EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks

团队：Datawhale 深度学习团队

汇报人： 苏静静

2019.8.1

目录

一 相关介绍

二 论文摘要

三 论文思路

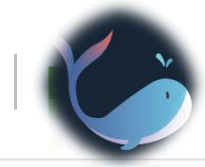
四 实验与结果

五 结论

六 总结

相关介绍

01 相关介绍



EfficientNets作者是来自谷歌大脑的工程师Mingxing Tan和首席科学家Quoc V. Le。
目前论文已被ICML(International Conference on Machine Learning)2019收录。

01 论文作者
02 相关知识

01 论文作者

02 相关知识

深度学习模型压缩

训练后的深度神经网络模型往往存在严重的 **过参数化** 问题，其中只有约5%的参数子集是真正有用的。为此，对模型进行 **时间** 和 **空间** 上的压缩，称为“**模型压缩**”。

模型压缩技术包括 **前端压缩** 和 **后端压缩** 。

	前端压缩	后端压缩
实现难度	较简单	较难
可逆否	可逆	不可逆
成熟运用	剪枝	低秩近似、参数量化
待发展运用	知识蒸馏	二值网络

01 论文作者

02 相关知识

■ **前端压缩：**不会改变原始网络结构的压缩技术。

□ **知识蒸馏**

教师—学生网络方法，属于迁移学习的，也就是将一个模型的性能迁移到另一个模型上。

□ **模型结构设计**

SqueezeNet的fire module，ResNet的Residual module，GoogLenet的Inception Module，基本都是由很小的卷积（1*1和3*3）组成，不仅参数运算量小，同时还具备了很好的性能效果。

□ **剪枝**

■ **后端压缩：**会大程度上改变原始网络结构的压缩技术，且不可逆。

□ **低秩近似**

使用结构化矩阵来进行低秩分解。

□ **参数量化**

从权重中 归纳出 若干的“代表”，由这些“代表”来 表示某一类权重 的 具体数值 。

□ **二值网络**

参数的取值 只能是 ±1 。

01 相关介绍

01 论文作者

02 相关知识

卷积神经网络（CNN）通常以固定成本开发，然后再按比例放大，从而在获得更多资源时可以达到更高的准确率。

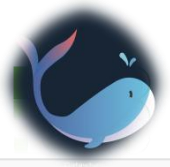
ResNet

可以通过增加网络层数，从 ResNet-18 扩展到 ResNet-200。

模型缩放的通常做法是任意增加 CNN 的深度或宽度，或者使用更大的输入图像分辨率进行训练和评估。

尽管这些方法确实可以改进准确率，但它们通常需要大量手动调参，且通常获得的是次优性能。那么，我们是否可以寻找更好的 CNN 扩展方法，来获得更高的准确率和效率呢？

论文摘要

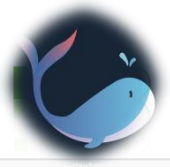


- 论文提出了一种新型**模型缩放**（model scaling）方法，该方法使用一种简单但高效的**复合系数**（compound coefficient）以更加结构化的方式扩展 CNN。与任意扩展网络维度（如宽度、深度、分辨率）的传统方法不同，该新方法使用固定的一组缩放系数扩展每个维度。

- 受益于上述方法和AutoML的最新进展，谷歌开发出了一系列模型——**EfficientNets**，该模型的准确率超越了当前最优模型，且效率是后者的 10 倍（模型更小，速度更快）。

论文思路

03 论文思路



01 模型缩放
02 EfficientNet

- **复合模型缩放：扩展 CNN 的更好方法**

研究人员系统研究了缩放模型不同维度的影响。虽然缩放单个维度可以改善模型性能，但研究人员发现**平衡网络的所有维度（宽度、深度和图像分辨率）和可用资源**才能最优地提升整体性能。

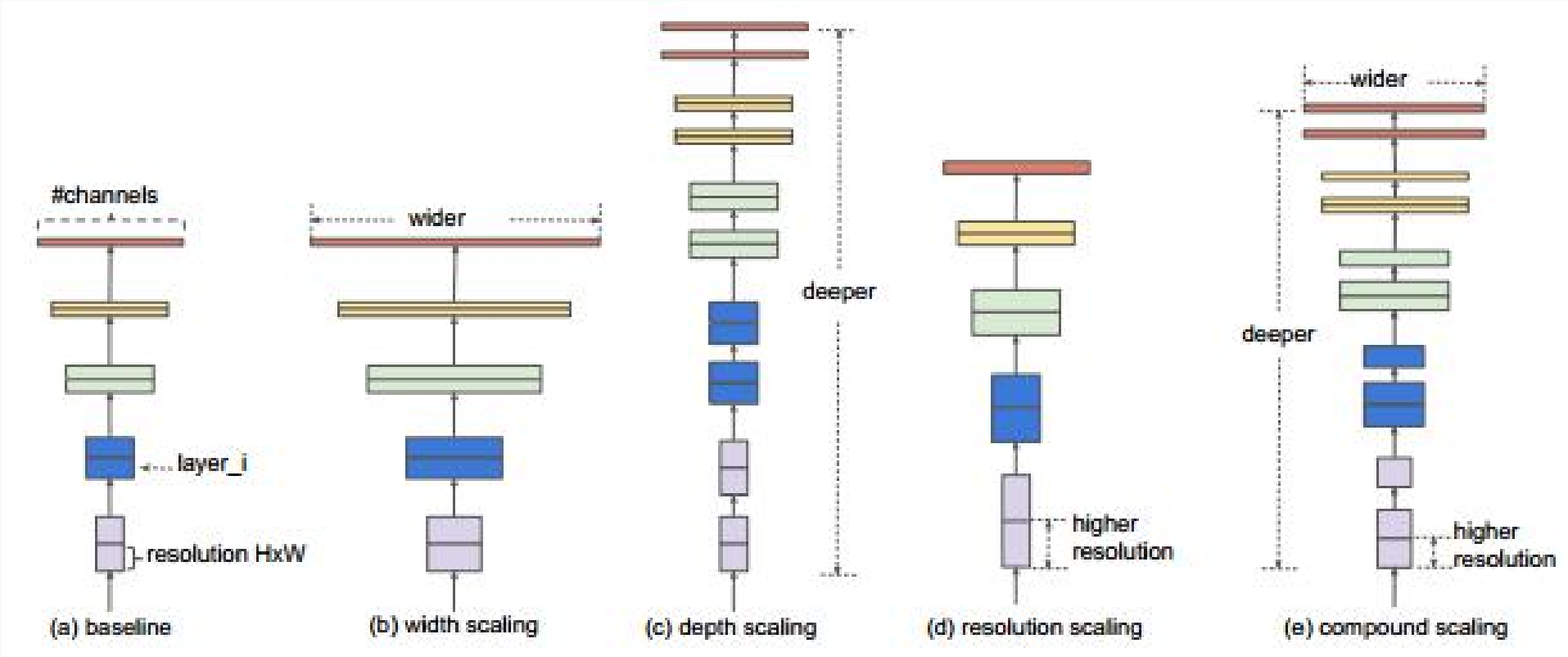
该复合缩放方法的第一步就是**执行网格搜索**，寻找固定资源限制下基线模型不同缩放维度之间的关系。这决定了每个维度的合适的缩放系数。第二步是**应用这些系数**，将基线网络扩展到目标模型大小或目标计算成本。

03 论文思路

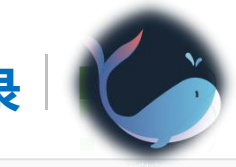
目录

01 模型缩放

02 EfficientNet



不同缩放方法对比。传统缩放方法 (b)-(d) 任意缩放模型的单个维度，而谷歌提出的新型复合缩放方法则不同，它扩展模型的所有维度。



01 模型缩放

02 EfficientNet

单个维度做scaling存在什么问题？

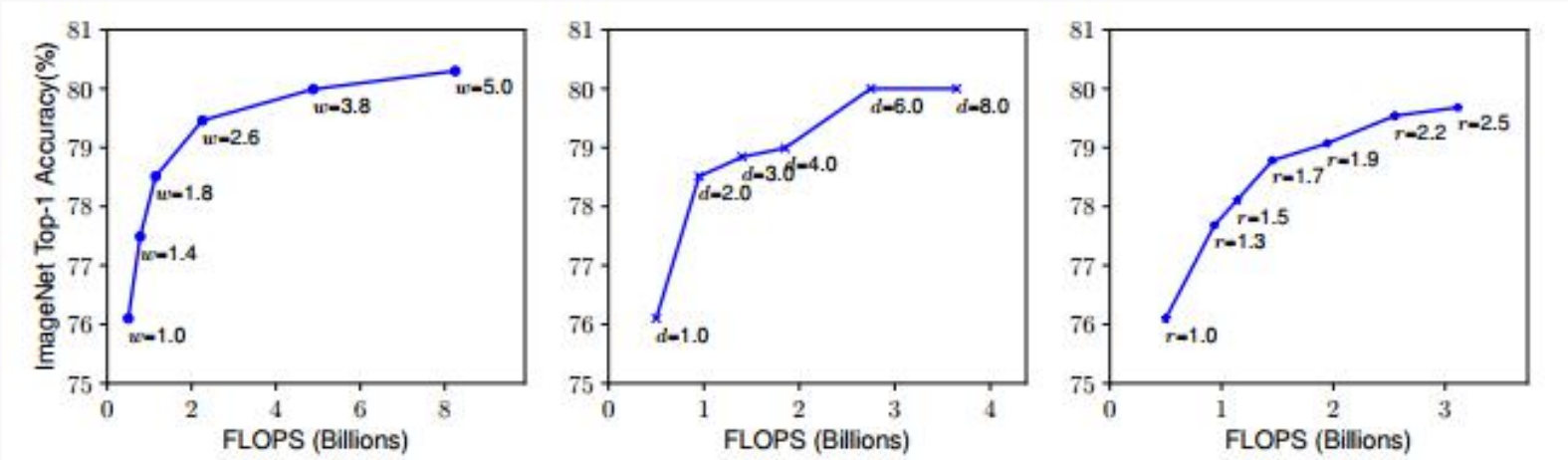
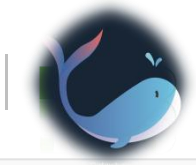


Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single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

单独优化这3个维度都能提升模型效果，但上限也比较明



01 模型缩放

02 EfficientNet

结合多个维度做scaling效果如何？

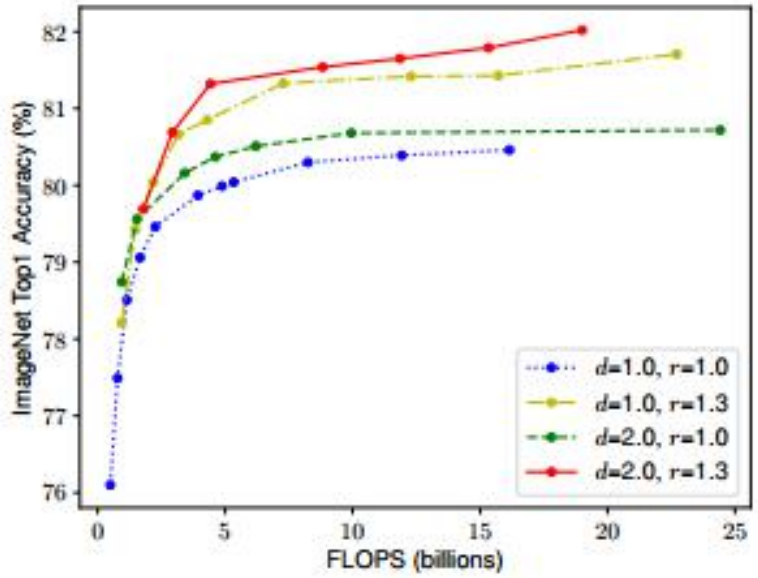
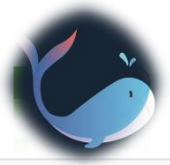


Figure 4. **Scaling Network Width for Different Baseline Networks**. Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline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

蓝线是只对宽度做model scaling的结果，每个点表示不同宽度的网络，因此不同线条上相同顺序的点表示的网络宽度设置是一样的。通过手动设置3个维度的model scaling参数就能有效提升模型效果，说明多维度融合是有效的。



01 模型缩放

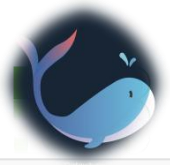
02 EfficientNet

如何选择最优的3个维度scaling参数？compound model scaling

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

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

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



01 模型缩放

02 EfficientNet

通过网络结构搜索设计了一个baseline网络， EfficientNet-B0

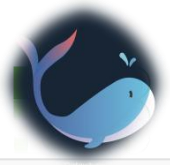
Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

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	28 × 28	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

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

- STEP 1: we first fix $\phi = 1$, 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 $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, under constraint of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.
- 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).

实验与结果



01 实验结果

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**76.3%**	**93.2%**	**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	**78.8%**	**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	**79.8%**	**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	**81.1%**	**95.5%**	**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	**82.6%**	**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	**83.3%**	**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%**	**96.9%**	**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).

01 实验结果

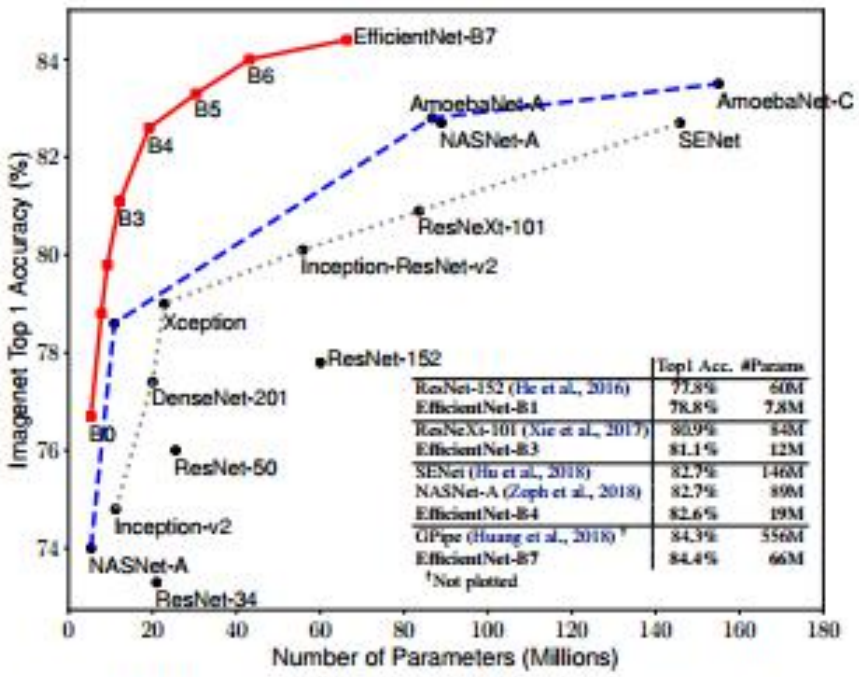


Figure 1. **Model Size vs. ImageNet Accuracy.** All numbers are for single-crop, single-model. Our EfficientNets significantly outperform other ConvNets. In particular, EfficientNet-B7 achieves new state-of-the-art 84.4% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

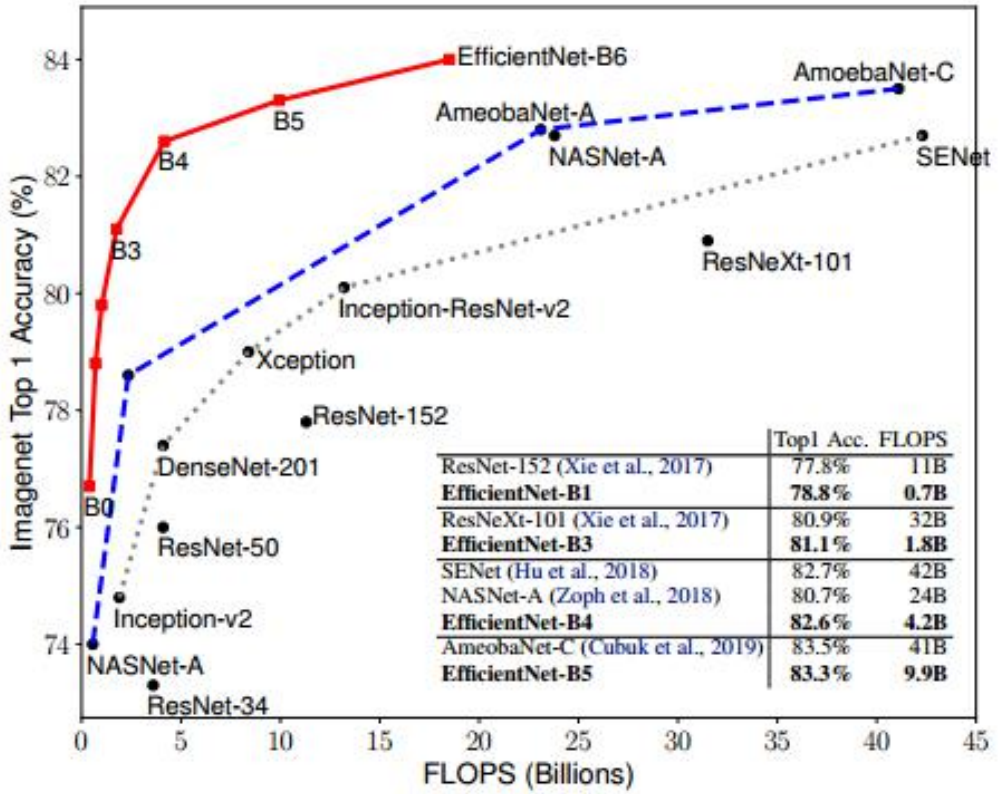
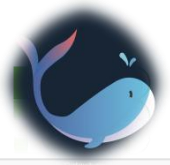


Figure 5. **FLOPS vs. ImageNet Accuracy –** Similar to Figure 1 except it compares FLOPS rather than model size.



01 实验结果

泛化性

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%**

01 实验结果

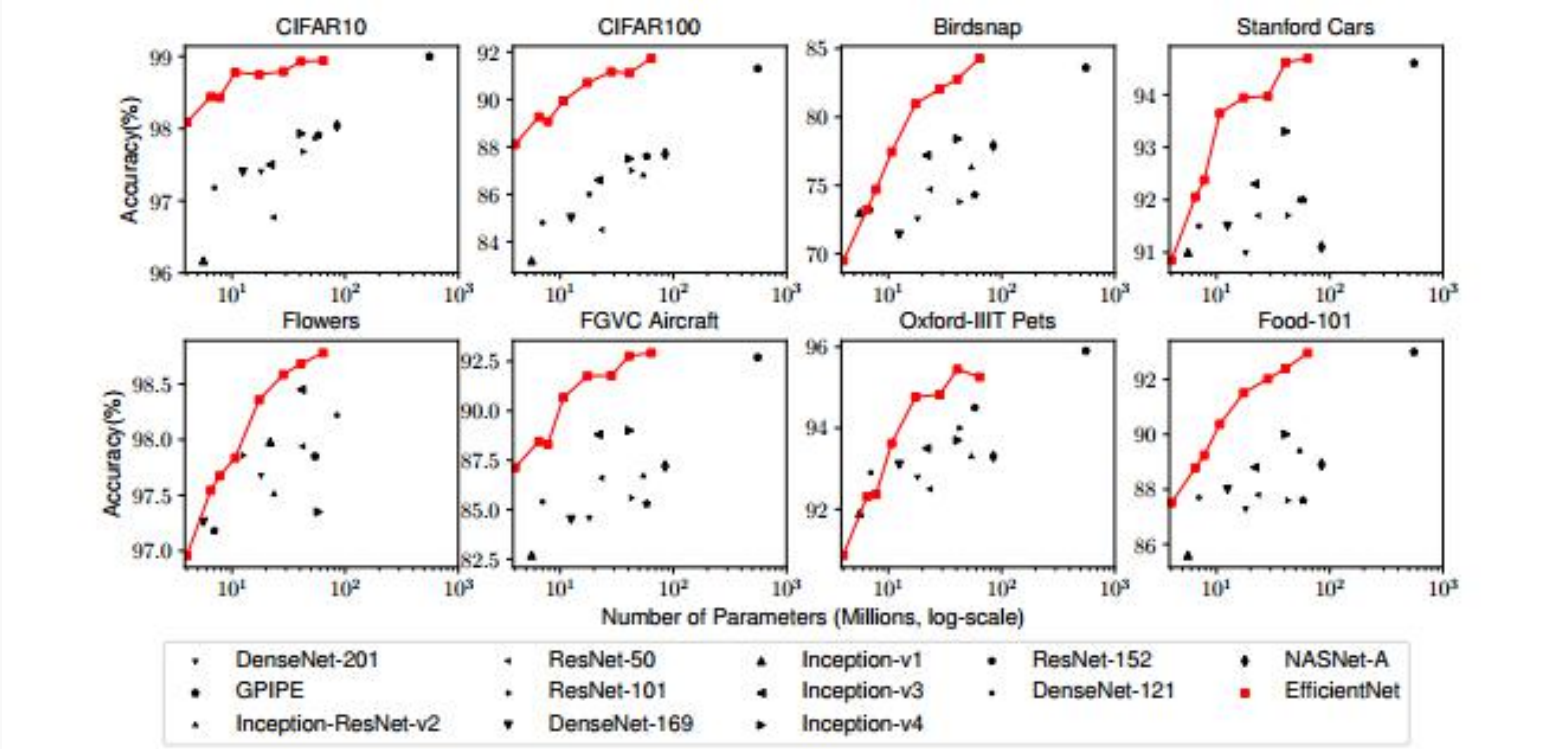


Figure 6. **Model Parameters vs. Transfer Learning Accuracy –** All models are pretrained on ImageNet and finetuned on new datasets.

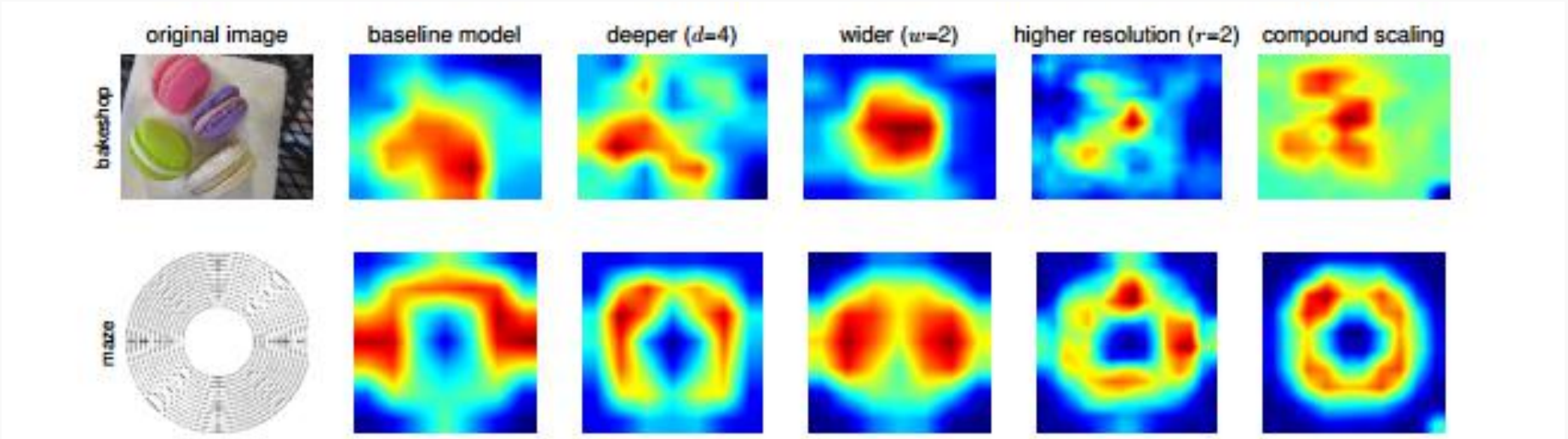


Figure 7. **Class Activation Map (CAM) (Zhou et al., 2016) for Different Models in Table 7 -** Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

01 实验结果

结论

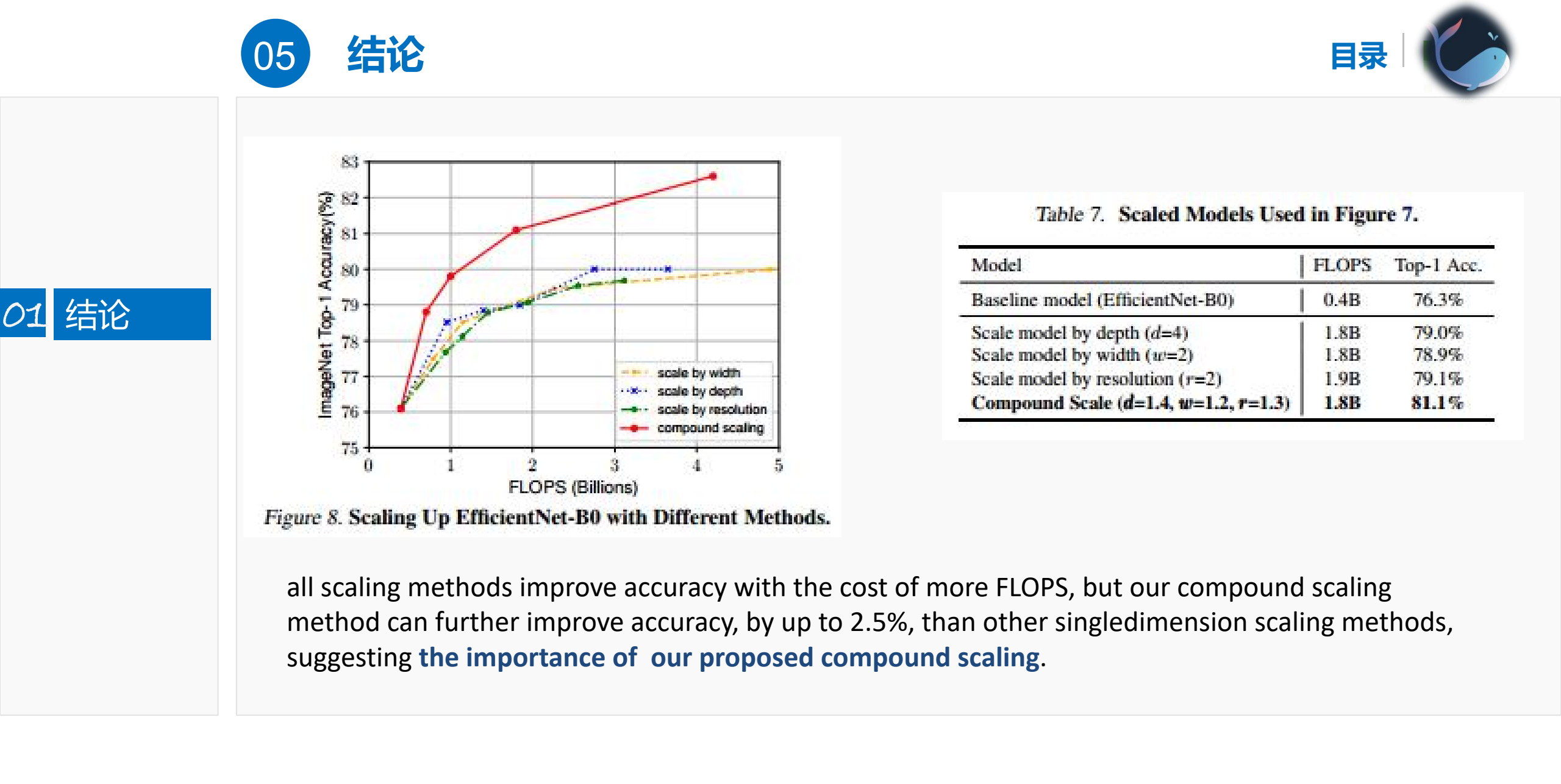
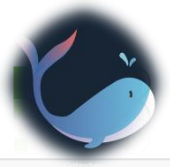


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

Table 7. **Scaled Models Used in Figure 7.**

Model	FLOPS	Top-1 Acc.
Baseline model (EfficientNet-B0)	0.4B	76.3%
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%**

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**.



- 该方法使用一种简单但高效的**复合系数**（compound coefficient）以更加结构化的方式扩展 CNN。与任意扩展网络维度（如宽度、深度、分辨率）的传统方法不同，该新方法使用固定的一组缩放系数扩展每个维度。

- **EfficientNets系列模型，**模型更小、效率和准确率更高。

总结

总结

Summary

总结 SUMMARY

- 传统CNN模型总有很多冗余参数
- 网络的设计调整受限于软件、硬件资源
- AutoML 和NAS(Neural Architecture Search)是s深度学习领域重要的方向

资源
Sources

资源 SOURCES

论文链接：
https://arxiv.org/abs/1905.11946

代码链接：
https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

PyTorch实现代码：
https://github.com/lukemelas/EfficientNet-PyTorch

敬请各位大佬批评指正