**EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**

Mingxing Tan, Qunc V.Le

**Abstract**

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 carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. We demonstrate the effectiveness of this method on scaling up MobileNets and ResNet.

Convolution Neural Network(ConvNets)는 일반적으로 고정된 자원 예산으로 개발되며, 더 많은 자원을 사용할 수 있는 경우 더 나은 정확도를 위해 확장됩니다. 이 백서에서는 모델 스케일링을 체계적으로 연구하여 네트워크의 깊이, 폭 및 해상도를 신중하게 균형 있게 조정함으로써 성능을 향상시킬 수 있음을 확인합니다. 이러한 관찰을 바탕으로 단순하지만 매우 효과적인 복합 계수를 사용하여 깊이/폭/분해능의 모든 차원을 균일하게 스케일링하는 새로운 스케일링 방법을 제안한다. 우리는 MobileNet과 ResNet의 스케일업에서 이 방법의 효과를 입증한다.

To go even further, we use neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets. In particular, our EfficientNet-B7 achieves state-of-the-art 84.3% top-1 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet. Our EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters. Source code is at https: //github.com/tensorflow/tpu/tree/master/models/official/efficientnet.

더 나아가 신경 아키텍처 검색을 사용하여 새로운 베이스라인 네트워크를 설계하고 확장하여 EfficientNets라 불리는 모델 패밀리를 얻습니다.이 모델 패밀리는 이전 ConvNets보다 훨씬 더 높은 정확성과 효율성을 달성합니다. 특히 EfficientNet-B7은 기존 ConvNet보다 8.4배 작고 추론 속도가 6.1배 빠르며 ImageNet에서 최고 84.3%의 정확도를 달성했습니다. 또한 EfficientNets는 CIFAR-100(91.7%), Flowers(98.8%), 기타 3가지 전송 학습 데이터셋에서 전송이 잘 되고 최신 정확도를 달성하며 매개 변수 수는 감소한다. 소스 코드는 <https://github.com/tensorflow/tpu/tree/> master/models/official/efficientnet에 있습니다.

1. **Introduction**

Scaling up ConvNets is widely used to achieve better accuracy. For example, ResNet (He et al., 2016) can be scaled up from ResNet-18 to ResNet-200 by using more layers; Recently, GPipe (Huang et al., 2018) achieved 84.3% ImageNet top-1 accuracy by scaling up a baseline model four time larger. However, the process of scaling up ConvNets has never been well understood and there are currently many ways to do it. The most common way is to scale up ConvNets by their depth (He et al., 2016) or width (Zagoruyko & Komodakis, 2016). Another less common, but increasingly popular, method is to scale up models by image resolution (Huang et al., 2018). In previous work, it is common to scale only one of the three dimensions – depth, width, and image size. Though it is possible to scale two or three dimensions arbitrarily, arbitrary scaling requires tedious manual tuning and still often yields sub-optimal accuracy and efficiency.

ConvNets의 스케일 업은 정확성을 높이기 위해 널리 사용되고 있습니다. 예를 들어 ResNet(He 등, 2016)은 더 많은 계층을 사용하여 ResNet-18에서 ResNet-200으로 확장할 수 있습니다. 최근에는 GPipe(Huang 등, 2018)가 기준 모델을 4배 더 크게 확장하여 84.3%의 ImageNet 상위 1 정확도를 달성했습니다. 그러나 ConvNets를 확장하는 과정은 잘 이해되지 않았으며 현재 이를 위한 많은 방법이 있습니다. 가장 일반적인 방법은 ConvNets의 깊이(He et al., 2016) 또는 폭(Zagoruyko & Komodakis, 2016)으로 스케일업하는 것이다. 일반적이지 않지만 점점 더 인기를 끌고 있는 또 다른 방법은 이미지 해상도에 따라 모델을 스케일업하는 것입니다(Huang et al. 이전 작업에서는 깊이, 폭 및 이미지 크기 중 하나의 치수만 축척하는 것이 일반적이었습니다. 2차원 또는 3차원을 임의로 확장할 수 있지만, 임의 스케일링에는 지루한 수동 조정이 필요하며, 여전히 최적의 정확성과 효율성이 떨어지는 경우가 많습니다.

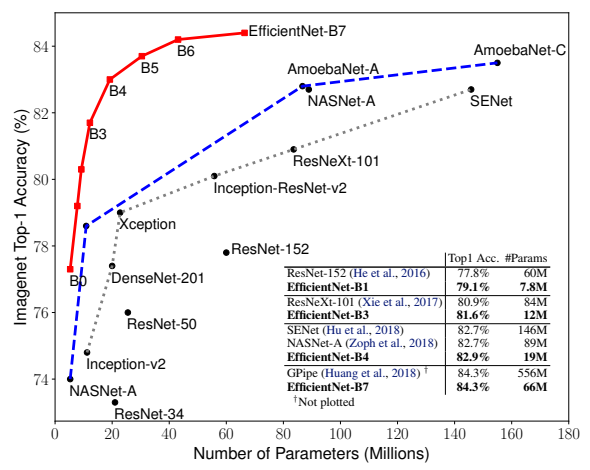


Figure 1. **Model Size vs. ImageNet Accuracy.** All numbers are for single-crop, single-model. Our EfficientNets significantly outperform other ConvNets. In particular, EfficientNet-B7 achieves new state-of-the-art 84.3% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

그림 1 모델 사이즈와 ImageNet 정밀도 모든 번호는 싱글 크롭, 싱글 모델용입니다. NAT의 EfficientNets는 다른 ConvNets를 크게 능가합니다. 특히, EfficientNet-B7은 84.3%의 새로운 최첨단 정확도를 달성하지만 GPipe보다 8.4배 작고 6.1배 더 빠릅니다. EfficientNet-B1은 ResNet-152보다 7.6배 작고 5.7배 빠릅니다. 자세한 내용은 표 2와 표 4에 있습니다.

In this paper, we want to study and rethink the process of scaling up ConvNets. In particular, we investigate the central question: is there a principled method to scale up ConvNets that can achieve better accuracy and efficiency? Our empirical study shows that it is critical to balance all dimensions of network width/depth/resolution, and surprisingly such balance can be achieved by simply scaling each of them with constant ratio. Based on this observation, we propose a simple yet effective compound scaling method. Unlike conventional practice that arbitrary scales these factors, our method uniformly scales network width, depth,and resolution with a set of fixed scaling coefficients. For example, if we want to use 2N times more computational resources, then we can simply increase the network depth by αN , width by βN , and image size by γN , where α, β, γ are constant coefficients determined by a small grid search on the original small model. Figure 2 illustrates the difference between our scaling method and conventional methods.

이 백서에서는 ConvNets의 스케일업 프로세스를 검토하고 다시 생각해 보고자 합니다. 특히, 우리는 핵심 질문을 조사합니다: 더 나은 정확성과 효율성을 달성할 수 있는 ConvNet을 확장하는 원칙적인 방법이 있습니까? 우리의 경험적 연구에 따르면 네트워크 폭/깊이/해상도의 모든 차원을 균형 있게 조정하는 것이 매우 중요하며, 놀랍게도 이러한 균형은 각각의 차원을 일정한 비율로 스케일링하는 것만으로 달성할 수 있다. 이러한 관찰을 바탕으로 단순하지만 효과적인 복합 스케일링 방법을 제안한다. 이러한 요인을 임의로 스케일링하는 기존의 관행과는 달리, 당사의 방법은 일련의 고정 스케일링 계수를 사용하여 네트워크 폭, 깊이 및 해상도를 균일하게 스케일링합니다. 예를 들어 2N배의 계산 자원을 사용하는 경우 네트워크 깊이를 αN, 폭을 βN, 이미지 크기를 βN만 늘리면 됩니다.여기서 α, β, θ는 원래 소형 모델에서의 작은 그리드 검색에 의해 결정되는 상수 계수입니다. 그림 2는 스케일링 방법과 기존 방법의 차이를 보여줍니다.

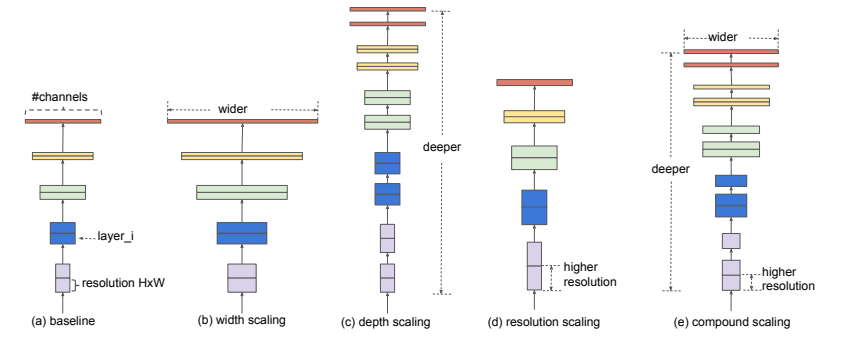


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

그림 2 모델 스케일링. (a)는 베이스라인 네트워크의 예시입니다. (b)-(d)는 네트워크 폭, 깊이 또는 분해능의 1차원만 증가시키는 일반적인 스케일링입니다. (e) 3차원 모두를 고정 비율로 균일하게 스케일링하는 제안 복합 스케일링 방법입니다.

Intuitively, the compound scaling method makes sense because if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image. In fact, previous theoretical (Raghu et al., 2017; Lu et al., 2018) and empirical results (Zagoruyko & Komodakis, 2016) both show that there exists certain relationship between network width and depth, but to our best knowledge, we are the first to empirically quantify the relationship among all three dimensions of network width, depth, and resolution.

직관적으로 복합 스케일링 방법은 입력 이미지가 클수록 네트워크는 수신 필드를 증가시키기 위해 더 많은 레이어를 필요로 하고 더 큰 이미지에서 더 세밀한 패턴을 캡처하기 위해 더 많은 채널을 필요로 하기 때문에 의미가 있습니다. 실제로 이전의 이론(Raghu et al., 2017; Lu et al., 2018)과 경험적 결과(Zagoruyko & Komodakis, 2016)는 모두 네트워크 폭과 깊이 사이에 일정한 관계가 있음을 보여주지만, 우리가 아는 한 네트워크 폭, 깊이, 분해능의 3차원 간의 관계를 경험적으로 수치화한 것은 처음이다.동작.

We demonstrate that our scaling method work well on existing MobileNets (Howard et al., 2017; Sandler et al., 2018) and ResNet (He et al., 2016). Notably, the effectiveness of model scaling heavily depends on the baseline network; to go even further, we use neural architecture search (Zoph & Le, 2017; Tan et al., 2019) to develop a new baseline network, and scale it up to obtain a family of models, called EfficientNets. Figure 1 summarizes the ImageNet performance, where our EfficientNets significantly outperform other ConvNets. In particular, our EfficientNet-B7 surpasses the best existing GPipe accuracy (Huang et al., 2018), but using 8.4x fewer parameters and running 6.1x faster on inference. Compared to the widely used ResNet-50 (He et al., 2016), our EfficientNet-B4 improves the top-1 accuracy from 76.3% to 83.0% (+6.7%) with similar FLOPS. Besides ImageNet, EfficientNets also transfer well and achieve state-of-the-art accuracy on 5 out of 8 widely used datasets, while reducing parameters by up to 21x than existing ConvNets.

스케일링 방법이 기존 MobileNets(Howard 등, 2017년, Sandler 등, 2018년) 및 ResNet(He 등, 2016년)에서 잘 작동함을 입증한다. 특히 모델 스케일링의 효과는 기준선 네트워크에 따라 크게 달라진다. 더 나아가 신경 아키텍처 검색(Zoph & Le, 2017; Tan et al., 2019)을 사용하여 새로운 기준선 네트워크를 개발하고 이를 확장하여 EfficientNets라는 모델군을 얻는다. 그림 1은 EfficientNet이 다른 ConvNet을 크게 능가하는 ImageNet의 성능을 요약한 것입니다. 특히, NAT의 EfficientNet-B7은 기존 GPIpe 정확도(Huang et al., 2018)를 능가하지만, 8.4배 적은 매개 변수를 사용하고 6.1배 더 빠르게 추론할 수 있습니다. 널리 사용되는 ResNet-50(He 등, 2016)과 비교하여, EfficientNet-B4는 유사한 FLOPS로 상위 1의 정확도를 76.3%에서 83.0%(+6.7%)로 향상시킵니다. ImageNet 이외에도 EfficientNets는 널리 사용되는 8개의 데이터 세트 중 5개에서 전송이 잘 되고 최신 정확도를 달성하며 기존 ConvNet보다 매개변수를 최대 21배까지 줄입니다.

1. **Related Work**

**ConvNet Accuracy**: Since AlexNet (Krizhevsky et al., 2012) won the 2012 ImageNet competition, ConvNets have become increasingly more accurate by going bigger: while the 2014 ImageNet winner GoogleNet (Szegedy et al., 2015) achieves 74.8% top-1 accuracy with about 6.8M parameters, the 2017 ImageNet winner SENet (Hu et al., 2018) achieves 82.7% top-1 accuracy with 145M parameters. Recently, GPipe (Huang et al., 2018) further pushes the state-of-the-art ImageNet top-1 validation accuracy to 84.3% using 557M parameters: it is so big that it can only be trained with a specialized pipeline parallelism library by partitioning the network and spreading each part to a different accelerator. While these models are mainly designed for ImageNet, recent studies have shown better ImageNet models also perform better across a variety of transfer learning datasets (Kornblith et al., 2019), and other computer vision tasks such as object detection (He et al., 2016; Tan et al., 2019). Although higher accuracy is critical for many applications, we have already hit the hardware memory limit, and thus further accuracy gain needs better efficiency.

**ConvNet 정확도**: 2012년 ImageNet 경쟁에서 AlexNet(Krizhevski 등, 2012년)이 우승한 이후 ConvNet의 정확도는 더욱 커지고 있습니다.한편 2014년 ImageNet 수상자인 GoogleNet(Szegedy 등, 2015년)은 약 680만개의 파라미터로 74.8%의 정확도를 달성했으며, 2017년 ImageNet Ethu(Senet)는 상위 1위에 올랐다.M 파라미터 최근 GPipe(Huang et al., 2018)는 557M 파라미터를 사용하여 최첨단 ImageNet Top-1 검증 정확도를 84.3%까지 끌어올렸습니다. 네트워크를 분할하고 각 부품을 서로 다른 가속기로 분산하여 전문 파이프라인 병렬화 라이브러리를 통해서만 교육할 수 있을 정도로 규모가 큽니다. 이러한 모델은 주로 ImageNet용으로 설계되었지만, 최근 연구에 따르면 더 나은 ImageNet 모델은 다양한 전달 학습 데이터셋(Kornblith 등, 2019)과 물체 감지와 같은 기타 컴퓨터 비전 작업에서도 더 나은 성능을 발휘하는 것으로 나타났다(He 등, 2016; Tan 등, 2019). 많은 애플리케이션에서 높은 정확도가 중요하지만 하드웨어 메모리 제한에 이미 도달했기 때문에 정확도가 더욱 향상되어야 합니다.

**ConvNet Efficiency**: Deep ConvNets are often overparameterized. Model compression (Han et al., 2016; He et al., 2018; Yang et al., 2018) is a common way to reduce model size by trading accuracy for efficiency. As mobile phones become ubiquitous, it is also common to handcraft efficient mobile-size ConvNets, such as SqueezeNets (Iandola et al., 2016; Gholami et al., 2018), MobileNets (Howard et al., 2017; Sandler et al., 2018), and ShuffleNets (Zhang et al., 2018; Ma et al., 2018). Recently, neural architecture search becomes increasingly popular in designing efficient mobile-size ConvNets (Tan et al., 2019; Cai et al., 2019), and achieves even better efficiency than hand-crafted mobile ConvNets by extensively tuning the network width, depth, convolution kernel types and sizes. However, it is unclear how to apply these techniques for larger models that have much larger design space and much more expensive tuning cost. In this paper, we aim to study model efficiency for super large ConvNets that surpass state-of-the-art accuracy. To achieve this goal, we resort to model scaling.

**ConvNet 효율성**: Deep ConvNet은 종종 과잉 파라미터화 됩니다. 모델 압축(Han et al., 2016; He et al., 2018; Yang et al., 2018)은 정확성과 효율성을 교환하여 모델 크기를 줄이는 일반적인 방법이다. 휴대전화가 보편화됨에 따라 효율적인 모바일 크기의 ConvNets(Squeeze 등)를 수작업으로 만드는 것도 일반적이다.Nets(Iandola et al., 2016; Golami et al., 2018), MobileNets(Howard et al., 2017; Sandler et al., 2018), ShuffleNets(Zhang et al., 2018; Ma et al., 2018). 최근 신경 아키텍처 검색은 효율적인 모바일 크기 ConvNet 설계(Tan et al., 2019; Cai et al., 2019)에서 점점 더 인기를 얻고 있으며, 네트워크 폭, 깊이, 컨볼루션 커널 유형 및 크기를 광범위하게 조정하여 수작업으로 만든 모바일 ConvNets보다 훨씬 더 나은 효율성을 달성한다. 그러나 설계 공간이 훨씬 넓고 튜닝 비용이 훨씬 비싼 대형 모델에 이러한 기술을 적용하는 방법은 불분명합니다. 본 백서에서는 최첨단 정확도를 능가하는 초대형 ConvNet의 모델 효율성을 연구하는 것을 목표로 한다. 이 목표를 달성하기 위해 모델 스케일링에 의존합니다.

**Model Scaling:** There are many ways to scale a ConvNet for different resource constraints: ResNet (He et al., 2016) can be scaled down (e.g., ResNet-18) or up (e.g., ResNet-200) by adjusting network depth (#layers), while WideResNet (Zagoruyko & Komodakis, 2016) and MobileNets (Howard et al., 2017) can be scaled by network width (#channels). It is also well-recognized that bigger input image size will help accuracy with the overhead of more FLOPS. Although prior studies (Raghu et al., 2017; Lin & Jegelka, 2018; Sharir & Shashua, 2018; Lu et al., 2018) have shown that network depth and width are both important for ConvNets’ expressive power, it still remains an open question of how to effectively scale a ConvNet to achieve better efficiency and accuracy. Our work systematically and empirically studies ConvNet scaling for all three dimensions of network width, depth, and resolutions.

**모델 스케일링**: 다양한 리소스 제약에 따라 ConvNet을 확장할 수 있는 방법은 여러 가지가 있습니다. ResNet(He 등, 2016)은 네트워크 깊이(#layer)를 조정하여 축소(예: ResNet-18) 또는 확대(예: ResNet-200)할 수 있으며, WideResNet(Zagoruyko & Komodakis, 2016) 및 MobileNet(Howard 등, 2017#)은 네트워크 폭에 따라 확장할 수 있습니다. 또한 입력 이미지 크기가 커지면 더 많은 FLOPS의 오버헤드에 대한 정확성에 도움이 된다는 것도 잘 알려져 있습니다. 이전 연구(Raghu et al., 2017; Lin & Jegelka, 2018; Sharir & Shashua, 2018; Lu et al., 2018)에서는 ConvNet의 표현력에 네트워크의 깊이와 폭이 모두 중요하다는 것이 밝혀졌지만, 효율성과 정확성을 높이기 위해 ConvNet을 효과적으로 확장하는 방법은 여전히 미해결 문제이다. 당사의 작업은 네트워크 폭, 깊이 및 해상도의 3차원 모두에 대한 ConvNet 스케일링을 체계적이고 경험적으로 연구합니다.

1. **Compound Model Scaling**

In this section, we will formulate the scaling problem, study different approaches, and propose our new scaling method.

이 섹션에서는 스케일링 문제를 공식화하고 다양한 접근 방식을 연구하며 새로운 스케일링 방법을 제안합니다.

* 1. **Problem Formulation**

A ConvNet Layer i can be defined as a function: Yi = Fi(Xi), where Fi is the operator, Yi is output tensor, Xi is input tensor, with tensor shape hHi, Wi, Cii1, where Hi and Wi are spatial dimension and Ci is the channel dimension. A ConvNet N can be represented by a list of composed layers: N = Fk F2 F1(X1) = J j=1k Fj (X1). In practice, ConvNet layers are often partitioned into multiple stages and all layers in each stage share the same architecture: for example, ResNet (He et al., 2016) has five stages, and all layers in each stage has the same convolutional type except the first layer performs down-sampling. Therefore, we can define a ConvNet as:

ConvNet Layer i는 함수로 정의할 수 있습니다. 여기서 Fi는 연산자이고, Yi는 출력 텐서이고, Xi는 입력 텐서이며, 텐서 모양 hHi, Wi, Cii1이며, Hi와 Wi는 공간 차원이고, Ci는 채널 차원입니다. ConvNet N은 N = Fk F2 F1(X1) = J J = 1k Fj(X1)의 합성층 리스트로 나타낼 수 있다. 실제로 ConvNet 계층은 종종 여러 단계로 분할되며 각 단계의 모든 계층은 동일한 아키텍처를 공유합니다. 예를 들어 ResNet(He 등, 2016)은 5개의 단계를 가지며, 각 단계의 모든 계층은 첫 번째 계층이 다운샘플링을 수행하는 것을 제외하고 동일한 컨볼루션 유형을 가집니다. 따라서 ConvNet은 다음과 같이 정의할 수 있습니다.

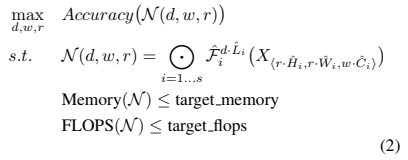


where F Li i denotes layer Fi is repeated Li times in stage i, (Hi, Wi, Ci) denotes the shape of input tensor X of layer i. Figure 2(a) illustrate a representative ConvNet, where the spatial dimension is gradually shrunk but the channel dimension is expanded over layers, for example, from initial input shape (224, 224, 3) to final output shape (7, 7, 512).

여기서 F Lii는 1단계에서 Fi층이 Li회 반복되는 것을 의미하며, (Hi, Wi, Ci)는 i층의 입력 텐서 X의 형태를 나타낸다. 그림 2(a)는 예를 들어 초기 입력 형상(224, 224, 3)에서 최종 출력 형상(7, 7, 512)까지 공간 치수는 점차 축소되지만 채널 치수는 레이어에 걸쳐 확장되는 대표적인 ConvNet을 나타낸다.

Unlike regular ConvNet designs that mostly focus on finding the best layer architecture Fi, model scaling tries to expand the network length (Li), width (Ci), and/or resolution (Hi, Wi) without changing Fi predefined in the baseline network. By fixing Fi, model scaling simplifies the design problem for new resource constraints, but it still remains a large design space to explore different Li, Ci, Hi, Wi for each layer. In order to further reduce the design space, we restrict that all layers must be scaled uniformly with constant ratio. Our target is to maximize the model accuracy for any given resource constraints, which can be formulated as an optimization problem

주로 최적의 계층 아키텍처 Fi를 찾는 데 초점을 맞춘 일반적인 ConvNet 설계와 달리 모델 스케일링은 기본 네트워크에서 미리 정의된 Fi를 변경하지 않고 네트워크 길이(Li), 폭(Ci) 및/또는 해상도(Hi, Wi)를 확장하려고 합니다. Fi를 고정함으로써 모델 스케일링은 새로운 자원 제약에 대한 설계 문제를 단순화하지만, 각 계층에 대해 서로 다른 Li, Ci, Hi, Wi를 탐색할 수 있는 큰 설계 공간으로 남아 있습니다. 설계 공간을 더욱 줄이기 위해 모든 레이어를 일정한 비율로 균일하게 스케일링해야 하는 것을 제한합니다. 우리의 목표는 최적화 문제로 공식화될 수 있는 주어진 리소스 제약에 대한 모델 정확도를 극대화하는 것이다.



where w, d, r are coefficients for scaling network width, depth, and resolution; Fˆi, Lˆi, Hˆi, Wˆi, Cˆi are predefined parameters in baseline network (see Table 1 as an example).

여기서 w, d, r은 네트워크 폭, 깊이 및 분해능을 확장하기 위한 계수입니다.fii, lii, hii, wii, cii는 기준선 네트워크에서 사전 정의된 파라미터입니다(예로서 표 1 참조).

* 1. **Scaling Dimensions**

The main difficulty of problem 2 is that the optimal d, w, r depend on each other and the values change under different resource constraints. Due to this difficulty, conventional methods mostly scale ConvNets in one of these dimensions:

문제 2의 가장 큰 어려움은 최적의 d, w, r이 서로 의존하며 값이 다른 자원 제약 조건 하에서 변화한다는 것입니다. 이러한 어려움으로 인해, 기존의 방법들은 대부분 다음 중 하나의 차원에서 ConvNets를 확장합니다.

**Depth (d):** Scaling network depth is the most common way used by many ConvNets (He et al., 2016; Huang et al., 2017; Szegedy et al., 2015; 2016). The intuition is that deeper ConvNet can capture richer and more complex features, and generalize well on new tasks. However, deeper networks are also more difficult to train due to the vanishing gradient problem (Zagoruyko & Komodakis, 2016). Although several techniques, such as skip connections (He et al., 2016) and batch normalization (Ioffe & Szegedy, 2015), alleviate the training problem, the accuracy gain of very deep network diminishes: for example, ResNet-1000 has similar accuracy as ResNet-101 even though it has much more layers. Figure 3 (middle) shows our empirical study on scaling a baseline model with different depth coefficient d, further suggesting the diminishing accuracy return for very deep ConvNets.

**깊이 (d):** 네트워크 깊이를 확장하는 것은 많은 ConvNets에서 가장 일반적으로 사용되는 방법이다(He 등, 2016; Huang 등, 2017; Szegedy 등, 2015; 2016). ConvNet은 보다 심층적인 ConvNet을 통해 보다 풍부하고 복잡한 기능을 포착할 수 있으며, 새로운 작업에서도 잘 일반화할 수 있습니다. 그러나, 깊은 네트워크도 구배 문제가 사라지기 때문에 훈련하기가 더 어렵다(Zagoruyko & Komodakis, 2016). 스킵 접속(He et al., 2016) 및 배치 정규화(Ioffe & Szegedy, 2015)와 같은 여러 기법이 훈련 문제를 완화하지만, 매우 깊은 네트워크의 정확도 향상은 감소한다. 예를 들어 ResNet-1000은 훨씬 더 많은 레이어를 가지고 있지만 ResNet-101과 유사한 정확도를 가진다. 그림 3(가운데)은 깊이 계수 d가 다른 기준선 모델의 스케일링에 대한 경험적 연구를 보여 주며, 매우 깊은 ConvNet의 정확도 수익 감소가 더욱 두드러진다.

**Width (w):** Scaling network width is commonly used for small size models (Howard et al., 2017; Sandler et al., 2018; Tan et al., 2019). As discussed in (Zagoruyko & Komodakis, 2016), wider networks tend to be able to capture more fine-grained features and are easier to train. However, extremely wide but shallow networks tend to have difficulties in capturing higher level features. Our empirical results in Figure 3 (left) show that the accuracy quickly saturates when networks become much wider with larger w.

**폭(w):** 스케일링 네트워크 폭은 소형 모델에 일반적으로 사용된다(Howard 등, 2017; Sandler 등, 2018; Tan 등, 2019). (Zagoruyko & Komodakis, 2016)에서 설명한 바와 같이, 넓은 네트워크는 보다 세밀한 특징을 포착할 수 있고 훈련하기 쉬운 경향이 있다. 단, 매우 광범위하지만 얕은 네트워크에서는 보다 높은 수준의 기능을 캡처하는 데 어려움을 겪는 경향이 있습니다. 그림 3(왼쪽)의 경험적 결과에 따르면 w가 클수록 네트워크가 훨씬 넓어지면 정확도가 빠르게 포화됩니다.

**Resolution (r):** With higher resolution input images, ConvNets can potentially capture more fine-grained patterns. Starting from 224x224 in early ConvNets, modern ConvNets tend to use 299x299 (Szegedy et al., 2016) or 331x331 (Zoph et al., 2018) for better accuracy. Recently, GPipe (Huang et al., 2018) achieves state-of-the-art ImageNet accuracy with 480x480 resolution. Higher resolutions, such as 600x600, are also widely used in object detection ConvNets (He et al., 2017; Lin et al., 2017). Figure 3 (right) shows the results of scaling network resolutions, where indeed higher resolutions improve accuracy, but the accuracy gain diminishes for very high resolutions (r = 1.0 denotes resolution 224x224 and r = 2.5 denotes resolution 560x560).

The above analyses lead us to the first observation:

**해상도(r):** 고해상도 입력 이미지를 사용하면 ConvNets는 잠재적으로 보다 세밀한 패턴을 캡처할 수 있습니다. 초기 ConvNets에서 224x224를 시작으로, 현대의 ConvNets는 더 나은 정확도를 위해 299x299(Szegedy 등, 2016) 또는 331x331(Zoph 등, 2018)을 사용하는 경향이 있다. 최근 GPipe(Huang et al., 2018)는 480x480 해상도로 최첨단 ImageNet 정확도를 달성했습니다. 600x600과 같은 고해상도 또한 물체 감지 ConvNets에 널리 사용된다(He 등, 2017; Lin 등, 2017). 그림 3(오른쪽)은 네트워크 해상도의 스케일링 결과를 보여줍니다. 여기서 고해상도에서는 정확도가 향상되지만, 매우 높은 해상도에서는 정확도 이득이 감소합니다. (r = 1.0은 224x224를 나타내고 r = 2.5는 560x4를 나타냅니다.)

위의 분석을 통해 첫 번째 관찰 결과를 얻을 수 있습니다.

**Observation 1** – Scaling up any dimension of network width, depth, or resolution improves accuracy, but the accuracy gain diminishes for bigger models.

**관찰 1** – 네트워크 폭, 깊이 또는 해상도의 모든 차원을 확대하면 정확도가 향상되지만 모델이 클수록 정확도가 낮아집니다.

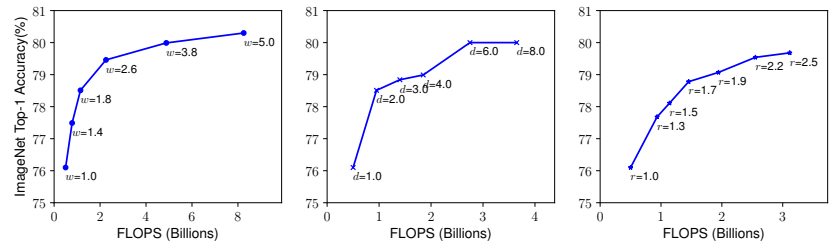


Figure 3. Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients. Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single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

그림 3 네트워크 폭(w), 깊이(d) 및 분해능(r) 계수가 다른 기준선 모델 스케일업 넓은 폭, 깊이 또는 해상도를 가진 큰 네트워크일수록 정확도가 높아지는 경향이 있지만 정확도 향상은 80%에 도달한 후 빠르게 포화되기 때문에 단일 치수 확장이 제한적이라는 것을 알 수 있습니다. 베이스라인 네트워크는 표 1에 설명되어 있습니다.

* 1. **Compound Scaling**

We empirically observe that different scaling dimensions are not independent. Intuitively, for higher resolution images, we should increase network depth, such that the larger receptive fields can help capture similar features that include more pixels in bigger images. Correspondingly, we should also increase network width when resolution is higher, in order to capture more fine-grained patterns with more pixels in high resolution images. These intuitions suggest that we need to coordinate and balance different scaling dimensions rather than conventional single-dimension scaling.

우리는 다른 스케일링 치수가 독립적이지 않다는 것을 경험적으로 관찰한다. 직관적으로 고해상도 이미지의 경우 네트워크 깊이를 늘려야 합니다.이것에 의해, 리셉티브 필드가 커지면, 보다 큰 이미지에 더 많은 픽셀이 포함되는 같은 기능을 캡쳐 할 수 있습니다. 따라서 고해상도 이미지에서 더 많은 픽셀로 더 세밀한 패턴을 포착하기 위해 해상도가 높을 때 네트워크 폭도 늘려야 합니다. 이러한 직관들은 우리가 기존의 단일 차원 축척이 아닌 다양한 축척 치수를 조정하고 균형을 맞출 필요가 있음을 시사합니다.

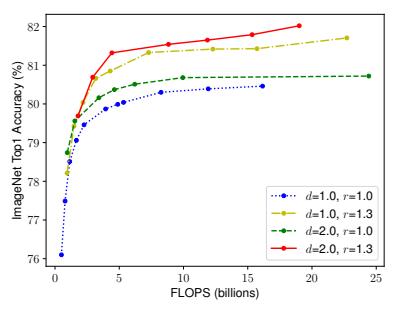


Figure 4. **Scaling Network Width for Different Baseline Networks**. Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline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

그림 4 **다른 기준선 네트워크에 대한 네트워크 폭 확장**. 선의 각 점은 다른 폭 계수 w를 가진 모형을 나타냅니다. 모든 베이스라인 네트워크는 표1에 기재되어 있습니다. 첫 번째 기준선 네트워크(d=1.0, r=1.0)는 분해능 224x224의 18개의 컨볼루션 계층을 가지고 있고, 마지막 기준선(d=2.0, r=1.3)은 분해능 299x12의 36개의 계층을 가지고 있다.

To validate our intuitions, we compare width scaling under different network depths and resolutions, as shown in Figure 4. If we only scale network width w without changing depth (d=1.0) and resolution (r=1.0), the accuracy saturates quickly. With deeper (d=2.0) and higher resolution (r=2.0), width scaling achieves much better accuracy under the same FLOPS cost. These results lead us to the second observation:

직관을 검증하기 위해 그림 4와 같이 다양한 네트워크 깊이와 해상도에서 폭 확장을 비교합니다. 깊이(d=1.0)와 해상도(r=1.0)를 변경하지 않고 네트워크 폭 w만 스케일링하면 정확도가 빠르게 포화됩니다. 더 깊은 (d=2.0) 및 더 높은 해상도 (r=2.0)로, 폭 스케일링은 동일한 FLOPS 비용으로 훨씬 더 나은 정확도를 달성합니다. 이러한 결과는 두 번째 관찰로 이어집니다.

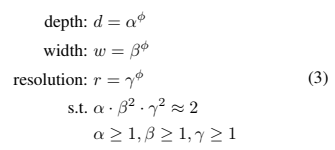
**Observation 2** – In order to pursue better accuracy and efficiency, it is critical to balance all dimensions of network width, depth, and resolution during ConvNet scaling.

In fact, a few prior work (Zoph et al., 2018; Real et al., 2019) have already tried to arbitrarily balance network width and depth, but they all require tedious manual tuning.

**관찰 2** – 정확성과 효율성을 높이기 위해서는 ConvNet 확장 시 네트워크 폭, 깊이 및 해상도의 모든 차원을 균형 있게 조정하는 것이 중요합니다.

실제로 몇몇 이전 연구(Zoph 등, 2018; Real 등, 2019)는 이미 네트워크 폭과 깊이의 균형을 임의로 맞추려고 시도했지만 모두 지루한 수동 조정이 필요하다.

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:



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, w2, r2, 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 α · β 2 · γ2 φ. In this paper, we constraint α · β 2 · γ 2 ≈ 2 such that for any new φ, the total FLOPS will approximately3 increase by 2φ.

여기서 α, β, β는 작은 그리드 검색에 의해 결정될 수 있는 상수이다. 직관적으로 α는 모델 스케일링에 사용할 수 있는 자원의 수를 제어하는 사용자 지정 계수이며, α, β는 이러한 추가 자원을 네트워크 폭, 깊이 및 분해능에 할당하는 방법을 각각 지정합니다. 특히, 일반 컨볼루션 동작의 FLOPS는 d, w2, r2에 비례합니다. 즉, 네트워크 깊이를 2배로 하면 FLOPS가 2배로 증가하지만, 네트워크 폭 또는 분해능을 2배로 하면 FLOPS가 4배로 증가합니다. 보통 ConvNets에서는 Convolution ops가 계산 비용을 지배하기 때문에 방정식 3을 사용하여 ConvNet을 스케일링하면 총 FLOPS가 α · β 2 · 2 2 φ 정도 증가합니다. 본서에서는 새로운 φ에 대해 총 FLOPS가 약 3 increase 증가하도록 α · β 2 · 2 2 2 2를 제한한다.

1. **EfficientNet Architecture**

Since model scaling does not change layer operators Fˆi in baseline network, having a good baseline network is also critical. We will evaluate our scaling method using existing ConvNets, but in order to better demonstrate the effectiveness of our scaling method, we have also developed a new mobile-size baseline, called EfficientNet.

모델 스케일링은 베이스라인 네트워크에서 레이어 연산자 fˆi를 변경하지 않으므로 양호한 베이스라인 네트워크를 구축하는 것도 중요합니다. 기존 ConvNet을 사용하여 스케일링 방법을 평가하지만 스케일링 방법의 효과를 더 잘 입증하기 위해 EfficientNet이라는 새로운 모바일 크기 기준선을 개발했습니다.

Inspired by (Tan et al., 2019), we develop our baseline network by leveraging a multi-objective neural architecture search that optimizes both accuracy and FLOPS. Specifically, we use the same search space as (Tan et al., 2019), and use ACC(m)×[F LOP S(m)/T] w as the optimization goal, where ACC(m) and F LOP S(m) denote the accuracy and FLOPS of model m, T is the target FLOPS and w=-0.07 is a hyperparameter for controlling the trade-off between accuracy and FLOPS. Unlike (Tan et al., 2019; Cai et al., 2019), here we optimize FLOPS rather than latency since we are not targeting any specific hardware device. Our search produces an efficient network, which we name EfficientNet-B0. Since we use the same search space as (Tan et al., 2019), the architecture is similar to Mnas-Net, except our EfficientNet-B0 is slightly bigger due to the larger FLOPS target (our FLOPS target is 400M). Table 1 shows the architecture of EfficientNet-B0. Its main building block is mobile inverted bottleneck MBConv (Sandler et al., 2018; Tan et al., 2019), to which we also add squeeze-and-excitation optimization (Hu et al., 2018).

(Tan 등, 2019)에서 영감을 받아 정확도와 FLOPS를 모두 최적화하는 다목적 신경 아키텍처 검색을 활용하여 기준 네트워크를 개발한다. 구체적으로는 (Tan et al., 2019)와 동일한 서치 공간을 사용하며, 최적화 목표로 ACC(m)×[F LOP S(m)/T] w를 사용한다. 여기서 ACC(m)와 F LOP S(m)는 모델 m의 정확도를 나타내며, FLOPS는 대상 FLOPS a(0)이다. (Tan et al., 2019; Cai et al., 2019)와는 달리, 여기서는 특정 하드웨어 장치를 대상으로 하지 않기 때문에 대기 시간이 아닌 FLOPS를 최적화한다. 이 검색은 효율적인 네트워크를 생성하며, 이를 EfficientNet-B0이라고 명명합니다. (Tan et al., 2019)와 동일한 서치 공간을 사용하기 때문에, 아키텍처는 Mnas-Net과 유사하지만, EfficientNet-B0은 FLOPS 타깃이 크기 때문에 약간 더 크다(우리의 FLOPS 타깃은 400M이다). 표 1은 EfficientNet-B0의 아키텍처를 나타내고 있습니다. 주요 구성 요소는 이동형 역병목 MBConv(Sandler 등, 2018; Tan 등, 2019)이며, 여기에 스퀴즈 앤 들뜸 최적화도 추가하였다(Hu 등, 2018).

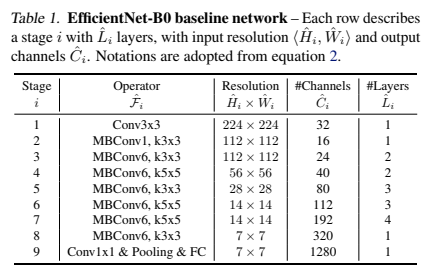


Table 1. EfficientNet-B0 baseline network – Each row describes a stage i with Lˆi layers, with input resolution hHˆi, Wˆ ii and output channels Cˆi. Notations are adopted from equation 2.

표 1. EfficientNet-B0 베이스라인 네트워크– 각 행은 입력 분해능hhii, W iiii 및 출력 채널cii를 가진 l layersi 레이어를 가진 스테이지i를 나타내고 있습니다. 표기는 등식 2에서 채택되었다.

Starting from the baseline EfficientNet-B0, we apply our compound scaling method to scale it up with two steps:

• STEP 1: we first fix φ = 1, assuming twice more resources available, and do a small grid search of α, β, γ based on Equation 2 and 3. In particular, we find the best values for EfficientNet-B0 are α = 1.2, β =1.1, γ = 1.15, under constraint of α · β 2· γ 2 ≈ 2.

• STEP 2: we then fix α, β, γ as constants and scale up baseline network with different φ using Equation 3, to obtain EfficientNet-B1 to B7 (Details in Table 2).

베이스라인 EfficientNet-B0부터 복합 스케일링 방법을 적용하여 다음 2단계로 스케일업합니다.

• 1단계: 먼저 θ = 1을 고정하고, 2배 더 많은 자원을 사용할 수 있다고 가정하고, 식 2와 3에 기초하여 α, β, θ의 소규모 그리드 검색을 실시한다. 특히, EfficientNet-B0에 대한 최선의 값은 α · β = 1.2, β = 1.1, β = 1.15이며, β · β 2 · 2 2 2 2 2 2이다.

• 스텝 2: 다음으로 α, β, θ를 상수로 고정하고 식 3을 사용하여 다른 θ로 베이스라인 네트워크를 스케일업하여 EfficientNet-B1에서 B7로 한다(표 2의 상세 정보).

Notably, it is possible to achieve even better performance by searching for α, β, γ directly around a large model, but the search cost becomes prohibitively more expensive on larger models. Our method solves this issue by only doing search once on the small baseline network (step 1), and then use the same scaling coefficients for all other models (step 2).

특히 대형 모델 주변에서 직접 α, β, δ를 검색하면 더 나은 성능을 얻을 수 있지만, 대형 모델에서는 검색 비용이 엄청나게 비싸다. 이 방법에서는 소규모 기준선 네트워크에서 한 번만 검색(1단계)한 다음 다른 모든 모델에 동일한 스케일링 계수를 사용하여 이 문제를 해결합니다(2단계).

1. **Experiments**

In this section, we will first evaluate our scaling method on existing ConvNets and the new proposed EfficientNets.

이 섹션에서는 먼저 기존 ConvNets 및 새로 제안된 EfficientNets에 대한 확장 방법을 평가합니다.

* 1. **Scaling Up MobileNets and ResNets**

As a proof of concept, we first apply our scaling method to the widely-used MobileNets (Howard et al., 2017; Sandler et al., 2018) and ResNet (He et al., 2016). Table 3 shows the ImageNet results of scaling them in different ways. Compared to other single-dimension scaling methods, our compound scaling method improves the accuracy on all these models, suggesting the effectiveness of our proposed scaling method for general existing ConvNets.

개념 증명으로서 우선 널리 사용되는 MobileNets(Howard 등, 2017년, Sandler 등, 2018년)와 ResNet(He 등, 2016년)에 스케일링 방법을 적용한다. 표 3은 다양한 방법으로 ImageNet을 스케일링한 결과를 보여줍니다. 다른 단일 차원 스케일링 방법과 비교하여 복합 스케일링 방법은 이러한 모든 모델에서 정확도를 개선하여 일반적인 기존 ConvNet에 대해 제안된 스케일링 방법의 효과를 제시한다.

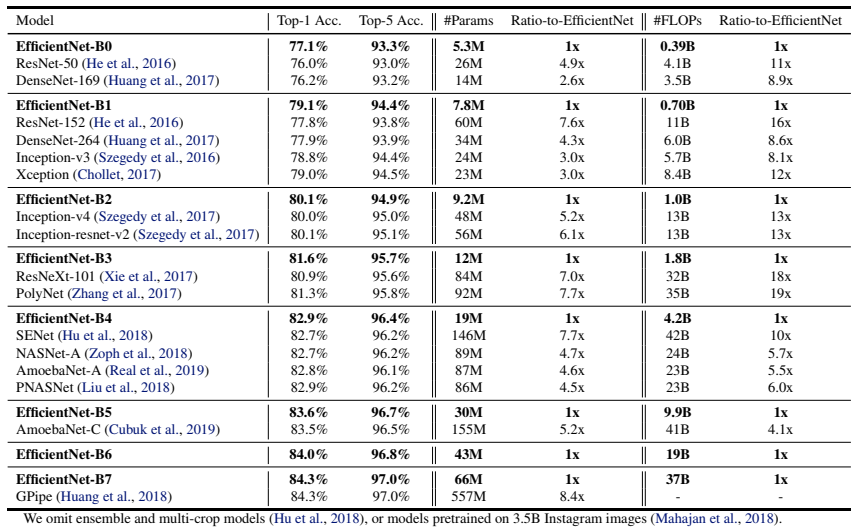
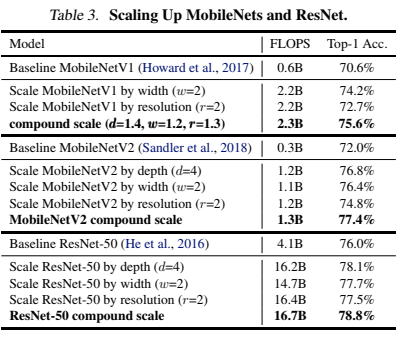
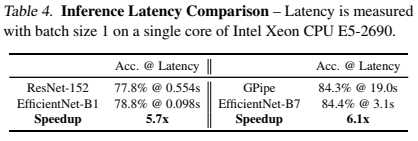
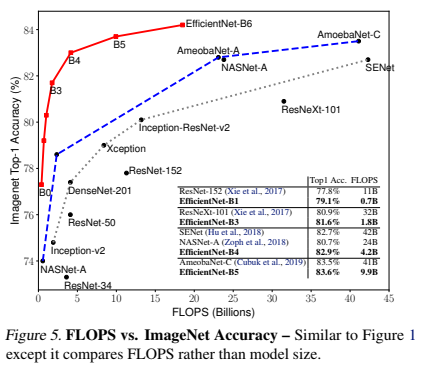


Table 2. **EfficientNet Performance Results on ImageNet** (Russakovsky et al., 2015). All EfficientNet models are scaled from our baseline EfficientNet-B0 using different compound coefficient φ in Equation 3. ConvNets with similar top-1/top-5 accuracy are grouped together for efficiency comparison. Our 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.

표 2. **ImageNet의 EfficientNet 퍼포먼스 결과**(Russakovsky 등, 2015). 모든 EfficientNet 모델은 식 3의 다른 복합계수 θ를 사용하여 베이스라인 EfficientNet-B0에서 스케일링됩니다. 유사한 상위 1/5 정확도의 ConvNet을 함께 그룹화하여 효율성을 비교합니다. NAT의 스케일링된 EfficientNet 모델은 기존 ConvNet보다 파라미터 및 FLOPS를 최대 8.4배 감소 및 최대 16배 감소 순서로 지속적으로 줄입니다.



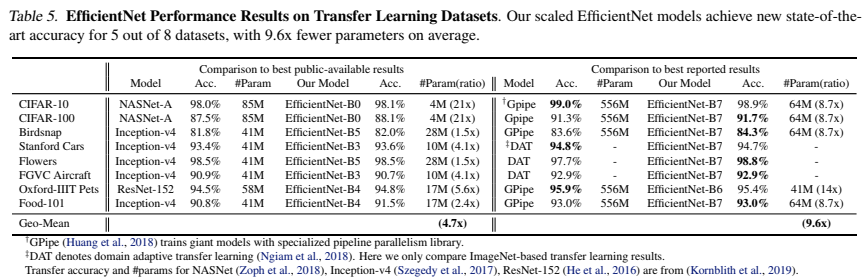




* 1. **ImageNet Results for EfficientNet**

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; weight decay 1e-5; initial learning rate 0.256 that decays by 0.97 every 2.4 epochs. We also use SiLU (Swish-1) activation (Ramachandran et al., 2018; Elfwing et al., 2018; Hendrycks & Gimpel, 2016), AutoAugment (Cubuk et al., 2019), and stochastic depth (Huang et al., 2016) with survival probability 0.8. As commonly known that bigger models need more regularization, we linearly increase dropout (Srivastava et al., 2014) ratio from 0.2 for EfficientNet-B0 to 0.5 for B7. We reserve 25K randomly picked images from the training set as a minival set, and perform early stopping on this minival; we then evaluate the earlystopped checkpoint on the original validation set to report the final validation accuracy.

ImageNet에서 EfficientNet 모델을 교육한다(Tan 등, 2019). 붕괴 0.9 및 운동량 0.9, 배치 기준 운동량 0.99, 무게 붕괴 1e-5, 2.4에폭마다 0.97씩 감소하는 초기 학습 속도 0.256과 유사한 설정을 사용한다. 또한 생존 확률이 0.8인 SiLU(Swish-1) 활성화(Ramachandran 등, 2018; Elfwing 등, 2018; Hendrycks & Gimpel, 2016), AutoAugment(Cubuk 등, 2019), 확률적 깊이(Huang 등, 2016)를 사용한다. 일반적으로 더 큰 모델은 정규화가 필요하다는 것을 알고 있듯이, EfficientNet-B0의 경우 0.2에서 B7의 경우 0.5로 드롭아웃 비율(Srivastava et al., 2014)을 선형적으로 높였습니다. 트레이닝 세트로부터 무작위로 추출한 25K개의 영상을 미니벌 세트로 예약해, 이 미니벌로 조기 정지를 실시해, 원래의 검증 세트상에서 조기 정지 체크 포인트를 평가해 최종 검증 정확도를 보고한다.



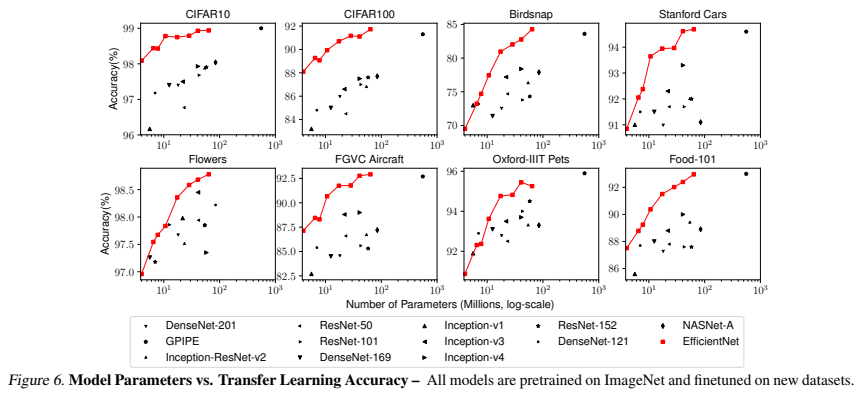


Table 2 shows the performance of all EfficientNet models that are scaled from the same baseline EfficientNet-B0. Our EfficientNet models generally use an order of magnitude fewer parameters and FLOPS than other ConvNets with similar accuracy. In particular, our EfficientNet-B7 achieves 84.3% top1 accuracy with 66M parameters and 37B FLOPS, being more accurate but **8.4x smaller** than the previous best GPipe (Huang et al., 2018). These gains come from both better architectures, better scaling, and better training settings that are customized for EfficientNet.

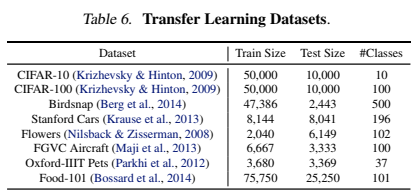
표 2는 동일한 기준 EfficientNet-B0에서 스케일링된 모든 EfficientNet 모델의 성능을 보여줍니다. EfficientNet 모델은 일반적으로 유사한 정확도의 다른 ConvNet보다 훨씬 적은 파라미터와 FLOPS를 사용합니다. 특히 EfficientNet-B7은 66M 파라미터와 37B FLOPS로 84.3%의 톱1 정확도를 달성하여 이전 최고의 GPIpe보다 정확도는 높지만 8.4배 감소하였습니다(Huang et al., 2018). 이러한 이점은 EfficientNet용으로 커스터마이즈된 더 나은 아키텍처, 더 나은 확장성 및 더 나은 교육 설정을 통해 얻을 수 있습니다.

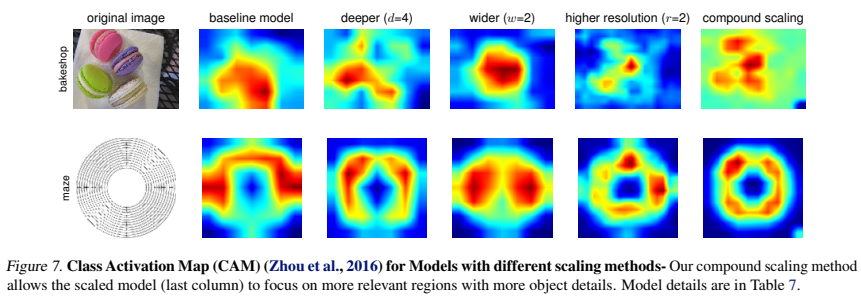
Figure 1 and Figure 5 illustrates the parameters-accuracy and FLOPS-accuracy curve for representative ConvNets, where our scaled EfficientNet models achieve better accuracy with much fewer parameters and FLOPS than other ConvNets. Notably, our EfficientNet models are not only small, but also computational cheaper. For example, our EfficientNet-B3 achieves higher accuracy than ResNeXt101 (Xie et al., 2017) using **18x fewer FLOPS.**

그림 1과 그림 5는 대표적인 ConvNet에 대한 매개변수 정확도 및 FLOPS 정확도 곡선을 보여줍니다. 이 곡선에서 스케일링된 EfficientNet 모델은 다른 ConvNet보다 훨씬 적은 매개변수 및 FLOPS로 더 나은 정확도를 달성합니다. 특히 델의 EfficientNet 모델은 소형일 뿐만 아니라 계산 비용도 저렴합니다. 예를 들어, EfficientNet-B3는 18배 적은 FLOPS를 사용하여 ResNeXt101(Xie 등, 2017)보다 높은 정확도를 달성합니다.

To validate the latency, we have also measured the inference latency for a few representative CovNets on a real CPU as shown in Table 4, where we report average latency over 20 runs. Our EfficientNet-B1 runs 5.7x faster than the widely used ResNet-152, while EfficientNet-B7 runs about 6.1x faster than GPipe (Huang et al., 2018), suggesting our EfficientNets are indeed fast on real hardware.

또한 지연을 검증하기 위해 실제 CPU 상의 몇 가지 대표적인 CovNet에 대한 추론 지연도 측정했습니다(표 4 참조). 여기서 평균 지연 시간은 20회 이상입니다. NAT의 EfficientNet-B1은 널리 사용되는 ResNet-152보다 5.7배 더 빠르게 실행되는 반면, EfficientNet-B7은 GPipe보다 약 6.1배 더 빠르게 실행되므로(Huang 등, 2018년) 실제 하드웨어에서 EfficientNets가 실제로 더 빠르다는 것을 알 수 있습니다.





* 1. **Transfer Learning Results for EfficientNet**

We have also evaluated our EfficientNet on a list of commonly used transfer learning datasets, as shown in Table 6. We borrow the same training settings from (Kornblith et al., 2019) and (Huang et al., 2018), which take ImageNet pretrained checkpoints and finetune on new datasets.

또, 표 6과 같이, 일반적으로 사용되는 전송 학습 데이터 세트 리스트에서도 EfficientNet을 평가했습니다. 동일한 교육 설정을 (Kornblith 등, 2019) 및 (Huang 등, 2018)에서 빌려 ImageNet 사전 교육 체크포인트를 취하고 새로운 데이터셋에 대해 미세 조정한다.

Table 5 shows the transfer learning performance: (1) Compared to public available models, such as NASNet-A (Zoph et al., 2018) and Inception-v4 (Szegedy et al., 2017), our EfficientNet models achieve better accuracy with 4.7x average (up to 21x) parameter reduction. (2) Compared to stateof-the-art models, including DAT (Ngiam et al., 2018) that dynamically synthesizes training data and GPipe (Huang et al., 2018) that is trained with specialized pipeline parallelism, our EfficientNet models still surpass their accuracy in 5 out of 8 datasets, but using 9.6x fewer parameters.

표 5는 전달 학습 성과를 보여준다. (1) NAT의 EfficientNet 모델은 NASNet-A(Zoph 등, 2018) 및 Inception-v4(Szegedy 등, 2017)와 같은 공공 이용 가능한 모델과 비교하여 4.7배(최대 21배)의 평균 매개변수 감소(2)를 달성한다.특수 파이프라인 병렬로 교육된 교육 데이터와 GPipe(Huang 등, 2018)를 완벽하게 통합한 NAT의 EfficientNet 모델은 8개의 데이터셋 중 5개의 데이터셋에서 정확성을 여전히 능가하지만 9.6배 적은 매개 변수를 사용합니다.

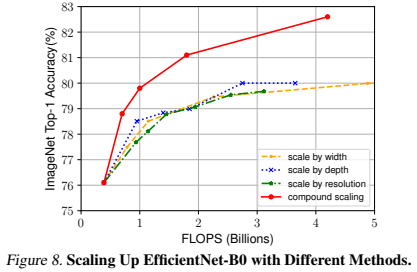
Figure 6 compares the accuracy-parameters curve for a variety of models. In general, our EfficientNets consistently achieve better accuracy with an order of magnitude fewer parameters than existing models, including ResNet (He et al., 2016), DenseNet (Huang et al., 2017), Inception (Szegedy et al., 2017), and NASNet (Zoph et al., 2018).

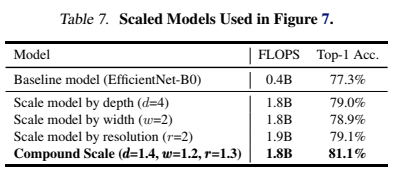
그림 6은 다양한 모델의 정확도-파라미터 곡선을 비교합니다. 일반적으로 EfficientNets는 ResNet(He 등, 2016), DenseNet(Huang 등, 2017), Inception(Szegedy 등, 2017) 및 NASNet(Zoph 등, 2018)을 포함하여 기존 모델보다 훨씬 적은 수의 매개 변수로 일관되게 더 나은 정확도를 달성한다.

1. **Discussion**

To disentangle the contribution of our proposed scaling method from the EfficientNet architecture, Figure 8 compares the ImageNet performance of different scaling methods for the same EfficientNet-B0 baseline network. In general, all scaling methods improve accuracy with the cost of more FLOPS, but our compound scaling method can further improve accuracy, by up to 2.5%, than other singledimension scaling methods, suggesting the importance of our proposed compound scaling.

EfficientNet 아키텍처에서 제안된 스케일링 방법의 기여도를 구별하기 위해 그림 8은 동일한 EfficientNet-B0 베이스라인 네트워크에 대한 다양한 스케일링 방법의 ImageNet 퍼포먼스를 비교하고 있습니다. 일반적으로 모든 스케일링 방법은 더 많은 FLOPS의 비용으로 정확도를 향상시키지만, 복합 스케일링 방법은 다른 단일 치수 스케일링 방법보다 최대 2.5% 더 정확성을 개선할 수 있어 제안된 복합 스케일링의 중요성을 시사한다.





In order to further understand why our compound scaling method is better than others, Figure 7 compares the class activation map (Zhou et al., 2016) for a few representative models with different scaling methods. All these models are scaled from the same baseline, and their statistics are shown in Table 7. Images are randomly picked from ImageNet validation set. As shown in the figure, the model with compound scaling tends to focus on more relevant regions with more object details, while other models are either lack of object details or unable to capture all objects in the images.

복합 스케일링 방법이 다른 모델보다 나은 이유를 더 잘 이해하기 위해 그림 7은 스케일링 방법이 다른 몇몇 대표 모델에 대한 클래스 활성화 맵(Zhou 등, 2016)을 비교한다. 이 모든 모델은 동일한 기준선에서 스케일링되며, 그 통계는 표 7과 같다. 이미지는 ImageNet 검증 세트에서 랜덤으로 선택됩니다. 그림에서 보듯이 복합 배율이 있는 모델은 더 많은 개체 세부 정보가 있는 관련 영역에 초점을 맞추는 경향이 있지만, 다른 모델은 개체 세부 정보가 없거나 이미지에서 모든 개체를 캡처할 수 없습니다.

1. **Conclusion**

In this paper, we systematically study ConvNet scaling and identify that carefully balancing network width, depth, and resolution is an important but missing piece, preventing us from better accuracy and efficiency. To address this issue, we propose a simple and highly effective compound scaling method, which enables us to easily scale up a baseline ConvNet to any target resource constraints in a more principled way, while maintaining model efficiency. Powered by this compound scaling method, we demonstrate that a mobilesize EfficientNet model can be scaled up very effectively, surpassing state-of-the-art accuracy with an order of magnitude fewer parameters and FLOPS, on both ImageNet and five commonly used transfer learning datasets.

이 백서에서는 ConvNet 스케일링을 체계적으로 연구하여 네트워크 폭, 깊이 및 해상도를 신중하게 균형 있게 조정하는 것이 중요하지만 누락되어 정확성과 효율성이 향상되지 않음을 확인합니다. 이 문제를 해결하기 위해 모델 효율성을 유지하면서 보다 원칙적인 방법으로 기본 ConvNet을 쉽게 타깃 리소스 제약으로 확장할 수 있는 단순하고 매우 효과적인 복합 스케일링 방법을 제안합니다. 이 복합 스케일링 방법을 통해 ImageNet과 일반적으로 사용되는 5개의 전송 학습 데이터 세트 모두에서 모바일 크기 EfficientNet 모델을 매우 효과적으로 스케일업하여 보다 적은 매개 변수와 FLOPS로 최첨단 정확도를 능가할 수 있음을 입증한다.

**Acknowledgements**

We thank Ruoming Pang, Vijay Vasudevan, Alok Aggarwal, Barret Zoph, Hongkun Yu, Jonathon Shlens, Raphael Gontijo Lopes, Yifeng Lu, Daiyi Peng, Xiaodan Song, Samy Bengio, Jeff Dean, and the Google Brain team for their help.

우리는 Ruoming Pang, Vijay Vasudevan, Alok Aggarwal, Barret Zoph, Hongkun Yu, Jonathon Shlens, Raphael Gontio Lopes, Yifeng Lu, Daiyi Peng, Xaodan Song, Samy Bengio, Je Dean, 그리고 그들의 브레인 팀에게 감사한다.

**Appendix**

Since 2017, most research papers only report and compare ImageNet validation accuracy; this paper also follows this convention for better comparison. In addition, we have also verified the test accuracy by submitting our predictions on the 100k test set images to http://image-net.org; results are in Table 8. As expected, the test accuracy is very close to the validation accuracy.

2017년 이후 대부분의 연구논문에서는 ImageNet 검증의 정확성만을 보고 및 비교하고 있습니다.또, 본서에서는, 보다 나은 비교를 위해서, 이 규약을 따르고 있습니다. 또한 10만 테스트 세트 이미지에 대한 예측을 http://image-net.org에 제출하여 테스트 정확성을 확인하였습니다.결과는 표 8에 있습니다. 예상대로 테스트 정확도는 검증 정확도에 매우 가깝습니다.

