MobileNetV2：倒置的残差和线性瓶颈

Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-ChiehChen
谷歌公司。
{sandler, howarda, menglong, azhmogin, lcchen}@google.com

摘要

在本文中，我们描述了一个新的移动架构，MobileNetV2，它提高了移动模型在多个任务和基准上的性能，以及在不同的模型规模的光谱上的性能。我们还描述了在我们称之为SSDLite的新框架中应用这些移动模型进行物体检测的有效方法。此外，我们还展示了如何通过DeepLabv3的简化形式来建立移动语义分割模型，我们称之为Mobile DeepLabv3。

是基于一个倒置的残差结构，其中的捷径连接是在薄的瓶颈层之间。中间的扩展层使用轻量级的深度卷积来过滤特征，作为非线性的来源。此外，我们发现，为了保持表现力，去除窄层中的非线性是很重要的。我们证明这可以提高性能，并提供了一个导致这种设计的in-tuition。

最后，我们的方法允许将输入/输出域与转换形成的表现力解耦，这为进一步分析提供了一个方便的框架。我们在ImageNet[1]分类、COCO对象检测[2]、VOC图像分割[3]上衡量我们的性能。我们评估了准确度、用乘加法（MAdd）衡量的操作数量以及实际延迟和参数数量之间的权衡。

1. 简介

神经网络已经彻底改变了机器智能的许多领域，使具有挑战性的图像识别任务具有超人的准确性。然而，提高准确性的动力往往是有代价的：现代最先进的网络需要大量的计算资源，超出了许多移动和嵌入式的能力。

本文介绍了一种新的神经网络结构，它是专门为移动和资源有限的环境量身定做的。我们的网络推动了移动定制计算机视觉模型的技术发展，在保持相同精度的同时，大大减少了所需的操作和内存数量。

我们的主要贡献是一个新的层模块：具有线性瓶颈的倒置残差。该模 块 将低维压缩表示作为输入，首先扩展到高维，并通过轻量级的深度对话进行过滤。随后用*线性卷积法将*特征投射回低维表示。正式的实现是作为TensorFlow-Slim模型库的一部分，见[4]。

这个模块可以在任何现代框架中使用标准操作有效地实现，并使我们的模型在多个性能点上超过了标准基准的水平。此外，这个卷积模块特别适合于移动设计，因为它可以通过不完全物化大的中间张量来显著地减少在融合过程中所需要的内存占用。这减少了许多嵌入式硬件设计中对主内存访问的需求，这些设计提供了少量非常快速的软件控制的高速缓存。

2. 相关工作

在过去的几年里，调整深度神经架构以实现准确性和性能之间的最佳平衡一直是一个活跃的研究领域。许多团队进行的人工架构搜索和训练算法的改进都导致了对早期设计的巨大改进，如AlexNet[5]、VGGNet[6]、GoogLeNet[7]。和ResNet[8]。最近，在算法结构探索方面有很多进展，包括超参数优化[9, 10, 11]以及各种

网络修剪[12, 13, 14, 15, 16, 17]和连接学习[18, 19]的方法。大量的工作也致力于改变内部卷积块的连接性结构，如ShuffleNet[20]或引入稀疏性[21]和其他[22]。

最近，[23, 24, 25, 26]，开辟了一个新的方向，将包括遗传算法和强化学习在内的优化方法引入建筑搜索。然而，一个缺点是所产生的网络最终会变得非常复杂。在本文中，我们追求的目标是发展关于神经网络工作原理的更好的直觉，并利用它来指导最简单的网络设计。我们的方法应该被看作是对[23]和相关工作中描述的方法的兼容。在这一点上，我们的方法与[20, 22]所采取的方法类似，可以进一步提高性能，同时提供对其内部运作的一瞥。我们的网络设计是基于MobileNetV1[27]。它重新保持了它的简单性，不需要任何特殊的运算器，同时显著提高了它的准确性，在移动应用的多种图像分类和检测任务上达到了最先进的水平。

3. 序言、讨论和直觉

3.1. 深度可分离的卷积

深度可分离卷积是许多高效神经网络架构的关键构建模块[27, 28, 20]，我们在本工作中也使用了它们。其基本思想是用一个将卷积分成两个独立层的因子化版本来取代全卷积运算。第一层被称为深度卷积，它通过对每个输入通道应用单个卷积滤波器来进行轻量级过滤。第二层是一个1×1的卷积，称为点式卷积。
卷积，它负责建立新的fea-。
通过计算输入通道的线性组合来实现。

标准卷积需要一个$h_i \times w_i \times d_i$的输入张量$L_i$，并应用卷积核$K \in \mathrm{R}^{k \times k \times d\ i \times d j}$来产生一个$h_i \times w_i \times d_j$的输出张量$L_j$。标准卷积层的编译成本为$h_i - w_i - d_i - d_j - k - k$。

深度可分离的卷积是一种落地式的再创造。
经验表明，它们几乎与普通卷积层一样好用，但只需花费：1,000-2,000美元。根据经验，它们的效果几乎与普通卷积层一样好，但只需要花费：

2

$$h_i - w_i - d_i\ (k + d)_j \qquad (1)$$

它是深度和1×1点状的总和
卷积。与传统的图层相比，有效的深度可分离卷积减少了几乎一个系数*的*计算量。[21].MobileNetV2使用$k = 3$
(3×3深度可分离的卷积)，因此，编译后的
其计算成本是标准卷积的8到9倍，而准确度仅有小幅下降[27]。

3.2. 线性瓶颈

考虑一个由n个层组成的深度神经网络L_i，每个层都有一个维度为$h_i \times w_i \times d_i$的激活张量。在本节中，我们将讨论这些激活张量的基本属性，我们将把这些张量视为具有d_i维度的$h_i \times w_i$"像素"容器。非正式地，对于一个真实图像的输入集，我们说层激活的集合（对于任何层L_i）形成一个"感兴趣的流形"。长期以来，人们一直认为神经网络中感兴趣的流形可以嵌入到低维的子空间中。换句话说，当我们看一个深度卷积层的所有单个*d通道*像素时，信息
编码的这些数值实际上位于某个流形中，而这个流形又可以嵌入到一个低维的子空间中。[2].

乍一看，这样的事实可以通过简单地减少一个层的维度，从而减少操作空间的维度来捕捉和利用。MobileNetV1[27]已经成功地利用了这一点，通过一个宽度乘数参数，有效地在com-putation和准确性之间进行权衡，并且也被纳入其他网络的有效模型设计中[20]。根据这一直觉，宽度乘数的方法允许人们减少激活空间的维度，直到感兴趣的模型横跨整个空间。然而，当我们回忆起深层对话神经网络实际上有非线性的同向变换（如ReLU）时，这种直觉就会被打破。比如说、
ReLU在一维空间中应用于一条线，会产生一条"射线"，而在R^n空间中，它通常会产生一个片状的有n个关节的线性曲线。

很容易看出，一般来说，如果一个层变换ReLU(Bx)的结果有一个非零的体积S，那么映射到内部S的点是通过输入的线耳变换B得到的，因此说明
输入空间中对应于全部二元输出的部分，被限制为线性转换。换句话说，深度网络只具有线性分类器在非零体积部分的能力。

[1] 更确切地说，是以系数$k\ d^2_j\ /(k^2 + d_j)$

[2] 注意流形的维度不同于子空间的维 度，后者可以通过线 性变 换 嵌入。

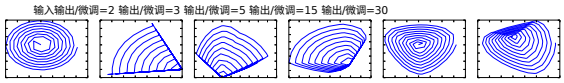


图1：低维流形嵌入高维空间的ReLU变换的例子。在这些例子中，初始螺旋被嵌入到一个n维空间，使用随机矩阵T，然后是ReLU，然后使用T^{-1}投射回二维空间。在上面的例子中，n=2，3会导致信息损失，其中流形的某些点会相互坍塌，而对于n=15到30的情况，转变是高度非凸的。

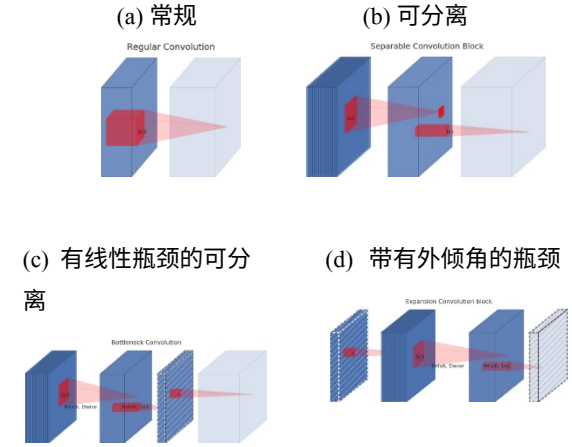


图2：可分离卷积块的演变。对角线阴影的纹理表示不包含非线性的层。最后一个（浅色）层表示下一个块的开始。注意：2d和2c在堆叠时是等价的块。最好以彩色观看。

输出域。我们参考补充材料，以获 得 更正式的声明。

另一方面，当ReLU折叠通道时，它不可避免地失去了*该通道*的信息。然而，如果我们有很多通道，并且有一个结构
在激活流形中，信息可能仍然保留在其他通道中。在补充材料中，我们表明，如果输入流形可以被嵌入到激活空间的一个明显的低维子空间中，那么ReLU变换就可以保留信息，同时引入所需的
复杂度为可表达函数的集合。

总而言之，我们强调了两个特性，它们表明了感兴趣的流形应该位于高维激活空间的低维子空间的要求：

1.如果感兴趣的流形在ReLU变换后仍保持非零体积，那么它就相当于一个线性变换。

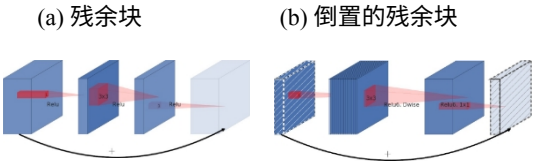


图3：残差块[8，30]和倒置残差的区别。对角线加盖的层不使用非线性。我们用每个块的厚度来表示它的相对通道数。请注意古典残差是如何连接通道数量较多的层的，而倒残差则是连接两个层的。最好以彩色观看。

2.ReLU能够保留关于输入流形的完整信息，但只有当输入流形位于输入空间的低维子空间时，才能保留。

这两个观点为我们提供了一个优化现有神经结构的经验提示：假设感兴趣的流形是低维的，我们可以通过将*线性瓶颈*层插入其中来实现这一目标。

卷积块。实验证据表明，使用线性层是至关重要的，因为它可以防止非线性破坏太多的信息。在第6节中，我们通过经验表明，在瓶颈中使用非线性层确实会损害几个百分点的性能，进一步验证了我们的假设[3].我们注意到在[29]中也有类似的报告，其中非线性被帮助了，非线性被从传统的残差块的输入中移除，这导致了CIFAR数据集的性能提高。

在本文的其余部分，我们将利用瓶颈卷积。我们将把输入瓶颈的大小和内部大小之间的比率称为*扩展率*。

3.3. 倒置的残差

瓶颈区块看起来类似于残差区块，每个区块包含一个输入，然后是几个瓶颈，接着是扩展[8]。然而，受瓶颈实际上包含所有必要信息的直觉启发，而扩展层只是作为一个执行细节，伴随着十进制的非线性转换，我们在瓶颈之间直接使用捷径。

[3] 我们注意到，在存在捷径的情况下，信息损失其实并不强烈。

图3提供了一个设计中差异的可视化示意图。插入短切的动机与经典的剩余连接相似：我们想提高梯度在乘法器层中传播的能力。然而，倒置设计的内存效率要高得多（详见第5节），而且在我们的实验中效果也略好。

瓶颈卷积的运行时间和参数数 基本的实现结构是il-在表1中列出。对于一个大小为$h \times w$，外倾系数为t，内核大小为k，具有d'输入通道和d''输出通道的块，所需的多层添加总数为$h \cdot w \cdot d' \cdot t$（$d' + k^2 + d''$）。与（1）相比，这个表达式有一个额外的项，因为事实上我们有一个额外的1×1卷积，然而我们网络的性质允许我们利用小得多的输入和输出尺寸。在表3中，我们比较了MobileNetV1、MobileNetV2和ShuffleNet之间每个分辨率所需的尺寸。

3.4. 信息流解释

我们架构的一个有趣的特性是，它在构件（瓶颈层）的输入/输出*领域*和*层的转换*之间提供了一个自然的分离--那是一个将输入转换为输出的非线性函数。前者可以被看作是网络在每一层的*能力*，而后者则是*表达能力*。这与传统的卷积块（包括常规卷积块和分离卷积块）形成对比，后者的表现力和容量都纠缠在一起，是输出层深度的函数。

特别是，在我们的案例中，当内层深度为0时，由于捷径连接，底层卷积是身份函数。当扩展比小于1时，这是一个经典的剩余卷积块[8, 30]。然而，对于我们的目的，我们表明，扩展率大于1是最有用的。

这种解释使我们能够将网络的表现力与它的能力分开研究，我们认为需要进一步探索这种分离，以更好地了解网络的特性。

4. 模型结构

现在我们详细描述一下我们的结构。正如上一节所讨论的，基本构件是一个带有残差的瓶颈深度可分卷积。这个模块的详细结构如图所示

输入	运营商	输出
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	线性1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

表1：*瓶颈剩余区块*转换，从k到k'通道，跨度为s，扩展系数为t。

表1.MobileNetV2的结构包含有32个过滤器的初始完全卷积层，然后是表2中描述的19个*剩余瓶颈*层。我们使用ReLU6作为非线性，因为它在用于低精度计算时具有鲁棒性[27]。我们总是使用内核大小为3×3的现代网络的标准，并利用dropout和batch nor...
训练期间的恶化。

除了第一层之外，我们在整个网络中使用恒定的扩展率。在我们的经验中，我们发现扩展率在5到10之间会产生几乎相同的性能曲线，较小的网络在扩展率稍小的情况下会更好，而较大的网络则有稍好的性能。
膨胀率较大时的性能。

在我们所有的主要实验中，我们使用适用于输入张量大小的扩展因子6。例如，对于一个接受64通道输入张量并产生128通道张量的瓶颈层，中间扩展层是64 · 6 = 384通道。

权衡超参数 在[27]中，我们通过使用输入图像的分辨率和宽度乘数作为可调整的超参数，使我们的架构适应不同的性能点，这些参数可以根据所需的精度/性能权衡进行调整。我们的首要
矩阵网络（宽度乘数1，224×224）的计算成本为3亿次乘法，使用340万个参数。我们探讨了输入分辨率从96到224，宽度乘数从0.35到1.4的性能权衡。该网络的功能是
投入成本从7倍增加到5.85亿不等
规模，而模型的规模在1.7M和1.5M之间变化。
6.9M的参数。

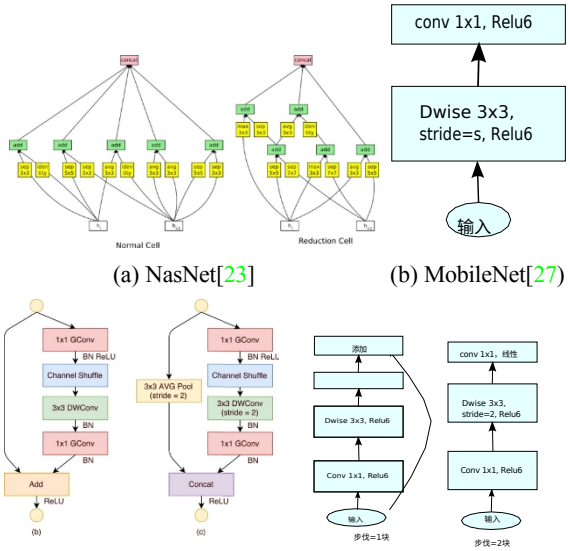
与[27]在实现上的一个小区别是，对于乘数小于1的情况，我们对所有的层都应用了宽度倍数，除了最后的卷积层。这提高了小模型的性能。

输入	运营商	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	瓶颈	1	16	1	1
$112^2 \times 16$	瓶颈	6	24	2	2
$56^2 \times 24$	瓶颈	6	32	3	2
$28^2 \times 32$	瓶颈	6	64	4	2
$14^2 \times 64$	瓶颈	6	96	3	1
$14^2 \times 96$	瓶颈	6	160	3	2
$7^2 \times 160$	瓶颈	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

表2：MobileNetV2：每一行描述了一个由1个或多个相同的（modulo stride）层组成的序列，重复n次。同一序列中的所有层都有相同数量的输出通道c。每个层的第一层序列有一个跨度s，其他都使用跨度1。所有的空间卷积都使用3×3核。扩张因子t总是应用于输入尺寸，如所述在表1中。

尺寸	移动网络V1	移动网络V2	甩手网 (2x,g=3)
112x112	64/1600	16/400	32/800
56x56	128/800	32/200	48/300
28x28	256/400	64/100	400/600K
14x14	512/200	160/62	800/310
7x7	1024/199	320/32	1600/156
1x1	1024/2	1280/2	1600/3
最大	1600K	**400K**	600K

表3：在不同的架构下，每个空间重复需要物化的通道/内存的最大数量（单位：Kb）。我们假设16位的浮点数用于激活。对于ShuffleNet，我们使用$2x$，g = 3，与MobileNetV1的性能相匹配。MobileNetV2。对于MobileNetV2的第一层和在ShuffleNet中，我们可以采用第5节中描述的技巧来减少内存需求。 尽管ShuffleNet在其他地方采用了瓶颈，由于非瓶颈的十进制之间存在捷径，非瓶颈的张量仍然需要被物化。检察官。



(a) NasNet[23] (b) MobileNet[27]

(c) ShuffleNet[20] (d) Mobilenet V2

图4：不同架构的卷积块的比较。ShuffleNet使用Group Convolu- tions[20]和shuffling，它也使用传统的resid- ual方法，内部块比外部窄。ShuffleNet和NasNet的插图来自各自的论文。

5. 实施说明

5.1. 内存高效推理

倒置的剩余瓶颈层允许一个特别有效的内存实现，这对移动应用非常重要。 一个标准的高效检察官。

使用TensorFlow[31]或Caffe[32]等推理的便捷实现，建立了一个有向无环计算超图G，由代表操作的边和代表中间计算的张量的结点组成。计算的安排是为了尽量减少需要存储在内存中的张量的总数。在最一般的情况下，它搜索所有可信的计算顺序$\Sigma(G)$并挑选出最小化的一个

$$M(G) = \min_{\pi \in \Sigma(G)} \max_{i \in 1 \ldots n} \left(\sum_{A \in R(i,\pi,G)} |A| + size(\pi_i) \right).$$

其中，$R(i, \pi, G)$是与$\pi_i \ldots \pi_n$节点中的任何一个相连的中间张量的列表，$|A|$代表张量A的大小，$size(i)$是运行期间内部存储所需的总内存量i.

对于只有琐碎的并行结构（如残余连接）的图，只有一个非琐碎的可行计算顺序，因此，推断所需的内存总量和界限

对计算图G*的影响*可以被简化：

$$M(G) = \max_{op \in G} \left(|A| + |B| + |op| \right) \quad (2)$$

或者重述一下，内存量只是所有操作中组合输入和输出的最大总大小。在下面的内容中，我们表明，如果我们把瓶颈剩余块当作一个单一的操作（并把内部卷积当作一个一次性的张量），总的
内存量将被瓶颈张量的大小所支配，而不是瓶颈内部张量的大小（而且大得多）。

瓶颈剩余块 图3b所示的瓶颈块运算符$F(x)$可以表示为三个运算符$F(x)=[A \circ N \circ B]x$*的*组合，其中$A$是线性变换$A: \mathbb{R}^{s \times s \times k} \rightarrow \mathbb{R}^{s \times s \times n}$，N是非线性的每通道变换：

$N: \mathbb{R}^{s \times s \times n} \rightarrow \mathbb{R}^{s' \times s' \times n}$，而$B$又是一个线性对输出域的转换：$B: \mathbb{R}^{s' \times s' \times n} \rightarrow \mathbb{R}^{s' \times s' \times k'}$。

对于我们的网络$N = \text{ReLU6} \circ \text{dwise} \circ \text{ReLU6}$，

但这些结果适用于任何每通道的转换。假设输入域的大小为|x|，输出域的大小为|y|，那么计算$F(X)$所需的内存可以低至$|s^2 k| + |s k'^2| + O(\max(s^2, s'^2))$。

该算法是基于这样一个事实，即内十索I可以表示为每个大小为n/t*的*张量的串联，然后我们的函数可以表示为

$$F(x) = \sum_{i=1}^{t} (A_i \circ N \circ B_i)(x)$$

通过累积总和，我们只需要在内存中一直保留一个大小为n/t*的*中间块。使用$n=t$，我们最终只需要在任何时候都保持一个通道的中间表示。使我们能够使用这个技巧的两个约束条件是
(a)内部转换（包括非线性和深度）是按通道进行的，(b)连续的非按通道运算符的输入大小与输出有明显的比例。对于大多数传统的神经网络，这种技巧不会产生明显的改善。

我们注意到，使用t-*way* split计算$F(X)$所需的乘法运算器的数量与t有关，但是在现有的实现中，我们发现用七个较小的乘法运算器来代替一个矩阵乘法，会损害运行时的性能，这是因为

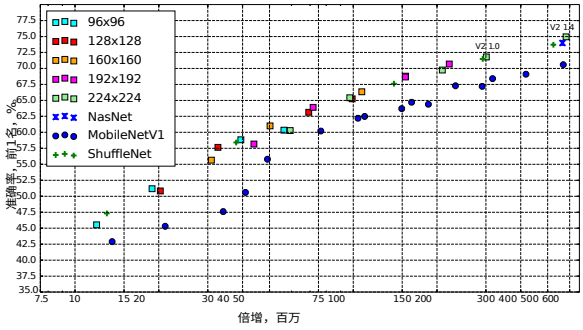
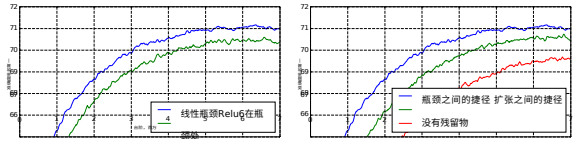


图5： MobileNetV2与MobileNetV1、ShuffleNet、NAS的性能曲线。对于我们的网络，我们在所有的决议中使用乘数0.35、0.5、0.75、1.0、
和额外的1.4为224。最好以彩色观看。



(a) (b)中非线性的影响 变化的影响
瓶颈层。 残余块。

图6：非线性和各种类型的捷径（剩余）连接的影响。

增加了高速缓存的失误。我们发现，在t为2到5之间的小常数时，这种方法是最有用的。它大大降低了对内存的要求，但仍然允许人们利用通过使用高度优化的矩阵乘法和卷积运算符所获得的大部分效率。
学习框架。特殊的框架层面的优化是否会导致进一步的运行时间改进，还有待观察。

6. 实验

6.1. 图像网分类

训练设置 我们使用TensorFlow[31]训练我们的模型。我们使用标准的RMSPropOp- timizer，衰减和动量都设置为0.9。我们在每一层之后使用批量归一化，标准权重衰减被设置为0.00004。按照MobileNetV1[27]的设置，我们使用初始学习率为0.045，学习率衰减率为0.98/ epoch。我们使用16个GPU异步工作者，批处理量为96。

结果 我们将我们的网络与MobileNetV1、ShuffleNet和NASNet-A模型进行比较。表4显示了几个选定模型的统计数据，图5显示了完整的性能图。

6.2. 物体检测

我们评估和比较了以下的性能
MobileNetV2和MobileNetV1作为特征提取器
[33]在COCO数据集[2]上用单次检测器（SSD）[34]的修改版进行物体检测。我们还与YOLOv2[35]和原始SSD（以VGG-16[6]为基础网络）作为基线进行了比较。我们没有与其他架构如Faster-RCNN[36]和RFCN[37]比较性能，因为我们的重点是移动/实时模型。

SSDLite：在本文中，我们介绍了一个对常规SSD友好的移动变体。我们在SSD预测层中用可分离的卷积（深度，然后是1×1的投影）取代所有的常规卷积。这一设计与 "Solidation "的整体设计是一致的。
我们认为这个版本的计算效率要高得多。我们称这个修改后的版本为SSDLite。与普通的SSD相比，SSDLite极大地减少了参数数量和计算成本，如表5所示。

对于MobileNetV1，我们遵循[33]中的设置。对于MobileNetV2，SSDLite的第一层被连接到第15层的扩展（输出跨度为16）。第二层和其余的SSDLite层连接在最后一层的顶部（输出跨度为32）。这个设置与MobileNetV1是一致的，因为所有的层都连接在一起。
到相同输出步数的特征图。

网络	前1名	参数	帐户	CPU
移动网络V1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
移动网络V2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

表4：ImageNet上的性能，不同网络的比较。按照操作的惯例，我们计算了Multiply-Adds的总数。在最后一栏，我们报告了谷歌Pixel 1手机（使用TF-Lite）单个大核心的运行时间，单位为毫秒（ms）。我们没有报告ShuffleNet的数字，因为高效的群组卷积和洗牌还没有得到支持。

	参数	帐户
SSD[34]。	14.8M	1.25B

表5：比较SSD和SSDLite配置MobileNetV2并对80个类进行预测的大小和计算成本。

网络	ÄÄÄ	参数	增长	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

表6：MobileNetV2 + SSDLite和其他实时检测器在COCO数据集物体检测任务上的性能比较。MobileNetV2 + SSDLite以更少的参数和更小的计算复杂度实现了具有竞争力的准确性。所有的模型都在`trainval35k`上训练，并在`test-dev`上评估。SSD/YOLOv2的数字来自[35]。运行时间是针对谷歌Pixel 1手机的大核心报告的，使用的是TF-Lite引擎的内部版本。

两个MobileNet模型都是用开源的TensorFlow物体检测API[38]进行训练和评估。两个模型的输入分辨率都是320×100。
320.我们对mAP（COCO
挑战指标），参数数量和数量
的乘法加法。结果显示在表6中。MobileNetV2 SSDLite不仅是最有效的模型，也是三者中最准确的。值得注意的是，MobileNetV2 SSDLite的效率是20倍，而且是最准确的。
缩小10倍，同时在COCO上仍优于YOLOv2
数据集。

6.3. 语义分割

在本节中，我们将MobileNetV1和MobileNetV2模型作为特征提取器与DeepLabv3[39]进行比较，以完成移动语义分割的任务。DeepLabv3采用了Atrous卷积[40, 41, 42]，这是一个明确控制计算出的特征图的重现的强大工具，并建立了五个准头，包括(a) Atrous Spatial Pyramid Pooling模块（ASPP）[43]，包含三个3×3的卷积。

不同心率下的运动，（b）1×1的卷积
头，以及(c)图像层面的特征[44]。我们用

*输出跨度*是指输入图像的空间分辨率与最终输出分辨率之比，这是由适当应用无序卷积来控制的。对于语义分离，我们通常采用*输出跨度*=16或8的密集特征图。我们在PASCAL VOC 2012数据集[3]上进行了实验，其中有额外的注解。
采集自[45]的图像和评估指标mIOU。

为了建立一个移动模型，我们尝试了三种设计变化：（1）不同的特征提取器，（2）简化DeepLabv3头以加快计算速度，以及（3）不同的推理策略以提高性能。我们的结果总结在表7中。我们观察到：(a) 推理策略，包括多尺度输入和添加左右翻转的图像，大大增加了MAdds，因此不适合设备上的应用、
(b) 使用*输出跨度*=16比*输出跨度*=8更有效，（c）MobileNetV1已经是一个强大的特征提取器，只需要比ResNet-101[8]少4.9-5.7倍的MAdds（*例如*，mIOU：
78.56*对*82.70，以及MAdds：941.9B*对*4870.6B），（d）。
在MobileNetV2的倒数第二层特征图上建立DeepLabv3头比在原始的最后一层特征图上建立DeepLabv3头更有效，因为倒数第二层特征图包含320个通道而不是1280个，通过这样做，我们达到了类似的性能，但需要的操作比MobileNetV1的对应操作少2.5倍，以及（e）DeepLabv3头在计算上很昂贵，删除ASPP模块大大降低了MAdds，而性能只是轻微下降。在表7的最后，我们确定了一个潜在的设备上应用的候选者（黑体字），它达到了75.32%的mIOU，只需要
27.5亿美元 MAdds.

6.4. 消融研究

颠倒的残余连接。剩余连接的重要性已经被广泛研究[8, 30, 46]。本文报告的新结果是，连接瓶颈的捷径比连接扩展层的捷径表现更好（对比见图6b）。

线性瓶颈的重要性。线性瓶颈模型严格来说不如具有非线性的模型强大，因为激活总是可以在线性系统中运行，并适当地改变偏移和缩放。然而，我们在图6a中的实验表明，线性瓶颈提高了性能，为非线性破坏了低维空间的信息提供了支持。

网络	操作系统	ASPP	基金会	硕士	参数	奖金
MNet V1	16	C		75.29	11.15M	14.25B
	8	C	C	78.56	11.15M	941.9B
MNet V2*	16	C		75.70	4.52M	5.8B
	8	C	C	78.42	4.52M	387B
MNet V2*	16			**75.32**	**2.11M**	**2.75B**
	8		C	77.33	2.11M	152.6B
里斯网-101	16	C		80.49	58.16M	81.0B
	8	C	C	82.70	58.16M	4870.6B

表7： MobileNet + DeepLabv3在PASCAL VOC 2012*验证*集上的推理策略。**MNet V2***：倒数第二张特征图用于DeepLabv3头部，其中包括（1）Atrous Spatial Pyramid Pool- ing（**ASPP**）模块，以及（2）1×1卷积以及图像池特征。**OS**：*输出跨度*，即con-
掌控着分割图的输出分辨率。**MF**：测试期间的多尺度和左右翻转的输入。所有的模型都在COCO上进行了预训练。在设备上应用的潜在候选模型以黑体字显示。PASCAL图像的维度为512×512，而Atrous卷积法允许我们控制输出特征。
在不增加参数数量的情况下，可以提高结构分辨率。

7. 结论和未来工作

我们描述了一个非常简单的网络结构，使我们能够建立一个高效的移动模型系列。我们的基本构建单元有几个特点，使它特别适合于移动应用。它允许非常有效的记忆推理，并依赖于所有神经框架中的标准操作。

对于ImageNet数据集，我们的架构在广泛的性能点上改善了技术现状。对于物体检测任务，我们的网络在准确度和模型复杂度方面都优于COCO数据集上的最先进的实时检测器。值得注意的是，我们的架构与SSDLite检测模型相结合，计算量减少20倍，参数减少10倍。比YOLOv2。

在理论方面：所提出的卷积块有一个独特的属性，允许将网络的表现力（由扩展层编码）与它的能力（由瓶颈输入编码）分开。对这一点进行研究是未来研究的一个重要方向。

鸣谢 我们要感谢Matt Streeter和Sergey Ioffe提供的

有益反馈和讨论。

参考文献

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, 和李飞飞。Imagenet 大规模视觉识别挑战。*Int.J. Comput.Vision*, 115(3):211-252, December 2015.1

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dolla´r, and C Lawrence Zitnick.微软COCO：语境中的常见对象。In *ECCV*, 2014.1, 7

[3] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserma.帕斯卡视觉对象类别挑战回顾。*IJCV*, 2014.1, 8

[4] Mobilenetv2源代码。可从 https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet。1

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton.用深度卷积神经网络进行图像网分类。在Bartlett等人[48]，第1106-1114页。1

[6] Karen Simonyan和Andrew Zisserman.用于大规模图像识别的极深卷积网络。*CoRR*, abs/1409.1556, 2014.1, 7

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Du- mitru Erhan, Vincent Vanhoucke, and Andrew Ra-binovich.更深入的卷积。在*IEEE计算机视觉和模式识别会议上，CVPR 2015，美国马萨诸塞州波士顿，2015年6月7-12日*，第1-9页。IEEE计算机学会、2015.1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.用于图像识别的深度残差学习.*CoRR*, abs/1512.03385, 2015.1, 3, 4, 8

[9] James Bergstra and Yoshua Bengio.超参数优化的随机搜索.*Journal of Machine Learning Research*, 13:281-305, 2012.1

[10] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams.医学学习算法的实用贝叶斯优化。In Bartlett et al. [48], pages 2960-2968.1

[11] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md.Mostofa Ali Patwary, Prabhat, and Ryan P. Adams.使用深度神经网络的可扩展的贝叶斯优化。In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd Interna- tional Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2171-2180.JMLR.org, 2015.1

[12] Babak Hassibi and David G. Stork.网络修剪的第二轨道导数：最佳的脑外科医生。In Stephen Jose Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural In- formation Processing Systems 5, [NIPS Confer- ence, Denver, Colorado, USA, November 30 -December 3, 1992]*, pages 164-171.摩根-考夫曼，1992.2

[13] Yann LeCun, John S. Denker, and Sara A. Solla.最佳的脑损伤。In David S. Touretzky, editor, *Advances in Neural Information Process- ing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 598-605.摩根-考夫曼，1989年。2

[14] Song Han, Jeff Pool, John Tran, and William J. Dally. 高效神经网络的权重和连接的学习。In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: An- nual Conference on Neural Information Process- ing Systems 2015, December 7-12, 2015, Mon-.*
加拿大魁北克省treal市，第1135-1143页，2015。2

[15] Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Shijian Tang, Erich Elsen, Bryan Catanzaro, John Tran, and William J. Dally。DSD：用密集-稀疏-密集训练流正则化深度神经网络。*CoRR*, abs/1607.04381, 2016.2

[16] Yiwen Guo, Anbang Yao, and Yurong Chen.高效dnns的动态网络手术。In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, ed- itors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Infor- mation Processing Systems 2016, December 5-10*、*2016年，西班牙巴塞罗那*，第1379-1387页，2016。2

[17] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf.高效信念网的修剪过滤器。*CoRR*, abs/1608.08710, 2016.2

[18] Karim Ahmed and Lorenzo Torresani.多分支网络中的联系性学习。*CoRR*, abs/1709.09582, 2017.2

[19] Tom Veniat and Ludovic Denoyer.用有预算的超级网络学习时间高效的深度架构。*CoRR*, abs/1706.00046, 2017.2

[20] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun.Shufflenet: 一个用于移动设备的极其有效的卷积神经网络。
CoRR, abs/1707.01083, 2017.2, 5

[21] Soravit Changpinyo, Mark Sandler, and Andrey Zhmoginov.卷积神经网络中稀疏性的力量。*CoRR*, abs/1702.06257, 2017.2

[22] Min Wang, Baoyuan Liu, and Hassan Foroosh.使用纯通道内卷积、拓扑细分和空间 "瓶颈 "结构设计高效卷积层。*CoRR*, abs/1608.04337, 2016.2

[23] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le.为可扩展的图像识别学习可转移的结构。*CoRR*, abs/1707.07012, 2017.2, 5

[24] Lingxi Xie和Alan L. Yuille. Genetic CNN. *CoRR*, abs/1703.01513, 2017.2

[25] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V. Le, and Alexey Kurakin.图像分类器的大规模演变。In Doina Pre- cup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2902-2911.
PMLR, 2017.2

[26] Barret Zoph and Quoc V. Le.带有强化学习的神经架构搜索.*CoRR*, abs/1611.01578, 2016.2

[27] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets：用于移动视觉应用的高效卷积神经网络-作品。*CoRR*, abs/1704.04861, 2017.2, 4, 5, 6

[28] Francois Chollet.Xception：带有深度可分离卷积的深度学习。*在IEEE计算机视觉和模式识别会议（CVPR）上*，2017年7月。2

[29] Dongyoon Han, Jiwhan Kim, and Junmo Kim.Deep pyramidal residual networks.*CoRR*, abs/1610.02915, 2016.3

[30] Saining Xie, Ross B. Girshick, Piotr Dolla´r, Zhuowen Tu, and Kaiming He.深度神经网络的聚合残差转换。
CoRR, abs/1611.05431, 2016.3, 4, 8

[31] Mart´ın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S．Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfel- low, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane´, Rajat Monga, Sherry Moore、 Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vie´gas, Oriol Vinyals, Pete Warden, Martin Wat- tenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng.TensorFlow：异构系统上的大规模机器学习，2015。软件可从tensorflow.org获得。5, 6

[32] 贾阳青、 Evan Shelhamer、 Jeff Donahue、 Sergey Karayev、Jonathan Long、Ross Girshick、Sergio Guadarrama 和 Trevor Darrell。Caffe：用于快速特征嵌入的卷积架构。*arXiv预印本arXiv:1408.5093*，2014。5

[33] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. 现代卷积物体检测器的速度/准确度权衡。In *CVPR*, 2017.7

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg.Ssd：单枪多盒检测器。In *ECCV*, 2016.7

[35] Joseph Redmon 和 Ali Farhadi.Yolo9000：更好、更快、更强。 *arXiv 预印本 arXiv:1612.08242*，2016。7

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian 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

[37] 戴继锋，李毅，何开明，孙健.R- fcn：通过基于区域的完全卷积网络进行物体检测。In *Advances in neural infor-mation processing systems*, pages 379-387, 2016.
7

[38] Jonathan Huang, Vivek Rathod, Derek Chow, Chen Sun, and Menglong Zhu.Tensorflow物体检测api, 2017.7

[39] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam.重新思考用于语义图像分割的反转卷积。
CoRR, abs/1706.05587, 2017.7

[40] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian.在小波变换的帮助下进行信号分析的一种实时算法。In *Wavelets：时间-频率方法和相位空间*，第289-297页。1989.7

[41] Pierre Sermanet, David Eigen, Xiang Zhang, Michae¨l Mathieu, Rob Fergus, and Yann Le-Cun.*ArXiv:1312.6229*, 2013.7

[42] George Papandreou, Iasonas Kokkinos, and Pierre-Andre Savalle.深度学习中的局部和全局deformations建模：表观卷积、多实例学习和滑动窗口检测。In *CVPR*, 2015.7

[43] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille.Deeplab：用深度卷积网、反转卷积和完全连接的crfs进行语义图像分割。*TPAMI*, 2017.7

[44] Wei Liu, Andrew Rabinovich, and Alexander C. Berg.Parsenet：看得更广，看得更清楚。*CoRR*, abs/1506.04579, 2015.7

[45] Bharath Hariharan, Pablo Arbela´ez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik.反向检测器的语义轮廓。在*ICCV*，2011年。8

[46] Christian Szegedy, Sergey Ioffe, and Vincent Van- houcke.Inception-v4, inception-resnet and the im- pact of residual connections on learning.*CoRR*, abs/1602.07261, 2016.8

[47] Guido Montu´far, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio.论深度神经网络的线性区域数量。在*第27届神经信息处理系统国际会议论文集*，NIPS'14，第2924-2932页，美国马萨诸塞州剑桥，2014。麻省理工学院出版社。13

[48] Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Le´on Bottou, and Kilian Q. Weinberger, editors. *神经信息处理系统的进展25：2012年第26届神经信息处理系统年度会议。2012年12月3-6日在美国内华达州塔霍湖举行的会议记录*，2012。
9

A. 瓶颈转化

在本节中，我们研究一个算子A ReLU(Bx)的特性，其中$x\in R^n$ 代表一个n*通道*像素，B是一个$m\times n$矩阵，A是一个$n\times m$矩阵。我们认为，如果$m\leq n$，这种形式的变换只能以损失信息为代价利用非线性。

矩阵。相反，如果n m，这样的变换可以是高度非线性的，但仍然可以以高概率倒置（对于初始随机权重）。

首先，我们表明ReLU对于任何位于其图像内部的点来说都是一种身份转换。

定理1 *让$S(X)$={ReLU(x)|$x\in X$}。如果$S(X)$的一个卷是非零的，那么*内部$S(X)\subseteq X$。

证明： 让S' = 内部ReLU(S)。首先我们注意到，如果$x\in S'$，那么$x_i > 0$，对于所有i。事实上，ReLU的图像不包含负坐标的点，并且

具有零值坐标的点不可能是内部点。因此对于每个$x\in S'$，x=ReLU(x)，如愿以偿。□

由此可见，对于一个任意的叶间线性变换和ReLU算子的组合，如果它保留了非零体积，那么输入空间的那部分

在这样的组合中保留的X是一个线耳变换，因此可能是深度网络力量的一个小贡献者。然而，这

是一个相当弱的声明。事实上，如果输入流形可以被嵌入到（n - 1）维流形中（共n个维度），那么该定理就基本成立，因为起始体积为0。在接下来的内容中，我们表明，当输入流形的维度明显降低时，我们可以确保不会有信息损失。

由于ReLU(x)非线性是一个将整个射线$x \leq 0$映射到0的射影函数，在神经网络中使用这种非线性会导致信息损失。一旦ReLU将输入流形的一个子集折叠成一个较小维度的输出，下面的网络布局------。

编码器不能再区分折叠的输入样本了。在下文中，我们表明，具有足够大的扩展层的瓶颈可以抵抗由ReLU激活 功能的存在所造成的信息损失。

定理2 (ReLU的可逆性) *考虑一个操作数*ReLU(Bx)，*其中B是一个$m \times n$矩阵，*$x \in \mathbb{R}^n$。*让*y_0 = ReLU(Bx_0)，*对于某些*$x_0 \in \mathbb{R}^n$，*那么方程*y_0 = ReLU(Bx)*对于x有唯一的解，当且仅当y_0至少有n个非零值，且B有n个线性独立的行对应于y_0的非零坐标。*

证明： 将y的非零坐标集$_0$为T，让y_T和B_T是y和B对T所定义的子空间的限制。如果 $|T| < n$，我们有

$y_T = B_T x_0$，其中B_T是欠确定的，至少有一个解x_0，因此有无限多的so-lutions。现在考虑$|T| \geq n$的情况，让

B_T的等级为 n。假设有一个额外的解 $x_1 \neq x_0$，使得 y_0 = ReLU(Bx_1)，那么我们有 $y_T = B_T x_0 = B_T x_1$，这不能满足，除非 $x_0 = x_1$。 □

这条定理的一个推论是，如果m n，我们只需要Bx的一小部分值是正的，ReLU(Bx)就可以反转。

定理2的约束条件可以通过经验来实现

我们对真实的网络和真实的输入进行了验证，因此我们可以保证信息确实得到了保留。我们进一步表明，就初始化而言，我们可以确保这些约束条件以高概率得到满足。请注意，对于随机初始化，由于初始化的对称性，秤2的条件得到满足。然而，即使是训练有素的图，这些约束也可以通过在有效输入上运行网络并验证所有或大多数输入都在阈值之上来进行经验验证。在图7中，我们展示了不同的MobileNetV2层的这种分布情况。在第0步激活模式集中在有一半的正通道（正如初始化症状所预测的那样）。对于经过充分训练的网络，虽然标准偏差明显增长，但除了两层之外，所有的网络仍然高于可逆性阈值。我们认为对此进行进一步的研究是有必要的，并可能导致对网络设计的有益见解。

定理1 *假设S是$\mathbb{R}^n$的一个紧凑的n维子man- ifold。考虑从$\mathbb{R}^n$到$\mathbb{R}^m$的一族函数f_B（x）= ReLU（Bx），参数为$m \times n$矩阵$B \in \mathbb{B}$。让p（B）是所有矩阵$\mathbb{B}$空间上的一个概率密度，满足：*

- *对于任何措施为零的子集$Z \subset \mathbb{B}$，P（Z）=0；*
- *(对称条件)对于任何$B \in \mathbb{B}$和任何$m \times m$对角线矩阵D，所有对角线元素不是+1就是-1，$p(DB)=p(B)$。*

那么，被f_B塌陷为低维流形的S的子集的平均n体积为

$$V - \frac{N_{m,n} V}{2^m},$$

其中V=体积S和

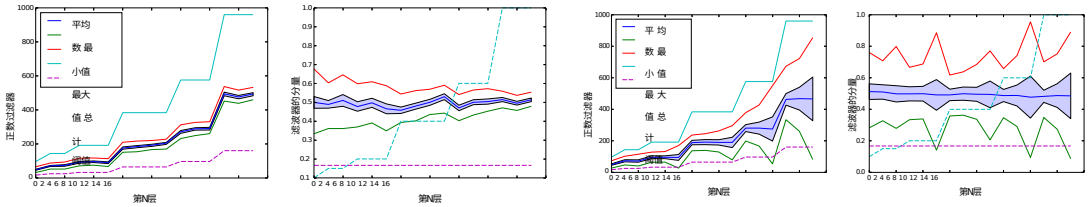
$$N_{m,n} \equiv \sum_{k=0}^{m} \binom{m}{k}.$$

证明： 对于任何$\sigma = (s_1, \ldots, s_m)$，其中$s \in \{-1, +1\}$，让$Q_\sigma = \{x \in \mathbb{R}^m \mid x\, s_i > 0\}$是$\mathbb{R}^m$中的一个对应象限。对于任何n维子流形$\Gamma \subset \mathbb{R}^m$，如果$\sigma$至少有$n$个正值，ReLU对$\Gamma \cap Q_\sigma$起到双射作用。[4] 否则就收缩 $\Gamma \cap Q_\sigma$。同时注意到，BS与$\mathbb{R}^m \setminus (\cup Q_{\sigma\sigma})$的交集几乎肯定是$(n - 1)$-维的。因此，将ReLU应用于$BS$而不被折叠的$S$的平均n*体积由*以下公式给出：

$$\sum_{\sigma \in \Sigma_n} E_B [V_\sigma (B)], \tag{3}$$

其中$\Sigma_n = \{(s_1, \ldots, s_m) \mid \sum_k \vartheta(s_k) \geq n\}$，$\vartheta$是一个阶梯函数，$V_\sigma (B)$是由$B$映射到$Q_\sigma$的$S$的最大子集的体积。现在让我们计算$E_B [V_\sigma (B)]$。回顾$p(DB)=p(B)$，对于任何$D$=diag($s_1, \ldots, s_m$)，$s_k \in \{-1, +1\}$，这个平均值可以改写为$E_{BD}\, E [V_\sigma (DB)]$。注意到由$DB$映射到$Q_\sigma$的$S$的子集也被$B$映射到$D\, Q^{-1}_\sigma$，我们立即得到

[4] 除非在所有$x \in \Gamma \cap Q_\sigma$的正坐标中至少有一个正坐标是固定的，对于几乎所有的B和$\Gamma = BS$来说，情况就不是这样。



(a) 在第0步　　(b) 全面培训

图7：激活模式的分布。x*轴*是层指数，我们显示了每次与ReLU卷积后的最小√/最大/平均的正通道数。*y轴*是通道的绝对数或相对数。阈值 "线表示ReLU的可逆性阈值--即阳性维度的数量要高于输入空间。在我们的例子中，这是*1/*6个通道的分数。请注意，在图7a的训练开始时，分布是如何更紧密地集中在平均值周围的。训练结束后
(图7b），平均数没有变化，但标准差急剧增长。最好以彩色观看。

$\Sigma_{\sigma'} V_\sigma$ [diag(σ')B] = $\Sigma_{\sigma'} V_\sigma$ ' [B] = vol S，因此，E_B [V_S(B)] = 2^{-m} vol S。将此和 $|\Sigma_n| = \sum_{k=0}^{nm} \binom{nm}{k}$ 进入公式3，证明完毕。□

请注意，对于有m n*的*足够大的膨胀层，塌陷空间的部分$N_{m,n}$ /2^m ，可以通过以下方式进行约束：

$$\frac{\text{谷米}n}{2^m} \geq 1 - \frac{mn^+}{2mn!} \geq 1 - 2^{(n+1)\log m - m} \geq 1 - 2^{-m/2}$$

因此，ReLU(*Bx*)进行了非线性转换，同时以较高的概率保存了信息。

我们讨论了瓶颈如何防止流形塌陷，但是增加瓶颈扩展的大小也可能使网络有可能代表更复杂的函数。按照[47]的主要再研究结果，我们可以证明，例如，对于任何整数L≥1和p>1，存在一个L ReLU的网络
层，每个层包含n*个*神经元和一个瓶颈前
大小*为*pn的pansion，使其将p^{nL} 输入量（线性同构为[0, 1]^n ）映射到相同的输出re gion [0, 1]^n 。因此，任何连接到网络输出的复杂的可能的非线性函数将有效地
计算p^{nL} 输入线性区域的函数值。

B. 语义分割可视化的结果

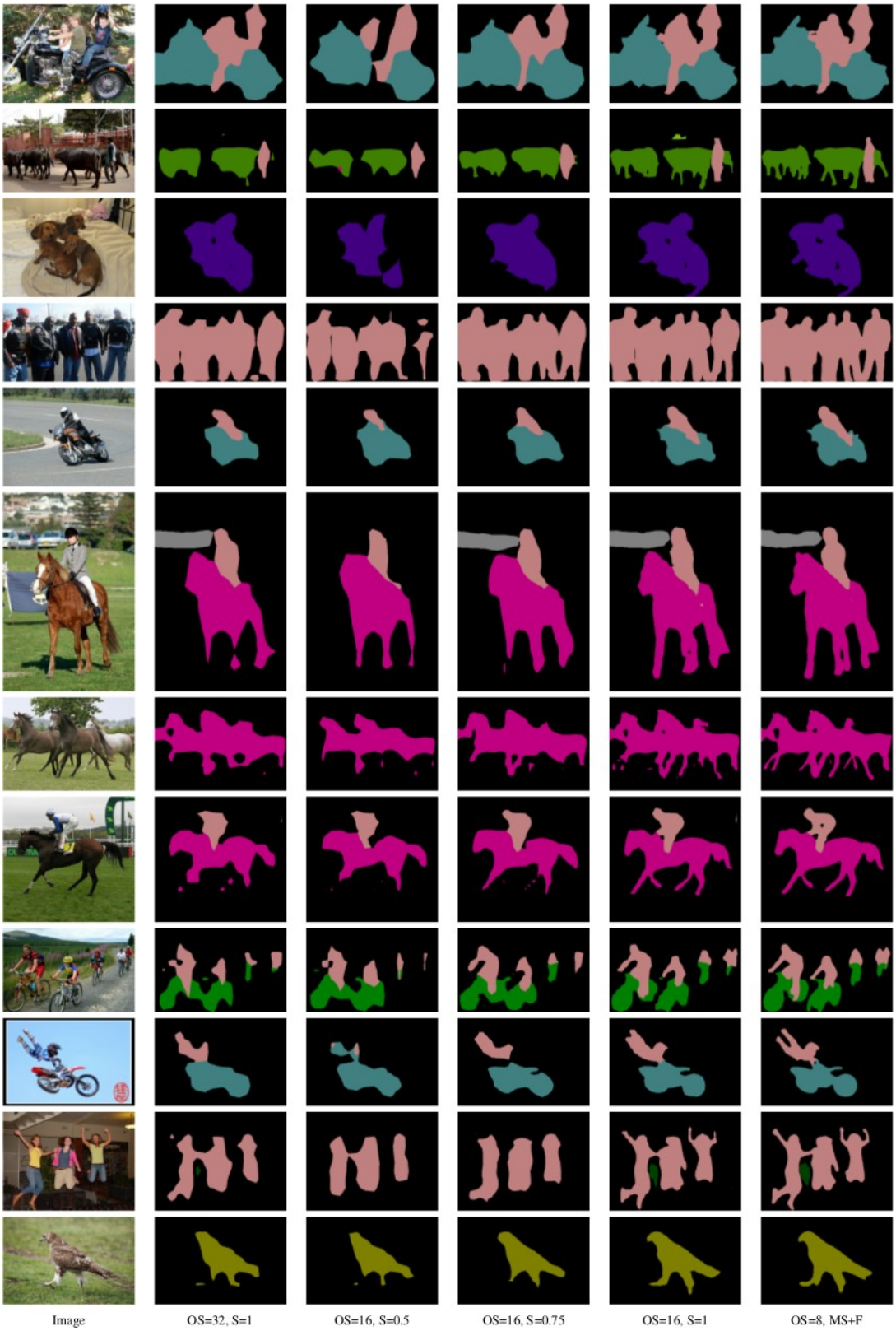


图8：MobileNetv2在PASCAL VOC 2012*估值*集上的语义分割可视化结果。**OS**：*输出跨度*。**S**：单尺度输入。**MS+F**：多尺度输入，尺度={0.5，0.75，1，1.25，1.5，1.75}和左右翻转的输入。采用*输出跨度*=16和单一输入比例=1，在FLOPS和精度之间取得了良好的权衡。