CVPR SEATTLE WASHINGTON
JUNE 16-18 2020

HRank: Filter Pruning using High-Rank Feature Map

Mingbao Lin[1], Rongrong Ji[1,5*], Yan Wang[2], Yichen Zhang[1],
Baochang Zhang[3], Yonghong Tian[4,5], Ling Shao[6]
[1]Media Analytics and Computing Laboratory, Department of Artificial Intelligence, School of Informatics, Xiamen University, China, [2]Pinterest, USA, [3]Beihang University, China
[4]Peking University, Beijing, China, [5]Peng Cheng Laboratory, Shenzhen, China
[6]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
lmbxmu@stu.xmu.edu.cn, rrji@xmu.edu.cn, yanw@pinterest.com, ethan.zhangyc@gmail.com,
bczhang@buaa.edu.cn, yhtian@pku.edu.cn, ling.shao@ieee.org

Outline

- Introduction
  - Motivation
  - Contributions
- Proposed Method
  - Framework
  - Theoretical Explanation
  - Tractability of Optimization
  - Pruning Procedure
- Experiments
  - Experimental Settings
  - Results on CIFAR-10
  - Results on ImageNet
  - Ablation Study(Variants of HRank)
  - Ablation Study(Freezing Filters during Fine-tuning)

Introduction

Motivation

- Filter pruning remains an open problem. On one hand we pursuit higher compression/acceleration ratios, while on the other hand we are restricted by heavy machine time and human labor.

- We attribute these problems to the lack of practical/theoretical guidance regarding to the filter importance and redundancy

Introduction

Motivation

- The proposed HRank performs as such a guidance, which is a property importance based filter pruner. It eliminates the need of introducing additional auxiliary constraints or retraining the model, thus simplifying the pruning complexity.

- With the empirical and quantitative observation, we have found that the average rank of feature maps generated by a single filter is always the same, regardless of how much data the CNN has seen. It suggests that the ranks of feature maps in deep CNNs can be accurately estimated using only a small portion of the input images, and thus can be highly efficient.

Introduction

Contributions

- Demonstrate empirically that the average rank of feature maps generated by a single filter is almost unchanged.

- Prove mathematically that filters with lower-rank feature maps are less informative and thus less important to preserve accuracy, which can be removed first.

- Demonstrate the efficiency and effectiveness of HRank in both model compression and acceleration over a variety of state-of-the-arts.

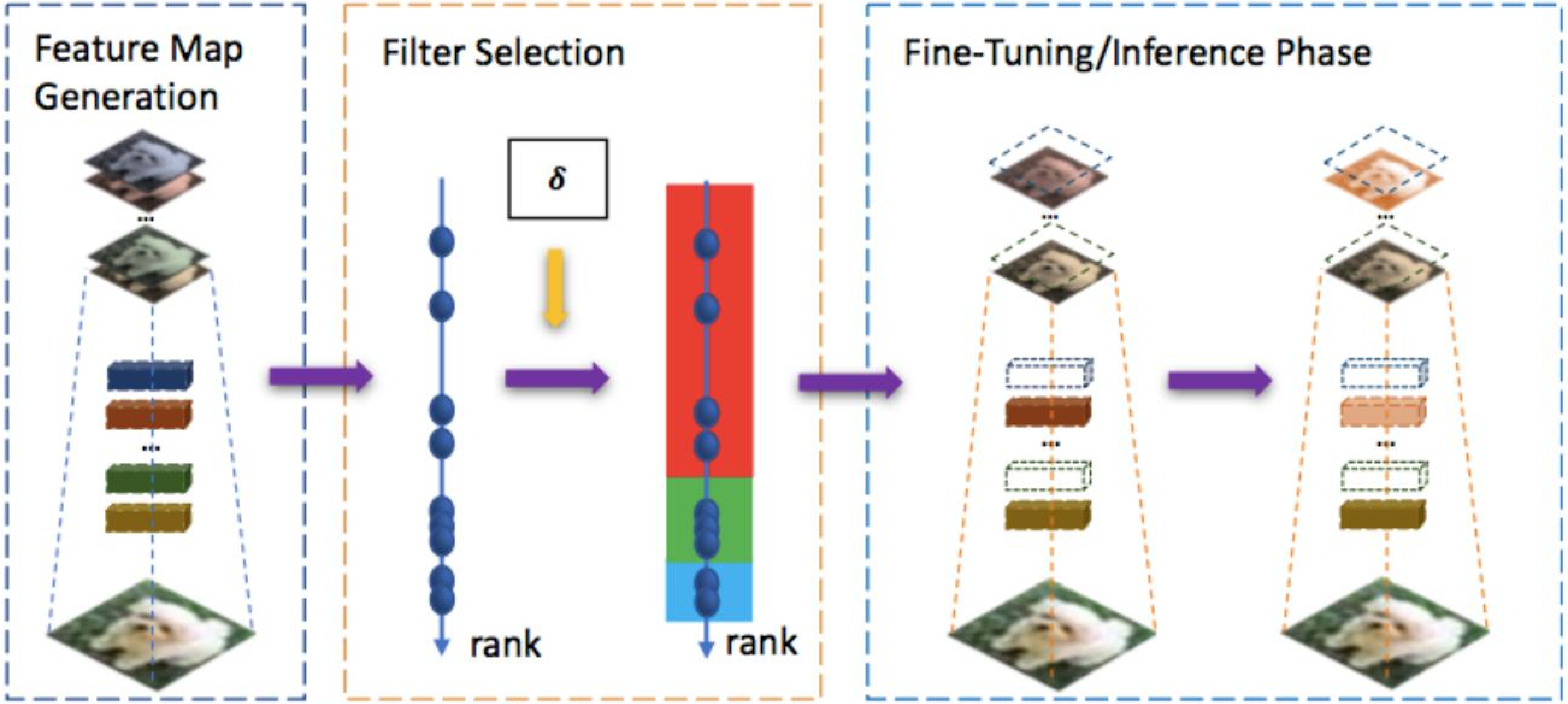
Outline

- Introduction
  - Motivation
  - Contributions
- Proposed Method
  - Framework
  - Theoretical Explanation
  - Tractability of Optimization
  - Pruning Procedure
- Experiments
  - Experimental Settings
  - Results on CIFAR-10
  - Results on ImageNet
  - Ablation Study(Variants of HRank)
  - Ablation Study(Freezing Filters during Fine-tuning)

Proposed Method

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

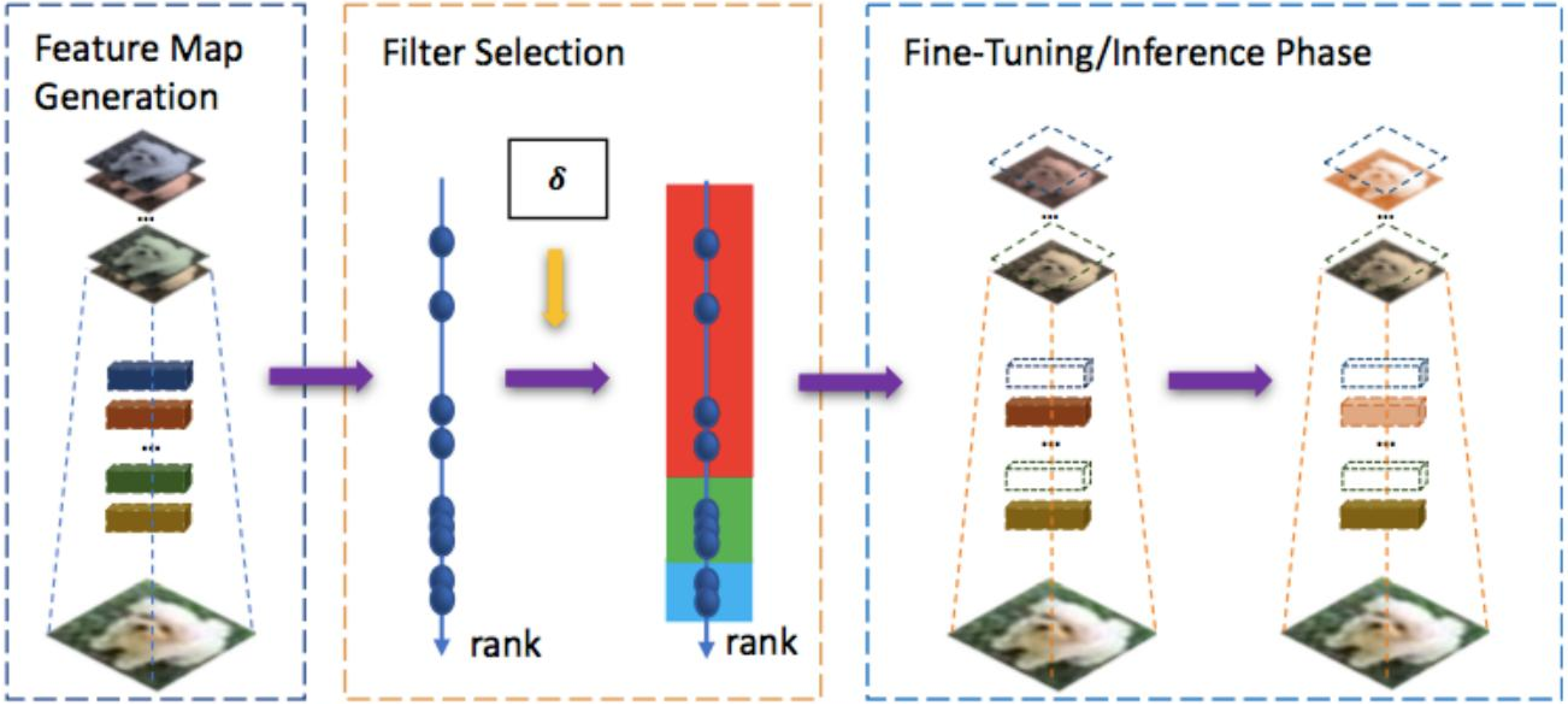
Framework



We first use images to run through the convolutional layers to get the feature maps.

Proposed Method

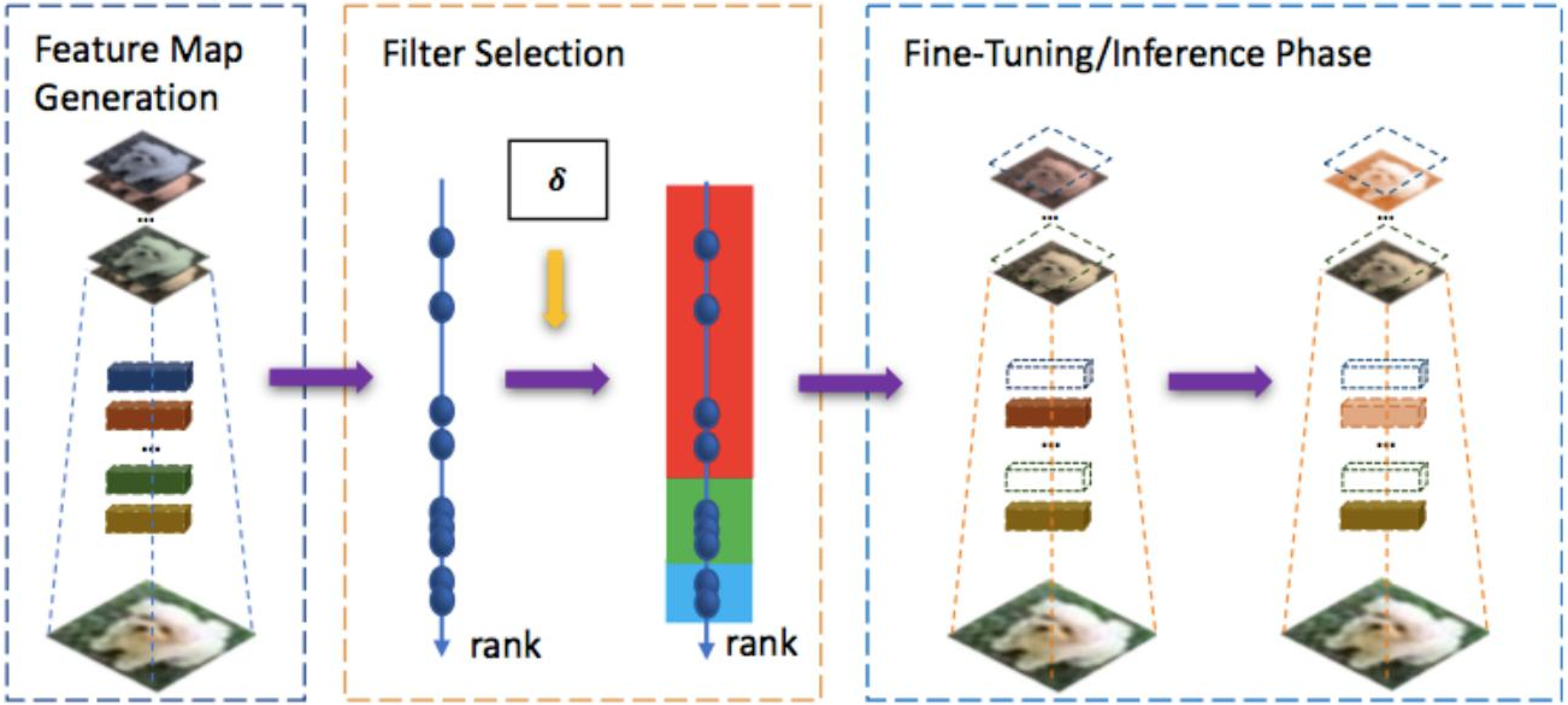
Framework



Then we estimate the rank of each feature map, which is used as the criteria for pruning.

Proposed Method

Framework



Last, we prune those red filters, and fine-tune where the green filters are updated and the blue filters are frozen.

Proposed Method

Theoretical Explanation

We exploit the rank of feature maps which is demonstrated to be not only an effective measure of information, but also a stable representation across the distribution of input images.

$$\hat{L}\big(\mathbf{o}_j^i(I,:,:)\big) = \mathbf{Rank}\big(\mathbf{o}_j^i(I,:,:)\big)$$

We conduct a Singular Value Decomposition (SVD) for $o_j^i(I,:,:)$:

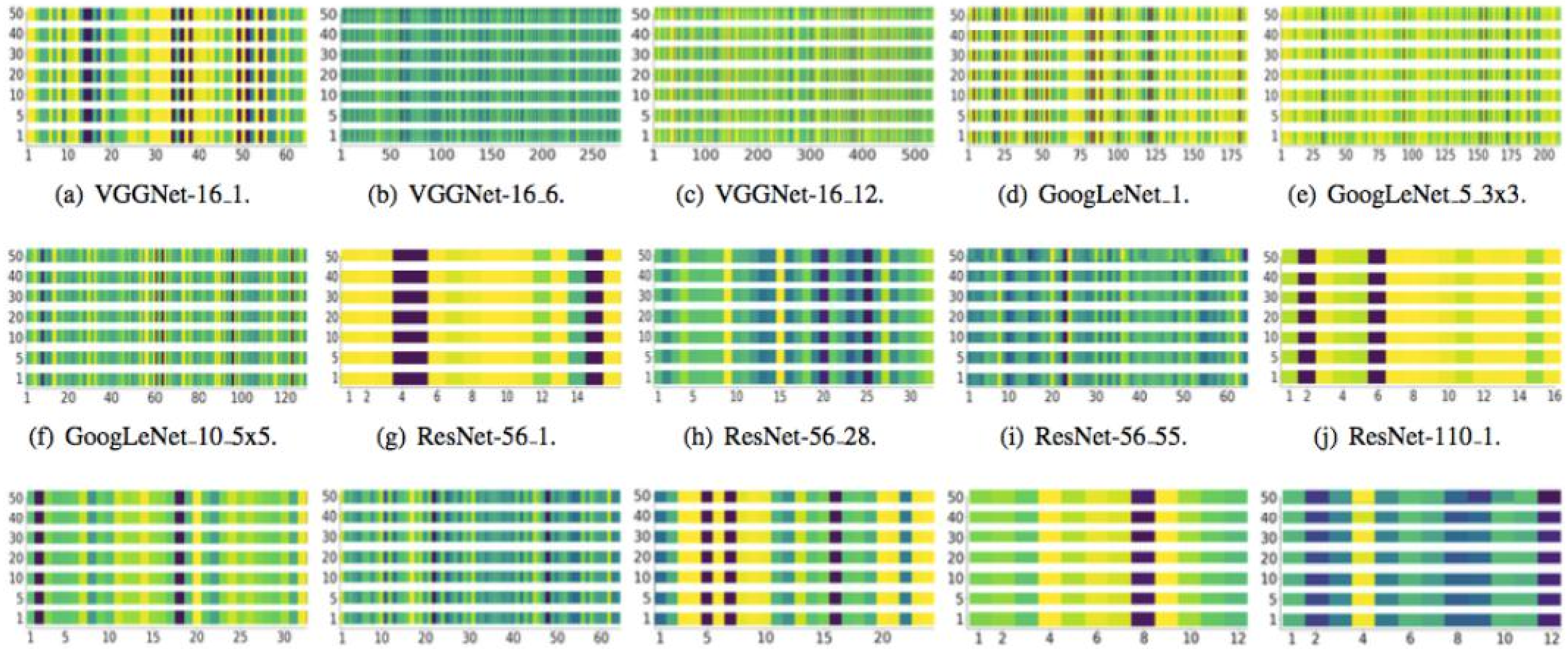
$$\begin{aligned}\mathbf{o}_j^i(I,:,:) &= \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \\ &= \sum_{i=1}^{r'} \sigma_i \mathbf{u}_i \mathbf{v}_i^T + \sum_{i=r'+1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T\end{aligned}$$

It can be seen that a feature map with rank r can be decomposed into a lower-rank feature map with rank r', $i.e.$, $\sum_{i=1}^{r'} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$, and some additional information, $i.e.$, $\sum_{i=r'+1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$. Hence, higher-rank feature maps actually contain more information than lower-rank ones.

Proposed Method

Tractability of Optimization

We empirically observe that the expectation of ranks generated by a single filter is robust to the input images, which means the variance is negligible. Hence, a small batch of input images can be used to accurately estimate the expectation of the feature map rank.



(a) VGGNet-16_1. (b) VGGNet-16_6. (c) VGGNet-16_12. (d) GoogLeNet_1. (e) GoogLeNet_5_3x3.

(f) GoogLeNet_10_5x5. (g) ResNet-56_1. (h) ResNet-56_28. (i) ResNet-56_55. (j) ResNet-110_1.

(k) ResNet-110_54. (l) ResNet-110_108. (m) DenseNet-40_1. (n) DenseNet-40_20. (o) DenseNet-40_39.

Figure 2. Average rank statistics of feature maps from different convolutional layers and architectures on CIFAR-10. For each subfigure, the x-axis represents the indices of feature maps and the y-axis is the batches of training images (each batch size is set to 128). Different colors denote different rank values. As can be seen, the rank of each feature map (column of the subfigure) is almost unchanged (the same color), regardless of the image batches. Hence, even a small number of images can effectively estimate the average rank of each feature map in different architectures.

Proposed Method

Pruning Procedure

Require: filters W_{C^i} in C^i and their generated feature maps $\mathcal{O}^i$

- Calculate the average rank of feature map o_j^i in $\mathcal{O}^i$, forming a rank set $\mathcal{R}^i = \{r_1^i, r_2^i, \dots, r_{n_i}^i\} \in \mathcal{R}^{n_i}$.
- Re-rank the rank set in decreasing order $\widehat{\mathcal{R}}^i = \{r^i_{I^i_1}, r^i_{I^i_2}, \dots, r^i_{I^i_{n_i}}\} \in \mathcal{R}^{n_i}$, where I_j^i is the index of the j-th top value in $\mathcal{R}^i$.
- Determine the values of n_{i1} (number of preserved filters) and n_{i2} (number of pruned filters).
- Obtain the important filter set $I_{C^i} = \{\boldsymbol{w}^i_{I^i_1}, \boldsymbol{w}^i_{I^i_2}, \dots, \boldsymbol{w}^i_{I^i_{n_{i1}}}\}$ where the rank of $\boldsymbol{w}^i_{I^i_j}$ is $r^i_{I^i_j}$. Similarly, we obtain the filter set $U_{C^i} = \{\boldsymbol{w}^i_{U^i_1}, \boldsymbol{w}^i_{U^i_2}, \dots, \boldsymbol{w}^i_{U^i_{n_{i2}}}\}$
- Remove set U_{C^i} and fine-tune the network with I_{C^i} as the initialization.

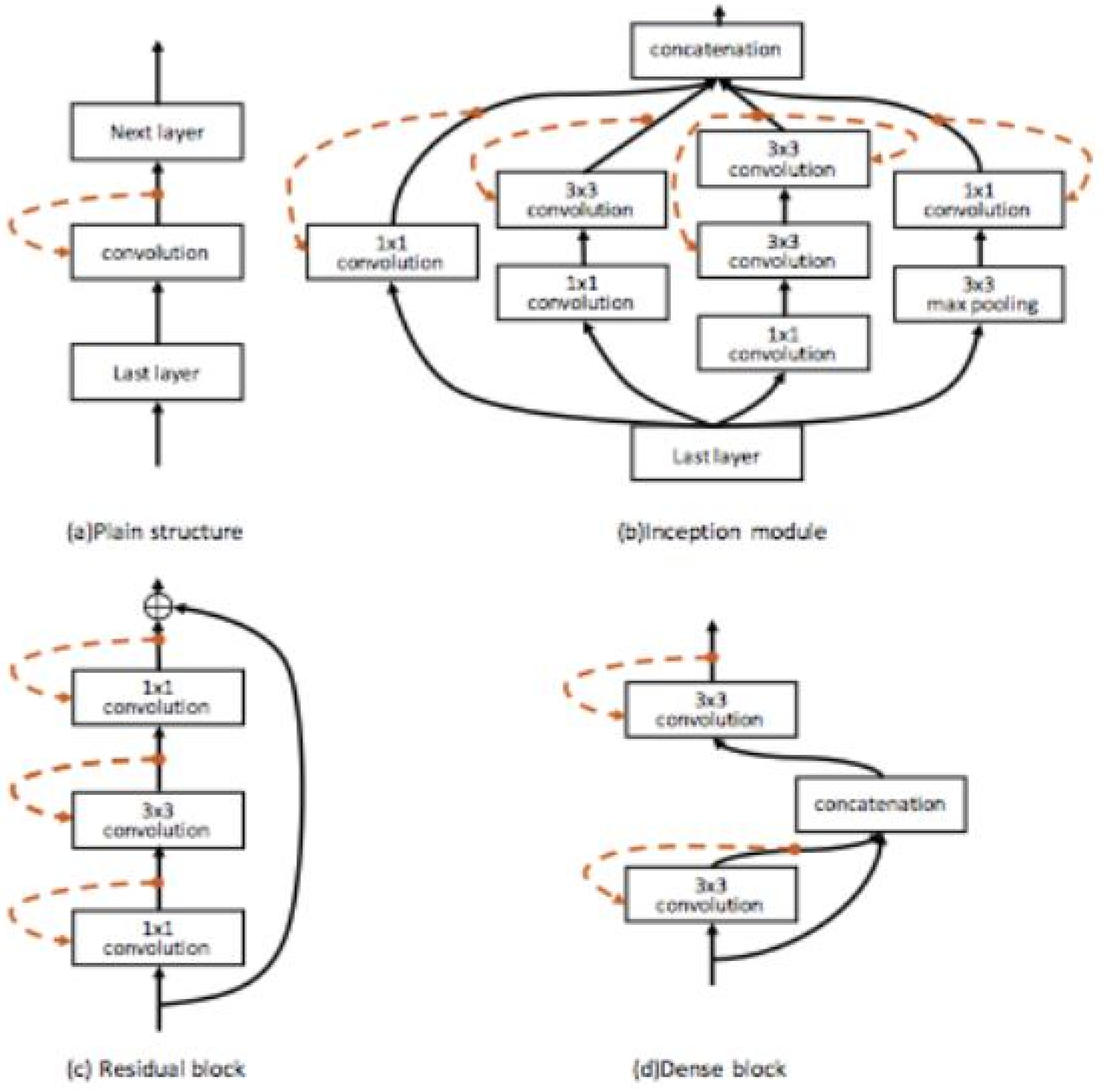
Outline

- Introduction
  - Motivation
  - Contributions
- Proposed Method
  - Framework
  - Theoretical Explanation
  - Tractability of Optimization
  - Pruning Procedure
- Experiments
  - Experimental Settings
  - Results on CIFAR-10
  - Results on ImageNet
  - Ablation Study(Variants of HRank)
  - Ablation Study(Freezing Filters during Fine-tuning)

Experiments

Experimental Settings

- Datasets: CIFAR-10, ImageNet
- Baselines: VGGNet, GoogLeNet, ResNet, DenseNet
- For all benchmarks and architectures, we randomly sample 500 images to estimate the average rank of each feature map.



(a)Plain structure

(b)Inception module

(c) Residual block

(d)Dense block

Figure 3. An illustration of mainstream network structures to be pruned, including Plain structure [32], Inception module [33], Residual block [11] and Dense block [15]. The black lines denote the inference streams of CNNs. The red dots denote the outputs of a convolutional layer (*i.e.*, feature maps). The red dashed lines denote the to-be-pruned layers after observing the ranks of feature maps. Note that, for 1×1 convolutions followed by $n \times n$ convolution ($n = 3$ in (b)), we do not consider pruning the 1×1 filter since it contains fewer parameters and less computation compared with an $n \times n$ filter.

Experiments

Results on CIFAR-10

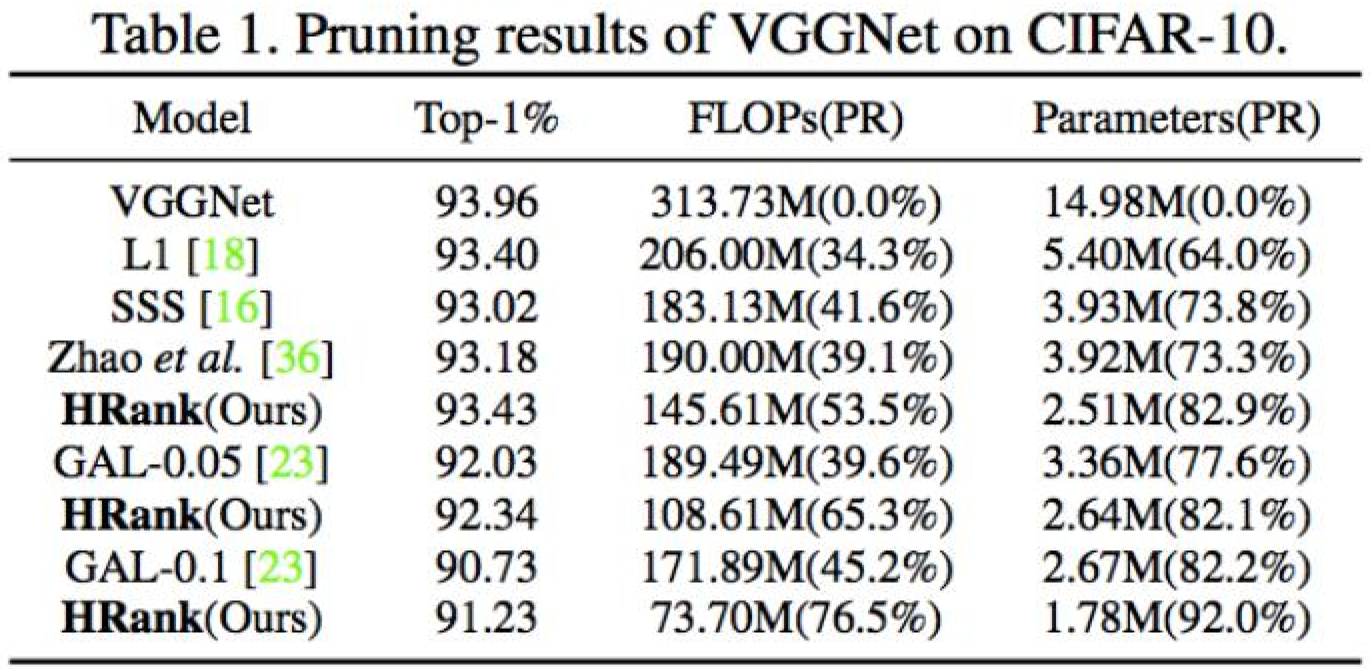


Table 1. Pruning results of VGGNet on CIFAR-10.

Model	Top-1%	FLOPs(PR)	Parameters(PR)
VGGNet	93.96	313.73M(0.0%)	14.98M(0.0%)
L1 [18]	93.40	206.00M(34.3%)	5.40M(64.0%)
SSS [16]	93.02	183.13M(41.6%)	3.93M(73.8%)
Zhao *et al.* [36]	93.18	190.00M(39.1%)	3.92M(73.3%)
HRank(Ours)	93.43	145.61M(53.5%)	2.51M(82.9%)
GAL-0.05 [23]	92.03	189.49M(39.6%)	3.36M(77.6%)
HRank(Ours)	92.34	108.61M(65.3%)	2.64M(82.1%)
GAL-0.1 [23]	90.73	171.89M(45.2%)	2.67M(82.2%)
HRank(Ours)	91.23	73.70M(76.5%)	1.78M(92.0%)

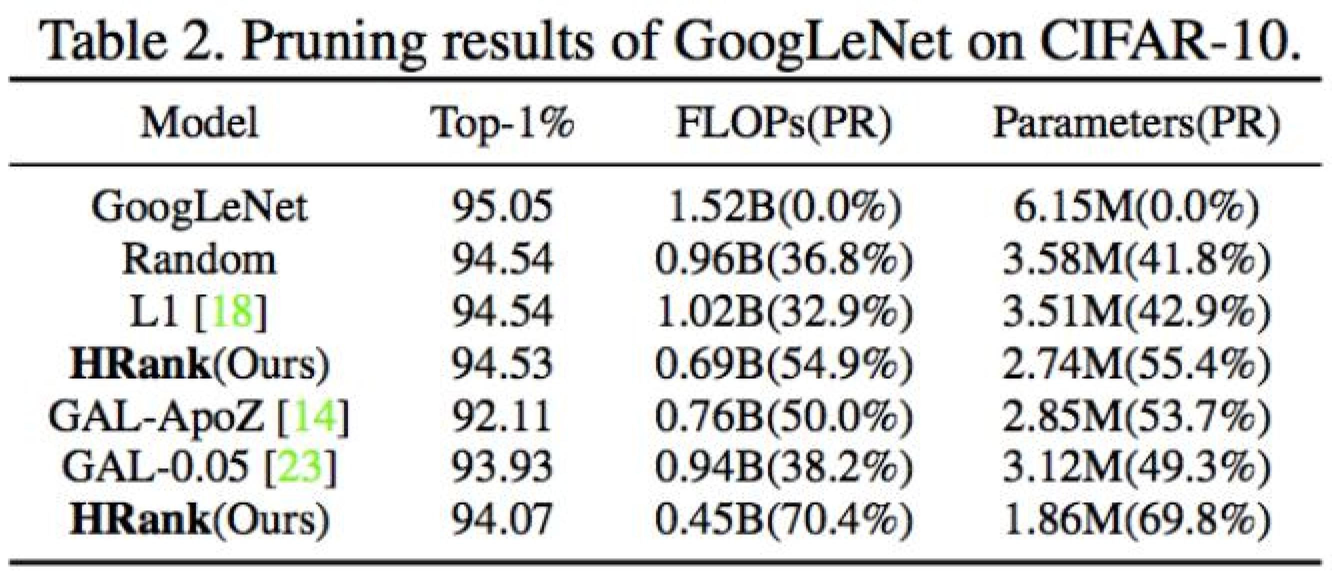


Table 2. Pruning results of GoogLeNet on CIFAR-10.

Model	Top-1%	FLOPs(PR)	Parameters(PR)
GoogLeNet	95.05	1.52B(0.0%)	6.15M(0.0%)
Random	94.54	0.96B(36.8%)	3.58M(41.8%)
L1 [18]	94.54	1.02B(32.9%)	3.51M(42.9%)
HRank(Ours)	94.53	0.69B(54.9%)	2.74M(55.4%)
GAL-ApoZ [14]	92.11	0.76B(50.0%)	2.85M(53.7%)
GAL-0.05 [23]	93.93	0.94B(38.2%)	3.12M(49.3%)
HRank(Ours)	94.07	0.45B(70.4%)	1.86M(69.8%)

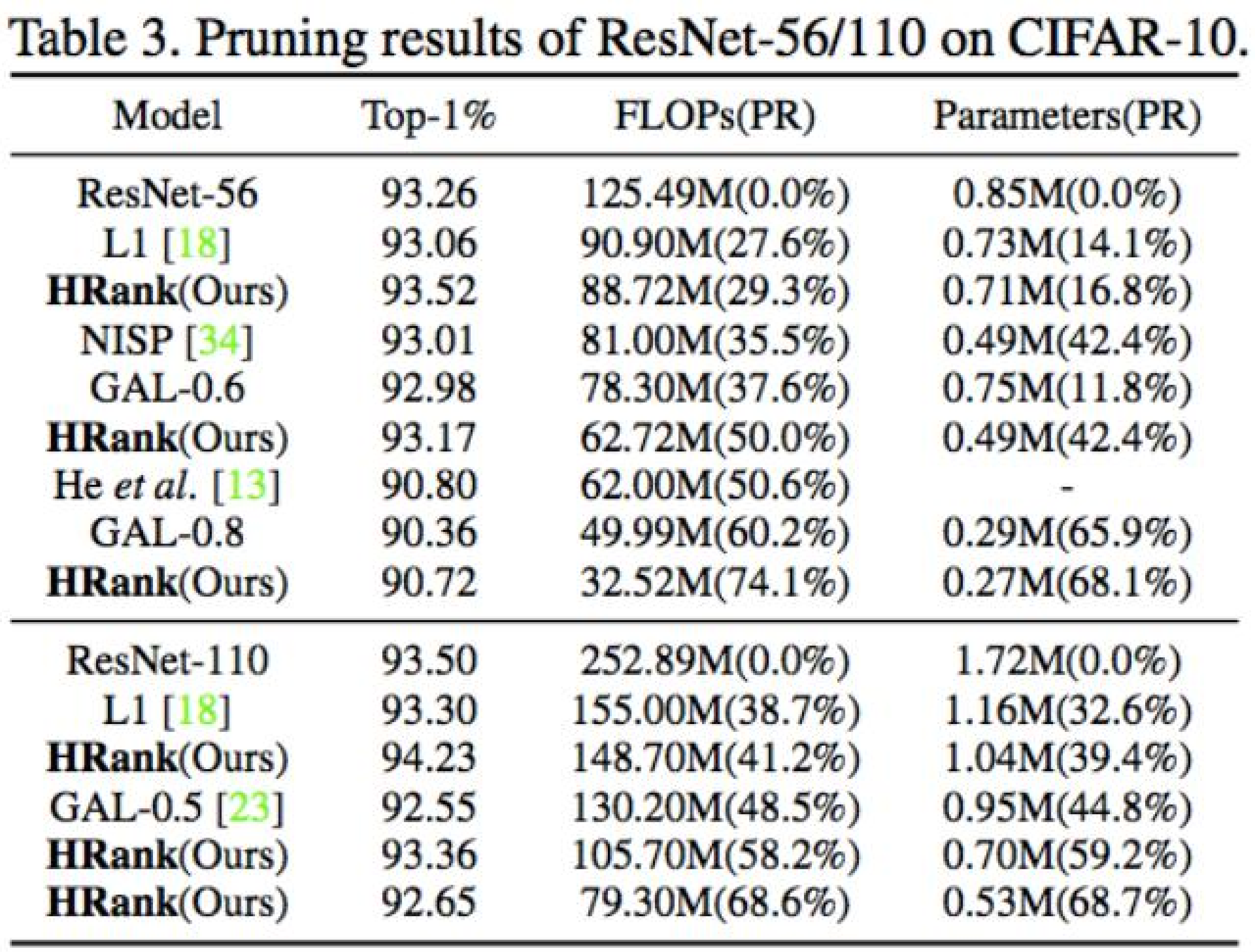


Table 3. Pruning results of ResNet-56/110 on CIFAR-10.

Model	Top-1%	FLOPs(PR)	Parameters(PR)
ResNet-56	93.26	125.49M(0.0%)	0.85M(0.0%)
L1 [18]	93.06	90.90M(27.6%)	0.73M(14.1%)
HRank(Ours)	93.52	88.72M(29.3%)	0.71M(16.8%)
NISP [34]	93.01	81.00M(35.5%)	0.49M(42.4%)
GAL-0.6	92.98	78.30M(37.6%)	0.75M(11.8%)
HRank(Ours)	93.17	62.72M(50.0%)	0.49M(42.4%)
He *et al.* [13]	90.80	62.00M(50.6%)	-
GAL-0.8	90.36	49.99M(60.2%)	0.29M(65.9%)
HRank(Ours)	90.72	32.52M(74.1%)	0.27M(68.1%)
ResNet-110	93.50	252.89M(0.0%)	1.72M(0.0%)
L1 [18]	93.30	155.00M(38.7%)	1.16M(32.6%)
HRank(Ours)	94.23	148.70M(41.2%)	1.04M(39.4%)
GAL-0.5 [23]	92.55	130.20M(48.5%)	0.95M(44.8%)
HRank(Ours)	93.36	105.70M(58.2%)	0.70M(59.2%)
HRank(Ours)	92.65	79.30M(68.6%)	0.53M(68.7%)

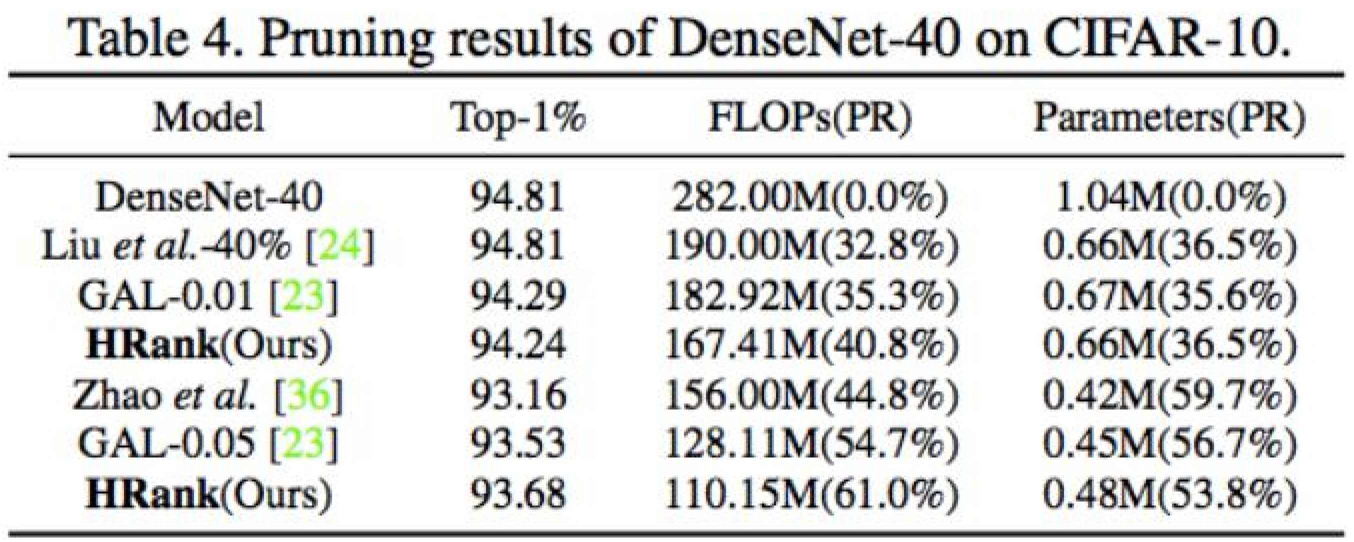


Table 4. Pruning results of DenseNet-40 on CIFAR-10.

Model	Top-1%	FLOPs(PR)	Parameters(PR)
DenseNet-40	94.81	282.00M(0.0%)	1.04M(0.0%)
Liu *et al.*-40% [24]	94.81	190.00M(32.8%)	0.66M(36.5%)
GAL-0.01 [23]	94.29	182.92M(35.3%)	0.67M(35.6%)
HRank(Ours)	94.24	167.41M(40.8%)	0.66M(36.5%)
Zhao *et al.* [36]	93.16	156.00M(44.8%)	0.42M(59.7%)
GAL-0.05 [23]	93.53	128.11M(54.7%)	0.45M(56.7%)
HRank(Ours)	93.68	110.15M(61.0%)	0.48M(53.8%)

Experiments

Results on ImageNet

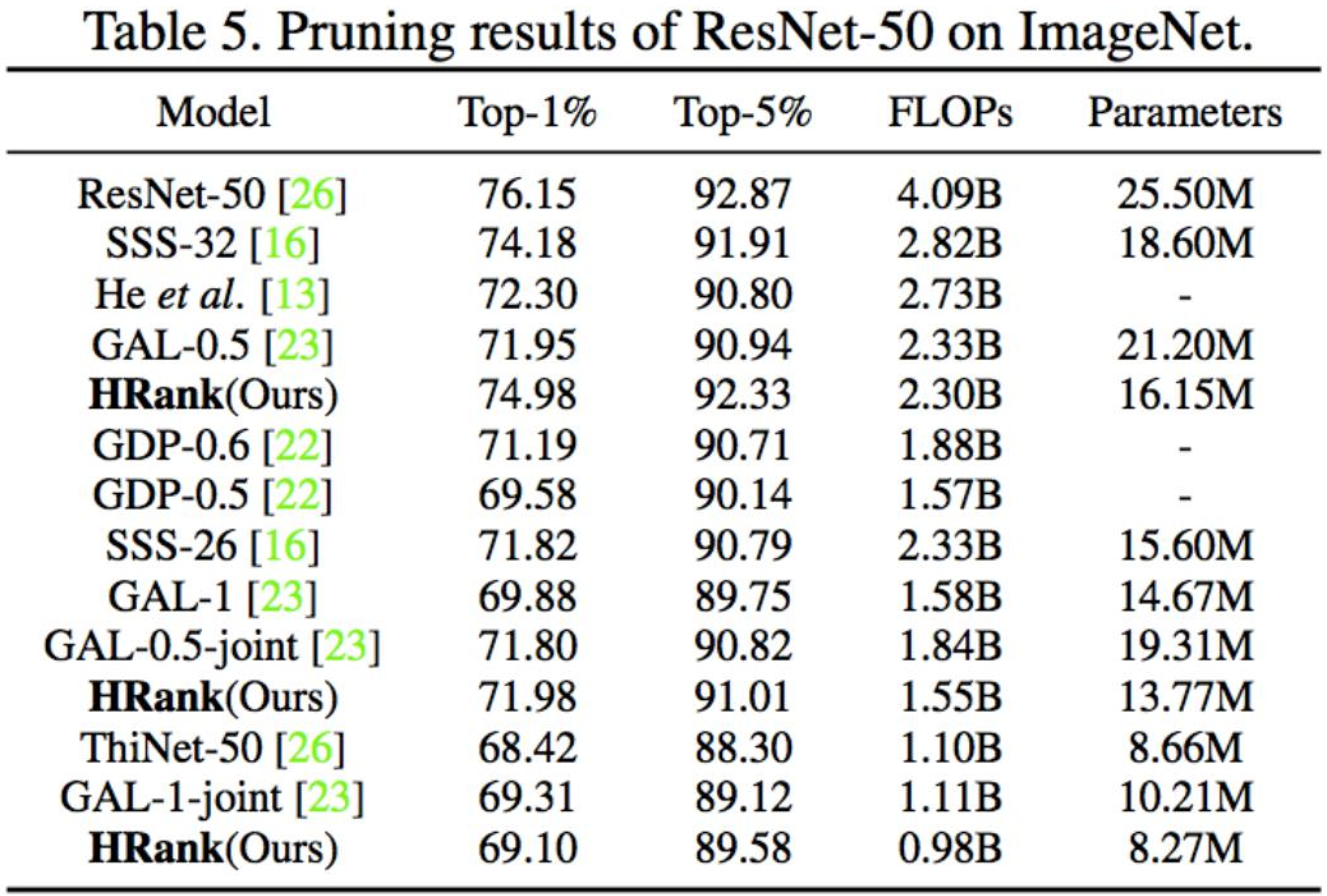


Table 5. Pruning results of ResNet-50 on ImageNet.

Model	Top-1%	Top-5%	FLOPs	Parameters
ResNet-50 [26]	76.15	92.87	4.09B	25.50M
SSS-32 [16]	74.18	91.91	2.82B	18.60M
He *et al.* [13]	72.30	90.80	2.73B	-
GAL-0.5 [23]	71.95	90.94	2.33B	21.20M
HRank(Ours)	74.98	92.33	2.30B	16.15M
GDP-0.6 [22]	71.19	90.71	1.88B	-
GDP-0.5 [22]	69.58	90.14	1.57B	-
SSS-26 [16]	71.82	90.79	2.33B	15.60M
GAL-1 [23]	69.88	89.75	1.58B	14.67M
GAL-0.5-joint [23]	71.80	90.82	1.84B	19.31M
HRank(Ours)	71.98	91.01	1.55B	13.77M
ThiNet-50 [26]	68.42	88.30	1.10B	8.66M
GAL-1-joint [23]	69.31	89.12	1.11B	10.21M
HRank(Ours)	69.10	89.58	0.98B	8.27M

Experiments

Ablation Study(Variants of HRank)

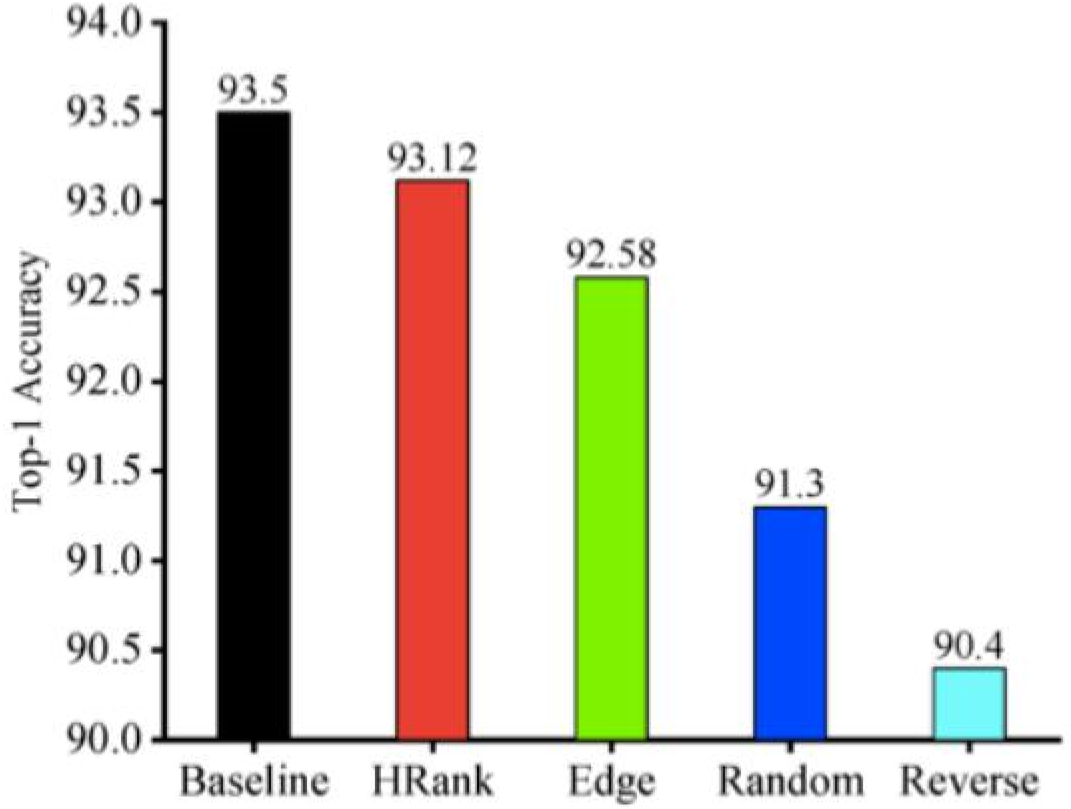


Figure 4. Top-1 accuracy for variants of HRank.

Three variants are proposed to demonstrate the appropriateness of preserving filters with high-rank feature maps, including:

(1) Edge: Filters generating both low- and high-rank feature maps are pruned.

(2) Random: Filters are randomly pruned.

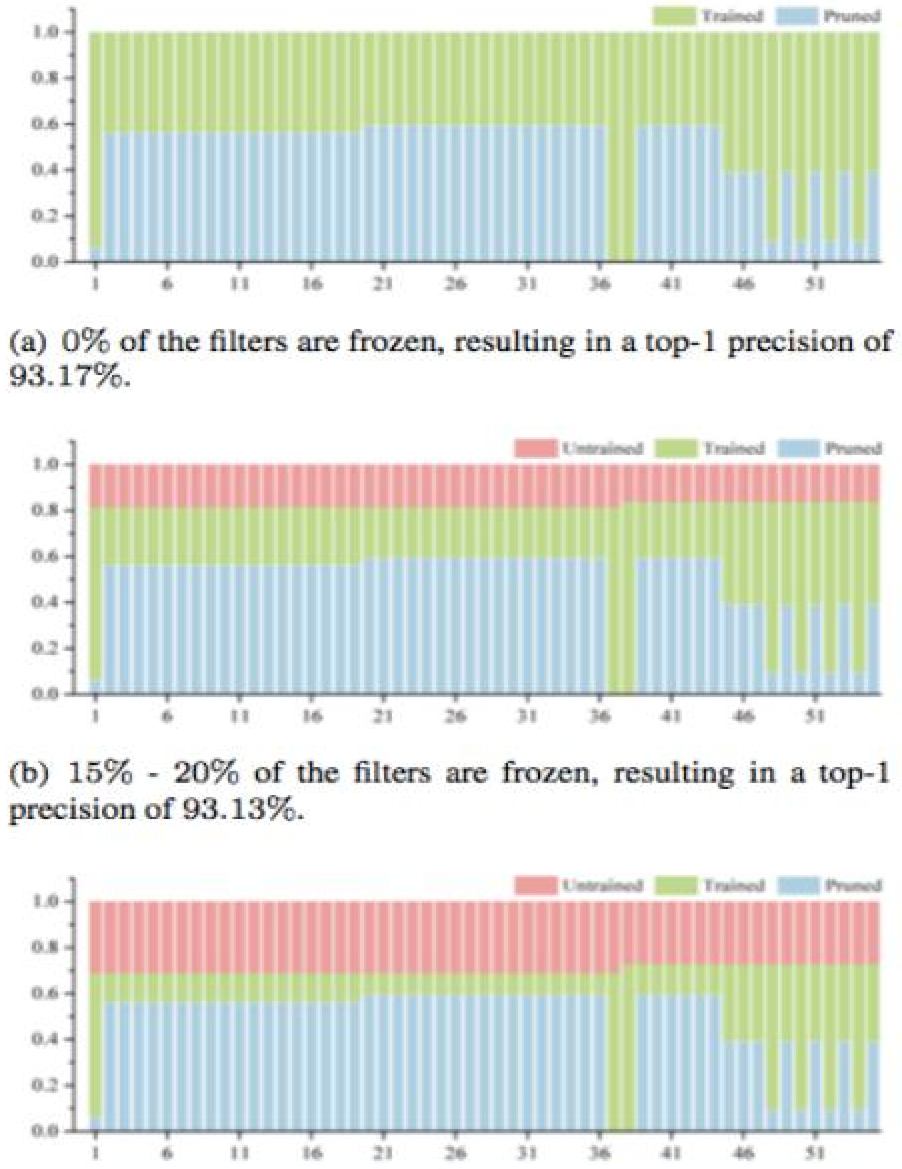
(3) Reverse: Filters generating high-rank feature maps are pruned.

Experiments

Ablation Study(Freezing Filters during Fine-tuning)

- We show that not updating filters with higher-rank feature maps does little damage to the model performance.

- Fig. 5 well supports our claim that feature maps with high ranks contain more information.



(a) 0% of the filters are frozen, resulting in a top-1 precision of 93.17%.

(b) 15% - 20% of the filters are frozen, resulting in a top-1 precision of 93.13%.

(c) 20% - 25% of the filters are frozen, resulting in a top-1 precision of 93.01%.

Figure 5. How freezing filter weights during fine-tuning affects the top-1 precision. The x-axes denote the indices of convolutional layers. The blue, green and red denote the percentage of pruned, trained and untrained filters, respectively.