Distilling Knowledge via Knowledge Review

Pengguang Chen[1] Shu Liu[2] Hengshuang Zhao[3] Jiaya Jia[1,2]
The Chinese University of Hong Kong[1] SmartMore[2] University of Oxford[3]
{pgchen, leojia}@cse.cuhk.edu.hk liushuhust@gmail.com hengshuang.zhao@eng.ox.ac.uk

Abstract

Knowledge distillation transfers knowledge from the teacher network to the student one, with the goal of greatly improving the performance of the student network. Previous methods mostly focus on proposing feature transformation and loss functions between the same level's features to improve the effectiveness. We differently study the factor of connection path cross levels between teacher and student networks, and reveal its great importance. For the first time in knowledge distillation, cross-stage connection paths are proposed. Our new review mechanism is effective and structurally simple. Our finally designed nested and compact framework requires negligible computation overhead, and outperforms other methods on a variety of tasks. We apply our method to classification, object detection, and instance segmentation tasks. All of them witness significant student network performance improvement.

1. Introduction

Deep convolution neural networks (CNNs) have achieved remarkable success in a variety of computer vision tasks. However, the success of CNN is often accompanied with considerable computation and memory consumption, making it a challenging topic to apply to devices with limited resource. There have been techniques for training fast and compact neural networks, including designing new architectures [10, 2, 11, 26], network pruning [20, 15, 34, 4, 19], quantization [13] , and knowledge distillation [9, 25].

We focus on knowledge distillation in this paper considering its practicality, efficiency, and most importantly the potential to be useful. It forms a very general line, applicable to almost all network architectures and can combine with many other strategies, such as network pruning and quantization [32], to further improve network design.

Knowledge distillation is first proposed in [9]. The process is to train a small network (also known as the student) under the supervision of a larger network (a.k.a. the teacher). In [9], knowledge is distilled though the teacher's logit, which means the student is supervised by both ground truth labels and teacher's logits. Recently, effort has been made to improve distillation effectiveness. FitNet [25] distilled knowledge though intermediate features. AT [38] further optimized FitNet and used the attention map of features to deliver knowledge. PKT [23] modeled knowledge of the teacher as a probability distribution while CRD [28] used a contrastive objective to transfer knowledge. All these solutions focused on transformation and loss functions.

Our New Finding We in this paper tackle this challenging problem from a new perspective regarding the connection path between the teacher and student. To briefly understand our idea, we first show how previous work deals with these paths. As shown in Figure 1(a)-(c), *all* previous methods only use the-same-level information to guide the student. For example, when supervising the student's fourth-stage output, *always* the teacher's fourth-stage information is utilized. This procedure looks intuitive and easy to construct. But we intriguingly reveal that it is in fact a bottleneck in the whole knowledge distillation framework – *quick update of the structure surprisingly improves the whole-system performance consistently for many tasks*.

We investigate the previously neglected importance of designing connection paths in knowledge distillation and propose a new effective framework accordingly. The key modification is to use low-level features in the teacher network to supervise deeper features for the student, which results in much improved overall performance.

We further analyze the network structure and discover the fact that the student high-level stage has the great capacity to learn useful information from the teacher's low-level features. More analysis is provided in Section 4.4. This process is analogous to human learning curve [35] where a young kid can only comprehend a small portion of knowledge that is taught. During the course of grow-up, more and more knowledge from past years may be gradually understood and remembered as experience.

Our Knowledge Review Framework Based on these discoveries, we propose to use multi-level information of the teacher to guide one-level learning of the student net-

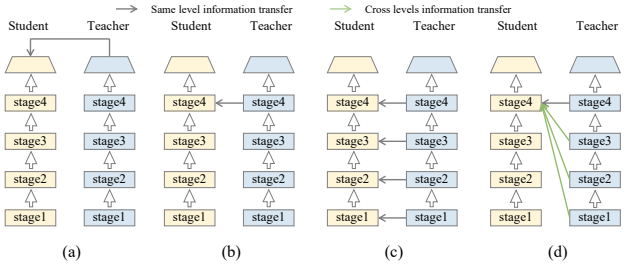


Figure 1. (a)-(c) Previous knowledge distillation frameworks. They only transfer knowledge within the same levels. (d) Our proposed "knowledge review" mechanism. We use multiple layers of the teacher to supervise one layer in the student. Thus, knowledge passing arises among different levels.

work. Our novel pipeline is shown in Figure 1(d), which we call "knowledge review". The review mechanism is to use previous (shallower) features to guide the current feature. It means a student has to always check what has been studied before for refreshing understanding and context of "old knowledge". It is a common practice for our human study to connect knowledge taught at different stages during a period of time of study.

However, how to extract useful information from multi-level information from the teacher and how to transfer them to the student are open and challenge problems. To tackle them, we propose a residual learning framework to make the learning process stable and efficient. Further, a novel attention based fusion (ABF) module and a hierarchical context loss (HCL) function are designed to boost performance. Our proposed framework makes the student network much improve the effectiveness of learning.

By applying this idea, we achieve better performance in many computer vision tasks. Extensive experiments in Sec. 4 manifest the vast advantage of our proposed knowledge review strategy.

Main Contributions

- We propose a new review mechanism in knowledge distillation, utlizing multi-level information of the teacher to guide one-level learning of the student net.
- We propose a residual learning framework to better realize the learning process of the review mechanism.
- To further improve the knowledge review mechanism, we propose an attentation based fusion (ABF) module and a hierarchical context loss (HCL) function.
- We achieve state-of-the-art performance of many compact models in multiple computer vision tasks by applying our distillation framework.

2. Related Work

Knowledge distillation concept was proposed in [9], where the student network learns from both the ground-truth labels and the soft-labels provided by the teacher. FitNet [25] distilled knowledge through one stage intermediate feature. The idea in FitNet is simple, where the student network feature is transferred to the same shape of the teacher though convolution layers. $\mathcal{L}_2$ distance is used to measure the distance between them.

Many methods follow FitNet and use one-stage feature to distill knowledge. PKT [23] modeled knowledge of the teacher as a probability distribution and used KL divergence to measure the distance. RKD [22] used multiple example relation to guide learning of the student. CRD[28] combined contrastive learning and knowledge distillation, and used a contrastive objective to transfer knowledge.

There are also methods using multi-stage information to transfer knowledge. AT [38] used multiple layer attention maps to transfer knowledge. FSP [36] generated FSP matrix from layer feature and used the matrix to guide the student. SP [29] further improved AT. Instead of single input information, SP uses the similarity between examples to guide the student. OFD [8] contained a new distance function to distill major information between the teacher and student using marginal ReLU.

All previous methods do not discuss the possibility to "review knowledge", which, however, is found in our work very effective to quickly improve system performance.

3. Our Method

We first formalize the knowledge distillation process and the review mechanism. Then we propose a novel framework and introduce attention based fusion module and hierarchical context loss function.

3.1. Review Mechanism

Given an input image $\mathbf{X}$ and student network $\mathcal{S}$, we let $\mathbf{Y}_s = \mathcal{S}(\mathbf{X})$ represent the output logit of the student. $\mathcal{S}$ can be separated into different parts $(\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_n, \mathcal{S}_c)$, where $\mathcal{S}_c$ is the classifier and $\mathcal{S}_1, \cdots, \mathcal{S}_n$ are different stages separated by downsample layers. Thus, the process of generating output $\mathbf{Y}_s$ can be denoted as

$$\mathbf{Y}_s = \mathcal{S}_c \circ \mathcal{S}_n \circ \cdots \circ \mathcal{S}_1(\mathbf{X}). \tag{1}$$

We refer to "$\circ$" as nesting of functions where $g \circ f(x) = g(f(x))$. $\mathbf{Y}_s$ is the output of student, and intermidate features are $(\mathbf{F}_s^1, \cdots, \mathbf{F}_s^n)$. The ith feature is calculated as

$$\mathbf{F}_s^i = \mathcal{S}_i \circ \cdots \circ \mathcal{S}_1(\mathbf{X}). \tag{2}$$

For the teacher network $\mathcal{T}$, the process is almost the same and we omit the details. Following previous notations, single-layer knowledge distillation can be represented as

$$\mathcal{L}_{SKD} = \mathcal{D}\left(\mathcal{M}_s^i(\mathbf{F}_s^i), \mathcal{M}_t^i(\mathbf{F}_t^i)\right), \tag{3}$$

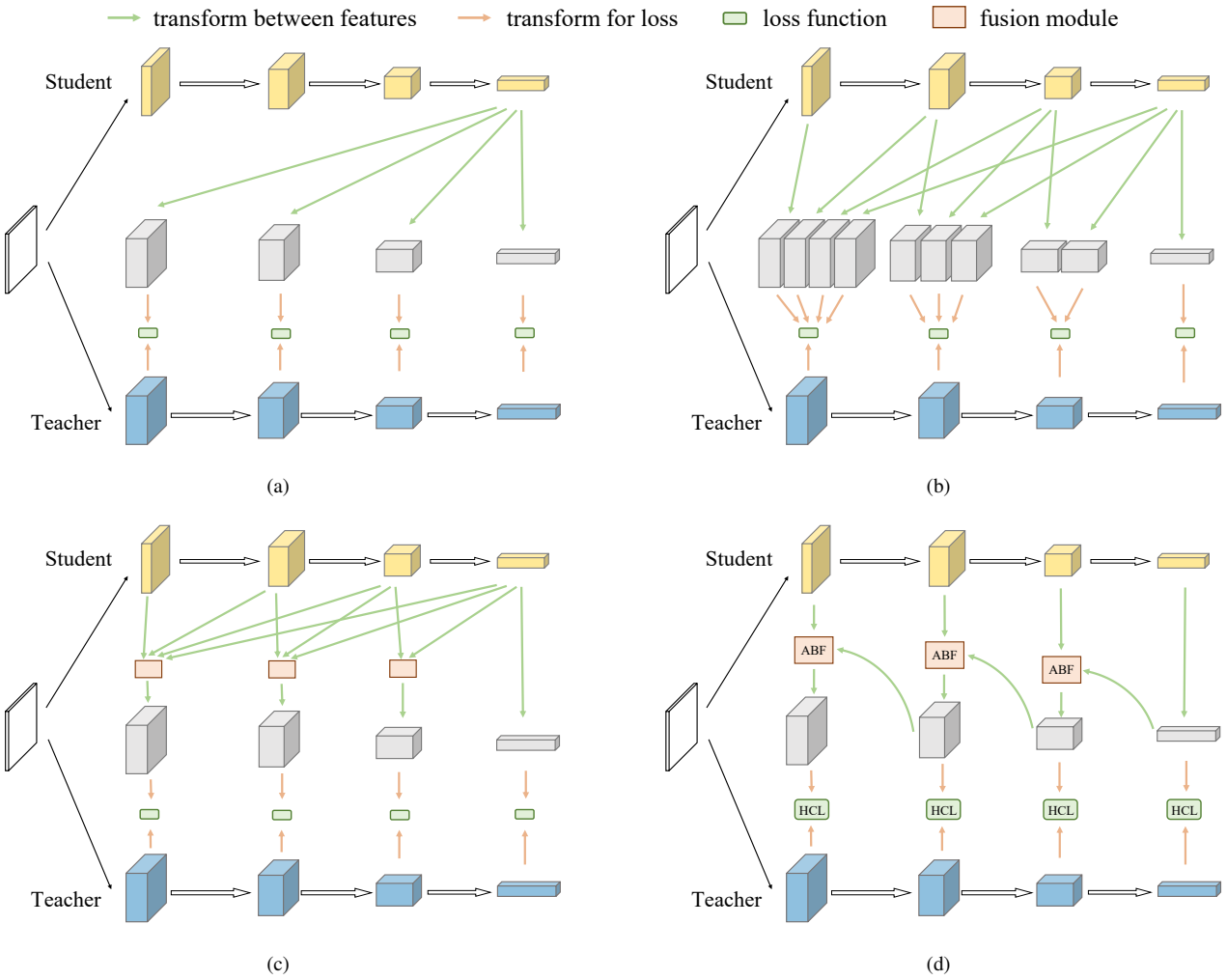


Figure 2. (a) Architecture for supervising one layer of the student according to the review mechanism. (b) Direct generalization from one layer to muliple ones. The process is straightforward but costly. (c) The architecture in (b) is optimized with fusion modules to obtain a compact framework. (d) We further improved the procudure in a progressive manner and utlize redisual learning as our final architecture. Structures of ABF and HCL are in Figure 3. This figure is best viewed in color.

where $\mathcal{M}$ is transformation that transfers the feature to target representation of attention maps [38] or factors [14]. $\mathcal{D}$ is the distance function that measures the gap between the student teacher. Similarly, multiple-layers knowledge distillation is written as

$$\mathcal{L}_{MKD} = \sum_{i \in \mathbf{I}} \mathcal{D}\left(\mathcal{M}_s^i(\mathbf{F}_s^i), \mathcal{M}_t^i(\mathbf{F}_t^i)\right), \tag{4}$$

where $\mathbf{I}$ stores the layers of features to transfer knowledge.

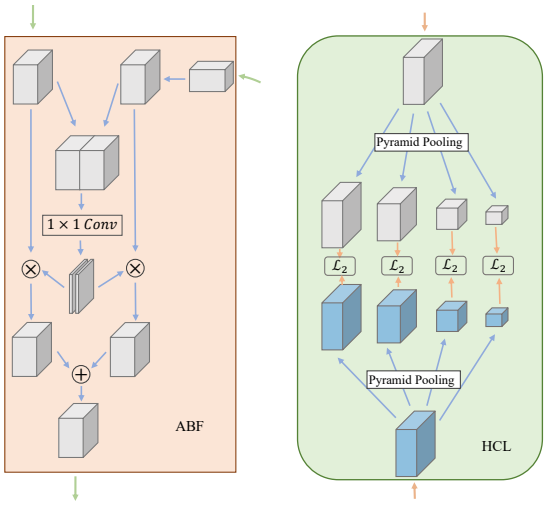
Our *review* mechanism is to use previous features to guide the current feature. The single-layer knowledge distillation with the review mechanism is formalized as

$$\mathcal{L}_{SKD\_R} = \sum_{j=1}^{i} \mathcal{D}\left(\mathcal{M}_s^{i,j}(\mathbf{F}_s^i), \mathcal{M}_t^{j,i}(\mathbf{F}_t^j)\right). \tag{5}$$

Although at the first glance it shares some similarity with multiple-layers knowledge distillation, it is in fact fundamentally different. Here feature of the student is fixed to $\mathbf{F}_s^i$, and we use the teacher's first i levels of features to guide $\mathbf{F}_s^i$. The review mechanism and multiple-layers distillation are complementary concepts. When combining the review mechanism with multiple-layers knowledge distillation, the loss function becomes

$$\mathcal{L}_{MKD\_R} = \sum_{i \in \mathbf{I}} \left(\sum_{j=1}^{i} \mathcal{D}\left(\mathcal{M}_s^{i,j}(\mathbf{F}_s^i), \mathcal{M}_t^{j,i}(\mathbf{F}_t^j)\right) \right). \tag{6}$$

In our experiments, the $\mathcal{L}_{MKD\_R}$ loss is simply added alone with original losses during the training process, and the inference is exactly the same as the original model. So our method is totally *cost-free at test time*. We use factor λ to



(a) (b)

Figure 3. (a) Architecture of ABF. Different levels' features of the student are aggregated together with attention maps. (b) Architecture of HCL. The student and teacher's features are pyramid pooled to extract different context information to distill.

balance the distillation loss and original losses. Taking the classification task as an example, the whole loss function is defined as

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{MKD\_R}. \tag{7}$$

In our proposed review mechanism, we only use shallower features of the teacher to supervise deeper features of the student. We found that the opposite brings marginal benefit and wastes many resources instead. The intuitive explanation is that deeper and more abstracted features are too complicated for early-stage learning. More analysis is in Section 4.4.

3.2. Residual Learning Framework

Following previous work, we first design a straightforward framework, as shown in Figure 2(a). The transformation $\mathcal{M}_s^{i,j}$ is simply composed of convolution layers and nearest interpolation layers to transfer the ith feature of the student to match the size of teacher's jth feature. We do not transform teacher features $\mathbf{F}_t$. The student feature is transformed into the same size as the teacher features.

Figure 2(b) shows directly applying the idea to multiple-layer distillation with all-stage features distilled. However, this strategy is not optimal because of the huge information difference between stages. Also, it yields a complicated process where all features are used. For instance, a network with n stages needs to calculate $n(n+1)/2$ pairs of features regarding the loss functions, which makes the learning process cumbersome and costs many resources.

To make the procedure more feasible and elegant, we reformulate Eq. (6) for Figure 2(b) as

$$\mathcal{L}_{MKD\_R} = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} \mathcal{D}\left(\mathbf{F}_s^i, \mathbf{F}_t^j\right) \right). \tag{8}$$

where the transform of features is omitted for simplicity. We now switch the order of two summations of i and j as

$$\mathcal{L}_{MKD\_R} = \sum_{j=1}^{n} \left(\sum_{i=j}^{n} \mathcal{D}\left(\mathbf{F}_s^i, \mathbf{F}_t^j\right) \right). \tag{9}$$

When j is fixed, Eq. (9) accumulates the distance between the teacher feature $\mathbf{F}_t^j$ and student features $\mathbf{F}_s^j$-$\mathbf{F}_s^n$. With fusion of features [40, 16], we approximate the summation of distance as the distance of fused features. It leads to

$$\sum_{i=j}^{n} \mathcal{D}\left(\mathbf{F}_s^i, \mathbf{F}_t^j\right) \approx \mathcal{D}\left(\mathcal{U}(\mathbf{F}_s^j, \cdots, \mathbf{F}_s^n), \mathbf{F}_t^j\right), \tag{10}$$

where $\mathcal{U}$ is a module to fuse features. This approximation is illustrated in Figure 2(c) where the structure is more effective now. But the calculation of fusion can be further optimized in a progressively manner as shown in Figure 2(d) for higher efficiency. Fusion of $\mathbf{F}_s^j, \cdots, \mathbf{F}_s^n$ is calculated by combination of $\mathbf{F}_s^j$ and $\mathcal{U}(\mathbf{F}_s^{j+1}, \cdots, \mathbf{F}_s^n)$, where the fusion operation is recursively defined as $\mathcal{U}(\cdot,\cdot)$, applied to consecutive feature maps. Denoting $\mathbf{F}_s^{j+1,n}$ as fusion of features from $\mathbf{F}_s^{j+1}$ to $\mathbf{F}_s^n$, the loss is written as

$$\mathcal{L}_{MKD\_R} = \mathcal{D}(\mathbf{F}_s^n, \mathbf{F}_t^n) + \sum_{j=n-1}^{1} \mathcal{D}\left(\mathcal{U}(\mathbf{F}_s^j, \mathbf{F}_s^{j+1,n}), \mathbf{F}_t^j\right), \tag{11}$$

Here we loop from $n-1$ down to 1 to make use of $\mathbf{F}_s^{j+1,n}$. $\mathbf{F}_s^{n,n} = \mathcal{M}_s^{n,n}(\mathbf{F}_s^n)$. The detailed structure is shown in Figure 2(d), where ABF and HCL are fusion module and loss function designed for this structure, respectively. Their details are discussed in Section 3.3.

The structure in Figure 2(d) is elegant and eases the distillation process with utilizing the concept of residual learning. For instance, the stage-4's feature of the student is aggregated with stage-3's feature of the student to mimic the stage-3's feature of the teacher. Therefore, stage-4's feature of the student learns the residual of stage-3's feature between the student and teacher. The residual information is very likely to be the key factor that the teacher yields higher-quality results.

This residual learning process is more stable and effective than directly letting high-level features of the student learned from low-level features of the teacher. With the residual learning framework, the high-level features of the student can better extract useful information progressively. Further, using Eq. (11), we eliminate the summation and reduce the total complexity to n pairs of distances.

Distillation Mechanism	Teacher Acc	ResNet56 72.34	ResNet110 74.31	ResNet32x4 79.42	WRN40-2 75.61	WRN40-2 75.61	VGG13 74.64
	Student Acc	ResNet20 69.06	ResNet32 71.14	ResNet8x4 72.50	WRN16-2 73.26	WRN40-1 71.98	VGG8 70.36
Logits	KD [9]	70.66	73.08	73.33	74.92	73.54	72.98
Single Layer	FitNet [25]	69.21	71.06	73.50	73.58	72.24	71.02
Single Layer	PKT [23]	70.34	72.61	73.64	74.54	73.54	72.88
Single Layer	RKD [22]	69.61	71.82	71.90	73.35	72.22	71.48
Single Layer	CRD [28]	71.16	73.48	75.51	75.48	74.14	73.94
Multiple Layers	AT [38]	70.55	72.31	73.44	74.08	72.77	71.43
Multiple Layers	VID [1]	70.38	72.61	73.09	74.11	73.30	71.23
Multiple Layers	OFD [8]	70.98	73.23	74.95	75.24	74.33	73.95
Review	Ours	**71.89**	**73.89**	**75.63**	**76.12**	**75.09**	**74.84**

Table 1. Results on CIFAR-100. The teacher and student have architectures of the same style.

Distillation Mechanism	Teacher Acc	ResNet32x4 79.42	WRN40-2 75.61	VGG13 74.64	ResNet50 79.34	ResNet32x4 79.42
	Student Acc	ShuffleNetV1 70.50	ShuffleNetV1 70.50	MobileNetV2 64.6	MobileNetV2 64.6	ShuffleNetV2 71.82
Logits	KD [9]	74.07	74.83	67.37	67.35	74.45
Single Layer	FitNet [25]	73.59	73.73	64.14	63.16	73.54
Single Layer	PKT [23]	74.10	73.89	67.13	66.52	74.69
Single Layer	RKD [22]	72.28	72.21	64.52	64.43	73.21
Single Layer	CRD [28]	75.11	76.05	69.73	69.11	75.65
Multiple Layers	AT [38]	71.73	73.32	59.40	58.58	72.73
Multiple Layers	VID [1]	73.38	73.61	65.56	67.57	73.40
Multiple Layers	OFD [8]	75.98	75.85	69.48	69.04	76.82
Review	Ours	**77.45**	**77.14**	**70.37**	**69.89**	**77.78**

Table 2. Results on CIFAR-100 with the teacher and student having different architectures.

3.3. ABF and HCL

There are two key components in Figure 2(d). They are attention based fusion (ABF) and hierarchical context loss (HCL). We explain them here.

ABF module utilizes the insight of [30, 12], as shown in Figure 3(a). The higher level features are first resized to the same shape as the lower level features. Then two features from different levels are concatenated together to generate two $H \times W$ attention maps. These maps are multiplied with two features, respectively. Finally, the two features are added to generate the final output.

The ABF module can generate different attention maps according to input features. So the two feature maps can be dynamically aggregated. The adaptive sum is better than direct sum because the two feature maps are from different stages of the network and their information is diverse. The low- and high-level features may focus on different partitions. The attention maps can aggregate them more reasonably. More experimental results are included in Section 4.4.

The detail of HCL is shown in Figure 3(b). Usually, we use $\mathcal{L}_2$ distance as the loss function between the two feature maps. The $\mathcal{L}_2$ distance is effective to transfer information between features from the same level. But in our framework, different levels' information is aggregated together to learn from the teacher. The trivial global $\mathcal{L}_2$ distance is not powerful enough to transfer compound levels' information.

Inspired by [41], we propose HCL, utilizing spatial pyramid pooling, to separate the transfer of knowledge into different levels' context information. In this way, the information is better distilled in different abstract levels. The structure is very simple: we first extract different levels' knowledge from the feature using spatial pyramid pooling,

Setting		Teacher	Student	KD [9]	AT [38]	OFD [8]	CRD [28]	Ours
(a)	Top-1	76.16	68.87	68.58	69.56	71.25	71.37	**72.56**
	Top-5	92.86	88.76	88.98	89.33	90.34	90.41	**91.00**
(b)	Top-1	73.31	69.75	70.66	70.69	70.81	71.17	**71.61**
	Top-5	91.42	89.07	89.88	90.01	89.98	90.13	**90.51**

Table 3. Results on ImageNet. (a) MobileNet as student, ResNet50 as teacher. (b) ResNet18 as student, ResNet34 as teacher.

and then use $\mathcal{L}_2$ distance to distill between them respectively. Despite the simple structure, HCL is suitable for our framework. More experimental results are shown in Section 4.4.

4. Experiments

We conduct experiments on various tasks. First, we compare our method with other knowledge distillation ones regarding classification. We experiment with different settings varying architecture and datasets. Also, we apply our method to the object detection and instance segmentation tasks. Our method also improves the baseline model by large margins consistently.

4.1. Classification

Datasets (1) CIFAR-100 contains 50K training images with 0.5K images per class and 10K test images. (2) ImageNet [3] is the most challenging dataset for classification, which provides 1.2 million images for training and 50K images for validation over 1,000 classes.

Implementation Details On CIFAR-100 dataset, we experiment with different representative network architectures, including VGG [27], ResNet [7], WideResNet [37], MobileNet [26], and ShuffleNet [39, 21]. We use the same training setting of [28], except for linearly scaling up the initial learning rate and setting batch size following [5].

Specifically, we train all models for 240 epochs with learning rate decayed by 0.1 for every 30 epochs after the first 150 epochs. We initialize the learning rate to 0.02 for MobileNet and ShuffleNet, and 0.1 for other models. The batch size is 128 for all models. We train all models for three times and report the mean accuracy. For fairness, previous method results are either reported in previous papers (when the training setting is the same as ours) or obtained using author released codes with our training setting.

On ImageNet, we use the standard training process that trains the model for 100 epochs and decays the learning rate for every 30 epochs. We initialize learning rate to 0.1 and set batch size to 256.

Results on CIFAR-100 Table 1 summarizes results on CIFAR-100 with the teacher and student having architectures of the same style. We separate previous methods in different groups according to the features they use. KD is the only method that uses logits. Methods in FitNet group use single-layer information, and methods in AT group use multiple-layer information. Our method employs multi-layer feature with the review mechanism. It outperforms all previous methods on all architectures.

We also experiment with the setting that the student and teacher have different architectural styles, and show results in Table 2. Method of OFD [8] and ours use multiple layers for distillation. They outperform those with distillation from the last layer, manifesting that our knowledge review mechanism successfully relaxes previously emphasized intermediate- or last-layer distillation condition [28].

Results on ImageNet The number of images in CIFAR-100 is small. So we also conduct experiments on ImageNet to verify the scalability of our method. We experiment with two settings of distillation from ResNet50 to MobileNet [11], and from ResNet34 to ResNet18 respectively. Our method, again, outperforms all other methods, as reported in Table 3. Setting (a) is challenging due to architecture difference. But the advantage of our method is consistently prominent. On setting (b), gap between the student and teacher is already reduced to a very small value 2.14 by previous best method. We further reduce it to 1.70, achieving 20% relative performance improvement.

4.2. Object Detection

We also apply our method to other computer vision tasks. On object detection, like the procedure for the classification task, we distill between the student and teacher's backbone output features. More details are presented in the supplementary file. We use the representative COCO2017 dataset [18] to evaluate our method and take the most popular open-source report Detectron2 [33] as our strong baseline. We use the best pre-trained model provided by Detrctron2 as teacher. Student models are trained using the standard training policy following tradition [31]. All performance is evaluated on COCO2017 validation set. We conduct experiments on both two- and one-stage methods.

Since only a few methods [31, 8] are claimed workable for detection, we reproduce the popular ones [9, 25] and the latest one [31]. The comparison is presented in Table 4. We note that knowledge distillation methods, such as KD and

	Method	mAP	AP50	AP75	APl	APm	APs
Teacher	Faster R-CNN w/ R101-FPN	42.04	62.48	45.88	54.60	45.55	25.22
Student	Faster R-CNN w/ R18-FPN	33.26	53.61	35.26	43.16	35.68	18.96
	w/ KD [9]	33.97 (+0.61)	54.66	36.62	44.14	36.67	18.71
	w/ FitNet [25]	34.13 (+0.87)	54.16	36.71	44.69	36.50	18.88
	w/ FGFI [31]	35.44 (+2.18)	55.51	38.17	47.34	38.29	19.04
	w/ Our Method	**36.75 (+3.49)**	**56.72**	**34.00**	**49.58**	**39.51**	**19.42**
Teacher	Faster R-CNN w/ R101-FPN	42.04	62.48	45.88	54.60	45.55	25.22
Student	Faster R-CNN w/ R50-FPN	37.93	58.84	41.05	49.10	41.14	22.44
	w/ KD [9]	38.35 (+0.42)	59.41	41.71	49.48	41.80	22.73
	w/ FitNet [25]	38.76 (+0.83)	59.62	41.80	50.70	42.20	22.32
	w/ FGFI [31]	39.44 (+1.51)	60.27	43.04	51.97	42.51	22.89
	w/ Our Method	**40.36 (+2.43)**	**60.97**	**44.08**	**52.87**	**43.81**	**23.60**
Teacher	Faster R-CNN w/ R50-FPN	40.22	61.02	43.81	51.98	43.53	24.16
Student	Faster R-CNN w/ MV2-FPN	29.47	48.87	30.90	38.86	30.77	16.33
	w/ KD [9]	30.13 (+0.66)	50.28	31.35	39.56	31.91	16.69
	w/ FitNet [25]	30.20 (+0.73)	49.80	31.69	39.69	31.64	16.39
	w/ FGFI [31]	31.16 (+1.69)	50.68	32.92	42.12	32.63	16.73
	w/ Our Method	**33.71 (+4.24)**	**53.15**	**36.13**	**46.47**	**35.81**	**16.77**
Teacher	RetinaNet101	40.40	60.25	43.19	52.18	44.34	24.03
Student	RetinaNet50	36.15	56.03	38.73	46.95	40.25	21.37
	w/ KD [9]	36.76 (+0.61)	56.60	39.40	48.17	40.56	21.87
	w/ FitNet [25]	36.30 (+0.15)	55.95	38.95	47.14	40.32	20.10
	w/ FGFI [31]	37.29 (+1.14)	57.13	40.04	49.71	41.47	21.01
	w/ Our Method	**38.48 (+2.33)**	**58.22**	**41.46**	**51.15**	**42.72**	**22.67**

Table 4. Results on object detection. We use AP on different settings to evaluate results. R101 represents using ResNet101 as backbone, and MV2 stands for MobileNetV2.

	Method	mAP	AP50	AP75	APl	APm	APs
Teacher	Mask R-CNN w/ R101-FPN	38.63	60.45	41.28	55.29	41.33	19.48
Student	Mask R-CNN w/ R18-FPN	31.25	51.07	33.10	45.53	32.80	14.18
	+ Our Method	**33.62 (+2.37)**	**53.91**	**35.96**	**50.30**	**35.31**	**15.03**
Teacher	Mask R-CNN w/ R101-FPN	38.63	60.45	41.28	55.29	41.33	19.48
Student	Mask R-CNN w/ R50-FPN	35.24	56.32	37.49	50.34	37.71	17.16
	+ Our Method	**36.98 (+1.74)**	**58.13**	**39.60**	**53.19**	**39.57**	**17.54**
Teacher	Mask R-CNN w/ R50-FPN	37.17	58.60	39.88	53.30	39.49	18.63
Student	Mask R-CNN w/ MV2-FPN	28.37	47.19	29.95	41.70	29.01	12.09
	+ Our Method	**31.56 (+3.19)**	**50.70**	**33.44**	**47.39**	**32.44**	**12.76**

Table 5. Instance segmentation results. R101 and MV2 stand for ResNet101 and MobileNetV2.

FitNet, also improve the performance of detection. But the gain is limited. FGFI [31] is directly designed for detection, and works better than other methods on this task. Still, our method outperforms it by a large margin.

We also vary experimental setting to check the generality. On the two-stage method FasterRCNN [24], we change backbone architectures. The knowledge distillation between architectures of the same style boosts mAP of ResNet18 and ResNet50 by 3.49 and 2.43 respectively. They are significant numbers. The distillation between ResNet50 and MobileNetV2 still promotes the baseline from 29.47 to 33.71. On one-stage detector RetinaNet [17], the gap between student and teacher is small, our method also improves the mAP by 2.33. The success on challenging object detection tasks demonstrates the generality and effectiveness of our method.

		Teacher Stage			
		1	2	3	4
Student Stage	1	69.5	69.0	68.2	66.3
	2	69.6	69.6	61.4	61.1
	3	69.2	69.8	71.0	50.4
	4	69.2	69.3	70.3	70.3

Table 6. Results of knowledge distillation between different stages of the teacher and student. The student's baseline result is 69.1. We use red color to mark numbers lower than baseline and blue for those higher than baseline. It is clear that using the lower level information of the teacher to supervise the deeper stage of the student is helpful.

4.3. Instance Segmentation

In this section, we apply our method to the even more challenging instance segmentation task. As far as we know, this is the first time for the knowledge distillation methods to apply to instance segmentation. We also use the strong baseline provided by Detectron2 [33]. We take Mask R-CNN [6] as our models and distill between different backbone architectures. The models are trained on the COCO2017 training set and are evaluated on the validation set. The results are shown in Table 5.

Our method also improves the performance of instance segmentation tasks notably. For distillation between architectures of the same style, we boost the performance of ResNet18 and ResNet50 by 2.37 and 1.74, and reduce the gap between the teacher and student by 32% and 51% relatively. Even for the distillation on architectures of different styles, we better MobileNetV2 by 3.19.

The fact that our method performs decently on all image classification, object detection, and instance segmentation tasks and accomplishes all SOTA results, manifest the remarkable efficacy and applicability of our method.

4.4. More Analysis

Knowledge Distillation across Stages We analyze the effectiveness of knowledge transfer across stages. We use ResNet20 as the student and ResNet56 as the teacher on CIFAR-100 dataset. There are four stages in ResNet20 and ResNet56. We choose the different stages in the student and vary stages in the teacher to supervise them. The results are summarized in Table 6 .

These results conclude that distilling student with the same stage information from the teacher is the best solution. This is in accordance with our intuition. Further, it is intriguing to observe that information from lower layers is also helpful. But distilling from teacher's higher levels adversely affects training of the student.

It indicates that deeper stages of the student are capable of learning useful information from lower stages of the teacher. In the other way around, deeper and more abstracted features from teacher are too complicated for early-stage of the student. This is consistent with our understanding and our proposed review mechanism, which uses shallow stages of the teacher to supervise deeper stages of the student.

RM	RLF	ABF	HCL	Accuracy
				74.3 ± 5e-2
✔				75.2 ± 6e-2
✔	✔			75.6 ± 6e-2
✔	✔	✔		76.0 ± 6e-2
✔	✔		✔	75.8 ± 5e-2
✔	✔	✔	✔	76.2 ± 4e-2

Table 7. RM: The proposed review mechanism (Section 3.1). RLF: Residual learning frame work (Section 3.2). ABF: Attentation based fusion module (Section 3.3). HCL: Hierarchical context loss function (Section 3.3).

Ablation Study Ablation experiments are conducted, in which the ablation components are added one-by-one to measure their effect. The results are summarized in Table 7 with accuracy and variance. We use WRN16-2 as the student and WRN40-2 as the teacher on CIFAR100 dataset. The baseline is trained with $\mathcal{L}_2$ distance between the same level's features of the student and the teacher.

With our proposed review mechanism, the result is improved over the baseline, as shown in the second line, which uses the trival structure as shown in Figure 2(b). When we further refine the structure with the residual learning framework, the student yields larger gains. The attention based fusion module and hierarchical context loss function also provide great improvement when utilized separately. And when we aggregate them together, the best results are obtained. It is surprising that they are even better than the teacher.

5. Conclusion

In this paper, we have studied knowledge distillation from a new perspective and accordingly proposed the review mechanism, which uses multiple layers in the teacher to supervise one layer in the student. Our method achieves significant improvement consistently on all classification, object detection and instance segmentation tasks, compared with all previous SOTA. We only use output of stages, and already accomplish decent results in general.

For future work, we will also employ features inside a stage. Also, other loss functions will be investigated in our framework.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, 2019. 5

[2] Jiequan Cui, Pengguang Chen, Ruiyu Li, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast and practical neural architecture search. In *ICCV*, 2019. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[4] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019. 1

[5] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. 6

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017. 8

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[8] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, 2019. 2, 5, 6

[9] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 1, 2, 5, 6, 7

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 1

[11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 1, 6

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5

[13] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, 2018. 1

[14] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NIPS*, 2018. 3

[15] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2017. 1

[16] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[17] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 7

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 6

[19] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *ICLR*, 2019. 1

[20] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *ICCV*, 2017. 1

[21] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018. 6

[22] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019. 2, 5

[23] Nikolaos Passalis and Anastasios Tefas. Probabilistic knowledge transfer for deep representation learning. *CoRR*, abs/1803.10837, 2018. 1, 2, 5

[24] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015. 7

[25] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 1, 2, 5, 6, 7

[26] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 6

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[28] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020. 1, 2, 5, 6

[29] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019. 2

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NIPS*, 2017. 5

[31] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *CVPR*, 2019. 6, 7

[32] Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan. Quantization mimic: Towards very tiny CNN for object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018. 1

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6, 8

[34] Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. Autoprune: Automatic network pruning by regularizing auxiliary parameters. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NIPS*, 2019. 1

[35] Louis E Yelle. The learning curve: Historical review and comprehensive survey. *Decision sciences*, 1979. 1

[36] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017. 2

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6

[38] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 1, 2, 3, 5, 6

[39] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 6

[40] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018. 4

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 5