applied sciences

MDPI

Article

Filter Pruning via Attention Consistency on Feature Maps

Huoxiang Yang [1,2], Yongsheng Liang [1,3,*], Wei Liu [2,4] and Fanyang Meng [2]

[1] College of Electronic and Information Engineering, Shenzhen University, Shenzhen 518060, China
[2] Pengcheng Laboratory, Shenzhen 518055, China
[3] Big Data and Internet College, Shenzhen University of Technology, Shenzhen 518118, China
[4] School of Computer Sciences, Shenzhen Institute of Information Technology, Shenzhen 518172, China
* Correspondence: liangys@hit.edu.cn

Abstract: Due to the effective guidance of prior information, feature map-based pruning methods have emerged as promising techniques for model compression. In the previous works, the undifferentiated treatment of all information on feature maps amplifies the negative impact of noise and background information. To address this issue, a novel filter pruning strategy called Filter Pruning via Attention Consistency (FPAC) is proposed, and a simple and effective implementation method of FPAC is presented. FPAC is inspired by the notion that the attention of feature maps on one layer is in high consistency of spatial dimension. The experiments also show that feature maps with lower consistency are less important. Hence, FPAC measures the importance of filters by evaluating the attention consistency on the feature maps and then prunes the filters corresponding to feature maps with lower consistency. The present experiments on various datasets further confirm the effectiveness of FPAC. For instance, applying VGG-16 on CIFAR-10, the classification accuracy even increases from 93.96% to 94.03% with 58.1% FLOPs reductions. Furthermore, applying ResNet-50 on ImageNet achieves 45% FLOPs reductions with only 0.53% accuracy loss.

Keywords: neural network compression; channel pruning; attention consistency

check for updates

Citation: Yang, H.; Liang, Y.; Liu, W.; Meng, F. Filter Pruning via Attention Consistency on Feature Maps. *Appl. Sci.* **2023**, *13*, 1964. https://doi.org/10.3390/app13031964

Academic Editor: José Salvador Sánchez Garreta

Received: 12 December 2022
Revised: 18 January 2023
Accepted: 20 January 2023
Published: 2 February 2023

1. Introduction

Neural networks have achieved great success across various application fields. Nevertheless, the enormous number of parameters as well as the high computational complexity of their implementation challenges the versatile deployment of such systems on resource-limited devices [1–6]. To address this issue, neural network compression was introduced which generates a more compact and efficient network based on the original design. These techniques often adequately reduce the number of computations and parameters in the neural network facilitating their practical deployment. The existing methods for neural network compression can be simply divided into four categories: weights low-rank decomposition [7–10], network pruning [11–16], parameter quantification [17,18] and distillation [19,20]. Among these categories, the network pruning method is simple, efficient, and easy to implement, hence attracting the attention of the research community. Main network pruning techniques include weight-based filter pruning [21–25] and feature map-based filter pruning [26–31]. Weight-based filter pruning is an efficient data-independent method with strong generalization ability. However, without the guidance of data preliminary information, it is not easy to achieve high performance on all datasets. In contrast, the feature map-based filter pruning methods use data distribution information to adjust the pruning strategy and improve pruning efficiency. The importance of a given feature map is positively correlated with the corresponding filter. Therefore, the feature map-based pruning methods perform filter pruning by defining an evaluation function to measure the importance of the feature maps. In the existing works, all information (pixels) on feature maps is utilized to estimate their importance. However, if the information on a feature map is dominated by unimportant information, the existing methods are unable to effectively evaluate the feature's importance.

In cognitive neurology, attention is considered a complex cognitive function that is indispensable for humans. Owing to the attention mechanism, people can focus on important information and ignore unimportant information. In this work, applying the attention mechanism to feature maps, a novel filter pruning strategy called Filter Pruning via Attention Consistency (FPAC) is proposed, as shown in Figure 1. In the proposed framework, the attention consistency discriminant module (ACDM) plays an essential role and includes the main contributions of this work. The main difference between FPAC and the existing methods is that ACDM presents an attention evaluation function instead of all information on feature maps. Given an input image to the network, Figure 2a illustrates the image with attention and Figure 2b shows the feature maps generated by the sequence of the fifth convolution, batch normalization and activation layer of ResNet50. The main motivation behind the work is empirical and quantitative observations. Firstly, the information of feature maps is mainly distributed in attention. Secondly, feature maps in red rectangles have more information in attention. Intuitively, the feature maps in red rectangles are more effective for network prediction than those in blue rectangles. In the existing methods, attention and background information are directly sent to the evaluation function without preprocessing. Since the analysis of attention enables avoiding most of the background information, it is more effective to analyze attention instead of considering all the information in feature maps. Therefore, in this work, a more reasonable method of analyzing attention to feature maps is proposed.

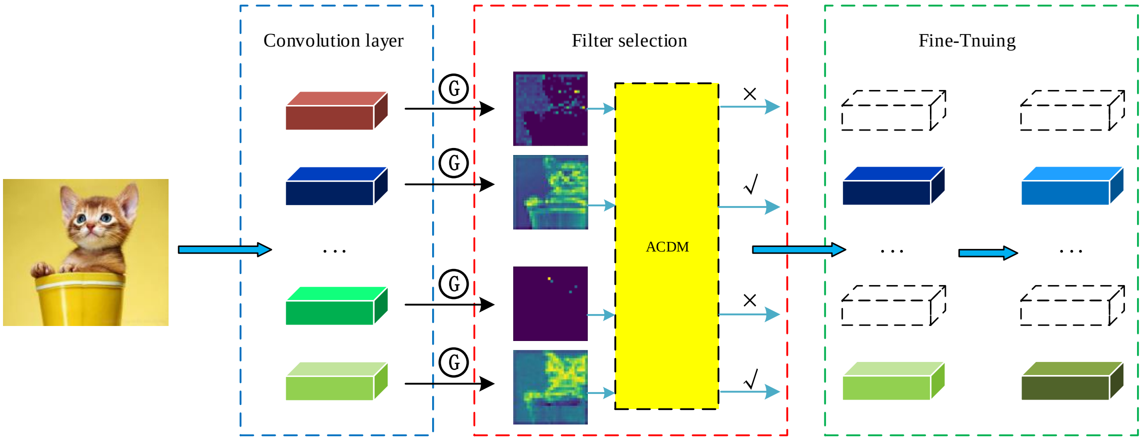


Figure 1. The overview of FPAC.

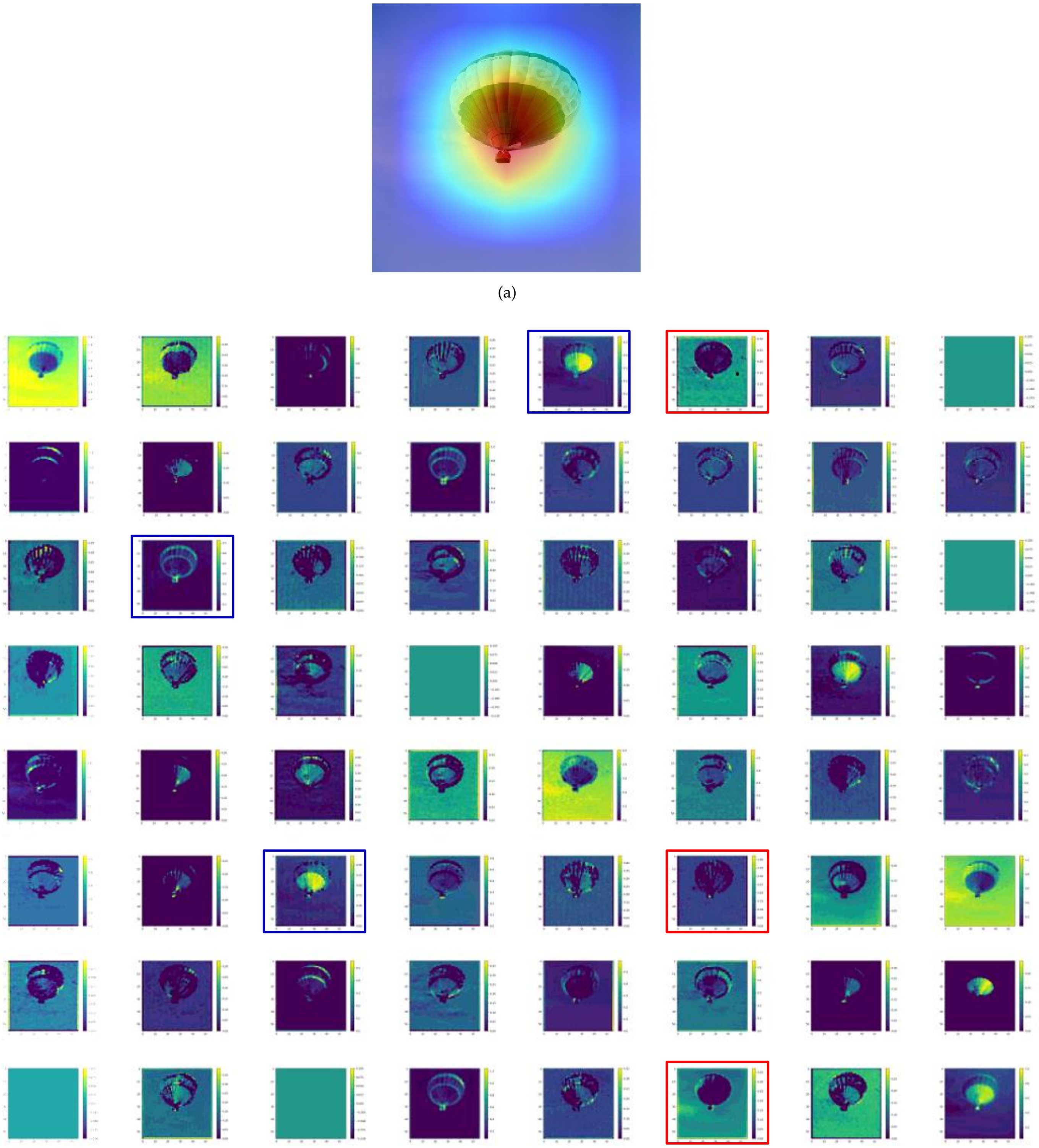
When using 3 × 3 convolution to calculate the pixel of the feature maps, only 8 adjacent pixels of the input features are relevant. Therefore, the general shapes of the feature maps are maintained. Due to the self-learning ability of convolution, it presents a higher response in the effective information area. Figure 2 shows that most feature maps have larger activation values in the attention. Although the information is mainly distributed in attention, the distribution of information in attention is quite different.

Based on the analysis above, an ideal evaluation function based on feature maps ought to satisfy the following two requirements: (1) it should express attention to feature maps; (2) it should measure the consistency of the attention of feature maps. The design of ACDM precisely meets these two requirements. In Section 3, ACDM is presented in detail and analytically investigated.

To summarize, the main contributions of this work are as the following:

(1) Based upon visualization of the feature maps, over the feature maps, the information is better distinguishable by means of attention values. A filter pruning strategy based on the attention of the feature maps is proposed. To the best of our knowledge, this is the first time that an attention mechanism is used for pruning.

(2) Based on the discovery that most feature maps have larger activation values in the attention, and the distribution of information in attention is quite different. A simple and effective filter selection method is proposed.



(a)

(b)

Figure 2. An image and its feature maps on the fifth convolution layer of ResNet50. (**a**) Input image with attention; (**b**) Feature maps on one layer.

2. Related Work

The main network pruning methods mainly focus on weight-based filter pruning and feature map-based filter pruning. In addition, the adaptive importance of filter pruning methods also received much attention. In this section, the existing research on these three approaches is discussed.

2.1. Weight-Based Filter Pruning

Weight-based filter pruning methods are data-independent and they implement pruning strategies by analyzing the weights of the filters. Norm-based filter pruning methods use the criterion that the filters with smaller norms are not important to prune the filters. For instance, the by calculating the ℓ_1-norm of filter weight, the L1 method [21] prunes the filter with a smaller ℓ_1-norm of weight. Soft Filter Pruning (SFP) [22] first calculates the ℓ_2-norm of the weights, then prunes those filters with the smaller weights norm in a soft manner. However, Filter Pruning via Geometric Median (FPGM) [23] indicates that the filters with smaller norms are not important and are not rigorous. According to the characteristics of the geometric median, FPGM finds that the information of the filters close to the geometric median can be represented by the remaining filters, which shows these filters can be pruned. Furthermore, Neuron Importance Score Propagation (NISP) [24] uses the properties of matrix power series to calculate the importance of a filter relative to all other filters, thus determining the unimportant filters. Furthermore, Spectral Clustering Filter Pruning (SCSP) [25] clusters the filters into K groups and sorts them according to their contributions, pruning the filters with smaller contributions.

Nevertheless, the weight-based filter pruning methods lack the guidance of data prior information and data labels, hence achieving sub-optimal performance on some datasets.

2.2. Feature Map-Based Filter Pruning

Feature map-based filter pruning methods utilize the prior information of training data to implement filter pruning strategies. For instance, Thinet [26] analyzes the filter contribution by the statistical information of the filter’s output features and accordingly prunes filters with a smaller contribution. Sampling [27] proposes a sampling-based pruning method. It assigns a score to the significance of filters and constructs an importance sampling distribution where the filters with a higher probability of sampling are regarded as the more important ones. Another method, Channel Pruning (CP) [28] first samples the points on the feature maps, then retains significant filters by minimizing the reconstruction errors of the sampling points. The PFA [29] also prunes the secondary filter by setting the energy ratio of principal component analysis (PCA) which applies to feature maps. Subspace clustering is used to research the correlation among the feature maps, and then the filters generating redundant feature maps are pruned accordingly [30]. Hrank [31] also presented a simple pruning method to prune filters with lower-rank feature maps.

Although feature map-based filter pruning methods are robust to different datasets, their performance is susceptible to noise. To solve this problem, a pruning method that focuses on the attention of the images is proposed.

2.3. Adaptive Importance Filter Pruning

Different from the above two methods, the adaptive importance filter pruning methods introduce additional discriminant modules into the network and use the optimization training methods to determine the filters that need pruning. Variational Pruning (VP) [32] and Slimming [33] modify the batch normalization layer by introducing the scaling factor, and the filters corresponding to the lower scaling factor are selected as the unimportant ones. Discrimination-aware Channel Pruning (DCP) [34] introduces the auxiliary loss of discrimination perception in the fine-tuning and channel selection stage, and the filters are selected to be pruned by joint optimization to minimize the reconstruction loss and the network loss. In the Sparse Structure Selection (SSS) [35], Global & Dynamic Filter Pruning (GDP) [36] and Generative Adversarial Learning (GAL) [37] pruning method, a mask is introduced to learn sparse structure pruning, and the mask is adjusted adaptively during the optimization training. After optimization, the filters with zero scaling factor are pruned.

However, because the network structures are changed, the retraining usually requires introducing additional parameters. These methods are not flexible enough to deploy in the existing network structure.

3. Methods

3.1. Preliminaries

For a given pre-trained CNN model with K convolution layers. The model weights are denoted by $\left\{\mathbf{W}_i \in \mathbb{R}^{N_{i+1}\times N_i\times k_i\times k_i}, 1 \le i \le K\right\}$. Here, N_i, N_{i+1} and k_i represent the number of input channels, the output channels, and the kernel size for i-th convolution layer, respectively. The j-th filter of the i-th convolution layer is also represented as $\mathbf{W}_i^j \in \mathbb{R}^{N_i\times k_i\times k_i}$. $\mathbf{F}_i^j \in \mathbb{R}^{h_i\times w_i}$ represents the feature maps generated by $\mathbf{W}_i^j$. Furthermore, h_i and w_i represent the height and width of the feature maps, respectively. In filter pruning, a binary indicator $m_i^j \in \{0,1\}$ is introduced to determine whether the j-th filter of i-th convolution layer ought to be pruned, where $m_i^j = 1$ means the j-th filter and the corresponding feature maps $\mathbf{F}_i^j$ are pruned.

3.2. Mathematical Model of Filter Pruning Framework

In this paper, the pruning rate of the i-th layer is denoted as P_i and prune each layer independently. For the i-th layer, the purpose of filter pruning is to obtain and remove the $N_{i+1} \times P_i$ less important filters from $\mathbf{W}_i$. This can be formulated as the following optimization problem:

$$\begin{aligned} \min_{m_i^j} \sum_{i=1}^{K}\sum_{j=1}^{N_{i+1}} m_i^j \mathcal{L}\left(\mathbf{W}_i^j\right) \\ \text{s.t.} \sum_{j=1}^{N_{i+1}} m_i^j = N_{i+1} \times P_i \end{aligned} \tag{1}$$

where $\mathcal{L}\left(\mathbf{W}_i^j\right)$ is a function to evaluate the importance of a filter $\mathbf{W}_i^j$. In this framework, most the existing weight-based techniques directly design $\mathcal{L}$ on weights. As reported in Hrank [31], weight-based pruning methods ignore the distribution of input images. Therefore, such methods might not accurately evaluate the filter importance for every dataset. Feature maps are the product of input images in network transmission, hence their importance of them can be regarded as a filter property as they are generated by matrix operations on weights. Considering the distribution of datasets, to achieve filter pruning, instead of weights, feature map-based pruning defines $\hat{\mathcal{L}}$ on the feature maps. Therefore, for evaluating the importance of feature maps, Equation (1) can be reformulated as:

$$\begin{aligned} \min_{m_i^j} \sum_{i=1}^{K}\sum_{j=1}^{N_{i+1}} m_i^j \mathbb{E}_{I\sim P(I)}\left[\hat{\mathcal{L}}\left(\mathbf{F}_i^j(I)\right)\right] \\ \text{s.t.} \sum_{j=1}^{N_{i+1}} m_i^j = N_{i+1} \times P_i \end{aligned} \tag{2}$$

where I represents an input image and $P(I)$ represents the distribution of input images. Furthermore, $\mathbf{F}_i^j(I)$ is a feature map generated by $\mathbf{W}_i^j$. $\hat{\mathcal{L}}\left(\mathbf{F}_i^j(I)\right)$ evaluates the importance of a feature map $\mathbf{F}_i^j(I)$. The difference between the feature map-based pruning algorithms mainly lies in the design of $\hat{\mathcal{L}}$. In this paper, an attention consistency discriminant module (ACDM) is designed as the evaluation function $\hat{\mathcal{L}}$. ACDM evaluates the attention consistency of the feature maps. It shows that the feature maps with higher attention consistency are more important.

3.3. Implementation of ACDM

Attention consistency can be regarded as a feature maps property measuring the distribution of attention on the feature maps of each layer. There exist several methods to measure the performance of attention consistency. For example, PCA and mean shift algorithms can be used to help measure attention consistency. Here the centroid-based implementation method is adopted.

A centroid is a hypothetical point on which the weight of the body is concentrated. Weights are expressed as a pixel value on a feature map. According to the findings, most feature maps have larger activation values in the attention area, hence, for such feature maps, the centroids will fall in the attention area. The centroid coordinates of a feature map are:

$$\begin{aligned} Ch_i^j(I) &= \frac{\sum_{x=1}^{h_i}\sum_{y=1}^{w_i} x \cdot \mathbf{F}_i^j(I,x,y)}{\sum_{h=1}^{h_i}\sum_{w=1}^{w_i} \mathbf{F}_i^j(I,x,y)} \\ Cw_i^j(I) &= \frac{\sum_{x=1}^{h_i}\sum_{y=1}^{w_i} y \cdot \mathbf{F}_i^j(I,x,y)}{\sum_{h=1}^{h_i}(I)\sum_{w=1}^{w_i} \mathbf{F}_i^j(I,x,y)} \end{aligned} \tag{3}$$

where x and y represent the coordinates on the h-axis and w-axis, respectively. $\mathbf{F}_i^j(I,x,y)$ is the pixel value of $\mathbf{F}_i^j(I)$. Ch_i^j and Cw_i^j represent h-axis and w-axis coordinates of the centroid on a feature map. Most of the feature maps have large activation values in the attention. In the i-th layer, the average centroid is:

$$\begin{aligned} Ch_i(I) &= \frac{\sum_{j=1}^{N_i+1} Ch_i^j(I)}{N_{i+1}} \\ Cw_i(I) &= \frac{\sum_{j=1}^{N_i+1} Cw_i^j(I)}{N_{i+1}} \end{aligned} \tag{4}$$

The distance of centroid deviation is also defined to evaluate the attention consistency of a feature map. In the i-th layer, the distance of centroid deviation is:

$$dis_i^j(I) = \left(Ch_i^j(I) - Ch_i(I)\right)^2 + \left(Cw_i^j(I) - Cw_i(I)\right)^2 \tag{5}$$

For a given input image, feature maps are generated in the forward propagation of every convolution layer. Figure 3 presents the centroid distribution of feature maps generated by the 6-th convolutional layer in Resnet-50. The x-axis and the y-axis represent centroid coordinates along the h and w directions of the feature maps. The blue dots represent the centroid of a feature map, and the red dot represents the average centroid. Several feature maps with different centroids at different distances are visualized. It is seen from these feature maps that the information is concentrated in the attention area in the feature maps with smaller deviations, whereas the information is concentrated in the non-attention area in feature maps with bigger deviations. It means that the feature maps with higher consistency have smaller deviations. Similar to such input images, the feature maps generated by the network with any other input images also have the same representation.

The prerequisite for this metric to guide pruning is that the metric can reflect the centroid deviations of entire datasets. The experimental results also show that the centroid deviations generated by a single filter are robust to the input images. It is also seen from the experiments that the expectation of centroid deviations generated by a single filter is robust to the input images. As shown in Figure 4, for each subfigure, the x-axis represents the indices of feature maps and the y-axis is the number of training images. Different colors denote distances of centroid deviation. It is also seen that the average centroid deviation of multiple feature maps generated by the same filter (column of the figure) are almost equal (the same color), regardless of the number of images (ordinate of the subfigure) the CNNs receive. Therefore, even a small number of input images can be used to estimate the average centroid of deviation of entire datasets. The centroid deviation expectation of feature maps is defined as $\sum_{t=1}^{g} dis_i^j(I_t)$, where g is the total number of input images sampled from the dataset.

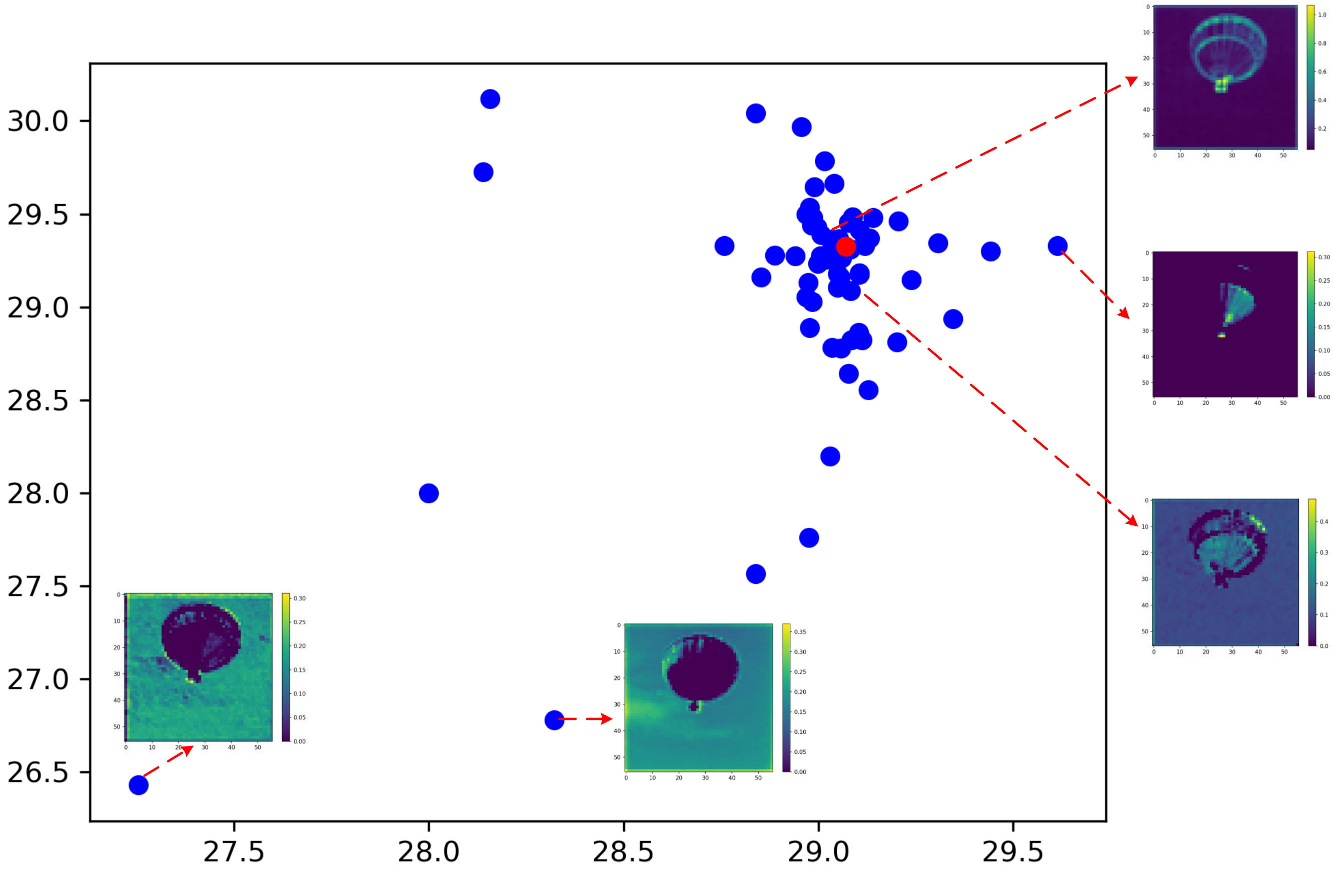
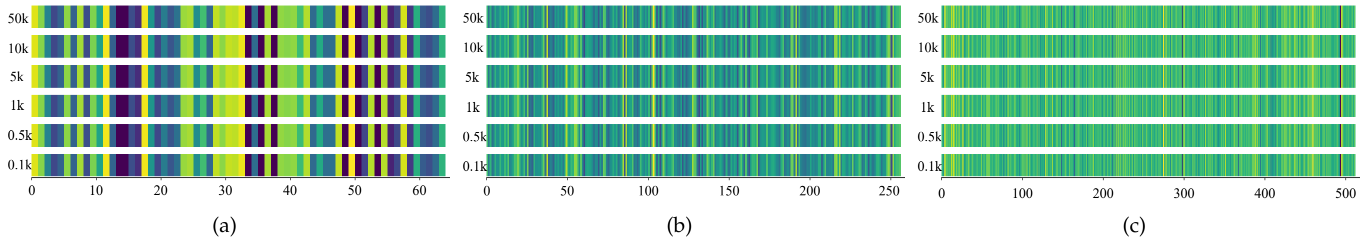


Figure 3. The centroid distribution of the feature maps.



(a)

(b)

(c)

Figure 4. The expectation of centroid deviations of the feature maps with different sampled images. (**a**) VGG16_1; (**b**) VGG16_5; (**c**) VGG16_10.

3.4. Filter Pruning Based on Attention

Note that the importance of a filter is measured by evaluating the centroid deviation of the feature maps generated by a filter. Therefore, the filter Pruning problem based on the centroid deviation can be reformulated as:

$$\begin{gathered}\min_{m_i^j} \sum_{i=1}^{K} \sum_{j=1}^{N_{i+1}} m_i^j \sum_{t=1}^{g} \left(-dis_i^j(I_t)\right) \\ \text{s.t.} \sum_{j=1}^{N_{i+1}} m_i^j = M_i \end{gathered} \tag{6}$$

The centroid deviation is a robust estimator of the importance of feature maps. For the i-th feature maps, the average centroid deviation is calculated in F_i. Then M_i feature maps with bigger average centroid deviation are chosen as the less important and the filters that generate these feature maps are pruned. After fine-tuning, the network might be able to achieve performance very close to its original performance. Therefore, the filters could be pruned with negligible effect on the final result of the neural network.

4. Experiment and Analysis

4.1. Experimental Settings

Evaluation metrics and Datasets: To evaluate the performance of FPAC, the widely-used CNN compressed evaluation metrics are adopted, i.e., required Floating Points Operations (FLOPs). The number of parameters is compared to the state-of-art pruning methods for the image classification task. FPAC is compared with the currently popular methods for image classification tasks. The benchmark datasets used in the experiment are CIFAR-10 [38] and ImageNet [39]. On the CIFAR-10 dataset, the top-1 accuracy and pruning ratio (PR) of the different methods are presented. On the ImageNet dataset, the top-5 accuracy is also presented.

Configurations. All of the experiments are implemented in the PyTorch framework with the Stochastic Gradient Descent algorithm (SGD) optimization strategy. Data argumentation strategies are the same as in the PyTorch official examples. On CIFAR-10, the networks are fine-tuned with a momentum of 0.9 for 150 epochs with a batch size of 256. The learning rate starts from 0.1 and decreases by a factor of 10 every 50 epochs. On ImageNet, the weight decay is set to 1e-4 and 90 epochs are given for fine-tuning. The learning rate is set as 0.1 and divided by 10 every 30 epochs. Referring to the implementation of HRank, pruning ratios are set at each layer to determine the number of filters pruned.

4.2. Results and Analysis

4.2.1. Results on CIFAR-10

To verify the generality of the proposed method, the pruning experiments are presented on three popular network structures including single-branch (VGG-16 [5]), multiple-branch (ResNet-56/110 [40] and DenseNet-40 [41]) and Inception (GoogLeNet [4]) structures. The proposed method is compared with the state-of-the-art on CIFAR-10 dataset. Note that, referring to the original paper, the initial accuracy of the baseline is different from FPAC in several comparison methods.

Single-branch structure: To verify pruning efficiency on the single-branch network, the pruning experiments on VGG-16 are presented. The experimental results on VGG-16 are shown in Table 1. The results with different methods including weight-based methods and server feature map-based methods are compared. Firstly, the weight-based methods, e.g., L1 and FPGM, perform well with a low pruning ratio, especially the L1 method. However, with the guidance of data prior information, the proposed method achieves higher accuracy with a larger FLOPs and parameters pruning ratio. For example, compared with FPGM, FPAC achieves better accuracy (0.07% increase by FPAC vs. 0.04% decrease by FPGM) with a larger FLOPs reductions ratio (58.1% by FPAC vs. 34.2% by FPGM). Secondly, HRank, as a feature map-based method, makes full use of the distribution information of feature maps and shows good performance with a higher pruning ratio. Because of the introduction of the attention mechanism to the feature maps, the proposed method outperforms in all aspects (accuracy and pruning ratio). For example, FPAC achieves a higher accuracy (93.86% by FPAC vs. 93.43% by FPGM) with the larger FLOPs reductions ratio (66.6% by FPAC vs. 53.5% by HRank). Thirdly, compared with adaptive importance-based methods, e.g., GAL, the proposed method provides significantly better accuracy and a higher reduction in the number of parameters and FLOPs.

Multiple-branch structure: The pruning efficiency is evaluated on three popular multiple-branch networks including ResNet-56, ResNet-110, and DenseNet-40. For ResNet-56 and ResNet-110, shortcuts are simply implemented using padding rather than 1×1 convolution similar to HRank. For DenseNet-40, shortcuts are implemented by channel-wise concatenation. For simplicity, in cases where the feature maps generated by the convolution layer have the same resolution, the compression ratios are set to the same. Because of the above treatment of ResNet and DenseNet, the convolution layer can be compressed such as a single-branch network and keep a balance of information from the shortcut and convolution layer. Considering fairness and comprehensive comparison, the implementation of the proposed method with several pruning ratios is presented. Under similar FLOPs reductions with the existing methods, the proposed method obtains an excellent top-1 accuracy which is even much better than the baseline model at a low pruning ratio. For example, for ResNet-56, the accuracy of FPAC is increased by 0.45% with 47.4% FLOPs reductions, which is higher than other methods. It shows that the proposed method still works on a multiple-branch network.

Inception structure: Except for single-branch and multiple-branch structures, inception structure is widely used in popular networks such as GoogLeNet. To verify the generality of the proposed method, the proposed method is tested on GoogLeNet. In the implementation, only the 3 × 3 convolution layers are pruned and obtain an equal pruning ratio on the convolution layer of one inception module. Due to the complexity of the inception structure, using most of the existing pruning methods reduces the accuracy. The last column of Table 1 presents the performance of several existing methods which have an experiment on GoogLeNet including weight-based methods, e.g., L1, feature map-based methods, e.g., HRank and adaptive importance-based method GAL. At a lower pruning ratio, all the existing methods yield a loss of accuracy of at least 0.5, while the loss of accuracy in the proposed method is almost none. Furthermore, the proposed method performs better at a higher pruning ratio than all three methods with a lower pruning ratio, which demonstrates the superiority of exploiting the attention of feature maps as an intrinsic property of CNNs.

4.2.2. Results on ImageNet

For the challenging ImageNet dataset, Table 2 presents the experiments on ResNet-18 and ResNet-50 and the results. It is seen that in general, the proposed method surpasses its counterparts in top-1 and top-5 accuracies, FLOPs and parameters reduction. For ResNet-18, 48.6% FLOPs and 48.7% parameters are reduced by the proposed method while it still yields 68.91% top-1 accuracy and 88.64% top-5 accuracy performing significantly better than other methods. For ResNet-50, three pruning ratios are tested to compare with all methods. Under low pruning ratios, the proposed method obtains 75.62% top-1 accuracy and 92.63% top-5 accuracy with 45.0% and 40.9% reductions of FLOPs and parameters, respectively. The proposed method, therefore, shows more advantages in FLOPs and parameter reductions and outperforms SSS, CP, SFP, and HRank. For high pruning ratios, the proposed method achieves higher complexity reductions (76.7% FLOPs and 68.6% parameters) and higher accuracy (72.30% top-1 accuracy and 90.74% top-5 accuracy) than ThiNet, GAL-1-joint, and HRank. Hence, the proposed method also works well on complex datasets.

Table 1. Pruning results on CIFAR-10. "Top-1": accuracy. "Top-1↓": decrease in accuracy, smaller is better. A negative value here indicates an improved model accuracy. "FLOPs(PR)": FLOPs(pruning ratio). "Params(PR).": parameters(pruning ratio). "-": results not reported in the corresponding paper.

Baseline	Method	Top-1(%)	Top-1↓(%)	FLOPs(PR)	Params(PR)
VGG-16	FPGM [23]	93.58 → 93.54	0.04	206.43M (34.2%)	-
	L1 [21]	93.25 → 93.40	−0.15	206.12M (34.3%)	5.40M (64.0%)
	VP [32]	93.25 → 93.18	0.07	190.00M (39.1%)	3.92M (73.3%)
	SSS [35]	93.96 → 93.02	0.84	183.13M (41.6%)	3.93M (73.8%)
	GAL-0.05 [37]	93.96 → 92.03	1.93	189.49M (39.6%)	3.36M (77.6%)
	FPAC	93.96 → 94.03	−0.07	131.17M (58.1%)	2.76M (81.6%)
	GAL-0.1 [37]	93.96 → 90.73	3.23	171.89M (45.2%)	2.67M (82.2%)
	HRank [31]	93.96 → 93.43	0.53	145.61M (53.5%)	2.51M (82.9%)
	FPAC	93.96 → 93.86	0.10	104.78M (66.6%)	2.50M (83.3%)
ResNet-56	GAL-0.6 [37]	93.26 → 92.98	0.28	78.30M (37.6%)	0.75M (11.8%)
	L1 [21]	93.04 → 93.06	−0.02	90.90M (27.6%)	0.73M (14.1%)
	HRank [31]	93.26 → 93.52	−0.26	88.72M (29.3%)	0.71M (16.8%)
	FPAC	93.26 → 94.35	−1.09	90.35M (28.0%)	0.66M (22.3%)
	NISP [24]	93.26 → 93.01	0.15	81.00M (35.5%)	0.49M (42.4%)
	FPAC	93.26 → 93.71	−0.45	65.94M (47.4%)	0.48M (42.8%)
	CP [28]	92.80 → 91.80	1.00	62.00M (50.6%)	-
	FPGM [23]	93.59 → 93.49	0.10	59.40M (52.6%)	-
	GAL-0.8 [37]	93.26 → 90.36	2.90	49.99M (60.2%)	0.29M (65.9%)
	HRank [31]	93.26 → 90.72	2.54	32.52M (74.1%)	0.27M (68.1%)
	FPAC	93.26 → 92.37	0.89	34.78M (74.1%)	0.24M (70.0%)
ResNet-110	L1 [21]	93.53 → 93.30	0.23	155.00M (38.6%)	1.16M (32.4%)
	HRank [31]	93.50 → 94.23	−0.73	148.70M (41.2%)	1.04M (39.4%)
	FPAC	93.50 → 94.52	−1.02	140.54M (44.4%)	1.04M (39.4%)
	FPGM [23]	93.68 → 93.85	−0.27	121.00M (52.3%)	-
	NISP [24]	93.50 → 93.32	0.18	142.17M (43.8%)	0.80M (43.3%)
	GAL-0.5 [37]	93.50 → 92.55	0.95	130.20M (48.5%)	0.95M (44.8%)
	HRank [31]	93.50 → 93.36	0.14	105.70M (58.2%)	0.70M (59.2%)
	FPAC	93.50 → 93.49	0.01	71.69M (71.6%)	0.54M (68.3%)
DenseNet-40	GAL-0.01 [37]	94.81 → 94.29	0.52	182.92M (35.3%)	0.67M (35.6%)
	Slimming [33]	94.81 → 94.81	0.00	190.00M (32.8%)	0.66M (36.5%)
	HRank [31]	94.81 → 94.24	0.57	167.41M (40.8%)	0.66M (36.5%)
	FPAC	94.81 → 94.51	0.30	173.39M (38.5%)	0.62M (40.1%)
	HRank [31]	94.81 → 93.68	1.13	110.15M (61.0%)	0.48M (53.8%)
	GAL-0.05 [37]	94.81 → 93.53	1.28	128.11M (54.7%)	0.45M (56.7%)
	VP [32]	94.11 → 93.16	0.95	156.00M (44.8%)	0.42M (59.7%)
	FPAC	94.81 → 93.66	1.15	113.08M (59.9%)	0.39M (61.9%)
GoogLeNet	L1 [21]	95.05 → 94.54	0.51	1.02B (32.9%)	3.51M (42.9%)
	GAL-0.05 [37]	95.05 → 93.93	1.12	0.94B (38.2%)	3.12M (49.3%)
	FPAC	95.05 → 95.04	0.02	0.65B (57.2%)	2.85M (53.5%)
	GAL-ApoZ [37]	95.05 → 92.11	2.94	0.76B (50.0%)	2.85M (53.7%)
	HRank [31]	95.05 → 94.53	0.52	0.69B (54.9%)	2.74M (55.4%)
	HRank [31]	95.05 → 94.07	0.98	0.45B (70.4%)	1.86M (69.8%)
	FPAC	95.05 → 94.42	0.63	0.40B (73.9%)	2.09M (65.8%)

Table 2. Pruning results of ResNet on ImageNet. "Top-1": accuracy. "Top-1↓": decrease in accuracy, smaller is better. A negative value here indicates an improved model accuracy. "Top-5↓": decrease of Top-5. "FLOPs(PR)": FLOPs(pruning ratio). "Params(PR).": parameters(pruning ratio). "-": results not reported in the corresponding paper.

Baseline	Method	Top-1(%)	Top-1↓(%)	Top-5(%)	Top-5↓(%)	FLOPs(PR)	Params(PR)
ResNet-18	Sampling [27]	69.76 → 67.38	2.38	89.55 → 88.43	1.12	1.28B (29.3%)	6.57M (43.8%)
	SFP [22]	70.28 → 67.10	3.18	89.63 → 87.78	1.85	1.05B (41.8%)	-
	FPGM [23]	70.28 → 68.41	1.87	89.63 → 88.48	1.15	1.05B (41.8%)	-
	Hrank [31]	70.28 → 68.56	1.72	89.63 → 88.58	1.05	1.01B (44.2%)	6.23M (46.7%)
	DCP [34]	69.76 → 67.35	2.41	89.55 → 88.45	1.10	0.98B (45.8%)	6.19M (46.6%)
	FPAC	70.28 → 68.91	1.67	89.64 → 88.64	1.00	0.93B (48.6%)	6.00M (48.7%)
ResNet-50	SSS-32 [35]	76.12 → 74.18	1.94	92.87 → 91.91	0.96	2.82B (31.1%)	18.60M (27.3%)
	CP [28]	74.99 → 72.84	2.15	92.20 → 90.80	1.40	2.73B (32.8%)	-
	SFP [22]	76.15 → 74.61	1.54	92.87 → 92.06	0.81	2.38B (41.2%)	-
	FPGM [23]	76.15 → 75.59	0.56	92.87 → 92.63	0.24	2.36B (42.2%)	-
	SSS-26 [23]	76.12 → 71.82	4.30	92.87 → 90.79	2.08	2.33B (43.3%)	15.60M (38.9%)
	GAL-0.5 [37]	76.15 → 71.95	4.20	92.87 → 90.04	2.83	2.33B (43.3%)	21.20M (17.0%)
	HRank [31]	76.15 → 74.98	1.17	92.87 → 92.33	0.54	2.30B (44.0%)	16.15M (36.8%)
	NISP [24]	-	0.89	-	-	2.29B (44.0%)	14.38M (43.8%)
	FPAC	76.15 → 75.62	0.53	92.87 → 92.63	0.24	2.26B (45.0%)	15.09M (40.9%)
	FPGM [23]	76.15 → 74.83	1.32	92.87 → 92.32	0.55	1.90B (53.5%)	-
	GDP-0.6 [36]	75.13 → 71.19	3.94	92.87 → 90.71	2.16	1.88B (54.3%)	-
	GAL-0.5-joint [37]	76.15 → 71.80	4.35	92.87 → 90.82	2.05	1.84B (55.2%)	19.31M (24.4%)
	ThiNet [26]	72.88 → 71.01	1.87	91.94 → 90.30	1.64	1.82B (55.8%)	12.40M (51.6%)
	GAL-1 [37]	76.15 → 69.88	6.27	92.87 → 90.14	2.73	1.58B (61.6%)	14.67M (42.4%)
	GDP-0.5 [36]	75.13 → 69.58	6.54	92.87 → 90.14	2.73	1.57B (61.8%)	-
	HRank [31]	76.15 → 71.98	4.17	92.87 → 91.01	1.86	1.55B (62.3%)	13.77M (46.1%)
	FPAC	76.15 → 74.17	1.85	92.87 → 91.84	1.03	1.52B (63.0%)	11.05M (56.7%)
	GAL-1-joint [37]	76.15 → 69.31	6.85	92.87 → 89.12	3.75	1.11B (73.0%)	10.21M (60.0%)
	ThiNet [26]	72.88 → 68.42	4.46	91.94 → 88.30	3.64	1.10B (73.2%)	8.66M (66.1%)
	HRank [31]	76.15 → 69.10	7.05	92.87 → 89.58	3.29	0.98B (76.2%)	8.27M (67.6%)
	FPAC	76.15 → 72.30	3.85	92.87 → 90.74	2.13	0.95B (76.7%)	8.02M (68.6%)

5. Ablation Study

The detailed ablation experiments are performed to demonstrate the effectiveness of pruning the feature maps with low attention consistency. For simplicity, only the performance of ResNet-56 on the CIFAR-10 dataset is discussed. The following four different comparison methods at the same pruning ratio are proposed to prove that the filter pruned by FPAC is reasonable: (1) Baseline: the original ResNet-56 network and it maintains the original number of filters; (2) Edge: filters that generate feature maps with lower and higher attention consistency are pruned; (3) Random: with the same number of filters removed from each layer of FPAC, several filters are selected randomly to prune; (4) Reverse: filters that generate feature maps with higher attention consistency are pruned. In all these experiments, the pruning rate is set to 47.4%, and the parameters rate is set to 42.8%.

Table 3 presents the corresponding top-1 accuracy of FPAC for the comparison methods. All five methods present higher classification accuracy. It shows that the filters of ResNet-56 are indeed redundant on the CIFAR-10 dataset. Due to the existence of redundancy, after fine-tuning, there is no significant decrease in these methods. It is seen that the performance of FPAC surpasses the baseline and all other comparison methods. It is also shown that FPAC retains important filters. Note that although part of the feature maps with lower or higher attention consistency is removed, the retained feature maps with lower attention consistency contain rich information. It is also seen that Edge performs well and Random is also better than Reverse. The latter is because significant filters are pruned by Reverse which further demonstrates the effectiveness of FPAC.

Table 3. Ablation experiments results of ResNet-56 on CIFAR-10. "Top-1": accuracy. "FLOPs(PR)": FLOPs(pruning ratio). "Params(PR).": parameters(pruning ratio).

Method	FLOPs(PR)	Params(PR)	Top-1(%)
Baseline	65.94M (47.4%)	0.48M (42.8%)	93.26%
FPAC			93.71%
Edge			92.34%
Random			92.35%
Reverse			91.25%

6. Conclusions

We investigated the attention of feature maps and proposed a novel pruning method, Filter Pruning via Attention Consistency on Feature maps (FPAC) to reduce the negative effects of the background information and noise. In the proposed method, the importance of filters is determined by estimating the attention consistency of the corresponding feature maps. Then, the distance of centroid deviation is adopted to calculate the attention consistency of feature maps. Feature maps with larger centroid deviation are then considered unimportant and their corresponding filters are accordingly pruned. The experiments showed the efficiency of FPAC on various popular network structures. It was also shown that even in cases with significant reductions in the number of network parameters and computational complexity, the model still maintained a similar accuracy. In the future, the relationship between the attention mechanism and network pruning will be further explored.

Author Contributions: Conceptualization, H.Y.; methodology, H.Y. and F.M.; software, H.Y.; formal analysis, Y.L.; investigation, H.Y. and W.L.; data curation, H.Y.; writing-original draft preparation, H.Y.; writing-review and editing, Y.L. and W.L.; visualization, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61871154, No. 62031013), by the Guangdong Province key construction discipline scientific research capacity improvement project (No. 2022ZDJS117).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was implied from all subjects involved in the study by completing the survey.

Data Availability Statement: Data are available upon reasonable request to the submitting author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef] [PubMed]
2. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
4. Szegedy, C.; Liu, W.; Jia Y.; Sermanet P.; Reed S.; Anguelov D.; Erhan D.; Vanhoucke V.; Rabinovic A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. Taigman, Y.; Yang, M.; Ranzato, M. A.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA , 23–28 June 2014; pp. 1701–1708.
7. Denil, M.; Shakibi, B.; Dinh, L.; Ranzato, M.A.; De Freitas, N. Predicting parameters in deep learning. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2148–2156.

8. Denton, E.L.; Zaremba, W.; Bruna, J.; LeCun, Y.; Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1269–1277.
9. Lin, S.; Ji, R.; Chen, C.; Tao, D.; Luo, J. Holistic cnn compression via low-rank decomposition with knowledge transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2889–2905. [CrossRef] [PubMed]
10. Zhang, X.; Zou, J.; Ming, X.; He, K.; Sun, J. Efficient and accurate approximations of nonlinear convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1984–1992.
11. Hou, Z.; Kung, S.Y. A feature map discriminant perspective for pruning deep neural networks. *arXiv* **2020**, arXiv:2005.13796.
12. Hu, H.; Peng, R.; Tai, Y.W.; Tang, C.K. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv* **2016**, arXiv:1607.03250.
13. Qian, X.; Klaban, D. A probabilistic approach to neural network pruning. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 8640–8649.
14. Rosenfeld, S.; Frankle, J.; Carbin, M.; Shavit, N. On the predictability of pruning across scales. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 9075–9083.
15. Amelio, A.; Bonifazi, G.; Corradini, E.; Ursino, D.; Virgili, L. A Multilayer Network-Based Approach to Represent, Explore and Handle Convolutional Neural Networks. *Cogn. Comput.* **2022**, 1–29. [CrossRef]
16. Amelio, A.; Bonifazi, G.; Cauteruccio, F.; Corradini, E.; Marchetti, M.; Ursino, D.; Virgili, L. Representation and compression of Residual Neural Networks through a multilayer network based approach. *Expert Syst. Appl.* **2023**, *215*, 119391. [CrossRef]
17. Chen, W.; Wilson, J.T.; Tyree, S.; Weinberger, K.; Chen, Y. Compressing neural networks with the hashing trick. In Proceedings of the International Conference on Machine Learning, Lille, France, 11 July 2015; pp. 2285–2294.
18. Wu, J.; Leng, C.; Wang, Y.; Hu, Q.; Cheng, J. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4820–4828.
19. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Deng, H.; Ju, Q. Fastbert: A self-distilling bert with adaptive inference time. *arXiv* **2020**, arXiv:2004.02178.
20. Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient knowledge distillation for bert model compression. *arXiv* **2019**, arXiv:1908.09355.
21. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning filters for efficient convnets. *arXiv* **2016**, arXiv:1608.08710.
22. He, Y.; Kang, G.; Dong, X.; Fu, Y.; Yang, Y. Soft filter pruning for accelerating deep convolutional neural networks. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 2234–2240.
23. He Y, Liu P, Wang Z.; Hu, Z.; Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4340–4349.
24. Yu, R.; Li, A.; Chen, C.F.; Lai, J.H.; Morariu, V.I.; Han, X.; Gao,M.; Lin, C.Y.; Davis, L.S. Nisp: Pruning networks using neuron importance score propagation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9194–9203.
25. Zhuo H, Qian X, Fu Y.; Yang, H.; Xue, X. Scsp: Spectral clustering filter pruning with soft self-adaption manners. *arXiv* **2018**, arXiv:1806.05320.
26. Luo, ; H.; Wu, J.; Lin, W. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5058–5066.
27. Liebenwein, L.; Baykal, C.; Lang, H.; Feldman, D.; Rus, D. Provable filter pruning for efficient neural networks. *arXiv* **2019**, arXiv:1911.07412.
28. He, Y.; Zhang, X.; Sun, J. Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1389–1397.
29. Suau, X.; Zappella, L.; Palakkode, V.; Apostoloff, N. Principal filter analysis for guided network compression. *arXiv* **2018**, arXiv:1807.10585.
30. Wang, D.; Zhou, L.; Zhang, X.; Bai, X.; Zhou, J. Exploring linear relationship in feature map subspace for convnets compression. *arXiv* **2018**, arXiv:1803.05729.
31. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1529–1538.
32. Zhao, C.; Ni, B.; Zhang, J.; Zhao, Q.; Zhang, W.; Tian, Q. Variational convolutional neural network pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2780–2789.
33. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
34. Zhuang, Z.; Tan, M.; Zhuang, B.; Liu, J.; Guo, Y; Wu, Q.; Huang, J.; Zhu, J. Discrimination-aware channel pruning for deep neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *40*, 4035–4051.
35. Huang, Z.; Wang, N. Data-driven sparse structure selection for deep neural networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 304–320.
36. Lin, S.; Ji, R.; Li, Y.; Wu, Y.; Huang, F.; Zhang, B. Accelerating Convolutional Networks via Global & Dynamic Filter Pruning. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 2425–2432.

37. Lin, S.; Ji, R.; Yan, C.; Zhang, B.; Cao, L.; Ye, Q.; Huang, F.; Doermann, D. Towards optimal structured cnn pruning via generative adversarial learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2790–2799.
38. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images (Technical Report). University of Toronto, Canada, 2009. Available online: https://www.cs.toronto.edu/ kriz/cifar.html (accessed on 23 June 2022).
39. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. Ournal Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
41. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.