清華大學
Tsinghua University

交叉信息研究院
Institute for Interdisciplinary
Information Sciences

Pruning-similar Optimizations

马恺声

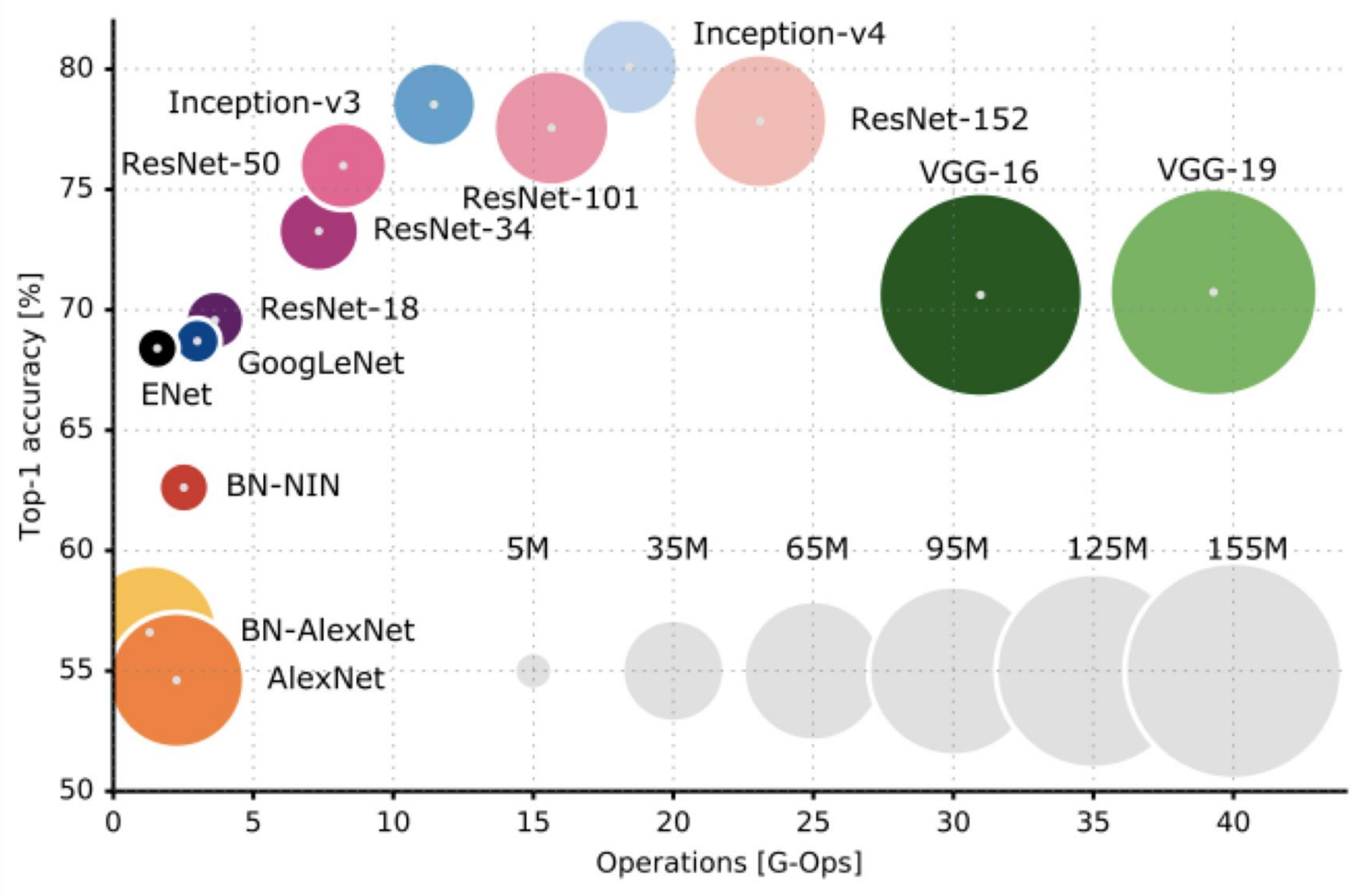
清华大学

Optimizations

- **Optimizations Directions:**
  - Compact Model Design
  - Pruning: Special Topic
  - Low-rank Matrix/Dictionary
  - Distillation: Special Topic

Efficient DNN Models

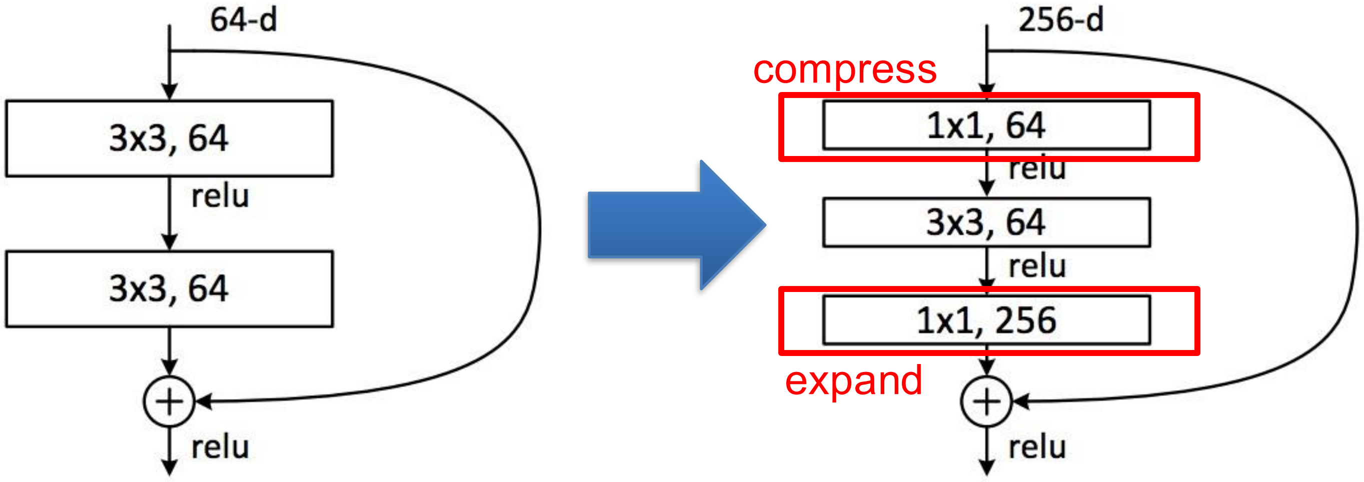
Accuracy vs. Weight & OPs



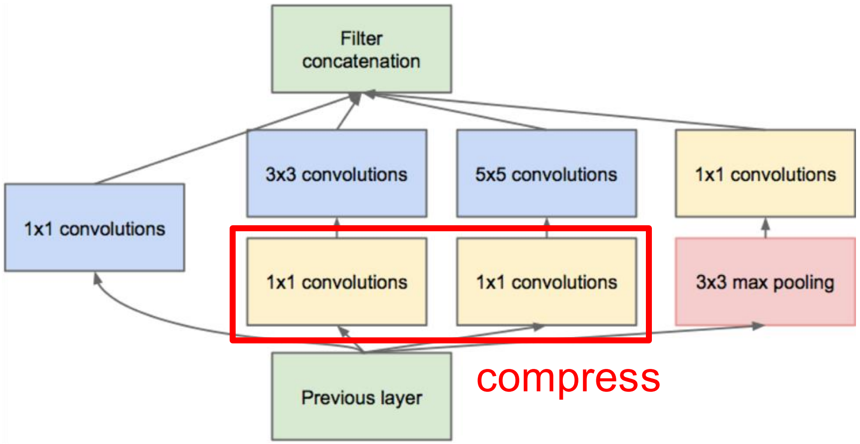
[Alfredo et al., arXiv, 2017]

Bottleneck in Popular DNN Models

ResNet



GoogleNet

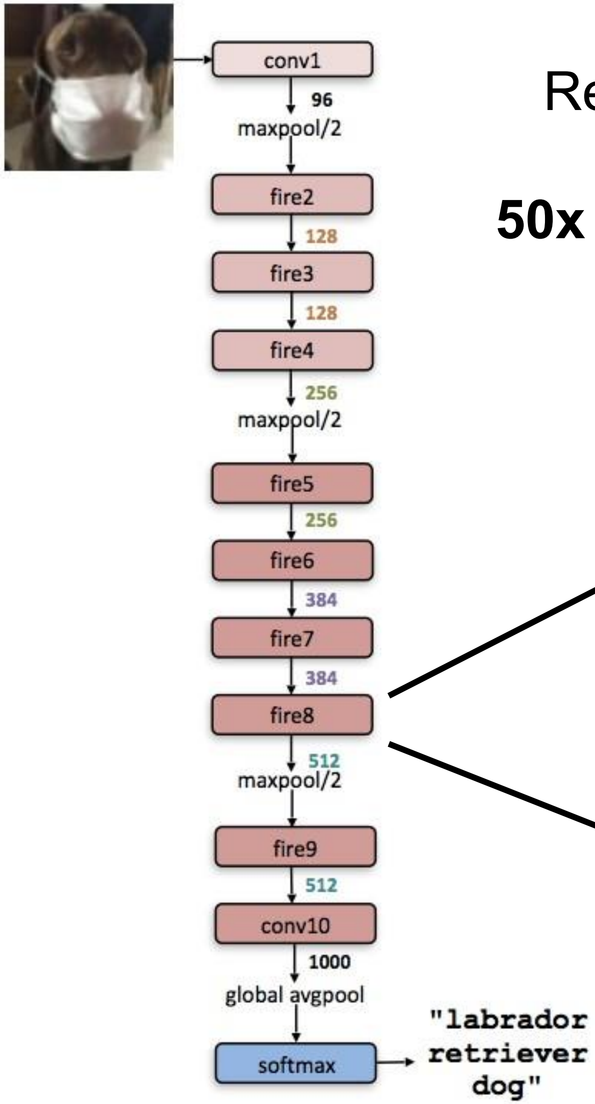


Example: SqueezeNet

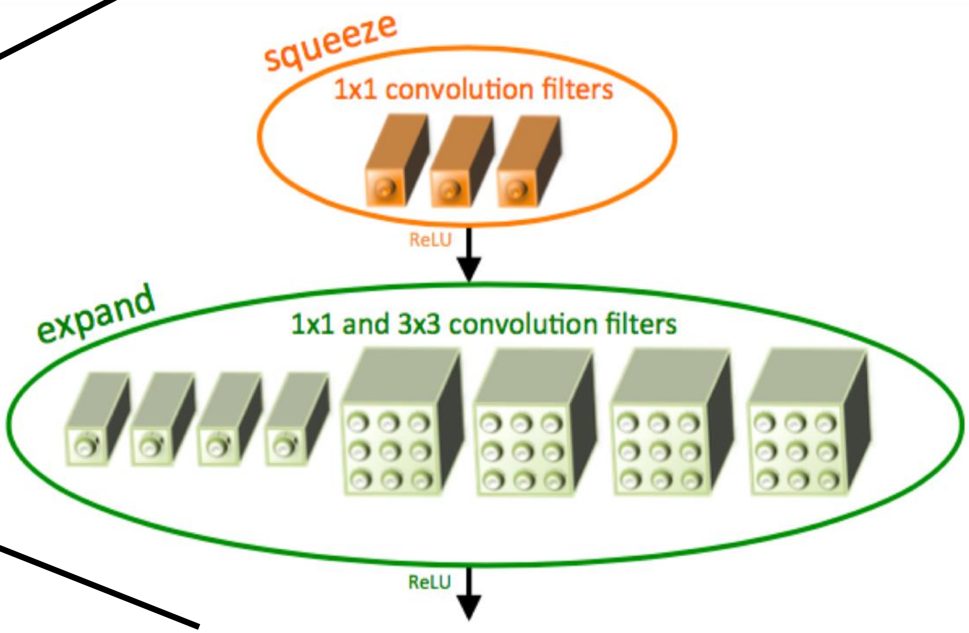
Reduce number of weights by reducing number of input channels by “squeezing” with 1x1

50x fewer weights than AlexNet (no accuracy loss)

However, 2.4x more operations than AlexNet*



Fire Module



[Iandola et al., arXiv 2016, ICLR 2017]

*SqueezeNetv1.0

Stacking Small Filters

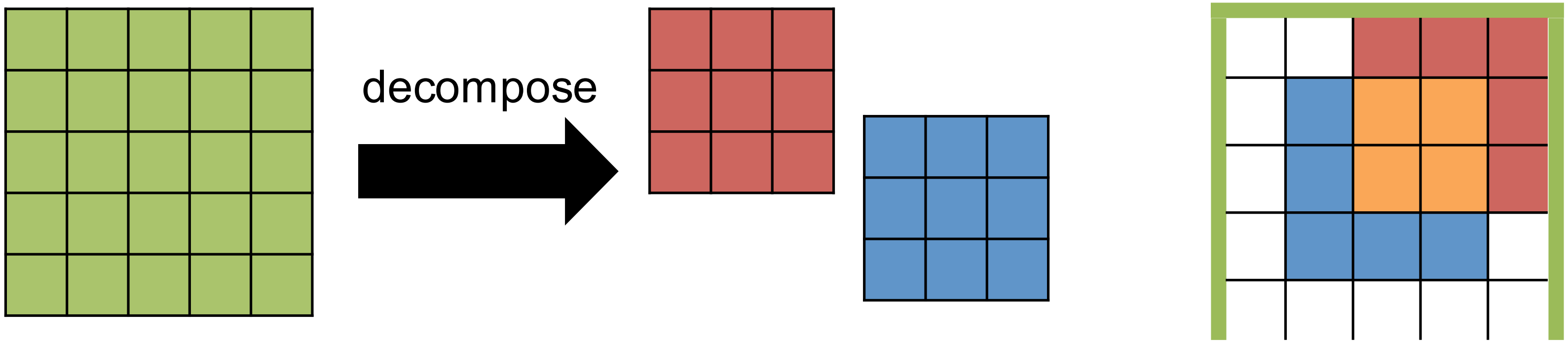
Build network with a **series of small filters**
(reduces degrees of freedom)

VGG-16

5x5 filter

Two 3x3 filters

Apply sequentially



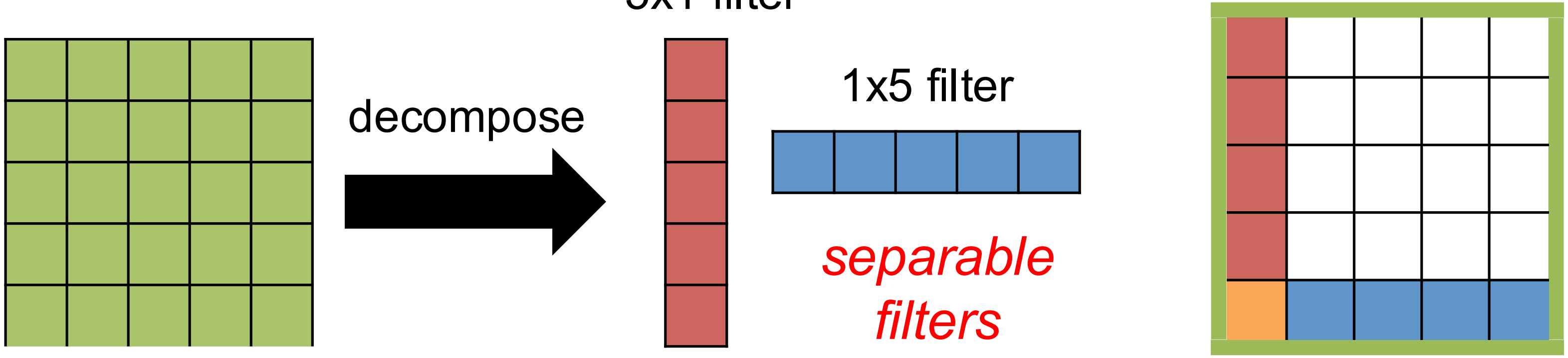
decompose

GoogleNet/Inception v3

5x5 filter

5x1 filter

Apply sequentially



decompose

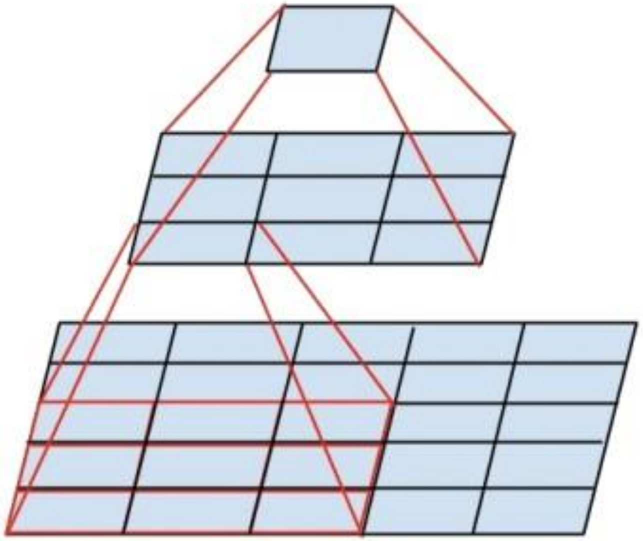
1x5 filter

separable filters

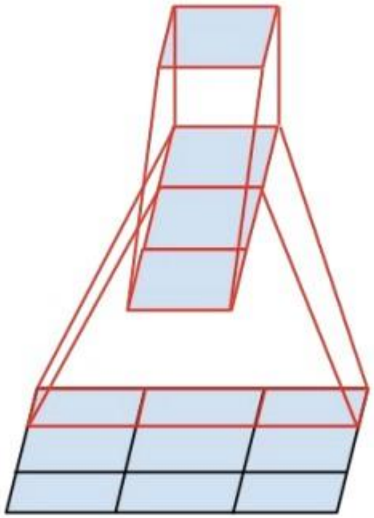
Example: Inception V3

Go deeper **(v1: 22 layers** à **v3: 40+ layers)** by reducing the number of weights per filter using **filter decomposition**
~3.5% higher accuracy than v1

5x5 filter à 3x3 filters



3x3 filter à 3x1 and 1x3 filters

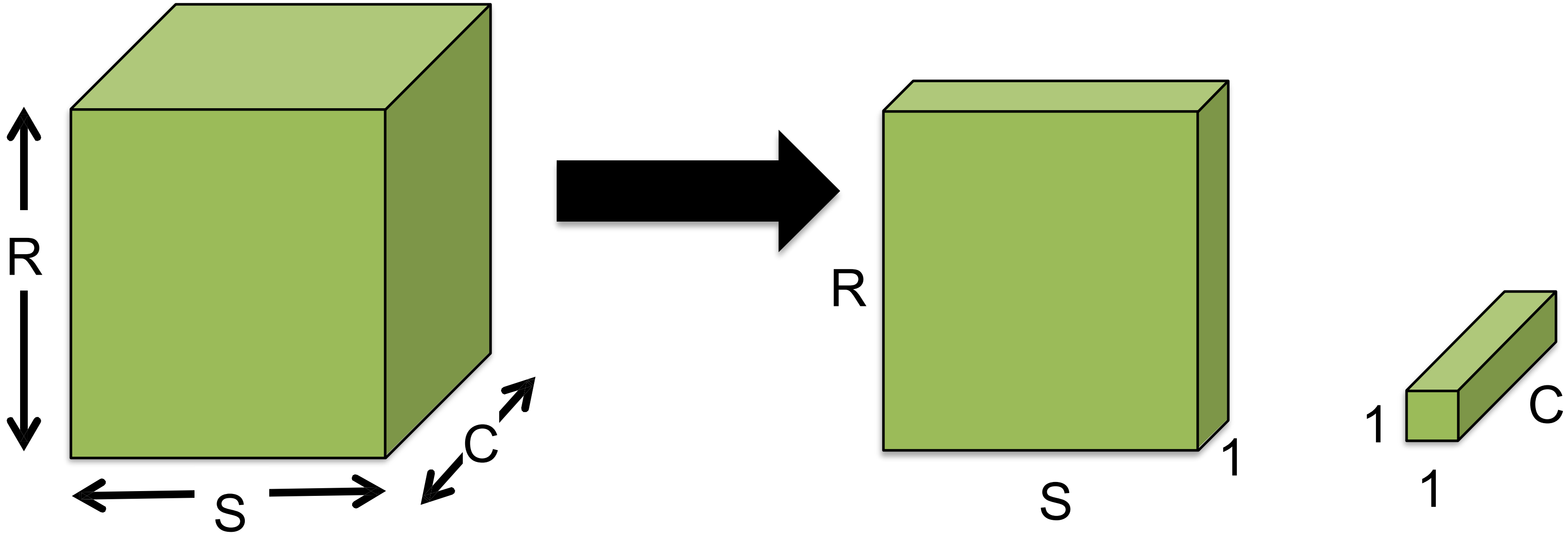


Separable filters

[Szegedy et al., arXiv 2015]

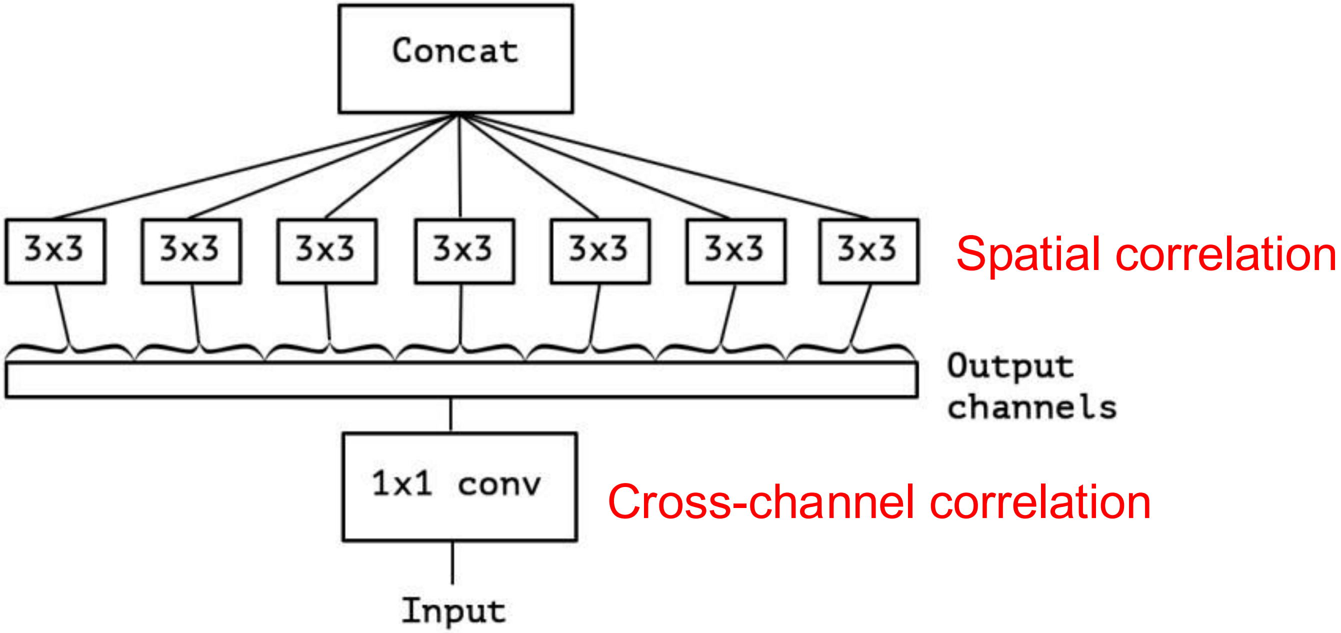
Depth-wise Separable

Decouple the **cross-channels correlations** and **spatial correlations** in the feature maps of the DNN



Example: Xception

- An Inception module based on depth-wise separable convolutions
- Claims to learn richer features with similar number of weights as Inception V3 (i.e. more efficient use of weights)
  - Similar performance on ImageNet; 4.3% better on larger dataset (JFT)
  - However, 1.5x more operations required than Inception V3



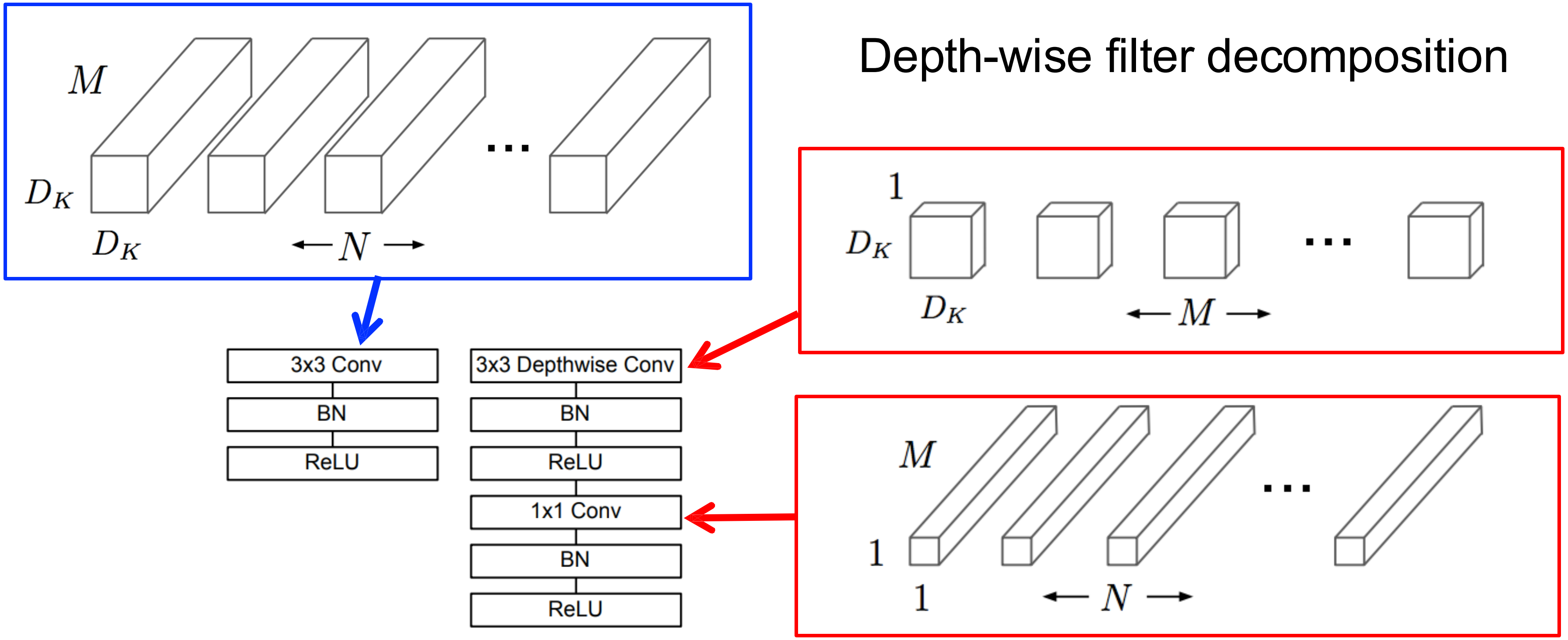
Spatial correlation

Cross-channel correlation

[Chollet, CVPR 2017]

Example: MobileNets

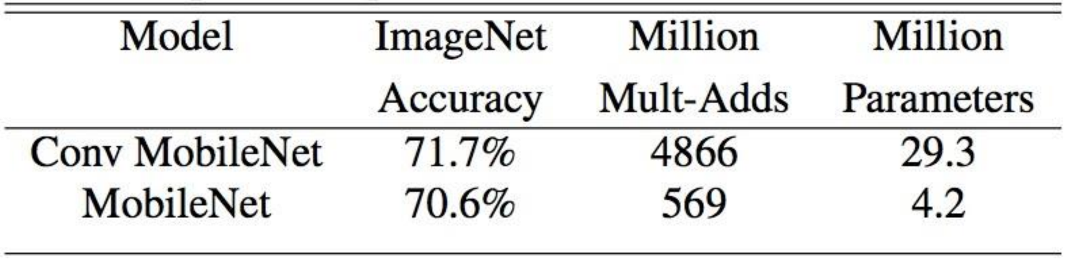
Depth-wise filter decomposition



3x3 Conv
BN
ReLU

3x3 Depthwise Conv
BN
ReLU
1x1 Conv
BN
ReLU

Table 4. Depthwise Separable vs Full Convolution MobileNet



Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

[Howard et al., arXiv, April 2017]

MobileNets: Comparison

Comparison with other DNN Models

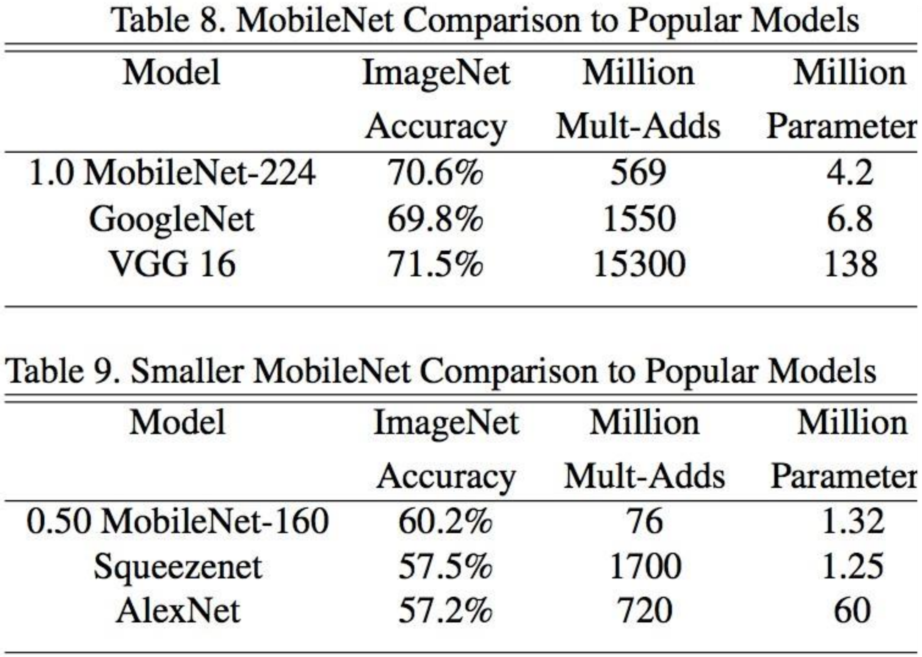
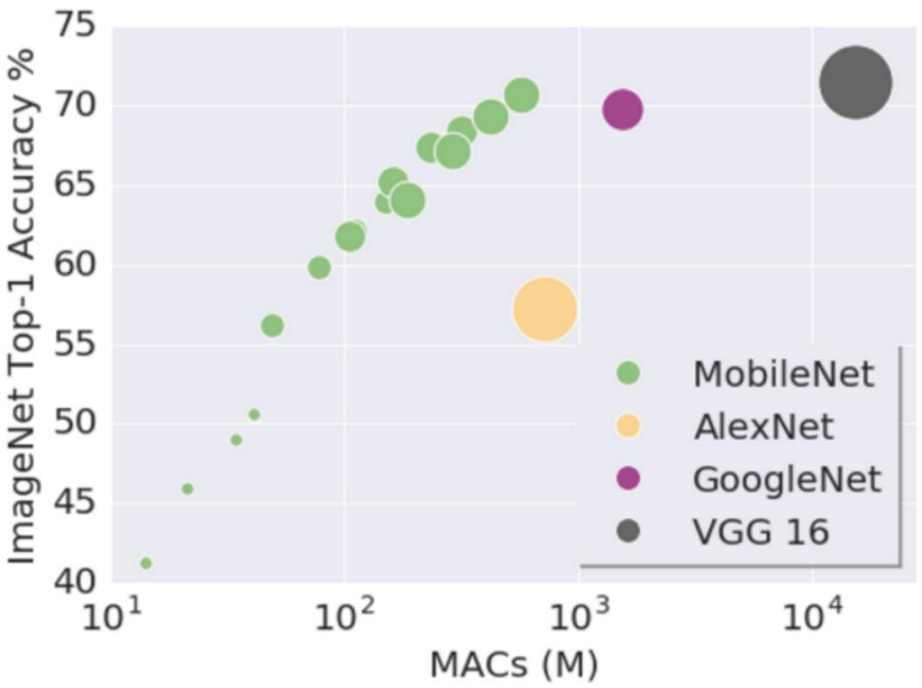


Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameter
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Table 9. Smaller MobileNet Comparison to Popular Models

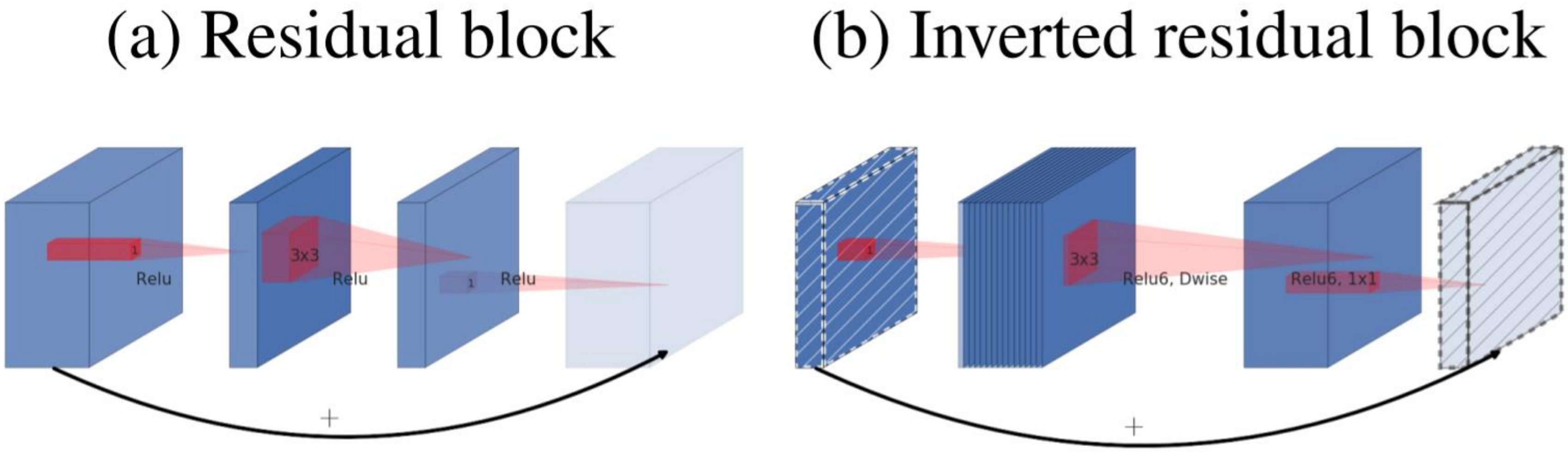
Model	ImageNet Accuracy	Million Mult-Adds	Million Parameter
0.50 MobileNet-160	60.2%	76	1.32
Squeezenet	57.5%	1700	1.25
AlexNet	57.2%	720	60



[Image source: Github]

[Howard et al., arXiv, April 2017]

MobileNetsV2: Comparison



Residual Block

- Conv 1x1 (Squeeze Channel)
- Conv 3x3
- Conv1x1 (Expand Channel)

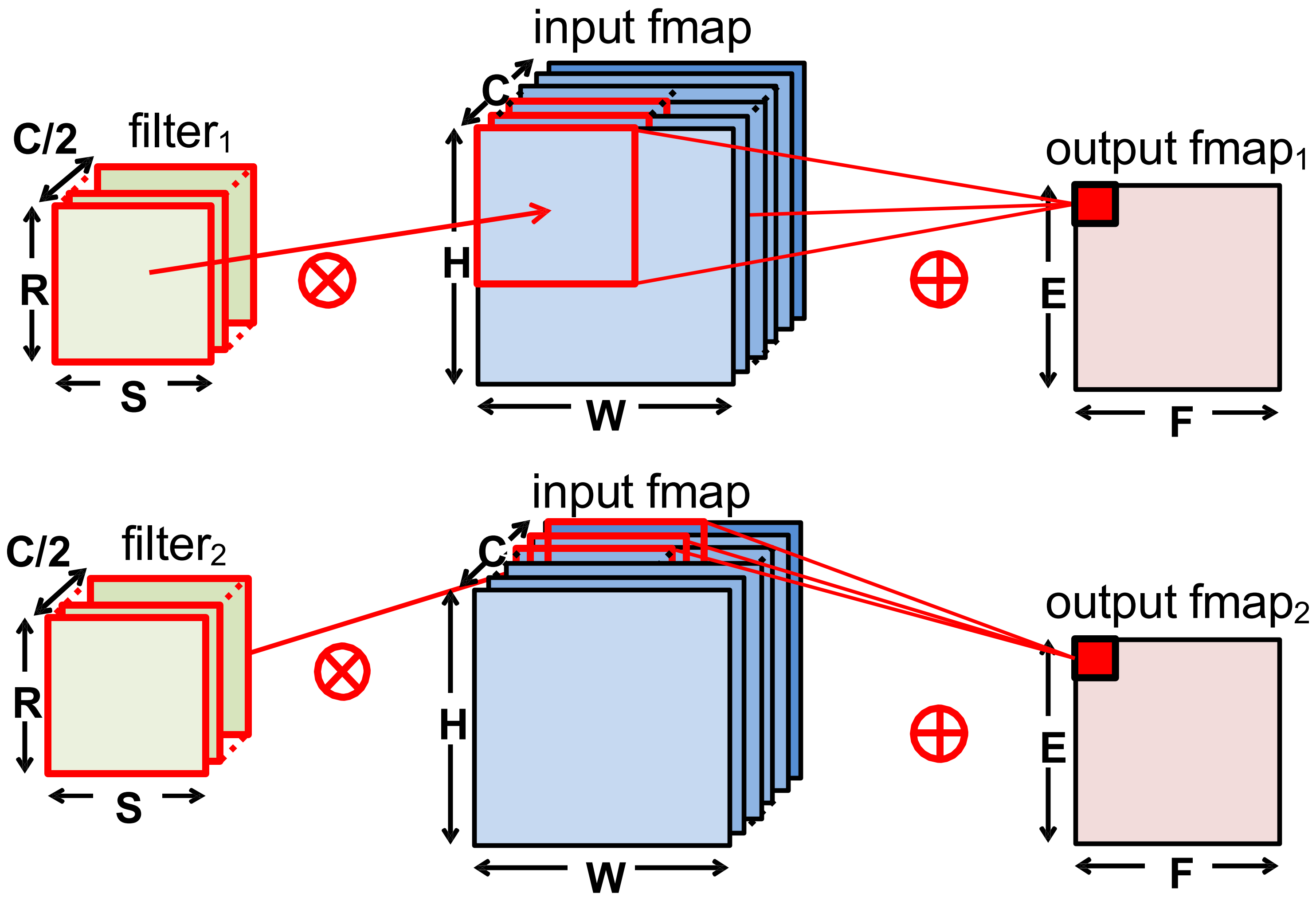
Inverted Residual Block

- Conv 1x1 (Expand Channel)
- Depthwise Conv 3x3
- Conv1x1 (Squeeze Channel)

[Sandler et al.,arxiv1801.14381

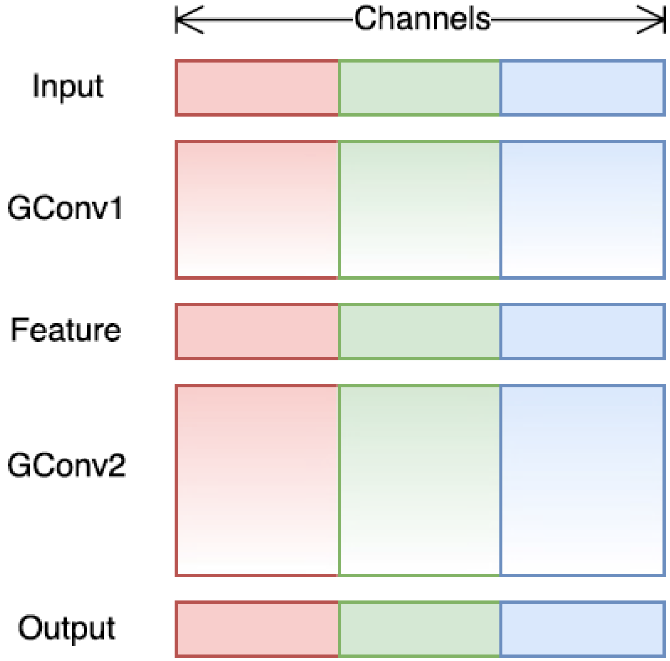
Grouped Convolutions

Grouped convolutions reduce the number of **weights and multiplications** at the cost of not sharing information between **groups**

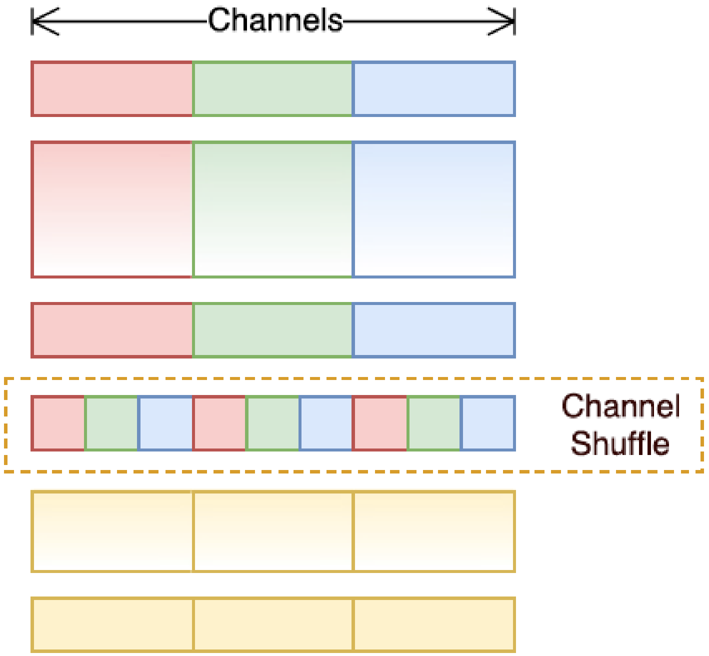


Example: ShuffleNet

Shuffle order such that channels are not isolated across groups
(up to 4% increase in accuracy)



No interaction between
channels from different groups



Shuffling allow interaction between
channels from different groups

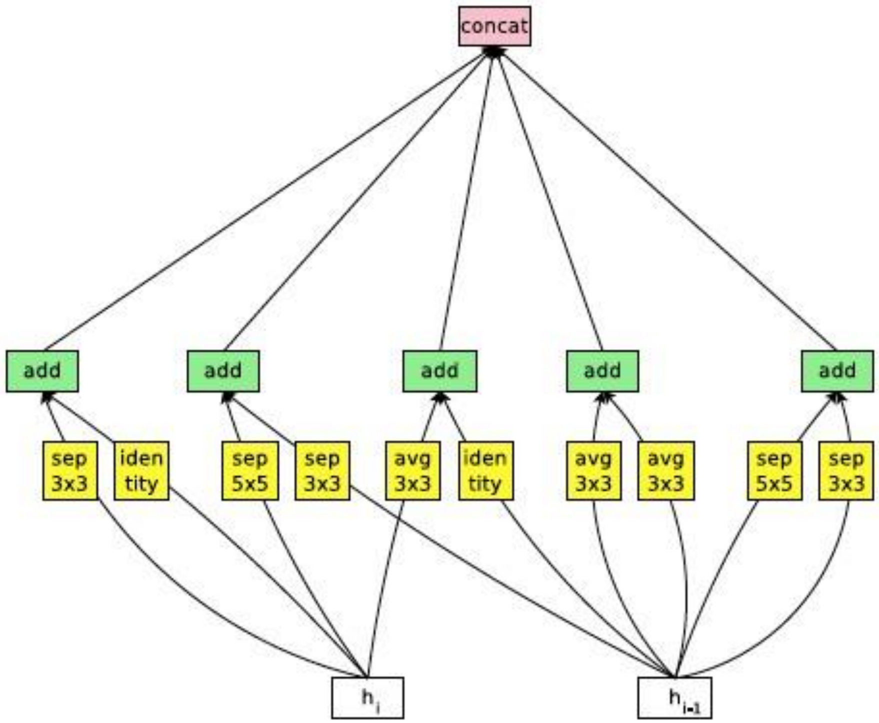
[Zhang et al., arXiv, July 2017]

Learn DNN Models

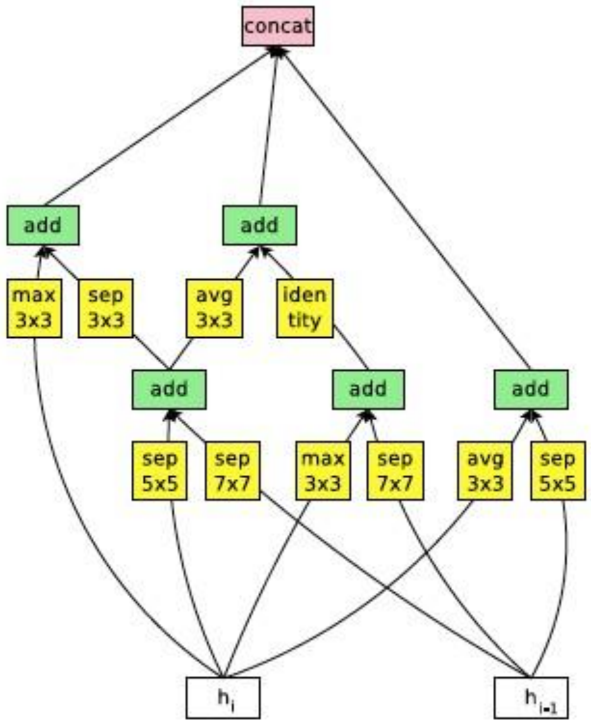
- Rather than handcrafting the model, learn the model
- More recent result uses Neural Architecture Search
- Build model from popular layers
    - Identity
    - 1x3 then 3x1 convolution
    - 1x7 then 7x1 convolution
    - 3x3 dilated convolution
    - 1x1 convolution
    - 3x3 convolution
    - 3x3 separable convolution
    - 5x5 separable convolution
    - 3x3 average pooling
    - 3x3 max pooling
    - 5x5 max pooling
    - 7x7 max pooling

[Zoph et al., arXiv, July 2017]

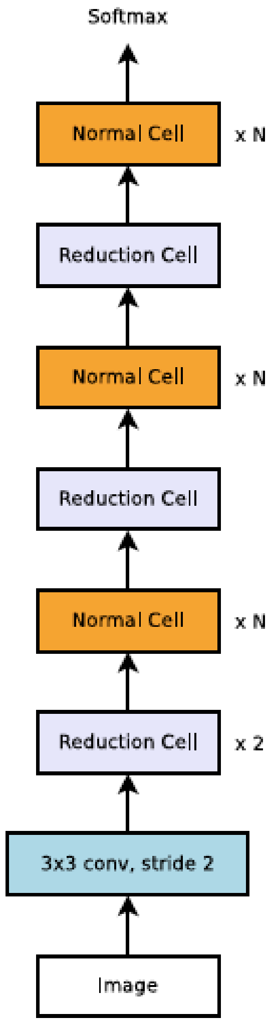
Learned Convolutional Cells



Normal Cell



Reduction Cell

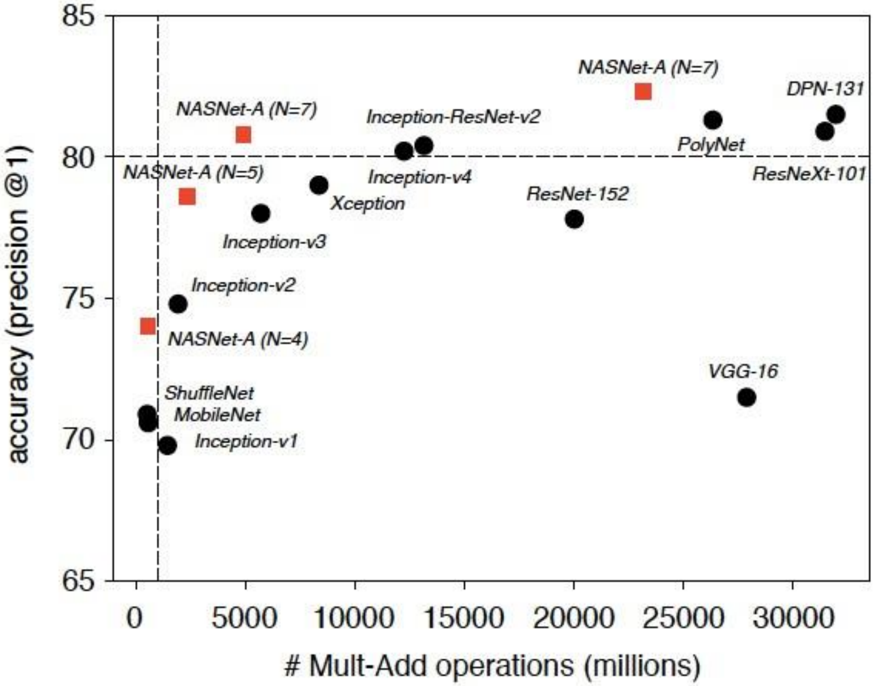


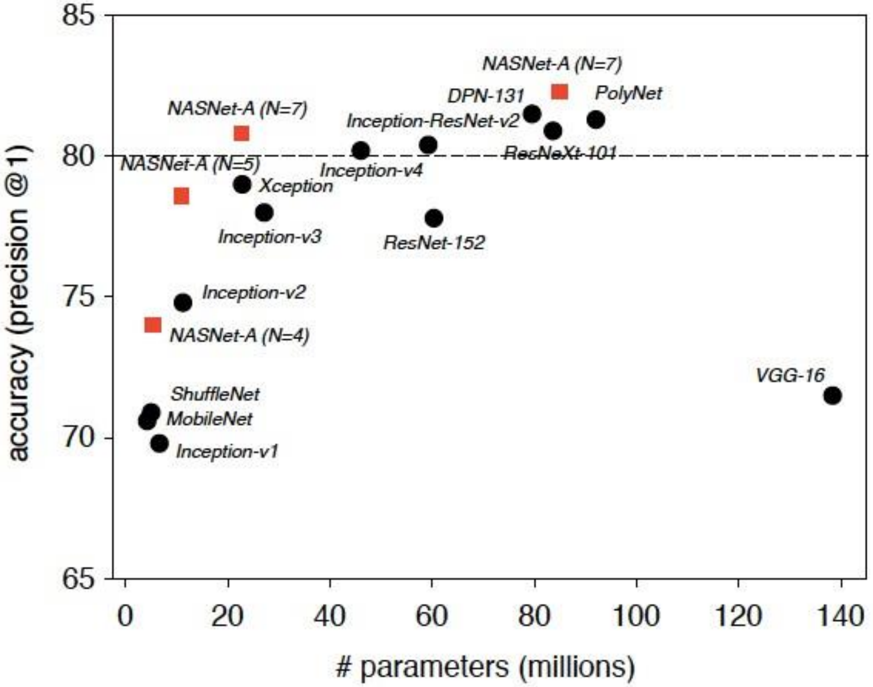
ImageNet Architecture

[Zoph et al., arXiv, July 2017]

Comparison with Existing Networks

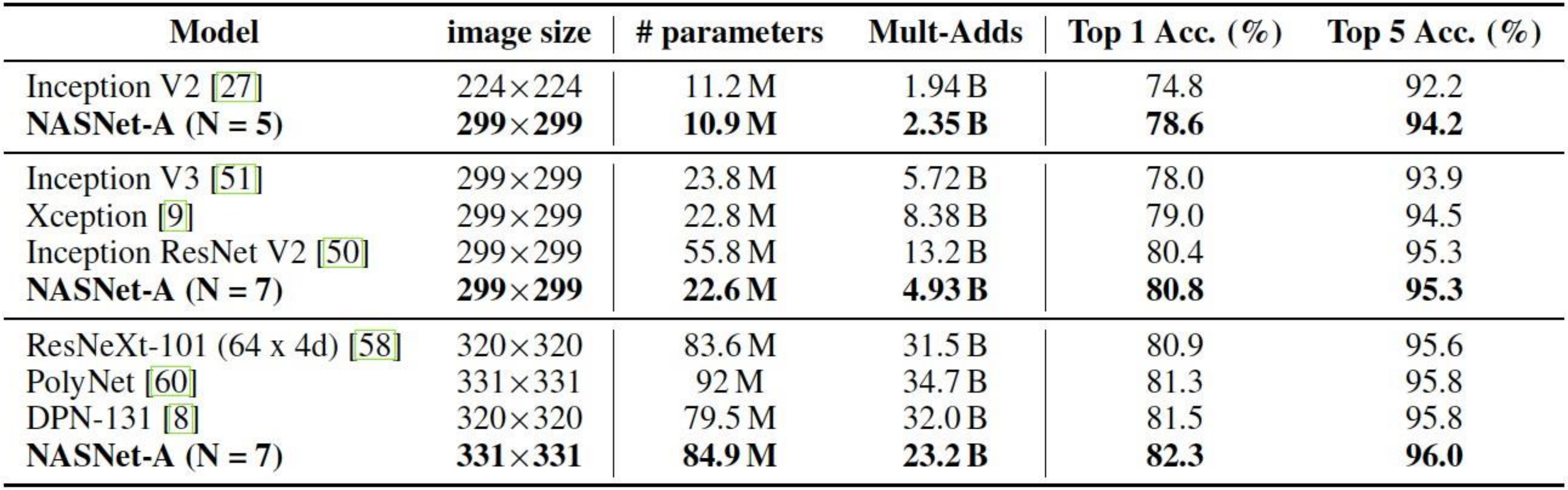
Learned models have improved accuracy vs. ‘complexity’ tradeoff compared to handcrafted models



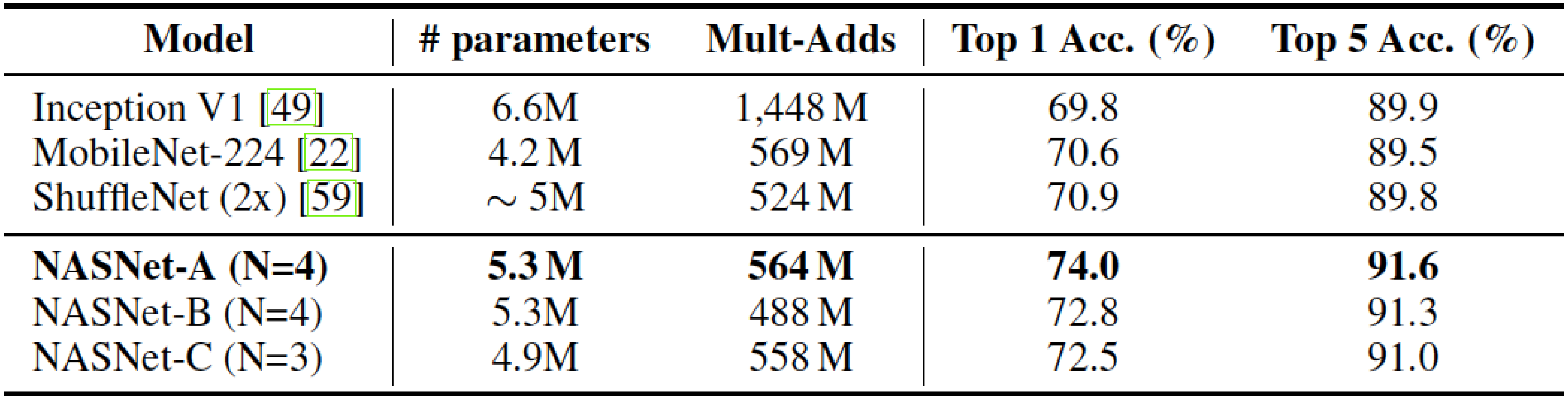


[Zoph et al., arXiv, July 2017]

Comparison with Existing Networks



Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [27]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (N = 5)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [51]	299×299	23.8 M	5.72 B	78.0	93.9
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [50]	299×299	55.8 M	13.2 B	80.4	95.3
NASNet-A (N = 7)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [58]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [60]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
NASNet-A (N = 7)	**331×331**	**84.9 M**	**23.2 B**	**82.3**	**96.0**



Model	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V1 [49]	6.6M	1,448 M	69.8	89.9
MobileNet-224 [22]	4.2 M	569 M	70.6	89.5
ShuffleNet (2x) [59]	∼ 5M	524 M	70.9	89.8
NASNet-A (N=4)	**5.3 M**	**564 M**	**74.0**	**91.6**
NASNet-B (N=4)	5.3M	488 M	72.8	91.3
NASNet-C (N=3)	4.9M	558 M	72.5	91.0

[Zoph et al., arXiv, July 2017]

Summary

- Approaches used to improve accuracy by popular DNN models in the ImageNet Challenge
  - Go deeper (i.e. more layers)
  - Stack smaller filters and apply 1x1 bottlenecks to reduce number of weights such that the deeper models can fit into a GPU (faster training)
  - Use multiple connections across layers (e.g. parallel and short cut)
- Efficient models aim to reduce number of weights and number of operations
  - Most use some form of filter decomposition (spatial, depth and channel)
  - Note: Number of weights and operations does not directly map to storage, speed and power/energy. Depends on hardware!
- Filter shapes vary across layers and models
  - Need flexible hardware!