EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

1. Introduction

- ConvNet의 크기를 키우는 것은 널리 쓰이는 방법이다.
- 그러나 제대로 된 이해를 바탕으로 이루어지지는 않았던 것 같다.
- 그래서 scaling하는 방법을 다시 한 번 생각해보고 연구하는 논문을 제안한다.
  - 그 방법이 *compound scaling method*이다.
- 이 방법을 MobileNets와 ResNet에서 검증해보고자 한다.
  - 그림 1이 결과를 보여주고 있다.

---

2. Related Work

ConvNet Accuracy

AlexNet 이후 ImageNet competition에서 더 깊어지고 커지면서 정확도가 높아지는 모델들이 여럿 발표되었다. 최근 발표되는 모델들은 ImageNet뿐만 아니라 다른 데이터셋에서도 잘 작동한다. 그러나 정확도는 높아졌지만, 사용하는 자원 역시 크게 늘어났다.

ConvNet Efficiency

깊은 ConvNets는 좀좀 over-parameterized된다. 효율을 높이기 위해 모델 압축하는 여러 기법이 제안되었다: SqueezeNets, MobileNets, ShuffleNets 등.

Model Scaling

- ResNet(ResNet-18, ResNet-50, ResNet-200)은 깊이를 달리 하였다.
- MobileNets는 network width를 달리 하였다.
- 또한 이미지 해상도가 높아지면 (찾아낼 정보가 많아서) 정확도를 높아진다. (물론 계산량도 많이 늘어난다.)

많은 연구가 진행되었으나 어떻게 효율적인 조합을 찾는지는 아직까지 정립되지 않았다.

---

3. Compound Model Scaling

3.1. Problem Formulation

뭔가 괜히 복잡하게 써 놨는데 그냥 ConvNet을 수식화해 정리해놓은 부분이다. H,W,C$_i$,$_i$,$_i$를 입력 tensor의 크기, F$_i$를 Conv layer라 하면 ConvNet은

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

로 표현 가능하다.

모델이 사용하는 자원이 제한된 상태에서 모델의 정확도를 최대화하는 문제를 풀고자 하는 것이므로, 이 문제는 다음과 같이 정리할 수 있다.

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

3.2. Scaling Dimensions

- **Depth**: 네트워크의 깊이가 증가할수록 모델의 capacity가 커지고 더 복잡한 feature를 잡아낼 수 있지만, vanishing gradient의 문제로 학습시키기가 더 어려워진다. 이를 해결하기 위해 Batch Norm, Residual Connection 등의 여러 기법들이 등장하였다.
- **Width**: 각 레이어의 width를 키우면 정확도가 높아지지만 계산량이 제곱에 비례하여 증가한다.
- **Resolution**: 입력 이미지의 해상도를 키우면 더 세부적인 feature를 학습할 수 있어 정확도가 높아지지만 마찬가지로 계산량이 제곱에 비례해 증가한다.

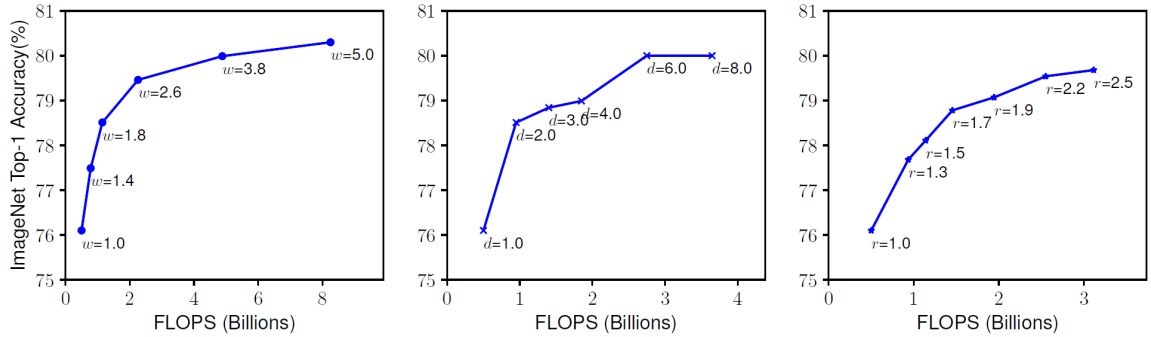

Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single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

공통적으로, 어느 정도 이상 증가하면 모델의 크기가 커짐에 따라 얻는 정확도 증가량이 매우 적어진다.

3.3. Compound Scaling

직관적으로, 더 높은 해상도의 이미지에 대해서는,

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

즉, 이 depth, width, resolution이라는 세 가지 변수는 밀접하게 연관되어 있으며, 이를 같이 움직이는 것이 도움이 될 것이라고 생각할 수 있다.

계산량은 깊이에 비례하고, 나머지 두 변수에 대해서 그 제곱에 비례하므로 다음과 같은 비율로 변수들이 움직이게 정할 수 있다.

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

이 논문에서는 α·β2·γ2≈2�·�2·�2≈2로 맞춰서 전체 계산량은 2ϕ2�에 비례하게 잡았다.

4. EfficientNet Architecture

MnasNet에 기반한 baseline network를 사용한다. 구체적인 모양은 다음과 같다.

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

이 baseline network에 기반해서 시작한다.

5. Experiments

5.1. Scaling Up MobileNets and ResNets

결과부터 보자.

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).

Efficient하다.

depth, width, resolution을 어떻게 늘리는지에 대한 비교도 진행해 보았다. 섹션 3의 직관적인 설명과 같은 결과를 보이고 있다.

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%**

5.2. ImageNet Results for EfficientNet

추론 latency에 대한 결과를 기록해 놓았다.

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**

8.4배 적은 연산량으로 더 높은 정확도를 갖는다는 것은 꽤 고무적이다. 결과에 따라서 18배 적거나, 아니면 5.7배, 6.1배 더 빠른 추론 시간을 보여주기도 한다. (표 1, 5 등)

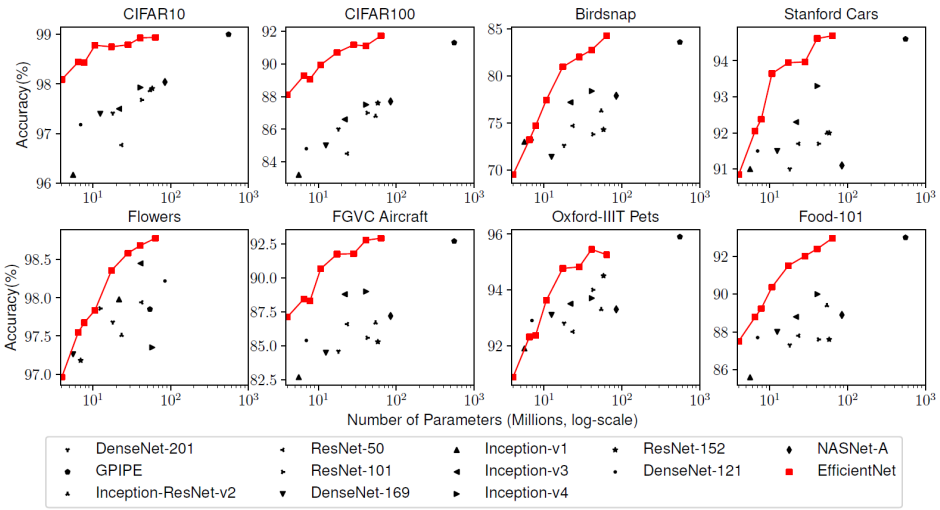

Figure 6. **Model Parameters vs. Transfer Learning Accuracy –** All models are pretrained on ImageNet and finetuned on new datasets.

5.3. Transfer Learning Results for EfficientNet

전이학습 dataset에 대한 결과를 기록해 놓았다.

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).

여기도 비슷하게 몇 배 더 작고 적은 연산량으로 더 좋은 정확도를 갖는다는 내용이다.

데이터셋에 대한 정보이다.

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

baseline 모델에 대해서 어떻게 scaling을 할지를 테스트해 보았다. 표 3과 같은 결과를 보여준다.

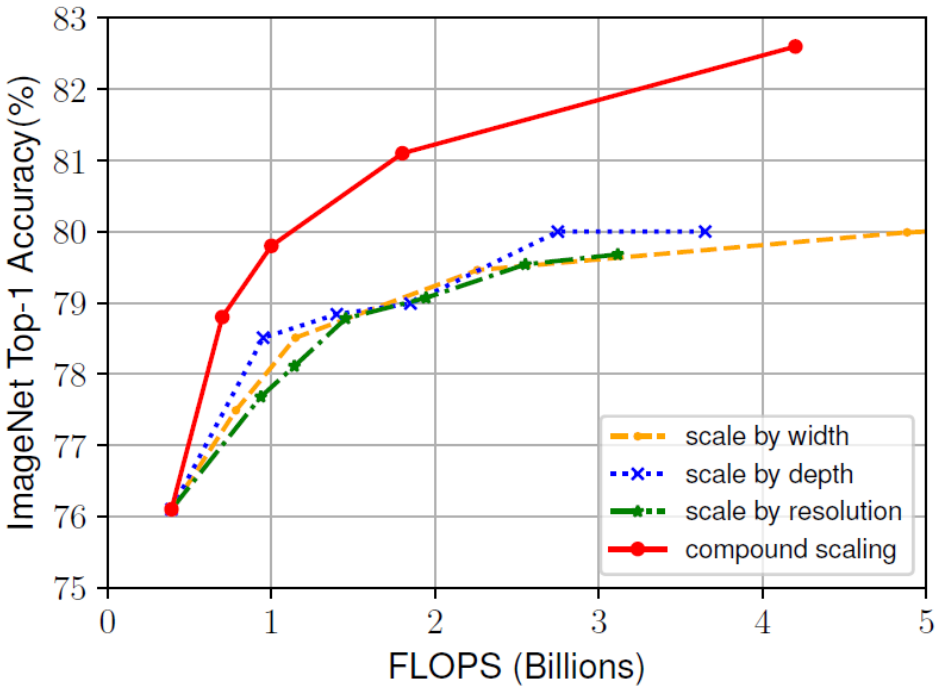

Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

6. Discussion

어떻게 scaling을 해야 하는지 아래 그림이 단적으로 보여준다. depth, width, resolution은 서로 긴밀히 연관되어 있으며 이들을 같이 키우는 것이 자원을 더 효율적으로 쓰는 방법이

다.

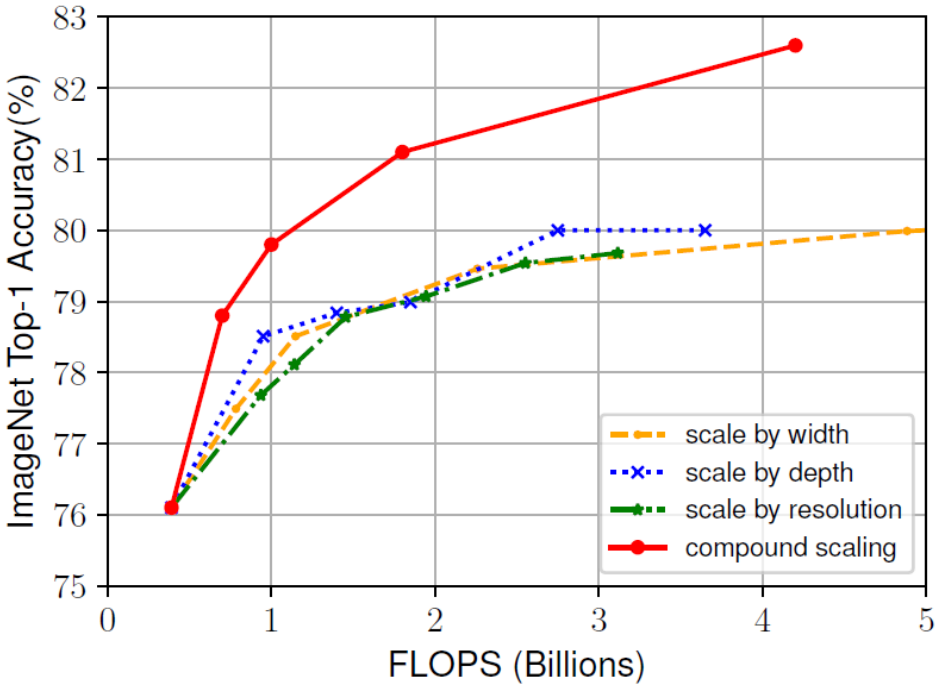

Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

어쨔서 compound scaling method라 다른 방법에 비해 더 좋은지를 나타내는 그림이 아래에 있다. 이미지의 어디에 집중하고 있는지를 보여준다. (근데 attention을 딱히 적용하진 않았다.)

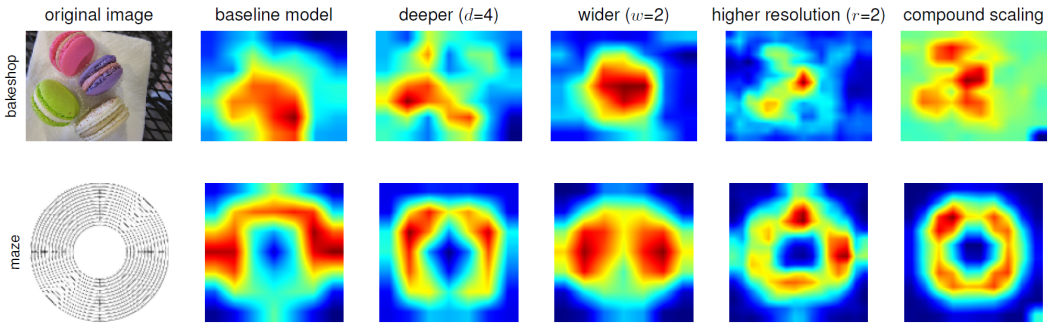

Figure 7. **Class Activation Map (CAM) (Zhou et al., 2016) for Models with different scaling methods-** Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

Table 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%**

7. Conclusion

한정된 자원을 갖고 있는 상황에서 Depth, Width, Resolution을 어떻게 적절히 조절하여 모델의 크기와 연산량을 줄이면서도 성능은 높일 수 있는지에 대한 연구를 훌륭하게 수행하였다.