Complex & Intelligent Systems (2023) 9:5779–5791
https://doi.org/10.1007/s40747-023-01036-0

ORIGINAL ARTICLE

Progressive multi-level distillation learning for pruning network

Ruiqing Wang[1] · Shengmin Wan[1] · Wu Zhang[1,2] · Chenlu Zhang[1] · Yu Li[1] · Shaoxiang Xu[1] · Lifu Zhang[1] · Xiu Jin[1,2] · Zhaohui Jiang[1,2] · Yuan Rao[1,2]

Received: 27 February 2022 / Accepted: 9 March 2023 / Published online: 7 April 2023

Abstract
Although the classification method based on the deep neural network has achieved excellent results in classification tasks, it is difficult to apply to real-time scenarios because of high memory footprints and prohibitive inference times. Compared to unstructured pruning, structured pruning techniques can reduce the computation cost of the model runtime more effectively, but inevitably reduces the precision of the model. Traditional methods use fine tuning to restore model damage performance. However, there is still a large gap between the pruned model and the original one. In this paper, we use progressive multi-level distillation learning to compensate for the loss caused by pruning. Pre-pruning and post-pruning networks serve as the teacher and student networks. The proposed approach utilizes the complementary properties of structured pruning and knowledge distillation, which allows the pruned network to learn the intermediate and output representations of the teacher network, thus reducing the influence of the model subject to pruning. Experiments demonstrate that our approach performs better on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets with different pruning rates. For instance, GoogLeNet can achieve near lossless pruning on the CIFAR-10 dataset with 60% pruning. Moreover, this paper also proves that using the proposed distillation learning method during the pruning process achieves more significant performance gains than after completing the pruning.

Keywords Deep neural network · Model compression · Network pruning · Knowledge distillation

Introduction

Deep neural networks (DNNs) based on deep learning have shown impressive results on tasks such as image classification [1–3], object detection [4–6], and natural language processing [7]. With the development of network models, it seems to be a new trend to build more sophisticated networks to achieve higher accuracy [8, 9]. These large, complex networks, however, do not work effectively on mobile devices or Internet of Things devices. Therefore, alleviating the model's operational burdens while ensuring high accuracy is one of the main problems facing DNNs. Pruning, knowledge distillation, quantization, and lightweight networks have developed into available ways needed to reduce the considerable computational resources required [10].

Pruning methods [11] allow the model to be simpler and more efficient by eliminating redundant parameters or connections through a certain measure, which is why pruning is a popular technique. Although this technique can minimize the size of the model while maintaining performance, precision loss is unavoidable in pruning networks. The goal of knowledge distillation is to guide student learning through a more robust teacher model, which enables more straightforward learners to have a certain degree of mastery over the teacher's skills [12]. This, however, requires the researchers to manually select models for both teachers and students.

Knowledge distillation is an effective way to compensate for the loss of precision due to pruning. However, using knowledge distillation only after the pruning has been completed, and not while it is in progress, may result in suboptimal model performance. In addition, most of the previous studies have focused on the problem of how to improve the performance of unstructured pruning [13, 14], while there

✉ Wu Zhang
zhangwu@ahau.edu.cn

1 School of Information and Computer, Anhui Agricultural University, 130 Changjiang West Road, Shushan District, Hefei, Anhui, China

2 Anhui Province Key Laboratory of Smart Agricultural Technology and Equipment, Anhui Agriculture University, Hefei, Anhui, China

have been few studies on structured pruning [15, 16]. In fact, non-structured pruning needs special software libraries or hardware to speed up the network model, whereas structured pruning can compress the network without any help [17]. Therefore, it is more realistic to combine structured pruning with distillation learning.

To solve the image classification problem, we present a new method of progressive multi-level distillation for structural pruning. In this paper, the original and pruned networks can be considered teacher and student models, thus avoiding the need for manual selection of teacher models. Moreover, we take full advantage of the characteristics of structured pruning, using each block of student network pruning and its corresponding teacher block as input for distillation loss based on feature representation. Its respective blocks gradually increase as pruning progresses, forming a progressive distillation. In addition to feature learning, our proposed multi-level distillation learning includes response representation-based learning, which allows students to mimic the logits output of the teacher's model. In this way, our approach can effectively reduce accuracy losses, allowing the pruned network to minimize the size of the model and the computational resources required within an acceptable range of accuracy degradation.

The contributions of this paper are as follows.

1. This paper proposes a progressive multi-level distillation learning approach for structured pruning networks. We also validate the proposed method on different pruning rates, pruning methods, network models, and three public datasets (CIFAR-10/100, and Tiny-ImageNet).
2. Compared with other knowledge distillation methods, our proposed method can better restore the structured pruning network's accuracy and improve the model's performance after each pruning.
3. We conduct ablation study experiments further to understand each loss's contribution to our proposed framework.
4. We show that distillation learning during pruning, rather than after pruning, improves model performance without additional inference time.

Related work

Network pruning

In earlier studies on pruning, the focus was more on the granularity of the pruning of individual neurons, i.e., unstructured pruning. Optimal Brain Damage [18] and Optimal Brain Surgeon [19] assessed the significance of weights on the basis of information related to the second-order derivatives of the loss function. More directly, Han et al. [20] determined whether the parameters were significant (insignificant) depending on whether they were larger (less) than a given threshold. While leading to high compression ratios, these methods only changed the weight matrix from dense to sparse. Unstructured pruning would not yield the expected results without specialized software libraries or hardware to help calculate [17].

On the other hand, the pruning granularity of structured pruning is an integrated structure. For example, Li et al. [21] ranked the filters of each layer according to the sum of the absolute filter weights (i.e., L1-norm) to determine their importance. Zhuang et al. [22] considered sparse filters non-critical, and removing unimportant filters by imposing a scaling factor on the Batch Normalization (BN) was also an efficient approach [23]. In a recent study, Lin et al. [24] concluded that the rank of the feature map is more representative of the amount of information contained in a filter, which can lead to promising results.

Knowledge distillation

The initial knowledge distillation [25] argued that one-hot labels limit the performance of the network model, and that the soft labels of a more robust network would provide more abundant information, which would allow the transfer of knowledge from a larger teacher network to a smaller one, thereby bridging the gap. Moreover, besides focusing on extracting logits output knowledge, intermediate representations of knowledge within the teacher in the form of feature maps can also be learned by the student model. FitNet [26] first proposed distillation learning for a single intermediate layer of knowledge. AT [27] extended this idea by extracting multiple intermediate layers knowledge of the teacher model to guide student learning, and by using L2-regularization on each feature map to ensure consistent dimensions for each pair of feature maps. However, knowledge from deeper intermediate layers may provide students with overly standardized guidance, while knowledge from shallower layers may not serve as a guiding role [12], which results in the inefficient transfer of knowledge. In relation-based distillation learning, knowledge transfer relationships between different layers or data are further explored. Yim et al. [28] used the relationship between layers of the teacher's network as the goal of student model learning. SP [29] aimed to preserve the student's pairwise similarity rather than mimicking the teacher's representation space, so that students could better understand the relationships between instances. Furthermore, in addition to the applications mentioned above in classification tasks, knowledge distillation methods have also proven their effectiveness in more complex tasks such as object detection [30, 31].

Model pruning and knowledge distillation are two independent parts of model compression. How to combine these two methods is one of the problems worth discussing. The simplest way to combine them is to use knowledge distillation after the completion of pruning [15, 32]. However, we have shown that the use of distillation learning in the fine-tuning process of pruning can yield better results, as demonstrated in "Two combined strategies". Furthermore, it is also necessary to validate the efficacy of distillation learning for structured pruning networks on various model architectures and public datasets.

Quantization

The memory footprints and inference speed of the model can be effectively decreased by reducing the number of representation bits of original weights. This technique is known as quantization. Gong et al. [33] quantizing of the weights using K-means clustering could compress the network model by a factor of 8–16 with minimal or no performance impairment. In addition, under exceptional cases, weights could be represented as one-bit data and constituted a binarized network [34], significantly reducing computational consumption. Han et al. [35] integrated pruning, quantization, and Hoffman coding for deep model compression, providing a solution for its deployment on devices with low energy consumption.

The proposed method

Figure 1 gives an overview of the progressive multi-level distillation learning approach for structured pruning. In the process of structured pruning, the original network and the pruned network are treated as a teacher and student model, respectively, and the proposed method is used in the fine-tuning process. In contrast to using knowledge distillation only after pruning is completed, our approach increases the training time but improves the performance of the model. Although the structure of the network model (i.e., the number of channels) is constantly changed with pruning, it has been shown that we can improve the performance after every pruning without the need to adjust the hyperparameters. The algorithm flow is illustrated in Algorithm 1. The proposed approach will be described in more detail in the following sections.

Algorithm 1 Using Progressive Multi-level Knowledge Distillation During Pruning.

Input: well-trained teacher T, numbers of pruning layers L, fine-tuning epochs N after pruning, pruning rate $\{r_l\}_{l=1:L}$.

Output: pruned model S

\# Initialization

$S \leftarrow T$

for $l = 1, 2, ..., L$ **do**

  Remove $r_l\%$ filters of the layer l in S by L1-norm [21] or Hrank [24] method

  for $n = 1, 2, ..., N$ **do**

    Fine-tuning model S by a progressive multi-level distillation loss (Eq. (7)) with T

  end for

end for

return the pruned model S

Progressive feature distillation

As mentioned in "Knowledge distillation", the intermediate knowledge from deep layers can easily lead to over-normalization of the students' models, and the intermediate knowledge from shallow layers will not be able to provide guidance. Therefore, effectively transferring the knowledge of teachers' models to students is a critical issue. As shown in Fig. 2, unlike FitNet [26] and AT [27] for distillation learning of fixed intermediate blocks of knowledge, we subtle used the characteristic of structured pruning in which each block is pruned in turn, so that each block that is pruned becomes a mentee. The corresponding unpruned block in the teacher model becomes a mentor. Although there is a significant deviation between the pruned block and the original one, the corresponding feature pairs can effectively transfer intermediate knowledge to achieve better performance recovery. As illustrated in Fig. 1, when pruning begins, the number of pruned blocks is small, and only shallow, intermediate knowledge can be used as a guide. But as the number of pruned blocks increases, the corresponding loss of information increases, so that the deep intermediate knowledge becomes useful, avoiding the over-standard of the student model and compensating for the loss of representation power caused by pruning.

In the pruning of the student model, the structured pruning removes the non-significant channels, which leads to a discrepancy in the number of channels between the two models. Using an adaptation layer consisting of a pointwise convolution (1×1 kernel) and a BN layer, we map the student channels to their corresponding teacher counterparts, allowing for more efficient knowledge extraction and reducing differences in feature maps between the pruned and the original model. We present the distillation losses of individual

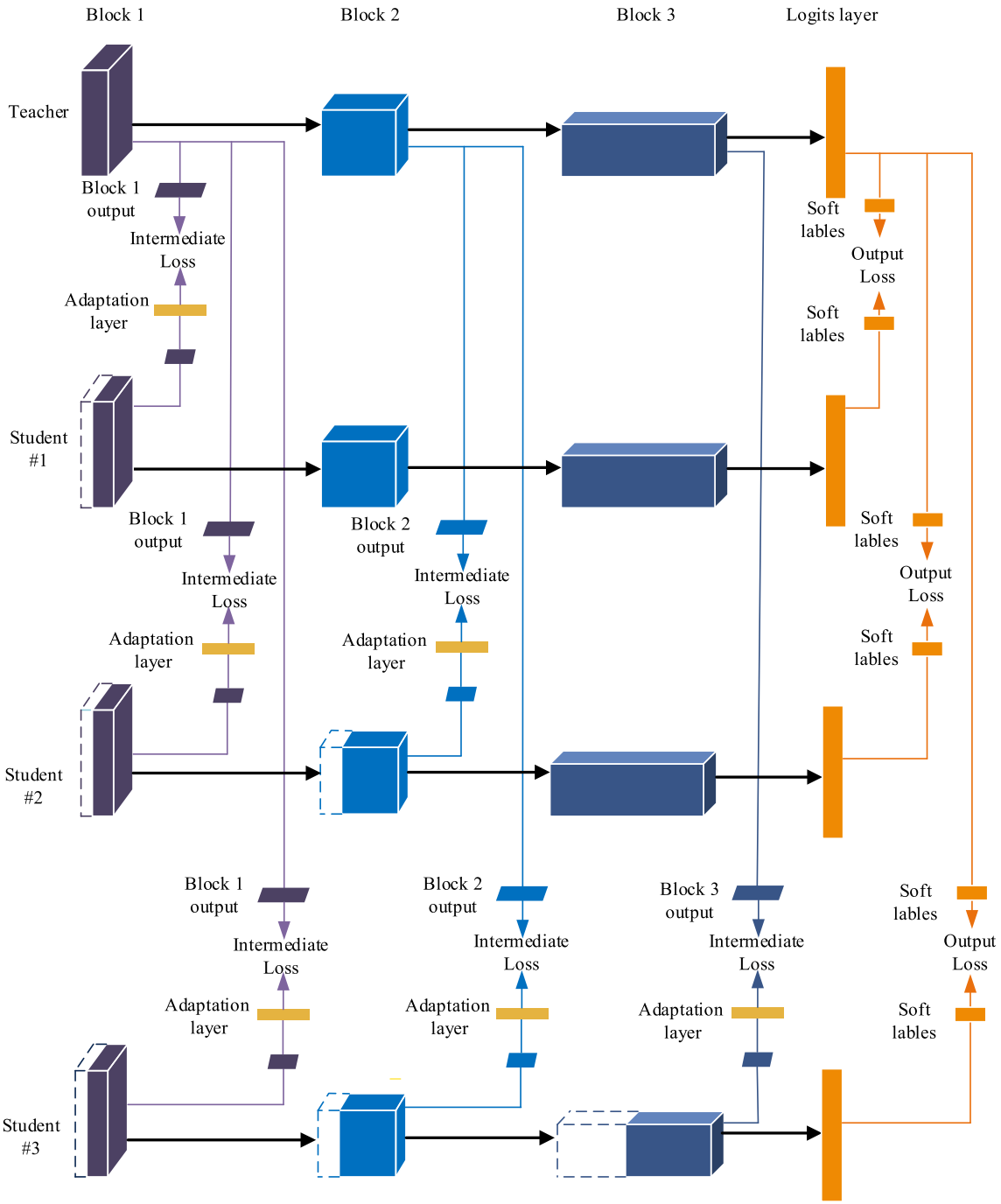

Fig. 1 An overview of the implementation of this method, which is based on a progressive multi-level distillation method for structured pruned networks. In the figure, the network is divided into three blocks, and we take the network after pruning each block as a student and the well-trained network as a teacher. Students #1 to #3 represent the student models obtained after sequential pruning of the first module of the teacher model to the third model. As pruning progresses, the intermediate features of the extracted knowledge are increased, which can maximize the utilization of pruning properties for distillation learning. The adaptation layer makes the feature mapping dimension of the student block the same as that of the teacher block. Note that a block can contain more than one convolution layer and block

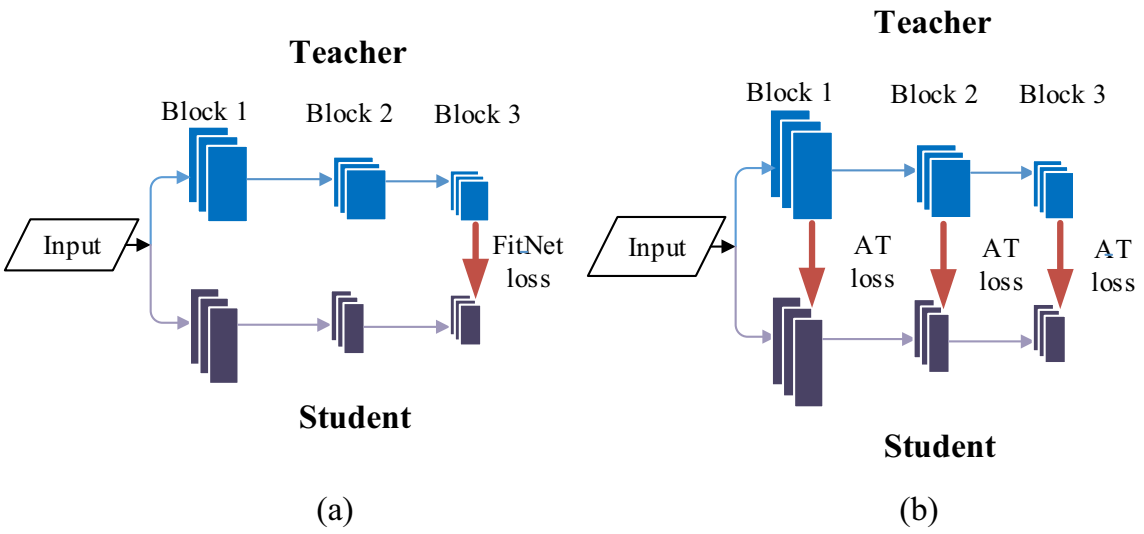

Fig. 2 **a** and **b** denote the extraction of the single and all intermediate features by FitNet [26] and AT [27], respectively

blocks as follows,

$$l_{block}^{intermediate} = D_p(\boldsymbol{F}_t, r(\boldsymbol{F}_s)), \tag{1}$$

in which $\boldsymbol{F}_s$ is denoted as a feature map of the student model and $\boldsymbol{F}_t$ is denoted as a feature map of its corresponding teacher model. $r(\cdot)$ is a regressor consisting of a 1×1 convolutional layer and a BN layer. D_p is a measure of the L_2 distance between student and teacher feature maps. The overall distillation loss based on feature representation can be expressed as follows,

$$L^{\text{intermediate}} = \sum_{b=1}^{B} l_b^{\text{intermediate}}, \tag{2}$$

where B is the number of pruned blocks. This loss makes it possible for the student model to learn the features of the teacher model efficiently during the structure pruning process.

Output logits distillation learning

Multi-level distillation learning has been shown to perform better than single knowledge distillation methods for image classification [36] and object detection [37]. Therefore, we extend this concept to the pruning process in a reasonable manner. Apart from the feature representation-based knowledge distillation described above, our approach also includes output logits mimicking distillation learning. It is also necessary to mimic the softened teacher outputs in order to learn more from the teacher model. We use the Kullback–Leibler Divergence loss between the student and teacher outputs as the distillation loss for output imitation. The temperature τ softens the outputs between each pair of students and teachers. This method enables the student model to learn the predictions of the high-performance teacher model more efficiently, which can significantly reduce the classification error rate. The softened softmax function and the overall output imitation loss are shown below,

$$X_{ij} = \frac{\exp(x_{ij}/\tau)}{\sum_{j=1}^{C} \exp(x_{ij}/\tau)}, \tag{3}$$

$$L^{output} = \sum_{i=1}^{S} \sum_{j=1}^{C} X_{ij}^{T} \log\left(X_{ij}^{T}/X_{ij}\right), \tag{4}$$

where x_{ij} represents the student single output logit for the j^{th} class of the i^{th} batch sample. X_{ij} and X_{ij}^{T} represent the softened softmax output of the student model and the teacher model for the j^{th} class of the i^{th} batch sample, respectively. The temperature hyperparameter T determines the softening degree of output. X_{ij}^{T} can also be calculated by Eq. (3).

Total loss

In addition to the feature and output imitation learning described above, each student model is trained in a classical cross-entropy function with ground-truth labels and student output logits, which aids the model to learn better about a given dataset, as shown in the following equation,

$$\widetilde{X_{ij}} = \frac{\exp(x_{ij})}{\sum_{j=1}^{C} \exp(x_{ij})}, \tag{5}$$

$$L^{CE} = \sum_{i=1}^{S} \sum_{j=1}^{C} -Y_{ij} \log(\widetilde{X_{ij}}), \tag{6}$$

where $\widetilde{X_{ij}}$ represents the student softmax output for the j^{th} class of the i^{th} batch sample. Y_{ij} denotes the ground-truth label for the j^{th} class of the i^{th} batch sample.

Our proposed progressive multi-level distillation learning is a weighted combination of these three losses mentioned above, updating the parameters of the student network only during the training phase to allow better accuracy recovery of the pruned model, which is mathematically represented as follows,

$$L = \alpha L^{\text{intermediate}} + \beta L^{\text{output}} + \gamma L^{CE}. \tag{7}$$

We find the optimal values of weights by grid search can be taken at $\alpha = 0.25$, $\beta = 0.1$, and $\gamma = 0.9$, and use these hyperparameters in all the subsequent experiments. Note that the proposed method does not increase the inference time of the model, and it is orthogonal to techniques such as quantization.

Experiments

The effectiveness of this method is evaluated by comparing it with the existing methods. See "Implementation details" for details of implementation. In "Main results", the superiority of our approach is demonstrated in publicly available datasets. The effects of the ablation experiments and different combination strategies will be discussed in later sections.

Implementation details

We perform L1-norm [21] pruning and HRank [24] pruning for VGGnet-16 [38] and ResNet-56 [39], GoogleNet [40]. The location of our selected feature distillation blocks is shown in Fig. 3. In addition, to enable a more comprehensive assessment of the usability of the proposed methods, we also validate it under different layer pruning rates: 60%, 70%, and the appropriate pruning rate (APR) given by HRank, as shown in Table 1. All experiments are performed using Pytorch and on an NVIDIA GeForce GTX 1080Ti GPU. The resource costs of the model at various pruning rates in CIFAR-10 are shown in Table 2.

In order to demonstrate the effectiveness of the proposed approach, we compare it with the following representative approaches. Baseline is the result of pruning without the use of a distillation method. Details are as follows.

(a) KD [25]: Makes use of KL divergence to close the softmax output of teacher and student, so as to transfer the knowledge and reduce the classification error of the student model.
(b) FitNet [26]: It extracts the knowledge of a single intermediate layer of the well-trained teacher network, and uses it to guide the students' study. Knowledge distillation is accomplished by optimizing the distance between student and teacher intermediate layer features.
(c) AT [27]: Improves student network performance by transferring the attention map of the teacher network so that the student can learn more useful information.
(d) SP [29]: It uses pairwise activation similarity in each mini-batch to train students. Thus, it is possible to encourage student models to maintain pairwise similarity in their representation space without mimicking the teacher's representation space.

Main results

CIFAR-10/100

CIFAR-10 [41] has 50,000 training and 10,000 test images divided into 10 classes. CIFAR-100 [41] has the same number of training and test set images as CIFAR-10; the difference is that these images are classified into 100 categories. To obtain the pre-trained model to be pruned, we execute an SGD optimizer with a momentum of 0.9, weight decay of 0.0005, initial learning rate of 0.1, training of 350 epochs, and multiplying the learning rate by 0.1 at 175 and 262 epochs. Batch size is set to 64 and fine tuning using 40 epochs after each layer pruning with a learning rate of 0.01 and divided by 10 at epochs 5, 10.

A more easily categorized CIFAR-10 dataset can be obtained from Table 3, as pruning can still cause performance impairments to the model even at lower pruning rates. Compared with other distillation methods, our method allows the model to recover the maximum lost accuracy during the fine-tuning phase. Especially on the GoogLeNet model with a 60% pruning rate, almost lossless pruning can be achieved (only a 0.04% decrease in accuracy compared to teacher). Our method improves only 0.28% accuracy on ResNet-56 at a 60% pruning rate, but the other methods improve at most 0.11%. GoogLeNet achieves an optimal 0.93% improvement at APR, while other methods achieve at most 0.4%. This result provides evidence for our framework to better transfer knowledge.

Figure 4 illustrates the time comparison of VGGNet with different knowledge distillation approaches at a 60% pruning rate in CIFAR-10. FitNet [26] is closer to running time than the proposed approach, but it only improves the performance by 0.15% (we have an improvement of 0.64%). SP [29] and AT [27] do not effectively compensate for the loss of accuracy while consuming significant runtime resources. The KD [25] method requires less runtime but has a relatively limited precision recovery. Figure 4 shows that our approach can get the best results with fewer resources.

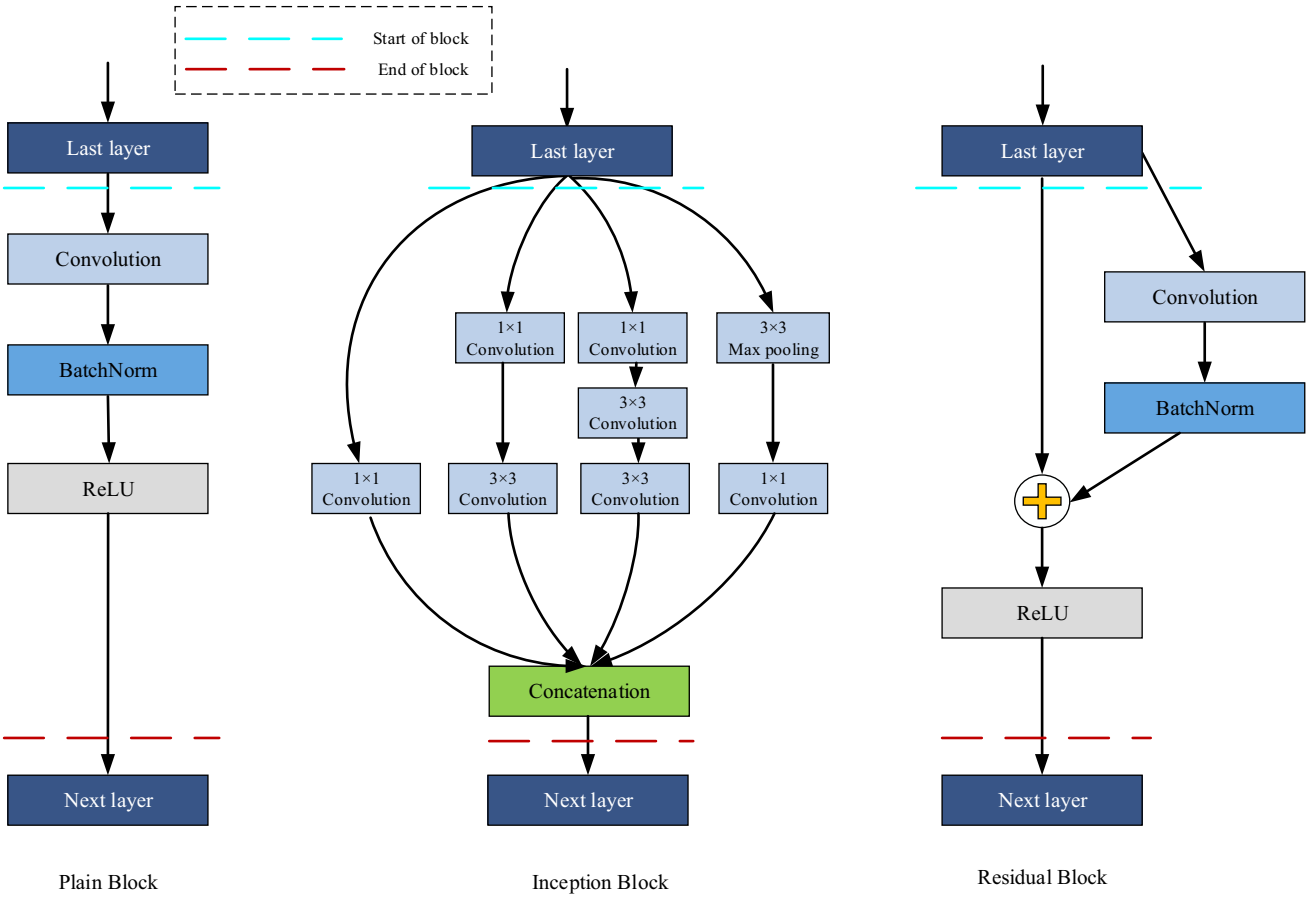

Fig. 3 Block selection locations of the network models. From left to right, VGGNet with plain block, GoogLeNet with inception block, and ResNet with residual block

Table 1 The appropriate pruning rate of models

Models	Appropriate pruning rate
VGGNet-16	[0.95], [0.5]*6, [0.9]*4, [0.8]*2
GoogLeNet	[0.10], [0.80]*5, [0.85], [0.80]*3
ResNet-56	[0.1], [0.60]*35, [0.0]*2, [0.6]*6, [0.4]*3, [0.1], [0.4], [0.1], [0.4], [0.1], [0.4]

[·] represents the pruning rate at each layer and $*n$ means the same pruning rate in the following n layers

Table 4 shows that on the more challenging to classify CIFAR-100 dataset, our method can achieve the most considerable boost on VGGNet 2.09%—over baseline (at 60% pruning rate). While ResNet-56 has only 0.41% accuracy improvement at a 70% pruning rate, the other methods only improve by 0.13% at most. Compared to the results in CIFAR-10, the proposed method generally improve more on CIFAR-100, which may explain the significant difference in performance between the teacher and student models. However, as described in [42], it is not the case that the better the teacher model performs, the better the distillation will be, as we can observe on ResNet56 at a 70% pruning rate for both datasets (note that the difference in performance between teacher and baseline in CIFAR-100 is much larger).

Both Tables 3 and 4 show that our method achieves effective and superior results on different pruning rates, models, and pruning methods and allows for continuous improvement.

Tiny-imageNet

Tiny-ImageNet [43] consists of 100,000 training and 10,000 validation images containing 200 classes, and we resize its input to 32 × 32. To obtain a pre-trained model, we use an SGD optimizer with a momentum of 0.9 and a weight decay rate of 0.0005, train 120 epochs with a learning rate of 0.01, and multiply by 0.1 at epochs 30, 60, and 90. The fine-tuning strategy after pruning is the same as CIFAR10/100.

Table 2 Comparison of the number of parameters and FLOPs for the model at various pruning rates in the CIFAR-10 dataset

Model		Params	FLOPs
VGGNet-16	Original	14.98 M	313.73 M
	60% pruning rate	5.88 M	126.42 M
	70% pruning rate	4.48 M	95.16 M
	APR	2.79 M	109.09 M
GoogLeNet	Original	6.15 M	1.52B
	60% pruning rate	2.83 M	0.73B
	70% pruning rate	2.33 M	0.59B
	APR	1.77 M	0.45B
ResNet-56	Original	0.85 M	125.49 M
	60% pruning rate	0.33 M	52.32 M
	70% pruning rate	0.25 M	39.21 M
	APR	0.47 M	62.72 M

Original represents the unpruned model

Table 3 The other methods are compared with ours in the case of different pruning rates on CIFAR-10

Model		VGGNet-16	GoogLeNet	ResNet-56
Teacher		93.85	95.21	94.23
60% pruning rate	Baseline	91.95	94.67	92.04
	KD [25]	92.34	94.71	92.04
	FitNet [26]	92.10	94.63	92.13
	AT [27]	92.38	94.89	92.15
	SP [29]	92.40	94.90	92.03
	Ours	**92.62 (+ 0.67)**	**95.17 (+ 0.5)**	**92.32 (+ 0.28)**
70% pruning rate	Baseline	91.10	94.16	90.97
	KD [25]	90.95	94.45	91.09
	FitNet [26]	91.06	94.32	88.56
	AT [27]	91.07	94.49	91.30
	SP [29]	91.22	94.51	91.35
	Ours	**91.55 (+ 0.45)**	**95.01 (+ 0.85)**	**91.49 (+ 0.52)**
ARP	Baseline	92.28	93.87	92.46
	KD [25]	92.46	93.87	92.50
	FitNet [26]	92.37	94.07	92.56
	AT [27]	92.21	94.27	92.75
	SP [29]	92.45	94.21	92.65
	Ours	**92.91 (+ 0.63)**	**94.80 (+ 0.93)**	**93.10 (+ 0.64)**

We report the Top-1 accuracy (%) of the results. The teacher refers to the pre-trained model without pruning. Baseline is the result obtained without the use of any distillation learning method

As shown in Table 5, on the larger dataset Tiny-IamgeNet, some knowledge distillation methods do not work as well as on CIFAR-10/100. However, our method still recovers the lost performance, clearly observed at a 60% pruning rate. The proposed method also obtains better results than other methods at a 70% pruning rate, allowing the model to recover 0.52% accuracy. This result proves that our method is still effective, even when it is more difficult to classify in larger datasets.

Improving each pruning

As described in the previous section, we show that our approach achieves promising results at the end of pruning. However, we hope it will improve the performance after

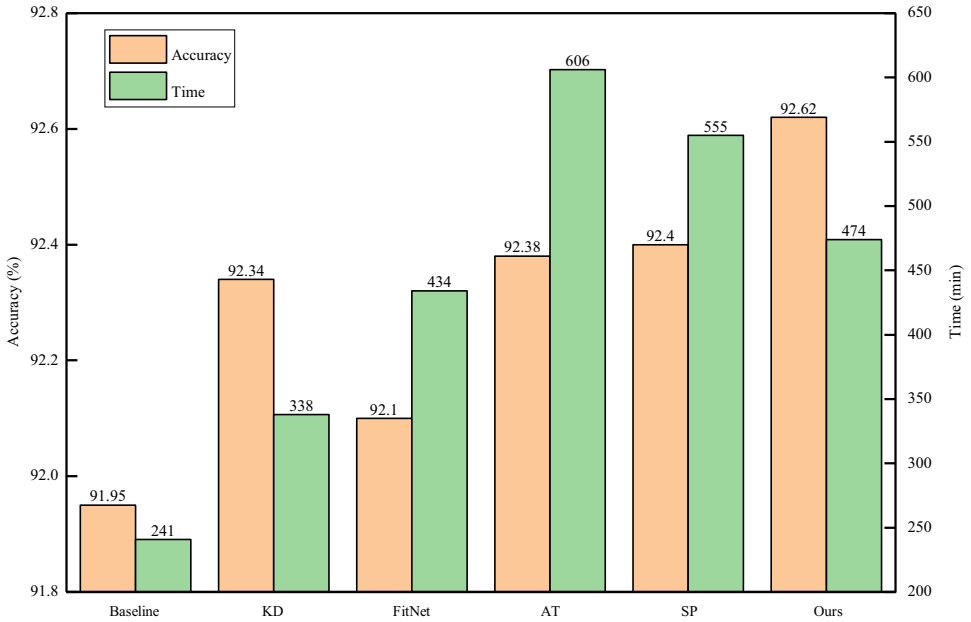

Fig. 4 Time comparison of various methods for VGGNet at 60% pruning rate of CIFAR-10. Baseline indicates that no knowledge distillation method is used

Table 4 The other methods are compared with ours in the case of different pruning rates on CIFAR-100

Model		VGGNet-16	GoogLeNet	ResNet-56
Teacher		73.95	80.49	73.34
60% pruning rate	Baseline	69.22	78.42	67.43
	KD [25]	69.55	77.73	67.67
	FitNet [26]	69.54	78.09	67.45
	AT [27]	70.93	78.92	67.88
	SP [29]	70.10	78.91	67.59
	Ours	**71.31 (+ 2.09)**	**79.20 (+ 0.78)**	**68.04 (+ 0.61)**
70% pruning rate	Baseline	66.93	77.41	64.69
	KD [25]	66.68	77.56	64.61
	FitNet [26]	66.77	77.65	64.82
	AT [27]	67.40	78.12	64.71
	SP [29]	67.13	78.11	64.82
	Ours	**67.50 (+ 0.57)**	**78.33 (+ 0.92)**	**65.10 (+ 0.41)**
ARP	Baseline	67.44	75.60	70.39
	KD [25]	67.72	75.62	70.44
	FitNet [26]	67.30	75.84	71.09
	AT [27]	67.63	76.12	71.10
	SP [29]	66.76	76.13	71.20
	Ours	**68.10 (+ 0.66)**	**76.63 (+ 1.01)**	**71.62 (+ 1.23)**

We report the Top-1 accuracy (%) of the results. The teacher refers to the pre-trained model without pruning. Baseline is the result obtained without the use of any distillation learning method

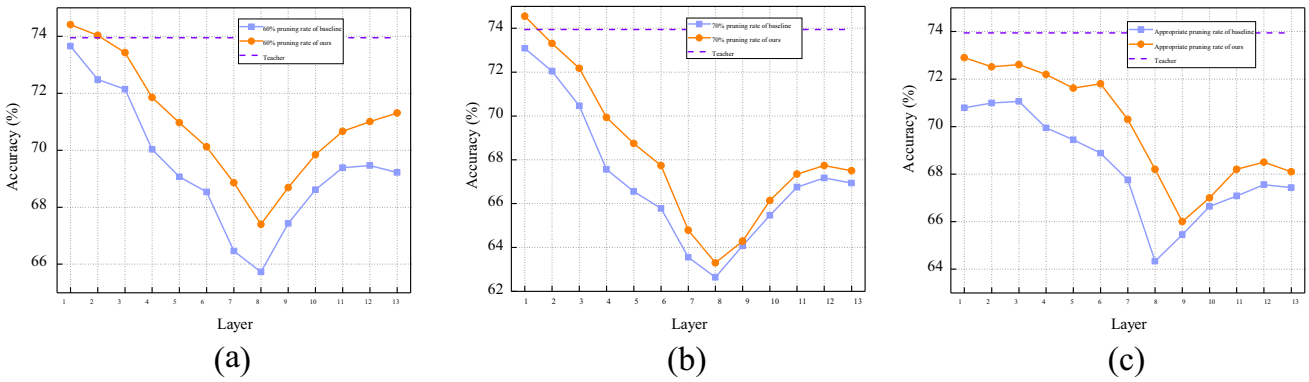

Fig. 5 Performance change of VGGNet after pruning per layer on CIFAR-100. **a**, **b** and **c** represent the different pruning rates, respectively

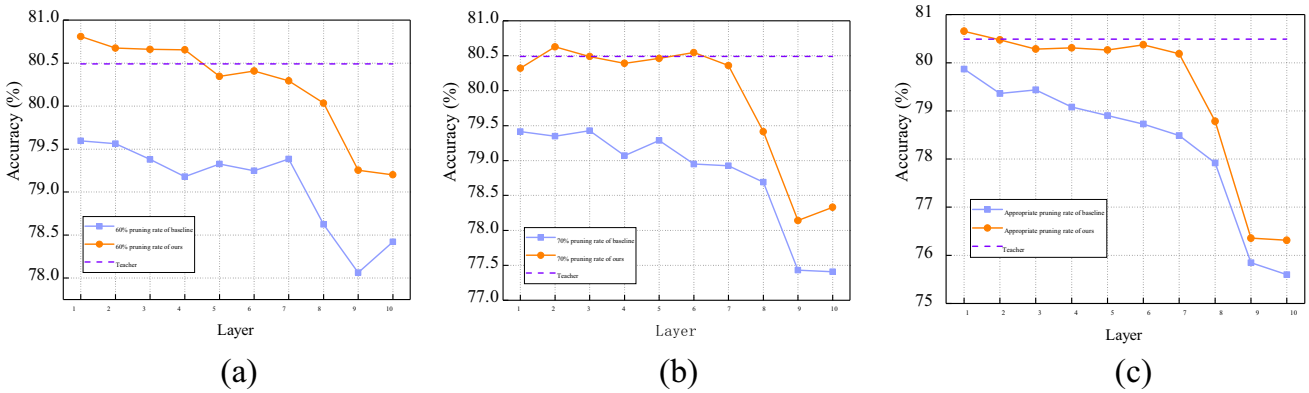

Fig. 6 Performance change of GoogLeNet after pruning per layer on CIFAR-100. **a**, **b** and **c** represent the different pruning rates, respectively

each pruning. Figures 5, 6, and 7 illustrate the performance changes of VGGNet-16, GoogLeNet, and ResNet-56 after each pruning completion on the CIFAR-100 dataset with different pruning rates, respectively. The blue line indicates the baseline accuracy, i.e., the results obtained without distillation. The orange line shows the results obtained with the proposed distillation. The dotted line represents the accuracy of the teacher model without pruning. The pruning of each layer in the convolution layer removes the irrelevant filters, which leads to a lower precision. As shown in Figs. 5, 6 and 7, our method still works during the pruning process, which also means that the proposed approach is still practical even if we haven't completed the pruning. Moreover, the performance of the student is better than that of the teacher model during the initial process of pruning proceeding, which also shows the ability of our approach to combine structured pruning and distillation learning methods better.

Ablation study

In order to further analyze the contribution of each of our proposed losses, we add the ablated portions step by step to observe their effects. We perform experiments related to VGGNet-16 with a 70% pruning rate on CIFAR-10, as shown in Table 6, where baseline refers to the pruning process without using our method. It can be observed that our proposed method in "Progressive feature distillation" improves the performance of the model to the maximum, and the proposed progressive mechanism based on the pruning process further improves the pruning process in terms of feature distillation. In conclusion, the weighted combination of the proposed components can be used to compensate for the loss of performance due to pruning as much as possible.

Two combined strategies

The strategy of combining distillation learning and pruning can be broadly divided into two categories: using after pruning is completed and using during pruning, and Fig. 8 shows the performance and time comparison of our proposed method on these two strategies. It has been shown that using distillation in the pruning process leads to higher precision recovery, but it also requires more training time. As a result of our progressive distillation process, it takes less time and achieves greater performance gains compared to AT [27]. Compared with the other one, FitNet [26], although it takes less time to train, its accuracy improvement is not even as

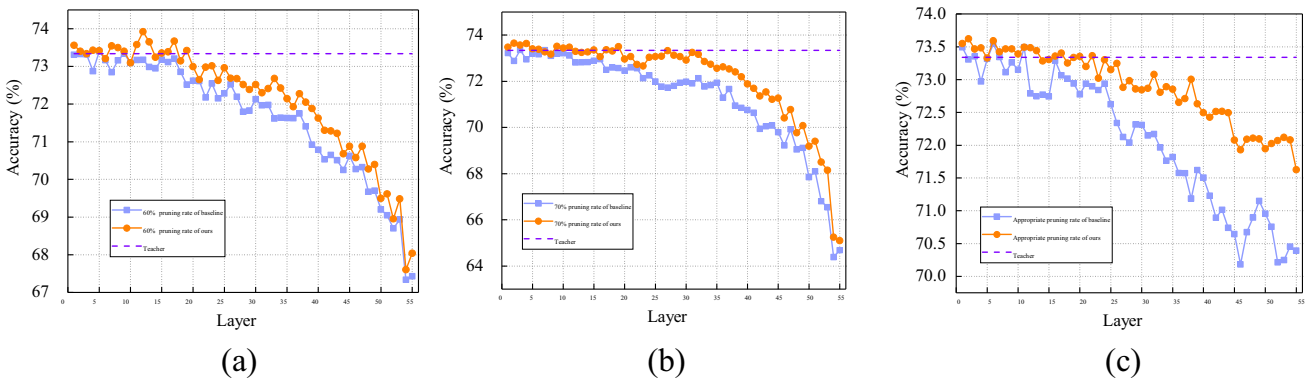

Fig. 7 Performance change of ResNet-56 after pruning per layer on CIFAR-100. **a**, **b** and **c** represent the different pruning rates, respectively

Table 5 The other methods are compared with ours in the case of different pruning rates on Tiny-ImageNet

Model		GoogLeNet
Teacher		61.81
60% pruning rate	Baseline	60.38
	KD [25]	60.25
	FitNet [26]	60.13
	AT [27]	60.34
	SP [29]	60.10
	Ours	**60.78 (+ 0.4)**
70% pruning rate	Baseline	58.53
	KD [25]	58.38
	FitNet [26]	58.44
	AT [27]	58.54
	SP [29]	58.64
	Ours	**59.05 (+ 0.52)**

We report here the Top-1 accuracy (%) of the results. The teacher refers to the pre-trained model without pruning. Baseline is the result obtained without the use of any distillation learning method

good as using our distillation method only after the completion of pruning. This result demonstrates that our method can recover the decreased accuracy within a relatively short training time without affecting the inference speed of the model.

Conclusion and future work

We propose a progressive multi-level distillation learning method to alleviate the accuracy drop by structured pruned networks. This method takes advantage of the characteristics of structured pruning, which allows the pruned network to learn more information from the teacher network. Experiments on different datasets, model architectures, and pruning rates show that the proposed approach achieves better performance than other approaches, and the accuracy of the model is improved after every pruning. Further experiments demonstrate that the proposed method in the pruning process enhances the model performance more effectively. Our approach has higher efficiency in training time and does not influence inference time. Our study provides a valuable approach to better integrate pruning and distillation learning.

In future work, we hope to extend this idea to more complicated tasks such as object detection and semantic segmentation. Furthermore, it is also worth exploring how to combine better pruning, knowledge distillation, and other compression techniques like quantization.

Table 6 Ablation experiments of the proposed method

	Baseline	+ Feature distillation	+ Progressive learning ("Progressive feature distillation")	+ Output distillation ("Output logits distillation learning")
Accuracy (%)	91.10	91.41	91.45	91.55
Diff	–	+ 0.31	+ 0.35	+ 0.45

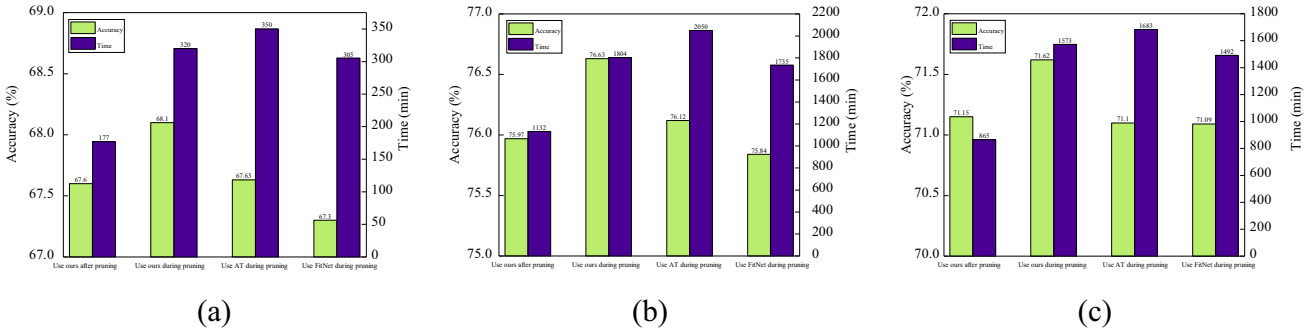

Fig. 8 The performance and training time of the proposed approach are compared only when pruning is finished and during pruning. **a**, **b**, and **c** represent the VGGNet-16, GoogLeNet, and ResNet-56 models under APR on CIFAR-100, respectively

Acknowledgements This is a part research accomplishment of the Key Research and Development Project of Anhui Province (202204c06020022, 202104a06020012, 201904a06020056), Independent Project of Anhui Key Laboratory of Smart Agricultural Technology and Equipment (APKLSATE2019X001).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. LeCun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
2. Krizhevsky A, Sutskever I, Hinton G E (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst
3. Huang G, Liu Z, Van Der Maaten L et al (2017) Densely connected convolutional networks. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2017:4700–4708
4. Girshick R, Donahue J, Darrell T et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2014:580–587
5. Ren S, He K, Girshick R, et al (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst
6. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv:1804.02767
7. Young T, Hazarika D, Poria S et al (2018) Recent trends in deep learning based natural language processing. IEEE Comput Intell Mag 13(3):55–75
8. Zagoruyko S, Komodakis N (2016) Wide residual networks. arXiv:1605.07146
9. Xie S, Girshick R, Dollár P et al (2017) Aggregated residual transformations for deep neural networks. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2017:1492–1500
10. Choudhary T, Mishra V, Goswami A et al (2020) A comprehensive survey on model compression and acceleration. Artif Intell Rev 53(7):5113–5155
11. Vadera S, Ameen S (2020) Methods for pruning deep neural networks. arXiv:2011.00241
12. Gou J, Yu B, Maybank SJ et al (2021) Knowledge distillation: a survey. Int J Comput Vision 129(6):1789–1819
13. Chen L, Chen Y, Xi J et al (2021) Knowledge from the original network: restore a better pruned network with knowledge distillation. Complex Intell Syst 2021:1–10
14. Kim J, Chang S, Kwak N (2021) PQK: model compression via pruning, quantization, and knowledge distillation. arXiv:2106.14681
15. Cui B, Li Y, Zhang Z (2021) Joint structured pruning and dense knowledge distillation for efficient transformer model compression. Neurocomputing 458:56–69
16. Wang R, Zhang W, Ding J et al (2021) Deep neural network compression for plant disease recognition. Symmetry 13(10):1769
17. Han S, Liu X, Mao H et al (2016) EIE: efficient inference engine on compressed deep neural network. ACM SIGARCH Comput Architecture News 44(3):243–254
18. LeCun Y, Denker J, Solla S (1989) Optimal brain damage. Adv Neural Inf Process Syst
19. Hassibi B, Stork D (1992) Second order derivatives for network pruning: Optimal brain surgeon. Adv Neural Inf Process Syst
20. Han S, Pool J, Tran J, et al (2015) Learning both weights and connections for efficient neural network. Adv Neural Inf Process Syst
21. Li H, Kadav A, Durdanovic I, et al (2016) Pruning filters for efficient convnets. arXiv:1608.08710
22. Liu Z, Li J, Shen Z et al (2017) Learning efficient convolutional networks through network slimming. Proc IEEE Int Conf Computer Vision (ICCV) 2017:2736–2744
23. Molchanov P, Tyree S, Karras T, et al (2016) Pruning convolutional neural networks for resource efficient inference. arXiv:1611.06440
24. Lin M, Ji R, Wang Y et al (2020) Hrank: filter pruning using high-rank feature map. Proc IEEE Confer Comput Vision Pattern Recogn (CVPR) 2020:1529–1538
25. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv:1503.02531

26. Romero A, Ballas N, Kahou S E, et al (2014) Fitnets: hints for thin deep nets. arXiv:1412.6550
27. Komodakis N, Zagoruyko S (2017) Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: International conference on learning representations (ICLR)
28. Yim J, Joo D, Bae J et al (2017) A gift from knowledge distillation: fast optimization, network minimization and transfer learning. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2017:4133–4141
29. Tung F, Mori G (2019) Similarity-preserving knowledge distillation. Proc IEEE Int Conf Comput Vision (ICCV) 2019:1365–1374
30. Li Q, Jin S, Yan J (2017) Mimicking very efficient network for object detection. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2017:6356–6364
31. Zhang L, Ma K (2020) Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In: International conference on learning representations (ICLR)
32. Xie H, Jiang W, Luo H et al (2021) Model compression via pruning and knowledge distillation for person re-identification. J Ambient Intell Humaniz Comput 12(2):2149–2161
33. Gong Y, Liu L, Yang M, et al (2014) Compressing deep convolutional networks using vector quantization. arXiv:1412.6115
34. Hubara I, Courbariaux M, Soudry D, et al (2016) Binarized neural networks: Training neural networks with weights and activations constrained to +1 or -1. arXiv:1602.02830
35. Han S, Mao H, Dally W J (2015) Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv:1510.00149
36. Walawalkar D, Shen Z, Savvides M (2020) Online ensemble model compression using knowledge distillation. Eur Conf Comput Vision (ECCV) 2020:18–35
37. Chen G, Choi W, Yu X, et al (2017) Learning efficient object detection models with knowledge distillation. Adv Neural Inf Process Syst
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
39. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. Proce IEEE Conf Comput Vision Pattern Recogn (CVPR) 2016:770–778
40. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. Proc IEEE Conf Comput Vision Pattern Recogn (CVPR) 2015:1–9
41. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
42. Mirzadeh SI, Farajtabar M, Li A et al (2020) Improved knowledge distillation via teacher assistant. Proc Conf AAAI Artif Intell 34(04):5191–5198
43. Le Y, Yang X (2015) Tiny imagenet visual recognition challenge. CS 231N 7(7):3