Going Deeper with Convolutions

Christian Szegedy, Wei Liu, Yangqing Jia. et al

翻译：莫天池

版本号：V1.0.0

2016 年 3 月

Going Deeper with Convolutions	
Abstract We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.	**摘要** 我们提出了一个名为“Inception”的深度卷积神经网结构，其目标是将分类、识别 ILSVRC14 数据集的技术水平提高一个层次。这一结构的主要特征是对网络内部计算资源的利用进行了优化。 这一目标的实现是通过细致的设计，使得在保持计算消耗稳定不变的同时增加网络的宽与深。 为了提高质量，网络结构基于赫布原则（Hebbian principle）和多尺度处理规则（intuition of multi-scale processing）设计。一个具体化的例子是所谓 GoogLeNet，也就是我们提交到 ILSVRC14 的成果，它是一个 22 层深的网络，其质量在分类和检测这两项指标中获得评估。
1 Introduction In the last three years, mainly due to the advances of deep learning, more concretely convolutional networks [10], the quality of image recognition and object detection has been progressing at a dramatic pace. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures. No new data sources were used, for example, by the top entries in the ILSVRC 2014 competition besides the classification dataset of the same competition for detection purposes. Our GoogLeNet submission to ILSVRC 2014 actually uses 12× fewer parameters than the winning	**1 引言** 最近三年，主要由于深度学习和越来越实际的卷积网络的发展【10】，图像识别以及物体检测的质量都在以惊人的速度提高。 一个振奋人心的消息是大多数进步并不只是更强大的硬件、更大的数据库和模型所带来的，而主要是一些新创意、新算法，以及优化的网络结构的成果。 现在，新的数据来源已经能够使用，比如最顶级的 ILSVRC 2014 不仅会进行分类方面的竞赛，也会进行物体检测方面的竞赛。我们提交到 ILSVRC 2014 的 GoogLeNet 实际上使用了 比赢得两年前比赛的 K【9，即 AlexNet】 少 12 倍的参数，但精确度提高了很多。

architecture of Krizhevsky et al [9] from two years ago, while being significantly more accurate. The biggest gains in object-detection have not come from the utilization of deep networks alone or bigger models, but from the synergy of deep architectures and classical computer vision, like the R-CNN algorithm by Girshick et al [6].	在物体识别方面，最大的收获其实并不来自于深度网络或是大型模型的单独使用，而是来自深度结构和传统机器视觉的协同作用，比如 G【6】提出来的 R-CNN 算法。
Another notable factor is that with the ongoing traction of mobile and embedded computing, the efficiency of our algorithms – especially their power and memory use – gains importance. It is noteworthy that the considerations leading to the design of the deep architecture presented in this paper included this factor rather than having a sheer fixation on accuracy numbers. For most of the experiments, the models were designed to keep a computational budget of 1.5 billion multiply-adds at inference time, so that the they do not end up to be a purely academic curiosity, but could be put to real world use, even on large datasets, at a reasonable cost.	另一个值得注意的要素是随着移动计算和嵌入式计算得到越来越广泛的认同，我们的算法的效率——尤其是其能量和存储利用率——变得越来越重要。值得注意的是，这篇文章中展现的深度结构在设计时就考虑了这些因素，而不仅是执着于单纯提高精度。 对于我们的大部分实验，模型计算量限制在预测时间内 15 亿次乘加运算左右，这让我们的实验并不仅仅是为了满足学术好奇心（而盲目提高精确度），而是可以在现实中使用，即使对于很大的数据集，开销也是合理的。
In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed Inception, which derives its name from the Network in network paper by Lin et al [12] in conjunction with the famous “we need to go deeper” internet meme [1]. In our case, the word “deep” is used in two different meanings: first of all, in the sense that we introduce a new level of organization in the form of the “Inception module” and also in the more direct sense of increased network depth. In general, one can view the Inception model as a logical culmination of [12] while taking inspiration and guidance from the theoretical work by Arora et al [2]. The	在本文中，我们所关注的是一个应用于计算机视觉的深度神经网络，名为“Inception”，它的名字来源于 Lin 等人【12】关于网络的论文，以及名言“我们要走向深度”。在我们这，“深”有两层含义：首先，我们引入了一种高水平的组织方式来构建 Inception 的模块，同时以更加直接的方式来增加网络深度。一般而言，把 Inception 模型看做一个在 Arora【2】的理论工作所激发的灵感的指引下所达到的巅峰是合理的。网络结构的优势已经在 ILSVRC 2014 分类与检测挑战中得到验证，在比赛中它大大超越了现有水平。

benefits of the architecture are experimentally verified on the ILSVRC 2014 classification and detection challenges, on which it significantly outperforms the current state of the art.	
2 Related Work Starting with LeNet-5 [10], convolutional neural networks (CNN) have typically had a standard structure – stacked convolutional layers (optionally followed by contrast normalization and max- pooling) are followed by one or more fully-connected layers. Variants of this basic design are prevalent in the image classification literature and have yielded the best results to-date on MNIST, CIFAR and most notably on the ImageNet classification challenge [9, 21]. For larger datasets such as Imagenet, the recent trend has been to increase the number of layers [12] and layer size [21, 14], while using dropout [7] to address the problem of overfitting.	**2 相关研究** 从 LeNet-5 开始【10】，卷积神经网络（CNN）就已经具有标准化的结构了——堆叠起来的卷积层（可能后面跟着对比度归一化层和最大池化层），后面跟随着全连接层。这种基础设计的变种在图像分类领域十分流行，并且在 MNIST，CIFAR 等数据集，尤其是 ImageNet 分类挑战赛【9，21】中产生了极佳的结果。对于 ImageNet 这样的大型数据集，最近流行的趋势是增加层数【12】和每一层的大小【21，14】，并利用 dropout 算法解决过拟合问题。
Despite concerns that max-pooling layers result in loss of accurate spatial information, the same convolutional network architecture as [9] has also been successfully employed for localization [9, 14], object detection [6, 14, 18, 5] and human pose estimation [19]. Inspired by a neuroscience model of the primate visual cortex, Serre et al. [15] use a series of fixed Gabor filters of different sizes in order to handle multiple scales, similarly to the Inception model. However, contrary to the fixed 2-layer deep model of [15], all filters in the Inception model are learned. Furthermore, Inception layers are repeated many times, leading to a 22-layer deep model in the case of the GoogLeNet model.	虽然对最大池化的关注造成了准确空间信息的丧失，文献【9】中的网络结构还是被成功地应用到了局部化【9，14】，物体检测【6，14，18，5】和人体姿势识别【19】等方面。受到神经科学对主要视觉皮层进行建模的启发，Serre 等人【15】用一系列不同大小的固定的（fixed）Gabor 过滤器去处理多尺度，这与 Inception 是相同的。然而，相比文献【15】中 fixed 的两层模型，Inception 中所有过滤器是学习得到的。进一步的，Inception 的各层都重复多次出现，形成了 GoogLeNet——一个 22 层网络模型。
Network-in-Network is an approach proposed by Lin et al. [12] in order to increase the	网中网（Network-in-Network）是 Lin 提出来的一种结构【12】，其目的是为了增加神经网络的表

representational power of neural networks. When applied to convolutional layers, the method could be viewed as additional 1 × 1 convolutional layers followed typically by the rectified linear activation [9]. This enables it to be easily integrated in the current CNN pipelines. We use this approach heavily in our architecture. However, in our setting, 1×1 convolutions have dual purpose: most critically, they are used mainly as dimension reduction modules to remove computational bottlenecks, that would otherwise limit the size of our networks. This allows for not just increasing the depth, but also the width of our networks without significant performance penalty.	现力。当应用于卷积层的时候，这一方法可以看做一个额外的 1×1 卷积层，后面通常跟着一个修正的线性激活（rectified linear activation）。这使得 Network-in-Network 能够轻松地集成到现有的 CNN 管道中。这种方法在我们的网络体系结构中被大量地使用。然而，在我们的设定中，1×1 卷积具有双重目的：最重要的一点是，它们被主要用于降维模块以打破计算瓶颈，否则我们的网络规模会受到限制。这使得我们不仅可以加深网络，同时还可以加宽，而不造成严重的性能下降。
The current leading approach for object detection is the Regions with Convolutional Neural Networks (R-CNN) proposed by Girshick et al. [6]. R-CNN decomposes the overall detection problem into two subproblems: to first utilize low-level cues such as color and superpixel consistency for potential object proposals in a category-agnostic fashion, and to then use CNN classifiers to identify object categories at those locations. Such a two stage approach leverages the accuracy of bounding box segmentation with low-level cues, as well as the highly powerful classification power of state-of-the-art CNNs. We adopted a similar pipeline in our detection submissions, but have explored enhancements in both stages, such as multi-box [5] prediction for higher object bounding box recall, and ensemble approaches for better categorization of bounding box proposals.	现在最好的物体检测方法是区域卷积神经网【？Regions with Convolutional Neural Networks (R-CNN)】，由 Girshick【6】提出。R-CNN 将整个检测问题分解为两个子问题：第一部使用低层线索比如组成潜在物体的颜色、超像素等，提取出一些类别不可知的信息，然后接下来利用 CNN 在这些区块信息上识别物体类别。这种双步方法中，低层线索会影响切分区块大小的准确性【Such a two stage approach leverages the accuracy of bounding box segmentation with low-level cues,】以及 CNN 分类的准确度。我们在提交的检测程序中采用了同样的管道，但我们对其中的每一步都进行了加强，比如采用多盒预测【5】以提高边界识别集合的召回率【？ such as multi-box [5] prediction for higher object bounding box recall】，还对 bounding box 提出的分类建议进行了不同方法的搭配合成，以获得更好的结果。
3 Motivation and High Level Considerations The most straightforward way of improving the	**3 动机与高层设计考虑** 最直接提高深度神经网络性能的方法是增加其规

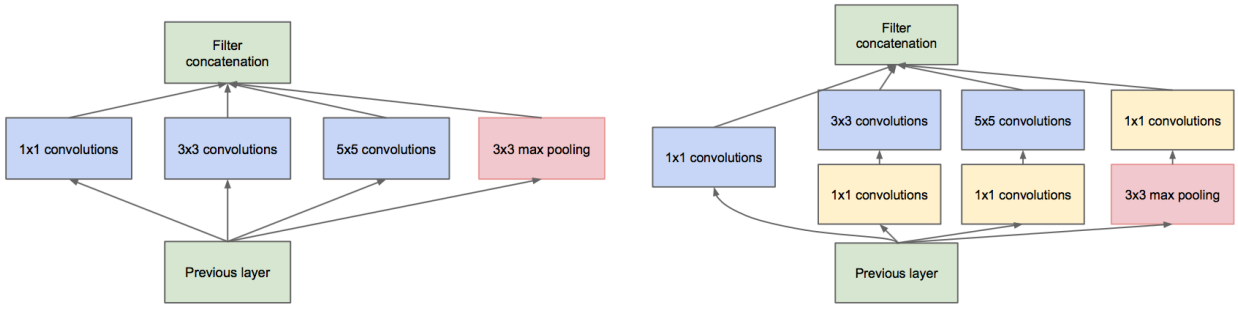
performance of deep neural networks is by increasing their size. This includes both increasing the depth – the number of levels – of the network and its width: the number of units at each level. This is as an easy and safe way of training higher quality models, especially given the availability of a large amount of labeled training data. However this simple solution comes with two major drawbacks.	模，包括通过增加层数以增大深度，通过增加每一层的节点数以增加宽度。这是训练高质量模型最简单安全的方法，特别是对于给定的大规模标签数据集。然而这种简单的解决方法有两大缺陷。
Bigger size typically means a larger number of parameters, which makes the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited. This can become a major bottleneck, since the creation of high quality training sets can be tricky and expensive, especially if expert human raters are necessary to distinguish between fine-grained visual categories like those in ImageNet (even in the 1000-class ILSVRC subset) as demonstrated by Figure 1.	更大的网络规模往往意味着更多的参数，这使得扩大后的网络更易过拟合，特别是当训练集中的标签样例有限的时候。这能够变成一个主要的瓶颈，因为制作高质量的训练集是要技巧的，也是很昂贵的，特别是人类专家对于类别力度的准确把握对于 ImageNet 这样的数据集而言是很重要的（即使是 ILSVRC 的 1000 类子集），如图一所示。
 (a) Siberian husky (b) Eskimo dog Figure 1: Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge.	
Another drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep	另一个统一增加网络大小的缺陷是计算资源需求的暴增。例如，在一个深度视觉网络，如果两个卷积层相连，任何增加过滤器数量的改动都会导

vision network, if two convolutional layers are chained, any uniform increase in the number of their filters results in a quadratic increase of computation. If the added capacity is used inefficiently (for example, if most weights end up to be close to zero), then a lot of computation is wasted. Since in practice the computational budget is always finite, an efficient distribution of computing resources is preferred to an indiscriminate increase of size, even when the main objective is to increase the quality of results.	致增加二次方倍数的计算量。如果增加的计算力没有被有效使用（比如大部分的权值趋于 0），那么大量的计算会被浪费。实际应用中可用的计算力是有限的，即使是以提高模型质量为主要目标，高效分布计算资源，其实也比盲目增加网络体积更加有效。
The fundamental way of solving both issues would be by ultimately moving from fully connected to sparsely connected architectures, even inside the convolutions. Besides mimicking biological systems, this would also have the advantage of firmer theoretical underpinnings due to the groundbreaking work of Arora et al. [2]. Their main result states that if the probability distribution of the data-set is representable by a large, very sparse deep neural network, then the optimal network topology can be constructed layer by layer by analyzing the correlation statistics of the activations of the last layer and clustering neurons with highly correlated outputs. Although the strict mathematical proof requires very strong conditions, the fact that this statement resonates with the well known Hebbian principle – neurons that fire together, wire together – suggests that the underlying idea is applicable even under less strict conditions, in practice.	解决这两个问题的基本方法最终一般是把全连接改成稀疏连接的结构，甚至包括在卷积中也这么做。除了模拟生物系统，根据 Arora【2】的突破性研究证明，这样做也可以在理论上获得更强健的系统。 Arora 等人的主要结果显示如果数据集的概率分布是一个十分稀疏的大型神经网络所能表达的，那么最合适的网络拓扑结构可以通过分析每一步的最后一层激活函数的统计关联性，并将具有高相关性输出的神经元进行聚类，而将网络一层一层地搭建起来。 虽然严格的数学证明需要很强的条件，但事实上这种情况符合著名的赫布原则——神经元如果激活条件相同，它们会彼此互联——这意味着在实践中，赫布原则在不那么严苛的条件下还是可以使用。【转自百度百科：Hebb 学习规则是一个无监督学习规则，这种学习的结果是使网络能够提取训练集的统计特性，从而把输入信息按照它们的相似性程度划分为若干类。这一点与人类观察和认识世界的过程非常吻合，人类观察和认识世界在相当程度上就是在根据事物的统计特征进行分类。Hebb 学习规则只根据神经元连接间的激活

	水平改变权值，因此这种方法又称为相关学习或并联学习。】
On the downside, todays computing infrastructures are very inefficient when it comes to numerical calculation on non-uniform sparse data structures. Even if the number of arithmetic operations is reduced by 100×, the overhead of lookups and cache misses is so dominant that switching to sparse matrices would not pay off. The gap is widened even further by the use of steadily improving, highly tuned, numerical libraries that allow for extremely fast dense matrix multiplication, exploiting the minute details of the underlying CPU or GPU hardware [16, 9]. Also, non-uniform sparse models require more sophisticated engineering and computing infrastructure. Most current vision oriented machine learning systems utilize sparsity in the spatial domain just by the virtue of employing convolutions. However, convolutions are implemented as collections of dense connections to the patches in the earlier layer. ConvNets have traditionally used random and sparse connection tables in the feature dimensions since [11] in order to break the symmetry and improve learning, the trend changed back to full connections with [9] in order to better optimize parallel computing. The uniformity of the structure and a large number of filters and greater batch size allow for utilizing efficient dense computation.	从负面而言，当涉及大量非统一的（non-uniform）稀疏的数据结构的计算时，现在的计算设施是很低效的。即使算术运算量降低 100 倍，查表运算和缓存失准（cache miss）也依然是主要瓶颈以至于稀疏矩阵的处理无法成功。如果使用稳定改进（steadily improving）、高度调制（highly tuned）、拥有大量库函数支持极快速密集矩阵相乘、关注 CPU 或 GPU 底层细节的方法，那么这种计算需求与计算资源之间的鸿沟甚至可能被进一步拉大。 另外，非统一（non-uniform？异构？？？？）的稀疏模型需要复杂的工程结构与计算结构。目前大部分面向机器学习的系统都利用卷积的优势在空间域中使用稀疏性。然而，卷积是通过一系列与前层区块的密集连接来实现的，文献【11】发表后，卷积神经网通常在特征维度中使用随机的稀疏的连接表，以打破对称性，提高学习水平，然而，根据文献【9】这种趋势会倒退回全连接模式，以便更好滴使用并行计算。 统一的结构、巨大的过滤器数量和更大的批次（batch）规模将允许使用高效的密集矩阵运算。
This raises the question whether there is any hope for a next, intermediate step: an architecture that makes use of the extra sparsity, even at filter level, as suggested by the theory, but exploits our current hardware by utilizing	这就导致了一个问题，是不是存在一个中间步骤，如同理论上所显示的，能够让整个结构即使在过滤器层面上都能使用额外的稀疏性，但依旧是利用现有硬件进行密集矩阵计算【an architecture that makes use of the extra sparsity, even at filter

computations on dense matrices. The vast literature on sparse matrix computations (e.g. [3]) suggests that clustering sparse matrices into relatively dense submatrices tends to give state of the art practical performance for sparse matrix multiplication. It does not seem far-fetched to think that similar methods would be utilized for the automated construction of non-uniform deep-learning architectures in the near future.	level, as suggested by the theory, but exploits our current hardware by utilizing computations on dense matrices】。大量关于稀疏矩阵计算的文献，比如文献【3】，都显示将稀疏矩阵聚类到相对密集的子矩阵上能够让稀疏矩阵相乘的性能达到实用水平，把同样的方法应用到自动构建非统一深度学习结构上，在不远的将来看起来并不过分。
The Inception architecture started out as a case study of the first author for assessing the hypothetical output of a sophisticated network topology construction algorithm that tries to approximate a sparse structure implied by [2] for vision networks and covering the hypothesized outcome by dense, readily available components. Despite being a highly speculative undertaking, only after two iterations on the exact choice of topology, we could already see modest gains against the reference architecture based on [12]. After further tuning of learning rate, hyperparameters and improved training methodology, we established that the resulting Inception architecture was especially useful in the context of localization and object detection as the base network for [6] and [5]. Interestingly, while most of the original architectural choices have been questioned and tested thoroughly, they turned out to be at least locally optimal.	Inception 的体系结构始于第一作者研究的一个例子——评估复杂拓扑结构的网络算法的假设输出，尝试近似地用一个密集的可获得的组件表示一个文献【2】提出的视觉网络的稀疏结构的假设输出。 然而这项工作在很大程度上是基于假设进行的，仅仅在两次迭代之后，我们就已经能够看到一些对于选定的拓扑结构非常不利的有限的成果【12】。在调节了学习速率、超系数，和采用了更好的训练方法之后，我们成功地建立了 Inception 的体系结构，使之能够在基于文献【5】和【6】提出的局部化和物体检测的上下文环境中非常好用。有趣的是，大多数最初的结构都被彻底地检测过，它们都至少能够达到局部最优。
One must be cautious though: although the proposed architecture has become a success for computer vision, it is still questionable whether its quality can be attributed to the guiding principles that have lead to its construction. Making sure would require much more thorough analysis and verification: for example, if	然而还是需要被谨慎考虑的是：虽然我们提出的体系结构在计算机视觉方面的应用很成功，但这能否归功于其背后的设计指导原则还不是很确定。 想要确定这一点还需要更加彻底的分析和验证：比如，基于这些规则的自动化工具是否能够找到

automated tools based on the principles described below would find similar, but better topology for the vision networks. The most convincing proof would be if an automated system would create network topologies resulting in similar gains in other domains using the same algorithm but with very differently looking global architecture. At very least, the initial success of the Inception architecture yields firm motivation for exciting future work in this direction.	与之类似但却更好的网络拓扑结构。最有说服力的证据将会是自动化系统能够利用相同的算法在不同的领域创建出具有相似结果，但整体架构有很大不同的网络拓扑。 最后，Inception 最初的成功为探索这一领域让人激动的未来产生了巨大的动力。
4 Architectural Details The main idea of the Inception architecture is based on finding out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components. Note that assuming translation invariance means that our network will be built from convolutional building blocks. All we need is to find the optimal local construction and to repeat it spatially. Arora et al. [2] suggests a layer-by layer construction in which one should analyze the correlation statistics of the last layer and cluster them into groups of units with high correlation. These clusters form the units of the next layer and are connected to the units in the previous layer. We assume that each unit from the earlier layer corresponds to some region of the input image and these units are grouped into filter banks. In the lower layers (the ones close to the input) correlated units would concentrate in local regions. This means, we would end up with a lot of clusters concentrated in a single region and they can be covered by a layer of 1×1 convolutions in the next layer, as suggested in	**4 结构细节** Inception 的体系结构的主要设计思路是要在一个卷积视觉网络中寻找一个局部最优的稀疏结构，这个结构需要能够被可获得的密集组件（dense component）覆盖和近似表达。 请注意，假定转义的不变性（translation invariance）意味着我们的网络将利用卷积砌块（convolutional building blocks）建立。我们所需要做的只是寻找局部最优化结构并在空间上对其进行重复。 Arora 等人在文献【2】中提出，一个逐层搭建的结构，需要分析其每一步的最后一层的统计关联性，并将高度相关的神经单元聚类为簇。 这些簇组成了下一层的单元并与前一层的各个单元相连。 我们假设前面一层的每个单元都对应输入图像的某些区域，而这些单元被分组分配给过滤器。在较低的层次（更靠近输入端），相关的单元聚焦于局部区域。这意味着我们能够得到大量聚焦于同一区域的簇，它们会被下一层的 1×1 卷积覆盖，如同文献【12】所述。

[12]. However, one can also expect that there will be a smaller number of more spatially spread out clusters that can be covered by convolutions over larger patches, and there will be a decreasing number of patches over larger and larger regions. In order to avoid patch alignment issues, current incarnations of the Inception architecture are restricted to filter sizes 1×1, 3×3 and 5×5, however this decision was based more on convenience rather than necessity. It also means that the suggested architecture is a combination of all those layers with their output filter banks concatenated into a single output vector forming the input of the next stage. Additionally, since pooling operations have been essential for the success in current state of the art convolutional networks, it suggests that adding an alternative parallel pooling path in each such stage should have additional beneficial effect, too (see Figure 2(a)).	然而，更少的在空间上传播更多的簇（a smaller number of more spatially spread out clusters）（这些簇会被区块更大的卷积所覆盖）是可以被期待的，这样的话，覆盖大型区域的区块数量就会减少。为了避免区块对齐问题（patch alignment issues），现有的 Inception 结构将过滤器大小限制为 1×1，3×3 和 5×5，然而这种设定更多是为了方便而不是必要的。 这也意味着合理的网络结构应该是将层次的输出过滤器 bank 结合起来，并将其合并为单一向量作为输出以及下一层的输入。 另外，因为池化操作对于现有水平的卷积网络是很重要的，建议最好在每一部增加一条并行池化通路，这样应该也会有一些额外的好处：如图 2a 所示。
 (a) Inception module, naïve version　　(b) Inception module with dimension reductions Figure 2: Inception module	
As these “Inception modules” are stacked on top of each other, their output correlation statistics are bound to vary: as features of higher abstraction are captured by higher layers, their spatial concentration is expected to decrease suggesting that the ratio of 3×3 and 5×5 convolutions should increase as we move to	Inception 模块是一层一层往上栈式堆叠的，所以它们输出的关联性统计会产生变化：更高层抽象的特征会由更高层次所捕获，而它们的空间聚集度会随之降低，因为随着层次的升高，3×3 和 5×5 的卷积的比例也会随之升高。

higher layers.	
One big problem with the above modules, at least in this naive form, is that even a modest number of 5 × 5 convolutions can be prohibitively expensive on top of a convolutional layer with a large number of filters. This problem becomes even more pronounced once pooling units are added to the mix: their number of output filters equals to the number of filters in the previous stage. The merging of the output of the pooling layer with the outputs of convolutional layers would lead to an inevitable increase in the number of outputs from stage to stage. Even while this architecture might cover the optimal sparse structure, it would do it very inefficiently, leading to a computational blow up within a few stages.	一个大问题是，上述模型，至少是朴素形式（naive form）的模型，即使只有很有限个数的 5×5 卷积，其最上层卷积层的巨量过滤器的开支都会让人望而却步。一旦把池化层加进来，这个问题会变得更加严重： 它们的输出过滤器个数与前面过程的过滤器个数相等。池化层输出与卷积层输出的合并会导致无法避免的每步输出暴增。 即使是当这种结构覆盖了最优的稀疏结构，它可能依然还是很低效，从而导致少数几步的计算量就会爆炸式增长。
This leads to the second idea of the proposed architecture: judiciously applying dimension reductions and projections wherever the computational requirements would increase too much otherwise. This is based on the success of embeddings: even low dimensional embeddings might contain a lot of information about a relatively large image patch. However, embeddings represent information in a dense, compressed form and compressed information is harder to model. We would like to keep our representation sparse at most places (as required by the conditions of [2]) and compress the signals only whenever they have to be aggregated en masse. That is, 1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions. Besides being used as reductions, they also include the use of rectified linear activation which makes them dual-purpose.	这种情况导致我们提出了第二种设想：审慎地把降维和投影使用到所有计算量可能急剧增加的地方。 这是基于嵌入的成功（success of embeddings）来设计的：相对于一个大型的图像区块，即使是低维的嵌入也可能包含大量的信息。 然而，嵌入会把信息以一种致密的，压缩的方式展现出来，而压缩信息是很难被建模的。 我们还是想在大部分位置保持稀疏性（如同文献【2】所要求的），而只在信号需要被聚合的时候压缩它们。 也就是说，1×1 卷积被用于在昂贵的 3×3 和 5×5 卷积之前降维。 除了用于降维，它们也被用于数据线性修正激活（rectified linear activation），这使之具有双重使命。最后的结果如图 2b。

The final result is depicted in Figure 2(b).	
In general, an Inception network is a network consisting of modules of the above type stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. For technical reasons (memory efficiency during training), it seemed beneficial to start using Inception modules only at higher layers while keeping the lower layers in traditional convolutional fashion. This is not strictly necessary, simply reflecting some infrastructural inefficiencies in our current implementation.	一般而言，一个 Inception 网络是由一系列上述结构栈式堆叠而成，有时候步长为 2 的最大池化层会把网络一分为二。 出于技术原因（更高效的训练），只在高层使用 Inception 结构而把低层保留为传统的卷积模式似乎是有利的。 这并不一定是必要的，只是反映了有些基础设施对于我们的设计而言很低效。
One of the main beneficial aspects of this architecture is that it allows for increasing the number of units at each stage significantly without an uncontrolled blow-up in computational complexity. The ubiquitous use of dimension reduction allows for shielding the large number of input filters of the last stage to the next layer, first reducing their dimension before convolving over them with a large patch size. Another practically useful aspect of this design is that it aligns with the intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from different scales simultaneously.	这一结构一个有利的方面是它允许每一步的神经元大量增加，而不会导致计算复杂度的暴增。 降维的普遍存在能够阻挡大量来自上一层的数据涌入下一层的过滤器，在大区块上对其进行卷积之前就对其进行降维。 该设计另一个在实践中很有用的方面是，它与【视觉信息应该被多层次处理，然后被汇集到下面层次汇总，同时抽取多尺度特征】的特性相一致。
The improved use of computational resources allows for increasing both the width of each stage as well as the number of stages without getting into computational difficulties. Another way to utilize the inception architecture is to create slightly inferior, but computationally cheaper versions of it. We have found that all the included the knobs and levers allow for a controlled balancing of computational resources that can	计算资源的优化利用允许我们增加每层网络的宽度以及层数，而无需面对增加的计算困难。 另一种使用 Inception 架构的方法是开发一种质量稍差，但计算起来更便宜的版本。 我们已经发现，用于平衡计算资源的控制因素 可以使得我们的网络比表现相同（译者注：这里可能是指精确度）而不使用 Inception 结构的网络快

result in networks that are 2~3× faster than similarly performing networks with non-Inception architecture, however this requires careful manual design at this point.

2~3 倍，只是这需要极为精细的人工调整。

5 GoogLeNet

We chose GoogLeNet as our team-name in the ILSVRC14 competition. This name is an homage to Yann LeCuns pioneering LeNet 5 network [10]. We also use GoogLeNet to refer to the particular incarnation of the Inception architecture used in our submission for the competition. We have also used a deeper and wider Inception network, the quality of which was slightly inferior, but adding it to the ensemble seemed to improve the results marginally. We omit the details of that network, since our experiments have shown that the influence of the exact architectural parameters is relatively minor. Here, the most successful particular instance (named GoogLeNet) is described in Table 1 for demonstrational purposes. The exact same topology (trained with different sampling methods) was used for 6 out of the 7 models in our ensemble.

5 GoogLeNet

我们选择 GoogLeNet 作为我们参加 ILSVRC14 比赛的队名。这个名字是为了纪念先驱者 Yann LeCuns 开发的 LeNet5 网络【10】。

我们也是用 GoogLeNet 作为我们在比赛中提交的 Inception 结构的具体实现的名字。

我们使用了一个更深、更宽的 Inception 网，其质量稍差，但如果把它进行合理搭配，会稍微改进其表现。

我们忽略了网络的实现细节，因为我们的实验表明，特定的某一结构参数的影响相对而言是很微小的。

在此，最成功的实现实例 GoogLeNet 是如表 1 所示的情况。一模一样的拓扑结构（用不同样例训练）在我们七分之六的合成模型中得到了应用。

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								

Table 1: GoogLeNet incarnation of the Inception architecture

All the convolutions, including those inside the Inception modules, use rectified linear activation. The size of the receptive field in our network is 224×224 taking RGB color channels with mean subtraction. "#3×3 reduce" and "#5×5 reduce" stands for the number of 1×1 filters in the reduction layer used before the 3×3 and 5×5 convolutions. One can see the number of 1×1 filters in the projection layer after the built-in max-pooling in the pool proj column. All these reduction/projection layers use rectified linear activation as well.	所有的卷积，包括那些 Inception 模块内的卷积，都使用修正线性激活函数（rectified linear activation）。我们网络的感知域是一个 RGB 三色通道的 224×224 区域，并且经过了减去均值的处理。"#3×3"降维和"#5×5"降维是 1×1 过滤器的等量代换【？？stands for the number of 1×1 filters】，用于在进行 3×3 和 5×5 卷积之前进行降维。1×1 过滤器的数量可以在池化投影列（pool proj column）中的最大池化层后面的投影层中看到。所有的降维层和投影层也都使用修正线性激活函数（rectified linear activation）。
The network was designed with computational efficiency and practicality in mind, so that inference can be run on individual devices including even those with limited computational resources, especially with low-memory footprint. The network is 22 layers deep when counting only layers with parameters (or 27 layers if we also count pooling). The overall number of layers (independent building blocks) used for the construction of the network is about 100. However this number depends on the machine learning infrastructure system used. The use of average pooling before the classifier is based on [12], although our implementation differs in that we use an extra linear layer. This enables adapting and fine-tuning our networks for other label sets easily, but it is mostly convenience and we do not expect it to have a major effect. It was found that a move from fully connected layers to average pooling improved the top-1 accuracy by about 0.6%, however the use of dropout remained essential even after removing the fully connected layers.	网络的设计是基于计算的效率与可实践性展开的，因此其推演过程可以在单台设备上进行，即使这些设备的运算资源极其有限（尤其是内存极其有限的设备）。 如果只计算有参数的层，我们的网络有 22 层深（算上池化层有 27 层）。 由于构建网络的总层数（独立砌块）有将近 100 个。 然而，这一数量需要依靠机器学习的基础设施，用于分类器之前的平均池化层是基于文献【12】设计的，虽然我们的实现方式有点不同：我们使用了一个多出来的线性层（use an extra linear layer）。 这使得在其它标签数据集上调整我们的网络变得容易，但这主要是为了方便，我们并不指望会有什么大的影响。 我们发现，从全连接层到平均池化的移动【？a move from fully connected layers to average pooling】会让 TOP-1 准确度提高 0.6%，然而，DROPOUT 的使用依然很重要，即使去掉了全连接层。
Given the relatively large depth of the network,	对于相对更深的网络，穿过所有层次高效向后梯

the ability to propagate gradients back through all the layers in an effective manner was a concern. One interesting insight is that the strong performance of relatively shallower networks on this task suggests that the features produced by the layers in the middle of the network should be very discriminative. By adding auxiliary classifiers connected to these intermediate layers, we would expect to encourage discrimination in the lower stages in the classifier, increase the gradient signal that gets propagated back, and provide additional regularization. These classifiers take the form of smaller convolutional networks put on top of the output of the Inception (4a) and (4d) modules. During training, their loss gets added to the total loss of the network with a discount weight (the losses of the auxiliary classifiers were weighted by 0.3). At inference time, these auxiliary networks are discarded.	度传播的能力是很关键的。 一个有趣的理论是，在这项任务中，相对浅层的网络的强大性能表明网络中层所产生的特征是具有很好的区分度的。 通过增加一些与这些中间层相连的附加的分类器，我们可以期待在分类器的低层增加向后传播的梯度信号，同时增加更多的正则化。 这些分类器采用较小的卷积网络形式，被安置在 Inception（4a）和（4d）模块的输出的顶部。 在训练中，它们的偏差被折扣后加到总偏差中（附加分类器的偏差乘以 0.3）。在预测过程中，这些附加网络会被抛弃。
The exact structure of the extra network on the side, including the auxiliary classifier, is as follows: • An average pooling layer with 5x5 filter size and stride 3, resulting in an 4x4x512 output for the (4a), and 4x4x528 for the (4d) stage. • A 1x1 convolution with 128 filters for dimension reduction and rectified linear activation. • A fully connected layer with 1024 units and rectified linear activation. • A dropout layer with 70% ratio of dropped outputs. • A linear layer with softmax loss as the classifier (predicting the same 1000 classes as the main classifier, but removed at inference time).	附加网络的结构，包括附加分类器的结构如下： ● 一个平均池化层，过滤器为 5×5，步长为 3，在 4（a）得到一个 4x4x512 的输出，在 4（d）得到一个 4x4x528 的输出。 ● 一个 1x1 卷积，有 128 个过滤器，用于降维和规范化线性激活（dimension reduction and rectified linear activation）。 ● 一个拥有 1024 个单元和规范化线性激活的全连接层。 ● 一个会抛弃 70%输出的 DROPOUT 层。 ● 一个使用 softmax 偏差的线性层，这一层被用作分类器（与主分类器一样，它进行 1000 类分类，但在预测阶段，它会被抛弃）

A schematic view of the resulting network is depicted in Figure 3.	最后得到的整个网络的示意图如图三所示。

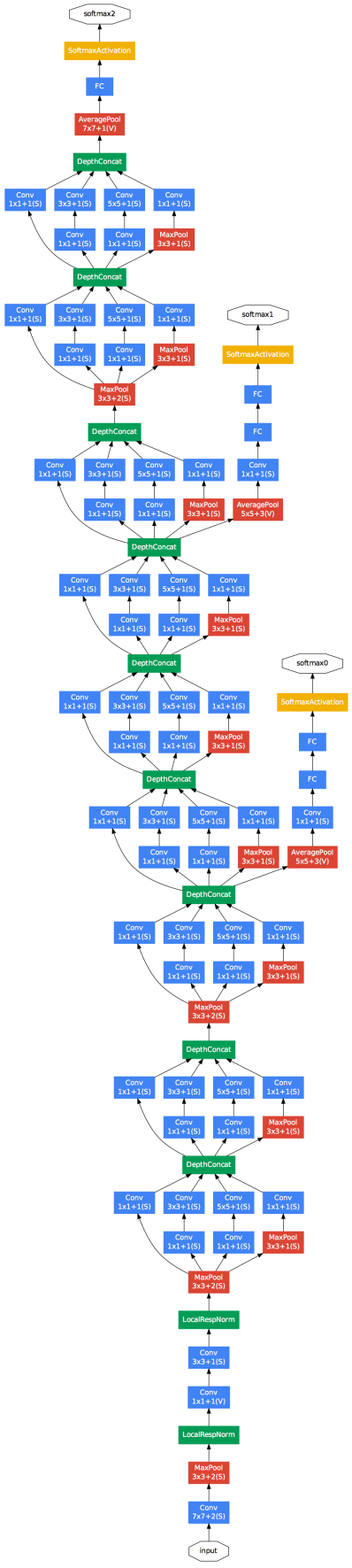


Figure 3: GoogLeNet network with all the bells and whistles

6 Training Methodology Our networks were trained using the DistBelief [4] distributed machine learning system using modest amount of model and data-parallelism. Although we used CPU based implementation only, a rough estimate suggests that the GoogLeNet network could be trained to convergence using few high-end GPUs within a week, the main limitation being the memory usage. Our training used asynchronous stochastic gradient descent with 0.9 momentum [17], fixed learning rate schedule (decreasing the learning rate by 4% every 8 epochs). Polyak averaging [13] was used to create the final model used at inference time.	**6 训练方法** 我们的网络使用文献【4】提出的分布置信网络，将机器学习系统分布为合适数量的模型和数据并行。 虽然我们只使用基于 CPU 的实现，一个粗略的估计证明 GoogLeNet 可以在少数几个高速 GPU 终端上进行训练并在一周内收敛，其主要限制是记忆体数量。 我们的训练使用动量（momentum）为 0.9 的异步随机梯度下降，并将学习速率固定为每八次迭代减少 0.04。Polyak 均值【13】被用于建立在推理过程中使用的最终模型
Our image sampling methods have changed substantially over the months leading to the competition, and already converged models were trained on with other options, sometimes in conjunction with changed hyperparameters, like dropout and learning rate, so it is hard to give a definitive guidance to the most effective single way to train these networks. To complicate matters further, some of the models were mainly trained on smaller relative crops, others on larger ones, inspired by [8]. Still, one prescription that was verified to work very well after the competition includes sampling of various sized patches of the image whose size is distributed evenly between 8% and 100% of the image area and whose aspect ratio is chosen randomly between 3/4 and 4/3. Also, we found that the photometric distortions by Andrew Howard [8] were useful to combat overfitting to some extent. In addition, we started to use random interpolation methods (bilinear, area, nearest	我们的图片采样方法在比赛前数个月就进行了彻底的修改，并在其他设置条件下通过了收敛测试——包括结合不同的超系数（比如 DROPOUT 率和学习速率），所以很难为【找到最高效的训练网络的方法】提供极为准确的指导。 更复杂的是，根据文献【8】的思路一些模型主要是在相对较小的粒度上进行训练，而另一些采用更大的粒度。 所以，一个在比赛之后已经被证明非常有效的方案是将取样区块的大小平均分布在图片区域的 8%到 100%之间，宽高比随机分布与 3/4 和 4/3 之间。 同时，我们发现 AH【8】提出的光度变换对于对抗过拟合在某种程度上是很有用的。 另外，我们开始的时候使用插入方法（等概率地使用双线性（bilinear 双曲线？）、区域、最近邻、

neighbor and cubic, with equal probability) for resizing relatively late and in conjunction with other hyperparameter changes, so we could not tell definitely whether the final results were affected positively by their use.	三次函数），以便在相对靠后的阶段重新确定取样大小，以及其他超系数的结合，所以我们无法明确知道这些方法的使用对于最后结果是不是真的有积极影响。
7 ILSVRC 2014 Classification Challenge Setup and Results The ILSVRC 2014 classification challenge involves the task of classifying the image into one of 1000 leaf-node categories in the Imagenet hierarchy. There are about 1.2 million images for training, 50,000 for validation and 100,000 images for testing. Each image is associated with one ground truth category, and performance is measured based on the highest scoring classifier predictions. Two numbers are usually reported: the top-1 accuracy rate, which compares the ground truth against the first predicted class, and the top-5 error rate, which compares the ground truth against the first 5 predicted classes: an image is deemed correctly classified if the ground truth is among the top-5, regardless of its rank in them. The challenge uses the top-5 error rate for ranking purposes.	**7 ILSVRC 2014 分类挑战的设置与结果** ILSVRC 2014 分类挑战包括将图片分类到 1000 个 ImageNet 层次结构的叶子节点类别中。 一共有 120 万张图片用于训练，5 万张图片用于验证，10 万张图片用于测试。 每张图片都与一个特定的类别相连，而性能则通过模型判断的可能性最高的类别是否合理进行检验。 两个指标被用于报告中：TOP-1 精确度——比较真实情况与预测认为可能性最高的情况；TOP-5 精确度——比较真实情况与预测认为可能性最高的前五种情况，一张图片的真实分类如果落入前五种预测分类之一，则视为分类正确，不考虑类别的排序位置。 挑战赛利用 TOP-5 错误进行排名。
We participated in the challenge with no external data used for training. In addition to the training techniques aforementioned in this paper, we adopted a set of techniques during testing to obtain a higher performance, which we elaborate below.	我们不利用任何附加数据参加这项挑战赛。 除了论文前述的训练技术，我们还采用了如下一系列测试技术去提高性能：
1. We independently trained 7 versions of the same GoogLeNet model (including one wider version), and performed ensemble prediction with them. These models were trained with the same initialization (even with the same initial weights, mainly because of an oversight) and	1，我们独立训练了七个版本的相同的 GoogLeNet 模型（包括一个宽度更大的版本） 然后将其联立起来进行预测。 这些模型训练基于相同的初始化（由于一个 oversight，甚至初始权值都是相同的）以及学习速率策略。

learning rate policies, and they only differ in sampling methodologies and the random order in which they see input images.	唯一的不同是采样方法和图片输入顺序不同。
2. During testing, we adopted a more aggressive cropping approach than that of Krizhevsky et al. [9]. Specifically, we resize the image to 4 scales where the shorter dimension (height or width) is 256, 288, 320 and 352 respectively, take the left, center and right square of these resized images (in the case of portrait images, we take the top, center and bottom squares). For each square, we then take the 4 corners and the center 224×224 crop as well as the square resized to 224×224, and their mirrored versions. This results in 4×3×6×2 = 144 crops per image. A similar approach was used by Andrew Howard [8] in the previous year's entry, which we empirically verified to perform slightly worse than the proposed scheme. We note that such aggressive cropping may not be necessary in real applications, as the benefit of more crops becomes marginal after a reasonable number of crops are present (as we will show later on).	2，在测试中，我们采取了比 Krizhevsky 等人【9】更大胆的裁切策略。特别地，我们将图片重设为四种不同的尺度（高和宽），分别是 256，288，320 和 352，包括左中右三块（如果说肖像图，我们取顶中底三块） 对于每一块，我们取其四角和中心，裁切出 5 个 224x224 的区块，同时取其镜像。 结果每张图就得到了 4×3×6×2 = 144 个区块。 同样的方法 AH【8】也在前些年的比赛中用了，根据我们的经验证明，其表现会比他们提出来的差一点。 我们注意到，如此激进的方法可能在实际应用中不是很有必要，因为当区块数超过合理范围之后，其带来的好处也就不那么重要了（我们后面会展示）。
3. The softmax probabilities are averaged over multiple crops and over all the individual classifiers to obtain the final prediction. In our experiments we analyzed alternative approaches on the validation data, such as max pooling over crops and averaging over classifiers, but they lead to inferior performance than the simple averaging.	3，softmax 概率分布被平均到不同的裁切以及所有的单分类器上以获取最终的预测结果。 在我们的试验中，我们在验证数据上分析了所有可选的方法，比如各个裁切区块上的最大池化，以及对分类器取平均。但它们会导致比简单平均更差的表现。

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

Table 2: Classification performance

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%

Table 3: GoogLeNet classification performance break down

In the remainder of this paper, we analyze the multiple factors that contribute to the overall performance of the final submission.	在余下的文章中，我们将分析各个因子在最终提交的作品中对性能的影响。
Our final submission in the challenge obtains a top-5 error of 6.67% on both the validation and testing data, ranking the first among other participants. This is a 56.5% relative reduction compared to the SuperVision approach in 2012, and about 40% relative reduction compared to the previous year's best approach (Clarifai), both of which used external data for training the classifiers. The following table shows the statistics of some of the top-performing approaches.	我们最后提交的挑战赛作品将 TOP-5 错误在验证集和测试集上都降到了 6.67%，在参赛者中排名第一。 与 2012 年的 SuperVision 方法相比，降低了 56.5%，与去年获得第一的 Clarifai 方法相比降低了 40%，而且这些方法都使用了外部数据来训练分类器。 如下表格展示了历年最优方法的统计数据。
We also analyze and report the performance of multiple testing choices, by varying the number of models and the number of crops used when predicting an image in the following table. When we use one model, we chose the one with the lowest top-1 error rate on the validation data. All numbers are reported on the validation dataset in order to not overfit to the testing data statistics.	我们还通过改变模型数量以及切分数量，分析并报告了其他几种测试策略对于图片进行预测的效果，结果见下表。 当我们使用一个模型，我们选择其在验证数据上的最低 TOP-1 错误率。 所有数据报告基于验证数据集,以避免测试集上的过拟合。
8 ILSVRC 2014 Detection Challenge Setup and	**8 ILSVRC 2014 识别挑战的设置与结果**

Results

The ILSVRC detection task is to produce bounding boxes around objects in images among 200 possible classes. Detected objects count as correct if they match the class of the groundtruth and their bounding boxes overlap by at least 50% (using the Jaccard index). Extraneous detections count as false positives and are penalized. Contrary to the classification task, each image may contain many objects or none, and their scale may vary from large to tiny. Results are reported using the mean average precision (mAP).

ILSVRC 的识别任务是在两百中可能类别上产生围绕物体的边界线（bounding boxes）。

如果边界线与事实重合至少 50%（使用交除以并的雅卡尔系数 Jaccard Index）则认为识别物体成功。

无关的识别将视为假正错误并遭受处罚。

与分类不同，每张图可能包含多个物体，也可能不包含任何物体，物体可大可小。结果报告采用平均精度（mAP）。

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

Table 4: Detection performance

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

Table 5: Single model performance for detection

The approach taken by GoogLeNet for detection is similar to the R-CNN by [6], but is augmented with the Inception model as the region classifier. Additionally, the region proposal step is improved by combining the Selective Search [20] approach with multi-box [5] predictions for higher object bounding box recall. In order to cut down the number of false positives, the

GoogLeNet 所采取的物体检测方法与文献【6】提出的 R-CNN 很类似，但因为在 Inception 模型中作为局部分类器使用而被放大了。

另外，为了获得更高的边界线召回率，通过将多边界预测【？multi-box predictions】【5】与选择性搜索（Selective Search）【20】相结合，区域提取的步骤【？the region proposal step】得到了改进。为了减少假正错误率，超像素的大小被扩大

superpixel size was increased by 2×. This halves the proposals coming from the selective search algorithm. We added back 200 region proposals coming from multi-box [5] resulting, in total, in about 60% of the proposals used by [6], while increasing the coverage from 92% to 93%. The overall effect of cutting the number of proposals with increased coverage is a 1% improvement of the mean average precision for the single model case. Finally, we use an ensemble of 6 ConvNets when classifying each region which improves results from 40% to 43.9% accuracy. Note that contrary to R-CNN, we did not use bounding box regression due to lack of time.	了两倍。这导致了选择搜索提取数量的减半【This halves the proposals coming from the selective search algorithm.】我们又把两百个多盒【5】提取区域加了回去，总共包括了文献【6】提出的60%，把覆盖率从92%提高到了93%。 利用增加覆盖率减少提取区域的总体效果是每个模型的平均精确度增加了1%。 最后，在分类每个区域的时候我们使用6个卷积神经网络的集合，从而将准确率从40%提高到了43.9%。请注意与R-CNN相比，限于时间，我们并未使用边界线回归（bounding box regression）
We first report the top detection results and show the progress since the first edition of the detection task. Compared to the 2013 result, the accuracy has almost doubled. The top performing teams all use Convolutional Networks. We report the official scores in Table 4 and common strategies for each team: the use of external data, ensemble models or contextual models. The external data is typically the ILSVRC12 classification data for pretraining a model that is later refined on the detection data. Some teams also mention the use of the localization data. Since a good portion of the localization task bounding boxes are not included in the detection dataset, one can pre-train a general bounding box regressor with this data the same way classification is used for pre-training. The GoogLeNet entry did not use the localization data for pre-training.	我们首先报告了可能性最高的检测结果，并从第一个版本的检测任务开始展示了整个过程。 与2013年的结果相比，准确率几乎翻了一倍。 系统性能最佳的队伍都使用了卷积神经网络。 我们在表4展示了官方分数以及相同的系统策略：是否使用外部数据、模型集成或是其他上下文模型。 外部数据主要是用ILSVRC12分类数据来进行预训练，然后再将模型限制在检测数据上。 一些队伍还提到了使用局部化数据。 因为适当比例的局部化任务的边界线并不包含在物体检测数据集中，可以预先将这些数据用到一个普适的边界线回归器上，用于最终预测相同的方式进行预训练。 GoogLeNet并不使用这种局部化数据进行预训练。
In Table 5, we compare results using a single model only. The top performing model is by Deep Insight and surprisingly only improves by 0.3	如表5，我们比较了使用不同单个模型的最终结果。 表现最好的是DeepInsight模型，让人惊讶的是，

points with an ensemble of 3 models while the GoogLeNet obtains significantly stronger results with the ensemble.	DeepInsight 使用三种模型的集成却只提高了 0.3 个点（的精度），而我们的模型集成后就要强大得多。
9 Conclusions Our results seem to yield a solid evidence that approximating the expected optimal sparse structure by readily available dense building blocks is a viable method for improving neural networks for computer vision. The main advantage of this method is a significant quality gain at a modest increase of computational requirements compared to shallower and less wide networks. Also note that our detection work was competitive despite of neither utilizing context nor performing bounding box regression and this fact provides further evidence of the strength of the Inception architecture. Although it is expected that similar quality of result can be achieved by much more expensive networks of similar depth and width, our approach yields solid evidence that moving to sparser architectures is feasible and useful idea in general. This suggest promising future work towards creating sparser and more refined structures in automated ways on the basis of [2].	**9 结论** 我们的结果似乎产生了一个坚实的结论——利用现有密集砌块逼近预想中的最佳稀疏结构，是一种可行的提高计算机视觉神经网络能力的方法。 这种模型的主要优势是与浅层且较窄的网络相比，只要适度增加计算需求就能极大地提升质量。 还请大家注意，我们的检测技术即使没有使用上下文和边界回归，依然很有竞争力，这一事实提供了进一步的证据证明 Inception 结构的强大。 虽然相同质量的网络可以被同样宽度和深度的更昂贵的网络实现，我们的方法却切实地证明了切换到一个更稀疏的结构上是一个在普遍情况下可行且有用的方法。 这意味着一个充满希望的未来——开发文献【2】提出的自动创建一个更稀疏，更有限的结构的方法。