Feature Decoupled Knowledge Distillation via Spatial Pyramid Pooling

Lei Gao[0000−0001−5958−4061][1] and Hui Gao *[0000−0002−9557−7739][1]

University of Electronic Science and Technology of China
202021080708@std.uestc.edu.cn, huigao@uestc.edu.cn

Abstract. Knowledge distillation (KD) is an effective and widely used technique of model compression which enables the deployment of deep networks in low-memory or fast-execution scenarios. Feature-based knowledge distillation is an important component of KD which leverages intermediate layers to supervise the training procedure of a student network. Nevertheless, the potential mismatch of intermediate layers may be counterproductive in the training procedure. In this paper, we propose a novel distillation framework, termed Decoupled Spatial Pyramid Pooling Knowledge Distillation, to distinguish the importance of regions in feature maps. Specifically, we reveal that (1) spatial pyramid pooling is an outstanding method to define the knowledge and (2) the lower activation regions in feature maps play a more important role in KD. Our experiments on CIFAR-100 and Tiny-ImageNet achieve state-of-the-art results.

Keywords: knowledge distillation · Spatial pyramid pooling.

1 Introduction

In the last few years, deep neural networks have been the basis of many successes in both industry and academia, especially for computer vision [16,12] and natural language processing [11] tasks. Nevertheless, the large depth or width and numbers of parameters account for the drawback that they may demand high-speed computing power and large memory to store, limiting their availability in applications or platforms with low memory or real-time requirements, e.g., mobile phones and embedded devices. This led to a rapidly increasing interest in research on exploring smaller and faster models. Therefore, a variety of ways including explicit prudent network design [19], model binarization [10,33], network pruning [25], model compression [14] and most attractively knowledge distillation [18].

Previous works [3,5] have revealed that small networks usually have comparable representation capacity to large networks; but compared with large networks they are hard to train and find proper parameters to realise the objective function. The limitation of small networks appears to be caused by the difficulty of optimization rather than the size of networks. To better train a small network

,the distillation approach starts with a powerful teacher network or network ensemble, and then designs learning rules, i.e., elaborate loss function, to train a smaller student network mimicking the teacher. In the vanilla KD framework, the knowledge is defined as the prediction of the final layer of the teacher,i.e., response-based KD, which is an intuitive understanding of how a model generalizes [18]. However, the high abstraction of knowledge transferring from teacher to student ignores the valuable information contained in the intermediate layers.

Benefit from the techniques of representation learning [4], networks are good at acquiring multiple levels of feature representation with increasing abstraction [13], arousing an increasing interest in the research of feature-based KD. Feature-based KD exploits both the output of the last layer and the output of intermediate layers, i.e., feature maps, to supervise the training of student network. Fitnets [34] first introduced intermediate representations in KD, providing hints [1] to alleviate the difficulty of training a fast yet accuracy student network. Inspired by this work, a variety of feature-based KD methods have been proposed to match the features between teacher and student network indirectly [42,1,39,22,21,7]. Nevertheless, existing efforts mainly rely on handcrafted layer assignments, i.e., random selection or one-to-one association and may cause semantic mismatch or negative regularization effect in student's training. Cheng et al. [9] conduct a series of experiments to illustrate why KD works and reveal that the position differences of visual concepts (VCs) that reflect the larger activation regions from the teacher and the student are marginal. Therefore, we infer that non-VCs reflecting the lower activation regions play a more important role in the process of KD. Back to the feature map itself, we should further dig more information to improve the performance of KD.

In this paper, we propose Decoupled Spatial Pyramid Pooling Knowledge Distillation (DSPP) to exploit intermediate knowledge by exploring a novel method to define knowledge and decoupling feature map to optimize KD training procedure. A spatial pyramid pooling architecture [15] is applied in our approach for automatically perceiving knowledge, which effectively captures informative knowledge at various scales of feature map. Then a decoupling module is designed to analyze region-level semantic loss between student and teacher network based on the observation that the lower activation regions in feature map plays a more important role in KD, i.e., lower activation regions contain more informative knowledge cues. To align the spatial dimension of teacher and student layer pair, feature map of the student layer is projected to the same dimension of the teacher layer. By taking advantage of spatial pyramid pooling and decoupled region-level loss assignment, the student network can be effectively optimized with more sophisticated supervision. Our main contributions are as follows:

- A new method of defining knowledge named Spatial Pyramid Pooling is proposed to perceive knowledge in the last feature map at various scales.

[1] Hints mean the output of a teacher's hidden layers that supervise the student's training.

- Decoupled semantic loss assignment is applied to improve the weights of lower activation regions which play a more important role in KD, aiming at alleviating the difficulty of training the student network.
- Extensive experiments on CIFAR-100 and Tiny-ImageNet with a variety of settings based on popular network architectures reveal that DSPP outperforms than most of state-of-the-art approaches.

The source code is available at https://github.com/luilui97/DSPP.

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation for model compression is similar to the way in which human beings perceive and learn knowledge. The distillation-based approach of model compression is first proposed by [5] and is re-popularised by [18], where soft targets from a pretrained teacher model are exploited to improve the performance of a given student model. Furthermore, recent knowledge distillation methods have extended to mutual learning [44], assistant teaching [27] and self-learning [41]. As pointed out in [41,32,28], soft targets predicted by the teacher model serve as an effective regularization to prevent the student model from making over-confident prediction. Moreover, some online KD variants have been proposed to reduce the expense of pre-training [2,6].

2.2 Response-based Knowledge Distillation

Response-based knowledge often refers to the logits or predictions of the teacher model. The main idea of response-based KD is to mimic the final prediction of the teacher model. The response-based KD is simple yet effective for model compression, and has been widely applied in various tasks. For example, the response in object detection task may contains classification logits with offsets of bounding boxes [8]. The most fashionable response-based knowledge for image classification is proposed by [18] named soft targets, which contains the informative dark knowledge from the teacher model. Besides dark knowledge, another interpretation for the effectiveness of response-based KD is the similarity between soft targets and label smoothing [8] or regularizers [28]. However, the compact reliability of the output of last layer lead to the absence of intermediate-level supervision from the teacher model, limiting the student's supervised learning. Our proposed approach incorporates more informative guidance from the teacher's last hint layer via spatial pyramid pooling.

2.3 Feature-based Knowledge Distillation

With the techniques of representation learning [4], both the output of last layer and the intermediate layers, i.e., feature maps, can offer informative knowledge guidance to the student model. Feature-based KD is first proposed in Fitnet

[34], in which hints are leveraged to improve the training of the student model. Inspired by Fitnet, various feature-based KD methods have been proposed to match the hint layers indirectly. Specifically, Zagoruyko et al. [42] derive an attention map from the teacher model to express knowledge and transferred it to the student model. To simplify the transfer of knowledge, Kim et al. [22] introduce factors to provide a more interpretive form of intermediate representations. Furthermore, Jin et al. [21] design a route constrained optimization to overcome the challenge of performance gap between teacher and student. Heo et al. [17] utilize activation boundary formed by hidden neurons rather than activation values to transfer knowledge. Recently, Chen et al. [7] suggest an adaptive semantic attention allocation mechanism, which matches teacher layers and student layers properly. SAKD [37] creatively proposes that distillation spots should be adaptive to training samples and distillation epochs.These feature-based KD methods pay more attention to fixing multiple potential mismatched intermediate layers, while our proposed method focuses on the last hint layer and leverages spatial pyramid pooling and decoupled semantic loss assignment to supervise the student, which relieves us from the exhausting layer matching process.

3 Method

We describe our proposed Decoupled Spatial Pyramid Pooling Knowledge Distillation (DSPP) method in this section. Firstly, we briefly recap the basic classic KD and illustrate it through additional necessary notations. Then, we provide an overview of our proposed DSPP architecture as well as the details of DSPP, e.g., loss function.

3.1 Preliminary

Firstly, we recap the procedure of classic knowledge distillations. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of N samples from K categories, and a powerful teacher network Θ_T pretrained on dataset $\mathcal{D}$, the goal of KD is training a student network Θ_S on $\mathcal{D}$ with less computational demand and shorter inference time under the supervision of Θ_T. Specifically, in response-based KD, the student Θ_S learns the knowledge from the last layer of the teacher Θ_T, while the student Θ_S learns knowledge from the hint layers of the teacher Θ_T in feature-based KD.

As is known to all, the probability of class k given by a network Θ is computed as

$$p\left(z_i\right) = \frac{exp\left(z_i\right)}{\sum_{j=1}^{K} exp\left(z_j\right)} \tag{1}$$

where the logit z_i is the output of the softmax layer in Θ for the i-th class. In response-based KD, a hyperparameter T is introduced to control the importance of each soft target. Accordingly, the soft targets can be estimated by a softmax function as:

$$p\left(z_i, T\right) = \frac{exp\left(z_i/T\right)}{\sum_{j=1}^{K} exp\left(z_j/T\right)} \tag{2}$$

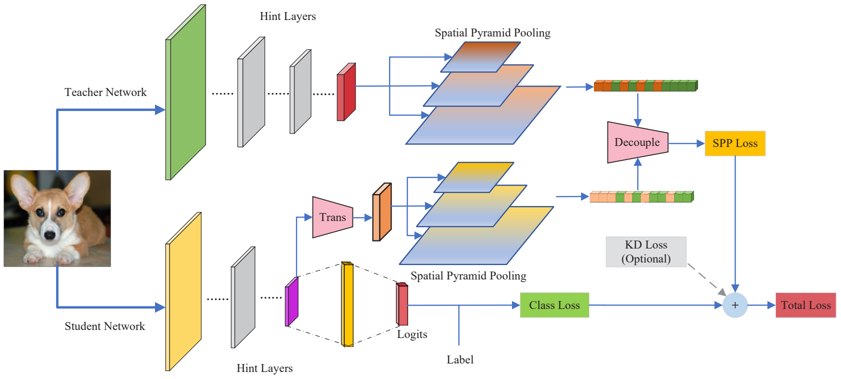

Fig. 1. The overview of Decoupled Spatial Pyramid Pooling Knowledge Distillation architecture. The last hint layer of the teacher is utilized to supervise the training of the student model via decoupled spatial pyramid pooling.

The distillation loss for response-based KD can be expressed as:

$$\begin{aligned} L_{KD}\left(p\left(\mathbf{z_t},T\right),p\left(\mathbf{z_s},T\right)\right) &= D_{KL}\left(p\left(\mathbf{z_t},T\right),p\left(\mathbf{z_s},T\right)\right) \\ &= p\left(\mathbf{z_s},T\right)\log\frac{p\left(\mathbf{z_s},T\right)}{p\left(\mathbf{z_t},T\right)} \end{aligned} \tag{3}$$

where D_{KL} denotes Kullback Leibler (KL) Divergence. Obviously, the optimization of Equ. (3) can match logits $\mathbf{z_s}$ of student model and $\mathbf{z_t}$ of teacher model.

For feature-based KD, the target is mimicking the teacher model's intermediate layers, the distillation loss can be formulated as:

$$L_{FD}\left(f_t\left(x\right),f_s\left(x\right)\right) = \mathcal{L}_F\left(\Phi\left(f_t\left(x\right)\right),\Phi\left(f_s\left(x\right)\right)\right) \tag{4}$$

where $f_t\left(x\right)$ and $f_s\left(x\right)$ denotes the feature maps, i.e., hint layers of teacher and student models respectively. $\Phi\left(\cdot\right)$ is transformation function, which is applied when the feature maps of student and teacher model are in different shape. $\mathcal{L}_F$ indicates the similarity measurement function which is applied to match the feature maps of student and teacher model. Specifically, $\mathcal{L}_F\left(\cdot\right)$ can be l_1-norm distance, l_2-norm distance, cross-entropy loss and maximum mean discrepancy loss in different feature-based KD methods.

3.2 Network Architecture

Fig. 1 illustrates our proposed DSPP architecture, which contains two subnetworks, i.e., student and teacher network, interacting with each other through decoupled spatial pyramid pooling. As mentioned earlier, the soft logits are actually the class probability distribution which is too abstract for the student to get informative knowledge. Furthermore, it's hard to find an appropriate match of hint layers of the teacher and student model. Therefore, we suggest leveraging the last hint layer to resolve the high abstraction of soft logits and avoid exhausting efforts on matching the hint layers. To align the last hint layers from the teacher and student model, a transformation operation is applied in the architecture. Inspired by [15], we introduce a new method to define knowledge in the hint layer, in which a spatial pyramid pooling is applied to capture informative knowledge in the hint layer at different scales. Besides, a decouple module is proposed to improve the importance of lower activation regions in the spatial pooling pyramid. The total loss consists of SPP loss from the Decouple module and class loss from the student network itself. Vanilla KD loss, i.e., response-based KD loss, is optional for our architecture. More details are introduced in the next part, Section 3.3.

3.3 Decoupled Spatial Pyramid Pooling

Spatial Pyramid Pooling. How to acquire informative knowledge from the last hint layer is a key issue for DSPP knowledge distillation. Spatial pyramid pooling is first proposed by He et al. [15] in visual recognition task, which liberates convolutional neural network from the limit of fixed input size. Considering the different architecture of the teacher and student network, we utilize spatial pyramid pooling to resolve the inconsistence issue that the student's last hint layer is of different shape from the teacher model. Moreover, contributed by hierarchical structure of spatial pyramid pooling, we can have a multi-scale receptive filed size of the last hint layer, which enables our model to perceive both global and local knowledge cue from the last hint layer. The procedure of spatial pyramid pooling can be formulated as follows:

$$f_{pyramid} = \left\{ Pooling\left(L, W\left(L\right)/k\right), \quad k = 1, 2, \cdots, n, \quad L \in \mathbb{R}^{b \times c \times h \times w} \right\} \tag{5}$$

where L denotes the input hint layer, and $W\left(\cdot\right)$ is adopted to get the width of L. Parameter k is the serial number of pyramid layer and starts from 1 to n. The function $Pooling\left(\cdot\right)$ takes 2 parameters: input feature map and the size of pooling kernel.

Why choose the last hint layer. As we know, the logits of the student model come from the fully-connected layer and are highly similar to the teacher model, i.e., their predictions are same on one dataset. However, for image classification task, the fully-connected layer is absence of spatial information of the input image which is two-dimensional or three-dimensional. As mentioned earlier, it's hard to find an appropriate match of hint layers of the teacher and student model and may reduce the interpretability of KD. These are the motivations why we choose the last hint layer. Moreover, the fully-connect layer is directly computed from the last hint layer which is theoretically and physically closest to the logits among all hint layers.

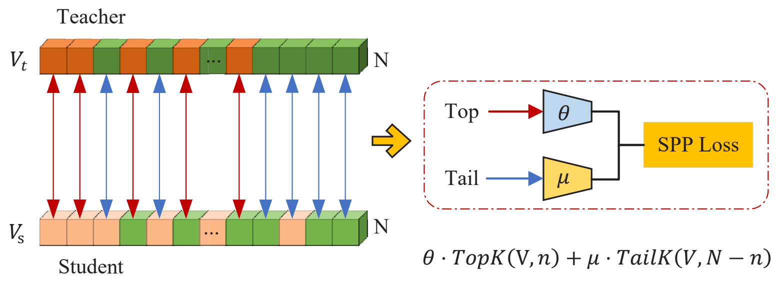

Fig. 2. An illustration of decouple module for SPP feature.

Decouple Module. To pay more attention to the lower activation regions, we propose a decouple module to handle the flattened feature from spatial pyramid pooling. In decouple module, the flattened feature is decoupled into two components according to value of each element in the feature. As shown in Fig. 2, the student feature V_s is element-wise matched to the teacher feature V_t by a two-way arrow. The red arrows point to the top-n largest element in V_t, whose the other ends point to the corresponding position in V_s. On the contrary, the blue arrows point to the last tail-$(N-n)$ element in V_t, where N denotes the length of V_s or V_t. The loss of SPP can be calculated as:

$$\begin{aligned} top\left(n\right) =& argmax\left(V_t\right)\left[0:n\right], \quad n = 0, 1, \cdots, N; \\ tail\left(m\right) =& argmin\left(V_t\right)\left[0:m\right], \quad m = N-n; \\ \mathcal{L}_{SPP} =& \theta\mathcal{L}_2\left(V_t\left[top\left(n\right)\right], V_s\left[top\left(n\right)\right]\right) + \\ & \mu\mathcal{L}_2\left(V_t\left[tail\left(m\right)\right], V_s\left[tail\left(m\right)\right]\right), \quad V_t, V_s \in \mathbb{R}^N \end{aligned} \tag{6}$$

where $top(\cdot)$ denotes the indices of top-$(\cdot)$ elements in V_t and $tail(\cdot)$ denotes the indices of tail-$(\cdot)$ elements in V_t. Function $\mathcal{L}_2\left(\cdot\right)$ indicates l_2-norm distance. θ and μ are hyperparameters to control the weight of decoupled components. To improve the importance of lower activation regions, we let μ greater than θ. The reason why we pay more attention to the lower activation regions is that the powerful teacher models with numerous parameters may have a more sophisticated mechanism to find more details of the input which reflect on the lower activation regions. The lower activation regions contribute to improve the accuracy and generalization of the student model.

3.4 Loss and Optimization

As mentioned earlier, the loss of our proposed consist of three parts: classification loss $\mathcal{L}_{cls}$, SPP loss $\mathcal{L}_{SPP}$, and KD loss $\mathcal{L}_{KD}$ (optional). For multi-class

Algorithm 1 DSPP Knowledge Distillation

```
Input:
Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$;
A pre-trained teacher model with parameter $\Theta^t$;
A randomly initialized student model with parameter $\Theta^s$
Output: A well-trained model;
while $\Theta^s$ is not converged do
    Sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$.
    Input $\mathcal{B}$ into $\Theta^t$ and $\Theta^s$ to obtain last hint layers $F_1^t$ and $F_1^s$.
    Align $F_1^t$ and $F_1^s$ then get $F_2^t$ and $F_2^s$.
    Perform Spatial Pyramid Pooling on $F_2^t$ and $F_2^s$.
    Backward propagate the gradients of loss by Equ. (8) and update $\Theta^s$.
end while
```

classification, the objective function $\mathcal{L}_{cls}$ is defined as the cross entropy error between the predictions $\hat{y}$ and the correct labels y:

$$\mathcal{L}_{cls} = -\left[y \log \hat{y} + (1-y) \log (1-\hat{y})\right]. \tag{7}$$

In the end, $\mathcal{L}_{total}$ is calculated as follows:

$$\mathcal{L}_{total} = \gamma \mathcal{L}_{cls} + \alpha \mathcal{L}_{KD} + \beta \mathcal{L}_{SPP} \tag{8}$$

where hyperparameters γ, α and β are adopted to balance the weight of $\mathcal{L}_{cls}$, $\mathcal{L}_{KD}$, and $\mathcal{L}_{SPP}$ respectively. The training procedure of our proposed DSPP is summarized in Algorithm 1.

4 Experiment

To demonstrate the effectiveness of our proposed DSPP knowledge distillation, we conduct a series of experiments with a variety of teacher-student combinations on popular network architectures, including VGG [36], MobileNet [19,35] , ResNet [16] and ShuffleNet [43,26]. The *CIFAR-100* [23] dataset is used in our experiments, which contains 50K training color images (32 × 32) with 0.5K images per class and 10K test images, 100 classes in total. Students and teachers of the same and different architecture style are both evaluated and compared with representative distillation approaches. Furthermore, ablation studies on the decouple module and the weights of lower activation regions are also conducted. We add our DSPP module in the KD collection established by [38] and follow their experiment settings. To evaluate the generalization of our method, we further conduct a series of experiments on Tiny-ImageNet [24].

4.1 Results

Table 1 gives the Top-1 test accuracy on CIFAR-100 based on five homogeneous network combinations and thirteen KD methods are compared with our proposed

Table 1. Test Top-1 accuracy (%) of homogeneous teacher-student networks on CIFAR-100 dataset of a variety of KD approaches.

Teacher Student	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32x4 resnet8x4	vgg13 vgg8
Teacher	72.34	74.31	74.31	79.42	74.64
Student	69.06	69.06	71.14	72.50	70.36
KD[18]	70.66	70.67	73.08	73.33	72.98
FitNet[34]	69.21	68.99	71.06	73.50	71.02
AT[42]	70.55	70.22	72.31	73.44	71.43
SP[39]	69.67	70.04	72.69	72.94	72.68
CC[31]	69.63	69.48	71.48	72.97	70.71
VID[1]	70.38	70.16	72.61	73.09	71.23
PKD[29]	69.61	69.25	71.82	71.90	71.48
PKT[30]	70.34	70.25	72.61	73.64	72.88
AB[17]	69.47	69.53	70.98	73.17	70.94
FT[22]	69.84	70.22	72.37	72.86	70.58
FSP[40]	69.95	70.11	71.89	72.62	70.23
NST[20]	69.60	69.53	71.96	73.30	71.53
CRD[38]	71.16	**71.46**	73.48	**75.51**	**73.94**
DSPP(OURS)	**71.45**	71.19	**73.56**	75.31	73.59
DSPP+KD	**71.51**	**71.88**	**73.74**	**75.63**	**73.99**

Table 2. Test Top-1 accuracy (%) of heterogeneous teacher-student networks on CIFAR-100 dataset of a variety of KD approaches.

Teacher Student	vgg13 MobileNetV2	ResNet50 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleNetV1	resnet32x4 ShuffleNetV2
Teacher	74.64	79.34	79.34	79.42	79.42
Student	64.60	64.60	70.36	70.50	71.82
KD	67.37	67.35	73.81	74.07	74.45
FitNet	64.14	63.16	70.69	73.59	73.54
AT	59.40	58.58	71.84	71.73	72.73
SP	66.30	68.08	73.34	73.48	74.56
CC	64.86	65.43	70.25	71.14	71.29
VID	65.56	67.57	70.30	73.38	73.40
PKD	64.52	64.43	71.50	72.28	73.21
PKT	67.13	66.52	73.01	74.10	74.69
AB	66.06	67.20	70.65	73.55	74.31
FT	61.78	60.99	70.29	71.75	72.50
NST	58.16	64.96	71.28	74.12	74.68
CRD	**69.73**	69.11	**74.3**	75.11	75.65
DSPP(OURS)	67.87	68.18	73.70	74.61	75.71
DSPP+KD	68.95	**69.21**	74.13	**75.25**	**76.33**

DSPP. The results of other approaches are partially cited from [38], as well as Table 2. According to Table 1, it is shown that DSPP consistently achieves higher accuracy than state-of-the-art distillation approaches with the participation of vanilla KD. Surprisingly, we found that our DSPP works well and none of the other methods except CRD consistently outperforms than vanilla KD.

The results of five heterogeneous network combinations are shown in Table 2. Obviously, while switching the teacher-student combinations from homogeneous to heterogeneous styles, methods that constructed on multiple intermediate layers tend to perform worse than methods that distill last few layers or logits. Even worse, some methods may play a opposed negative role in the training procedure of student network. For instance, the AT and FitNet even preform worse than the vanilla student. As mentioned earlier, the mismatch of hint layers may account for this phenomenon. Tian et al. [38] gives another explanation that networks of different style have their unique hyperspace and paths mapping from the input to the output and the forced mimics of intermediate layers then conflicts with these kind of misleading. This why we have a almost equal performance compared with CRD which utilizes a family of contrastive objectives.

To evaluate the generalization of our method, we conduct a series of experiments on Tiny-ImageNet on three classical teacher-student architecture as shown in Table 3. The results show that DSPP outperforms other methods including a combination of CRD and SAKD [37] and further demonstrate the effectiveness

Table 3. Test Top-1 accuracy (%) of a variety of KD approaches on Tiny-ImageNet dataset.

T→S	ResNet56→ResNet20	ResNet110→ResNet20	Vgg13→Vgg8
Teacher	58.34	58.46	60.09
Student	51.89	51.89	56.03
KD	53.04	53.40	57.33
FitNet	54.43	54.04	58.33
AT	54.39	54.57	58.85
FT	53.90	54.46	58.87
PKT	54.29	54.70	58.87
SP	54.23	54.38	58.78
VID	53.89	53.94	58.55
CC	54.22	54.26	58.18
RKD	53.95	53.88	58.58
NST	53.66	53.82	58.85
CRD	55.04	54.69	58.88
SAKD [37]+CRD	55.06	55.28	59.38
DSPP(OURS)	**55.23**	**55.56**	**59.69**

Table 4. Distillation (ResNet110→ResNet32) accuracy at different μ on *CIFAR-100*.

μ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
acc	70.78	71.14	71.26	71.35	71.32	71.62	71.90	71.82	71.90	72.12
μ	1.5	2	3	4	5	6	7	8	9	10
acc	72.17	72.61	72.65	73.21	73.42	73.5	73.52	73.59	73.50	73.49

of DSPP. Images in Tiny-ImageNet are two times larger than CIFAR-100, so the feature map is two times larger which can provide more information. Consequently, the performance of DSPP on Tiny-ImageNet is better than CIFAR-100.

4.2 Ablation Study

Firstly, we conduct a series of experiments on ResNet110→ResNet32 architecture to demonstrate that regions of lower activation values play a more important role in KD. We set μ from 0.1 to 10 to verify whether we should pay more attention to regions of lower activation values. As shown in Table 4, the performances of DSPP are better when $\mu > 1$ compared with $\mu \leq 1$. The cases $\mu > 1$ denote that the model concentrate more on lower activation regions.

To evaluate the performance of decouple module, we conduct a series experiments on how much improvement the decouple module brings. The results in Table 5 demonstrate the effectiveness of decouple module. Obviously, the decouple module contributes a lot to the improvement of DSPP, where the largest improvement is 1.59 percent in (vgg13-MobileNetV2) teacher-student architecture.

Table 5. Test Top-1 accuracy (%) of ten teacher-student networks with/without decouple module. "w" denotes with decouple module, while "w/o" denotes without decouple module.

teacher student	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32x4 resnet8x4	vgg13 vgg8
w	71.45	71.19	73.56	75.31	73.59
w/o	71.44	70.34	72.56	74.21	72.24
improvement	0.01	0.85	1.00	1.10	1.35
teacher student	vgg13 MobileNetV2	ResNet50 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleNetV1	resnet32x4 ShuffleNetV2
w	67.87	68.18	73.70	74.61	75.56
w/o	66.28	66.67	72.47	73.22	74.84
improvement	1.59	1.51	1.23	1.39	0.72

Another interesting observation is that teacher-student architectures of heterogeneous style benefit more from the decouple module than those of homogeneous style.

5 Conclusion

Intermediate layers of a powerful teacher model contain various informative semantic knowledge, but mismatch of hint layers may lead to a counterproductive result. An urgent challenge for knowledge distillation is to establish a mechanism of correctly leveraging logits and intermediate layers. To reduce dependence on multiple intermediate layers and improve the interpretability of KD, we propose feature decoupled knowledge distillation via spatial pyramid pooling. Decoupled spatial pyramid pooling operation is applied on aligned last hint layer of both teacher and student model to acquire multi-scale knowledge cues. Experimental results show that distillation via DSPP outperforms the compared approaches. Additional ablation studies also demonstrate the effectiveness of our decouple module. For the future work, the proposed method can be combined with other KD methods and the way to select KD spot can be also explored further.

References

1. Ahn, S., Hu, S.X., Damianou, A.C., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 9163–9171 (2019). https://doi.org/10.1109/CVPR.2019.00938

2. Anil, R., Pereyra, G., Passos, A., Ormándi, R., Dahl, G.E., Hinton, G.E.: Large scale distributed neural network training through online distillation. In: 6th International Conference on Learning Representations, ICLR (2018)
3. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in Neural Information Processing Systems. pp. 2654–2662 (2014)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013). https://doi.org/10.1109/TPAMI.2013.50
5. Buciluundefined, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 535–541. New York, NY, USA (2006). https://doi.org/10.1145/1150402.1150464
6. Chen, D., Mei, J., Wang, C., Feng, Y., Chen, C.: Online knowledge distillation with diverse peers. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI. pp. 3430–3437 (2020)
7. Chen, D., Mei, J., Zhang, Y., Wang, C., Wang, Z., Feng, Y., Chen, C.: Cross-layer distillation with semantic calibration. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI. pp. 7028–7036 (2021)
8. Chen, G., Choi, W., Yu, X., Han, T.X., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 742–751 (2017)
9. Cheng, X., Rao, Z., Chen, Y., Zhang, Q.: Explaining knowledge distillation by quantifying the knowledge. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12925–12935 (2020)
10. Courbariaux, M., Bengio, Y., David, J.: Binaryconnect: Training deep neural networks with binary weights during propagations. In: Advances in Neural Information Processing Systems. pp. 3123–3131 (2015)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
13. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. Int. J. Comput. Vis. **129**(6), 1789–1819 (2021). https://doi.org/10.1007/s11263-021-01453-z
14. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In: 4th International Conference on Learning Representations, ICLR (2016)
15. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision, ECCV. Lecture Notes in Computer Science, vol. 8691, pp. 346–361 (2014). https://doi.org/10.1007/978-3-319-10578-9_23
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

17. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI. pp. 3779–3787 (2019). https://doi.org/10.1609/aaai.v33i01.33013779
18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Computer Science **14**(7), 38–39 (2015)
19. 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)
20. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
21. Jin, X., Peng, B., Wu, Y., Liu, Y., Liu, J., Liang, D., Yan, J., Hu, X.: Knowledge distillation via route constrained optimization. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV. pp. 1345–1354 (2019). https://doi.org/10.1109/ICCV.2019.00143
22. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 2765–2774 (2018)
23. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases **1**(4) (2009)
24. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
25. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: 5th International Conference on Learning Representations, ICLR (2017)
26. Ma, N., Zhang, X., Zheng, H., Sun, J.: Shufflenet V2: practical guidelines for efficient CNN architecture design. In: European Conference on Computer Vision, ECCV. Lecture Notes in Computer Science, vol. 11218, pp. 122–138 (2018). https://doi.org/10.1007/978-3-030-01264-9_8
27. Mirzadeh, S., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI. pp. 5191–5198 (2020)
28. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 4696–4705 (2019)
29. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 3967–3976 (2019). https://doi.org/10.1109/CVPR.2019.00409
30. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: European Conference on Computer Vision, ECCV. Lecture Notes in Computer Science, vol. 11215, pp. 283–299 (2018). https://doi.org/10.1007/978-3-030-01252-6_17
31. Peng, B., Jin, X., Li, D., Zhou, S., Wu, Y., Liu, J., Zhang, Z., Liu, Y.: Correlation congruence for knowledge distillation. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV. pp. 5006–5015 (2019). https://doi.org/10.1109/ICCV.2019.00511
32. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.E.: Regularizing neural networks by penalizing confident output distributions. In: 5th International Conference on Learning Representations, ICLR (2017)

33. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European Conference on Computer Vision, ECCV. Lecture Notes in Computer Science, vol. 9908, pp. 525–542 (2016). https://doi.org/10.1007/978-3-319-46493-0_32
34. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: 3rd International Conference on Learning Representations, ICLR (2015)
35. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR (2015)
37. Song, J., Chen, Y., Ye, J., Song, M.: Spot-adaptive knowledge distillation. IEEE Transactions on Image Processing **31**, 3359–3370 (2022)
38. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: 8th International Conference on Learning Representations, ICLR (2020)
39. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV. pp. 1365–1374 (2019). https://doi.org/10.1109/ICCV.2019.00145
40. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 7130–7138 (2017). https://doi.org/10.1109/CVPR.2017.754
41. Yuan, L., Tay, F.E.H., Li, G., Wang, T., Feng, J.: Revisiting knowledge distillation via label smoothing regularization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR. pp. 3902–3910 (2020). https://doi.org/10.1109/CVPR42600.2020.00396
42. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: 5th International Conference on Learning Representations, ICLR (2017)
43. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 6848–6856 (2018). https://doi.org/10.1109/CVPR.2018.00716
44. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4320–4328 (2018). https://doi.org/10.1109/CVPR.2018.00454