



[논문리뷰] EfficientNet [2019]

Introduction

✨ 해당 논문은 연구에 따른 new scaling method을 적용한 **EfficientNet**을 소개합니다.

논문에서는 Model을 scaling하고 Network의 Depth, width, resolution의 조절한다면, better performance를 낸다고 말합니다.

이러한 방법은, MobileNet, ResNet을 통하여 증명되었으며, **단순하고, 뛰어난 compound coefficient**를 사용하여, 깊이/높이/해상도의 모든 차원을 균일하게 확장합니다.

지금까지 ConvNets을 scaling up하는 과정은 제대로 이해되지 않았으며, 일반적인 방법으로 깊이, 차원, 이미지의 크기 중에서 한가지만 scale up하였습니다.

2~3차원으로 임의로 확장하는 것이 가능할지라도, 지루한 수동 조정이 필요하며, 차선의 정확도와 효율성을 나타내는 경우가 많습니다.



논문에서는 ConvNets의 확장 프로세스를 연구하고 재고하고 있습니다.

정확성과 효율성을 달성할 수 있는 일반화된 ConvNet ScaleUp 방법에 대한 질문을 통하여 연구하였을 때,

Network의 **Depth/width/resolution**의 균형을 맞추는 것이 매우 중요하다는 것을 보여주었으며, 그러한 균형은 **차원을 일정한 비율로 확장**하기만 하면 달성할 수 있다고 보았다.

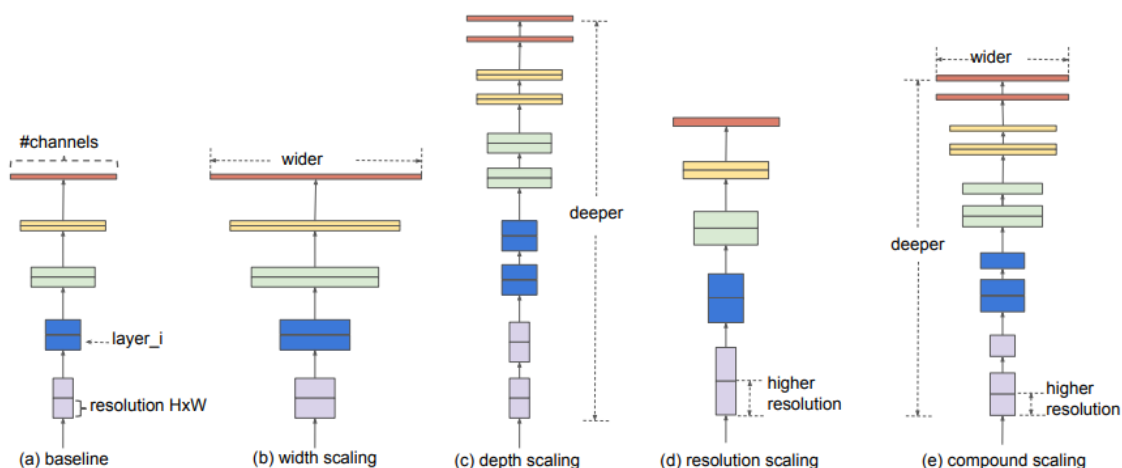


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

임의적으로 Scaling하는 관행과는 다르게, 우리의 방법은 일반적으로 너비, 깊이, 해상도를 고정된 계수를 사용하여, Scaling합니다.

[A는 CNN BaseModel이며, B~D는 관행적으로 이루어진 Scaling 방식이고, E는 새로운 방식이다.]

만약, 우리가 2^N 많은 Computational resource를 사용한다면,

단순히 $a^N(\text{depth})B^N(\text{width})r^N(\text{imagesize})$ 를 조절하면 됩니다.

a, B, r 는 Original small model내의 small grid에 의하여 결정나는 constant coefficients입니다.

Compound Model Scaling

$$Y_i = F_i(X_i)$$

X_i : input tensor

X_i shape : $\langle H_i, W_i, C_i \rangle$

- H_i, W_i : spatial dimension
- C_i : channel dimension

$$\begin{aligned} \max_{d,w,r} \quad & \text{Accuracy}(\mathcal{N}(d, w, r)) \\ \text{s.t.} \quad & \mathcal{N}(d, w, r) = \bigodot_{i=1 \dots s} \hat{\mathcal{F}}_i^{d \cdot \hat{L}_i}(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle}) \\ & \text{Memory}(\mathcal{N}) \leq \text{target_memory} \\ & \text{FLOPS}(\mathcal{N}) \leq \text{target_flops} \end{aligned} \quad (2)$$

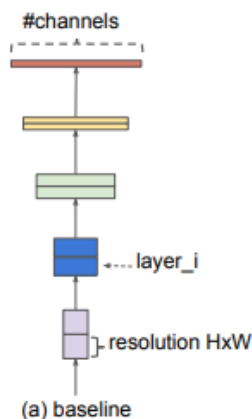
where w, d, r are coefficients for scaling network width, depth, and resolution; $\hat{\mathcal{F}}_i, \hat{L}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i$ are predefined parameters in baseline network (see Table 1 as an example).

$$\mathcal{N} = F_k \odot \dots \odot F_2 \odot F_1(X_1) = \bigodot_{j=1, \dots, k} F_j(X_1)$$

ConvNet은 보통 multiple stage로 분할되고 각 stage의 모든 layer들은 같은 구조임

따라서 ConvNet은

$$\mathcal{N} = \bigodot_{i=1 \dots s} F_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle}) \text{로 표현할 수 있음}$$



representative ConvNet

Spatial dimension은 줄어들고, channel dimension은 늘어나는 것을 알 수 있음

ex) $\langle 224, 224, 3 \rangle \rightarrow \langle 7, 7, 512 \rangle$

이전의 ConvNet은 F를 찾는데 중점인 것과는 다르게,

Model Scaling은 F의 변화가 없이, Width, Length, Resolution(H,W)를 확장하는데 중점을 둡니다.

이때, **Width, Length, Resolution의 Scale**을 찾을 때, 제각각의 **Scale**을 사용하면, 상당히 오랜 시간이 걸리므로, **Constant Ratio** 즉 일정한 비율을 모든 layer에 적용하여 제한합니다.

즉, 주어진 제약 조건 (fixed f , Constant Ratio)에서 Model Accuracy를 최대화하는게 목표입니다.

Scaling Dimension

Depth, Width, Resolution이 각각에 의존하기에 해결에 어려움이 있어, 이전 연구에서는 한 가지의 Dimension만을 변경해왔습니다.

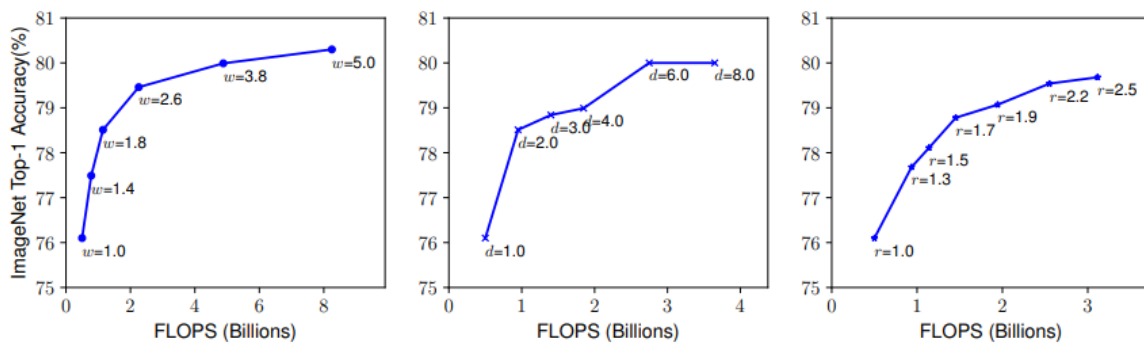


Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single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

Compound Scaling

다양한 실험을 통하여, **Width, Depth, Resolution**간의 균형을 맞추는 것이 중요함을 보여줍니다.

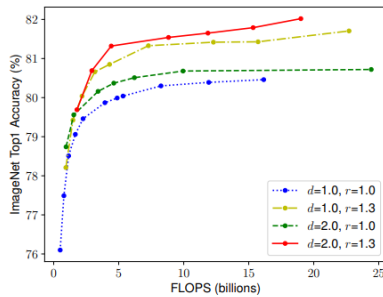


Figure 4. **Scaling Network Width for Different Baseline Networks.** Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network ($d=1.0, r=1.0$) has 18 convolutional layers with resolution 224x224, while the last baseline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

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:

$$\begin{aligned}
 \text{depth: } d &= \alpha^\phi \\
 \text{width: } w &= \beta^\phi \\
 \text{resolution: } r &= \gamma^\phi
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
 \alpha &\geq 1, \beta \geq 1, \gamma \geq 1
 \end{aligned}$$

Compound Coefficient로 Width, Depth, Resolution를 균일하게 Scale하는 방법을 보여줍니다.

Compound Coefficients의 경우 사용자가 사용가능한 만큼 지정하면 됩니다.

일반적인 Convolution연산에서는 FLOP은 d, w^2, r^2 의 비율로 계산을 합니다.

$FLOP = (d, w^2, r^2)^\phi$ 로 계산되며, $d * w^2 * r^2 \approx 2$ 의 조건에 의해 $FLOP$ 이 대략 2^ϕ 증가합니다.

EfficientNet Architecture



Scaling method을 활용해서, EfficientNet의 effectiveness를 증명할 것입니다.

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

Accuracy와 FLOPS를 최적화하는 Multi-Objective neural Architecture로부터 영감을 받아, Base Network을 개발하였습니다.

EfficientNet-B0의 주요 Block은 mobile inverted bottleNeck인 MBConv2로 생성되었으며, Squeeze-and-excitation optimization을 사용하였습니다.

EfficientNet-B0을 기초로 시작하여, Compound Scaling 방법을 2단계 거쳐 적용하였습니다.

- **STEP1**: ϕ 를 1로 고정한 후, 제한 조건 내에서 GridSearch를 활용하여, D, W, R 를 구합니다.

EfficientNet-B0의 경우 $D = 1.2$, $W = 1.1$, $R = 1.15$ 가 최적의 값이었다.

- **STEP2**: D, W, R 을 상수로 고정하고, ϕ 를 증가시키면서, 네트워크의 scale을 확장하면서 **EfficientNet B1~B7까지의 모델**을 얻었습니다.

Experiments

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

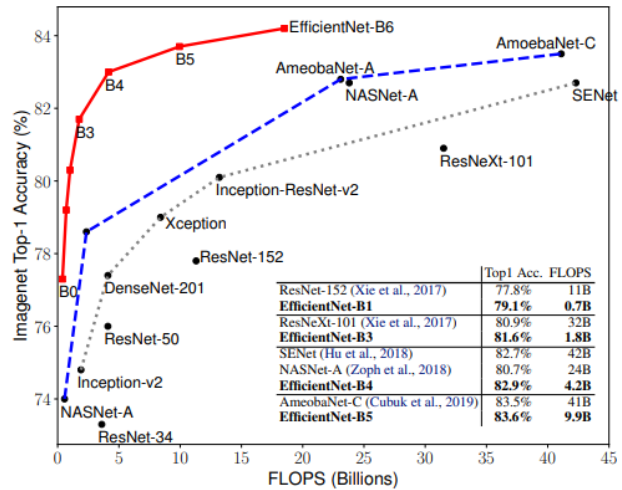


Figure 5. FLOPS vs. ImageNet Accuracy – Similar to Figure 1 except it compares FLOPS rather than model size.

Conclusion


본 논문에서는 체계적으로 ConvNet을 확장 및 신중하게 width, depth, resolution의 identity를 균형을 조절합니다.

Compound Scaling Method의 Powered에 의하여, EfficientNet의 effectively를 증명하였습니다.

Reference

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that

 <https://arxiv.org/abs/1905.11946>

arXiv