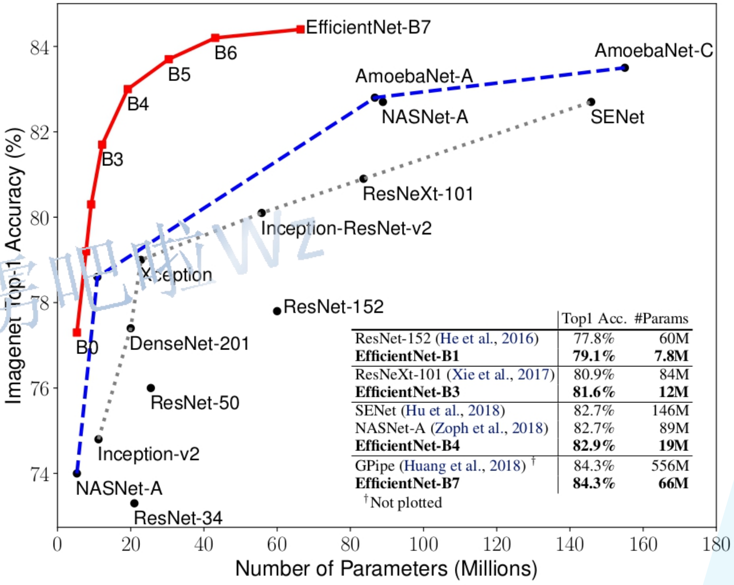
EfficientNet

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Mingxing Tan [1] **Quoc V. Le** [1]

Google2019发表的文章

在论文中提到，本文提出的EfficientNet-B7在Imagenet top-1上达到了当年最高准确率84.3%，与之前准确率最高的GPipe相比，参数数量仅为其1/8.4，推理速度提升了6.1倍。



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

- 同时探索输入分辨率，网络的深度、宽度的影响

论文下载地址：https://arxiv.org/abs/1905.11946
推荐博文：https://blog.csdn.net/qq_37541097/article/details/114434046

EfficientNet

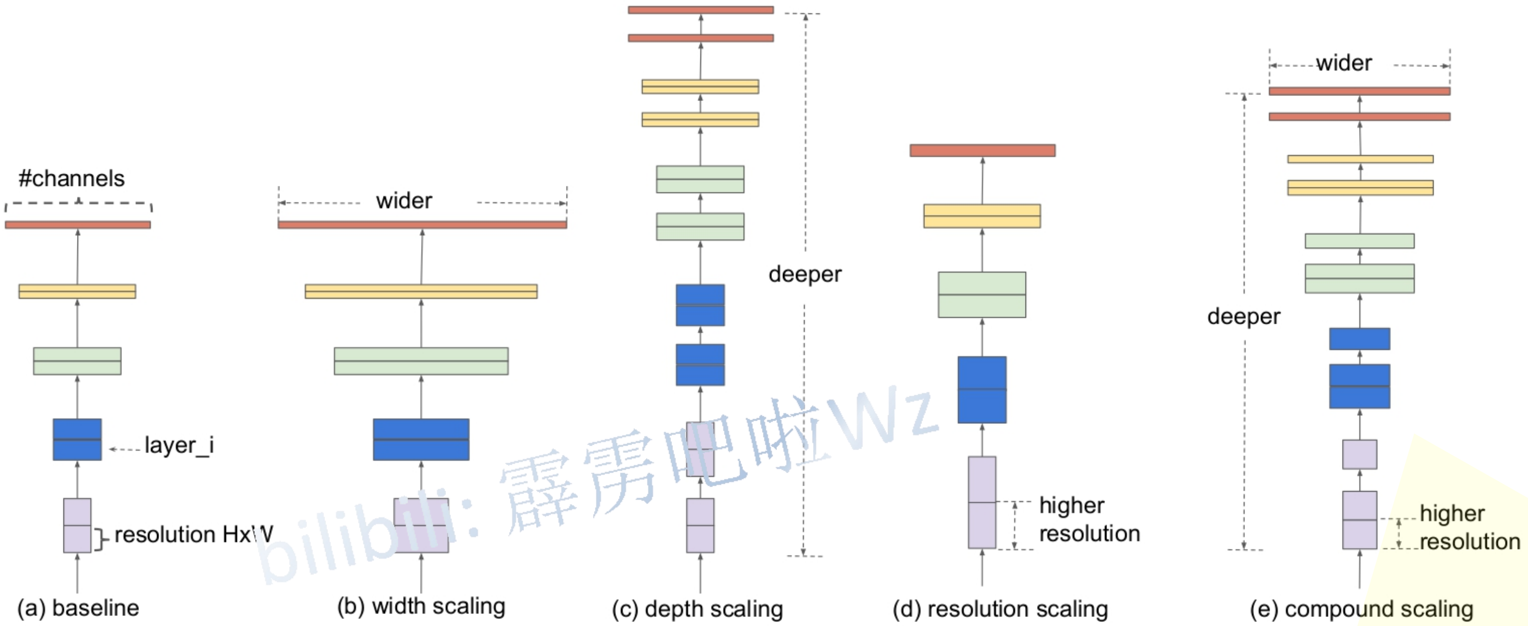
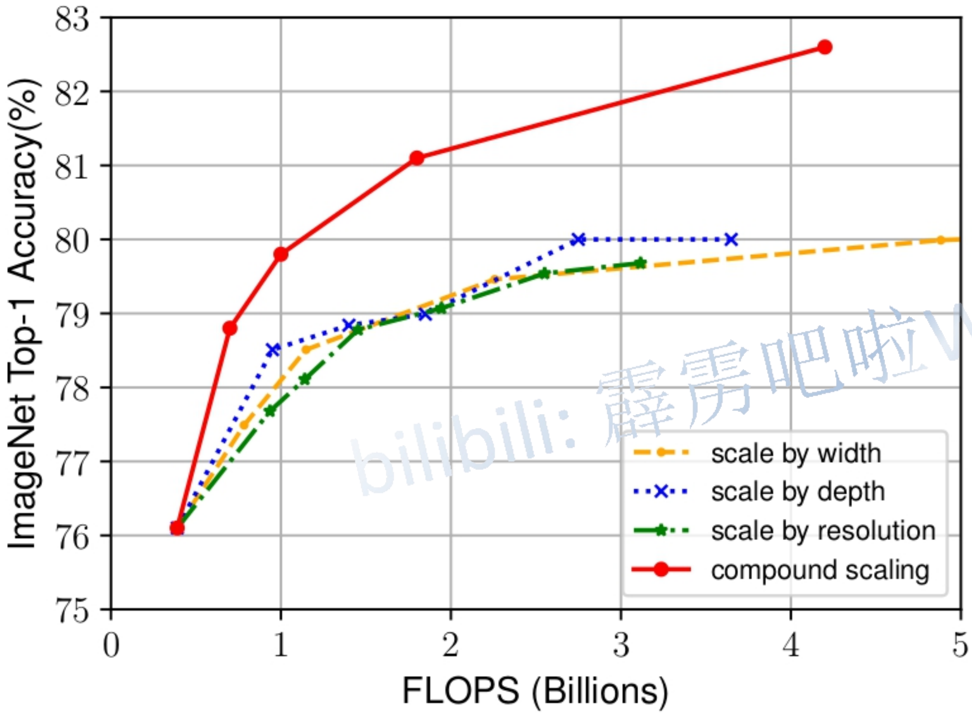


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

EfficientNet



- 根据以往的经验，增加网络的深度depth能够得到更加丰富、复杂的特征并且能够很好的应用到其它任务中。但网络的深度过深会面临梯度消失，训练困难的问题。
- 增加网络的width能够获得更高细粒度的特征并且也更容易训练，但对于width很大而深度较浅的网络往往很难学习到更深层次的特征。
- 增加输入网络的图像分辨率能够潜在得获得更高细粒度的特征模板，但对于非常高的输入分辨率，准确率的增益也会减小。并且大分辨率图像会增加计算量。

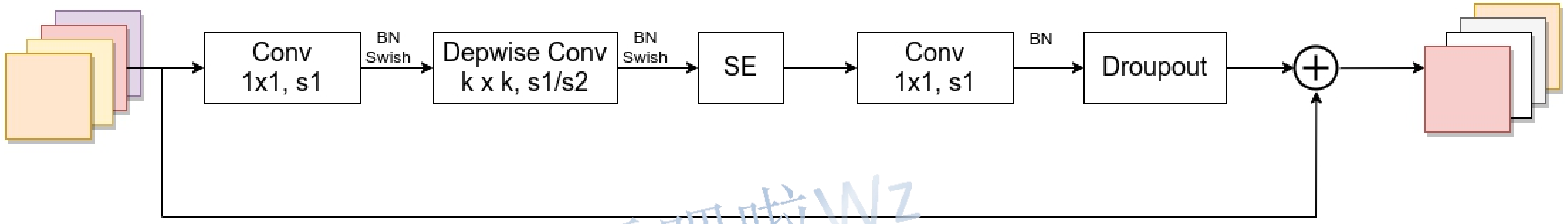
EfficientNet

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 $\hat{L}_i$	stride
1	Conv3x3	224×224	32	1	2
2	MBConv1, k3x3	112×112	16	1	1
3	MBConv6, k3x3	112×112	24	2	2
4	MBConv6, k5x5	56×56	40	2	2
5	MBConv6, k3x3	28×28	80	3	2
6	MBConv6, k5x5	14×14	112	3	1
7	MBConv6, k5x5	14×14	192	4	2
8	MBConv6, k3x3	7×7	320	1	1
9	Conv1x1 & Pooling & FC	7×7	1280	1	

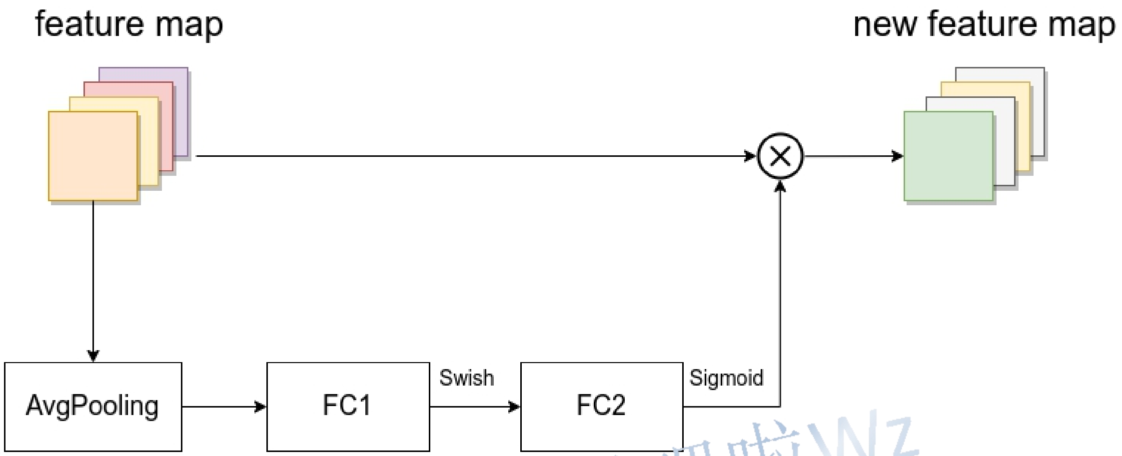
EfficientNet

MBConv



- 第一个升维的1x1卷积层，它的卷积核个数是输入特征矩阵channel的n倍
- 当 n = 1 时，不要第一个升维的1x1卷积层，即Stage2中的MBConv结构都没有第一个升维的1x1卷积层（这和MobileNetV3网络类似）
- 关于shortcut连接，仅当输入MBConv结构的特征矩阵与输出的特征矩阵shape相同时才存在

EfficientNet



SE模块如图所示，由一个全局平均池化，两个全连接层组成。第一个全连接层的节点个数是输入该MBConv特征矩阵channels的1/4，且使用Swish激活函数。第二个全连接层的节点个数等于Depthwise Conv层输出的特征矩阵channels，且使用Sigmoid激活函数。

EfficientNet

Model	input_size	width_coefficient	depth_coefficient	drop_connect_rate	dropout_rate
EfficientNetB0	224x224	1.0	1.0	0.2	0.2
EfficientNetB1	240x240	1.0	1.1	0.2	0.2
EfficientNetB2	260x260	1.1	1.2	0.2	0.3
EfficientNetB3	300x300	1.2	1.4	0.2	0.3
EfficientNetB4	380x380	1.4	1.8	0.2	0.4
EfficientNetB5	456x456	1.6	2.2	0.2	0.4
EfficientNetB6	528x528	1.8	2.6	0.2	0.5
EfficientNetB7	600x600	2.0	3.1	0.2	0.5

- width_coefficient代表channel维度上的倍率因子，比如在 EfficientNetB0中Stage1的3x3卷积层所使用的卷积核个数是32，那么在B6中就是 $32 \times 1.8 = 57.6$接着取整到离它最近的8的整数倍即56，其它Stage同理。

- depth_coefficient代表depth维度上的倍率因子（**仅针对Stage2到Stage8**），比如在EfficientNetB0中Stage7的 $L = 4$ ，那么在B6中就是 $4 \times 2.6 = 10.4$，接着向上取整即11.

EfficientNet

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).

沟通方式

1.github

https://github.com/WZMIAOMIAO/deep-learning-for-image-processing

2.bilibili

https://space.bilibili.com/18161609/channel/index

3.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003