

# WHC: WEIGHTED HYBRID CRITERION FOR FILTER PRUNING ON CONVOLUTIONAL NEURAL NETWORKS

Shaowu Chen, Weize Sun, Lei Huang

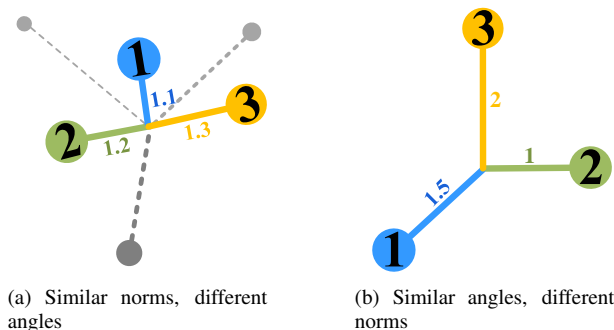
Department of Electronic and Information Engineering, Shenzhen University

## ABSTRACT

Filter pruning has attracted increasing attention in recent years for its capacity in compressing and accelerating convolutional neural networks. Various data-independent criteria, including norm-based and relationship-based ones, were proposed to prune the most unimportant filters. However, these state-of-the-art criteria fail to fully consider the dissimilarity of filters, and thus might lead to performance degradation. In this paper, we first analyze the limitation of relationship-based criteria with examples, and then introduce a new data-independent criterion, Weighted Hybrid Criterion (WHC), to tackle the problems of both norm-based and relationship-based criteria. By taking the magnitude of each filter and the linear dependence between filters into consideration, WHC can robustly recognize the most redundant filters, which can be safely pruned without introducing severe performance degradation to networks. Extensive pruning experiments in a simple one-shot manner demonstrate the effectiveness of the proposed WHC. In particular, WHC can prune ResNet-50 on ImageNet with more than 42% of floating point operations reduced without any performance loss in top-5 accuracy.

**Index Terms**— Filter pruning, CNN compression, acceleration.

## 1. INTRODUCTION



**Fig. 1:** Examples in which relationship-based criteria lose efficacy.

Deep convolutional neural networks (CNNs) have achieved great success in various research fields in recent years, and broader

Corresponding author: Weize Sun (proton198601@hotmail.com).

This research was supported by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515011706, the National Science Fund for Distinguished Young Scholars under Grant 61925108, the Foundation of Shenzhen under Grant JCYJ20190808122005605, the Open Research Fund from the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, under Grant B10120210117-OF07.

or deeper architectures have been derived to obtain better performance [1]. However, state-of-the-art CNNs usually come with an enormously large number of parameters, costing prohibitive memory and computational resources, and thus are problematic to be deployed on resource-limited platforms such as mobile devices [2].

To tackle this problem, pruning methods, including weight pruning [3, 4] and filter pruning approaches [5, 6, 7], are developed to compress and accelerate CNNs. Weight pruning evaluates the importance of elements of weight tensors using criteria such as absolute value [3], and sets those with the lowest scores to zero to achieve element-wise sparsity. Nevertheless, as the sparsity is unstructured, weight pruning relies on customized software and hardware to accelerate CNNs. By contrast, filter pruning methods remove structured filters, which yields slimmer CNNs that can be applied to general-purpose hardware using the common BLAS libraries directly with less memory and inference time consumption, and thus have attracted more attention in recent years.

One of the core tasks in filter pruning is to avoid severe performance degradation on CNNs. To this end, many pruning criteria, including data-driven [8, 9, 10] and data-independent ones [11, 12], are proposed to find the most redundant filters, which can be safely deleted. In this paper, we focus on data-independent criteria, which can be further divided into two categories: norm-based [13] and relationship-based [12, 14, 15, 16]. The former believe that norms such as the  $\ell_1$  and  $\ell_2$  [11, 13] of filters indicate their importance, and thus prune filters with more minor norms. However, He *et al.* [14] argue that it is difficult for norm-based criteria to verify unimportant filters when the variance of norms is insignificant. To solve the problem, they propose a relationship-based criterion, FPGM, which prunes filters with the shortest Euclidean distances from the others. In line with FPGM, cosine criterion [15] and CFP [16] are also developed, in which filters with the highest angles and the weakest correlation with others are considered the most valuable, respectively.

Generally speaking, relationship-based methods can overcome the problems introduced by norm-based criteria but still have their imperfections. For example, considering the colored filters shown in Figure 1(a) with similar small norms but different angles, FPGM and the cosine distance criterion will delete the 1st filter while keeping the inverted 2nd and 3rd since they have the largest Euclidean or angle-wise distance with others. However, the 2nd and 3rd filters will extract strongly correlated (although negatively) feature maps that contain highly redundant information, while the 1st filter orthogonal to them may extract totally different features. Deleting the 1st filter will weaken the representative capacity of CNNs, therefore, it is the 2nd or 3rd filter that should be deleted instead of the 1st one. Furthermore, considering that the 2nd filter has a smaller norm than the 3rd one, it is more reasonable to prune the 2nd filter. Whereas, the cosine distance [15] and CFP [16] criteria may rate the 2nd and 3rd the same score and remove one of them randomly. Similarly, this situation would also happen when filters have similar angles, such as

the example shown in the Figure 1(b). These examples demonstrate that relationship-based criteria also need improvement.

In this paper, we propose a Weighted Hybrid Criterion (WHC) that considers both magnitude and relationship of filters to address the problems mentioned above and alleviate performance degradation robustly. Specifically, we value filters that have both more significant norms and higher dissimilarity from others (manifesting as orthogonality, instead of antiphase in FPGM [14] and cosine distance criterion [15]) while deleting the others. Moreover, we weigh filters' dissimilarity terms differently rather than equally. That is, when evaluating a filter, its dissimilarity terms with filters of more significant norms are assigned a greater weight, while for filters of more minor norms, lower weights are appointed. The reason for this is that the dissimilarity evaluated by the degree of orthogonality is more trustworthy if the norm of a counterpart is more significant. In this manner, WHC can rationally score filters and prune those with the lowest scores, *i.e.*, the most redundant ones, and thus alleviate the degradation in CNNs' performance caused by pruning.

## 2. METHODOLOGY

### 2.1. Notation and Symbols

**Weight tensors of a CNN.** Following the conventions of PyTorch, we assume that a pre-trained  $L$ -layers CNN has weight tensors  $\{\mathcal{W}_l \in \mathbb{R}^{N_{l+1} \times N_l \times K \times K} | l = 1, 2, \dots, L\}$ , where  $\mathcal{W}_l$ ,  $K \times K$ ,  $N_l$  and  $N_{l+1}$  stand for the weight tensor of the  $l$ -th convolutional layer, the kernel sizes, the number of input and output channels of the  $l$ -th convolutional layer, respectively.

**Filters.** We use  $\mathcal{F}_{li}$  to represent the  $i$ -th filter of the  $l$ -th layer, where  $\mathcal{F}_{li} = \mathcal{W}_l[i, :, :, :] \in \mathbb{R}^{N_l \times K \times K}$ , *i.e.*, the  $i$ -th slide of  $\mathcal{W}_l$  along the first dimension.

**Pruning rates.**  $r_l = \frac{\#\text{pruned filters}}{N_{l+1}} \in [0, 1]$  denotes the proportion of pruned filters in the  $l$ -th layer.

### 2.2. Weighted Hybrid Criterion (WHC)

Norm-based criteria degrades when the variance of norms of filters is small [14], and relationship-based ones may fail to distinguish unimportant filters in several cases, as illustrated in Figure 1(a) and 1(b). To address these problems, we propose a data-independent Weighted Hybrid Criterion (WHC) to robustly prune the most redundant filters, which scores the importance of the  $i$ -th filter  $\mathcal{F}_{li}$  in the  $l$ -th layer by taking into account not only the norm of a filter but also the linear dissimilarity as follows:

$$\text{score}_{li} = \|\mathcal{F}_{li}\|_2 \sum_{j=1, j \neq i}^{N_{l+1}} \|\mathcal{F}_{lj}\|_2 (1 - |\cos \theta_{i,j}|), \quad (1)$$

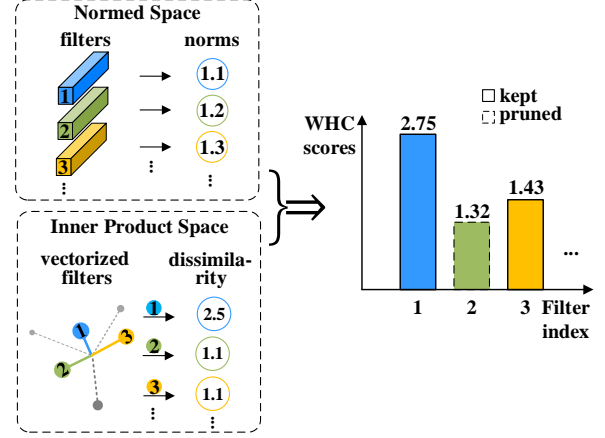
where

$$\cos \theta_{i,j} = \frac{\langle \mathcal{F}_{li}, \mathcal{F}_{lj} \rangle}{\|\mathcal{F}_{li}\|_2 \cdot \|\mathcal{F}_{lj}\|_2}, \quad (2)$$

and  $\|\mathcal{F}_{ij}\|_2$  represents the  $\ell_2$  norm of the vectorized  $\mathcal{F}_{ij}$ . Note that for a pre-trained model, we can assume that  $\|\mathcal{F}_{lj}\|_2 > 0$ .

When applying WHC in Eq. 1 for pruning, filters with lower scores are regarded as more redundant and thus will be deleted, while those with the opposite should be retained. To explain how WHC works in theory, we first discuss the unweighted variant of (1), Hybrid Criterion (HC):

$$\text{score}'_{li} = \|\mathcal{F}_{li}\|_2 \sum_{j=1}^{N_{l+1}} (1 - |\cos \theta_{i,j}|). \quad (3)$$



**Fig. 2:** Left top: Magnitude information of filters. Left bottom: Measuring dissimilarity of filters. Right: WHC scores filters and prune the most redundant ones.

Here  $1 - |\cos \theta_{i,j}| \in [0, 1]$ , the dissimilarity measurement (DM) between  $\mathcal{F}_{li}$  and  $\mathcal{F}_{lj}$ , acts as a scaling factor for  $\|\mathcal{F}_{li}\|_2$ , which in fact widens relative gaps between filters' norms and thus tackles the problem of invalidation of norm-based criteria [14] caused by a small variance of norms. Moreover, unlike euclidean or angle-wise distance-based criteria [14, 15] that prefer filters having  $180^\circ$  angles with others, WHC (1) and HC (3) value filters that are more orthogonal to the others, since they have shorter projected lengths with others and can extract less redundant features.

Note that in HC (3), the DM terms  $1 - |\cos \theta_{i,j}|$  for  $j = 1, \dots, N_{l+1}$  are considered equally valuable. However, WHC has a different view: when evaluating a filter  $\mathcal{F}_{li}$ , weights for the DM terms should be proportional to  $\|\mathcal{F}_{lj}\|_2$  for  $i \neq j$ . The reason for this is that it is less robust to count on filters with smaller norms when scoring filters. To explain this, consider two orthogonal filters,  $\mathcal{F}_{l1} = (100, 0)$  and  $\mathcal{F}_{l2} = (0, 0.1)$ . Since the norm of  $\mathcal{F}_{l2}$  is minor, a small additive interference  $(-0.1, -0.1)$  can easily change  $\mathcal{F}_{l2} = (0, 0.1)$  to  $\mathcal{F}'_{l2} = (-0.1, 0)$ , which radically modifies the DM term of  $\mathcal{F}_{l1}$  and  $\mathcal{F}_{l2}$  from the ceiling 1 to the floor 0. To improve robustness, the DM term should be weighted.

AutoML techniques such as meta-learning [17] can be used to learn the weights, but it will be time-consuming. Alternatively, we directly take norms of filters as weights in WHC (1), such that the blind spots of norm-based and relationship-based criteria can be eliminated easily but effectively. As illustrated in Figure 2, when encountering the case shown in Figure 1(a) where norms of filters differ insignificantly, WHC can utilize dissimilarity information to score the filters and recognize the most redundant one. Furthermore, WHC is also robust to the case shown in Figure 1(b) in which the filters have similar linear relationships but with varied norms, while relationship-based criteria such as the cosine criterion [15] will lose effectiveness since it scores the filters equally. There is a scenario that WHC may lose efficacy, *i.e.*, filters have the same norms and DM terms at the time. However, this indicates that there is no redundancy, and therefore it is not necessary to prune the corresponding model.

### 2.3. Algorithm Description

As described in Algorithm 1, we perform filter pruning using WHC following the common ‘‘Pretrain-Prune-Finetune’’ pipeline in a sim-

ple single-shot manner, with all layers pruned under the same pruning rate, *i.e.*,  $r_1 = r_2 = \dots = r_L$ . Although iterative mechanism [18], Knowledge Distillation [19, 20, 21], sensitive analysis that decides layer-wise pruning rates [11], and some fine-tuning techniques [22] can improve the performance of pruned CNNs, none of them are included in this paper for ease of presentation and validation.

---

**Algorithm 1** WHC for single-shot filter pruning

---

**Input:** Pre-trained model  $\{\mathcal{W}_l\}_{l=1}^L$ , pruning rates  $r_l$ , training data, fine-tuning epoch  $epoch_f$

- 1: **for**  $l = L \rightarrow 1$  **do**
- 2:   Score  $\{\mathcal{F}_l\}_{i=1}^{N_{l+1}}$  using WHC (1);
- 3:   Prune  $r_l * N_{l+1}$  filters with the lowest scores to get  $\mathcal{W}'_l$ ;
- 4:   Replace  $\mathcal{W}_l$  with  $\mathcal{W}'_l$ .
- 5: **end for**
- 6: Fine-tune  $\{\mathcal{W}'_l\}_{l=1}^L$  for  $epoch_f$  epochs.

**Output:** Compact model  $\{\mathcal{W}'_l\}_{l=1}^L$

---

By pruning  $r_l \cdot N_{l+1}$  filters in the  $l$ -th layer, WHC also reduces the same number of input channels in the  $l+1$ -th layer. Suppose the input feature maps for the  $l$ -th layer are of dimensions  $H_l \times W_l \times N_l$ , and output feature maps for the  $l$ -th,  $l+1$ -th layer are of dimensions  $H_{l+1} \times W_{l+1} \times N_{l+1}$  and  $H_{l+2} \times W_{l+2} \times N_{l+2}$ , respectively, pruning the  $l$ -th layer with  $r_l$  will reduce  $H_{l+1}W_{l+1}(N_{l+1}r_l)K^2N_l + H_{l+2}W_{l+2}N_{l+2}K^2(N_{l+1}r_l)$  floating point operations (**FLOPs**) totally, which greatly accelerates the forwarding inference.

### 3. EXPERIMENT

#### 3.1. Experimental Settings

**Datasets and baseline CNNs.** Following [13, 14], we evaluate the proposed WHC on the compact and widely used ResNet-20/32/56/110 for CIFAR-10 [23] and ResNet-18/34/50/101 for ILSVRC-2012 (ImageNet) [24]. For a fair comparison, we use the same pre-trained models for CIFAR-10 as [14]. Whereas, for ILSVRC-2012, since part of the pre-trained parameters of [14] are not available, we use official Pytorch pre-trained models [25] with slightly lower accuracy. Code and CKPT are available at <https://github.com/ShaoWuChen/WHC>.

**Pruning and fine-tuning.** The experiments are implemented with Pytorch 1.3.1 [25]. We keep all our implementations such as data argumentation strategies, pruning settings, and fine-tuning epochs the same as [13, 14], except that we use the straightforward single-shot mechanism. In the pruning stage, all convolutional layers in a network are pruned with the same pruning rate, and we report the proportion of “FLOPs” dropped for ease of comparison.

**Compared methods.** We compare WHC with several criteria, including the data-independent norm-based PFEC [11], SPF [13], ASFP [26], relationship-based FPGM [14], and several data-dependent methods HRank [27], GAL [28], LFPC [29], CP [30], NISP [31], ThiNet [18] and ABC [32].

#### 3.2. Evaluation on CIFAR-10

For CIFAR-10, we repeat each experiment three times and report the average accuracy after fine-tuning. As shown in Table 1, WHC outperforms several state-of-the-art counterparts. WHC can prune 52.3% of FLOPs in ResNet-110 with even 0.39% improvement, while the norm-based SFP under the same settings suffers 0.78% of degradation. The improvement shows that under moderate pruning

Depth	Method	Baseline acc. (%)	Pruned acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)
20	<b>WHC</b>	92.20 ( $\pm 0.18$ )	<b>91.62 (<math>\pm 0.14</math>)</b>	<b>0.58</b>	42.2
	<b>WHC</b>	92.20 ( $\pm 0.18$ )	90.72 ( $\pm 0.16$ )	1.48	<b>54.0</b>
32	<b>WHC</b>	92.63 ( $\pm 0.70$ )	<b>92.71 (<math>\pm 0.08</math>)</b>	<b>-0.08</b>	41.5
	<b>WHC</b>	92.63 ( $\pm 0.70$ )	92.44 ( $\pm 0.12$ )	0.19	<b>53.2</b>
56	PFEC [11]	93.04	93.06	-0.02	27.6
	<b>WHC</b>	<b>93.59 (<math>\pm 0.58</math>)</b>	<b>93.91 (<math>\pm 0.06</math>)</b>	<b>-0.32</b>	<b>28.4</b>
	GAL [28]	93.26	93.38	0.12	37.6
	SFP [13]	<b>93.59 (<math>\pm 0.58</math>)</b>	93.78 ( $\pm 0.22$ )	-0.19	<b>41.1</b>
	<b>WHC</b>	<b>93.59 (<math>\pm 0.58</math>)</b>	<b>93.80 (<math>\pm 0.33</math>)</b>	<b>-0.21</b>	<b>41.1</b>
	HRank [27]	93.26	93.17	0.09	50.0
	SFP [13]	<b>93.59 (<math>\pm 0.58</math>)</b>	93.35 ( $\pm 0.31$ )	0.24	<b>52.6</b>
	ASFP [26]	<b>93.59 (<math>\pm 0.58</math>)</b>	93.12 ( $\pm 0.20$ )	0.47	<b>52.6</b>
	FPGM [14]	<b>93.59 (<math>\pm 0.58</math>)</b>	93.26 ( $\pm 0.03$ )	0.33	<b>52.6</b>
	<b>WHC</b>	<b>93.59 (<math>\pm 0.58</math>)</b>	<b>93.47 (<math>\pm 0.18</math>)</b>	<b>0.12</b>	<b>52.6</b>
110	LFPC [29]	<b>93.59 (<math>\pm 0.58</math>)</b>	93.24 ( $\pm 0.17$ )	0.35	52.9
	ABC [32]	93.26	93.23	0.03	54.1
	<b>WHC</b>	<b>93.59 (<math>\pm 0.58</math>)</b>	<b>93.66 (<math>\pm 0.19</math>)</b>	<b>-0.07</b>	<b>54.8</b>
	GAL [28]	93.26	91.58	1.68	60.2
	<b>WHC</b>	<b>93.59 (<math>\pm 0.58</math>)</b>	<b>93.29 (<math>\pm 0.11</math>)</b>	<b>0.30</b>	<b>63.2</b>
	GAL [28]	93.50	93.59	-0.09	18.7
	PFEC [11]	93.53	93.30	0.23	38.6
	SFP [13]	<b>93.68 (<math>\pm 0.32</math>)</b>	93.86 ( $\pm 0.21$ )	-0.18	<b>40.8</b>
	ASFP [26]	<b>93.68 (<math>\pm 0.32</math>)</b>	93.37 ( $\pm 0.12$ )	0.31	<b>40.8</b>
	<b>WHC</b>	<b>93.68 (<math>\pm 0.32</math>)</b>	<b>94.32 (<math>\pm 0.17</math>)</b>	<b>-0.64</b>	<b>40.8</b>
110	GAL [28]	93.26	92.74	0.76	48.5
	SFP [13]	<b>93.68 (<math>\pm 0.32</math>)</b>	92.90 ( $\pm 0.18$ )	0.78	<b>52.3</b>
	FPGM [14]	<b>93.68 (<math>\pm 0.32</math>)</b>	93.74 ( $\pm 0.10$ )	-0.06	<b>52.3</b>
	ASFP [26]	<b>93.68 (<math>\pm 0.32</math>)</b>	93.10 ( $\pm 0.06$ )	0.58	<b>52.3</b>
	<b>WHC</b>	<b>93.68 (<math>\pm 0.32</math>)</b>	<b>94.07 (<math>\pm 0.20</math>)</b>	<b>-0.39</b>	<b>52.3</b>
	HRank [27]	93.50	93.36	0.14	58.2
110	LFPC [29]	<b>93.68 (<math>\pm 0.32</math>)</b>	93.07 ( $\pm 0.15$ )	0.61	60.3
	ABC [32]	93.50	93.58	-0.08	65.0
	<b>WHC</b>	<b>93.68 (<math>\pm 0.32</math>)</b>	<b>93.82 (<math>\pm 0.07</math>)</b>	<b>-0.14</b>	<b>65.8</b>

**Table 1:** Pruning results on CIFAR-10. “↓” means “drop”. In “Acc. ↓”, the smaller, the better; a negative drop means improvement. In “FLOPs ↓”, a larger number indicates that more FLOPs are reduced.

rates, WHC can alleviate the overfitting problem of models without hurting their capacity.

Compared with the iterative ASFP [26], data-driven HRank [27], AutoML-based ABC [32] and LFPC [29], WHC in a single-shot manner can also achieve competitive performance. For example, although more FLOPs are reduced, WHC still achieves 0.42% and 0.75% higher accuracy than LFPC in ResNet-56 and ResNet-110, respectively, which demonstrates that WHC can recognize the most redundant filters effectively. Furthermore, under similar pruned rates, as the depth of CNNs increases, the pruned models obtained by WHC suffer less performance degradation. The reason for this is that deeper CNNs contain more redundancy, which can be removed by WHC robustly without hurting CNNs’ capacity severely.

#### 3.3. Evaluation on ILSVRC-2012

The results are shown in Table 2. Not surprisingly, compared with several state-of-the-art methods, WHC not only achieves the highest top-1 and top-5 accuracy, but also suffers the slightest performance degradation. In ResNet-50, WHC reduces more than 40% of FLOPs but barely brings loss in the top-1 and top-5 accuracy, while the norm-based SFP suffers 14% degradation in the top-1 accuracy and other methods more than 0.5%. Compared with norm-based and relationship-based criteria, the superior performance of WHC can be attributed to the utilization of both norm and linear similarity information of filters, and the assigned weights on different DM terms can provide more robust results.

Depth	Method	Baseline top-1 acc. (%)	Pruned top-1 acc. (%)	Top-1 acc. ↓ (%)	Baseline top-5 acc. (%)	Pruned top-5 acc. (%)	Top-5 acc. ↓ (%)	FLOPs ↓ (%)
18	SFP [13]	70.23	60.79	9.44	89.51	83.11	6.40	<b>41.8</b>
	ASFP [26]	70.23	68.02	2.21	89.51	88.19	1.32	<b>41.8</b>
	FPGM [14]	<b>70.28</b>	68.41	1.87	<b>89.63</b>	88.48	1.15	<b>41.8</b>
	WHC	69.76	<b>68.48</b>	<b>1.28</b>	89.08	<b>88.52</b>	<b>0.56</b>	<b>41.8</b>
34	PFEC [11]	73.23	72.17	1.06	-	-	-	24.2
	ABC [32]	73.28	70.98	2.30	91.45	90.05	1.40	41.0
	SFP [13]	<b>73.92</b>	72.29	1.63	<b>91.62</b>	90.90	0.72	<b>41.1</b>
	ASFP [26]	<b>73.92</b>	72.53	1.39	<b>91.62</b>	91.04	0.58	<b>41.1</b>
	FPGM [14]	<b>73.92</b>	72.54	1.38	<b>91.62</b>	91.13	0.49	<b>41.1</b>
	WHC	73.31	<b>72.92</b>	<b>0.40</b>	91.42	<b>91.14</b>	<b>0.28</b>	<b>41.1</b>
50	ThiNet [18]	72.88	72.04	0.84	91.14	90.67	0.47	36.7
	SFP [13]	<b>76.15</b>	62.14	14.01	<b>92.87</b>	84.60	8.27	41.8
	ASFP [26]	<b>76.15</b>	75.53	0.62	<b>92.87</b>	92.73	0.14	41.8
	FPGM [14]	<b>76.15</b>	75.59	0.56	<b>92.87</b>	92.63	0.24	<b>42.2</b>
	WHC	76.13	<b>76.06</b>	<b>0.07</b>	92.86	<b>92.86</b>	<b>0.00</b>	<b>42.2</b>
50	HRank [27]	<b>76.15</b>	74.98	1.17	<b>92.87</b>	92.33	0.54	43.8
	NISP [31]	-	-	0.89	-	-	-	44.0
	GAL [28]	<b>76.15</b>	71.95	4.20	<b>92.87</b>	90.94	1.93	43.0
	CFP [16]	75.30	73.40	1.90	92.20	91.40	0.80	49.6
	CP [30]	-	-	-	92.20	90.80	1.40	50.0
101	FPGM [14]	<b>76.15</b>	74.83	1.32	<b>92.87</b>	92.32	0.55	<b>53.5</b>
	WHC	76.13	<b>75.33</b>	<b>0.80</b>	92.86	<b>92.52</b>	<b>0.34</b>	<b>53.5</b>
	GAL [28]	<b>76.15</b>	71.80	4.35	92.87	90.82	2.05	55.0
	ABC [32]	76.01	73.86	2.15	<b>92.96</b>	91.69	1.27	54.3
101	ABC [32]	76.01	73.52	2.49	<b>92.96</b>	91.51	1.45	56.6
	LFPC [29]	<b>76.15</b>	74.46	1.69	92.87	92.04	0.83	60.8
	WHC	76.13	<b>74.64</b>	<b>1.49</b>	92.86	<b>92.16</b>	<b>0.70</b>	<b>60.9</b>
	FPGM [14]	<b>77.37</b>	77.32	0.05	<b>93.56</b>	93.56	0.00	<b>42.2</b>
101	WHC	<b>77.37</b>	<b>77.75</b>	<b>-0.38</b>	93.55	<b>93.84</b>	<b>-0.30</b>	<b>42.2</b>
	ABC [32]	<b>77.38</b>	75.82	1.56	<b>93.59</b>	92.74	0.85	59.8
101	WHC	77.37	<b>76.63</b>	<b>0.74</b>	93.55	<b>93.30</b>	<b>0.25</b>	<b>60.8</b>

**Table 2:** Pruning results on ILSVRC-2012 (ImageNet). “acc.” and “↓” stand for “accuracy” and “drop”, respectively.

### 3.4. Ablation Study

**Decoupling experiment.** To further validate the effectiveness of WHC, we progressively decouple WHC into several criteria, as shown in Table 3. The cosine criterion [15] is also added for comparison. We repeat pruning 40% of filters in ResNet-32 and ResNet-56 three times and report the raw accuracy (without fine-tuning) and the average drop in accuracy after fine-tuning. Compared with the cosine criterion [15], the DM criterion suffers less degradation in accuracy and is therefore more rational. Taking into account both norm and dissimilarity, HC achieves better performance than  $\ell_2$  and DM. Furthermore, by assigning different weights to the DM terms, WHC outperforms all counterparts stably, especially in the more compact ResNet-32. The  $\ell_2$  achieves similar performance as WHC in ResNet-56, but fails to maintain the same performance in ResNet-32, demonstrating the robustness of the proposed WHC.

**Types of norm and dissimilarity measurement.** We replace  $\ell_2$  norm and  $\cos \theta_{i,j}$  in WHC (1) with  $\ell_1$  norm and correlation coefficient, respectively. The correlation can be regarded as the centralized version of  $\cos \theta_{i,j}$ . We conduct experiments on ResNet-32 with  $r_l = 40\%$ . The fine-tuned accuracy of the  $\ell_1$  and correlation version of WHC are  $(92.50 \pm 0.11)\%$  and  $(92.62 \pm 0.18)\%$ , respectively, which are slightly higher than the naive WHC,  $(92.44 \pm 0.12)\%$ . The results indicate that WHC can be further improved with more suitable types of norm and dissimilarity measurement.

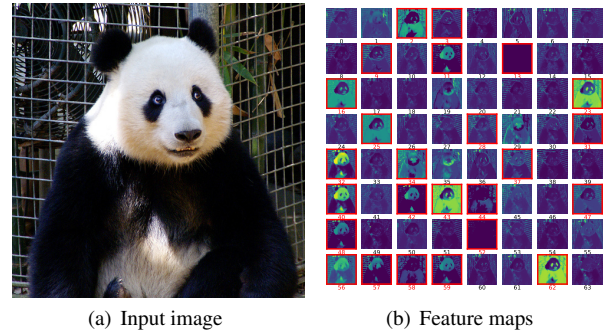
### 3.5. Visualization

We prune 40% of filters in the first layer of ResNet-50 for ImageNet and visualize the output feature maps, as shown in Figure 3. Among the pruned filters, 3 and 34 can be replaced by 10 and 29, respec-

Depth& acc.	Criterion	Raw acc. (%)	Fine-tuned acc. ↓ (%)
32 92.63%	$\ \mathcal{F}_{li}\ _2$ [13]	10.06	0.42
	$\sum_{j=1}^{N_{l+1}} \cos \theta_{i,j}$ [15]	10.00	0.78
	$\sum_{j=1}^{N_{l+1}} (1 -  \cos \theta_{i,j} )$ (DM)	11.30	0.56
	$\ \mathcal{F}_{li}\ _2 \sum_{j=1}^{N_{l+1}} (1 -  \cos \theta_{i,j} )$ (HC)	11.18	0.32
	$\ \mathcal{F}_{li}\ _2 \sum_{j=1}^{N_{l+1}} \ \mathcal{F}_{lj}\ _2 (1 -  \cos \theta_{i,j} )$ (WHC)	11.25	<b>0.19</b>
56 93.59%	$\ \mathcal{F}_{li}\ _2$ [13]	16.99	0.20
	$\sum_{j=1}^{N_{l+1}} \cos \theta_{i,j}$ [15]	10.00	0.51
	$\sum_{j=1}^{N_{l+1}} (1 -  \cos \theta_{i,j} )$ (DM)	9.94	0.37
	$\ \mathcal{F}_{li}\ _2 \sum_{j=1}^{N_{l+1}} (1 -  \cos \theta_{i,j} )$ (HC)	17.94	0.14
	$\ \mathcal{F}_{li}\ _2 \sum_{j=1}^{N_{l+1}} \ \mathcal{F}_{lj}\ _2 (1 -  \cos \theta_{i,j} )$ (WHC)	19.73	<b>0.12</b>

**Table 3:** Decoupling results on ResNet-32 and ResNet-56 for CIFAR-10. “acc.” and “↓” stand for “accuracy” and “drop”, respectively

tively, and [13, 32, 48, 62, *et al*] fail to extract valuable features. We also compare WHC with the  $\ell_2$  norm criterion [13] and relationship-based FPGM [14], finding that  $\ell_2$  and FPGM rank the filters differently from WHC but finally give a similar pruning list under the given pruning rate. The only divergence of WHC and the  $\ell_2$  criterion arises over filters 29 and 56: WHC keeps 29 and prunes 56, while  $\ell_2$  criterion takes the opposite action. We consider filter 29 to be more valuable than filter 56, since the latter fails to extract significant features, while the former highlights the eyes of the input image. The difference of WHC and  $\ell_2$  in the pruning list for a single layer is insignificant, but an accumulation of tens of layers finally results in a wide gap and makes WHC more robust in finding redundant filters.



**Fig. 3:** Visualization of ResNet-50-conv1 output feature maps (after ReLU, BN and MaxPooling, *i.e.*, the input of the second Conv layer). The feature maps bounded by red boxes correspond to the pruned filters; 40% of the filters are pruned. The highest values are colored in the brightest green, while the lowest in the darkest blue.

## 4. CONCLUSION

We propose a simple but effective data-independent criterion, Weighted Hybrid Criterion (WHC), for filter pruning. Unlike previous norm-based and relationship-based criteria that use a single type of information to rank filters, WHC takes into consideration both magnitude of filters and dissimilarity between filter pairs, and thus can recognize the most redundant filters more efficiently. Furthermore, by reweighting the dissimilarity measurements according to the magnitude of counterpart filters adaptively, WHC is able to alleviate performance degradation on CNNs caused by pruning robustly.



## 5. REFERENCES

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
- [2] Shaowu Chen, Jiahao Zhou, Weize Sun, and Lei Huang, “Joint matrix decomposition for deep convolutional neural networks compression,” *Neurocomputing*, 2022.
- [3] Song Han, Jeff Pool, John Tran, and William J. Dally, “Learning both weights and connections for efficient neural network,” in *NeurIPS*, 2015, pp. 1135–1143.
- [4] Jonathan Frankle and Michael Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” in *ICLR*, 2019.
- [5] S. H. Shabbeer Basha, Sheethal N. Gowda, and Dakala Jayachandra, “A simple hybrid filter pruning for efficient edge inference,” in *ICASSP*, 2022, pp. 3398–3402.
- [6] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang, “Learning efficient convolutional networks through network slimming,” in *ICCV*, 2017, pp. 2755–2763.
- [7] Yuan Zhang, Yuan Yuan, and Qi Wang, “ACP: adaptive channel pruning for efficient neural networks,” in *ICASSP*, 2022, pp. 4488–4492.
- [8] Jianbo Ye, Xin Lu, Zhe Lin, and James Z. Wang, “Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers,” in *ICLR*, 2018.
- [9] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz, “Importance estimation for neural network pruning,” in *CVPR*, 2019, pp. 11264–11272.
- [10] Shixing Yu, Zhewei Yao, Amir Gholami, Zhen Dong, Sehoon Kim, Michael W Mahoney, and Kurt Keutzer, “Hessian-aware pruning and optimal neural implant,” in *WACV*, 2022, pp. 3880–3891.
- [11] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, “Pruning filters for efficient convnets,” in *ICLR*, 2017.
- [12] ZhaoJing Zhou, Yun Zhou, Zhuqing Jiang, Aidong Men, and Haiying Wang, “An efficient method for model pruning using knowledge distillation with few samples,” in *ICASSP*, 2022, pp. 2515–2519.
- [13] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang, “Soft filter pruning for accelerating deep convolutional neural networks,” in *IJCAI*, 2018, pp. 2234–2240.
- [14] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang, “Filter pruning via geometric median for deep convolutional neural networks acceleration,” in *CVPR*, 2019, pp. 4340–4349.
- [15] Yang He, Ping Liu, Linchao Zhu, and Yi Yang, “Meta filter pruning to accelerate deep convolutional neural networks,” *arXiv preprint arXiv:1904.03961*, 2019.
- [16] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay P. Nambodiri, “Leveraging filter correlations for deep model compression,” in *WACV*, 2020, pp. 824–833.
- [17] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, 2022.
- [18] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin, “Thinet: A filter level pruning method for deep neural network compression,” in *ICCV*, 2017, pp. 5068–5076.
- [19] Haoli Bai, Hongda Mao, and Dinesh Nair, “Dynamically pruning segformer for efficient semantic segmentation,” in *ICASSP*, 2022, pp. 3298–3302.
- [20] ZhaoJing Zhou, Yun Zhou, Zhuqing Jiang, Aidong Men, and Haiying Wang, “An efficient method for model pruning using knowledge distillation with few samples,” in *ICASSP*, 2022, pp. 2515–2519.
- [21] Donggyu Joo, Eojindl Yi, Sunghyun Baek, and Junmo Kim, “Linearly replaceable filters for deep network channel pruning,” in *AAAI*, 2021, pp. 8021–8029.
- [22] Duong H. Le and Binh-Son Hua, “Network pruning that matters: A case study on retraining variants,” in *ICLR*, 2021.
- [23] Alex Krizhevsky and Geoffrey Hinton, “Learning multiple layers of features from tiny images,” *Tech. Rep.*, 2009.
- [24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, 2015.
- [25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, “Automatic differentiation in pytorch,” 2017.
- [26] Yang He, Xuanyi Dong, Guoliang Kang, Yanwei Fu, Chenggang Yan, and Yi Yang, “Asymptotic soft filter pruning for deep convolutional neural networks,” *IEEE Trans. Cybern.*, pp. 3594–3604, 2020.
- [27] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao, “Hrunk: Filter pruning using high-rank feature map,” in *CVPR*, 2020, pp. 1526–1535.
- [28] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David S. Doermann, “Towards optimal structured CNN pruning via generative adversarial learning,” in *CVPR*, 2019, pp. 2790–2799.
- [29] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang, “Learning filter pruning criteria for deep convolutional neural networks acceleration,” in *CVPR*, 2020, pp. 2006–2015.
- [30] Yihui He, Xiangyu Zhang, and Jian Sun, “Channel pruning for accelerating very deep neural networks,” in *ICCV*, 2017, pp. 1398–1406.
- [31] 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 *CVPR*, 2018, pp. 9194–9203.
- [32] Mingbao Lin, Rongrong Ji, Yuxin Zhang, Baochang Zhang, Yongjian Wu, and Yonghong Tian, “Channel pruning via automatic structure search,” in *IJCAI*, 2020, pp. 673–679.