Deeply-supervised Knowledge Synergy

Supplementary Materials

Dawei Sun[1,2*] Anbang Yao[1*] Aojun Zhou[1] Hao Zhao[1,2]
[1]Intel Labs China [2]Tsinghua University
{dawei.sun, anbang.yao, aojun.zhou, hao.zhao}@intel.com

1. Theoretical Insights

In this section, we give some theoretical insights about why the synergy mechanism in our DKS method shows better performance than the Deeply-Supervised (DS) learning method [5, 7]. For simplicity, we focus on the optimization of a CNN regression model with one auxiliary branch. Inspired by [6], we give a formal proof for that the pairwise synergy term behaves as a regularizer which penalizes the inconsistency between the gradients of the two branches w.r.t. their shared intermediate feature map. Such a proof can be generalized to the optimization of a deep CNN classification model.

Suppose we need to train a CNN model to fit target data distribution $x, y \sim D$ where x denotes the input and y denotes the expected output. The model has two output heads, the top-most one $\hat{y}_1$ and the auxiliary one $\hat{y}_2$ as illustrated in Fig. 1. The forward process is as follows, $z = f(x)$, $\hat{y}_1 = g_1(z)$, $\hat{y}_2 = g_2(z)$, where z denotes the intermediate feature map. In DS configuration, the loss function used to guide the training process can be written as

$$L_{DS} = \mathop{\mathbb{E}}_{x,y\sim D,\ \epsilon} \left(\frac{1}{2} \left|\left| g_1(z+\epsilon) - y \right|\right|^2 + \frac{1}{2} \left|\left| g_2(z+\epsilon) - y \right|\right|^2 \right),$$

where ϵ denotes the random perturbations on the intermediate feature map caused by data augmentation. Here, we assume $\mathbb{E}(\epsilon) = 0$, $\mathbb{E}(\epsilon^2) = \sigma^2$. In DKS configuration, the loss function can be written as

$$L_{DKS} = \mathop{\mathbb{E}}_{x,y\sim D,\ \epsilon} \left(\frac{1}{2} \left|\left| g_1(z+\epsilon) - y \right|\right|^2 + \frac{1}{2} \left|\left| g_2(z+\epsilon) - y \right|\right|^2 + \frac{1}{2} \left|\left| g_1(z+\epsilon) - g_2(z+\epsilon) \right|\right|^2 \right).$$

*Equal contribution. This work was done when Dawei Sun was an intern at Intel Labs China, supervised by Anbang Yao who is responsible for correspondence. Interns Aojun Zhou and Hao Zhao contributed to early theoretical analysis.

Proposition 1. *The synergy term in DKS guarantees more consistent gradients of the two branches w.r.t. z compared with DS. That is, the synergy term penalizes $\left|\left| \frac{\partial \hat{y}_1}{\partial z} - \frac{\partial \hat{y}_2}{\partial z} \right|\right|^2$.*

Proof. The synergy term in DKS:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x,y\sim D,\ \epsilon} \left(\frac{1}{2} \left|\left| g_1(z+\epsilon) - g_2(z+\epsilon) \right|\right|^2 \right) \\
&= \mathop{\mathbb{E}}_{x,y\sim D,\ \epsilon} \left(\frac{1}{2} \left|\left| g_1(z) + \epsilon \frac{\partial g_1(z)}{\partial z} + O(\epsilon^2) - g_2(z) - \epsilon \frac{\partial g_2(z)}{\partial z} + O(\epsilon^2) \right|\right|^2 \right) \\
&= \frac{1}{2} \mathop{\mathbb{E}}_{x,y\sim D} \left(\left|\left| g_1(z) - g_2(z) \right|\right|^2 \right) \\
&\quad + \mathbb{E}(\epsilon) \mathop{\mathbb{E}}_{x,y\sim D} \left(\left|\left| g_1(z) - g_2(z) \right|\right| \cdot \left|\left| \frac{\partial g_1(z)}{\partial z} - \frac{\partial g_2(z)}{\partial z} \right|\right| \right) \\
&\quad + \frac{1}{2} \mathbb{E}(\epsilon^2) \mathop{\mathbb{E}}_{x,y\sim D} \left(\left|\left| \frac{\partial g_1(z)}{\partial z} - \frac{\partial g_2(z)}{\partial z} \right|\right|^2 \right) + O(\epsilon^4) \\
&= \frac{1}{2} \mathop{\mathbb{E}}_{x,y\sim D} \left(\left|\left| g_1(z) - g_2(z) \right|\right|^2 \right) \\
&\quad + \frac{1}{2} \sigma^2 \mathop{\mathbb{E}}_{x,y\sim D} \left(\left|\left| \frac{\partial g_1(z)}{\partial z} - \frac{\partial g_2(z)}{\partial z} \right|\right|^2 \right) + O(\epsilon^4)
\end{aligned}$$

□

2. Design Details of Auxiliary Classifiers

As we described in the paper, we append carefully designed auxiliary classifiers on top of some intermediate layers of a given backbone network when applying our DKS method to CIFAR-100 and ImageNet classification datasets. In this section, we provide the design details of the auxiliary classifiers.

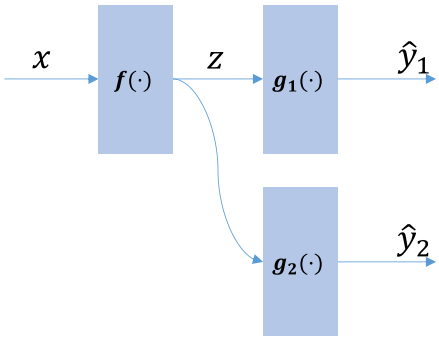


Figure 1: The regression model used in the proof.

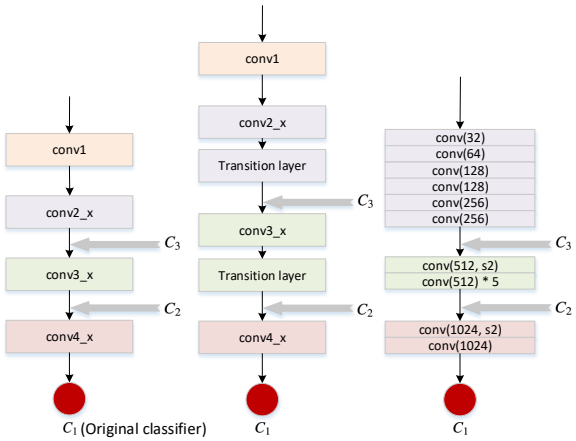


Figure 2: Locations of the auxiliary classifiers added to the backbone networks evaluated on the CIFAR-100 dataset. The left figure is for ResNets and WRNs, and the middle figure is for DenseNets, and the right figure is for MobileNet. The grey thick arrows indicate the locations where auxiliary classifiers are added. We denote these three classifiers as C_1, C_2 and C_3 respectively, where C_1 is the original classifier of the backbone network.

2.1. Auxiliary Classifiers for CIFAR-100

On the CIFAR-100 dataset, we test several kinds of backbone networks including ResNets [1], DenseNets [4], WRNs [8] and MobileNet [2]. In this sub-section, we describe the auxiliary classifiers used in the Section 4.1, the Section 4.3 'Analysis of Auxiliary Classifiers', 'DKS on Very Deep Network', 'DKS with Strong Regularization' and 'DKS on Noisy Data' of our paper, respectively.

2.1.1 Auxiliary Classifiers Used in Section 4.1

Locations. In the experiments, we add two auxiliary classifiers to every backbone network. Their locations for different backbone networks are shown in Fig.2.

Structures. In the experiments, we append relatively complex auxiliary supervision branches on top of certain intermediate layers during network training. Specifically, every auxiliary supervision branch 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], 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 different backbone networks in Table 2, Table 3, Table 4 and Table 5, respectively.

2.1.2 Auxiliary Classifiers Used in Section 4.3 'Analysis of Auxiliary Classifiers'

In order to analyze the impact of the complexity of auxiliary classifiers, we evaluate different auxiliary classifier designs for ResNet-32 in the Section 4.3 'Analysis of Auxiliary Classifiers'.

Locations. The locations are the same as that described in Fig. 2.

Structures. As described in the Table 3 of our paper, we evaluated four more types of auxiliary classifiers besides our final design. AP+2FC refers to one average pooling layer + two fully connected layers. AP+1Conv+2FC refers to one average pooling layer + one convolutional layer + two fully connected layers. Narrow Blocks means the auxiliary classifiers are narrower than the original design (i.e., the top-most classifier connected to the last layer of the backbone network). Shallow Blocks means the auxiliary classifiers are shallower than the original design. Please refer to Fig. 3 and Table 6 for more details.

2.1.3 Auxiliary Classifiers Used in Section 4.3 'DKS on Very Deep Network'

We conduct a set of experiments to analyze the performance of DKS on very deep CNNs. In the experiments, we consider the training of a ResNet variant with 1202 layers [1] on the CIFAR-100 dataset. Unlike auxiliary classifiers used in the other experiments, we study DKS with shallow but wide auxiliary classifiers in the experiments.

Locations. In the experiments, we add three auxiliary classifiers to the ResNet-1202 backbone network. Their lo-

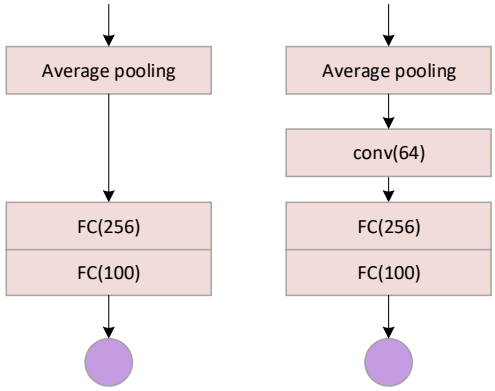


Figure 3: Details of the AP+2FC and AP+1Conv+2FC auxiliary classifiers added to the ResNet-32 backbone network evaluated on the CIFAR-100 dataset. The output size of the average pooling layer is 4×4. The number of output channels of every layer is shown in the parentheses.

	C_2	C_3	C_4
conv1	512	256	128
conv2	-	512	256
conv3	-	-	512

Table 1: Details of the convolutional layers of the auxiliary classifiers added to the ResNet-1202 backbone network evaluated on the CIFAR-100 dataset. In the table, the number in every cell indicates how many filters are in this convolutional layer. For example, C_3 has two convolutional layers where the first layer has 256 filters and the second layer has 512 filters.

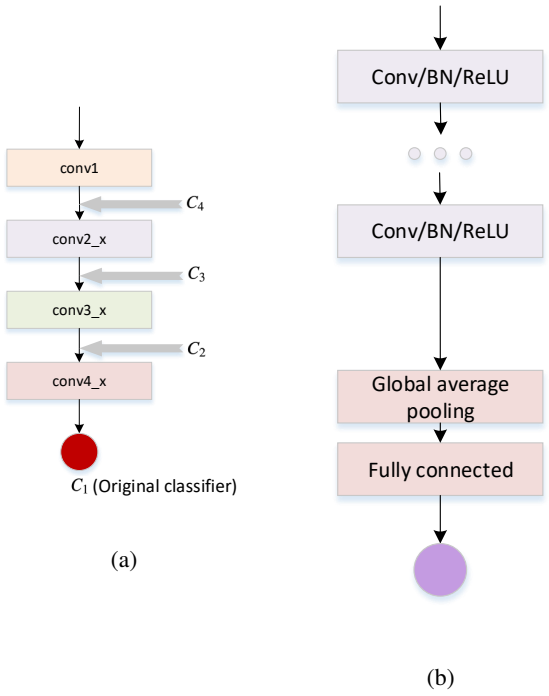
cations are shown in Fig.4a.

Structures. The auxiliary classifiers added to the ResNet-1202 backbone network have the macro-structures defined in Fig.4b. The number of convolutional layers for every auxiliary classifier can be found in Table 1.

2.2. Auxiliary Classifiers for ImageNet

On the ImageNet classification dataset, we use popular ResNet-18, ResNet-50 and ResNet-152 [1] as the backbone networks. In this sub-section, we describe the auxiliary classifiers used in the Section 4.2 and some experiments of Section 4.3 of our paper.

Locations. In the experiments, we add at most 3 auxiliary classifiers to every backbone network. Following the definition in our paper, 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, C_3 and C_4, as shown in Fig.5. Recall that C_4 is an extra auxiliary classifier which is added for analyzing the accuracy effect of the increasing number of the auxiliary classifiers, as described in Table 4 of our paper. For the main experiments in the Section 4.2 of our paper, we add 2 auxiliary classifiers (i.e., C_2 and C_3) to every backbone network.



(a) (b)

Figure 4: (a) Locations of the auxiliary classifiers added to the ResNet-1202 backbone network evaluated on the CIFAR-100 dataset. The grey thick arrows indicate the layer locations where auxiliary classifiers are added. We denote these three auxiliary classifiers as C_2, C_3 and C_4, respectively. (b) Structure of the auxiliary classifiers. All the convolutional layers in this structure have the same kernel size (= 3×3) and the same stride (= 1), but have different number of filters (yielding different number of output channels). The numbers of convolutional layers and the corresponding filters for every auxiliary classifier can be found in Table 1.

Structures. In all of the experiments except for the one regarding Fig.3 'DS with simple aux. classifiers' of our paper, the auxiliary classifiers added to all backbone networks have the same macro-structure. Generally, we design these auxiliary classifiers with the same building blocks as the backbone network. To guarantee that all the paths from

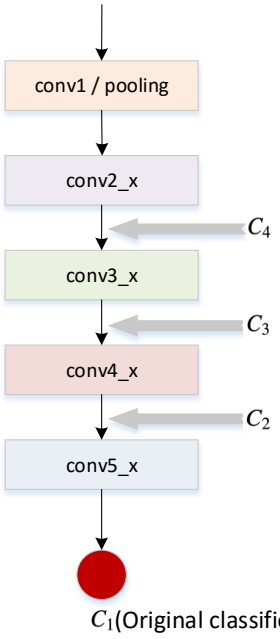


Figure 5: Locations of the auxiliary classifiers added to the ResNet backbone networks evaluated on the ImageNet classification dataset. The grey thick arrows indicate the locations where auxiliary classifiers are added. Following the definition in our paper, we denote these three auxiliary classifiers as C_2, C_3 and C_4, respectively.

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. For example, the auxiliary classifier C_3 has its own conv_4x and conv_5x blocks 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. We show the details of the convolutional blocks of the auxiliary classifiers in Table 7.

In the experiment regarding Fig.3 'DS with simple aux. classifiers' of our paper, we use a very simple structure as suggested in [5] for the auxiliary classifiers. The structure is shown in Fig.6. Specifically, the hyper-parameters of the average pooling layer in C_2 are kernel size = 5×5, stride = 3 and padding = 1, and in C_3 are kernel size = 7×7, stride = 7 and padding = 3. The feature map with size of $4 \times 4 \times 256$ or $4 \times 4 \times 128$ is fed into its respective fully connected layer with a Softmax function for final predication.

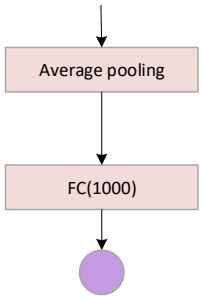


Figure 6: Structure of the simple auxiliary classifiers added to the ResNet-18 backbone network evaluated on the ImageNet classification dataset. The average pooling layer will down-sample the input feature map into a new one with the spatial size of 4×4. Then the feature map will be flattened to be a one-dimensional vector which will be fed into the fully connected layer. Here, the purple circle denotes the Softmax layer which will output a probability distribution.

3. Accuracy Curves of ResNet Models Trained on ImageNet

Fig. 7 shows the curves of Top-1 training error and test error of the ResNet models trained on the ImageNet classification dataset. Compared with the standard training scheme and DS, it can be seen that DKS has the worst training accuracy but the best test accuracy for all backbone networks, showing better capability to suppress over-fitting during training.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 6

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[3] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *ICLR*, 2018. 2

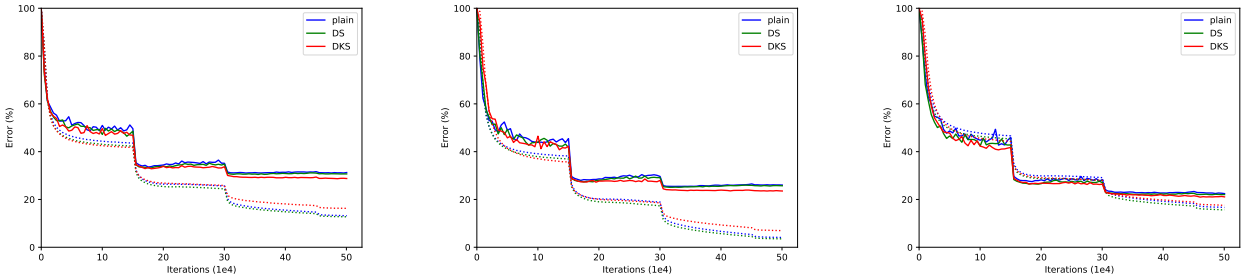
[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[5] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 1, 4

[6] S. Srinivas and F. Fleuret. Knowledge transfer with Jacobian matching. In *ICML*, 2018. 1

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *CVPR*, 2015. 1

[8] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 2



(a) (b) (c)

Figure 7: Curves of Top-1 training error (dashed line) and test error (solid line) on the ImageNet classification dataset with ResNet-18 (a), ResNet-50 (b) and ResNet-152 (c).

	ResNet(d=32)			ResNet(d=110)		
	C_1	C_3	C_2	C_1	C_3	C_2
conv1	3×3, 16	-	-	3×3, 16	-	-
conv2_x	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 5$	-	-	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 18$	-	-
conv3_x	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 5$	-	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 18$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 9$	-
conv4_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 18$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 9$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 18$

Table 2: Details of the convolutional blocks of the auxiliary classifiers added to the ResNet backbone networks evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding number of output channels.

	WRN-28-4			WRN-28-10		
	C_1	C_3	C_2	C_1	C_3	C_2
conv1	3×3, 16	-	-	3×3, 16	-	-
conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$	-	-	$\begin{bmatrix} 3 \times 3, 160 \\ 3 \times 3, 160 \end{bmatrix} \times 4$	-	-
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	-	$\begin{bmatrix} 3 \times 3, 320 \\ 3 \times 3, 320 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 320 \\ 3 \times 3, 320 \end{bmatrix} \times 4$	-
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 640 \\ 3 \times 3, 640 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 640 \\ 3 \times 3, 640 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 1280 \\ 3 \times 3, 1280 \end{bmatrix} \times 4$

Table 3: Details of the convolutional blocks of the auxiliary classifiers added to the WRN backbone networks evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding number of output channels.

	DenseNet(d=40,k=12)			DenseNet(d=100,k=12)		
	C_1	C_3	C_2	C_1	C_3	C_2
conv1	3×3, 24	-	-	3×3, 24	-	-
conv2_x	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 12$	-	-	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 32$	-	-
conv3_x	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 12$	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 12$	-	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 32$	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 16$	-
conv4_x	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 12$	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 36 \end{bmatrix} \times 12$	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 32$	$\begin{bmatrix} 3 \times 3, 12 \end{bmatrix} \times 16$	$\begin{bmatrix} 3 \times 3, 36 \end{bmatrix} \times 32$

Table 4: Details of the convolutional blocks of the auxiliary classifiers added to the DenseNet backbone networks evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of building blocks and the corresponding growth rate.

C_1	32	64	128	128	256	256	(512,s2)	512×5	(1024,s2), 1024
C_3	-	-	-	-	-	-	(512,s2)	512×3	(1024,s2), 1024
C_2	-	-	-	-	-	-	-	-	(2048,s2), 2048

Table 5: Details of the convolutional blocks of the auxiliary classifiers added to the MobileNet backbone network evaluated on the CIFAR-100 dataset. In the table, every cell shows the number of output channels, and $s2$ denotes the stride of the convolution operation in this layer is 2.

	Original			Narrow		Shallow	
	C_1	C_3	C_2	C_3	C_2	C_3	C_2
conv1	3×3, 16	-	-	-	-	-	-
conv2_x	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 5$	-	-	-	-	-	-
conv3_x	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 5$	-	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 5$	-	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 2$	-
conv4_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$

Table 6: Details of the narrow and shallow auxiliary classifiers added to the ResNet-32 backbone network 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-18			ResNet-50		ResNet-152	
	C_2	C_3	C_4	C_2	C_3	C_2	C_3
conv3_x	-	-	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 1$	-	-	-	-
conv4_x	-	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 1$	-	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$	-	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 18$
conv5_x	$\begin{bmatrix} 3 \times 3, 1024 \\ 3 \times 3, 1024 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 4096 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 4096 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 2$

Table 7: Details of the convolutional blocks of the auxiliary classifiers added to the ResNet backbone networks evaluated on the ImageNet classification 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 ResNet-152) and their parameter sizes. For comparison with the backbone networks, please refer to the Table 1 of the ResNet paper [1].