CFDP: Common Frequency Domain Pruning

Samir Khaki, Weihan Luo
University of Toronto
Toronto, Canada
{samir.khaki, weihan.luo}@mail.utoronto.ca

Abstract

As the saying goes, sometimes less is more – and when it comes to neural networks, that couldn't be more true. Enter pruning, the art of selectively trimming away unnecessary parts of a network to create a more streamlined, efficient architecture. In this paper, we introduce a novel end-to-end pipeline for model pruning via the frequency domain. This work aims to shed light on the interoperability of intermediate model outputs and their significance beyond the spatial domain. Our method, dubbed ***C****ommon* ***F****requency* ***D****omain* ***P****runing (CFDP) aims to extrapolate common frequency characteristics defined over the feature maps to rank the individual channels of a layer based on their level of importance in learning the representation. By harnessing the power of CFDP, we have achieved state-of-the-art results on CIFAR-10 with GoogLeNet reaching an accuracy of* 95.25%*, that is,* $+0.2\%$ *from the original model. We also outperform all benchmarks and match the original model's performance on ImageNet, using only* 55% *of the trainable parameters and* 60% *of the FLOPs. In addition to notable performances, models produced via CFDP exhibit robustness to a variety of configurations including pruning from untrained neural architectures, and resistance to adversarial attacks. The implementation code can be found at* https://github.com/Skhaki18/CFDP.

1. Introduction

Convolutional Neural Networks (CNNs) have emerged as a popular technology in computer vision, enabling breakthroughs in many fields including image classification [44], segmentation [36], and detection [11]. This surge in interest led to the development of modern-day architectures that incorporate novel features including skip-connections [17], concatenations [23], and inception modules [44] that vastly outperform traditional models. Unfortunately, these new innovations have given rise to a significantly increased model size and energy consumption thus limiting the global community's ability to effectively leverage these powerful tools in various domains. This poses a significant challenge to the widescale adoption of newer CNN architectures in the real world where applications generally enforce energy constraints and real-time processing. As a result, several solutions were proposed to tackle this issue including quantization [5], low-rank factorization [8], knowledge distillation [21], and pruning [27].

Network pruning has emerged as a particularly promising approach over various domains [15] and can be further divided into two categories: *unstructured pruning* and *structured pruning*. *Unstructured pruning* aims to reduce the total number of trainable parameters in a model by masking individual elements of the weight matrix, effectively obtaining a sparse representation [9, 10]. The major drawback of this method is that in order to leverage the acceleration and compression from sparse matrix computations, dedicated hardware/libraries must be provided, hence limiting the scope of application [14]. On the flip side, *structured pruning* [25, 48, 52] doesn't suffer from the same deficiency as the entire filter (or equivalently channel) is being removed from the layer, thus resulting in faster inferencing and training time as well as lower memory consumption [34]. Despite the advantages of structured pruning, there still exists the open problem of developing an effective saliency metric that can rank the individual channels of a layer based on their level of importance in learning the representation. Several works have attempted to solve this problem using either the model weights or feature maps to determine their respective importance [15, 27, 30]. One work in particular, FDNP [35], leveraged the frequency domain interpretation of the convolutional operator to develop its own saliency metric. Despite the compression brought about by these methods, they still suffer from either reduced performance or additional labor costs [30].

Inspired by these works, we introduce our novel pruning metric that leverages a combination of information in the frequency and spatial domains to achieve competitive performance on state-of-the-art (SOTA) benchmarks [19, 20, 24, 27, 30, 32, 37, 46, 50] without the intensive labor costs of iteration. These benchmarks were selected based

on consistent choices of datasets and performance metrics. Our method, ***C**ommon **F**requency **D**omain **P**runing*, dubbed CFDP, was benchmarked for image classification on the CIFAR-10 [26] and ImageNet [7] datasets across a variety of architectures. Additionally, we conduct exhaustive testing through ablation studies to examine the robustness of our method. The results of our experiments demonstrate the dominant performance of CFDP in terms of **accuracy**, **acceleration**, and **robustness** across different settings. In summary, our main contributions are threefold:

- We propose a novel pruning metric rooted in frequency-based traditional signal processing techniques to more effectively estimate the performance of each channel in a CNN.
- Our novel pruning metric achieves state-of-the-art performance across all benchmarks, including ImageNet, with a high cross-architecture generalization and superior results over an extended range of pruning.
- Our framework for pruning offers increased robustness including the ability to generalize well on untrained neural networks and produce models that are more resistant to adversarial attacks.

2. Related work

2.1. Model Compression

Structured Pruning.

Structured pruning aims to find a subset of a CNN architecture that contains fewer filters (herein referred to as channels) while maintaining comparable accuracy. As opposed to unstructured pruning, structured pruning doesn't suffer from the problem of producing sparse matrices, which allows it to effectively utilize the BLAS library. Previous works [27, 28, 30] have explored metrics for evaluating the importance of filters via their corresponding L^1-Norm, average ranks, or sparsity respectively. Alternatively, one work explored combining pruning into a training pipeline with Soft Pruning [18], where pruned filters had a possibility of being updated during training. Finally, another work, named Filter Pruning Geometric Median [19], found that the ideal filters satisfied large norm deviation and small minimum norm.

Frequency Domain Representation. It is well-known that there is spatial redundancy within most filters in a CNN [35]. Consequently, recent works have started to explore training, feature extraction, and pruning in the frequency domain. For instance, [39] trained CNNs directly in the frequency domain, which significantly accelerated the training time. Other works [13, 47] use the Discrete Cosine Transform (DCT) on the YCbCr color space of the original input image for feature extraction. In [47], the authors showed that learning in the frequency domain achieved superior image information preservation in the pre-processing stage as opposed to its spatial domain counterpart. The authors in [45] used the K-means algorithm to extract similar components between filters in the frequency domain. Finally, [4] extended filter pruning to 3D CNNs to eliminate the temporal redundancy using the DCT.

Discussion. The collection of art shows a diversified pool of research wherein concepts from other domains are applied in an effort of improving computer vision models. In the domain of pruning, current methods suffer from large labor costs due to additional hyperparameter tunings and more complex training pipelines; as a result, they tend to exhibit inferior acceleration/performance. Our approach leverages information from feature maps by jointly considering its spatial and frequency information, leading to better acceleration, reduced labor costs, and more robust performance.

3. Network Pruning via the Frequency Domain

3.1. Notations

Let's assume a standard CNN model contains N convolutional layers indexed with i where $i \in \{0, .., N-1\}$. For the ith layer, we define the weight parameter by $W_i \in \mathbb{R}^{D_i \times C_i \times K_i \times K_i}$, where D_i and C_i represent the input and output channels of the ith convolutional layer respectively, while K_i is the corresponding kernel size. Under the definition of filter pruning, we can extend the notation to define two sets: P_i and S_i to represent the indices of pruned and saved output channels for layer i. Thus we have $|P_i| = T_i$ and $|S_i| = C_i - T_i$ where T_i is the number of channels pruned in layer i. For formality, we can state that there are a finite number of channels per layer, of which each channel can either be pruned or saved – analytically we have $P_i \cap S_i = \emptyset$. For the sake of simplicity, we omit the batch dimension in defining the feature maps. We define the intermediate feature maps for layer i on a single image as $\mathcal{F}_i \in \mathbb{R}^{C_i \times M_i \times M_i}$, where M_i represents the width and height of the square feature map. To simplify the notation, $\mathcal{F}_{i,j}$ references the intermediate feature map from the jth channel in the ith layer. Finally, with respect to our proposed methodology, we define $D(\cdot)$ as the discrete cosine transform operator to convert any 2D time-domain signal into the frequency domain. In particular, when converting a feature map through the $D(\cdot)$ operator, it retains its dimensional configuration but exists in a different space of analysis. We denote the frequency representation of a feature map as $\tilde{\mathcal{F}}_i \in \mathbb{R}^{C_i \times M_i \times M_i}$.

3.2. CFDP

In this work, we introduce a novel pruning metric defined over the intermediate feature maps of a CNN. Our motiva-

tion for defining the metric on the feature maps stems from the idea that features maps incorporate an additional data-centric component into the pruning algorithm making it better tuned for the specific model and dataset, as similarly seen in recent works [33, 51, 52].

We begin by denoting the saliency metric, a measure of importance, as $\mathcal{L}$, wherein we can formulate the pruning optimization problem, on the basis of a single feature map as:

$$\min_{\mathbb{1}_{i,j}} \sum_{i=0}^{N-1} \sum_{j=0}^{C_i-1} \mathbb{1}_{i,j}[\mathcal{L}(\mathcal{F}_{i,j})] \tag{1}$$

$$s.t. \sum_{j=0}^{C_i-1} \mathbb{1}_{i,j} = T_i \tag{2}$$

where $\mathbb{1}_{i,j}$ is an indicator function that is 1 if $j \in P_i$, else 0.

Before proceeding in resolving the optimization, we note that several works [22, 29] have shown the importance of averaging saliency metrics over a large batch of images. We can integrate this into our optimization problem by introducing the expectation value over the set of images. Specifically, we arrive at the following notation:

$$\mathcal{L}(\mathcal{F}_{i,j}) \equiv \mathbb{E}_{b\sim P(b)}[\mathcal{L}(\mathcal{F}_{i,j}(b))] \tag{3}$$

where the input to $\mathcal{L}$ is of dimension $[1 \times 1 \times M_i \times M_i]$, and it represents the feature map derived from image b sampled from the distribution $P(b)$ for layer i and channel j. For easier computation, we define $P(b)$ to be an empirically determined distribution of the data. Finally, solving this non-convex minimization problem can be executed by pruning all T_i filters in P_i for each layer i. In order to assign channel j into a particular set, we use the saliency metric $\mathcal{L}(\cdot)$ on its respective feature map. Designing this saliency metric is an open problem, and in this paper, we introduce a novel approach to the design of this metric by incorporating traditional signal processing techniques. Specifically, we begin with understanding feature map representations in the frequency domain.

Analzying Feature Map Information in the Frequency Domain.

As seen in recent works [51], there has been a growing trend of deriving channel-wise performance correlation with its spatial output - the respective feature map. We argue, however, that there exists more information, by transforming this map into the frequency domain.

Before analyzing the frequency transformations, we investigate some preprocessing to augment the information concentration in our feature maps. Specifically, we incorporate an additional level of data-centric design into the preprocessing of our frequency representation. Particularly, since the datasets used in this paper include CIFAR10 [26], and ImageNet [7]), we are guaranteed that the images are derived from a subset of natural images. In particular, natural images tend to exhibit the majority of their information in the lower frequencies of the spectrum [6]. Ultimately we leverage this information into the design of a suitable filter to isolate the main information from background noise in the feature maps. Since the majority of information is in the low-frequency range, we use a Gaussian filter, where $g(\cdot)$ is the Gaussian filter operator applied element-wise on the input.

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4}$$

Next, we investigated the two common methods to transform any N-dimensional (discrete) signal from the spatial domain into the frequency space: Discrete Fourier Transform (FFT/DFT) [2] and the Discrete Cosine Transform (DCT) [1]. In this paper, we chose to implement our transformations via the DCT for two reasons: **(1)** The DFT follows an assumption of signal periodicity wherein it uses a sawtooth basis function for analysis resulting in a strong presence of high-frequency components – however in the cases of natural images, as are those in our datasets, periodicity is too strict of an assumption to describe the distribution of data [40]. **(2)** The DFT occupies twice as much space complexity by storing the phase (imaginary) components of a signal which aren't necessarily valuable for determining the level of information in an image [40]. In particular, image compaction is the process of storing the most important information in an image, and it is predominantly run using the DCT [3, 16, 43]. For these reasons, we have selected the DCT for our domain transformation. We can now apply the DCT to the task of determining a layer-wise ranking of different channels.

In order to transform this representation into the frequency domain, we subdivide the feature map into a set of non-overlapping patches G with each patch being of size $B_s \times B_s$ and $|G| = \frac{2 \cdot M_i}{B_s}$. From our experiments, we have determined that $B_s = 4$ works best, refer to the ablation on block size Section 4.4.3. If however, $M_i < B_s$ which may be the case in very deep networks, we simply treat the entire feature map as one patch. For each patch p in the set G, we define the DCT pixel-to-pixel encoding from spatial into frequency domain: $(x, y) \rightarrow (u, v)$, considering the Gaussian filter operator, as:

$$\tilde{p}_{u,v} = D(p_{x,y}) \tag{5}$$

$$= \sum_{x}^{B_s} \sum_{y}^{B_s} cos(\frac{(\pi u)(2x+1)}{2B_s}) cos(\frac{(\pi v)(2y+1)}{2B_s}) g(p_{x,y}) \tag{6}$$

Given the processed frequency domain representation, we can now introduce the main component to our saliency metric $\mathcal{L}_{Fq}$. This metric accounts for two phenomena in the frequency domain: the magnitude of representation, and the distribution of frequencies.

Magnitude of Representation. In the frequency domain, we can assign the magnitude of information contained within the frequency spectrum as the spectral energy of the DCT coefficients.

$$Spectral_{i,j} = \left[\sum_{w=0}^{M_i-1} \sum_{h=0}^{M_i-1} \left[\tilde{\mathcal{F}}_{i,j,w,h} \right]^2 \right]^{\frac{1}{2}} \tag{7}$$

where $Spectral_{i,j}$ represents the spectral energy of the jth channel in the ith layer. Thus, we are able to measure the magnitude of representation for each channel based on the feature map's energy.

Distribution of Frequencies. Each sub-block in the DCT representation, contains various frequencies, increasing as we move diagonally down the block. Recognizing that the majority of the information is in the upper triangle (low frequencies) and that the DC component (upper left index) has the property of dominating the spectral energy, thus it's important to scale the energetic magnitude by how well the information is spread over the relevant frequencies [1]. Let's define the mean value of our frequency representation for the ith layer and jth channel as $\mu_{i,j}$ We can calculate this effective spread using:

$$Dist_{i,j} = \frac{1}{M_i^2} \sum_{w=0}^{M_i-1} \sum_{h=0}^{M_i-1} \mathbb{1}_{[\tilde{\mathcal{F}}_{i,j,w,h} \geq \mu_{i,j}]} \tag{8}$$

To justify the introduction of energy distributions into the ranking metric, we empirically test its performance in Table 5. From these findings, we can conclude that although the magnitude of representation is a core metric for information in the domain, the distribution of frequencies can further augment the performance.

Frequency Based Saliency Metric. Jointly considering both magnitude and distribution, we define our frequency-based saliency metric as:

$$\mathcal{L}_{Fq} = Dist(\tilde{\mathcal{F}}_i) \cdot Spectral(\tilde{\mathcal{F}}_i) \tag{9}$$

3.3. Challenges with the Frequency only Metric

One challenge that was presented when comparing the frequency metrics was the closeness in the ranking of the channels. In particular, referencing Figure 1, we can see that when sorted by $\mathcal{L}_{Fq}$ the overall score assignment between neighboring indices was hard to distinguish as there are limited natural breaking points. Thus, in order to reinforce the rankings, but add enough disturbance to accurately determine the final few channels in the saved set S_i, we introduce a regularizer to sort channels with very similar frequency representations. In particular, we introduce a common measure of spatial energy, dubbed $\mathcal{L}_{Sp}$, derived from the spatial representation of the feature maps. The idea, with this regularizer, is that the majority of the channels will be selected based on the frequency domain and that in order to distinguish the final few channels that would be saved or pruned, the spatial domain will introduce enough perturbation to make the correct choice. From Figure 1, it is easy to tell that the main ranking metric is the frequency domain, while the spatial domain has added sufficient separation between close frequencies.

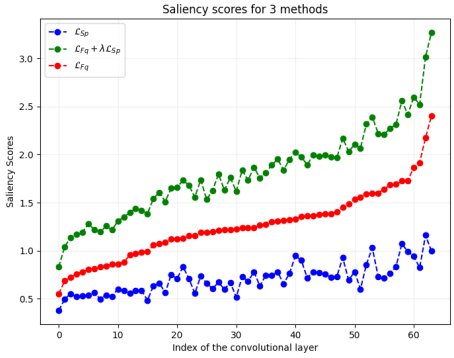

Figure 1. This figure shows the performance score for the first layer in VGG-16-BN on CIFAR-10, using 3 methods: (1) $\mathcal{L}_{Sp}$ only, (2) $\mathcal{L}_{Fq}$ only, and (3) $\mathcal{L}_{Fq} + \lambda \mathcal{L}_{Sp}$ (our combined metric, where λ is introduced in Section 3.5). The channel indices have been sorted by $\mathcal{L}_{Fq}$ in order to illustrate the motivation behind incorporating spatial regularization

3.4. Space Domain Regularization

Referencing Figure 1, we can see that the scores $\mathcal{L}_{Sp}$ operates on a lower scale than that of the $\mathcal{L}_{Fq}$ and this is because we want the primary driver of the ranking to be derived from the frequency domain, and simply use the spatial domain for regularization. In particular, we define ranking the metric in the spatial domain as:

$$Spatial_{i,j} = \left[\sum_{w=0}^{M_i-1} \sum_{h=0}^{M_i-1} \left[\mathcal{F}_{i,j,w,h} \right]^2 \right]^{\frac{1}{2}} \tag{10}$$

where $Spatial_{i,j}$ represents the spatial magnitude of the feature maps in the time domain. Past works including [34] have explored using only spatial norms for channel ranking, however, we find this method to be inadequate as it doesn't accurately compare the encoded information in the feature maps. Further, we augment our motivation with an evaluation of each component of our ranking metric in the ablation Section 4.4.1, wherein we show that it is due to the combination of our frequency metrics and a regularizer that we are

able to truly improve the results. We can express the spatial metric as a function of the keyword $Spatial$ operating on the spatial feature map.

$$\mathcal{L}_{Sp} = Spatial(\mathcal{F}_i) \tag{11}$$

3.5. CFDP Ranking Metric

Finally, we conclude with the full formulation of our ranking metric defined over the feature maps to be:

$$\mathcal{L} = \mathcal{L}_{Fq} + \lambda \mathcal{L}_{Sp} \tag{12}$$

where λ is a hyperparameter to modulate the regularization power of $\mathcal{L}_{Sp}$ We discover the value of $\lambda = 0.03$ empirically, through the ablation study in Section 4.4.2.

3.6. CFDP Framework

The CFDP framework for a single layer is visualized in Figure 2. An initial batch of images B is sent into the network, where intermediate feature maps are extracted and converted into the inputs for both $\mathcal{L}_{Sp}$ and $\mathcal{L}_{Fq}$. Next, the regularizing coefficient λ is applied on the spatial loss via the gain block and combined with the scores from the frequency domain to generate the two sets P_i and C_i with threshold T_i. This sets reconstruct the appropriate layer $layer_i^{'} \in \mathbb{R}^{|S_i| \times M_i \times M_i}$.

4. Experiments

4.1. Implentation Details

Datasets. We evaluate our performance on the CIFAR-10 [26] and ImageNet [7] dastasets. CIFAR-10 is a 10-class dataset containing 60K 32x32 color images with 50K training and 10K testing images. ImageNet contains over 1.2 million training images with 50K validation images across 1000 classes with a resolution of 224x224.

Evaluation metrics. Following the current SOTA, we accurately benchmark our performance using three common metrics: Top-1%, Params, and FLOPs. Top-1% is an indicator of how well our model is able to discriminate classes on the specific dataset, while Params and FLOPs evaluate the model size and computational footprint respectively. For ImageNet, due to the difficulty of the dataset, we include Top-5% following common SOTA benchmarks.

Configurations. For fair comparison, we adopt the same training configurations as HRank [30], a leading SOTA method, for each architecture. We use a newer implementation of HRank for benchmarking dubbed HRankPlus [30] as it vastly outperforms the preceding paper. We used a learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.005, following standard configurations, as well as the commonly scheduled learning rate decay of 0.1. CIFAR-10 pruning and training were done on an NVIDIA P5000 GPU, while ImageNet was done on an Ampere A100 GPU.

Algorithm 1 CFDP Pruning Framework

Input Variables & Functions:
Pre-Trained Weights θ
Saved and Pruned Sets for the model $S = \{\}, P = \{\}$
▷ Network Initialization Function
$Init(weights, savedChannels, prunedChannels)$
▷ Training Pipeline given network
$Train(network)$
Output: Fine Tuned Pruned Model

Network $\leftarrow Init(\theta, \{C_0, ..., C_N\}, \emptyset)$
▷ Compute forward pass of batch B
FM = Network(B)
for i in N **do**
  for f in FM **do**
    ▷ Calculate Batch averaged Proxy Ranking
    ▷ Leverage Equations 9, 11, 12
    $\mathcal{L}_i \leftarrow \frac{1}{|B|}\left(\mathcal{L}_{Fq}(f) + \lambda \mathcal{L}_{Sp}(f)\right)$
  end for
  ▷ Sort Channels in layer i based on the Proxy Metric
  $C_i \leftarrow$ argsort($\mathcal{L}_i$)
  $S[i] \leftarrow C_i[:T_i]$
  $P[i] \leftarrow C_i[T_i:]$
end for
▷ Re-initialize model using pruned and saved channel sets
Network $\leftarrow Init(\theta, S, P)$
$Train$(Network)

4.2. CIFAR10 results

VGG-16-BN. We use a variant of VGG-16-BN that was fine-tuned on CIFAR10 with results reported in Table 1. The table is divided into two parts: in the upper part, we include common SOTA methods, while the lower part focuses on the comparison between our method and Hrank on the same pruning configuration. Compared with importance-based methods such as L^1 and FPGM, CFDP is shown to perform better in terms of accuracy (94.10% vs 93.40% vs 94.00%) and in terms of acceleration (58.1% vs 34.3% vs 35.9% for FLOPs and 2.76M vs 5.40M for Params). Given that these two methods operate only using properties in the spatial domain (L^1-norm, L^2-norm), the results suggest that incorporating properties from both the spatial and the frequency domain lead to better performance with greater acceleration. Finally, CFDP achieves the best validation accuracy when compared with HRank under the same configurations by surpassing their validation accuracy by 0.37%.

ResNet-56. Table 2 shows different pruning algorithms on ResNet-56. Compared with L^1, CFDP achieves better Top-1 Accuracy (93.97% vs 93.06%), with an increase in FLOPs reduction (28.0% vs 27.6%) and a decrease in the number of parameters (0.66M vs 0.73M). Under the same

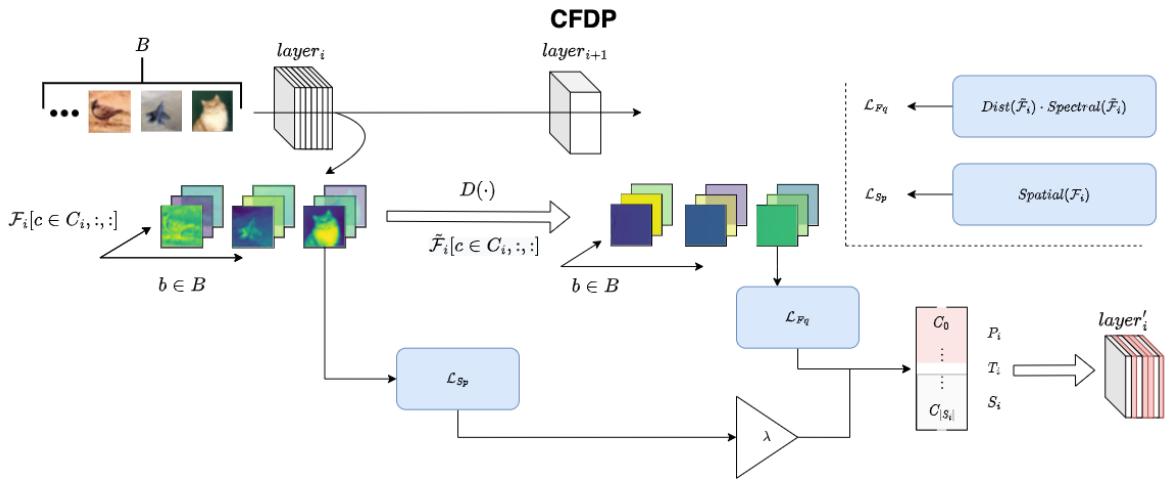

Figure 2. This figure shows CFDP method computation for $layer_i$ based on feature maps $\mathcal{F}_i$ and the associate transformation into $layer_i^{'}$ (the pruned layer). Further information discussed in Section 3.6

Model	Top-1%	FLOPs (↓)	Params (↓)
VGG-16-BN	93.96	313.73M(0.0%)	14.98M(0.0%)
L^1 [27]	93.40	206.00M(34.3%)	5.40M(64.0%)
SSS [24]	93.02	183.13M(41.6%)	3.93M(73.8%)
Zhao *et al.* [50]	93.18	190.00M(39.1%)	3.92M(73.3%)
GAL-0.05 [32]	92.03	189.49M(39.6%)	3.36M(77.6%)
GAL-0.1 [32]	90.78	171.89M(45.2%)	2.67M(82.2%)
FPGM [19]	94.00	201.10M(35.9%)	–
Wang *et al.* [46]	93.63	156.86M(50.0%)	–
HRank [30]	93.73	131.17M(58.1%)	2.76M(81.6%)
CFDP	**94.10**	**131.17M**(58.1%)	**2.76M**(81.6%)

Table 1. Pruning results of VGG-16-BN on CIFAR-10.

Model	Top-1%	FLOPs (↓)	Params (↓)
ResNet-56	93.26	125.49M(0.0%)	0.85M(0.0%)
L^1 [27]	93.06	90.90M(27.6%)	0.73M(14.1%)
He *et al.* [20]	90.80	62.00M(50.6%)	–
NISP [49]	93.01	81.00M(35.5%)	0.49M(42.4%)
GAL-0.6 [32]	92.98	78.30M(37.6%)	0.75M(11.8%)
GAL-0.8 [32]	90.36	49.44M(60.2%)	0.29M(65.9%)
FPGM [19]	93.49	59.44M(52.6%)	–
Wang *et al.* [46]	93.05	62.75M(50.0%)	–
HRank [30]	93.85	90.35M(28.0%)	0.66M(22.3%)
CFDP	**93.97**	**90.35M**(28.0%)	**0.66M**(22.3%)

Table 2. Pruning results of ResNet-56 on CIFAR-10.

Model	Top-1%	FLOPs(PR)	Parameters(PR)
GoogLeNet	95.05	1.52B(0.0%)	6.15M(0.0%)
Random	94.54	0.96B(36.8%)	3.58M(41.8%)
L^1 [27]	94.54	1.02B(32.9%)	3.51M(42.9%)
Hrank [30]	94.53	0.69B(54.9%)	2.74M(55.4%)
GAL-ApoZ [22]	92.11	0.76B(50.0%)	2.85M(53.7%)
GAL-0.05 [32]	93.93	0.94B(38.2%)	3.12M(49.3%)
Hrank [30]	95.04	0.65B(57.2%)	2.86M(53.5%)
CFDP	**95.25**	**0.65B**(**57.2**%)	**2.86M**(**53.5**%)

Table 3. Pruning results of GoogLeNet on CIFAR-10.

Model	Top-1%	Top-5%	FLOPs	Params
ResNet-50 [37]	**76.15**	**92.87**	4.09B	25.50M
He *et al.* [20]	72.30	90.80	2.73B	–
ThiNet-50 [37]	68.42	88.30	1.10B	8.66M
SSS-26 [24]	71.82	90.79	2.33B	15.60M
SSS-32 [24]	74.18	91.91	2.82B	18.60M
GDP-0.5 [31]	69.58	90.14	1.57B	–
GDP-0.6 [31]	71.19	90.71	1.88B	–
GAL-0.5 [32]	71.95	90.94	2.33B	21.20M
GAL-1 [32]	69.88	89.75	1.58B	14.67M
GAL-0.5-joint [32]	71.80	90.82	1.84B	19.31M
GAL-1-joint [32]	69.31	89.12	1.11B	10.21M
FPGM [19]	75.91	92.63	2.36B	–
HRank [30]	75.56	92.63	2.26B	15.09M
CFDP	**76.10**	**92.93**	2.26B	15.09M

Table 4. Pruning Results of ResNet-50 on ImageNet.

configurations, we outperform Hrank with a Top-1 Accuracy of 93.97% vs 93.85%. This shows our algorithm's robustness towards skip connections-based architectures.

GoogLeNet. GoogLeNet results are shown in Table 3. Our method greatly surpasses all methods in the upper part of the table, including the original model (95.25% vs 95.05%) while benefiting from a 57.2% reduction in FLOPs and a 53.5% reduction in the number of parameters. Compared with HRank, our method still prevails, with a 95.25% Top-1% vs 95.04%. These results show our method's robustness to models with Inception modules.

4.3. ImageNet results

ResNet-50. Table 4 includes our experimental results for ResNet-50 on ImageNet. CFDP achieves a Top-1 Accuracy of 76.10% and a Top-5 Accuracy of 92.93%, beating all methods in the upper portion of the table, with the exception of the original ResNet-50, where the Top-1 Accuracy is only 0.05% lower. Compared with HRank, CFDP also manages to obtain a close to 1% boost in Top-1 Accuracy and a 0.3% boost in Top-5 Accuracy. These results demonstrate that combining spatial and frequency informa-

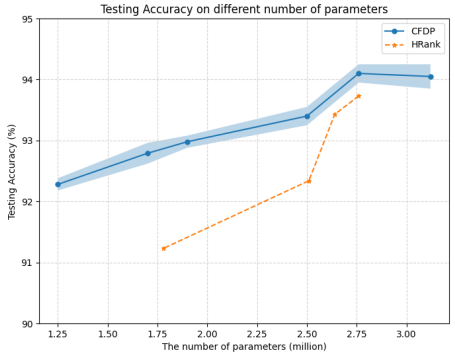

Figure 3. This figure shows the performance trend for extended pruning configurations on VGG-16-BN. We plot the performance (accuracy) over various model sizes on CIFAR-10 for both CFDP and HRank, to illustrate the superior performance and scalability.

tion allows for more than 50% compression and acceleration while having comparable performance to the original model on large-scale datasets such as ImageNet.

4.3.1 Extending the Pruning Configurations

In this experiment, we investigate the scalability of our model, as we test an extended range of model compression for VGG-16-BN on CIFAR-10. In Figure 3, we see that our method scales well as we increase the number of parameters. In particular, we can see a consistently increasing trend indicating that each additional parameter added to the model is carefully and correctly selected by our method. As expected this upward trend tapers off at a higher level of parameters, where the model has achieved its maximum learning potential given the structure of the architecture and dataset complexity, however, it still outperforms the SOTA and the original unpruned model. Additionally, we have achieved an accuracy of 82.5% with only 0.5M parameters, which is recovering 87.8% of the performance with only 3.3% of the parameters from the original model.

4.4. Ablation

In this section, we examine the motivation behind our design choices in CFDP. In particular, we look into how each component of our framework affects the overall performance of our method. For consistency, we run all ablation studies on VGG-16-BN using the CIFAR10 Dataset.

4.4.1 Components of $\mathcal{L}$ in CFDP

The goal of this study is to examine how each component affects the final performance. Referencing Table 5, we can see that $\mathcal{L}_{Fq}$ is a much stronger metric than the case without considering the distribution of frequencies $\mathcal{L}_{Fq}$ w/o $Dist$.

$\mathcal{L}_{Fq}$	$\mathcal{L}_{Fq}$ w/o $Dist$	$\mathcal{L}_{Sp}$	Performance
✓			93.70
	✓		93.58
		✓	93.50
✓		✓	**94.10**

Table 5. The effect of each component in CFDP on performance

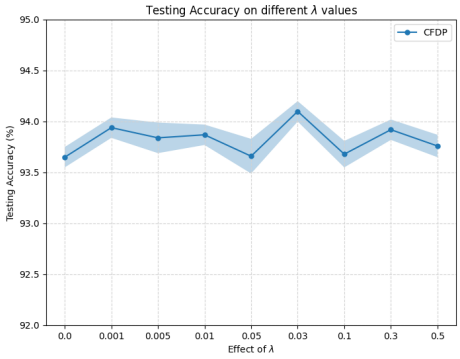

Figure 4. The effect of different λ regularization coefficients on the novel pruning metric

We also see that the frequency methods seem to yield better performance than the spatial metric $\mathcal{L}_{Sp}$. However, their combined performance is much stronger indicating that both the frequency and spatial metrics must bring some degree of unique information into the ranking allowing us to converge with an overall better channel configuration.

4.4.2 Effect of λ regularization on performance

The goal of this ablation study is to determine the impact of the regularizer on our framework's performance. Referencing Figure 4, we can see the performance of our method actually decreases as we diverge from the centrally selected value of $\lambda = 0.03$. Interestingly, it seems that this regularization value is a local maximum over the spectrum of tested values. We can see that a weak coefficient will default the performance to that of just $\mathcal{L}_{Fq}$, while too strong of a value will begin to interfere too heavily with the rankings generated from the frequency domain. Empirically, we determine the ideal λ to be 0.03.

4.4.3 Effect of B_s on performance

In this experiment, we investigate the effectiveness of different B_s values for patching the feature maps before applying DCT. Some studies have shown that different values of B_s can affect the performance of DCT compression [42]. Empirically we see that a very large block size can result in some degradation in accuracy while too small a size isn't able to properly correlate multiple pixels into the frequency spectrum, thus $B_s = 4$ was used in all experiments as it resulted in the best performance.

Block Size B_s	1	2	4	8
Performance	93.68	93.63	**94.10**	93.53

Table 6. Measuring the effect of block size B_s on DCT conversion, through empirical performance results

	Random	0-25	25-50	50-75	75-90	$\geq$90
Performance	93.60	93.88	93.37	93.72	93.58	**94.10**

Table 7. The effect of network training on the novel pruning metric

FSGM [12]	$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$	$\epsilon = 0.25$	$\epsilon = 0.3$
Original	93.96	61.85	55.01	50.64	46.62	41.70	36.01
CFDP	94.10	66.52	58.84	52.38	50.05	44.97	40.47
PGD [38]	$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$	$\epsilon = 0.25$	$\epsilon = 0.3$
Original	93.96	28.39	20.07	20.08	20.08	20.08	20.07
CFDP	94.10	30.92	24.99	24.99	24.99	24.99	24.99

Table 8. Measuring the effect of the novel pruning metric on defending against Adversarial Attacks

4.4.4 Effect of Network Training

In this section, we investigate the effect of pre-training on CFDP's ability to rank channels in a layer. The goal is to determine if pre-training is necessary for pruning under CFDP's method. Referencing Table 7 we can see that our ranking method is fairly robust to different levels of pre-training. Particularly interesting is the fact that determining the pruned subset of channels on an untrained model still outperforms many of the methodologies in Table 1, while slightly underperforming the original model by less than 0.4%. This approach shows that pre-training for network channel selection may not be required if CFDP is employed, due to its network initialization robustness.

4.4.5 Robustness to Adversarial Attacks

In this section, we evaluate the robustness of a model produced by our pruning framework with regard to adversarial attacks. The motivation behind this ablation study is to show that by removing several parameters from the model through pruning, our produced models are more robust to a potential attack on the data at inference time. To accomplish this, we incorporate two common types of attacks, FSGM [12] and PGD [38] on the testing data and evaluate the original and pruned models' testing performance. We can see from Table 8, that although the initial accuracies for both the original and pruned are quite close, our pruned model is far more robust to the adversarial attacks over the spectrum of their strength.

4.5. Interpretable Visualizations

In this section, we visualize how the pruned structure preserves the internal encodings and discriminative power of the CNN. To further demonstrate the effectiveness of our technique, we use the Grad-CAM algorithm [41] in Figure 5

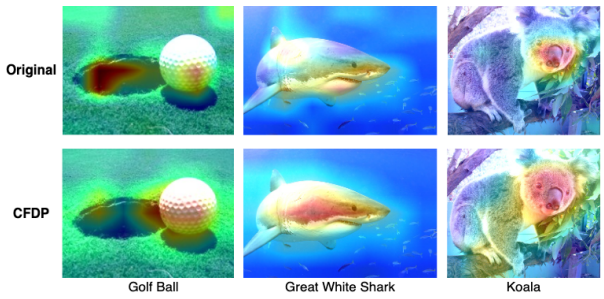

Figure 5. This figure shows 3 images sampled from ImageNet [7] with attention maps overlayed for both the Original and CFDP pruned versions of the ResNet-50 architecture. Despite the significant pruning of the architecture by almost 50% in parameters, it is able to create a reduced n-dimensional embedding space that retains the importance of feature recognition from the original 512-dimensional embedding space (and in some cases, improves it).

to show that the heat map produced by the pruned model is similar to that of the original model. As we can see from the three sets of pictures, the pruned model does an even better job of capturing the important parts of an image than the original pre-trained model. For instance, on the first row, the heatmap is centered around the golf ball for the pruned model, whereas the heatmap is mostly centered around the hole. Given that this picture corresponds to the class of golf balls and not of golf ball holes, the pruned model most correctly identified the vital features used for prediction.

5. Conclusion

In this paper, we introduced CFDP, a novel pruning method, to generate layer-wise rankings of channels with regard to the degree of information they contribute to the final model. We provide in-depth analysis and empirical investigation into the motivation behind each component of our saliency metric as well as its overall formulation. Further, we achieve state-of-the-art performance on CIFAR-10 and ImageNet across a variety of architectures. Lastly, we conduct several ablative studies testing each component of our metric, demonstrating the robustness of our frameworks to initializations, and the defensive capability of the resulting models to adversarial attacks. Overall we have shown the true merit of our framework and metric with regard to benchmarks standards. Through experiments on CIFAR10 we demonstrate superior performance (Top-1%), on ImageNet we show superior acceleration (through Params and FLOPs), and finally improved robustness over several ablative studies. In future work, we hope to dig further into the theoretical analysis of how the dominant frequencies are related to the scale of the feature maps, as well as expand our experimental work to potentially multi-class datasets.

References

[1] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. 3, 4

[2] E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967. 3

[3] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 3

[4] Hanting Chen, Yunhe Wang, Han Shu, Yehui Tang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. Frequency domain compact 3d convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1641–1650, 2020. 2

[5] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International conference on machine learning*, pages 2285–2294. PMLR, 2015. 1

[6] Yaosen Chen, Renshuang Zhou, Bing Guo, Yan Shen, Wei Wang, Xuming Wen, and Xinhua Suo. Discrete cosine transform for filter pruning. *Applied Intelligence*, pages 1–17, 2022. 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 3, 5, 8

[8] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014. 1

[9] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 1

[10] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019. 1

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 8

[13] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[14] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016. 1

[15] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016. 1

[16] Miska M. Hannuksela, Jani Lainema, and Vinod K. Malamal Vadakital. The high efficiency image file format standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, 32(4):150–156, 2015. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[18] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018. 2

[19] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi 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*, pages 4340–4349, 2019. 1, 2, 6

[20] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1389–1397, 2017. 1, 6

[21] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1

[22] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016. 3, 6

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[24] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 304–320, 2018. 1, 6

[25] Jaedeok Kim, Chiyoun Park, Hyun-Joo Jung, and Yoonsuck Choe. Plug-in, trainable gate for streamlining arbitrary neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4452–4459, 2020. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 3, 5

[27] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1, 2, 6

[28] Yuchao Li, Shaohui Lin, Baochang Zhang, Jianzhuang Liu, David Doermann, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploiting kernel sparsity and entropy for interpretable cnn compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2800–2809, 2019. 2

[29] Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proceedings of Machine Learning and Systems*, 3:93–138, 2021. 3

[30] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1529–1538, 2020. 1, 2, 5, 6

[31] Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Accelerating convolutional networks via global & dynamic filter pruning. In *IJCAI*, volume 2, page 8. Stockholm, 2018. 6

[32] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2790–2799, 2019. 1, 6

[33] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 3

[34] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. 1, 4

[35] Zhenhua Liu, Jizheng Xu, Xiulian Peng, and Ruiqin Xiong. Frequency-domain dynamic pruning for convolutional neural networks. *Advances in neural information processing systems*, 31, 2018. 1, 2

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[37] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017. 1, 6

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 8

[39] Harry Pratt, Bryan Williams, Frans Coenen, and Yalin Zheng. Fcnn: Fourier convolutional neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 786–798. Springer, 2017. 2

[40] Willie L. Scott and Subhash C. Kak. Block-level discrete cosine transform coefficients for autonomic face recognition. 2003. 3

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8

[42] S. Singh, V. Kumar, and H. K. Verma. Optimization of block size for dct-based medical image compression. *Journal of Medical Engineering & Technology*, 31(2):129–143, 2007. PMID: 17365437. 7

[43] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 3

[44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[45] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. Cnnpack: Packing convolutional neural networks in the frequency domain. *Advances in neural information processing systems*, 29, 2016. 2

[46] Yulong Wang, Xiaolu Zhang, Lingxi Xie, Jun Zhou, Hang Su, Bo Zhang, and Xiaolin Hu. Pruning from scratch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12273–12280, 2020. 1, 6

[47] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1740–1749, 2020. 2

[48] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in neural information processing systems*, 32, 2019. 1

[49] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9194–9203, 2018. 6

[50] Chenglong Zhao, Bingbing Ni, Jian Zhang, Qiwei Zhao, Wenjun Zhang, and Qi Tian. Variational convolutional neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2019. 1, 6

[51] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018. 3

[52] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. *Advances in neural information processing systems*, 31, 2018. 1, 3