This ACCV 2020 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2020
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Online Knowledge Distillation via Multi-branch Diversity Enhancement

Zheng Li[1,2], Ying Huang[1,3], Defang Chen[4],
Tianren Luo[1], Ning Cai[1], and Zhigeng Pan[1⋆]

[1] Virtual Reality and Intelligent Systems Research Institute,
Hangzhou Normal University
[2] School of Information Science & Engineering, Hangzhou Normal University
[3] Alibaba business school, Hangzhou Normal University
{lizheng1, caining, luotianren}@stu.hznu.edu.cn,
{yw52, zgpan}@hznu.edu.cn,
[4] College of Computer Science, Zhejiang University
defchern@zju.edu.cn

Abstract. Knowledge distillation is an effective method to transfer the knowledge from the cumbersome teacher model to the lightweight student model. Online knowledge distillation uses the ensemble prediction results of multiple student models as soft targets to train each student model. However, the homogenization problem will lead to difficulty in further improving model performance. In this work, we propose a new distillation method to enhance the diversity among multiple student models. We introduce **F**eature **F**usion **M**odule (**FFM**), which improves the performance of the attention mechanism in the network by integrating rich semantic information contained in the last block of multiple student models. Furthermore, we use the **C**lassifier **D**iversification(**CD**) loss function to strengthen the differences between the student models and deliver a better ensemble result. Extensive experiments proved that our method significantly enhances the diversity among student models and brings better distillation performance. We evaluate our method on three image classification datasets: CIFAR-10/100 and CINIC-10. The results show that our method achieves state-of-the-art performance on these datasets.

1 Introduction

Knowledge distillation[1], as one of the key methods in model compression, the distillation process usually starts by training a high-capacity teacher model. A student model will actively learn the soft label or feature representation[11] generated by teacher model. The purpose of distillation is to train a more compact and accurate student model through the knowledge transferred from the teacher network. In recent years, the convolutional neural network has made very impressive achievements in many vision tasks[2–6]. But it requires high cost of

⋆ Corresponding author

computation and memory in inference, making the deployment of CNN difficult in resource-limited mobile devices. Knowledge distillation was proposed to solve these problems. In the meantime, other types of model compression techniques such as network pruning[7–9] and network quantization[10–12] have also been proposed.

Traditional knowledge distillation[13–15] is a two-stage process. We should first train a teacher model, then get a student model by distilling the teacher model. Although this approach can obtain a higher quality student model by aligning the predictions of the teacher model, it is still a complex approach that requires more computational resources. Online knowledge distillation[16] successfully simplifies the training process by reducing the need for pretrained teacher model. Existing online knowledge distillation methods[17–19] learns not only from the ground truth labels but also from the ensemble results of multiple branches. We refer to each branch as a separate student model. This method can improve the performance of models with arbitrary capacity and obtain better generalization ability.

Averaging the predictions of each branch is a very simple way to get the ensemble results. This approach tend to cause branches to quickly homogenize, hurting the distillation performance[20, 21]. However, [17, 19] found that the accuracy of the final result improves if different weights were applied to each peer. In OKDDip[19], this paper introduces the concept of two-level distillation method, builds diverse peers by applying a self-attention mechanism[22]. Self-attention in OKDDip needs two fully connected layers separately as transformation matrices to obtain importance scores, which increases the complexity of time and space. In ONE[17], the gate module uses features from the second block of its backbone network as input to generate the importance score of the corresponding branch. However, this feature contains little semantic information which leads to limited improvement in image classification tasks.

In this work, we propose a new distillation strategy to enhance the diversity among branches which can significantly improve the effectiveness of online knowledge distillation. By introducing Feature Fusion Module(FFM) to fuse the features of the last layer of multiple branches, we make full use of the diversity of semantic information contained in multiple branches to improve the attention performance[23]. Since a large diversity of branches can help ensemble-based online KD methods achieve better results, inspired by [24], we propose the CD loss to prevent homogeneity between branches by explicitly forcing their features to be learned orthogonally. This loss function serves as a regularization term to prevent group performance degradation caused by homogenization. Unlike other methods in which all branches converge into similar one. By using our method, each branch keeps their uniqueness. Based on [19], a two-level knowledge distillation framework is adopted. We build a network with m branches, including m-1 auxiliary branches and a group leader. The knowledge generated by these diverse peers will be distilled into the group leader, and the remaining peers will be discarded. In order to reduce the consumption of computing resources, we only keep the group leader as the final deployment model.

Our contributions of this work can be summarized as follows:

- We propose Feature Fusion Module(FFM) which can better fuse diverse semantic information from multiple branches and improves the performance of the attention mechanism.
- We introduce the Classifier Diversification(CD) loss function. As a regularization term, it effectively reduces the homogenization among branches, improves the accuracy of ensemble results and leads to a better student model.
- The extensive experiments and analysis verify that our proposed method can effectively enhance branch diversity and train better student models on different image classification datasets: CIFAR-10/100[25] and CINIC-10[26].

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation[1] has been widely used in many scenarios involving deep learning algorithms, such as virtual experiments in VR, autonomous driving and so on. It provides an useful method that allows the complex teacher model to be compressed to a more lightweight student model by aligning the student model with the teacher model. When training the target model, this method takes advantage of the extra supervisory signal provided by the soft output of the teacher model. there are also many works[13–15, 27] made explorations based on this idea. In FitNets[13], the student model attempts to mimic the intermediate representation directly from the teacher network. Attention Transfer[14] transfers an attention map of a teacher model into a student and [28] proposes a similar method using mean weights. In flow-based knowledge distillation[15], the student is encouraged to mimic the teacher's flow matrices, which are derived from the inner product between feature maps in two layers. [29] saves the computation by using singular value decomposition to compress feature maps.

There are also innovative works exploring alternatives to the usual student-teacher training paradigm. Generative Adversarial Learning[30] is proposed to generate realistic-looking images from random noise using neural networks. The ideas in the adversarial network are applied to knowledge distillation[31–33]. In MEAL[31], the generators were employed to synthesize student features and the discriminator was used to discriminate between teacher and student outputs for the same image. In [33], this work adopts adversarial method to discover adversarial samples supporting decision boundary. With the supervision of discriminator, student can better mimic the behavior of teacher model. In addtion, many works[34–37] have also explored the relationship between the samples. [34] propose that similar input pairs in the teacher network tends to produce similar activations in the student network. A few recent papers[37–39] have shown that models of the same architecture can also be distilled. Snapshot distillation[39] uses the cyclic learning rate policy, in which the last snapshot of each cycle is used as the teacher for all iterations in the next cycle, and the teacher is used to provide supervision signal.

2.2 Online Knowledge Distillation

Traditional knowledge distillation methods have two stages that require a pretrained teacher model to provide soft output for distillation. Different from above complex training methods, several works adopts collaboratively training strategy. Simultaneously training a group of student models based on each other's predictions is an effective single-stage distillation method, which can be a good substitute for pretrained teacher models. Some methods[16, 18] solve this problem. The online knowledge distillation was completed through mutual instruction between two peers[16]. However, the lack of a high-capacity teacher model will decrease the distillation efficiency. In [17, 40], each student model learns from the average of the predictions generated by a group of students and obtains a better teacher model effect. ONE found that simply averaging the results would reduce the diversity among students, affecting the training of branch-based models. ONE generates the importance score corresponding to each student through the gate module. By assigning different importance score to each branch, a high-capacity teacher model is constructed, which can leverage knowledge from training data more effectively. OKDDip[19] proposed the concept of two-level distillation. The ensemble results of auxiliary peer networks were distilled into the group leader. The diversified peer network plays a key role in improving distillation performance.

3 Online Knowledge Distillation via Multi-branch Diversity Enhancement

The architecture of our proposed method is illustrated in Fig. 1. Our method is based on a two-level distillation procedure. The network has $m-1$ auxiliary branches and one group leader. In the first level distillation, each branch learns not only from the ground truth label but also from the weighted ensemble targets obtained through Feature Fusion Module. These results play the role of a teacher model to teach each branch. In the second level distillation, the knowledge learned by the group is further distilled into the group leader. To save computing resources, we use the group leader for the final deployment.

3.1 Formulation

In knowledge distillation, the student uses the output of the teacher as an additional supervisory signal for network training. Given a dataset of N training samples $D = \{(x_i, y_i)\}_i^N$, where $y_i \in \{1, 2, ..., C\}$. Here, x_i is the i^{th} training sample, y_i is the corresponding ground truth label and C is the total number of image classes. Take the training sample as the input of the teacher network, we will get the output logits $t_i = (t_i^1, ..., t_i^c)$. The logits vector after softmax will get the i^{th} probability value q_i^j,

$$q_i^j = \frac{exp(t_i/T)}{\sum_{j=1}^{C} exp(t_i^j/T)} \tag{1}$$

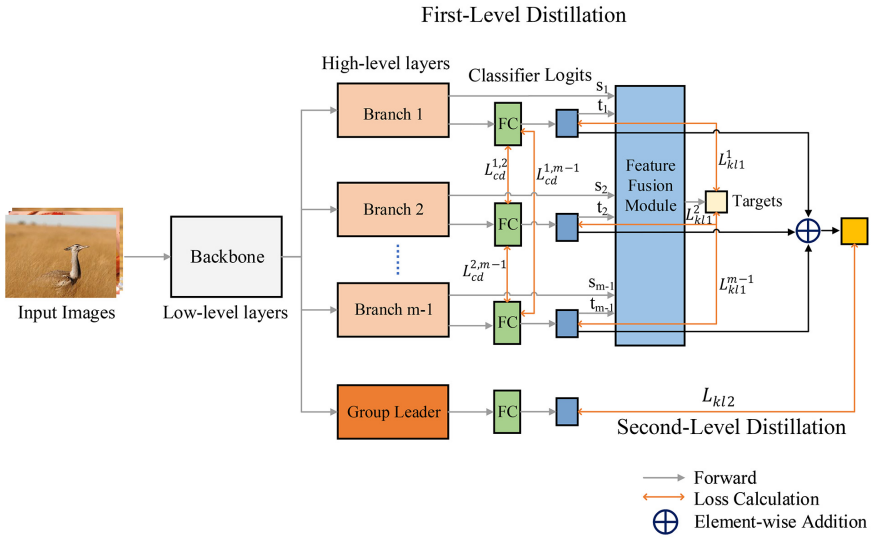

Fig. 1. The overall framework of our proposed method. Each branch and shared low-level layers constitute an individual student model. This is a two-level distillation process. For the first-level distillation, each auxiliary branch learns from their ensemble targets. The second-level distillation transfers the knowledge learned by the group to the group leader. L_{cd} denotes the proposed classifier diversification loss. L_{kl} denotes the KL divergence loss. We omit the cross entropy loss L_{ce} for simplicity. We will introduce these loss functions in detail in the third section. Best viewed in color.

where T is the temperature parameter. An increase in the parameter T will make the probability distribution smoother. When training teachers, T is set to 1. When distilling knowledge from the teacher model to the student model, T is usually set to 3.

In order to train a multi-class image classification model, our goal is to minimize the cross entropy between the predicted class probabilities q_i and the corresponding ground truth label distribution y_i,

$$L_{ce} = H(y_i, q_i) \tag{2}$$

where $H(p, q) = -\sum_i p_i log q_i$.

Knowledge transfer is achieved by aligning the probability distribution q generated by the student with the target distribution t. The temperature parameter T should be the same for teacher and student networks. Specifically, we use KL(Kullback-Leibler) Divergence as the loss function:

$$L_{kl} = KL(t, q) = \sum_{i,j} t_{ij} log \frac{t_{ij}}{q_{ij}} \tag{3}$$

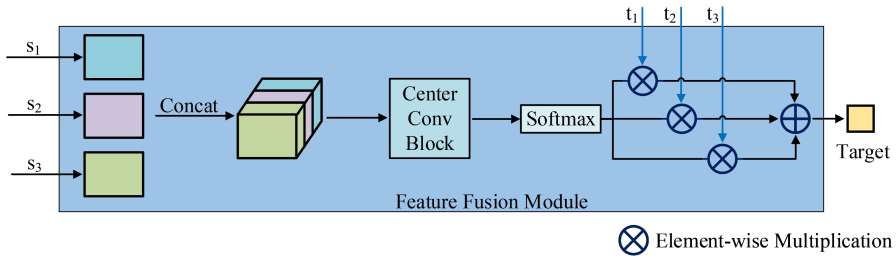

Fig. 2. We take the case of three auxiliary branches as an example. Feature maps s_i from each branch will be concatenated together, and then fed into the center convolution block. The center block is made of several convolutional layers, batch normalization and ReLU activation function. The last layer of this block is fully connected layer. This block is designed to fuse the semantic representation from multiple branches. Compared with other methods, more semantic information can effectively improve the performance of the module. The final target is obtained by the weighted sum of logits t_i of all auxiliary branches.

3.2 Feature Fusion Module

An overview of the Feature Fusion Module is described in Fig. 2. Features from a single layer contain less information than features from multiple layers. Many approaches[41–44] try to take advantage of more diversed features to get better model performance. We take the features of the last block from multiple branches as the input of the Feature Fusion Module. Since deeper layers in the network lead to richer semantic information, this approach can enrich features with high-level semantic information. Our experiment proves that the weights generated from this method can achieve better results.

$$t_e = \sum_{i=0}^{m-1} f_i(s_1, s_2, ..., s_{m-1}) \cdot t_i \tag{4}$$

where $f(\cdot)$ denotes the function of center block in the FFM. This function will output the corresponding importance score for each branch and also satisfy $\sum_{i=1}^{m} f_i(s_1, s_2, ..., s_{m-1}) = 1$. s_k denotes the feature map of the last block from the $(m-1)^{th}$ branch. t_k denotes the logits from the m^{th} branch. t_e denotes the weighted ensemble target.

3.3 Classifier Diversification Loss

The diversity has an important influence on the accuracy of the final ensemble results. For better results, we expect peer classifiers to classify samples based on different viewpoints. So we restrict the weight of classifiers, force them to be diversed. We use

Algorithm 1 Online knowledge distillation via multi-branch diversity enhancement

Input: Training dataset D ; Training Epoch Number ϵ ; Branch Number β
Output: Trained group leader model θ^0 and auxiliary models $\{\theta^i\}_{i=1}^m$
Initialize: e=1; Randomly initialize $\{\theta^i\}_{i=0}^m$

while e $\leq \epsilon$ **do**
    Compute the predictions of all auxiliary branches $\{\theta^i\}_{i=1}^m$ with Eq. (1);
    Get each branch's weight through FFM;
    Compute the target logits with Eq. (4);
    Compute the CD loss L_{cd} with Eq. (5);
    Compute the distillation loss L_{kl1} and L_{kl2} with Eq. (3);
    Compute the total loss function with Eq. (8);
    Update the model parameters $\{\theta^i\}_{i=0}^m$
    e=e+1
end while
Model eployment: Use group leader θ^0;

$$L_{cd} = \sum_{i=0}^{m-1} \sum_{j=i+1}^{m} L_{cd}^{i,j} = \sum_{i=0}^{m-1} \sum_{j=i+1}^{m} |W_i^T W_j| \tag{5}$$

where W_i is the fully connected layers' weights of peer classifiers. If the weights of fully connected layers between peers get similar, it means there are more homogenization among them. This loss function acts as a regularization term to prevent homogenization. This will force each classifier to learn different features under this limit. Experiments show that this loss function improves the diversity of peer classifiers and improves the distillation efficiency. We will explain in detail in the ablation study.

3.4 Loss function and algorithm

To get a better understanding of our method, we describe the process in Algorithm 1. Our distillation method is a two-level procedure. For the first level distillation, each auxiliary branch learns the knowledge distilled from the soft targets t_e generated by FFM. The distillation loss of all auxiliary branches is

$$L_{kl1} = \sum_{i=1}^{m-1} KL(t_e, q_i) \tag{6}$$

In the second-level distillation, the knowledge learned by the group will be distilled to the group leader. Same as OKDDip, we average the predictions of all branches to get t_{avg}. The distillation of the group leader is

$$L_{kl2} = KL(t_{avg}, q_{gl}) \tag{7}$$

To sum up, the loss function of the whole neural network is:

$$L = \sum_{i=1}^{m} L_{ce}^{i} + \alpha T^2 L_{kl1} + \beta T^2 L_{kl2} + \gamma L_{cd} \tag{8}$$

where α, β and γ are the balance parameter to balance the loss term. The first term is the sum of all branches' cross entropy loss.

4 Experiment

In this section, we evaluate our method on five popular neural networks (ResNet-50, ResNet-110[4], ResNext-50(32x4d)[45], Xception[46], ShuffleNet V2-1.0[47]) and three image classification benchmark dataset: CIFAR-10/100[25] and CINIC-10[26]. We also compare our method with closely related works, including ONE and OKDDip. In addition to the classification ability, we also conduct several ablation studies on the feature fusion module and classifier diversification loss, of which the result indicates that the proposed method has better generalization performance compared with other methods. All the reported results are averaged based on three runs.

4.1 Datasets and Settings

Datasets. There are three datasets in our experiments. CIFAR-10 and CIFAR-100[25] both contains 50,000 training images and 10,000 test images, which come from 10/100 classes. CINIC-10 consists of images from both CIFAR and ImageNet[48]. It has 270,000 images and 10 classes. The size in CINIC-10 is the same as in CIFAR. It contains 90,000 training images and 90,000 test images, all at a resolution of 32 x 32. The top-1 classification error rate are reported.

Settings. We implement all the networks and training procedures in Pytorch[49]. We conduct all experiments on an NVIDIA GeFore RTX 2080Ti GPU. For all datasets, we follow the experimental setting of [19]. For data augmentation, we apply standard random crop and horizontal flip to all images. We use SGD[50] as the optimizer with Nesterov momentum 0.9 and weight decay $5e-4$ during training. We set mini-batch size to 128. We use the standard learning schedule. The learning rate starts from 0.1 and divided by 10 at 150 and 225 iterations, for a total of 300 iterations. We set m=4, means that there are three auxiliary branches and a group leader. We separate the last two blocks of each backbone network for CIFAR-10/100 and CINIC-10. We empirically set T=3 to generate soft predictions. We set α=1, β=2 and γ=$5e-8$ to balance the loss term in Equation 6.

We compare our method with several online knowledge distillation methods. In OKDDip, it has two network settings: branch-based and network-based. The branch-based approach refers to student models sharing multiple convolutional layers, separated from each other after a specified layer. The network-based method means that all student models do not share any convolutional layers, and each student is an independent model. The principles of these two

Table 1. Error Rate(Top-1, %) on CIFAR-10.

Models	Baseline	Our Method	Gain
ResNet-32	6.38 ± 0.10	**5.45 ± 0.07**	0.93
ResNet-110	5.46 ± 0.02	**4.47 ± 0.02**	0.99
ResNext-50(32x4d)	5.05 ± 0.12	**4.66 ± 0.05**	0.39
Xception	5.70 ± 0.08	**5.19 ± 0.05**	0.51
ShuffleNetV2-1.0	9.21 ± 0.04	**8.36 ± 0.03**	0.85

Table 2. Error Rate(Top-1, %) on CIFAR-100.

Models	Baseline	ONE	OKDDip	Our Method
ResNet-32	28.39 ± 0.04	25.76 ± 0.04	25.45 ± 0.10	**24.84 ± 0.06**
ResNet-110	23.85 ± 0.17	21.94 ± 0.13	21.01 ± 0.16	**20.52 ± 0.13**
ResNext-50(32x4d)	20.43 ± 0.19	18.24 ± 0.03	17.90 ± 0.06	**17.55 ± 0.06**
Xception	21.71 ± 0.06	19.69 ± 0.06	19.66 ± 0.07	**19.55 ± 0.11**
ShuffleNetV2-1.0	28.76 ± 0.12	25.23 ± 0.11	25.28 ± 0.18	**25.17 ± 0.10**

approaches are close, so the branch-based method can well validate the effectiveness of our method. In all the experiments, we use branch-based setting for comparison. Baseline means the original model trained on the dataset without any modification.

4.2 Results on CIFAR-10/100

Table 1 and Table 2 compares the top-1 classification error rate on CIFAR-10 and CIFAR-100 based on five different backbone networks. The result generated by ONE is the averaged accuracy of all branches. The results of OKDDip and ours are the accuracy of the group leader. From these two tables, it clearly shows that our method achieves a lower error rate on the same backbone network. Specifically, our method improves the accuracy of various baseline network by 3% to 4% on CIFAR-100. The network with higher capacity generally benefits more from our method. Our methods improves the state-of-the-art methods by 0.61%, 0.49% and 0.35% with ResNet-32, ResNet-110 and ResNext-50, respectively. These results showing that our method is more effective than existing methods. When the baseline model has lower capacity, our method can also slightly improve the accuracy compared with other methods.

In Table 3, we compare our method with another two-level distillation method OKDDip on three backbone networks. The results of compared methods are the averaged ensemble results of three branches on three backbone networks in the second-level distillation. Since the ensemble results act as a teacher to teach the group leader, a more accurate result can train a better group leader. It is also seen that our method improves the OKDDip method by 0.59%, 0.57% and 0.34%

Table 3. Error Rate(Top-1, %) of ensemble results on CIFAR-100.

Models	OKDDip	Our Method	Gain
ResNet-32	23.22	**22.63**	0.59
ResNet-110	19.42	**18.85**	0.57
ResNext-50(32x4d)	17.02	**16.68**	0.34

with ResNet-32, ResNet-110 and ResNext-50. Generally, our method successfully enhanced the diversity among different branches and brings improvement to distillation performance.

Diversity Measurement. We use the interrater agreement in [21] as the metric to measure the branch diversity. This method is defined as:

$$s = 1 - \frac{\frac{1}{T}\sum_{k=1}^{m}\rho(x_k)(T-\rho(x_k))}{m(T-1)\bar{p}(1-\bar{p})} \tag{9}$$

where T is the total number of classifiers, $\rho(x_k)$ is the number of classifiers that classify x correctly, $\bar{p}$ is the average accuracy of individual classifiers and m is the total number of test samples. OKDDip and our method obtained 0.633 and 0.549 respectively (CIFAR-100 & ResNet-32). The smaller the s measurement, the larger the diversity. From this results, we can see that our method actually increase the branch diversity.

4.3 Results on CINIC-10

CINIC-10 dataset is larger and more challenging than CIFAR-10 but not as difficult as ImageNet. We adopt the same data preprocessing as those of CIFAR-10/100 experiments.

Table 4. Error Rates(Top-1, %) on CINIC-10.

Models	Baseline	ONE	OKDDip	Our Method
ResNet-32	15.96 ± 0.13	14.60 ± 0.09	14.41 ± 0.10	**14.28 ± 0.12**
ResNet-110	13.99 ± 0.06	12.29 ± 0.09	12.21 ± 0.11	**11.86 ± 0.08**
ResNext-50(32x4d)	13.65 ± 0.12	12.19 ± 0.04	12.20 ± 0.06	**12.02 ± 0.07**

Table 4 compares the top-1 classification error rates based on three backbone networks trained by different methods. From this table, we observed that our method outperforms baseline by 1.68%, 2.13% and 1.63% on ResNet-32, ResNet-110 and ResNext-50 respectively. Our method also improves the state-of-the-art method by 0.13%, 0.35% and 0.18% on three backbone networks. We can

Table 5. Error Rates(Top-1, %) of ensemble results on CINIC-10.

Models	OKDDip	Our Method	Gain
ResNet-32	13.55	**13.44**	0.11
ResNet-110	11.35	**10.98**	0.37
ResNext-50(32x4d)	11.77	**11.54**	0.23

find that the improvement in generalization performance is very limited on this dataset. High-capacity networks tend to perform better. But the accuracy of ResNext-50 is slightly lower than ResNet-110 although its baseline performance is better.

In Table 5, we compare our method with OKDDip. We can find that our method outperforms OKDDip by 0.11%, 0.37% and 0.23% on ResNet-32, ResNet-110 and ResNext-50. While it can be observed that all the methods seem not to increase as much as that in CIFAR-100 experiments. We guess it is because the homogenization problem becomes serious when we conduct experiments on easier datasets. We still need to explore solutions to solve the homogenization problem in the future.

4.4 Ablation Study

Table 6. Ablation Study: Error rates(Top-1, %) for ResNet-32 on CIFAR-100.

Gate	SA	FFM	CD	Top-1 error	Top-5 error
		✓		25.40	6.19
	✓			25.45	6.33
✓				25.76	6.39
		✓	✓	**24.84**	**6.08**
	✓		✓	25.18	6.10
✓			✓	25.31	6.11

In this section, we conduct various ablation studies to validate the effectiveness of our proposed FFM and CD loss. We use ResNet-32 on the CIFAR-100 dataset to show the benefit of our components. We also compare our FFM with other knowledge distillation methods, including gate module in ONE and self-attention(SA) mechanism in OKDDip.

In Table 6, we report the top-1 and top-5 error rates of different methods. The remaining experimental settings are consistent with previous experiment. We carefully conducted six experiments on the network components. We compared the performance of three attention modules in the same experimental

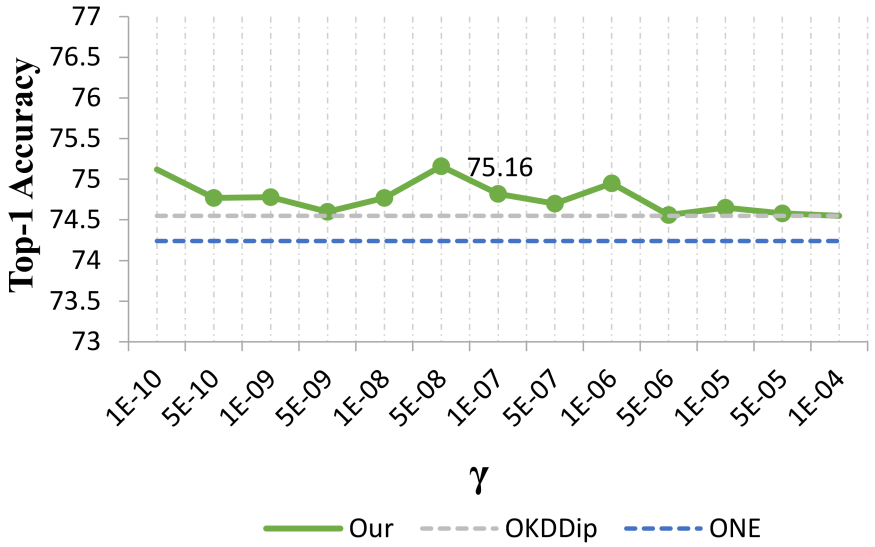

Fig. 3. Sensitivity to γ on CIFAR-100 for ResNet-32.

settings. When FFM is used only, the performance of our method has slightly exceeded other methods. This shows that FFM makes the student network learns more knowledge during the distillation. Compared with gate module in ONE, our method improves the top-1 error rates by 0.36% and top-5 error rates by 0.2%. This result proves that our method effectively utilizes the rich semantic information of multiple branches. When we combine different attention mechanism with classifier diversification loss, our results clearly show that our method surpasses other methods. The combination of FFM and CD loss has more obvious improvement. Compared with the independent FFM, the combination improves the top-1 error rates by 0.56% and the top-5 error rates by 0.08%. Our method clearly enhances the diversity among branches and improves the generalization ability of the student model. From this table, we observe that CD loss really plays the most important role in the overall improvements.

Fig. 3 demonstrates how the performance of our method is affected by the choice of hyperparameter γ of the CD loss. We plot the top-1 accuracy on the CIFAR-100 for ResNet-32 group leader trained with γ ranging from $1e-10$ to $1e-4$. In this figure, the dash line indicates the mean accuracy of other methods. We can find that our method still has robust performance against varying γ values. The green dot indicates the parameter we are using. We should note that the choice of parameters will affect the optimization process. If the parameter is too large, this will lead to too much diversity among the branches, and eventually will not converge. If the parameter is too small, the CD loss function will be difficult to play the role of regularization. In that case, the value of this loss function will be very small, making the loss function ineffective. This figure

shows that CD loss has a significant effect on distillation performance within a proper range.

5 Conclusion

In online knowledge distillation, diversity is always an important and challenging issue. In this work, we proposed the Feature Fusion Module and Classifier Diversification loss, which effectively enhances the diversity among multiple branches. By increasing branch diversity and using more diversed semantic information, we have significantly improved the performance of online knowledge distillation. Experiments show that our method achieves the state-of-the-art performance among several popular datasets without additional training and inference costs.

Acknowledgement

This work is supported by National Key Research and Development Project of China (Grant No. 2018YFB1004901).

References

1. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012) 1097–1105
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 770–778
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
6. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 779–788
7. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
8. Lebedev, V., Lempitsky, V.: Fast convnets using group-wise brain damage. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2554–2564
9. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440 (2016)

10. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision, Springer (2016) 525–542
11. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4820–4828
12. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 8612–8620
13. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
14. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
15. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4133–4141
16. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4320–4328
17. Zhu, X., Gong, S., et al.: Knowledge distillation by on-the-fly native ensemble. In: Advances in Neural Information Processing Systems. (2018) 7517–7527
18. Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G.E., Hinton, G.E.: Large scale distributed neural network training through online distillation. arXiv preprint arXiv:1804.03235 (2018)
19. Chen, D., Mei, J.P., Wang, C., Feng, Y., Chen, C.: Online knowledge distillation with diverse peers. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2020) 3430–3437
20. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine learning **51** (2003) 181–207
21. Zhou, Z.H.: Ensemble methods: foundations and algorithms. CRC press (2012)
22. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning. (2019) 7354–7363
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. (2017) 5998–6008
24. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: International Conference on Machine Learning. Volume 70. (2017) 2988–2997
25. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical Report (2009)
26. Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: Cinic-10 is not imagenet or cifar-10. arXiv preprint arXiv:1810.03505 (2018)
27. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in Neural Information Processing Systems. (2014) 2654–2662
28. Yun, S., Park, J., Lee, K., Shin, J.: Regularizing class-wise predictions via self-knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020) 13876–13885

29. Lee, S.H., Kim, D.H., Song, B.C.: Self-supervised knowledge distillation using singular value decomposition. In: European Conference on Computer Vision, Springer (2018) 339–354
30. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. (2014) 2672–2680
31. Shen, Z., He, Z., Xue, X.: Meal: Multi-model ensemble via adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 4886–4893
32. Xu, Z., Hsu, Y.C., Huang, J.: Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. arXiv preprint arXiv:1709.00513 (2017)
33. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge distillation with adversarial samples supporting decision boundary. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 3771–3778
34. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3967–3976
35. Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 5007–5016
36. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems. (2017) 1195–1204
37. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020) 10687–10698
38. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. arXiv preprint arXiv:1805.04770 (2018)
39. Yang, C., Xie, L., Su, C., Yuille, A.L.: Snapshot distillation: Teacher-student optimization in one generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 2859–2868
40. Song, G., Chai, W.: Collaborative learning for deep neural networks. In: Advances in Neural Information Processing Systems. (2018) 1832–1841
41. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4700–4708
42. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
43. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9522–9531
44. Sun, D., Yao, A., Zhou, A., Zhao, H.: Deeply-supervised knowledge synergy. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 6997–7006
45. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1492–1500

46. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1251–1258
47. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: European Conference on Computer Vision. (2018) 116–131
48. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, Ieee (2009) 248–255
49. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. (2019) 8026–8037
50. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)