Rui Lin\textsuperscript{1,}\textsuperscript{*} Jie Ran\textsuperscript{1,}\textsuperscript{*} Dongpeng Wang\textsuperscript{2} King Hung Chiu\textsuperscript{2} Ngai Wong\textsuperscript{1}\\\textsuperscript{1} Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong\textsuperscript{2} United Microelectronics Centre (Hong Kong) Limited, Hong Kong Science Park, N.T., Hong Kong\\Email Address: \{linrui, jieran, nwong\}@eee.hku.hk, \{khchiu\}@umechk.com

\section*{Abstract}
Recent results have revealed an interesting observation in a trained convolutional neural network (CNN), namely, the rank of a feature map channel matrix remains surprisingly constant despite the input images. This has led to an effective rank-based channel pruning algorithm [23], yet the constant rank phenomenon remains mysterious and unexplained. This work aims at demystifying and interpreting such rank behavior from a frequency-domain perspective, which as a bonus suggests an extremely efficient Fast Fourier Transform (FFT)-based metric for measuring channel importance without explicitly computing its rank. We achieve remarkable CNN channel pruning based on this analytically sound and computationally efficient metric, and adopt it for repetitive pruning to demonstrate robustness via our scheme named Energy-Zoned Channels for Robust Output Pruning (EZCrop), which shows consistently better results than other state-of-the-art channel pruning methods. The codes and Appendix are publicly available at: https://github.com/ruilin0212/EZCrop.

1. Introduction

Convolutional neural networks (CNNs) are among the most popular models for deep learning which have achieved breakthroughs in vision applications including classification [41], object detection [28], semantic segmentation [34], etc. However, deeper and larger CNNs render them challenging to deploy on edge devices with constrained hardware resources. This dilemma has motivated the research on compressing CNNs for lower storage and faster inference. Numerous neural network compression techniques have been proposed, including optimized implementation [16], quantization [2, 5], network pruning [10], low-rank decomposition [25, 17] and knowledge distillation [12, 33], etc. Among the categories mentioned above, this work focuses on channel pruning, a subclass of structured network pruning.

\*RL, JR contributed equally to this work.
ponents dynamically. In [10] “smaller-norm-less-important” filter pruning is proposed, which removes filters with smaller norms. Recently, Lin et al. [23] observe that the average rank of feature maps generated by a fixed filter remains surprisingly constant irrespective of the input images. They then propose a scheme called HRank to prune channels with small corresponding ranks and thereby less information content. Noticeably, most existing pruning schemes operate in the spatial domain, with only a few of them utilizing information from the frequency or spectral domain. Inspired by the work of Sedghi et al. [39] that studies the singular values in a convolution layer from a Fast Fourier Transformation (FFT) viewpoint, we take FFT as a tool for deciphering the constant rank behavior [23], and develop a filter pruning scheme with a novel, analytical, energy-based explanation. Our key insight is that the energy distributions of feature maps generated by a fixed filter can reflect the filter’s importance (Figure 1). Alongside this observation, we propose a novel frequency-domain channel pruning approach called Energy-Zoned Channel for robust output pruning (EZCrop), which entails a natural, accurate, and more robust selection of important filters at a lower computational cost than the singular value decomposition (SVD) needed for evaluating ranks in HRank [23].

In order to make a comprehensive comparison versus HRank [23] and other state-of-the-art algorithms [20, 13, 29, 32, 45, 15, 27, 10, 44, 19, 3, 42, 22, 30, 6, 24, 21], we benchmark EZCROP for image classification tasks on CIFAR-10 [18] and ImageNet [4], using VGGNet [40], ResNet [9], and DenseNet [14]. The experimental results demonstrate the consistent superiority of EZCROP in both model reduction and acceleration. Moreover, we conduct repetitive pruning with HRank and EZCROP, which further illustrates the high robustness of EZCROP. Our main contributions are:

- We analytically bridge the rank-based channel importance metric in the spatial domain to an energy perspective in the frequency domain, and for the first time explain the interesting constant-rank phenomenon in a channel matrix.
- We propose a computationally efficient FFT-based metric for channel importance, which reduces the computational complexity from $O(n^3)$ to $O(n^2 \log n)$.
- The proposed EZCROP algorithm for channel pruning is simple, intuitive and robust. It outperforms other state-of-the-art pruning schemes and consistently delivers good performance, which is not a result of normal variations as confirmed through extensive experiments.

2. Related Work

2.1. Filter Pruning

In contrast to weight pruning that results in unstructured sparsity, channel or filter pruning belongs to the class of structured pruning and can readily leverage the efficient BLAS library without specific software or hardware requirements. Generally, there are two categories of CNN filter pruning schemes, namely, by utilizing post-training properties of the CNN itself or by adaptive pruning during training. The first category employs algebraic metrics. For example, in [20], the authors use $l_1$-norm of the filters or the corresponding feature maps to evaluate the importance, where smaller norms mean less informative channels. In HRank [23], filters with small corresponding average ranks calculated by a batch of feature map slices are pruned. For the second category, adaptive pruning approaches make decisions through retraining with a specialized loss function taking pruning into account. For example, Liu et al. [29] impose sparsity-induced regularization on scaling factors in the batch normalization layers, where channels with zero-valued scaling factors are pruned. By reformulating the batch normalization layer, the authors of [46] propose a new parameter called saliency, and only channels with saliency beyond a threshold during training can remain.

2.2. Application of Frequency Information in the Convolutional Neural Networks

In recent years, the properties of CNNs in the frequency domain have attracted increasing attention due to the acceleration of CONV calculations and the additional information arising from the frequency domain. In [38], the authors propose spectral pooling that preserves more information and complex-coefficient spectral parameterization of filters that improves training convergence. Pratt et al. [37] propose a frequency-domain CNN, whose training can be done entirely in the frequency-domain without alternating between spatial and spectral domains. In [39], by using FFT properties, the authors design a regularizer to constrain the range of CONV kernels’ singular values during model training, which improves the model performance. Including [31], there are other works using spectral information for network compression. For example, in [1], the model is compressed in a frequency-sensitive fashion which preserves parameters of low-frequency components better. The filters are treated as images in [43], where the authors compress the model by discarding the low-energy coefficients of filters in the frequency domain.

Summarizing over the above works on filter pruning and frequency-domain CNN design, we notice the research on filter pruning through exploiting frequency-domain information is insufficient. The method in [31], though doing dynamic pruning in the frequency domain, belongs to unstructured and adaptive pruning, and suffers from machine and labor costs. Therefore, further exploring the use of spectral information for filter pruning is desirable.
3. Constant Ranks in Feature Map: From a Perspective in the Frequency Domain

The idea proposed in [39] by using FFT to constrain the singular values of kernels inspires us to use FFT to relate the rank-based metric in the spatial domain to an energy perspective in the frequency domain.

3.1. Notation

For a trained CNN model, we use $Y_i \in \mathbb{R}^{B \times T_i \times H_i \times W_i}$ to denote the output\(^1\) of the \(i\)-th CONV layer \((i = 1, 2, \ldots, N)\), where \(B\) is the batch size, \(T_i\) is the number of channels, \(H_i\) and \(W_i\) are the height and width, respectively. We denote the kernels of the \(i\)-th CONV layer as $K_i \in \mathbb{R}^{D_i \times D_i \times S_i \times T_i}$, where \(D_i\) represents the kernel size, \(S_i\) and \(T_i\) are the number of input and output channels. We use $Y_i[b, :, :, :] \in \mathbb{R}^{H_i \times W_i}$ to denote the feature map generated by the \(b\)-th sample in the batch passing through the \(j\)-th filter in the \(i\)-th CONV layer. The set $I_i = \{I_1^i, I_2^i, \ldots, I_{T_i}^i\}$ is used to record the importance of filters in the \(i\)-th CONV layer. Every element in the set $I_i$ records the index of one filter. We use $F(\cdot)$ and $F^{-1}(\cdot)$ to denote FFT and inverse FFT. The notation $\text{fftwshift}(\cdot)$ stands for the operation swapping four quadrants to center the DC component, $\text{abs}(\cdot)$ means element-wise magnitude, and $\text{ceil}(\cdot)$ means rounding up.

3.2. Convolution in the Frequency Domain

It is well-known that convolution in the time/spatial domain is equivalent to multiplication in the frequency domain. In [36], the author illustrates how to transform a general 2D convolution into a Fourier-based one. In brief, there are three steps: 1) for the given convolution kernel and input data, we apply 2D FFT on them respectively, 2) then we do point-wise multiplication between the newly obtained kernel and input, 3) lastly, we perform inverse 2D FFT on the multiplied result. Theorem 1 formalizes the convolution process in the frequency domain, and the detailed proof is provided in Appendix 1.

**Theorem 1** For a single 3-D input $X \in \mathbb{R}^{S \times H \times W}$ and a given filter $K \in \mathbb{R}^{D \times D \times S \times T}$, their convolution result $Y \in \mathbb{R}^{T \times H \times W}$ can be formalized as:

$$Y[j, :, :] = \sum_{i=1}^{S} F^{-1}(F(K[:, i, j]) \odot F(X[:, :, :]))$$

(1)

where $\odot$ stands for the point-wise multiplication (also called Hadamard product), and $K \in \mathbb{R}^{H \times W \times S \times T}$ is the expanded filter, whose slices $K[:, i, j]$ are the torus form of the corresponding slices $K[:, i, j]$ in the original filter.

By writing Eq. 1 alternatively as

$$F(Y[j, :, :]) = \sum_{i=1}^{S} F(K[:, i, j]) \odot F(X[:, :, :])$$

(2)

the input, filter, and the corresponding output are all in the frequency domain. The equivalence between spatial convolution and spectral point-wise product establishes the credibility of using frequency-domain information. Taking a sample image from CIFAR-10 and three filters from the first CONV layer in a pre-trained VGGNet, Figure 2 visualizes convolution in the frequency domain. More visualization results are available in Appendix 2.1.

---

\(^1\)The output selected here is after the operations like ReLU, batch normalization, and max pooling, etc. In other words, the output we extract for the \(i\)-th CONV layer is the input for the \((i + 1)\)-th CONV layer.
3.3. Energy Zoning

According to the rightmost column row of Figure 2, the concentration of low-frequency components of the generated feature map is inversely proportional to the average rank of the feature maps. Specifically, the average rank of the feature maps generated by the filter selected in the first row is the smallest, its low-frequency components of the feature map are the most concentrated. In the third row, the selected filter generates feature maps with high average ranks, and it is obvious that the frequency components of its feature map are much more dispersed. This phenomenon shows that a filter generating feature maps with high average ranks in the spatial domain is expected to generate feature maps with dispersed frequency components. Figure 3 uses a toy example to conceptually depict a low-rank channel matrix and a high-rank one. We use zero-valued and nonzero-valued elements to represent high-frequency and low-frequency components, respectively. It is obvious that the right matrix with a high rank has more dispersed nonzero elements.

To precisely quantify, we define a new metric called energy zone due to its underlying nature of summing spectral components related to power. The energy zone here refers to an area obtained by expanding outward from the DC center. Due to the conjugate symmetry after applying \( F(\cdot) \) and \( \text{fftshift}(\cdot) \) on the feature map, we define the energy zone with a symmetrical shape. For best visualization, we choose circle as the energy-zone shape, and choose a feature map with a large spatial size.

Figure 4 shows the results taking the first CONV in VGGNet having 64 filters as an example, the energy in each circle takes up 70% of the total feature map. The details of how to draw the circles are provided in Appendix 2.1. The 64 subfigures are arranged in an ascending order of the rank-based HRank scheme in [23] from left to right and top to bottom. In general, we can see that the radius of the circle increases as the average rank of the feature maps increases, which shows the high consistency between the rank-based scheme in the spatial domain and the dispersion of the feature maps’ low-frequency components in the frequency domain. However, the circle radii are not monotonically increasing, which means there are still subtle differences in the evaluation of certain filters based on information in the spatial and frequency domains. We remark that analyzing the low-frequency components’ concentration brings in higher resolution in channels’ importance evaluation since every element in \( H_i \times W_i \) is considered. In contrast, the rank of each feature map slice is only an integer in \([0, \min(H_i, W_i)]\). This is the reason why HRank and energy metric may produce subtle discrepancy in their channel importance measure. More discussions on the rank-based and energy-based metrics are in Appendix 3.

To further benchmark the difference in filter importance evaluation, we propose a pruning metric based on the energy zone in Section 4 and evaluate it through various experiments in Section 5.

4. EZCrop

Recall from Section 3.3 that the radius can be used to evaluate the concentration of the feature map’s frequency components. In particular, filters with a large average corresponding radius are treated as important ones. However, this method loses resolution when \( H_i \) and \( W_i \) of the feature maps \( Y_i \) are small. For example, it is difficult to determine a proper radius when the feature map slice is of size \( 4 \times 4 \) or smaller, such as \( 2 \times 2 \). Subsequently, a simple metric based on the concept of energy zone regardless of the feature maps’ size is desired. Here, we set the shape of an energy zone for filter importance evaluation.

---

2 We remark that the rank of a channel matrix is invariant in the spatial and spectral domains as 2D FFT is a bijective linear mapping.
Figure 5. An example to illustrate how to find the square center, and determine the selected energy area.

Figure 6. The two curves indicate the importance of the 64 filters using HRank and our energy-based metrics (Eq. 5) in the spatial and spectral domains, respectively. The dashed lines joining the points are only for the ease of visualization wherein the points are discrete in nature.

zone to be square and evaluate the importance of filters by four steps: 1) find the square center, 2) decide the size of the square, 3) calculate the energy-zone ratios, and 4) sort the filters according to the ratios. These steps are detailed below and the pseudocode can be found in Appendix 4.2.

**Step 1: Find the Square Center.** For a given feature map $Y_i$, as long as $H_i$ and $W_i$ are known, we can determine the coordinate of the DC center by

$$x_i = \begin{cases} \frac{H_i}{2} + 1, & H_i \text{ is even} \\ \frac{H_i + 1}{2}, & H_i \text{ is odd} \end{cases},$$

$$y_i = \begin{cases} \frac{W_i}{2} + 1, & W_i \text{ is even} \\ \frac{W_i + 1}{2}, & W_i \text{ is odd} \end{cases},$$

regardless of the parity of $H_i$ and $W_i$ or whether $H_i$ and $W_i$ are equal. Figure 5 visualize the square center when $H_i$ and $W_i$ are equal and are both even, and the schematic diagrams for other scenarios are available in Appendix 4.1.

**Step 2: Decide the Expanding Distance.** The size of a energy-zoned square is constrained by limiting the distance away from the center, accordingly:

$$l_{ih} = H_i - x_i, \quad l_{iw} = W_i - y_i,$$

$$d_i = \begin{cases} 0, & x_i - 1 = 0 \text{ or } y_i - 1 = 0 \\ \lceil \beta \cdot \min(l_{ih}, l_{iw}) \rceil, & \text{else} \end{cases},$$

where $\beta \in (0, 1)$ is a hyper-parameter to limit the area. It is worth noting that, $d_i = 0$ means only the DC component itself is contained in the energy-zoned square. Figure 5 shows the examples when $d_i = 0, 1, 2$.

**Step 3: Calculate the Energy Zone Ratios.** Different from Section 3.3, we determine the size of the energy zone first instead of the ratio of the sum of the elements in the energy zone. In this case, we evaluate the concentration of feature map’s low-frequency components by the proportion of the sum of elements outside the energy-zoned square:

$$\eta_i^j = \frac{1}{B} \left( 1 - \sum_{b=1}^{B} \frac{S(d_i[b])}{S(E^j_i[b, :, :])} \right), \quad (5)$$

where $S(d_i[b])$ and $S(E^j_i[b, :, :])$ denote the sum of elements in the energy-zoned square with $d_i$ as the extended length and the sum of elements in the whole square, respectively, for the $b$-th sample in a batch, and $\eta_i^j$ is the energy zone ratio of $j$-th filter in the $i$-th CONV layer, where larger value means more important the filter is. In Figure 6, we compare the results of the evaluation of the filters’ importance through HRank [23] and EZCrop. Consistent with the conclusion of Section 3.3, the two metrics generally track each other, especially for particularly important or unimportant channels/filters. For filters between the two poles, the evaluation results are slightly different. In Section 5, we use sufficient experiments to show EZCrop indeed provides a better scrutiny of important filters.

**Step 4: Sort the Filters.** After finishing the above steps, for each CONV layer, we can obtain a set of energy zone ratios $\eta_i = \{\eta_i^1, \eta_i^2, \cdots, \eta_i^T_i\}$. The set $I_i = \{I_i^1, I_i^2, \cdots, I_i^T_i\}$ records the channel index of the elements in $\eta_i$ after sorting them in a descending order. For example, if $\eta_i^1$ ranks first in $\eta_i$, then $I_i^1 = 12$. Figure 7 shows an example how we sort the filters according to the energy ratios.

**Complexity Analysis.** For a given feature map slice $Y_i^b[b, :, :] \in \mathbb{R}^{H_i \times W_i}$, we compare the complexity of HRank and EZCrop to generate the pruning metrics in spatial and frequency domains, respectively. HRank [23] employs the function torch.matrix_rank(·) in PyTorch, which uses SVD to compute the matrix rank at a complexity of $O(\max(H_i, W_i)^3)$. For EZCrop, applying 2D FFT to transform $Y_i^b[b, :, :]$ to the frequency domain is the most time-consuming step, which we implemented by np.fft.fft2(·) in NumPy with a complexity of $O(H_i W_i \log(H_i W_i))$. For a simpler expression, we rewrite it as $O(n^2 \log n)$ which indicates the high efficiency of EZCrop versus that of HRank.

5. Experimental Results

We conduct extensive experiments on image classification tasks, using CIFAR-10 [18] and ImageNet [4] datasets, to demonstrate the superiority of the proposed EZCrop channel pruning scheme. Popular CNN models are used, including VGGNet [40], ResNet [9], and DenseNet [14]. We present the performance of state-of-the-art algorithms for a comprehensive overview, and we mainly compare EZCrop against its closest scheme HRank which uses channel matrix rank in the spatial domain to measure channel importance.
The pruning results shown in Tables 2 to 5 are directly acquired from the corresponding papers or their official GitHub repositories. We obtained the result of multi-pass HRank in Table 6 ourselves by generalizing the single-pass HRank, since the authors of [23] did not report the results of applying HRank repetitively. For simplicity, we fix $\beta = 0.25$ in Eq. 4 in all experiments when generating the energy zone ratios, more training details can be found in Appendix 5. All experiments are run on a computer equipped with four NVIDIA GeForce GTX1080Ti Graphics Cards, each with 11GB frame buffer.

**Time Comparison.** In Table 1, we further show the actual runtimes of EZCrop and HRank to generate the required pruning metrics for those CIFAR-10/ImageNet networks on one GTX1080Ti GPU. Apparently, the time required by EZCrop to generate the pruning metric is much shorter in all models. Noticeably, the larger the dataset, the more pronounced the efficiency of EZCrop is.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>HRank [23]</th>
<th>EZCrop ($)</th>
<th>Params (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>VGGNet</td>
<td>1505.5s</td>
<td>356.94s (76.29%)</td>
<td>14.98M (0.0%)</td>
</tr>
<tr>
<td></td>
<td>ResNet-56</td>
<td>1247.51s</td>
<td>381.97s (69.38%)</td>
<td>5.40M (64.0%)</td>
</tr>
<tr>
<td></td>
<td>DenseNet-40</td>
<td>1431.17s</td>
<td>171.50s (63.76%)</td>
<td>3.93M (73.8%)</td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet-56</td>
<td>7.96h</td>
<td>3.45h (56.66%)</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 1. Runtimes of HRank and EZCrop to compute pruning metrics on CIFAR-10 / ImageNet for different models.

### 5.1. CIFAR-10

**VGGNet.** The VGG-16 used in the experiments is a variation of the original VGGNet for CIFAR-10 dataset taken from [20]. In Table 2, the upper part is the performance of state-of-the-art pruning approaches excluding HRank, while the lower part focuses on the comparison between HRank and EZCrop (such convention also applies to subsequent tables). First, we compare EZCrop with the upper-part approaches. Compared with FPGA, though both can reach the accuracy of around 94%, EZCrop has higher computational efficiency with more significant FLOPs reduction (58.1% vs. 35.9%). When compared with L1, SSS, Zhao et al., GAL, and PScratch, EZCrop excels in all aspects wherein it obtains the fastest and most compact model with the best performance. Furthermore, when compared with HRank under the same settings, namely, same FLOPs and Params reductions, EZCrop outperforms HRank in both cases (94.06% vs. 93.73% / 93.70% vs. 93.56%). This reveals that though HRank can evaluate the filters well using the information in the spatial domain, EZCrop offers more accurate evaluation benefiting from the additional frequency information.

**ResNet-56.** Table 3 shows the results of different pruning schemes on ResNet-56. When the FLOPs reductions are around 30%, EZCrop outperforms L1, LeGR, and HRank. Compared with L1 and HRank (first row in the lower part), EZCrop discards more parameters (34.1%) than L1 (14.1%) and HRank (22.3%), while reaches the highest accuracy (94.18%). As for LeGR, EZCrop squeezes out more accuracy (94.18%) than it (94.08%) under the same FLOPs reduction (30%). Compared with He et al., FPGA and PScratch, when their FLOPs reductions are both around 50%, EZCrop has the highest accuracy of 93.80%. When compared with GAL and NISP, EZCrop excels in all aspects. EZCrop also outperforms DHP, as the gap between their

---

Figure 7. By assuming the pictures as the outputs generated by four filters, we give an example to show intuitively how energy-zone ratios reflect the importance of the corresponding filters using the frequency information. (Left) The first row contains four feature map slices, the quality of which decreases from left to right. The second row shows the corresponding selected energy areas. It is worth noting that the concentration of the energy distribution increases as the feature maps become less informative. (Right) The more information the feature map contains, the larger the energy-zone ratio, and the more important the corresponding filter is.
When the number of parameters is around 10M, EZCrop achieves higher Top-1 accuracy in complicated datasets than CIFAR-10, as shown in Table 5.

5.2. ImageNet

EZCrop not only outperforms HRank on networks with plain reduction (59%) but also on networks with higher FLOPs reduction. When the FLOPs reduction is around 10%, EZCrop achieves higher Top-1 accuracy in both cases. This verifies that EZCrop beats HRank also on networks with inception modules.

5.2. ImageNet

ResNet-50. The results on ImageNet, which is a more complicated dataset than CIFAR-10, are shown in Table 5. When the number of parameters is around 15M, EZCrop (75.68%) has higher accuracy than SSS-26 (71.82%) and GAL-1 (69.31%). Though GAL-1 can reduce the FLOPs more than EZCrop (FLOPs: 1.58B vs. 2.26B), its accuracy drop of 6.37% is drastic. When the number of parameters is around 10M, EZCrop outperforms HRank and GAL-1-joint with obvious advantages on the performance (74.33% vs. 66.42% / 69.31%). Compared with MetaPruning, EZCrop has better performance (75.68% vs. 75.40%) with higher FLOPs reduction.

For DMCP and EagleEye, though they obtain more compact models with higher accuracy, they require dramatically more GPU resources. DMCP needs 8~15h for training to find the possible structure, using 16 32GB 1080Ti cards. EagleEye comprises three stages, namely, strategy proposal, candidate selection and fine-tuning. Excluding the fine-tuning stage, EagleEye takes more than 1 GPU day (an Nvidia 2080Ti card) to search for acceptable candidates with no guarantee that the best one can be selected. As depicted in Table 1, EZCrop costs only 3.45h (a single 11GB 1080Ti card) to obtain the pruning metric. In other words, EagleEye requires > 10× more GPU resources than EZCrop, which leads to a mere 0.7% accuracy improvement after fine-tuning. For other upper part approaches, EZCrop excels in all aspects. For the comparative experiments with HRank, EZCrop boosts the accuracy in all cases. This indicates that EZCrop can deal with large and complicated datasets as well.

Table 3. Pruning results of ResNet-50 on CIFAR-10.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1%</th>
<th>FLOPs (%)</th>
<th>Params (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRank [23]</td>
<td>93.85</td>
<td>90.35 (28%)</td>
<td>0.66M (22.3%)</td>
</tr>
<tr>
<td>EZCrop</td>
<td>94.18</td>
<td>87.84 (30.0%)</td>
<td>0.56M (34.1%)</td>
</tr>
<tr>
<td>HRank [23]</td>
<td>93.57</td>
<td>65.94 (47.4%)</td>
<td>0.48M (42.8%)</td>
</tr>
<tr>
<td>EZCrop</td>
<td>93.80</td>
<td>65.94 (47.4%)</td>
<td>0.48M (42.8%)</td>
</tr>
<tr>
<td>HRank [23]</td>
<td>92.32</td>
<td>34.78 (74.1%)</td>
<td>0.24M (70.0%)</td>
</tr>
<tr>
<td>EZCrop</td>
<td>92.52</td>
<td>34.78 (74.1%)</td>
<td>0.24M (70.0%)</td>
</tr>
</tbody>
</table>

Table 4. Pruning results of DenseNet-40 on CIFAR-10.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1%</th>
<th>FLOPs (%)</th>
<th>Params (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseNet-40</td>
<td>94.82</td>
<td>282.00 (0.0%)</td>
<td>1.04M (0.0%)</td>
</tr>
<tr>
<td>Liu et al.-40 [29]</td>
<td>94.81</td>
<td>190.00 (32.8%)</td>
<td>0.66M (36.5%)</td>
</tr>
<tr>
<td>GAL-0.01 [27]</td>
<td>94.29</td>
<td>182.92 (35.3%)</td>
<td>0.67M (35.6%)</td>
</tr>
<tr>
<td>GAL-0.05 [27]</td>
<td>91.53</td>
<td>128.11 (54.7%)</td>
<td>0.45M (56.7%)</td>
</tr>
<tr>
<td>Zhao et al. [46]</td>
<td>93.16</td>
<td>156.00 (44.8%)</td>
<td>0.42M (57.9%)</td>
</tr>
<tr>
<td>HRank [23]</td>
<td>94.51</td>
<td>173.39 (38.5%)</td>
<td>0.62M (40.1%)</td>
</tr>
<tr>
<td>EZCrop</td>
<td>94.72</td>
<td>173.39 (38.5%)</td>
<td>0.62M (40.1%)</td>
</tr>
<tr>
<td>HRank [23]</td>
<td>93.76</td>
<td>113.08 (59.9%)</td>
<td>0.39M (61.9%)</td>
</tr>
<tr>
<td>EZCrop</td>
<td>93.76</td>
<td>113.08 (59.9%)</td>
<td>0.39M (61.9%)</td>
</tr>
</tbody>
</table>

Table 5. Pruning Results of ResNet-50 on ImageNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1%</th>
<th>Top-5%</th>
<th>FLOPs</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [32]</td>
<td>76.15</td>
<td>92.87</td>
<td>4.09B</td>
<td>25.50M</td>
</tr>
<tr>
<td>He et al. [11]</td>
<td>72.30</td>
<td>90.80</td>
<td>2.73B</td>
<td>12M</td>
</tr>
<tr>
<td>ThyNet-50 [32]</td>
<td>68.42</td>
<td>88.30</td>
<td>1.10B</td>
<td>8.66M</td>
</tr>
<tr>
<td>SSS-26 [15]</td>
<td>71.82</td>
<td>90.79</td>
<td>2.33B</td>
<td>15.60M</td>
</tr>
<tr>
<td>SSS-32 [15]</td>
<td>71.18</td>
<td>91.91</td>
<td>2.82B</td>
<td>18.60M</td>
</tr>
<tr>
<td>GDP-0.5 [26]</td>
<td>69.58</td>
<td>90.14</td>
<td>1.57B</td>
<td>18.60M</td>
</tr>
<tr>
<td>GDP-0.5 [26]</td>
<td>71.19</td>
<td>90.71</td>
<td>1.88B</td>
<td>21.20M</td>
</tr>
<tr>
<td>GAL-0.5 [27]</td>
<td>71.95</td>
<td>90.94</td>
<td>2.33B</td>
<td>18.60M</td>
</tr>
<tr>
<td>GAL-0.5-joint [27]</td>
<td>71.80</td>
<td>90.82</td>
<td>1.84B</td>
<td>19.31M</td>
</tr>
<tr>
<td>GAL-1-joint [27]</td>
<td>69.31</td>
<td>89.12</td>
<td>1.11B</td>
<td>20.12M</td>
</tr>
<tr>
<td>FPGM [10]</td>
<td>75.91</td>
<td>92.63</td>
<td>2.36B</td>
<td>15.60M</td>
</tr>
<tr>
<td>MetaPruning [30]</td>
<td>75.40</td>
<td>92.63</td>
<td>2.36B</td>
<td>15.60M</td>
</tr>
<tr>
<td>DMCP [6]</td>
<td>76.20</td>
<td>92.60</td>
<td>2.29B</td>
<td>15.60M</td>
</tr>
<tr>
<td>EagleEye [19]</td>
<td>76.40</td>
<td>92.60</td>
<td>2.29B</td>
<td>15.60M</td>
</tr>
<tr>
<td>ABCPruner-0.8% [21]</td>
<td>73.86</td>
<td>91.69</td>
<td>1.98B</td>
<td>11.75M</td>
</tr>
</tbody>
</table>

Table 5. Pruning Results of ResNet-50 on ImageNet.

5.3. Robustness Under Repetitive Pruning

We further test HRank and EZCrop in a multi-pass manner, i.e., pruning models repetitively. Repetitive pruning requires multiple evaluations of channels’ importance, which amplifies the difference between the two schemes. We conduct the experiments using ResNet-56 on CIFAR-10, and keep adopting HRank and EZCrop to get a more compact model without a fixed compression rate, then compare their
performance. For fairness, all settings of HRank and EZCrop for every single-pass are the same. Table 6 shows that when getting more compact models, the gap of EZCrop over HRank is enlarged after each single-pass. When we reduce the #Params from 0.85M to 0.63M, the gap between the accuracy of HRank (93.76%) and EZCrop (93.95%) is 0.19%. After the second pass, the gap is enlarged to 0.27% with 0.17M more parameters discarded. In the final pass, the gap is further expanded to 0.6% while more than 50% parameters are reduced. This obviously shows that EZCrop enjoys higher robustness under repetitive pruning, and is able to evaluate the filters more accurately in each single-pass.

<table>
<thead>
<tr>
<th>#Passes (#epochs)</th>
<th>FLOPs</th>
<th>Params</th>
<th>HRank [23]</th>
<th>EZCrop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (300)</td>
<td>90.86M</td>
<td>0.63M</td>
<td>93.76%</td>
<td>93.95%</td>
</tr>
<tr>
<td>2 (300)</td>
<td>66.25M</td>
<td>0.46M</td>
<td>93.15%</td>
<td>93.42%</td>
</tr>
<tr>
<td>3 (300)</td>
<td>36.03M</td>
<td>0.22M</td>
<td>91.58%</td>
<td>92.18%</td>
</tr>
</tbody>
</table>

Table 6. Repetitive pruning of ResNet-56 on CIFAR-10.

5.4. Standard Deviation (STD) Analysis

To further show the effectiveness and robustness of EZCrop while dispelling the worries that the improvements compared with HRank are due to standard variations, we conduct the first comparative experiments in Tables 2-4 for 20 times and compute the average and STD of the accuracy, respectively. The results in Table 7 prove that the improvements are not accidental, and EZCrop actually makes more accurate and robust evaluations of the importance of filters.

<table>
<thead>
<tr>
<th>Mean Acc. (%) / STD</th>
<th>VGGNet</th>
<th>ResNet-56</th>
<th>DenseNet-40</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZCrop</td>
<td>93.98% / 0.073</td>
<td>93.99% / 0.097</td>
<td>94.63% / 0.066</td>
</tr>
<tr>
<td>HRank [23]</td>
<td>93.75% / 0.140</td>
<td>93.76% / 0.168</td>
<td>94.26% / 0.165</td>
</tr>
</tbody>
</table>

Table 7. The mean and STD of the accuracy by repeating each selected experimental setting for 20 times.

5.5. Ablation Study

Excluding the pruning metrics, the compression rate and training settings for VGGNet / ResNet-56 / DenseNet-40 in this section are the same as the first comparative experiments in Tables 2-4, respectively.

5.5.1 Effectiveness of Energy-zone Rate

To show that EZCrop makes reliable filter importance evaluation, we compare the results by pruning the models: 1) using EZCrop, 2) randomly, 3) using inverse EZCrop (filters with high energy-zone ratio will be removed). For each network in Figure 8, we observe that EZCrop outperforms random pruning, while the inverse EZCrop has the worst performance. The gaps in the accuracy confirm the effectiveness of our energy-zone metric in the frequency domain.

5.5.2 Appropriate Expanding Distance

There is a hyper-parameter $\beta$ in EZCrop when we decide the expanding distance. In this paper, we set $\beta = 0.25$ for all experiments for the sake of simplicity. However, this general setting may not be an optimal setting for all. In Figure 9, we explore the relations between the accuracy and $\beta$. It is worth noting that the three curves have similar trends, the curve shows an “M” shape when $\beta$ is between 0.2 and 0.6. Based on this observation, we suggest to manually set $\beta$ around 0.3 or 0.5, namely, two peaks of the curve.

6. Conclusion

This work has connected the previously mysterious constant-rank phenomenon in a CNN feature map channel to a novel, analytical view in the frequency domain. Via a spectral perspective, an efficient FFT-based energy-zone metric has been proposed for evaluating the importance of a channel and its corresponding filter. This leads to a channel pruning scheme named EZCrop (energy-zoned channels for robust output pruning) which readily outperforms existing state-of-the-art in terms of compression rates, runtimes and FLOPs. EZCrop also constitutes a robust way for repetitive channel pruning to stretch the limit of CNN compression, while maintaining high output accuracy.

Acknowledgement

This work is supported in part by the General Research Fund (GRF) projects 17209721 & 17206020, and in part by the in-kind support of United Microelectronics Centre (Hong Kong) Limited.
References


