IEEE WCCI 2020
IEEE World Congress on Computational Intelligence
Virtual Conference – July 19-24, 2020

Pruning Depthwise Separable Convolutions for MobileNet Compression

Cheng-Hao Tu, Jia-Hong Lee, Yi-Ming Chan, and Chu-Song Chen

Institute of Information Science, Academia Sinica, Taipei, Taiwan

MOST Joint Research Center for AI Technology and All Vista Healthcare, Taipei, Taiwan

SPONSORS:

IEEE Advancing Technology for Humanity
IEEE Computational Intelligence Society
IET
The International Neural Network Society (INNS)
EPS Evolutionary Programming Society

Outline

- Introduction
- Related Work
- Depthwise Separable Convolution
- Multistage Gradual Pruning
- Experiments
- Conclusions

Introduction

- Although convolution neural networks (CNNs) have brought many breakthroughs in computer vision tasks, they usually consume lots of computational resources in the inference stage.

- This limits the applications of CNNs on resource-limited devices, such as home robots and mobile phones.

- How to build lightweight CNNs that require less resources is a crucial problem.

Related Work

- **Lightweight architectures designing** looks for efficient blocks for CNNs so that parameters can be used more effectively, and thus CNNs can be smaller.

- MobileNetV1 [1] utilizes depthwise separable convolutions that decompose standard convolution into depthwise and pointwise convolutions for reducing model parameters.

- MobileNetV2 [2] further enhances the information flow by introducing linear bottlenecks so that it requires much less parameters and has performance improvements.

[1] Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017
[2] Sandler et al., ""Mobilenetv2: Inverted residuals and linear bottlenecks," " in CVPR 2018, pp. 4510–4520.

Related Work

- **Network pruning** looks for redundant parameters in CNNs so that removing them will not degrade the performance too much.

- Unimportant filters can be pruned from CNNs so that the inference can be accelerated directly using existing deep learning frameworks.

- In [3], they prune weights from CNNs via an iterative pruning and finetuning process so that the performance can be regained. However, weight pruning needs special libraries or hardware to speed up the pruned networks.

[3] Zhu al., “To prune, or not to prune: exploring the efficacy of pruning for model compression, ” ” in ICLR Workshops, 2018.

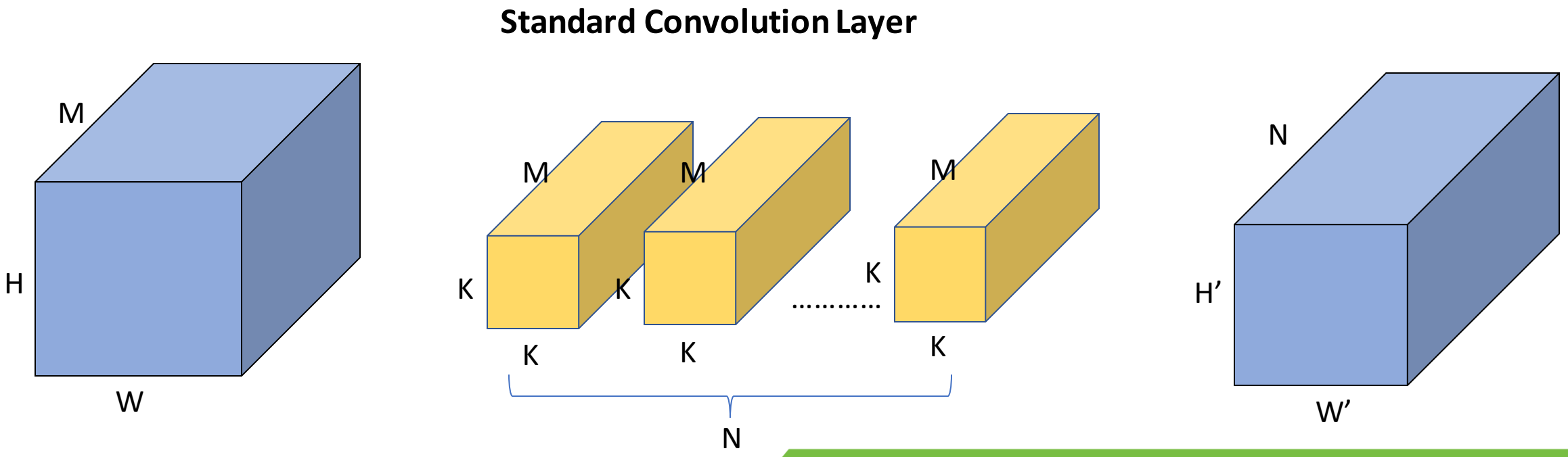
Objective

- In this paper, we focus on further compressing the lightweight networks, like MobileNets, to gain more speedups by filter pruning.

- Given a pretrained MobileNet, we would like to prune unimportant filters to accelerate its inference.

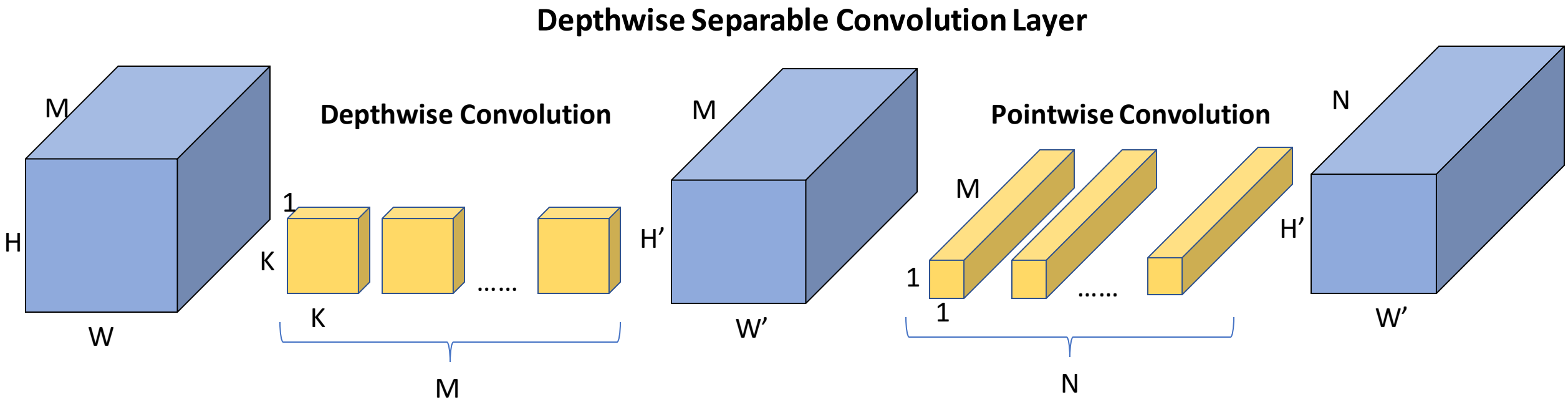
Depthwise Separable Convolution

- A Depthwise separable convolution layer consists of a depthwise convolution and a pointwise convolution.



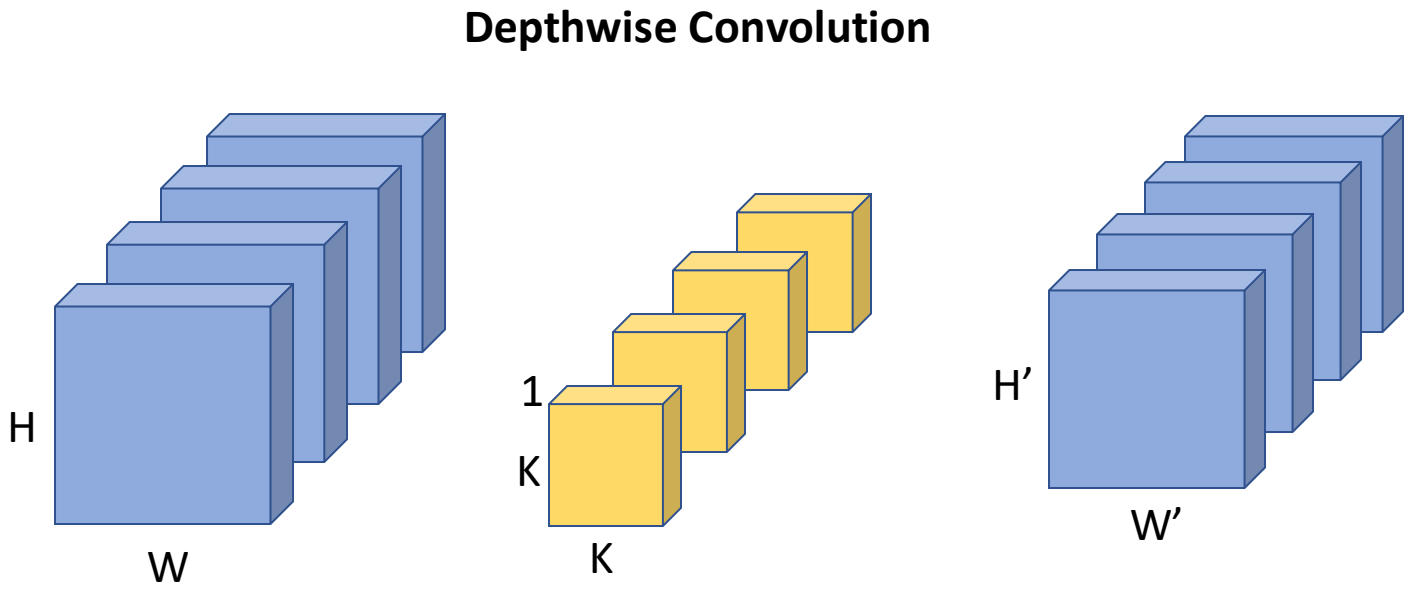
Depthwise Separable Convolution

- A Depthwise separable convolution layer consists of a depthwise convolution and a pointwise convolution.



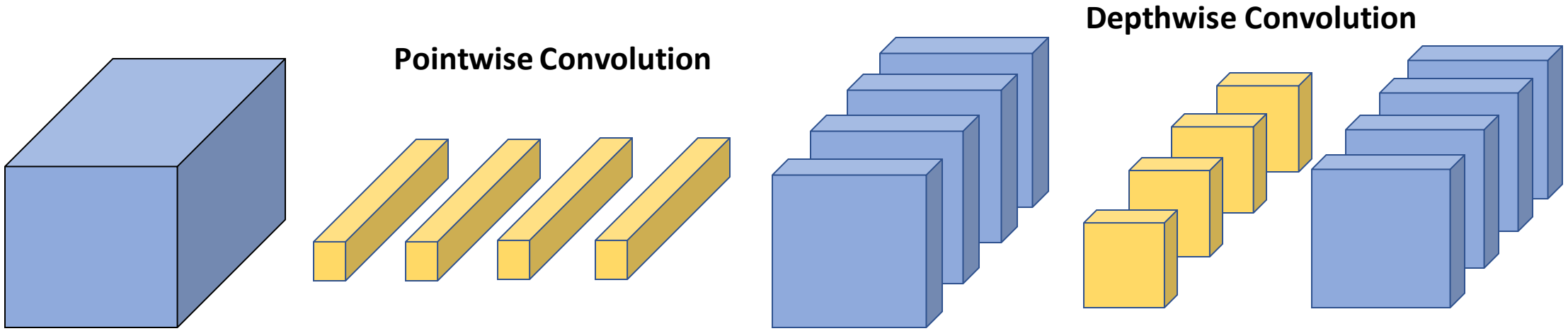
Depthwise Separable Convolution

- In the first depthwise convolution, each filter is applied on the corresponding input channel to produce the corresponding output channel.



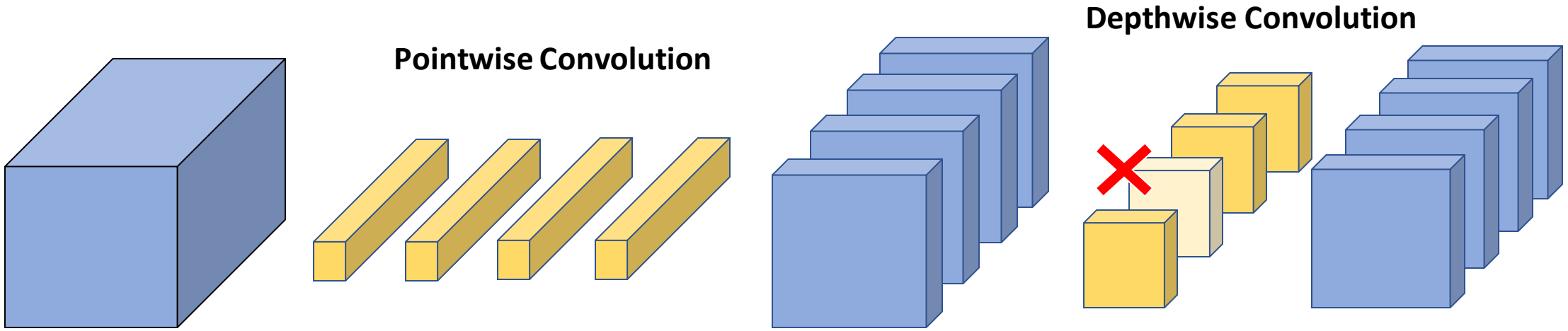
Depthwise Separable Convolution

- Pruning a filter in a depthwise convolution will remove the corresponding filter in the previous pointwise convolution.



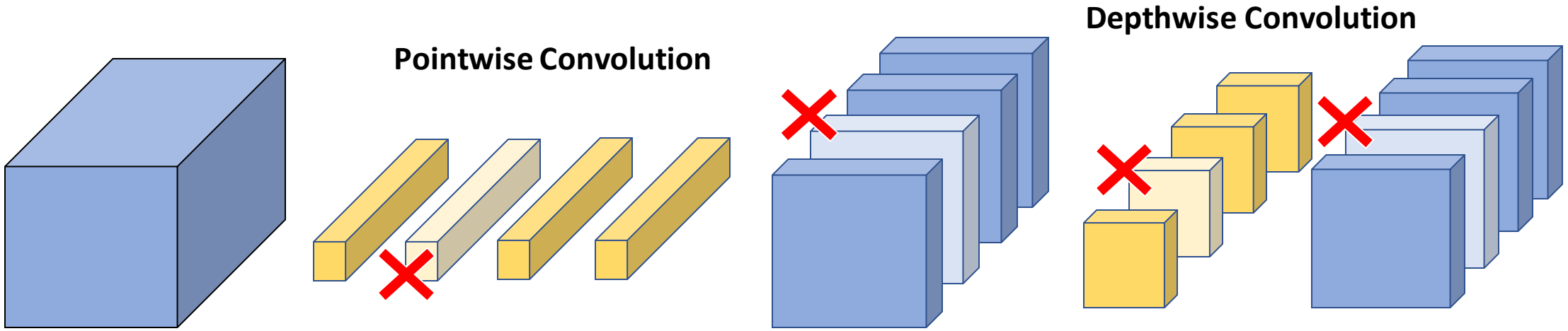
Depthwise Separable Convolution

- Pruning a filter in a depthwise convolution will remove the corresponding filter in the previous pointwise convolution.



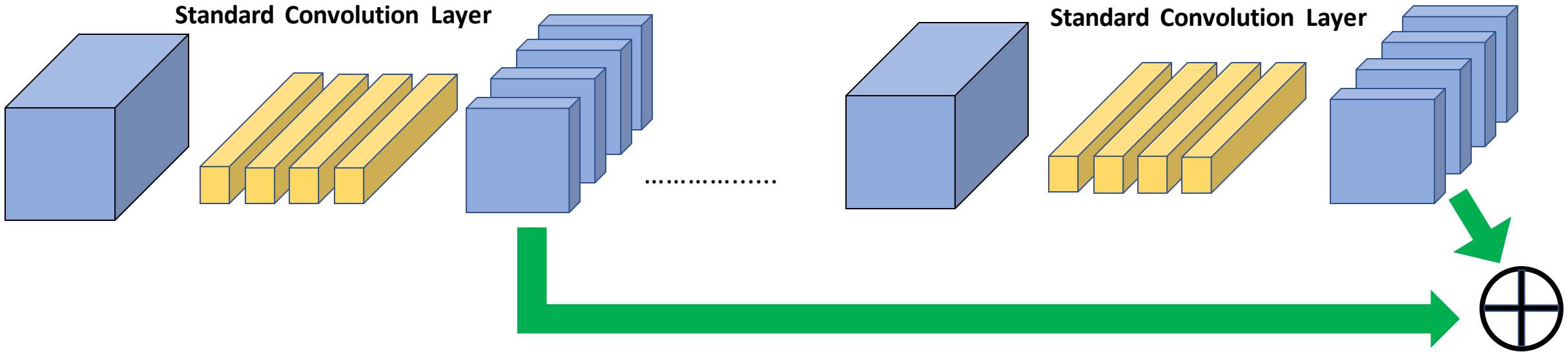
Depthwise Separable Convolution

- Pruning a filter in a depthwise convolution will remove the corresponding filter in the previous pointwise convolution.



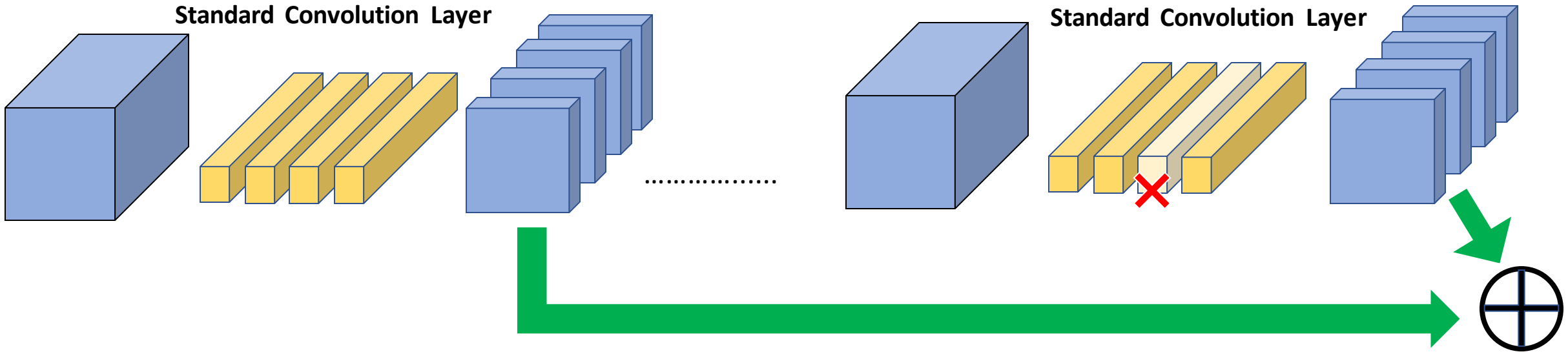
Shortcut connections

- As MobileNetV2 contains shortcut connections, such structure also introduces constraints in filter pruning.



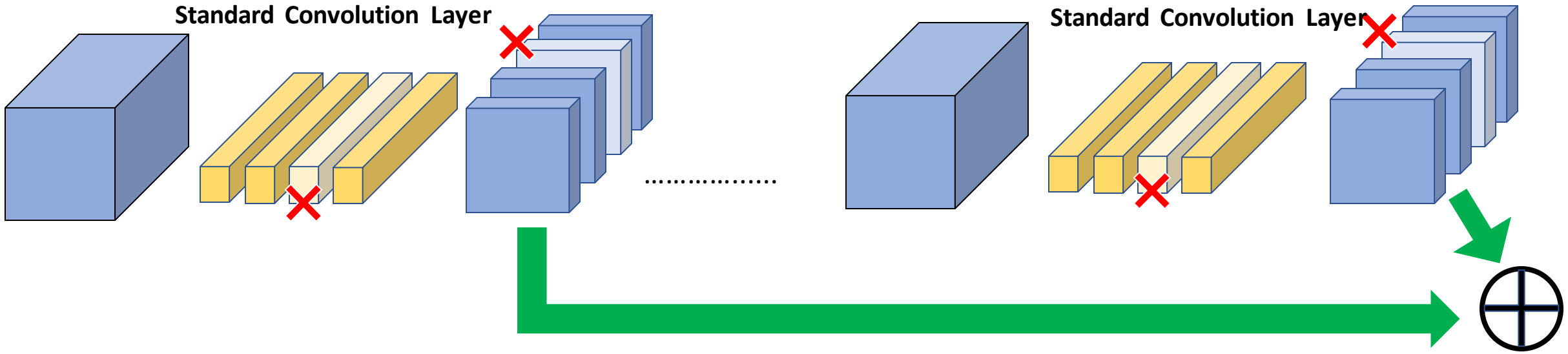
Shortcut connections

- As MobileNetV2 contains shortcut connections, such structure also introduces constraints in filter pruning.



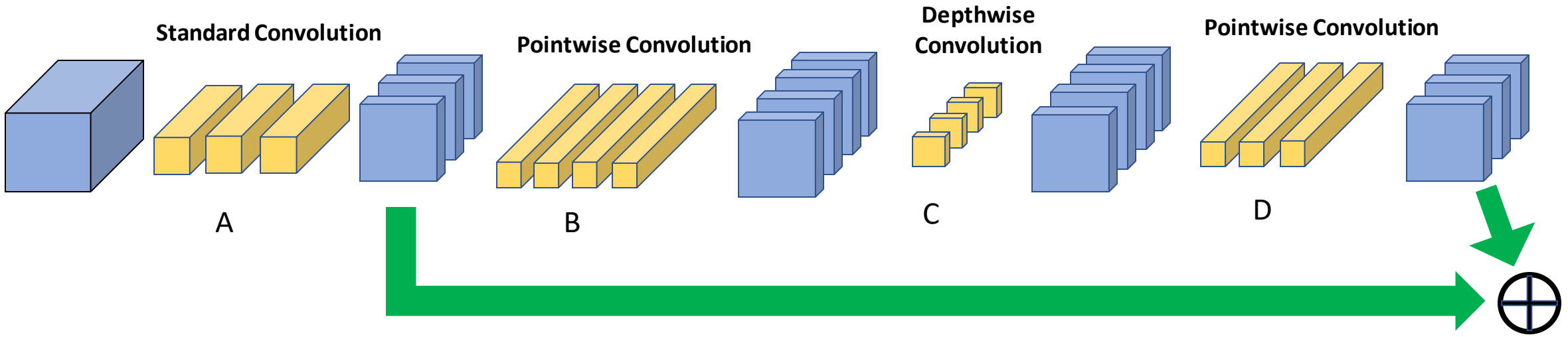
Shortcut connections

- As MobileNetV2 contains shortcut connections, such structure also introduces constraints in filter pruning.



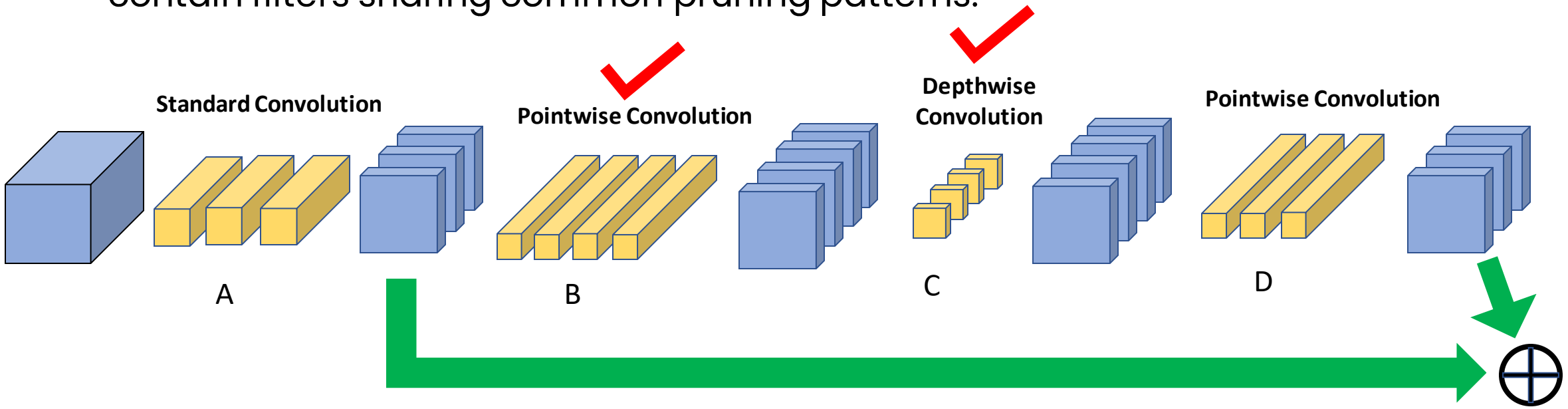
Combining together

- Given a pretrained network, we can group convolutions into various groups that contain filters sharing common pruning patterns.



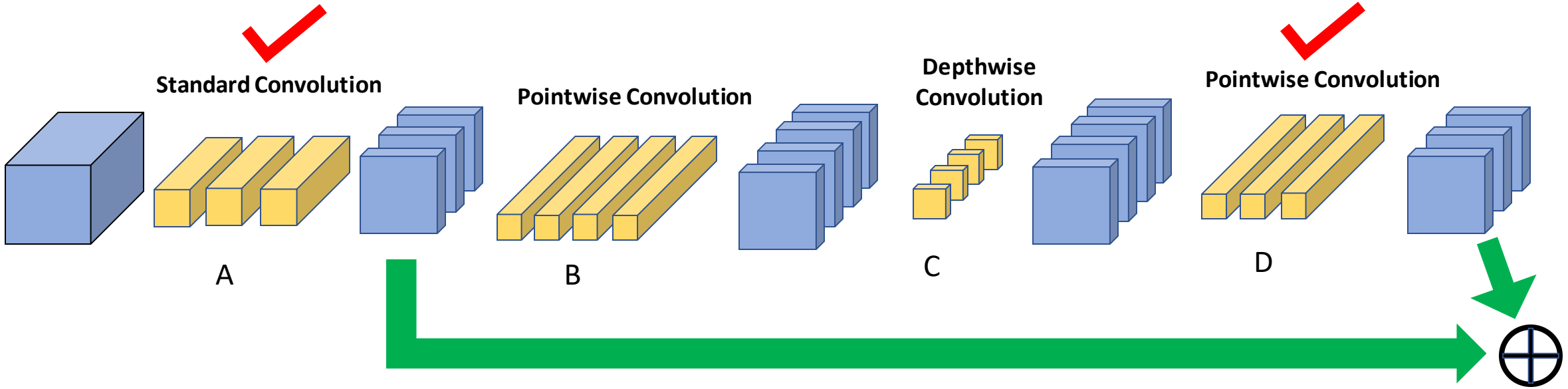
Combining together

- Given a pretrained network, we can group convolutions into various groups that contain filters sharing common pruning patterns.



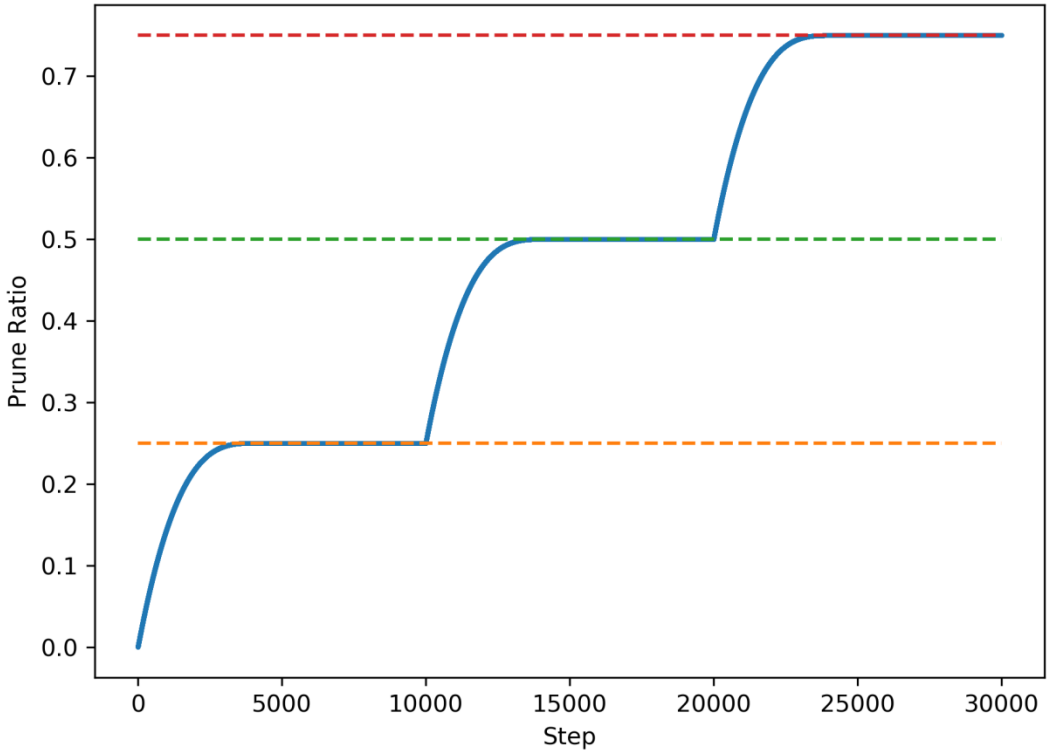
Combining together

- Given a pretrained network, we can group convolutions into various groups that contain filters sharing common pruning patterns.



Multistage Gradual Pruning

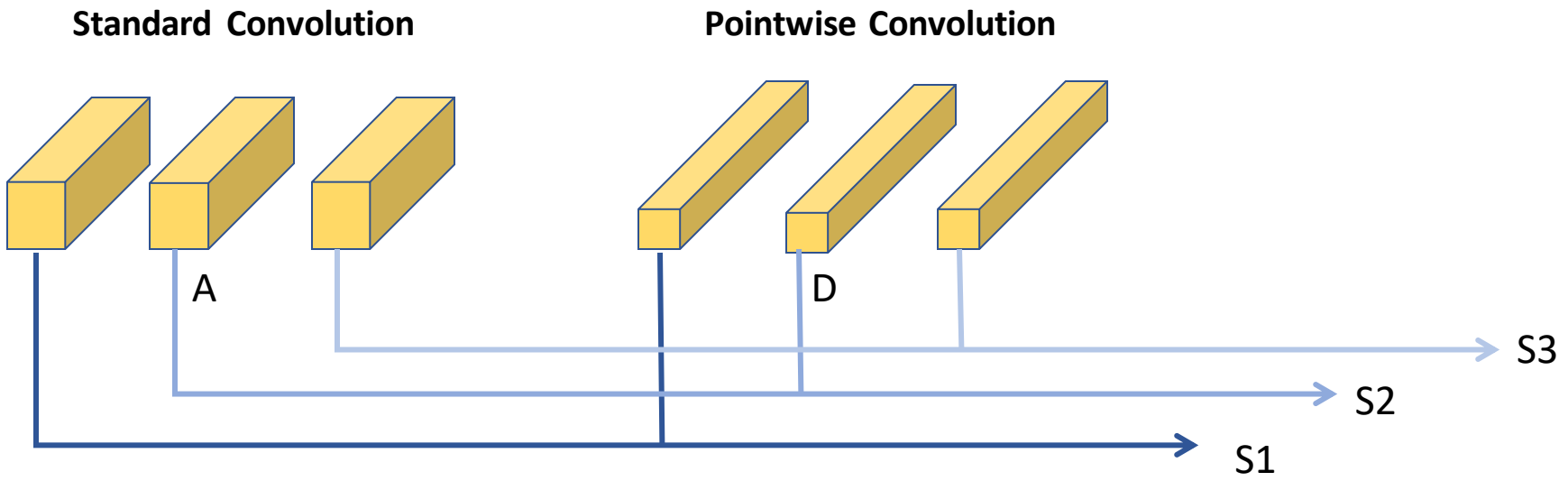
- We prune filters from the network using gradual pruning with multiple stages.



3-stage gradual pruning, and the pruning ratio increases 0.25 in each stage.

Multistage Gradual Pruning

- In each convolution group, we compute scores for filters by summing their absolute values and consider filters with smaller scores as less important ones.



Experiments

- We conduct experiments on 2 small-scale datasets (CIFAR10, SVHN) and 1 large-scale dataset (ImageNet)
- The evaluated architectures are MobileNetV1 and MobileNetV2 that both contain depthwise separable convolutions. MobileNetV2 also contains shortcut connections.
- We implement our method using Pytorch.

Experiments

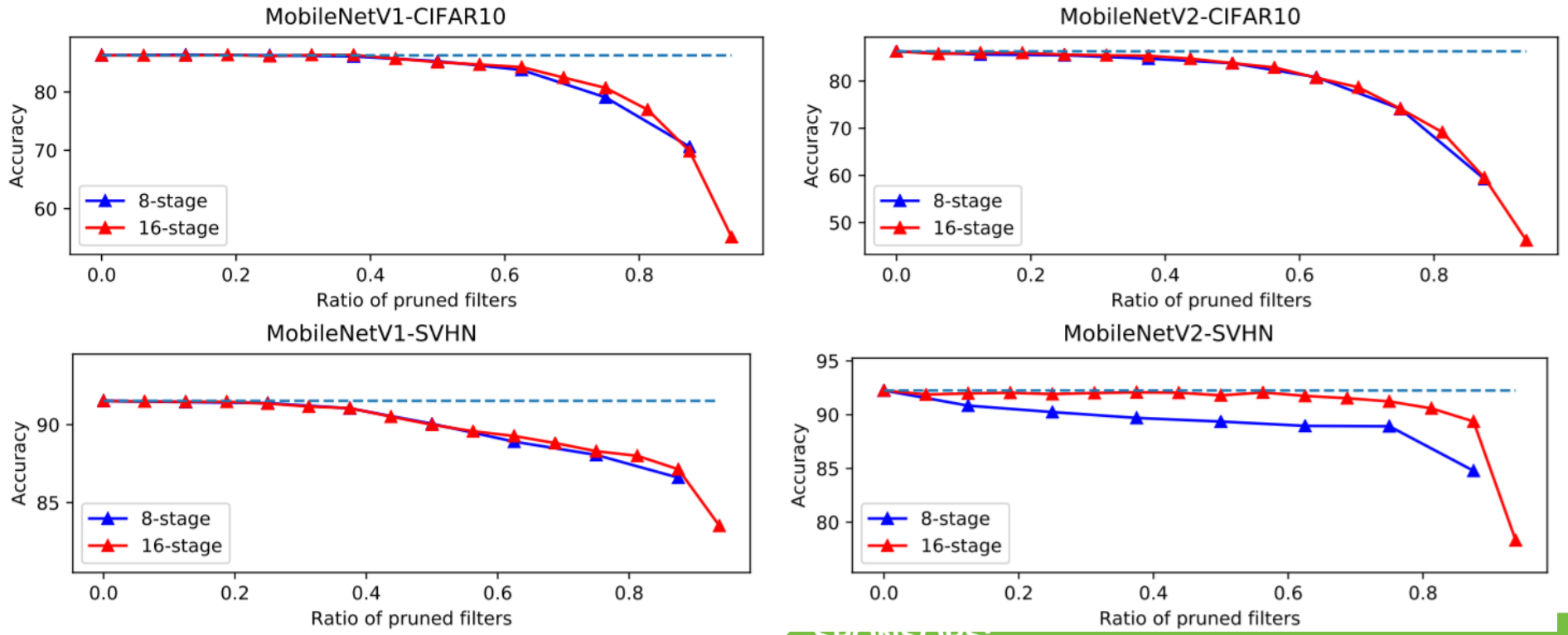
- Results with pruning ratio 0.25 using 2 stages.

MobileNetV1	Baseline Top1 Acc.	Pruned Top1 Acc.	Rel. ↓ %	FLOPs ↓ %
CIFAR10	86.28	86.15	0.15	42.5
SVHN	91.53	91.36	0.19	
ImageNet	70.69	68.84	2.61	

MobileNetV2	Baseline Top1 Acc.	Pruned Top1 Acc.	Rel. ↓ %	FLOPs ↓ %
CIFAR10	86.31	85.61	0.81	41.0
SVHN	92.25	91.91	0.37	
ImageNet	71.88	67.25	6.44	

Experiments

- Results under different pruning stages.



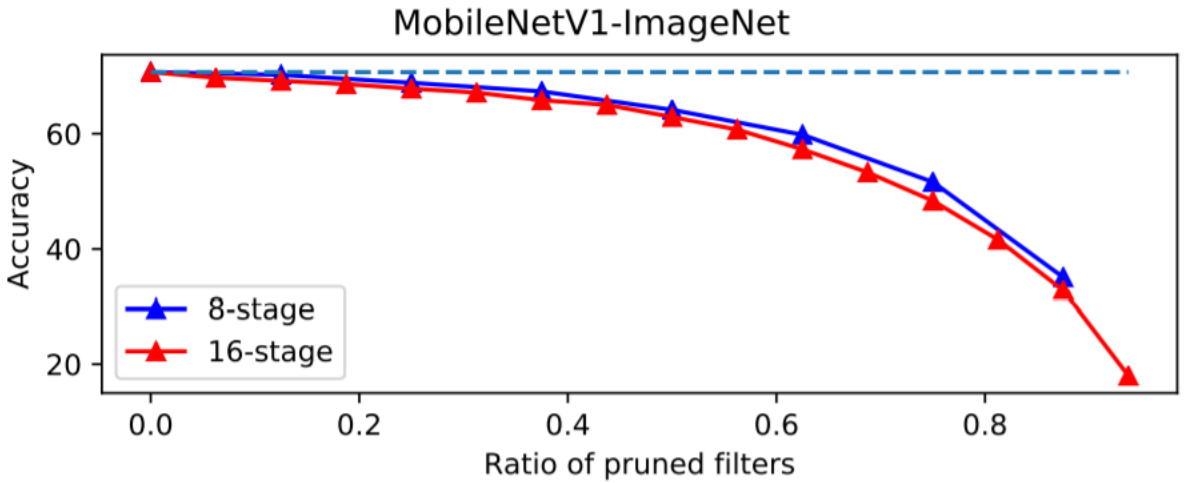
Experiments

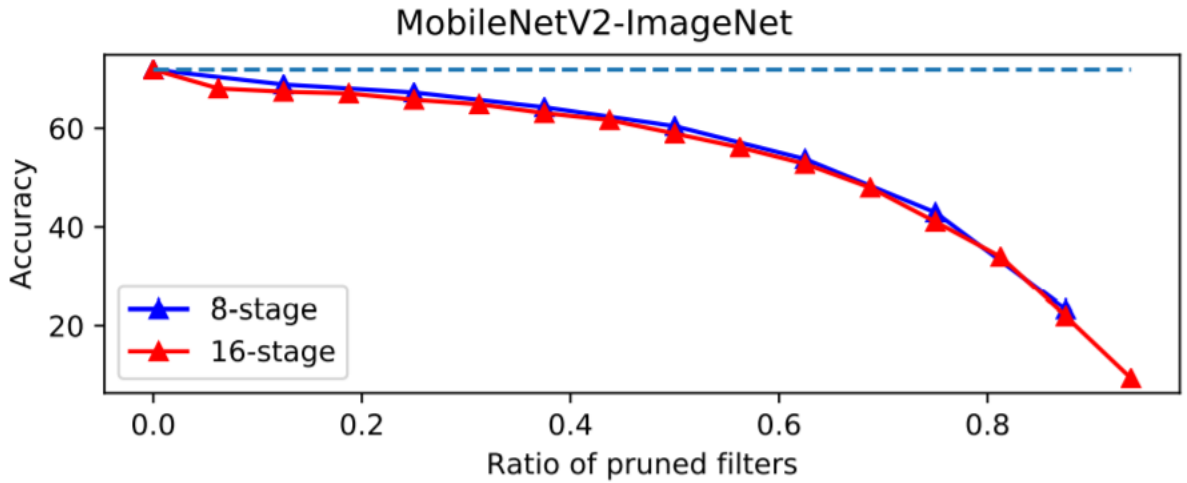
- Results of MobileNetV1 on ImageNet.

Width Multipliers	Ours Acc.	Scratch [1] Acc.	FLOPs ↓ %	Params ↓ %
1.0×	70.69	70.6	-	-
0.75×	68.84	68.4	42.44%	38.90%
0.5×	64.15	63.7	73.26%	68.53%
0.25×	51.62	50.6	92.45%	88.89%

Experiments

- Results on ImageNet.





Conclusions and Future Work

- We demonstrate that our method can compress the MobileNets to a certain extent with satisfiable performance.
- In our method, finer pruning ratios and more pruning stages help maintain the accuracy better.
- We plan to investigate the performance on more networks with depthwise separable convolutions, like MobileNetV3.