EfficientNet：重新思考卷积神经网络的模型扩展

Mingxing Tan Quoc V. Le

Abstract

卷积神经网络（ConvNets）通常是在固定的资源预算下开发的，然后在有更多资源的情况下进行扩展，以获得更高的精度。在本文中，我们对模型缩放进行了系统研究，发现仔细平衡网络深度、宽度和分辨率可以带来更好的性能。基于这一观点，我们提出了一种新的缩放方法，利用简单而高效的复合系数统一缩放深度/宽度/分辨率的所有维度。我们演示了这种方法在扩展 MobileNets 和 ResNet 时的有效性。

为了更进一步，我们利用神经架构搜索设计了一个新的基线网络，并将其扩展，从而获得了一系列称为 EfficientNets 的模型，这些模型比以前的 ConvNets 获得了更高的准确率和效率。特别是，我们的 EfficientNet-B7 在 ImageNet 上达到了最先进的 84.3% top-1 准确率，同时比现有最佳 ConvNet 的体积小 8.4 倍，推理速度快 6.1 倍。在 CIFAR-100 (91.7%)、Flowers (98.8%) 和其他 3 个迁移学习数据集上，我们的 EfficientNets 也能很好地进行迁移，并达到最先进的准确率，而且参数数量级更低。源代码在 https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet.

1. 导言

扩展 ConvNets 被广泛用于实现更高的精度。例如，ResNet（He 等人，2016）可以通过使用更多层从 ResNet-18 扩展到 ResNet-200；最近，GPipe（Huang 等人，2018）通过将基线模型扩展四倍，实现了 84.3% 的 ImageNet top-1 准确率。然而，ConvNets 的扩展过程一直没有得到很好的理解，目前有很多方法可以做到这一点。最常见的方法是通过深度（He 等人，2016 年）或宽度（Zagoruyko & Komodakis，2016 年）来放大 ConvNets。另一种不太常见但越来越流行的方法是按图像分辨率放大模型（Huang 等人，2018 年）。在以往的工作中，通常只对深度、宽度和图像大小这三个维度中的一个维度进行缩放。虽然可以任意缩放两个或三个维度，但任意缩放需要繁琐的手动调整，而且通常仍会产生次优精度和效率。

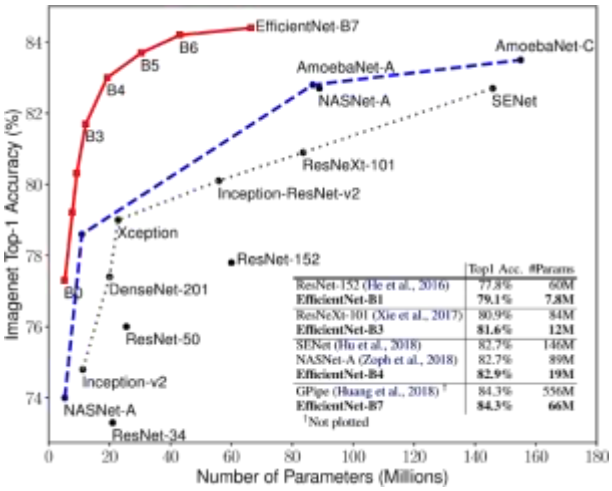
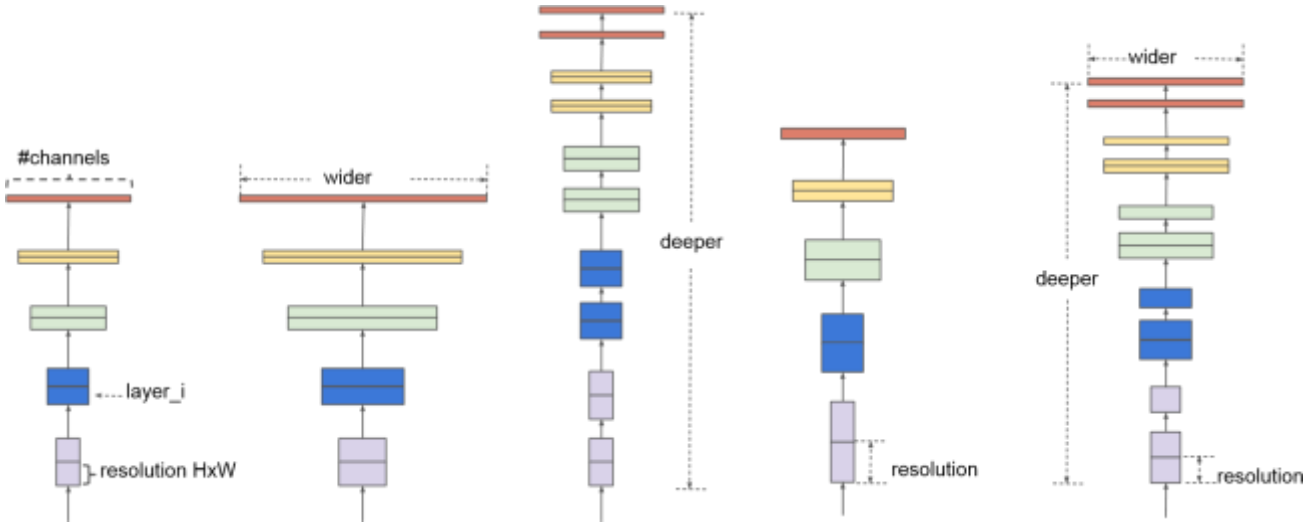


图 1.模型大小与 ImageNet 精度的关系。所有数据均为单作物、单模型数据。我们的高效网络明显优于其他 ConvNets。其中，EfficientNet-B7 达到了最先进的 84.3% top-1 准确率，但体积比 GPipe 小 8.4 倍，速度比 GPipe 快 6.1 倍。EfficientNet-B1 比 ResNet-152 小 7.6 倍，速度快 5.7 倍。详情见表 2 和表 4。



(a) Baseline (b) width scaling (c) depth scaling (d) resolution scaling (e)compound scaling

图 2：模型缩放模型缩放。(a) 为基线网络示例；(b)-(d) 为传统缩放法，只增加网络宽度、深度或分辨率中的一个维度。(e)是我们提出的复合缩放方法，以固定比例统一缩放所有三个维度。

本文希望研究并重新思考 ConvNets 的扩展过程。特别是，我们研究的核心问题是：是否有一种原则性的方法来扩展 ConvNets，从而达到更好的精度和效率？我们的实证研究表明，平衡网络的宽度/深度/分辨率等所有维度至关重要，而令人惊讶的是，这种平衡可以通过简单地以恒定比率缩放每个维度来实现。基于这一观察结果，我们提出了一种简单而有效的复合缩放方法。与任意缩放这些因子的传统做法不同，我们的方法通过一组固定的缩放系数来统一缩放网络宽度、深度和分辨率。例如，如果我们想使用 2N 倍的计算资源，那么我们只需将网络深度增加 α^N，宽度增加 β^N，图像尺寸增加 γ^N，其中 α、β、γ 是通过对原始小型模型进行小网格搜索确定的常数。图 2 说明了我们的缩放方法与传统方法的区别。

直观地说，复合缩放法是有道理的，因为如果输入图像更大，那么网络就需要更多的层来增加感受野，需要更多的通道来捕捉更大图像上更细粒度的模式。事实上，之前的理论（Raghu 等人，2017 年；Lu 等人，2018 年）和实证结果（Zagoruyko & Komodakis，2016 年）都表明，网络宽度和深度之间存在一定的关系，但据我们所知，我们是第一个对网络宽度、深度和分辨率这三个维度之间的关系进行实证量化的人。

我们证明，我们的扩展方法在现有的 MobileNets（Howard 等人，2017 年；Sandler 等人，2018 年）和 ResNet（He 等人，2016 年）上运行良好。值得注意的是，模型缩放的效果在很大程度上取决于基线网络；为了更进一步，我们使用神经架构搜索（Zoph & Le，2017；Tan 等人，2019）开发了一个新的基线网络，并将其放大以获得一个模型系列，称为 EfficientNets。图 1 总结了 ImageNet 的性能，其中我们的 EfficientNets 明显优于其他 ConvNets。特别是，我们的 EfficientNet-B7 超过了现有最佳 GPipe 准确率（Huang 等人，2018 年），但使用的参数减少了 8.4 倍，推理速度提高了 6.1 倍。与广泛使用的 ResNet-50（He 等人，2016 年）相比，我们的 EfficientNet-B4 在 FLOPS 相近的情况下，将 top-1 准确率从 76.3% 提高到 83.0%（+6.7%）。除 ImageNet 外，效能网还在 8 个广泛使用的数据集中的 5 个数据集上实现了很好的转移并达到了最先进的准确率，同时比现有的 ConvNets 减少了多达 21 倍的参数。

2. 相关工作

ConvNet 精确度：自 AlexNet（Krizhevsky 等人，2012 年）在 2012 年 ImageNet 比赛中获胜以来，ConvNets 的规模越来越大，精确度也越来越高：2014 年 ImageNet 冠军 GoogleNet（Szegedy 等人，2015 年）使用约 680 万个参数实现了 74.8% 的 top-1 精确度，而 2017 年 ImageNet 冠军 SENet（Hu 等人，2018 年）使用 1.45 亿个参数实现了 82.7% 的 top-1 精确度。最近，GPipe（Huang 等人，2018）使用 5.57 亿个参数将最先进的 ImageNet top-1 验证准确率进一步推高到 84.3%：它是如此之大，以至于只能通过分割网络并将每个部分分散到不同的加速器上，使用专门的流水线并行库进行训练。虽然这些模型主要是为 ImageNet 设计的，但最近的研究表明，更好的 ImageNet 模型在各种迁移学习数据集（Kornblith 等人，2019 年）和其他计算机视觉任务（如物体检测）中也有更好的表现（He 等人，2016 年；Tan 等人，2019 年）。虽然更高的精度对许多应用来说至关重要，但我们已经达到了硬件内存的极限，因此进一步提高精度需要更好的效率。

ConvNet 效率：深度 ConvNet 通常参数过高。模型压缩（Han 等人，2016 年；He 等人，2018 年；Yang 等人，2018 年）是通过以精度换效率来缩小模型大小的常用方法。随着手机变得无处不在，手工制作高效的移动大小 ConvNets 也很常见，如 SqueezeNets（Iandola 等人，2016 年；Gholami 等人，2018 年）、MobileNets（Howard 等人，2017 年；Sandler 等人，2018 年）和 ShuffleNets（Zhang 等人，2018 年；Ma 等人，2018 年）。最近，神经架构搜索在设计高效移动大小 ConvNets 方面变得越来越流行（Tan 等人，2019；Cai 等人，2019），通过广泛调整网络宽度、深度、卷积核类型和大小，实现了比手工制作的移动 ConvNets 更好的效率。然而，目前还不清楚如何将这些技术应用于设计空间更大、调整成本更高的大型模型。本文旨在研究超大 ConvNets 的模型效率，以超越最先进的精度。为了实现这一目标，我们采用了模型缩放技术。

模型缩放：针对不同的资源限制，有很多方法可以缩放 ConvNet：ResNet（He 等人，2016）可以通过调整网络深度（#层）来缩小（如 ResNet-18）或放大（如 ResNet-200），而 WideResNet（Zagoruyko & Komodakis，2016）和 MobileNets（Howard 等人，2017）可以通过网络宽度（#通道）来缩放。此外，人们还认识到，输入图像尺寸越大，精确度越高，而 FLOPS 的开销也越大。尽管之前的研究（Raghu 等人，2017；Lin & Jegelka，2018；Sharir & Shashua，2018；Lu 等人，2018）已经表明，网络深度和宽度对 ConvNets 的表现力都很重要，但如何有效地扩展 ConvNet 以实现更好的效率和准确性，仍然是一个未决问题。我们的工作从网络宽度、深度和分辨率三个维度对 ConvNet 的扩展进行了系统性的实证研究。

3. 复合模型缩放

在本节中，我们将提出缩放问题，研究不同的方法，并提出我们新的缩放方法。

3.1. 问题的提出

ConvNet 第 i 层可定义为一个函数：Yi = Fi(Xi)，其中 Fi 是算子，Yi 是输出张量，Xi 是输入张量，张量形状为 hHi,Wi,Cii，其中 Hi 和 Wi 是空间维度，Ci 是通道维度。ConvNet N 可以用组成层的列表来表示： 。在实践中，ConvNet 层通常被划分为多个阶段，每个阶段中的所有层都采用相同的架构：例如，ResNet（He et al、例如，ResNet（He 等人，2016）有五个阶段，除了第一层执行下采样外，每个阶段的所有层都具有相同的卷积类型。因此，我们可以将 ConvNet 定义为：

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

其中，F 表示层 Fi 在阶段 I 中重复了 Li 次，<Hi,Wi,Ci> 表示层 I 的输入张量 X 的形状。图 2(a) 展示了一个具有代表性的 ConvNet，其中空间维度逐渐缩小，但通道维度却逐层扩大，例如，从初始输入形状 <224,224,3> 到最终输出形状 <7,7,512> 。

一般的 ConvNet 设计主要集中在寻找最佳层架构 Fi，而模型缩放则不同，它试图在不改变基线网络中预定义的 Fi 的情况下，扩展网络长度（Li）、宽度（Ci）和/或分辨率（Hi,Wi）。通过固定 Fi，模型缩放简化了新资源限制的设计问题，但要为每一层探索不同的 Li、Ci、Hi、Wi，设计空间仍然很大。为了进一步缩小设计空间，我们限制所有层必须以恒定比率均匀缩放。我们的目标是在任何给定的资源限制条件下最大限度地提高模型精度，这可以表述为一个优化问题：

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

其中，w,d,r 为网络宽度、深度和分辨率的缩放系数；F^i,L^i,H^i,W^i,C^i 为基线网络的预定义参数（以表 1 为例）。

3.2. 缩放尺寸

问题 2 的主要难点在于最佳 d、w、r 值相互依赖，而且在不同的资源限制条件下，其值也会发生变化。由于这一困难，传统方法大多在其中一个维度上对 ConvNets 进行缩放：

深度（d）：缩放网络深度是许多 ConvNets 最常用的方法（He 等人，2016；Huang 等人，2017；Szegedy 等人，2015；2016）。其直觉是，更深的 ConvNet 可以捕捉到更丰富、更复杂的特征，并能很好地泛化到新任务中。然而，由于梯度消失问题（Zagoruyko & Komodakis，2016 年），深度网络也更难训练。虽然跳过连接（He 等人，2016 年）和批量归一化（Ioffe & Szegedy，2015 年）等几种技术可以缓解训练问题，但深度网络的准确率增益却在减少：例如，ResNet-1000 的准确率与 ResNet-101 相近，尽管它的层数要多得多。图 3（中）显示了我们用不同深度系数 d 对基线模型进行缩放的实证研究，进一步说明了深度 ConvNets 的准确率收益在不断降低。

宽度（w）：缩放网络宽度通常用于小型模型（Howard 等人，2017 年；Sandler 等人，2018 年；Tan 等人，2019 年）。正如（Zagoruyko & Komodakis, 2016）中所讨论的，更宽的网络往往能够捕捉到更多细粒度的特征，也更容易训练。然而，极宽但较浅的网络往往难以捕捉更高层次的特征。图 3（左）中的实证结果表明，当网络变得更宽、w 越大时，准确率很快就会达到饱和。

分辨率 (r)：有了更高分辨率的输入图像，ConvNets 就有可能捕捉到更精细的模式。从早期 ConvNets 的 224x224 开始，现代 ConvNets 倾向于使用 299x299（Szegedy 等人，2016 年）或 331x331（Zoph 等人，2018 年），以获得更高的精度。最近，GPipe（Huang 等人，2018 年）以 480x480 的分辨率达到了最先进的 ImageNet 准确度。更高的分辨率，如 600x600，也广泛应用于物体检测 ConvNets（He 等人，2017；Lin 等人，2017）。图 3（右）显示了网络分辨率缩放的结果，分辨率越高，准确率越高，但分辨率越高，准确率的提高就越小（r = 1.0 表示分辨率为 224x224，r = 2.5 表示分辨率为 560x560）。

通过上述分析，我们得出了第一个结论：

观察结果 1 - 放大网络宽度、深度或分辨率的任何维度都能提高精确度，但模型越大，精确度的提高幅度就越小。

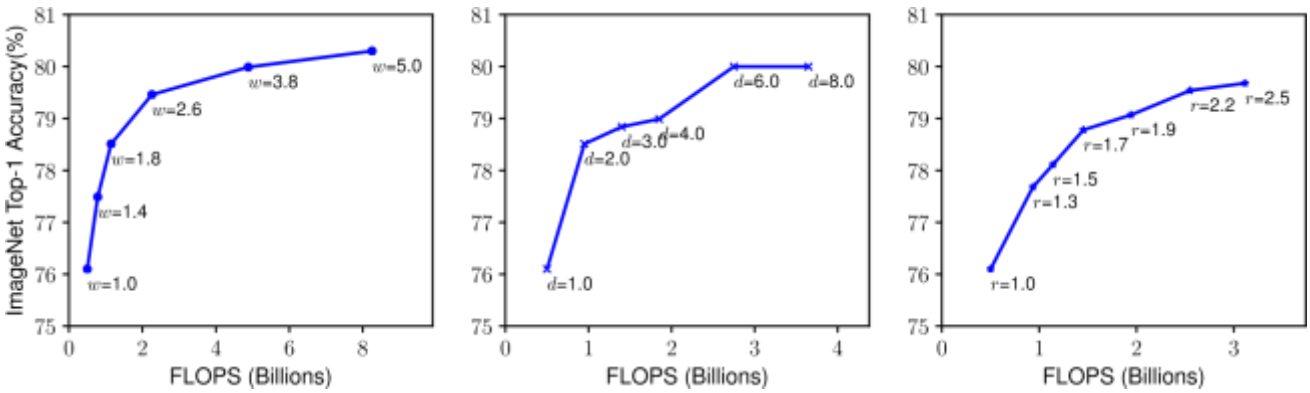


图 3.使用不同的网络宽度 (w)、深度 (d) 和分辨率 (r) 系数放大基准模型。宽度、深度或分辨率更大的网络往往能获得更高的准确度，但准确度在达到 80% 后很快就会饱和，这说明了单维度扩展的局限性。基线网络见表 1。

3.3. 复合缩放

我们根据经验观察到，不同的缩放维度并不是独立的。直观地说，对于更高分辨率的图像，我们应该增加网络深度，这样更大的感受野有助于捕捉更大图像中包含更多像素的类似特征。相应地，当分辨率较高时，我们也应该增加网络宽度，以便在高分辨率图像中捕捉到像素更多的精细模式。这些直觉表明，我们需要协调和平衡不同的缩放维度，而不是传统的单一维度缩放。

为了验证我们的直觉，我们比较了不同网络深度和分辨率下的宽度缩放，如图 4 所示。如果我们只缩放网络宽度 w 而不改变深度（d=1.0）和分辨率（r=1.0），精度很快就会达到饱和。如果深度（d=2.0）和分辨率（r=2.0）更高，在 FLOPS 成本相同的情况下，宽度扩展能获得更好的精度。这些结果使我们得出第二个结论：

观察 2 - 为了追求更高的精度和效率，在 ConvNet 扩展过程中平衡网络宽度、深度和分辨率的所有维度至关重要。

事实上，此前已有一些工作（Zoph 等人，2018 年；Real 等人，2019 年）尝试任意平衡网络宽度和深度，但都需要繁琐的手动调整。

在本文中，我们提出了一种新的复合缩放方法，它使用复合系数 ϕ 以一种原则性的方式均匀缩放网络宽度、深度和分辨率：

$$\text{depth: } d = \alpha^{\varphi}$$

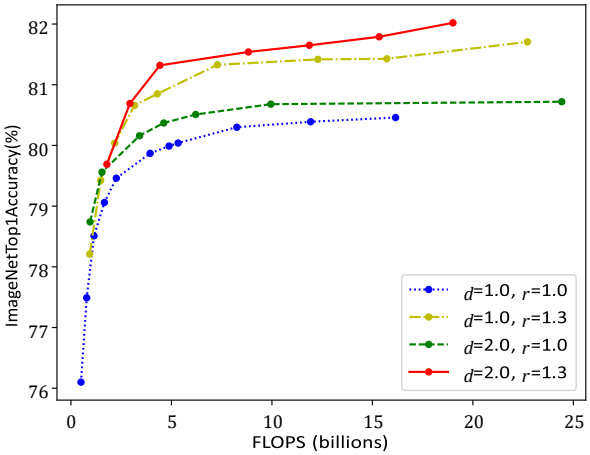


图 4.不同基线网络的网络宽度缩放。线条中的每个点表示不同宽度系数（w）的模型。所有基线网络均来自表 1。第一个基线网络（d=1.0，r=1.0）有 18 个卷积层，分辨率为 224x224，而最后一个基线网络（d=2.0，r=1.3）有 36 个卷积层，分辨率为 299x299。

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

其中 α、β、γ 为常数，可通过小网格搜索确定。直观地说，ϕ 是用户指定的系数，用于控制模型缩放所需的额外资源，而 α、β、γ 则分别指定如何将这些额外资源分配给网络宽度、深度和分辨率。值得注意的是，常规卷积运算的 FLOPS 与 d、w2、r2 成正比，即网络深度增加一倍，FLOPS 增加一倍，但

网络宽度或分辨率增加一倍，FLOPS 增加四倍。由于卷积运算通常在 ConvNet 的计算成本中占主导地位，因此根据公式 3 对 ConvNet 进行缩放，总 FLOPS 大约会增加 。在本文中，我们限制 α - β2 - γ2 ≈ 2，这样对于任何新的 ϕ，总 FLOPS 大约会增加 2ϕ。

4. 高效网络架构

由于模型缩放不会改变基线网络中的层算子 Fˆi，因此拥有一个良好的基线网络也至关重要。我们将使用现有的 ConvNets 评估我们的缩放方法，但为了更好地展示缩放方法的有效性，我们还开发了一个新的移动大小基线，称为 EfficientNet。

受（Tan 等人，2019 年）的启发，我们开发了基线网络，利用多目标神经架构搜索同时优化准确率和 FLOPS。具体来说，我们使用与（Tan 等，2019）相同的搜索空间，并使用 ACC(m)×[FLOPS(m)/T]w 作为优化目标，其中 ACC(m) 和 FLOPS(m) 表示模型 m 的准确度和 FLOPS，T 是目标 FLOPS，w=-0.07 是控制准确度和 FLOPS 之间权衡的超参数。与（Tan 等人，2019；Cai 等人，2019）不同的是，这里我们优化的是 FLOPS 而不是延迟，因为我们的目标不是任何特定的硬件设备。我们的搜索产生了一个高效网络，我们将其命名为 EfficientNet-B0。由于我们使用了与（Tan 等人，2019）相同的搜索空间，因此架构与 Mnas-Net 类似，只是由于 FLOPS 目标更大（我们的 FLOPS 目标是 400M），因此我们的 EfficientNet-B0 略大一些。表 1 显示了 EfficientNet-B0 的架构。它的主要构件是移动倒置瓶颈 MBConv（Sandler 等人，2018 年；Tan 等人，2019 年），我们还在其中添加了挤压-激发优化（Hu 等人，2018 年）。

从基线 EfficientNet-B0 开始，我们采用复合扩展方法，分两步将其扩展：

Stage i	Operator $\hat{F}_i$	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

表 1.EfficientNet-B0 基线网络 - 每行描述一个具有 Lˆi 层的阶段 i，输入分辨率为 hHˆi,Wˆii，输出通道为 Cˆi。符号取自公式 2。

- 步骤 1：我们首先固定 ϕ = 1，假设可用资源多一倍，并根据公式 2 和 3 对 α、β、γ 进行小网格搜索。
- 步骤 2：我们将 α、β、γ 设为常数，并使用公式 3 放大不同 ϕ 的基线网络，得到 EfficientNet-B1 至 B7（详见表 2）。

值得注意的是，在大型模型周围直接搜索 α、β、γ 可以获得更好的性能，但搜索成本在大型模型上会变得过于昂贵。我们的方法解决了这个问题，只需在小型基线网络上搜索一次（第 1 步），然后在所有其他模型上使用相同的缩放系数（第 2 步）。

5. 实验

在本节中，我们将首先在现有的 ConvNets 和新提出的 EfficientNets 上评估我们的缩放方法。

5.1. 扩大移动网和驻地网的规模

作为概念验证，我们首先将缩放方法应用于广泛使用的 MobileNets（Howard 等人，2017 年；Sandler 等人，2018 年）和 ResNet（He 等人，2016 年）。表 3 显示了用不同方法缩放它们的 ImageNet 结果。与其他单维度缩放方法相比，我们的复合缩放方法提高了所有这些模型的准确性，这表明我们提出的缩放方法对一般现有 ConvNets 非常有效。

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	-	-

我们省略了集合模型和多作物模型（Hu 等人，2018 年），或在 3.5B Instagram 图像上预先训练的模型（Mahajan 等人，2018 年）。

表 2.EfficientNet 在 ImageNet 上的性能结果（Russakovsky 等人，2015 年）。所有 EfficientNet 模型都是在基线 EfficientNet-B0 的基础上，使用等式 3 中不同的复合系数 ϕ 缩放而成。为了进行效率比较，我们将具有相似 top-1/top-5 准确率的 ConvNets 归为一组。与现有的 ConvNets 相比，我们的扩展 EfficientNet 模型始终能将参数和 FLOPS 减少一个数量级（参数减少高达 8.4 倍，FLOPS 减少高达 16 倍）。

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%**

表 3.扩大 MobileNets 和 ResNet 的规模

	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**

表 4.推理延迟比较 - 延迟是在英特尔至强 CPU E5-2690 单核心上以批量大小 1 测量的。

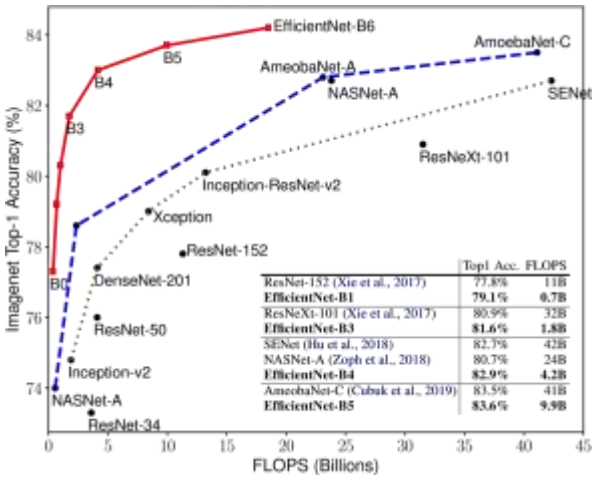


图 5.FLOPS 与 ImageNet 准确率对比 - 与图 1 类似，只是对比的是 FLOPS 而不是模型大小。

	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).

表 5.效能网络在迁移学习数据集上的性能结果。我们的扩展 EfficientNet 模型在 8 个数据集中的 5 个数据集上达到了最新的准确率，平均参数减少了 9.6 倍。

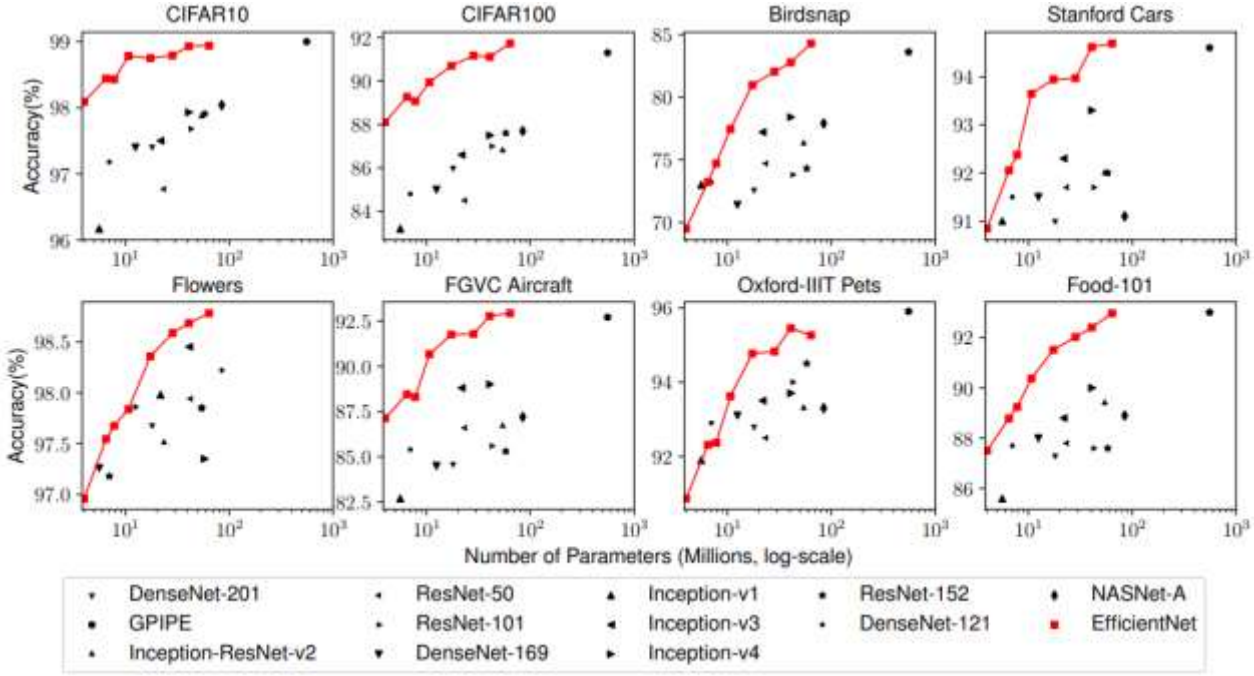


图 6.模型参数与迁移学习准确率对比 - 所有模型都在 ImageNet 上进行了预训练，并在新数据集上进行了微调。

5.2. ImageNet Results for EfficientNet

我们使用与（Tan 等人，2019 年）类似的设置在 ImageNet 上训练 EfficientNet 模型：RMSProp 优化器，衰减为 0.9，动量为 0.9；批量规范动量为 0.99；权重衰减 1e-5；初始学习率 0.256，每 2.4 个历时衰减 0.97。我们还使用了 SiLU (Swish-1) 激活（Ramachandran 等人，2018 年；Elfwing 等人，2018 年；Hendrycks & Gimpel，2016 年）、AutoAugment（Cubuk 等人，2019 年）和随机深度（Huang 等人，2016 年），生存概率为 0.8。众所周知，更大的模型需要更多的正则化，因此我们线性增加了 dropout（Srivastava 等人，2014 年）比率，从 EfficientNet-B0 的 0.2 增加到 B7 的 0.5。我们从训练集中随机抽取 25K 张图像作为最小集，并在此最小集上执行早期停止；然后在原始验证集上评估早期停止的检查点，以报告最终的验证准确率。

表 2 显示了从相同基线 EfficientNet-B0 扩展而来的所有 EfficientNet 模型的性能。与精度相似的其他 ConvNets 相比，我们的 EfficientNet 模型使用的参数和 FLOPS 通常要

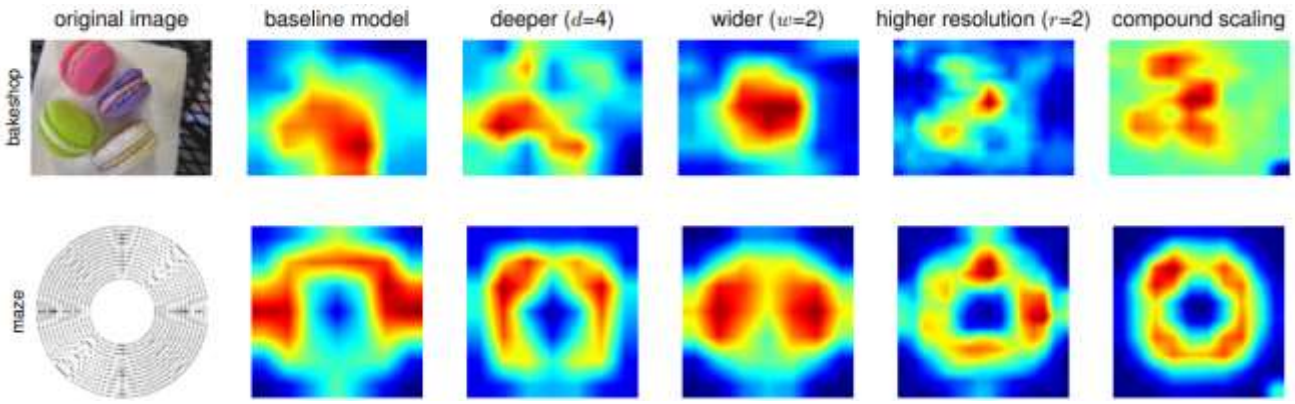


图 7.采用不同缩放方法的模型的类激活图（CAM）（Zhou 等人，2016 年）--我们的复合缩放方法可以使缩放模型（最后一列）聚焦于更多对象细节的相关区域。模型详情见表 7。

少一个数量级。特别是，我们的 EfficientNet-B7 在使用 6600 万个参数和 37B FLOPS 的情况下实现了 84.3% 的 top1 准确率，比之前的最佳 GPipe（Huang 等人，2018 年）更准确，但体积却小了 8.4 倍。这些进步既来自更好的架构、更好的扩展，也来自为 EfficientNet 定制的更好的训练设置。

图 1 和图 5 展示了具有代表性的 ConvNets 的参数-准确率和 FLOPS-准确率曲线，与其他 ConvNets 相比，我们的扩展 EfficientNet 模型以更少的参数和 FLOPS 达到更高的准确率。值得注意的是，我们的 EfficientNet 模型不仅体积小，而且计算成本更低。例如，与 ResNeXt101（Xie 等人，2017 年）相比，我们的 EfficientNet-B3 使用的 FLOPS 减少了 18 倍，却获得了更高的准确率。

为了验证延迟，我们还在实际 CPU 上测量了几个具有代表性的 CovNets 的推理延迟，如表 4 所示，我们报告了 20 次运行的平均延迟。我们的 EfficientNet-B1 比广泛使用的 ResNet-152 快 5.7 倍，而 EfficientNet-B7 比 GPipe（Huang 等人，2018 年）快约 6.1 倍，这表明我们的 EfficientNets 在实际硬件上确实很快。

5.3. EfficientNet 的迁移学习结果

我们还在一系列常用的迁移学习数据集上评估了我们的 EfficientNet，如表 6 所示。

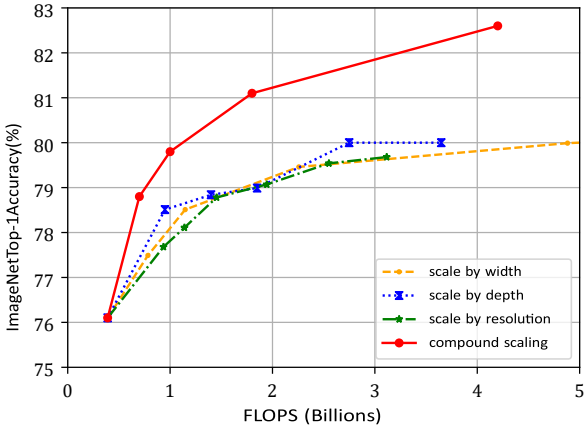


图 8.使用不同方法扩展 EfficientNet-B0。

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%**

表 7.图 7 中使用的缩放模型。

我们借鉴了（Kornblith 等人，2019 年）和（Huang 等人，2018 年）的相同训练设置，采用 ImageNet 预训练检查点，并在新数据集上进行微调。

表 5 显示了迁移学习的性能：（1）与 NASNet-A（Zoph 等人，2018 年）和 Inception-v4（Szegedy 等人，2017 年）等公开可用的模型相比，我们的 EfficientNet 模型实现了更好的准确性，参数平均减少了 4.7 倍（最多减少 21 倍）。(2) 与最先进的模型（包

括动态合成训练数据的 DAT（Ngiam 等人，2018 年）和使用专门管道并行性训练的 GPipe（Huang 等人，2018 年））相比，我们的 EfficientNet 模型在 8 个数据集中的 5 个数据集上的准确率仍然超过了它们，但使用的参数减少了 9.6 倍。

图 6 比较了各种模型的准确率-参数曲线。总体而言，我们的 EfficientNets 始终能以比现有模型少一个数量级的参数获得更高的准确率，现有模型包括 ResNet（He 等人，2016 年）、DenseNet（Huang 等人，2017 年）、Inception（Szegedy 等人，2017 年）和 NASNet（Zoph 等人，2018 年）。

6. 讨论

为了将我们提出的缩放方法与 EfficientNet 架构的贡献区分开来，图 8 比较了相同 EfficientNet-B0 基线网络下不同缩放方法的 ImageNet 性能。一般来说，所有缩放方法都能以更多 FLOPS 为代价提高准确性，但我们的复合缩放方法能比其他单维度缩放方法进一步提高准确性，最高可提高 2.5%，这表明了我们提出的复合缩放方法的重要性。

为了进一步理解为什么我们的复合缩放方法比其他方法更好，图 7 比较了一些具有代表性的模型在不同缩放方法下的类激活图（Zhou 等人，2016 年）。所有这些模型都是从相同的基线进行缩放的，它们的统计数据如表 7 所示。图像是从 ImageNet 验证集中随机挑选的。如图所示，采用复合缩放的模型倾向于聚焦于具有更多物体细节的相关区域，而其他模型要么缺乏物体细节，要么无法捕捉到图像中的所有物体。

7. 结论

在本文中，我们系统地研究了 ConvNet 的缩放，发现仔细平衡网络宽度、深度和分辨率是一个重要但缺失的环节，阻碍了我们获得更好的精度和效率。为了解决这个问题，我们提出了一种简单而高效的复合缩放方法，它能让我们以一种更有原则的方式轻松地将基线 ConvNet 扩展到任何目标资源限制，同时保持模型效率。在这种复合扩展方法的支持下，我们证明了移动大小的 EfficientNet 模型可以非常有效地扩展，在 ImageNet 和五个常用迁移学习数据集上，以数量级更少的参数和 FLOPS 超越了最先进的精度。

致谢

我们感谢庞若明、Vijay Vasudevan、Alok Aggarwal、Barret Zoph、于洪坤、Jonathon Shlens、Raphael Gontijo Lopes、吕一峰、彭大毅、宋晓丹、Samy Bengio、Jeff Dean 和谷歌大脑团队的帮助。

附录

自 2017 年以来，大多数研究论文只报告和比较 ImageNet 验证准确率；为了更好地进行比较，本文也遵循了这一惯例。此外，我们还通过向 http://image-net.org 提交我们对 100k 测试集图像的预测，验证了测试准确率；结果见表 8。不出所料，测试准确率与验证准确率非常接近。

	B0	B1	B2	B3	B4	B5	B6	B7
Val top1	77.11	79.13	80.07	81.59	82.89	83.60	83.95	84.26
Test top1	77.23	79.17	80.16	81.72	82.94	83.69	84.04	84.33
Val top5	93.35	94.47	94.90	95.67	96.37	96.71	96.76	96.97
Test top5	93.45	94.43	94.98	95.70	96.27	96.64	96.86	96.94

表 8.ImageNet 验证与测试 Top-1/5 的准确率对比。

参考文献

Berg, T., Liu, J., Woo Lee, S., Alexander, M. L., Jacobs, D. W., and Belhumeur, P. N. Birdsnap: Large-scale fine-grained visual categorization of birds. *CVPR*, pp. 2011–2018, 2014.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101– mining discriminative components with random forests. *ECCV*, pp. 446–461, 2014.

Cai, H., Zhu, L., and Han, S. Proxylessnas: Direct neural architecture search on target task and hardware. *ICLR*, 2019.

Chollet, F. Xception: Deep learning with depthwise separable convolutions. *CVPR*, pp. 1610–02357, 2017.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *CVPR*, 2019.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., and Keutzer, K. Squeezenext: Hardware-aware neural network design. *ECV Workshop at CVPR'18*, 2018.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CVPR*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. Mask´ r-cnn. *ICCV*, pp. 2980–2988, 2017.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. Amc: Automl for model compression and acceleration on mobile devices. *ECCV*, 2018.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. *CVPR*, 2018.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger,
K. Q. Deep networks with stochastic depth. *ECCV*, pp. 646–661, 2016.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. *CVPR*, 2017.

Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q. V., and Chen, Z. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *arXiv preprint arXiv:1808.07233*, 2018.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, pp. 448–456, 2015.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? *CVPR*, 2019.

Krause, J., Deng, J., Stark, M., and Fei-Fei, L. Collecting a large-scale dataset of fine-grained cars. *Second Workshop on Fine-Grained Visual Categorizatio*, 2013.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Lin, H. and Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. *NeurIPS*, pp. 6172– 6181, 2018.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B.,′ and Belongie, S. Feature pyramid networks for object detection. *CVPR*, 2017.

Liu, C., Zoph, B., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., and Murphy, K. Progressive neural architecture search. *ECCV*, 2018.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *NeurIPS*, 2018.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *ECCV*, 2018.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. Exploring the limits of weakly supervised pretraining. *arXiv preprint arXiv:1805.00932*, 2018.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, Q. V., and Pang, R. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. *ICVGIP*, pp. 722–729, 2008.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. *CVPR*, pp. 3498–3505, 2012.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and SohlDickstein, J. On the expressive power of deep neural networks. *ICML*, 2017.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2018.

Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. Regularized evolution for image classifier architecture search. *AAAI*, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. *CVPR*, 2018.

Sharir, O. and Shashua, A. On the expressive power of overlapping architectures of deep learning. *ICLR*, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,
A. Going deeper with convolutions. *CVPR*, pp. 1–9, 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *CVPR*, pp. 2818–2826, 2016.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 4:12, 2017.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V.

MnasNet: Platform-aware neural architecture search for mobile. *CVPR*, 2019.

Xie, S., Girshick, R., Dollar, P., Tu, Z., and He, K. Aggre-′ gated residual transformations for deep neural networks. *CVPR*, pp. 5987–5995, 2017.

Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sze, V., and Adam, H. Netadapt: Platform-aware neural network adaptation for mobile applications. *ECCV*, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *BMVC*, 2016.

Zhang, X., Li, Z., Loy, C. C., and Lin, D. Polynet: A pursuit of structural diversity in very deep networks. *CVPR*, pp. 3900–3908, 2017.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CVPR*, 2018.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba,
A. Learning deep features for discriminative localization. *CVPR*, pp. 2921–2929, 2016.

Zoph, B. and Le, Q. V. Neural architecture search with reinforcement learning. *ICLR*, 2017.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning transferable architectures for scalable image recognition. *CVPR*, 2018.