Supplementary Material for CICC: Channel Pruning via the Concentration of Information and Contributions of Channels

Yihao Chen[1]
yihaochen@zju.edu.cn
Zhishan Li[2]
zhishanli@zju.edu.cn
Yingqing Yang[1]
yingqingyang@zju.edu.cn
Lei Xie[2]
leix@iipc.zju.edu.cn
Yong Liu[3]
yongliu@iipc.zju.edu.cn
Longhua Ma[4]
lhma_zju@zju.edu.cn
Shanqi Liu[2]
shanqiliu@zju.edu.cn
Guanzhong Tian*[1]
gztian@zju.edu.cn

[1] Ningbo Research Institute
Zhejiang University
Ningbo, China

[2] College of Control Science and Engineering
Zhejiang University
Hangzhou, China

[3] Zhejiang – Singapore Innovation and AI Joint Research Lab

[4] College of Information Science and Engineering
NingboTech University
Ningbo, China

1 Algorithm Description

In the pruning phase, we employ three pruning schedules. (1) One-shot: We remove the channels with the least contributions evaluated by Shapley values in multiple convolutional layers at once. (2) Iterative static: We prune the layers via the Shapley values calculated from the pre-trained model, and fine-tune the network after pruning each layer. (3) Iterative dynamic: When iteratively pruning and fine-tuning, we re-calculate the Shapley values in each convolutional layer with the fine-tuned weights, then prune the channels with the least contributions.

Alg. 1 illustrates our method in an iterative static pruning manner, which we employ in the experiments on CIFAR-10. To be specific, we first initialize the network with the weights of the pre-trained model. Then we calculate the overall indicator of the rank and entropy via the fusion function for the convolutional layers by feeding batches of images. Based on the fusion values, we assign the numbers of channels to be removed for the layers. When forward passing the model to the sampled batches of images, we also obtain the Shapley values for

* Corresponding author

Algorithm 1 Algorithm Description

```
Input: The pre-trained model $T$ consisting of $L$ layers, model weights $W$, training data
    $X = \{X_1, X_2, \dots, X_N\}$ with $N$ samples, and convolutional layers of $T$: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_L\}$.
 1: Initialize: $W$ with the weights of $T$
 2: Inference for obtaining the fusion of rank and entropy:
        • Forward pass $T$ to sample minibatch of $B$ examples $\{X_1, X_2, \dots, X_B\}$ to calculate
          $O(\mathcal{C}) = \{O(\mathcal{C}_1), O(\mathcal{C}_2), \dots, O(\mathcal{C}_L)\}$ and $\varphi_{\mathcal{C}}(\hat{f}) = \{\varphi_{\mathcal{C}_1}(\hat{f})\}, \varphi_{\mathcal{C}_2}(\hat{f})\}, \dots, \varphi_{\mathcal{C}_L}(\hat{f})\}$.
        • $u = \{u_1, u_2, \dots, u_L\}$ target numbers of channels to be removed are assigned based
          on $O(\mathcal{C})$.
 3: for $i = 1; i \leq L; ++i$ do
 4:     Sort the Shapley values of the channels: $\{\varphi_{\mathcal{C}_i^1}(\hat{f}), \varphi_{\mathcal{C}_i^2}(\hat{f}), \dots, \varphi_{\mathcal{C}_i^{n_i}}(\hat{f})\}$.
 5:     Remove $u_i$ unimportant channels and the corresponding filters.
 6:     Fine-tune the model weights $W$ with $N$ samples.
 7: end for
 8: Obtain the layer-wise pruned and fine-tuned model with weights $W^{'}$.
 9: for $epoch = 1; epoch \leq epoch_{max}; ++epoch$ do
10:     Retrain the model weights $W^{'}$ with $N$ samples.
11: end for
12: Obtain the compact retrained model with weights $W^{''}$.
Output: the compact model with parameters $W^{''}$
```

the channels in the convolutional layers. In the pruning phase, we consider the channels in a convolutional layer as the players. Hence, the Shapley values represent their average marginal contributions to the output of the layer. Thus, we sort the Shapley values to identify the smallest ones and discard the unimportant channels with the fewest contributions and the corresponding filters. After pruning a layer, we fine-tune the model with several epochs to reduce the loss resulting from pruning, and the process repeats until the final layer is pruned. To minimize the error caused by channel pruning, we train the pruned model again after layer-wise pruning and fine-tuning, then the accuracy of the compressed model is recovered.

2 More Results on CIFAR-10

Tab. 1 shows the performance of the proposed method with the iterative pruning schedule compared with other methods for pruning ResNet-20 and ResNet-32 on the CIFAR-10 dataset.

For ResNet-20, compared with SFP and FPGM, CICC is advantageous in accuracy drop (0.35% *v.s.* 1.37% by SFP and 1.11% by FPGM) and FLOPs reduction (45.2% *v.s.* 42.2% by SFP and 42.2% by FPGM). Besides, CICC achieves an accuracy drop of 0.93% with an acceleration ratio of 57.9%, outperforming DSA and FPGM (1.06% by DSA and 1.76% by FPGM in accuracy drop, 50.3% by DSA and 54.0% by FPGM in FLOPs reduction).

For ResNet-32, CICC achieves an accuracy increase of 0.01%, better than SFP and FPGM which degrade the accuracy by 0.55% and 0.32%, respectively. Moreover, CICC yields a larger FLOPs reduction (38.0% by ACTD, 46.8% *v.s.* 41.5% by SFP and 41.5% by FPGM). Compared with PScratch and FPGM, CICC achieves an accuracy drop (0.70%) better than PScratch (1.00%) and the same as FPGM (0.70%). Moreover, 58.0% of FLOPs are reduced

Model	Method	Base. Acc. (%)	Accl. Acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)	Params ↓ (%)
ResNet-20	SFP [1]	92.20	90.83	1.37	42.2	–
	FPGM [2]	92.20	91.09	1.11	42.2	-
	CICC	91.73	91.38	**0.35**	**45.2**	**40.3**
	DSA [5]	92.17	91.38	1.06	50.3	–
	FPGM [2]	92.20	90.44	1.76	54.0	–
	CICC	91.73	90.80	**0.93**	**57.9**	**43.9**
ResNet-32	ACTD [6]	93.18	93.27	**-0.09**	38.0	**49.0**
	SFP [1]	92.63	92.08	0.55	41.5	–
	FPGM [2]	92.63	92.31	0.32	41.5	–
	CICC	92.63	92.64	-0.01	**46.8**	40.3
	PScratch [7]	93.18	92.18	1.00	50.0	–
	FPGM [2]	92.63	91.93	**0.70**	53.2	–
	CICC	92.63	91.93	**0.70**	**58.0**	**43.9**

Table 1: Comparison of pruned ResNet-20 and ResNet-32 on CIFAR-10.

Model	Method	Base. Top-1 Acc. (%)	Accl. Top-1 Acc. (%)	Base. Top-5 Acc. (%)	Accl. Top-5 Acc. (%)	Top-1 Acc. ↓ (%)	Top-5 Acc. ↓ (%)	FLOPs ↓ (%)	Params ↓ (%)
ResNet-18	CICC	69.76	68.27	89.08	88.26	**1.49**	**0.82**	40.7	**40.6**
	SFP [1]	70.28	67.10	89.63	87.78	3.18	1.85	**41.8**	–
	FPGM [2]	70.28	68.34	89.63	88.53	1.94	1.10	**41.8**	–
	CICC	69.76	67.96	89.08	88.10	**1.80**	**0.98**	**45.1**	**41.7**
ResNet-34	SFP [1]	73.92	71.83	91.62	90.33	2.09	1.29	41.1	–
	FPGM [2]	73.92	72.54	91.62	91.13	1.38	**0.49**	41.1	–
	CICC	73.31	72.74	91.42	90.86	**0.57**	0.56	**45.2**	**36.3**
	CICC	73.31	72.25	91.42	90.63	**1.06**	**0.73**	**50.1**	**36.0**

Table 2: Comparison of pruned ResNet-18 and ResNet-34 on ImageNet.

and 43.9% of parameters are removed.

3 More Results on ImageNet

Tab. 2 shows the performance of our proposed method with the one-shot pruning schedule compared with other methods for pruning ResNet-18 and ResNet-34 on the ImageNet dataset.

For ResNet-18, CICC achieves 1.49%/0.82% Top-1/Top-5 accuracy drop, better than SFP and FPGM (3.18%/1.85% Top-1/Top-5 accuracy drop by SFP and 1.94%/1.10% Top-1/Top-5 accuracy drop by FPGM). Moreover, CICC reduces 40.7% FLOPs and removes 40.6% parameters. Besides, CICC performs better than SFP and FPGM under the FLOPs and parameters reduction ratio of 45.1% and 41.7%, respectively, achieving 1.80%/0.98% Top-1/Top-5 accuracy drop.

For ResNet-34, CICC achieves a Top-1 accuracy drop (0.57%) better than SFP (2.09%) and FPGM (1.38%), with a larger FLOPs reduction (45.2% *v.s.* 41.1% by SFP and 41.1% by FPGM) and a parameters reduction of 36.3%. Besides, CICC achieves loss in Top-1/Top-5 accuracy of 1.06%/0.73% when reducing 50.1% FLOPs and removing 36.0% parameters.

4 Comparison of Multiple Pruning Schedules

Tab. 3 shows the performance of the models with multiple pruning schedules on CIFAR-10. The one-shot pruning schedule prunes the network all at once, which is the most time-saving. But in our experiments, only VGG-16 under the acceleration ratio of 52.3% and compression ratio of 45.7% with the one-shot schedule performs the best among the three schedules. The pruned networks with iterative static and iterative dynamic pruning schedule do not harm the accuracy seriously and even lead to an accuracy increase on ResNet-32 under the FLOPs drop of 46.8%, ResNet-56 under the FLOPs drop of 45.5% and ResNet-110 under the FLOPs drop of 45.6% and 58.1%. But it is worth noting that the iterative static pruning schedule does not require calculating the Shapley values on each convolutional layer when layer-wisely fine-tuning the network, hence, it achieves the most satisfying accuracy-efficiency trade-off.

5 All Visualization of the Rank and Entropy Under Image Batches

Fig. 1 shows the scaled average rank and entropy per channel for the outputs of convolutional layers under different image batches.

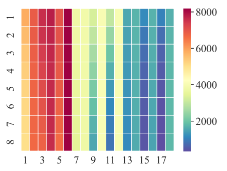
6 All Visualization for Concentration of Information

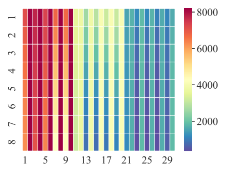
Fig. 2 and Fig. 3 show the rank and entropy for the outputs of convolutional layers and the corresponding fusion values of the stages.

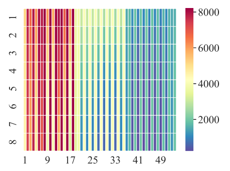
7 Setting of Pruning Rates

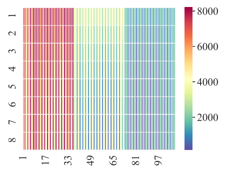
A constant pruning rate for all layers: The idea of employing a pruning rate to prune the same percentage of filters/channels for each convolutional layer is widely used by structured pruning algorithms, but we provide a new perspective to assign the pruning rates for different convolutional layers based on the concentration of information for the convolutional layers. Tab. 4 compares the performance of setting a constant pruning rate to prune each layer and our method for ResNet-56 leading to the FLOPs reduction of about 45% and parameters reduction of about 47% on CIFAR-10. For the three pruning schedules, the pruned models whose numbers of channels for pruning the convolutional layers are assigned by our method all outperform those with a constant pruning rate for all convolutional layers.

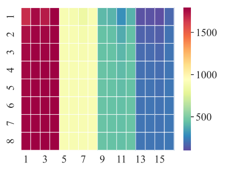
Different pruning rates for each layer: PFEC [3] sets different numbers of channels for pruning the convolutional layers based on the sensitivity of each layer, while HRank [4] pre-defines pruning rates for the layers empirically. Hence, we compare our method for assigning

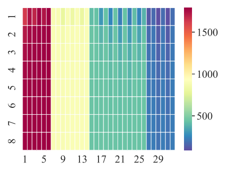

(a) ResNet-20 - Rank

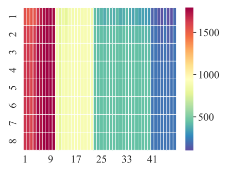

(b) ResNet-32 - Rank

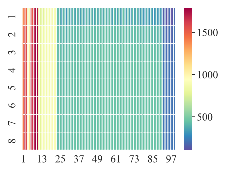

(c) ResNet-56 - Rank

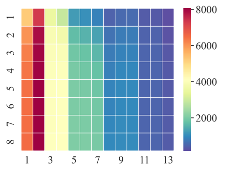

(d) ResNet-110 - Rank

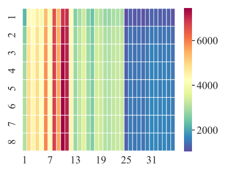

(e) ResNet-18 - Rank

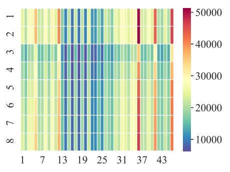

(f) ResNet-34 - Rank

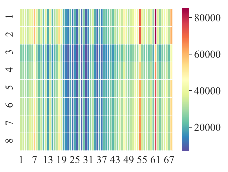

(g) ResNet-50 - Rank

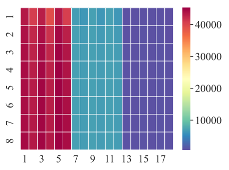

(h) ResNet-101 - Rank

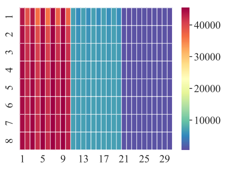

(i) VGG-16 - Rank

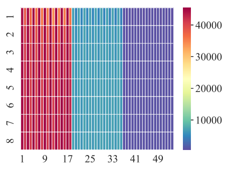

(j) DenseNet-40 - Rank

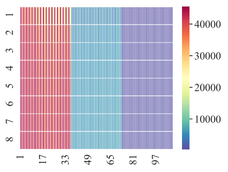

(k) YOLOv5s - Rank

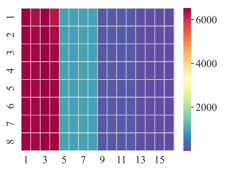

(l) YOLOv5m - Rank

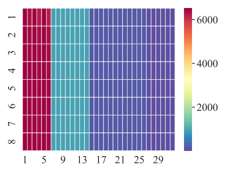

(m) ResNet-20 - Entropy

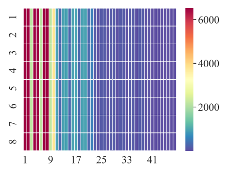

(n) ResNet-32 - Entropy

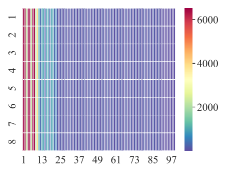

(o) ResNet-56 - Entropy

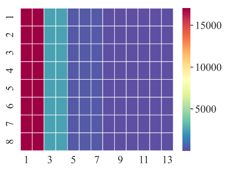

(p) ResNet-110 - Entropy

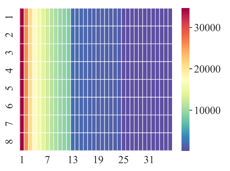

(q) ResNet-18 - Entropy

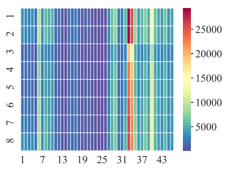

(r) ResNet-34 - Entropy


(s) ResNet-50 - Entropy


(t) ResNet-101 - Entropy


(u) VGG-16 - Entropy


(v) DenseNet-40 - Entropy


(w) YOLOv5s - Entropy

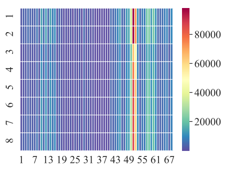
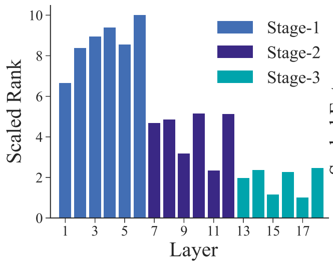

(x) YOLOv5m - Entropy

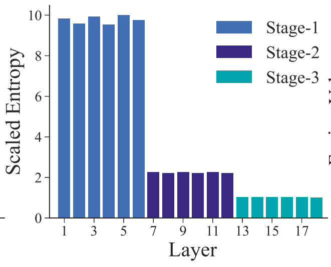
Figure 1: The average rank and entropy per channel for the outputs of convolutional layers under different batches of input images. The x-axis represents the indices of convolutional layers and the y-axis is the number of image batches. The columns of subfigures demonstrate that the rank and entropy for the outputs of convolutional layers is almost unchanged, regardless of the image batches.

Model	Pruning Schedule	Base. Acc. (%)	Accl. Acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)	Params ↓ (%)
VGG-16	OS	93.91	93.27	**0.64**	52.3	45.7
	IS		93.17	0.74		
	ID		93.10	0.81		
	OS	93.91	92.92	0.99	61.0	50.7
	IS		93.24	0.67		
	ID		93.41	**0.50**		
ResNet-20	OS	91.73	91.05	0.68	45.2	40.3
	IS		91.38	**0.35**		
	ID		91.14	0.59		
	OS	91.73	90.70	1.03	57.9	43.9
	IS		90.80	**0.93**		
	ID		90.46	1.27		
ResNet-32	OS	92.63	92.21	0.42	46.8	40.3
	IS		92.64	**-0.01**		
	ID		92.33	0.30		
	OS	92.63	91.57	1.06	58.0	43.9
	IS		91.92	**0.71**		
	ID		91.87	0.76		
ResNet-56	OS	93.39	93.12	0.27	45.5	40.3
	IS		93.60	-0.21		
	ID		93.66	**-0.27**		
	OS	93.39	93.03	0.36	58.1	43.9
	IS		93.11	**0.28**		
	ID		92.86	0.53		
ResNet-110	OS	93.68	94.01	-0.33	45.6	40.4
	IS		94.56	**-0.88**		
	ID		94.28	-0.60		
	OS	93.68	93.62	0.06	58.1	44.0
	IS		94.16	-0.48		
	ID		94.20	**-0.52**		
DenseNet-40	OS	94.22	92.71	1.51	44.4	60.8
	IS		93.56	**0.66**		
	ID		93.34	0.88		
	OS	94.22	91.58	2.64	59.6	68.6
	IS		92.54	1.68		
	ID		92.92	**1.30**		

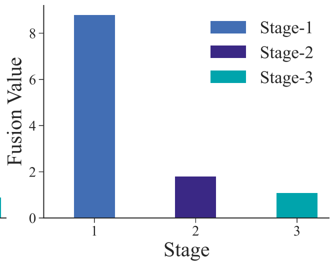
Table 3: Comparison of pruned VGGNet, ResNet and DenseNet with different pruning schedules on CIFAR-10.



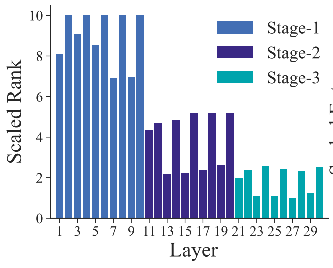
(a) ResNet-20 - Rank



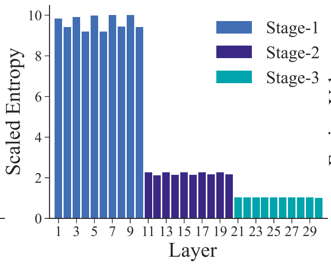
(b) ResNet-20 - Entropy



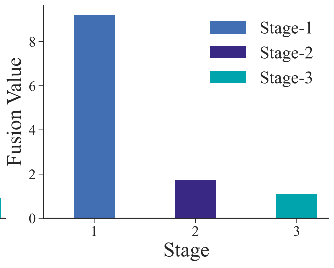
(c) ResNet-20 - Fusion Value



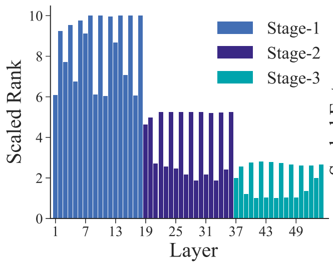
(d) ResNet-32 - Rank



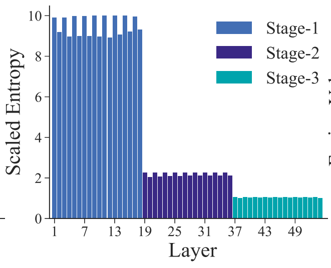
(e) ResNet-32 - Entropy



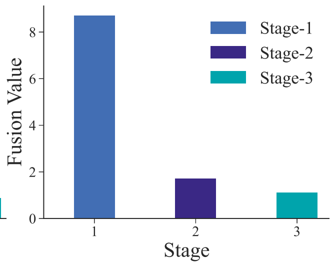
(f) ResNet-32 - Fusion Value



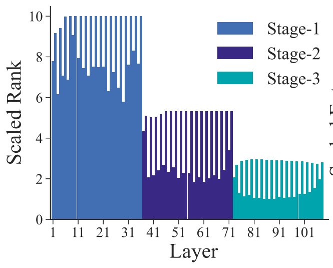
(g) ResNet-56 - Rank



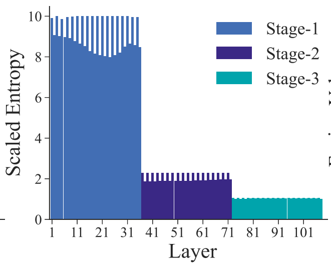
(h) ResNet-56 - Entropy



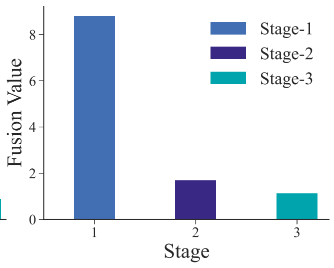
(i) ResNet-56 - Fusion Value



(j) ResNet-110 - Rank



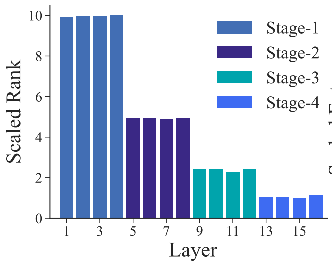
(k) ResNet-110 - Entropy



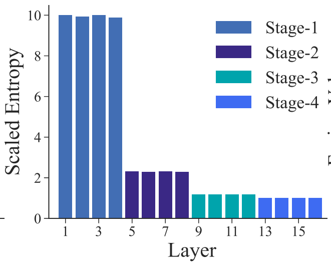
(l) ResNet-110 - Fusion Value

Figure 2: The scaled average rank and entropy per channel for the outputs of convolutional layers and the corresponding fusion values of the stages for architectures.

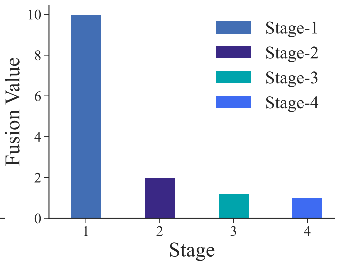
the layer-wise pruning ratio via the concentration of information for the convolutional layers with PFEC and HRank. Note that PFEC prunes the network with one-shot pruning schedule,



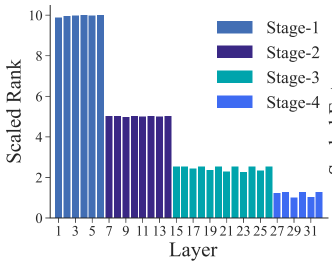
(a) ResNet-18 - Rank



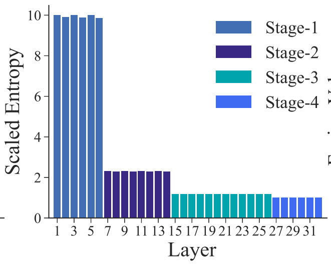
(b) ResNet-18 - Entropy



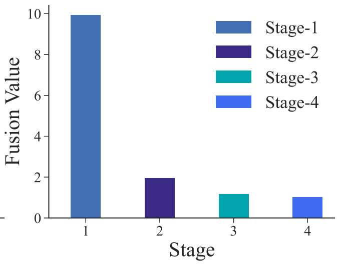
(c) ResNet-18 - Fusion Value



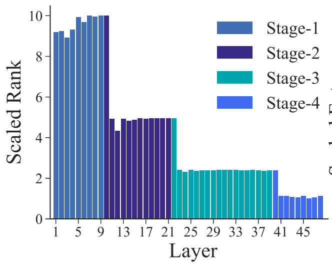
(d) ResNet-34 - Rank



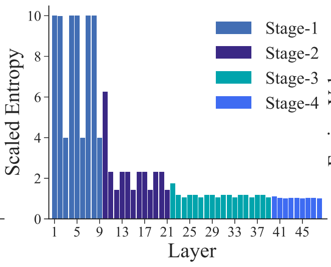
(e) ResNet-34 - Entropy



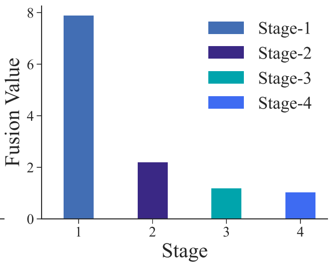
(f) ResNet-34 - Fusion Value



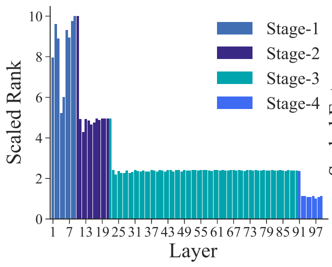
(g) ResNet-50 - Rank



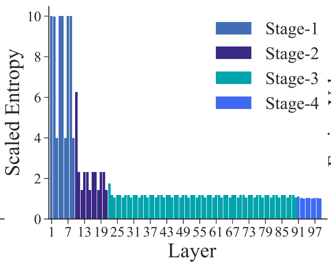
(h) ResNet-50 - Entropy



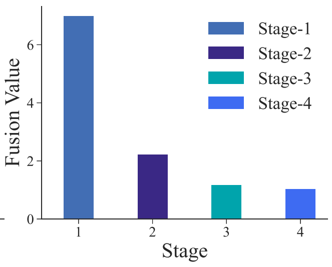
(i) ResNet-50 - Fusion Value



(j) ResNet-101 - Rank



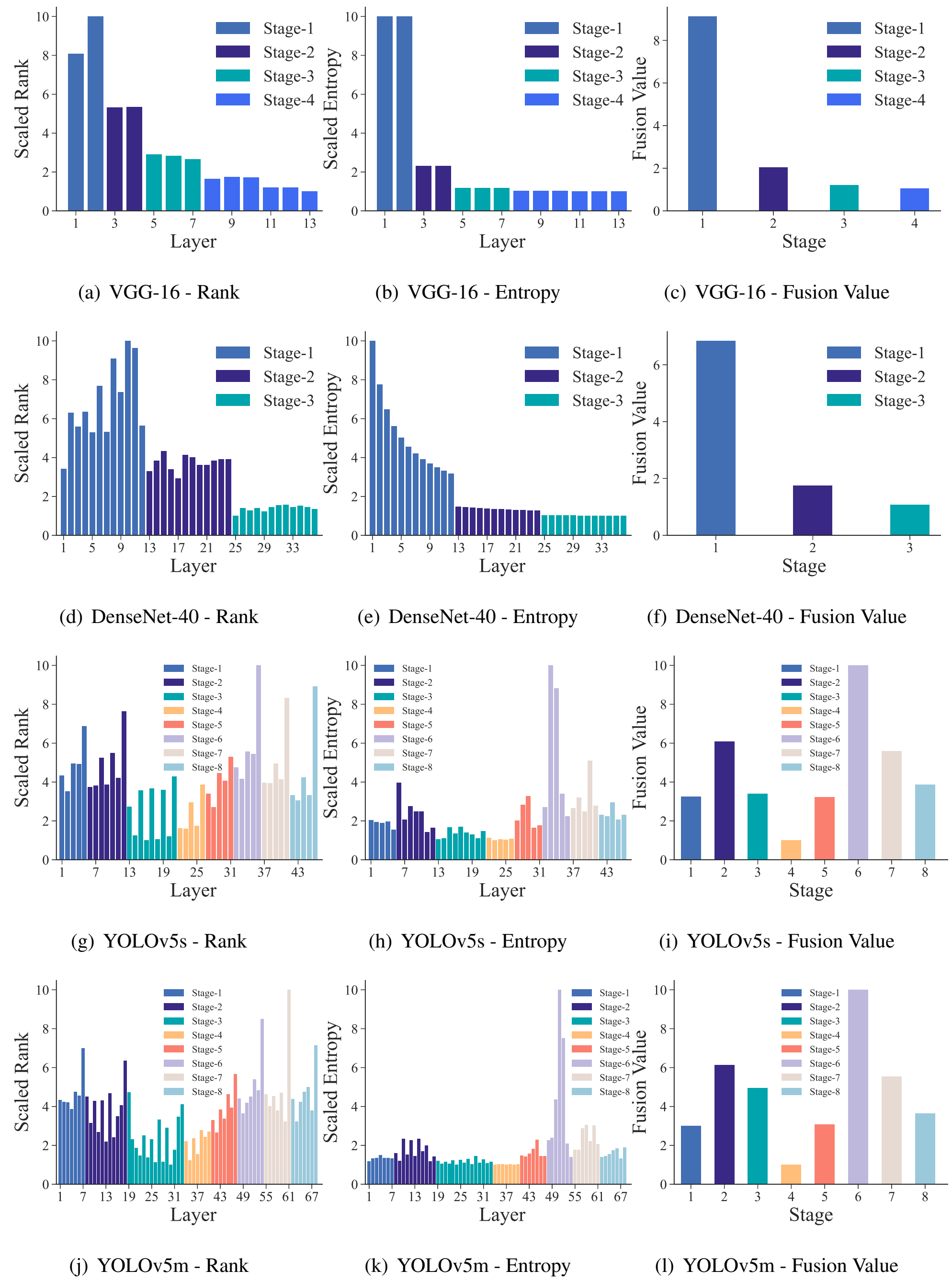
(k) ResNet-101 - Entropy



(l) ResNet-101 - Fusion Value

Figure 3: The scaled average rank and entropy per channel for the outputs of convolutional layers and the corresponding fusion values of the stages for architectures.

and HRank prunes with iterative static pruning schedule via the feature map rank generated from the pre-trained model, so we prune in a one-shot manner and in an iterative static manner



(a) VGG-16 - Rank (b) VGG-16 - Entropy (c) VGG-16 - Fusion Value

(d) DenseNet-40 - Rank (e) DenseNet-40 - Entropy (f) DenseNet-40 - Fusion Value

(g) YOLOv5s - Rank (h) YOLOv5s - Entropy (i) YOLOv5s - Fusion Value

(j) YOLOv5m - Rank (k) YOLOv5m - Entropy (l) YOLOv5m - Fusion Value

Figure 4: The scaled average rank and entropy per channel for the outputs of convolutional layers and the corresponding fusion values of the stages for architectures.

compared with PFEC and HRank, respectively. As shown in Tab. 5, under similar FLOPs drop and larger parameters drop, our method outperforms PFEC (0.55% *v.s.* 0.02% in accuracy

Pruning Schedule	Pruning Rate Setting	Base. Acc. (%)	Accl. Acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)	Params ↓ (%)
One-shot	Fusion (Ours)	93.39	93.15	**0.24**	45.5	47.4
	Constant (0.48)		92.85	0.54	45.4	46.6
Iterative Static	Fusion (Ours)	93.39	93.52	**-0.13**	45.5	47.4
	Constant (0.48)		93.38	0.01	45.4	46.6
Iterative Dynamic	Fusion (Ours)	93.39	93.72	**-0.33**	45.5	47.4
	Constant (0.48)		93.24	0.15	45.4	46.6

Table 4: Comparison of the results achieved by the pruned models whose pruning rates are assigned by our concentration-based method and constantly of 0.48 for ResNet-56 on CIFAR-10.

Pruning Schedule	Pruning Rate Setting	Base. Acc. (%)	Accl. Acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)	Params ↓ (%)
One-shot	Fusion (Ours)	93.39	93.94	**-0.55**	27.9	28.5
	PFEC [3]	93.04	93.06	-0.02	27.6	13.7
Iterative Static	Fusion (Ours)	93.39	93.39	**0.00**	50.6	52.2
	HRank [4]	93.26	93.17	0.09	50.0	42.4

Table 5: Comparison of the results achieved by our method, PFEC and HRank for ResNet-56 on CIFAR-10.

increase) and HRank (0.00% *v.s.* 0.09% in accuracy drop) for ResNet-56 on CIFAR-10. Thus, it demonstrates the excellence to assign specific pruning rates for different layers via the concentration of information, which provides a theoretical guidance for the settings of pruning rates.

8 The Numbers of Removed Channels

Tab. 6 shows the numbers of channels that we assign for pruning the convolutional layers in multiple network architectures, based on the fusion values which represent the concentration of information. In the networks, shallower layers contain less information than the deeper ones, so we assign larger numbers of channels to be removed for pruning the deeper layers.

9 Comparison of Different Scoring Metrics

Tab. 7 compares the performance under different scoring metrics for pruning ResNet-56 on CIFAR-10 including rank, entropy and Shapley Values. Among them, Shapley values perform the best, thus we choose Shapley values to evaluate the importance of the channels to discard the unimportant ones.

VGG - CIFAR-10						
Depth	Stage-1	Stage-2	Stage-3	Stage-4	FLOPs ↓ (%)	Params ↓ (%)
16	16	64	80	128	52.3	45.7
	32	64	96	144	61.0	50.7

ResNet - CIFAR-10					
Depth	Stage-1	Stage-2	Stage-3	FLOPs ↓ (%)	Params ↓ (%)
20	8	16	24	45.2	40.3
	12	20	24	57.9	43.9
32	8	16	24	46.8	40.3
	12	20	24	58.0	43.9
56	8	16	24	45.5	40.3
	12	20	24	58.1	43.9
110	8	16	24	45.6	40.4
	12	20	24	58.1	44.0

DenseNet - CIFAR-10					
Depth	Stage-1	Stage-2	Stage-3	FLOPs ↓ (%)	Params ↓ (%)
40	3	6	9	44.4	60.8
	5	8	9	59.6	68.6

ResNet - ImageNet						
Depth	Stage-1	Stage-2	Stage-3	Stage-4	FLOPs ↓ (%)	Params ↓ (%)
18	8	24	48	80	40.7	40.6
	8	36	48	80	45.1	41.7
34	8	12	24	32	45.2	36.3
	8	20	24	28	50.1	36.0
50	8	48	64	80	41.6	35.0
	8	64	80	112	50.4	44.2
101	16	24	32	96	43.7	42.6
	16	32	42	128	54.4	54.0

Table 6: The numbers of channels for pruning VGGNet, ResNet and DenseNet.

Scoring Metric	Accl. Acc. (%)
Rank	92.55
Entropy	91.17
Shapley values	**92.60**

Table 7: The performance under different scoring metrics.

References

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

2018.

[2] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

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

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

[5] Xuefei Ning, Tianchen Zhao, Wenshuo Li, Peng Lei, Yu Wang, and Huazhong Yang. Dsa: More efficient budgeted pruning via differentiable sparsity allocation. In *European Conference on Computer Vision*, pages 592–607. Springer, 2020.

[6] Wenxiao Wang, Minghao Chen, Shuai Zhao, Long Chen, Jinming Hu, Haifeng Liu, Deng Cai, Xiaofei He, and Wei Liu. Accelerate cnns from three dimensions: A comprehensive pruning framework. In *International Conference on Machine Learning*, pages 10717–10726. PMLR, 2021.

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