Paper Reading

周争光

Paper

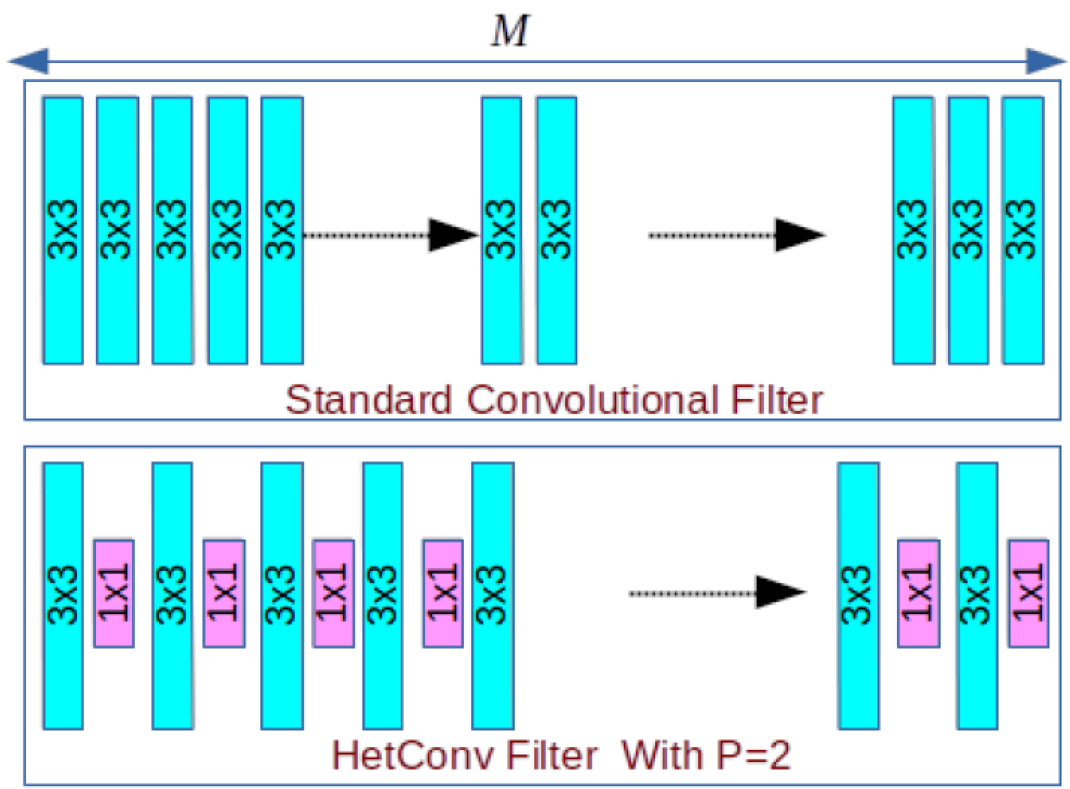
- HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs, CVPR, 2019
- Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Paper

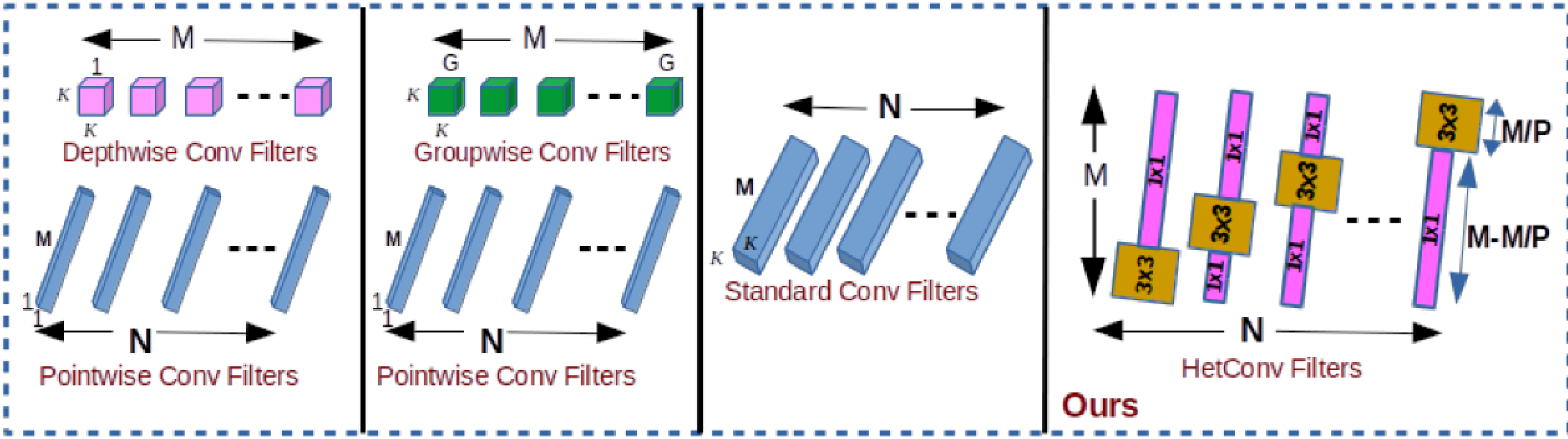
- HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs, CVPR, 2019
- Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ICML, 2019

HetConv

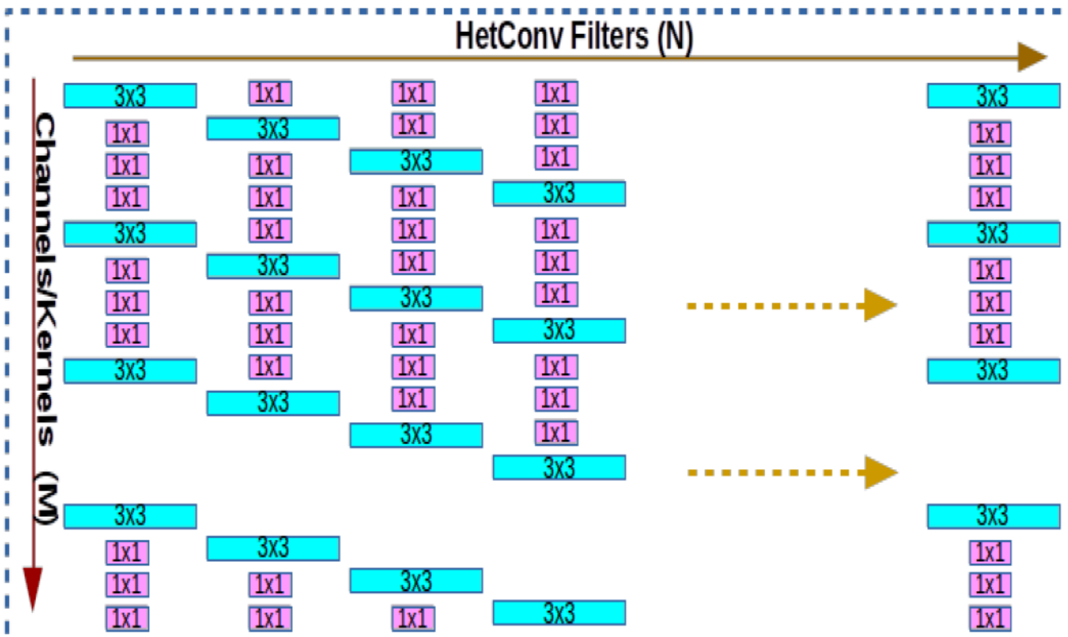
- Reduce the FLOPs of the given model/architecture by designing new kernels
- Homogeneous: each kernel is of the same size
- Heterogeneous: contains different sizes of kernels



HetConv

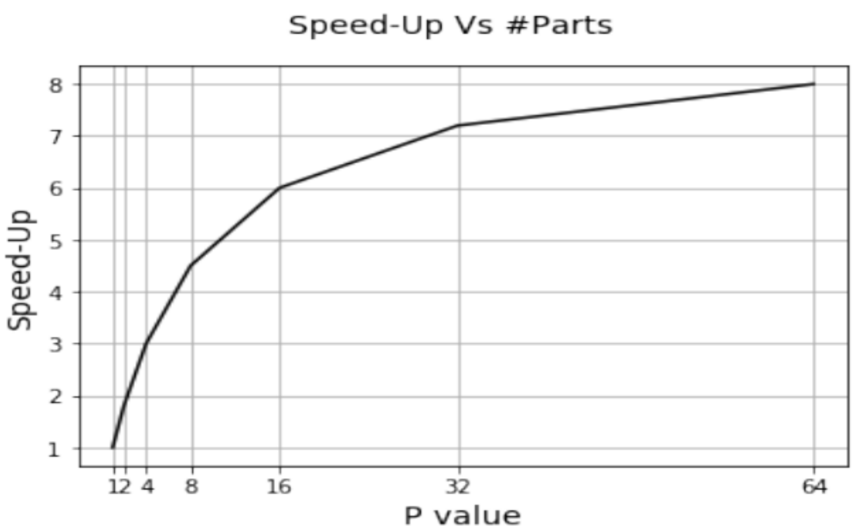


- Filters



HetConv

- Standard conv: $FL_S = D_o \times D_o \times M \times N \times K \times K$
- HetConv with part P:
  - KxK: $FL_K = (D_o \times D_o \times M \times N \times K \times K)/P$
  - 1x1 $FL_1 = (D_o \times D_o \times N) \times \left(M - \frac{M}{P}\right)$
- Total reduction: $R_{HetConv} = \frac{FL_K + FL_1}{FL_S}$
- Speed-up $= \frac{1}{P} + \frac{(1 - 1/P)}{K^2}$



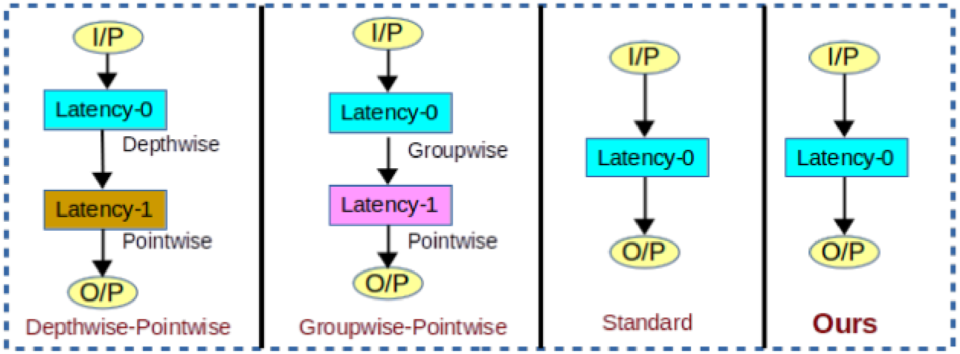


Figure 4. The figure shows the comparison with the different types of convolution in terms of latency.

HetConv

- VGG-16 on CIFAR10

Model	Acc%	FLOPs	FLOPs Reduced (%)	Parameters	Parameters Reduced (%)
VGG-16_P1	94.06	313.74M	–	15.00M	–
VGG-16_P1_SE	94.13	314.19M	–	15.22M	–
VGG-16_P2	93.89	175.23M	44.15	8.45M	43.68
VGG-16_P2_SE	94.11	175.67M	44.00	8.68M	42.99
VGG-16_P4	93.93	105.98M	66.22	5.17M	65.45
VGG-16_P4_SE	94.29	106.42M	66.08	5.41M	64.48
VGG-16_GWC4_PWC	92.76	107.67M	65.68	5.42M	–
VGG-16_P8	93.92	71.35M	77.26	3.54M	76.40
VGG-16_P8_SE	93.97	71.79M	77.12	3.77M	75.22
VGG-16_P16	93.96	54.04M	82.78	2.72M	81.86
VGG-16_P16_SE	93.63	54.48M	82.64	2.95M	80.59
VGG-16_P32	93.73	45.38M	85.54	2.31M	84.58
VGG-16_P32_SE	93.41	45.82M	85.39	2.54M	83.28
VGG-16_P64	93.42	41.05M	86.92	2.11M	85.95
VGG-16_P64_SE	93.33	41.49M	86.77	2.34M	84.63
VGG-16_DWC_PWC	91.27	38.53M	87.72	1.97M	–
VGG-16_PC	92.53	38.18M	87.83	1.93M	–
VGG-16_PC_SE	93.08	38.62M	87.69	2.15M	–

Table 1. The table shows the detail results for VGG-16 on CIFAR-10 in different setups.

HetConv

ImageNet

Method	Acc%(Top-1)	Acc%(Top-5)	FLOPs Reduced %
RNP (3X)[22]	–	87.57	66.67
ThiNet-70 [24]	69.8	89.53	69.04
CP 2X [11]	–	89.90	50.00
VGG-16_P1	71.3	90.2	–
VGG-16_P4	**71.2**	**90.2**	**65.8**

Table 5. Table shows the results for the VGG-16 on ImageNet [29]. Our model has no loss in accuracy as compare to state-of-art [11, 24] pruning approaches while significantly higher FLOPs reduction.

Method	Error (top-1)%	FLOPs	FLOPs Reduced(%)
ThiNet-70 [24]	27.90	–	36.8
NISP [41]	27.33	–	27.31
ResNet-50_P1	23.86	4.09G	–
ResNet-50_P4	**23.84**	2.85G	**30.32**

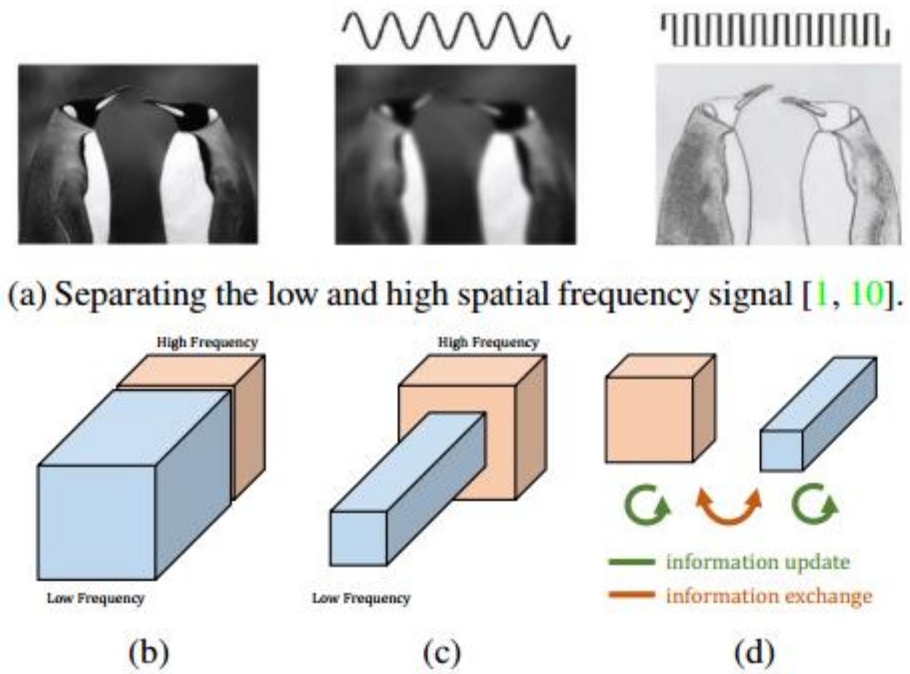
Table 7. Table shows the results for ResNet-50 on ImageNet [29]. Our model has no loss in accuracy as compare to state-of-art [24, 41] flop pruning approaches.

Paper

- HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs, CVPR, 2019
- Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ICML, 2019

OctConv

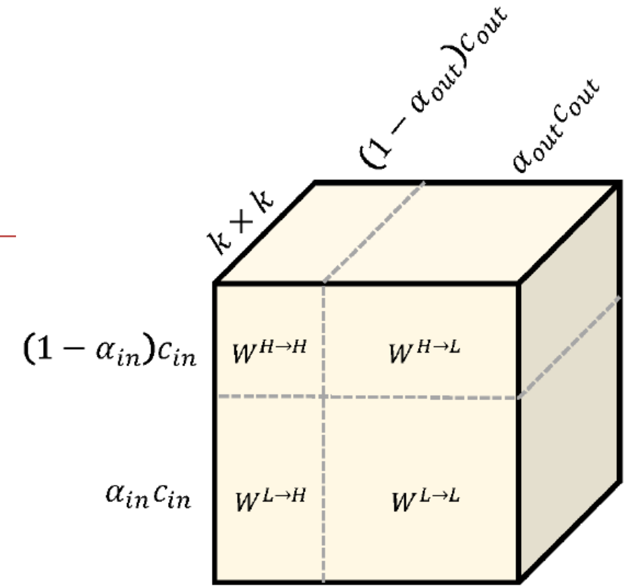
- The output maps of a convolutional layer can also be factorized and grouped by their spatial frequency.
- OctConv focuses on reducing the spatial redundancy in CNNs and is designed to replace vanilla convolution operations.

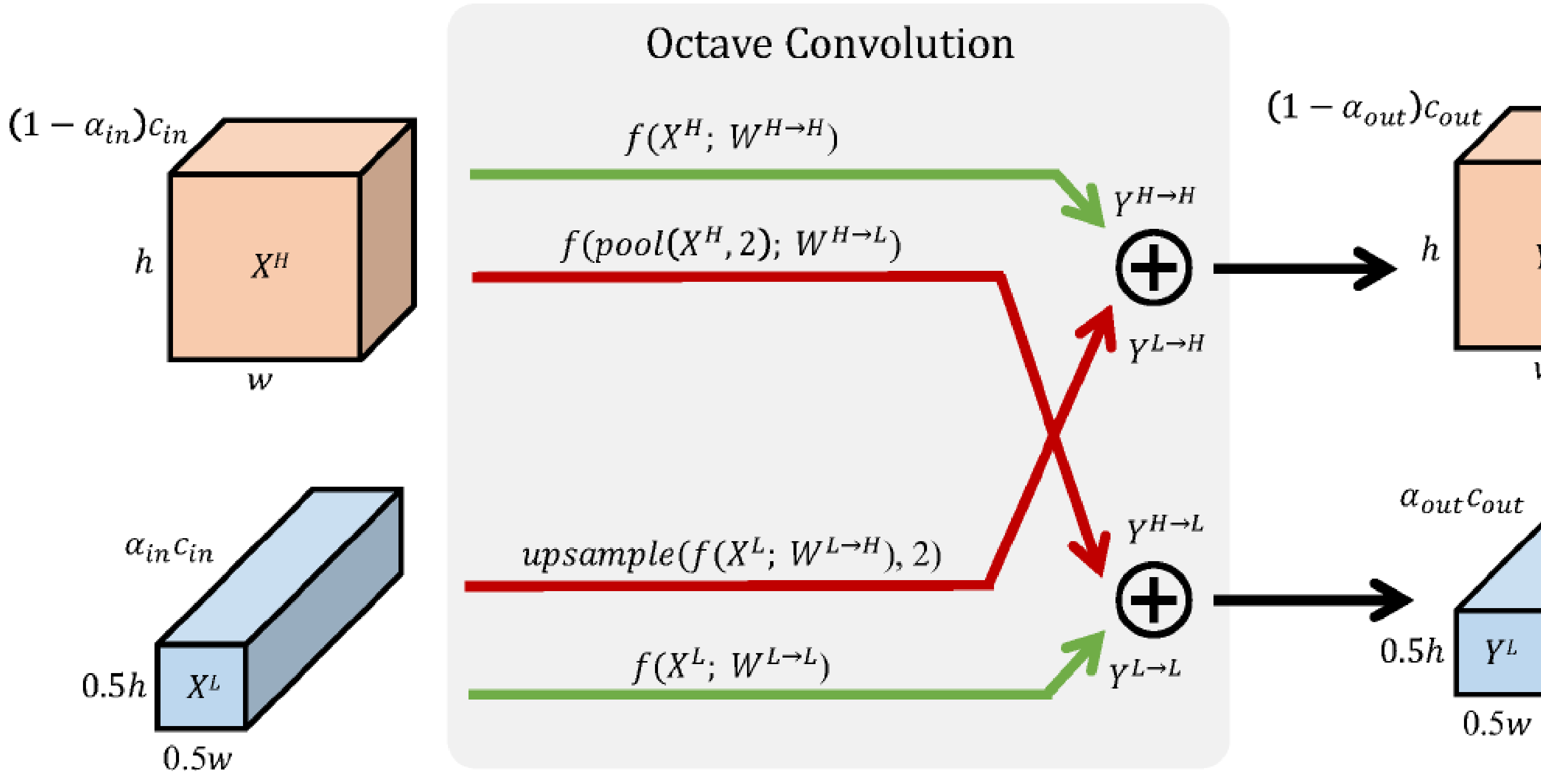


OctConv

- Implementation Details

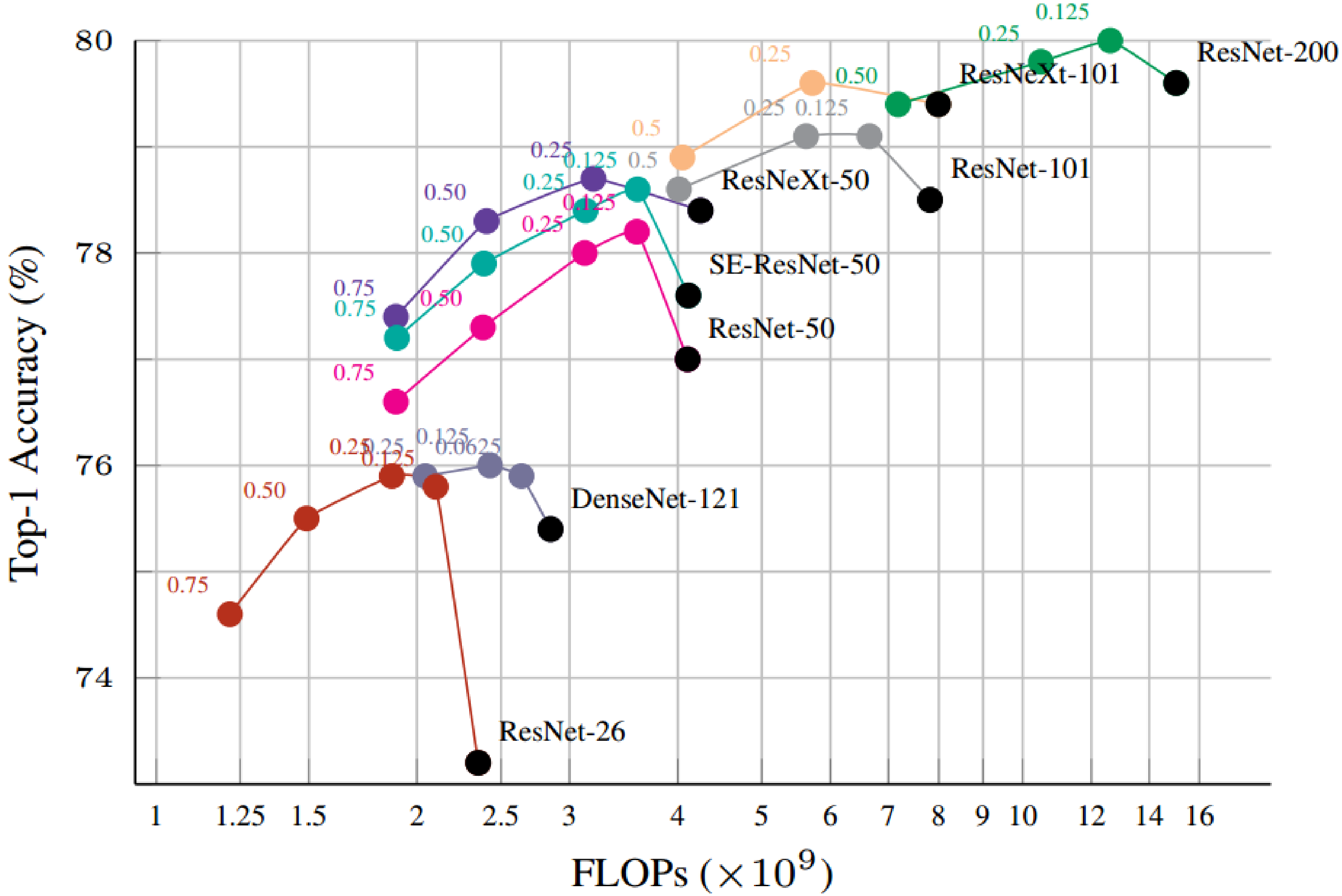
$$Y^H = f(X^H; W^{H\to H}) + \mathtt{upsample}(f(X^L; W^{L\to H}), 2),$$
$$Y^L = f(X^L; W^{L\to L}) + f(\mathtt{pool}(X^H, 2); W^{H\to L})),$$





OctConv

- ImageNet



OctConv

ImageNet

ratio (α)	Testing Scale (*small* $\rightarrow$ *large*)							
	256	320	384	448	512	576	640	740
N/A	77.2	78.6	78.7	78.7	78.3	77.6	76.7	75.8
.5	**+0.7**	**+0.7**	**+0.9**	**+0.9**	**+0.8**	**+1.0**	**+1.1**	**+1.2**

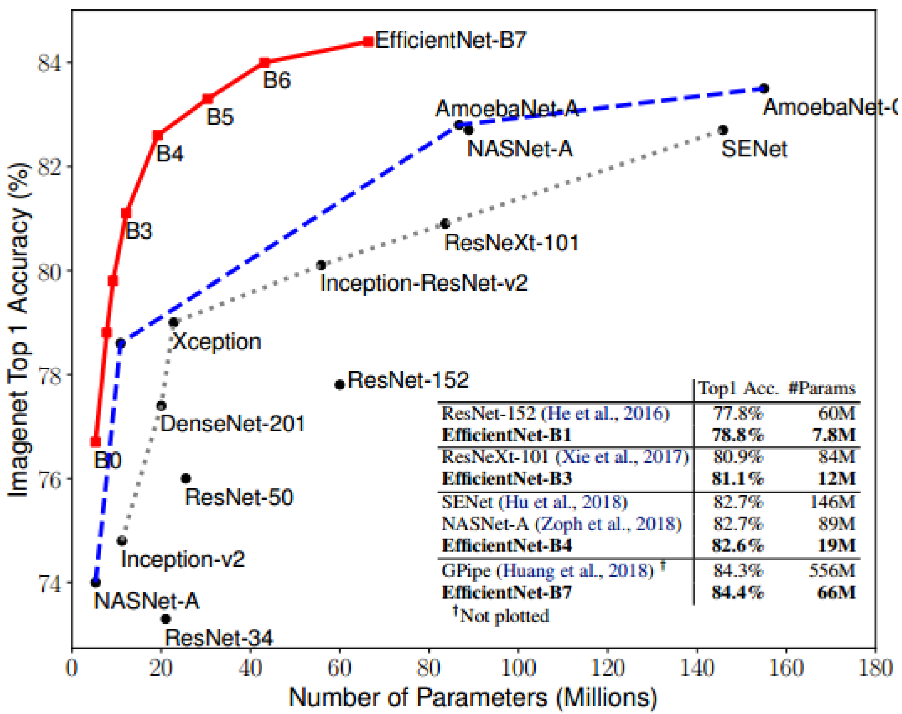
Method	ratio (α)	#Params (M)	#FLOPs (M)	CPU (ms)	Top-1 (%)
CondenseNet ($G = C = 8$) [21]	-	2.9	274	-	71.0
1.5 ShuffleNet (v1) [50]	-	3.4	292	-	71.5
1.5 ShuffleNet (v2) [32]	-	3.5	299	-	**72.6**
0.75 MobileNet (v1) [18]	-	2.6	325	13.4	70.3*
0.75 Oct-MobileNet (v1) (ours)	.375	2.6	**213**	**11.9**	70.6
1.0 Oct-MobileNet (v1) (ours)	.5	4.2	321	18.4	**72.4**
1.0 MobileNet (v2) [34]	-	3.5	300	24.5	72.0
1.0 Oct-MobileNet (v2) (ours)	.375	3.5	**256**	**17.1**	72.0
1.125 Oct-MobileNet (v2) (ours)	.5	4.2	295	26.3	**73.0**

Paper

- HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs, CVPR, 2019
- Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ICML, 2019

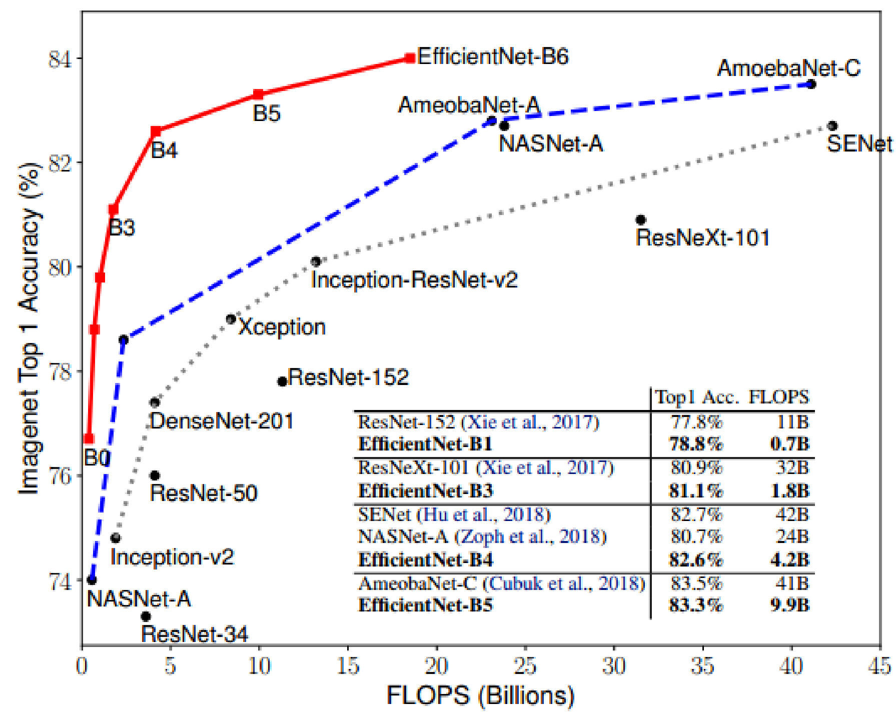
EfficientNet

- ☐ Uniformly scales depth/width/resolution.
- ☐ New SOTA 84.4% top-1 accuracy.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted



	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**78.8%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.1%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**82.6%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2018)	83.5%	41B
EfficientNet-B5	**83.3%**	**9.9B**

EfficientNet

- Compound scaling method

$$\mathcal{N} = \bigodot_{i=1\ldots s} \mathcal{F}_i^{L_i}\left(X_{\langle H_i,W_i,C_i\rangle}\right)$$

$$\max_{d,w,r} \quad Accuracy\big(\mathcal{N}(d,w,r)\big)$$

$$s.t. \quad \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\left(X_{\langle r\cdot\hat{H}_i,r\cdot\hat{W}_i,w\cdot\hat{C}_i\rangle}\right)$$

$$\text{Memory}(\mathcal{N}) \leq \text{target\_memory}$$

$$\text{FLOPS}(\mathcal{N}) \leq \text{target\_flops}$$

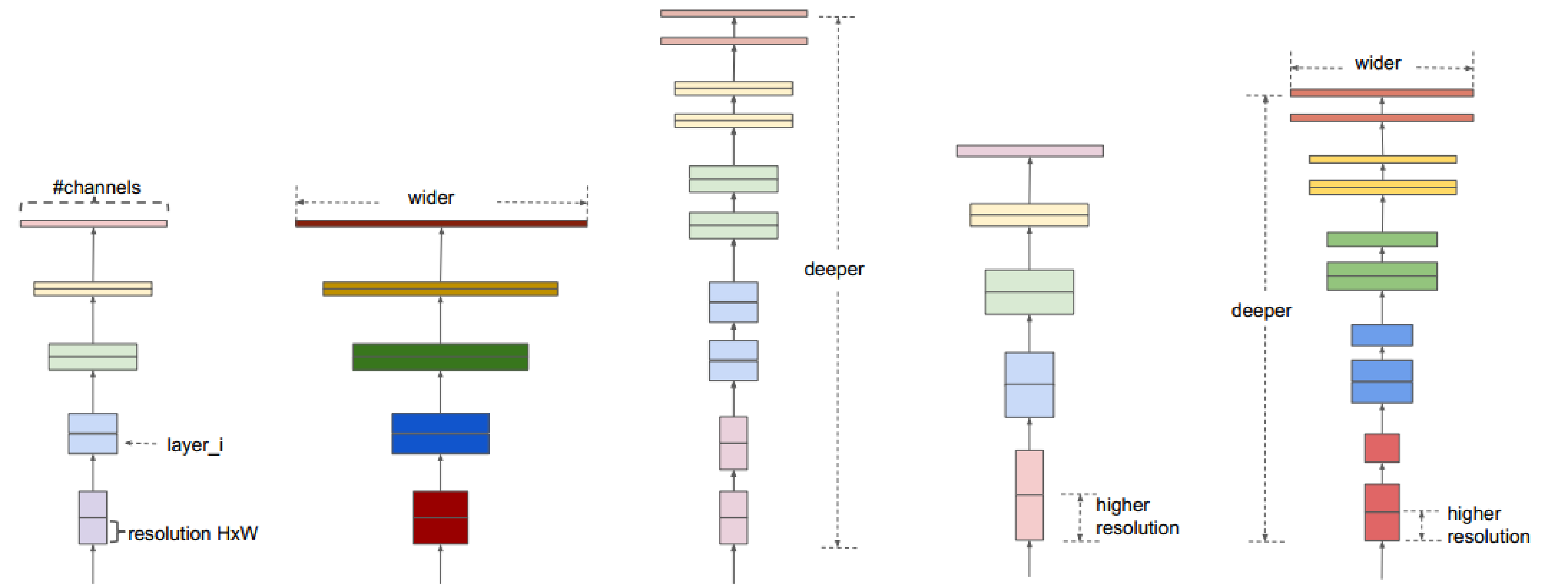
$$\text{depth: } d = \alpha^{\phi}$$

$$\text{width: } w = \beta^{\phi}$$

$$\text{resolution: } r = \gamma^{\phi}$$

$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

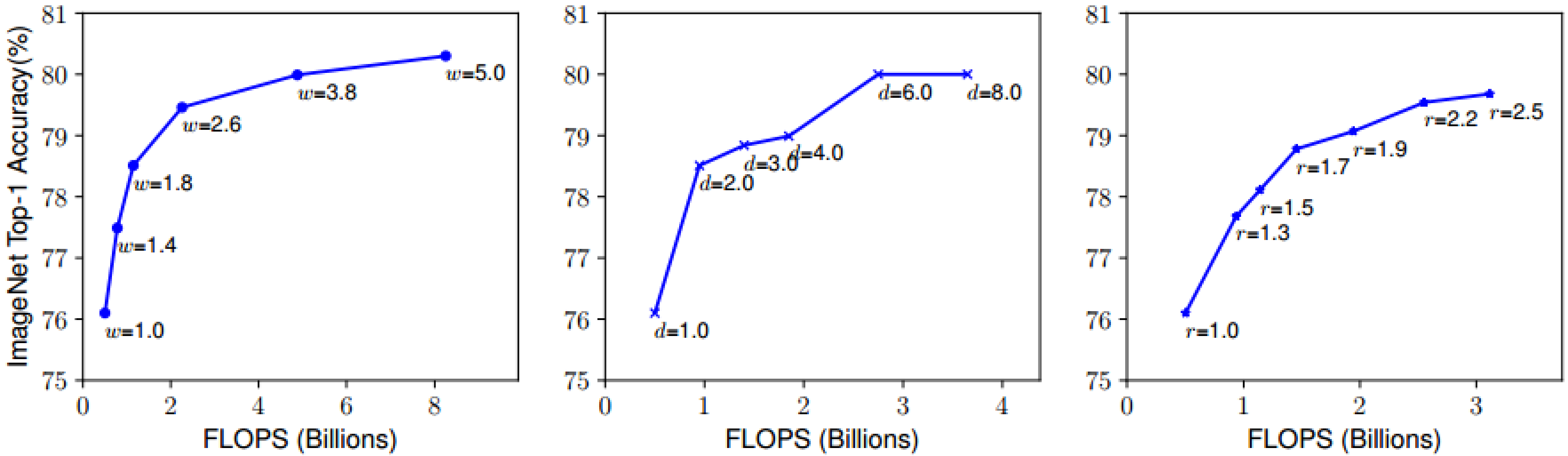
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$



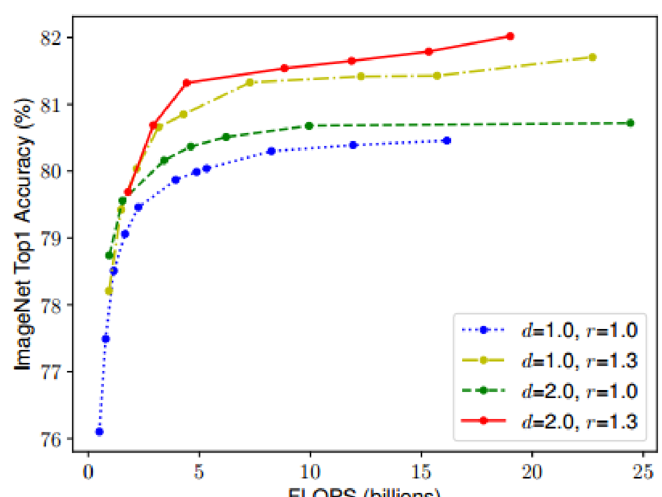
(a) baseline network
(b) width scaling (increase #channels)
(c) depth scaling (increase #layers)
(d) resolution scaling (increase #image size)
(e) compound scaling (increase width/depth/resolution uniformly)

EfficientNet

- Single dimension scaling



- Scaling Network Width for Different Baseline



EfficientNet

- Scaling Up MobileNets and ResNets

Table 3. **Scaling Up MobileNets and ResNet.**

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1 (M1)	0.6B	70.6%
Scale M1 width (w=2)	2.2B	74.2%
Scale M1 resolution (r=2)	2.2B	72.7%
M1 compound scale (d=1.4, w=1.2, r=1.3)	**2.3B**	**75.6%**
Baseline MobileNetV2	0.3B	72.0%
Scale MobileNetV2 depth (d=4)	1.2B	76.8%
Scale MobileNetV2 width (w=2)	1.1B	76.4%
Scale MobileNetV2 resolution (r=2)	1.2B	74.8%
MobileNetV2 compound scale	**1.3B**	**77.4%**
Baseline ResNet-50	4.1B	76.0%
Scale ResNet-50 depth (d=4)	16.2B	78.1%
Scale ResNet-50 width (w=2)	14.7B	77.7%
Scale ResNet-50 resolution (r=2)	16.4B	77.5%
ResNet-50 compound scale	**16.7B**	**78.8%**

EfficientNet

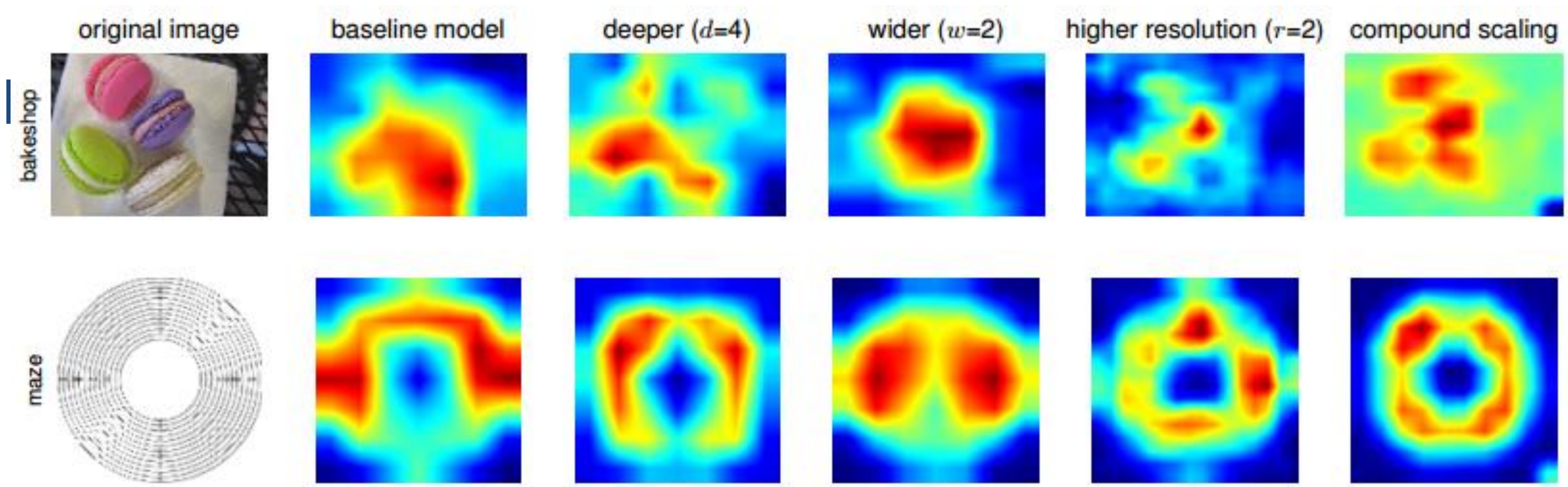
Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**76.3%**	**93.2%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**78.8%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**79.8%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.1%**	**95.5%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.6%**	**96.3%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.3%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2018)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.9%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.4%**	**97.1%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

EfficientNet

- Results on Transfer Learning Datasets
  - achieve new state-of-the-art accuracy for 5 out of 8 datasets

	Comparison to best public-available results						Comparison to best reported results					
	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)
CIFAR-10	NASNet-A	98.0%	85M	EfficientNet-B0	98.1%	4M (21x)	†Gpipe	**99.0%**	556M	EfficientNet-B7	98.9%	64M (8.7x)
CIFAR-100	NASNet-A	87.5%	85M	EfficientNet-B0	88.1%	4M (21x)	Gpipe	91.3%	556M	EfficientNet-B7	**91.7%**	64M (8.7x)
Birdsnap	Inception-v4	81.8%	41M	EfficientNet-B5	82.0%	28M (1.5x)	GPipe	83.6%	556M	EfficientNet-B7	**84.3%**	64M (8.7x)
Stanford Cars	Inception-v4	93.4%	41M	EfficientNet-B3	93.6%	10M (4.1x)	‡DAT	**94.8%**	-	EfficientNet-B7	94.7%	-
Flowers	Inception-v4	98.5%	41M	EfficientNet-B5	98.5%	28M (1.5x)	DAT	97.7%	-	EfficientNet-B7	**98.8%**	-
FGVC Aircraft	Inception-v4	90.9%	41M	EfficientNet-B3	90.7%	10M (4.1x)	DAT	92.9%	-	EfficientNet-B7	**92.9%**	-
Oxford-IIIT Pets	ResNet-152	94.5%	58M	EfficientNet-B4	94.8%	17M (5.6x)	GPipe	**95.9%**	556M	EfficientNet-B6	95.4%	41M (14x)
Food-101	Inception-v4	90.8%	41M	EfficientNet-B4	91.5%	17M (2.4x)	GPipe	93.0%	556M	EfficientNet-B7	**93.0%**	64M (8.7x)
Geo-Mean						**(4.7x)**						**(9.6x)**



Thanks!