Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich Google
Inc.

摘要

我们在 ImageNet 大规模视觉识别挑战赛 2014（ILSVRC14）上提出了一种代号为 Inception 的深度卷积神经网络结构，并在分类和检测上取得了新的最好结果。这个架构的主要特点是提高了网络内部计算资源的利用率。通过精心的手工设计，我们在增加了网络深度和广度的同时保持了计算预算不变。为了优化质量，架构的设计以赫布理论和多尺度处理直觉为基础。我们在 ILSVRC14 提交中应用的一个特例被称为 GoogLeNet，一个 22 层的深度网络，其质量在分类和检测的背景下进行了评估。

1 引言

过去三年中，由于深度学习和卷积网络的发展[10]，图像识别和目标检测的能力一直在以惊人的速度进步。一个令人鼓舞的消息是，大部分的进步不仅仅是更强大硬件、更大数据集、更大模型的结果，而主要是新的想法、算法和网络结构改进的结果。例如，ILSVRC 2014 竞赛中最靠前的输入除了用于检测目的的分类数据集之外，没有使用新的数据资源。我们在 ILSVRC 2014 中的 GoogLeNet 提交实际使用的参数只有两年前 Krizhevsky 等人[9]获胜结构参数的 1/12，而结果明显更准确。在目标检测前沿，最大的收获不是来自于越来越大的深度网络的简单应用，而是来自于深度架构和经典计算机视觉的协同，像 Girshick 等人[6]的 R-CNN 算法那样。

另一个值得注意的因素是，随着移动计算和嵌入式计算的持续发展，我们的算法的效率—特别是它们的能力和内存使用—变得越来越重要。值得注意的是，正是包含了这个因素的考虑才得出了本文中呈现的深度架构设计，而不是单纯的为了提高准确率。对于大多数实验来说，模型被设计为在一次推断中保持 15 亿乘加的计算预算，所以最终它们不是单纯的学术好奇心，而是能在现实世界中应用，甚至是以合理的代价在大型数据集上使用。

在本文中，我们将关注一个高效的计算机视觉深度神经网络架构，代号为 Inception，它的名字来自于 Lin 等人[12]网络论文中的 Network 与著名的“we need to go deeper”网络迷因[1]的结合。在我们的案例中，单词“deep”用在两个不同的含义中：首先，在某种意义上，我们以“Inception module”的形式引入了一种新层次的组织方式，在更直接的意义上增加了网络的深度。一般来说，可以把 Inception 模型看作论文[12]的逻辑顶点同时从 Arora 等人[2]的理论工作中受到了鼓舞和引导。这种架构的好处在 ILSVRC 2014 分类和检测挑战赛中通过实验得到了验证，它明显优于目前的最好水平。

2 近期工作

从 LeNet-5 [10]开始，卷积神经网络（CNN）通常有一个标准结构——堆叠的卷积层（后面可以选择有对比归一化和最大池化）后面是一个或更多的全连接层。这个基本设计的变种在图像分类著作流行，并且目前为止在 MNIST，CIFAR 和更著名的 ImageNet 分类挑战赛中[9, 21]的已经取得了最佳结果。对于更大的数据集例如 ImageNet 来说，最近的趋势是增加层的数目[12]和层的大小[21, 14]，同时使用丢弃[7]来解决过拟合问题。

尽管担心最大池化层会引起准确空间信息的损失，但与[9]相同的卷积网络结构也已经成功的应用于定位[9, 14]，目标检测[6, 14, 18, 5]和行人姿态估计[19]。受灵长类视觉皮层神

经科学模型的启发，Serre 等人[15]使用了一系列固定的不同大小的 Gabor 滤波器来处理多尺度。我们使用一个了类似的策略。然而，与[15]的固定的 2 层深度模型相反，Inception 结构中所有的滤波器是学习到的。此外，Inception 层重复了很多次，在 GoogleNet 模型中得到了一个 22 层的深度模型。

Network-in-Network 是 Lin 等人[12]为了增加神经网络表现能力而提出的一种方法。当应用于卷积层时，该方法可以看作是额外的 1×1 个卷积层，随后通常是 ReLU[9]。这使得它能够很容易地集成到当前的 CNN pipelines 中。我们的架构中大量的使用了这个方法。但是，在我们的设置中，1 × 1 卷积有两个目的：最关键的是，它们主要是用来作为降维模块来移除卷积瓶颈，否则将会限制我们网络的大小。这不仅允许了深度的增加，而且允许我们网络的宽度增加但没有明显的性能损失。

最后，目前最好的目标检测是 Girshick 等人[6]的基于区域的卷积神经网络（R-CNN）方法。R-CNN 将整个检测问题分解为两个子问题：利用低层次的信号例如颜色，纹理以跨类别的方式来产生目标位置候选区域，然后用 CNN 分类器来识别那些位置上的对象类别。这样一种两个阶段的方法利用了低层特征分割边界框的准确性，也利用了目前的 CNN 非常强大的分类能力。我们在我们的检测提交中采用了类似的方式，但是在两个阶段都进行了更强的探索，例如使用多框预测（multi-box prediction）[5]来提高对象边界框的召回率，以及使用集成方法来更好地对边界框提议进行分类。

3 动机和高层思考

提高深度神经网络性能最直接的方式是增加它们的尺寸。这不仅包括增加深度—网络层次的数目—也包括它的宽度：每一层的单元数目。这是一种训练更高质量模型容易且安全的方法，尤其是在可获得大量标注的训练数据的情况下。但是这个简单方案有两个主要的缺点。

更大的尺寸通常意味着更多的参数，这会使增大的网络更容易过拟合，尤其是在训练集的标注样本有限的情况下。这是一个主要的瓶颈，因为要获得强标注数据集费时费力且代价昂贵，经常需要专家评委在各种细粒度的视觉类别进行区分，例如图 1 中显示的 ImageNet 中的类别（甚至是 1000 类 ILSVRC 的子集）。

(a)西伯利亚哈士奇(Siberian husky)　　(b)爱斯基摩狗(Eskimo dog)

图 1：来自 2014 年 ILSVRC 分类挑战的 1000 类中的两个不同的类。

统一增加网络尺寸的另一个缺点是计算资源使用的显著增加。例如，在一个深度视觉网络中，如果两个卷积层相连，它们的滤波器数目的任何统一增加都会引起计算量平方式的增加。如果增加的能力使用时效率低下（例如，如果大多数权重结束时接近于 0），那么会浪费大量的计算能力。由于在实践中，计算预算总是有限的，因此即使其主要目标是提高结果的质量，也会首选有效地分配计算资源，而不是不加选择地增加计算资源的大小。

解决这两个问题的根本方法是最终从全连接架构转向稀疏连接架构，甚至在卷积层内部也是如此。除了模仿生物系统之外，由于 Arora 等人[2]的开创性工作，这也具有更坚固的理论基础优势。他们的主要成果说明如果数据集的概率分布可以通过一个大型稀疏的深度神经网络表示，则最优的网络拓扑结构可以通过分析前一层激活的相关性统计和聚类高度相关的神经元来一层层的构建。虽然严格的数学证明需要在很强的条件下，但事实上这个声明与著

名的赫布理论产生共鸣—神经元一起激发，一起连接—实践表明，即使在不那么严格的条件下，其基本思想也很适用。

遗憾的是，当碰到在非均匀的稀疏数据结构上进行数值计算时，现在的计算架构效率非常低下。即使算法运算的数量减少 100 倍，查询和缓存丢失上的开销仍占主导地位：切换到稀疏矩阵可能是不可行的。通过使用稳步改进、高度调优的数字库，允许极其快速的密集矩阵乘法，利用底层 CPU 或 GPU 硬件[16,9]的微小细节，进一步扩大了差距。此外，非均匀的稀疏模型也要求更多的复杂工程和计算基础结构。目前大多数面向视觉的机器学习系统通过采用卷积的优点来利用空域的稀疏性。然而，卷积被实现为对上一层块的密集连接的集合。为了打破对称性，提高学习水平，从论文[11]开始，ConvNets 习惯上在特征维度使用随机的稀疏连接表，然而为了打破对称性和提高学习能力，为了更好地优化并行计算，趋势重新转向与[9]的完全连接。目前最新的计算机视觉架构有统一的结构。更多的滤波器和更大的批大小要求密集计算的有效使用。

这提出了下一个中间步骤是否有希望的问题：一个架构能利用滤波器水平的稀疏性，正如理论所建议的那样，但能通过利用密集矩阵计算来利用我们目前的硬件。稀疏矩阵乘法的大量文献（例如[3]）认为对于稀疏矩阵乘法，将稀疏矩阵聚类为相对密集的子矩阵会有更佳的性能。在不久的将来会利用类似的方法来进行非均匀深度学习架构的自动构建，这样的想法似乎并不牵强。

Inception 架构开始是作为案例研究，用于评估一个复杂网络拓扑构建算法的假设输出，该算法试图近似[2]中所示的视觉网络的稀疏结构，并通过密集的、容易获得的组件来覆盖假设结果。尽管这是一个高度投机的尝试，在对拓扑结构进行两次迭代后，我们已经可以看到与基于[12]的参考架构相比的适度收益。经过进一步调整学习率、超参数和改进训练方法，我们确定了所得到的 Inception 架构在定位和物体检测的背景下作为[6]和[5]的基础网络特别有用。有趣的是，尽管大部分原始的架构选择都受到了质疑和充分测试，但它们最终被证明至少在局部上是最优的。

然而，我们必须谨慎：尽管这种提出的架构对于计算机视觉已经取得了成功，但仍然有疑问是否可以将其质量归因于导致其构建的指导原则。要确保这一点，需要进行更加彻底的分析和验证：例如，基于下面描述的原则的自动化工具是否能够找到类似但更好的视觉网络拓扑结构。最有说服力的证据是，如果一个自动化系统能够使用相同的算法但具有非常不同的全局架构，在其他领域中创建出类似的网络拓扑结构并获得类似的收益。至少，Inception 架构的初步成功为在这个方向上进行激动人心的未来工作提供了坚实的动力。

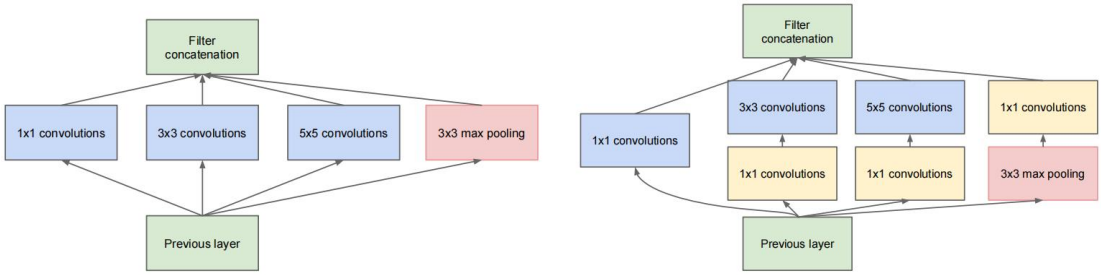
4 架构细节

Inception 架构的主要想法是考虑怎样近似卷积视觉网络的最优稀疏结构并用容易获得的密集组件进行覆盖。注意假设转换不变性，这意味着我们的网络将以卷积构建块为基础。我们所需要做的是找到最优的局部构造并在空间上重复它。Arora 等人[2]提出了一个层次结构，其中应该分析最后一层的相关统计并将它们聚集成具有高相关性的单元组。这些聚类形成了下一层的单元并与前一层的单元连接。我们假设较早层的每个单元都对应输入层的某些区域，并且这些单元被分成滤波器组。在较低的层（接近输入的层）相关单元集中在局部区域。因此，如[12]所示，我们最终会有许多聚类集中在单个区域，它们可以通过下一层的 1×1 卷积层覆盖。然而也可以预期，将存在更小数目的在更大空间上扩展的聚类，其可以被更大块上的卷积覆盖，在越来越大的区域上块的数量将会下降。为了避免块校正的问题，目前 Inception 架构形式的滤波器的尺寸仅限于 1×1、3×3、5×5，然而这个决定更多的是基于便易性而不是必要性。这也意味着提出的架构是所有这些层的组合，其输出滤波器组连接成单个输出向量形成了下一阶段的输入。另外，由于池化操作对于目前卷积网络的成功至关重要，因此建议在每个这样的阶段添加一个替代的并行池化路径应该也应该具有额外的有益效果（看图 2(a)）。

由于这些“Inception 模块”在彼此的顶部堆叠，其输出相关统计必然有变化：由于较高层会捕获较高的抽象特征，其空间集中度预计会减少。这表明随着转移到更高层，3×3 和 5×5 卷积的比例应该会增加。

上述模块的一个大问题是在具有大量滤波器的卷积层之上，即使适量的 5×5 卷积也可能是非常昂贵的，至少在这种朴素形式中有这个问题。一旦池化单元添加到混合中，这个问

题甚至会变得更明显：输出滤波器的数量等于前一阶段滤波器的数量。池化层输出和卷积层输出的合并会导致这一阶段到下一阶段输出数量不可避免的增加。虽然这种架构可能会覆盖最优稀疏结构，但它会非常低效，导致在几个阶段内计算量爆炸。



(a)Inception 模块，naive 版本　　(b)降维后的 Inception 模块

图 2:Inception 模块

这引出了提出架构的第二个思想：在计算要求会增加太多的地方，果断地降维和投影。这是基于嵌入的成功：甚至低维嵌入可能包含大量关于较大图像块的信息。然而嵌入以密集、压缩形式表示信息并且压缩信息更难处理。这种表示应该在大多数地方保持稀疏（根据[2]中条件的要求】）并且仅在它们必须汇总时才压缩信号。也就是说，在昂贵的 3×3 和 5×5 卷积之前，1×1 卷积用来计算降维。除了用来降维之外，它们也包括使用 ReLU 使其两用。最终的结果如图 2(b)所示。

通常，Inception 网络是一个由上述类型的模块互相堆叠组成的网络，偶尔会有步长为 2 的最大池化层将网络分辨率减半。出于技术原因（训练过程中内存效率），只在更高层开始使用 Inception 模块而在更低层仍保持传统的卷积形式似乎是有益的。这不是绝对必要的，只是反映了我们目前实现中的一些基础结构效率低下。

这种架构的一个主要好处是，它允许在每个阶段显著增加单元的数量，而不会导致计算复杂度无法控制地急剧增加。广泛使用维度缩减使得可以将上一阶段的大量输入滤波器屏蔽到下一层，首先在卷积之前减小它们的维度，然后使用尺寸较大的块进行卷积。此外，设计遵循了实践直觉，即视觉信息应该在不同的尺度上处理然后聚合，为的是下一阶段可以从不同尺度同时提取特征。。

计算资源的改善使用允许增加每个阶段的宽度和阶段的数量，而不会陷入计算困境。可以利用 Inception 架构创建性能略差但计算成本更低的版本。我们发现，所有包括的参数和选项允许对计算资源进行可控的平衡，从而使得具有非 Inception 架构的网络比同样性能的网络快 2-3 倍，但这需要仔细的手动设计。

5 GoogLeNet

我们在 ILSVRC14 比赛中选择了 GoogLeNet 作为团队名称。这个名称是对 Yann LeCun 开创性的 LeNet 5 网络的致敬[10]。我们还使用 GoogLeNet 来指代我们在比赛中提交的 Inception 架构的特定版本。我们还使用了一个更深更宽的 Inception 网络，其质量略有下降，但将其添加到集成模型中似乎稍微改善了结果。我们省略了该网络的细节，因为我们的实验证明了确切的架构参数的影响相对较小。为了演示的目的，表 1 描述了最成功的特定实例（命名为 GoogLeNet）。我们的集成模型中有 7 个模型中的 6 个使用了完全相同的拓扑结构（使用不同的采样方法进行训练）。

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	Pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M

inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

表 1:Inception 架构的 GoogLeNet 版本

所有的卷积都使用了 ReLU，包括 Inception 模块内部的卷积。我们网络中的感受野大小为 224×224，采用 RGB 颜色通道并进行均值减法处理。“#3×3 reduce”和“#5×5 reduce”表示在 3×3 和 5×5 卷积之前，降维层使用的 1×1 滤波器的数量。在 pool proj 列中可以看到内置最大池化后投影层中 1×1 滤波器的数量。所有这些降维/投影层也使用修正线性激活函数。

由于在网络的设计过程中考虑了计算效率和实用性，因此推理可以在单独的设备上运行，甚至包括那些计算资源有限的设备，尤其是低内存占用的设备。当只计算有参数的层时，网络有 22 层（如果我们也计算池化层是 27 层）。构建网络的全部层（独立构建块）的数目大约是 100。确切的数量取决于机器学习基础设施对层的计算方式。分类器之前的平均池化是基于[12]的，尽管我们的实现有一个额外的线性层。这使得我们可以轻松地适应和微调网络以适用于其他标签集，但这主要是为了方便起见，我们并不指望它会产生重大影响。实验证明，从全连接层转换为平均池化层可以将 top-1 准确率提高约 0.6%，然而，即使在去除全连接层后，使用 dropout 仍然是必不可少的。

由于网络的深度相对较大，因此以有效的方式将梯度传播回所有层的能力是一个值得关注的问题。有趣的发现是，相对较浅的网络在这个任务上表现出色，这表明网络中间层产生的特征应该具有很强的区分能力。通过将辅助分类器添加到这些中间层，我们希望在分类器的较低阶段鼓励区分，并增加被传播回来的梯度信号，并提供额外的正则化。这些分类器采用较小的卷积网络形式，放置在 Inception（4a）和（4d）模块的输出之上。在训练期间，它们的损失以折扣权重（辅助分类器损失的权重是 0.3）加到网络的整个损失上。在推理期间，这些辅助网络被丢弃。

包括辅助分类器在内的附加网络的具体结构如下：

- 使用 5×5 的滤波器大小和步长为 3 的平均池化层，对(4a)阶段的输入进行处理，得到 4×4×512 的输出；对(4d)阶段的输入进行处理，得到 4×4×528 的输出。
- 一个 1×1 的卷积层，使用 128 个滤波器进行降维，并使用 ReLU。
- 一个具有 1024 个单元的全连接层，并使用修正线性激活函数。
- 一个 dropout 层，丢弃输出的比率为 70%。
- 一个线性层，使用 softmax 损失作为分类器（预测与主分类器相同的 1000 个类别，但在推断时移除）。

最终的网络模型图如图 3 所示。

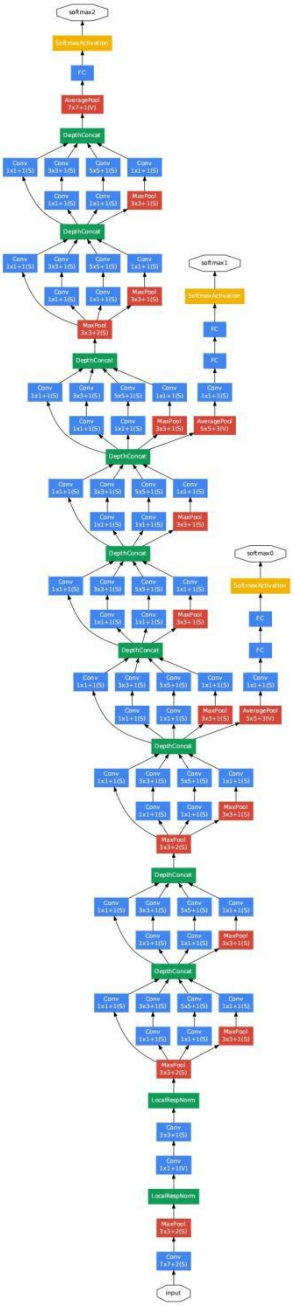


图 3：使用了所有先进技术和创新的 GoogLeNet 网络。

6 训练方法

GoogLeNet 网络使用 DistBelief[4]分布式机器学习系统进行训练，该系统使用适量的模型和数据并行。尽管我们只使用基于 CPU 的实现，但粗略的估计表明 GoogLeNet 可以用少量的高端 GPU 在一周之内训练到收敛，主要的限制是内存使用。我们的训练使用异步随机梯度下降，动量参数为 0.9[17]，固定的学习率计划（每 8 次遍历下降学习率 4%）。Polyak 平均[13]在推断时用来创建最终的模型。

在领先比赛的几个月中，我们的图像采样方法发生了很大变化，并且已经收敛的模型也经过了其他选项的训练，有时还与改变的超参数（如丢弃率和学习率）结合使用，因此很难给出对这些网络训练最有效的单一方法的明确指导。更复杂的是，一些模型主要是在相对较小的裁剪图像上训练的，而其他模型则是在较大的裁剪图像上训练的，受到[8]的启发。然而，在比赛之后经过验证非常有效的一种方法是对图像进行各种尺寸的图像块采样，其大小均匀分布在图像面积的 8%到 100%之间，其宽高比在 3/4 和 4/3 之间随机选择。此外，我们发现 Andrew Howard [8]的光度畸变在一定程度上对抗过拟合是有用的。此外，我们开始相

对较晚地使用随机插值方法（双线性插值、区域插值、最近邻插值和三次插值，概率相等）进行调整，并与其他超参数的更改同时使用，因此我们无法确定最终结果是否受到它们的积极影响。

7 ILSVRC 2014 分类挑战赛设置和结果

ILSVRC 2014 分类挑战赛包括将图像分类到 ImageNet 层级中 1000 个叶子结点类别的任务。训练图像大约有 120 万张，验证图像有 5 万张，测试图像有 10 万张。每一张图像与一个实际类别相关联，性能度量基于分类器预测的最高分。通常报告两个数字：top-1 准确率，比较实际类别和第一个预测类别，top-5 错误率，比较实际类别与前 5 个预测类别：如果图像实际类别在 top-5 中，则认为图像分类正确，不管它在 top-5 中的排名。挑战赛使用 top-5 错误率来进行排名。

我们参加竞赛时没有使用外部数据来训练。除了本文中前面提到的训练技术之外，我们在获得更高性能的测试中采用了一系列技巧，描述如下。

1. 我们独立训练了 7 个版本的相同的 GoogLeNet 模型（包括一个更广泛的版本），并用它们进行了整体预测。这些模型的训练具有相同的初始化（甚至具有相同的初始权重，主要是因为一个疏忽）和学习率策略。它们仅在采样方法和随机输入图像顺序方面不同。

2. 在测试阶段，我们采用比 Krizhevsky 等人[9]更积极的裁剪方法。具体来说，我们将图像调整为四个尺度，其中较短维度（高度或宽度）分别为 256，288，320 和 352，取这些调整后图像的左，中，右方块（在纵向图片中，我们采用顶部，中心和底部方块）。对于每个方块，我们将采用 4 个角以及中心 224×224 裁剪图像以及调整为为 224×224 的方块，还有它们的镜像版本。这导致每张图像会得到 4×3×6×2 = 144 的裁剪图像。前一年的输入中，Andrew Howard[8]采用了类似的方法，经过我们实证验证，其方法略差于我们提出的方案。我们注意到，在实际应用中，这种积极裁剪可能是不必要的，因为存在合理数量的裁剪图像后，进一步增加裁剪的效益会逐渐减小（我们将在后面展示）。

3. softmax 概率在多个裁剪图像上和所有单个分类器上进行平均，然后获得最终预测。在我们的实验中，我们在验证数据上分析了替代方法，比如对裁剪进行最大池化和对分类器进行平均，但它们的性能不如简单的平均方法。

在本文的其余部分，我们分析了有助于最终提交整体性能的多个因素。

竞赛中我们的最终提交在验证集和测试集上得到了 top-5 6.67%的错误率，在其它的参与者中排名第一。与 2012 年的 SuperVision 方法相比相对减少了 56.5%，与前一年的最佳方法（Clarifai）相比相对减少了约 40%，这两种方法都使用了外部数据训练分类器。下表显示了一些性能最好的方法的统计数据。

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

表 2：分类性能

我们还分析和报告了在以下表格中改变模型数量和预测图像时使用的裁剪数量时的多个测试选择的性能。当我们使用一个模型时，我们选择在验证数据上具有最低 top-1 错误率的模型。为了不过度拟合测试数据的统计数据，所有数字都是在验证数据集上报告的。

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

表 3：GoogLeNet 分类性能详细分析

8 ILSVRC 2014 检测挑战赛设置和结果

ILSVRC 检测任务是为了在 200 个可能的类别中生成图像中目标的边界框。如果检测到的对象匹配的它们实际类别并且它们的边界框重叠至少 50%（使用 Jaccard 索引），则将检测到的对象记为正确。无关的检测记为假阳性且被惩罚。与分类任务相反，每张图像可能包含多个对象或没有对象，并且它们的尺度可能是变化的。报告的结果使用平均精度均值（mean average precision,mAP）。

GoogLeNet 检测采用的方法类似于 R-CNN[6]，但用 Inception 模块作为区域分类器进行了增强。此外，为了更高的目标边界框召回率，通过选择搜索[20]方法和多框[5]预测相结合改进了区域生成步骤。为了减少假阳性的数量，超分辨率的尺寸增加了 2 倍。这将选择搜索算法的区域生成减少了一半。我们总共补充了 200 个来自多框结果的区域生成，大约 60%的区域生成用于[6]，同时将覆盖率从 92%提高到 93%。减少区域生成的数量，增加覆盖率的整体影响是对于单个模型的情况平均精度均值增加了 1%。最后，在对每个区域进行分类时，我们使用了 6 个 ConvNets 的集合，将结果的准确率从 40%提高到 43.9%。请注意，与 R-CNN 相反，由于缺乏时间，我们没有使用边界框回归。

我们首先报告了最好检测结果，并显示了从第一版检测任务以来的进展。与 2013 年的结果相比，准确率几乎翻了一倍。所有表现最好的团队都使用了卷积网络。我们在表 4 中报告了官方的分数和每个队伍的常见策略：使用外部数据、集成模型或上下文模型。外部数据通常是 ILSVRC12 的分类数据，用来预训练模型，后面在检测数据集上进行改善。一些团队也提到使用定位数据。由于定位任务的边界框很大一部分不在检测数据集中，所以可以用该数据预训练一般的边界框回归器，这与分类预训练的方式相同。GoogLeNet 输入没有使用定位数据进行预训练。

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

表 4：检测性能

在表 5 中，我们仅比较了单个模型的结果。最好性能模型是 Deep Insight 提供的，令人惊讶的是 3 个模型的集合仅提高了 0.3 个点，而 GoogLeNet 在模型集成时明显获得了更好的结果。

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

表 5：检测任务的单模型性能

9 总结

我们的结果似乎提供了一个坚实的证据，即通过现成的密集构建块来逼近期望的最优稀疏结构是改进计算机视觉神经网络的可行方法。这种方法的主要优势是在与更浅和更窄的网络相比，仅在计算需求上适度增加的情况下显著提高了质量。还要注意的是，尽管我们的检测工作既没有利用上下文，也没有执行边界框回归，但它仍具有竞争力，这进一步证明了 Inception 架构的强大之处。虽然预计通过更昂贵的深度和宽度相似的网络可以实现类似的结果质量，但我们的方法提供了坚实的证据，即采用更稀疏的架构是可行且有用的想法。这表明在基于[2]的基础上以自动化方式创建更稀疏和更精细的结构是有前景的未来工作。

10 致谢

我们要感谢 Sanjeev Arora 和 Aditya Bhaskara 对[2]的富有成果的讨论。同时，我们要感谢 DistBelief [4]团队的支持，特别是 Rajat Monga，Jon Shlens，Alex Krizhevsky，Jeff Dean，Ilya Sutskever 和 Andrea Frome。我们还要感谢 Tom Duerig 和 Ning Ye 对光度失真方面的帮

助。此外，没有 Chuck Rosenberg 和 Hartwig Adam 的支持，我们的工作将无法实现。

References

[1] Know your meme: We need to go deeper. http://knowyourmeme.com/memes/we-need-to-go-deeper. Accessed:2014-09-15.

[2] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. CoRR, abs/1310.6343, 2013.

[3] ¨Umit V. C¸ataly¨urek, Cevdet Aykanat, and Bora Uc¸ar. On two-dimensional sparse matrix partitioning: Models, methods, and a recipe. SIAM J. Sci. Comput., 32(2):656–683, February 2010.

[4] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large scale distributed deep networks. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, editors, Advances in Neural Information Processing Systems 25,pages 1232–1240. 2012.

[5] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition, 2014.CVPR 2014. IEEE Conference on, 2014.

[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, 2014.

[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580, 2012.

[8] Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. CoRR, abs/1312.5402, 2013.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, pages 1106–1114, 2012.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541–551, December 1989.

[11] Yann LeCun, L´eon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[12] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.

[13] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. SIAM J. Control Optim., 30(4):838–855, July 1992.

[14] Pierre Sermanet, David Eigen, Xiang Zhang, Micha¨el Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013.

[15] Thomas Serre, Lior Wolf, Stanley M. Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell., 29(3):411–426, 2007.

[16] Fengguang Song and Jack Dongarra. Scaling up matrix computations on shared-memory manycore systems with 1000 cpu cores. In Proceedings of the 28th ACM International Conference on Supercomputing, ICS '14, pages 333–342, New York, NY, USA, 2014. ACM.

[17] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, volume 28 of JMLR Proceedings, pages 1139–1147. JMLR.org, 2013.

[18] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In Christopher J. C. Burges, L´eon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 2553–2561, 2013.

[19] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. CoRR, abs/1312.4659, 2013.

[20] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 1879–1886, Washington, DC, USA, 2011. IEEE Computer Society.

[21] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tom´as Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, volume 8689 of Lecture Notes in Computer Science, pages 818–833. Springer,2014.