EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

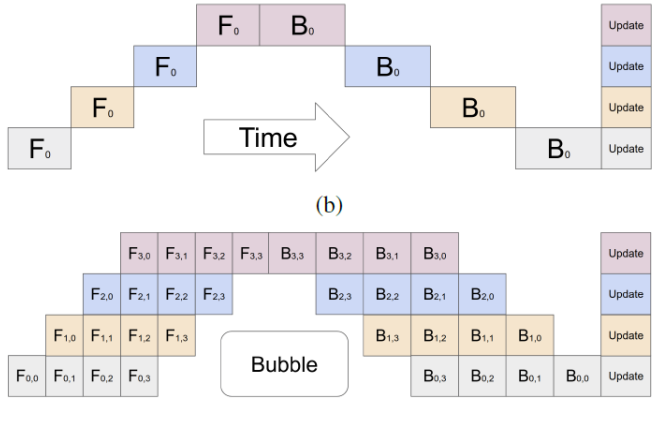
Mingxing Tan [1] Quoc V. Le [1]

김 성 철

Contents

Introduction

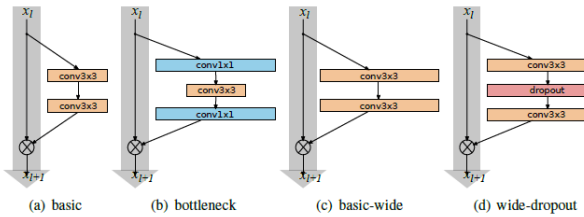
- Scaling up ConvNets은 요즘 대세
  - ResNet : ResNet-18 → ResNet-200
  - GPipe : scaling up a baseline model four time larger
- Scaling up하는 방법도 여러가지
  - Depth (ResNet) / Width (Wide Residual Networks)
  - Image resolution (GPipe: 480x480)



(b)

(c)

GPipe



(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

Figure 1: Various residual blocks used in the paper. Batch normalization and ReLU precede each convolution (omitted for clarity)

Wide Residual Networks

Introduction

- Rethink the process of scaling up ConvNets
  - A principled method to scale up ConvNets for accuracy and efficiency
    - Network의 width / depth / resolution의 균형이 매우 중요
    - 단순히 상수 비로 scaling 한 경우 균형이 맞춰지기도 함
  - **Compound scaling method**
    - **Uniformly scale network width, depth, and resolution**
    - **2^N배 많은 resource 사용**

      → depth α^N, width β^N, image size γ^N

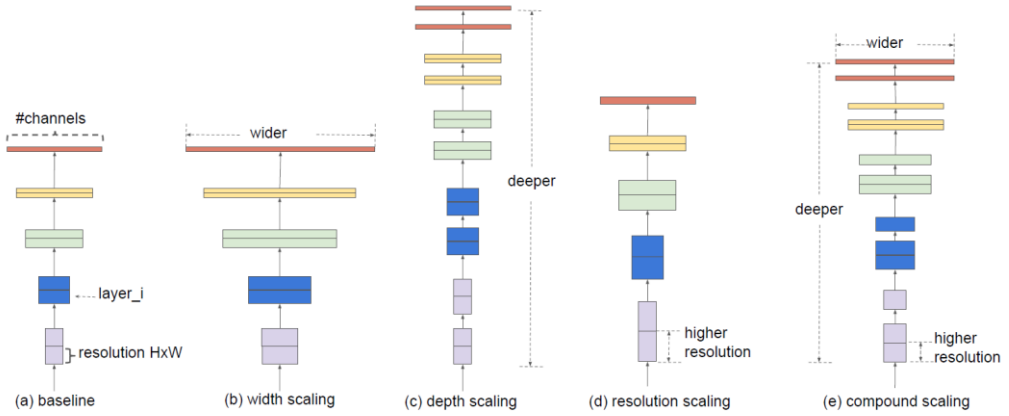


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Introduction

- Model scaling의 effectiveness는 baseline network의 영향을 많이 받음
  - 좋은 baseline network를 만들기 위해 Neural Network Search(NAS)를 사용
  - **Compound scaling method**를 사용해 **EfficientNet** 시리즈를 만듦

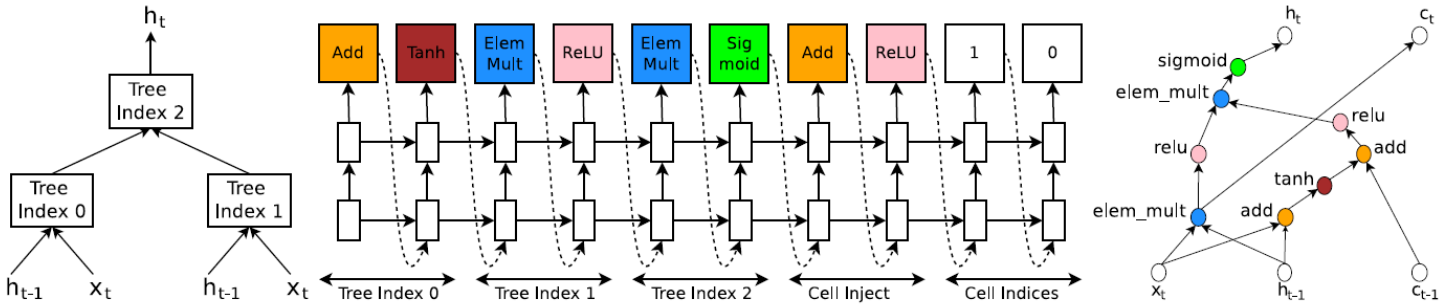


Figure 5: An example of a recurrent cell constructed from a tree that has two leaf nodes (base 2) and one internal node. Left: the tree that defines the computation steps to be predicted by controller. Center: an example set of predictions made by the controller for each computation step in the tree. Right: the computation graph of the recurrent cell constructed from example predictions of the controller.

Neural Architecture Search(NAS)

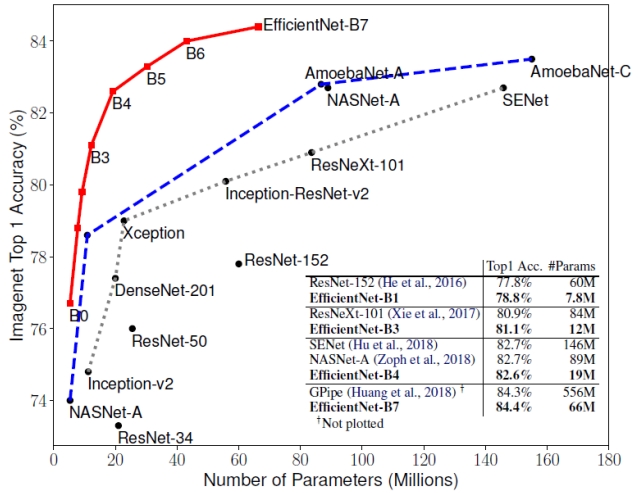


Figure 1. **Model Size vs. ImageNet Accuracy.** All numbers are for single-crop, single-model. Our EfficientNets significantly outperform other ConvNets. In particular, EfficientNet-B7 achieves new state-of-the-art 84.4% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

Related Work

- **ConvNet Accuracy**
  - AlexNet, GoogleNet, SENet, Gpipe를 거치며 네트워크는 점점 커짐
  - ImageNet 뿐만 아니라 다른 데이터셋이나 task에도 적용되어야 함
  - **Hardware memory limit 때문에 효율적으로 accuracy를 얻어야 함**
- **ConvNet Efficiency**
  - Deep ConvNets은 over-parameterized인 경우가 종종 있음
    - Model compression (Deep compression, Amc, Netadapt) → accuracy를 포기하고 efficiency를 선택
    - Handcraft efficient mobile-size ConvNets (SqueezeNets, MobileNets, ShuffleNets)
    - Neural Network Search
  - **이런 기법들이 큰 모델에서 어떻게 적용되는지 불명확함**

Related Work

- **Model Scaling**
  - Depth : ResNet (18 – 200)
  - Width (channels) : WideResNet, MobileNets
  - Image resolution
  - **하지만 효율적으로 accuracy를 얻는 방법은 open question으로 여전히 남아있음**

Compound Model Scaling

- **Problem Formulation**
  - ConvNet Layer i : $Y_i = \mathcal{F}_i(X_i)$

    ($\mathcal{F}_i$: operator, Y_i : output tensor, X_i : input tensor with tensor shape $\langle H_i, W_i, C_i \rangle$)
  - ConvNet $\mathcal{N} = \mathcal{F}_k \odot \cdots \odot \mathcal{F}_1 \odot \mathcal{F}_1(X_1) = \odot_{j=1\ldots k} \mathcal{F}_j(X_1)$

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

  - Model scaling은 best layer architecture $\mathcal{F}_i$를 찾기보다 $L_i, C_i, (H_i, W_i)$를 확장

$$\begin{aligned} \max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\ s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\left(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i \rangle}\right) \\ & \text{Memory}(\mathcal{N}) \le \text{target\_memory} \\ & \text{FLOPS}(\mathcal{N}) \le \text{target\_flops} \end{aligned} \qquad (2)$$

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Compound Model Scaling

- **Scaling Dimensions**
  - Depth (d)
    - Scaling network depth : ResNet, DenseNet, Inception
    - Deeper ConvNet은 **richer and more complex feature**를 찾고 **robust**함
    - **Vanishing gradient 때문에 학습이 어려움**
  - Width (w)
    - Scaling network width : MobileNets v1/v2, MnasNet
    - WRNs : wider network가 more fine-grained feature를 잡고 학습이 비교적 쉬움
    - **Extremely wide but shallow network는 higher level feature를 잡기 힘듬**
  - Resolution (r)
    - ConvNets with higher resolution input images는 more fine-grained patterns를 잡을 수 있음
    - 224x224 → 299x299 (Inception v3) → 331x331 (NASNet) → 480x480 (GPipe) → 600x600 (Mask R-CNN, FPN)

Compound Model Scaling

- **Scaling Dimensions**
  - Width, depth, or resolution의 scaling up은 accuracy를 증가
  - **하지만 모델이 커질수록 accuracy gain은 감소**

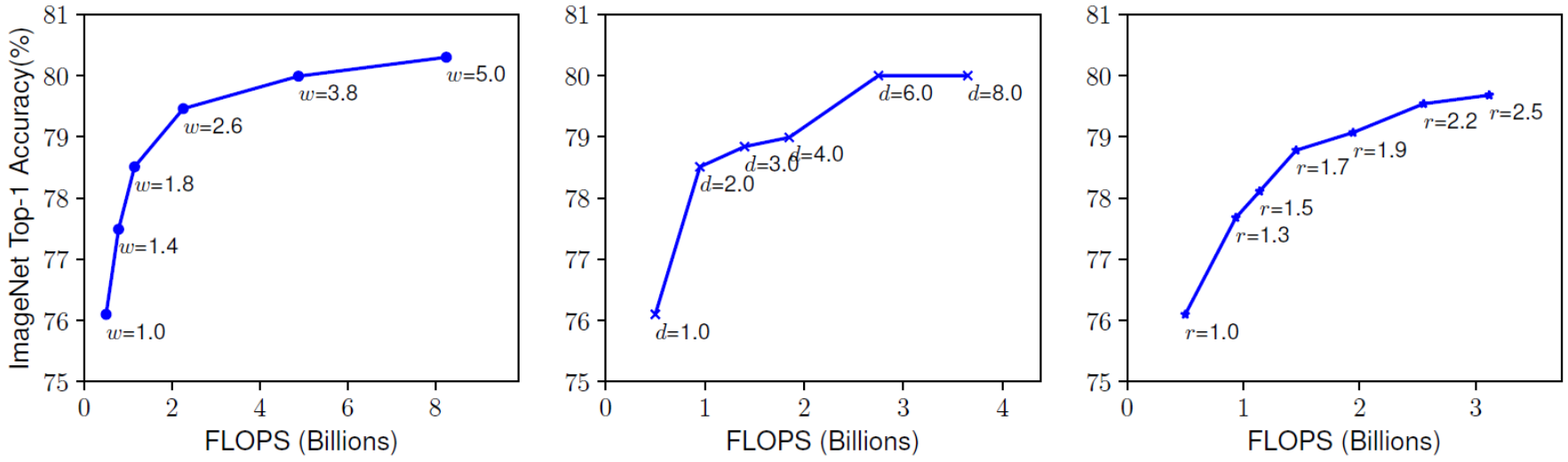


Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling. Baseline network is described in Table 1.

Compound Model Scaling

- **Compound Scaling**
  - Different scaling dimension은 서로에게 종속적임!
    - **Higher resolution image는**
      - **network depth를 늘려서**
      - **larger receptive field와**
      - **more fine-grained patterns를 잡을 수 있게 해야함**

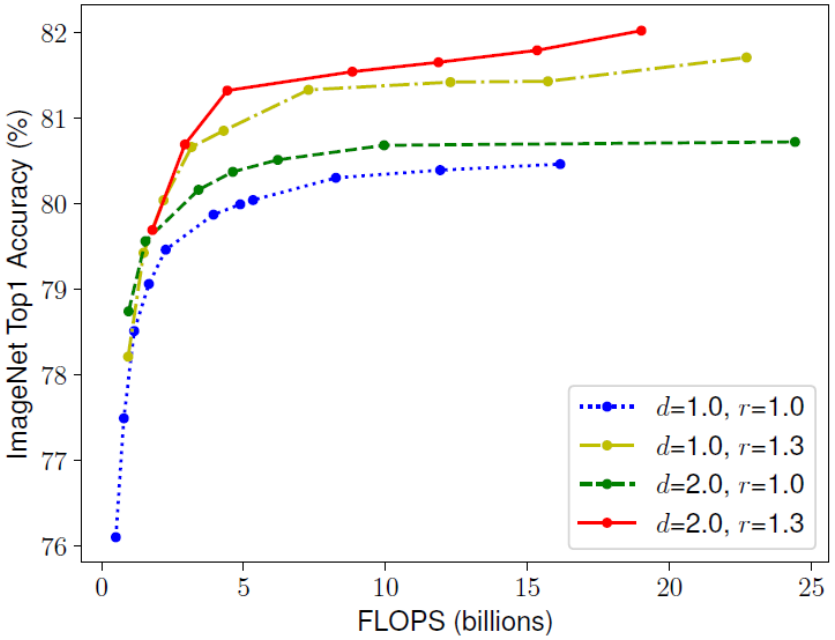


Figure 4. **Scaling Network Width for Different Baseline Networks**. Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline (d=2.0, r=1.3) has 36 layers with resolution 299x299.

Compound Model Scaling

- **Compound Scaling**
  - **Compound scaling method**
    - α, β, γ는 small grid search를 통해 결정
    - ϕ : user-specified coefficient
    - FLOPS $\propto d, w^2, r^2$
    - Total FLOPS = $(\alpha \cdot \beta^2 \cdot \gamma^2)^\phi \approx 2^\phi$

$$\begin{aligned} \text{depth: } & d = \alpha^\phi \\ \text{width: } & w = \beta^\phi \\ \text{resolution: } & r = \gamma^\phi \\ \text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned} \tag{3}$$

EfficientNet Architecture

- $\hat{\mathcal{F}}_i$을 바꾸지 않기 때문에, 좋은 baseline network 선택이 중요!!!
  - Accuracy와 FLOPS 모두 최적화하는 Multi-objective neural architecture search (MnasNet) 사용
  - $ACC(m) \times [FLOPS(m)/T]^w$ 를 최적화
    - T : target FLOPS
    - $w = -0.07$: accuracy와 FLOPS의 trade-off를 조절

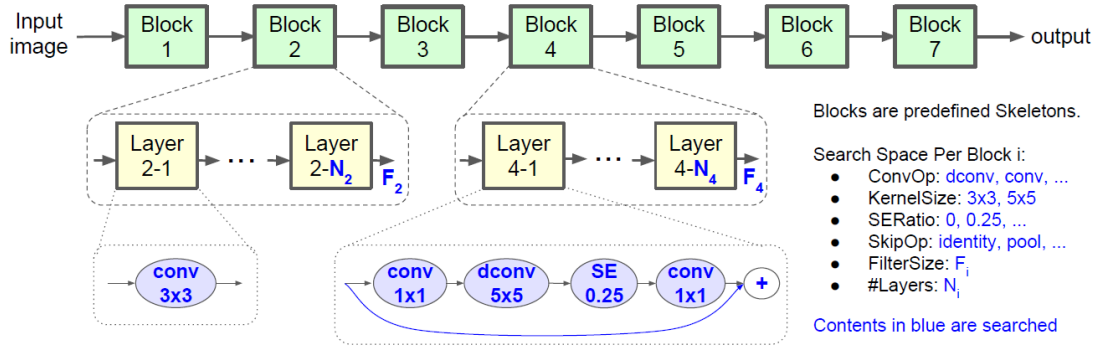
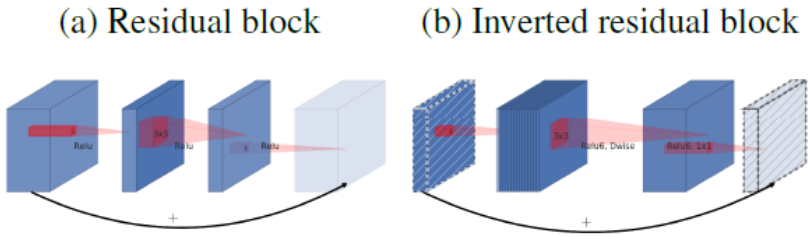


Figure 4: **Factorized Hierarchical Search Space.** Network layers are grouped into a number of predefined skeletons, called blocks, based on their input resolutions and filter sizes. Each block contains a variable number of repeated identical layers where only the first layer has stride 2 if input/output resolutions are different but all other layers have stride 1. For each block, we search for the operations and connections for a single layer and the number of layers N, then the same layer is repeated N times (e.g., Layer 4-1 to 4-N_4 are the same). Layers from different blocks (e.g., Layer 2-1 and 4-1) can be different.

MnasNet

EfficientNet Architecture

- **EfficientNet-B0**
  - Main building block : mobile inverted bottleneck MBConv + SE optimization



MBConv in MobileNet V2

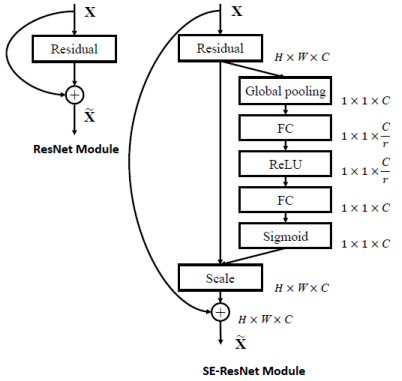


Fig. 3. The schema of the original Residual module (left) and the SE-ResNet module (right).

SE Block in SENet

- **EfficientNet family**
  - STEP 1
    - Fix $\phi = 1$, resource를 2배 사용 가능하다고 가정하고 α, β, γ를 small grid search

      → $\boldsymbol{\alpha = 1.2,\ \beta = 1.1,\ \gamma = 1.15}$ **unser constraint of** $\boldsymbol{\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2}$
  - STEP 2
    - Fix α, β, γ, scale up baseline network with different ϕ

      → **EfficientNet-B1 to B7**

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Table 2. **EfficientNet Performance Results on ImageNet** (Russakovsky et al., 2015). All EfficientNet models are scaled from our baseline EfficientNet-B0 using different compound coefficient ϕ in Equation 3. ConvNets with similar top-1/top-5 accuracy are grouped together for efficiency comparison. Our scaled EfficientNet models consistently reduce parameters and FLOPS by an order of magnitude (up to 8.4x parameter reduction and up to 16x FLOPS reduction) than existing ConvNets.

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., 2019)	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).

Experiments

- **Scaling Up MobileNets and ResNets**
  - MobileNets과 ResNets에 compound scaling method 적용

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

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1 (Howard et al., 2017)	0.6B	70.6%
Scale MobileNetV1 by width (w=2)	2.2B	74.2%
Scale MobileNetV1 by resolution (r=2)	2.2B	72.7%
compound scale (d=1.4, w=1.2, r=1.3)	**2.3B**	**75.6%**
Baseline MobileNetV2 (Sandler et al., 2018)	0.3B	72.0%
Scale MobileNetV2 by depth (d=4)	1.2B	76.8%
Scale MobileNetV2 by width (w=2)	1.1B	76.4%
Scale MobileNetV2 by resolution (r=2)	1.2B	74.8%
MobileNetV2 compound scale	**1.3B**	**77.4%**
Baseline ResNet-50 (He et al., 2016)	4.1B	76.0%
Scale ResNet-50 by depth (d=4)	16.2B	78.1%
Scale ResNet-50 by width (w=2)	14.7B	77.7%
Scale ResNet-50 by resolution (r=2)	16.4B	77.5%
ResNet-50 compound scale	**16.7B**	**78.8%**

Experiments

- **ImageNet Results for EfficientNet**
  - MnasNet과 유사한 환경으로 EfficientNet을 ImageNet으로 학습
    - Swish activation ($= x \cdot \sigma(\beta x)$)
    - Fix AutoAugment policy
    - Stochastic depth with drop connect ratio 0.3 ($p_t = 1 - \frac{l}{L}(1 - P_L)$)
    - Dropout ratio 0.2 for B0 / 0.5 for B7

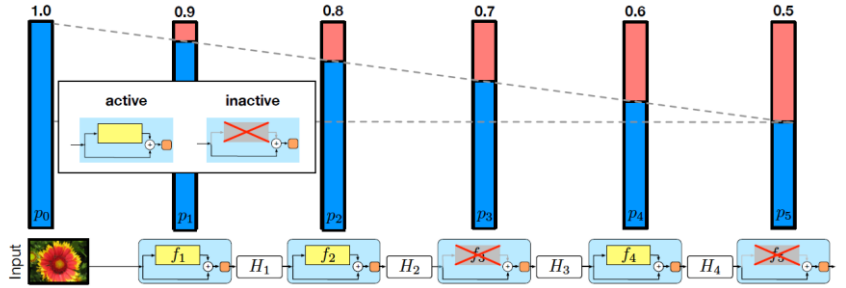


Fig. 2. The linear decay of p_ℓ illustrated on a ResNet with stochastic depth for $p_0 = 1$ and $p_L = 0.5$. Conceptually, we treat the input to the first ResBlock as H_0, which is always active.

Stochastic depth

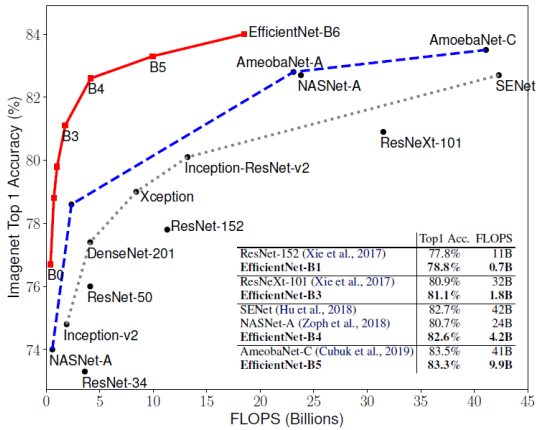


Figure 5. **FLOPS vs. ImageNet Accuracy –** Similar to Figure 1 except it compares FLOPS rather than model size.

Table 4. **Inference Latency Comparison** – Latency is measured with batch size 1 on a single core of Intel Xeon CPU E5-2690.

	Acc. @ Latency		Acc. @ Latency
ResNet-152	77.8% @ 0.554s	GPipe	84.3% @ 19.0s
EfficientNet-B1	78.8% @ 0.098s	EfficientNet-B7	84.4% @ 3.1s
Speedup	**5.7x**	**Speedup**	**6.1x**

Experiments

- Transfer Learning Results for EfficientNet

Table 6. **Transfer Learning Datasets.**

Dataset	Train Size	Test Size	#Classes
CIFAR-10 (Krizhevsky & Hinton, 2009)	50,000	10,000	10
CIFAR-100 (Krizhevsky & Hinton, 2009)	50,000	10,000	100
Birdsnap (Berg et al., 2014)	47,386	2,443	500
Stanford Cars (Krause et al., 2013)	8,144	8,041	196
Flowers (Nilsback & Zisserman, 2008)	2,040	6,149	102
FGVC Aircraft (Maji et al., 2013)	6,667	3,333	100
Oxford-IIIT Pets (Parkhi et al., 2012)	3,680	3,369	37
Food-101 (Bossard et al., 2014)	75,750	25,250	101

Table 5. **EfficientNet Performance Results on Transfer Learning Datasets**. Our scaled EfficientNet models achieve new state-of-the-art accuracy for 5 out of 8 datasets, with 9.6x fewer parameters on average.

	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)**

†GPipe (Huang et al., 2018) trains giant models with specialized pipeline parallelism library.
‡DAT denotes domain adaptive transfer learning (Ngiam et al., 2018). Here we only compare ImageNet-based transfer learning results.
Transfer accuracy and #params for NASNet (Zoph et al., 2018), Inception-v4 (Szegedy et al., 2017), ResNet-152 (He et al., 2016) are from (Kornblith et al., 2019).

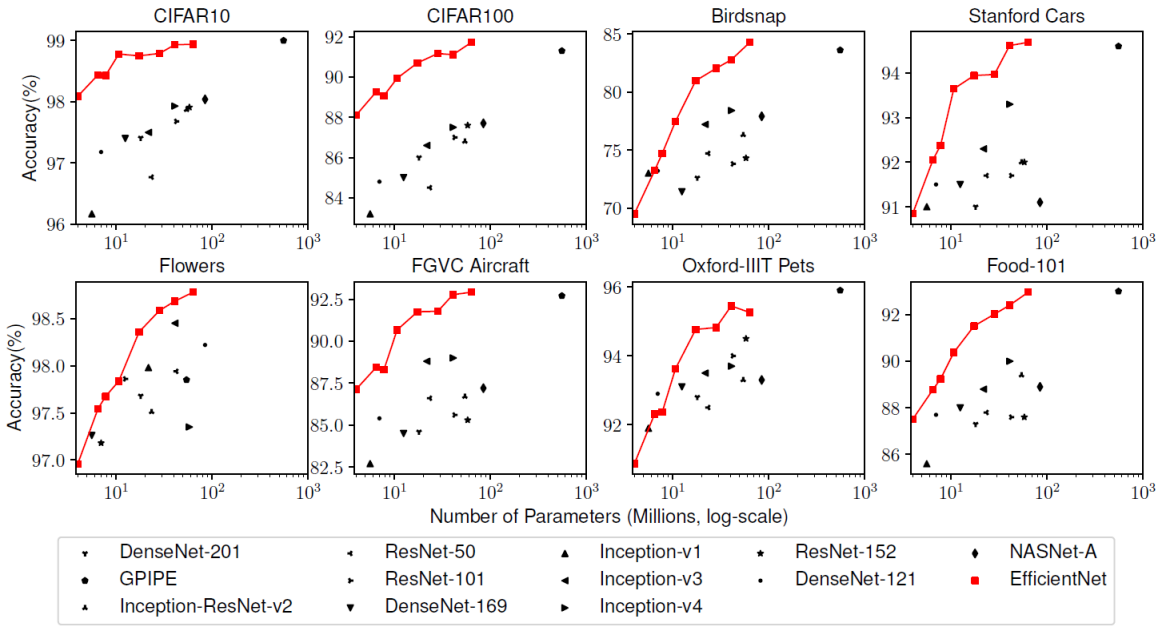


Figure 6. **Model Parameters vs. Transfer Learning Accuracy –** All models are pretrained on ImageNet and finetuned on new datasets.

Discussion

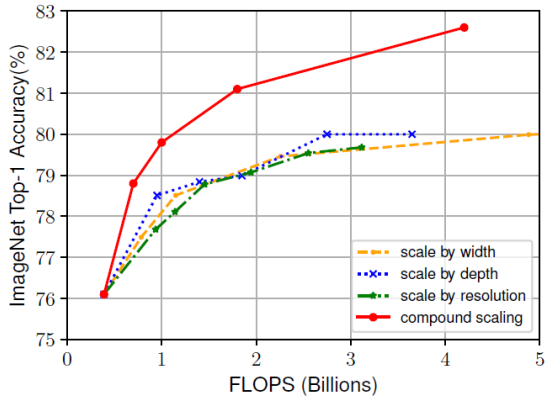


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

Table 7. **Scaled Models Used in Figure 7.**

Model	FLOPS	Top-1 Acc.
Baseline model (EfficientNet-B0)	0.4B	76.3%
Scale model by depth (d=4)	1.8B	79.0%
Scale model by width (w=2)	1.8B	78.9%
Scale model by resolution (r=2)	1.9B	79.1%
Compound Scale (d=1.4, w=1.2, r=1.3)	**1.8B**	**81.1%**

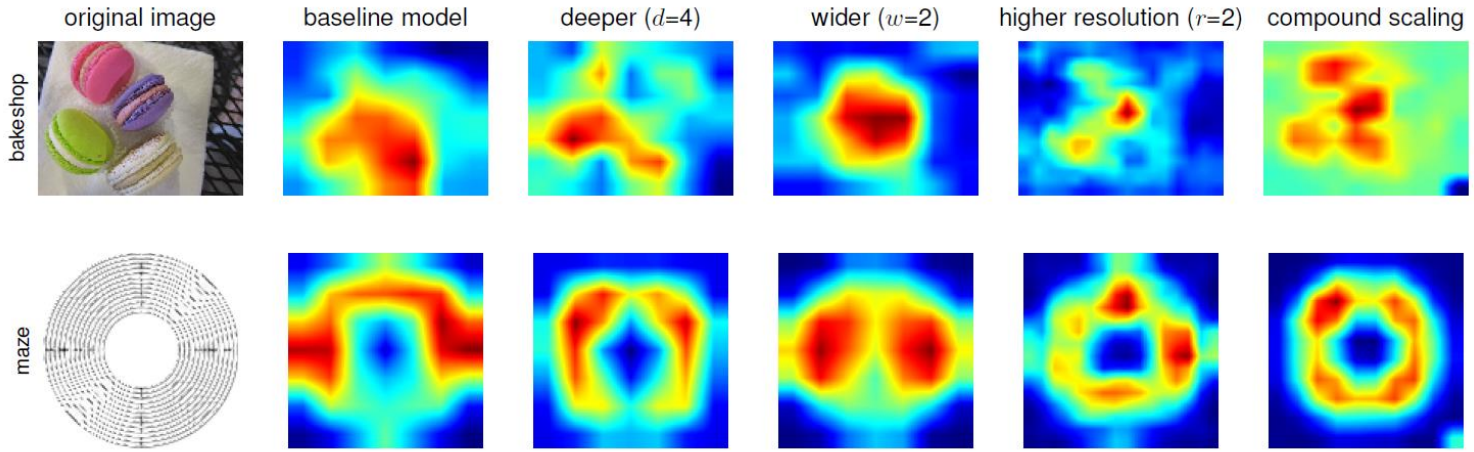


Figure 7. **Class Activation Map (CAM) (Zhou et al., 2016) for Different Models in Table 7 -** Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

감 사 합 니 다