使用卷积进行更深层次的研究

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

我们提出了一个名为 Inception 的深度卷积神经网络结构，在 ImageNet 大规模视觉识别挑战赛 2014（ILSVRC14）的分类和检测任务中创造了新的技术水平。这个架构的主要特点是改进了网络内部计算资源的利用。通过精心设计，我们实现了在保持计算预算不变的情况下增加网络的深度和宽度。为了优化质量，架构决策基于 Hebbian 原则和多尺度处理的直觉。在我们提交的 ILSVRC14 中使用的一个特定版本被称为 GoogLeNet，这是一个 22 层深的网络，模型质量是在分类和检测的背景下进行评估的。

1 引言

过去三年来，主要是由于深度学习的进步，具体来说是卷积神经网络，图像识别和目标检测的质量已经以惊人的速度在进步。一个令人鼓舞的消息是，这种进步大部分不仅仅是由于更强大的硬件、更大的数据集和更大的模型，而主要是由于新的思想、算法和改进的网络架构。举例来说，除了用于检测目的的 ILSVRC 2014 竞赛的相同分类数据集外，在 ILSVRC 2014 竞赛中排名靠前的参赛作品没有使用新的数据来源。我们提交给 ILSVRC 2014 的 GoogLeNet 实际上比两年前 Krizhevsky 等人的获胜架构少了 12 个参数，而且更加准确。目标检测方面的最大收益不是仅仅来自于深度网络的利用或更大的模型，而是来自于深度架构和经典计算机视觉的协同作用，例如 Girshick 等人的 R-CNN 算法。

另一个值得注意的因素是，随着移动计算和嵌入式计算的不断发展，我们算法的效率（尤其是功耗和内存使用）变得越来越重要。值得注意的是，本文所介绍的深度架构的设计考虑因素包括了这一因素，而不是一味地追求准确率。在大多数实验中，模型的设计都是为了在推理时保持 15 亿次乘法

加法的计算预算，这样它们就不会最终成为纯粹的学术奇观，而是可以以合理的成本投入实际应用，即使是在大型数据集上。

在本文中，我们将重点讨论一种用于计算机视觉的高效深度神经网络架构，其代号为 “Inception”，其名称源自 Lin 等人的网络论文[12]中的 “Network in network”，以及著名的 “we need to go deeper ”网络流行语[1]。在我们的案例中，“深入 ”一词有两种不同的含义：首先是指我们以 “Inception 模块 ”的形式引入了一个新的组织层次，还有更直接的含义，即增加网络深度。总的来说，我们可以将 Inception 模型视为 [12] 的逻辑结晶，同时从 Arora 等人 [2] 的理论研究中汲取灵感和指导。该架构的优势在 ILSVRC 2014 分类和检测挑战赛上得到了实验验证，其性能明显优于目前的技术水平。

2 相关工作

从 LeNet-5 [10]开始，卷积神经网络（CNN）通常采用标准结构--堆叠卷积层（可选择进行对比度归一化和最大池化），然后是一个或多个全连接层。这种基本设计的变体在图像分类文献中非常普遍，并在 MNIST、CIFAR 以及最著名的 ImageNet 分类挑战中取得了迄今为止最好的结果[9, 21]。对于像 Imagenet 这样的大型数据集，最近的趋势是增加层数 [12] 和层大小 [21, 14]，同时使用 dropout [7] 来解决过拟合问题。

尽管有人担心最大池化层会导致准确空间信息的丢失，但与 [9] 相同的卷积网络架构也被成功用于定位 [9, 14]、物体检测 [6, 14, 18, 5] 和人体姿态估计 [19]。受到灵长类动物视觉皮层神经科学模型的启发，Serre 等人[15]使用了一系列不同大小的固定 Gabor 滤波器，以处理多个尺度，这与 Inception 模型类似。不过，与 [15] 中的固定 2 层深度模型不同，Inception 模型中的所有滤波器都是通过学习获得的。此外，Inception 层会重复多次，因此 GoogLeNet 模型有 22 层深度模型。

“网络中的网络（network in network）”是 Lin 等人[12]提出的一种方法，旨在提高神经网络的表征能力。当应用到卷积层时，该方法可被视为额外的 1×1 卷积层，之后是典型的整流线性激活[9]。这样，它就能很容易地集成到当前的 CNN 管道中。我们在架构中大量使用了这种方法。不过，在我们的设置中，1×1 卷积具有双重目的：最关键的是，它们主要用作降维模块，以消除计算瓶颈，否则会限制我们网络的规模。这样，我们不仅可以增加网络的深度，还可以增加网络的宽度，而不会对性能造成显著影响。

Girshick 等人提出的卷积神经网络区域（Regions with Convolutional Neural Networks，R-CNN）是目前物体检测的主要方法[6]。R-CNN 将整体检测问题分解为两个子问题：首先利用颜色和超像素一致性等低级线索，以类别无关的方式提出潜在的物体建议，然后使用 CNN 分类器识别这些位置上的物体类别。这种两阶段方法充分利用了边界框分割与低级线索的准确性，以及最先进的 CNN 的强大分类能力。我们在提交的检测报告中也采用了类

似的方法，但在这两个阶段都进行了改进，例如采用多边框[5]预测来提高物体边框的召回率，以及采用集合方法对边框提议进行更好的分类。

3 动机和高层次考虑因素

提高深度神经网络性能的最直接方法就是扩大其规模。这包括增加网络的深度（层级数量）和宽度（每个层级的单元数量）。这是训练更高质量模型的一种简单而安全的方法，尤其是在有大量标注训练数据的情况下。然而，这种简单的解决方案有两大缺点。

更大的规模通常意味着更多的参数，这使得扩大后的网络更容易出现过度拟合，尤其是在训练集中标注示例数量有限的情况下。这可能会成为一个主要瓶颈，因为创建高质量的训练集既麻烦又昂贵，尤其是在需要人类专家来区分细粒度视觉类别的情况下，如图 1 所示，ImageNet 中的类别（即使是在 1000 个类别的 ILSVRC 子集中）。

(a) 西伯利亚哈士奇　　(b) 爱斯基摩犬

图 1：ILSVRC 2014 分类挑战赛 1000 个类别中的两个不同类别。

均匀增加网络规模的另一个缺点是会大幅增加计算资源的使用。例如，在深度视觉网络中，如果两个卷积层是连锁的，其滤波器数量的均匀增加会导致计算量的二次方增加。如果增加的容量使用效率不高（例如，如果大多数权重最终接近于零），那么就会浪费大量计算资源。由于在实践中计算预算总是有限的，因此即使主要目标是提高结果质量，有效分配计算资源也比盲目增加计算量要好。

解决这两个问题的根本方法是最终从全连接转移到稀疏连接的架构，甚至在卷积内部也是如此。除了模仿生物系统，这样做还有一个优势，即由于 Arora 等人的开创性工作，具有更坚实的理论基础。他们的主要结果表明，如果数据集的概率分布可以由一个大型、非常稀疏的深度神经网络表示，那么可以通过分析最后一层的激活的相关统计信息并对具有高度相关输出的神经元进行聚类，逐层构建最优网络拓扑结构。尽管严格的数学证明需要非常强的条件，但这个说法与著名的 Hebbian 原则 resonates——即一起激活的神

经元会相互连接——这表明这个基本思想即使在不太严格的条件下在实践中也是适用的。

另一方面，当今的计算基础设施在对非均匀稀疏数据结构进行数值计算时效率非常低。即使算术运算次数减少 100 倍，查找和高速缓存缺失的开销也非常大，因此转而使用稀疏矩阵不会带来任何收益。由于使用了不断改进、高度调整的数值库，可以利用底层 CPU 或 GPU 硬件的微小细节实现极快的密集矩阵乘法，因此差距进一步拉大[16, 9]。此外，非均匀稀疏模型需要更复杂的工程和计算基础设施。目前大多数面向视觉的机器学习系统只是通过采用卷积来利用空间领域的稀疏性。然而，卷积是作为与前一层斑块的密集连接集合来实现的。自[11]以来，ConvNets 传统上在特征维度上使用随机稀疏连接表，以打破对称性并提高学习效率，但为了更好地优化并行计算，[9]的趋势又变回了全连接。结构的统一性、大量的滤波器和更大的批处理规模，使得密集计算的效率得以提高。

这就提出了一个问题：是否有希望实现下一个中间步骤：如理论所建议的那样，利用额外的稀疏性（甚至在滤波器级），但通过利用密集矩阵计算来利用我们当前的硬件的架构。有关稀疏矩阵计算的大量文献（如 [3]）表明，将稀疏矩阵聚类为相对密集的子矩阵，往往能为稀疏矩阵乘法带来最先进的实用性能。在不久的将来，利用类似方法自动构建非均匀深度学习架构似乎并不牵强。

Inception 架构最初是第一作者的一项案例研究，用于评估复杂的网络拓扑结构构建算法的假设输出，该算法试图近似[2]为视觉网络隐含的稀疏结构，并通过密集的、随时可用的组件覆盖假设结果。尽管这是一项高度推测性的工作，但在对拓扑结构的精确选择进行了两次迭代之后，我们已经可以看到与基于[12]的参考架构相比取得了适度的收益。在对学习率、超参数和改进的训练方法进行进一步调整后，我们发现，作为 [6] 和 [5] 的基础网络，Inception 架构在定位和物体检测方面特别有用。有趣的是，虽然大多数最初的架构选择都受到了质疑和全面测试，但事实证明它们至少是局部最优的。

不过，我们必须谨慎行事：尽管所提出的架构已在计算机视觉领域取得成功，但其质量是否归功于构建该架构的指导原则仍值得怀疑。要确定这一点，需要进行更深入的分析和验证：例如，基于下述原则的自动化工具是否能为视觉网络找到类似但更好的拓扑结构。最有说服力的证明是，如果一个自动化系统能够使用相同的算法创建网络拓扑结构，从而在其他领域获得类似的收益，但其全局架构却截然不同。至少，Inception 架构的初步成功为未来在这一方向开展令人兴奋的工作提供了坚实的动力。

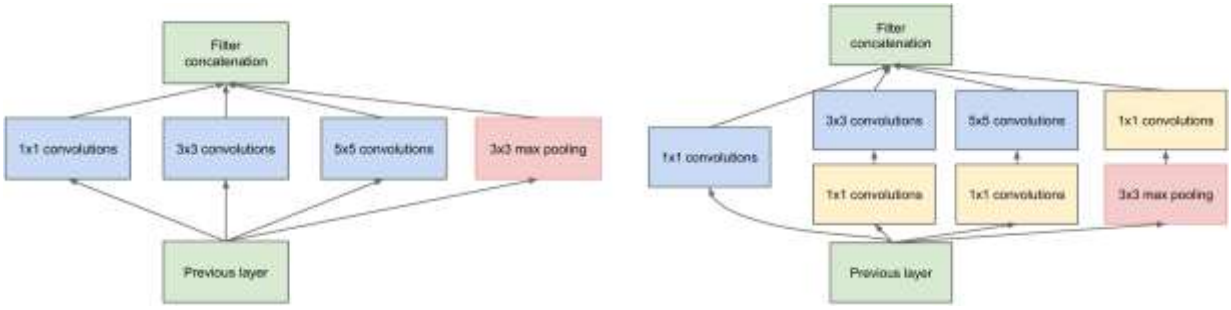
4 架构细节

Inception 架构的主要思想是找出如何在卷积视觉网络中逼近和覆盖一个最优的局部稀疏结构，而这个结构可以由现成的密集组件来实现。需要注意的是，假设平移不变性意味着我们的网络将由卷积构建块构建。我们所需

要的就是找到最优的局部构造，并在空间上重复它。Arora 等人建议进行逐层构建，分析最后一层的相关统计信息，并将其聚类成具有高相关性的单元组。这些聚类形成下一层的单元，并与前一层的单元相连。我们假设较早层的每个单元对应于输入图像的某个区域，并且这些单元被分组成滤波器组。在较低层（靠近输入的层）中，相关单元会集中在局部区域。这意味着我们最终会得到许多集中在单个区域的聚类，并且它们可以由下一层的 1×1 卷积覆盖，正如[12]中所建议的。然而，人们也可以预期到将会有更少、更空间分散的聚类，这些聚类可以由覆盖更大补丁的卷积来处理，而且随着区域的增大，覆盖的补丁数量会逐渐减少。为了避免补丁对齐问题，Inception 架构的当前版本限制了滤波器尺寸为 1×1、3×3 和 5×5，然而这个决定更多地基于方便而非必要性。这也意味着建议的架构是所有这些层的组合，它们的输出滤波器组被连接成一个单一的输出向量，形成下一阶段的输入。此外，由于池化操作对于当前最先进的卷积网络的成功至关重要，因此建议在每个阶段中添加一个替代的并行池化路径也应该具有额外的益处（见图 2(a)）。

随着这些"Inception 模块"的堆叠，它们的输出相关统计信息必然会有所变化：随着更高层次捕捉到更抽象的特征，它们的空间集中度预计会降低，这意味着在向更高层次移动时，3×3 和 5×5 卷积的比例应该增加。

对于上述模块来说，至少在这种天真的形式中，一个较小数量的 5×5 卷积在具有大量滤波器的卷积层之上可能会成本过高。一旦池化单元被添加到混合中，这个问题变得更加突出：它们的输出滤波器数量等于前一阶段的滤波器数量。池化层的输出与卷积层的输出合并将不可避免地导致网络表达能力的截断。因此，在向架构中添加 5x5 卷积和池化单元时，需要仔细考虑计算成本和对网络表达能力的影响。



(a)Inception 模块，朴素版　　（b）inception 模块，带维度缩减

图 2，inception 模块

池化层的输出与卷积层的输出合并会导致输出数量在每个阶段中不可避免地增加。即使这种架构可能覆盖了最佳的稀疏结构，但它的效率非常低下，会在几个阶段内导致计算量激增。

这引出了所提出的架构的第二个想法：在计算要求过高的地方，明智地应用维度缩减和投影。这基于嵌入的成功：即使低维嵌入可能包含相对较大的图像块的大量信息。然而，嵌入以稠密、压缩的形式表示信息，而压缩信息更难建模。我们希望在大多数地方保持我们的表示稀疏（根据[2]的条件要求），并且只在需要大量聚合信号时才压缩信号。也就是说，1×1 卷积用于在昂贵的 3×3 和 5×5 卷积之前计算缩减。除了用作缩减外，它们还包括使用修正线性激活，使它们具有双重用途。最终结果如图 2(b)所示。

一般来说，Inception 网络是由上述类型的模块堆叠在一起组成的网络，偶尔会使用步幅为 2 的最大池化层来减半网格的分辨率。出于技术原因（在训练期间的内存效率），似乎有益于仅在较高层开始使用 Inception 模块，同时保持较低层以传统的卷积方式进行处理。这并非严格必要，只是反映了当前实现中的一些基础设施效率低下。

这种架构的一个主要益处在于，它允许显著增加每个阶段的单元数量，而不会导致计算复杂性失控地激增。普遍使用维度缩减可以将上一阶段的大量输入滤波器屏蔽到下一层，首先在对它们进行大块尺寸的卷积之前减少它们的维度。这种设计的另一个实用方面是，它符合视觉信息应该以各种尺度进行处理，然后聚合，以便下一阶段可以同时从不同尺度抽象特征的直觉。

对计算资源的改进使得可以增加每个阶段的宽度以及阶段的数量，而不会陷入计算困难。另一种利用 Inception 架构的方法是创建略逊一筹但计算成本更低的版本。我们发现，所有包括的调节和控制手段可以实现对计算资源的可控平衡，从而导致比非 Inception 架构性能相似的网络快 2-3 倍，但这需要在设计上进行仔细的手动设计。

5 GoogLeNet

我们选择将 GoogLeNet 作为我们在 ILSVRC14 比赛中的团队名称。这个名字是向 Yann LeCun 的开创性 LeNet 5 网络[10]致敬。我们也用 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，其质量略低，但将其添加到集成中似乎略微改善了结果。我们省略了该网络的细节，因为我们的实验表明，确切的架构参数的影响相对较小。

在这里，为了演示目的，描述了最成功的特定实例（名为 GoogLeNet）如表 1 所示。完全相同的拓扑结构（使用不同的采样方法进行训练）被用于我们集成中的 7 个模型中的 6 个。

所有的卷积，包括 Inception 模块内部的卷积，都使用修正线性激活。我们网络中的感受野大小为 224×224，采用 RGB 颜色通道进行均值减法。"#3×3 reduce"和"#5×5 reduce"代表 3×3 和 5×5 卷积之前使用的减少层中 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。所有这些减少/投影层也使用修正线性激活。

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: GoogLeNet inception 的结构

所有的卷积操作，包括 Inception 模块内部的卷积，都使用修正线性激活。我们网络中的感受野大小为 224×224，采用 RGB 颜色通道进行均值减法。“#3×3 reduce”和“#5×5 reduce”代表 3×3 和 5×5 卷积之前的缩小层中 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。所有这些缩小/投影层也使用修正线性激活。

该网络的设计考虑了计算效率和实用性，因此推断可以在个别设备上运行，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。该网络在仅计算具有参数的层时深度为 22 层（如果还计算池化层，则为 27 层）。用于构建网络的总层数（独立的构建模块）约为 100。然而，这个数字取决于所使用的机器学习基础设施系统。在分类器之前使用平均池化是基于[12]的，尽管我们的实现不同之处在于我们使用了额外的线性层。这使得我们可以轻松地调整和微调网络以适应其他标签集，但这主要是为了方便，我们并不指望它会产生重大影响。研究发现，从全连接层转移到平均池化层可以将 top-1 准确率提高约 0.6%，然而即使去除了全连接层，使用 dropout 仍然是必不可少的。

鉴于网络的相对较大深度，有效地将梯度传播回所有层是一个值得关注的问题。有趣的一点是，相对较浅的网络在这个任务上表现良好，这表明网络中间层产生的特征应该具有很强的区分性。通过添加连接到这些中间层的辅助分类器，我们期望能够促进分类器中较低阶段的区分度，增加被传播回去的梯度信号，并提供额外的正则化。这些分类器采用较小的卷积网络的形式，放置在 Inception (4a)和(4d)模块的输出之上。在训练期间，它们的损失被加到整个网络的总损失中，使用折扣权重（辅助分类器的损失以 0.3 的权重进行加权）。在推断时，这些辅助网络被丢弃。

,

图 3: 功能齐全的 GoogLeNet 网络

包括辅助分类器在内的侧边额外网络的具体结构如下：

- 一个平均池化层，使用 5×5 的滤波器大小和 3 的步长，在(4a)阶段得到 4×4×512 的输出，在(4d)阶段得到 4×4×528 的输出。
- 一个 1×1 卷积，使用 128 个滤波器进行维度缩减，并采用修正线性激活。
- 一个具有 1024 个单元和修正线性激活的全连接层。
- 一个 droout 层，丢弃输出的比例为 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）。 图 3 中展示了该网络的示意图。

图 3 是最终网络的示意图。

6 训练方法

我们的网络是使用 DistBelief [4]分布式机器学习系统进行训练的，使用了适度数量的模型和数据并行性。尽管我们仅使用基于 CPU 的实现，但粗略估计表明，GoogLeNet 网络可以在一周内使用少量高端 GPU 训练到收敛，主要限制是内存使用。我们的训练使用了异步随机梯度下降和 0.9 的动量[17]，固定的学习率调度（每 8 个 epochs 将学习率降低 4%）。在推断时，使用了 Polyak 平均[13]来创建最终的模型。

在竞赛前几个月，我们的图像采样方法发生了重大变化，已经收敛的模型也使用了其他选项进行训练，有时还结合改变的超参数，比如 dropout 和学习率，因此很难给出对这些网络进行训练最有效的单一方法的明确指导。更进一步复杂化问题的是，一些模型主要是在相对较小的裁剪图像上训练的，另一些则是在较大的裁剪图像上训练的，这受到了[8]的启发。然而，经过竞赛验证为非常有效的一种方法包括对图像进行各种尺寸的采样，其尺寸均匀分布在图像面积的 8%到 100%之间，其宽高比在 3/4 到 4/3 之间随机选择。此外，我们发现 Andrew Howard 提出的光度扭曲方法在一定程度上对抗过拟合很有用。此外，我们开始比较晚地使用随机插值方法（双线性插值、区域插值、最近邻插值和立方插值，概率相等）来调整大小，并结合其他超参数的更改，因此我们无法确定最终结果是否受到了它们的积极影响。

7 ILSVRC 2014 分类挑战赛的设置和结果

ILSVRC 2014 分类挑战任务涉及将图像分类为 Imagenet 层次结构中的 1000 个叶节点类别之一。训练集包含大约 120 万张图像，验证集包含 5 万张图像，测试集包含 10 万张图像。每个图像都与一个真实类别相关联，性能是基于最高得分的分类器预测来衡量的。通常报告两个数字：top-1 准确率，将真实类别与第一个预测类别进行比较；top-5 错误率，将真实类别与

前 5 个预测类别进行比较：如果真实类别在前 5 个中，不论其在其中的排名如何，都认为图像被正确分类。挑战使用 top-5 错误率进行排名。

我们参加了挑战，没有使用外部数据进行训练。除了本文中提到的训练技术之外，我们在测试过程中采用了一系列技术来获得更高的性能，下面我们将详细阐述。

1. 我们独立训练了 7 个版本的相同 GoogLeNet 模型（包括一个更宽的版本），并对它们进行了集成预测。这些模型使用相同的初始化（甚至是相同的初始权重，主要是由于疏忽），以及学习率策略进行训练，它们唯一的区别在于采样方法和观察输入图像的随机顺序。

2. 在测试过程中，我们采用了比 Krizhevsky 等人[9]更激进的裁剪方法。具体来说，我们将图像调整为 4 个尺度，其中较短的尺度（高度或宽度）分别为 256、288、320 和 352，然后取这些调整大小的图像的左、中、右方形（对于纵向图像，我们取顶部、中部和底部方形）。对于每个方形，我们再取 4 个角和中心的 224×224 裁剪，以及正方形调整为 224×224、以及它们的镜像版本。这样，每幅图像的裁剪次数为 4×3×6×2=144。安德鲁-霍华德（Andrew Howard）[8] 在前一年的参赛作品中也使用了类似的方法，我们通过经验验证，其性能略逊于所提出的方案。我们注意到，在实际应用中可能并不需要如此激进的裁剪，因为当裁剪数量达到一定程度后，更多裁剪的好处就变得微不足道了（正如我们稍后将展示的那样）。

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

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 分类性能细分

3. 对多个作物和所有分类器的软最大概率取平均值，得出最终预测结果。在实验中，我们分析了验证数据的其他方法，如作物最大池化和分类器平均，但这些方法的性能都不如简单平均。

在本文的其余部分，我们将分析对最终提交的整体性能产生影响的多个因素。

我们在挑战中的最终提交在验证和测试数据上获得了 6.67%的 top-5 错误率，在其他参与者中排名第一。与 2012 年的 SuperVision 方法相比，这是相对减少了 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来训练分类器。以下表格显示了一些表现最佳方法的统计数据。

我们还通过改变模型数量和预测图像时使用的裁剪数量，分析并报告了多种测试选择的性能。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。所有数字均在验证数据集上报告，以避免过度拟合测试数据统计。

8 ILSVRC 2014 检测挑战的设置和结果

ILSVRC 检测任务是在可能的 200 个类别中，在图像中产生物体的边界框。如果检测到的对象与真实类别匹配，并且它们的边界框重叠至少 50%（使用 Jaccard 指数），则被视为正确。多余的检测被视为误报，并会受到惩罚。与分类任务相反，每个图像可能包含许多对象或没有对象，并且它们的尺度可能从大到小不等。结果使用平均精度均值（mAP）进行报告。

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

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：单一模型的检测性能

GoogLeNet 在检测方面采用的方法类似于[6]中的 R-CNN，但增加了 Inception 模型作为区域分类器。此外，通过将选择性搜索[20]方法与多盒[5]预测相结合，改进了区域提议步骤，以提高对象边界框的召回率。为了减少误报，超像素尺寸增加了 2 倍。这样可以减少选择性搜索算法提出的建议数量。我们重新添加了来自多盒[5]的 200 个区域建议，总共使用了大约 60%的建议，而覆盖范围从 92%增加到 93%。减少建议数量并增加覆盖范围的整体效果是，对于单模型情况，平均精度均值提高了 1%。最后，在对每个区域进行分类时，我们使用了 6 个 ConvNets 的集成，将准确率从 40%提高到 43.9%。需要注意的是，与 R-CNN 不同，由于时间不足，我们没有使用边界框回归。

我们首先报告了顶级的检测结果，并展示了自检测任务首次发布以来的进展。与 2013 年的结果相比，准确性几乎翻了一番。表现最好的团队都使用了卷积网络。我们在表 4 中报告了官方得分以及每个团队的常见策略：使用外部数据、集成模型或上下文模型。外部数据通常是 ILSVRC12 分类数据，用于预训练模型，然后在检测数据上进行微调。一些团队还提到了使用定位数据。由于定位任务的大部分边界框未包含在检测数据集中，可以像分类预训练一样，使用此数据预先训练一个通用边界框回归器。GoogLeNet 条目没有使用定位数据进行预训练。

在表 5 中，我们只使用单个模型进行了比较。表现最佳的模型是 Deep Insight，令人惊讶的是，即使使用 3 个模型的集成，其结果仅提高了 0.3 个百分点，而 GoogLeNet 在使用集成模型时获得了明显更强的结果。

9 结论

我们的研究结果似乎提供了一个确凿的证据，即用现成的密集构件来逼近预期的最佳稀疏结构，是改进计算机视觉神经网络的一种可行方法。这种方法的主要优势在于，与较浅和较宽的网络相比，只需适度增加计算要求，就能显著提高质量。我们还注意到，尽管我们既没有利用上下文，也没有进行边界框回归，但我们的检测工作仍具有竞争力，这一事实进一步证明了 Inception 架构的优势。虽然预计类似深度和宽度的昂贵网络也能达到类似的

结果质量，但我们的方法提供了确凿的证据，证明转向更稀疏的架构总体上是可行且有用的想法。这表明，在 [2] 的基础上，未来以自动化方式创建更稀疏、更精细的结构的工作大有可为。

10 致谢

我们要感谢 Sanjeev Arora 和 Aditya Bhaskara 就 [2] 进行的富有成效的讨论。我们还要感谢 DistBelief [4] 团队的支持，特别是 Rajat Monga、Jon Shlens、Alex Krizhevsky、Jeff Dean、Ilya Sutskever 和 Andrea Frome。我们还要感谢汤姆-杜里格（Tom Duerig）和叶宁（Ning Ye）在测光失真方面提供的帮助。此外，如果没有查克-罗森伯格和哈特维格-亚当的支持，我们的工作也不可能完成。

11 参考资料

[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 par-¨ titioning: 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, Leon 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, Michael Mathieu, Rob Fergus, and Yann Le-¨ Cun. 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, Leon 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, Tomas 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.