Information Fusion 90 (2023) 12–22

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Diversified branch fusion for self-knowledge distillation

Zuxiang Long [a,1], Fuyan Ma [a,1], Bin Sun [a,*], Mingkui Tan [b], Shutao Li [a]

[a] College of Electrical and Information Engineering, Hunan University, Changsha 430072, China
[b] School of Software Engineering, South China University of Technology, Guangzhou 510641, China

ARTICLE INFO

Keywords:
Deep learning
Self-knowledge distillation
Diversity loss
Knowledge fusion
Multiple branches

ABSTRACT

Knowledge distillation improves the performance of a compact student network by adding supervision from a pre-trained cumbersome teacher network during training. To avoid the resource consumption of acquiring an extra teacher network, the self-knowledge distillation designs a multi-branch network architecture with shared layers for teacher and student models, which are trained collaboratively in a one-stage manner. However, this method ignores the knowledge of shallow branches and rarely provides diverse knowledge for effective collaboration of different branches. To solve these two shortcomings, this paper proposes a novel Diversified Branch Fusion approach for Self-Knowledge Distillation (DBFSKD). Firstly, we design lightweight networks for adding to the middle layers of the backbone. They capture discriminative information by global–local attention. Then we introduce a diversity loss between different branches to explore diverse knowledge. Moreover, the diverse knowledge is further integrated to form two knowledge sources by a Selective Feature Fusion (SFF) and a Dynamic Logits Fusion (DLF). Thus, the significant knowledge of shallow branches is efficiently utilized and all branches learn from each other through the fused knowledge sources. Extensive experiments with various backbone structures on four public datasets (CIFAR100, Tiny-ImageNet200, ImageNet, and RAF-DB) show superior performance of the proposed method over other methods. More importantly, the DBFSKD achieves even better performance with fewer resource consumption than the baseline.

1. Introduction

The Knowledge Distillation (KD) [1,2] is one of the most effective techniques to compress and accelerate the over-parameterized deep models. It is essential to apply the deep learning in the application scenarios with limited computational resources, such as embedded human–computer interaction platforms, in-vehicle driver fatigue detectors, etc.

The conventional knowledge distillation first trains a cumbersome high-performance model as the teacher model. Then a compact student is trained by taking the knowledge of the teacher model as supervision, including the logits as soft targets [3], the feature-map activation boundary [4], and intermediate layer feature maps [5]. So the compact student network can achieve similar performance with less resource consumption and replace the cumbersome teacher network on the deployment stage. Since the training of deep models is computationally expensive and time-consuming, the enormous cost of the two-stage training develops into the ultimate obstacle to the practical application of conventional knowledge distillation.

To reduce the resource consumption of conventional knowledge distillation, the collaborative knowledge distillation methods, such as DML [6], DCM [7], and KDCL [8], employ the same compact architecture for both the student and teacher networks and simplify the execution steps by learning from each other in a one stage manner. Although the collaborative knowledge distillation downsizes the teacher to reduce the resource cost of training, we still need to train two networks with similar performance, one of which is discarded on the deployment stage.

The self-knowledge distillation methods add weak networks [9,10] or early exits [11,12] at the middle layers of the backbone to form multiple branches. As shown in Fig. 1(a), these methods share the backbone with all branches, treating the deeper branch as the teacher network and the shallow branch as the student network. The self-knowledge distillation transfers the knowledge from the deeper branch rather than an extra pre-trained teacher network, which avoids training a cumbersome teacher and reduces the overhead of the training process. On the actual deployment stage, we can choose the more lightweight branch according to different resource constraints.

* Corresponding author.
E-mail addresses: longzx@hnu.edu.cn (Z. Long), mafuyan@hnu.edu.cn (F. Ma), sunbin611@hnu.edu.cn (B. Sun), mingkuitang@scut.edu.cn (M. Tan), shutao_li@hnu.edu.cn (S. Li).

[1] The first two authors Zuxiang Long and Fuyan Ma contributed equally to the work.

https://doi.org/10.1016/j.inffus.2022.09.007
Received 25 May 2022; Received in revised form 3 September 2022; Accepted 6 September 2022
Available online 11 September 2022

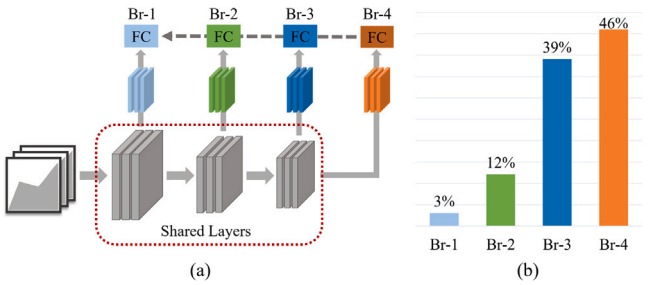

Fig. 1. The complementary performance of different branches in self-knowledge distillation. (a) The multi-branch architecture of the Self-D. (b) The percentages of best classified categories obtained with different branches on CIFAR100.

However, the self-knowledge distillation ignores the useful knowledge of the shallow branches. We obtain the category-wise accuracies of different branches in the Self-D [9] on CIFAR100 and illustrate the percentages of best-classified classes for them in Fig. 1(b). It can be observed that the Br-4 (i.e. the backbone) only achieves the best performance on 46 categories among all the branches. Over half of the all categories are best classified by the three shallower branches (i.e., Br-1 to Br-3). This indicates that the knowledge of the shallow branches can be complementary to the deeper ones. Therefore, it is better to utilize such complementary knowledge to enhance the entire network, rather than simply discarding them. To achieve this goal, we propose a novel knowledge fusion approach to maximize the utilization of shallow branch knowledge.

In addition, the features and logits of different branches become similar with the guidance of the same knowledge source during the training process. Such similarity may impair effectiveness of collaborative learning for different branches [13–15]. Therefore, we introduce an extra diversity loss between different branches to provide richer knowledge for the performance improvement of the whole network.

Based on the above analysis, we propose a new Diversified Branch Fusion approach for Self-Knowledge Distillation (DBFSKD). Our method adds well-designed lightweight networks in the middle layers of the backbone, which extract discriminative features from the backbone by global–local attention and transform these features into a high-dimensional space under few computations. We construct two fused knowledge sources through a Selective Feature Fusion (SFF) and a Dynamic Logits Fusion (DLF) for maximizing the usage of shallow branch knowledge. The SFF can adaptively select important feature maps to generate an additional feature map with richer information. The logits based on the additional feature map are further dynamically fused with the logits of other branches by DLF. The proposed method aggregates the branch knowledge effectively by fusion in both features and logits level. Furthermore, the DBFSKD encourages branches to explore different knowledge by the diversity loss, which improves the performance of the whole network. In summary, the main contributions of this work are as follows:

- We design a lightweight network for adding to the middle layer of the backbone. The network has an efficient attention module to extract required features from the backbone, enabling shallow branches to perform better.
- We explore how to effectively utilize each-branch knowledge to improve the multi-branch self-knowledge distillation methods. Specifically, we design two simple but effective fusion modules named SFF and DLF for adaptively selecting the vital knowledge from each branch to construct two fused knowledge sources at the features and logits level.
- We introduce a diversity loss during training, which is implemented by minimizing the similarity between feature maps. This supervision encourages different branches to mine diverse knowledge for improving the effect of knowledge fusion and producing better teacher.
- Extensive experiments with various networks on three public image classification datasets, i.e., CIFAR100, Tiny-ImageNet200, and ImageNet, prove that the DBFSKD outperforms several state-of-the-art knowledge distillation methods. Additionally, we also verify the effectiveness of the DBFSKD for the application of facial expression recognition on the RAF-DB dataset.

2. Related work

The related work of knowledge distillation mainly includes three modes: (1) Conventional Knowledge Distillation: an experienced teacher network transfers knowledge to a student network in a two-stage manner; (2) Collaborative Knowledge Distillation: networks with the same architecture learn from each other in a one-stage manner; (3) Self-Knowledge Distillation: a network learns its own knowledge in a one-stage manner.

2.1. Conventional knowledge distillation

The classical "teacher–student" two-stage knowledge distillation mode is first proposed by Hinton [3]. The main idea is that the student network simulates the soft target of a pre-trained teacher network. Fitnets [16] extends this idea by allowing the student feature to match the teacher feature and using the L2 norm to narrow the distance between the two. Based on the Fitnets paradigm, many methods have been derived. For example, Guan et al. [17] exploit neural architecture search to find multi-teacher feature aggregation paths for feature-map distillation. Heo et al. [18] propose to distill the activation boundary of the feature map for knowledge transfer. However, simply one-to-one assigning a teacher network layer as the supervisor to the student network may not be conducive to its learning. Chen et al. [5] propose cross-layer distillation with semantic calibration (SemCKD), which can adaptively assign weights to feature maps of different layers through the attention mechanism, so that the student network can find the best matching feature map from the teacher network. In addition, He et al. [19] propose a novel paradigm to transfer the integrated knowledge to the baseline model and improve the performance of incremental segmentation. Conventional knowledge distillation can expand the improvement space of the student network by continuously enhancing the performance of the teacher network. However, it usually needs to preserve a reservoir of persistent data and train a strong teacher network to guide the student network [20], which is time-consuming. Moreover, how to build a suitable bridge between the two for improving the efficiency of knowledge transfer is an urgent problem.

2.2. Collaborative knowledge distillation

The "student-student" collaboration is a one-stage knowledge distillation mode, where the student networks learn from each other during the joint training process. Zhang et al. [6] propose the deep mutual learning (DML) to use soft targets to supervise each other. Inspired by the DML, Yao et al. [7] propose the dense cross-layer mutual-distillation (DCM), which adds many complex auxiliary classifiers to the hidden layers of two networks. And then, the knowledge distillation operations occur between the same-staged and different-staged classifiers of these networks. In the DCM, the paths from the input to all auxiliary classifiers have the same structure as the backbone, so these classifiers cannot reduce resource consumption in the deployment phase. In [8], Guo et al. make the networks produce different predictions by distorting the input image, and achieve cooperation by learning the ensemble results of these predictions. Furthermore, Chung et al. [21] suggest using adversarial learning to perform feature-map distillation between networks. Kim et al. [22] recommend constructing a fusion branch based on multiple independent networks, which acts as an intermediary for collaborative learning. Dissimilar to the above methods, the proposed

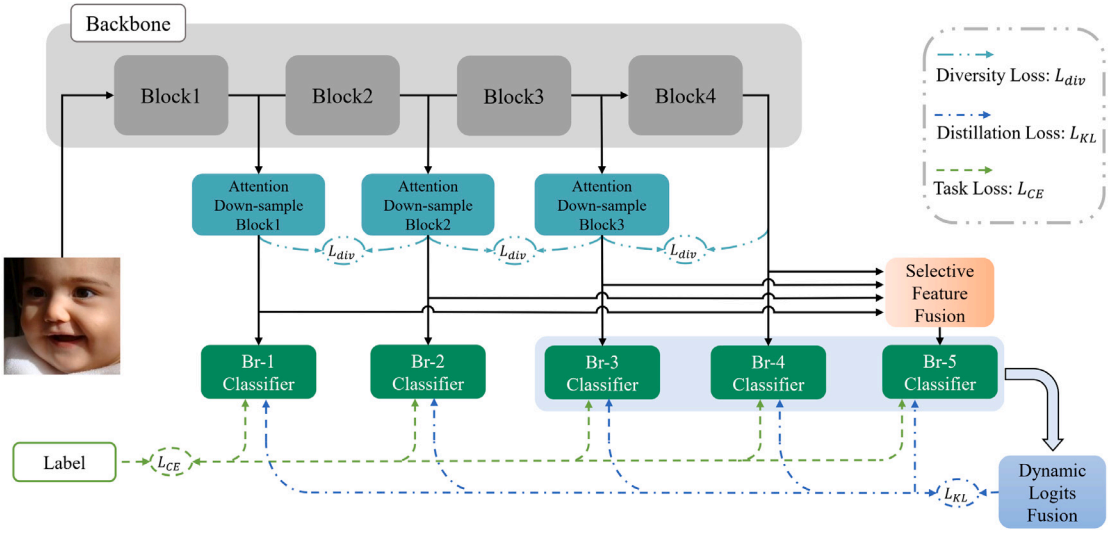

Fig. 2. Our DBFSKD mainly consists of a backbone, three Attention Down-sample Blocks (ADB), a Selective Feature Fusion (SFF), a Dynamic Logits Fusion (DLF), and five classifiers. The ADB includes an attention module and some downsampling layers, which aims to extract features from the backbone and resize them. The SFF is designed to fuse the feature maps from ADB and backbone adaptively, while the DLF is used to fuse the logits. The module passed from the input to a particular classifier output is called a branch.

DBFSKD does not introduce extra networks. Our method builds multiple networks for collaborative learning by adding lightweight networks at different depths of the backbone, which further reduces training costs. In addition, the ultimate goal of the DBFSKD is not only to improve the backbone performance but also to make the shallow branches with fewer parameters and calculations more competitive.

2.3. Self-knowledge distillation

The first two modes need to consider how to select the appropriate guidance network and spend profuse resources to train it. The self-knowledge distillation avoids this problem ingeniously. Zhang et al. [9,10] proposes to add several weak networks at the middle layers of the backbone, regarding the backbone as the deepest branch (i.e., teacher network) and the shallow branches as student networks. It transfers knowledge of features and logits to the shallow branches by deep supervision manner. The self attention distillation (SAD) [23] further learns representation through performing top-down and layer-wise attention distillation within the network itself. Meanwhile, the multi-exits knowledge distillation [11] uses the logits of the deepest branch to guide the shallow branches. To further improve the generalization ability of the multi-exits knowledge distillation, Wang et al. [12] propose a dense multi-exits knowledge distillation framework prompting the shallow branch to learn from all deeper branches. It is worth mentioning that Chen et al. [13] observes the homogenization of the multi-branch knowledge distillation impairs the performance of the whole network. To ensure the branch diversity, they apply a self-attention mechanism to calculate the dependency score of the current branch on others while learning more knowledge from the branch with a high score, which is the key to ensuring branch diversity. However, this method brings a large amount of calculation. Our method utilizes a simple but effective approach to enhance branch diversity by minimizing feature similarity between branches. Different from the above four self-knowledge distillation methods, Yun et al. [24] propose to increase the generalization ability of the network by distilling the logits of different samples with the same label. Later on, Ji et al. [25] propose to design a self-teacher network for extracting feature knowledge from the student network. The self-teacher generates refined feature maps and soft targets for guiding the student network. Since no auxiliary networks are added, the two methods cannot provide different compact branches customized for platforms with variant resources.

3. Proposed method

As shown in Fig. 2, the proposed DBFSKD method utilizes the multi-branch architecture, including three shallow branches (Br-1 to Br-3), one backbone branch (Br-4), and a fusion branch (Br-5). The shallow branch includes backbone blocks and an Attention Down-sample Block (ADB). The backbone branch is consistent with the original backbone network. The fusion branch adds Selective Feature Fusion (SFF) to the previous modules for integrating feature maps from different branches. The final features of all branches will be sent to their own classifiers for calculating the category logits. Then we select the logits to construct a soft target by the Dynamic Logits Fusion (DLF) and apply the fused soft target to transfer knowledge to all branches. More importantly, the DBFSKD introduces the diversity loss between the adjacent branch to encourage different branches to turn up diverse knowledge, which effectively enhances the feature fusion and the logits fusion.

3.1. Background and notations

The key conceptions of knowledge distillation are training the student to learn the original supervision signal and mimic the teacher output. To achieve this goal, the cross-entropy loss and Kullback–Leibler divergence are employed to optimize student training.

Given $Q = \{(x_i, y_i)\}_{i=1}^{N}$ is a dataset with N samples collected from M categories, where x_i represents the ith input sample and y_i represents the corresponding ground truth. The cross-entropy loss L_{CE} of the student is defined as

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} y_i \log\left(\mathrm{softmax}(c_i^{s,m})\right), \tag{1}$$

where $c_i^{s,m} = f_s(x_i)$ is the logit output from the student for the ith sample, which represents the probability that the sample x_i belongs to the mth class. The f_s means the forward inference process of the student.

The loss L_{KL} of Kullback–Leibler divergence is defined as follows

$$L_{KL} = \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M}\sigma(c_i^{t,m})\log\frac{\sigma(c_i^{t,m})}{\sigma(c_i^{s,m})}, \tag{2}$$

where $c_i^{t,m} = f_t(x_i)$ is the logit output from the teacher for the ith sample. The σ is a function to soften the logits to obtain more valuable

knowledge, such as class similarity

$$\sigma(c_i^m) = \frac{\exp(c_i^m/T)}{\sum_{m=1}^{M}\exp(c_i^m/T)}, \tag{3}$$

where the class probability distribution in the soft target becomes smoother with the increase of T.

As a result, the final loss function of the conventional knowledge distillation is written with the balancing parameter γ as follows:

$$Loss = (1-\gamma)L_{CE} + \gamma L_{KL}. \tag{4}$$

3.2. Diversified branch fusion for self-knowledge distillation

In our DBFSKD, we try to solve two shortcomings: how to take full advantage of each branch knowledge and how to encourage different branches to mine diverse knowledge. As mentioned in Section 1, the DBFSKD introduces knowledge fusion and the diversity loss to solve these two shortcomings. The feature maps output from the last convolutional layer of the network contains highly abstract information about the samples. The logits output from fully connected layer of the network, which directly reflects the probability of the category. Therefore, it is feasible to utilize the knowledge of branches by fusing their feature maps and logits. According to the above analysis, we construct two fused knowledge sources in the DBFSKD.

Fused feature-map knowledge source: Many works adopt simple summation and concatenation to perform feature fusion [22,26]. They treat feature maps from different layers equally and suppress those features that contribute to the task instead of useless features. The attention mechanism is an effective method for solving this problem, which determines the importance of different channels and spatial locations in the feature map through learnable parameters. The SENet [27] pioneers channel attention which uses average pooling to collect global information and captures channel-wise relationships by two fully connected layers. The CBAM [28] proposes to use average pooling and maximum pooling to collect global information for channel-wise relationships modeling.

Based on these two outstanding works, we design a simple but efficient feature fusion module named SFF to integrate feature-map knowledge from different branches and construct a more powerful branch. Specifically, the final feature maps of Br-1 to Br-4 are forwarded to the SFF. Given i feature maps $F_n \in \mathbb{R}^{H\times W\times C}$, $(n = 1, 2, 3...i)$, we first concatenate them in the channel dimension to obtain $F_{cat} \in \mathbb{R}^{H\times W\times iC}$. And then, applying channel attention and convolution operation to re-weight the F_{cat} to obtain the feature map $M \in \mathbb{R}^{H\times W\times iC}$. Finally, the fused feature map $F_{Re} \in \mathbb{R}^{H\times W\times C}$ is acquired through 1×1 convolution.

Fused logits knowledge source: Unlike the original self-knowledge distillation treats the logits of the backbone as the soft target for knowledge transfer. The DBFSKD adopts a dynamic fusion manner to construct a fused knowledge source with a more robust generalization capability [8]. Note that the proposed DBFSKD produces five different logits denoted as $c_i^{j,m} = f_i(x_i), j = 1, 2, 3, 4, 5$, which represent the logits of Br-1 to Br-5, respectively. The c_i^3, c_i^4 and c_i^5 are employed for logits fusion. The c_i^1 and c_i^2 are not employed because they contain larger errors, which reduce the robustness of the fused knowledge source. The corresponding experimental analysis is given in Section 4.5. The logits fusion is performed by solving the following optimization problem

$$\begin{aligned} &\boldsymbol{\beta}^* = \min_{\boldsymbol{\beta}\in R^3} L_{CE}(\boldsymbol{\beta}[c_i^3, c_i^4, c_i^5]^T, y_i) \\ &\text{Subject to } \textstyle\sum_{j=1}^{3}\boldsymbol{\beta}_j = 1,\ \boldsymbol{\beta}_j \geq 0, \end{aligned} \tag{5}$$

where $\boldsymbol{\beta}^*$ is the weight vector. We construct the DLF with two fully connected layers for optimizing the objective (5) to obtain the vector. Later on, the $E_i = \boldsymbol{\beta}[c_i^3, c_i^4, c_i^5]^T$ as the soft target to supervise all branches

$$L_{KL} = \frac{1}{N}\sum_{j=1}^{5}\sum_{i=1}^{N}\sum_{m=1}^{M}\sigma(E_i^m)\log\frac{\sigma(E_i^m)}{\sigma(c_i^{j,m})}, \tag{6}$$

where the E_i^m represents the probability that x_i belongs to the mth category.

Diversity loss: The purpose of self-knowledge distillation is to make the output of all shallow branches as similar as possible to the backbone. This approach makes the feature representations of different branches become infinitely close, which may damage the effect of feature fusion and logits fusion. For feature fusion, it is expected that the feature maps provided by different branches have a semantic discrepancy. Thus, the fused feature map may contain richer semantic information. For logits fusion, all logits are expected to have independent error distributions so that others may correct the error made by one branch. Therefore, we introduce an additional loss term in the training process to encourage different branches to discover diverse knowledge. Since adjacent branches have greater similarity in network capacity, the diversity loss L_{div} is defined between two adjacent branches to minimize the similarity of their feature maps.

$$L_{div} = -\frac{1}{NS}\sum_{i=1}^{N}\sum_{j=1}^{3}\|F_i^j - F_i^{j+1}\|_2^2, \tag{7}$$

where the $F_i^j \in \mathbb{R}^{H\times W\times C}$ is the feature map output by the jth branch for the input sample x_i and the $S = HWC$. To sum up, the optimization function of our DBFSKD is

$$Loss = \sum_{j=1}^{5}L_{CE} + \gamma L_{KL} + \alpha L_{div}. \tag{8}$$

where, α and γ are the balance parameters of L_{div} and L_{KL}, respectively.

3.3. Attention down-sample block

The proposed DBFSKD performs adaptive feature map extraction on the backbone by an attention module and makes its resolution consistent with the last convolutional layer of backbone by the down-sampling layers. We design the attention module from the perspective of reducing the number of parameters and calculations. As shown in Fig. 3, the attention module contains two parallel paths on the top and bottom. The bottom path calculates spatial attention, while the top path calculates channel attention. Given the input feature map $F \in \mathbb{R}^{W\times H\times C}$, the channel attention calculation can be expressed as

$$A_c = \sigma(\mathbb{C}(\mathbb{AP}(F))),\ A_c \in \mathbb{R}^{1\times 1\times C}, \tag{9}$$

where the σ, $\mathbb{C}$, and $\mathbb{AP}$ symbolize activation function, 1×1 convolution, and adaptive pooling, respectively. The size of F is reduced to $1\times 1\times C$ by $\mathbb{AP}$ and then a pre-weight map is generated through $\mathbb{C}$. Finally, we use the σ (i.e., Sigmoid) to normalize the pre-weight map to generate the final weight map A_c.

The spatial attention calculation can be expressed as

$$\begin{aligned} &F' = \mathbb{C}(F),\ F' \in \mathbb{R}^{W\times H\times 1} \\ &A_s = \sigma(F'_x) \otimes \sigma(F'_y), \end{aligned} \tag{10}$$

we first adopt 1×1 convolution to reduce the dimension of F to obtain F', then perform pooling in the x and y axes of F' to obtain $F'_x \in \mathbb{R}^{1\times H\times 1}$ and $F'_y \in \mathbb{R}^{W\times 1\times 1}$. After normalization by the activation function, the attention weight map in the x and y directions is obtained. The spatial attention weight map with the size of $W\times H\times 1$ is obtained by multiplication. Finally, the overall attention is calculated as

$$\widetilde{F} = F \odot A_c \odot A_s. \tag{11}$$

The down-sampling layer includes two deep-wise convolutions and two point-wise convolutions. Specifically, the resolution of the input feature map is altered to half after the first deep-wise convolution. We borrow from MobileNetV2 [29], forming an inverted residual block with the remaining three convolutions and controlling the number of channels by an expansion factor. The number of down-sampling layers N is determined according to the backbone. In our method, since all feature maps need to be concatenated, the N of three added networks is 3, 2, and 1, which can ensure all feature maps with consistent resolution.

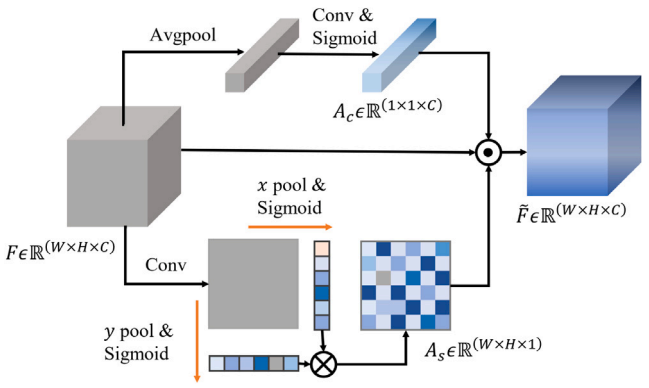

Fig. 3. Illustration of the attention module. x pool and y pool represent pooling in the x-axis and y-axis directions, respectively. The $\otimes$ denote the kronecker product and the $\odot$ denote broadcasting element-wise multiplication. The bottom path calculates spatial attention, while the top path calculates channel attention. The attention calculations of spatial and channel dimensions are computed in parallel.

4. Experiments

To verify the effectiveness of our method, extensive experiments are conducted on two tasks with four benchmark datasets. We first narrate the experimental details and then compare the DBFSKD with other methods. Subsequently, we discuss the impact of both the number of branches and the fusion manner in logits fusion. And then, the impact of each component in the proposed method is quantitatively analyzed through an ablation experiments. Finally, we further verify the superiority of DBFSKD through visual analysis.

4.1. Datasets and implementation details

Datasets: We evaluate our proposed DBFSKD on two tasks: image classification and facial expression recognition. The datasets include: CIFAR100, Tiny-ImageNet200, ImageNet, and RAF-DB.

(1) CIFAR100: CIFAR100 is the most commonly used benchmark dataset for image classification tasks, which contains 100 categories, each with 500 training images and 100 testing images. We resize the images to 32×32 for evaluation.

(2) Tiny-ImageNet200: Tiny-ImageNet200 is a more challenging dataset than the CIFAR100. The dataset contains 200 classes, each class has 500 training images and 50 testing images. we resize all images to meet the same size of the CIFAR100 for evaluation.

(3) ImageNet: ImageNet is the largest classification datasets with more than 1.2 million RGB images collected from the Internet. It provides 1000 categories, which makes classification significantly more difficult.

(4) RAF-DB: RAF-DB contains 29,672 real-world facial images collected from Flickr and about 40 independent annotators label each image. We only use single-label subset of the RAF-DB in our experiments, including seven basic emotions (neutral, happy, surprise, sad, angry, disgust, and fear). The images from the single-label subset are split into 12,271 training images and 3068 testing images. We resize the images to 224×224 for evaluation and use overall sample accuracy for performance measurement.

Implementation details: All experiments are done with Pytorch1.7.0, CUDA11.0, and cudnn8.0.4 on GPU devices. The SGD optimizer with momentum is used to optimize the whole networks and the initial learning rate is set to 0.1 for image classification, and 0.01 for facial expression recognition, respectively. On CIFAR100, the networks are trained for 300 epochs and the learning rate is divided by 10 at epoch 130, 220, and 280. On ImageNet, the total epochs reduce to 100 and the learning rate is divided by 10 every 30 epochs. We do not use the AutoAugment [30] for ImageNet, which is different from Self-D [9]. On Tiny-ImageNet200, we set total epochs as 200 and divide the learning rate by 10 at epoch 100 and 150. On RAF-DB, we set the total epoch to 60 and use an exponential decay. The batch size is set as 256 for CIFAR100 and Imagenet, 128 for Tiny-ImageNet200, and 64 for RAF-DB. The network structure needs to be modified for the small resolution of the images in the CIFAR100 and Tiny-ImageNet200. For ResNet, we follow the Self-D. For MobileNetv2 and ShuffleNetv2, we modify the stride of the first convolution layer and the first block to 1 and add lightweight networks at each down-sampling block. In addition, the diversity loss and the KL divergence loss balance parameters α and γ are set to 5e–5 and 1.5, respectively.

Table 1
Accuracy (%) comparison of DBFSKD with conventional and collaborative knowledge distillation methods on the CIFAR100. The best results of our method and other methods are shown in bold and underlined.

T-Model S-Model	Year	ResNet50 ResNet18	ResNet101 ResNet18	ResNet101 ResNet50
T-Acc	–	80.88	82.37	82.37
S-Acc|Baseline	–	79.01	79.01	80.88
KD [3]	2015	80.49	80.31	82.09
DML [6]	2018	80.52	80.57	82.37
RKD [31]	2019	80.69	80.67	82.29
SPKD [32]	2019	80.57	80.45	82.16
Feat [33]	2019	<u>80.91</u>	<u>80.80</u>	<u>82.40</u>
DBFSKD	2022	**82.65**	**82.65**	**85.18**

4.2. Comparison with state-of-the-art methods

Comparison with Self-D: We propose DBFSKD after analyzing the shortcomings of the multi-branch self-knowledge distillation methods, so we first comprehensively compare these two methods on the CFIAR100. Fig. 4 shows the comparison between the proposed DBFSKD and Self-D in terms of parameters, MACs, and accuracy. The orange and blue points represent the four classifiers of DBFSKD and Self-D, while the green points represent the baseline. It is observed that (i) Both the DBFSKD and the Self-D can effectively improve the accuracy of the backbone branch compared with the baseline. (ii) The accuracy of the branches in DBFSKD surpasses the corresponding ones of the Self-D. (iii) Especially, Our method achieves better performance improvement on the shallow branches. For example, the gain of the second branch of ResNet50 exceeds that of the backbone branch of Self-D.

Comparison with Other Methods: To further confirm the advantages of the proposed method, we compare it with a variety of representative knowledge distillation methods. The experiments are divided into two cases: (1) Distilling knowledge from other networks, which are conventional knowledge distillation and collaborative knowledge distillation: KD [3], SemCKD [5], DML [6], RKD [31], SPKD [32], Feat [33], and CL-ILR [34]. (2) Distilling knowledge from itself, which is self-knowledge distillation: FFL [22], BYOT [10], FRSKD [25], CS-KD [24], OKDDip [13], and ONE [35]. For a fair comparison, we only compare the accuracy of the Br-4 with other methods, which has the same structure as the baseline network.

The experimental results of case (1) are shown in Table 1. As we can see, the DBFSKD with ResNet18 and ResNet50 as the backbone network acquires 82.65% and 85.18% on CIFAR100, respectively. Compared with the best methods Feat, our method obtains gains of 1.74% and 2.78% on the ResNet18 and ResNet50. The experimental results of case (2) are shown in Table 2. Overall, our proposed method achieves 60.88% on Tiny-ImageNet200. In detail, the DBFSKD achieves accuracy improvements of 6.28% and 1.13% over the baseline and the previous best method OKDDip.

We select some state-of-the-art methods and compare them with DBFSKD on the ImageNet validation set. The results are reported in Table 3. When using ResNet18 and ResNet34 as the backbone, we obtain 71.09% and 74.71% on ImageNet, respectively. The Self-D uses

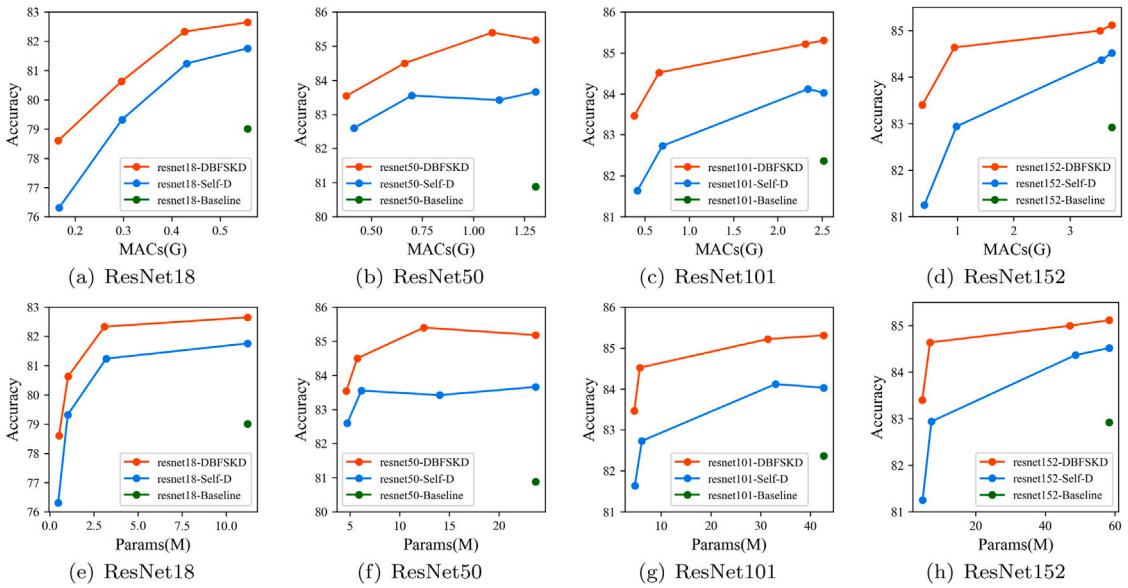

Fig. 4. Comparison of DBFSKD and Self-D on the CIFAR100 test set. The orange and blue lines represent DBFSKD and Self-D, respectively, where each point represents a branch. The green point represents the baseline. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

AutoAugment for data augmentation, achieving 70.51% with ResNet18 as the backbone. Our method not uses the AutoAugment for training but also outperforms the benchmark method Self-D by 0.58%. The SemCKD achieves higher accuracy by using ResNet34 as the teacher network, but our method still outperforms SemCKD by 0.22%. On ResNet34, the DBFSKD obtains a gain of 0.5% over the best method DML. These results demonstrate the effectiveness and the superiority of our method in knowledge transfer.

4.3. Experiments with more network architectures

To examine the effectiveness of our method on different network architectures and scales, we also performed multiple experiments on CIFAR100 and ImageNet with ResNet34, WideResNet50, ResNeXt50, SENet18, MobiltNetV2, and ShuffleNetV2.

Results on CIFAR100: Table 4 shows the comparison of accuracy between the DBFSKD and the baseline. These conclusions can be found by careful analysis: (i) Compared with the standard trained baseline, our proposed method effectively improves the performance of the whole network. With the same parameters and MACs, the Br-4 obtains an average accuracy improvement of 3.06%. (ii) Our DBFSKD outperforms the baseline on the shallowest branch Br-1 in WideResNet50, ResNext50, and Shufflenetv2. Moreover, the shallowest branch has huge computing and storage advantages. For example, the Br-1 of DBFSKD with WideResNet50 as the backbone gets a gain of 1.96% over the baseline, which is only 23% and 9% of the baseline in terms of computation and parameters. (iii) In all network architectures, the Br-5 performs better than other branches, which indicates that the SFF produces feature maps with richer semantics.

Results on ImageNet: The results are shown in Table 5. According to observations, we can draw two conclusions: (i) Compared with the standard trained baseline, the accuracy of the backbone trained with our method increased by 1.64% on average. (ii) Consistent with the experimental results on the CIFAR100 dataset, the Br-5 obtains the highest accuracy, which shows that the proposed SFF module has robust performance.

4.4. Application of facial expression recognition

Facial expression recognition is the most important way for computers to understand human emotions. Therefore, numerous research endeavours have been invested for promoting the development of FER [36–42,44]. However, these works use complex networks to obtain higher performance, ignoring the application of FER systems in resource-constrained environments, such as embedded human–computer interaction platforms, in-vehicle driver fatigue detectors. In this section, we present the experiments on the RAF-DB. For fair comparison with previous state-of-the-art methods, we use the same backbone ResNet18 pre-trained on ImageNet.

Comparison with other state-of-the-art methods can be found in Table 6. The results of DBFSKD in the table are the average of three experiments. Overall, our proposed method shows stable advantages on the FER task, setting a new state-of-the-art on RAF-DB with 88.62%.

Table 2
Accuracy (%) comparison of DBFSKD with the self-knowledge distillation methods on the Tiny-ImageNet200. The best results of our method and other methods are shown in bold and underlined.

Methods	Year	Accuracy ResNet18
Baseline	–	54.60
ONE [35]	2018	57.53
BYOT [10]	2019	56.61
CS-KD [24]	2020	56.46
OKDDip [13]	2020	<u>59.75</u>
FRSKD [25]	2021	59.61
DBFSKD	2022	**60.88**

Table 3
Accuracy (%) comparison of DBFSKD with state-of-the-art methods on the ImageNet val set, where "SemCKD" use ResNet34 as teacher. The "–" indicates that the corresponding result is not provided. The best results of our method and other methods are shown in bold and underlined.

Methods	Year	Accuracy	
		ResNet18	ResNet34
Baseline	–	69.30	73.21
DML [6]	2018	70.65	<u>73.97</u>
BYOT [10]	2019	69.84	–
Self-D[a] [9]	2021	70.51	–
SemCKD [5]	2021	<u>70.87</u>	–
FRSKD [25]	2021	70.17	73.75
DBFSKD	2022	**71.09**	**74.71**

[a]The method uses AutoAugment for training.

Table 4
Accuracy (%) results of DBFSKD on the CIFAR100 test set. Where "Acc" represents the accuracy of the classifier, "P" and "M" represent the number of parameters and MACs of the branch, and the "–" indicates that the corresponding result is not available. The parameters and MACs of the baseline are consistent with the Br-4 and the value in brackets represents the accuracy improvement of the Br-4 compared to the baseline.

Models	Baseline	Br-1		Br-2		Br-3		Br-4		Br-5
		Acc(%)	M(G) | P(M)	Acc(%)	M(G) | P(M)	Acc(%)	M(G) | P(M)	Acc(%)	M(G) | P(M)	
WideResNet50	81.26	83.22	0.836 | 5.91	84.44	1.717 | 9.34	84.57	3.034 | 29.91	84.60 (3.34 ↑)	3.694 | 67.03	84.77
ResNext50	82.65	83.25	0.398 | 5.48	84.07	0.693 | 6.63	84.27	1.132 | 13.48	84.18 (1.53 ↑)	1.353 | 23.18	84.61
ResNet34	79.26	79.05	0.240 | 0.55	82.44	0.522 | 1.65	83.52	0.955 | 8.45	83.75 (4.49 ↑)	1.162 | 21.33	83.91
SENet18	79.53	78.62	0.166 | 0.53	80.95	0.296 | 1.05	82.27	0.427 | 3.16	82.43 (2.90 ↑)	0.557 | 11.31	82.55
MobileNetv2	78.12	73.34	0.026 | 0.61	76.38	0.038 | 0.65	79.59	0.070 | 1.36	80.85 (2.73 ↑)	0.093 | 2.35	80.92
Shufflenetv2	74.59	76.07	0.022 | 0.83	77.45	0.038 | 1.08	–	–	77.93 (3.34 ↑)	0.046 | 1.36	78.27

Table 5
Accuracy (%) results of DBFSKD on the ImageNet val set.

Models	Baseline	Br-1	Br-2	Br-3	Br-4	Br-5
ResNet18	69.30	62.75	65.78	70.29	71.09 (1.79 ↑)	72.30
ResNet34	73.21	63.46	68.03	73.99	74.71 (1.50 ↑)	75.21

Table 6
Accuracy (%) comparison of DBFSKD with other FER methods on the RAF-DB. The best results of our method and other methods are shown in bold and underlined.

Methods	Year	Accuracy
VGG [36]	2018	69.34
DLP-CNN [36]	2018	82.74
FSN [37]	2018	81.14
gACNN [38]	2018	85.07
RAN [39]	2020	86.90
SCN [40]	2020	87.03
DSAN-VGG-RACE [41]	2020	85.37
SPWFA-SE [42]	2020	86.31
STSN [43]	2021	87.52
VTFF [44]	2021	88.14
DBFSKD (Br-1)	2022	87.45
DBFSKD (Br-2)	2022	87.80
DBFSKD (Br-3)	2022	88.18
DBFSKD (Br-4)	2022	88.28
DBFSKD (Br-5)	2022	**88.62**

Table 7
The influence of classifier scheme in the logits fusion. ResNet18 is used as backbone on the CIFAR100 test set."Avg" means the average accuracy improvement compared to setting (e) and the best results are demonstrated in bold.

Setting	Br-1	Br-2	Br-3	Br-4	Br-5	Avg
a	**79.05**	80.51	81.81	81.87	82.35	1.85
b	78.85	80.57	82.02	82.08	82.60	1.95
c	78.60	**80.63**	**82.33**	**82.65**	**82.96**	**2.16**
d	78.52	80.45	81.67	81.78	82.25	1.66
e	77.63	78.85	80.26	80.10	79.51	–

In detail, the Br-5 of DBFSKD has acquired gains of 19.28% and 0.48% over VGG and VTFF, which are the baseline method and the previous SOTA method, respectively. The VTFF pioneers the application of Transformers for FER and achieves better performance than previous CNN methods. However, the self-attention mechanism increases computational complexity, making VTFF unsuitable for deployment on resource-constrained devices. Under the same amount of parameters and computation constraints, the Br-4 of DBFSKD obtains a gain of 1.25% compared to SCN. Even the most lightweight Br-1 still achieves a gain of 0.42%. STSN uses ResNet50/18 as teacher and student respectively, which is more resource-intensive than our method. In addition, DBFSKD achieves better performance than STSN in the shallow branch Br-2. The results show that our method has outstanding advantages for deployment on resource-constrained devices.

4.5. Influence from different strategies in logits fusion

In this subsection, we explore the impact of the number of branches and fusion manner in the logits fusion on our method.

Table 8
The influence of fusion method in the logits fusion. ResNet18 is used as backbone on the CIFAR100 test set, and the best results are demonstrated in bold.

Setting	Br-1	Br-2	Br-3	Br-4	Br-5
x	78.60	**80.63**	**82.33**	**82.65**	**82.96**
y	**78.61**	80.12	81.44	81.87	82.30
z	78.51	80.38	81.78	81.96	82.32

Influence from number of classifiers: Consider the following five settings:

(1) Setting (a): The logits of all branches are used for dynamically weighted fusion.

(2) Setting (b): The logits of all branches except the Br-1 are used for dynamically weighted fusion.

(3) Setting (c): The logits of Br-3, Br-4, and Br-5 are used for dynamically weighted fusion.

(4) Setting (d): Only the logits of the Br-4 and Br-5 are used for dynamically weighted fusion.

(5) Setting (e): No logits fusion is utilized to construct teacher while the Br-5 is taken as the teacher of the other branches. This setting is baseline method in the experiment.

Table 7 presents the experimental results of using ResNet18 as the backbone on the CIFAR100 data set. According to observation, we can find the setting (a, b, c, d) with fusion policy achieve better performance over setting (e), which indicates that the proposed dynamic logits fusion has a positive effect. Specifically, we can see the settings(a, b, c, d) exceed the baseline on the Br-2, while the setting (e) needs to achieve it on the Br-3. This phenomenon evinces that the fused logits knowledge source helps shallow branches to learn more task-oriented knowledge. Among the four settings with fusion policy, the setting (c) obtains the best performance. Despite the fact that Br-1 and Br-2 can bring more complementary knowledge, but their network capacity are smaller compared with other branches, which leads to their prediction having a larger error. The setting (c) guarantees the information source required for fusion and avoids the errors caused by the Br-1 and Br-2. Therefore, setting (c) achieves the highest average accuracy improvement by 2.16% over the baseline.

Influence from fusion manner: As mentioned earlier, the DBFSKD regards the logits fusion as an optimization problem and uses the gradient descent method to solve the optimal weight in each iteration. In this subsection, we carefully compare the performance difference between three settings: (1) setting (x) is the proposed dynamic fusion manner; (2) setting (y) is the naive fusion manner, which simply averages the objects participating in logits fusion; (3) setting (z) is the fixed-weight fusion manner, which assigns weights to different objects based on experience. In this experiment, we assign the weight of Br-3, Br-4, and Br-5 is 0.25, 0.35, 0.4, respectively. We conduct experiments with these three logits fusion settings on CIFAR100 using ResNet18 as the backbone network of our method. Table 8 shows the experimental results. It can be observed that the proposed dynamic fusion achieves the best performance.

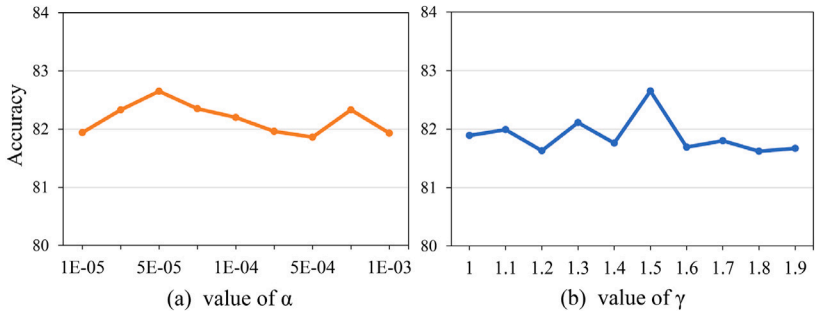

Fig. 5. Sensitivity studies of α and γ on CIFAR100 with ResNet18. We only report the accuracy of Br-4 in the experiments.

Table 9
The effect of the number of branches on DBFSKD, where setting (g) is the default configuration in the paper, while the baseline is the normally trained network. ResNet18 is used as the backbone on the CIFAR100 test set.

Setting	Number of Br	Br-1	Br-e2	Br-2	Br-e3	Br-3	Br-4	Br-5
Baseline	1	–	–	–	–	–	79.01	–
a	3	77.48	–	–	–	–	80.67	80.86
b	3	–	–	79.64	–		81.09	81.37
c	3	–	–	–	–	81.06	81.69	81.81
d	4	78.02	–	80.51	–	–	81.44	81.94
e	4	77.72	–	–	–	81.41	81.87	82.46
f	4	–	–	79.77	–	81.85	82.10	82.57
g	5	78.60	–	80.63	–	82.33	82.65	82.96
h	6	78.77	79.74	81.31	–	82.22	82.55	82.89
i	6	78.57	–	81.12	81.59	82.30	82.60	83.07
j	7	78.43	79.66	81.20	81.80	82.38	82.64	83.18

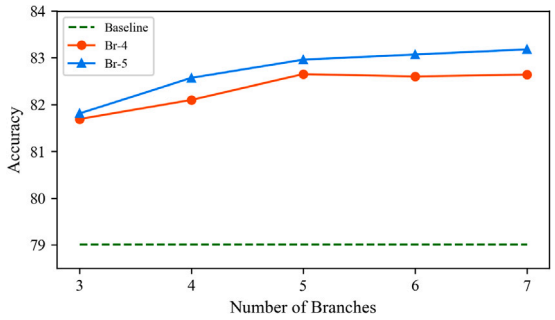

Fig. 6. The effect of the number of branches on Br-4 and Br-5.

4.6. Sensitivity studies of hyper-parameters

In DBFSKD, we introduce two parameters α and γ to control the ratio of L_{div} and L_{KL}, respectively. Fig. 5 gives a sensitivity study on CIFAR100 for these two parameters. According to subfigure (a), when α ranges from 1e-5 to 1e-3, the classification accuracy of DBFSKD ranges 81.86% to 82.65%. In detail, the best result of 82.65% is obtained when α is set to 5e–5. As α increases, the accuracy decreases slightly, floating around 82.00%. In the worst case, α set as 5e–4, our method still leads to a gain of 2.85% over the baseline. According to subfigure (b), when γ ranges from 1 to 1.9, the accuracy of the proposed method ranges from 81.28% to 82.65%. The best result is obtained at γ set as 1.5. In the worst case, γ set as 1.8, the DBFSKD still achieves 2.27% higher than the baseline.

4.7. Influence from the number of branches

To investigate the effect of different branch numbers on the proposed method, we conduct experiments with different branch settings. As shown in Table 9, Br-4 and Br-5 are preserved in all the settings and used for the performance comparison. Setting (g) is the default configuration used in our method. Settings (a-f) are the results with fewer branches, where the first 3 branches in setting (g) are removed or shifted to different positions. Settings (h-j) are the results with more branches, where two extra branches Br-e1 and Br-e2 are inserted after the middle layers of block2 and block3.

By comparing the results for different numbers of branches, we can observe that our method achieves better performance with more branches. For example, setting (c) has acquired gains of 2.68% over the baseline on Br-4, while setting (g) with one more branch has improved 0.96% over setting (c) on Br-4. The reason may be that more branches provide more diverse knowledge under the supervision of the diversity loss. From Fig. 6, we can get an intuitive conclusion that the accuracies of Br-4 and Br-5 increase with the increasing of the branch number, which indicates the proposed method works for different branch numbers, not only for 5 branches. However, from the results with settings (g, h, i, j), we can find that when the performance gains with more than 5 branches are no longer as significant as those with less than 5 branches. The reason may be that too dense branches lead to more shared parameters and less diversity between adjacent branches, which limits the performance of knowledge fusion. Further increasing the number of branches obtains less performance gains but requires more training cost. To effectively balance network performance and training overhead, we propose to choose five branches for the ResNet.

Table 10
Ablation study of loss terms on the CIFAR100 test Set. ResNet18 is used as backbone. The best results are demonstrated in bold.

Setting	SFF	L_{div}	L_{KL}	Br-1	Br-2	Br-3	Br-4	Br-5
i	✗	✗	✗	74.98	77.25	79.92	80.08	–
ii	✗	✗	✓	78.10	79.68	81.54	82.00	–
iii	✗	✓	✓	78.18	80.23	81.76	82.28	–
iv	✓	✗	✗	75.30	78.15	80.30	80.09	81.18
v	✓	✗	✓	78.35	80.40	81.92	82.06	82.45
vi	✓	✓	✓	**78.60**	**80.63**	**82.33**	**82.65**	**82.96**

Additionally, for the results with the same number of branches, we investigate the effect of the branch position. By comparing the results with settings (a–c), with sufficient different (one block in our experiment), the model obtains better performance when the branch is added at the deeper layer. This can also be observed from the results with settings (d–f). The reason may be that the deeper branch can extract more discriminative features, benefiting the subsequent SFF and DLF.

In different application scenarios, we propose to choose the branch number according to the block number of the backbone network and the available computational resources. The backbone with deeper architecture may allow larger number of diversified branches. As long as the computational resource is sufficient, more and deeper branches should be used to improve the model performance. In case of limited computational resources, the number and position of branches can be adjusted to harmonize the model size and the computational efficiency.

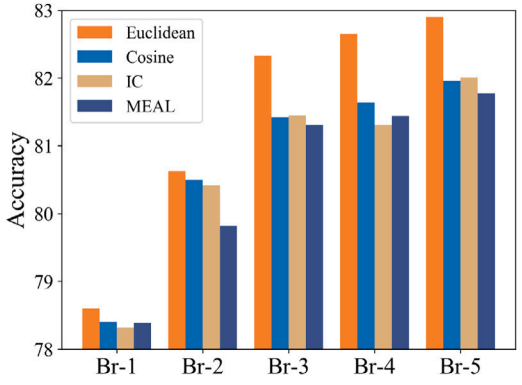

Fig. 7. Comparison of different diversity losses on the CIFAR100.

Table 11
Ablation study of attention module on the CIFAR100 test Set. ResNet18 is used as backbone. "Avg" means the average accuracy improvement compared to DBFSKD without the attention module. The best results are in bold.

Method	Br-1	Br-2	Br-3	Br-4	Br-5	Avg
w/attention module	**78.60**	**80.63**	**82.33**	**82.65**	**82.96**	**1.06**
w/spatial path	78.42	80.62	81.65	81.93	82.31	0.61
w/channel path	78.09	80.15	81.42	81.89	81.95	0.32
w/o attention module	77.59	79.70	80.97	81.68	81.94	–

4.8. Ablation studies

Ablation experiments of loss and SFF: On the basis of previous work, we propose the selective feature fusion module SFF, the distillation loss L_{KL}, and the diversity loss L_{div}. To explore the importance of different components in the proposed method, we adopt ResNet18 as the backbone network to conduct a series of ablation experiments with different settings. Note that the diversity loss is introduced under the premise of multi-branch collaborative learning, which is specifically implemented by SFF and DLF, so that the effects of L_{div} is not separately verified in the ablation study. We divided the ablation experiments into two settings with and without SFF to verify the effect of the fusion branch. Then, we verify the effect of different loss terms under these two settings, respectively. All experimental results are shown in Table 10.

The setting (i) without the SFF and only trained under the supervision of the task loss L_{CE}, which as the baseline setting. According to setting(i, iv), the SFF improves the baseline setting (i) on shallow branches by 0.32%, 0.9%, and 0.38%. Furthermore, our method obtains a fusion branch with 81.18%, indicating the SFF fuses features from different branches is positive to the whole network. According to setting(i, ii) and the setting (iv, v), the L_{KL} leads to an increase on classification accuracy when branches learn from the fused logits knowledge source, showing the usefulness of the proposed DLF. Specifically, we can see from settings (i, ii) that the L_{KL} further improves the average performance by 2.27% without the SFF. Setting (ii, iii) and setting (v, vi) demonstrate that the diversity loss enhances the collaboration between multiple branches by promoting different branches to explore different knowledge. In detail, the L_{div} brings average gains of 0.28% and 0.40% when the DBFSKD without SFF and with SFF, respectively.

From the results of the experiments performed, we can observe that the DBFSKD obtains greater improvements on each branch after adding the L_{KL}. Since the L_{div} is employed to make branches acquire diversified knowledge, which is not directly related to knowledge transfer. While the L_{KL} is utilized to promote the knowledge transfer from the fused soft target to all branches.

Comparison of different diversity losses: The proposed DBFSKD adopts a simple way to motivate the branches to explore diverse knowledge, i.e., maximize the Euclidean distance similarity between feature maps of adjacent branch. To demonstrate that this approach is simple but effective, we compare it with three other approaches, including cosine similarity (Cosine), intra-channel similarity (IC) [45], and using discriminative networks to identify similarities (MEAL) [46]. Fig. 7 shows the accuracy comparison of these diversity losses on the CIFAR100. All experiments use ResNet18 as the backbone network of DBFSKD. It can be found that in Br-1 and Br-2, the four diversity losses obtain similar accuracies. However, on other branches, the Euclidean distance similarity shows a more significant advantage. So it is reasonable that we adopt this simple way as diversity loss in the DBFSKD.

Ablation experiments of attention module: The branches of the DBFSKD employ an Attention Module (AM) to extract important features from the backbone. To investigate the impact of the attention module on our method, we evaluate the method with and without the attention module. All experimental detail are shown in Table 11. We compare the accuracy of DBFSKD under four different settings.

The fourth row is DBFSKD without the AM, taken as the baseline method. The first row is DBFSKD with the full AM. Adding a full AM achieves an average gain of 1.06% compared to the baseline method. The AM aggregates vital features on channels and spatial, which further improves the classification performance. In addition, we compare the importance of spatial path and channel path in the AM. The channel path simply adopts SENet mode and obtains an average gain of 0.32%. The spatial path is carefully designed to reduce computation by pooling in two dimensions, achieving an average gain of 0.61%, which indicates that the vital local features are more conducive to fine-grained classification.

4.9. Visualization analysis

Feature map visualization: The SFF module is designed to generate feature maps with richer semantics by fusing feature maps from different branches. From all results of previous experiments, the Br-5 has the highest accuracy than others, which supports the effectiveness of the SFF. To visually shows the superior of our method, the raw images and the corresponding attention weights of each branch and the baseline are visualized by Grad-CAM [47]. The Grad-CAM algorithm projects the attention weights of network to the input image space and visualizes them via a heatmap.

As shown in Fig. 8, we randomly select three images from the ImageNet dataset to visualize. The first column shows the original image, the 2–6 columns show the interest area of the branches, and the last column shows the interest area of the baseline. Overall, all branches focus on the foreground while ignoring the background. In detail, from the Br-1 to Br-5, their attention is more inclined to discriminative features. Take the eagle image as an example. The Br-1 puts more attention on the background, while the Br-2 reduces the attention on these spatial positions. The Br-3 and the Br-4 focus more on the eagle body. The Br-5 increases attention to the head and beak of the eagle. Obviously, the area that Br-5 focuses on is conducive to classification, which reveals that our SFF module can generate feature maps with richer semantics. This trend can also be discovered in the triceratops and dog image. From Fig. 8, we can conclude that the Br-1 and Br-2 seem to be more error-prone compared with the baseline, while others are superior. For instance, the baseline pays less attention to the background than the Br-1 and Br-2 but reduces attention to the body of the eagle than the Br-4 and Br-5.

Diversity analysis: In the proposed DBFSKD, we introduce the diversity loss L_{div} to encourage branches to mine different feature representations. The effectiveness of L_{div} is confirmed by the ablation experiments. In this subsection, we further analyze the role of L_{div} by visualizing the intra-channel correlation (ICC) matrix of each branch. Given a feature map $F \in \mathbb{R}^{C\times H\times W}$, we first reshape it to $F_r \in \mathbb{R}^{C\times HW}$. The ICC matrix is calculated as follows

$$M_{ICC} = F_r \times F_r^T \tag{12}$$

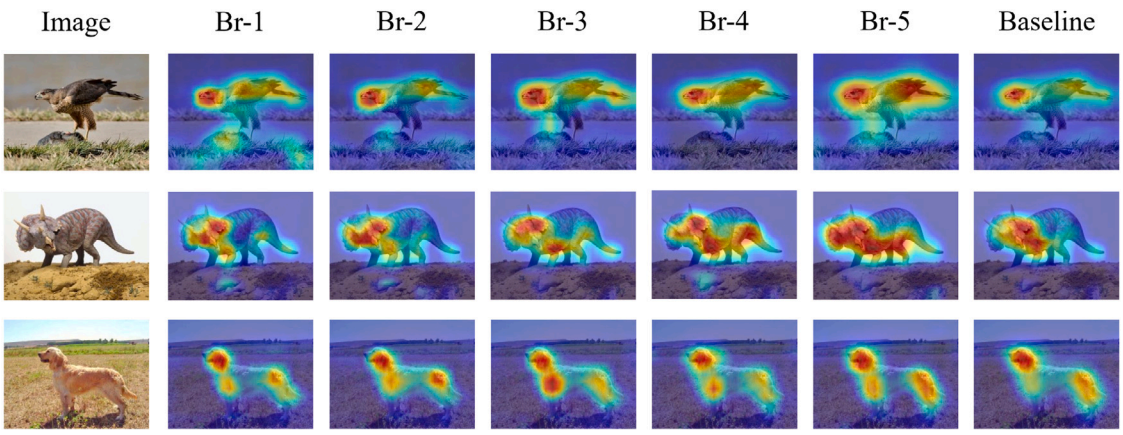

Fig. 8. Visualization comparison of feature map between DBFSKD and baseline. We randomly sample two images from the ImageNet dataset as input samples. The backbone network is ResNet18.

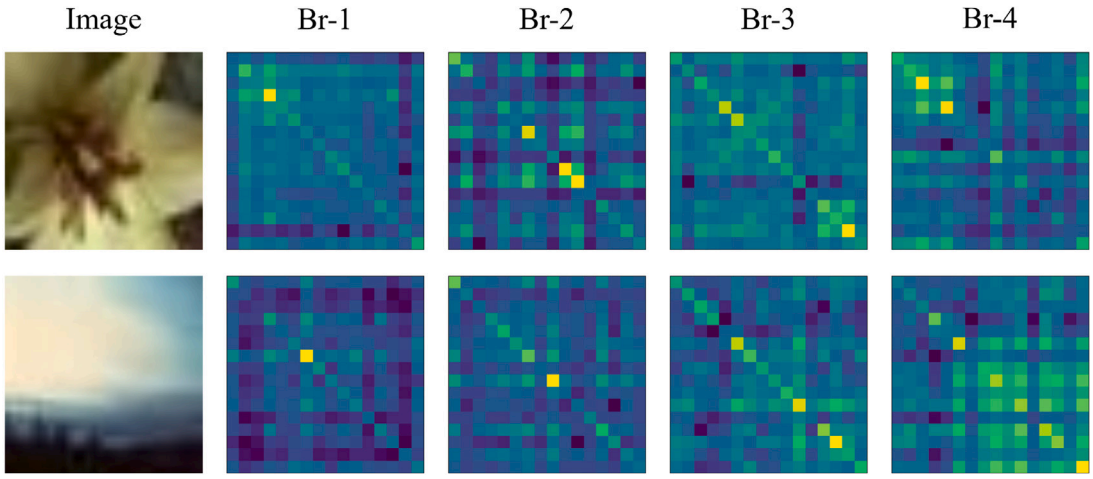

Fig. 9. ICC matrix visualization of the four branches of DBFSKD. We randomly sample two images from the CIFAR100 dataset as input samples. The backbone network of DBFSKD is ResNet18.

The resulting ICC matrix has a size of $C \times C$ regardless of the spatial dimensions H and W. Fig. 9 shows the visualization of the ICC matrix. For randomly selected samples, the intra-channel correlations of the four branches of the DBFSKD have significant differences, which show that the L_{div} has achieved the desired goal of encouraging branches to explore diverse knowledge.

4.10. Limitations

Although the comparison results and the ablation study verify the effectiveness of the proposed method, the limitations of our method are twofold. Our proposed method works well for CNNs-based backbones but is not suitable for the Transformer-based ones. Increasing the number of branches introduces more diversity, distillation and classification losses to be optimized, that may require more computation resources during training compared to conventional KD methods.

5. Conclusion

In this paper, we focus on solving two shortcomings in self-knowledge distillation with multi-branch architecture: how to utilize the knowledge of shallow branches and promote different branches to mine diverse knowledge. We propose a new Diversified Branch Fusion approach for Self-Knowledge Distillation (DBFSKD). The proposed method adds well-designed lightweight networks at middle layers of the backbone to form multiple branches. The DBFSKD facilitates branches to mine diverse knowledge by introducing the diversity loss. On this basis, the selective feature fusion module is designed to fuse the feature maps from different branches. To further utilize the logits of each branch, we propose the dynamic logits fusion module for obtaining a more robust soft target. And then, we apply the fused soft target as the prediction of the teacher to supervise other branches. Extensive experiments are conducted on four public datasets to verify the effectiveness of the DBFSKD and demonstrate the superiorities of reducing parameters and MACs without compromising the performance, which is vital for practical scenarios with limited resources. Furthermore, the DBFSKD provides multiple networks with various capabilities for practical applications.

CRediT authorship contribution statement

Zuxiang Long: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Fuyan Ma:** Writing – original draft, Writing – review & editing, Visualization, Software, Investigation. **Bin Sun:** Data curation, Resources, Project administration. **Mingkui Tan:** Data curation, Validation. **Shutao Li:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the National Key&D Program of China (2018YFB1305200), the National Natural Science Fund of China (62171183) and the Hunan Provincial Natural Science Foundation of China (2022JJ20017).

References

[1] Z. Feng, J. Lai, X. Xie, Resolution-aware knowledge distillation for efficient inference, IEEE Trans. Image Process. 30 (2021) 6985–6996, http://dx.doi.org/10.1109/TIP.2021.3101158.
[2] K. Zhang, C. Zhanga, S. Li, D. Zeng, S. Ge, Student network learning via evolutionary knowledge distillation, IEEE Trans. Circuits Syst. Video Technol. 32 (2021) 2251–2263, http://dx.doi.org/10.1109/TCSVT.2021.3090902.
[3] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint arXiv:1503.02531.
[4] B. Heo, M. Lee, S. Yun, J.Y. Choi, Knowledge transfer via distillation of activation boundaries formed by hidden neurons, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3779–3787.
[5] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, C. Chen, Cross-layer distillation with semantic calibration, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 7028–7036.
[6] Y. Zhang, T. Xiang, T.M. Hospedales, H. Lu, Deep mutual learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4320–4328.
[7] A. Yao, D. Sun, Knowledge transfer via dense cross-layer mutual-distillation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 294–311.
[8] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, P. Luo, Online knowledge distillation via collaborative learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11020–11029.
[9] L. Zhang, C. Bao, K. Ma, Self-distillation: Towards efficient and compact neural networks, IEEE Trans. Pattern Anal. Mach. Intell. (2021) http://dx.doi.org/10.1109/TPAMI.2021.3067100.
[10] 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: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 3713–3722.
[11] M. Phuong, C.H. Lampert, Distillation-based training for multi-exit architectures, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1355–1364.
[12] X. Wang, Y. Li, Harmonized dense knowledge distillation training for multi-exit architectures, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 10218–10226.
[13] D. Chen, J.-P. Mei, C. Wang, Y. Feng, C. Chen, Online knowledge distillation with diverse peers, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 3430–3437.
[14] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.
[15] S. Feng, H. Chen, X. Ren, Z. Ding, K. Li, X. Sun, Collaborative group learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 7431–7438.
[16] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, 2014, arXiv preprint arXiv:1412.6550.
[17] Y. Guan, P. Zhao, B. Wang, Y. Zhang, C. Yao, K. Bian, J. Tang, Differentiable feature aggregation search for knowledge distillation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 469–484.
[18] B. Heo, M. Lee, S. Yun, J.Y. Choi, Knowledge transfer via distillation of activation boundaries formed by hidden neurons, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3779–3787.
[19] W. He, X. Wang, L. Wang, Y. Huang, Z. Yang, X. Yao, X. Zhao, L. Ju, L. Wu, L. Wu, et al., Incremental learning for exudate and hemorrhage segmentation on fundus images, Inf. Fusion 73 (2021) 157–164.
[20] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, N. Díaz-Rodríguez, Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges, Inf. Fusion 58 (2020) 52–68.
[21] I. Chung, S. Park, J. Kim, N. Kwak, Feature-map-level online adversarial knowledge distillation, in: Proceedings of the International Conference on Machine Learning (ICML), 2020, pp. 2006–2015.
[22] J. Kim, M. Hyun, I. Chung, N. Kwak, Feature fusion for online mutual knowledge distillation, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2021, pp. 4619–4625.
[23] Y. Hou, Z. Ma, C. Liu, C.C. Loy, Learning lightweight lane detection cnns by self attention distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1013–1021.
[24] S. Yun, J. Park, K. Lee, J. Shin, Regularizing class-wise predictions via self-knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13876–13885.
[25] M. Ji, S. Shin, S. Hwang, G. Park, I.-C. Moon, Refine myself by teaching myself: Feature refinement via self-knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10664–10673.
[26] Z. Li, Y. Huang, D. Chen, T. Luo, N. Cai, Z. Pan, Online knowledge distillation via multi-branch diversity enhancement, in: Proceedings of the Asian Conference on Computer Vision (ACCV), 2020.
[27] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.
[28] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.
[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.
[30] E.D. Cubuk, B. Zoph, D. Manšš, V. Vasudevan, Q.V. Le, AutoAugment: Learning augmentation strategies from data, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 113–123.
[31] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3967–3976.
[32] F. Tung, G. Mori, Similarity-preserving knowledge distillation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1365–1374.
[33] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, J.Y. Choi, A comprehensive overhaul of feature distillation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1921–1930.
[34] G. Song, W. Chai, Collaborative learning for deep neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[35] X. Zhu, S. Gong, et al., Knowledge distillation by on-the-fly native ensemble, Adv. Neural Inf. Process. Syst. 31 (2018).
[36] S. Li, W. Deng, Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition, IEEE Trans. Image Process. 28 (1) (2019) 356–370, http://dx.doi.org/10.1109/TIP.2018.2868382.
[37] S. Zhao, H. Cai, H. Liu, J. Zhang, S. Chen, Feature selection mechanism in CNNs for facial expression recognition, in: BMVC, 2018, p. 317.
[38] Y. Li, J. Zeng, S. Shan, X. Chen, Occlusion aware facial expression recognition using CNN with attention mechanism, IEEE Trans. Image Process. 28 (5) (2019) 2439–2450, http://dx.doi.org/10.1109/TIP.2018.2886767.
[39] K. Wang, X. Peng, J. Yang, D. Meng, Y. Qiao, Region attention networks for pose and occlusion robust facial expression recognition, IEEE Trans. Image Process. 29 (2020) 4057–4069, http://dx.doi.org/10.1109/TIP.2019.2956143.
[40] K. Wang, X. Peng, J. Yang, S. Lu, Y. Qiao, Suppressing uncertainties for large-scale facial expression recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6897–6906.
[41] Y. Fan, V. Li, J.C. Lam, Facial expression recognition with deeply-supervised attention network, IEEE Trans. Affect. Comput. (2020) http://dx.doi.org/10.1109/TAFFC.2020.2988264.
[42] Y. Li, G. Lu, J. Li, Z. Zhang, D. Zhang, Facial expression recognition in the wild using multi-level features and attention mechanisms, IEEE Trans. Affect. Comput. (2020) http://dx.doi.org/10.1109/TAFFC.2020.3031602.
[43] H. Li, N. Wang, X. Ding, X. Yang, X. Gao, Adaptively learning facial expression representation via C-F labels and distillation, IEEE Trans. Image Process. 30 (2021) 2016–2028, http://dx.doi.org/10.1109/TIP.2021.3049955.
[44] F. Ma, B. Sun, S. Li, Facial expression recognition with visual transformers and attentional selective fusion, IEEE Trans. Affect. Comput. (2021) http://dx.doi.org/10.1109/TAFFC.2021.3122146.
[45] L. Liu, Q. Huang, S. Lin, H. Xie, B. Wang, X. Chang, X. Liang, Exploring inter-channel correlation for diversity-preserved knowledge distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 8271–8280.
[46] Z. Shen, Z. He, X. Xue, Meal: Multi-model ensemble via adversarial learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4886–4893.
[47] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.