EfficientNet

Akshan Sameullah and Maxwell Bai

December 2019

1 Introduction

In George Lucas's *Return of the Jedi* Han Solo awakens after being frozen in Carbonite for 6 months, as a result, he now has temporary blindness and will get eaten by a monster unless he can defeat his enemy, Boba Fett, and escape. Luckily C3-PO, his robot friend, can use a CNN to process what he sees to help Han reach safety. C3-PO knows that CNNs are created with a fixed resource cost that is then scaled up as more resources are made available. For example, Res-net can be scaled up by adding more layers. Because C3-PO is always on an adventure he wants to use a CNN that is reasonably fast, possibly using a MobileNet framework (AutoML approach). First introduced using Compound Model Scaling in 2018, **EfficientNet** surpasses state-of-the-art accuracy and is up to 10x more efficient.

2 Concept

Conventional approach to scaling a model involves increasing depth, width, and resolution. EfficientNet uses a compound coefficient to scale those dimensions in a more balanced manner.

1. Perform a grid search to find the relationship between different scaling dimensions
2. Apply the coefficients found to match the amount of computational resource.

3 Grid Search

Grid Search is the traditional way of hyperparameter tuning and is used here to find the constant coefficients used in compound scaling. Typically, with larger hyperparameters, grid search is more expensive since it is harder to tune. The new scaling method allows us to make smaller grid searches on smaller hyperparameters, then scale it up to get better results in a less expensive way.

4 Scaling

As mentioned earlier, we can scale up a CNN in three ways: by width, by depth, and by resolution. A combination of these is known as compound scaling. The EfficientNet paper proposes a highly efficient **compound coefficient**.

The proposed compound scaling method uses a compound coefficient, ϕ, to balance the dimensions of depth, width, and resolution by scaling them at a certain ratio.

$$depth: \ d = \alpha^{\phi} \quad width: \ w = \beta^{\phi} \quad resolution: \ r = \gamma^{\phi}$$

Computational Resources are leveled in relation to the equation stating that the constant coefficients are limited by the function

$$\alpha \times \beta^2 \times \gamma^2 \approx 2$$

$$\alpha \geq 1, \ \beta \geq 1, \ \gamma \geq 1$$

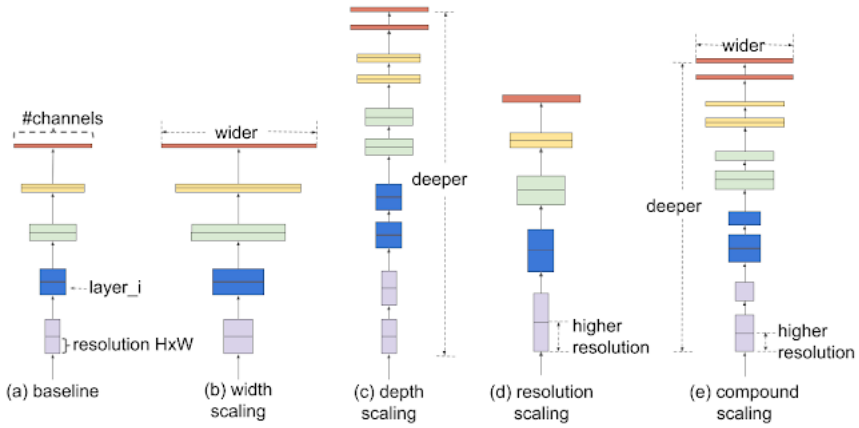


Figure 1: Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

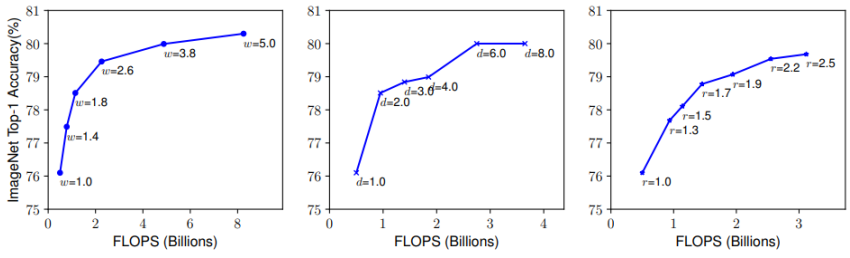


Figure 2: Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients. Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling.

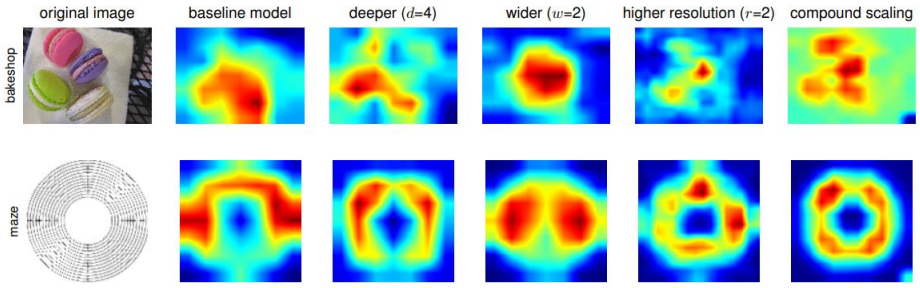


Figure 3: Here is a map of the details of the images on the left that the model can focus on

5 Architecture

EfficientNet still however, depends on the baseline network since that is something model scaling depends on. A Neural Architecture search is done to develop a new baseline net. It also uses a MobileNet to the extent of maintaining FLOPS (Floating Point Operations per Second) budget. The baseline is then scaled up to result a family of models known as EfficientNets. EfficientNet has a high efficiency partially because of the optimization goal denoted by the following equation:

$$ACC(m) \times [FLOPS(m)/T]^w$$

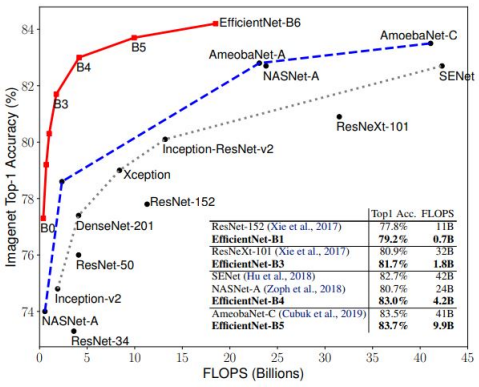
The baseline network, EfficientNet-B0, was produced using a search space similar to the one used to produce Mnas-Net, and as such has a similar architecture, only slightly bigger because of a larger FLOPS target. The main building block of the EfficientNet-B0 architecture is the inverted residual and linear bottleneck layer, introduced in MobileNetV2.

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**77.3%**	**93.5%**	**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	**79.2%**	**94.5%**	**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	**80.3%**	**95.0%**	**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	**81.7%**	**95.6%**	**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	**83.0%**	**96.3%**	**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	**83.7%**	**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	**84.2%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.4%**	**97.1%**	**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%**



6 Performance on ImageNet

At EfficientNet-B3, an accuracy better than ResNeXt-101 was reached using 18x fewer FLOPS. By EfficientNet-B7, results as good as 84.4% top1 / 97.1% top-5 accuracy was reached, more accurate than GPipe but 8.4 times smaller, and 6.1 times faster.

7 Extra

If you want to read more about EfficientNets and compound scaling you can find it on the Google AI Blog. It is very interesting.

https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

If you want to know the specifics of the inverted residual and linear bottleneck layer, you can find more information from the MobileNetV2 paper.

https://arxiv.org/abs/1801.04381

8 Acknowledgements

The figures used in this lecture were not created by us. Credit goes to:

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks by Mingxing Tan, Quoc V. Le