Filter Pruning via Feature Discrimination in Deep Neural Networks

Zhiqiang He[1], Yaguan Qian[1][0000−0003−4056−9755]⋆, Yuqi Wang[1], Bin Wang[2], Xiaohui Guan[3], Zhaoquan Gu[4], Xiang Ling[5], Shaoning Zeng[6], Haijiang Wang[1], and Wujie Zhou[1]

[1] Zhejiang University of Science and Technology, Hangzhou Zhejiang 310023, China qianyaguan@zust.edu.cn
[2] Zhejiang Key Laboratory of Multidimensional Perception Technology, Application and Cybersecurity, Hangzhou Zhejiang 310052, China wbin2006@gmail.com
[3] Zhejiang University of Water Resources and Electric Power, Hangzhou Zhejiang 310023, China
[4] Cyberspace Institute of Advanced Technology (CIAT), Guang Zhou University, Guangzhou Guangdong 510006, China
[5] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[6] Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Huzhou Zhejiang 313000, China

Abstract. Filter pruning is one of the most effective methods to compress deep convolutional networks (CNNs). In this paper, as a key component in filter pruning, We first propose a feature discrimination based filter importance criterion, namely Receptive Field Criterion (RFC). It turns the maximum activation responses that characterize the receptive field into probabilities, then measure the filter importance by the distribution of these probabilities from a new perspective of feature discrimination. However, directly applying RFC to global threshold pruning may lead to some problems, because global threshold pruning neglects the differences between different layers. Hence, we propose Distinguishing Layer Pruning based on RFC (DLRFC), i.e., discriminately prune the filters in different layers, which avoids measuring filters between different layers directly against filter criteria. Specifically, our method first selects relatively redundant layers by hard and soft changes of the network output, and then prunes only at these layers. The whole process dynamically adjusts redundant layers through iterations. Extensive experiments conducted on CIFAR-10/100 and ImageNet show that our method achieves state-of-the-art performance in several benchmarks.

Keywords: Model Compression; Filter Pruning; Receptive Field Criterion; Distinguishing Layer Pruning

1 Introduction

In the past few years, CNNs have achieved the most excellent performance in various computer vision tasks, such as target classification [15,9], object detec-

⋆ Corresponding Author

Fig. 1: Feature discrimination. On ImageNet, we randomly select 9 images and visualize the 20^{th} filter in the last convolutional layer of VGG16. If the filter responds more uniformly to each class, then the features learned by the filter will play a small role in discriminating those classes, i.e., contribute less to the classification. We also experimentally demonstrate the effectiveness of this new pruning angle in subsection 4.7

tion [1] and super-resolution [16]. However, over-parameterization is still a severe challenge to the CNNs' deployment in the edge intelligent device. For this reason, many network compression methods have been proposed at present, such as quantification [7], knowledge distillation [12], and network pruning [17,25,28]. Network pruning, as one of the most widely used methods, has attracted extensive attention. According to the pruned objects, network pruning methods are mainly divided into two types: unstructured pruning [8,29] and structured pruning [17,23]. Unstructured pruning removes the specific weights in a filter to obtain a sparse network, which requires special hardware and software for acceleration. Structured pruning directly removes the entire filter without any specially designed acceleration. Therefore, we focus on structured pruning in this paper.

In structured pruning, the filter importance criterion plays a vital role to determine which filters need to be pruned. Previous pruning methods design the filter importance criteria from various perspectives, such as heuristic experience [17], mathematical statistics [28], and network loss [4,26]. In this paper, we create a novel filter importance criterion named Receptive Field Criterion (RFC) from the view of feature discrimination (as illustrated in Fig 1). Inspired by the receptive field analysis of network interpretability [5,42], we convert the maximum activation responses of a receptive field into probabilities that represent the filter's contribution to each class. Then, from the perspective of feature discrimination, we introduce the information entropy to represent the uniformity of the filter's contribution to all class. Thus, filters with higher information entropy is prone to be pruned. We also compared the RFC with other criterion , and the results show that RFC can better measure filter importance.

Many existing works directly conduct pruning according to a global threshold determined by filtering criteria. However, we find that global threshold pruning ignores the relative importance between layers. To address this problem, we propose a new pruning method: distinguishing layer pruning based on RFC (DLRFC). We select redundant layers by comparing the impact of each layer on

network performance. This effect is quantified by both hard and soft labels of the network output vector, and then we prune filters only within redundant layers. The redundant layers are dynamically adjusted in the whole process iteratively. Our contributions are summarized as follows:

- We propose a new filter criterion that utilizes network interpretability to construct a filter maximum response set, and then judges redundancy based on the consistency of filter's response to the class.
- We propose a discriminative layer pruning method that avoids direct global comparison of importance according to filter criteria, which results in better network structures.
- We evaluate the effectiveness of our method on various networks and datasets. Experiment results show that our method achieves state-of-the-art performance.

2 Related work

Many previous works studied pruning methods and their filter importance criteria. Li [17] used the L1 norm of filters while Liu [23] proposed SD score as the importance score of filters to prune the unimportant filters. Yu [40] applied the feature ranking technique to measure the importance of filters in final response layer. Meng [27] used the stripe of filter as the granularity and took L1 norm of each stripe as the importance criterion to obtain irregular filters. Ding [3] proposed a centripetal SGD to train the network in the hyperspace of parameters and prune the similar filters after training. He [10] proposed a pruning method to prune filters closed to the geometric center of the network. Besides filter importance, Liu [25] considered the scale factor in BN layer as the channel importance to determine which channel to be pruned. Molchanov [28] proposed a new criterion based on the Taylor expansion of network loss change. After scoring all filters using importance criteria, many methods directly perform global pruning. But Wang [39] claimed that pruning in layers with the most structural redundancy outperforms pruning the least important filters across all layers. Therefore, it is better to prune in relatively redundant layers every time. Recently, some researchers have introduced network structure search into pruning. Liu [24,36] applied meta learning to pruning and Lin [21] used artificial bee colony.

Insight into network interpretability can help researchers better understand the network behavior. Zeiler and Fergus [41] use a multi-layer deconvolutional network to project feature activation back to pixel space to learn the representation. Experiments showed that filters of low layers learn low-level features such as color and edge, while filters of high layers learn high-level features such as body parts. Therefore, different layers may not be well pruned according to the filter criterion alone. Zhou [42] visualized the receptive fields by the images with top 5% activation response to filters. Girshick [6] demonstrated that convolutional layers complete most of the learning ability of a CNN and the extracted features are universal.

3 Method

3.1 Receptive Field Criterion (RFC)

Let $N \in \mathbb{R}$ be the number of input images and $C \in \mathbb{R}$ be the number of classes. $M_{ij}(k) \in \mathbb{R}^{h_i \times w_j}$ is the channel generated from the k-th image by the j-th filter in the i-th layer after the activation function. $S_{ij} = \{(M_{ij}(k), v_k), \quad k = 1, 2, \cdots, N\}$ is a set consisting of channel-label pairs, where v_k is the class label of the k-th image. Next, we define the filter's response score to the input image. Define the response score set A_{ij} to represent the activation response of the j-th filter to all images, then the stronger the filter response to the image, the higher the response score is, as shown below:

$$A_{ij} = \{\|M_{ij}(k)\|, \quad k = 1, 2, \cdots, N\} \tag{1}$$

where $\|M_{ij}(k)\| = \sum_{n=1}^{h_i \times w_j} \left\|m_{ij}^n(k)\right\|$ and n is the component of the corresponding channel. Zhou et al. [42] used 5% of images with the largest filter response to detect the filter's receptive field, which means that these images reflect the filter's learned features. Therefore, after obtaining all responses in A_{ij}, we construct a maximum response set A_{ij}^* consisting of the top 5% activation responses. With the channel-label pairs corresponding to the responses in A_{ij}^*, we construct the channel-label set:

$$S_{ij}^* = \{(M_{ij}(k), v_k) | \, \|M_{ij}(k)\|_2 \in A_{ij}^*, k = 1, 2, \cdots, N\}. \tag{2}$$

Then we extract the label set V_{ij} from S_{ij}^* as follows:

$$V_{ij} = \{v_k | (M_{ij}(k), v_k) \in S_{ij}^*, \quad k = 1, 2, \cdots, N\} \tag{3}$$

Here, V_{ij} is used to generate the probability distribution of maximum response of the corresponding filter. We convert V_{ij} to the maximum response probability as follows:

$$p_{ij}^n = \frac{e^{\left|V_{ij}^n\right|}}{\sum_{n=1}^{C} e^{\left|V_{ij}^n\right|}}, n = 1, 2, ..., C \tag{4}$$

where $|V_{ij}|$ is the number of elements in set V_{ij} and $\left|V_{ij}^n\right|$ represents the number of elements belong to class n. Note that only when $p_{ij}^n > 0$ will it be added to the next calculation. From the perspective of feature discrimination, we take information entropy as our RFC. With those positive p_{ij}^n, we calculate their information entropy [33]:

$$H_{ij} = \sum_{n=1}^{C} p_{ij}^n \cdot \log \frac{1}{p_{ij}^n} \tag{5}$$

where H_{ij} is the RFC score of the j-th filter in the i-th layer. The greater RFC score indicates the more average contribution to each class, and vice versa. We

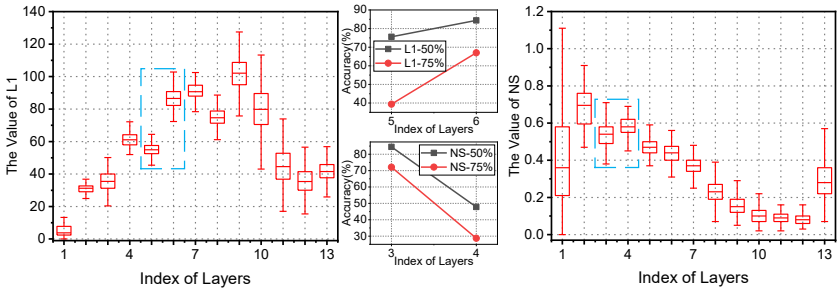


Fig. 2: The upper part is a boxplot of the filter scores (L1[17], NS[25]) of each layer of VGG16 on CIFAR-10. The middle of the boxplot shows the measured accuracy after only the index layer is selected for trimming. For example, when the index of the middle graph is 5, L1-50% represents the accuracy after pruning only 50% filters of the fifth layer of VGG16 according to the L1 criterion.

demonstrate the effectiveness of RFC in Section 4.7, showing that RFC is a good measure of filter importance. Besides, we state in Section 4.5 that most of the filters with uniform response are redundant because of their invalid response to the common background of the picture.

For conducting pruning, we define an RFC-score set T_i of all H_{ij} in the i-th layer as follows:

$$T_i = \{H_{ij}, \quad j = 1, 2, ..., O_i\} \tag{6}$$

where O_i is the number of filters in the i-th layer. We denote T_i^* as a set consisting of low RFC values (corresponding to the retained filters). Then the retained filter set K_i^* is defined as follows:

$$K_i^* = \{K_{ij} | H_{ij} \in T_i^*, j = 1, 2, \cdots, O_i\} \tag{7}$$

where K_{ij} is the correspond filter of H_{ij}. In our pruning procedure, we retain the filters in K_i^* to rebuild the network.

3.2 Distinguishing layer pruning based on RFC

Thoughts on directly using criterion pruning for global pruning: Traditional global threshold pruning methods combine all T_i into a global set $T = \bigcup_{i=1}^{W} T_i$, where W is the total number of convolutional layers in the network. In this case, we cannot simply measure all filters at the same level to construct T^*. As shown in Fig 2, we observe that the value range of the pruning criterion is usually significantly different between layers. If the global set pruning is simply used, a certain layer may be completely pruned. In addition, we found that criteria for this difference do not compare the importance of filters across layers. As observed in the L1 boxplot of Fig 2, layer 5 is considered more redundant than layer 6 under global threshold pruning. But from the plot in the middle of the boxplot we know that layer 5 has a bigger impact on network performance,

which is more important. The same situation exists in NS. This means that some filters are considered equally important in global threshold pruning, but their importance to the network varies greatly from layer to layer.

The above shows that there are obvious differences in the pruning situation of different layers. Therefore, we cannot directly compare the filter importance of each layer according to the criteria, but first select the relatively redundant layers, and then prune only in these layers. To this end, we consider each T_i in $T = \bigcup_{i=1}^{W} T_i$ separately during pruning using RFC, first select redundant layers, and then perform pruning. Moreover, the redundant layers are dynamically adjusted in the whole process iteratively.

Distinguishing Layer Pruning based on RFC: Inspired by the filter redundancy measurement in [26,40,4], we remove each layer in rotation and calculate the network output change to determine the redundant layer.

For a network $\mathcal{F} = (K_1, K_2, \cdots, K_W)$, where K_i represents the filter set of the i-th layer and corresponds to the RFC score set T_i. We remove the same ratio λ of filters in each layer according to T_i to obtain T_i^* as follows:

$$T_i^* = sort_{0.5}(T_i) \tag{8}$$

where $sort_{0.5}$ represents removing 50% of the elements in T_i with the highest H values (50% for a trade-off between effect and efficiency [4]). Thus, T_i^* represents the set of the lowest part of H values in T_i. Corresponding to T_i^*, we construct our network by replacing K_i with K_i^* as presented in Eq. (7):

$$\mathcal{F}_i = (K_1, \cdots, K_i^*, \cdots, K_W) \tag{9}$$

where $\mathcal{F}_i$ represents the network obtained by only pruning the i-th layer of $\mathcal{F}$. Denote $\mathbf{g}_i(\mathbf{g})$ and $y_i(y)$ as the output probability vector and class value of $\mathcal{F}_i(\mathcal{F})$. Since calculating the change of network output on the entire training dataset is time-consuming, we randomly select 5% images from the training set to construct a proxy dataset D_{proxy}. To measure output changes before and after pruning, we measure hard and soft changes. A hard change is a change in the output target class, and a soft change is a change in the output of all classes. We introduce cosine similarity as a soft change, which is widely used to measure similarity [37,34] between vectors. A hard change is combined with the change of target class. We define the layer redundancy R_i as follows:

$$R_i = \gamma E_k\left[sign(|y^k - y_i^k|)\right] + (1-\gamma)\left(1 - E_k\left[\frac{g^k \cdot g_i^k}{|g^k|\,|g_i^k|}\right]\right), \quad i = 1, 2, \ldots, W \tag{10}$$

where k is the index of the image in D_{proxy} and γ is the balance coefficient between [0,1]. The smaller the value of R_i, the smaller the change of the network performance, the higher the redundancy of the i-th layer. If R_i is large, the corresponding layer will be considered important. We introduce a redundancy threshold ε to control redundant layers in this iteration, so we only prune layers with high redundancy. as follows:

$$\mathcal{F} \leftarrow \begin{cases} K_i^*, & R_i < \varepsilon \\ K_i, & R_i \geq \varepsilon \end{cases} \tag{11}$$

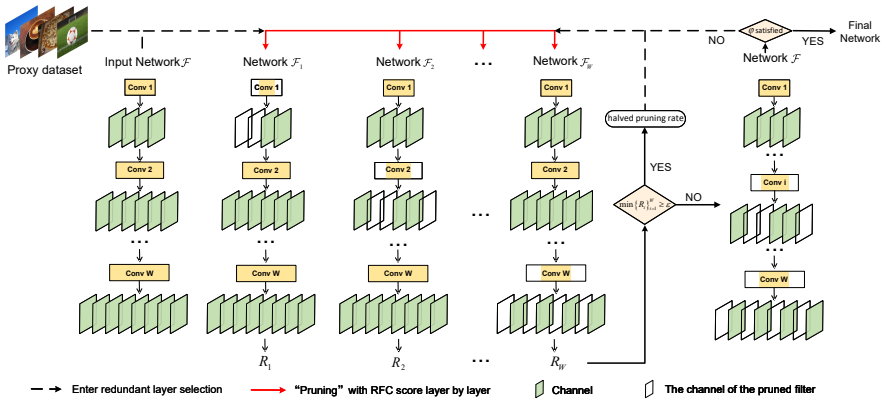


Fig. 3: Distinguishing layer pruning procedure. Conv represents the convolutional layer. Input the proxy dataset to original network $\mathcal{F}$. Then "prune" each layer respectively to obtain each layer's redundancy R_i.If $\min\{R_i\}_{i=1}^{W} \geq \varepsilon$ holds when halved pruning granularity re-enters redundant layer selection. Otherwise, prune the layer of R_i smaller than ε to obtain the network $\mathcal{F}$. Repeat this process until the constraint φ is satisfied.

Here K_i^* represents the layer that has been pruned, and K_i represents the layer that has not been pruned. The network $\mathcal{F}$ is then reconstructed using these layers. The fixed pruning rate in Eq. (8) may result in redundant layers satisfying Eq. (11) not appearing later in the iteration. Therefore, we adjust the ratio by ε, reducing the pruning rate by half to reduce the network output change, and thus perform better pruning. Therefore, when $\min\{R_i\}_{i=1}^{W} \geq \varepsilon$ is satisfied, we permanently halve the pruning rate in Eq. (8) to automatically adjust the pruning granularity. After we prune the redundant layer, we fine-tune the network for one epoch to maintain its stability, which is the same as [13]. The entire procedure (named as DLRFC) is illustrated in Fig. 3 and its pseudocode is shown in Algorithm 1.

4 Experimental setup

4.1 Experimental setup

Dataset and Models: As stated in [43], for different size of datasets, a pruning method may lead to different results. We use CIFAR-10/100 [14] and ImageNet [15] for our experiments, which are three popular datasets for pruning evaluation. On CIFAR datasets, we evaluate our method through two classic types of network structures: VGG16 without shortcut connection and ResNet56 with shortcut connection. On ImageNet, we use ResNet50 [9] and MobileNetV2 [31].
Baseline Setting: Our baseline settings are consistent with those in [25,27,21]. On CIFAR-10/100, the model is trained for 160 epochs and the batch size is

Algorithm 1 Distinguishing Layer Pruning based on RFC

Input: Proxy dataset $\mathcal{D}_{proxy}$, redundancy threshold ε, network $\mathcal{F} = (K_1, \cdots, K_W)$, and balance coefficient γ.
Output: Pruned network.

repeat
  Calculate all $\mathbf{g}^k$ and $\mathbf{y}^k$ of $\mathcal{F}$ on the proxy dataset $\mathcal{D}_{proxy}$
  for $i = 1$ to W **do**
    Obtain T_i by RFC
    $T_i^* \leftarrow sort_{0.5}(T_i)$
    $K_i^* \leftarrow \{K_{ij} | H_{ij} \in T_i^*\}$
    $\mathcal{F}_i \leftarrow (K_1, \cdots, K_i^*, \cdots, K_W)$
    Calculate all $\mathbf{g}_i^k$ and $\mathbf{y}_i^k$ of $\mathcal{F}_i$ on the proxy dataset $\mathcal{D}_{proxy}$
    $R_i = \gamma E_k \left[sign(|y^k - y_i^k|) \right] + (1-\gamma)\left(1 - E_k \left[\frac{g^k \cdot g_i^k}{|g^k||g_i^k|} \right] \right)$
  end for
  if $\min\{R_i\}_{i=1}^{W} \geq \varepsilon$ **then**
    go to 3 **and** halve the pruning granularity in step (5)
  end if
  prune only redundant layers for $\mathcal{F}$
  Stablize the network, return to 3.
until Pruned network satisfies the predefined constraint.

64. The initial learning rate is set to 0.1 and divided by 10 at 50% and 75% of the epoch. Simple data augmentation (random cropping and random horizontal flip) is used for training images. On ImageNet, our settings are as same as the popular settings [11,26].

Pruning Setting: We complete all experiments on NVIDIA RTX 2080 Ti and NVIDIA RTX 3090. We set the balance coefficient $\gamma = 0.8$ and the redundancy threshold ε =0.1. When pruning ResNet and MobileNetV2, our strategy is similar to [26], i.e. consider all shortcuts at each stage and prune shortcuts and non-shortcuts separately. On CIFAR-10/100, we train the pruned model from scratch using the same FLOPs. On ImageNet, the fine-tuning settings for the pruned model are the same as those in [43].

4.2 Experimental results on CIFAR-10/100

We prune VGG16 and ResNet56 on CIFAR-10/100 datasets and comprehensively compare our method with other state-of-the-art methods. On CIFAR-10, we obtain two structures (DLRFC-1 and DLRFC-2) from VGG16 and one structure from ResNet56; on CIFAR-100, we obtain one structure for VGG16 and ResNet56, respectively. We record the basline accuracy, pruned model accuracy, accuracy drop, FLOPs and parameters reduction of each model and compare them with other state-of-the-art pruning methods. Comparison results are summarized in Tab. 1 and Tab. 2, respectively.

For VGG16, DLRFC-2 reduces 77% FLOPs and 94% parameters; DLRFC-1 reduces 61% FLOPs and 92% parameters on CIFAR-10. Compared with others,

Table 1: Comparison result of VGG16 and ResNet56 on CIFAR-10. Acc↓ is the accuracy drop of pruned model compared to the baseline model. FLOPs↓ and Param↓ represent the reduction of FLOPs and parameters in percentage, respectively.

Model	Method	Baseline Acc.(%)	Pruned Acc.(%)	Acc. ↓ (%)	FLOPs ↓ (%)	Param. ↓ (%)
VGG16	Hinge [18]	93.59	94.02	-0.43	39.07	19.95
	NSPPR [43]	93.88	93.92	-0.04	54.00	-
	AOFP [4]	93.38	93.84	-0.46	60.17	-
	DLRFC-1	**93.25**	**93.93**	**-0.68**	**61.23**	**92.86**
	DPFPS [30]	93.85	93.67	0.18	70.85	93.92
	PFF [27]	93.25	93.65	-0.40	71.16	92.66
	ABC [21]	93.02	93.08	-0.06	73.68	88.68
	HRank [22]	93.96	91.23	2.73	76.50	92.00
	AOFP [4]	93.38	93.28	0.10	75.27	-
	DLRFC-2	**93.25**	**93.64**	**-0.39**	**76.95**	**94.38**
ResNet56	NISP [40]	93.04	93.01	0.03	43.60	42.60
	FPGM [11]	93.59	93.49	0.10	53.00	-
	NSPPR [43]	93.83	93.84	-0.03	47.00	-
	ABC [21]	93.26	93.23	0.03	54.13	54.20
	SRR-GR [39]	93.38	93.75	-0.37	53.80	-
	DPFPS [30]	93.81	93.20	0.61	52.86	46.84
	DLRFC	**93.06**	**93.57**	**-0.51**	**52.58**	**55.63**

Table 2: Comparison result of VGG16 and ResNet56 on CIFAR-100.

Model	Method	Baseline Acc.(%)	Pruned Acc.(%)	Acc. ↓ (%)	FLOPs ↓ (%)	Param. ↓ (%)
VGG16	NS [25]	73.83	74.20	-0.37	38.00	-
	COP [38]	72.59	71.77	0.82	43.10	73.20
	NSPPR [43]	73.83	74.25	-0.42	43.00	-
	DLRFC	**73.54**	**74.09**	**-0.55**	**43.40**	**82.50**
ResNet56	NS [25]	72.49	71.40	1.09	24.00	-
	NSPPR [43]	72.49	72.46	0.03	25.00	-
	DLRFC	**71.14**	**71.41**	**-0.27**	**25.50**	**25.90**

DLRFC has the best result, which is higher by 0.49% in accuracy gain than AOFP [4]. On CIFAR-100, DLRFC reduces 43% FLOPs and 82% parameters. In addition, the accuracy gain under the same FLOPs is better than NSPPR.

For ResNet56, DLRFC reduces 52% FLOPs and 55% parameters on CIFAR-10. Compared with ABC [21], our model achieves a higher accuracy gain. On CIFAR-100, DLRFC reduces 25% FLOPs and 26% parameters, which outperforms other methods. In brief, DLRFC can produce a more compact model and give better performance.

Table 3: Comparison results of the Top-1 ImageNet accuracy of our method and state-of-the-art pruning methods on ResNet50 and MobileNetV2.

Model	Method	Baseline Acc.(%)	Pruned Acc.(%)	Acc. ↓ (%)	FLOPs ↓ (%)	Param. ↓ (%)
ResNet50	G-SD-B [23]	76.15	75.85	0.30	44	23
	MetaPruning [24]	76.60	75.40	1.20	50	-
	NSPPR [43]	76.15	75.63	0.52	54	-
	DPFPS [30]	76.15	75.55	0.60	46	-
	S-COP [35]	76.15	75.26	0.89	54	52
	LRF-60 [13]	76.15	75.71	0.50	56	53
	DLRFC	**76.13**	**75.84**	**0.29**	**54**	**40**
MobileNetV2	W-Gates [20]	71.80	70.90	0.90	25	-
	DPFPS [30]	72.00	71.10	0.90	25	-
	ManiDP [43]	71.80	71.41	0.39	30	-
	CC [19]	71.88	70.91	0.97	29	-
	DLRFC	**71.80**	**71.88**	**-0.08**	**30**	**-**

Table 4: Comparison results of the Top-1 accuracy and latency of Resnet34 and Resnet50 on ImageNet. Uniform means we set the same pruning ratio for each layer. The network input test batch is set to 100.

	DLRFC			Uniform		
Model	FLOPs(G)	Acc(%)	Latency(ms)	FLOPs(G)	Acc(%)	Latency(ms)
ResNet34	-	-	-	3.7(1X)	73.88	54.04
	2.9	**73.86**	**46.70**	2.9	72.56	49.23
	2.3	**73.28**	**40.65**	2.4	72.05	44.27
	2.0	**72.99**	**37.13**	2.1	71.32	43.47
ResNet50	-	-	-	4.1(1X)	76.15	105.75
	3.0	**76.33**	**95.52**	3.1	75.59	97.87
	2.5	**76.18**	**87.66**	2.6	74.77	91.53
	1.9	**75.81**	**79.83**	2.1	74.42	85.20

Compared with ResNet56 on both CIFAR-10/100 datasets, we observe that even at a higher pruning ratio, the pruned VGG16 model can still achieve good performance. A possible explanation is that VGG16 is an extra-large model for CIFAR-10/100, resulting in excessive redundancy. Consequently, pruning can make the model more perfect to fit different datasets.

4.3 Experimental results on ImageNet

We prune ResNet50 and MobileNetV2 on ImageNet, record the performance of our pruned model and compare it with other methods. As shown in Tab. 3, our pruned model achieved the best performance. The FLOPs of ResNet50 dropped

Table 5: Results of ablation study. This table shows the results of VGG16 on CIFAR-10 and CIFAR-100 datasets according to different pruning criteria and pruning methods. Different dataset distributions are compared fairly under the same pruning rate.

	cifar10			cifar100		
criterion	L1(%)	NS(%)	RFC(%)	L1(%)	NS(%)	RFC(%)
GTP	90.56	92.64	92.25	70.90	71.12	71.25
DLP	93.12	93.25	93.38	73.49	73.55	73.82

by 54%, and the accuracy dropped by 0.29%, which was 0.60% lower than S-COP [35] and 0.91% lower than MetaPruning [24]. The FLOPs of MobileNetV2 dropped by 30% and the accuracy increased by 0.08%. This shows that the experimental results still demonstrate the effectiveness of our method on more complex datasets.

Moreover, we demonstrate the effectiveness of our method for wall-clock time speedup on ImageNet. As set by [20], we set the same compression ratio for each layer as a uniform baseline and measure latency using Pytorch on an NVIDIA RTX 2080 Ti GPU. As shown in Tab. 4, the results show that under the same FLOPs conditions, ResNet50 and ResNet34 pruned by our method can save 17% and 14% of hardware latency without notable accuracy loss. And the model is significantly better than the Uniform baseline in terms of both accuracy and latency.

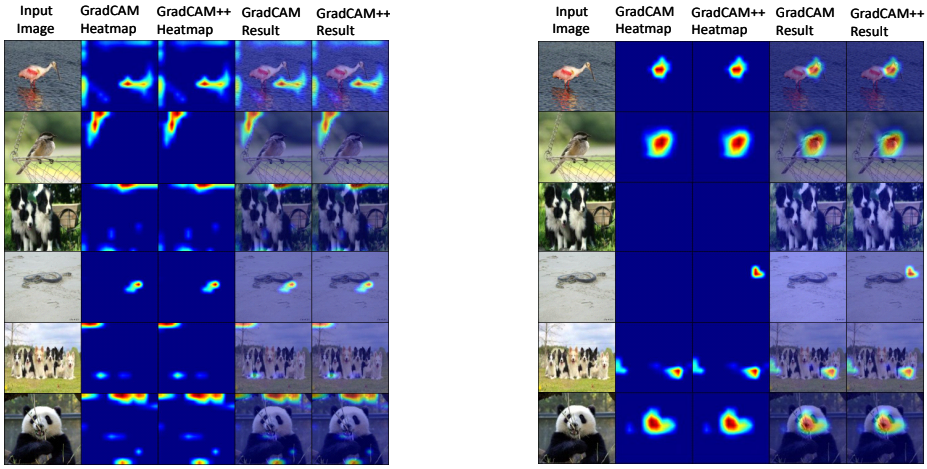
4.4 Ablation studies

In this subsection, we investigate the importance of RFC and discriminative layer pruning (called DLP) separately. The results of the ablation studies are summarized in Tab. 5. For the purpose of ablation experiments, we use the L1 norm [17] (denoted as L1) and the BN scale factor [25] (denoted as NS) to replace the RFC criterion to investigate its importance. We use global threshold pruning (denoted as GTP) to replace DLP to study its importance.

Distinguishing Layer Pruning (DLP): We first study our distinguishing layer pruning algorithm with the fixed criterion RFC. In the RFC column of Tab. 5, the accuracy of our method is significantly higher than the global threshold pruning of CIFAR-10 and CIFAR-100. This result indicates the effectiveness of our distinguishing layer pruning method. Moreover, we observe that the accuracy of other filter criteria on the DLP method is higher than the global threshold pruning. It shows that the filter criterion alone cannot be used for global comparison, so it is correct that we care about the difference in importance between different layers.

Receptive Field Criterion (RFC): We fixed the pruning method and then replaced RFC with L1 and BN as criteria during pruning. The resulting RFC for the DLP row in Tab. 5 has better accuracy than L1 and BN, showing that RFC can select filters that should be pruned more to achieve better pruning effect.

4.5 Visualization of filter selection



(a) Large H value. (b) Small H value.

Fig. 4: GradCAM and GradCAM++ visualization for filter selection.

Recall in Sec. 3.1 that H represents the distribution uniformity of the responses. We randomly select 6 animal images in ImageNet. Since the last layer learns the highest-level semantics, we use GradCAM [32] and GradCAM++ [2] to make an visualization analysis of the filter selection in the last layer of the pretrained VGG16, as shown in Fig. 4.

Large H value: As shown in Fig. 4a, the filters with large H value make high response to most classes, which indictes that the features learned by these filters are not distinguishable and contribute less to classification. Moreover, the response region of these filters with large H values is not on the object. A possible reason is that these filers tend to extract common information of the similar background in the images of different classes.

Small H value: As shown in Fig. 4b, filters with small H value do not respond to most classes, which shows that these features are distinguishable and contributive to classification. Meanwhile, the response regions generated by the filters with small H values are roughly located on the object, i.e., these features are relevant to the object. This phenomenon further confirms the reasonability of RFC.

4.6 Hyperparameters

Our DLRFC contains two hyperparameters: (1) balance coefficient γ controlling the relative proportion of soft change and hard change, as shown in Eq. (10); and

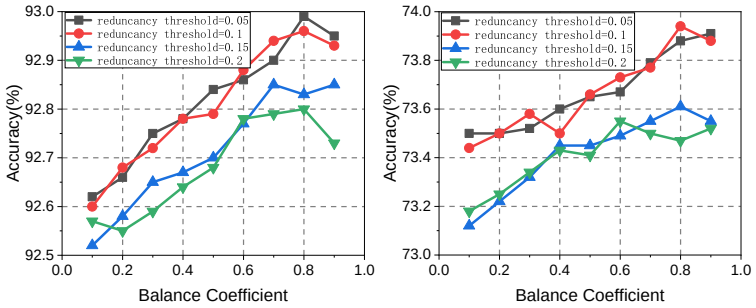


Fig. 5: Hyperparameter analysis of VGG16 on CIFAR10 (left) and CIFAR100 (right).

(2) The redundancy threshold ε controls the network change size and redundancy layer relationship, as shown in Eq. (11).

As shown in Fig. 5, when ε is small, the network selects redundant layers with strict criteria each time. Though this may lead to an increase in accuracy, the entire pruning process will take more time. When ε is too large, the network selects more layers as redundant layers, which will result in the existence of very important layers in the redundant layers and cause accuracy serious decline. So we need to choose ε within a reasonable range to ensure a balance between accuracy and speed. Although different ε may have different fluctuation curves, we found that when $\gamma = 0.8$, there will be better results, that is, we should probably pay more attention to hard changes. But $\gamma = 0.9$ shows that we can't focus too much on hard changes either.

4.7 Other studies

Feature discrimination angle effectiveness: The level of feature discrimination also represents the uniformity of the filter's response to all class, so we prune filters with high and low uniformity respectively. Figure 6 (left) shows a slight decrease in network performance when pruning the Unif-unimportant filter. However, pruning Unif-important filters wreaks havoc on network performance, suggesting that they are indeed able to measure filter importance. The figure also shows that NS-important has higher accuracy in stage 3 than NS-unimportant, suggesting that NS-important chooses less important filters than NS-unimportant, which is contradictory. It seems that NS cannot distinguish the importance of filters in stage 3 and stage 4, but Unif can distinguish well. The above shows that the idea of class uniformity can be used to judge the redundancy of the filter.

RFC measuring redundancy: Recall that the RFC measures redundancy from a new perspective, which is expressed as the H value, as shown in Eq. (5). The higher the H value, the more redundant the filters and the less they contribute to classification. We prune filters with different criteria: RFC, L1 norm,

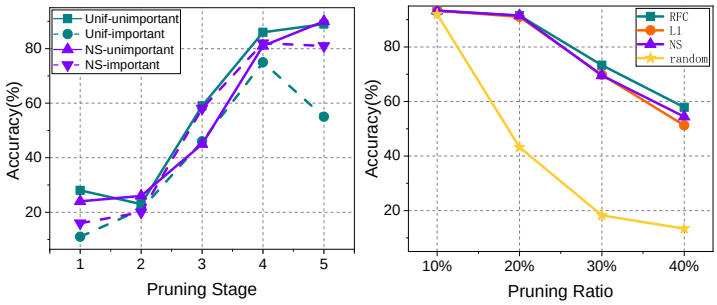


Fig. 6: Left: We divide VGG16 into 5 stages according to the pooling layer, stage x means that the filter of each layer in stage x is only pruned by half, and then the accuracy is obtained. unif-unimportant means to prune all filters that uniformity considers unimportant (that is, filters with high uniformity), unif-important means to prune filters that uniformity considers important, and the same is true for NS. Right: The accuracy of different criteria under the same network pruning structure.

BN, and random selection. After pruning the network by the above criteria, we directly record the accuracy. Figure 6 (right) shows the change in accuracy as the pruning rate increases. When pruning the RFC's filters, the accuracy drop was the smallest among these criteria, i.e. the RFC found the most redundant filters. This result further demonstrates the validity of the RFC guidelines.

5 Conclusion

In this paper, we propose a novel filter importance criterion named as Receptive Field Criterion from the feature discrimination. Our criterion scientifically measures the filter redundancy and effectively guides the pruning procedure. The distinguishing layer pruning based on RFC proposed by us can effectively consider the relative redundancy between the layers of the network. Extensive experiments conducted on CIFAR-10/100 and ImageNet show that our method achieves the state-of-the-art performance in some benchmarks. In the future, we will further explore the more intrinsic relationship between pruning and response class distribution.

Acknowledgement

This work is sponsored by the Zhejiang Provincial Natural Science Foundation of China (LZ22F020007, LGF20F020007), Major Research Plan of the National Natural Science Foundation of China (92167203), National Key R&D Program of China (2018YFB2100400), Natural Science Foundation of China (61902082, 61972357), and the project funded by China Postdoctoral Science Foundation under No.2022M713253.

References

1. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3319–3327 (2017)
2. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: IEEE Winter Conference on Applications of Computer Vision. pp. 839–847 (2018)
3. Ding, X., Ding, G., Guo, Y., Han, J.: Centripetal sgd for pruning very deep convolutional networks with complicated structure. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition. pp. 4943–4953 (2019)
4. Ding, X., Ding, G., Guo, Y., Han, J., Yan, C.: Approximated oracle filter pruning for destructive CNN width optimization. In: International Conference on Machine Learning. pp. 1607–1616 (2019)
5. Dong, Y., Bao, F., Su, H., Zhu, J.: Towards interpretable deep neural networks by leveraging adversarial examples (2017), *arXiv preprint arXiv:*1708.05493
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
7. Han, S., Mao, H., Dally, W.J.: Deep Compression: compressing deep neural networks with pruning, trained quantization and huffman coding. In: International Conference on Learning Representations (2016)
8. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. pp. 1135–1143 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
10. He, Y., Ding, Y., Liu, P., Zhu, L., Zhang, H., Yang, Y.: Learning filter pruning criteria for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2009–2018 (2020)
11. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4340–4349 (2019)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Advances in 28th Neural Information Processing Systems (2015)
13. Joo, D., Yi, E., Baek, S., Kim, J.: Linearly replaceable filters for deep network channel pruning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 8021–8029 (2021)
14. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009), pH.D Thesis in University of Toronto
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. pp. 1097—-1105 (2012)
16. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 4681–4690 (2017)

17. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient ConvNets (2016), *arXiv preprint arXiv:*1608.08710
18. Li, Y., Gu, S., Mayer, C., Gool, L.V., Timofte, R.: Group sparsity: The hinge between filter pruning and decomposition for network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8018–8027 (2020)
19. Li, Y., Lin, S., Liu, J., Ye, Q., Wang, M., Chao, F., Yang, F., Ma, J., Tian, Q., Ji, R.: Towards compact cnns via collaborative compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6438–6447 (2021)
20. Li, Y., Wu, W., Liu, Z., Zhang, C., Zhang, X., Yao, H., Yin, B.: Weight-dependent gates for differentiable neural network pruning. In: European Conference on Computer Vision. pp. 23–37 (2020)
21. Lin, M., Ji, R., Zhang, Y., Zhang, B., Wu, Y., Tian, Y.: Channel pruning via automatic structure search. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. pp. 673–679 (2020)
22. Lin, M., Ji, R., Zhang, Y., Zhang, B., Wu, Y., Tian, Y.: Hrank: filter pruning using high-rank feature map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1529–1538 (2020)
23. Liu, Y., Wentzlaff, D., Kung, S.: Rethinking class-discrimination based cnn channel pruning (2020), *arXiv preprint arXiv:*2004.14492
24. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, T.K.T., Sun, J.: Metapruning: meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3296–3305 (2019)
25. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2736–2744 (2017)
26. Luo, J.H., Wu, J.: Neural network pruning with residual-connections and limited-data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1458–1467 (2020)
27. Meng, F., Cheng, H., Li, K., Luo, H., Guo, X., Lu, G., Sun, X.: Pruning filter in filter. In: Advances in 33rd Neural Information Processing Systems (2020)
28. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. In: International Conference on Learning Representations (2017)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in 32nd Neural Information Processing Systems. pp. 8024–8035 (2019)
30. Ruan, X., Liu, Y., Li, B., Yuan, C., Hu, W.: DPFPS: Dynamic and progressive filter pruning for compressing convolutional neural networks from scratch. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 2495–2503 (2021)
31. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
32. 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/CVF International Conference on Computer Vision. pp. 618–626 (2017)

33. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (1948)
34. Tang, C., Lv, J., Chen, Y., Guo, J.: An angle-based method for measuring the semantic similarity between visual and textual features. In: Soft Computing. pp. 4041–4050 (2019)
35. Tang, Y., Wang, Y., Xu, Y., Tao, D., Xu, C., Xu, C., Xu, C.: SCOP: Scientific control for reliable neural network pruning (2020), *arXiv preprint arXiv:*2010.10732
36. Tian, H., Liu, B., Yuan, X.T., Liu, Q.: Meta-learning with network pruning. In: European Conference on Computer Vision. pp. 675–700 (2020)
37. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: CosFace: large margin cosine loss for deep face recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5265–5274 (2018)
38. Wang, W., Fu, C., Guo, J., Cai, D., , He, X.: COP: Customized deep model compression via regularized correlation-based filter-level pruning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 3785–3791 (2019)
39. Wang, Z., Li, C., Wang, X.: Convolutional neural network pruning with structural redundancy reduction. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 14913–14922 (2021)
40. Yu, R., Li, A., Chen, C.F., Lai, J.H., Morariu, V.I., Han, X., Gao, M., Lin, C.Y., Davis, L.S.: NISP: Pruning networks using neuron importance score propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9194–9203 (2018)
41. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. pp. 818–833 (2014)
42. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene CNNs. In: International Conference on Learning Representations (2015)
43. Zhuang, T., Zhang, Z., Huang, Y., Zeng, X., Shuang, K., Li, X.: Neuron-level structured pruning using polarization regularizer. In: Advances in 33rd Neural Information Processing Systems (2020)