

Refine Myself by Teaching Myself : Feature Refinement via Self-Knowledge Distillation

Mingi Ji¹ Seungjae Shin¹ Seunghyun Hwang² Gibeom Park¹ Il-Chul Moon^{1,3}

¹Korea Advanced Institute of Science and Technology (KAIST) ²Looko Inc. ³Summary.AI

qwertgfdcvb@kaist.ac.kr tmdwo0910@kaist.ac.kr shhwang@acloset.app pgb1227@kaist.ac.kr icmoon@kaist.ac.kr

Abstract

Knowledge distillation is a method of transferring the knowledge from a pretrained complex teacher model to a student model, so a smaller network can replace a large teacher network at the deployment stage. To reduce the necessity of training a large teacher model, the recent literatures introduced a self-knowledge distillation, which trains a student network progressively to distill its own knowledge without a pretrained teacher network. While Self-knowledge distillation is largely divided into a data augmentation based approach and an auxiliary network based approach, the data augmentation approach loses its local information in the augmentation process, which hinders its applicability to diverse vision tasks, such as semantic segmentation. Moreover, these knowledge distillation approaches do not receive the refined feature maps, which are prevalent in the object detection and semantic segmentation community. This paper proposes a novel self-knowledge distillation method, Feature Refinement via Self-Knowledge Distillation (FRSKD), which utilizes an auxiliary self-teacher network to transfer a refined knowledge for the classifier network. Our proposed method, FRSKD, can utilize both soft label and feature-map distillations for the self-knowledge distillation. Therefore, FRSKD can be applied to classification, and semantic segmentation, which emphasize preserving the local information. We demonstrate the effectiveness of FRSKD by enumerating its performance improvements in diverse tasks and benchmark datasets. The implemented code is available at <https://github.com/MingiJi/FRSKD>.

1. Introduction

Deep neural networks (DNNs) have been applied to various fields of computer vision due to the exponential advancement of convolutional neural networks [7, 27, 12]. To distribute the success at the mobile devices, the vision task needs to overcome the limited computing re-

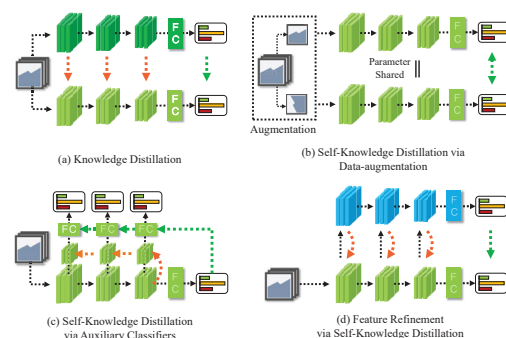


Figure 1: Comparison of various distillation methods. The black line is the forward path; the green line is the soft label distillation; and the orange line is the feature distillation. (a) Conventional knowledge distillation method with pretrained teacher [9, 26, 36, 14, 1]. (b) Self-knowledge distillation method via data augmentation [32, 35, 18]. (c) Auxiliary weak classifier based self-knowledge distillation, which creates a set of layer-wise classifiers to generate a backpropagation signal at each layer, and the layer-wise classifier produce its estimation from the layer’s feature distillations of the orange line and the logit distillations of the green line [39]. (d) Our proposed method. The original classifier provides original feature as an input to the auxiliary self-teacher network (blue blocks). Afterwards, the self-teacher network distills the refined feature-map to the original classifier (orange lines).

sources [11, 41]. To solve this problem, the model compression has been a crucial research task, and knowledge distillation has been a prominent technology with a good compression and equivalent performances [9].

Knowledge distillation is a method of transferring the knowledge from a pretrained teacher network to a student network, so a smaller network can replace a large teacher network at the deployment stage. Knowledge distillation utilizes the teacher’s knowledge through receiving either 1) class predictions as soft labels [9]; 2) penultimate layer outputs [29, 21, 23], or 3) feature-maps including spatial in-

formation at the intermediate layer [26, 2, 8]. Whereas the knowledge distillation enables utilizing the larger network in a condensed manner, the inference on such large network, a.k.a. the teacher network, becomes an ultimate burden of its practical usages. In addition, pretraining the large network requires substantial computational resources to prepare the teacher network.

To reduce such necessity of training a large network, the recent literatures introduce an alternative knowledge distillation [6, 42]; which is a distillation from a pretrained network with the same architecture of the student network. This knowledge distillation is still known to be informative to the student network with the same scale. Moreover, there are literatures on a self-knowledge distillation, which trains the student network progressively to distill and to regularize its own knowledge without the pretrained teacher network [43, 35, 18, 32, 39]. The self-knowledge distillation is different from the previous knowledge distillation because it does not require a prior preparation of the teacher network.

Self-knowledge distillation is largely divided into a data augmentation based approach and an auxiliary network based approach. The data augmentation based self-knowledge distillation induces a consistent prediction of relevant data, i.e. the different distorted versions of a single instance or a pair of instances from the same class [32, 35, 18]. The auxiliary network based approach utilizes additional branches in the middle of the classifier network, and the additional branches are induced to make similar outputs via knowledge transfer [43, 39]. However, these approaches depend on the auxiliary network, which has the same or less complexity than classifier network; so it is hard to generate a refined knowledge, either by features, which are the output of the convolutional layers, or soft labels, for the classifier network [32, 35, 18, 43, 39]. Furthermore, the data augmentation based approaches are vulnerable to lose the local information between instances, such as differently distorted instances or rotated instances. Therefore, it is hard to utilize the feature distillation, which is well known technique for improving the performances in general knowledge distillations [32, 35, 18].

To deal with the limitation of existing self-knowledge distillation, this paper proposes a novel self-knowledge distillation method, Feature Refinement via Self-Knowledge Distillation (FRSKD), which introduces an auxiliary self-teacher network to enable the transfer of a refined knowledge to the classifier network. Figure 1 shows the difference between FRSKD and existing knowledge distillation methods. Our proposed method, FRSKD, can utilize both soft label and feature-map distillations for the self-knowledge distillation.

Therefore, FRSKD can be applied to classification and semantic segmentation, which emphasize preservation of the local information. FRSKD shows the state-of-the-

art performances in image classification task on various datasets, compared to other self-knowledge distillation methods. In addition, FRSKD improves the performances on semantic segmentation. Besides, FRSKD is compatible with other self-knowledge distillation methods as well as data augmentation. We demonstrate the compatibility of FRSKD with large performance improvements through various experiments.

2. Related Work

Knowledge distillation The goal of *knowledge distillation* is to effectively train a simpler network, a.k.a. a student network, by transferring the knowledge of a pretrained complex network, a.k.a. teacher network. Here, knowledge includes the features at the hidden layers or the logits at the final layer, etc. Hinton et al. proposed a method of knowledge distillation by transferring the teacher network's output logit to the student network [9]. The intermediate layer distillation methods were then introduced, so such methods utilize the teacher network's knowledge from either the convolutional layer with feature-map level preserving localities [26, 36, 14, 34, 16]; or penultimate layer [24, 29, 21, 23, 30]. For the feature-map distillations, the prior works induced the student to imitate 1) feature of the teacher network [26], 2) abstracted attention map of the teacher network [36], or 3) FSP matrix of the teacher network [34]. For the penultimate layer distillations, the existing literature utilized the relation between instances as knowledge, i.e. the cosine similarity between feature sets at the same penultimate layer from pair of instances [24, 29, 21, 23, 30]. In addition, prior experiments concluded that these distillation methods on different layers perform better when different distillations are jointly applied. However, there are two distinct limitations: 1) knowledge distillation requires pretraining of the complex teacher model, and 2) variation of the teacher networks will result in different performances with the same student network.

Self-knowledge distillation *Self-knowledge distillation* enhances the effectiveness of training a student network by utilizing its own knowledge without a teacher network. First, some approaches utilize an auxiliary network for self-knowledge distillation. For example, BYOT introduced a set of auxiliary weak classifier networks that classify the output with the features from the middle hidden layers [39]. The weak classifier networks of BYOT are trained by the joint supervision of estimated logit values and true supervision. ONE utilized additional branches to diversify model parameters and estimated features at the middle layer. This diversification is aggregated through the ensemble method, and the ensembled output creates the joint back-propagation signal shared by the branches [43]. The commonality of these auxiliary network approaches is the utilization of Ad-hoc structure at the same level or weak classifier network

at the middle layer. Hence, these approaches without enhanced network may suffer from a lack of more refined knowledge.

Secondly, data augmentation was used for self-knowledge distillation as well. DDGSD induced the consistent prediction by providing differently augmented instances, so the classifier network would face variations of instances [32]. CSKD used logits of other instances belonging to the same class for regularization purposes, so the classifier network would predict similar outcomes for the same classes [35]. However, the data augmentations do not necessarily preserve the spatial information, i.e. simple flipping would ruin the feature locality, so the feature-map distillation is difficult to be applied in the line of data augmentations. SLA proposed augmenting the data label by combining the self-supervision task with the original classification task. The self-supervision takes augmentations such as input rotations, and the ensembled output of the augmented instances provides additional supervision for the backpropagation [18].

Ideally, the feature-map distillation can be improved by extracting the refined knowledge from a complex model. However, the auxiliary structure of the prior works did not provide such a method to make feature more complex. On the contrary, the data augmentation may increase data variations or refinements in some cases, but their variability can hinder the feature-map distillation because the variation will prevent consistent locality modelling from the parameter side. Therefore, we conjecture that supplying the refinements to the feature-map distillation can be a breakthrough in the auxiliary network-based self-knowledge distillation. Hence, we suggest the auxiliary self-teacher network to generate a refined feature-map as well as its soft label. To our knowledge, this paper is the first work on a self-teacher network to generate a refined feature-maps from a single instance.

Feature networks Our suggested structure of the auxiliary self-teacher network is developed from the feature networks used in the object detection field. The aggregation of feature with various scales is one of key for processing multi-scaled features, and the object detection field has investigated this scale variations in [19, 15, 40, 20, 28]. Our auxiliary self-teacher network generates a refined feature-map by adapting the network that deals with multi scaled features to the knowledge distillation purpose. While we will explain the adaptation of this structure in Section 3.1, this subsection enumerates the recent developments of the feature networks. FPN utilized a top-down network to simultaneously exploit 1) abstract information from the upper layers and 2) information on small objects at the lower layers of the backbone network [19]. PANet introduced additional bottom-up network for FPN to enable a short path connection between a layer for detection and a layer

of backbone [20]. BiFPN proposed a more efficient network structure, using top-down and bottom-up networks as same as PANet [28]. This paper proposes an auxiliary self-teacher network, which is altered from the structure of BiFPN to be appropriate for the classification task. In addition, the feature-map distillation has become efficient by the alteration of the network structure because the self-teacher network requires fewer computations than BiFPN by varying the channel dimension according to the depth of the feature-maps.

3. Method

This section introduces a feature refinement self-knowledge distillation (FRSKD). Figure 2 shows the overview of our distillation method, which is further discussed in Section 3.1 from the perspective of the self-teacher network. Then, we review the training procedure of our self-knowledge distillation in Section 3.2.

Notations Let $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ be a set of labeled instances where N is its size; let $F_{i,j}$ be a feature map of the j -th block of the classifier network for the i -th sample; and let c_j be a channel dimension of the j -th block of the classifier network. For notation simplicity, we omit the index i in the rest of this paper.

3.1. Self-Teacher Network

The main purpose of the self-teacher network is providing a refined feature-map and its soft label for the classifier network to itself. The inputs of the self-teacher network are the feature-maps of the classifier network, F_1, \dots, F_n , which assumes the n blocks of the classifier network. We model the self-teacher network by modifying the structure of BiFPN for the classification task. Specifically, we adapt the top-down path and the bottom-up path from PANet and BiFPN [28, 20]. Before the top-down path, we utilize the lateral convolutional layers as following:

$$L_i = \text{Conv}(F_i; d_i) \quad (1)$$

Conv is a convolutional operation with an output dimension of d_i . Unlike the existing lateral convolutional layers with d_i dimensions that is fixed at the network setup, we design d_i that depends upon the channel dimension of the feature map, c_i . We set d_i as $d_i = w \times c_i$ with a channel width parameter, w . For the classification task, it is natural to set a higher channel dimension for the deeper layers. Therefore, we adjust the channel dimension of each layer to contain information of its feature-map depth. Also, this design reduces the computation of the lateral layers.

The top-down path and the bottom-up path aggregate different features as the below:

$$P_i = \text{Conv}(w_{i,1}^P \cdot L_i + w_{i,2}^P \cdot \text{Resize}(P_{i+1}); d_i) \quad (2)$$

$$T_i = \text{Conv}(w_{i,1}^T \cdot L_i + w_{i,2}^T \cdot P_i + w_{i,3}^T \cdot \text{Resize}(T_{i-1}); d_i)$$

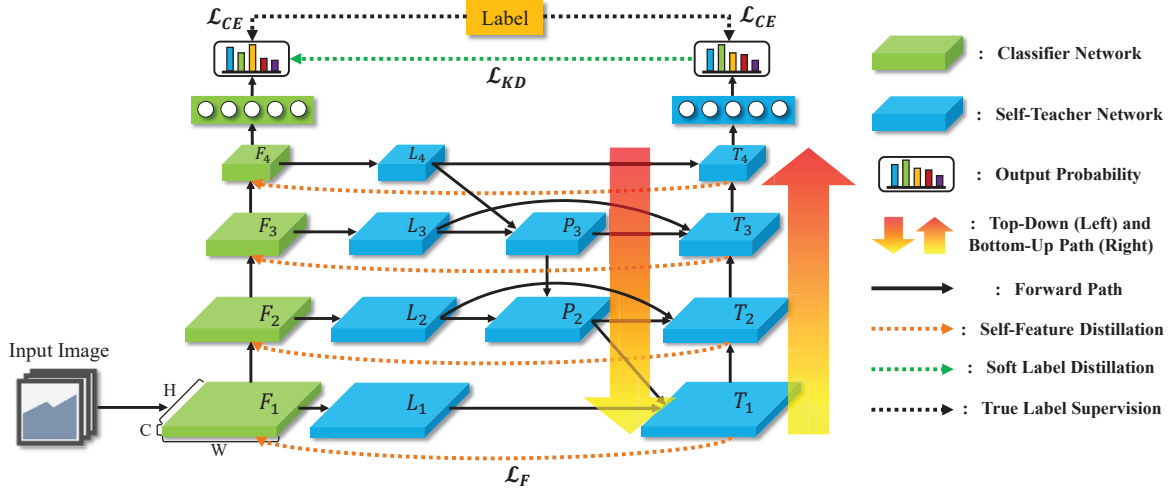


Figure 2: Overview of our proposed self-knowledge distillation method, Feature Refinement via Self-Knowledge Distillation (FRSKD). The top-down path and the bottom-up path aggregates different sized features and provide the refined feature-map to the original classifier network. Exploiting feature-map of the self-teacher network, FRSKD performs distillation on refined feature-map and soft label.

P_i represents the i -th layer of the top-down path; and T_i is the i -th layer of the bottom-up path. Similar to BiFPN [28], the forward pass connection has a different structure depending on the depth of the layers. In Figure 2, in the case of the shallowest bottom-up path layer T_1 and the deepest bottom-up path layer T_4 , each directly utilizes the lateral layer L_1 and L_4 as inputs respectively for efficiency, instead of using the features of the top-down path. In these settings, to create a top-down structure that connects all the shallowest layer, middle layer and deepest layer, the two diagonal connections for forward propagations are added: 1) the connection from the last lateral layer, L_4 , to the penultimate layer of the top-down path, P_3 ; and 2) the connection from the P_2 , to the first layer of the bottom-up path, T_1 .

We apply a fast normalized fusion with parameters, such as w^P and w^T [28]. We use a bilinear interpolation for the up-sampling, and we use the max-pooling for the down-sampling, as *Resize* operator. For efficient calculations, we use a depth-wise convolution for convolutional operations [11]. We conduct various experiments, and we analyze the results, according to the self-teacher network structure in Section 4.3. Finally, we attach the fully connected layer on the top of the bottom-up path to predict the output class, and the self-teacher network provides its soft label, $\hat{p}_t = \text{softmax}(f_t(\mathbf{x}; \theta_t))$ where f_t denotes the self-teacher network, parameterized by θ_t .

3.2. Self-Feature Distillation

Our proposed model, FRSKD, utilizes the output of the self-teacher network, the refined feature-map, T_i , and the

soft label, \hat{p}_t . Firstly, we add the feature distillation, which induces the classifier network to mimic the refined feature-map. For feature distillations, we adapt the attention transfer [36]. Equation 3 defines the feature distillation loss, \mathcal{L}_F :

$$\mathcal{L}_F(T, F; \theta_c, \theta_t) = \sum_{i=1}^n \|\phi(T_i) - \phi(F_i)\|_2 \quad (3)$$

where ϕ is a combination of channel-wise pooling function with L_2 normalization [36]; and θ_c is parameter of the classifier network. ϕ abstracts the spatial information of the feature-map. Therefore, \mathcal{L}_F makes the classifier network learn the locality of the refined feature-map from the self-teacher network. In addition, it is possible to train a classifier network to mimic the refined feature-map exactly [26, 8]; or utilize the transformation of the feature-map [34]. Unless noted, this paper utilizes the attention transfer-based feature distillation, and we discuss the methodology of the feature distillation in Section 4.3.

Similar to other self-knowledge distillation methods, FRSKD also performs the distillation through the soft label, \hat{p}_t , as following:

$$\begin{aligned} \mathcal{L}_{KD}(\mathbf{x}; \theta_c, \theta_t, K) \\ = D_{KL}(\text{softmax}(\frac{f_c(\mathbf{x}; \theta_c)}{K}) || \text{softmax}(\frac{f_t(\mathbf{x}; \theta_t)}{K})) \end{aligned} \quad (4)$$

where f_c is the classifier network; and K is the temperature scaling parameter. Also, the classifier network and the self-teacher network learn a ground-truth label using a cross-entropy loss, \mathcal{L}_{CE} . By integrating the loss functions above,

we construct the following optimization objective:

$$\begin{aligned} \mathcal{L}_{FRSKD}(\mathbf{x}, y; \theta_c, \theta_t, K) \\ = \mathcal{L}_{CE}(\mathbf{x}, y; \theta_c) + \mathcal{L}_{CE}(\mathbf{x}, y; \theta_t) \\ + \alpha \cdot \mathcal{L}_{KD}(\mathbf{x}; \theta_c, \theta_t, K) + \beta \cdot \mathcal{L}_F(T, F; \theta_c, \theta_t) \end{aligned} \quad (5)$$

where α and β are hyperparameters; and we choose $\alpha \in [1, 2, 3]$ and $\beta \in [100, 200]$; further details are explained in Appendix *Sensitivity Analysis*. Optimization is initiated by backpropagations at the same time for both classifier and self-teacher networks. To prevent the model collapse issue [22], FRSKD updates the parameters by the distillation loss, \mathcal{L}_{KD} and \mathcal{L}_F , which is only applied to the student network.

4. Experiments

We evaluate our self-knowledge distillation method on various tasks: classification and semantic segmentation. Throughout this section, we mainly use three settings; utilizing the distillation of soft label only (FRSKD\F); optimizing by \mathcal{L}_{FRSKD} with the distillation of refined feature-map and its soft label (FRSKD); and attaching the data augmentation based on the self-knowledge distillation, SLA-SD, with our method (FRSKD+SLA) [18].

4.1. Classification

Datasets We demonstrate the effectiveness of FRSKD on various classification datasets: CIFAR-100 [17], TinyImageNet, Caltech-UCSD Bird (CUB200) [31], MIT Indoor Scene Recognition (MIT67) [25], Stanford 40 Actions (Stanford40) [33], Stanford Dogs (Dogs) [13], and ImageNet [3]. Cifar-100 and TinyImageNet consist of small scaled images, and we resized TinyImageNet images to meet the same size of CIFAR-100 (32×32). CUB200, MIT67, Stanford40 and Dogs are datasets for the fine-grained visual recognition (FGVR) tasks. FGVR contains fewer data instances per class compared to CIFAR-100 and ImageNet. ImageNet is a large scaled dataset to validate our method to practically test the model.

Implementation details We use ResNet18 and WRN-16-2 [7, 37] for CIFAR-100 and TinyImageNet. To adapt ResNet18 to small sized datasets, we modify the first convolution layer of ResNet18 as a kernel size of 3×3 , a single stride, and a single padding. We remove a max-pooling operation, as well. We use the standard ResNet18 for FGVR tasks; and we apply both ResNet18 and ResNet34 to ImageNet.

For all classification experiments, we use stochastic gradient descents (SGD) with an initial learning rate of 0.1 and weight decay as 0.0001. We set total epochs as 200, and we divide the learning rate by 10 at epoch 100 and 150 for CIFAR-100, TinyImageNet and FGVR. For ImageNet, we set total epochs as 90, and we divide the learning rate

by 10 at epoch 30 and 60. We set the batch size as 128 for CIFAR-100 and TinyImageNet; 32 for FGVR; and 256 for ImageNet. We use standard data augmentation methods for all experiments, i.e. random cropping and flipping. In terms of hyperparameters, we set α as two and β as 100 for CIFAR-100; α as three and β as 100 for TinyImageNet; and α as one and β as 200 for FGVR and ImageNet. Additionally, we set the temperature scaling parameter, K , as four; and we set the channel width parameter, w , as two for all experiments. Further details are enumerated in Appendix *Implementation Details*.

Baselines We compare FRSKD to a standard classifier (named as Baseline), which doesn't utilize the distillation, with cross entropy based loss and six self-knowledge distillation methods, which make a total of seven baselines.

- **ONE** [43] exploits an ensembled prediction of additional branches as the soft label.
- **DDGSD** [32] generates different distorted versions of a single instance, and DDGSD trains to produce consistent prediction for the distorted data.
- **BYOT** [39] applies auxiliary classifiers utilizing the outputs of intermediate layers, and BYOT trains auxiliary classifiers by ground-truth labels and signals from network itself, such as the predicted logit or the feature-map.
- **SAD** [10] focuses on a lane detection by a layer-wise attention distillation in network itself.
- **CS-KD** [35] forces a consistent prediction for the same class by utilizing the prediction of other instances within the same class as the soft label.
- **SLA-SD** [18] trains a network with an original classification task and a self-supervised task jointly by utilizing the label augmentation. SLA-SD exploits an aggregated prediction as the soft label.

We utilize the available official code for implementation [35, 18, 43]. Otherwise, we implement models according to the corresponding papers. We apply same training setting according to the dataset, and we tune the hyperparameters of baseline models.

Performance comparison Table 1 shows the classification accuracy on CIFAR-100 and TinyImageNet with two different classifier network structures. Most of the self-knowledge distillation methods improve the performance of the standard classifier. Compared to baselines, FRSKD consistently shows better performance than other self-knowledge distillation methods. Furthermore, FRSKD\F, which does not utilize the feature distillation, shows better performance than other baselines. This result shows that the soft label of the self-teacher network outperforms

Methods	CIFAR100		TinyImageNet	
	WRN-16-2	ResNet18	WRN-16-2	ResNet18
Baseline	70.42±0.08	73.80±0.60	51.05±0.20	54.60±0.33
ONE	73.01±0.23	76.67±0.66	52.10±0.20	57.53±0.39
DDGSD	71.96±0.05	76.61±0.47	51.07±0.24	56.46±0.24
BYOT	70.22±0.26	76.68±0.07	50.33±0.03	56.61±0.30
SAD	70.31±0.45	74.65±0.33	51.26±0.39	54.45±0.06
CS-KD	71.79±0.68	77.19±0.05	50.08±0.18	56.46±0.10
SLA-SD	73.00±0.45	77.52±0.30	50.77±0.33	58.48±0.44
FRSKD\F	73.12±0.06	77.64±0.12	<u>52.91±0.30</u>	59.50±0.15
FRSKD	<u>73.27±0.45</u>	<u>77.71±0.14</u>	53.08±0.33	<u>59.61±0.31</u>
FRSKD+SLA	75.43±0.21	82.04±0.16	51.83±0.37	63.58±0.04

Table 1: Performance comparison on CIFAR-100 and TinyImageNet. Experiments are repeated three times, and we report average and standard deviation of the accuracy of the last epoch. The best performing model is indicated as boldface. The second-best model is indicated as underline.

the data augmentation based methods. Additionally, the effect of the feature distillation is demonstrated by the out-performance of FRSKD compared to FRSKD\F. Our proposed model, FRSKD, is not dependent on the data augmentation, so FRSKD is compatible with other data augmentation based self-knowledge distillation methods, such as SLA-SD. Hence, we conduct experiments by integrating FRSKD and SLA-SD (FRSKD+SLA), and FRSKD+SLA shows performance improvements with large margins on most experiments.

Methods	CUB200	MIT67	Dogs	Stanford40
Baseline	51.72±1.17	55.00±0.97	63.38±0.04	42.97±0.66
ONE	54.71±0.42	56.77±0.76	65.39±0.59	45.35±0.53
DDGSD	58.49±0.55	59.00±0.77	69.00±0.28	45.81±1.79
BYOT	58.66±0.51	58.41±0.71	68.82±0.15	48.51±1.02
SAD	52.76±0.57	54.48±1.30	63.17±0.56	43.52±0.06
CS-KD	64.34±0.08	57.36±0.37	68.91±0.40	47.23±0.22
SLA-SD	56.17±0.71	61.57±1.06	67.30±0.21	54.07±0.38
FRSKD\F	62.29±1.65	61.32±0.67	69.48±0.84	53.16±0.44
FRSKD	<u>65.39±0.13</u>	<u>61.74±0.67</u>	<u>70.77±0.20</u>	<u>56.00±1.19</u>
FRSKD+SLA	67.80±1.24	66.04±0.31	72.48±0.34	61.96±0.57

Table 2: Performance comparison on FGVR. Experiments are repeated three times, and we report average and standard deviation of the accuracy of the last epoch. The best performing model is indicated as boldface. The second-best model is indicated as underline.

Table 2 shows the classification accuracy on FGVR tasks. Similar to the result of Table 1, FRSKD shows better performance than other self-knowledge distillation methods. The superior performance of FRSKD against FRSKD\F indicates that the effect of feature distillation is greater when using a larger image. Also, FRSKD+SLA outperforms all of the other methods with large margins, so

the compatibility of FRSKD with data augmentation based self-knowledge distillation methods provides a significant advantage in FGVR tasks.

Model	Method	Top-1	Top-5
ResNet18	Baseline	69.76	89.08
	FRSKD	70.17	89.78
ResNet34	Baseline	73.31	91.42
	FRSKD	73.75	92.11

Table 3: Performance comparison on ImageNet. The best performing model is indicated as boldface.

To demonstrate FRSKD on large-scaled datasets, we evaluate FRSKD on ImageNet with two backbone network alternatives, ResNet18 and ResNet34. Table 3 shows that FRSKD improves the performance on ImageNet.

4.2. Semantic Segmentation

We conduct an experiment on semantic segmentation to verify the efficiency of FRSKD in various domains. We follow the most of the experimental settings from [8]. We use a combined dataset of VOC2007 and VOC2012 *trainval* as a training set; and we use the test set of VOC2007 as a validation set, which are widely used settings in semantic segmentation [4, 5]. This experiment utilizes an EfficientDet with stacked BiFPN structure [28] as a baseline. For our experiments, we stack three BiFPN layers, and we use additional two BiFPN layer as the self-teacher network. We set an initial learning rate as 0.01; we set the total epochs as 60; and we divide the learning rate by 10 at epoch 40. We describe further details in Appendix *Implementation Details*. Table 4 shows that FRSKD improves the performance of the semantic segmentation models by utilizing the self-knowledge distillation from the self-teacher network.

Model	Method	mIOU
EfficientDet-d0	Baseline	79.07
	FRSKD	80.55
EfficientDet-d1	Baseline	81.95
	FRSKD	83.88

Table 4: Performance comparison on semantic segmentation task. The best performing model is indicated as boldface.

4.3. Further Analyses on FRSKD

Qualitative Attention map comparison In order to check whether the classifier network is receiving a meaningful distillation from the self-teacher network, we conduct qualitative analyses by comparing the attention maps of each block from the classifier network and the self-teacher network.

Method	CIFAR-100	TinyImageNet	CUB200	MIT67	Dogs	Stanford40
Baseline	73.80 \pm 0.60	54.60 \pm 0.33	51.72 \pm 1.17	55.00 \pm 0.97	63.38 \pm 0.04	42.97 \pm 0.66
Fit+SKD	77.03 \pm 0.05	59.06 \pm 0.20	<u>61.05\pm1.05</u>	<u>57.69\pm0.28</u>	<u>67.50\pm0.32</u>	<u>51.66\pm1.32</u>
OD+SKD	<u>77.12\pm0.09</u>	<u>59.14\pm0.20</u>	57.44 \pm 0.92	54.83 \pm 2.63	66.51 \pm 0.87	49.09 \pm 0.47
FRSKD	77.71\pm0.14	59.61\pm0.31	65.39\pm0.13	61.74\pm0.67	70.77\pm0.20	56.00\pm1.19

Table 5: Performance comparison according to feature distillation method of FRSKD. The feature distillation method of Fit+SKD is based on FitNet [26]; OD+SKD is based on overhaul distillation [8]; and FRSKD is based on attention transfer [36]. ResNet18 is used as classifier network. The best performing model is indicated as boldface. The second-best model is indicated as underline.

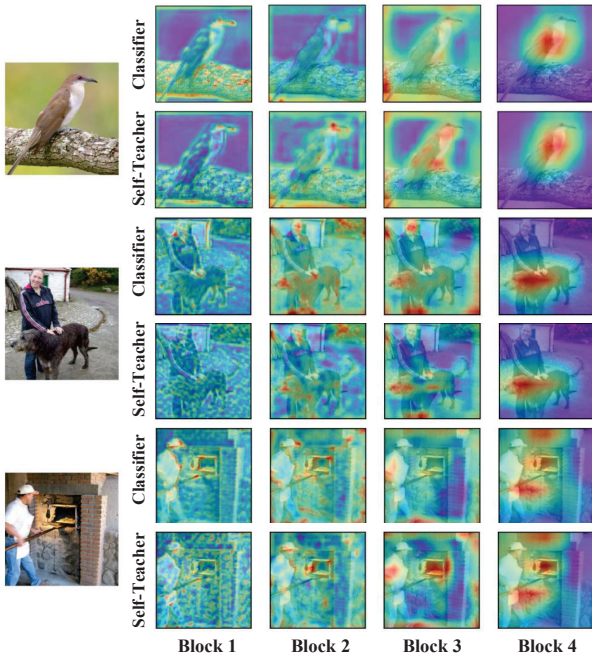


Figure 3: The block-wise attention map comparison between classifier network (first row from each data) and self-teacher network (second row from each data). From above, each data was taken from CUB200, Dogs and MIT67.

We obtain the attention map by applying a channel-wise pooling to the feature map for each block from the various datasets: CUB200, MIT67 and the Dogs dataset. We select the attention maps at the 50-th epoch to observe the distillation behaviors in the learning process.

Figure 3 shows the differences in the block-wise attention maps for the classifier network and the self-teacher network. For the data from CUB200 dataset, which is designed to distinguish the species of bird, the cases of Block 2 and 3 illustrate that the classifier network does not capture the proper attention on the main object (bird). In contrast, the cases of Block 2 and 3 in a self-teacher network show coherent attention maps on the target object by utiliz-

Type	#channel	Parameters	FLOPs	CIFAR-100
BiFPN	128	$\times 0.30$	$\times 0.67$	72.64 \pm 0.12
BiFPN	256	$\times 0.97$	$\times 2.38$	73.54 \pm 0.41
BiFPNc	128	$\times 0.19$	$\times 0.21$	71.70 \pm 0.19
BiFPNc	256	$\times 0.59$	$\times 0.68$	73.27 \pm 0.45

Table 6: Performance and efficiency comparison between the self-teacher network structures. WRN-16-2 is used as classifier network on CIFAR-100. BiFPN is a structure with the same channel dimension for each layer, and BiFPNc is a structure in which the channel dimension is different depending on the depth of the layer as proposed in Section 3.1. #channel is the channel dimension of the deepest layer, i.e. L_n, P_n, T_n , of the self-teacher network. #channel of BiFPNc depends on the channel width parameter, w . Parameters and FLOPs are the ratio of those of the classifier network.

ing the aggregated features. This trend can also be found in the data from the Dogs dataset. The self-teacher’s attention map points to the main object (dog) relatively with a concentration compared to the attention map of the classifier that is biased toward the human, which is not the main object. The attention map comparison is also conducted on MIT67 dataset, which performs indoor scene recognition by reflecting the overall context, not the task of concentrating on a single object. In order to successfully recognize the scene class (bakery) of the data, it is important to utilize the contextual cues inside the data. From the case of block 3, unlike the classifier network, the self-teacher network pays more attention to the bread, which can be an important clue to the scene class (bakery).

Ablation with the feature distillation methods FRSKD is able to integrate diverse feature distillation methods, so we experiment such variations of the integrated feature distillations. To analyze the performance differences, we compare 1) exact feature distillation methods, FitNet and Overhaul distillation; and 2) attention transfer methods used in FRSKD. Table 5 shows that the attention transfer of FRSKD achieves better accuracy in various datasets than the accu-

Method	CIFAR-100	TinyImageNet	CUB200	MIT67	Dogs	Stanford40
Baseline	73.80 \pm 0.60	54.60 \pm 0.33	51.72 \pm 1.17	55.00 \pm 0.97	63.38 \pm 0.04	42.97 \pm 0.66
FitNet	76.65 \pm 0.25	59.38 \pm 0.10	58.97 \pm 0.07	59.15 \pm 0.41	67.18 \pm 0.10	46.64 \pm 0.24
ATT	77.16 \pm 0.15	59.83\pm0.28	59.21 \pm 0.34	59.33 \pm 0.22	67.54 \pm 0.18	47.04 \pm 0.17
Overhaul	74.59 \pm 0.32	59.50 \pm 0.09	58.82 \pm 0.12	58.81 \pm 0.58	66.43 \pm 0.08	47.06 \pm 0.26
FRSKD	77.71\pm0.14	59.61 \pm 0.31	65.39\pm0.13	61.74\pm0.67	70.77\pm0.20	56.00\pm1.19

Table 7: Performance comparison on knowledge distillation. ResNet18 is used as classifier network. The best performing model is indicated as boldface. The second-best model is indicated as underline.

Method	CIFAR-100	TinyImageNet	CUB200	MIT67	Dogs	Stanford40
Baseline	73.80 \pm 0.60	54.60 \pm 0.33	51.72 \pm 1.17	55.00 \pm 0.97	66.38 \pm 0.04	42.97 \pm 0.66
Mixup	76.26 \pm 0.41	56.28 \pm 0.24	57.60 \pm 0.42	56.77 \pm 1.45	65.96 \pm 0.03	47.15 \pm 0.60
FRSKD + Mixup	78.74 \pm 0.19	60.30 \pm 0.38	67.98\pm0.58	62.11 \pm 0.81	71.64 \pm 0.29	56.50\pm0.36
CutMix	79.23 \pm 0.23	58.97 \pm 0.29	51.54 \pm 1.12	60.87 \pm 0.30	67.71 \pm 0.14	46.90 \pm 0.29
FRSKD + CutMix	80.49\pm0.05	61.92\pm0.11	65.92 \pm 0.59	66.19\pm0.49	72.81\pm0.23	55.75 \pm 0.43

Table 8: Performance of data augmentation method with FRSKD. ResNet18 is used as classifier network. The best performing model is indicated as boldface. The second-best model is indicated as underline.

racy from the integration of exact feature distillations.

Structure of self-teacher network In order to show the proposed self-teacher network efficiency, we experiment various self-teacher network structures, and Table 6 shows the experimented variations. BiFPN with high channel dimension (256) achieves the best performance, but the parameter and FLOPs of BiFPN are even larger or similar to the classifier network. In terms of efficiency, while BiFPN is an inadequate choice for the self-teacher network because of its large parameter size, BiFPNc with high channel dimension (256) shows a compatible performance to BiFPN with much less computations. Since the increased computation of BiFPNc is smaller than those of the classifier network, FRSKD is more efficient than the data augmentation based self-knowledge distillation methods, which use the classifier network in duplicates.

Compare to knowledge distillation Knowledge distillation with a teacher network easily utilizes the refined feature-map and its soft label. Therefore, we compare the existing knowledge distillation and FRSKD, which play similar roles. Under the assumption that we have a pre-trained teacher network, we compare exact feature distillation methods, FitNet and overhaul distillation [26, 8]; and attention transfer methods [36]. For knowledge distillation, we set the teacher network as the pretrained ResNet34 on each dataset, and we set the student network as an untrained ResNet18. For fair comparisons, each knowledge distillation method exploits the feature distillation as well as the soft label distillation. FRSKD utilizes ResNet18 as a classifier network to meet the identical conditions. Table 7 shows that FRSKD outperforms the experimented knowledge dis-

tillation method with a pretrained teacher network on most datasets.

Training with data augmentation The method of data augmentation is compatible with FRSKD. To verify this compatibility of FRSKD, we experiment our proposed method with recent data augmentation methods. Mixup utilizes a convex combination between two images and their labels [38]. Cutmix mixes a pair of images and labels by cutting an image into patches and pasting a patch on the other image. It is known that such data augmentations improve the accuracy on the most datasets. Table 8 shows that FRSKD has a large performance improvement when being used with data augmentations.

5. Conclusion

This paper presents a specialized neural network structure for self-knowledge distillation with top-down and bottom-up paths. The addition of these paths are expected to provide refined feature-maps and their soft labels to the classifier network. Moreover, the change of channel dimensions is applied to reduce the parameters while maintaining the feature-map refinement. Finally, FRSKD is able to apply self-knowledge distillation to the vision tasks of classification and semantic segmentation. We confirmed the large performance improvements quantitatively, and we show the efficiency of working mechanisms with various ablation studies.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A2C2009816).

References

- [1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1
- [2] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. *arXiv preprint arXiv:2002.01775*, 2020. 2
- [3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>. 6
- [5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>. 6
- [6] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616, 2018. 2
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5
- [8] Byeongho Heo, Jeessoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1921–1930, 2019. 2, 4, 6, 7, 8
- [9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2
- [10] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1013–1021, 2019. 5
- [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. *arXiv preprint arXiv:1704.04861*, 2017. 1, 4
- [12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1
- [13] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011. 5
- [14] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in neural information processing systems*, pages 2760–2769, 2018. 1, 2
- [15] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5936–5944, 2017. 3
- [16] Animesh Koratana, Daniel Kang, Peter Bailis, and Matei Zaharia. Lit: Learned intermediate representation training for model compression. In *International Conference on Machine Learning*, pages 3509–3518, 2019. 2
- [17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5
- [18] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *International Conference on Machine Learning*, 2020. 1, 2, 3, 5
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3
- [20] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 3
- [21] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019. 1, 2
- [22] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 5
- [23] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 1, 2
- [24] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5007–5016, 2019. 2
- [25] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009. 5
- [26] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 2, 4, 7, 8
- [27] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 1

- [28] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 3, 4, 6
- [29] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 1, 2
- [30] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019. 2
- [31] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5
- [32] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5565–5572, 2019. 1, 2, 3, 5
- [33] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *2011 International Conference on Computer Vision*, pages 1331–1338. IEEE, 2011. 5
- [34] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 4
- [35] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13876–13885, 2020. 1, 2, 3, 5
- [36] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 1, 2, 4, 7, 8
- [37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5
- [38] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 8
- [39] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3713–3722, 2019. 1, 2, 5
- [40] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018. 3
- [41] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 1
- [42] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 2
- [43] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *Advances in neural information processing systems*, pages 7517–7527, 2018. 2, 5