**MobileNets: Efﬁcient Convolutional Neural Networks for Mobile Vision Applications**

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

**Abstract:** We present a class of efﬁcient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. We introduce two simple global hyper-parameters that efﬁciently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classiﬁcation. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, ﬁnegrain classiﬁcation, face attributes and large scale geo-localization.

**摘要：**我们提出了一类称为MobileNets的高效模型,它能用于移动和嵌入式的视觉应用。 **MobileNets是使用深度可分离卷积来构建轻量级深度神经网络的流线型架构。** 我们引进了两个简单的全局超参数，它们在延迟**【这儿的延迟应该泛指效率】**和准确性之间进行了有效的平衡。 **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【通过超参数的调节可以改变模型的大小】。** 我们展示了大量**资源和准确性平衡**的实验结果，相比ImageNet分类上的其他流行模型来说性能更强大。 然后，我们展示了MobileNets在各种应用和实例中的有效性，包括目标检测，细粒度分类，人脸属性和大规模地理定位。**【根据一张图片，确定出其拍摄的地理位置？有点意思。】**

1. **Introduction**

Convolutional neural networks have become ubiquitous in computer vision ever since AlexNet [19] popularized deep convolutional neural networks by winning the ImageNet Challenge: ILSVRC 2012 [24]. The general trend has been to make deeper and more complicated networks in order to achieve higher accuracy [27, 31, 29, 8]. However, these advances to improve accuracy are not necessarily making networks more efficient with respect to size and speed. In many real world applications such as robotics, self-driving car and augmented reality, the recognition tasks need to be carried out in a timely fashion on a computationally limited platform.

AlexNet [19]赢得了ImageNet挑战赛ILSVRC 2012 [24]，极大地推广了深度卷积神经网络的发展，自此以后，卷积神经网络已经在计算机视觉中无处不在。 **CNN的总体设计趋势是更深层，更复杂的网络，以实现更高的准确率[27,31,29,8]。** 然而，这些准确度提高的进步并不一定使网络在尺寸和速度方面更有效**【更深更复杂的网络往往需要的算力更大，时间复杂度更高】。** 在诸如机器人、自动驾驶汽车和增强现实等许多现实应用中需要在计算有限的平台上及时地执行识别任务**【嵌入式开发板的算力有限，往往在服务器上训练所得到的模型需要压缩和量化】。**

This paper describes an efficient network architecture and a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications. Section 2 reviews prior work in building small models. Section3 describes the MobileNet architecture and two hyper-parameters width multiplier and resolution multiplier to define smaller and more efficient MobileNets. Section 4 describes experiments on ImageNet as well a variety of different applications and use cases. Section 5 closes with a summary and conclusion.

**本文描述了一种高效的网络架构，使用两个超参数，以便构建极度精简，低延迟的模型，从而可以轻松地满足移动和嵌入式视觉应用的设计要求。** 第2节简要回顾了一下建立小型模型历史研究工作。 第3节描述了MobileNet的架构和两个超参数：宽度因子和分辨率因子，以定义更小，更高效的MobileNets。 第4节描述了ImageNet上的实验以及各种不同的应用程序和用例。 第5节以概述和结论结束。

1. **Prior Work**

There has been rising interest in building small and efficient neural networks in the recent literature, e.g. [16, 34, 12, 36, 22]. Many different approaches can be generally categorized into either compressing pretrained networks or training small networks directly. This paper proposes a class of network architectures that allows a model developer to specifically choose a small network that matches the resource restrictions (latency, size) for their application. MobileNets primarily focus on optimizing for latency but also yield small networks. Many papers on small networks focus only on size but do not consider speed.

在最近的文献中，研究者对建立精简而有效的神经网络越来越感兴趣,如文献 [16,34,12,36,22]。 这些不同的方法通常可以分为压缩预训练好的网络或直接训练小网络。 **本文提出了一类网络体系结构，允许模型开发人员专门选择与其应用程序的资源限制（延迟，大小）匹配的小型网络。 MobileNets主要专注于优化延迟，但同时也会产生精简的网络。**许多关于轻量化网络的论文只关注模型大小但不考虑速度。

MobileNets are built primarily from depth wise separable convolutions initially introduced in [26] and subsequently used in Inception models [13] to reduce the computation in the first few layers. Flattened networks [16] build a network out of fully factorized convolutions and showed the potential of extremely factorized networks. Independent of this current paper, Factorized Networks [34] introduces a similar factorized convolution as well as the use of topological connections. Subsequently, the Xception network [3] demonstrated how to scale up depthwise separable filters to out perform Inception V3 networks. Another small network is Squeeze net [12] which uses a bottleneck approach to design a very small network. Other reduced computation networks include structured transform networks [28] and deep fried convnets [37].

**MobileNets主要由深度可分离卷积构建，其最初在[26]中引入**，随后用于Inception模型[13]以减少前几层中的计算量。 扁平化网络[16]利用完全分解的卷积构建网络，展示了其极度分解网络的潜力。与当前论文的思路不同，因式分解网络[34]引入了类似因式化的卷积以及使用拓扑连接。 随后，Xception网络[3]演示了如何扩展深度可分离卷积核以实现Inception V3网络。 另一个小型网络是Squeeze net [12]，它使用瓶颈方法设计一个非常小的网络。 其他减少计算的网络包括结构化变换网络[28]和深度fried网络[37]**【这些个网络轻量化的论文，从事这方面的你还是要看一看的】。**

A different approach for obtaining small networks is shrinking, factorizing or compressing pretrained networks. Compression based on product quantization [36], hashing [2], and pruning, vector quantization and Huffman coding [5] have been proposed in the literature. Additionally various factorizations have been proposed to speed up pretrained networks [14, 20]. Another method for training small networks is distillation [9] which uses a larger network to teach a smaller network. It is complementary to our approach and is covered in some of our use cases in section 4. Another emerging approach is low bit networks [4, 22, 11].

获得小型网络的另一种方法是缩减**【剪枝。。。】**，分解或压缩预训练的网络。 有相关文献已经提出了基于乘积量化[36]，散列[2]和剪枝，矢量量化和霍夫曼编码[5]**【这几个典型的大爷了解一下呢】**的压缩。 另外，已经提出了各种因式分解的技术来加速预训练网络[14,20]。 训练小型网络的另一种方法是蒸馏[9]，它使用更大的网络来教授更小的网络。 它是我们方法的补充，在第4节的一些用例中有所介绍。另一种新兴方法是低位网络[4,22,11]**【目前深鉴科技所采用的方法基本上是基于粗粒度的剪枝和模型参数的量化，可以参考韩松的相关论文研究】。**

1. **MobileNet Architecture**

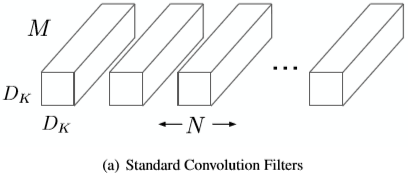
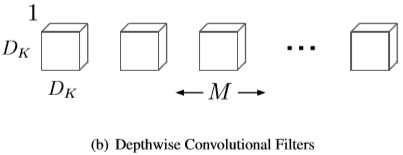
In this section we first describe the core layers that MobileNet is built on which are depthwise separable filters. We then describe the MobileNet network structure and conclude with descriptions of the two model shrinking hyperparameters width multiplier and resolution multiplier.

**在本节中，我们首先描述构建MobileNet的核心层，它们是深度可分离的卷积核。 然后，我们描述了MobileNet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. **Depthwise Separable Convolution**

The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. Figure 2 shows how a standard convolution 2(a) is factorized into a depthwise convolution 2(b) and a 1×1 pointwise convolution 2(c).

**MobileNet模型基于深度可分离卷积，这是一种因式分解的卷积形式，它将标准卷积分解为深度卷积和称为逐点卷积的1×1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1×1卷积来组合输出的深度卷积**【通道信息融合】**。 标准卷积可以在一个步骤中完成滤波并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，一个用于滤波的单独层和一个用于组合的单独层。【可以去看看深度可分离卷积的具体文章】** 这种因式分解的方法具有显着减少计算量和模型大小的效果。 图2展示了如何将标准卷积2（a）分解为深度卷积2（b）和1×1逐点卷积2（c）。

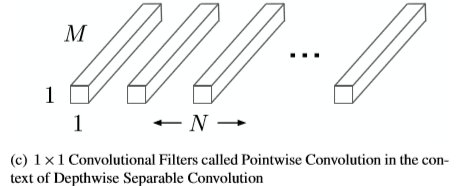


Figure 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

A standard convolutional layer takes as input a DF ×DF × M feature map F and produces a DF × DF × N feature map G where DF is the spatial width and height of a square input feature map (We assume that the output feature map has the same spatial dimensions as the input and both feature maps are square. Our model shrinking results generalize to feature maps with arbitrary sizes and aspect ratios.), M is the number of input channels (input depth), DG is the spatial width and height of a square output feature map and N is the number of output channel (output depth).

标准卷积层以DF×DF×M维度的特征图F作为输入并生成DF×DF×N的特征图G，其中DF是方形输入特征图的空间宽度和高度（我们假设输出特征图具有与输入特征图相同的空间维度且两个特征图都是正方形。我们的模型收缩结果可以推广到具有任意大小和纵横比的特征图。），M是输入通道的数量（输入深度），DG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）**【结合后文，此处的DG的尺寸应该为DK×DK×N】**。

The standard convolutional layer is parameterized by convolution kernel K of size DK×DK×M×N where DK is the spatial dimension of the kernel assumed to be square and M is number of input channels and N is the number of output channels as defined previously.

标准卷积层由大小为DK×DK×M×N的卷积核K参数化表示，其中DK是假定为正方形的核的空间维度，M是输入通道的数量，N是预先定义的输出通道的数量。

The output feature map for standard convolution assuming stride one and padding is computed as:



Standard convolutions have the computational cost of:



where the computational cost depends multiplicatively on the number of input channels M, the number of output channels N the kernel size Dk × Dk and the feature map size DF × DF. MobileNet models address each of these terms and their interactions. First it uses depthwise separable convolutions to break the interaction between the number of output channels and the size of the kernel.

标准卷积的输出特征图（假设步长为1且有padding）计算如下：



标准卷积的计算成本为**【浮点运算次数】**：



其中计算成本取决于输入通道的数量M、输出通道的数量N、卷积核大小Dk×Dk和特征映图大小DF×DF的乘积。 MobileNet模型解决了这些术语及其相互作用的问题【意思大概是MobileNet完成了标准卷积该完成的工作】。 首先，它使用深度可分离卷积来打破输出通道数量和卷积核大小之间的依赖关系**【意思是使卷积核大小不一定非要和输出通道数有关联】。**

The standard convolution operation has the effect of filtering features based on the convolutional kernels and combining features in order to produce a new representation. The filtering and combination steps can be split into two steps via the use of factorized convolutions called depthwise separable convolutions for substantial reduction in computational cost.

标准卷积操作能使用卷积核过滤特征并组合特征以产生新表示**【特征】**的作用。 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，从而显着降低计算成本。

Depthwise separable convolution are made up of two layers: depthwise convolutions and pointwise convolutions. We use depthwise convolutions to apply a single filter per each input channel (input depth). Pointwise convolution, a simple 1×1 convolution, is then used to create a linear combination of the output of the depthwise layer. MobileNets use both batchnorm and ReLU nonlinearities for both layers.

**深度可分离卷积由两层组成：深度卷积和逐点卷积。 我们使用深度卷积对每个输入通道（输入深度）应用单个滤波器。 然后使用简单的1×1逐点卷积来创建深度卷积层输出的线性组合。 MobileNets对上述两个层都使用batchnorm和ReLU非线性激活。**

Depthwise convolution with one filter per input channel (input depth) can be written as:



Where is the depthwise convolutional kernel of size DK × DK × M where the mth filter in is applied to the mth channel in F to produce the mth channel of the filtered output feature map .

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：



其中是大小为DK×DK×M的深度卷积核，其中中的第m个滤波器被应用于F中的第m个通道以产生滤波后的输出特征图的第m个通道。

Depthwise convolution has a computational cost of:



深度卷积的计算成本为：



Depthwise convolution is extremely efficient relative to standard convolution. However it only filters input channels, it does not combine them to create new features. So an additional layer that computes a linear combination of the output of depthwise convolution via 1×1 convolution is needed in order to generate these new features.

相对于标准卷积，深度卷积效率极高。 **但是它只对输入通道进行了过滤，不会将它们组合起来产生新特征。 因此，为了生成这些新特征，需要附加1×1的卷积层来计算深度卷积层输出特征的线性组合。【将标准卷积分两步走，第一步卷积实现滤波功能，第二步跨通道融合】**

The combination of depthwise convolution and 1 × 1 (pointwise) convolution is called depthwise separable convolution which was originally introduced in [26].

深度卷积和1×1（逐点）卷积的组合称为深度可分离卷积，最初在[26]中引入。

Depthwise separable convolutions cost:



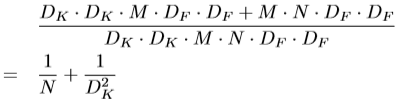
Which is the sum of the depthwise and 1×1 pointwise convolutions.

深度可分离卷积的计算成本为：

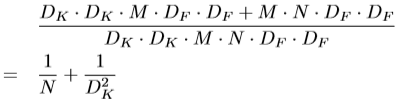


这是深度卷积和1×1逐点卷积的成本总和。

By expressing convolution as a two step process of filtering and combining we get a reduction in computation of:



通过将卷积表示为过滤和组合的两步过程，我们得到的计算量占原有计算量的比例为：



MobileNet uses 3×3 depthwise separable convolutions which uses between 8 to 9 times less computation than standard convolutions at only a small reduction in accuracy as seen in Section 4.

MobileNet使用3×3的深度可分离卷积，**其使计算量比标准卷积少8到9倍**，而精度只有很小的损失，如第4节所示。

Additional factorization in spatial dimension such as in [16,31] does not save much additional computation as very little computation is spent in depthwise convolutions.

如[16,31]所用的空间维度中的附加因子分解，由于其在深度卷积中花费的计算量很少，因此无法节省太多额外的计算量。

* 1. **Network Structure and Training**

The MobileNet structure is built on depthwise separable convolutions as mentioned in the previous section except for the first layer which is a full convolution. By defining the network in such simple terms we are able to easily explore network topologies to find a good network. The MobileNet architecture is defined in Table1. All layers are followed by a batchnorm[13] and ReLU nonlinearity with the exception of the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification. Figure 3 contrasts a layer with regular convolutions, batchnorm and ReLU nonlinearity to the factorized layer with depthwise convolution, 1×1 pointwise convolution as well as batchnorm and ReLU after each convolutional layer. Down sampling is handled with strided convolution in the depthwise convolutions as well as in the first layer. A final average pooling reduces the spatial resolution to 1 before the fully connected layer. Counting depthwise and pointwise convolutions as separate layers, MobileNet has 28 layers.

如前一节所述，MobileNet结构建立在深度可分离卷积上，除了第一层是完全卷积。以如此简单的术语，完成了网络的定义，我们能够轻松地探索网络拓扑以找到良好的网络。 MobileNet的架构在表1中定义。最终的全连接层除外，所有层后面都跟一个batchnorm [13]和ReLU非线性激活，全连接层是输入softmax层进行分类。图3将具有标准卷积，batchnorm和ReLU非线性激活的层与具有深度卷积，1×1逐点卷积以及每个卷积层之后接Batchnorm和ReLU的分解层进行对比。包括第一层在内，我们在深度方向使用跨步卷积处理下采样。最后在全连接层之前平均池化将空间分辨率降低到1。将深度卷积和逐点卷积计为单独的层，MobileNet总共有28层**【这段是MobileNet网络结构的详细叙述】**。

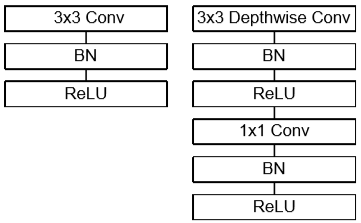
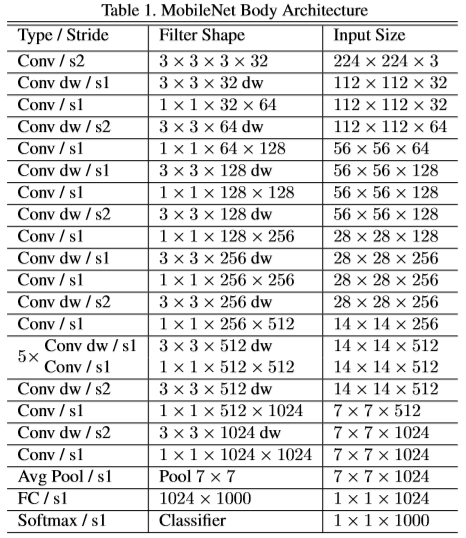
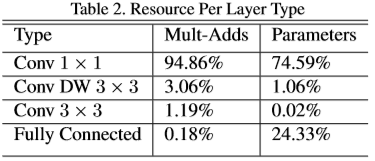


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.



It is not enough to simply define networks in terms of a small number of Mult-Adds. It is also important to make sure these operations can be efficiently implementable. For instance unstructured sparse matrix operations are not typically faster than dense matrix operations until a very high level of sparsity. Our model structure puts nearly all of the computation into dense 1×1 convolutions. This can be implemented with highly optimized general matrix multiply (GEMM) functions. Often convolutions are implemented by a GEMM but require an initial reordering in memory called im2col in order to map it to a GEMM. For instance, this approach is used in the popular Caffe package [15]. 1×1 convolutions do not require this reordering in memory and can be implemented directly with GEMM which is one of the most optimized numerical linear algebra algorithms. MobileNet spends 95% of it’s computation time in 1 × 1 convolutions which also has 75% of the parameters as can be seen in Table 2. Nearly all of the additional parameters are in the fully connected layer.

仅通过统计Mult-Adds运算的次数来定义网络的简单与否是不够的**【老哥们觉得这样翻译正确吗？】，**确保这些操作在实际实施中有效也很重要。例如，非结构化稀疏矩阵运算通常不比密集矩阵运算快，除非稀疏度非常高。我们的模型结构几乎将所有计算都放入密集的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。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。例如，将这种方法用于流行的Caffe包[15]。 1×1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。 MobileNet在1×1卷积中花费95％的计算时间，其中还有75％的参数，如表2所示。几乎所有其它参数都在全连接的层中。



MobileNet models were trained in TensorFlow [1] using RMSprop[33] with asynchronous gradient descent similar to Inception V3 [31]. However, contrary to training large models we use less regularization and data augmentation techniques because small models have less trouble with over fitting. When training MobileNets we do not use side heads or label smoothing and additionally reduce the amount image of distortions by limiting the size of small crops that are used in large Inception training [31]. Additionally, we found that it was important to put very little or no weight decay (l2 regularization) on the depthwise filters since there are so few parameters in them. For the ImageNet benchmarks in the next section all models were trained with same training parameters regardless of the size of the model.

MobileNet模型在TensorFlow [1]中使用RMSprop [33]进行训练，其中异步梯度下降类似于Inception V3 [31]。 然而，与训练大型模型相反，我们很少使用正则化和数据增强技术，因为小型模型在过拟合方面的麻烦较少。 在训练MobileNets时，我们不使用侧头**【这个是个啥？】**或标签平滑，也不像训练Inception那样通过限制小的剪切块的大小来减少扭曲图像的数量[31]。 另外一点很重要，我们发现在深度滤波器上只需放置很少或没有权重衰减（l2正则化），因为它们中的参数很少。 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。

* 1. **Width Multiplier: Thinner Models**

Although the base MobileNet architecture is already small and low latency, many times a specific use case or application may require the model to be smaller and faster. In order to construct these smaller and less computationally expensive models we introduce a very simple parameter α called width multiplier. The role of the width multiplier α is to thin a network uniformly at each layer. For a given layer and width multiplier α, the number of input channels M becomes αM and the number of output channels N becomes αN.

虽然MobileNet基础架构已经很小且延迟很低，但是很多时候特定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，我们引入了一个非常简单的参数α，称为宽度系数。 **宽度系数α的作用是在每层均匀地稀疏网络。** **对于给定的层和宽度系数α，输入通道M的数量变为αM，输出通道的数量N变为αN。**

The computational cost of a depthwise separable convolution with width multiplier α is:



where α ∈ (0,1] with typical settings of 1, 0.75, 0.5 and 0.25. α = 1 is the baseline MobileNet and α < 1 are reduced MobileNets. Width multiplier has the effect of reducing computational cost and the number of parameters quadratically by roughly α2. Width multiplier can be applied to any model structure to define a new smaller model with a reasonable accuracy, latency and size trade off. It is used to define a new reduced structure that needs to be trained from scratch.

具有宽度系数α的深度可分离卷积的计算成本是：



其中α∈（0,1]，典型值为1,0.75,0.5和0.25.α= 1是基础的MobileNet，α<1的时候构建更简化的MobileNets。**宽度乘数具有降低计算成本和参数数量的效果，参数量大致减少为1/α2，**宽度乘数可以应用于任何模型结构，以定义新的精度、延迟和大小权衡的小模型。α用于定义需要从头开始训练的新的简化结构。

* 1. **Resolution Multiplier: Reduced Representation**

The second hyper-parameter to reduce the computational cost of a neural network is a resolution multiplier ρ. We apply this to the input image and the internal representation of every layer is subsequently reduced by the same multiplier. In practice we implicitly set ρ by setting the input resolution.

降低神经网络计算成本的第二个超参数是分辨率系数ρ。 我们将其应用于输入图像，并且每个层的内部表示随后减少相同的倍数。 在实践中，我们通过设置输入图像的分辨率隐式地设置ρ。

We can now express the computational cost for the core layers of our network as depthwise separable convolutions with width multiplier α and resolution multiplier ρ:



where ρ ∈ (0,1] which is typically set implicitly so that the input resolution of the network is 224, 192, 160 or 128. ρ = 1 is the baseline MobileNet and ρ < 1 are reduced computation MobileNets. Resolution multiplier has the effect of reducing computational cost by ρ2.

现在，我们可以将具有宽度系数α和分辨率系数ρ的深度可分离卷积的网络核心层的计算成本表示为：



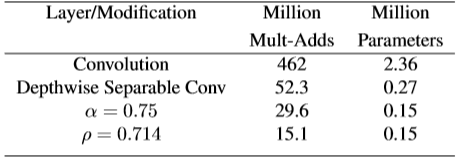
其中ρ∈（0,1），通常是隐式设置的，因此网络的输入分辨率为224,192,160或128.ρ= 1是基本的MobileNet，ρ<1是减少计算的MobileNets。**分辨率系数能降低计算成本ρ2倍**。

As an example we can look at a typical layer in MobileNet and see how depthwise separable convolutions, width multiplier and resolution multiplier reduce the cost and parameters. Table3 shows the computation and number of parameters for a layer as architecture shrinking methods are sequentially applied to the layer. The first row shows the Mult-Adds and parameters for a full convolutional layer with an input feature map of size 14×14×512 with a kernel K of size 3 × 3 × 512 × 512. We will look in detail in the next section at the trade offs between resources and accuracy.

举一个例子，我们可以看一下MobileNet中的典型层，看看深度可分离卷积，宽度系数和分辨率系数如何降低运算成本和参数。 表3显示了层的参数和计算的数量，因为体系结构收缩方法被顺序地应用于层。 第一行显示了标准卷积层的Mult-Adds操作次数和参数量，其输入特征图大小为14×14×512，卷积核K大小为3×3×512×512。我们将在下一节中详细介绍在资源和准确性之间的权衡。

Table3.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 DK = 3, M = 512, N = 512, DF = 14.

表3.修改版和标准版卷积的资源使用情况。 请注意，每行都是在上一行之上添加的累积效果。 此示例适用于DK = 3，M = 512，N = 512，DF = 14的内部MobileNet层。



1. **Experiments**

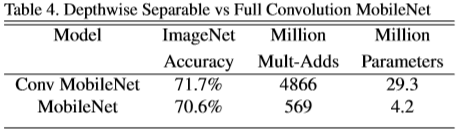
In this section we first investigate the effects of depthwise convolutions as well as the choice of shrinking by reducing the width of the network rather than the number of layers. We then show the trade offs of reducing the network based on the two hyper-parameters: width multiplier and resolution multiplier and compare results to a number of popular models. We then investigate MobileNets applied to a number of different applications.

在本节中，**我们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，我们展示了减小网络的两个超参数的权衡：宽度系数和分辨率系数，并将结果与许多流行模型进行比较。** 然后，我们研究了将MobileNets应用于许多不同应用程序。

* 1. **Model Choices**

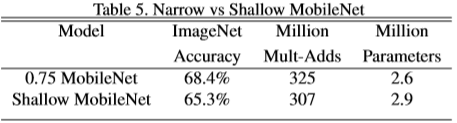
First we show results for MobileNet with depthwise separable convolutions compared to a model built with full convolutions. In Table 4 we see that using depthwise separable convolutions compared to full convolutions only reduces accuracy by 1% on ImageNet was saving tremendously on mult-adds and parameters.

首先，我们展示了具有深度可分离卷积的MobileNet与使用标准卷积构建的模型相比的结果。 在表4中，我们看到使用深度可分离卷积与标准卷积相比，ImageNet上的精度仅降低了1％，但大大节省乘加操作次数和参数量。



We next show results comparing thinner models with width multiplier to shallower models using less layers. To make MobileNet shallower, the 5 layers of separable filters with feature size 14 × 14 × 512 in Table 1 are removed. Table 5 shows that at similar computation and number of parameters, that making MobileNets thinner is 3% better than making them shallower.

我们接下来展示的结果是:将使用宽度系数训练的简化模型与使用更少层的较浅模型进行比较。 为了使MobileNet更浅，删除了表1中特征尺寸为14×14×512的5个可分离滤波器层。 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效果好3％。



* 1. **Model Shrinking Hyperparameters**

Table 6 shows the accuracy, computation and size trade offs of shrinking the MobileNet architecture with the width multiplier α. Accuracy drops off smoothly until the architecture is made too small at α = 0.25.

表6显示了使用宽度系数α缩小MobileNet架构的准确性，计算和大小权衡。 精度随着α的减小平滑下降，直到在α= 0.25时由于结构太小而下降很多。

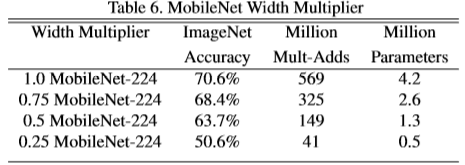


Table 7 shows the accuracy, computation and size trade offs for different resolution multipliers by training MobileNets with reduced input resolutions. Accuracy drops off smoothly across resolution.

表7显示了通过使用不同分辨率系数训练具有降低的输入分辨率MobileNets的精度，计算量和参数大小的权衡。 精度在分辨率下平滑下降。

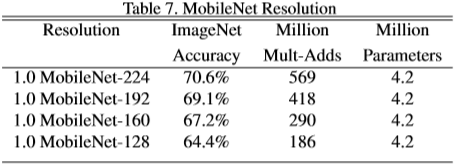


Figure 4 shows the trade off between ImageNet Accuracy and computation for the 16 models made from the cross product of width multiplier α ∈ {1,0.75,0.5,0.25} and resolutions{224,192,160,128}. Results are log linear with a jump when models get very small at α = 0.25.

图4显示了16个模型在ImageNet上的精度与计算量之间的权衡，这些模型由宽度系数α∈{1,0.75,0.5,0.25}和分辨率系数{224,192,160,128}的交叉组合构成。 模型在α= 0.25时变得很小，结果是对数线性跳跃的。

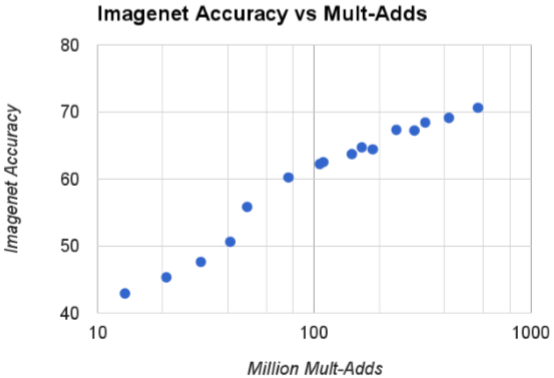


Figure 4. This figure shows the trade off between computation (Mult-Adds) and accuracy on the ImageNet benchmark. Note the log linear dependence between accuracy and computation.

Figure 5 shows the trade off between ImageNet Accuracy and number of parameters for the 16 models made from the cross product of width multiplier α ∈ {1,0.75,0.5,0.25}and resolutions{224,192,160,128}.

图5显示了16个模型在ImageNet上的精度与参数数量之间的权衡，这些模型由宽度系数α∈{1,0.75,0.5,0.25}和分辨率系数{224,192,160,128}的交叉组合构成。

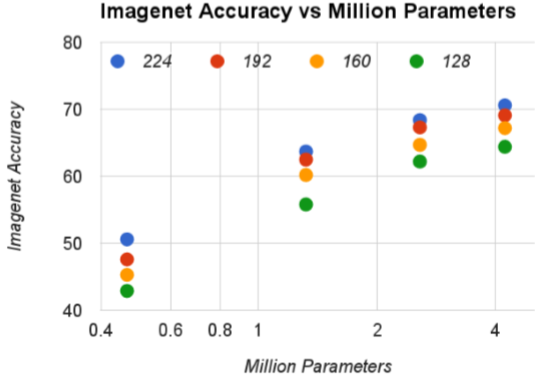


Figure 5. This figure shows the trade off between the number of parameters and accuracy on the ImageNet benchmark. The colors encode input resolutions. The number of parameters do not vary based on the input resolution.

Table 8 compares full MobileNet to the original GoogleNet [30] and VGG16 [27]. MobileNet is nearly as accurate as VGG16 while being 32 times smaller and 27 times less compute intensive. It is more accurate than GoogleNet while being smaller and more than 2.5 times less computation.

表8将完整的MobileNet与原始GoogleNet [30]和VGG16 [27]进行了比较。 MobileNet几乎与VGG16一样精确，同时模型缩小了32倍，计算密集度降低了27倍。 它比GoogleNet更准确，而且计算量减少了2.5倍。

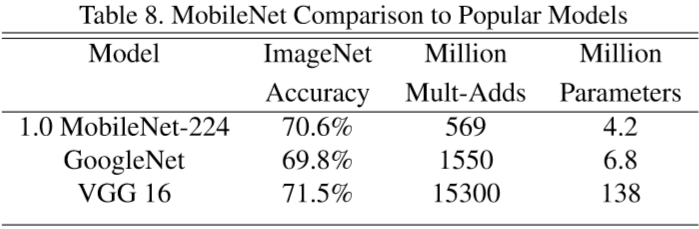
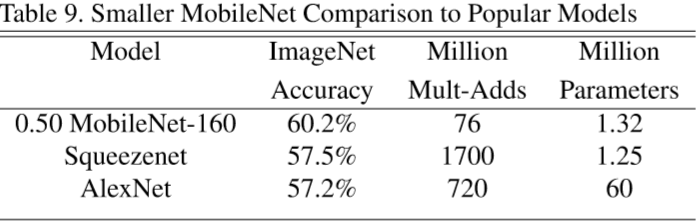


Table 9 compares a reduced MobileNet with width multiplier α = 0.5 and reduced resolution 160×160. Reduced MobileNet is 4% better than AlexNet[19] while being 45× smaller and 9.4×less compute than AlexNet. It is also 4% better than Squeezenet [12] at about the same size and 22× less computation.

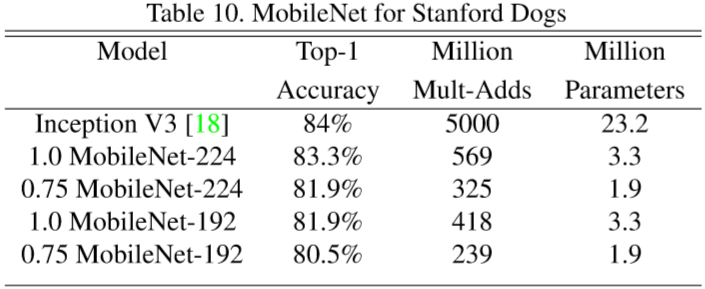
表9比较了使用宽度系数α= 0.5和降低分辨率到160×160的MobileNet和AlexNet，MobileNet效果比AlexNet好4％[19]**【不是3%么？】**，同时参数量比AlexNet小45倍，计算量减少9.4倍。 它也比Squeezenet [12]好4％，模型大小相同，计算量减少22倍。



* 1. **Fine Grained Recognition**

We train MobileNet for fine grained recognition on the Stanford Dogs dataset[17]. We extend the approach of [18] and collect an even larger but noisy training set than [18] from the web. We use the noisy web data to pretrain a fine grained dog recognition model and then fine tune the model on the Stanford Dogs training set. Results on Stanford Dogs test set are in Table 10. MobileNet can almost achieve the state of the art results from [18] at greatly reduced computation and size.

我们在Stanford Dogs数据集上训练MobileNet进行细粒度识别[17]。 我们扩展了[18]的方法，并从网上收集了比[18]更大但嘈杂的训练集。 我们使用嘈杂的网络数据预先训练细粒度识别狗的模型，然后在Stanford Dogs训练集上微调模型。 Stanford Dogs测试集的结果见表10.MobileNet几乎可以在大大减少计算量和模型尺寸的情况下实现[18]的结果。



* 1. **Large Scale Geolocalizaton**

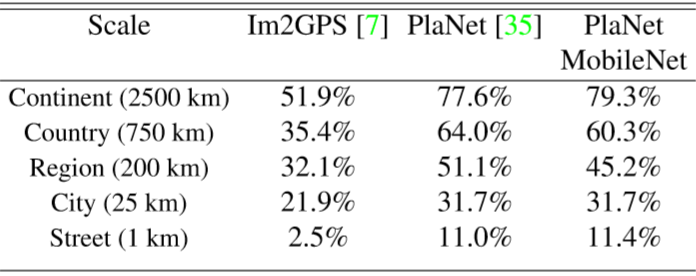
PlaNet [35] casts the task of determining where on earth a photo was taken as a classification problem. The approach divides the earth into a grid of geographic cells that serve as the target classes and trains a convolutional neural network on millions of geo-tagged photos. PlaNet has been shown to successfully localize a large variety of photos and to outperform Im2GPS [6, 7] that addresses the same task.

PlaNet [35]的任务是将确定照片的拍照位置作为分类问题。 该方法将地球划分为一个个地理单元格网格，作为目标类别，并在数百万张带地理标记的照片上训练卷积神经网络。 PlaNet已经被证明可以成功地定位各种各样的照片，并且胜过可以解决相同的任务的Im2GPS [6,7]。

We re-train PlaNet using the MobileNet architecture on the same data. While the full PlaNet model based on the Inception V3 architecture [31] has 52 million parameters and 5.74 billion mult-adds. The MobileNet model has only 13 million parameters with the usual 3 million for the body and 10 million for the final layer and 0.58 Million mult-adds. As shown in Tab. 11, the MobileNet version delivers only slightly decreased performance compared to PlaNet despite being much more compact. Moreover, it still outperforms Im2GPS by a large margin.

我们使用MobileNet架构在相同数据上重新训练PlaNet。基于Inception V3架构的完整版PlaNet模型[31]拥有5200万个参数和57.4亿个乘加操作。 MobileNet模型只有1300万个参数，通常是网络主体300万个参数，最终层1000万个参数和58万个乘加操作。 如表格11所示，与PlaNet相比，MobileNet版本的性能略有下降，但是更紧凑。 而且，它仍然极大地优于Im2GPS。

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



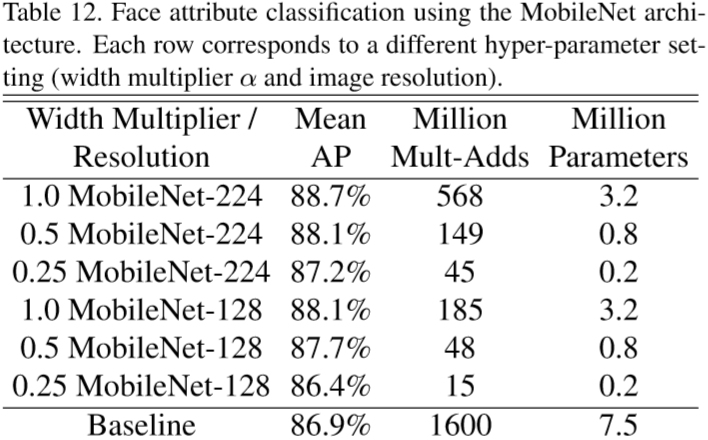
* 1. **Face Attributes**

Another use-case for MobileNet is compressing large systems with unknown or esoteric training procedures. In a face attribute classification task, we demonstrate a synergistic relationship between MobileNet and distillation [9], a knowledge transfer technique for deep networks. We seek to reduce a large face attribute classifier with 75 million parameters and 1600 million Mult-Adds. The classifier is trained on a multi-attribute dataset similar to YFCC100M [32].

MobileNet的另一个用例是以一种未知的或深奥（黑盒？）的训练过程压缩大型系统。 在面部属性分类任务中，我们展示了MobileNet和蒸馏之间的协同关系[9]，这是一种深度网络的知识转移技术。 我们寻求将一个大型的面部属性分类器的参数减少到7500万个，Mult-Adds操作数为1600万个。 分类器是在类似于YFCC100M [32]的多属性数据集上训练的。

We distill a face attribute classifier using the MobileNet architecture. Distillation [9] works by training the classifier to emulate the outputs of a larger model (The emulation quality is measured by averaging the per-attribute cross-entropy over all attributes) instead of the ground-truth labels, hence enabling training from large (and potentially infinite) unlabeled datasets. Marrying the scalability of distillation training and the parsimonious parameterization of MobileNet, the end system not only requires no regularization (e.g. weight-decay and early-stopping), but also demonstrates enhanced performances. It is evident from Tab. 12 that the MobileNet-based classifier is resilient to aggressive model shrinking: it achieves a similar mean average precision across attributes (mean AP) as the in-house while consuming only 1% the Multi-Adds.

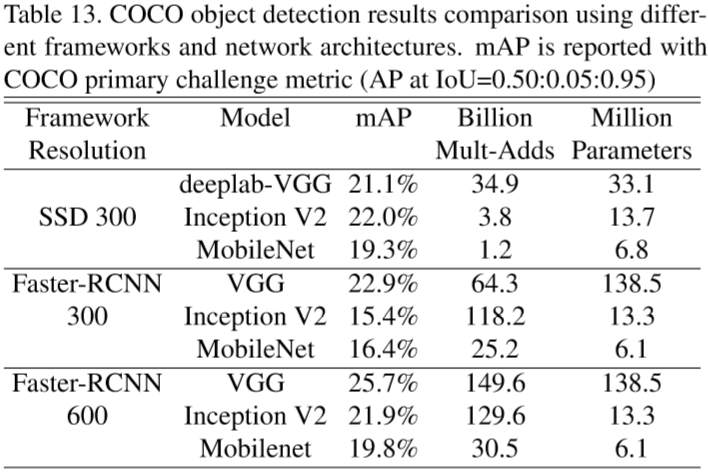
我们使用MobileNet架构提取面部属性分类器。 蒸馏[9]通过训练分类器来模拟较大模型的输出（模拟质量通过平均每个属性的交叉熵来测量）而不是真实标签，从而实现训练大型（ 并且可能是无限的）未标记的数据集。 结合蒸馏训练的可扩展性和MobileNet的简约参数化，终端系统不仅不需要正规化（例如，权重衰减和早停），而且还表现出更好的性能。 从Tab12中可以看出这一点。基于MobileNet的分类器对于积极的模型**【积极的模型？】**收缩具有弹性：它在内部实现了与属性（平均AP）相似的平均精度，同时仅占用了1％的乘加操作。



* 1. **Object Detection**

MobileNet can also be deployed as an effective base network in modern object detection systems. We report results for MobileNet trained for object detection on COCO data based on the recent work that won the 2016 COCO challenge [10]. In table 13, MobileNet is compared to VGG and Inception V2 [13] under both Faster-RCNN [23] and SSD [21] framework. In our experiments, SSD is evaluated with 300 input resolution (SSD 300) and Faster-RCNN is compared with both 300 and 600 input resolution (Faster RCNN 300, Faster-RCNN 600). The Faster-RCNN model evaluates 300 RPN proposal boxes per image. The models are trained on COCO train+val excluding 8k minival images and evaluated on minival. For both frameworks, MobileNet achieves comparable results to other networks with only a fraction of computational complexity and model size.

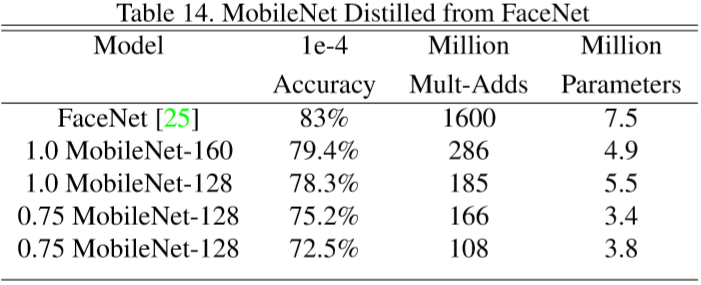
MobileNet还可以作为现代物体检测系统中的有效基础网络进行部署。 根据最近的工作报告，我们针对COCO数据集进行物体检测训练的MobileNet的结果[10]赢得2016年COCO挑战赛。 在表13中，MobileNet与以VGG和Inception V2 [13]为骨干网络的Faster-RCNN [23]和SSD [21]框架进行了比较。 在我们的实验中，SSD使用300作为输入分辨率（SSD 300）进行评估，并将Faster-RCNN与300和600作为输入分辨率（Faster RCNN 300，Faster-RCNN 600）进行比较。 Faster-RCNN模型每个图像评估300个RPN提议框。 这些模型在COCO train + val上训练，拿出了8k的minival，并在minival上进行评估。 对于这两个框架，MobileNet实现了与其相当的结果，但是计算复杂性和模型大小相当少。



* 1. **Face Embeddings**

The FaceNet model is a state of the art face recognition model [25]. It builds face embeddings based on the triplet loss. To build a mobile FaceNet model we use distillation to train by minimizing the squared differences of the output of FaceNet and MobileNet on the training data. Results for very small MobileNet models can be found in table 14.

FaceNet模型是最先进的人脸识别模型[25]。 它基于三元组损失构建face embeddings。 为了构建移动FaceNet模型，我们使用蒸馏来训练，从而使得FaceNet和MobileNet的输出在训练数据上的平方差最小。 可以在表14中看到， MobileNet的模型非常小。



1. **Conclusion**

We proposed a new model architecture called MobileNets based on depthwise separable convolutions. We investigated some of the important design decisions leading to an efficient model. We then demonstrated how to build smaller and faster MobileNets using width multiplier and resolution multiplier by trading off a reasonable amount of accuracy to reduce size and latency. We then compared different MobileNets to popular models demonstrating superior size, speed and accuracy characteristics. We concluded by demonstrating MobileNet’s effectiveness when applied to a wide variety of tasks. As a next step to help adoption and exploration of MobileNets, we plan on releasing models in Tensor Flow.

我们提出了一种基于深度可分离卷积的称为MobileNets的新模型架构。 我们研究了一些训练高效模型的重要设计决策。 然后，我们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，通过折衷合理的精度来减小模型大小和延迟。 然后，我们将不同的MobileNets与流行的模型进行了比较，展示了出众的模型大小，速度和准确性。 最后，我们通过展示MobileNet应用于各种任务时的有效性得出结论。 作为帮助采用和探索MobileNets的下一步，我们计划在Tensor Flow中发布模型。