EfficientNet

Abstract

- CNN의 depth, width, resolution을 수학적인 공식을 통해 균형있게 증가시켜 더 나은 성능을 이끌 수 있도록 연구하였다.
- 본 논문은 기존 모델보다 사이즈가 작으며 더 빠른 `EfficientNet` 모델을 제안하였다.

Introduction

ConvNets의 scale을 키우는 방법은 정확히 이해되어 연구되지 않았다. 가장 흔한 방법은 ConvNets의 `depth` (깊이, 레이어 개수), `width` (각 레이어의 파라미터 수), `resolution` (인풋 이미지 해상도)을 증가시켜 scale을 증가시키는 것이었다. 본 논문에서는 scale up 하는 방법에 대해 다시 생각해보았고 "어떻게 scale up 해야 정확도(accuracy)와 효율성(efficiency)을 증가시킬 수 있을까?"에 대한 연구를 진행하였다.

`width/depth/resolution` 을 특정한 상수 비율로 균형있게 scaling up 하면 유의미한 결과를 보여준다는 것을 관측했고, 세 가지 차원인 `width` , `depth` , `resolution` 에 대해 양적 관계를 정립(quantify)하였다.

Compound Model Scaling

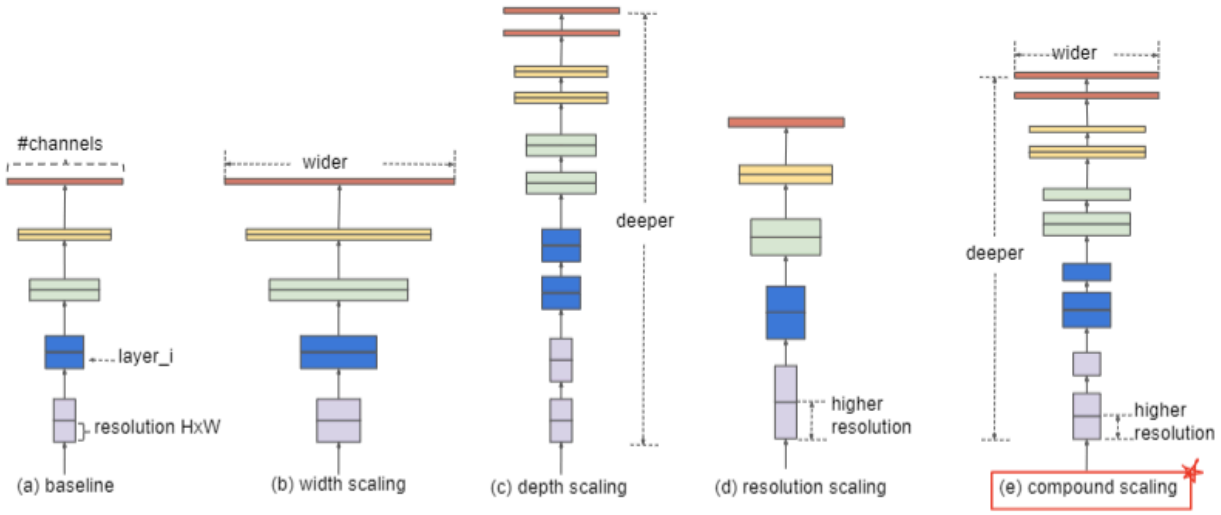


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Problem Formulation

$$N = F_k \odot \cdots \odot F_2 \odot F_1(X_1) = \bigodot_{j=1 \ldots k} F_j(X_1)$$

N은 `ConvNet` 을 의미하고 F_i 는 `operator` X_i는 `input tensor` 를 의미한다. 위 수식을 CNN 레이어의 `Height` , `Width` , `Channel` 로 다시 정의를 하면 아래와 같다.

$$N = \bigodot_{i=1 \dots s} F_i^{L_i}(X_{<H_i, W_i, C_i>})$$

$F_i^{L_i}$는 layer F_i가 stage i일 때 L_i번 반복된다는 것을 의미한다. 본 논문에서는 자원에 규제를 가해 `width`, `depth`, `resolution`의 수치를 조절하는 계수 값을 최적화하기 위해 아래 수식을 도입하였다.

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

$$s.t. \quad N(d,w,r) = \bigodot_{i=1 \dots s} \hat{F}_i^{d \cdot \hat{L}i}(X_{<r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i>})$$

$$Memory(N) \leq target\_memory$$

$$FLOPS(N) \leq target\_flops$$

w, d, r은 `width`, `depth`, `resolution`을 scaling 하기 위한 계수이며 최적의 w, d, r 값을 찾는 것이 목표이다.

Scaling Dimensions

Depth (d): 모델의 깊이가 깊어질수록 `vanishing gradient`로 인해 학습하기 어려워지는 경향이 있으며, `batch normalization`등의 기술로 해결하여도 정확도 상승의 증가 폭이 낮아지는 경향을 보인다. 즉, 모델의 깊이가 깊어질수록 깊어짐에 따른 성능의 향상 폭이 감소한다.

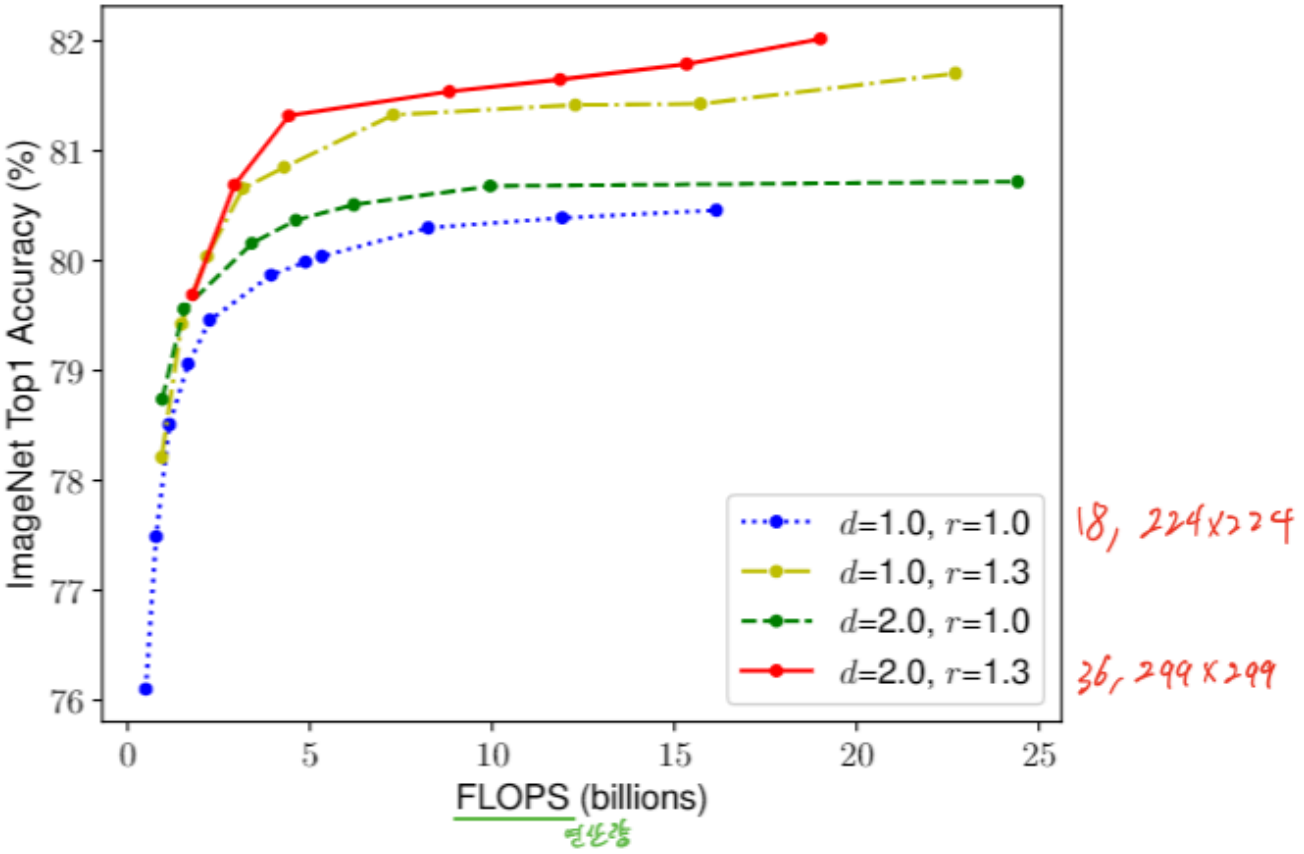
Width(w): 극히 wide 하지만 깊이가 얕은 모델은 `higher level features`를 포착하기 힘들다.

Resolution(r): 고해상도 이미지의 입력 텐서일수록 `fine-grained` 패턴을 더욱 잘 포착하는 경향이 있다. 그러나, 해상도의 크기가 극히 높아질수록 성능 향상의 폭이 감소한다.

> **관측 1 -** `width`, `depth`, `resolution` 중 하나의 scale을 키우는 것은 정확도 향상을 불러오나, scale이 커질수록 정확도 향상의 폭이 감소한다.

Compound Scaling

위 관측으로 `width`, `depth`, `resolution` 각 차원은 서로 독립적이지 않은(not independent) 것을 알아냈다. 각 차원의 scaling은 한 차원만 키우는 것이 아니라 균형있게 키워야 할 필요가 있다.



해당 가설을 검증하기 위해 18개의 convolutional layer와 224x224 resolution을 가진 `baseline network` ($d=1.0, r=1.0$)의 d, r 수치를 조절하며 실험해본 결과 하나의 차원의 scale만 증가시킨 것 보다 깊고(d) 고해상도(r)인 모델이 같은 연산량 수준에서 더 높은 정확도를 보이었다.

> **관측 2 -** `width`, `depth`, `resolution` 모든 차원의 균형을 유지하는 것이 중요하다.

본 논문에서는, 새로운 **compound scaling method**를 제안하였다.

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

α, β, γ는 `small grid search`에 의해 결정되는 상수 값이며 ϕ는 자원 제약을 컨트롤할 수 있는 계수이다. β^2, γ^2 인 이유는 β, γ가 증가하면 **FLOPS**가 제곱으로 늘어나기 때문이다. 위 수식에 따르면, ϕ의 수치에 따라 연산량 **FLOPS**가 약 2^ϕ에 근접한다.

EfficientNet Architecture

본 논문에서 `EfficientNet-B0`의 `baseline network`를 구성하였다.

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

(`EfficientNet-B0` 이 만들어지게 된 과정은 본 논문을 참고하자)

`EfficientNet-B0` 을 이용하여 아래의 두 과정을 거쳐 scaling method를 시행한다.

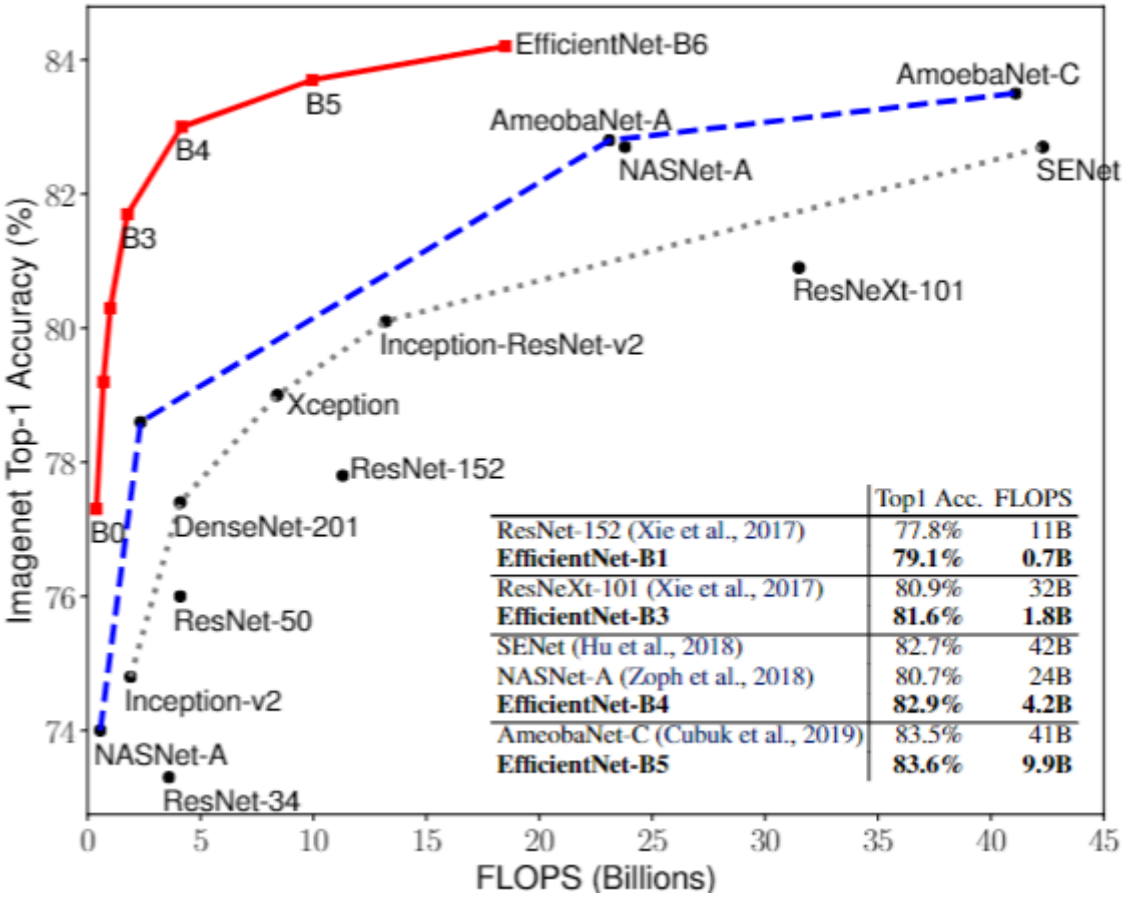
1. $\phi = 1$ 로 고정한 후 α, β, γ에 대해 앞서 소개한 optimizaion 수식을 통해 `small grid search` 를 수행한다. `EfficientNet-B0` 에 가장 적합한 수치는 $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$ 로 나타났다.
2. 결정된 α, β, γ의 값을 고정한 후 optimization 수식을 통해 ϕ을 조절하여 scale up을 수행하였다. 해당 과정으로 `EfficientNet-B1` 부터 `B7` 까지 도출하였다.

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).



	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**

`EfficientNet` 모델은 타 모델에 비해 적은 연산량 및 적은 파라미터로 높은 정확도를 거두었다.

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).

위 표를 보면, CIFAR-100의 경우 `Gpipe` 모델은 $556M$의 파라미터를 가진 반면에 `EfficientNet-B7` 모델은 $64M$의 파라미터를 가지고도 비슷한 정확도 수치를 달성하였다. 다른 데이터 셋에 대해서도 비슷한 양상을 보이었다.

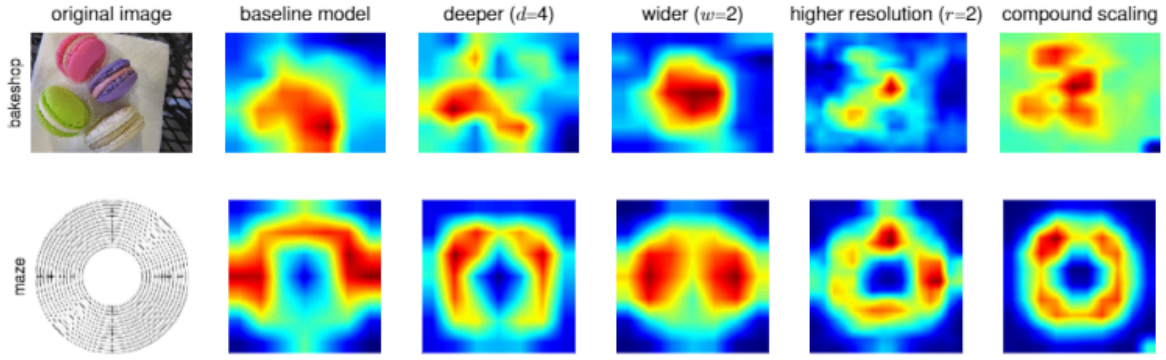


Figure 7. **Class Activation Map (CAM) (Zhou et al., 2016) for Models with different scaling methods-** Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

Grad CAM을 통해 분석한 결과, `compound scaling`을 한 모델의 경우 더욱 대상과 관련된 영역을 집중하는 것을 확인할 수 있다.

`EfficientNet`은 전이 학습(Transfer Learning) 결과도 타 모델에 비해 좋은 성능을 보였다.

Discussion & Conclusion

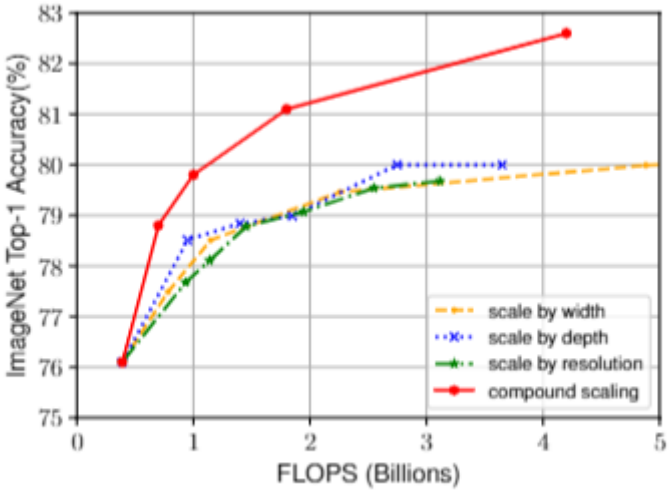


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

`compound scaling`을 적용하면 같은 연산량의 수준에서 하나의 차원만을 `scaling`한 모델에 비해 더 높은 정확도를 보였다. 위 그래프를 보았을 때, `compound scaling`이 다른 `scale`방법에 비해 정확도 증가율 감소가 적은 수치로 나타났다.

본 논문을 통해 `width, depth, resolution`의 균형있는 관계가 중요한 것으로 드러났고 `compound scaling` 방법을 통해 작은 사이즈의 모델에 대해 효과적으로 `scale`을 수행할 수 있다는 것을 입증하였다.