Supplementary Material for "Knowledge Transfer via Dense Cross-Layer Mutual-Distillation"

Anbang Yao*✉ and Dawei Sun*

Intel Labs China
{anbang.yao,dawei.sun}@intel.com

1 Experimental Settings

In this section, we provide detailed settings of the experiments conducted on the CIFAR-100 and ImageNet datasets.

1.1 Experimental Settings on CIFAR-100

As stated in Section 4.1 of the main paper, we consider two different scenarios on the CIFAR-100 dataset when jointly training two CNNs from scratch: (1) Two CNNs with the same backbone (e.g., WRN-28-10 & WRN-28-10); (2) Two CNNs with the different backbones (e.g., WRN-28-10 & ResNet-110). Generally, we follow the same settings as reported in the original papers [1,4,8,9]. Here, we first describe the training hyper-parameters in the first scenario. At the training phase, for ResNet-110/ResNet-164 and MobileNet, we use SGD with momentum, and we set the batch size as 64, the weight decay as 0.0001, the momentum as 0.9 and the number of training epochs as 200. The initial learning rate is 0.1, and it is divided by 10 every 60 epochs. For DenseNet-40-12, we use SGD with Nesterov momentum, and we set the batch size as 64, the weight decay as 0.0001, the momentum as 0.9 and the number of training epochs as 300. The initial learning rate is set to 0.1, and is divided by 10 at 50% and 75% of the total number of training epochs. For WRN-28-10/WRN-28-4, we use SGD with momentum, and we set the batch size as 128, the weight decay as 0.0005, the momentum as 0.9 and the number of training epochs as 200. The initial learning rate is set to 0.1, and is divided by 5 at 60, 120 and 160 epochs. In the second scenario of jointly training two CNNs with the different backbones, we use the training hyper-parameters of WRN-28-10 to train both networks.

1.2 Experimental Settings on ImageNet

On the ImageNet dataset, we use popular ResNet-18/ResNet-50 [1] and MobileNetV2 [5] as the backbone networks, and consider two training scenarios

* Equal contribution. ✉ Corresponding author. Experiments were mostly done by Dawei Sun when he was an intern at Intel Labs China, supervised by Anbang Yao.

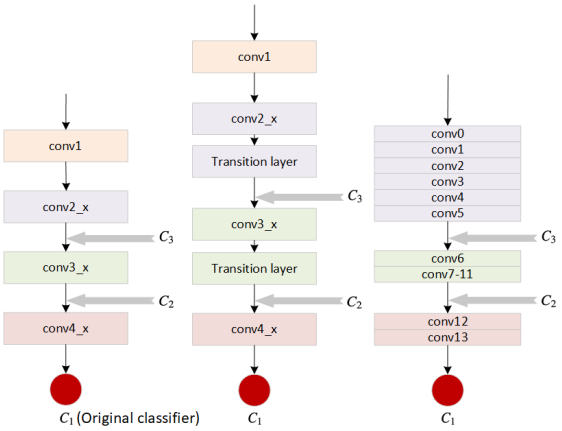

Fig. 1. Locations of the auxiliary classifiers added to the CNN architectures evaluated on the CIFAR-100 dataset. The left figure is for ResNet-110/ResNet-164 and WRN-28-10/WRN-28-4, and the middle figure is for DenseNet-40-12, and the right figure is for MobileNet. The grey thick arrows indicate the locations where auxiliary classifiers are added. For a backbone network, we denote the original classifier as C_1, and denote two auxiliary classifiers as C_2 and C_3 respectively.

which are the same as on the CIFAR-100 dataset. For all these CNN backbones, we use the same settings as reported in the original papers. For the experiments with ResNet backbones, two models with either the same depth configuration or the different depth configurations are trained with SGD for 100 epochs. We set the batch size as 256, the weight decay as 0.0001 and the momentum as 0.9. The learning rate starts at 0.1, and is divided by 10 every 30 epochs. For the experiment with MobileNetV2 backbone, two models with the same configuration are trained with SGD for 150 epochs using batch size 256. The momentum is set as 0.9 and the weight decay is set as 4e-5. The learning rate initiates from 0.05 and declines with a cosine function shaped decay strategy approximating to zero.

2 Structures of Auxiliary Classifiers

On the CIFAR-100 dataset, we test several kinds of prevailing CNN architectures including ResNet-110/ResNet-164 [1], DenseNet-40-12 [4], WRN-28-10/WRN-28-4 [8] and MobileNet [2]. As stated in the main paper, we append well-designed auxiliary classifiers on top of certain different-staged down-sampling layers of a CNN backbone when applying our method to CIFAR-100 and ImageNet datasets. In this section, we provide the structures of our auxiliary classifiers.

2.1 Auxiliary Classifiers for CIFAR-100

In this sub-section, we describe the auxiliary classifiers used in the CIFAR-100 experiments.

Table 1. Details of the convolutional blocks of the auxiliary classifiers added to the ResNet backbones evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding number of output channels.

	ResNet-110		ResNet-164	
	C_3	C_2	C_3	C_2
conv1	-	-	-	-
conv2_x	-	-	-	-
conv3_x	$\begin{bmatrix} 3\times3,\ 32 \\ 3\times3,\ 32 \end{bmatrix} \times 9$	-	$\begin{bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128 \\ 1\times1,\ 128 \end{bmatrix} \times 9$	-
conv4_x	$\begin{bmatrix} 3\times3,\ 64 \\ 3\times3,\ 64 \end{bmatrix} \times 9$	$\begin{bmatrix} 3\times3,\ 128 \\ 3\times3,\ 128 \end{bmatrix} \times 18$	$\begin{bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256 \\ 1\times1,\ 256 \end{bmatrix} \times 9$	$\begin{bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512 \\ 1\times1,\ 512 \end{bmatrix} \times 18$

Table 2. Details of the convolutional blocks of the auxiliary classifiers added to the DenseNet backbone evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding growth rate.

	DenseNet-40-12	
	C_3	C_2
conv1	-	-
conv2_x	-	-
conv3_x	$\begin{bmatrix} 3\times3,\ 12 \end{bmatrix} \times 12$	-
conv4_x	$\begin{bmatrix} 3\times3,\ 12 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times3,\ 36 \end{bmatrix} \times 12$

Locations. In the experiments, we add 2 auxiliary classifiers to every backbone network. We denote the original classifier (i.e., the top-most classifier added to the last layer of a backbone network) as C_1 and the auxiliary classifiers as C_2 and C_3 as shown in Fig. 1.

Structures. In each joint training experiment, every auxiliary classifier is composed of the same building block (e.g., residual block in ResNet) as in the backbone network. The differences lie in the numbers and parameter sizes of convolutional layers. As empirically verified in [3,10,6], early layers lack coarse-level features which are helpful for image-level classification. In order to address this problem, we use a heuristic principle making the paths from the input to all classifiers have the same number of down-sampling layers. We detail the hyper-parameter settings of the convolutional layers of auxiliary classifiers w.r.t. different backbone networks in Table 1, Table 2, Table 3 and Table 4 respectively.

2.2 Additional Auxiliary Classifiers regarding Experiments for Analyzing How to Set $\boldsymbol{Q}$.

As stated in Section 4.3 of the main paper, in order to analyze how to set $\boldsymbol{Q}$ in our method, we first conduct ablative experiments to jointly train two WRN-28-10 models considering different settings by adding auxiliary classifiers to at most three down-sampling layers. Besides two basic auxiliary classifiers C_2 and C_3 used in our DCM, one additional auxiliary classifier C_4 is added after "conv1"

Table 3. Details of the convolutional blocks of the auxiliary classifiers added to the WRN backbones evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding number of output channels. C_4 is used as an additional auxiliary classifier for the analysis of how to set $\boldsymbol{Q}$.

	WRN-28-4		WRN-28-10		
	C_3	C_2	C_4	C_3	C_2
conv1	-	-	-	-	-
conv2_x	-	-	$\begin{bmatrix} 3\times3,\ 160 \\ 3\times3,\ 160 \end{bmatrix} \times 4$	-	-
conv3_x	$\begin{bmatrix} 3\times3,\ 128 \\ 3\times3,\ 128 \end{bmatrix} \times 4$	-	$\begin{bmatrix} 3\times3,\ 320 \\ 3\times3,\ 320 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 320 \\ 3\times3,\ 320 \end{bmatrix} \times 4$	-
conv4_x	$\begin{bmatrix} 3\times3,\ 256 \\ 3\times3,\ 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 512 \\ 3\times3,\ 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3,\ 640 \\ 3\times3,\ 640 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 640 \\ 3\times3,\ 640 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 1280 \\ 3\times3,\ 1280 \end{bmatrix} \times 4$

Table 4. Details of the convolutional blocks of the auxiliary classifiers added to the MobileNet backbone evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of convolutional blocks and the number of output channels, and $s2$ denotes the stride of the convolution operation in this layer is 2. Each convolutional block is composed of a 3×3 depthwise convolution and a 1×1 pointwise convolution. Please see Fig. 1 for different layer locations.

	conv0	conv1	conv2	conv3	conv4	conv5	conv6	conv7-11	conv12	conv13
C_3	-	-	-	-	-	-	$\begin{bmatrix} 3\times3,\ 256s2 \\ 1\times1,\ 512 \end{bmatrix}$	$\begin{bmatrix} 3\times3,\ 512 \\ 1\times1,\ 512 \end{bmatrix}$	$\begin{bmatrix} 3\times3,\ 512s2 \\ 1\times1,\ 1024 \end{bmatrix}$	$\begin{bmatrix} 3\times3,\ 1024 \\ 1\times1,\ 1024 \end{bmatrix}$
C_2	-	-	-	-	-	-	-	-	$\begin{bmatrix} 3\times3,\ 512s2 \\ 1\times1,\ 2048 \end{bmatrix}$	$\begin{bmatrix} 3\times3,\ 2048 \\ 1\times1,\ 2048 \end{bmatrix}$

layer of WRN-28-10 as illustrated in Fig. 1, and its hyper-parameter setting of the convolutional layers is provided in Table 3.

As for two additional kinds of auxiliary classifiers evaluated by DCM, namely "APFC" and "Narrow", their structures are: (1) "APFC" is composed of an average pooling layer (with the spatial output size of 4×4), a fully connected layer (with the output size of 100) and a softmax function; (2) "APFC" refers to narrower versions whose growth rate (for DenseNet-40-12) or width (for WRN-28-10) values are at half of our basic designs shown in Table 2 and Table 3 respectively.

2.3 Auxiliary Classifiers for ImageNet

On the ImageNet dataset, we use popular ResNet-18/ResNet-50 and MobileNetV2 as the backbone networks. In this sub-section, we describe their respective auxiliary classifiers.

Locations. The locations of the auxiliary classifiers for ResNet-18/ResNet-50 and MobileNetV2 are shown in Fig. 2 respectively.

Structures. The auxiliary classifiers added to all backbone networks have the same macro-structure. Generally, we design these auxiliary classifiers with the

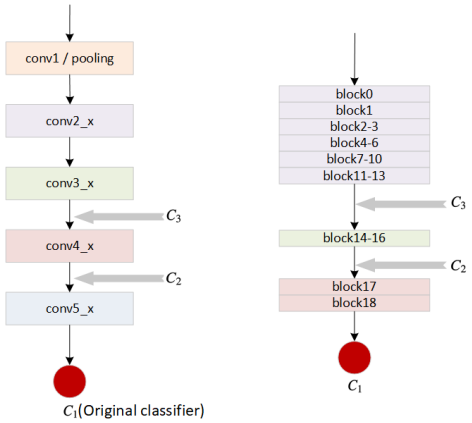

Fig. 2. Locations of the auxiliary classifiers added to the CNN architectures evaluated on the ImageNet dataset. The left figure is for ResNet-18/ResNet-50, and the right figure is for MobileNetV2. The grey thick arrows indicate the locations where auxiliary classifiers are added. For a backbone network, we denote the original classifier as C_1, and two auxiliary classifiers as C_2 and C_3 respectively.

same building blocks as the backbone network. To guarantee that all the paths from the input to different classifier outputs have the same down-sampling process, we design the auxiliary classifiers according to the corresponding building blocks in the backbone network. Taking ResNet-18/ResNet-50 as an example, the auxiliary classifier C_3 has its own conv_4x and conv_5x acting as down-sampling stages, whose parameter size is smaller than that of the corresponding stages in the backbone network. After these down-sampling stages, there are also a global average pooling layer and a fully connected layer. Auxiliary classifiers for MobileNetV2 are designed with the same principles. We show the details of the convolutional blocks of the auxiliary classifiers for these backbone networks in Table 5 and Table 6 respectively.

3 Advantages over KD and Its Variants

Recall that in this paper we focus on improving two-way knowledge transfer design. Compared with conventional one-way KD and it variants (mostly relying on hidden layer feature/attention distillation), our method has new properties: (1) collaborative training of two models from scratch (no need of the pre-trained and fixed teacher); (2) bidirectional cross-layer knowledge transfer (a smaller model also improves a large model); (3) soft hidden layer knowledge obtained in a supervised manner (no need of the layer-wise feature/attention matching usually conducted with multi-step strategies due to different hidden layer map dimensions between two models). Note that a direct performance comparison of two-way DCM with one-way KD based methods is not fair due to their quite

Table 5. Details of the convolutional blocks of the auxiliary classifiers added to the ResNet backbones evaluated on the ImageNet dataset. In the table, every cell shows the corresponding number of convolutional blocks (including basic blocks for ResNet-18 and bottleneck blocks for ResNet-50) and their parameter sizes. For comparison with the backbone networks, please refer to the Table 1 of the ResNet paper [1].

	ResNet-18		ResNet-50	
	C_3	C_2	C_3	C_2
conv1	-	-	-	-
conv2_x	-	-	-	-
conv3_x	-	-	-	-
conv4_x	$\begin{bmatrix} 3\times 3,\ 256 \\ 3\times 3,\ 256 \end{bmatrix} \times 1$	-	$\begin{bmatrix} 1\times 1,\ 256 \\ 3\times 3,\ 256 \\ 1\times 1,\ 1024 \end{bmatrix} \times 3$	-
conv5_x	$\begin{bmatrix} 3\times 3,\ 512 \\ 3\times 3,\ 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3,\ 1024 \\ 3\times 3,\ 1024 \end{bmatrix} \times 2$	$\begin{bmatrix} 1\times 1,\ 512 \\ 3\times 3,\ 512 \\ 1\times 1,\ 2048 \end{bmatrix} \times 2$	$\begin{bmatrix} 1\times 1,\ 1024 \\ 3\times 3,\ 1024 \\ 1\times 1,\ 4096 \end{bmatrix} \times 3$

Table 6. Details of the separable convolutional blocks of the auxiliary classifiers added to the MobileNetV2 backbone evaluated on the ImageNet dataset. In the table, every cell shows the parameter configuration of separable convolutional blocks, and (t,c,n,s) denotes the expansion factor, the number of output channels, the repeated times of bottleneck unit and the stride respectively. For comparison with the backbone network, please refer to the Table 2 of the MobileNetV2 paper [5].

	block0	block1	block2-3	block4-6	block7-10	block11-13	block14-16	block17	block18
C_3	-	-	-	-	-	-	(6,160,3,2)	(3,320,1,1)	(-,1280,1,1)
C_2	-	-	-	-	-	-	-	(6,480,1,1)	(-,1920,1,1)

different training paradigms. Recently, a comprehensive benchmark of fourteen state of the art KD based methods on the CIFAR-100 dataset was published in [7] from which we can observe: Merely considering the accuracy gain to the smaller student model regardless of training paradigm differences, our results reported in the main paper are mostly better than those one-way KD based methods using the fixed teacher model.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
2. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
3. Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. In: ICLR (2018)
4. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
5. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR (2018)
6. Sun, D., Yao, A., Zhou, A., Zhao, H.: Deeply-supervised knowledge synergy. In: CVPR (2019)
7. Tian, y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2020)
8. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)
9. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: CVPR (2018)
10. Zhang, Z., Zhang, X., Peng, C., Cheng, D., Sun, J.: Exfuse: Enhancing feature fusion for semantic segmentation. In: ECCV (2018)