A Comprehensive Overhaul of Feature Distillation

Byeongho Heo[1,3]* Jeesoo Kim[2]* Sangdoo Yun[1] Hyojin Park[2]* Nojun Kwak[2] Jin Young Choi[3]

{bhheo, kimjiss0305, wolfrun, nojunk, jychoi}@snu.ac.kr, sangdoo.yun@navercorp.com

[1]Clova AI Research, NAVER Corp, Korea
[2]GSCST, Seoul National University, Korea
[3]Department of ECE, ASRI, Seoul National University, Korea

Abstract

We investigate the design aspects of feature distillation methods achieving network compression and propose a novel feature distillation method in which the distillation loss is designed to make a synergy among various aspects: teacher transform, student transform, distillation feature position and distance function. Our proposed distillation loss includes a feature transform with a newly designed margin ReLU, a new distillation feature position, and a partial L_2 distance function to skip redundant information giving adverse effects to the compression of student. In ImageNet, our proposed method achieves 21.65% of top-1 error with ResNet50, which outperforms the performance of the teacher network, ResNet152. Our proposed method is evaluated on various tasks such as image classification, object detection and semantic segmentation and achieves a significant performance improvement in all tasks. The code is available at bhheo.github.io/overhaul

1. Introduction

Experiencing remarkable advances in many machine learning tasks using neural networks, researchers have started to work on network compression and enhancement. Several approaches such as model pruning, model quantization and knowledge distillation have been suggested to make the model smaller and cost-efficient. Among them, *knowledge distillation* is being actively investigated. Knowledge distillation refers to the method that helps the training process of a smaller network (student) under the supervision of a larger network (teacher). Unlike other compression methods, it can downsize a network regardless of the structural difference between the teacher and the student network. Allowing architectural flexibility, knowledge distillation is emerging as a next generation approach of net-

---
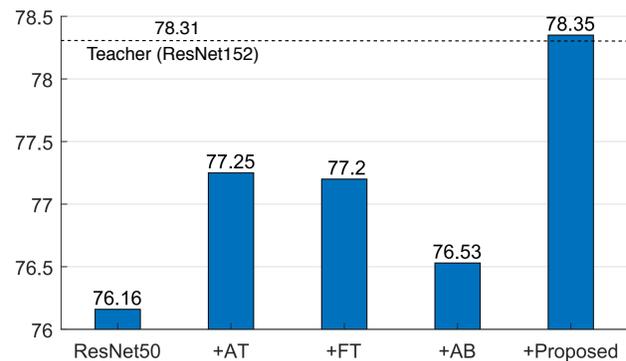*This work was done when authors were in research internship at *Clova AI Research*, NAVER corp.

Figure 1. Performance of distillation methods: AT [30], FT [13], AB [7] and proposed method on ImageNet. The graph shows accuracy(%) of ResNet50 trained with each distillation method. Note that ResNet152 with 78.31% accuracy is used as a teacher.

work compression.

Hinton *et al.* [8] proposed a knowledge distillation (KD) method using the softmax output of the teacher network. This method can be applied to any pair of network architectures since the dimensions of both outputs are the same. However, the output of a high-performance teacher network is not significantly different from the ground truth. Thus, transferring only the output is similar to training the student with the ground truth, making the performance of output distillation limited. To make better use of the information contained in the teacher network, several approaches have been proposed for feature distillation instead of output distillation. FitNets [22] have proposed a method that encourages a student network to mimic the hidden feature values of a teacher network. Although feature distillation was a promising approach, the performance improvement by FitNets was not significant.

After FitNets, variant methods of feature distillation have been proposed as follows. The methods in [30, 28] transform the feature into a representation having a reduced dimension and transfer it to the student. In spite of the reduced dimension, it has been reported that the ab-
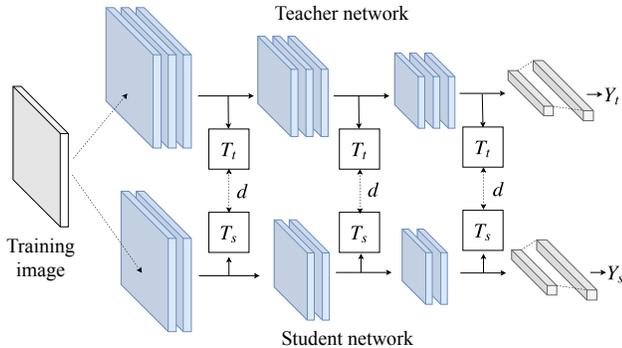
Figure 2. The general training scheme of feature distillation. The form of teacher transform T_t, student transform T_s and distance d differ from method to method.

stracted feature representation does lead to an improved performance. Recent methods (FT [13], AB [7]) have been proposed to increase the amount of transferred information in distillation. FT [13] encodes the feature into a 'factor' using an auto-encoder to alleviate the leakage of information. AB [7] focuses on activation of a network with only the sign of features being transferred. Both methods show a better distillation performance by increasing the amount of transferred information. However, FT [13] and AB [7] deform feature values of the teacher, which leaves a further room for the performance to be improved.

In this paper, we further improve the performance of feature distillation by proposing a new feature distillation loss which is designed via investigation of various design aspects: teacher transform, student transform, distillation feature position and distance function. Our method aims to transfer two factors from features. The first target is the magnitude of feature response after ReLU, since it carries most of the feature information. The second is the activation status of each neuron. Recent studies [20, 7] have shown that the activation of neurons strongly represents the expressiveness of a network, and it should be considered in distillation. To this purpose, we propose a margin ReLU function, change the distillation feature position to the front of ReLU, and use a partial L_2 distance function to skip the distillation of unnecessary information. The proposed loss significantly improves performance of feature distillation. In our experiments, we have evaluated our proposed method in various domains including classification (CIFAR [15], ImageNet [23]), object detection (PASCAL VOC [2]) and semantic segmentation (PASCAL VOC). As shown in Fig. 1, in our experiments, the proposed method shows a performance superior to the existing state-of-the-art methods and even the teacher model.

2. Motivation

In this section, we investigate the design aspects of feature distillation methods achieving network compression and present novel aspects of our approach distinctive to the preceding methods. First, we describe a general form of loss function in feature distillation. As shown in Fig. 2, the feature of the teacher network is denoted as $\boldsymbol{F}_t$ and the feature of the student network is $\boldsymbol{F}_s$. To match the feature dimension, T_t and T_s respectively, we transform the feature $\boldsymbol{F}_t$ and $\boldsymbol{F}_s$. A distance d between the transformed features is used as a loss function $L_{distill}$. In other words, the loss function of feature distillation is generalized as

$$L_{distill} = d(T_t(\boldsymbol{F}_t), T_s(\boldsymbol{F}_s)). \tag{1}$$

The student network is trained by minimizing the distillation loss $L_{distill}$.

It is desirable to design the distillation loss so as to transfer all feature information without missing any important information from the teacher. To achieve this, we aim to design a new feature distillation loss in which all important teacher's information is transferred as much as possible to improve the distillation performance. To get an idea for this purpose, we analyze the design aspects of feature distillation loss. As described in Table 1, the design aspects of feature distillation loss are categorized into 4 categories: teacher transform, student transform, distillation feature position and distance function.

Teacher transform. A teacher transform T_t converts the teacher's hidden features into an easy-to-transfer form. It is an important part of feature distillation and also a main cause of the information missing in distillation. AT [30], FSP [28] and Jacobian [26] reduce the dimension of the feature vector via the teacher transform which causes serious information missing. FT [13] uses a compression ratio determined by the user and AB [7] utilizes the original feature in the form of binarized values, making both methods to use features different from the original ones. Except Fit-Nets [22], most teacher transforms of existing approaches cause an information missing in the teacher's feature used in the distillation loss. Since features include both adverse and beneficial information, it is important to distinguish them and avoid missing the beneficial information. In the proposed method, we use a new ReLU activation, called margin ReLU, for the teacher transform. In our margin ReLU, the positive (beneficial) information is used without any transformation while the negative (adverse) information is suppressed. As a result, the proposed method can perform distillation without missing the beneficial information.

Student transform. Typically, the student transform T_s uses the same function as the teacher transform T_t. Therefore, in methods like AT [30], FSP [28], Jacobian [26] and FT [13], the same amount of information is lost in both the

Method	Teacher transform	Student transform	Distillation feature position	Distance	Missing information
FitNets [22]	None	1×1 conv	Mid layer	L_2	None
AT [30]	Attention	Attention	End of group	L_2	Channel dims
FSP [28]	Correlation	Correlation	End of group	L_2	Spatial dims
Jacobian [26]	Gradient	Gradient	End of group	L_2	Channel dims
FT [13]	Auto-encoder	Auto-encoder	End of group	L_1	Auto-encoded
AB [7]	Binarization	1×1 conv	Pre-ReLU	Marginal L_2	Feature values
Proposed	Margin ReLU	1×1 conv	Pre-ReLU	Partial L_2	Negative features

Table 1. Difference in various kinds of feature distillation. Most distillation use teacher transform with information loss.

teacher transform and the student transform. FitNets and AB do not reduce the dimension of teacher's feature and use a 1×1 convolutional layer as a student transform to match the feature dimension with the teacher. In this case, the feature size of the student does not decrease, but rather increases, so there is no information missing. In our method, we use this asymmetric format of transformations as the student transform.

Distillation feature position. Besides the types of feature transformation, we should be careful in picking the location in which distillation occurs. FitNets uses the end of an arbitrary middle layer as the distillation point, which has been shown to have a poor performance. We refer to a group of layers with the same spatial size as a layer group [29, 3]. In AT [30], FSP [28] and Jacobian [26], the distillation point lies at the end of each layer group, whereas in FT it lies at the end of only the last layer group. This has led to better results than FitNets but still lacks consideration about the ReLU-activated values of the teacher. ReLU allows the beneficial information (positive) to pass through and filters out the adverse information (negative). Therefore, knowledge distillation must be designed under the acknowledgement of this information dissolution. In our method, we design the distillation loss to bring the features in front of the ReLU function, called pre-ReLU. Positive and negative values are preserved in the pre-ReLU position without any deformation. So, it is suitable for distillation.

Distance function. Most distillation methods naively adopt L_2 or L_1 distance. However, in our method, we need to design an appropriate distance function according to our teacher transform, and our distillation point in the pre-ReLU position. In our design, the pre-ReLU information is transferred from teacher to student, but negative values of the pre-ReLU feature contain adverse information. The negative values of the pre-ReLU feature are blocked by the ReLU activation and not used by the teacher network. The transfer of all values may have an adverse effect to the student network. To handle this issue, we propose a new distance function, called partial L_2 distance, which is designed to skip the distillation of information on a negative region.
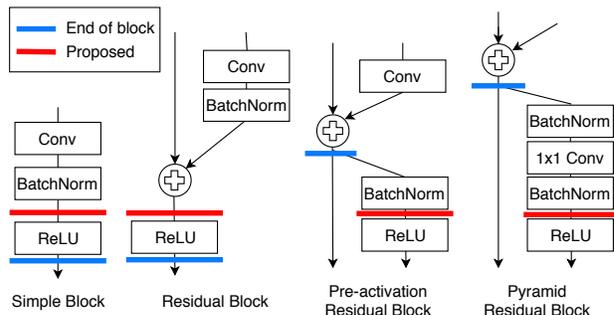

Figure 3. Position of distillation target layer. We place the distillation layer between the last block and the first ReLU. The exact location differs according to the network architecture.

3. Approach

In this section, we describe our distillation method outlined in section 2. We first describe the location where the distillation occurs in our method and then explain about the newly designed loss function.

3.1. Distillation position

The activation function is a crucial component of neural networks. The non-linearity of a neural network attributes to this function. The performance of the model is significantly influenced by the type of activation function. Among various activation functions, rectified linear unit (ReLU) [19] is used in most computer vision tasks. Most networks [16, 25, 27, 5, 6, 29, 3, 9, 24] use ReLU or modified versions very similar to ReLU [18, 14]. ReLU simply applies a linear mapping for positive values. For negative values, it eliminates the values and fixes them to zero, which prevents the unnecessary information from going backward. With a careful design of knowledge distillation considering ReLU, it is possible to transfer only the necessary information. Unfortunately, most of the preceding research don't take a serious consideration of the activation function. We define the minimum unit of the network, such as the residual block in ResNet [5] and the Conv-ReLU in VGG [25], as a layer block. The distillation in most methods occurs at
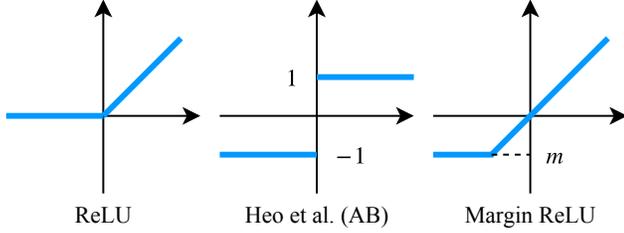
| ReLU | Heo et al. (AB) | Margin ReLU |

Figure 4. A comparison of the conventional ReLU, teacher transforms in Heo *et al.* [7] and our proposed method.

the end of the layer block ignoring whether it is related to ReLU or not.

In our proposed method, the position of the distillation lies between the first ReLU and the end of layer block. This positioning enables the student to reach the preserved information of the teacher before it passes through ReLU. Fig. 3 depicts the distillation position of some architectures. In case of the simple block [16, 25, 27, 9] and the residual block [5], the fact of whether the distillation happens before or after the ReLU constitutes the difference between our proposed method and other methods. However, for networks using the pre-activation [6, 29], the difference is larger. Since there is no ReLU at the end of each block, our method has to find the ReLU in the next block. In a structure like PyramidNet [3, 24], our proposed method can reach the ReLU after the 1×1 convolution layer. Though the positioning strategy may be complicated according to the architecture, it has a significant influence on the performance. Our new distillation position significantly improves the performance of the student as demonstrated in our experiments.

3.2. Loss function

Based on the format of section 2, we explain the teacher transform T_t, student transform T_s and the distance function d of our proposed method. Since the feature values of teacher $\boldsymbol{F}_t$ are the values before ReLU, positive values have the information utilized by the teacher while negative values do not. If a value in the teacher is positive, the student must produce the same value as in the teacher. On the contrary, if a value of the teacher is negative, the student should produce a value less than zero to make same activation status of neurons. Heo *et al.* [7] noted that a margin is required to make the student's value less than zero. Thus, we propose a teacher transform that preserves positive values while giving a negative margin.

$$\sigma_m(x) = \max(x, m). \tag{2}$$

Here, m is a margin value less than zero. We name this function as margin ReLU. Several types of teacher transforms are depicted in Fig. 4. Margin ReLU is designed to give a negative margin which is easier to follow than the negative

value of the teacher. Heo *et al.* set the margin by an arbitrary scalar value, which does not reflect the weight values of the teacher. In our proposed method, the margin value m is defined as the channel-by-channel expectation value of the negative response, and the margin ReLU uses values that correspond to each channel of the input. For a channel $\mathcal{C}$ and the i-th element of the teacher's feature F_t^i, the margin value of a channel $m_{\mathcal{C}}$ is set to an expectation value over all training images.

$$m_{\mathcal{C}} = E[F_t^i | F_t^i < 0, i \in \mathcal{C}]. \tag{3}$$

The expectation value can be calculated directly in the training process, or it can be calculated using parameters of the previous batch normalization layer. The margin ReLU $\sigma_{m_{\mathcal{C}}}(\cdot)$ is used as a teacher transform T_t in our proposed method and produces the target feature value for the student network. For the student transform, a regressor consisting of an 1×1 convolution layer [22, 7] and a batch normalization layer is used.

We now explain our distance function d. Our proposed method transfers the representation before ReLU. Therefore, the distance function should be changed considering ReLU. In the feature of the teacher, the positive responses are actually used for the network which implies that the positive responses of the teacher should be transferred by their exact values. However, negative responses are not. For a negative teacher response, if the student response is higher than the target value, it should be reduced, but if the student response is lower than the target value, it doesn't need to be increased since negatives are equally blocked by ReLU regardless of their values. For the feature representation of the teacher and student, $\boldsymbol{T}, \boldsymbol{S} \in \mathbb{R}^{W \times H \times C}$, let the i-th component of the tensor be $T_i, S_i \in \mathbb{R}$. Our partial L_2 distance (d_p) is defined as

$$d_p(\boldsymbol{T}, \boldsymbol{S}) = \sum_i^{WHC} \begin{cases} 0 & \text{if } S_i \leq T_i \leq 0 \\ (T_i - S_i)^2 & \text{otherwise.} \end{cases} \tag{4}$$

where $\boldsymbol{T}$ is the position for the teacher's feature and $\boldsymbol{S}$ is the position for the student's feature.

Our proposed method uses margin ReLU $\sigma_{m_{\mathcal{C}}}(x)$ as a teacher transform T_t and a regressor $r(\cdot)$ consisting of an 1×1 convolution layer as a student transform T_s, and uses partial L_2 distance (d_p) as the distance function. Distillation loss of the proposed method is:

$$L_{distill} = d_p(\sigma_{m_{\mathcal{C}}}(\boldsymbol{F}_t), r(\boldsymbol{F}_s)). \tag{5}$$

Our proposed method is conducted as *continuous distillation* using the distillation loss $L_{distill}$. Thus, the final loss function is the sum of distillation loss and task loss:

$$loss = L_{task} + \alpha L_{distill}. \tag{6}$$

The task loss refers to the loss specified by the task of a network. The feature position for distillation is after the last block of one spatial size and before ReLU as depicted in Fig. 3. In a network with a 32×32 input, such as CI-FAR [15], there are three target layers, and in the case of ImageNet [23], the number of target layers is four.

3.3. Batch normalization

We further investigate batch normalization in knowledge distillation. Batch normalization [11] is used in most recent network architectures to stabilize training. A recent study on batch normalization [10] explains the difference between training mode and evaluation mode of batch normalization. Each mode of batch-norm layer acts differently in the network. Therefore, when performing knowledge distillation, it is necessary to consider whether to use the teacher in training mode or evaluation mode. Typically, the feature of the student is normalized batch by batch. Therefore, the feature from the teacher must be normalized in the same way. In other words, the mode of the teacher's batch normalization layers should be training mode when distilling the information. To do this, we attach a batch normalization layer after the 1×1 convolutional layer and use it as a student transform and bring the knowledge from the teacher in training mode. As a result, our proposed method achieves additional performance improvements. This issue holds for all knowledge distillation methods including the proposed method. We empirically analyze various knowledge distillation methods for batch normalization issues in Section 4.5.3.

4. Experiments

We have evaluated the efficiency of our distillation method in several domains. The first task is the classification problem which is a fundamental problem in machine learning. As most of the other distillation methods have reported their performance under this domain, we also have compared our results to those of others. The performance of knowledge distillation depends on which network architecture is used, how well the teacher performs and what kind of training scheme is used. To control other factors and make a fair comparison, we reproduced the algorithms of other methods based on their codes and papers. All experiments were implemented and evaluated on NAVER Smart Machine Learning (NSML) [12] platform with PyTorch [21].

4.1. CIFAR-100

CIFAR-100 [15] is the dataset that most knowledge distillation methods use to validate their performance. Composed of 50,000 images with 100 classes, we use CIFAR-100 to compare various settings of all methods. To be practically used in any task, knowledge distillation must be applicable to any network structure. Therefore, we provide the experimental results of knowledge distillation using various structures of the teacher and student. Table. 2 shows the settings of each experiment such as the architecture used for each model, model size and compression rate. Majority of our experiments utilize Wide Residual Network [29] since the number of layers and the depth of each layer can be easily modified. Distillation between different types of architectures has also been experimented with the setting (d), (e), (f). In the case of (f), network names are similar, but the teacher is based on the bottleneck block and the student uses the basic block. Note that (e) and (f) use a teacher network PyramidNet-200 [3] trained with the Mixup augmentation [31]. All models have been trained for 200 epochs with a learning rate of 0.1, multiplied by 0.1 at epoch 100 and 150. To produce the best results from other methods, some algorithms [22, 13, 7] are trained with an output distillation loss [8] along with the feature distillation loss. The rest of the algorithms have shown better results when trained without the output distillation loss. The results of each method on every setting are presented in Table 3. Our proposed method outperforms the state-of-the-art in the settings of depth and channel compression (a), (b), (c) and different architecture (d), (e), (f). Especially, in the setting of depth compression (a), the student network trained by our proposed method outperforms the teacher network. The proposed method consistently shows a good performance regardless of the compression rate and even when distilling to different types of network architecture. Note that the error rate of 17.8% in (e) is better than any network reported in the paper of Wide Residual Network [29]. Therefore, our proposed method can be applied not only to small networks but also to large networks with high performance.

4.2. ImageNet

The image size of 32×32 in CIFAR is not enough to represent real world images. For this reason, we have conducted experiments on the ImageNet dataset [23] as well. ImageNet includes images with an average size of 469×387, which allows us to verify distillation performance in large images. In this paper, we have used the dataset in ILSVRC 2012 [23]. This dataset consists of 1.2 million training images and 50 thousand validation images. Images are cropped to the size of 224×224 for training and evaluation. The student network is trained for 100 epochs, and the learning rate begins at 0.1 multiplied by 0.1 at every 30th epoch. For a fair comparison and a simple reproduction, we used the pre-trained model in the PyTorch [21] library as the teacher network.

The experiments have been conducted on two pairs of networks. The first one is distillation from ResNet152 [5] to ResNet50, and the second one is distillation from ResNet50 to MobileNet [9]. In this section, we present the results of three latest algorithms [30, 13, 7], which have shown

Setup	Compression type	Teacher network	Student network	# of params teacher	# of params student	Compress ratio
(a)	Depth	WideResNet 28-4	WideResNet 16-4	5.87M	2.77M	47.2%
(b)	Channel	WideResNet 28-4	WideResNet 28-2	5.87M	1.47M	25.0%
(c)	Depth & channel	WideResNet 28-4	WideResNet 16-2	5.87M	0.70M	11.9%
(d)	Different architecture	WideResNet 28-4	ResNet 56	5.87M	0.86M	14.7%
(e)	Different architecture	PyramidNet-200 (240)	WideResNet 28-4	26.84M	5.87M	21.9%
(f)	Different architecture	PyramidNet-200 (240)	PyramidNet-110 (84)	26.84M	3.91M	14.6%

Table 2. Experiments settings with various network architectures on CIFAR-100. Network architecture is denoted as WideResNet (depth)-(channel multiplication) for Wide Residual Networks [29] and PyramidNet-(depth) (channel factor) for PyramidNet [3].

Setup	Teacher	Baseline	KD [8]	FitNets [22]	AT [30]	Jacobian [26]	FT [13]	AB [7]	Proposed
(a)	21.09	22.72	21.69	21.85	22.07	22.18	21.72	21.36	**20.89**
(b)	21.09	24.88	23.43	23.94	23.80	23.70	23.41	23.19	**21.98**
(c)	21.09	27.32	26.47	26.30	26.56	26.71	25.91	26.02	**24.08**
(d)	21.09	27.68	26.76	26.35	26.66	26.60	26.20	26.04	**24.44**
(e)	15.57	21.09	20.97	22.16	19.28	20.59	19.04	20.46	**17.80**
(f)	15.57	22.58	21.68	23.79	19.93	23.49	19.53	20.89	**18.89**

Table 3. Performance of various knowledge distillation methods on CIFAR-100. Measurement is error rate (%) of classification. lower is better. 'Baseline' represents a result without distillation.

the best results in the previous subsection. The results are represented in Table 4. Our proposed method shows a great improvement. In particular, our method has made ResNet50 perform better than ResNet152, which is a remarkable achievement. In addition, it has shown a considerable improvement in the recently proposed lightweight architecture, MobileNet. In case of MobileNet, it is hard to reproduce the performance of the paper (29.4) because the training scheme, such as training epochs, is not reported. Thus, we measured the performance in a standard setting.

4.3. Object detection

In this section, we apply our method to other computer vision tasks. The first one is object detection which is one of the most frequently used neural network techniques. Since the purpose of distillation is to improve speed, we applied our proposed method on a high-speed detector, Single Shot Detector (SSD) [17]. Networks are trained on a mixture of VOC2007 and VOC2012 [2] *trainval* set, which are widely used in object detection. The backbone network in all models is pre-trained using the ImageNet dataset. Networks have been trained for 120k iterations with a batch size of 32. To show the improvement of our method, we set the SSD trained with no distillation as the baseline, referred to as 'Baseline' in Table 5. SSD detector based on ResNet50 [5] or VGG [25] is used as the teacher network to examine the difference of performance according to the teacher architecture. As the student networks, SSD based on ResNet18 and SSD lite based on Mobilenet [9] have been used.

Network	# of param (ratio)	Method	Top-1 error(%)	Top-5 error(%)
ResNet152	60.19M	Teacher	21.69	5.95
ResNet50	25.56M (42.5%)	Baseline	23.72	6.97
		KD [8]	22.85	6.55
		AT [30]	22.75	6.35
		FT [13]	22.80	6.49
		AB [7]	23.47	6.94
		Proposed	**21.65**	**5.83**
ResNet50	25.56M	Teacher	23.84	7.14
MobileNet	4.23M (16.5%)	Baseline	31.13	11.24
		KD [8]	31.42	11.02
		AT [30]	30.44	10.67
		FT [13]	30.12	10.50
		AB [7]	31.11	11.29
		Proposed	**28.75**	**9.66**

Table 4. Results on ILSVRC 2012 validation set. Networks are trained and evaluated in 224×224 size with single-crop. 'Baseline' represents a result without distillation.

Detection performance has been evaluated in the VOC 2007 *test* set and all results are presented in Table 5. In the case of ResNet18, the performance improvement of ResNet18-T1 using ResNet teacher is larger than T2 using a VGG teacher. Though both student architectures outperform the baseline, the distillation between similar structure

Network	# of params	Method	mAP(%)
ResNet50-SSD	36.7M	Teacher (T1)	76.79
VGG-SSD	26.3M	Teacher (T2)	77.50
ResNet18-SSD	20.0M	Baseline	71.61
		Proposed-T1	**73.08**
		Proposed-T2	72.38
MobileNet -SSD lite	6.5M	Baseline	67.58
		Proposed-T1	**68.54**
		Proposed-T2	68.45

Table 5. Object detection results of SSD300 [17] in PASCAL VOC2007 *test* set [2]. Results are described in mean Average Precision (mAP). Higher is better.

Backbone	# of params	Method	mIoU
ResNet101	59.3M	Teacher	77.39
ResNet18	16.6M (28.0%)	Baseline	71.79
		Proposed	**73.24**
MobileNetV2	5.8M (9.8%)	Baseline	68.44
		Proposed	**71.36**

Table 6. Semantic segmentation based on DeepLabV3+ [1] on the PASCAL VOC 2012 *test* set [2]. Measurement of performance is mean Intersection over Union (mIoU).

shows a better quality of knowledge distillation. In the case of MobileNet, our proposed method shows a constant performance improvement regardless of the type of the teacher. Student models in all experiments have experienced improvements in performance and this implies that our method can be applied to any SSD-based object detector.

4.4. Semantic segmentation

In this section, we verify the performance of our proposed method on semantic segmentation. Applying distillation on semantic segmentation is challenging since the output size is much larger than any other task. We have selected the latest study, DeepLabV3+ [1] as our base model for semantic segmentation. DeepLabV3+ based on ResNet101 has been used as the teacher network, and DeepLabV3+ based on MobileNetV2 [24] and ResNet18 [5] has been used as the student network. Experiments have been performed on the PASCAL VOC 2012 segmentation [2] dataset. We also use an augmentation of the dataset provided by the extra annotations in [4] as in the baseline paper [1]. All models have been trained for 50 epochs, and the learning rate schedule is the same as the baseline paper [1]. In similar fashion to our detection task, the student network is initialized to a network pre-trained on ImageNet without distillation. Results are presented in Table 6. Our proposed method significantly improves the performance of

Method	KL divergence with teacher	Cross-entropy with GT	Error rate(%)
Baseline	0.7318	1.0741	27.32
KD [8]	0.7064	1.0758	26.47
FitNets [22]	0.7993	1.1585	26.30
AT [30]	0.7047	1.0303	26.56
Jacobian [26]	0.7122	1.0495	26.71
FT [13]	0.6872	1.0561	25.91
AB [7]	0.7555	1.1197	26.02
Proposed	**0.5723**	**0.9585**	**24.08**

Table 7. Output similarity analysis between teacher and student on test set of CIFAR-100.

ResNet18 and MobilenetV2. Taking MobileNetV2, in particular, our proposed method improves the performance by almost 3 points in mIoU and contributes to computation reduction of the segmentation algorithm. We have shown that our proposed method can be applied to image classification, object detection and semantic segmentation. Being able to be applied to many tasks without major changes is an advantage of feature distillation and shows that our proposed method has a wide range of applications.

4.5. Analysis

We analyze possible factors which would have lead to the performance improvement by our proposed method. The first analysis is the output similarity between the teacher and the student learned by distillation. By this, we verify how well our method forces the student to follow the teacher. After that, we provide an ablation study of our proposed method. We measure how much each component of our proposed method contributes to the performance. Finally, we discuss about how the mode of batch normalization affects knowledge distillation, as mentioned in Section 3.3. All experiments are based on setting (c) of Table 2.

4.5.1 Teacher-student similarity

KD [8] forces the output of the student to be similar to output of the teacher. The purpose of output distillation is quite intuitive, i.e., if a student produces an output similar to that of the teacher, its performance will also be similar. However, in the case of feature distillation, it is necessary to investigate how the output of the student changes. To see how well the student mimics the teacher, we measure the similarity of the teacher's and student's output under a consistent setting. On the test set of CIFAR-100, we measure the KL divergence between the teacher and student output. The cross-entropy with the ground truth has also been measured since classification performance also contributes to the reduction of KL divergence. Results are presented in Table 7. Methods that apply distillation only in the early stage of the

Mode of batch-norm	KD [8]	FitNets [22]	AT [30]	Jacobian [26]	FT [13]	AB [7]	Proposed
Training mode	26.47	26.61	26.56	26.71	25.91	26.02	24.08
Evaluation mode	26.45	26.92	26.42	26.75	26.15	26.36	24.54

Table 8. Analysis of the mode of batch normalization in teacher network on CIFAR-100. Table shows error rate(%). The first row shows the results of teacher's batch-norm in training mode while the second row shows the results of using the batch-norm in evaluation mode.

training, *Initial distillation* (FitNets [22], AB [7]), increase the KL divergence both with the teacher and ground-truth. With this result, it is hard to say that the student networks of these methods are mimicking their teacher networks. Meanwhile, distillation methods with *continuous distillation* in Eq. 6 (KD [8], AT [30], Jacobian [26], FT [13]) as well as our proposed method reduce the KL divergence, which implies that the similarity between the teacher and student is relatively high. Specifically, our method shows a considerably high similarity compared to other *continuous distillation* methods. In other words, our proposed method trains the student to produce an output most similar to that of the teacher. This similarity is one of the main reasons of improved performance of our proposed method.

4.5.2 Ablation study

Ablation experiments were conducted in which the ablation components were added one-by-one to measure their effects. The result is shown in Table 9. The baseline is a distillation method based on L_2 loss at end of block position. The version that uses the preReLU position (Section 3.1) provides the greatest improvement because it is helpful to transfer the activation boundary effectively with both negative and positive values before ReLU. The second improvement is achieved by the loss function (Section 3.2), which prevents the transfer of useless and harmful negative values of less than a small negative margin. The batch-norm mode (Section 3.3) also contributes to the performance improvement. In conclusion, a combination of all proposed components leads to a significant improvement in performance of the proposed method.

4.5.3 Batch normalization

In Section 3.3, we mentioned about the issue related to the mode of the batch normalization in knowledge distillation. To investigate this, we measure the performance variation of knowledge distillation methods when differing the mode of the teacher's batch norm layer. The experimental results are shown in Table 8. The distillation methods that use additional information other than feature (KD [8], Jacobian [26]) show marginal differences between each mode of batch normalization. AT [30], which uses a diminished feature for distillation, has shown a better result in the evaluation mode. However, methods that do not squeeze the fea-

	Baseline	+ Position (Sec.3.1)	+ BN (Sec.3.3)	+ loss (Sec.3.2)
Error	26.37	24.81	24.68	24.08
Diff	-	-1.56	-0.13	-0.60

Table 9. Ablation study of proposed method. The results are presented in the form of error rate (%).

ture (FitNets [22], FT [13], AB [7]) consistently work better in the training mode. Our method especially shows a substantial improvement when using the training mode. Note that all experiments in previous sections exploit the better mode of the batch-norm layer as there is no mention about it in each paper. In conclusion, an appropriate type of batch normalization should be carefully chosen in many distillation methods including ours.

5. Conclusion

We propose a new knowledge distillation method along with several investigations about various aspects of the existing feature distillation methods. We have discovered the effectiveness of pre-ReLU location and proposed a new loss function to improve the performance of feature distillation. The new loss function consists of a teacher transform (margin ReLU) and a new distance function (partial L_2) and enables an effective feature distillation at pre-ReLU location. We have also investigated about the mode of batch normalization in teacher network and achieved additional performance improvements. Through experiments, we examined the performance of the proposed method using various networks in various tasks, and proved that the proposed method substantially outperforms the state-of-the-arts of feature distillation.

Acknowledgement

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[2] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015.

[3] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[7] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] 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.

[10] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in Neural Information Processing Systems (NIPS)*. 2017.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[12] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018.

[13] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[14] Alex Krizhevsky and Geoffrey Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[18] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[19] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010.

[20] Xingyuan Pan and Vivek Srikumar. Expressiveness of rectifier networks. In *International Conference on Machine Learning (ICML)*, 2016.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017.

[22] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[24] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[26] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning (ICML)*, 2018.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016.

[30] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[31] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.