Filter Pruning with Uniqueness Mechanism in the Frequency Domain for Efficient Neural Networks

Shuo Zhang [1], Mingqi Gao [2,3], Qiang Ni[1] and Jungong Han [4]

[1] *School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, United Kingdom;*
[2] *WMG Data Science, The University of Warwick, Coventry, CV4 7AL, United Kingdom;*
[3] *Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, 518055, China;*
[4] *Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, United Kingdom;*

Abstract

Filter pruning has drawn extensive attention due to its advantage in reducing computational costs and memory requirements of deep convolutional neural networks. However, most existing methods only prune filters based on their intrinsic properties or spatial feature maps, ignoring the correlation between filters. In this paper, we suggest the correlation is valuable and consider it from a novel view: the frequency domain. Specifically, we first transfer features to the frequency domain by Discrete Cosine Transform (DCT). Then, for each feature map, we compute a uniqueness score, which measures its probability of being replaced by others. This way allows to prune the filters corresponding to the low-uniqueness maps without significant performance degradation. Compared to the methods focusing on intrinsic properties, our proposed method introduces a more comprehensive criterion to prune filters, further improving the network compactness while preserving good performance. In addition, our method is more robust against noise than the spatial ones since the critical clues for pruning are more concentrated after DCT. Experimental results demonstrate the superiority of our method. To be specific, our method outperforms the baseline ResNet-56 by 0.38% on CIFAR-10 while reducing the floating-point operations (FLOPs) by 47.4%. In addition, a consistent improvement can be observed

when pruning the baseline ResNet-110: 0.23% performance increase and up to 71% FLOPs drop. Finally, on ImageNet, our method reduces the FLOPs of the baseline ResNet-50 by 48.7% with only 0.32% accuracy loss.

Keywords: Deep Learning; Model Compression; Computer Vision; Image Classification; Frequency-Domain Transformation

1. Introduction

The rapid progress in Convolutional Neural Networks (CNNs) has revolutionalized various computer vision tasks, e.g., image classification [1, 2, 3, 4], object detection [5, 6, 7], and segmentation [8, 9, 10]. To pursue better performance, most methods resort to complex network architectures. However, such implementations require heavy computations and memory footprints, limiting their applications in resource-limited systems (e.g., embedded or mobile devices). Therefore, how to compress complex networks while preserving competitive performance has recently drawn much attention.

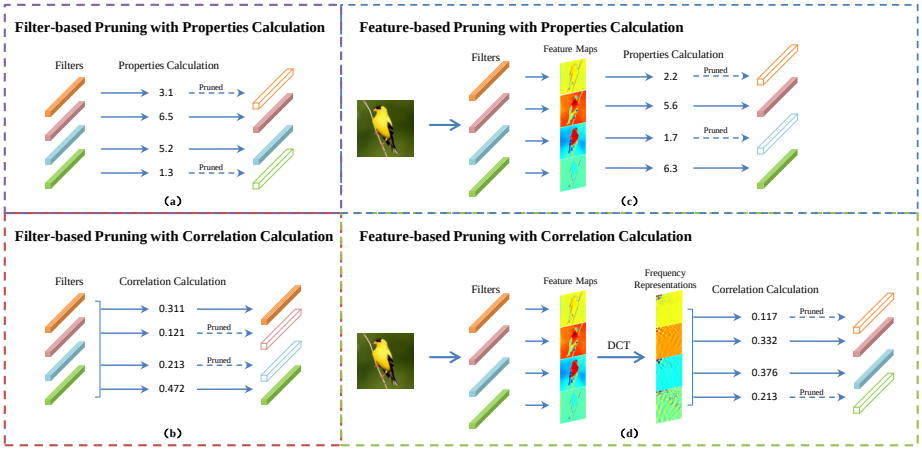

Figure 1: Four different pruning methods.

The existing techniques for network compression can be mainly grouped into four categories: knowledge distillation [11, 12], parameter binarization [13, 14],

compact network design [15, 16], and network pruning [17, 18]. Unlike the former three techniques, which build lightweight networks from scratch or change the parameter storage types, network pruning methods achieve compactness by finding and removing redundant parameters from the off-the-shelf networks. By doing so, the pruned networks can retain the most knowledge from the original networks while reducing the requirements for memory and computation resources. In general, there are two techniques to achieve network pruning: weight pruning [19, 20, 21] and filter pruning [22, 23, 24]. Weight pruning methods reduce network parameters by seeking a sparse weight matrix. However, the network slimmed down by those methods is unstructured as each parameter is likely to be removed, which disorders the integrity of the network components and makes it unable to achieve optimal efficiency via the Basic Linear Algebra Subprograms (BLAS) library [23]. By contrast, the latter methods prune all the related parameters once one filter is unimportant. Therefore, the pruned network still consists of integral components and can fully leverage the BLAS library for accelerated training and inference.

Due to its excellent efficiency, filter pruning has become prevalent in network compression. One intuitive attempt is to prune filters based on their importance measured by their intrinsic properties. For example, the method shown in Figure 1 (a) only considers the norm value to determine its importance. Specifically, it removes the filters whose norms are below a threshold. However, since the correlation between filters is ignored, one type of redundancy still exists in the pruned network: the filters that are prone to be replaced by others, even though their norm values are high enough. To address this issue, some methods [24, 25] take the correlation into account, as shown in Figure 1 (b). Despite achieving better performance, these methods mainly focus on filter weights but ignore more comprehensive information generated by the filters: feature maps.

Unlike the filter-based pruning methods, Lin *et al.* [26] consider the rank of each feature map to measure the corresponding filter's importance. As shown in Figure 1 (c), the method performs pruning with the properties derived from feature maps. With more comprehensive guidance, the feature-based methods

outperform the filter-based ones. However, there is still room to improve them further: (1) For each feature map, current methods only compute the rank based on its intrinsic properties, ignoring the correlation between maps; (2) For visual tasks, low-frequency channels are generally more informative than high-frequency ones [27], and almost all the energy in spatial features is concentrated in the low-frequency spectrum after DCT. Therefore, in contrast to the spatial domain, it seems easier and more efficient to find unimportant filters in the frequency domain.

Inspired by the above observations, we measure the filter importance based on the frequency-domain correlation between features. Figure 1 (d) illustrates the basic idea of our method. For each feature map, we compute the *uniqueness* to indicate whether it consists of unique enough information which is not easily replaced by others. Since the *uniqueness* comes from the interaction between feature maps, our method is more robust against the interference from intrinsic properties, e.g., norm values. With the uniqueness and the effective convergence of DCT, our proposed method can prune various networks with complex architectures. Specifically, we first transform the feature maps into the frequency domain by DCT. Then, we compute the uniqueness for each feature map and remove the filters corresponding to the low-uniqueness maps. The remaining filters form the final network with less complexity and performance preserved. Figure 2 illustrates the framework of our proposed method, and our main contributions are summarized as follows:

1) We propose uniqueness, a novel criterion for filter pruning. Unlike intrinsic properties of filters, uniqueness is measured from the correlation between feature maps. It implicitly indicates how much and how unique a feature map embeds the critical information. Therefore, a more comprehensive pruning strategy can be achieved.

2) We propose to determine to-be-pruned filters in the frequency domain. With the advantages of the frequency-domain operations, our proposed method can find and prune unimportant filters more efficiently, without

much interference as in the spatial domain.

3) The extensive experiments involving various network architectures and two different scales of image datasets demonstrate that our proposed method outperforms the state-of-the-art in accuracy and model compression.

The rest of this paper is organized as follows. Section 2 briefly introduces the representative works for weight/filter-based pruning, and the current frequency-domain works for vision tasks. Our proposed method is illustrated in detail in Section 3. Section 4 reports the experimental results on two different scales of datasets. The ablation study and the conclusion for our research are drawn in Section 5 and Section 6, respectively.

2. Related Works

2.1. Weight pruning

Weight pruning methods prune redundant activations and weights to achieve a more compact network. For instance, Hassibi and Stork [28] select and prune unimportant weights based on the second-order derivatives of an error function. Han *et al.* [29] first generate a sparse network by pruning the connections whose weights are below a threshold, and then retrain the network to improve the accuracy. Moreover, Deep Compression [19] combines network pruning with the quantization and Huffman encoding techniques to compress the network. Dong *et al.* [20] compress the weights of each layer independently based on the second derivative of the corresponding layer-wise errors. Xiao *et al.* [30] compress the model by optimizing a set of trainable and removable weights, which come from the product of the original weights and auxiliary parameters. Ding *et al.* [31] divide all the weights into two subsets through the Taylor expansion estimation. After this, the subsets are trained with different updating rules to achieve more sparse weights. Sanh *et al.* [21] propose a first-order weight pruning method,

which leverages the movement pruning to make the pre-trained model fine-tuning more adaptive. Lee *et al.* [32] propose a layer-adaptive and magnitude-based method to approximate the distortion of the pruned network.

2.2. Filter pruning

Filter pruning methods prune all the contained weights once one filter is redundant. Therefore, the pruned network can retain the integrity of basic components and better leverage the BLAS-based acceleration. For example, Li *et al.* [22] first measure the importance of each filter based on its absolute weight sum. Then, the compact network is achieved by pruning unimportant filters. Liu *et al.* [33] consider the scaling factors in batch norm layers as the pruning criterion and prune the filters corresponding to the low-factor feature channels. For the first time, He *et al.* [23] propose a soft pruning method that allows the pruned filters to be updated in the next training epoch. With the aid of LSTM, Zhong *et al.* [34] retrieve and prune potentially redundant filters. He *et al.* [25] prune alternative filters rather than unimportant ones by calculating their geometric median points. Li *et al.* [35] apply structural regularization to the out-in-channels from consecutive layers, removing more redundant channels with less precision loss. Lin *et al.* [26] suggest that the high-rank feature maps are more informative than the low-rank ones, and therefore they prune the channels corresponding to the low-rank feature maps. Tang *et al.* [36] generate knockoff features irrelevant to ground-truth labels to facilitate the redundant filter selection. Instead of pruning redundant channels, Lin *et al.* [37] perform an automatic structure search to build the optimal and compact model structure. Ning *et al.* [38] propose gradient-based optimization to calculate the pruning rate for each layer after training from scratch.

2.3. Frequency-Domain Analysis

Frequency information has recently attracted attention in the computer vision community due to its unique data representation. For example, Gueguen

et al. [39] incorporate frequency information into the decoding stage to accelerate the model training. Besides, Ehrlich and Davis [40] run ResNet in the frequency domain to improve the inference speed. Xu *et al.* [27] employ DCT transformation to replace the original spatial sub-sampling approach to better preserve the image information in the pre-processing stage. In addition to better performance, these works reveal that low-frequency feature channels are usually more informative than high-frequency ones in visual reasoning tasks. The following works further extend the advantages of the frequency information to multiple applications. To be specific, Qin *et al.* [41] propose channel attention based on the features in the frequency domain and achieve excellent performance in classification, detection, and segmentation tasks. Jiang *et al.* [42] develop a frequency-domain loss for image reconstruction and synthesis, optimized by weighting different frequencies. Cai *et al.* [43] use frequency information to enhance the image generation process. For model compression, Chen *et al.* [44] first observe that the trained weights are usually low-frequency and they prune the filters containing high-frequency components using hashing techniques. Differently, Liu *et al.* [45] prune filters dynamically by converting convolution operations into DCT multiplications, which is different from the pruning method in [44] and ours. Although the above methods employ frequency to refine pruning, they ignore the properties and correlations of filters and corresponding feature maps in the frequency domain. Our method considers correlations of feature maps in the frequency domain since the feature maps contain richer information than filters. However, this paper finds they are valuable for pruning and bring superior performance to our proposed method.

3. Methodology

3.1. Preliminaries

Given a CNN model with L layers, let $\boldsymbol{W}^l \in \mathbb{R}^{c^l \times c^{l-1} \times k \times k}$ be the filter tensor of the l-th convolutional layer, where k, c^l, and c^{l-1} represent the kernel size, output channel, and input channel of the filter, respectively. The filter-based

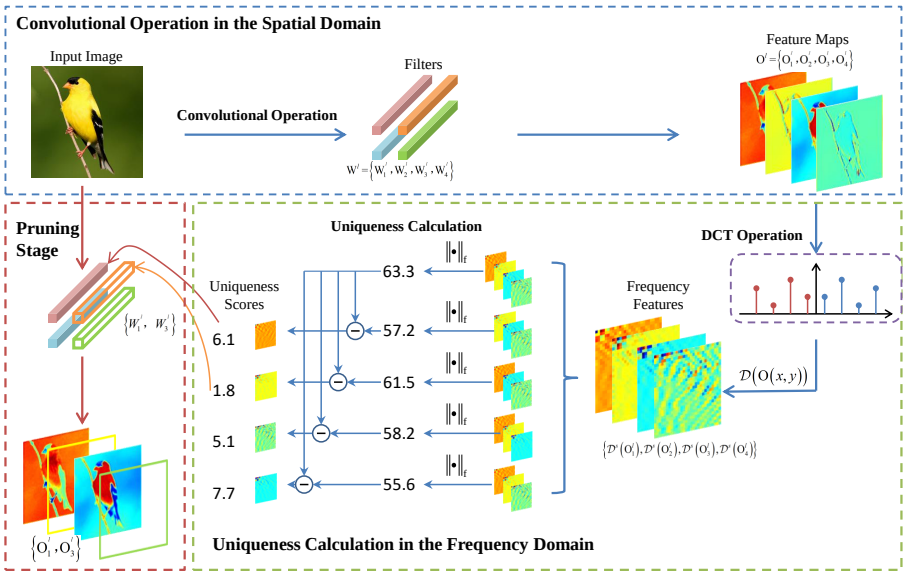

Figure 2: The framework of our proposed method. At first, we encode features in the spatial domain, where each feature map comes from the convolution between the input image and one filter (shown in the blue box, the top region). Then, these features are transferred to the frequency domain by DCT, which are in turn used to compute the uniqueness score for each feature map (shown in the green box, the bottom-right region). Finally, the filters corresponding to the low-uniqueness feature maps will be pruned and will not participate in the subsequent computation (shown in the red box, the bottom-left region). Best viewed in color.

pruning methods generally formulate the objective function as follows:

$$\begin{aligned} \min_{k_l} &= \sum_{l=1}^{L} \sum_{j=1}^{c^l} k_l E(\boldsymbol{W}_j^l) \\ \text{s.t. } & 0 < \|k_l\|_0 \leq (1-\alpha^l)c^l \end{aligned} , \tag{1}$$

where k_l is a list of indices, indicating the filters to prune in the l-th layer. The amount of the pruned filters $\|k_l\|_0$ is limited by both the compression ratio α^l and the initial number c^l. $E(\cdot)$ estimates the intrinsic property of each filter $\boldsymbol{W}_j^l$, which implicitly measures the importance of the filter and, therefore, can serve as a criterion for pruning. In more recent pruning methods, however,

feature maps have gradually dominated the criterion measurement since they embed more comprehensive information regarding filters and input data. In this case, the above objective function can be rewritten as the following form:

$$\begin{aligned} \min_{k_l} &= \sum_{l=1}^{L} \sum_{j=1}^{c^l} k_l E(\boldsymbol{I}^l * \boldsymbol{W}_j^l), \\ \text{s.t. } & 0 < \|k_l\|_0 \leq (1-\alpha^l)c^l \end{aligned} \tag{2}$$

where $\boldsymbol{I}^l$ is the input tensor to the l-th layer and $*$ denotes the convolution operation. So far, several prior works [23, 26, 25] have been implemented to prune unimportant filters through the above optimization processes. The main focus of these methods usually lies in designing the function $E(\cdot)$ for importance measurement.

3.2. Uniqueness Calculation in the Image Domain

Unlike the previous works, which focus on spatial operations and prune unimportant filters using their intrinsic properties or feature maps, we perform pruning from a novel view: the frequency domain. Specifically, we design an effective mechanism to measure the uniqueness of each filter, based on the correlation between its corresponding feature map and others in the frequency domain. With the uniqueness, we can infer if one filter can be replaced by others. Compared with intrinsic properties (e.g., norm values), it better reflects the filter redundancy. Therefore, the over-pruning or under-pruning due to inaccurate redundancy measurement can be effectively alleviated.

We first introduce the uniqueness in the spatial domain. For simplicity, we assume $\boldsymbol{O}^l = \boldsymbol{I}^l * \boldsymbol{W}^l \in \mathbb{R}^{c^l \times h^l \times w^l}$ as the feature maps generated by the l-th convolutional layer, ignoring the activations and biases. c^l, h^l, and w^l indicate the channel, height, and width dimensions, respectively. For each feature map $\boldsymbol{O}_j^l \in \mathbb{R}^{h^l \times w^l}$, its uniqueness can be defined as:

$$E(\boldsymbol{O}_j^l) = \|\boldsymbol{O}^l\|_f - \|\boldsymbol{O}_{j^*}^l\|_f, \tag{3}$$

where $\|\cdot\|_f$ computes the Frobenius norm. We reshape feature maps $\boldsymbol{O}^l$ and $\boldsymbol{O}_{j^*}^l$ to $c^l \times h^l w^l$ to meet the requirement for the input dimensions. $\boldsymbol{O}_{j^*}^l$ is ini-

tialized from $\boldsymbol{O}^l$ and all its values in the j-th row are set to zero after dimension transformation.

3.3. Uniqueness Calculation in the Frequency Domain

Given the efficient energy convergence of DCT, it is easier to determine the uniqueness of each feature map in the frequency domain than in the spatial domain. Therefore, we transform the feature maps into the frequency domain through DCT for uniqueness computation, which is defined by reformulating Eqn. 3 as the following form:

$$E(\mathcal{D}(\boldsymbol{O}_j^l)) = \|\mathcal{D}(\boldsymbol{O}^l)\|_f - \|\mathcal{D}(\boldsymbol{O}_{j^*}^l)\|_f, \tag{4}$$

where $\mathcal{D}(\cdot)$ denotes the DCT operation. For each position (x, y) in the j-th feature map $\boldsymbol{O}_j^l$, its counterpart in the frequency domain is obtained as below:

$$\mathcal{D}(\boldsymbol{O}_j^l(x,y)) = \frac{s}{\sqrt{h^l w^l}} \sum_{x=0}^{h^l-1} \sum_{y=0}^{v-1} \boldsymbol{O}_j^l(x,y) \cos(\frac{\pi}{h^l} u(x+\frac{1}{2})) \cos(\frac{\pi}{w^l} v(y+\frac{1}{2})), \tag{5}$$

where (u, v) indicates the location in the frequency domain. s is a scaling factor and conditioned on (u, v):

$$\begin{cases} s = 1, \text{ if } (u,v) = (0,0) \\ s = 2, \text{ if } (u,v) \neq (0,0). \end{cases} \tag{6}$$

After DCT, most low-frequency information will be concentrated in a local region in the frequency domain, which comprises the most informative knowledge of the input data. Therefore, such a local area can support robust and more efficient data processing compared with the whole region. We further reformulate Eqn. 4 to measure the uniqueness of the j-th filter in the l-th layer:

$$\mathrm{E}(\mathcal{D}^s(\boldsymbol{O}_j^l)) = \|\mathcal{D}^s(\boldsymbol{O}^l)\|_f - \|\mathcal{D}^s(\boldsymbol{O}_{j^*}^l)\|_f, \tag{7}$$

where $\mathcal{D}^s(\cdot)$ only focuses on the frequency-domain features from the concentrated local region after DCT. For the j-th filter in the l-th layer, its importance is evaluated by $E(\mathcal{D}^s(\boldsymbol{O}_j^l))$, the correlation of its corresponding feature

map with others in the frequency domain. In this way, we can prune filters more accurately and efficiently if we replace the original criterion ($E(\boldsymbol{W}_j^l)$) in Eqn. 1 with $E(\mathcal{D}^s(\boldsymbol{O}_j^l))$. After obtaining the uniqueness score of each filter, the number of removable filters is determined based on the pruning rate for each layer. Therefore, we prune the filter according to its uniqueness score, from small to large, until the number of pruning requirements is reached.

4. Experiments

In this section, we provide extensive experimental results and analysis to illustrate the superior performance of our algorithm on two different scales of datasets: CIFAR-10 [46] and ImageNet [47].

4.1. Experimental Settings

4.1.1. Baselines and Datasets

To evaluate our proposed method comprehensively, we apply it to various CNN architectures and compare the pruning performance with state-of-the-art methods on different datasets. Specifically, we first consider two ResNet architectures [3] (ResNet-56 and ResNet-110) and VGG-16 [2] on a small benchmark dataset: CIFAR-10 [46]. Then, we conduct experiments with ResNet-50 on a larger dataset (ImageNet [47]) for further analysis.

4.1.2. Evaluation Metrics

Similar to existing methods, we utilize the number of floating-point operations (FLOPs) and parameters (Params) to measure the complexity and size of the pruned CNN models, respectively. As for the accuracy evaluation, we compute models' Top-1 accuracy on CIFAR-10 and Top-1 and Top-5 accuracy on ImageNet. It is worth noting that ΔTop-1(%) and ΔTop-5(%) represent the difference in Top-1 and Top-5 accuracies before and after pruning. Params(%)$\downarrow$ and FLOPs(%)$\downarrow$ indicate the drops (percentage) in parameters and FLOPs between the pruned and baseline models. The results of competitors shown in Tables 1, 2, 3, and 4 are reported from their original publications.

4.1.3. Implementation Details

During the fine-tuning stage on CIFAR-10, we set the number of epochs as 300 and batch size as 128. We also consider Stochastic Gradient Descent (SGD) as the optimizer with an initial learning rate, weight decay, and momentum of 0.01, 0.05, and 0.9, respectively. As for the experiment setting on ImageNet, we retrain the pruned network for 180 epochs with a batch size of 256. The initial learning rate, weight decay, and momentum are set as 0.1, 0.0001, and 0.9, respectively. We use a similar setting of HRank [26] for choosing the pruning rate of each layer to make a fair comparison. Since the sensitivity and importance of different convolution layers are various [51], some layers are pruned under different pruning rates while others share the same pruning rate during the pruning stage. All the experiments are implemented with PyTorch on 4 NVIDIA GTX1080Ti GPUs.

4.2. Results and Analysis

4.2.1. Results on CIFAR-10

In this section, we firstly deploy ResNet-56 to verify the performance of our proposed method on CIFAR-10. As shown in Table 1, our method obtains the best performance compared to state-of-the-art methods on both moderate and deep compression. Specifically, the Top-1 accuracy of our method is 0.49% higher than LSTM [51] with almost the same FLOPs reduction. Moreover, our results are even better than the baseline model (93.88% vs. 93.26%). Compared with NISP [49], DNAL [48], and GAL [50], we obtain a better performance in Top-1 accuracy, parameter reduction, and FLOPs reduction. Although CP [52], SEP [23], and FPGM [25] reduce more FLOPs than ours, their Top-1 accuracy is much lower than ours (-2.52%, -1.95%, -1.28%). For deep compression, our method achieves the best Top-1 accuracy while obtaining the maximum parameter reduction and FLOPs reduction compared with GAL [50] and DNAL [48]. Besides, our method achieves better quantitative results (92.48% vs. 90.72%) even though we share a similar model compression with Hrank [26].

Table 1: Comparison of pruned Resnet-56 on CIFAR-10.

Method	**Top1(%)**	**ΔTop1(%)**	**Params(%)↓**	**FLOPs(%)↓**
DNAL [48]	94.15 → 93.76	-0.39	22.3	25.1
DNAL [48]	94.15 → 93.75	-0.40	33.8	30.6
NISP [49]	94.15 → 93.01	-0.14	42.4	35.5
GAL [50]	93.26 → 93.38	+0.12	11.8	37.6
Ours	**93.26 → 93.88**	**+0.62**	**42.8**	**47.4**
LSTM [51]	93.04 → 92.93	-0.11	N/A	47.5
CP [52]	92.80 → 90.90	-1.90	N/A	50.0
FPSST [53]	93.57 → 93.28	-0.29	N/A	51.1
SFP [23]	93.59 → 92.26	-1.33	N/A	52.6
FPGM [25]	93.59 → 92.93	-0.66	N/A	52.6
GAL [50]	93.26 → 91.58	-1.68	65.9	60.2
DNAL [48]	94.15 → 93.20	-0.95	70.5	70.5
Hrank [26]	93.26 → 90.72	-2.54	68.1	74.1
Ours	**93.26 → 92.48**	**-0.78**	**71.8**	72.3

Analogous to ResNet-56, our method achieves excellent performance on ResNet-110. From Table 2, it can be seen that we obtain the best accuracy improvement with the most significant parameter and FLOPs reductions, compared to SEP [23], Rethink [54], HRank [26] and GAL [50]. Notably, Experiments on the CIFAR-10 dataset using ResNet-110 might encounter the overfitting since it contains too many parameters for such a small dataset. Although FPGM [25] obtains a similar FLOPs reduction as ours, it refines the accuracy by +0.06% only, which is much lower than ours (+1.01%). In addition to moderate compression, we observe that our method with deep compression outperforms the existing competitors in both Top-1 accuracy and FLOPs reduction, further illustrating the consistent superiority of our proposed method.

Table 2: Comparison of pruned Resnet-110 on CIFAR-10.

Method	**Top1(%)**	**ΔTop1(%)**	**Params(%)↓**	**FLOPs(%)↓**
SFP [23]	93.68 → 93.38	-0.30	N/A	40.8
Rethink [54]	93.77 → 93.70	-0.07	N/A	40.8
Hrank [26]	93.50 → 94.23	+0.73	39.4	41.2
GAL [50]	93.50 → 92.55	-0.95	44.8	48.5
FPSST [53]	93.70 → 93.62	-0.08	N/A	50.6
Ours	**93.50 → 94.51**	**+1.01**	**48.3**	52.1
FPGM [25]	93.68 → 93.74	+0.06	N/A	52.3
FalCon [55]	93.68 → 93.79	+0.11	N/A	60.3
FalCon [55]	93.68 → 93.63	-0.05	N/A	62.3
CCPrune [56]	94.11 → 93.36	-0.75	N/A	68.0
Hrank [26]	93.50 → 92.65	-0.85	68.7	68.6
Ours	**93.50 → 93.73**	**+0.23**	68.3	**71.6**

Table 3: Comparison of pruned VGG-16 on CIFAR-10.

Method	**Top1(%)**	**ΔTop1(%)**	**Params(%)↓**	**FLOPs(%)↓**
PFEC [22]	93.58 → 93.28	-0.30	N/A	34.2
FPGM [25]	93.58 → 93.23	-0.35	N/A	35.9
SSS [57]	93.96 → 93.02	-0.94	73.8	41.6
GAL [50]	93.96 → 93.42	-0.54	82.2	45.2
RL-MCTS [58]	93.51 → 93.90	+0.39	N/A	45.5
FalCon [55]	93.32 → 93.63	+0.31	N/A	50.0
Hrank [26]	93.96 → 93.43	-0.53	82.9	53.5
FalCon [55]	93.32 → 91.92	-1.40	N/A	67.3
Ours	**93.96 → 93.76**	**-0.20**	80.8	73.7
Hrank [26]	93.96 → 91.23	-2.73	92.0	76.5
Ours	**93.96 → 93.61**	**-0.35**	**84.8**	**76.7**

Finally, we employ VGG-16 to verify the performance of our proposed method on CIFAR-10, as shown in Table 3. Not surprisingly, our method achieves bet-

ter performance with deep compression. Compared to PEFC [22], FPMG [25], SSS [57], GAL [50], and Hrank [26], our method has apparent advantages in Top-1 accuracy and FLOPs compression ratio. To be specific, our method achieves up to 73.7% FLOPs reduction with only a 0.2% accuracy loss (from 93.96% to 93.76%). Although RL-MCTS [58] and FalCon [55] further improve the Top-1 accuracy, their FLOPs reductions (45.5% and 50.0%) are much lower than ours (73.7%). Moreover, our method achieves fewer Top-1 accuracy loss than HRank [26](−0.35% vs.−2.73%) even though we yield approximately the same FLOPs reduction. These results illustrate the effectiveness of our proposed method.

4.2.2. Results on ImageNet

Table 4: Comparison of pruned Resnet-50 on ImageNet.

Method	**Top1(%)**	**ΔTop1(%)**	**Top5(%)**	**ΔTop5(%)**	**Params(%)↓**	**FLOPs(%)↓**
SFP [23]	76.15 → 74.61	-1.54	92.87 → 92.06	-0.81	N/A	41.8
BNFI [59]	76.33 → 75.47	-0.86	N/A	N/A	N/A	42.8
LSTM [51]	76.12 → 75.00	-1.12	93.00 → 92.67	-0.33	N/A	43.0
HRank [26]	76.15 → 74.98	-1.17	92.87 → 92.33	-0.54	36.6	43.7
CPS [60]	76.15 → 75.59	-0.46	N/A	N/A	N/A	44.3
RL-MCTS [58]	77.34 → 76.80	-0.54	93.27 → 93.00	-0.27	N/A	46.1
Hinge [61]	76.10 → 74.70	-1.40	N/A	N/A	N/A	46.5
Ours	**76.15 → 75.83**	**-0.32**	**92.87 → 92.76**	**-0.11**	**44.2**	48.7
CFP [62]	75.30 → 73.40	-1.90	92.20 → 91.40	-0.80	N/A	49.6
DSA [38]	76.02 → 74.69	-1.33	N/A	-0.80	N/A	50.0
Autopruner [63]	76.15 → 74.76	-1.39	92.87 → 92.15	-0.72	N/A	51.2
GDP [64]	75.13 → 71.89	-3.24	92.30 → 90.71	-1.59	N/A	51.3
BNFI [59]	76.33 → 75.02	-1.29	N/A	N/A	N/A	52.6
FalCon [55]	75.83 → 74.59	-1.24	92.78 → 92.51	-0.27	N/A	53.5
FPGM [25]	76.15 → 74.83	-1.32	92.87 → 92.32	-0.55	N/A	53.5
ABCPruner [37]	76.01 → 73.86	-2.15	92.96 → 91.69	-1.27	N/A	54.3
RL-MCTS [58]	77.34 → 76.46	-0.88	93.27 → 92.83	-0.44	N/A	55.0
Hrank [26]	76.15 → 71.98	-2.73	92.87 → 92.68	-1.86	46.0	62.1
Ours	**76.15 → 74.80**	**-1.35**	**92.87 → 92.39**	**-0.48**	**56.7**	62.8
FalCon [55]	75.83 → 73.55	-2.28	92.78 → 91.99	-0.79	N/A	63.4
Hrank [26]	76.15 → 69.10	-7.05	92.87 → 89.58	-3.29	67.5	76.0
Ours	**76.15 → 73.18**	**-2.97**	**92.87 → 91.32**	**-1.55**	**68.6**	**76.7**

In this section, we verify the performance of our method on a large-scale dataset (ImageNet). Table 4 shows that our method achieves competitive performance compared to state-of-the-art methods. For moderate compression, our approach outperforms HRank [26], LSTM [51], BNFI [59], and SFP [23] in parameter reduction, FLOPs reduction, and Top-1 accuracy. Besides, our method shares similar FLOPs reduction with Hinge [61] and RL-MCTS [58] but obtains less accuracy degradation (both in Top-1 and Top-5). Although the pruning rate of our method is slightly lower than CFP [62], DSA [38], Autopruner [63], GDP [64], BNFI [59], FalCon [55], and FPGM [25], it can retain much more performance of the original model than them.

For deep compression, the Top-1 accuracy of our method exceeds Hrank [26] and FalCon [55] by 2.82% and 0.93% respectively under similar FLOPs reduction. Notably, compared to Hrank [26], which reduces the FLOPs up to 76%, our method achieves slightly more parameters and FLOPs drop but with much less accuracy degradation (-2.97% vs.-7.05%). From the quantitative results, our pruning method can not only outperform the state-of-the-art methods in moderate compression but also maintain higher accuracy even in deep compression.

5. Ablation Study

5.1. The Comparison of High and Low Representations

Frequency information is able to measure the pixel value change in images. Specifically, low-frequency and high-frequency signals reflect the image regions with smooth and sharp differences, respectively. In general, low-frequency features are more informative than high-frequency ones in visual tasks [27]. To analyze the impact of different frequencies of information in pruning, we conduct experiments with various networks and compare the results on CIFAR-10. The "low-frequency" that is shown in the blue bar shows that only the low-frequency information is extracted and used for generating uniqueness scores of feature maps after the DCT operation. Similarly, the green bar representing

"high-frequency" indicates that only high-frequency information is collected and utilized to compute uniqueness scores for feature maps following the DCT process. The "spatial domain" indicates that the uniqueness scores of feature maps are generated by the information of feature maps without the DCT operation.

Figure 3 shows the Top-1 accuracy achieved by different networks and different frequencies of features. Under the same pruning rate, it is observed that pruning with low-frequency features is better than with high-frequency ones, which means the low-frequency components can effectively measure the uniqueness of feature maps. In addition, we follow Eqn. 3 to prune filters based on their spatial uniqueness scores, whose Top-1 accuracy is in-between the methods with low-frequency and high-frequency features. The comparison results show that the low-frequency components in our method indeed consist of more valuable information than others, validating the effectiveness of our frequency-based strategy.

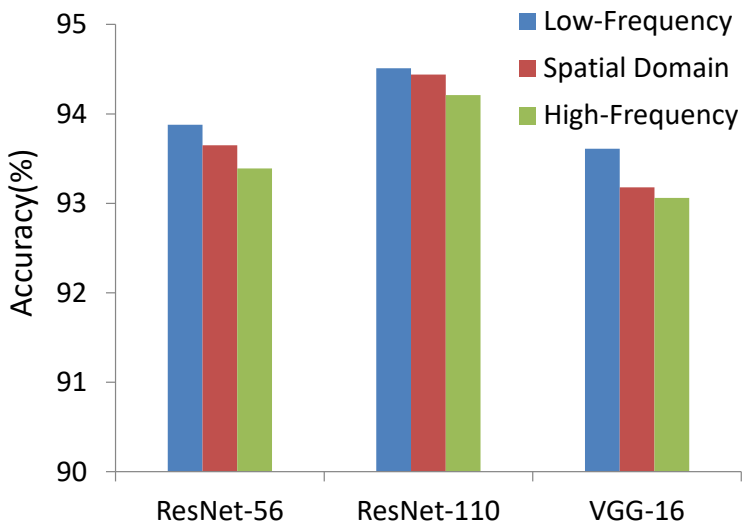

Figure 3: Top-1 accuracy of various models pruned with different frequencies/domains of features.

5.2. Feature Map Visualization

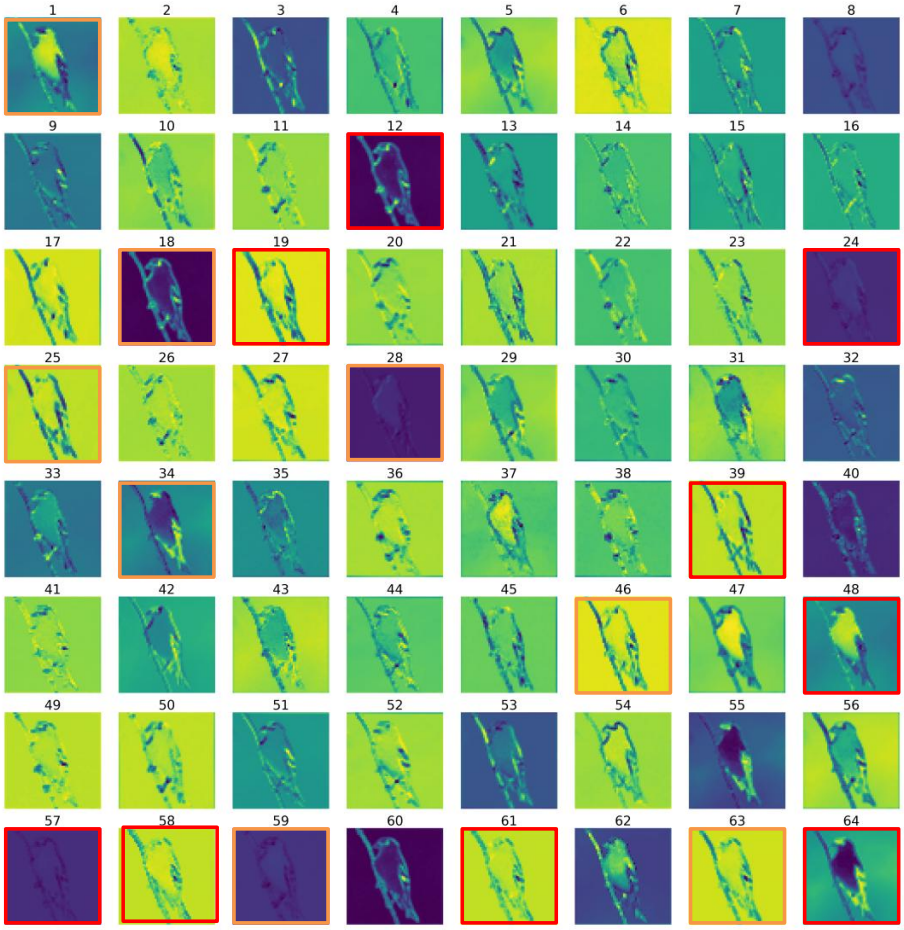

Figure 4: The visualization of feature maps, generated by ResNet-50 (block1, conv1), where red boxes highlight the pruned feature maps by our method, orange boxes indicate the feature maps (retained) which can replace the pruned ones.

To explore the effectiveness of our uniqueness computation, we visualize one layer of feature maps before and after pruning, as shown in Figure 4. Specifically, we select the features generated by the first convolution layer in the first block of ResNet-50, trained on ImageNet. Under the pruning rate of 0.15, our method removes the feature maps with indices 12, 19, 24, 39, 48, 57, 58, 61, and 64

(marked by red boxes) due to their low uniqueness scores. Although the removed maps cannot contribute to the subsequent computation, it will not degrade the final results considerably since the remaining maps can easily replace them. From Figure 4, we can find the substitute for each removed map and show them as a pair with the format (removed map index - substitute map index with **bold** text): (12-**18**), (19-**46**), (24-**28**), (39-**25**), (48-**1**), (57-**59**), (58-**63**), (61-**63**), and (64-**34**). These substitutes are marked by orange boxes in Figure 4 to highlight their relationships to the removed ones. To sum up, the visualized results show that the feature maps pruned by our method are indeed replaceable and unimportant, further validating our uniqueness-based strategy.

5.3. The Difference in Selecting Redundant Filters in the Spatial and the Frequency Domains

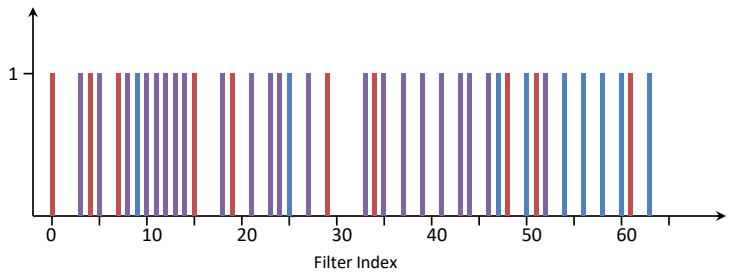

Figure 5: The difference in selecting redundant filters in the frequency and spatial domains, respectively. The red bar indicates that the index of the redundant filter is selected in the spatial domain only, while the blue bar illustrates that the index of the filter needs to be pruned in the frequency domain only. The purple bar shows the index of the redundant filter chosen to be removed both in the frequency domain and the spatial domain.

To explore the difference in choosing redundant filters in the spatial and frequency domains, we view the filter indices in the 40th layer of the ResNet-56 model. In Figure 5, it can be seen that most redundant filters are selected in both the frequency domain and the spatial domain, which are shown as purple bars. Notably, the pruning method in the frequency domain tends to select large indexes of filters as redundant ones, while some small indexes of

filters are picked to remove by the pruning method in the spatial domain. For determining 32 redundant filters, the spatial and frequency domain approaches have ten different selections in this layer, which makes a difference in model performance.

5.4. The Effectiveness between Various Pruning Methods

To further evaluate the effectiveness of our proposed method, we study the accuracy-pruning rate trade-off curve for the ResNet-50 model on ImageNet dataset. The results are shown in Figure 6:

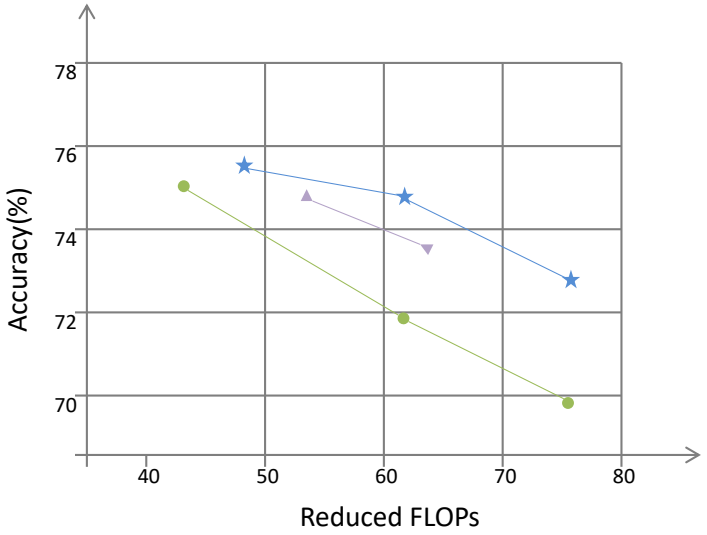

Figure 6: The comparison of the accuracy-pruning rate trade-off curve for various pruning methods using the ResNet-50 model on the ImageNet dataset.

In this figure, the blue star represents the accuracy of our method under the specific FLOPs reduction, whereas the green circle and purple triangle indicate the accuracy of HRank and FalCon with given FLOPs reduction, respectively. As we can see that our proposed method achieves the best performance throughout various pruning rates.

6. Conclusion

This paper presented a novel pruning method, which operates mainly in the frequency domain and computes uniqueness as the critical criteria for removing filters. Unlike the previous spatial methods, we further transform the encoded features into the frequency domain by DCT to mine more valuable and concentrated information from the input data. After this, we compute uniqueness scores from each feature map, considering both the properties within and across maps. The network pruning is achieved by removing the filters corresponding to the low-uniqueness maps, which can be easily replaced by others. This way, the proposed method can effectively reduce the network complexity while maintaining its performance to the largest extent. We evaluated our method with various network architectures on two different scales of datasets. The experimental results showed that our method achieves superior performance compared to the state-of-the-art approaches.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[4] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28.

[7] B. Singh, M. Najibi, L. S. Davis, Sniper: Efficient multi-scale training, Advances in neural information processing systems 31.

[8] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[9] Z. Zhu, M. Xu, S. Bai, T. Huang, X. Bai, Asymmetric non-local neural networks for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 593–602.

[10] M. Gao, F. Zheng, J. J. Yu, C. Shan, G. Ding, J. Han, Deep learning for video object segmentation: a review, Artificial Intelligence Review (2022) 1–75.

[11] A. Polino, R. Pascanu, D. Alistarh, Model compression via distillation and quantization, arXiv preprint arXiv:1802.05668.

[12] N. Aghli, E. Ribeiro, Combining weight pruning and knowledge distillation for cnn compression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3191–3198.

[13] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: Imagenet classification using binary convolutional neural networks, in: European conference on computer vision, Springer, 2016, pp. 525–542.

[14] X. Wang, B. Zhang, C. Li, R. Ji, J. Han, X. Cao, J. Liu, Modulated convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 840–848.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

[17] S. Guo, Y. Wang, Q. Li, J. Yan, Dmcp: Differentiable markov channel pruning for neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1539–1547.

[18] B. Li, B. Wu, J. Su, G. Wang, Eagleeye: Fast sub-net evaluation for efficient neural network pruning, in: European conference on computer vision, Springer, 2020, pp. 639–654.

[19] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149.

[20] X. Dong, S. Chen, S. Pan, Learning to prune deep neural networks via layer-wise optimal brain surgeon, Advances in Neural Information Processing Systems 30.

[21] V. Sanh, T. Wolf, A. Rush, Movement pruning: Adaptive sparsity by fine-tuning, Advances in Neural Information Processing Systems 33 (2020) 20378–20389.

[22] H. Li, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, Pruning filters for efficient convnets, arXiv preprint arXiv:1608.08710.

[23] Y. He, G. Kang, X. Dong, Y. Fu, Y. Yang, Soft filter pruning for accelerating deep convolutional neural networks, arXiv preprint arXiv:1808.06866.

[24] S. Zhang, G. Wu, J. Gu, J. Han, Pruning convolutional neural networks with an attention mechanism for remote sensing image classification, Electronics 9 (8) (2020) 1209.

[25] Y. He, P. Liu, Z. Wang, Z. Hu, Y. Yang, Filter pruning via geometric median for deep convolutional neural networks acceleration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4340–4349.

[26] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, L. Shao, Hrank: Filter pruning using high-rank feature map, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1529–1538.

[27] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, F. Ren, Learning in the frequency domain, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1740–1749.

[28] B. Hassibi, D. Stork, Second order derivatives for network pruning: Optimal brain surgeon, Advances in neural information processing systems 5.

[29] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, Advances in neural information processing systems 28.

[30] X. Xiao, Z. Wang, S. Rajasekaran, Autoprune: Automatic network pruning by regularizing auxiliary parameters, Advances in neural information processing systems 32.

[31] X. Ding, X. Zhou, Y. Guo, J. Han, J. Liu, et al., Global sparse momentum sgd for pruning very deep neural networks, Advances in Neural Information Processing Systems 32.

[32] J. Lee, S. Park, S. Mo, S. Ahn, J. Shin, Layer-adaptive sparsity for the magnitude-based pruning, arXiv preprint arXiv:2010.07611.

[33] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2736–2744.

[34] J. Zhong, G. Ding, Y. Guo, J. Han, B. Wang, Where to prune: Using lstm to guide end-to-end pruning., in: IJCAI, 2018, pp. 3205–3211.

[35] J. Li, Q. Qi, J. Wang, C. Ge, Y. Li, Z. Yue, H. Sun, Oicsr: Out-in-channel sparsity regularization for compact deep neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7046–7055.

[36] Y. Tang, Y. Wang, Y. Xu, D. Tao, C. Xu, C. Xu, C. Xu, Scop: Scientific control for reliable neural network pruning, Advances in Neural Information Processing Systems 33 (2020) 10936–10947.

[37] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, Y. Tian, Channel pruning via automatic structure search, arXiv preprint arXiv:2001.08565.

[38] X. Ning, T. Zhao, W. Li, P. Lei, Y. Wang, H. Yang, Dsa: More efficient budgeted pruning via differentiable sparsity allocation, in: European Conference on Computer Vision, Springer, 2020, pp. 592–607.

[39] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, J. Yosinski, Faster neural networks straight from jpeg, Advances in Neural Information Processing Systems 31.

[40] M. Ehrlich, L. S. Davis, Deep residual learning in the jpeg transform domain, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3484–3493.

[41] Z. Qin, P. Zhang, F. Wu, X. Li, Fcanet: Frequency channel attention networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 783–792.

[42] L. Jiang, B. Dai, W. Wu, C. C. Loy, Focal frequency loss for image reconstruction and synthesis, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13919–13929.

[43] M. Cai, H. Zhang, H. Huang, Q. Geng, Y. Li, G. Huang, Frequency domain image translation: More photo-realistic, better identity-preserving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13930–13940.

[44] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, Y. Chen, Compressing convolutional neural networks in the frequency domain, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1475–1484.

[45] Z. Liu, J. Xu, X. Peng, R. Xiong, Frequency-domain dynamic pruning for convolutional neural networks, Advances in neural information processing systems 31.

[46] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[48] Q. Guo, X.-J. Wu, J. Kittler, Z. Feng, Differentiable neural architecture learning for efficient neural networks, Pattern Recognition (2022) 108448.

[49] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V. I. Morariu, X. Han, M. Gao, C.-Y. Lin, L. S. Davis, Nisp: Pruning networks using neuron importance score propagation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9194–9203.

[50] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, D. Doermann, Towards optimal structured cnn pruning via generative adversarial learn-

ing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2790–2799.

[51] G. Ding, S. Zhang, Z. Jia, J. Zhong, J. Han, Where to prune: Using lstm to guide data-dependent soft pruning, IEEE Transactions on Image Processing 30 (2020) 293–304.

[52] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 1389–1397.

[53] Y. Lian, P. Peng, W. Xu, Filter pruning via separation of sparsity search and model training, Neurocomputing 462 (2021) 185–194.

[54] Z. Liu, M. Sun, T. Zhou, G. Huang, T. Darrell, Rethinking the value of network pruning, arXiv preprint arXiv:1810.05270.

[55] Z. Xu, F. Yu, C. Liu, Z. Wu, H. Wang, X. Chen, Falcon: Fine-grained feature map sparsity computing with decomposed convolutions for inference optimization, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 350–360.

[56] Y. Chen, X. Wen, Y. Zhang, W. Shi, Ccprune: Collaborative channel pruning for learning compact convolutional networks, Neurocomputing 451 (2021) 35–45.

[57] Z. Huang, N. Wang, Data-driven sparse structure selection for deep neural networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 304–320.

[58] Z. Wang, C. Li, Channel pruning via lookahead search guided reinforcement learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2029–2040.

[59] J. Oh, H. Kim, S. Baik, C. Hong, K. M. Lee, Batch normalization tells you which filter is important, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2645–2654.

[60] C. Yang, H. Liu, Channel pruning based on convolutional neural network sensitivity, Neurocomputing 507 (2022) 97–106.

[61] Y. Li, S. Gu, C. Mayer, L. V. Gool, R. Timofte, 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, 2020, pp. 8018–8027.

[62] P. Singh, V. K. Verma, P. Rai, V. Namboodiri, Leveraging filter correlations for deep model compression, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 835–844.

[63] J.-H. Luo, J. Wu, Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference, Pattern Recognition 107 (2020) 107461.

[64] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, B. Zhang, Accelerating convolutional networks via global & dynamic filter pruning., in: IJCAI, Vol. 2, 2018, p. 8.