Make Baseline Model Stronger: Embedded Knowledge Distillation in Weight-Sharing Based Ensemble Network

Shuchang Lyu
lyushuchang@buaa.edu.cn
Qi Zhao
zhaoqi@buaa.edu.cn
Yujing Ma
zy1902407@buaa.edu.cn
Lijiang Chen
chenlijiang@buaa.edu.cn

Department of Electronics and Information Engineering, Beihang University
Beijing, China

Abstract

Recently, many notable convolutional neural networks have powerful performance with compact and efficient structure. To further pursue performance improvement, previous methods either introduce more computation or design complex modules. In this paper, we propose an elegant weight-sharing based ensemble network embedded knowledge distillation (EKD-FWSNet) to enhance the generalization ability of baseline models with no increase of computation and complex modules. Specifically, we first design an auxiliary branch alongside with baseline model, then set branch points and shortcut connections between two branches to construct different forward paths. In this way, we form a weight-sharing ensemble network with multiple output predictions. Furthermore, we integrate the information from diverse posterior probabilities and intermediate feature maps, which are then transferred to baseline model through knowledge distillation strategy. Extensive image classification experiments on CIFAR-10/100 and tiny-ImageNet datasets demonstrate that our proposed EKD-FWSNet can help numerous baseline models improve the accuracy by large margin (sometimes more than 4%). We also conduct extended experiments on remote sensing datasets (AID, NWPU-RESISC45, UC-Merced) and achieve state-of-the-art results.

1 Introduction

Recent years have witnessed significant progress in various computer vision tasks [15, 21, 36, 48, 51, 52, 61] using deep convolutional neural networks. In practical tasks, many devices strictly require networks to have high accuracy with limited memory and computation resource. To address this issue, various methods have been proposed including efficient network design [26, 29, 43, 54], network pruning [17, 19] and knowledge distillation [25].

As a recent popular and powerful method, knowledge distillation (KD) is widely applied in practical application to enhance the generalization ability of baseline models. With

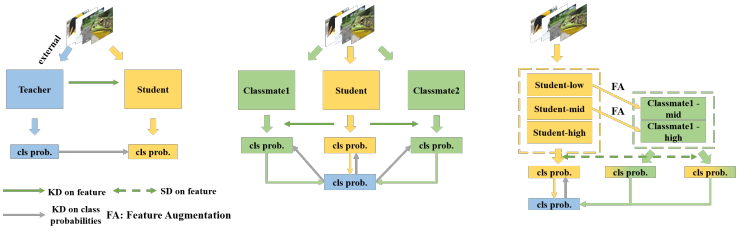

Figure 1: The diagram of previous knowledge distillation based networks and our proposed EKD-FWSNet: **left:** teacher-student network, **middle:** student-classmate ensemble network, **right:** EKD-FWSNet

the help of well-trained "teacher" model, student model can gain performance boost without adding more parameters and complex modules. As shown in Fig.1, there are two main types of KD based networks. In the **left** network, student model is trained to inherit the knowledge of large and high-efficiency teacher model. However, the problem of offline "teacher-student" mechanism is that the dependence of accompanying cumbersome teacher models complicates the training process and increases memory and computation cost. The **middle** network constructs multi-branch ensemble network and integrates KD methods to optimize baseline model in an end-to-end manner. Some recent novel works [1, 3, 56] show the power of this kind of KD based ensemble network. However, with the increase of classmate branches, the training burden increase obviously. In addition, distillation loss functions also increase rapidly when more branches are added, which will also complicate training process.

The above-mentioned issues from the two typical types of KD based networks motivate us to propose a novel and easy-optimized end-to-end training framework (EKD-FWSNet) to make the baseline model (student model) stronger. As shown in Fig. 1 **right**, we first respectively set branch points on student and classmate branch. Then we create connections between two neighboring branch points from different branches to construct multiple forward paths. To enhance the generalization ability of baseline model, we first integrate the multiple predicted posterior probabilities and intermediate feature maps from different forward paths. Then, we apply knowledge distillation to transfer the integrated information to baseline model. Particularly, we do not distill integrated knowledge to classmate branch because the weight-sharing blocks have been optimized once in baseline model. With this ingenious design, EKD-FWSNet has simple yet efficient distillation loss functions. The number of loss functions will not increase with more branches involved. Compared to previous KD based ensemble networks (Fig. 1 **middle**), EKD-FWSNet has much more efficient training process. Moreover, according to the theory of ensemble learning, the final predicted posterior probabilities and intermediate feature maps should have diversity to make voting (soft voting) meaningful. However, the diversity may decrease caused by weight-sharing blocks. To compensate diversity decrease, we insert feature augmentation modules including SE (squeeze-and-excitation) [29], CAM (channel attention module) [14] and Dropout [53] after weight-sharing blocks of baseline model.

To verify the effectiveness of EKD-FWSNet, we conduct extensive experiments on multiple benchmark classification datasets (CIFAR-10/100 [35] and tiny-ImageNet[1]) using cur-

[1] https://tiny-imagenet.herokuapp.com

rent notable baseline models (e.g., ResNet [20, 21] and EfficientNet [54]). Baseline models optimizing in EKD-FWSNet perform much better. Some baseline models, especially lightweight models (e.g., ResNet-20/32) gain more than 4% classification accuracy improvement, which achieves the new state-of-the-art results. We also extend our experiments on remote sensing classification datasets (AID [13], NWPU-RESISC45 [9], UC-Merced [62]) and also achieve state-of-the-art results. The main contributions are listed as follows.

- We propose a novel and efficient KD embedded weight-sharing based ensemble network to improve the performance of baseline model without adding extra structures and modules, so no extra memory and computation cost are introduced during inference.
- EKD-FWSNet maximally explores the potential of weight-sharing blocks, which eases the training burden and provides an insight of designing high-efficiency KD based ensemble training framework.
- We propose online feature augmentation blocks to compensate knowledge diversity decrease caused by large number of weight-sharing blocks.
- Baseline models training in EKD-FWSNet perform much better (sometimes improve over 4%) than individually training. On some benchmark datasets, our proposed method achieves state-of-the-art classification results.

2 Related Work

Efficient convolutional neural networks. In recent years, researchers have introduced many high-accuracy efficient networks for embedded devices with limited computing resource. Some networks like SqueezeNet [33], MobileNet [27, 28, 50], ShuffleNet [66] and EfficientNet [54] utilize elegant structures to make the model compact and efficient. To compress networks, some researchers also apply low-bit quantization technique [18, 32, 40, 47], low-rank decomposition [12, 34] and network pruning [2, 41, 46].

Knowledge distillation. KD is a notable method [25] to transfer knowledge from a larger teacher model to a small student model. It is now widely used in various computer vision tasks such as classification [49, 55, 64], detection [4, 42], segmentation [44, 58], and re-identification [7]. Recently, many notable works apply online knowledge distillation to improve the generalization performance in an end-to-end manner without using pretrained teacher models. [59, 60] propose an elegant KD based training framework to obtain high-accuracy lightweight models by exploiting the data representation invariance within student network itself. [31, 63, 65] dynamically generate several student networks from a full-size network in depth-level or width-level. Some recent excellent works integrate ensemble learning method into KD embedded frameworks [1, 68]. [30, 38] construct ensemble KD from snapshots of iterative pruning, which achieves competitive performance. [3, 56] design student-classmate network (Fig.1 **middle**) to obtain ensemble knowledge as teacher knowledge, which can guide student and classmate efficiently.

Unsupervised representation learning methods. Our proposed method also has close relation with some famous unsupervised representation learning methods, such as MoCo [23], simCLR/simCLRv2 [5, 6] and BYOL [16] in terms of "knowledge transferring" and self-supervision among different branches. The difference can be summarized as follows. 1)

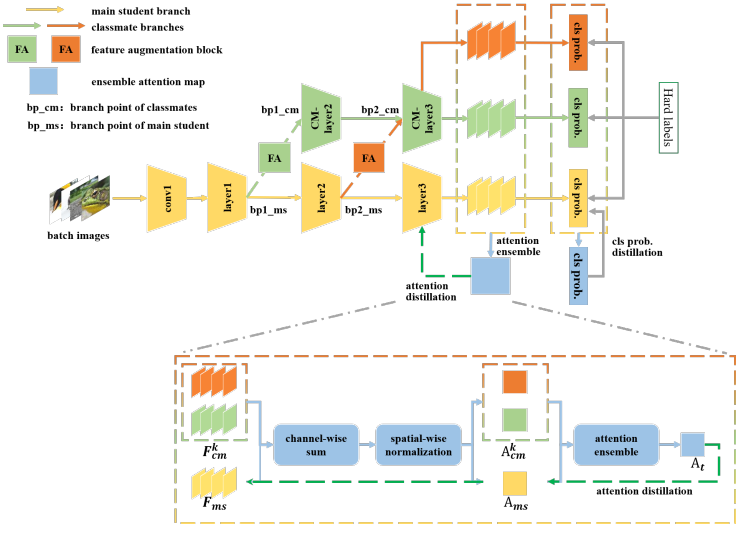

Figure 2: **The architecture of EKD-FWSNet.** We set four branch points (bp1_cm, bp2_cm, bp1_ms, bp2_ms) on main student and classmate branch to form three-branch ensemble network. CM_layer2 and CM_layer3 are convolutional blocks, which respectively have same structure as layer1 and layer2. With four branch points and classmate branch, we form three forward paths ("conv1→layer1→layer2→layer3", "conv1→layer1→CM_layer2→CM_layer3" and "conv1→layer1→layer2→CM_layer3".) To transfer knowledge to baseline model ("main student"), We adopt ensemble attention distillation and class probability distillation respectively on intermediate feature maps and posterior class probabilities. Meanwhile, all branches are optimized with hard labels.

Unsupervised representation learning methods always employ contrastive learning to transfer knowledge between feature embeddings of two transformed images, which can enhance the representation capacity by maximally exploring the intra-class similarity and inter-class variation in latent space. In our proposed method, we adopt "knowledge distillation" mainly for transferring more mature knowledge from stronger ensemble model to naive baseline model. 2) The motivation of constructing multiple branches is different. Unsupervised representation learning methods construct two branches to provide two transformations of input images, which enriches the representation of each category in latent space. We construct multiple branches for providing diverse predictions to obtain a stronger ensemble prediction.

3 Methodology

3.1 Architecture of EKD-FWSNet

To enhance the generalization ability of baseline model without adding extra parameters and modules, we propose EKD-FWSNet with simple and efficient KD embedded training process. As shown in Fig. 2, we respectively set two branch points on main student and classmate branch and create connections to construct three-branch ensemble network. In

EKD-FWSNet, "conv1 layer1", "layer2" and "CM_layer3" are weight-sharing blocks. If more diverse output predictions are required, we can flexibly set more branch points and construct more forward paths. During training, images are first fed into "conv1 layer1". And then the feature maps respectively pass through layers of main student and classmate branch to generate different predictions. During inference, we prune classmate branch and only leave main student branch. With this architecture, EKD-FWSNet saves more training cost and maximally explores the representation potential of weight-sharing blocks.

3.2 Online Feature Augmentation

Our proposed EKD-FWSNet utilizes weight-sharing blocks to save memory cost. However, when more weight-sharing blocks involve, each forward path will get similar representation, which will harm the diversity of different predictions. To solve this problem, we propose online feature augmentation blocks (FA) to enrich the representation of each path. As shown in Fig. 2, FA blocks are set between the two neighboring branch points of main student branch and classmate branch. In this paper, we mainly employ SE [29], CAM [14] and Dropout [53] blocks as FA blocks. Among them, SE and CAM blocks provide channel-wise attention respectively utilizing global receptive field and relation among channels. Dropout can enrich feature through random occupation and improve the representation ability of architecture.

3.3 Knowledge Distillation in EKD-FWSNet

Designing easy-optimized KD mechanism is important to better optimize baseline model. In previous "student-classmate" ensemble networks [4, 56], distillation loss functions of previous works will rapidly increase when integrating more classmate branches. Too many auxiliary distillation loss functions complicates the training process and makes the optimization become harder. To ease the training burden, only two distillation loss functions are designed and the number of loss functions which will not increase with more forward paths involved.

Distillation on class probabilities. Generally, we use temperature-scaled softmax operation to generate the posterior class probability $p(\hat{y}=y_i|\boldsymbol{x})$ for input sample $\boldsymbol{x}$. The formulation can be expressed as $p(\hat{y}=y_i|\boldsymbol{x})=\frac{e^{(z_i/T)}}{\sum_{j=1}^{M}e^{(z_j/T)}}$. Here, z denotes the logits, which is the output feature vector of last fully-connected layer. M is total number of classes. T is the temperature value, which is set to 3 in this paper. Then, we formulate the predicted class probability vector of main student branch as $\boldsymbol{p}_{ms}(\boldsymbol{x})=\{p_{ms}(\hat{y}=1|\boldsymbol{x}),\cdots,p_{ms}(\hat{y}=M|\boldsymbol{x})\}$. Similarly, the predicted probability vector of k^{th} forward path is defined as $\boldsymbol{p}^{k}_{cm}(\boldsymbol{x})$.

If the number of forward paths is set as K, we first average the class probabilities of all student branches to generate an ensemble teacher class probability $\boldsymbol{p}_{et}(\boldsymbol{x})$, which shows in Eq. 1. Then we adopt Kullback-Leibler (KL) divergence as distillation loss to guide main student. The formulation is shown in Eq. 2, where N is the number of samples of a mini-batch.

$$p_{et}(\hat{y}=y_i|\boldsymbol{x})=\frac{e^{(\frac{1}{K+1}\sum_{k=1}^{K+1}z_{ki})/T}}{\sum_{j=1}^{M}e^{(\frac{1}{K+1}\sum_{k=1}^{K+1}z_{kj})/T}},\quad \boldsymbol{p}_{et}(\boldsymbol{x})=\{p_{et}(\hat{y}=1|\boldsymbol{x}),\ldots,p_{et}(\hat{y}=M|\boldsymbol{x})\} \tag{1}$$

$$KL_{ms}=\frac{1}{N}\sum_{n=1}^{N}KL\big(\boldsymbol{p}_{ms}(\boldsymbol{x}_n)\|\boldsymbol{p}_{et}(\boldsymbol{x}_n)\big)=-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{p}_{et}(\boldsymbol{x}_n)\log\frac{\boldsymbol{p}_{ms}(\boldsymbol{x}_n)}{\boldsymbol{p}_{et}(\boldsymbol{x}_n)} \tag{2}$$

Trained by distillation on class probabilities, main student can "discuss" with classmates and learn from each other. In addition, with the involvement of more forward paths, no extra loss items will be added. Training in this concise way, main student will benefit from mature class probability and get huge enhancement.

Ensemble attention distillation on feature map. To further guide main student in intermediate feature map, we introduce ensemble-attention distillation. As shown in Fig. 2, main branch will be guided by an ensemble attention teacher. We utilize the channel-wise joint information of feature maps to represent attention. We formulate attention generation process in Eq. 3.

$$\boldsymbol{A}_{ms} = \sum_{c=1}^{C} (\boldsymbol{F}_{ms})_{\boldsymbol{c}}, \ \boldsymbol{A}_{cm}^{\boldsymbol{k}} = \sum_{c=1}^{C} (\boldsymbol{F}_{cm}^{\boldsymbol{k}})_c \Rightarrow \boldsymbol{A}_t = \frac{1}{K+1}(norm(\boldsymbol{A}_{ms}) + \sum_{k=1}^{K} norm(\boldsymbol{A}_{cm}^{\boldsymbol{k}})) \tag{3}$$

Note that, C denotes the total number of channels. $norm(\cdot)$ is spatial-wise normalization operation to keep the value consistency of attention maps from different branches. Moreover, $\boldsymbol{F}_{ms}, \boldsymbol{F}_{cm}^{k} \in \mathbb{R}^{C\times H\times W}$ indicate feature maps of main student and classmates at same stage of each branch. $\boldsymbol{A}_{ms}, \boldsymbol{A}_{cm}^{k} \in \mathbb{R}^{1\times H\times W}$ indicate attention map, which will be fused to form ensemble attention teacher ($\boldsymbol{A}_t$). To distill knowledge from $\boldsymbol{A}_t$ to $\boldsymbol{A}_{ms}$, we design mean-squared-error (MSE) loss to implement online ensemble attention learning, which can be denoted as Eq. 4.

$$MSE_{ms} = \frac{1}{N}\sum_{n=1}^{N} MSE(\boldsymbol{A}_{ms}(\boldsymbol{x}_n) \| \boldsymbol{A}_{\boldsymbol{t}}(\boldsymbol{x}_n)) = \frac{1}{N}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} \left((a_{ms}(\boldsymbol{x}_n))_{hw} - (a_t(\boldsymbol{x}_n))_{hw}\right)^2 \tag{4}$$

Here, H and W are height and width of the attention map and $((a_{ms})(\boldsymbol{x}_n))_{hw}$ denotes the pixel value at position (h,w) on attention map of n^{th} batch image.

Models	layer combination
ResNet-20/32/44/56	[3, 3, 3]/[5, 5, 5]/[7, 7, 7]/[9, 9, 9]
ResNet-18/34	[2, 2, 2, 2]/[3, 4, 6, 3]
EfficientNet-b0	[1, 2, 2, 3, 3, 4, 1]
EfficientNet-b2	[2, 3, 3, 4, 4, 5, 2]
EfficientNet-b4	[2, 4, 4, 6, 6, 8, 2]

Table 1: The structure configuration of baseline models [21, 54]. Layer combination represents the network layer structure. E.g. ResNet-34 has four stages. The number of basic blocks in each layer are respectively 3, 4, 6, 3. In addition, the block types of ResNet and EfficientNet are respectively basic block and MBConv block.

3.4 Optimizing EKD-FWSNet

To optimize EKD-FWSNet, we apply conventional cross-entropy loss (L_{ms} for main student and L_{cm}^{k} for k^{th} classmates) to train network with hard labels. We also apply the above mentioned two types of distillation loss functions (Eq. 2, Eq. 4). Finally, the total loss of EKD-FWSNet (L) is defined in Eq. 5.

$$L = \underbrace{L_{ms} + \frac{1}{K}\sum_{k=1}^{K} L_{cm}^{k}}_{\text{cross-entropy}} + \underbrace{w \cdot KL_{ms}}_{\text{KL distillation}} + \underbrace{\alpha \cdot MSE_{ms}}_{\text{ensemble attention}} \tag{5}$$

Models	CIFAR-10		CIFAR-100	
	BL	EKD-FWSNet	BL	EKD-FWSNet
ResNet-20	8.37	6.69	32.66	28.54
ResNet-32	7.35	6.03	30.73	26.46
ResNet-44	6.78	5.83	29.43	25.74
ResNet-56	6.13	5.49	28.91	25.63

Table 2: Top-1 error rate (%) of lightweight models on CIFAR-10/100. "BL" means individually training baseline model.

Models	CIFAR-100		tiny-ImageNet	
	BL	EKD-FWSNet	BL	EKD-FWSNet
ResNet-18	23.71	20.49	30.91	26.81
ResNet-34	22.16	19.94	23.83	21.74
EfficientNet-b0	12.64	11.33	18.91	17.73
EfficientNet-b2	11.34	10.21	16.41	15.02
EfficientNet-b4	10.03	8.90	13.78	12.59

Table 3: Top-1 error rate (%) of high-efficiency models on CIFAR-100/tiny-ImageNet.

Here, we use w and α as hyper-parameters to adjust the proportions of KL distillation loss and ensemble attention loss respectively. Usually, w and α are respectively set to 60 and 1. In addition, we always apply ensemble attention distillation only at the position before fully-connected layers. Because low-level features naturally have very different attention regions. Forcing main student to learn in early stage may obtain more noise than useful knowledge. In supplementary materials, we will elaborate on the principle of hyper-parameter adjusting though experiments.

4 Experiments

4.1 Datasets and Implementation Details

In this paper, we mainly evaluate our method on CIFAR-10/100 [35] and tiny-ImageNet. CIFAR-10/100 contains 50000 training samples and 10000 testing samples which are tiny RGB images with 32×32 pixels. Tiny-ImageNet dataset consists of a subset of ImageNet images [11]. It contains 200 classes, each of which has 500 RGB images for training and 50 RGB images for validation. The size of image is 64×64. Recently, remote sensing (RS) scene classification task becomes popular. To prove the generalization of our method, we also make experiments on some notable RS scene classification benchmark datasets including AID [13], NWPU-RESISC45 [9] and UC-Merced [62].

In this paper, we select ResNet [21] and EfficientNet [54] as baseline models. The detailed structure of baseline models are shown in Tab.1. Moreover, some baseline models like DenseNet-121 are only designed for RS scene classification task to make fair comparison with previous RS works. On CIFAR-10/100, we use Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 0.0001. The initial learning rate is set to 0.1 and the mini-batch size is set to 128. The total number of training epochs is 300 and learning rate will be divided by 10 at epoch 150 and 250. On tiny-ImageNet, we follow the optimizer setting of CIFAR. Differently, every networks are trained 100 epochs. The learning rate is set to 0.01 initially and decreases by 10 times at epoch 50 and 75. Specifically, followed the setting of [38], we load ImageNet pretrained parameters when training on tiny-ImageNet.

4.2 Experimental Results

Classification on lightweight baseline models. In real applications, lightweight baseline models are always required. Lightweight models indicate those models with few parameters and computation cost, which sacrifice the performance to meet the demand of limited

computation resource. Therefore, we conduct experiments on ResNet20/32/44/56, which are notable lightweight model series. Followed [21], we evaluate lightweight ResNet series on CIFAR-10/100. The results are shown in Tab. 2. In EKD-FWSNet, the first branch point is set at layer1. We construct a three-forward-paths structure using Dropout as FA block. From the results, we observe that baseline models training in EKD-FWSNet make huge improvement. On CIFAR-10 classification experiments, ResNet-20 and ResNet-32 respectively get improved by 1.68% and 1.32% training in EKD-FWSNet. Larger baseline models ResNet-44/56 also improve by large margin. On a more challenging task CIFAR-100, our proposed EKD-FWSNet also shows encouraging performance. Compared to individually training baseline models, the average accuracy improvement of ResNet-20/32/44/56 training with EKD-FWSNet is 3.89%.

Models	CIFAR-10			CIFAR-100		
	SD	OEM	EKD-FWSNet	SD	OEM	EKD-FWSNet
ResNet-20	6.89	7.87	**6.69±0.10**	30.05	-	**28.54±0.08**
ResNet-32	6.10	7.05	**6.03±0.08**	28.22	29.03	**26.46±0.14**
ResNet-44	-	6.55	**5.83±0.05**	-	28.24	**25.74±0.13**
ResNet-56	6.02	-	**5.49±0.05**	26.78	27.84	**25.63±0.17**

Table 4: Top-1 error rate (%) lightweight model comparison on CIFAR-10/100. All results are "average value ± standard deviation" of three runs. In addition, "SD" and "OEM" respectively indicate method of [60] and [56].

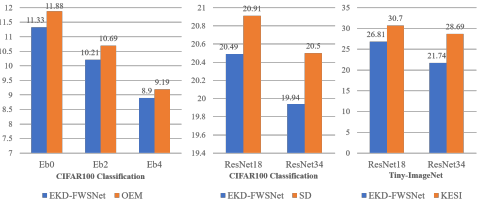

Figure 3: Top-1 error rate (%) high-efficiency model comparison on CIFAR-100 and tiny-ImageNet. All results are "average value ± standard deviation" of three runs. In addition, "SD", "OEM" and "KESI" respectively indicate method of [60], [56] and [38]

Classification on high-efficiency baseline models. Recently, some high-efficiency baseline models have shown strong performance with compact structure. High-efficiency models denote some recent notable compact models, which can achieve high performance with fewer cost using high-efficiency design. Therefore, further enhancing their generalization ability is hard. In this paper, we conduct experiments to show that training with EKD-FWSNet, high-efficiency models can also get improved. As shown in Tab.3, ResNet-18/34 training in EKD-FWSNet achieve surprising improvement. Especially on tiny-ImageNet, ResNet18 in EKD-FWSNet surpasses baseline model by 4.10%! Experiments on EfficientNet further prove the effectiveness of EKD-FWSNet. Even though EfficientNet series have already achieved very competitive results, they can still be improved by more than 1%.

Comparison of KD based networks. [60] and [56] are two recent notable methods which have similar motivation as our work. [60] proposes a typical KD based training framework to enhance the capability of baseline models. [56] uses KD based student-classmate network (Fig. 2 **middle**) to explore the potential of student and each classmate. Tab. 4 shows the comparison results on lightweight model. It is clear that baseline models training in EKD-FWSNet perform better. Especially on CIFAR-100, our networks can surpass [60] and [56] by more than 1%. On high-efficiency models, EKD-FWSNet is also more competitive. As shown in Fig. 3, after optimizing with EKD-FWSNet, high-efficiency baseline models obviously obtain much lower error rate on CIFAR-100. On tiny-ImageNet, we compare the classification results of main student with the ensemble results of KESI (Knowledge Distillation from Ensembles of Snapshots of Iterative Pruning) [38]. EKD-FWSNet has much better performance (Fig. 3), Since we reinplement baseline models with higher accu-

Models	CIFAR-10				CIFAR-100			
	KD-ONE	DML	OEM	EKD-FWSNet	KD-ONE	DML	OEM	EKD-FWSNet
ResNet-32	**5.99**	6.68	7.05	6.03±0.08	26.61	28.90	29.03	**26.46±0.14**
ResNet-32-E	-	-	5.73	**5.14±0.07**	24.63	-	26.06	**24.15±0.09**

Table 5: Comparison of KD and ensemble based networks. We select ResNet-32 as baseline model and respectively compare the "main student"(ResNet-32: baseline model after distilling) and "ensemble teacher"(ResNet-32-E: ensemble teacher) results with previous notable KD and ensemble based networks, KD-ONE [57], DML [67] and OEM [56].

Methods	AID		NWPU-RESISC45		UC-Merced
	T.R.=20%	T.R.=50%	T.R.=10%	T.R.=20%	T.R.=80%
MSCP [43]	92.21±0.17	96.56±0.18	88.07±0.18	90.81±0.13	98.40±0.34
DCNN [11]	90.82±0.16	96.89±0.10	89.22±0.50	91.89±0.22	98.93±0.10
RTN [8]	92.44	-	89.90	92.71	98.96
SCCov [24]	93.12±0.25	96.10±0.16	89.30±0.35	92.10±0.25	99.05±0.25
MG-CAP [53]	93.34±0.18	96.12±0.12	90.83±0.12	92.95±0.13	99.0±0.10
Hydra [45]	-	-	92.44±0.34	94.51±0.21	-
KFBNet [39]	95.50±0.27	97.40±0.10	93.08±0.14	95.11±0.10	**99.88±0.12**
EKD-FWSNet	**95.89±0.14**	**97.60±0.16**	**93.24±0.15**	**95.17±0.07**	99.81±0.10

Table 6: Comparison of classification results (%) on UC-Merced, AID and NWPU-RESISC45 datasets.

racy, it is unfair to directly compare the accuracy. Therefore, we also compare the margin of improvement. On ResNet-18, EKD-FWSNet and KESI respectively improve by 4.10% and 2.08%. On ResNet-34, EKD-FWSNet and KESI respectively improve by 2.09% and 2.50%. Obviously, on ResNet-18, EKD-FWSNet has huge advantage while on ResNet-34, EKD-FWSNet has minor inferiority.

Comparison of KD and ensemble based networks. KD-ONE [57], DML [67] and OEM [56] are three recent notable methods which integrate knowledge distillation and ensemble learning. All this three methods can be regarded as typical student-classmate networks (Fig. 2 **middle**). As shown in Tab. 5, when using ensemble teacher to teach "main student", EKD-FWSNet achieves comparable results (0.04% less on CIFAR-10 while 0.15% better on CIFAR-100) with KD-ONE. We also compare the accuracy of our ensemble teacher to other ensemble knowledge distillation based methods to prove the overall superiority of EKD-FWSNet. From Tab. 5, we observe that EKD-FWSNet has obvious better ensemble teacher, which proves the effectiveness of our method (better teacher, better student).

Classification comparison on RS datasets. As shown in Tab.6, we list recent notable methods of RS scene classification task. The baseline network is DenseNet-121 (same setting as SOTA method, KFBNet [39]). It is clear that training in EKD-FWSNet, DenseNet-121 achieves SOTA results on RS datasets. Specifically, UC-Merced dataset only has 2100 images with 21 categories. The training rate is 80%, which means only 420 images are served as test data. Our models training in EKD-FWSNet achieve high accuracy close to full marks. We run every experiments five times and calculate the mean and standard deviation of average accuracy. On UC-Merced dataset, EKD-FWSNet obtains twice 100% accuracy and three times 99.76%. From comparison results, we can observe that our networks and KFBNet both reach the ultimate limit and have obvious advantage against other methods.

Models	CIFAR-100	
	EKD-FWSNet(w/o)	EKD-FWSNet
ResNet-20	29.39	28.54
ResNet-32	26.91	26.46
ResNet-44	26.03	25.74
ResNet-56	25.78	25.63
ResNet-18	21.09	20.49
ResNet-34	20.22	19.94

Table 7: Top-1 error rate (%) comparison on CIFAR-100. "(w/o)" means ensemble attention distillation is not applied.

Networks	Dropout []	SE []	CAM []	Branch-num
ResNet-20@layer1	28.54	28.76	29.20	3
ResNet-32@layer1	26.46	26.69	27.07	3
ResNet-44@layer1	25.74	25.69	26.03	3
ResNet-56@layer1	25.63	25.93	26.21	3
ResNet-18@layer1	20.49	20.95	21.59	4
ResNet-34@layer1	19.94	20.55	20.92	4

Table 8: Effectiveness of different feature augmentation blocks on EKD-FWSNet. We set first branch point after "layer1" (@layer1). Therefore, the branch number of lightweight networks (ResNet20/32/44/56) and high-efficiency networks (ResNet18/34) is respectively 3 and 4.

4.3 Ablation Study

Effectiveness of ensemble attention distillation. To show the separate effectiveness of ensemble attention distillation on intermediate feature maps. We conduct experiments shown in Tab. 7. When adding ensemble attention distillation, baseline models improve by average 0.41%. Although some baseline models improve little, it still works at most situations. Theoretically, [60] and [56] employ distillation approach in intermediate feature maps by constructing distillation loss between two 3-dimension feature maps (tensor-level), i.e., they construct high-dimension approximation, which will imposes an extra load on optimizing (Curse of Dimensionality). Moreover, with the increasing of branches, distillation loss terms increase. Our proposed EKD-FWSNet avoids the above problems. No matter how many forward paths are added, we only apply 2-dimension (matrix-level) ensemble attention distillation with only one loss terms, which can easily optimize baseline models.

Effectiveness of different FA blocks. In our paper, we apply feature augmentation blocks to prevent the diversity loss of each branch's final logits caused by weight-sharing layers. To analyze the effectiveness of online feature augmentation blocks, we compare the influence of different FA blocks. We use ResNet series and run classification experiments on CIFAR-100. Results in Tab. 8 shows that Dropout performs better in most cases.

5 Conclusion

In this paper, we propose EKD-FWSNet to explore the generalization ability of baseline models. Baseline models training in EKD-FWSNet gain improvement by ensemble distillation on class probabilities and attention maps. To ease training burden when involving knowledge distillation, we design flexible weight-sharing mechanism and concise distillation loss. Experiments prove that EKD-FWSNet is more competitive than previous methods on improving both lightweight and high-efficiency models. All in all, we provide a novel ensemble training framework with easy-optimized knowledge distillation strategy, which makes baseline model stronger without adding extra parameters and computation costs.

Acknowledgment This work was supported in part by the National Natural Science Foundation of China under Grant 62072021 and in part by the Fundamental Research Funds for the Central Universities under Grant YWF-21-BJ-J-534.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *CoRR*, abs/2012.09816, 2020.

[2] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems*, 13(3):32:1–32:18, 2017.

[3] Umar Asif, Jianbin Tang, and Stefan Harrer. Ensemble knowledge distillation for learning improved and efficient networks. In *European Conference on Artificial Intelligence (ECAI)*, volume 325, pages 953–960, 2020.

[4] Guobin Chen, Wongun Choi, Xiang Yu, Tony X. Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning ICML*, volume 119, pages 1597–1607, 2020.

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, 2020.

[7] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2852–2859, 2018.

[8] Zan Chen, Shidong Wang, Xingsong Hou, and Ling Shao. Recurrent transformer network for remote sensing scene categorisation. In *British Machine Vision Conference*, page 266, 2018.

[9] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[10] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE Trans. Geosci. Remote. Sens.*, 56(5):2811–2821, 2018.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[12] Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, pages 1269–1277, 2014.

[13] G. Xia et al. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

[14] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3146–3154, 2019.

[15] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, and et al. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[17] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[18] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[19] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–645, 2016.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9726–9735, 2020.

[23] Nanjun He, Leyuan Fang, Shutao Li, Antonio Plaza, and Javier Plaza. Remote sensing scene classification using multilayer stacked covariance pooling. *IEEE Trans. Geosci. Remote. Sens.*, 56(12):6899–6910, 2018.

[24] Nanjun He, Leyuan Fang, Shutao Li, Javier Plaza, and Antonio Plaza. Skip-connected covariance network for remote sensing scene classification. *IEEE Trans. Neural Networks Learn. Syst.*, 31(5):1461–1474, 2020.

[25] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

[26] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1314–1324. IEEE, 2019.

[27] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.

[28] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

[29] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[30] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations, ICLR*, 2017.

[31] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations (ICLR)*, 2018.

[32] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems*, pages 4107–4115, 2016.

[33] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016. URL http://arxiv.org/abs/1602.07360.

[34] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *British Machine Vision Conference (BMVC)*, 2014.

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[37] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, pages 7528–7538, 2018.

[38] Duong H. Le, Vo Trung Nhan, and Nam Thoai. Paying more attention to snapshots of iterative pruning: Improving model compression via ensemble distillation. In *British Machine Vision Conference (BMVC)*, 2020.

[39] F. Li, R. Feng, W. Han, and L. Wang. High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 58(11):8077–8092, 2020.

[40] Hao Li, Soham De, Zheng Xu, Christoph Studer, Hanan Samet, and Tom Goldstein. Training quantized nets: A deeper understanding. In *Advances in Neural Information Processing Systems*, pages 5811–5821, 2017.

[41] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*, 2017.

[42] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7341–7349, 2017.

[43] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019.

[44] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2604–2613, 2019.

[45] Rodrigo Minetto, Maurício Pamplona Segundo, and Sudeep Sarkar. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Trans. Geosci. Remote. Sens.*, 57(9):6530–6541, 2019.

[46] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*, 2017.

[47] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 525–542, 2016.

[48] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[49] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[50] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4510–4520, 2018.

[51] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[53] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[54] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning(ICML)*, volume 97, pages 6105–6114, 2019.

[55] Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Özlem Aslan, Shengjie Wang, Abdelrahman Mohamed, Matthai Philipose, Matthew Richardson, and Rich Caruana. Do deep convolutional nets really need to be deep and convolutional? In *International Conference on Learning Representations (ICLR)*, 2017.

[56] Devesh Walawalkar, Zhiqiang Shen, and Marios Savvides. Online ensemble model compression using knowledge distillation. In *European Conference on Computer Vision (ECCV)*, volume 12364, pages 18–35, 2020.

[57] Shidong Wang, Yu Guan, and Ling Shao. Multi-granularity canonical appearance pooling for remote sensing scene classification. *IEEE Trans. Image Process.*, 29:5396–5407, 2020.

[58] Jiafeng Xie, Bing Shuai, Jianfang Hu, Jingyang Lin, and Wei-Shi Zheng. Improving fast segmentation with teacher-student learning. In *British Machine Vision Conference (BMVC)*, page 205, 2018.

[59] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 5565–5572, 2019.

[60] Ting-Bing Xu and Cheng-Lin Liu. Deep neural network self-distillation exploiting data representation invariance. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2020.

[61] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 899–980 2019.

[62] Yi Yang and Shawn D. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Sigspatial International Symposium on Advances in Geographic Information Systems*, pages 270–279, 2010.

[63] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas S. Huang. Slimmable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[64] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[65] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3712–3721, 2019.

[66] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6848–6856, 2018.

[67] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 4320–4328, 2018.

[68] Kaikai Zhao, Tetsu Matsukawa, and Einoshin Suzuki. Retraining: A simple way to improve the ensemble accuracy of deep neural networks for image classification. In *International Conference on Pattern Recognition (ICPR)*, pages 860–867, 2018.