Rethinking Online Knowledge Distillation with Multi-Exits

Hojung Lee[0000−0001−9312−3904] and Jong-Seok Lee[0000−0002−8038−1119]

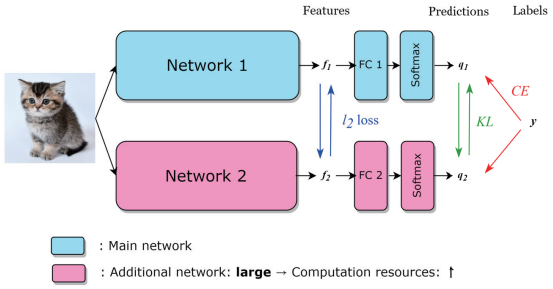
School of Integrated Technology, Yonsei University, Incheon 21983, South Korea
{hjlee92,jong-seok.lee}@yonsei.ac.kr

Abstract. Online knowledge distillation is a method to train multiple networks simultaneously by distilling the knowledge among each other from scratch. An efficient way for this is to attach auxiliary classifiers (called exits) to the main network. However, in this multi-exit approach, there are important questions that have not been answered in previous studies: *What structure should be used for exits? What can be a good teacher for distillation? How should the overall training loss be constructed?* In this paper, we propose a new online knowledge distillation method using multi-exits by answering these questions. First, we examine the influence of the structure of the exits on the performance of the main network, and propose a bottleneck structure that leads to improved performance for a wide range of main network structures. Second, we propose a new distillation teacher using an ensemble of all the classifiers (main network and exits) by exploiting the diversity in the outputs and features of the classifiers. Third, we propose a new technique to form the overall training loss, which balances classification losses and distillation losses for effective training of the whole network. Our proposed method is termed balanced exit-ensemble distillation (BEED). Experimental results demonstrate that our method achieves significant improvement of classification performance on various popular convolutional neural network (CNN) structures. Code is available at https://github.com/hjdw2/BEED.

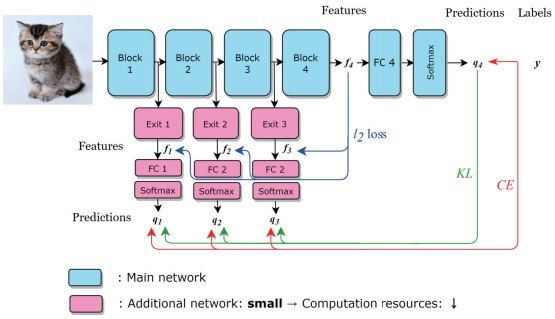
Keywords: Online knowledge distillation · Multi-exits · Ensemble.

1 Introduction

Deep neural networks have made remarkable achievements in the field of image classification with the advancement of convolutional neural networks. These achievements are often based on deep and wide networks [11]. In order to successfully use deep learning in resource-limited environments such as mobile or embedded systems, model compression approaches have been studied. Knowledge distillation [10, 13] is one such approach, which transfers learned knowledge, such as predictions or intermediate feature maps, from a large pre-trained teacher network to a smaller student network. The student network tries to mimic the



(a) Using independent networks



(b) Using auxiliary classifiers (exits)

Fig. 1. Comparison of online distillation methods using additional networks. A black arrow indicates the forward path; a red arrow indicates the cross-entropy loss; a green arrow implies the prediction distillation; and a blue arrow implies the feature distillation.

teacher's knowledge, which improves the performance of the student network. Then, the student network can be deployed instead of the teacher network for a resource-constrained environment. However, pre-training a large teacher network is a significant burden, and this issue becomes even worse when an ensemble model is used as a teacher [1].

Knowledge distillation can also be used without a pre-trained model, which is called online distillation. Online distillation trains multiple networks simultaneously by distilling the knowledge with each other from scratch. There are generally two ways to configure the networks as shown in Figure 1. One is to use multiple independent networks in addition to the main network [42, 4, 15], and the other is to attach auxiliary classifiers (called exits) in the middle of the main network [27, 40, 23]. In the former case, independent networks are used and thus computational complexity is high, which reduces the advantage of online distillation. In the latter case, on the other hand, the exits are usually small and the complexity can be effectively decreased. For example, when ResNet34 is trained for CIFAR-100 [18] using two independent networks, the total number of parameters is 42.96M; when three exits are used, however, only 31.30M parameters are

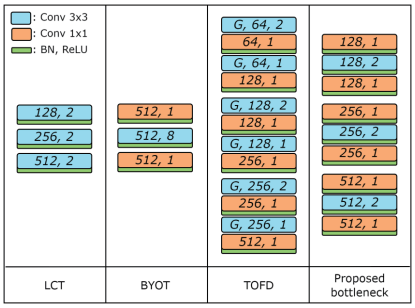


Fig. 2. Structures of the exits (the first exit in ResNet18) in the previous studies (LCT [21], BYOT [40], TOFD [39]) and this paper. G means group convolution.

required (and four classifiers are obtained). Therefore, the approach using exits has the advantages of involving reduced complexity and obtaining more outputs that can be used for distillation.

While the success of online distillation using multi-exits was shown in the previous studies [27, 40, 23], we note that there are three important questions that have not been answered. In this paper, we offer solutions for the questions and propose a new online distillation method using multi-exits.

Q1. What structure should be used for exits? In most previous studies [40, 27, 39], the structure of an exit is determined mainly to match the resolution of the final feature maps to that of the main network without much consideration about its influence on the performance (Figure 2). However, our analysis reveals that the performance of the main network can be significantly changed depending on which structure is used for exits in the same main network. In particular, we show that it is beneficial to use a different block type from the main network for exits. In addition, we present a bottleneck structure for exits, which has a simple structure but yields higher performance than other previously proposed bottleneck structures.

Q2. What can be a good teacher for distillation? The previous distillation methods using multi-exits [27, 40, 23] consider the exists as students and the main network as a teacher, and expect that distillation of the teacher's knowledge to the students eases learning of the network, especially for the early layers. However, we rethink this typical role assignment: Since the multi-exits inherently provide multiple outputs, we can use them to constitute a better teacher. Thus, we propose a new ensemble distillation teacher using an ensemble of all the classifiers in the network, which allows us to exploit diversity in the outputs and features of the classifiers. In particular, we apply an importance coefficient to adjust the relative contribution of each exit in the ensemble, which maximizes the advantage of the ensemble.

Q3. How should the overall training loss be constructed? After the distillation losses and the classification losses are obtained, how to properly reflect them in

the overall training loss is still an open problem. In general, the learning ability of a network is closely related to its size. Thus, it is difficult for early exits having small sizes to learn properly only with the classification loss using the one-hot labels, and they often find a shortcut for classification criteria [9]. The distillation loss can alleviate this limitation, but applying the distillation loss at the same rate to all exits, as in the previous studies [27, 40, 23], does not consider the size-dependent learning ability of each exit. Thus, we propose a new loss-balancing technique, which adjusts the weights of the classification loss and the distillation loss for each exit. This technique can improve the performance of the main network by helping the exits learn appropriate features.

To sum up, our contributions are as follows: 1) We propose a new online knowledge distillation method using multi-exits. 2) We investigate the effect of the structure of the exits and present a simple but effective structure. 3) We propose a method to form an ensemble of the classifiers as a teacher for distillation. 4) We propose a loss-balancing technique to combine classification losses and distillation losses.

2 Related Work

Knowledge distillation is a method to use a pre-trained network as a teacher network and distill its learned knowledge to a smaller student network [13]. The teacher's knowledge can be extracted at different levels, including logits [13] and features [28], which can be used directly or after transformation using another network or a kernel function [12, 3, 17, 43]. In addition, a teacher can be a single pre-trained network or an ensemble of multiple pre-trained networks [1].

Online distillation trains multiple networks simultaneously by distilling the knowledge with each other without a pre-trained teacher. To this end, several independent networks can be used [42, 4, 35, 24, 15, 30, 31] or auxiliary classifiers (i.e., exits) attached to the main network can be used [27, 40, 23]. The latter approach is preferable in terms of complexity, since the former approach requires higher complexity due to the large network size. In a multi-exit architecture [27, 40, 23], the exits can be used as paths to deliver the information at the main network's output to its early layers via distillation. Through this, each exit and shared parts of the main network can learn more general features, by not only following the true label but also receiving the knowledge of the main network from the final output. Consequently, the performance of not only the exits but also the main network is enhanced. While these studies have shown the success of online distillation using multi-exits, we focus on the research questions mentioned in the introduction, which have not been addressed yet. We show that careful consideration of the questions can significantly improve the classification performance of the trained network.

The on-the-fly native ensemble (ONE) method in [19] attaches multiple exits at a certain location of the main network, where the exits have the same structure to the main network and the ensemble of the exits form a teacher. However, since the exits are attached at the same location and have the same structure,

the added value of using them for an ensemble is limited. Our experiments show that this method is outperformed by our proposed method.

3 Structure of Auxiliary Classifiers

In the previous studies, the structure of the exits is usually designed heuristically. More specifically, in an exit, the number of convolutional layers having a stride of two is determined so that the dimension of the features at the penultimate layer of the exit becomes the same to that of the main network. However, it is well known that the network structure is important in determining its performance [16]. From the same point of view, we can infer that the structure of the exits can also have a significant impact on the performance, which is investigated in this section.

The role of the exits is to provide additional training objectives to the network other than the classification loss at the final output. These additional objectives act as regularizers to prevent overfitting, which improves the generalization performance of the main network [34]. Considering this, we can hypothesize as follows: It is beneficial to use exit structures different from that of the main network so that the exits produce features and outputs that are distinguished from those of the main network, and this diversity in turn can result in informative regularizers.

Most popular CNNs are constructed based on the ResNet structure, which consist of either *basic blocks* or *bottleneck blocks*. Therefore, as a way of using a different structure for exits from the main network, we suggest using bottleneck blocks for an exit if the main network consists of basic blocks, and using basic blocks for an exit if the main network consists of bottleneck blocks.

To verify our argument, we train multi-exit architectures using different exit structures shown in Figure 2. When we build a multi-exit architecture by inserting $k-1$ exits into a CNN (e.g., $k=4$ in Figure 1b), the exits, denoted as c_i $(i=1,...,k-1)$, divide the CNN (main network) into k blocks, which are denoted as g_i $(i=1,...,k)$. Then, the input data x produces k predictions (logits), denoted as z_i $(i=1,...,k)$, by a single feedforward process, i.e.,

$$z_i = \begin{cases} c_i(g_i(\cdots g_1(x))) & \text{if } i < k \\ g_i(\cdots g_1(x)) & \text{if } i = k \end{cases} \tag{1}$$

In addition, the feature information right before the fully connected layer of each exit or the main network is denoted as f_i $(i=1,...,k)$. Then, the basic way to train the multi-exit architecture is to use the joint classification loss L_C with cross entropy (CE) for true label y, i.e.,

$$L_C = \sum_{i=1}^{k} CE(q_i, y), \tag{2}$$

where $q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$ is the softmax output and T is the temperature.

Table 1. Test accuracy (%) of the main network trained with different multi-exit structures (LCT [21], BYOT [40], TOFD [39]) for CIFAR-100. 'Baseline' indicates the case without multi-exits and 'Self' indicates the case where the structure of each exit is the same as the remaining main network. Basic block-based networks are marked in red and bottleneck block-based networks are marked in blue. The best results are marked in bold.

Network \ Exit	Baseline	LCT	BYOT	TOFD	Self	Proposed
ResNet18	77.70	78.22	78.51	79.01	78.79	**79.39**
ResNet34	78.01	79.80	79.41	79.92	79.41	**80.22**
WRN16-4	76.42	75.71	76.61	76.81	77.31	**77.49**
WRN28-4	78.50	78.19	78.56	79.02	79.48	**80.05**
MobileNet-V2	71.92	**74.31**	74.10	73.51	72.91	73.73
EfficientNetB0	72.01	**75.01**	74.51	74.38	73.52	74.21

The test accuracy of the main network trained for CIFAR-100 is shown in Table 1 for various main network architectures (see Section 6 for more details of the experimental setup). Significant differences in performance are observed depending on the combination of the main network structure and the exit structure. When the results of the previous exit structures are compared (LCT, BYOT, and TOFD), the exits consisting of a different kind of blocks from the main network lead to higher accuracy than the exits consisting of the same kind of blocks in most cases. For the main networks composed of basic blocks (i.e., ResNets and WRNs), the exit structures using bottleneck blocks are beneficial, including BYOT and TOFD. On the other hand, even though LCT uses a very shallow structure, it shows the best performance among the exit structures when MobileNet-V2 and EfficientNetB0 are used as the main network. These demonstrate that an exit structure using a different type of blocks from that of the main network is preferable as a regularizer.

However, BYOT uses only one bottleneck block, which is too simple to obtain sufficiently high performance. TOFD uses a complex structure, but the rationale for selecting such a structure is not clear. Thus, we propose to use the standardized bottleneck structure employed in the original ResNet [11] for exits as shown in Figure 2. We make slight modifications by excluding channel expansion in the last convolution layer in the block (in order to match the dimension of the penultimate layer's features to that of the main network) and the residual connection (to reduce the computational burden as discussed in [22]). As shown in Table 1, our proposed bottleneck exit achieves better performance than the other bottleneck-based exit structures for the main networks based on basic blocks. In addition, our bottleneck achieves the best performance also for MobileNet-V2 and EfficientNetB0 when distillation is applied, which is shown in the supplementary material.

It is possible to use the exits having the same structure to the main network but initialize them differently to impose diversity in the additional training ob-

Table 2. Test accuracy (%) of each classifier (main network or exit) and diversity (prediction disagreement / cosine similarity) between the main network and each exit when ResNet18 is used as the main network for CIFAR-100. The exit structure is denoted as (Exit1 structure)-(Exit2 structure)-(Exit3 structure). For example, 'C-B-B' means that Exit1 uses the LCT structure and Exit2 and Exit3 use the proposed bottleneck structure. 'X' means that Exit1 is not attached.

Exit Structure	Exit1 Acc.(Div.)	Exit2 Acc.(Div.)	Exit3 Acc.(Div.)	Main Acc.
LCT (C-C-C)	70.90 (0.2731/0.7914)	75.20 (0.2016/0.8577)	77.72 (0.0547/0.9831)	78.22
BYOT	66.88 (0.3146/0.7578)	73.33 (0.2176/0.8433)	77.37 (0.0766/0.9671)	78.51
TOFD	72.17 (0.2590/0.8009)	75.30 (0.2119/0.8447)	77.82 (0.0952/0.9577)	79.01
Proposed Bottleneck (B-B-B)	74.55 (0.2376/0.8263)	77.19 (0.1897/0.8670)	78.84 (0.1017/0.9485)	79.39
B-B-C	74.12 (0.2456/0.8160)	76.90 (0.1970/0.8613)	77.71 (0.0637/0.9805)	78.18
C-C-B	71.58 (0.2663/0.8014)	75.24 (0.2012/0.8612)	78.09 (0.0964/0.9545)	79.12
C-C-ResNet50	70.79 (0.2776/0.7951)	74.48 (0.2107/0.8518)	78.68 (0.1061/0.9408)	79.35
B-C-C	73.80 (0.2435/0.8179)	74.73 (0.2102/0.8497)	77.36 (0.0560/0.9485)	78.20
C-B-B	70.85 (0.2714/0.7942)	77.19 (0.1860/0.8692)	78.75 (0.1059/0.9450)	79.30
ResNet50-C-C	78.17 (0.2068/0.8437)	73.36 (0.2296/0.8333)	76.87 (0.0629/0.9784)	77.31
X-C-C	-	74.05 (0.2188/0.8474)	77.42 (0.0626/0.9813)	77.82
X-B-B	-	76.55 (0.1989/0.8605)	78.58 (0.1138/0.9395)	79.11

jectives through the exits. This case is denoted as 'Self' in Table 1, which is outperformed by the existing exit structures for several main networks and by our proposed structure for all main networks. Thus, different initializations do not provide a sufficient regularization effect. In addition, the structure of the exits becomes excessively large, which is inefficient in terms of memory complexity and computational complexity.

3.1 Diversity of Exits

We perform further analysis on how the diversity of exits, achieved by their structures, influences the classification performance. We employ two measures for diversity of each exit. One is the prediction disagreement, which is defined as the

ratio of the number of test samples that the exit classifier and the main network classifier classify differently [8]. The other is the cosine similarity between the predictions of the exit classifier and the main network classifier [7]. Table 2 shows the test accuracy and the diversity of each classifier when different exit structures are used with ResNet18 as the main network for CIFAR-100. Rows 1 to 4 compare the existing and proposed structures. In addition, we examine several variations on the block type of Exit3 (rows from 5 to 7) or Exit1 (rows from 8 to 12).

When the existing and proposed exit structures are compared, it is observed that high diversity of Exit3 leads to improving the accuracy of the main network (rows 1 to 4). The classifier obtained by Exit3 shares most of the layers with the main network, thus they inherently tend to produce highly similar outputs. Therefore, in order for Exit3 to act as a proper regularizer, its structure needs to be different from that of the main network so that it can provide meaningful additional information to improve the performance of the main network. Even though when we use the bottleneck structure for Exit1 and Exit2, if Exit3 uses the basic block structure (B-B-C), the accuracy of the main network becomes lower due to the low diversity of Exit3. In contrast, even if Exit1 and Exit2 use the basic block structure, using the bottleneck block (C-C-B) or even a large ResNet (C-C-ResNet50) allows Exit3 to have relatively high diversity and thus can improve the performance of the main network.

In the case of the earlier exits (Exit1 and Exit2), their diversity is already high since they share only small numbers of layers with the main network. Thus, their diversity is not necessarily correlated to the performance of the main network. For instance, when different structures for Exit1 are used while Exit2 and Exit3 are kept as the basic block structure (B-C-C and ResNet50-C-C), the accuracy of the main network does not change much. And, changing the structure of Exit1 (from C-C-C to B-C-C or B-B-B to C-B-B) does not change the accuracy of the main network much, either. Nevertheless, without Exit1, the main network does not achieve high performance (X-C-C and X-B-B).

In summary, the key to improving the performance of the main network is to set the structure of the exit at the later stage different from that of the main network (e.g., bottleneck structure for ResNet18). And, the proposed bottleneck exit is a reasonable choice for all exits when the performance and compactness are considered.

4 Ensemble Classifier Distillation

For online distillation using a multi-exit architecture, distillation losses are used together with the classification loss given by (2). The predictions (q_k) and/or feature information (f_k) can be used as teacher signals. Thus, the joint distillation loss L_D is generally written as

$$L_D = \sum_{i=1}^{k-1} \left\{ \alpha KL(q_i, q_k) + \beta \|f_i - f_k\|_2^2 \right\}, \tag{3}$$

where KL is the Kullback-Leibler (KL) divergence and α and β are coefficients. Thus, the overall loss is given by

$$L = \sum_{i=1}^{k} CE(q_i, y) + \sum_{i=1}^{k-1} \Big\{ \alpha KL(q_i, q_k) + \beta \|f_i - f_k\|_2^2 \Big\}. \tag{4}$$

The joint distillation loss (3) assumes that the outputs of the main network can act as a teacher for the exits because the performance of the main network is usually higher than that of the exits.

However, we pay attention to the potential advantage of an ensemble teacher [1]. In the multi-exit architecture, we can inherently obtain multiple outputs (from the exit classifiers and the main network) simultaneously, which can form an ensemble teacher. Moreover, the classifiers in the multi-exit architecture have significant diversity as shown in the previous section due to their structural differences, which maximizes the infomativeness of the ensemble [41]. Thus, we exploit this superior ensemble knowledge, which can be transferred to not only the exits but also the main network.

To construct an effective ensemble teacher, we propose a non-uniform strategy using an importance coefficient λ to reflect the logit of each exit at a different rate, i.e.,

$$z_E = \frac{1}{\sum_{i=1}^{k} \lambda^{i-1}} \sum_{i=1}^{k} \Big(\lambda^{i-1} \cdot z_i \Big), \tag{5}$$

where $\lambda > 1$. The idea is to allow high-performing exits at the later stage to contribute more in order to obtain a teacher of good quality while the exits at the earlier stage mainly enhance diversity in the ensemble teacher at appropriate levels.

Using z_E, the prediction ensemble teacher is defined as

$$q_E = \frac{\exp(z_E/T)}{\sum_j \exp(z_E/T)}. \tag{6}$$

Similarly, we define an ensemble teacher for feature distillation as

$$f_E = \frac{1}{\sum_{i=1}^{k} \lambda^{i-1}} \sum_{i=1}^{k} \Big(\lambda^{i-1} \cdot f_i \Big). \tag{7}$$

Thus, our new joint distillation loss is written as

$$L_D = \sum_{i=1}^{k} \Big\{ \alpha KL(q_i, q_E) + \beta \|f_i - f_E\|_2^2 \Big\}, \tag{8}$$

which applies the distillation mechanism to both the exits and the main network.

Finally, the overall loss becomes

$$L = \sum_{i=1}^{k} \Big\{ CE(q_i, y) + \alpha KL(q_i, q_E) + \beta \|f_i - f_E\|_2^2 \Big\}. \tag{9}$$

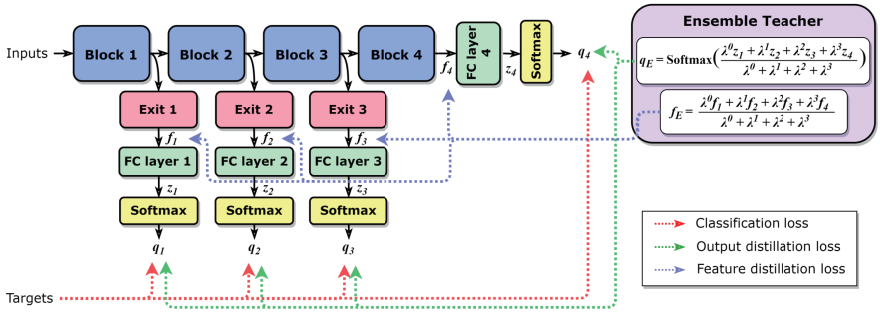


Fig. 3. Illustration of the proposed method. f_i, z_i, and q_i mean the features, logits, and predictions, respectively.

The whole network including the main network and the attached exits is trained using this training loss from scratch as depicted in Figure 3. We call this ensemble distillation *Exit-Ensemble Distillation* (EED), which will be further improved in the following section.

5 Loss-Balancing

In general, if the size of the network is not sufficiently large, the characteristics of the data cannot be learned properly through the classification loss, and instead, shortcuts are learned [9], which yield degraded generalization performance. Therefore, in multi-exit architectures, since the sizes of the early exits are relatively small, they are more difficult to learn proper features if they depend more on the classification loss during training. In this case, a distillation loss can be of help since it additionally provides the information of the non-true classes [33].

In particular, we propose to control relative contributions of the classification loss and the distillation loss for each classifier in a way that an exit at the early stage is trained more with the distillation loss than the classification loss. Therefore, the total loss using this loss-balancing method can be expressed as

$$L = \sum_{i=1}^{k} \left\{ (1 + \alpha - \gamma^{k-i}) \cdot CE(q_i, y) + \gamma^{k-i} \cdot KL(q_i, q_E) + \beta \| f_i - f_E \|_2^2 \right\} \quad (10)$$

where γ is a balancing constant satisfying $\gamma > 1$ and $\gamma^{k-1} < 1 + \alpha$. We call this *Balanced EED* (BEED), which is the final proposed method. Note that the coefficient for feature distillation (β) is fixed as a small value for simplicity, since we found that its contribution to the overall learning is only secondary.

Table 3. Test accuracy (%) of the main network trained by different methods using multi-exits for CIFAR-100.

Network	CE	KD	EED	BEED
ResNet18	79.39	79.21	80.03	**80.58**
ResNet34	80.22	80.17	81.61	**81.62**
WRN16-4	77.49	77.75	78.26	**78.51**
WRN28-4	80.05	80.06	80.55	**80.93**
MobileNet-V2	73.73	73.68	76.63	**76.74**
EfficientNetB0	74.21	74.44	77.47	**77.62**
MSDNet	74.44	75.13	74.63	**75.79**

Table 4. Test accuracy (%) (top-1 / top-5) of the main network trained by different methods using multi-exits for ImageNet.

Network	CE	KD	BEED
ResNet18	69.91 / 88.75	70.12 / 89.14	70.28 / 89.50
ResNet34	73.13 / 91.30	73.75 / 91.66	73.96 / 91.75

6 Experiments

We evaluate our proposed BEED in comparison to existing methods for multi-exits and other online distillation methods using the CIFAR-100 [18] and ImageNet [5] datasets. We use several different CNN architectures composed of residual blocks as main networks such as ResNet [11], WideResNet (WRN) [37], MobileNet-V2 [29], and EfficientNetB0 [32]. MSDNet [14] is also considered, which was specially designed for anytime prediction.

We divide the main network before each residual block containing the convolutional layer having a stride of two, resulting in three or four parts. Then, we insert our proposed bottleneck structure as an exit network between the residual blocks. We use the same number of bottlenecks for an exit as the remaining residual blocks in the main network in order to match the dimension of the feature map for feature distillation. For ResNet, as an example, we use three bottlenecks for the first exit, two for the second exit, and one for the third exit. The number of channels of the bottleneck is the same to that of the corresponding residual block in the main network.

We set α to 1.0 , β to 0.1, γ to 1.15, and λ to 1.6 by default. We conduct all experiments three times with different random seeds and report the average accuracy. Other implementation details and the ablation study of the hyper-parameters tuning are given in the supplementary material.

6.1 Main Network

To prove the effectiveness of our method for training the main network, we compare the test performance of the methods using multi-exits, including the method

Table 5. Test accuracy (%) of the main network trained for CIFAR-100 using online distillation methods.

Network	DML	ONE	DCL	OKDDip	BEED
ResNet18	78.97	78.89	79.58	79.83	**80.58**
ResNet34	78.98	78.84	79.71	79.54	**81.62**
WRN16-4	78.10	78.23	78.25	78.49	**78.51**
WRN28-4	80.35	80.67	80.71	80.89	**80.93**

Table 6. Test accuracy (%) (and GFLOPs required for one feedforward pass) of ensemble inference for CIFAR-100. The ensemble of the four classifiers (three exits and main network) trained by our BEED is compared to the case when four independent networks having the same structure are trained and used for ensembling.

Network	Indep. ens.	BEED ens.
ResNet18	80.56 (2.24)	81.45 (0.86)
ResNet34	80.80 (4.64)	82.50 (1.47)
MobileNet-V2	75.94 (0.40)	78.86 (0.27)
EfficientNetB0	76.58 (0.48)	79.36 (0.32)

in [34] using (2) (denoted by CE), the method in [40] using (4) (denoted by KD), and the proposed BEED method using (10). We also show the performance of EED using (9) to verify the effectiveness of our loss-balancing.

The results for CIFAR-100 are shown in Table 3. Our BEED method achieves the best performance for all networks. The KD method is not always better than CE, but using an ensemble teacher on top of KD (i.e., EED) brings clear performance improvement. Especially, the performance of MoblieNet-V2 and EfficientNetB0 is greatly enhanced by the ensemble teacher (73.68% $\rightarrow$ 76.63% and 74.44% $\rightarrow$ 77.47%, respectively). In addition, our loss-balancing strategy with EED (i.e., BEED) enhances the performance further. With MSDNet, EED does not yield performance gain compared to KD, but by using BEED for proper loss-balancing, performance improvement is obtained (75.13% $\rightarrow$ 75.79%). Our BEED also achieves better performance than CE and KD for ImageNet as shown in Table 4.

In addition, we compare the performance of our BEED with that of the representative online distillation methods for image classification [4] (deep mutual learning (DML) [42] and on-the-fly native ensemble (ONE) [19]) and recent methods (deep collaborative learning (DCL) [26] and online knowledge distillation with diverse peers (OKDDip) [2]). As shown in Table 5, BEED achieves better performance than all methods with large performance gaps.

6.2 Ensemble Inference

Although we originally used the multi-exits to improve the performance of the main network, we can also use them to perform ensemble inference. As discussed

Table 7. Test accuracy (%) of the main network with data augmentation for CIFAR-100.

Network	Aug.	KD	FRSKD	BEED
ResNet18	No aug.	79.21	77.88	80.58
	Mixup	80.32	78.83	81.12
	Cutout	80.16	79.80	80.89
	CutMix	**80.81**	81.22	**81.37**
	SLA	80.68	**81.25**	80.75
ResNet34	No aug.	80.17	77.02	81.62
	Mixup	81.01	78.37	81.57
	Cutout	81.82	80.08	82.75
	CutMix	**82.47**	81.16	**83.13**
	SLA	82.12	**81.54**	82.14

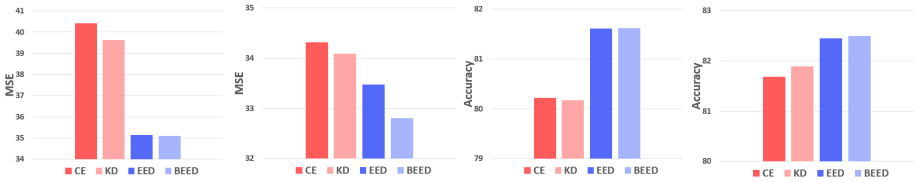
in Section 3, our design of the structure of the exits aimed to enhance diversity. We can also make good use of this diversity to form a strong ensemble for ensemble inference.

Table 6 compares the performance of the ensemble formed by all classifiers (exits and main network) in the mult-exit structure trained by BEED and the ensemble of independently trained main networks, showing that our BEED achieves better ensemble performance. Note that the performance of BEED in this case is significantly higher than that in Table 3 (e.g., 77.62% $\rightarrow$ 79.36% for EfficientNetB0); in other words, when a multi-exit structure is trained by our BEED, ensembling the obtained classifiers further enhances the classification performance. Besides, the FLOPs required for ensemble inference is significantly reduced in BEED because the exit classifiers are smaller than the main network. In addition, even when we compare the ensemble performance of BEED to that of CE and KD, our method achieves better performance, which is shown in the supplementary material. Thus, our BEED is a good option even when ensemble inference is considered.

6.3 Performance with Data Augmentation

The recent online distillation method, feature refinement via self-knowledge distillation (FRSKD) [15], showed that data augmentation can improve performance of online distillation. Thus, we evaluate the performance of our BEED when applying the popular data augmentation methods, including Mixup [38], Cutout [6], CutMix [36], and self-supervised label augmentation (SLA) [20]. For Mixup, Cutout, and CutMix, we apply them to all exits, but SLA is applied only to the main network due to the excessive complexity of applying it to all exits.

The results in Table 7 show that CutMix is effective for KD and BEED, and SLA is effective for FRSKD. When the performance of the best data augmentation strategy in each method is compared, our BEED with CutMix outperforms the other methods for both ResNet18 and ResNet34.



(a) MSE (main network) (b) MSE (ensemble) (c) Acc. (main network) (d) Acc. (ensemble)

Fig. 4. MSE and test accuracy (%) of different methods with ResNet34 for CIFAR-100.

6.4 Good Teacher

In a recent study [25], it was found that teachers with low mean square error (MSE) between the output probabilities and the one-hot labels produce better students via distillation. Thus, we compare the MSE of the teacher in each method to verify whether our ensemble teacher is good in this criterion.

In Figure 4, we show the MSE of the main network output (a) and ensemble output (b) with their accuracy (c and d). Overall, there exists a tendency that the lower the MSE is, the higher the accuracy is. The teacher in KD is the main network and the teacher in BEED is the ensemble output. Thus, when we compare the MSE for KD in Figure 4a and the MSE for BEED in Figure 4b, the latter (32.80) is much lower than the former (39.62). The ensemble teacher in BEED is the best teacher showing the smallest MSE among all cases (Figures 4a and 4b), which results in the highest accuracy in Figures 4c and 4d.

7 Conclusion

We proposed a new online knowledge distillation method using auxiliary classifiers (exits), called BEED. Our method is based on the selection of the structure of the exits to promote diversity, the newly proposed ensemble distillation method to obtain an improved teacher signal, and the new loss-balancing strategy to control the contributions of different losses. The experimental results showed that our method outperforms the existing online distillation methods. Further improvement was achieved by ensemble inference and data augmentation.

Acknowledgements This work was supported by the Artificial Intelligence Graduate School Program, Yonsei University under Grant 2020-0-01361.

References

1. Asif, U., Tang, J., Harrer, S.: Ensemble knowledge distillation for learning improved and efficient networks. In: Proceedings of the European Conference on Artificial Intelligence (ECAI) (2020)

2. Chen, D., Mei, J.P., Wang, C., Feng, Y., Chen, C.: Online knowledge distillation with diverse peers. In: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI). pp. 3430–3437 (2020)
3. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling knowledge via knowledge review. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5006–5015 (2021)
4. Chung, I., Park, S., Kim, J., Kwak, N.: Feature-map-level online adversarial knowledge distillation. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 119, pp. 2006–2015 (2020)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
6. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
7. Dvornik, N., Mairal, J., Schmid, C.: Diversity with cooperation: Ensemble methods for few-shot classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3722–3730 (2019)
8. Fort, S., Hu, H., Lakshminarayanan, B.: Deep ensembles: A loss landscape perspective. arXiv preprint arXiv:1912.02757 (2019)
9. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**, 665âĂŞ673 (2020)
10. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**, 1789–1819 (2021)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. Las Vegas, Nevada (2016)
12. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1921–1930 (2019)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Proceedings of the Neural Information Processing Systems (NeurIPS) Workshop (2014)
14. Huang, G., Chen, D., Li, T., Wu, F., v. d. Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
15. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10659–10668 (2021)
16. Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review pp. 1 – 62 (2020)
17. Kim, Y., Park, J., Jang, Y., Ali, M., Oh, T.H., Bae, S.H.: Distilling global and local logits with densely connected relations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6290–6300 (2021)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
19. Lan, X., Zhu, X., Gong, S.: Knowledge distillation by on-the-fly native ensemble. In: Proceedings of the Neural Information Processing Systems (NeurIPS). pp. 7528–7538 (2018)

20. Lee, H., Hwang, S.J., Shin, J.: Self-supervised label augmentation via input transformations. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 119, pp. 5714–5724 (2020)
21. Lee, H., Lee, J.S.: Local critic training of deep neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN) (2019)
22. Li, G., Zhang, J., Wang, Y., Liu, C., Tan, M., Lin, Y., Zhang, W., Feng, J., Zhang, T.: Residual distillation: Towards portable deep neural networks without shortcuts. In: Proceedings of the Neural Information Processing Systems (NeurIPS). vol. 33, pp. 8935–8946 (2020)
23. Li, H., Zhang, H., Qi, X., Ruigang, Y., Huang, G.: Improved techniques for training adaptive deep networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1891–1900 (2019)
24. Liu, B., Rao, Y., Lu, J., Zhou, J., Hsieh, C.J.: Metadistiller: Network self-boosting via meta-learned top-down distillation. In: Proceedings of the European Conference on Computer Vision (ECCV). vol. 12359, pp. 694–709 (2020)
25. Menon, A.K., Rawat, A.S., Reddi, S., Kim, S., Kumar, S.: A statistical perspective on distillation. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 139, pp. 7632–7642 (2021)
26. Minami, S., Hirakawa, T., Yamashita, T., Fujiyoshi, H.: Knowledge transfer graph for deep collaborative learning. In: Proceedings of the Asian Conference on Computer Vision (ACCV). p. 203âĂŞ217 (2020)
27. Phuong, M., Lampert, C.: Distillation-based training for multi-exit architectures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1355–1364 (2019)
28. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: Hints for thin deep nets. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4510–4520 (2018)
30. Song, J., Chen, Y., Ye, J., Song, M.: Spot-adaptive knowledge distillation. IEEE Transactions on Image Processing **31**, 3359–3370 (2022)
31. Song, J., Zhang, H., Wang, X., Xue, M., Chen, Y., Sun, L., Tao, D., Song, M.: Tree-like decision distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13488–13497 (2021)
32. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 97, pp. 6105–6114 (2019)
33. Tang, J., Shivanna, R., Zhao, Z., Lin, D., Singh, A., Chi, E.H., Jain, S.: Understanding and improving knowledge distillation. arXiv preprint arXiv:2002.03532 (2020)
34. Teerapittayanon, S., McDanel, B., Kung, H.T.: BranchyNet: Fast inference via early exiting from deep neural networks. In: Proceedings of the International Conference on Pattern Recognition (ICPR). pp. 2464–2469 (2016)
35. Yao, A., Sun, D.: Knowledge transfer via dense cross-layer mutual-distillation. In: Proceedings of the European Conference on Computer Vision (ECCV). vol. 12360, pp. 294–311 (2020)
36. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6022–6031 (2019)

37. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC) (2016)
38. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
39. Zhang, L., Shi, Y., Shi, Z., Ma, K., Bao, C.: Task-oriented feature distillation. In: Proceedings of the Neural Information Processing Systems (NeurIPS). vol. 33, pp. 14759–14771 (2020)
40. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3712–3721 (2019)
41. Zhang, S., Liu, M., Yan, J.: The diversified ensemble neural network. In: Proceedings of the Neural Information Processing Systems (NeurIPS). vol. 33, pp. 16001–16011 (2020)
42. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
43. Zhu, Y., Wang, Y.: Student customized knowledge distillation: Bridging the gap between student and teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5057–5066 (2021)