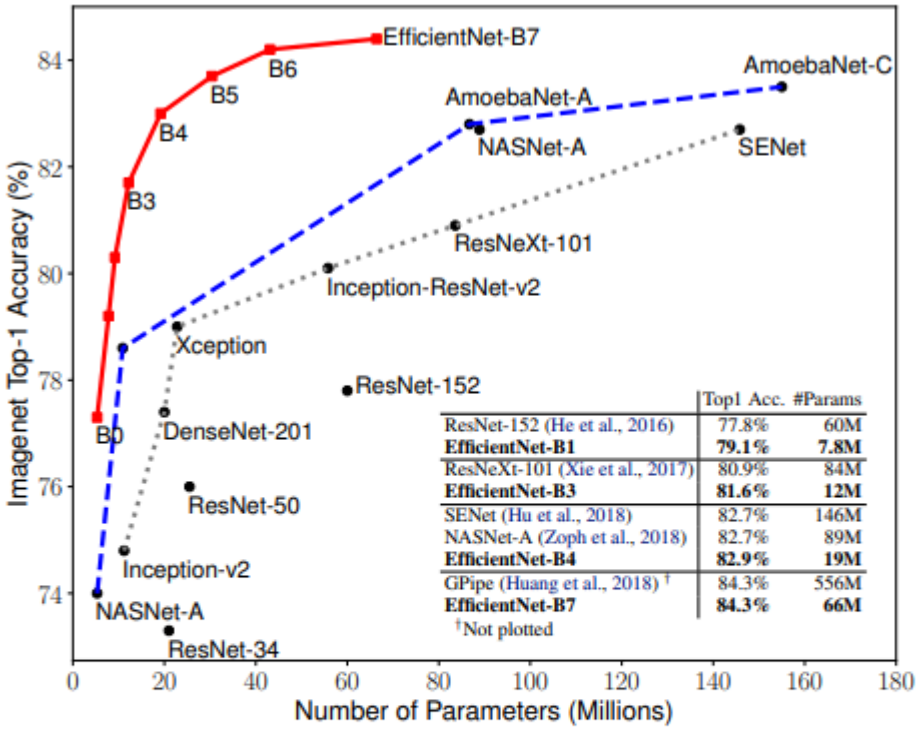
EfficientNet

Category | Computer Vision

들어가며



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

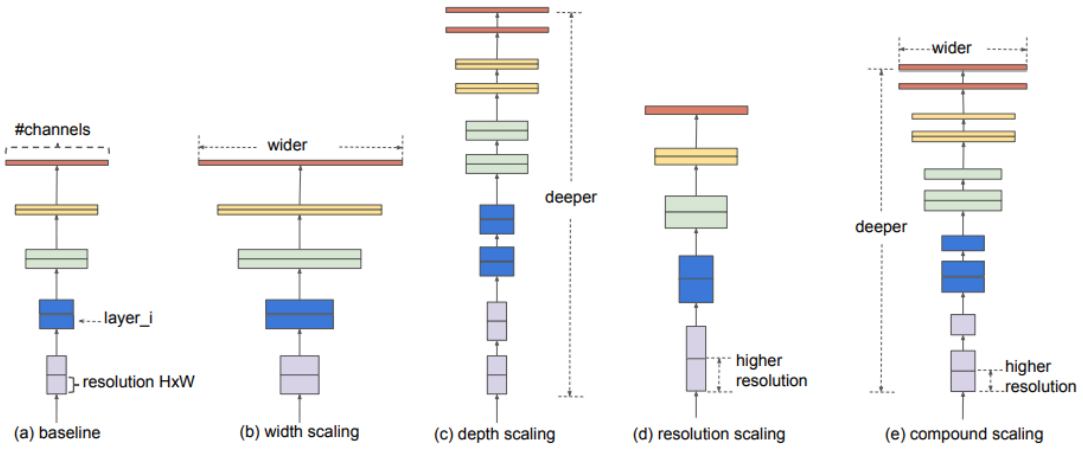
†Not plotted

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 84.3% 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.

- CNN은 한정된 예산 내에서 개발 후, 리소스가 더 주어진 가면 정확도를 향상시키기 위해 확장하는 방식으로 설계.
- 모델 크기 조정(model scaling)을 체계적으로 연구하고, 네트워크의 깊이(depth), 너비(width), 해상도(resolution) 간의 균형을 신중히 조정하면 더 나은 해상도를 얻을 수 있음
- 이 논문은 모든 차원(깊이, 너비, 해상도)를 균일하게 조정하는 방법을 제안

- 단순하지만 효율적인 복합 계수(compound coefficient)을 확장하여 모델 확장
- MobileNet, ResNet 확장으로 효과 입증
- 신경망 아키텍처 탐색(Neural Architecture Search)을 이용하여 새로운 기본 네트워크를 설계하고 확장하여 EfficientNets이라는 모델 패밀리를 만듬.(모델 모음)
- CNN에 비해 훨씬 더 높은 정확도와 효율성을 보임
- EfficientNet-B7은 ImageNet 데이터셋에서 최첨단 성능인 84.3%의 top-1 정확도 기록
    - 기존의 CNN보다 8.4배 작고, 추론 속도가 6.1배 빠름
- EfficientNet은 CIFAR-100에서 91.7%, Flowers에서는 98.8%의 정확도를 기록
- 기존 모델보다 매개변수 수가 10배 이상 적은 효율적 모델

소개



- 더 높은 정확도를 달성하기 위해서는 CNN를 확장하는 방식이 사용됨
- CNN 확장하는 과정에서는 너비나 이미지 크기 중 하나를 선택해서 확장하는게 일반적임.
    - 두개 또는 세개를 동시에 확장하게 되는 경우, 수작업으로 세밀하게 조정이 필요해 최적의 정확도와 효율성이 보장되지 않음.
- CNN 확장시 더 높은 정확도와 효율성을 달성할 수 있는 체계적 방법이 존재할까?

    - 실험에서, 네트워크의 너비, 깊이, 해상도의 모든 차원을 균형 있게 확장하는 것이 중요하고, 이 균형은 각 차원을 일정한 비율로 확장함으로써 달성할 수 있음
- 고정된 스케일링 계수(fixed scaling coefficients)를 사용하여 신경망의 너비, 깊이, 해상도를 균일하게 확장
    - 컴퓨터 자원을 2^n만큼 확장시키고 싶으면, 가볍게 네트워크의 너비, 깊이, 해상도를 작은 원본 모델에서 작게 grid search하여 결정된 상수 계수들의 n승만큼 증가가 가능함.
    - Compound scaling은 큰 이미지에서 receptive field를 확장하기 위해 더 많은 레이어가 필요하고, 세밀한 패턴(fine-grained pattern)을 파악하기 위한 더 많은 채널이 필요함.
- 이 논문은 실험적으로 신경망의 너비, 깊이, 해상도라는 세 가지 차원의 관계를 실험적으로 정량화함.
- 기존의 MobileNet과 Resnet에서 잘 작동했고, 모델 확장의 효과를 더 발전시키기 위해, Neural Architecture Search(신경망 아키텍처 탐색)을 사용하여 새로운 기본 네트워크를 개발하고, 확장해 EfficientNet 모델군을 개발함.

관련 연구

CNN 정확도

- CNN 정확도는 점점 더 정확해지고 있음
- 다른 전이 학습 데이터셋에서도 좋은 성능을 보임(Kornblith et al, 2019)
- 객체 탐지와 같은 다른 컴퓨터 비전 작업에서 성능 발휘
- 하지만, 하드웨어 메모리 한계(hardware memory limit) 문제에 다다랐기에, 효율성 개선이 필요.

CNN 효율성

- CNN 깊게 쌓으면, 과도하게 매개변수화(overparameterized)됨.
- 모델 압축(Model Compression, Han et al, 2016;He et al.. , 2018;Yang et al.., 2018;)은 효율성을 위해 정확도를 희생하면서 모델 크기를 줄이는 일반적 방법
    - 모바일 기기가 널리 보급되면서. 모바일 크기의 효율적인 CNN 설계가 일반적.

    - 예) SqueezeNet, MobileNEts, ShuffleNEts
- 최근에는 Neural Architecture Search가 모바일 크기의 효율적인 CNN 설계하는 데 인기를 끌고, 수작업으로 광범위하게 너비, 깊이, 커널 타입과 사이즈를 튜닝하는 것보다 더 효율적.
- 그러나 디자인 공간이 훨씬 더 크고 튜닝 비용이 매우 비싼 큰 모델에 어떻게 적용될지는 확실하지 않음.
- 따라서 기존의 최첨단 정확도를 능가하는 매우 큰 CNN의 모델 효율성을 연구하고자, 모델 스케일링(model scaling)에 의존함.

모델 스케일링(Model Scaling)

- ResNet은 신경망의 깊이(layers)를 조절함으로써 업스케일링, 다운스케일링이 가능했음.
- MobileNets와 WideResNet은 신경망의 너비(channels)을 조정함으로써 조정이 가능
- 큰 입력 이미지의 크기는 더 많은 부동 소숫점 연산량(FLOPS)가 발생하지만 정확도 향상에 도움을 줌.
- 이전 연구에서는 CNN의 표현력에 있어 신경망의 깊이, 너비 둘 다 중요했으나, 더 나은 효율성과 정확성을 도달하기 위해 어떻게 조정하느냐에 대한 질문은 여전히 열려 있음.
- 이 논문은 신경망의 너비, 깊이, 해상도의 세 가지 차원을 대상으로 CNN scaling을 체계적이고 실험적으로 연구함

Compound Model Scaling

- 단순함을 위해서 Batch Noramlization은 하지 않음.(논문에서 설명시, 실제로는 뒤에서 보듯 해야 함)

문제 공식화

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

where w, d, r are coefficients for scaling network width, depth, and resolution; $\hat{\mathcal{F}}_i, \hat{L}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i$ are predefined parameters in baseline network (see Table 1 as an example).

- 일반적으로 CNN 설계시에는 가장 좋은 레이어 아키텍처를 찾는데 집중하나, 모델 스케일링은 미리 정의된 모델 변경 없이, 신경망의 길이와 너비, 해상도를 확장하는 것이 목표임.
- 모델을 고정하면, 모델 스케일링은 새 자원 제약에서의 설계 문제를 단순화하나, 각 레이어에 대해서 다양한 길이, 너비, 해상도를 탐색해야하는 넓은 설계 공간이 남음.
- 설계 공간을 줄일려면, 모든 레이어를 일정한 비율로 균일하게 스케일링되야함.
- 주어진 자원 안에서, 모델 정확도를 최적화하는 것.

차원 확장

- 신경망의 너비, 깊이, 해상도를 확장하는 최적의 계수들은 다양한 자원 제약 하에서 값들이 변하며, 서로 의존적인 문제가 있음
  - 기존 방법론들은 대개 깊이, 너비, 해상도 중 하나를 조정해야 했음.
  - 깊이(depth)
    - 더 깊은 CNN은 풍부하고 복잡한 feature를 잡아내서, 새로운 작업을 잘 일반화할 수 있음.
    - 하지만 무조건 깊게 쌓는다고 좋은게 아닌건, Vanishing gradient(기울기 소실) 문제로 인해 학습이 어려워짐.
      - 여러 기법들이 있으나, 엄청 깊은 신경망에서 정확도 향상은 미미함.
        - 예) ResNet-1000 vs ResNet-101(ResNet-1000의 정확도는 ResNet-101과 유사)

- 너비(width)
  - 작은 모델에서 너비를 확장하는 건 흔함.
  - 더 넓은 신경망은 더 세밀한 특징을 잡아내고, 학습에 용이함
  - 하지만, 매우 넓고 얕은 신경망은 고수준의 특징(feature)을 캡처하는 데 어려움
- 해상도(resolution)
  - 고해상도 이미지를 입력으로 사용하면, CNN이 더 세밀한 패턴을 잡아낼 수 있음
  - 초기 CNN에서는 224x224 해상도를 사용했으나, 현대에는 299x299, 331×331 해상도의 이미지도 사용하고, 480x480 해상도에서도 최첨단의 정확도를 달성하기도 함.
  - 하지만 매우 높은 해상도에서는 정확도 향상이 둔화되는 문제가 있음.

- 이 관찰을 통해 우리가 알 수 있는 점은, 신경망의 너비, 깊이, 해상도를 확장하면 정확도는 올라가지만, 큰 모델에서 정확도 향상은 둔화됨.

Compound Scaling(복합 스케일링)

- 경험적으로 서로 다른 스케일링 차원이 독립적이지 않음
- 고해상도 이미지 사용시 신경망을 깊게 쌓아야 하며, 깊게 쌓으면, 더 큰 receptive field(수용 영역)이 더 많은 픽셀을 갖는 더 큰 이미지에서 유사한 특징들을 잡아내는데 도움이 됨.
- 따라서 해상도를 높이면, 신경망 너비도 늘려서, 더 많은 픽셀로 더 세밀한 패턴을 잡아낼 수 있음.
- 서로 다른 스케일링 차원을 조정하고 균형을 맞춰야 함.

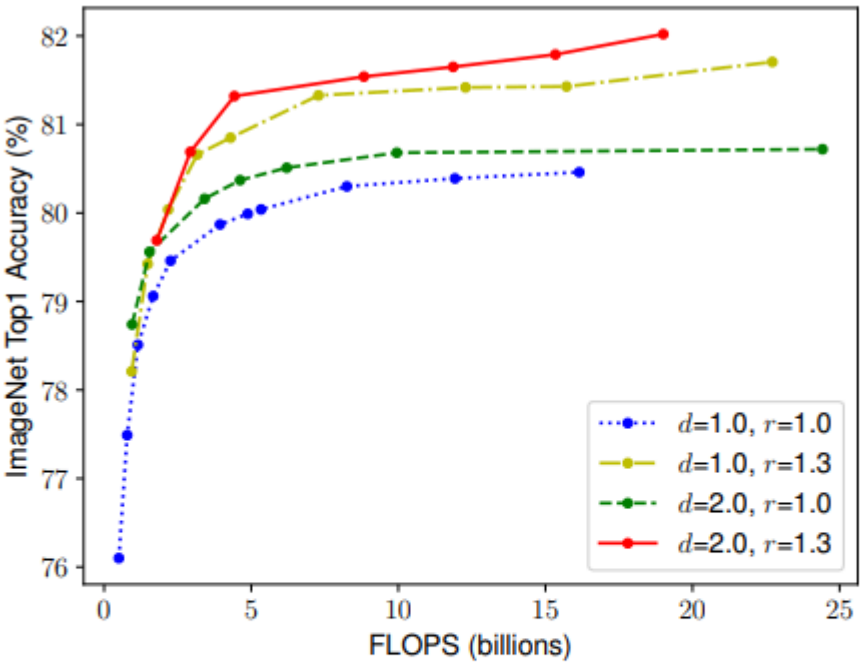


Figure 4. **Scaling Network Width for Different Baseline Networks**. Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline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

- 위의 그림을 토대로 얻을 수 있는 결론은, CNN 스케일링 시 신경망의 너비, 깊이, 해상도를 균형 있게 조절해야 더 나은 정확도와 효율성 추구 가능!

In this paper, we propose a new **compound scaling method**, which use a compound coefficient ϕ to uniformly scales network width, depth, and resolution in a principled way:

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

where α, β, γ are constants that can be determined by a small grid search. Intuitively, ϕ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively. Notably, the FLOPS of a regular convolution op is proportional to d, w^2, r^2, i.e., doubling network depth will double FLOPS, but doubling network width or resolution will increase FLOPS by four times. Since convolution ops usually dominate the computation cost in ConvNets, scaling a ConvNet with equation 3 will approximately increase total FLOPS by $\left(\alpha \cdot \beta^2 \cdot \gamma^2\right)^\phi$. In this paper, we constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ such that for any new ϕ, the total FLOPS will approximately[3] increase by 2^ϕ.

- 복합 계수(compound coefficient)승을 하여 균형있게 신경망의 너비, 깊이, 해상도를 조정할 수 있음.
- 알파, 베타, 감마는 작은 그리드 탐색으로 결정되는 상수이자, 추가 자원을 신경망 너비, 깊이, 해상도에 어떻게 분배할지 결정함
- 복합 계수는 모델 스케일링을 위한 추가 자원이 얼마나 있는지 제어하는 사용자 지정 계수
- 네트워크 깊이를 두배로 늘리면, FLOPS가 두 배 증가하나, 너비나 해상도를 두 배 늘리면, FLOPS가 4배 증가함.
- 이 수식에서 제약 조건을 검으로써 전체 FLOPS가 2^ϕ배 증가하게 함.

Architecture

- 좋은 기본 모델을 갖는게 중요
- Tan et al의 MobileNet 모델(2019)과 같은 탐색 공간을 갖고, 최적화 목표로는 다음 수식을 만족하게 함.

$$ACC(m) \times [FLOPS(m)/T]^{\omega}$$

- ACC : 모델 m의 정확도, FLOPS는 모델 m의 FLOPS
- T는 목표 FLOPS, ω는 정확도와 FLOPS간 균형을 제어하는 하이퍼파라미터로서 -0.07로 설정됨.
- 특정 하드웨어 장치를 목표로 하지 않기에, 지연 시간(latency)보다는 FLOPS를 최적화하여 EfficientNet-B0 설계
  - 이 모델은 MnasNet과 유사하나, 400M이라는 FLOPS 목표로 인해 약간 큼.
  - 구성 요소는 MobileNet의 inverted bottleneck인 MBConv로 이루어져 있음
    - 여기에 Squeze-and-excitation 최적화를 추가.
- EfficientNet-B0을 기반으로 compound scaling을 두 단계로 적용함
  - 첫 번째는, 복합 계수를 1로 고정하고, 자원이 두 배 더 제공된다 가정하고, 알파와 베타와 감마를 찾는 작은 그리드 탐색을 위의 (2),(3)번 방정식에 기반하여 진행함.
  - 특히, EfficientNet-B0에 대한 최적 값은 α=1.2, β=1.1, γ=1.15이며,

    $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$를 만족해야 함.
  - 이후 알파, 베타, 감마를 상수로 고정하고, (3)번 방정식을 사용해 기존 모델을 다양한 복합 계수(파이값)으로 확장하여 EfficientNet-B1에서 B7까지 얻음.
- 더 큰 모델을 찾기 위해 알파와 베타와 감마를 탐색하면 더 나은 성능을 얻으나, 큰 탐색 비용으로 인해 큰 모델에서 비용이 너무 많이 듬.
  - 따라서 1단계에서 한 번만 작은 기본 모델에서 탐색을 수행 후, 2단계에서는 동일한 스케일링 계수를 다른 모델들에 적용하여 해결함.

실험

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).

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%**

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**

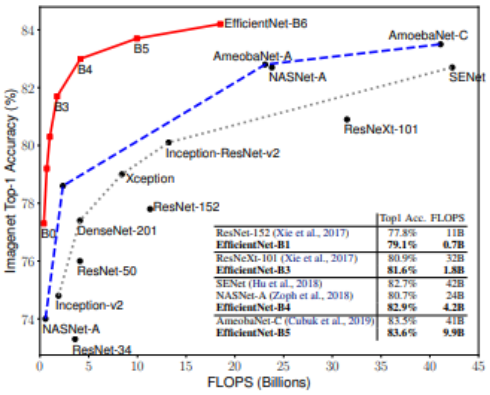


Figure 5. **FLOPS vs. ImageNet Accuracy** – Similar to Figure 1 except it compares FLOPS rather than model size.

5.2. ImageNet Results for EfficientNet

We train our EfficientNet models on ImageNet using similar settings as (Tan et al., 2019): RMSProp optimizer with decay 0.9 and momentum 0.9; batch norm momentum 0.99;

ImageNet에서의 EfficientNet 결과

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 (Komblith et al., 2019).

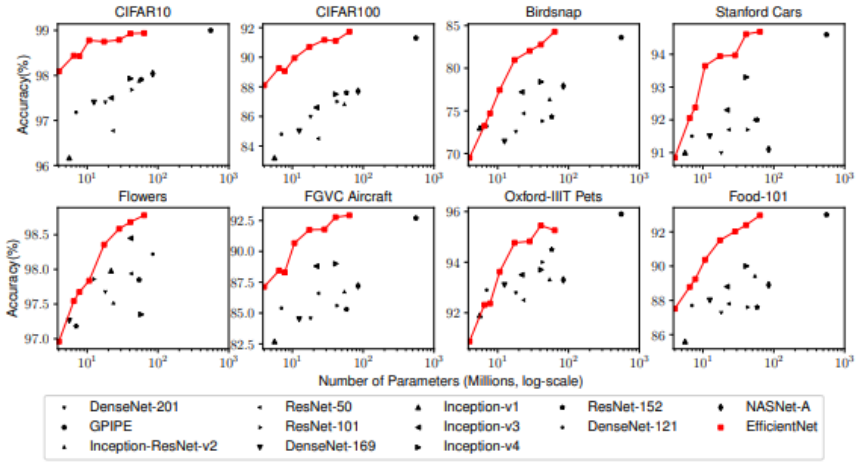


Figure 6. **Model Parameters vs. Transfer Learning Accuracy –** All models are pretrained on ImageNet and finetuned on new datasets.

- 이미지넷에서 훈련시 다음과 같이 설정함
  - 최적화 알고리즘은 RMSProp 사용 (momentum=0.9, decay(감쇠계수)=0.9)
  - Batch Normalization 사용하고 momentum=0.99로 설정
  - weight decay : 1e-5
  - 초기 learning rate : 0.256(epoch 2.4마다 0.97배씩 감소)
  - 활성화 함수는 SiLU(Swish-1) 사용(시그모이드에 입력값을 곱해주는 것, beta값 1)
  - AutoAugment 기법과 생존률 0.8로 Stochastic Depth 설정
  - 큰 모델일수록 정규화가 더 많이 필요하여, B0에서는 dropout을 0.2, B7까지 선형적으로 dropout을 증가시켜 0.5까지 설정.
- 학습 데이터셋에서 무작위로 25000개의 이미지를 미니 검증 세트(minival set)로 분류하고, early stopping(조기 종료) 사용.
- 보고서에서 최종 검증 정확도는 미니 검증세트에서 조기 종료된 체크포인트를 원래 validation set(검증 데이터셋)에서 평가.
- 성능

    - 유사한 정확도를 가진 다른 CNN 모델들보다 파라미터와 FLOPS를 10배나 적게 사용함
    - 특히, EfficientNet-B7은 84.3%의 Top-1 정확도와 66M개의 매개변수와 37B FLOPS 사용해 GPipe(Huang et al, 2018)보다 8.4배 더 작으면서도 높은 구조를 보임
    - EfficientNet-B3은 ResNeXt-101보다 높은 정확도를 갖고, FLOPS가 18배나 더 적음.
- 성능 향상은 더 나은 네트워크 구조와 더 효율적인 스케일링, 모델에 최적화된 훈련 설정에서 기인함.
- Latency (지연) 시간을 CPU 환경에서 추론 지연 시간(inference latency)를 검증해 본 결과, EfficientNet-B1은 ResNet-152보다 5.7배 빠르고 실행되고, EfficientNet-B7은 GPipe보다 6.1배 빠르게 실행됨.
    - 실제 하드웨어에서도 매우 빠르게 동작함!

EfficientNet에서의 전이학습 데이터 결과

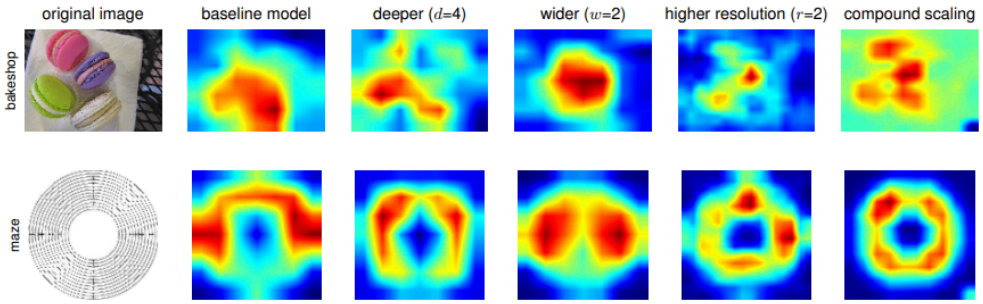


Figure 7. **Class Activation Map (CAM) (Zhou et al., 2016) for Models with different scaling methods-** Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

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

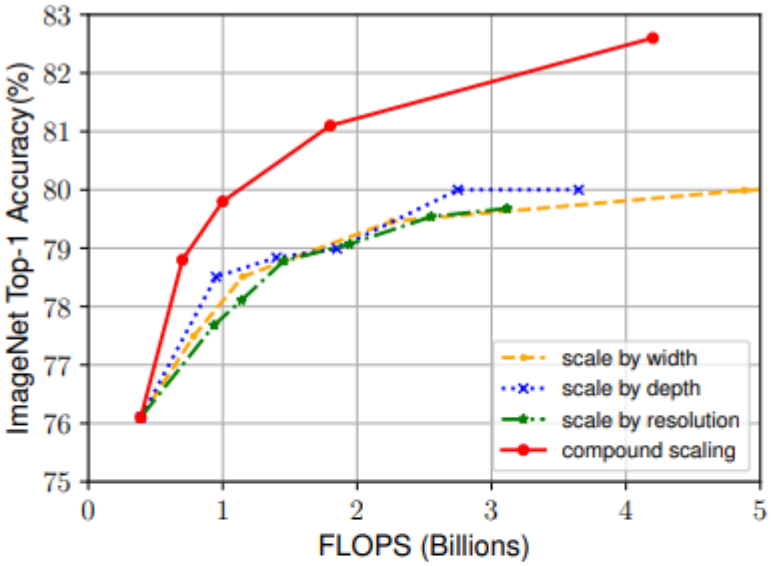


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

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

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

- 널리 사용되는 전이학습 데이터를 활용해 평가 진행
- 학습 설정은 Kornblith et al, 2019와 Huang et al, 2018을 따르고, 이미지넷의 미리 학습된 체스포인트를 가져와서 새 데이터셋에 맞게 파인튜닝(미세 조정)을 진행하였음
- 성능
  - NASNET-A(Zoph et al, 2018), Inception-v4(Szegedy et al, 2017)과 비교 시, EfficientNet은 평균적으로 4.7배 적은 매개변수(최대 21배감소)로 더 높은 정확도 달성
  - 학습 데이터를 동적으로 합성하는 기법을 사용한 DAT(Ngiam et al, 2018)과 파이프라인 병렬화를 활용한 대형 모델인 GPipe(Huang et al, 2018)과 비교하자면, EfficientNEt은 8개의 데이터셋 중 5개에서 최고 정확도를 기록하였고, 9.6배 더 적은 파라미터를 사용함.
- 복합 스케일링 모델은 더 관련성 높은 영역을 집중적으로 학습해 더 많은 객체의 세부 정보를 캡처하는 경향이 있었으나, 단일 차원 스케일링 모델들은 세부 정보가 부족하고, 이

미지 내 모든 객체를 완전히 포착하지 못하는 문제가 발생함

결론

- 신경망의 너비, 깊이, 해상도를 균형 있게 조정하는 것이 중요하나, 기존 연구에서는 간과됨
- 단순하면서도 효과적인 복합 스케일링(compound scaling) 기법을 통해, CNN 모델이 제한된 자원 하에서 효율적으로 확장하고, 모델의 효율성을 유지할 수 있음
- Compound Scaling이 적용된 EfficientNet 모델은 모바일 크기의 작은 모델들에서도 효과적으로 확장이 가능하고, 정확도도 높고, 파라미터 수와 연산량(FLOPS)가 기존 모델 대비 10배 이상 감소했으며, 5개의 전이 데이터셋과 ImageNet에서 우수한 성능을 입증함.