计算机视觉课程报告

沈鹏
11821018
zju
shenp94@163.com

摘要

我本身不是做AI方向的，本报告若有表述不当或错误之处，还请老师批评指正。

自从AlexNet[1]通过赢得ImageNet挑战：ILSVRC 2012[2]推广深度卷积神经网络依赖，卷积神经网络已经在计算机视觉中无处不在。总的趋势是制造更深入，更复杂的网络，以实现更高的准确性。然而，这些提高精度的进步不一定使网络在尺寸和速度方面更有效。在许多现实世界的应用中，例如机器人、自动驾驶汽车和增强现实，需要在计算有限的平台上及时地执行识别任务。本文描述了一个高效的网络架构和一组两个超参数，以构建非常小，低延迟的模型，可以轻松地匹配移动和嵌入式视觉应用的设计要求。

本文提出了一类称为MobileNets的高效模型，用于移动和嵌入式视觉应用。MobileNets基于流线型架构，使用深度可分离卷积来构建轻量级神经网络。本文介绍了两个简单的全局超参数，它们在延迟和准确性之间进行了有效的权衡。这些超参数允许模型构建器根据问题的约束为其应用选择合适大小的模型。本文提供了有关资源和准确性权衡的广泛实验，并且与ImageNet分类上的其他流行模型相比显示出强大的性能。然后，本文展示了MobileNets在各种应用和用例中的有效性，包括对象检测，细粒分类，面部属性和大规模地理定位。

1. 介绍

本节中，本文首先描述构建MobileNets的核心层，它们是深度可分离的过滤器。然后，本文描述了MobileNets网络结构，并总结了两个模型收缩超参数宽度乘数和分辨率乘数的描述。

1.1. 深度可分离卷积

MobileNets模型基于深度可分离卷积，这是一种因式化卷积形式，它将标准卷积分解为深度卷积和称为逐点卷积的1x1卷积。对于MobileNets，深度卷积将单个滤波器应用于每个输入通道。然后，逐点卷积应用1x1卷积来组合输出深度卷积。标准卷积既可以过滤，也可以在一个步骤中将输入组合成一组新的输出。深度可分离卷积将其分成两层，一个用于过滤的单独层和一个用于组合的单独层。这种因子分解具有显著减少计算和模型大小的效果。如图1所示。

标准卷积层将$D_F \times D_F \times M$特征映射F作为输入，并生成$D_F \times D_F \times N$特征映射G，其中D_F是方形输入特征映射的空间宽度和高度，M是输入通道的数量（输入深度），D_G是方形输出特征图的空间宽度和高度，N是输出通道的数量（输出深度）。

标准卷积层由大小为$D_K \times D_K \times M \times N$的卷积核K参数化，其中$D_K$是假设为正方形的核的空间维度，M是输入通道的数量，N是先前定义的输出通道的数量。

标准卷积的输出特征映射（假设步长为1和填充）计算如下：

$$G_{k,l,m} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m}$$

标准卷积的计算成本为：

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

其中计算成本乘以输入通道M的数量，输出通道的数量N，内核大小$D_K \times D_K$的特征映射大小$D_F \times D_F$。MobileNets模型解决了这些术语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和内核大小之间的相互作用。

标准卷积操作具有基于卷积内核和组合特征来过滤特征以产生新表示的效果。通过使用称为深度可分离卷积的分解卷积，可以将滤波和组合步骤分成两个步骤，从而显著降低计算成本。

深度可分离卷积由两层组成：深度卷积和逐点卷积。本文使用深度卷积来为每个输入通道（输入深度）应用单个滤波器。然后使用简单的1x1卷积的逐点卷积来创建深度层的输出的线性组合。MobileNets对两个层都使用batchnorm和ReLU非线性。

每个输入通道（输入深度）使用一个滤波器进行深度卷积可写为：

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m}$$

其中$\hat{K}$是大小为$D_K \times D_K \times M$的深度卷积核，其中$\hat{K}$中的第m_{th}个滤波器应用于F中的第m_{th}个通道，以产生滤

波器输出特征映射$\hat{G}$的第m_{th}个通道。

深度卷积的计算成本为：

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F$$

相对于标准卷积，深度卷积非常有效。但是它只会过滤输入通道，它不会讲它们组合起来创建新功能。因此，为了生成这些新特性，需要通过 1x1 卷积计算深度卷积输出的线性组合的附加层。

深度卷积和 1x1（逐点）卷积的组合成为深度可分离卷积。

深度可分离卷积成本：

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

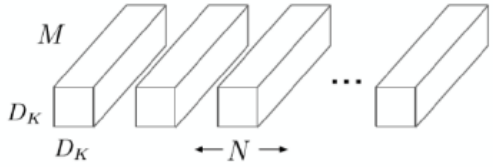
这是深度和 1x1 逐点卷积的总和。

通过将卷积表示为过滤和组合的两步过程，本文可以减少计算：

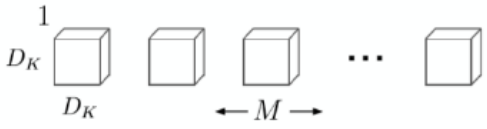
$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

MobileNets使用 3x3 深度可分离卷积，其使用的计算量比标准卷积少 8 到 9 倍，而精度只有很小的降低。

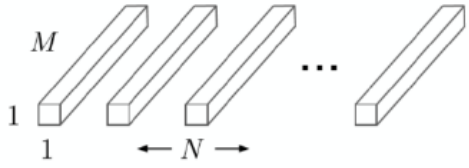
空间维度中的附加因子分解中没有节省太多额外的计算，因为在深度卷积中花费的计算非常少。



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



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

Figure 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

1.2. 网络结构和训练

MobileNets结构建立在深度可分离的卷积上，如前一节所述，除了第一层是完全卷积。通过以如此简单的方式定义网络，本文能够轻松探索网络拓扑以找到一个好的网络。MobileNets架构在表 1 中定义。所有层后面都是一个batchnorm[3]和ReLU非线性，除了最终完全连接的层没有非线性，并输入softmax层进行分类。图 3 将具有规则卷积，batchnorm和ReLU的分解层形成对比。在深度方向卷积中以及在第一层中使用跨步卷积来处理下采样。最终平均池化层在完全连接层之前将空间分辨率降低到 1。将深度和逐点卷积记为单独的层，MobileNets有 28 层。

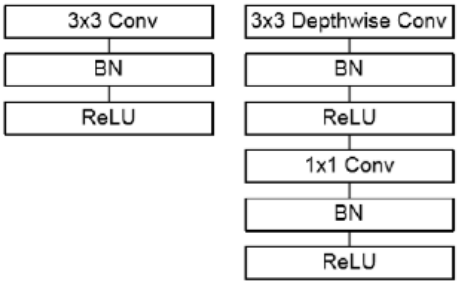


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

仅仅根据少量的Mult-Adds来定义网络是不够的。确保这些操作可以有效实施也很重要。例如，非结构化稀疏矩阵运算通常不比秘籍矩阵运算快，直到非常高的稀疏度。本文的模型结构几乎将所有计算都放入秘籍的 1x1 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，通常，卷积由GEMM实现，但需要在内存中进行初始重新排序，称为im2col，以便将其映射到GEMM。1x1 卷积不需要在存储器中重新排序，并且可以直接用GEMM实现，GEMM是最优化的数值线性代数算法之一。MobileNets在 1x1 卷积中花费 95%的计算时间，其中还有 75%的参数，如表 2 所示。几乎所有附加参数都在完全连接的层中。

MobileNets模型咋TensorFlow[4]中使用RMSprop进行训练，其异步梯度下降类似于InceptionV3。然而，与训练大型模型相反，本文使用较少的正则化和数据增强技术，因为小模型在过度配置方面的麻烦较少。在训练MobileNets时，本文不使用侧头或标签平滑，并通过限制大型初始训练中使用的小作物的大小来减少扭曲图像的数量。另外，本文发现在深度滤波器上放置很少或没有重量衰减（L2 正则化）很重要，因为它们的参数很少，对于下一部分中的ImageNet基准测试，无论模型的大小如何，所有模型都使用相同的训练参数进行训练。

Table 1. MobileNet Body Architecture

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

Table 2. Resource Per Layer Type

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%

1.3. 宽度乘数：更小的模型

尽管基础MobileNets架构已经很小且延迟很低，但很多时候特定的用例或应用程序可能要求模型更小更快。为了构造这些更小且计算量更小的模型，本文引入了一个非常简单的参数α，称为宽度乘数。宽度乘数α的作用实在每层均匀地稀疏网络。对于给定的层和宽度乘数α，输入通道M的数量变为αM，输出通道的数量N变为αN。

具有宽度乘数α的深度可分为卷积的计算成本是：

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F$$

其中$\alpha \in (0,1]$，典型设置为 1,0.75,0.5 和 0.25。$\alpha = 1$是基线MobileNets，$\alpha < 1$是减少的MobileNets。宽度乘数具有降低计算成本和参数数量的效果α^2。宽度乘数可以应用于任何模型结构，以定义一个新的较小模型，具有合理的精度、延迟和大小权衡。它用于定义需要从头开始训练的新的简化结构。

1.4. 分辨率乘数：减少的表示

减少神经网络的计算成本的第二个超参数是解乘数ρ。本文将其应用于输入图像，并且每个层内部表示随后被相同的乘数减少。在实践中，本文通过设置输入分辨率隐式设置ρ。

本文现在可以将网络核心层的计算成本表示为具有宽度乘数α和分辨率乘数ρ的深度可分离卷积：

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$$

其中$\rho \in (0,1]$通常是隐式设置的，因此网络的输入分辨率为 224,192,160 或 128。$\rho = 1$是基线MobileNets，$\rho < 1$是减少计算的MobileNets。分辨率乘数通过ρ^2降低计算成本的效果。

作为一个例子，本文可以看一下MobileNets中的典型层，看看深度可分离卷积，宽度乘数和分辨率乘数如何降低成本和参数。表 3 显示了层的参数的计算和数量，因为架构收缩方法被顺序地应用于层。第一行显示完整卷积层的Mult-Adds和参数，输入特征映射大小为14x14x512，内核K大小为 3x3x512x512。

Table 3. Resource usage for modifications to standard convolution. Note that each row is a cumulative effect adding on top of the previous row. This example is for an internal MobileNet layer with $D_K = 3, M = 512, N = 512, D_F = 14$.

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

2. 实验结果

在本节中，本文首先通过减小网络宽度而不是层数来研究深度卷积的影响以及收缩的选择。然后，本文展示了基于两个超参数减少网络的权衡宽度乘数和分辨率乘数，并将结果与许多流行模型进行比较。然后，我们调查应用于许多不同应用程序的MobileNets。

2.1. 模型选择

首先，本文展示了具有深度可分离卷积的MobileNets的结果，与使用完整卷积构建的模型相比。在表 4 中，本文看到使用深度可分离卷积与完全卷积相比，ImageNet只能将精度降低 1%，从而大大节省了多次加法和参数。

接下来，本文展示了使用较小层将较薄模型与宽度模型与较浅模型进行比较的结果。为了使MobileNets更浅，删除表 1 中具有 14x14x512 特征尺寸的 5 层可分离滤波器。表 5 显示，在相似的计算和参数数量下，使MobileNets更薄的效果比使它们更浅更好 3%。

Table 4. Depthwise Separable vs Full Convolution MobileNet

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

Table 5. Narrow vs Shallow MobileNet

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

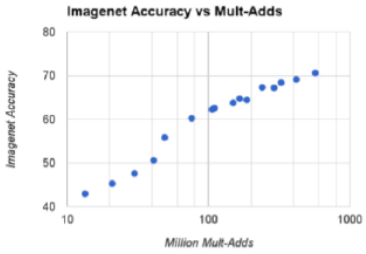


Figure 4. This figure shows the trade off between computation (Mult-Adds) and accuracy on the ImageNet benchmark. Note the log linear dependence between accuracy and computation.

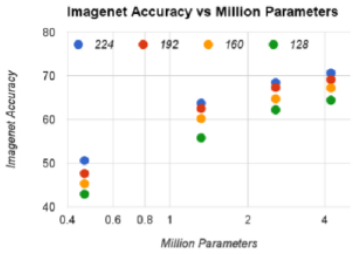


Figure 5. This figure shows the trade off between the number of parameters and accuracy on the ImageNet benchmark. The colors encode input resolutions. The number of parameters do not vary based on the input resolution.

2.2. 模型收缩超参数

表 6 显示了使用宽度乘数α缩小MobileNets架构的精度，计算和大小权衡。精度下降平滑，直到在$\alpha = 0.25$时结构太小。

表 7 显示了通过训练具有降低的输入分辨率的MobileNets，不同分辨率乘法器的精度，计算和大小权衡。精度在分辨率下平滑下降。

Table 6. MobileNet Width Multiplier

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

Table 7. MobileNet Resolution

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

图 4 显示了ImageNet精度与 16 个模型的计算之间的权衡，这些模型由宽度乘数$\alpha \in 1,0.75,0.5,0.25$和分辨率224,192,160,128 的叉积制成。当模型在$\alpha = 0.25$时非常小时，结果是对数线性的跳跃。

图 5 显示了ImageNet准确度与 16 个模型参数之间的折中，这些模型由宽度乘数$\alpha \in 1,0.75,0.5,0.25$和分辨率224,192,160,128 的叉积制成。

表 8 将完整的MobileNets与原始GoogleNet[5]和VGG16[6]进行了比较。MobileNets几乎与VGG16 一样准确，同时缩小了 32 倍，计算密集度降低了 27 倍。它比GoogleNet更精准，同时体积更小，计算量减少 2.5 倍以上。

表 9 比较了降低的MobileNets与宽度乘数$\alpha = 0.5$和降低分辨率 160x160。MobileNets比AlexNet降低 4%，同时比AlexNet小 45 倍，计算小 9.4 倍。它也比Squeezenet[7]好 4%，大小相同，计算量减少 22 倍

Table 8. MobileNet Comparison to Popular Models

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

Table 9. Smaller MobileNet Comparison to Popular Models

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

2.3. 细粒度识别

本文训练MobileNet在Stanford Dogs数据集上进行细粒度识别。本文使用嘈杂的网络数据预先训练细粒度的狗识别模型，然后在斯坦福犬训练集上进行模型调整。Standford Dogs测试集的结果见表 10。MobileNet几乎

可以在计算和尺寸大大减少的情况下达到最先进的结果。

Table 10. MobileNet for Stanford Dogs

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

2.4. 大规模地理定位

PlaNet投入的任务是确定照片被拍摄的地方作为分类问题。该方法将地球划分为一个地理单元格网格，作为目标类别，并在数百万张带地理标记的照片中训练卷积神经网络。PlaNet已被证明可以成功的定位各种各样的照片，并且胜过Im2GPS，可以解决相同的任务。

本文使用MobileNet架构在相同数据上重新训练PlaNet。虽然基于InceptionV3 架构的完整PlaNet模型拥有 5200 万个参数和 57.4 亿个多重增加。MobileNet模型只有 1300 万个参数，通常 300 万个主体，1000 万个最终层和 0.58 百万个多重增加。如表格 11 所示。与PlaNet相比，MobileNet版本的性能略有下降，尽管更紧凑。而且，它仍大大优于Im2GPS。

Table 11. Performance of PlaNet using the MobileNet architecture. Percentages are the fraction of the Im2GPS test dataset that were localized within a certain distance from the ground truth. The numbers for the original PlaNet model are based on an updated version that has an improved architecture and training dataset.

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%

2.5. 人脸分布

MobileNet的另一个用例是压缩具有未知或深奥培训程序的大型系统。在面部属性分类任务中，本文展示了MobileNet和蒸馏之间的协同关系，这是一种深度网络的知识转移技术。本文寻求减少一个具有 7500 万个参数和 1600 万个Mult-Adds的大型面部属性分类器。分类器在类似于YFCC100M的多属性数据集上进行训练。

本文使用MobileNet架构提取面部属性分类器。蒸馏通过训练分类器来模拟较大模型 2 的输出而不是地面实况标签，从而实现大型（可能是有限的）未标记数据集的训练。结合蒸馏训练的可扩展性和MobileNet的简约参数化，终端系统不仅不需要正规化（例如，重量衰减和早期停止），而且还表现出增强的性能。从表 12 中可以看出这一点。基于MobileNet的分类器对于积极的模型收缩具有弹性：它实现了与内部属性（平均AP）相似的平均精度，同时仅消耗了多次添加的 1%。

Table 12. Face attribute classification using the MobileNet architecture. Each row corresponds to a different hyper-parameter setting (width multiplier α and image resolution).

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

2.6. 目标检测

MobileNet还可以作为现代物体检测系统中的有效基础网络进行部署。本文根据最近赢得 2016 年COCO年挑战的工作报告针对COCO数据进行物体检测培训的MobileNet的结果。在表 13 中，MobileNet与Faster-RCNN和SSD框架下的VGG和Inception V2 进行比较。在本文的实验中，SSD[8]使用 300 输入分辨率（SSD 300）进行评估，并将Faster-RCNN模型评估每个图像 300 个RPN提议框。这些，模型在COCO train + val上训练，不包括 8k的迷你图像，并在迷你上进行评估。对于这两个框架，MobileNets实现了与其他网络相当的结果，只有一小部分计算复杂性和模型大小。

Table 13. COCO object detection results comparison using different frameworks and network architectures. mAP is reported with COCO primary challenge metric (AP at IoU=0.50:0.05:0.95)

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

2.7. 人脸嵌入

FaceNet模型是最先进的人脸识别模型[10]。它基于三元组损失构建面嵌入。为了构建移动FaceNet模型，本文使用蒸馏来训练FaceNet和MobileNets输出在训练数据上的平方差异。非常小的MobileNets模型的结果可以在表 14 中找到。

Table 14. MobileNet Distilled from FaceNet

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

3. 小结与讨论

本文提出了一种基于深度可分离卷积的称为MobileNets的新模型架构。本文研究了一些导致高效模型的重要设计决策，然后，本文演示了如何使用宽度乘数和分辨率乘数构建更小，更快的MobileNets，通过折中合理的精度来减小大小和延迟。然后，本文将不同的MobileNets与流行的模型进行了比较，展示出了出众的尺寸，速度和准确性。最后本文通过展示MobileNets在应用于各种任务时的有效性得出结论。

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in International Conference on Neural Information Processing Systems, 2012.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[4] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[6] G. Song, B. Leng, L. Yu, C. Hetang, and S. Cai, "Region-based quality estimation network for large-scale person re-identification," 2017.

[7] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size," 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European Conference on Computer Vision, 2016.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," 2015.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2015.