EfficientNet

Introduction

기존 Network보다 파라미터 대비 정확도가 높은 효율적인 Network를 제시

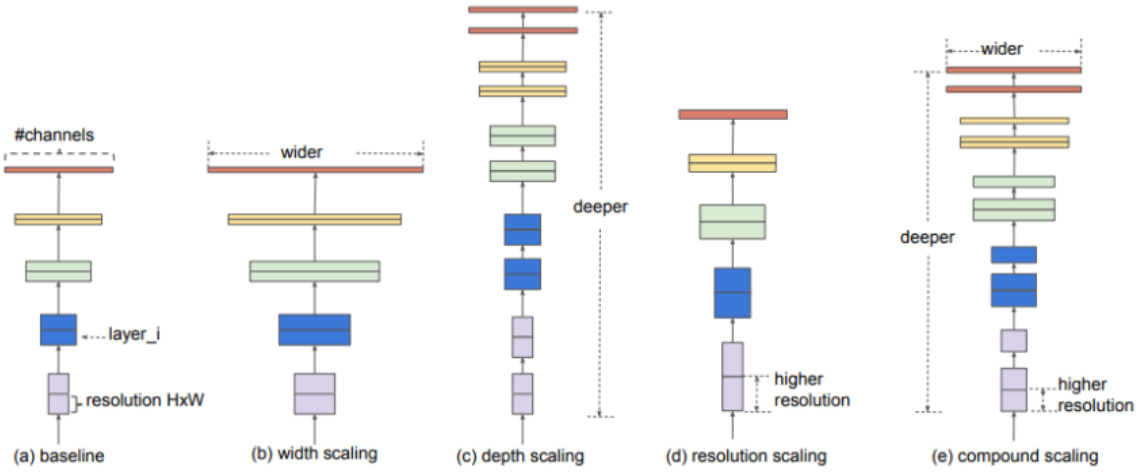
효율적인 Network라는 이름을 본따서 EfficientNet으로 명명

모델의 정확도를 높이기 위하여 일반적으로 모델의 깊이, 너비, 입력 이미지의 크기를 조절

→ 기존에는 이를 수동적으로 조절했기 때문에 최적의 성능과 효율 X

EfficientNet은 모델의 깊이, 너비, 입력 이미지의 크기를 효율적으로 조절할 수 있는 compound scaling 방법을 제안

Model Scaling



일반적으로 모델을 Scaling 하는 방법은 너비, 깊이, 입력 해상도를 조절

1. network의 depth를 깊게 만드는 것
2. channel width(filter 개수)를 늘리는 것(width가 넓을수록 미세한 정보가 많이 담아짐)
3. input image의 해상도를 올리는 것

→ 이러한 세가지 방법을 잘 종합하여 모델을 확장시키는 방법 Compound Scaling

(a) Baseline에서 입력값은 각 레이어 함수를 거쳐서 최종 출력값을 생성

$$\mathcal{N} = \mathcal{F}_k \odot ... \odot \mathcal{F}_2 \odot \mathcal{F}_1(X_1) = \bigodot_{j=1...k} \mathcal{F}_j(X_1).$$

입력값 X가 각 레이어의 함수 F연산을 거쳐서 출력값 N이 생성

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

여기서 각 레이어에서 수행하는 연산(F)를 고정하고, 레이어 수, 채널 수, 입력 이미지 크기에만 집중하기 때문에 search space가 감소

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

해당 부분에서 w, d, r 상수들의 관계를 연구하는 것이 EfficientNet

Depth (깊이)

- 네트워크의 깊이가 증가할수록 모델의 capacity가 커지고 더 복잡한 feature를 잡아낼 수 있지만, vanishing gradient의 문제로 학습시키기가 더 어려워짐
- 이를 해결하기 위해 Batch Norm, Residual Connection 등의 여러 기법들이 등장
- ResNet101 과 ResNet1000의 정확도가 비슷한 것 처럼 너무 깊은 Layer를 가진 모델의 성능은 더 좋아지지 않음

Width (너비)

- 보통 Width(Channel)는 작은 모델을 만들기 위해 scale down(MobileNet 등)을 하는 데에 사용
- 더 넓은 channel은 더 세밀한 특징을 추출할 수 있고 train 하기가 더 쉬움
- width의 증가에 따른 성능은 빠르게 saturate

- 각 레이어의 width를 키우면 정확도가 높아지지만 계산량이 제곱에 비례하여 증가

Resolution (해상도)

- 입력 이미지의 해상도를 키우면 더 세부적인 feature를 학습할 수 있어 정확도가 높아지지만 계산량이 제곱에 비례해 증가
- 성능을 높이기 위해서 Resolution을 크게 가져가고 있으며 최근에는 480x480(GPipe) 600x600(object detection)의 size를 사용
- 하지만 마찬가지로 너무 큰 해상도는 효율적이지 않음

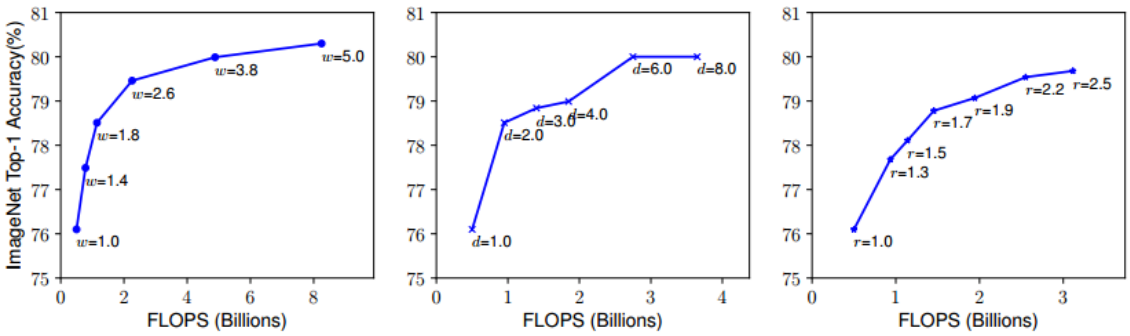


Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single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

width, depth, resoultion에 따른 정확도를 표현

width, depth, resoultion이 일정 값 이상되면 정확도가 빠르게 saturate, 값이 적을 때는 약간만 값을 조절해도 효과가 크게 나타남

어느 정도 이상 증가하면 모델의 크기가 커짐에 따라 얻는 정확도 증가량이 매우 적어짐

→ width, depth, resoultion 중 하나만 조절하는 것보다는 d와 r을 함께 조절하여 최고의 효율을 찾아내는 것이 Compound Scaling

더 높은 해상도의 이미지는 네트워크를 깊게 만들어서 더 넓은 영역에 걸쳐 있는 feature(by larger receptive fields)를 더 잘 잡아낼 수 있도록 하는 것이 유리하고, 더 큰 이미지일수록 세부적인 내용도 많이 담고 있어서, 이를 잘 잡아내기 위해서는 layer의 width를 증가시킬 필요가 있다.

→ 이 depth, width, resolution이라는 세 가지 변수는 밀접하게 연관, 이를 같이 움직이는 것이 도움

Compound Scaling

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

α,β,γ는 small grid search로 결정되는 상수

ϕ는 주어진 연산량에 따라 사용자가 결정하는 상수

FLOPs는 너비와 해상도에 따라 제곱배가 상승 (α·β^2·γ^2)2배 만큼 증가

논문에서는 α·β^2·γ^2=2로 제한 → 제한된 범위에서 α,β,γ를 찾음

논문에서 찾은 값은 α=1.2, β=1.1, γ=1.15

계산량은 깊이에 비례하고, 나머지 두 변수에 대해서 그 제곱에 비례

FLOPs는 2^ϕ 만큼 증가

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%**

ResNet과 MoblieNet을 Compound Scaling 했을 때의 결과 비교

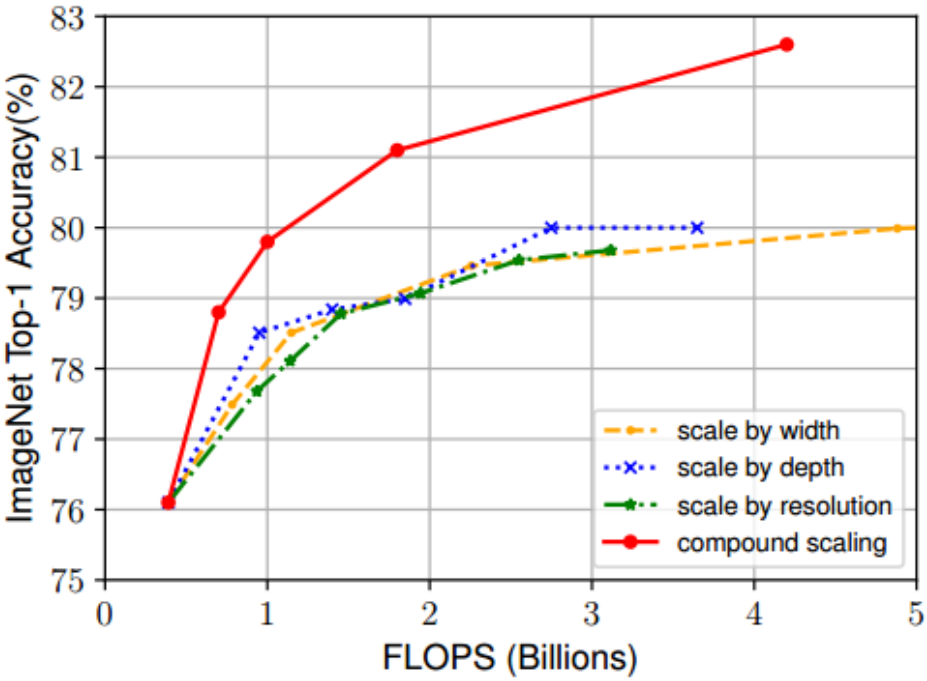


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

EfficientNet의 너비, 깊이, 입력 해상도를 하나만 조절했을 때와 Compound Scaling 했을 때의 성능 비교

depth, width, resolution은 서로 긴밀히 연관되어 있으며 이들을 같이 키우는 것이 자원을 더 효율적으로 쓰는 방법

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

Model	FLOPS	Top-1 Acc.
Baseline model (EfficientNet-B0)	0.4B	77.3%
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%**

Compound Scale는 비슷한 연산량을 가지면서 성능이 가장 좋음

EfficientNet Architecture

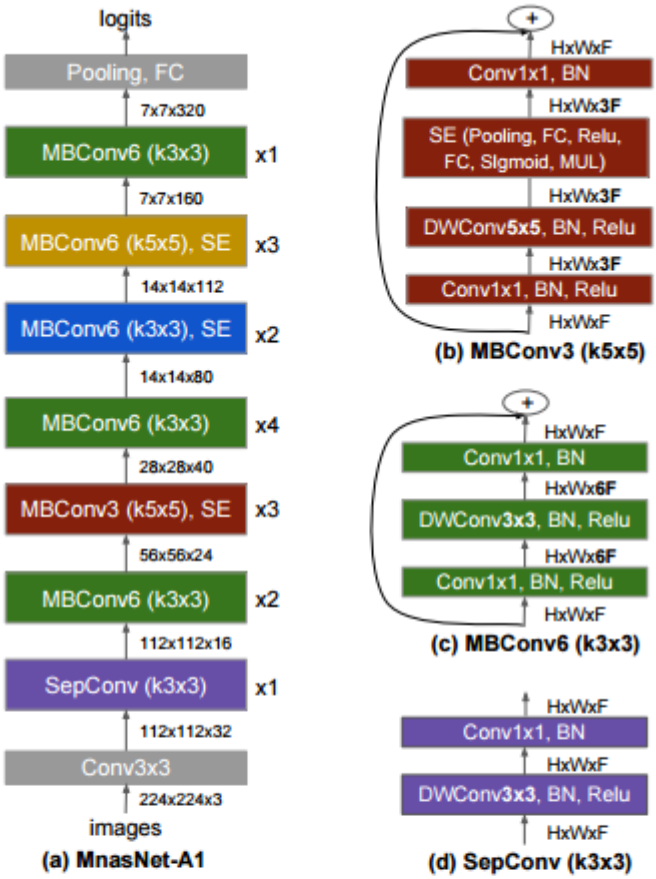


Figure 7: **MnasNet-A1 Architecture –** (a) is a representative model selected from Table 1; (b) - (d) are a few corresponding layer structures. *MBConv* denotes mobile inverted bottleneck conv, *DWConv* denotes depthwise conv, k3x3/k5x5 denotes kernel size, *BN* is batch norm, HxWxF denotes tensor shape (height, width, depth), and $\times 1/2/3/4$ denotes the number of repeated layers within the block.

EfficientNet은 MnasNet과 동일한 search space를 사용해서 찾는 구조이기 때문에 Mnasnet 구조와 비슷

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	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

EfficientNet의 구조 (scaling을 하지 않은 기본 B0모델 구조)

여기서 MBConv는 Mobilenet v2에서 제안된 inverted residual block을 의미

숫자 1 또는 6은 expand ratio

ImageNet dataset의 size인 224x224를 input size로 사용

Activation function으로 ReLU가 아닌 Swish 사용

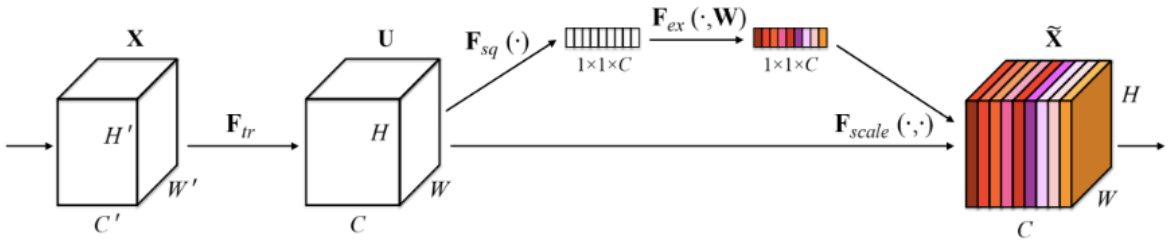
Swish 는 매우 깊은 신경망에서 ReLU 보다 높은 정확도를 달성

맨 처음 ϕ=1로 고정하고, α,β,γ에 대해서 작게 grid search를 수행

후에 α,β,γ를 고정하고, ϕ를 변화시키면서 전체적인 크기를 키우는 방식으로 학습

squeeze-and-excitation

squeeze-and-excitation optimization을 추가



pooling을 통해 1x1 size로 줄여서 정보를 압축한 뒤에 압축된 정보들을 weighted layer와 비선형 activation function으로 각 채널별 중요도를 계산하여 기존 input에 곱을 해주는 방식 (SENet 참고)

Experiments

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.

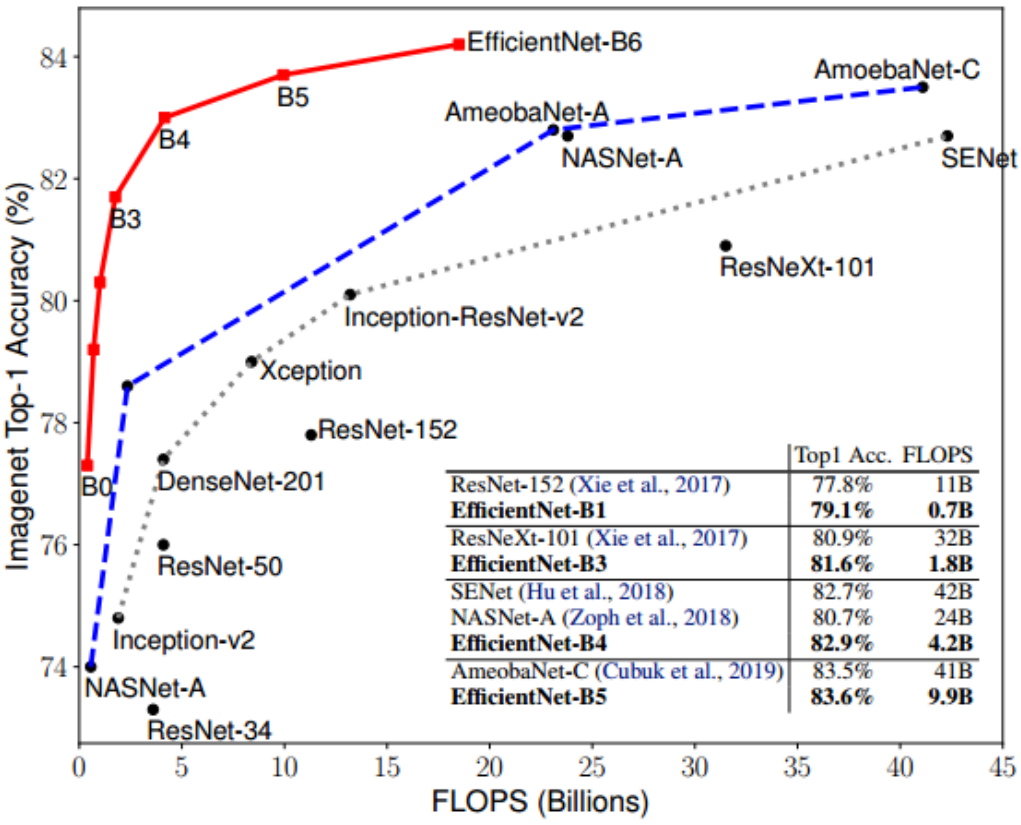
Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**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	**80.1%**	**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	**81.6%**	**95.7%**	**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	**82.9%**	**96.4%**	**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	**83.6%**	**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%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**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).

ImageNet 데이터에서의 EfficientNet의 성능

EfficientNet의 B0~B7 까지 각각의 성능과 비슷한 SOTA모델을과 비교

같은 성능 대비 파라미터와 FLOPS가 EfficentNet 모델들이 압도적으로 적은 것 확인



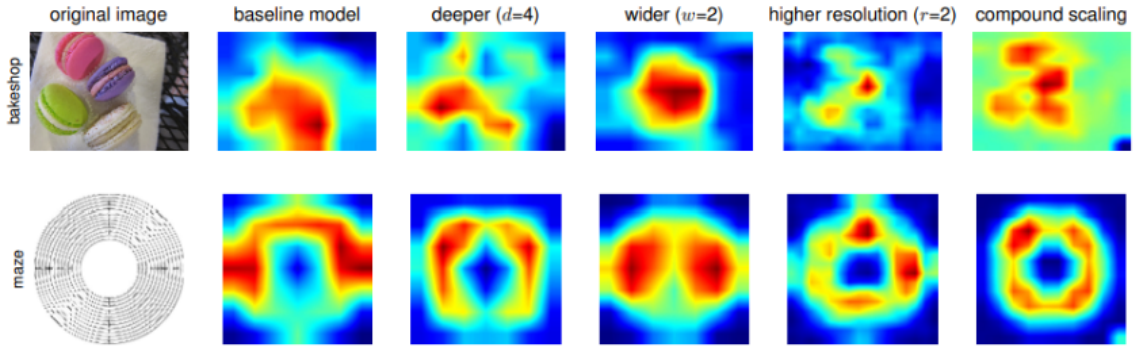
	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**79.1%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.6%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**82.9%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.6%**	**9.9B**

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

FLOPs와 정확도를 비교한 그래프

EfficientNet이 SOTA image classification network보다 효율적인 모델

ImageNet에서 기존 ConvNet보다 8.4배 작으면서 6.1배 빠르고 더 높은 정확도



width, height, resolution을 각각 따로 scaling up 했을때와 compund scaling 했을때의 Class Activation Map을 나타낸 사진

compound scaling이 각각의 객체들을 잘 담고 있으며 좀 더 정확