org/10.1109/ICCV48922.2021.00082>
- [22] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: *Proc. 2018 European Conference on Computer Vision (ECCV)*, 2018.
https://doi.org/10.1007/978-3-030-01234-2_1
- [23] J. Park, S. Woo, J.-Y. Lee, I.S. Kweon, BAM: Bottleneck attention module, in: *Proc. 2018 British Machine Vision Conference (BMVC)*, 2018.
- [24] X. Li, X. Hu, J. Yang, Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. <<https://arxiv.org/abs/1905.09646>>, 2019 (accessed 19.12.2024).
- [25] L. Yang, R.-Y. Zhang, L. Li, X. Xie, SimAM: A simple, parameter-free attention module for convolutional neural networks, in: *Proc. 2021 International Conference on Machine Learning*, 2021.
<https://proceedings.mlr.press/v139/yang21o/yang21o.pdf>

- [26] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, <<https://arxiv.org/abs/1503.02531>>, 2015. (accessed 19.12.2024).
- [27] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. <<https://arxiv.org/abs/1612.03928>>, 2016 (accessed 19.12.2024).
- [28] Y. Guan, P. Zhao, B. Wang, Y. Zhang, C. Yao, K. Bian, J. Tang, Differentiable feature aggregation search for knowledge distillation, in: Proc. 2020 European Conference on Computer Vision (ECCV), 2020. https://doi.org/10.1007/978-3-030-58520-4_28
- [29] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. <https://doi.org/10.1109/CVPR.2019.00409>
- [30] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, Y. Duan, Knowledge distillation via instance relationship graph, in: Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. <https://doi.org/10.1109/CVPR.2019.00726>
- [31] L. Zhang, C. Bao, K. Ma, Self-distillation: Towards efficient and compact neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 44(8)(2022) 4388-4403. <https://doi.org/10.1109/TPAMI.2021.3067100>
- [32] Y. Zhang, T. Xiang, T.M. Hospedales, H. Lu, Deep mutual learning, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018. <https://doi.org/10.1109/CVPR.2018.00454>
- [33] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, University of Toronto, Toronto, Canada, 2009. <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, International Journal of Computer Vision 115(2015) 211-252.
- [35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016. <https://doi.org/10.1109/CVPR.2016.90>
- [36] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proc. 2016 British Machine Vision Conference (BMVC), 2016.
- [37] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017. <https://doi.org/10.1109/CVPR.2017.634>
- [38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. 2015 International Conference on Learning Representations (ICLR), 2015.
- [39] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. <<https://arxiv.org/abs/1602.07360>>, 2016 (accessed 19.12.2024).
- [40] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications. <<https://arxiv.org/abs/1704.04861>>, 2017 (accessed 19.12.2024).
- [41] B. Zhao, Q. Cui, R. Song, Y. Qiu, J. Liang, Decoupled knowledge distillation, in: Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. <https://doi.org/10.1109/CVPR52688.2022.01165>
- [42] S. Sun, W. Ren, J. Li, R. Wang, X. Cao, Logit standardization in knowledge distillation, in: Proc. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. <https://doi.org/10.1109/CVPR52733.2024.01489>
- [43] Z. Chi, T. Zheng, H. Li, Z. Yang, B. Wu, B. Lin, D. Cai, NormKD: Normalized logits for knowledge distillation, <<https://arxiv.org/abs/2308.00520>>, 2023 (accessed 19.12.2024).