Article

Multiple-Stage Knowledge Distillation

Chuanyun Xu 1,2, Nanlan Bai 1,*, Wenjian Gao 1, Tian Li 3, Mengwei Li 1, Gang Li 1,† and Yang Zhang 2

1 School of Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China
2 College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
3 Computer Science Department, RWTH Aachen University, 52074 Aachen, Germany
* Correspondence: bainanlan@stu.cqut.edu.cn (N.B.); ligang@cqut.edu.cn (G.L.)

Abstract: Knowledge distillation (KD) is a method in which a teacher network guides the learning of a student network, thereby resulting in an improvement in the performance of the student network. Recent research in this area has concentrated on developing effective definitions of knowledge and efficient methods of knowledge transfer while ignoring the learning ability of the student network. To fully utilize this potential learning ability and improve learning efficiency, this study proposes a multiple-stage KD (MSKD) method that allows students to learn the knowledge delivered by the teacher network in multiple stages. The student network in this method consists of a multi-exit architecture, and the students imitate the output of the teacher network at each exit. The final classification by the student network is achieved through ensemble learning. However, because this results in an unreasonable gap between the number of parameters in the student branch network and those in the teacher branch network, as well as a mismatch in learning capacity between these two networks, we extend the MSKD method to a one-to-one multiple-stage KD method. The experimental results reveal that the proposed method applied to the CIFAR100 and Tiny ImageNet datasets exhibits good performance gain. The proposed method of enhancing KD by changing the style of student learning provides new insight into KD.

Keywords: knowledge distillation; multi-exit architecture; teacher–student network; ensemble learning

1. Introduction

In the past few decades, deep neural network (DNN) models have achieved remarkable success in various computer vision tasks, such as image classification [1], object detection [2], and semantic segmentation [3]. Additionally, this has had an impact on various fields [4]. DNN models can effectively extract features by augmenting “width” and “depth” to different degrees; however, increasing the number of layers in the model will increase the training difficulty. Therefore, other methods must be explored to improve the performance of the model.

Knowledge distillation (KD) is a method of model compression [5–7] that can be applied to improve the performance of a model [8,9]. In the conventional KD method, the teacher network has a deeper layer and more parameters than those of the student network. When the student network mimics the output of the teacher network, the accuracy of the student network is increased and model compression is achieved. KD utilizes the “dark knowledge” of the teacher network to direct and increase the performance of the student network. Existing research reveals that the student network can outperform the teacher network [10]. The current focus of KD research is on studying the categories and methods of knowledge transfer; however, the methods of activating the learning potential of student networks are yet to be investigated.

The most effective teaching method fully utilizes the learning ability of a student. Additionally, to maximize the learning ability of a student, an appropriate learning style must be adopted. Furthermore, the general process of human learning is usually divided into several stages, such as a preview before class, classroom learning, and review after...
class. Each stage contributes to a different level of understanding, which can be revised with the assistance of the teacher and books. Eventually, the imparted knowledge can be correctly understood. Figure 1 describes our idea. The teaching process includes the knowledge imparted by the teacher and learning by the student, both of which are essential. Students’ efforts alone cannot be used to improve the effectiveness of teaching; the teacher’s methods of teaching are equally crucial. Since students have limited ability to accept the knowledge transferred by powerful teachers, teachers must provide targeted instruction according to the different stages of learning of the student and impart knowledge that is easy to understand at the current learning stage to achieve better teaching outcomes.

Figure 1. Illustration of the proposed methods.

Inspired by the human learning process, a multiple-stage KD method is proposed in this study that modifies the student network in the conventional KD method by dividing the network into several parts according to different depths. Additionally, simplified attention modules, convolutional blocks, and classifiers are added to form a multi-exit architecture student network with a general network backbone. The multiple exits of the student network are used to learn the information delivered by the teacher network in stages for improving network performance. The entire student network is considered to be involved in the learning process with multiple exit branch networks representing various learning stages. Each stage of learning has a particular focus; for example, the early stage of learning focuses on gaining general knowledge of the content, progressing through multiple stages of learning, and finally consolidating the total learning from all the stages. The current KD research focuses on the types of knowledge imparted by the teacher to students while neglecting the exploration of the different learning styles of the student network. Therefore, under the premise of using simple logits as the transmission knowledge, we altered the learning mode of the student network using the multi-exit architecture to fully utilize the learning ability of the student network. However, because the performance of the student network decreases, owing to the lack of learning ability when the capability gap between the teacher and student networks is too large [11], the teacher network possesses a multi-exit architecture similar to that of the student network, in which the early exit of the teacher network is used to guide the corresponding early exit of the student network to achieve targeted guidance. This allows students to adapt to the current stage of learning, and the teacher network can impart comprehensive knowledge, thereby improving teaching efficacy. We validated these methods using two classification datasets, namely CIFAR100 [12] and Tiny ImageNet [13], and analyzed the experimental results.

The main objectives of this study are summarized as follows:
• We propose a multi-stage learning approach where the student can learn the teacher’s understanding of the original knowledge at different stages and the student is able to gradually get an identical understanding with the teacher, even becoming superior to the teacher.
• We propose a multiple-stage KD (MSKD) method for the supervised training of different stages of a multi-exit network. In the MSKD, the output of the teacher network is used to guide the different stages of the student network, and the trained early stage network is trained later in the training process, thereby resulting in phased recursive training.
• We extend the MSKD method to a one-to-one MSKD (OtO) method that allows a multi-exit teacher network to provide one-to-one targeted instruction to a multi-exit student network.
• Extensive experiments were conducted using CIFAR100 and Tiny ImageNet to test different teacher–student network architectures and verify the effectiveness of the proposed method.

2. Related Work

2.1. KD

KD is a common technique for knowledge transfer between two models, which was first discussed in detail by Hinton et al. [14]. In this technique, a temperature was used to soften the output of the teacher network and minimize the Kullback–Leibler divergence of the outputs of the teacher and student networks. Several relevant theoretical explanations [15–18], improved methods [5,19,20], and other applications [21,22] of KD exist. Instead of focusing on the output of the network, Komodakis et al. [5] investigated the feature map of the teacher–student network to minimize the differences in this feature map. Passalis et al. [23] focused on the geometry of the feature space of the teacher network and mapped it effectively to the feature space of the student network. Liu et al. [24] reported the important role of inter-channel correlations in retaining features and aligning the diversity and isomorphism of the feature spaces of the student and teacher networks. Wen et al. [25] adapted the knowledge transferred by the teacher network and used dynamic temperature penalization in case of poor supervision to improve the student network. Tian et al. [26], Xu et al. [27], and Yang et al. [28] combined self-supervision with KD using different definitions of a pretext task, thereby increasing the effectiveness of the knowledge transfer. Notably, these methods usually use data augmentation operations, such as rotating images, to define the pretext task. Unlike the previous model, in which students learned unilaterally from the teacher, Fu et al. [29] utilized an interactive teaching model that increased the effectiveness of KD by the teacher–student network. Most studies have focused on exploring different knowledge categories and how knowledge is transferred while neglecting to explore the different learning styles of student networks. To explore the impact of student learning styles, we chose to use a simple definition of knowledge in our method.

2.2. Multi-Exit Architectures

Teerapittayanon et al. [30] was the first study to propose the use of early exit branch networks to achieve fast inference in DNNs. Subsequently, Huang et al. [31] discovered that added early exits affect the performance of the final exits and the early exits do not possess coarse-scale features; therefore a multi-scale dense network was proposed. Additionally, two experimental conditions were set for image classification under different resource constraints. Since then, many researchers have investigated multi-exit networks from different perspectives. Li et al. [32] explored methods to train multi-exit networks effectively and used gradient equalization algorithms to solve the learning conflicts among the exits and improved the learning speed of the exits. There has been research on combining KD with a multi-exit architecture. Scan [33] and Zhang et al. [34] prevented multiple classifiers of different depths from negatively affecting each other and enabled them to benefit from each other by adding KD and attention mechanisms. Phuong et al. [35] used
KD to encourage early exits to mimic the previous exit, thereby resulting in a new form of KD. Each exit in harmonized dense KD [36] learns from all other exits and uses adaptive weight learning to optimize multiple losses. Such studies ignore teacher guidance, which can reduce resource use but also misses out on more effective guidance.

In terms of the overall framework, hierarchical self-supervised augmented KD (HSAKD) [28] is similar to this study. HSAKD adds multiple auxiliary classifiers at different depths of the network to transfer different self-supervised augmented distributions extracted from the hierarchical features. In contrast, the purpose of constructing the teacher-student multi-exit architecture, construction methods, and knowledge transfer in this study are different from those in HSAKD. The purpose of constructing the multi-exit student network is to stimulate the learning ability of the student, to allow multiple stages of learning, and to use the attention mechanism to ensure a different focus for each stage of learning. These can ultimately achieve better ensemble learning. The purpose of constructing a teacher multi-exit network is to transfer knowledge comprehensively and allow the students to adapt to the current stage of learning. Additionally, HSAKD uses a complex knowledge definition, whereas we only use logit, which is a simple knowledge definition in the traditional approach.

3. Proposed Method

In this section, the basics of KD are briefly introduced and the proposed MSKD method is described. Finally, the OtO method is introduced.

3.1. KD

Given a dataset $D = \{(x_n, y_n)\}_{n=1}^{N}$ consisting of $N$ samples with $C$ classes, $y_n \in \{1 \ldots C\}$. The KD method uses the student network to learn the information delivered by the teacher network, thereby improving the performance of the student network. For ease of presentation, $t$ and $s$ are used to distinguish between the teacher and student networks, respectively. The temperature-softened network output is introduced in the conventional KD [14] to obtain supervised information. At this point, the probability of the network output for each category $j \in \{1 \ldots C\}$ is expressed as:

$$p(x, \tau) = \sigma(z_j / \tau) = \frac{e^{(z_j / \tau)}}{\sum_{i=1}^{C} e^{(z_i / \tau)}}$$ (1)

where $x$ denotes the input sample, $z_j$ is the output of the network before the softmax function [37] is applied, and $\sigma(\cdot)$ is the softmax function. $\tau$ is the hyper-parameter temperature, which controls the smoothness of the probability distribution of the output. The conventional KD method attempts to minimize the gap between the output of the teacher and student networks at the same temperature to allow the student network to mimic the teacher network. The distance between the teacher and student, i.e., the distillation loss, is denoted as:

$$\sum_{i=1}^{N} D_{KL}(p_t(x_i, \tau), p_s(x_i, \tau))$$ (2)

where $D_{KL}$ is the distance function Kullback–Leibler divergence [38] that measures the distance between the teacher and student networks. $p_t(x_i, \tau)$ and $p_s(x_i, \tau)$ denote the output of the teacher network and that of the student network after temperature smoothing, respectively. The distance between the student network and true label, i.e., the classification loss, is expressed as:

$$\sum_{i=1}^{N} L_{CE}(p_s(x_i, 1), y_i)$$ (3)
where \( L_{CE} \) denotes the cross-entropy function and \( y_i \) refers to the true label of this sample. The overall loss of the student network consists of the distillation and classification losses.

\[
Loss_{KD} = Loss_{cla} + Loss_{dis}
\]

\[
= \sum_{i=1}^{N} \left( (1 - \alpha)L_{CE}(p_s(x_i, 1), y_i) + \alpha \tau^2 D_{KL}(p_t(x_i, \tau), p_s(x_i, \tau)) \right)
\]

(4)

where \( \alpha \) is the hyper-parameter used to balance the two losses and since the order of magnitude of the classification loss gradient is approximately \( \tau^2 \) times that of the distillation loss gradient. The distillation loss is multiplied by \( \tau^2 \) to keep the contribution of the two loss gradients consistent.

### 3.2. MSKD

To make the student network available for multi-stage learning, we divided the general network into several parts according to different depths and set the corresponding early exits. This network with multi-exit architecture and \( K \) exits can be expressed as \( \{c_k(\cdot)\}_{k=1}^{K} \), where \( c_k(\cdot) \) denotes the \( k \)-th exit of the network. The loss of our proposed MSKD method possesses two main components: the classification loss \( (Loss_{cla}) \), i.e., the loss obtained by comparing the output of the student network with the true labels, and distillation loss \( (Loss_{dis}) \), i.e., the loss obtained by comparing the output of the student network with the output of the teacher network. The true labels are the actual category of the samples. The output of the teacher network refers to the probability distribution of the teacher network’s judgment on the sample category, and the output of the teacher network is softened by temperature to obtain a smoother probability distribution. The framework is shown in Figure 2 with a multi-exit student network learning true labels in multiple stages, as well as knowledge transferred by the teacher network. The loss corresponding to the \( k \)-th exit of the student network can be expressed as:

\[
Loss^{k}_{MSKD} = Loss^{k}_{cla} + Loss^{k}_{dis}
\]

\[
= \sum_{i=1}^{N} \left( (1 - \alpha)L_{CE}(p_s^k(x_i, 1), y_i) + \alpha \tau^2 D_{KL}(p_s^k(x_i, \tau), p_t(x_i, \tau)) \right)
\]

(5)

where \( p_s^k(x_i, 1) \) denotes the output of the \( k \)-th exit of the student network without temperature softening and \( p_s^k(x_i, \tau) \) represents the output of the \( k \)-th exit of the student network when the temperature is \( \tau \). At this point, the student network output is softened by the temperature to obtain a smoother probability distribution. The value of \( \tau \) ranges from greater than or equal to 1. For a student network with a multi-exit architecture, the overall loss is the combination of the loss at each exit, which can be expressed as:

\[
Loss_{MSKD} = \sum_{k=1}^{K} Loss^{k}_{MSKD}
\]

\[
= \sum_{k=1}^{K} \sum_{i=1}^{N} \left( (1 - \alpha)L_{CE}(p_s^k(x_i, 1), y_i) + \alpha \tau^2 D_{KL}(p_s^k(x_i, \tau), p_t(x_i, \tau)) \right)
\]

(6)
The pseudo-code for this method, which utilizes the standard gradient-based optimization method, is provided in Algorithm 1 below. During testing, we used an ensemble algorithm to combine the knowledge learned at each stage to constitute the final learning.

$$p_s(x) = \frac{1}{K} \sum_{k=1}^{K} p^k_s(x, 1)$$

**Algorithm 1: Multiple-Stage Knowledge Distillation**

```plaintext
Given: Dataset $D$, Model $t$, Multi-exit Model $s$

1. for $(x, y)$ in $D$ do
   2. $p_t = \text{Model}_t(x)$
      // The Model $s$ outputs include multiple exit outputs.
   3. $p_s = \text{Model}_s(x)$
      // Ensure that only students can learn from the teacher.
   4. $p_t_.detach()$
   5. for $k$ in $1 \ldots K$ do
      6. loss = loss + $(1 - \alpha)\text{CrossEntropy}(p^k_s(x, 1), y) + $
         $\alpha\tau^2 KLDivLoss\left(p^k_s(x, \tau), p_t(x, \tau)\right)$
   7. end
   8. loss.backward()
   9. update()
10. end
```

### 3.3. OtO Method

The multi-stage learning of the final output of the teacher network stimulates the learning potential of the student network; however, the performance improvement of the student network is limited because the teacher network only delivers the probabilistic information available at the final exit. The student network receiving information is a multi-exit branch network, and the gap between the learning ability of the pre-branch
student network and that of the teacher network is too large. This gap affects learning to a certain extent. Therefore, the teacher network is also constructed as a multi-exit architecture network and the teaching style is changed to one-to-one instruction such that the teacher network is targeted to guide the student network through different stages of learning. The overall framework of our proposed OtO method is shown in Figure 3, and the supervisory information consists of two parts: (1) the true label and (2) the branch network of teachers corresponding to that stage. The loss of the $k$-th exit of the student network in this method can be expressed as:

$$\text{Loss}_{k_{oto}} = \text{Loss}_{k_{cla}} + \text{Loss}_{k_{dis'}}$$

$$= \sum_{i=1}^{N} \left((1 - \alpha) L_{CE}\left(p_{k_{s}}^{k}(x_i, 1), y_i\right) + \alpha \tau^{2} D_{KL}\left(p_{k_{s}}^{k}(x_i, 1), p_{k_{t}}^{k}(x_i, \tau)\right)\right)$$

where $p_{k_{t}}^{k}(x_i, \tau)$ denotes the output of the probability distribution of the $k$-th exit of the teacher network after softening. As can be seen with the above description, the main difference between Equation (5) and Equation (8) is the source of guidance of the teacher network, one is the final output of the teacher network with the specific formula $p_{t}(x_i, \tau)$ in Equation (5) and the other corresponds to the output of the teacher branch network with the specific formula $p_{k_{t}}^{k}(x_i, \tau)$ in Equation (8). Therefore, the overall loss of the student network can be expressed as:

$$\text{Loss}_{oto} = \sum_{k=1}^{K} \text{Loss}_{k_{oto}}$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{N} \left((1 - \alpha) L_{CE}\left(p_{k_{s}}^{k}(x_i, 1), y_i\right) + \alpha \tau^{2} D_{KL}\left(p_{k_{s}}^{k}(x_i, 1), p_{k_{t}}^{k}(x_i, \tau)\right)\right)$$

Figure 3. Overall framework of the OtO method. Each of these exits has two types of losses: a distillation loss from the branch network corresponding to the teacher network and classification loss from the true label.
When each exit of the student network corresponds to an exit of the teacher network for one-to-one learning, the difference in the learning capacities between the teacher and student networks becomes small. We can then seek commonalities and differences in the training process with students sharing parameters in the backbone of the network and varying independent learning in the branches. Additionally, information is shared from different stages and relative differences are maintained. The same ensemble approach as that of MSKD is used to integrate multi-stage learning outcomes as the final learning outcome during testing.

4. Experiments

In this section, we first present the datasets used and the relevant setup of the experiment. The experimental data are subsequently reported in relation to the proposed method comparing the benchmark and KD methods.

4.1. Experimental Setup

Datasets. Relevant experiments were conducted on two standard classification datasets, namely CIFAR100 [12] and Tiny ImageNet [13]. CIFAR100 contains a total of 60,000 color images with a size of $32 \times 32$. Additionally, this dataset contains 100 categories with 600 images in each category. We followed the default split using 50,000 images for training and 10,000 images for testing the accuracy of the model. During training, random cropping and horizon flipping are applied to the data augmentation. In contrast, during testing, we do not perform any operations on the images. The Tiny ImageNet dataset contains RGB images of size $64 \times 64$ with 100,000 images in 200 classes for training and 5000 images each for validation and testing. During training, the images are randomly rotated for data augmentation, whereas no operations were performed on the images during testing. The details of the data set are shown in Table 1.

Table 1. The details of the dataset.

| Dataset          | Total | Category | Size     | Data Augmentation              |
|------------------|-------|----------|----------|-------------------------------|
| CIFAR100         | 60k   | 100      | $32 \times 32$ | random cropping and horizon flipping |
| Tiny ImageNet    | 120k  | 200      | $64 \times 64$ | random rotation               |

Optimization and hyper-parameters. Following previous research [26], we set the hyper-parameter temperature $\tau$ to 3 and $\alpha$ to 0.3. Each experiment was trained for 240 epochs with a batch size of 128. We use a stochastic gradient descent optimizer with a momentum of 0.9 and the weight decay was set to $5 \times 10^{-4}$. The initial learning rate was 0.1, which was then divided by 10 at the 150th, 180th, and 210th epochs. All the experiments were performed in PyTorch on GPU (RTX2080s) devices.

Model architecture. We selected widely used neural networks including residual network (ResNet) [1], visual geometry group (VGG) [3], and wide residual network (WRN) [39]. Additionally, we followed the most common structure of multi-exit networks constructed by dividing the above networks into four parts as the backbone network, i.e., by adding three early exit classifiers, as shown in Figures 2 and 3. We divide the network according to its own constituent blocks, for example, the ResNet family of networks adds early exits after the ResBlock. Additionally, the FC in the figure indicates the fully connected layer. To prevent the early exits from interfering with the performance of the final exit and ensure that the final exit can benefit from them, we added simplified attention modules [33] at each exit to enable each exit classifier to acquire the unique features of these modules. The attention module mainly consists of a convolutional layer and a deconvolutional layer. Additionally, mutually different exit classifiers ensure the effectiveness of ensemble learning.
4.2. Comparison with the Benchmark Method

To demonstrate the performance improvement of the proposed method over the network in the training approach of the benchmark method, we display the performance of our OTO method for CIFAR100 and Tiny ImageNet datasets in Tables 2 and 3, respectively. The benchmark method trains the student using standard cross-entropy. As shown in the two tables, the final accuracy obtained from testing the proposed OTO method is at least 4% higher than that of the benchmark method for CIFAR100 and Tiny ImageNet for a variety of teacher–student architectures. Selecting VGG13 and VGG8 [3] as the teacher and student networks, respectively, improves the performance of the student network by 5.91% for CIFAR100. ResNet54 and ResNet18 [1] are selected as the teacher and student networks, respectively, for Tiny ImageNet, and the student network performance is improved by 5.46%. In addition, the proposed method can outperform the benchmark method in the earlier exit. For example, when VGG13–VGG8 is used as the teacher–student architecture for CIFAR100, VGG8 outperforms the benchmark method at the first exit. This demonstrates that the proposed method can achieve better performance than that of the benchmark method. According to the accuracy of each exit branch network shown in the table, the performance of the later branch network is not necessarily better than that of the earlier branch network; however, the performance of each exit exhibits a gradually increasing trend. This is because the proposed method jointly optimizes multiple exit branch networks to improve each of them. The final result of the total approach is obtained using an ensemble to combine the knowledge learned in multiple exit branch networks.

Table 2. Accuracies (%) obtained for the CIFAR100 dataset. The first early exit that exceeds the performance of the benchmark method is marked in bold.

| Teacher | Baseline | Student | Baseline | Exit1 | Exit2 | Exit3 | Exit4 | Ensemble |
|---------|----------|---------|----------|-------|-------|-------|-------|----------|
| ResNet152 | 80.88 | ResNet50 | 79.32 | 80.80 | 81.78 | 81.74 | 83.20 |
| ResNet152 | 80.88 | ResNet18 | 78.47 | 77.40 | 80.15 | 80.50 | 81.73 |
| ResNet152 | 80.88 | ResNet10 | 76.19 | 75.12 | 77.52 | 78.02 | 79.74 |
| ResNet50 | 79.34 | ResNet18 | 78.47 | 77.62 | 80.09 | 80.65 | 81.42 |
| ResNet50 | 79.34 | VGG8 | 70.36 | 71.20 | 70.95 | 72.98 | 76.05 |
| VGG13 | 74.64 | VGG8 | 70.36 | **71.17** | 70.59 | 71.10 | 72.65 | 76.27 |

Table 3. Accuracies (%) obtained for the Tiny ImageNet dataset. The first early exit that exceeds the performance of the benchmark method is marked in bold.

| Teacher | Baseline | Student | Baseline | Exit1 | Exit2 | Exit3 | Exit4 | Ensemble |
|---------|----------|---------|----------|-------|-------|-------|-------|----------|
| ResNet34 | 66.40 | ResNet34 | 64.40 | 59.86 | 62.30 | 65.30 | 68.04 | 69.86 |

4.3. Comparison with Other KD Methods

To demonstrate the effectiveness and robustness of the proposed method, we compared it with other mainstream KD methods for the CIFAR100 and Tiny ImageNet datasets, and the experimental results are shown in Tables 4 and 5. All KD methods can improve network performance to some extent, as seen by the statistics presented in the tables. However, not all improved KD methods consistently outperform conventional KD methods under different teacher–student structures. The proposed method consistently outperforms the traditional KD methods under a variety of teacher–student architectures. Specifically, for a ResNet18 student trained by a ResNet50 teacher on CIFAR100, the proposed method results in a 2.7% improvement in the student network performance, and the accuracy of the student network is 1.83% higher than that of the teacher network. The OTO method, which is an extension of MSKD, further improves the accuracy, its performance is slightly worse than the SOTA distillation method HSAKD.
when the teacher network is ResNet152, but is excellent with the rest of the teacher–student architectures. The SOTA HSAKD [28] method augments data for self-supervised learning, and data augmentation aids its effectiveness. Furthermore, we used the AutoAugment [40] strategy in OtO, and Table 4 shows that the proposed method superior to the previous KD method at this time.

Table 4. Comparison of the accuracy (%) of our method with those of other KD methods for CIFAR100. The best and second-best results are shown in bold and underlined, respectively. “OtO∗” denotes the results after data augmentation.

| Teacher  | ResNet152 | ResNet128 | ResNet152 | ResNet50 | ResNet50 | VGG13 | WRN40-2 |
|----------|-----------|-----------|-----------|----------|----------|-------|---------|
| ResNet50 | 79.69     | 79.86     | 77.85     | 79.72    | 73.81    | 72.98 | 74.92   |
| FitNet   | 80.51     | 79.24     | 78.02     | 76.27    | 73.24    | 72.22 | 73.07   |
| AT       | 80.41     | 80.19     | 78.45     | 78.28    | 74.01    | 73.48 | 73.97   |
| SP       | 80.72     | 79.87     | 78.25     | 79.32    | 73.52    | 73.49 | 74.34   |
| CC       | 79.89     | 79.82     | 78.06     | 77.42    | 73.48    | 73.04 | 73.18   |
| VID      | 79.24     | 79.67     | 77.80     | 76.79    | 73.46    | 72.97 | 74.07   |
| RKD      | 80.22     | 79.60     | 77.90     | 78.75    | 73.51    | 73.19 | 73.36   |
| PKT      | 80.57     | 79.44     | 78.41     | 78.89    | 73.61    | 73.25 | 74.42   |
| AB       | 81.21     | 79.50     | 78.18     | 76.77    | 73.65    | 73.35 | 72.65   |
| FT       | 80.37     | 79.26     | 77.53     | 76.29    | 72.98    | 73.44 | 73.30   |
| CRD      | 80.53     | 79.81     | 78.60     | 74.58    | 74.58    | 74.29 | 75.79   |
| SSKD     | 80.29     | 80.36     | 78.60     | 79.72    | 75.36    | 75.27 | 76.14   |
| HSAKD    | 83.33     | 82.17     | 79.75     | 81.22    | 75.20    | 75.07 | 78.20   |
| TSKD     | 83.09     | 81.27     | 79.34     | 81.28    | 75.63    | 75.42 | 78.31   |

5. Analysis

In this section, we first explore the validity of each part of the proposed method using ablation experiments. Subsequently, we explore the validity of the ensemble in the proposed method.
5.1. Ablation Study

To verify the effectiveness of each part of our OtO method, several teacher–student architectures are selected to conduct an ablation study for CIFAR100. Overall, we explored the following: (i) a multi-exit architecture network (Mu) using only classification loss, i.e., setting $\alpha$ in Equation (5) to 0; (ii) applying a conventional KD method on a multi-exit architecture network (Mu+KD), i.e., only the final exit of the multiple-exit student network architecture receives guidance from the original teacher network; (iii) OtO and the use of ensemble learning in the above three training modalities. The experimental results are shown in Figure 4. As shown in the figure, the multi-exit architecture student network with only one last exit to receive guidance from the teacher network improves the accuracies of the single last exit and ensemble learning. This illustrates the effectiveness of KD applied to multi-exit architectures. Furthermore, our proposed method is substantially better than the original multi-exit architecture network and conventional KD method, which indicates that a multi-exit architecture teacher–student network for one-on-one instruction stimulates the learning ability of the student network and results in a greater amount of knowledge. As shown in Figure 4b, in several teacher–student architectures, the accuracy is eventually improved by the ensemble, which illustrates the practical relevance of the ensemble for combining multiple stages. In conclusion, our proposed method can achieve better results using the ensemble.

![Figure 4. Cont.](image-url)
Figure 4. Ablation study of multiple teacher–student architectures for CIFAR100. The horizontal coordinate refers to the teacher–student architectures. (a) Accuracy of the final exit corresponding to different training methods. (b) Accuracy after the ensemble of multiple exits corresponding to different training methods.

5.2. Ensemble Effectiveness of the Proposed Method

Here we will discuss the effectiveness of ensemble learning in proposing methods. In our approach, we first construct the student network as a multi-exit architecture network and add early classifiers to different depths of the neural network to form a multi-exit network. The multi-exit architecture network jointly optimizes the loss of multiple exits, each of which is equivalent to a regularization term for the remaining exits, which prevents overfitting and improves accuracy. Additionally, the early exit can reduce the tendency of gradient disappearance [50]. The student model is divided into multiple stages sharing the first few layers of the network structure, which can use the same low-level features, isolate unique features at their respective exits, and then use individual branches for higher-level features to achieve better classification results. To ensure that the ensemble is effective, we must preserve as much branch diversity as possible. Firstly, we added multiple exits at different depths when constructing a multi-exit architecture student network, so that each exit uses a different depth of the shared backbone. Secondly, we used a simplified attention module at the diversion in order to make that exit acquire unique features, and finally, we constructed branching networks at different depths according to the depth of the shared backbone of the exits. This set of measures allows each branch network to acquire different features and therefore different classification results, and the experimental data are shown in Tables 2 and 3. As can be seen from the data in the table, the classification accuracy of each branch network varies, showing a gradual increase in general, and the accuracy can be further improved after the final ensemble. For instance, when VGG13 is the teacher network and VGG8 is the student network in Table 2, the final integrated accuracy is improved by 3.62% compared to the final exit accuracy.

Relevant experiments were conducted to illustrate the differences in the classification performance of each exit. Herein, we choose ResNet34 and ResNet18 as the teacher and student network, respectively, for the Tiny ImageNet dataset and explore the samples that
are correctly classified for each exit. The results reveal that the samples classified correctly in each exit are not identical, and there are cases in which samples classified correctly at the earlier exit are misclassified at the later exit. Figure 5 shows a set of samples that are correctly classified at the first exit only. Grad-CAM [51] can be used to visualize the importance of this set of samples at the first exit versus that at the last exit in the prediction process for each position of the image for the corresponding class. Notably, the first and last exits are concerned with the general structure of the input image and image details, respectively. The candle sample in Figure 5 is an example in which the first exit focuses only on the simple structure of the candle to arrive at the correct classification. In contrast, the final exit focuses on the image details. More specifically, the final exit only focuses on the local information of the candle, which results in the classifier mistaking the image for that of a jellyfish and finally misclassifying it. The different exits of the student network focus on different aspects of the samples, which results in different classifications. Furthermore, variabilities exist among the multiple exits of a multi-exit architecture student network and the validity of the ensemble is ensured.

| The original image | Exit1 (Correct) | Exit4 (Wrong) | The original image | Exit1 (Correct) | Exit4 (Wrong) |
|--------------------|-----------------|---------------|--------------------|-----------------|---------------|
| Pole               | Pole            | Oboe          | Barbershop         | Barbershop      | Oboe          |
| Tailed frog        | Tailed frog     | Tabby         | Candle             | Candle          | Jellyfish     |

Figure 5. A set of Tiny ImageNet samples with only the first exit classified correctly in the case of a teacher–student architecture of ResNet34–ResNet18, and heat maps drawn with Grad-CAM. Below the original image column is the true label for that sample, and below each sample in each exit column is the predicted category label for that exit.

5.3. Exploratory Experiments

In the multi-teacher KD method in the KD approach [6], the student network can receive guidance from multiple teachers simultaneously. We added a final exit guide of the teacher network to the OtO method, i.e., the early exit of the student network had two teachers: one at the corresponding exit and another at the final exit. This was equivalent to combining the MSKD and OtO methods. Herein, we used a simple average teacher output to instruct the student network, and the results of the experiment are shown in Figure 6. Here, we chose two teacher–student architectures for our experiments and compare Mu+KD, MSKD, OtO, and the combination of multiple-stage KD and OtO (OtO+last). To visualize the differences in the methods, the framework diagram of the above methods is displayed in Figure 7. From the experimental results in Figure 6, the classification accuracy of MU+KD and MSKD is relatively low, OtO+last is observed to possess the best accuracy at the first exit; however, its performance does not exceed that of OtO at the later exits, and classification accuracy is still the best for OtO method after the final ensemble. Additionally, the accuracies of the early exits improve when guided by multiple teachers; however, when the learning goals of multiple exits tend to converge, the diversity among exits reduces and results in a poor ensemble. Therefore, to achieve the best final ensemble, a balance of
accuracy and variability must be maintained across multiple exits. Of the four methods compared, the OtO method has the best balance of accuracy and variability.

Figure 6. Comparison of the accuracies of different methods for multiple exits on CIFAR100.

Figure 7. The framework diagram of each method. The blue line indicates the supervision of the true label, the green line indicates the supervision of the teacher network, and the red line indicates the supervision of the early exit of the teacher network. (a–d) denote the four training methods mentioned in the context, respectively.

6. Conclusions

Inspired by the learning process of humans, this study divides the learning process of a DNN into multiple stages to stimulate the learning ability of students for improved learning. Herein, we propose an MSKD method in which the student network is constructed as a multi-exit architecture network that mimics the output of the teacher network at different
depths. Additionally, an ensemble is used to combine the results of the multiple stages of learning to form the final output. To match the learning capacities of the teacher and student networks, the teacher network also possesses a multi-exit architecture, thereby allowing the teacher branch network to provide one-to-one guidance to the student branch network. This results in the formation of the OtO method. The proposed OtO method is at least 4% more accurate than the benchmark method for a variety of teacher–student architectures, and can achieve a high level compared with other mainstream distillation methods. Numerous experimental results revealed that the proposed method can stimulate the learning potential of the student by changing their learning styles, which provides a new perspective on KD.

In this study, we only used logits as the knowledge being transferred. We can be considered to introduce the definition of knowledge from other studies or will be able to achieve a better distillation effect. Furthermore, we treated all the stages of learning equally. In the future, we may consider highlighting certain learning stages using learning adaptive weights.

**Author Contributions:** Conceptualization, C.X. and N.B.; methodology, C.X. and N.B.; software, G.L.; validation, W.G., T.L. and M.L.; formal analysis, W.G.; investigation, M.L.; resources, G.L.; data curation, T.L.; writing—original draft preparation, N.B.; writing—review and editing, C.X. and W.G.; visualization, N.B.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, C.X. and G.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the China Chongqing Science and Technology Commission under Grants cstc2020jscx-msxmX0086, cstc2019jscx-zdztzx0043, cstc2019jcyj-msxmX0442, and stc2021jcyj-msxmX0605; China Chongqing Banan District Science and Technology Commission project under Grant 2020QC413; China Chongqing Municipal Education Commission under Grant KJQN202001137.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
3. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
4. Aggarwal, K.; Mijwil, M.M.; Al-Mistarehi, A.H.; Alomari, S.; Gök, M.; Alaabdin, A.M.Z.; Abdulrhman, S.H. Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning. *IRAQI J. Comput. Sci. Math.* 2022, 3, 115–123.
5. Komodakis, N.; Zagoruyko, S. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
6. Liu, Y.; Zhang, W.; Wang, J. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing* 2020, 415, 106–113. [CrossRef]
7. Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; Guo, C. Knowledge distillation from internal representations. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7350–7357.
8. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* 2021, 129, 1789–1819. [CrossRef]
9. Wang, L.; Yoon, K.J. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2021, 44, 3048–3068. [CrossRef] [PubMed]
10. Yuan, L.; Tay, F.E.; Li, G.; Wang, T.; Feng, J. Revisiting knowledge distillation via label smoothing regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3903–3911.
11. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5191–5198.

12. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Master’s Thesis, University of Tront, Toronto, ON, Canada, 2009.

13. Deng, J. A large-scale hierarchical image database. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.

14. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. Stat 2015, 1050, 9.

15. Cheng, X.; Rao, Z.; Chen, Y.; Zhang, Q. Explaining knowledge distillation by quantifying the knowledge. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 12925–12935.

16. Mobahi, H.; Farajtabar, M.; Bartlett, P. Self-distillation amplifies regularization in hilbert space. Adv. Neural Inf. Process. Syst. 2020, 33, 3351–3361.

17. Phuong, M.; Lampert, C. Towards understanding knowledge distillation. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5142–5151.

18. Allen-Zhu, Z.; Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv 2020, arXiv:2012.09816.

19. Fukuda, T.; Suzuki, M.; Kurata, G.; Thomas, S.; Cui, J.; Ramabhadran, B. Efficient Knowledge Distillation from an Ensemble of Teachers. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 3697–3701.

20. Beyer, L.; Zhai, X.; Royer, A.; Markeeva, L.; Anil, R.; Kolesnikov, A. Knowledge distillation: A good teacher is patient and consistent. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–23 June 2022; pp. 10925–10934.

21. Li, J.; Guo, J.; Sun, X.; Li, C.; Meng, L. A Fast Identification Method of Gunshot Types Based on Knowledge Distillation. Appl. Sci. 2022, 12, 5526. doi: 10.3390/app1215526. [CrossRef]

22. Chen, L.; Ren, J.; Mao, X.; Zhao, Q. Electroglottograph-Based Speech Emotion Recognition via Cross-Modal Distillation. Appl. Sci. 2022, 12, 4338. doi: 10.3390/app12094338. [CrossRef]

23. Passalis, N.; Tefas, A. Learning deep representations with probabilistic knowledge transfer. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 268–284.

24. Liu, L.; Huang, Q.; Lin, S.; Xie, H.; Wang, B.; Chang, X.; Liang, X. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8271–8280.

25. Wen, T.; Lei, S.; Qian, X. Preparing lessons: Improve knowledge distillation with better supervision. Neurocomputing 2021, 454, 25–33. [CrossRef]

26. Tian, Y.; Krishnan, D.; Isola, P. Contrastive Representation Distillation. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

27. Xu, G.; Liu, Z.; Li, X.; Loy, C.C. Knowledge distillation meets self-supervision. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin, Germany, 2020; pp. 588–604.

28. Yang, C.; An, Z.; Cai, L.; Xu, Y. Hierarchical Self-supervised Augmented Knowledge Distillation. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 19–27 August 2021; pp. 1217–1223.

29. Fu, S.; Li, Z.; Liu, Z.; Yang, X. Interactive knowledge distillation for image classification. Neurocomputing 2021, 449, 411–421. [CrossRef]

30. Teerapittayanon, S.; McDanel, B.; Kung, H.T. Branchynet: Fast inference via early exiting from deep neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2464–2469.

31. Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; Weinberger, K. Multi-Scale Dense Networks for Resource Efficient Image Classification. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

32. Li, H.; Zhang, H.; Qi, X.; Yang, R.; Huang, G. Improved techniques for training adaptive deep networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1891–1900.

33. Zhang, L.; Tan, Z.; Song, J.; Chen, J.; Bao, C.; Ma, K. Scan: A scalable neural networks framework towards compact and efficient models. Adv. Neural Inf. Process. Syst. 2019, 32, 1–10.

34. Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; Ma, K. Be your own teacher: Improve the performance of convoludional neural networks via self distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 3713–3722.

35. Phuong, M.; Lampert, C.H. Distillation-based training for multi-exit architectures. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1355–1364.

36. Wang, X.; Li, Y. Harmonized dense knowledge distillation training for multi-exit architectures. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 10–218.
37. Duan, K.; Keerthi, S.S.; Chu, W.; Shevade, S.K.; Poo, A.N. Multi-category classification by soft-max combination of binary classifiers. In Proceedings of the International Workshop on Multiple Classifier Systems, Guilford, UK, 11–13 June 2003; Springer: Berlin, Germany, 2003; pp. 125–134.

38. Johnson, D.; Sinanovic, S. Symmetrizing the kullback-leibler distance. IEEE Trans. Inf. Theory 2001, 57, 5455–5466.

39. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016; British Machine Vision Association: York, UK, 2016.

40. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. arXiv 2018, arXiv:1805.09501.

41. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. arXiv 2015, arXiv:1412.6550.

42. Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1365–1374.

43. Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; Zhang, Z. Correlation congruence for knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 5007–5016.

44. Ahn, S.; Hu, S.X.; Damianou, A.; Lawrence, N.D.; Dai, Z. Variational information distillation for knowledge transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9163–9171.

45. Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3967–3976.

46. Heo, B.; Lee, M.; Yun, S.; Choi, J.Y. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3779–3787.

47. Kim, J.; Park, S.; Kwak, N. Paraphrasing complex network: Network compression via factor transfer. Adv. Neural Inf. Process. Syst. 2018, 31, 2760–2769.

48. Xu, C.; Gao, W.; Li, T.; Bai, N.; Li, G.; Zhang, Y. Teacher-student collaborative knowledge distillation for image classification. Appl. Intell. 2022, 1–13. [CrossRef]

49. Ji, M.; Heo, B.; Park, S. Show, attend and distill: Knowledge distillation via attention-based feature matching. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 7945–7952.

50. Scardapane, S.; Scarpiniti, M.; Baccarelli, E.; Uncini, A. Why should we add early exits to neural networks? Cogn. Comput. 2020, 12, 954–966. [CrossRef]

51. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.