MobileNet: 用于视觉移动应用的高效卷积神经网络

Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko
Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

Google Inc.
{howarda,menglong,bochen,dkalenichenko,weijunw,weyand,anm,hadam}@google.com

摘要

我们为移动和嵌入式视觉应用提出了一类名为 MobileNets 的高效模型。MobileNets 基于精简的架构，使用深度可分离卷积来构建轻量级深度神经网络。我们引入了两个简单的全局超参数，可在延迟和准确性之间进行有效权衡。这些超参数允许模型构建者根据问题的限制条件为其应用选择合适大小的模型。我们对资源和准确性之间的权衡进行了广泛的实验，结果表明，与其他流行模型相比，MobileNets 在 ImageNet 分类中表现出强劲的性能。然后，我们展示了 MobileNets 在各种应用和用例中的有效性，包括物体检测、细粒度分类、人脸属性和大规模地理定位。

1. 引言

自从 AlexNet [19] 赢得 "ImageNet Challenge: ILSVRC 2012"[24]，普及了深度卷积神经网络之后，卷积神经网络在计算机视觉领域变得无处不在。为了达到更高的精确度，总的趋势是建立更深、更复杂的网络[27, 31, 29, 8]。然而，这些旨在提高精确度的改进并不一定能使网络在规模和速度上更加高效。在机器人、自动驾驶汽车和增强现实等许多现实应用中，需要在计算能力有限的平台上及时执行识别任务。

本文介绍了一种高效的网络架构和一组两个超参数，以便建立非常小的、低延迟的模型，从而轻松满足移动和嵌入式视觉应用的设计要求。第 2 节回顾了之前在建立小型模型方面的工作。第 3 节介绍了 MobileNet 架构以及两个超参数宽度乘数和分辨率乘数，以定义更小、更高效的 MobileNet。第 4 节介绍了在 ImageNet 上的实验以及各种不同的应用和用例。第 5 节最后是总结和结论.

2. 以往的工作

近年来，人们对构建小型高效神经网络的兴趣日益浓厚，例如 [16, 34, 12, 36, 22]。许多不同的方法一般可分为压缩预训练网络或直接训练小型网络。本文提出了一类网络架构，允许模型开发者根据其应用的资源限制（延迟、大小），有针对性地选择小型网络。MobileNet 主要侧重于优化延迟，但也会产生小型网络。许多关于小型网络的论文只关注规模而不考虑速度。

MobileNet 主要由深度可分离卷积构建而成，最初是在文献[26]中提出的，随后被用于 Inception 模型[13]，以减少前几层的计算量。扁平化网络[16]用全因式卷积构建网络，展示了极因式网络的潜力。与本文无关，Factorized Networks[34] 引入了类似的因式卷积以及拓扑连接的使用。随后，Xception 网络[3] 展示了如何扩展深度可分离滤波器，以超越 Inception V3 网络。另一个小型网络是 Squeezenet [12]，它使用瓶颈方法设计了一个非常小的网络。其他减少计算量的网络包括结构转换网络 [28] 和深度炸裂卷积网络 [37]。

获得小型网络的另一种方法是缩小、分解或压缩预训练网络。文献中提出了基于乘积量化[36]、散列[2]以及剪枝、矢量量化和哈夫曼编码[5]的压缩方法。此外，还有人提出了各种因子化方法来加快预训练网络的速度[14, 20]。另一种训练小型网络的方法是 "蒸馏法"[9]，它使用较大的网络来训练较小的

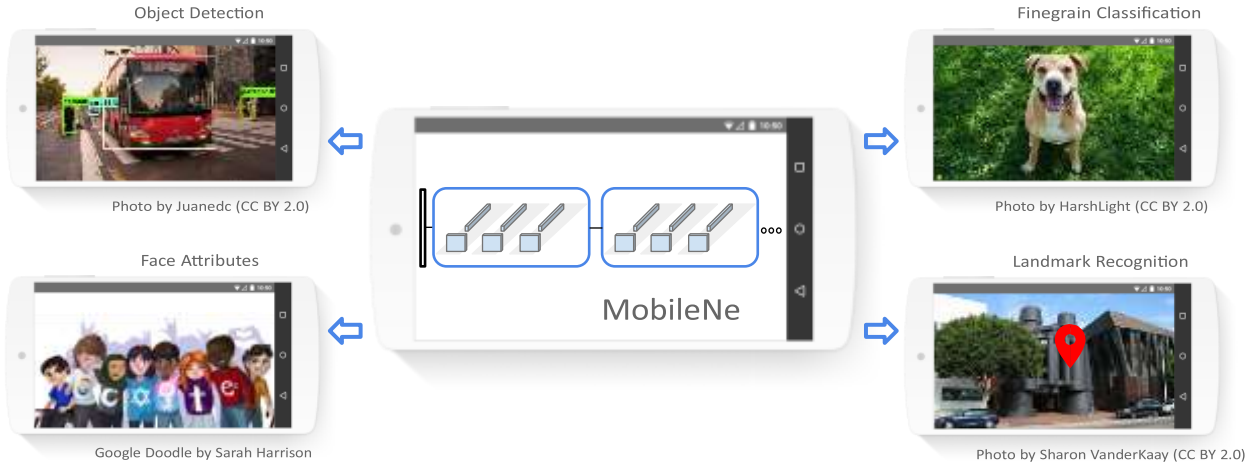


图 1.MobileNet 模型可应用于各种识别任务，实现高效的设备智能。

网络。这种方法与我们的方法相辅相成，在第 4 节的一些用例中也有所涉及。另一种新兴方法是低比特网络 [4, 22, 11]

3. MobileNet 架构

在本节中，我们首先介绍 MobileNet 的核心层，即深度可分离卷积。然后，我们将介绍 MobileNet 的网络结构，最后介绍两个模型缩小超参数：宽度乘数和分辨率乘数。

3.1. 深度可分离卷积

MobileNet 模型基于深度可分离卷积，这是一种因式卷积，它将标准卷积因式分解为深度卷积和 1×1 卷积（称为点式卷积）。对 MobileNets 而言，深度卷积对每个输入通道应用一个滤波器。然后，点式卷积应用 1×1 卷积来合并深度卷积的输出。标准卷积在一个步骤中既能过滤输入，又能将输入合并为一组新的输出。深度可分离卷积将其分为两层，一层用于过滤，一层用于合并。这种因式分解可以大大减少计算量和模型大小。图 2 显示了如何将标准卷积 2(a) 分解为深度卷积 2(b) 和 1 × 1 点式卷积 2(c)。

标准卷积层将 DF × DF × M 特征图 F 作为输入，并生成 DF × DF × N 特征图 G，其中 DF 是正方形输入特征图的空间宽度和高度，M 是输入通道数（输入深度），DG 是正方形输出特征图的空间宽度和高度，N 是输出通道数（输出深度）。

标准卷积层的参数是大小为 DK×DK×M×N 的卷积核 K，其中 DK 是假定为正方形的核的空间维度，M 是输入通道数，N 是输出通道数，如前所述。

标准卷积的输出特征图假定步长为 1，填充计算公式为：

$$\mathbf{G}_{k,l,n} = \sum_{i,j,m} \mathbf{K}_{i,j,m,n} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \qquad (1)$$

标准卷积的计算成本为：

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \qquad (2)$$

其中，计算成本与输入通道数 M、输出通道数 N、内核大小 Dk × Dk 和特征图大小 DF × DF 成倍数关系。MobileNet 模型解决了上述各项及其相互作用的问题。首先，它使用深度可分离卷积来打破输出通道数量与内核大小之间的相互影响。

标准卷积操作的作用是根据卷积核过滤特征，并将特征组合起来，以产生新的表示。通过使用称为深度可分离卷积的因子化卷积，可以将过滤和组合步骤分成两个步骤，从而大大降低计算成本。

深度可分离卷积由两层组成：深度卷积和点深度卷积。我们使用深度卷积为每个输

[1] 我们假设输出特征图与输入特征图具有相同的空间维度，并且两个特征图都是正方形。我们的模型缩小结果适用于任意尺寸和长宽比的特征图。

入通道（输入深度）应用一个滤波器。点向卷积是一种简单的 1×1 卷积，用于创建深度卷积层输出的线性组合。MobileNets 的两个层都使用批量规范和 ReLU 非线性。

每个输入通道（输入深度）使用一个滤波器的深度卷积可写成：

$$\hat{\mathbf{G}}_{k,l,m} = \sum_{i,j} \hat{\mathbf{K}}_{i,j,m} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \qquad (3)$$

其中，K^ 是大小为 DK × DK × M 的深度卷积核，K^ 中的第 m 个滤波器应用于 F 中的第 m 个通道，以产生滤波输出特征图 G^ 的第 m 个通道。

深度卷积的计算成本为：

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \qquad (4)$$

与标准卷积相比，深度卷积的效率极高。但是，它只能过滤输入通道，并不能将它们组合起来生成新的特征。因此，为了生成这些新特征，需要增加一个层，通过 1 × 1 卷积计算深度卷积输出的线性组合。

深度卷积和 1 × 1（点卷积）卷积的组合被称为深度可分离卷积，最初是在 [26] 中提出的。深度可分离卷积的成本是:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \qquad (5)$$

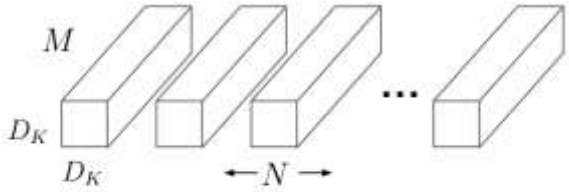
这是深度卷积和 1 × 1 点阵卷积之和。

将卷积表示为滤波和组合的两步过程，可以减少计算量：

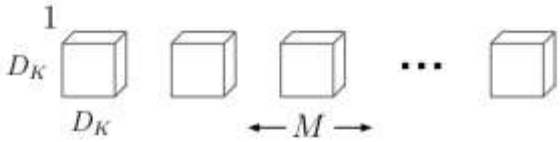
$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

MobileNet 使用 3 × 3 深度可分离卷积，与标准卷积相比，计算量减少了 8 到 9 倍，但准确率却略有下降，这一点在第 4 节中可以看到。

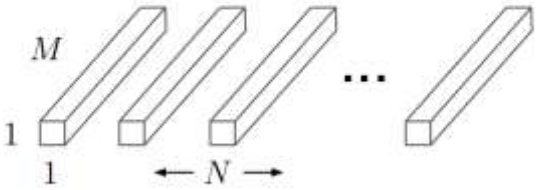
在空间维度上的额外因式分解（如 [16, 31] 中的因式分解）并不能节省多少额外的计算量，因为深度卷积所耗费的计算量非常少。



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1 × 1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

图 2.图（a）中的标准卷积滤波器被两层滤波器取代：图（b）中的深度卷积滤波器和图（c）中的点卷积滤波器，从而建立了一个深度可分离滤波器

上图中的标注：

a) 标准卷积滤波器

b) 深度卷积滤波器

c) 1×1 卷积滤波器在深度可分离卷积中被称为点式卷积

3.2. 网络结构和培训

正如上一节所述，MobileNet 结构基于深度可分离卷积，只有第一层是完全卷积。通过用如此简单的术语定义网络，我们可以轻松探索网络拓扑结构，找到一个好的网络。MobileNet 的架构如表 1 所示。除了最后的全连接层没有非线性，并进入 softmax 层进行分类外，所有层后都有 batchnorm [13] 和 ReLU 非线性。图 3 将具有常规卷积、批处理规范和 ReLU 非线性的层与具有深度卷积、1 × 1 点卷积以及每个卷积层之后的批处理规范和 ReLU 的因子层进行了对比。在深度卷积和第一层中，采用分步卷积处理向下采样。在全连接层之前，最后的平均池化将空间分辨率降为 1。将深度卷积和点卷积作为独立层计算，MobileNet 共有 28 层。

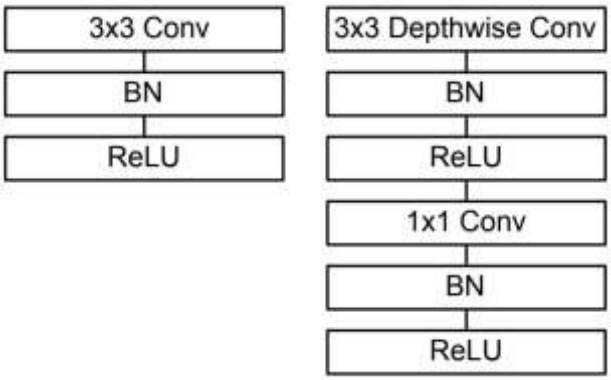


图 3.左图：带有批规范和 ReLU 的标准卷积层。右图深度可分离卷积层与深度和点深度卷积层，然后是批规范和 ReLU。

仅仅用少量的 Mult-Adds 来定义网络是不够的。同样重要的是，要确保这些运算可以高效地实现。例如，非结构化稀疏矩阵运算通常不会比密集矩阵运算快，除非稀疏程度非常高。我们的模型结构将几乎所有计算都置于密集的 1×1 卷积中。这可以通过高度优化的通用矩阵乘法（GEMM）函数来实现。卷积通常由 GEMM 实现，但需要在内存中进行名为 im2col 的初始重新排序，以便将其映射到 GEMM。例如，流行的 Caffe 软件包 [15] 就采用了这种方法。1×1 卷积不需要在内存中重新排序，可以直接用 GEMM 实现，而 GEMM 是最优化的数值线性代数算法之一。从表 2 中可以看出，MobileNet 95% 的计算时间都花在了 1×1 卷积上，而 1×1 卷积也有 75% 的参数。几乎所有附加参数都在全连接层。

MobileNet 模型在 TensorFlow [1] 中使用 RMSprop [33] 和异步梯度下降技术进行训练，与 Inception V3 [31] 相似。不过，与训练大型模型相反，我们使用了较少的正则化和数据增强技术，因为小型模型的过拟合问题较少。在训练 MobileNets 时，我们不使用侧头或标签平滑，此外，我们还通过限制用于大型 Inception 训练的小作物的大小来减少失真图像的数量[31]。此外，我们还发现，由于深度滤波器的参数非常少，因此必须对深度滤波器进行很少或不进行权重衰减（l2 正则化）。对于下一节中的 ImageNet 基准，无论模型大小如何，所有模型都使用相同的训练参数进行训练。

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

表 1.MobileNet 主体架构

Type	Mult-Adds	Parameters
Conv 1 × 1	94.86%	74.59%
Conv DW 3 × 3	3.06%	1.06%
Conv 3 × 3	1.19%	0.02%
Fully Connected	0.18%	24.33%

表 2.每层资源类型

3.3. 宽度倍增器：较薄模型

虽然 MobileNet 的基本架构已经很小，延迟也很低，但很多时候，特定用例或应用可能要求模型更小、更快。为了构建更小、计算成本更低的模型，我们引入了一个非常简单的参数 α，称为宽度乘数。宽度乘数 α 的作用是在每一层均匀地减薄网络。对于给定的层和宽度乘数 α，输入通道数 M 变为 αM，输出通道数 N 变为 αN。

宽度乘数为 α 的深度可分离卷积的计算成本为：

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F \qquad (6)$$

其中 α∈ (0,1]，典型设置为 1、0.75、0.5 和 0.25。α = 1 为基准移动网络，α < 1 为缩小移动网络。宽度乘法器的作用是以大约 α2 的四倍降低计算成本和参数数量。宽度乘法器可应用于任何模型结构，以定义一个新的较

Layer/Modification	Million Mult-Adds	Million Parameters
Convolution	462	2.36
Depthwise Separable Conv	52.3	0.27
$\alpha = 0.75$	29.6	0.15
$\rho = 0.714$	15.1	0.15

表 3.修改标准卷积的资源使用情况。请注意，每一行都是在前一行基础上的累积效果。本例为内部 MobileNet 层，DK = 3，M = 512，N = 512，DF = 14。

小模型，并在精度、延迟和大小方面进行合理权衡。它可用于定义需要从头开始训练的新缩小结构。

3.4. 分辨率乘法器：简化表示法

降低神经网络计算成本的第二个超参数是分辨率乘数 ρ。我们将其应用于输入图像，随后每一层的内部表示都会以相同的乘数进行缩减。在实际应用中，我们通过设置输入分辨率来隐式地设置 ρ。

现在，我们可以将网络核心层的计算成本表示为宽度乘数 α 和分辨率乘数 ρ 的深度可分离卷积：

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \qquad (7)$$

其中，ρ∈（0,1]通常是隐式设置，以便网络的输入分辨率为 224、192、160 或 128。ρ = 1 是基准移动网络，ρ < 1 是计算量减少的移动网络。分辨率乘数的作用是将计算成本降低 ρ^2。

以 MobileNet 中的一个典型层为例，我们可以看到深度可分离卷积、宽度乘法器和分辨率乘法器是如何降低成本和参数的。表 3 显示了当架构缩减方法依次应用于一个层时，该层的计算量和参数数量。第一行显示的是输入特征图大小为 14×14×512 且内核 K 大小为 3 × 3 × 512 × 512 的全卷积层的多重乘法和参数。我们将在下一节详细讨论资源和精度之间的权衡。

4. 实验

在本节中，我们首先研究了深度卷积的效果，以及通过减少网络宽度而不是层数来缩小网络的选择。然后，我们展示了基于两个超参数（宽度乘数和分辨率乘数）缩减网络的权衡结果，并将结果与一些流行模型进行了比较。

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

表 4.深度可分与全卷积 MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

表 5.窄移动网络与浅移动网络

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

表 6.移动网络宽度乘数

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

表 7.移动网络分辨率

然后，我们研究了应用于多种不同应用的移动网络。

4.1. 模型选择

首先，我们展示了使用深度可分离卷积的 MobileNet 与使用完全卷积建立的模型的对比结果。从表 4 中我们可以看出，与完全卷积相比，使用深度可分离卷积在 ImageNet 上仅降低了 1%的准确率，而在多重添加和参数方面却节省了大量成本。

接下来，我们将展示使用宽度乘数的较薄模型与使用较少层的较浅模型的比较结果。为了使 MobileNet 更浅，我们删除了表 1 中特征大小为 14 × 14 × 512 的 5 层可分离滤波器。表 5 显示，在计算量和参数数量相似的情况下，使 MobileNet 变薄的效果比变浅的效果好 3%。

4.2. 模型收缩超参数

表 6 显示了用宽度乘数 α 缩减 MobileNet 架构时的精度、计算量和大小权衡。精度平稳下降，直到 α = 0.25 时架构变得太小。

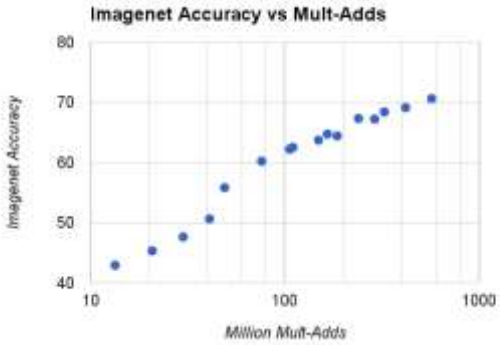


图 4 该图显示了在 ImageNet 基准上计算量（Mult-Adds）与准确率之间的权衡。请注意准确率和计算量之间的对数线性关系。

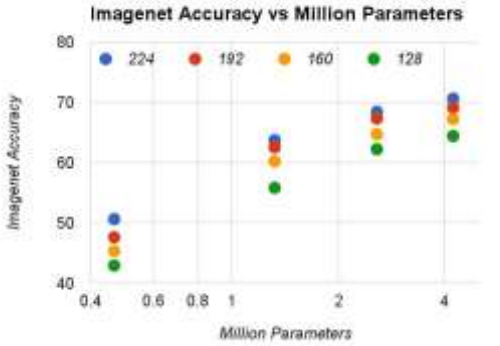


图 5.该图显示了在 ImageNet 基准测试中参数数量与准确率之间的权衡。颜色表示输入分辨率。参数数量不随输入分辨率而变化。

表 7 显示了通过降低输入分辨率来训练 MobileNets 时，不同分辨率乘数在精度、计算量和大小方面的折衷情况。精度在不同分辨率下平稳下降。

图 4 显示了由宽度乘数 $\alpha \in \{1,0.75,0.5,0.25\}$ 和分辨率 {224,192,160,128} 的交叉乘积组成的 16 个模型的 ImageNet 精度和计算量之间的权衡。结果是对数线性的，当模型在 $\alpha = 0.25$ 时变得非常小时，会出现跳跃。

图 5 显示了由宽度乘数 $\alpha \in \{1,0.75,0.5,0.25\}$ 和分辨率 {224,192,160,128} 的交叉乘积组成的 16 个模型的 ImageNet 准确度与参数数量之间的权衡。

表 8 将完整的 MobileNet 与原始 GoogleNet [30] 和 VGG16 [27] 进行了比较。MobileNet 的准确度几乎与 VGG16 相当，但体积小 32 倍，计算密集度低 27 倍。它比 GoogleNet 更准确，同时体积更小，计算量减少 2.5 倍以上。

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

表 8.MobileNet 与流行模型的比较

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.50 MobileNet-160	60.2%	76	1.32
Squeezenet	57.5%	1700	1.25
AlexNet	57.2%	720	60

表 9.小型 MobileNet 与流行型号的比较

Model	Top-1 Accuracy	Million Mult-Adds	Million Parameters
Inception V3 [18]	84%	5000	23.2
1.0 MobileNet-224	83.3%	569	3.3
0.75 MobileNet-224	81.9%	325	1.9
1.0 MobileNet-192	81.9%	418	3.3
0.75 MobileNet-192	80.5%	239	1.9

表 10.斯坦福狗（Stanford Dogs）的 MobileNet

Scale	Im2GPS [7]	PlaNet [35]	PlaNet MobileNet
Continent (2500 km)	51.9%	77.6%	79.3%
Country (750 km)	35.4%	64.0%	60.3%
Region (200 km)	32.1%	51.1%	45.2%
City (25 km)	21.9%	31.7%	31.7%
Street (1 km)	2.5%	11.0%	11.4%

表 11.使用 MobileNet 架构的 PlaNet 性能。百分比是 Im2GPS 测试数据集中与地面实况距离在一定范围内的定位分数。原始 PlaNet 模型的数据基于经过改进的架构和训练数据集的更新版本。

表 9 比较了宽度乘数为 $\alpha = 0.5$、分辨率为 160 × 160 的缩小版 MobileNet。缩小后的 MobileNet 比 AlexNet [19] 好 4%，但体积比 AlexNet 小 45 倍，计算量比 AlexNet 少 9.4 倍。在大小和计算量减少 22 倍的情况下，它也比 Squeezenet [12] 好 4%。

4.3. 精细识别

我们在斯坦福狗数据集 [17] 上训练 MobileNet 进行精细识别。我们扩展了 [18] 的方法，从网络上收集了比 [18] 更大但噪声更高的训练集。我们使用嘈杂的网络数据预训练细粒度狗识别模型，然后在 Stanford Dogs 训练集上对模型进行微调。Stanford Dogs 测试集的结果见表 10。MobileNet 几乎可以达到 [18] 的最先进结果，而且计算量和体积大大减少。

4.4. 大规模地理定位

PlaNet [35] 将确定照片拍摄地点的任务作为一个分类问题。该方法将地球划分为一个个地理单元网格，作为目标类别，并在数百万张有地理标记的照片上训练卷积神经网络。事实证明，PlaNet 能成功定位大量照片，其性能优于处理相同任务的 Im2GPS [6, 7]。

我们使用 MobileNet 架构在相同数据上重新训练 PlaNet。基于 Inception V3 架构[31]的完整 PlaNet 模型有 5200 万个参数和 57.4 亿次多重添加。而 MobileNet 模型只有 1300 万个参数，其中 300 万个通常用于主体层，1000 万个用于最终层，以及 58 万个多重附加值。如表 11 所示如表 11 所示，与 PlaNet 相比，MobileNet 版本尽管更加紧凑，但性能却略有下降。此外，它的性能仍然远远超过 Im2GPS。

4.5. 面部（face）属性

MobileNet 的另一个用例是压缩具有未知或深奥训练程序的大型系统。在人脸属性分类任务中，我们展示了 MobileNet 与深度网络知识转移技术 "蒸馏"（distillation）[9] 之间的协同关系。我们试图减少一个包含 7500 万个参数和 1600 万个 Mult-Adds 的大型人脸属性分类器。该分类器是在类似于 YFCC100M [32] 的多属性数据集上训练的。

我们利用 MobileNet 架构蒸馏出一个人脸属性分类器。蒸馏法[9]的工作原理是通过训练分类器来模拟更大模型的输出，而不是地面实况标签，因此可以从大型（可能是无限的）无标签数据集进行训练。结合蒸馏训练的可扩展性和 MobileNet 的简易参数化，最终系统不仅不需要正则化（如权重衰减和早期停止），而且性能更强。从表从表 12 中可以看出，基于 MobileNet 的分类器对激进的模型缩减有很强的适应能力：它能达到与内部分类器相似的跨属性平均精度（平均 AP），但只消耗了 1%的多重添加。

Width Multiplier / Resolution	Mean AP	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	88.7%	568	3.2
0.5 MobileNet-224	88.1%	149	0.8
0.25 MobileNet-224	87.2%	45	0.2
1.0 MobileNet-128	88.1%	185	3.2
0.5 MobileNet-128	87.7%	48	0.8
0.25 MobileNet-128	86.4%	15	0.2
Baseline	86.9%	1600	7.5

表 12.使用 MobileNet 架构进行的人脸属性分类。每一行对应不同的超参数设置（宽度乘数 α 和图像分辨率）。

Framework Resolution	Model	mAP	Billion Mult-Adds	Million Parameters
SSD 300	deeplab-VGG	21.1%	34.9	33.1
	Inception V2	22.0%	3.8	13.7
	MobileNet	19.3%	1.2	6.8
Faster-RCNN 300	VGG	22.9%	64.3	138.5
	Inception V2	15.4%	118.2	13.3
	MobileNet	16.4%	25.2	6.1
Faster-RCNN 600	VGG	25.7%	149.6	138.5
	Inception V2	21.9%	129.6	13.3
	Mobilenet	19.8%	30.5	6.1

4.6. 物体检测

图 6.使用 MobileNet SSD 的物体检测结果示例。

MobileNet 还可作为有效的基础网络部署到现代物体检测系统中。我们报告了基于最近赢得 2016 COCO 挑战赛[10]的工作，在 COCO 数据上训练 MobileNet 进行物体检测的结果。在表 13 中，MobileNet 在 Faster-RCNN [23] 和 SSD [21] 框架下与 VGG 和 Inception V2 [13] 进行了比较。在我们的实验中，SSD 以 300 输入分辨率（SSD 300）进行评估，Faster-RCNN 则以 300 和 600 输入分辨率（FasterRCNN 300、Faster-RCNN 600）进行比较。Faster-RCNN 模型对每幅图像评估 300 个 RPN 提议框。模型在 COCO train+val 上进

Model	1e-4 Accuracy	Million Mult-Adds	Million Parameters
FaceNet [25]	83%	1600	7.5
1.0 MobileNet-160	79.4%	286	4.9
1.0 MobileNet-128	78.3%	185	5.5
0.75 MobileNet-128	75.2%	166	3.4
0.75 MobileNet-128	72.5%	108	3.8

表 14.从 FaceNet 提炼出的 MobileNet

行训练，不包括 8k minival 图像，并在 minival 上进行评估。对于这两个框架，MobileNet 的计算复杂度和模型大小仅为其他网络的一小部分，却取得了与其他网络相当的结果。

4.7. 面部嵌入

FaceNet 模型是最先进的人脸识别模型 [25]。它基于三重损失建立人脸嵌入。为了建立移动 FaceNet 模型，我们使用蒸馏法，通过最小化 FaceNet 和 MobileNet 在训练数据上的输出平方差来进行训练。非常小的 MobileNet 模型的结果见表 14。

5. 结论

我们提出了一种基于深度可分离卷积的新模型架构，称为 MobileNets。我们研究了导致高效模型的一些重要设计决策。然后，我们演示了如何利用宽度乘法器和分辨率乘法器构建更小更快的 MobileNets，并通过合理的精度权衡来减少尺寸和延迟。然后，我们将不同的 MobileNet 与流行的模型进行了比较，这些模型在尺寸、速度和准确性方面都具有优势。最后，我们展示了 MobileNet 在各种任务中的应用效果。下一步，我们计划在 Tensor Flow 中发布模型，以帮助 MobileNets 的应用和探索。

参考文献

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*, 1,, 2015. 4

[2] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. *CoRR, abs/1504.04788*, 2015. 2

[3] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357v2*, 2016. 1

[4] M. Courbariaux, J.-P. David, and Y. Bengio. Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*, 2014. 2

[5] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, 2, 2015.2

[6] J. Hays and A. Efros. IM2GPS: estimating geographic information from a single image. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008. 7

[7] J. Hays and A. Efros. Large-Scale Image Geolocalization. In J. Choi and G. Friedland, editors, *Multimodal Location Estimation of Videos and Images*. Springer, 2014. 6, 7

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[9] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 7

[10] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016. 7

[11] Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *arXiv preprint arXiv:1609.07061*, 2016. 2

[12] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 1mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 1, 6

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1, 3, 7

[14] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014. 2

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[16] J. Jin, A. Dundar, and E. Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014. 1, 3

[17] A.Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision*

and Pattern Recognition, Colorado Springs, CO, June 2011. 6

[18] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. *arXiv preprint arXiv:1511.06789*, 2015. 6

[19] A.Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 6

[20] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*, 2014. 2

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 7

[22] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. *arXiv preprint arXiv:1603.05279*, 2016. 1, 2

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7

[24] O.Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 8

[26] L. Sifre. *Rigid-motion scattering for image classification*. PhD thesis, Ph. D. thesis, 2014. 1, 3

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 6

[28] V. Sindhwani, T. Sainath, and S. Kumar. Structured transforms for small-footprint deep learning. In *Advances in Neural Information Processing Systems*, pages 3088–3096, 2015. 1

[29] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 1

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 6

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 1, 3, 4, 7

[32] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 7

[33] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012. 4

[34] ,M. Wang, B. Liu, and H. Foroosh. Factorized convolutional neural networks. *arXiv preprint arXiv:1608.04337*, 2016. 1

[35] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *European Conference on Computer Vision (ECCV)*, 2016. 6, 7

[36] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. *arXiv preprint arXiv:1512.06473*, 2015. 1

[37] Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, and Z. Wang. Deep fried convnets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1483, 2015. 1