Received: 27 March 2018 | Revised: 1 August 2018 | Accepted: 5 September 2018

DOI: 10.4218/etrij.2018-0152

ORIGINAL ARTICLE

WILEY ETRI Journal

Layer-wise hint-based training for knowledge transfer in a teacher-student framework

Ji-Hoon Bae[1] | **Junho Yim**[2] | **Nae-Soo Kim**[1] | **Cheol-Sig Pyo**[1] | **Junmo Kim**[2]

[1]KSB Convergence Research Department, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea.

[2]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea.

Correspondence
Junmo Kim, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea.
Email: junmo.kim@kaist.ac.kr

Funding information
National Research Council of Science & Technology (NST); Korean government (MSIP), Rep. of Korea, Grant/Award Number: CRC-15-05-ETRI

We devise a layer-wise hint training method to improve the existing hint-based knowledge distillation (KD) training approach, which is employed for knowledge transfer in a teacher-student framework using a residual network (ResNet). To achieve this objective, the proposed method first iteratively trains the student ResNet and incrementally employs hint-based information extracted from the pretrained teacher ResNet containing several hint and guided layers. Next, typical softening factor-based KD training is performed using the previously estimated hint-based information. We compare the recognition accuracy of the proposed approach with that of KD training without hints, hint-based KD training, and ResNet-based layer-wise pretraining using reliable datasets, including CIFAR-10, CIFAR-100, and MNIST. When using the selected multiple hint-based information items and their layer-wise transfer in the proposed method, the trained student ResNet more accurately reflects the pretrained teacher ResNet's rich information than the baseline training methods, for all the benchmark datasets we consider in this study.

KEYWORDS

knowledge transfer, layer-wise hint training, residual networks, teacher-student framework

1 | INTRODUCTION

Recently, deep neural network (DNN) models based on convolutional neural networks (CNNs) [1], such as AlexNet [2], GoogleNet [3], VGGNet [4], and the residual network (ResNet) [5,6], have produced promising results, particularly in the field of computer vision. Applications using state-of-the-art DNN models continue to expand [7-19]. However, DNN models have a deep and wide neural network structure with a large number of learning parameters that must generally be optimized. Thus, the direct reuse of pretrained DNN models is limited in many applications, such as the Internet of Things environment [20]. Knowledge extracted from a complex pretrained network and its efficient transfer to other, relatively less complex networks is useful for improving the training ability of the simpler networks. Therefore, to extend the application of DNN models to improving classification accuracy, rapidly obtaining inference times, and reducing network sizes for limited-computing environments, efficient knowledge extraction, and knowledge transfer techniques are crucial.

To achieve these requirements, several studies on knowledge distillation (KD) and knowledge transfer in a teacher-student framework (TSF) have been conducted in recent years [21–25]. Li and others [21] proposed a knowledge transfer method using a network output distribution based on Kullback-Leibler (KL) divergence in speech recognition tasks. Based on model compression [26], the researchers trained a small student network by matching the class probabilities of a large pretrained teacher network. This approach was implemented by minimizing the KL divergence of the output distribution between the teacher and student networks. In relation to [21], Hinton and others [22] introduced KD terminology from the TSF. Unlike in [21], Hinton and others introduced relaxation by applying a softening factor to the signal, originating from the teacher network's output. This approach can provide more information to the student network during training. Therefore, the softened version of the final output of the teacher network is regarded as the teacher's KD information, which small student networks strive to learn. Romero and others [23] proposed a hint-based KD training method in a TSF called FitNet, which improved the earlier KD training performance by introducing hint-based training, in which a hint is defined as the output of a teacher network's hidden layer. This method enables the student network to learn additional information that corresponds to the teacher's parameters up to the hint layer, as well as existing KD information. The trained deep and narrow VGGNet-like student network can then provide better recognition accuracy with fewer parameters than the original wide and shallow maxout [24] teacher network, owing to this stage-wise training procedure. In addition, Net2Net [25] was proposed for the rapid transfer of knowledge from a small teacher network to a large student network. In [25], a function-preserving transform was applied to initialize the parameters of the student network based on the parameters of the teacher network.

This study aims to improve the recognition accuracy of hint-based KD training for effective knowledge transfer. To achieve this objective, we propose a layer-wise hint-training TSF that uses multiple hint and guided layers. First, multiple hint layers in the teacher network—and the same number of guided layers in the student network—are selected. Next, the student network is iteratively and incrementally trained from the lowest guided layer to the highest guided layer with the help of the teacher's hints from multiple selected hint layers. Finally, the student network learns further using multiple hints extracted from the previous step and existing KD information from the teacher's softened output [22]. To verify the effectiveness of the proposed training approach, we employ ResNet with the latest DNN model for all training methods, where the teacher ResNet is deeper than the student ResNet. Therefore, we focused on knowledge transfer to improve the performance of a small student network by extracting distilled knowledge from a deep teacher network. For our experimental analysis, we employed Caffe [27,28], which is a reliable deep-learning open framework.

Meanwhile, the proposed training approach can be regarded as a layer-wise CNN-based pretraining scheme [29], in terms of training the student network, because multiple hints extracted from the pretrained teacher network are propagated layer-by-layer into the student network. Therefore, we also compare the recognition accuracy of the proposed method with that of layer-wise pretraining using ResNet.

The remainder of this paper is organized as follows: In Section 2, we detail the proposed TSF using layer-wise hint training. In Section 3, we demonstrate the recognition accuracy of the proposed training approach through experimental results on several widely used benchmark datasets. In Sections 4 and 5, respectively, we present a discussion of our results and our conclusions.

2 | TRAINING IN A TEACHER-STUDENT FRAMEWORK

2.1 | Original training algorithm for knowledge transfer

In this section, we employ an existing hint-based KD training method [23] to introduce the proposed training approach using multiple hint and guided layers, specifically when ResNet models with the same spatial dimensions are used in a TSF. The traditional knowledge transfer scheme is composed of two stages: hint training and KD training. First, hint training is achieved by minimizing the following l_2 loss function [23]:

$$(\hat{\mathbf{W}}_G) = \arg\min_{\mathbf{W}_G} \frac{1}{2} \|\mathbf{F}_H^{\text{mid}}(x; \mathbf{W}_H) - \mathbf{F}_G^{\text{mid}}(x; \mathbf{W}_G)\|^2, \quad (1)$$

where $\mathbf{W}_H$ are the weights of a teacher ResNet up to the selected hint layer, $\mathbf{W}_G$ are the weights of a student ResNet up to the selected guided layer, and $\mathbf{F}_H^{\text{mid}}$ and $\mathbf{F}_G^{\text{mid}}$ represent N_l feature maps ($\in \mathscr{R}^{N_h \times N_w}$) generated from their respective hint and guided layers with $\mathbf{W}_H$ and $\mathbf{W}_G$. Here, N_h and N_w are the height and width of the feature map. Note that each hint and guided layer is selected as the middle layer of the teacher and student ResNets, respectively.

After hint training, the extracted $\hat{\mathbf{W}}_G$ is used to construct the initial weights of the student ResNet, $\mathbf{W}_S = \left[\hat{\mathbf{W}}_G; \mathbf{W}_{S_r}\right]$, where $\mathbf{W}_{S_r}$ denotes the remaining weights of the student ResNet, which are randomly initialized from the guided layer to the output layer.

Second, after initially loading all weights $\mathbf{W}_S$ of the student ResNet, KD training using the softening factor (τ) is implemented by minimizing the weighted sum of the two cross entropies [22,23]:

$$(\hat{\mathbf{W}}_S) = \arg\min_{\mathbf{W}_S} \left\{ \text{CE}(y_{\text{true}}, P_S)|_{\mathbf{W}_S} + \lambda \text{CE}(P_T, P_S)|_{\mathbf{W}_S} \right\}, \quad (2)$$

where CE(·) denotes cross entropy, λ indicates a control parameter that adjusts the weight between the two CEs,

$P_T = \text{softmax}(p_t/\tau)$, $P_S = \text{softmax}(p_s/\tau)$, and p_t and p_s are the pre-softmax outputs of the teacher and student ResNets, respectively. Based on the recommended range of 2.5 to 4 for τ [22,23], we set $\tau = 3$ for all experiments.

2.2 | Proposed training algorithm for knowledge transfer

In this section, we introduce a layer-wise hint training method based on the existing hint-based learning approach to enhance the knowledge transfer capability in the TSF. The goal of the proposed approach is to perform layer-wise training among multiple hint and guided layers, unlike the original method, which uses only the intermediate hint and guided layers. In other words, knowledge transfer across multiple hint and guided layers is achieved using repeated incremental bottom-up training between the teacher and student networks.

Based on (1), the proposed hint training procedure using N hint/guided layers (layers H_i–G_i, $i = 1, 2, \ldots, N$) is detailed as follows (Stage 1):

Step 1: Estimate weights $\hat{\mathbf{W}}_{G_1}$ from the first hint/guided layers (H_1–G_1) by solving the optimization problem in (3).

$$\left(\hat{\mathbf{W}}_{G_1}\right) = \underset{\mathbf{W}_{G_1}}{\arg\min} \frac{1}{2} \|\mathbf{F}_H^1(x; \mathbf{W}_{H_1}) - \mathbf{F}_G^1(x; \mathbf{W}_{G_1})\|_2^2, \quad (3)$$

where $\mathbf{W}_{H_1}$ are the teacher ResNet's weights up to layer H_1, $\mathbf{W}_{G_1}$ are the student ResNet's weights up to layer G_1, and the initial weights $\mathbf{W}_{G_1} = \mathbf{W}_{S_1}$. $\mathbf{W}_{S_1}$ comprise randomly initialized weights from the input layer to layer G_1.

Step 2: Estimate weights $\hat{\mathbf{W}}_{G_2}$ from the second hint/guided layers (H_2–G_2) using the previously estimated weights $\hat{\mathbf{W}}_{G_1}$ ($\hat{\mathbf{W}}_{G_1} \subset \mathbf{W}_{G_2}$), as follows:

$$\left(\hat{\mathbf{W}}_{G_2}\right) = \underset{\mathbf{W}_{G_2}}{\arg\min} \frac{1}{2} \|\mathbf{F}_H^2(x; \mathbf{W}_{H_2}) - \mathbf{F}_G^2(x; \mathbf{W}_{G_2})\|_2^2, \quad (4)$$

where $\mathbf{W}_{H_2}$ are the teacher ResNet's weights up to layer H_2, $\mathbf{W}_{G_2}$ are the student ResNet's weights up to layer G_2, and the initial weights $\mathbf{W}_{G_2} = \left[\hat{\mathbf{W}}_{G_1}; \mathbf{W}_{S_2}\right]$. $\mathbf{W}_{S_2}$ denotes randomly initialized weights between layers G_1 and G_2.

Step i: Estimate weights $\hat{\mathbf{W}}_{G_i}$ up to the ith guided layer with (5) from the ith hint/guided layers (H$_i$–G_i).

$$\left(\hat{\mathbf{W}}_{G_i}\right) = \underset{\mathbf{W}_{G_i}}{\arg\min} \frac{1}{2} \|\mathbf{F}_H^i(x; \mathbf{W}_{H_i}) - \mathbf{F}_G^i(x; \mathbf{W}_{G_i})\|_2^2, \quad (5)$$

where $\mathbf{W}_{H_i}$ are the teacher ResNet's weights up to the selected layer H$_i$, $\mathbf{W}_{G_i}$ are the student ResNet's weights up to the selected layer G_i, $\mathbf{F}_H^i$ denotes the ith feature maps generated from the ith hint layer using weights $\mathbf{W}_{H_i}$, $\mathbf{F}_G^i$ denotes the ith feature maps generated from the ith guided layer using weights $\mathbf{W}_{G_i}$, and $\hat{\mathbf{W}}_{G_i}$ are the ith estimated weights using the previously identified ($i - 1$)th weights $\hat{\mathbf{W}}_{G_{i-1}}$, as

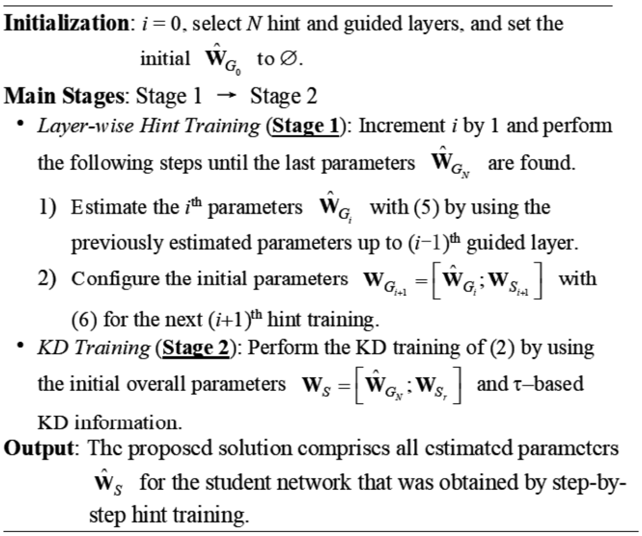
$$\mathbf{W}_{G_i} = \left[\hat{\mathbf{W}}_{G_{i-1}}; \mathbf{W}_{S_i}\right], \hat{\mathbf{W}}_{G_{i-1}} \subset \mathbf{W}_{G_i}, \quad (6)$$

where $\mathbf{W}_{S_i}$ represents randomly initialized weights from the ($i - 1$)th guided layer to the ith guided layer. The previous steps are then repeated until the last weights $\hat{\mathbf{W}}_{G_N}$, up to the Nth guided layer (G_N), are found. As per this procedure, each hint training is performed incrementally from the bottom to the top by minimizing the corresponding l_2 loss function. Through iterative and layer-wise hint training, the teacher network's rich information can be delivered more precisely to the student network than the original training approach of simply considering the teacher network's intermediate result.

Next, we implemented a softening factor—the τ-based KD training from (2) (Stage 2 in the proposed method)—using all initial weights $\mathbf{W}_S = \left[\hat{\mathbf{W}}_{G_N}; \mathbf{W}_{S_r}\right]$, where $\hat{\mathbf{W}}_{G_N}$ consists of weights obtained from the proposed layer-wise hint training procedure, and $\mathbf{W}_{S_r}$ comprises randomly initialized weights from the Nth guided layer to the output layer. We set $\tau = 3$ for all experiments. Figure 1 presents a description of the proposed approach to using multiple hints for knowledge transfer in the TSF.

3 | EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed method for knowledge transfer in the TSF. For several benchmark datasets, we compare the recognition accuracies of the proposed method and that of existing TSF-based training methods. All experiments used a



Initialization: $i = 0$, select N hint and guided layers, and set the initial $\hat{\mathbf{W}}_{G_0}$ to ∅.

Main Stages: Stage 1 → Stage 2

- *Layer-wise Hint Training* (**Stage 1**): Increment i by 1 and perform the following steps until the last parameters $\hat{\mathbf{W}}_{G_N}$ are found.
    1) Estimate the i^{th} parameters $\hat{\mathbf{W}}_{G_i}$ with (5) by using the previously estimated parameters up to $(i-1)^{\text{th}}$ guided layer.
    2) Configure the initial parameters $\mathbf{W}_{G_{i+1}} = \left[\hat{\mathbf{W}}_{G_i}; \mathbf{W}_{S_{i+1}}\right]$ with (6) for the next $(i+1)^{\text{th}}$ hint training.
- *KD Training* (**Stage 2**): Perform the KD training of (2) by using the initial overall parameters $\mathbf{W}_S = \left[\hat{\mathbf{W}}_{G_N}; \mathbf{W}_{S_r}\right]$ and τ–based KD information.

Output: The proposed solution comprises all estimated parameters $\hat{\mathbf{W}}_S$ for the student network that was obtained by step-by-step hint training.

FIGURE 1 Description of the proposed iterative layer-wise hint training method in a TSF

ResNet model with a total of $6n + 2$ stacked weighted layers ($n = 1, 2$, etc.) as the base architecture [5] (Figure 2).

Note that the ResNet structure is realized using feedforward neural networks with shortcut connections (used to make an ensemble structure that enables training overly deep networks by enhancing information propagation) and batch normalization (BN) [30].

The ResNet considered in this study has three sections in which the feature map dimensions and number of filters are changed. For example, as shown in Figure 2, the first stack has $2n$ residual modules with sixteen 32×32 feature maps per layer for 32×32 input images; the second stack has $2n$ residual modules with thirty-two 16×16 feature maps per layer; and the third stack has $2n$ residual modules with sixty-four 8×8 feature maps per layer. For all subsequent experiments, the original ResNet (without teacher knowledge) was implemented with a training procedure [5] using Softmax cross entropy loss for true labels. As in [5], we also used a weight decay of 0.0001 and a momentum of 0.9 with MSRA weight initialization (introduced in [31]).

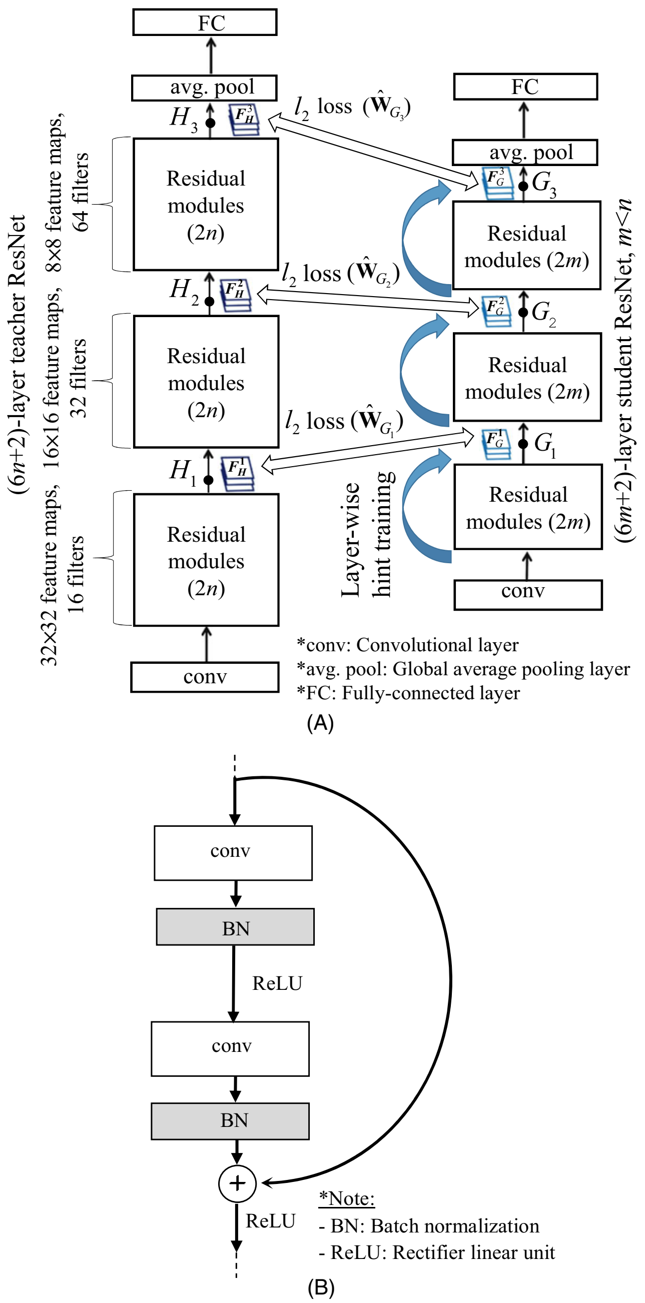

FC
avg. pool
H_3
F_H^3
l_2 loss ($\hat{\mathbf{W}}_{G_3}$)
FC
avg. pool
F_G^3
G_3
Residual modules (2n)
Residual modules (2m)
H_2
F_H^2
l_2 loss ($\hat{\mathbf{W}}_{G_2}$)
F_G^2
G_2
Residual modules (2n)
Residual modules (2m)
l_2 loss ($\hat{\mathbf{W}}_{G_1}$)
F_G^1
G_1
H_1
F_H^1
Residual modules (2n)
Residual modules (2m)
Layer-wise hint training
conv
conv
(6n+2)-layer teacher ResNet
32×32 feature maps, 16 filters
16×16 feature maps, 32 filters
8×8 feature maps, 64 filters
(6m+2)-layer student ResNet, m<n
*conv: Convolutional layer
*avg. pool: Global average pooling layer
*FC: Fully-connected layer
(A)
conv
BN
ReLU
conv
BN
+
ReLU
*Note:
- BN: Batch normalization
- ReLU: Rectifier linear unit
(B)

FIGURE 2 TSF using a state-of-the-art ResNet model to implement the proposed method: (A) overall architecture and (B) residual module in ResNet

For the proposed method, although there were no constraints on selecting multiple hint/guided layers, we selected three pairs of hint/guided layers, whose feature map dimensions changed (ie, $N = 3$; $\{(H_1, G_1), (H_2, G_2), (H_3, G_3)\}$ in Figure 2) to maintain the consistency of the criteria for selecting multiple hint/guided layers for two ResNet structures with different layers.

3.1 | Proposed hint training using CIFAR-10/100

We first experimentally evaluated the proposed training approach using CIFAR-10 [32], a widely used reliable benchmark image dataset composed of 50,000 32×32 color training images and 10,000 test images belonging to ten classes (Figure 3). For all experiments, we applied the data preprocessing technique presented in [5] to the training dataset using a mini-batch size of 128. Four pixels were padded on each side to create a 40×40 pixel image. Randomly cropped 32×32 pixel images were used for training, whereas the original 32×32 pixel images were used for testing.

For the existing hint-based KD training method (Section 2.1), we first trained Stage 1 by minimizing (1) using a learning rate of 1e-4. We stopped the training when there was no improvement in hint training loss after 25,000 iterations; therefore, hint-based training in Stage 1 was implemented for 25,000 iterations, where the hint and guided layers were set to the middle layer of each teacher and student ResNet, respectively. Next, KD training was implemented over 64,000 iterations in Stage 2. According to [5], which started at 0.1, the learning rate changed from 0.01 to 0.001 at 32,000 and 48,000 iterations, respectively, and terminated at 64,000 iterations. For the tunable parameter λ for KD training in (2), simulation results revealed that $\lambda = 5$ provided better accuracy than other values, ranging from 3 to 7. Therefore, in this experiment, we set $\lambda = 5$ for KD training in Stage 2.

In the proposed method, Stage 1 was trained incrementally using a learning rate of 1e-4 over the same 25,000 iterations. First, $\hat{\mathbf{W}}_{G_1}$ was estimated for 3,000 iterations. Then, $\hat{\mathbf{W}}_{G_2}$ was extracted for 7,000 iterations. Finally, $\hat{\mathbf{W}}_{G_3}$

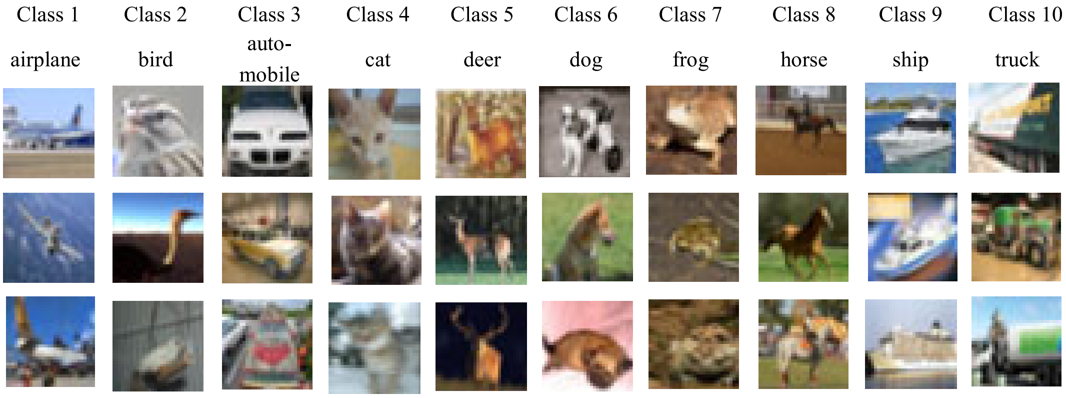


FIGURE 3 CIFAR-10 dataset [32] showing three random 32 × 32 images from each class

was obtained over 15,000 iterations. The remaining KD training (Stage 2) in the proposed method was performed in the same manner as existing hint-KD training methods.

Figure 4 represents recognition accuracies and test losses in Stage 2 for the two knowledge transfer methods, considering a pretrained 14-layer ResNet (recognition rate [P_c] = 90.79%) and an 8-layer ResNet in the TSF. The recognition accuracy of the original eight-layer student ResNet without teacher knowledge was 88.09% (Case 5 in Figure 4). The trained student ResNet, using the proposed method (Case 2 and Case 4 in Figure 4), performed better in terms of both accuracy and loss than the existing method (Case 1 and Case 3 in Figure 4). Hence, the proposed layer-wise hint training scheme using multiple hint and guided layers provided a well-trained student network via layer-wise transfer of multiple hints from the pretrained teacher network.

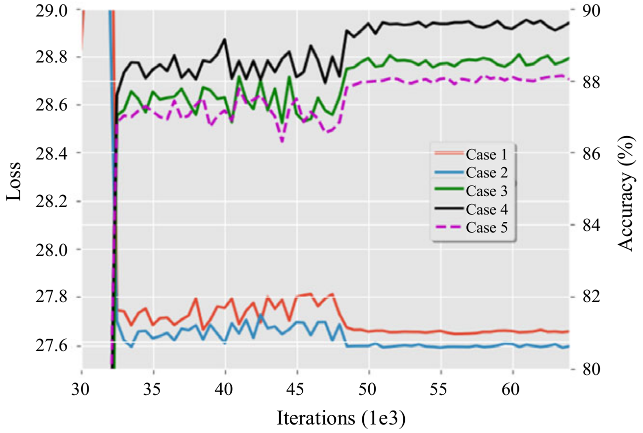


FIGURE 4 Comparison of recognition accuracy and test loss in Stage 2 in the TSF. Case 1: Test loss of student ResNet using the existing hint-KD training method. Case 2: Test loss of student ResNet using the proposed method. Case 3: Recognition accuracy of student ResNet using the existing hint-KD training method. Case 4: Recognition accuracy of student ResNet using the proposed method. Case 5: Recognition accuracy of the original student ResNet without teacher knowledge

Table 1 compares the recognition accuracies of the proposed method and existing knowledge transfer methods of the pretrained 26-layer teacher ResNet (with 91.75% accuracy) and 14-layer student ResNet. All experimental specifications applied to each training method were the same as those described in Figure 4. Note that, for all methods except the existing KD method, we copied the result from Stage 1 to several student ResNets with the same topology (Net 1, Net 2, and Net 3 in Table 1) for the subsequent Stage 2. To train the student ResNets in Stage 2, the three Nets used different random parameter initializations for the remaining weights that did not participate in the training of Stage 1. In this experiment, we added the existing KD training method without hint information and a hint-KD$^+$ training method for performance comparison, where the latter method (Hint-KD$^+$ in Table 1) utilized the whole of each teacher and student ResNet—except the fully connected (FC) layer—instead of using the intermediate hidden layer pair, thus applying a single hint layer and a single guided layer to the hint-based training. Compared to the KD training method, the existing hint-KD training method showed better recognition accuracy owing to the stage-wise training that used the intermediate result-based hint information and τ–based KD

TABLE 1 P_c (%) on CIFAR-10 for the 26-layer teacher ResNet and 14-layer student ResNet in the TSF

Method	Net 1	Net 2	Net 3	Avg.	Reference
KD without Hint	90.77	90.78	90.74	90.76	P_c = 90.79% for the original 14-layer ResNet
Original Hint-KD	91.05	91.23	91.19	91.15	
Hint-KD+	90.77	90.37	90.75	90.63	
Proposed approach	91.66	91.80	91.64	**91.70**	

Bold value means the highest average recognition rate in Table 1.

information (P_c = 90.76% → 91.15%). In addition, it can be seen in Table 1 that hint training of the whole network is inferior to the original hint training approach using intermediate hint/guided layers (P_c = 90.63% → 91.15%). However, the trained student network using the proposed method outperformed the student network using the existing hint-KD training method (P_c = 91.15% → 91.7%). Furthermore, although the number of layers in the student ResNet is reduced to 46.15% of those in the 26-layer teacher ResNet, the 14-layer student ResNet trained using the proposed method clearly showed a high level of performance, close to that of the teacher ResNet.

Next, we analyze the recognition accuracy of the proposed method using CIFAR-100 [32]; this dataset is similar to CIFAR-10, except it has 100 classes, containing 600 images each. Because of the small number of images per class, we adopted wide ResNet structures ({64, 128, 256} filters)—four times more than those described in the CIFAR-10 case. A 20-layer teacher ResNet model was pre-trained with the CIFAR-100 dataset (batch size = 128), achieving 74.43% accuracy. The same data augmentation as in CIFAR-10 was adopted in this experiment. The accuracy of the original eight-layer student ResNet without teacher knowledge was 69.51%, using the normal training procedure [5] over 64,000 iterations.

For the first stage of the existing hint-based KD training method, hint-based training was implemented using a learning rate of 1e-4, to minimize the l_2 loss between outputs of the two hint/guided layers over 35,000 iterations. Then, we followed the same KD training procedure described in the CIFAR-10 case for 64,000 iterations. In the proposed method, layer-wise hint training was implemented in Stage 1, using a learning rate of 1e-4 for the same 35,000 iterations (5,000 for $\hat{\mathbf{W}}_{G_1}$, 15,000 for $\hat{\mathbf{W}}_{G_2}$, and 15,000 for $\hat{\mathbf{W}}_{G_3}$). The remaining KD training (Stage 2) in the proposed method was also performed over 64,000 iterations under the previous learning rate policy (ie, learning rates of 0.1, 0.01, and 0.001 until 32,000, 48,000, and 64,000 iterations, respectively). We also compared the recognition accuracy of the KD training method without hints and the hint-KD$^+$ training method on CIFAR-100 by averaging the predictions of three trained eight-layer student ResNets (Figure 5). The recognition accuracy of the proposed method, shown in Figure 5, is better than that of the three knowledge transfer methods.

Table 2 shows the recognition accuracies when a 26-layer teacher ResNet model (with 74.65% accuracy) was applied to all knowledge transfer methods with the same learning rate policy and training iterations as in Figure 5. We also copied the result from Stage 1 to several student ResNets (Net 1, Net 2, and Net 3 in Table 2). The trained eight-layer student ResNet using hint-based KD training demonstrates improved performance compared to the existing KD training method, as well as to the original student ResNet trained using a standard learning method without the teacher's knowledge. In this case, as in CIFAR-10, observe that hint-KD$^+$ training using the whole network is inferior to the original hint-based KD method.

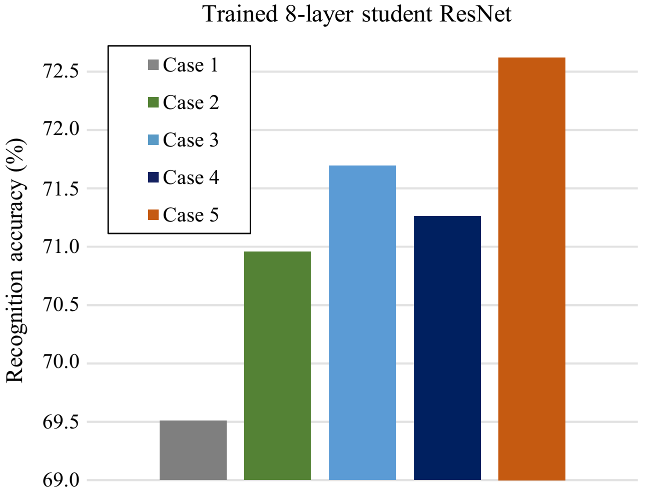


FIGURE 5 P_c (%) on CIFAR-100 for the 20-layer teacher ResNet and 8-layer student ResNet in the TSF. Case 1: Original 8-layer student ResNet without teacher knowledge. Case 2: KD without hints. Case 3: Original Hint-KD. Case 4: Hint-KD$^+$. Case 5: Proposed method

TABLE 2 P_c (%) on CIFAR-100 for the 26-layer teacher ResNet and 8-layer student ResNet in the TSF

Method	Net 1	Net 2	Net 3	Avg.	Reference
KD without Hints	70.60	71.25	70.66	70.83	P_c = **69.51%** for the original 8-layer ResNet
Original Hint-KD	71.73	71.57	71.93	71.74	
Hint-KD$^+$	71.24	71.60	71.35	71.39	
Proposed method	72.49	72.91	73.07	**72.82**	

Bold value means the highest average recognition rate in Table 2.

However, similar to the CIFAR-10 example, the proposed training approach also outperformed the existing hint-KD training method for the CIFAR-100 dataset (71.74% → 72.82%). Consequently, as shown in Figure 5 and Table 2, the trained student ResNet using the proposed hint training method was superior to student ResNets with the existing KD or hint-KD training, as well as to the original student ResNet without teacher knowledge.

3.2 | Proposed hint training using MNIST

To further validate the performance of the proposed training approach, we used the MNIST dataset, a large database of handwritten digits that consists of 60,000 grayscale training images and 10,000 test images [33]. In this experiment, the ResNet architecture was the same as that in Figure 2 ({16, 32, 64} filters). The only difference was the

TABLE 3 P_e (%) on MNIST for the 32-layer teacher ResNet and 8-layer student ResNet in the TSF

Method	Net 1	Net 2	Net 3	Avg.	Reference
KD without Hints	0.56	0.57	0.55	0.56	P_e = **0.6**% for the original 8-layer ResNet
Original Hint-KD	0.51	0.56	0.45	0.506	
Hint-KD$^+$	0.60	0.50	0.61	0.57	
Proposed approach	0.41	0.40	0.44	**0.416**	

Bold value means the lowest average error rate in Table 3.

feature map size, which was {28, 14, 7}, because the input images were 28 × 28 pixels. We prepared a pretrained 32-layer teacher ResNet that achieved an error rate P_e of 0.39% using learning rates of 0.1, 0.01, and 0.001 for 18,000, 27,000, and 36,000 iterations, respectively. P_e is defined as $1 - P_c$. A mini-batch size of 64 was used to train the 32-layer teacher ResNet without data preprocessing.

For the existing hint-KD training method in Stage 1, we used 25,000 iterations to train a TSF using the 32-layer teacher ResNet and 8-layer student ResNet. A learning rate of 1e-4 was used for 25,000 iterations. In Stage 2, we used the same learning rate policy and training iterations described above, up to 36,000 iterations. In this experiment, we set λ = 5, which also provided better accuracy than other λ values.

For the proposed method, Stage 1 was also trained using a learning rate of 1e-4 over 25,000 iterations (3,000 for $\hat{\mathbf{W}}_{G_1}$, 7,000 for $\hat{\mathbf{W}}_{G_2}$, and 15,000 for $\hat{\mathbf{W}}_{G_3}$). Next, we performed Stage 2 for 36,000 iterations using the same parameters as for the existing hint-based KD training method. Table 3 summarizes the comparative recognition accuracy results for the knowledge transfer methods. Note that the original eight-layer student ResNet without teacher knowledge achieves $P_e = 0.6\%$. The average accuracy of the trained student network using the proposed method is superior to those of the three other TSF-based training methods. Considering all experimental results presented in Sections 3.1 and 3.2, we can conclude that the proposed method is more useful for knowledge transfer using hint and KD information, than existing methods.

3.3 | Comparison with layer-wise pretraining

Of the TSF-based knowledge transfer methods using multiple hints mentioned in Section 2.2, the proposed training approach can be categorized as a layer-wise pretraining method [29,34,35] because the student ResNet learns the teacher's hints from bottom to top layer-by-layer. Traditional unsupervised layer-wise pretraining using restricted Boltzman machines is difficult to apply directly to the skip connections and BNs of the ResNet structure. Instead, supervised CNN-based layer-wise pretraining [29] can be applied to the ResNet topology. Therefore, we introduce supervised layer-wise pretraining of the ResNet (SL-ResNet), which also comprises two stages: layer-wise training and fine-tuning.

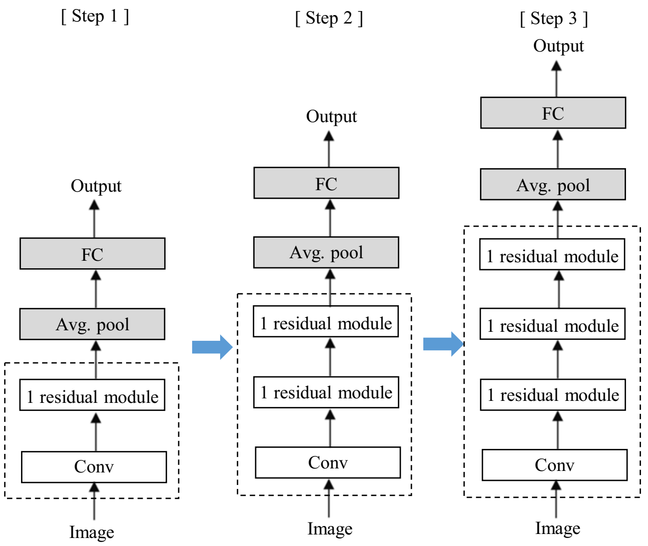


FIGURE 6 Block diagram of the eight-layer SL-ResNet in Stage 1

For example, when considering the eight-layer ResNet model for the SL-ResNet, Stage 1 is implemented with layer-by-layer training per residual module in three steps, as shown in Figure 6. First, Stage 1 of the SL-ResNet is performed by building the model incrementally by adding a residual module and training it before adding more residual models. Based on [29], the global average pooling layer (avg. pool in Figure 6) and a fully connected layer (FC in Figure 6) are added every time when a new residual module is added for each step. Here, the old pooling layer and fully connected layer are obviously removed before the addition of new ones. As in [29], each step was trained for the same number of iterations: 12,000 iterations for CIFAR-10 and 8,000 iterations for MNIST. To train each step, we used a momentum gradient descent (MGD) optimizer instead of the RMSprop [29] because MGD performed better in this study when using the ResNet structure. After completing layer-wise training in Stage 1, fine-tuning is implemented for 64,000 iterations using the same training procedures as in Sections 3.1 and 3.2. Note that fine-tuning usually employs small learning rates; however, because we found that a base learning rate of 0.1 was better than smaller values, we decided on the following learning rate policy: learning rates of 0.1, 0.01, and 0.001 until 32,000, 48,000, and 64,000 iterations, respectively, for CIFAR-10, and learning rates of 0.1, 0.01, and 0.001 until 18,000, 27,000, and 36,000 iterations, respectively, for MNIST. To compare the performance of the proposed

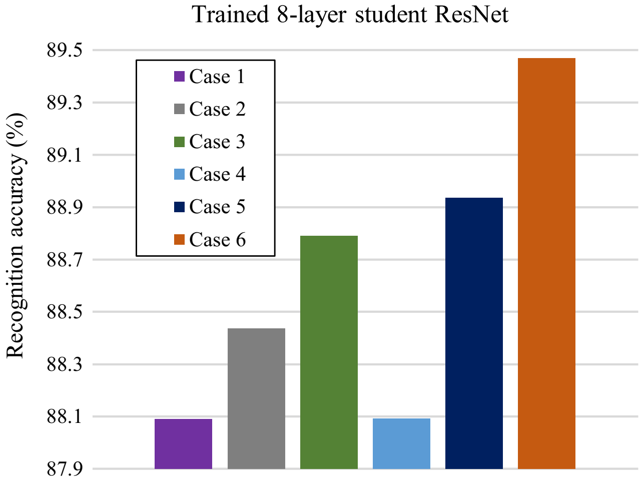


FIGURE 7 P_c (%) on CIFAR-10 for the 14-layer teacher ResNet and 8-layer student ResNet. Case 1: Original 8-layer student ResNet without teacher knowledge. Case 2: KD without hints. Case 3: Original Hint-KD. Case 4: SL-ResNet. Case 5: SL-ResNet with KD. Case 6: Proposed method

TABLE 4 P_e (%) on MNIST for the 32-layer teacher ResNet and 8-layer student ResNet

Method	Net 1	Net 2	Net 3	Avg.	Reference
KD without Hints	0.58	0.58	0.56	0.573	P_e = 0.6% for the original 8-layer ResNet
Original Hint-KD	0.52	0.54	0.62	0.56	
SL-ResNet	0.49	0.49	0.51	0.496	
SL-ResNet with KD	0.49	0.47	0.41	0.456	
Proposed method	***0.38**	0.40	0.39	**0.39**	

Bold value means the lowest average error rate in Table 4.
*This value means the lowest individual error rate in Table 4.

method, we also applied KD training to Stage 2 of the SL-ResNet under the same KD training procedure described in Sections 3.1 and 3.2.

As reported in Figure 7 and Table 4, the SL-ResNet with KD exhibited better performance than the existing KD and hint-KD training methods. For the SL-ResNet with KD, λ was set to 5 for CIFAR-10 and 3 for MNIST. In addition, the trained eight-layer ResNet using SL-ResNet without KD outperformed the eight-layer student ResNets using the two existing knowledge transfer methods for MNIST, although the performance of the SL-ResNet without KD (Case 4 in Figure 7) was worse than that of both other methods (Case 2 and Case 3 in Figure 7) for CIFAR-10. The proposed method clearly surpassed the SL-ResNet both with and without KD. Furthermore, eight-layer student Net 1, using the proposed method, surpassed the performance ($P_e = 0.38\%$ in Table 4) of the 32-layer teacher ResNet ($P_e = 0.39\%$). Note that the layers were reduced by 75% compared to the original 32-layer teacher ResNet. These results verified that the proposed hint training can provide good initial weights for training Stage 2 compared to the layer-wise training of an SL-ResNet, as well as the existing hint training.

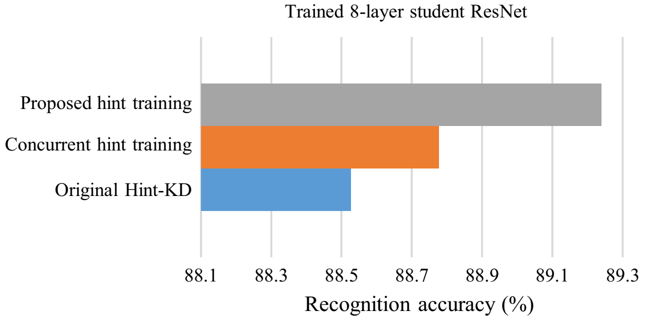


FIGURE 8 P_c (%) on CIFAR-10 for the 26-layer teacher and 8-layer student ResNet in the TSF

4 | DISCUSSION

In the TSF, when using multiple hint/guided layers to transfer teacher knowledge, concurrent hint training using multiple loss functions can also be considered in lieu of iterative layer-wise hint training. By simultaneously using N l_2 loss functions applied to N hint/guided layers, the concurrent hint training approach is given as:

$$\left(\hat{\mathbf{W}}_G\right) = \underset{\mathbf{W}_{G_i}}{\arg\min}\, \frac{1}{2N}\sum_{i=1}^{N} a_i \cdot \|\mathbf{F}_H^i(x;\mathbf{W}_{H_i}) - \mathbf{F}_G^i(x;\mathbf{W}_{G_i})\|_2^2, \tag{7}$$

where a_i denotes the weighting factor for each loss function, and $\hat{\mathbf{W}}_G$ comprises the parameters obtained from concurrent hint training. All N selected hint/guided layers were used to simultaneously minimize loss terms of (7) during Stage 1 of hint training. Note that, unlike concurrent hint training, the proposed method performs bottom-up step-by-step hint training using multiple hint/guided layers.

Figure 8 shows the comparative results of the average recognition accuracy using each method under the same experimental conditions as in Section 3.1 (CIFAR-10). For the concurrent training approach, we selected the same three pairs of hint/guided layers as the proposed method and set equal weighting for $\{a_i\}_{i=1}^{N=3}$. Figure 8 shows that the accuracy of the concurrent hint training (orange bar) is definitely inferior to that of the proposed hint training (gray bar).

Furthermore, Figure 9 compares hint training losses under two knowledge transfer methods (concurrent hint training and proposed hint training) for three pairs of hint/guided layers in the TSF. In the results, three loss terms (loss 1, loss 2, and loss 3)—corresponding to $\{\mathrm{H}_i - G_i \text{ layers}\}_{i=1}^{3}$—are presented, where each loss term is normalized. It can be seen in Figure 9 that the proposed training approach (solid line) achieves a lower hint training

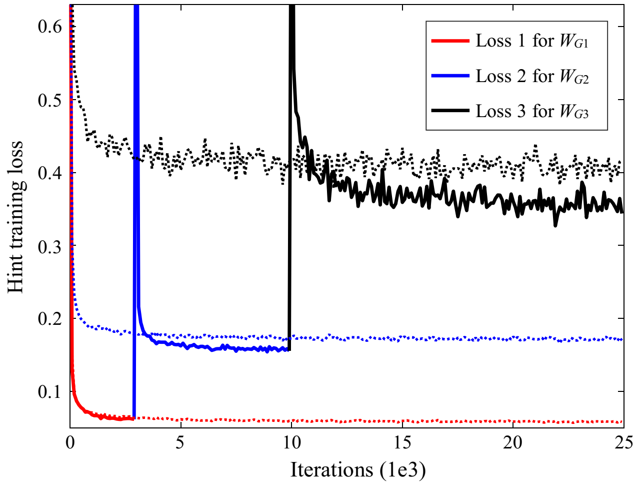


FIGURE 9 Hint training on CIFAR-10 for the 26-layer teacher ResNet and 8-layer student ResNet. Dashed line: Concurrent hint training approach. Solid line: Proposed hint training approach

loss than the concurrent training approach (dashed line), although knowledge transfer is considered in the higher hint/guided layers.

However, it can be difficult to obtain a well-trained student network using simultaneous training with multiple loss functions on multiple hint/guided layers, which can lead to a higher hint training loss than the proposed training approach. In contrast, layer-wise training in the proposed approach can overcome this problem by incrementally training the student network using each single loss function, even when multiple hint/guided layers are used. This implies that layer-wise hint training is preferable to concurrent hint training when transferring teacher knowledge using multiple hint/guided layers in the TSF.

As shown in Figure 10, this phenomenon is also observed for the CIFAR-100 dataset (Section 3.1). The proposed hint training method (gray bar) exhibits superior performance over both concurrent hint training (orange bar) and the existing hint-based KD method (blue bar). Consequently, as shown in Figures 8–10, a TSF based on the proposed hint training in Section 2.2 is preferable to a framework based on the concurrent hint training (as in (7)) to efficiently transfer teacher knowledge to a student network through multiple hint/guided layers.

Next, for hint-based training of Stage 1, the teacher and student ResNets considered in this study were both ResNet models with the same spatial dimensions; that is, the student ResNet acquires hint-based teacher information such that hidden layer features of the student ResNet directly resemble that of the teacher ResNet by minimizing the l_2 loss between the two layer features. In contrast, if the teacher and student ResNets have different spatial dimensions, an additional regression function should be added theoretically between the hint layer feature and the guide layer feature (as in [23]) to match the spatial dimension. Table 5 represents the recognition accuracy with a convolutional regressor function in the original hint-KD training method and the proposed method, where the two methods used the same parameter settings as in Table 1. In this experiment, we prepared a 26-layer teacher ResNet with {32, 64,128} filters, which has spatial dimensions two times wider than the teacher ResNet structure (with {16, 32, 64} filters) in Table 1. The accuracy of the wider teacher 26-layer ResNet was 93.36% when using the normal training procedure [5] during 64,000 iterations. Based on [23], we adopted a convolutional regressor with Gaussian initialization and no bias term. Note that a TSF without a regressor has teacher and student ResNet models with the same spatial dimensions. For both methods, the 14-layer student ResNet trained using the TSF with the regressor (Table 5) showed better accuracy than the student ResNet trained using the TSF without the regressor (Table 1). As expected, even when using the regressor, the proposed method outperformed

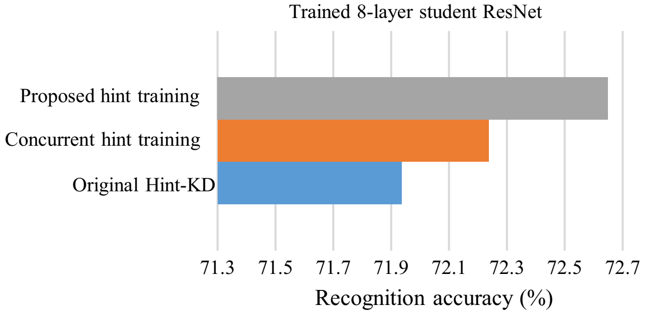


FIGURE 10 P_c (%) on CIFAR-100 for the 14-layer teacher and 8-layer student ResNet in the TSF

TABLE 5 P_c (%) on CIFAR-10 for the 26-layer teacher ResNet and 14-layer student ResNet in the TSF

Method	Net 1	Net 2	Net 3	Avg.	Reference
Original Hint-KD	91.25	91.31	91.09	91.21 ([a]91.15)	26-layer teacher ResNet with {32, 64, 128} filters
Proposed method	92.01	92.05	91.97	**92.01** ([a]91.7)	

Bold value means the highest average recognition rate in Table 5.
[a]TSF without regressor in Table 1 is used for both methods.

TABLE 6 P_c (%) on CIFAR-100 for the 14-layer teacher ResNet and 8-layer student ResNet in the TSF

Method	Net 1	Net 2	Net 3	Avg.	Reference
Original Hint-KD	71.88	72.16	72.35	72.13 ([a]71.85)	14-layer teacher ResNet with {80, 160, 320} filters
Proposed method	73.07	72.94	73.32	**73.11** ([a]72.65)	

Bold value means the highest average recognition rate in Table 6.
[a]TSF without regressor in Figure 10 is used for both methods.

the existing hint-KD training method (P_c = 91.21% → 92.01%).

Next, recognition accuracies on CIFAR-100 for the two methods were compared in Table 6 for a 14-layer teacher ResNet with {80, 160, 320} filters and an 8-layer student ResNet with {64, 128, 256} filters in a TSF. The pretrained 14-layer teacher ResNet using {80, 160, 320} filters provided 73.49% accuracy. Because the teacher and student ResNets have different spatial dimensions, we used the same type of convolutional regressor as that used in Table 5. As shown in Table 6, the proposed method is superior to the existing hint-KD training method for the TSF with the regressor (P_c = 72.13% → 73.11%), as well as that for the TSF without the regressor (P_c = 71.85% → 72.65%). From the results shown in Tables 5 and 6, the TSF using the regressor provided much better student ResNets than the TSF without the regressor for both training methods. Using the regressor can allow the student ResNet with narrow hidden layers to learn from this wider teacher ResNet, where the teacher ResNet with the wider hidden layers showed better recognition accuracy than the original teacher ResNet used in the TSF without regressor. Therefore, the student ResNet can benefit from this wider teacher ResNet, which provides higher accuracy performance, despite having different spatial dimensions. This also preserves the computational efficiency of using the student ResNet with narrow hidden layers. In future work, we will further address efficient knowledge transfer for TSF structures with various spatial dimensions.

5 | CONCLUSION

In this paper, we proposed a layer-wise hint training method to improve existing knowledge transfer methods using the TSF. To efficiently transfer pretrained teacher knowledge to a student network, the proposed method is composed of two main stages: (i) iterative and layer-wise training using pretrained hints between multiple hint layers and guided layers, and (ii) τ-based KD training using the hint-based information extracted from Stage 1.

To validate the effectiveness of the proposed method, we compared its recognition accuracy to that of the ResNet-based layer-wise pretraining method as well as the existing TSF-based training methods on several reliable datasets. State-of-the-art ResNets with different layers and the same spatial dimensions were utilized in the TSF.

Based on the step-by-step hint training approach described in Section 2.2, the advantages of the proposed method can be summarized as follows. First, by selecting multiple hint and guided layers, more pretrained teacher knowledge, including low-level detailed features and high-level abstracted features—from the lower hint layer to the upper hint layer—is considered for knowledge transfer than the existing hint-based training approach using the intermediate hint layer feature and intermediate guided layer feature. Next, repetitive layer-wise training and layer-wise knowledge transfer from the bottom to top can improve the recognition accuracy of the small student network. As a result, useful information that is inherent in the hidden layers of the complex teacher network can be more accurately conveyed.

Consequently, the results showed that the proposed method of using layer-wise hint-based information was superior to the existing hint-KD training method of using the intermediate result-based hint information when transferring the pretrained teacher-network hint and KD information to the student network. In addition, although KD was applied to the teacher SL-ResNet, the proposed method provided a more accurately optimized student network than both the SL-ResNet without KD and SL-ResNet with KD.

ACKNOWLEDGMENTS

The authors thank the associate editor and the anonymous reviewers for their valuable comments and suggestions, which improved the quality of this paper.

ORCID

Ji-Hoon Bae https://orcid.org/0000-0002-0035-5261

REFERENCES

1. Y. LeCun et al., *Gradient-based learning applied to document recognition*, Proc. IEEE **86** (1998), 1–46.
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *26th Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, Stateline, NV, USA, Dec. 3–8, 2012, pp. 1106–1114.
3. C. Szegedy et al., Going deeper with convolutions, *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Boston, MA, USA, June 7–12, 2015, pp. 1–9.
4. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Proc. 5th Int. Conf. Learning Represent. (ICLR)*, San Diego, CA, USA, May 7–9, 2015, pp. 1–14.
5. K. He et al., Deep residual learning for image recognition, *Proc. IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Las Vegas, NV, USA, June 26–July 1, 2016, pp. 1–12.
6. A. Veit, M. Wilber, and S. Belongie, *Residual networks are exponential ensembles of relatively shallow networks*, arXiv preprint arXiv: 1605.06431 (2016), 1–12.
7. S. L. Phung and A. Bouzerdoum, *A pyramidal neural network for visual pattern recognition*, IEEE Trans. Neural Netw. **18** (2007), 329–343.
8. K. Simonyan and A. Zisserman, Two-stream convolutional networks for action recognition in videos, *Proc. 27th Neural Inform.*

Process. Sys. Conf. (NIPS), Montreal, Canada, Dec. 8–13, 2014, pp. 1–9.

9. M. Lin, Q. Chen, and S. Yan, Network in network, *Proc. Int. Conf. Learning Represent. (ICLR)*, Banff, Canada, Apr. 14–16, 2014, pp. 1–10.
10. R. Girshick, Fast R-CNN, *Proc. Int. Conf. Compu. Vision (ICCV)*, Santiago, Chile, Dec. 11–18, 2015, pp. 1440–1448.
11. M. Liang and X. Hu, Recurrent convolutional neural network for object recognition, *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Boston, June 7–12, 2015, pp. 3367–3375.
12. J. Donahue et al., Long-term recurrent convolutional networks for visual recognition and description, *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Boston, MA, USA, June 7–12, 2015, pp. 1–14.
13. J. Yim et al., Rotating your face using multi-task deep neural network, *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Boston, MA, USA, June 7–12, 2015, pp. 676–684.
14. C. Park et al., *Korean conference resolution with guided mention pair model using the deep learning*, ETRI J. **38** (2016), 1207–1217.
15. Y. Zhang et al., *Adaptive convolutional neural network and its application in face recognition*, Neural Process. Lett. **43** (2016), 389–399.
16. D. Han, J. Kim, and J. Kim, Deep pyramidal residual networks, *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Honolulu, HI, USA, July 21–26, 2017, pp. 5927–5935.
17. G. Huang, Z. Liu, and L. Maaten, Densely connected convolutional networks, *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Honolulu, HI, USA, July 21–26, 2017, pp. 2261–2269.
18. G. Huang et al., Densely connected convolutional networks, *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Honolulu, HI, USA, July 21–26, 2017, pp. 2261–2269.
19. M. Brahimi, *Deep learning for tomato diseases: classification and symptoms visualization*, Appl. Artif. Intell. **31** (2017), no. 4, 1–17.
20. J. Yun and B.-J. Jang, *Ambient light backscatter communication for IoT applications*, J. Kor. Electromag. Eng. Soc. **16** (2016), no. 4, 214–218.
21. J. Li et al., Learning small-size DNN with output-distribution-based criteria, *Proc. INTERSPEECH*, Singapore, Sept. 14–18, 2014, pp. 1910–1914.
22. G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531, 2015, pp. 1–19.
23. A. Romero et al., Fitnets: hints for thin deep nets, *Proc. 5th Int. Conf. Learning Represent. (ICLR)*, San Diego, CA, USA, May 7–9, 2015, pp. 1–13.
24. I. J. Goodfellow et al., *Maxout networks*, arXiv:1302.4389, 2013, pp. 1–9.
25. T. Chen, I. Goodfellow, and J. Shlens, Net2Net: accelerating learning via knowledge transfer, *Proc. 6th Int. Conf. Learning Represent. (ICLR)*, San Juan, Puerto Rico, May 2–4, 2016, pp. 1–12.
26. C. Bucilua, R. Caruana, and A. Niculescu-Mizil, Model compression, *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Philadelphia, PA, USA, Aug. 20–23, 2016, pp. 535–541.
27. Y. Jia et al., Caffe: convolutional architecture for fast feature embedding, *Proc. 22th ACM Int. Conf. Multimedia (ACM MM)*, Orlando, FL, USA, Nov. 3–7, 2014, pp. 675–678.
28. Caffe, *Deep learning framework*, available at http://caffe.berkeleyvision.org/.
29. S. Roy et al., *Handwritten isolated Bangla compound character recognition: a new benchmark using a novel deep learning approach*, Pattern Recogn. Lett. **90** (2017), 15–21.
30. S. Ioffe and C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariant shift, *Proc. 32nd Int. Conf. Machine Learning (ICML)*, Lille, France, July 6–11, 2015, pp. 1–9.
31. K. He et al., Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, Boston, MA, USA, June 7–12, 2015. pp. 1026–1034.
32. *CIFAR-10 and CIFAR-100 datasets*, available at https://www.cs.toronto.edu/~kriz/cifar.html.
33. *MNIST dataset*, available at http://yann.lecun.com/exdb/mnist/.
34. G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science **313** (2006), 504–507.
35. Y. Bengio et al., Greedy layer-wise training of deep neural networks, *Proc. Neural Inform. Process. Syst. Conf. (NIPS)*, Vancouver, Canada, Dec. 4–7, 2006, pp. 153–160.

AUTHOR BIOGRAPHIES

Ji-Hoon Bae received his BS degree in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2000, and his MS and PhD degrees in electrical engineering from Pohang University of Science and Technology, Pohang, Kyungbuk, Rep. of Korea, in 2002 and 2016, respectively. Since 2002, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a principal researcher. His research interests include deep learning, transfer learning, radar imaging, HF/UHF RFID digital modem, array antennas, and optimized techniques.

Junho Yim received his BS and MS degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2012 and 2015, respectively. He is currently working toward his PhD degree in Statistical Inference and Information Theory Laboratory in the Department of Electrical Engineering, KAIST. His research interests include deep learning in computer vision, image classification, and image detection. He is a student member of the IEEE.

Nae-Soo Kim received his BS and MS degrees in Mathematics from Hannam University, Rep. of Korea, in 1985 and 1989, respectively. He received a PhD degree in Computer Engineering from Hannam University, Rep. of Korea, in 2001. After having worked for the Agency for Defense Development, Rep. of Korea from 1976 to 1990, he joined the Electronics and Telecommunications Research Institute (ETRI), Rep. of Korea in 1990, and has been involved with satellite communication and broadcasting system development projects until 2004. Since 2005, he has worked on Internet of Things (IoT) key technology and IoT S/W platform based on smart devices, including RFID and wireless sensor networks in ETRI. He worked as a project manager of several projects and also as the team leader. His current research interests involve developing machine learning and domain knowledge base technologies for artificial intelligence. Currently, he is a director of the Knowledge-converged Super Brain (KSB) Convergence Research Department at ETRI.

Cheol-Sig Pyo received his BS in electronic engineering from Yonsei University, Rep. of Korea, in 1991, and his MS in electrical engineering from Korea Advanced Institute of Science and Technology, in 1999. Since 1991, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is currently a managing director of the Knowledge-converged Super Brain (KSB) Convergence Research Department. His research interests include artificial intelligence, machine/deep learning, and Internet of Things (IoT).

Junmo Kim received his BS degree in electrical engineering from Seoul National University, Seoul, Rep. of Korea, in 1998 and his MS and PhD degrees in Electrical Engineering and Computer Science from Massachusetts Institute of Technology, Cambridge, USA, in 2000 and 2005, respectively. From 2005 to 2009, he was with the Samsung Advanced Institute of Technology, Suwon, Gyeonggi-do, Rep. of Korea, as a Research Staff Member. He joined the faculty of Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2009, where he is currently an associate professor of Electrical Engineering. His research interests include deep learning, image processing, computer vision, statistical signal processing, and information theory.