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.

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.

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 $12\times$ fewer parameters than the winning 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].

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.

深入卷积网络

Christian Szegedy Google Inc. Wei Liu 北卡罗来纳大学教堂山分校 Yangqing Jia Google Inc. Pierre Sermanet Google Inc. Scott Reed 密歇根大学 Dragomir Anguelov Google Inc. Dumitru Erhan Google Inc. Vincent Vanhoucke Google Inc. Andrew Rabinovich Google Inc.

摘要

我们提出了一种代号为Inception的深度卷积神经网络架构，该架构在2014年ImageNet大规模视觉识别挑战赛（ILSVRC14）中为分类和检测任务创造了新的技术标杆。该架构的主要特点在于提升网络内部计算资源的利用率。这是通过精心设计实现的，使得网络在保持计算量不变的前提下，能够增加深度和宽度。为了优化性能，架构设计基于赫布原理和多尺度处理的直观理念。我们在ILSVRC14中提交的一个具体实现名为GoogLeNet，这是一个22层的深度网络，其性能在分类和检测任务中得到了验证。

1 引言

在过去的三年中，主要由于深度学习的进步，更具体地说是卷积网络[10]的发展，图像识别和目标检测的质量以惊人的速度提升。一个令人鼓舞的消息是，这种进步大多不仅仅是更强大的硬件、更大的数据集和更大的模型的结果，而主要是新思想、算法和改进的网络架构的产物。例如，在ILSVRC 2014竞赛中，除了用于检测目的的同一竞赛分类数据集外，顶级参赛作品并未使用新的数据源。我们在ILSVRC 2014中提交的GoogLeNet实际上比两年前Krizhevsky等人[9]的获奖架构少用了12{v*}个参数，同时显著提高了准确性。目标检测方面的最大收益并非仅仅来自深度网络或更大模型的使用，而是来自深度架构与经典计算机视觉的协同作用，例如Girshick等人[6]提出的R-CNN算法。

另一个值得注意的因素是，随着移动和嵌入式计算的持续发展，我们算法的效率——尤其是其功耗和内存使用——变得愈发重要。值得注意的是，本文提出的深度架构在设计过程中考虑了这一因素，而非仅仅执着于准确率数字。在大多数实验中，模型的设计目标是在推理阶段保持15亿次乘加运算的计算预算，以确保它们不会沦为纯粹的学术探索，而是能够在合理成本下应用于现实世界，即使是处理大规模数据集也不例外。

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 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.

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.

Network-in-Network is an approach proposed by Lin et al. [12] in order to increase the 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.

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.

3 Motivation and High Level Considerations

The most straightforward way of improving the 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.

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

本文聚焦于一种高效的计算机视觉深度神经网络架构，代号为Inception。该名称源于Lin等人[12]提出的"网络中的网络"论文，并结合了著名的网络流行语"我们需要更深入"[1]。此处"深度"一词具有双重含义：首先指我们通过"Inception模块"形式引入了新的组织层级，同时也更直接地指网络深度的增加。总体而言，可将Inception模型视为[12]研究的逻辑终局，并受到Arora等人[2]理论工作的启发与指导。该架构的优势在ILSVRC 2014分类与检测挑战赛中得到实验验证，其性能显著超越了当时的最先进水平。

2 相关工作

从LeNet-5 [10]开始，卷积神经网络（CNN）通常具有标准结构——堆叠的卷积层（可选地后接对比度归一化和最大池化）之后是一个或多个全连接层。这种基础设计的变体在图像分类文献中十分普遍，并在MNIST、CIFAR数据集上取得了迄今最佳结果，尤其是在ImageNet分类挑战赛上表现突出[9, 21]。对于更大规模的数据集（如ImageNet），近期的趋势是增加网络层数[12]和层尺寸[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精确空间信息的丢失，但[9]中采用的相同卷积网络架构已成功应用于定位[9, 14]、目标检测[6, 14, 18, 5]以及人体姿态估计[19]。受灵长类视觉皮层神经科学模型的启发，Serre等人[15]采用了一系列不同尺寸的固定Gabor滤波器来处理多尺度问题，这与Inception模型类似。然而，与[15]中固定的双层深度模型不同，Inception模型中的所有滤波器都是通过学习得到的。此外，Inception层被多次重复堆叠，在GoogLeNet模型中形成了22层的深度架构。

Network-in-Network 是由 Lin 等人 [12] 提出的一种方法，旨在增强神经网络的表示能力。当应用于卷积层时，该方法可被视为添加额外的 1×1 卷积层，通常后接线性整流激活函数 [9]。这使得它能够轻松集成到当前的 CNN 流程中。我们在架构中大量使用了这种方法。然而，在我们的设置中，1×1 卷积具有双重作用：最关键的是，它们主要被用作降维模块，以消除计算瓶颈，否则这些瓶颈会限制我们网络的规模。这不仅允许我们增加网络的深度，还能增加网络的宽度，而不会造成显著的性能损失。

当前物体检测的主流方法是Girshick等人提出的基于区域的卷积神经网络（R-CNN）[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，以类别无关的方式生成潜在物体候选区域；随后使用CNN分类器对这些区域进行物体类别识别。这种两阶段方法既利用了低级线索在边界框分割上的准确性，又发挥了先进CNN强大的分类能力。我们在检测任务中采用了类似流程，但对两个阶段均进行了改进探索：例如采用多框预测[5]提升物体边界框召回率，并运用集成学习方法优化候选边界框的分类效果。

3 动机与高层考量

提升深度神经网络性能最直接的方法是增加其规模。这既包括增加网络的深度——即层数，也包括增加其宽度：即每层的单元数量。这是一种简单且安全的训练高质量模型的方法，尤其是在有大量标注训练数据可用的情况下。然而，这种简单的解决方案存在两个主要缺点。

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过拟合，尤其是在训练集中有标签样本数量有限的情况下。这可能成为一个主要瓶颈，因为创建高质量的训练集可能很棘手。

(a) Siberian husky (b) Eskimo dog

Figure 1: Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge.

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.

Another drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep 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.

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.

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 $100\times$, 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.

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

(a) Siberian husky (b) Eskimo dog

图1：ILSVRC 2014分类挑战赛1000个类别中的两个不同类别。

且成本高昂，尤其是在需要专家人工评估员来区分细粒度视觉类别（如图像网络中的那些类别，即使在1000类的ILSVRC子集中也是如此）时，如图1所示。

网络规模均匀增大的另一个缺点是计算资源的使用急剧增加。例如，在一个深度视觉网络中，如果两个卷积层串联，其滤波器数量的任何均匀增加都会导致计算量的二次增长。如果新增的容量被低效利用（例如，如果大多数权重最终接近于零），那么大量计算就被浪费了。由于实际计算预算总是有限的，即使主要目标是提高结果质量，也应优先考虑计算资源的有效分配，而非盲目扩大规模。

解决这两个问题的根本方法，最终在于从全连接架构转向稀疏连接架构，即使在卷积内部也是如此。除了模仿生物系统外，由于Arora等人[2]的开创性工作，这种方法还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通过一个大型、高度稀疏的深度神经网络表示，那么可以通过分析最后一层激活值的相关统计量，并将输出高度相关的神经元聚类，从而逐层构建出最优的网络拓扑结构。尽管严格的数学证明需要非常强的条件，但这一结论与著名的赫布原理——"同时激活的神经元会相互连接"——产生共鸣的事实表明，其核心思想在实践中即使在不那么严格的条件下也适用。

从不利的一面来看，当今的计算基础设施在处理非均匀稀疏数据结构的数值计算时效率极低。即使算术运算的数量减少了100×，查找和缓存未命中的开销仍然占据主导地位，以至于转向稀疏矩阵并不会带来回报。随着持续改进、高度优化的数值库的使用，这一差距进一步扩大，这些库能够实现极快的稠密矩阵乘法，并充分利用底层CPU或GPU硬件的细微特性[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。当前大多数面向视觉的机器学习系统仅通过采用卷积来利用空间域的稀疏性。然而，卷积是通过与前一层的图像块建立稠密连接集合来实现的。自[11]以来，卷积神经网络传统上在特征维度使用随机和稀疏的连接表，以打破对称性并改善学习效果；但为了更好优化并行计算，[9]之后趋势又回归到全连接。结构的均匀性、大量的滤波器以及更大的批处理规模，使得高效的稠密计算得以充分利用。

这就引出了一个问题：是否存在一种有希望的中间步骤：一种架构，它能够利用理论所建议的额外稀疏性，甚至在滤波器级别上，同时充分利用我们的

current hardware by utilizing 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.

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.

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 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.

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 [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)).

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 higher layers.

One big problem with the above modules, at least in this naïve 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

当前硬件通过利用密集矩阵计算来提升性能。关于稀疏矩阵计算的大量文献（例如[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。可以合理推测，类似方法在不久的将来将被用于非均匀深度学习架构的自动化构建中。

Inception架构最初是作为第一作者的一项案例研究，旨在评估一种复杂网络拓扑构建算法的假设输出。该算法试图逼近[2]中为视觉网络所暗示的稀疏结构，并通过密集且易于获取的组件来覆盖这一假设结果。尽管这是一项高度推测性的尝试，但在仅对拓扑结构进行两次精确选择的迭代后，我们已经能看到相对于基于[12]的参考架构所取得的适度收益。经过对学习率、超参数及改进训练方法的进一步调优，我们确认了由此产生的Inception架构在定位和物体检测背景下作为[6]和[5]的基础网络尤为有效。有趣的是，尽管最初的大部分架构选择曾受到质疑并经过彻底测试，但它们最终被证明至少是局部最优的。

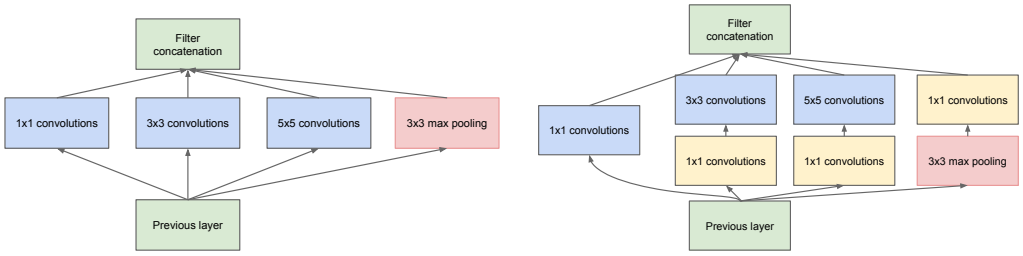
不过，我们必须谨慎：尽管所提出的架构在计算机视觉领域取得了成功，但其质量是否可归功于指导其构建的原则仍存疑问。要确证这一点，需要更深入的分析和验证：例如，基于下述原则的自动化工具能否为视觉网络找到类似但更优的拓扑结构。最有力的证明将是，一个自动化系统能够使用相同算法，在其他领域创建出带来类似性能提升的网络拓扑，即使其全局架构外观迥异。至少，Inception架构的初步成功为这一方向的未来激动人心的工作提供了坚实的动力。

4 架构细节

Inception架构的核心思想在于探索如何利用现成的密集组件来近似并覆盖卷积视觉网络中最优的局部稀疏结构。需注意，平移不变性的假设意味着我们的网络将由卷积构建模块组成。我们只需找到最优的局部结构并在空间上重复它。Arora等人[2]提出了一种逐层构建方法：分析前一层的相关性统计量，将高相关性的单元聚类成组。这些聚类将形成下一层的单元，并与前一层的单元相连接。我们假设早期层的每个单元对应输入图像的某个区域，这些单元被分组为滤波器组。在靠近输入的底层网络中，相关单元会集中在局部区域。这意味着我们将得到大量集中于单个区域的聚类，它们可以通过下一层的1×1卷积层来覆盖，正如文献[12]所建议的。然而，我们也可以预期会出现数量较少、空间分布更广的聚类，这些聚类需要更大尺寸的卷积核来覆盖，且随着区域增大，所需卷积核数量会递减。为避免图像块对齐问题，当前Inception架构的实现将滤波器尺寸限制为1×1、3×3和5×5，但这更多是出于便利性而非必要性考量。该架构实质上是将各卷积层的输出滤波器组拼接为单一输出向量，作为下一阶段的输入。此外，由于池化操作对当前先进卷积网络的成功至关重要，在每一阶段增加并行的池化路径也应能带来额外增益（见图2(a)）。

随着这些“初始模块”层层叠加，它们的输出相关统计量必然发生变化：当更高层次捕捉到更高抽象的特征时，其空间集中度预计会降低，这表明随着向更高层级移动，3×3和5×5卷积的比例应当增加。

上述模块的一个大问题，至少在这种天真的形式下，是即使数量不多的5×5卷积，在具有大量滤波器的卷积层之上也可能成本过高。一旦将池化单元加入其中，这个问题变得更加明显：它们的输出滤波器数量等于前一阶段的滤波器数量。将池化层的输出与卷积层的输出合并，将不可避免地



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

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.

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. The final result is depicted in Figure 2(b).

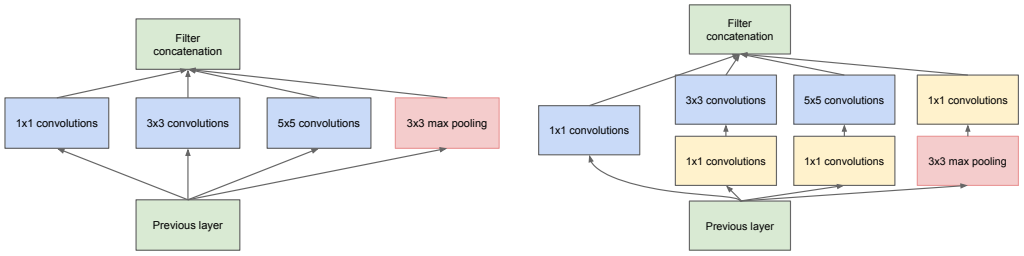
In general, an Inception network is a network consisting of modules of the above type stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. For technical reasons (memory efficiency during training), it seemed beneficial to start using Inception 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.

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.

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

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



(a) Inception module, naïve version (b) Inception module with dimension reductions

图2：Inception模块

阶段间输出数量的增加。即使这种架构可能覆盖了最优稀疏结构，其实现效率也会非常低下，导致在少数几个阶段内计算量急剧膨胀。

这引出了所提出架构的第二个核心理念：在计算需求可能过度增长的地方，审慎地应用降维和投影操作。这一理念基于嵌入技术的成功实践：即使是低维嵌入，也可能包含相对较大图像块的大量信息。然而，嵌入以密集压缩的形式表示信息，而压缩信息往往更难建模。我们希望保持表征在多数位置具有稀疏性（如文献[2]所述条件的要求），仅当信号需要大规模聚合时才进行压缩。具体而言，在计算密集的3×3和5×5卷积之前，会使用1×1卷积执行降维处理。这些卷积不仅用于降维，还结合了整流线性激活函数，从而实现双重功能。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相互堆叠而成的网络，偶尔会使用步长为2的最大池化层来将网格分辨率减半。出于技术原因（训练期间的内存效率），似乎只在较高层开始使用Inception模块，而将较低层保持为传统的卷积形式更为有利。这并非绝对必要，只是反映了我们当前实现中的一些基础设施效率不足。

这种架构的主要优势之一在于，它允许在每个阶段显著增加单元数量，而不会导致计算复杂度失控性激增。通过普遍采用降维操作，能够将最后阶段的大量输入滤波器屏蔽到下一层——先降低其维度，再使用大尺寸卷积核进行卷积运算。该设计另一个具有实际价值的方面是：它符合视觉信息应在多尺度处理后再聚合的直觉，从而使下一阶段能同时从不同尺度中提取特征。

计算资源的优化使用使得在不陷入计算困境的情况下，既能增加每个阶段的宽度，也能增加阶段的数量。利用Inception架构的另一种方式是创建性能稍逊但计算成本更低的版本。我们发现，所有内置的调节机制允许对计算资源进行可控平衡，从而构建出比性能相当的非Inception架构网络快2–至3×倍的网络，但这目前仍需谨慎的人工设计。

5 GoogLeNet

在ILSVRC14竞赛中，我们选择GoogLeNet作为团队名称。此名旨在致敬Yann LeCun开创性的LeNet 5网络[10]。我们亦用GoogLeNet指代本次竞赛提交方案中采用的Inception架构具体实现。我们还使用了更深更广的Inception网络变体，其质量略逊一筹，但将其加入集成模型后，结果似乎有微弱提升。由于实验表明具体架构参数的影响相对有限，此处不再赘述该网络细节。

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

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.

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.

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.

Given the relatively large depth of the network, 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.

The exact structure of the extra network on the side, including the auxiliary classifier, is as follows:

- An average pooling layer with 5×5 filter size and stride 3, resulting in an $4\times4\times512$ output for the (4a), and $4\times4\times528$ for the (4d) stage.

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实现

次要。这里，为了演示目的，表1描述了最成功的特定实例（名为GoogLeNet）。在我们的集成模型中，有6个模型使用了完全相同的拓扑结构（采用不同的采样方法训练）。

所有卷积，包括Inception模块内部的卷积，均使用修正线性激活函数。我们网络中的感受野大小为224×224，采用经过均值减除处理的RGB色彩通道。“#3×3 reduce”和“#5×5 reduce”表示在3×3和5×5卷积之前使用的降维层中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。所有这些降维/投影层同样使用修正线性激活函数。

该网络在设计时充分考虑了计算效率和实用性，以便推理过程能够在包括计算资源有限的设备上运行，尤其是内存占用较低的情况。仅计算带参数的层时，网络深度为22层（若计入池化层则为27层）。用于构建网络的整体层数（独立构建模块）约为100层，但具体数值取决于所使用的机器学习基础设施系统。分类器前的平均池化操作借鉴了文献[12]，但我们的实现方式有所不同——我们增加了一个额外的线性层。这使得网络能够轻松适配和微调至其他标签集，虽然这主要是出于便利性考虑，我们预计其不会产生重大影响。实验发现，将全连接层替换为平均池化能使Top-1准确率提升约0.6%，但即使移除全连接层后，dropout技术的使用仍然至关重要。

考虑到网络深度相对较大，如何以有效方式将梯度反向传播至所有层是一个值得关注的问题。一个有趣的发现是：相对较浅的网络在此任务上表现优异，这表明网络中间层生成的特征应具备极强的区分性。通过在中间层添加辅助分类器，我们期望能够增强分类器低层阶段的判别能力，增加反向传播的梯度信号，并提供额外的正则化效果。这些分类器采用小型卷积网络的形式，附加在Inception（4a）和（4d）模块的输出端。训练过程中，其损失值会以折扣权重（辅助分类器损失权重设为0.3）加入网络总损失。在推理阶段，这些辅助网络将被移除。

的确切结构　　侧边的额外网络，包括辅助分类器　　如下：

- 一个平均池化层，使用5×5的滤波器尺寸和步幅3，为(4a)阶段生成4×4×512的输出，为(4d)阶段生成4×4×528的输出。

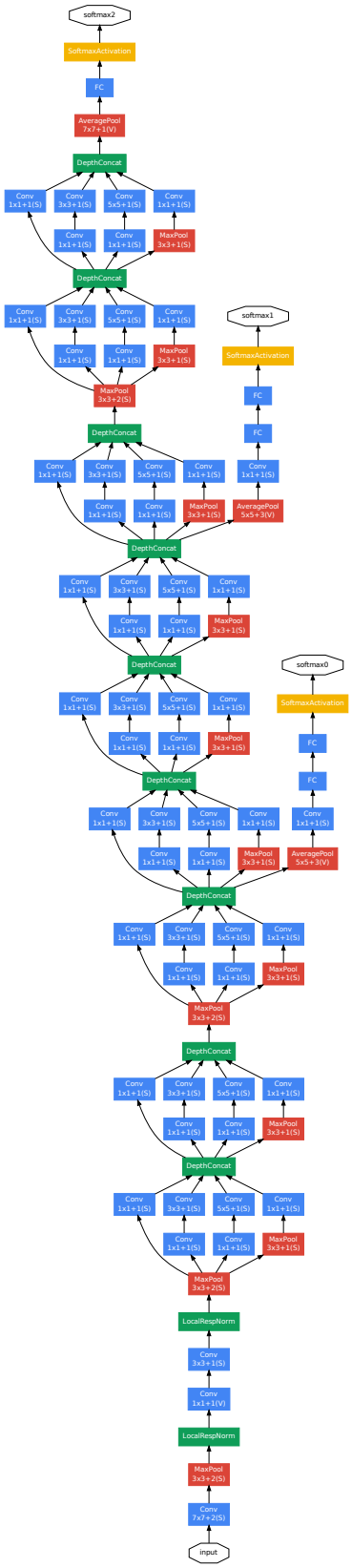


Figure 3: GoogLeNet network with all the bells and whistles

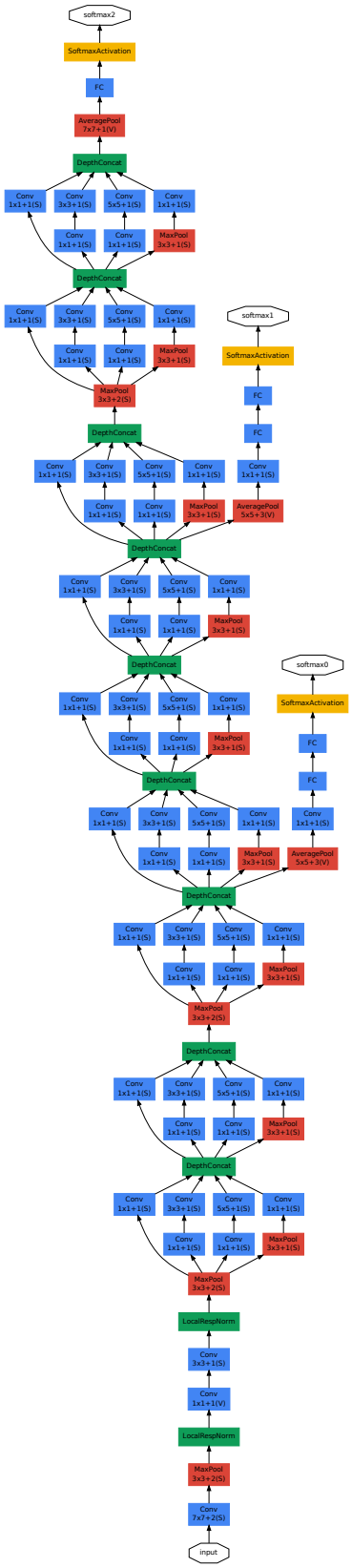


图3：配备所有附加功能的GoogLeNet网络

- A 1×1 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).

A schematic view of the resulting network is depicted in Figure 3.

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.

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 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.

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.

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.

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 learning rate policies, and they only differ in sampling methodologies and the random order in which they see input images.
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

- 一个具有128个滤波器用于降维和修正线性激活的1×1卷积。
- 一个具有1024个单元和修正线性激活的全连接层。
- 丢弃率为70%的dropout层。
- 一个带有softmax损失的线性层作为分类器（预测与主分类器相同的1000个类别，但在推理时被移除）。

所得网络的示意图如图 {v*} 所示。 关于第3点。

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，采用了适度的模型并行与数据并行策略。尽管我们仅使用了基于CPU的实现，但粗略估计表明，GoogLeNet网络可以在大约一周内通过少量高端GPU训练至收敛，主要限制在于内存使用量。我们的训练采用异步随机梯度下降法，动量为0.9 [17]，并采用固定的学习率调整策略（每8轮迭代将学习率降低4%）。在推理阶段使用的最终模型通过Polyak平均法 [13] 生成。

我们的图像采样方法在比赛前的几个月里发生了显著变化，已经收敛的模型曾使用其他选项进行训练，有时还结合了改变的超参数（如dropout和学习率），因此很难给出一个明确的指导来确定训练这些网络最有效的单一方法。更复杂的是，部分模型主要训练于相对较小的裁剪区域，另一些则基于较大的区域，这受到了[8]的启发。尽管如此，比赛后验证效果极佳的一种方案包括：采样不同尺寸的图像块，其大小均匀分布在图像面积的8%到100%之间，长宽比则在3/4和4/3之间随机选择。此外，我们发现Andrew Howard[8]提出的光度失真方法在一定程度上有助于对抗过拟合。另外，我们较晚才开始使用随机插值方法（双线性、区域、最近邻和立方插值，每种方法等概率选择）进行尺寸调整，并且与其他超参数更改结合使用，因此无法确定最终结果是否因其使用而获得了积极影响。

7 国际SVRC 2014 分类挑战赛设置与 R 结果

ILSVRC 2014分类挑战赛的任务是将图像分类到ImageNet层次结构中的1000个叶节点类别之一。该赛事提供约120万张训练图像、5万张验证图像和10万张测试图像。每张图像对应一个真实类别标签，评估标准基于分类器预测得分最高的类别。通常报告两个指标：top-1准确率（将真实类别与预测最高概率的类别比较）和top-5错误率（将真实类别与预测概率前五的类别比较：只要真实类别出现在前五预测中即视为正确分类，不要求具体排名）。本挑战赛采用top-5错误率作为排名依据。

我们参与挑战时未使用任何外部数据进行训练。除了本文前述的训练技术外，我们在测试阶段还采用了一系列技术以获取更高性能，具体内容如下详述。

1. 我们独立训练了7个相同版本的GoogLeNet模型（包括一个更宽的版本），并对它们进行了集成预测。这些模型采用相同的初始化（甚至使用相同的初始权重，主要是由于疏忽）和学习率策略，它们仅在采样方法以及输入图像的随机顺序上有所不同。 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，我们将图像缩放到4个尺度，其中较短边（高度或宽度）分别为256、288、320和352，取这些缩放后图像的左、中、右正方形区域（对于纵向图像，则取上、中、下正方形区域）。接着，对每个正方形区域，我们取4个角落和中心224{v*}224的裁剪区域，以及

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

square resized to 224×224, and their mirrored versions. This results in $4\times3\times6\times2 = 144$ 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).

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.

In the remainder of this paper, we analyze the multiple factors that contribute to the overall performance of the final submission.

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.

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.

8 ILSVRC 2014 Detection Challenge Setup and 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

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

将图像调整为224×224的正方形，并生成其镜像版本。这样每张图像可得到4×3×6×2 = 144个裁剪区域。Andrew Howard [8] 在去年的参赛方案中采用了类似方法，我们通过实验验证其效果略逊于本文提出的方案。需要指出的是，在实际应用中可能无需如此密集的裁剪，因为当裁剪数量达到合理范围后，继续增加裁剪区域带来的收益会逐渐递减（后续将对此进行说明）。

3. 通过对多个裁剪区域和所有独立分类器的softmax概率进行平均，以获得最终预测。在我们的实验中，我们在验证数据上分析了其他方法，例如对裁剪区域进行最大池化并对分类器进行平均，但这些方法的性能均低于简单的平均方法。

在本文的剩余部分，我们将分析影响最终提交整体性能的多种因素。

我们在挑战赛中的最终提交在验证和测试数据上均获得了6.67%的前五错误率，在所有参与者中排名第一。与2012年的SuperVision方法相比，这实现了56.5%的相对降低；与上一年度的最佳方法（Clarifai）相比，也实现了约40%的相对降低——这两种方法均使用了外部数据来训练分类器。下表展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，在下表中分析并报告了多种测试选择的性能。当使用单一模型时，我们选择在验证数据上 top-1 错误率最低的模型。为避免过度拟合测试数据统计特征，所有数据均在验证数据集上报告。

8 ILSVRC 2014 检测挑战赛设置与结果

ILSVRC检测任务是在图像中为200个可能类别中的物体生成边界框。若检测到的物体与真实类别一致，且其边界框重叠率至少达到50%（使用Jaccard指数），则视为正确检测。多余的检测会被视为误报并受到惩罚。与分类任务不同，每张图像可能包含

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

many objects or none, and their scale may vary from large to tiny. Results are reported using the mean average precision (mAP).

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 superpixel size was increased by $2\times$. 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.

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 pre-training 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 pretraining.

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 points with an ensemble of 3 models while the GoogLeNet obtains significantly stronger results with the ensemble.

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

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：检测任务的单模型性能

许多物体或没有物体，且它们的尺度可能从大到微小不等。结果使用平均精度均值（mAP）进行报告。

GoogLeNet采用的检测方法与[6]提出的R-CNN相似，但通过引入Inception模型作为区域分类器进行了增强。此外，区域建议步骤通过将选择性搜索[20]方法与multi-box[5]预测相结合得到改进，从而提高了目标边界框的召回率。为减少误报数量，超像素尺寸被增大了2×，这使得选择性搜索算法产生的建议区域减半。我们额外补充了200个来自multi-box[5]的区域建议，最终使用的建议区域总数约为[6]所用数量的60%，同时将覆盖率从92%提升至93%。在提高覆盖率的同时削减建议区域数量，使得单模型情况下的平均精度均值提升了1%。最后，在对每个区域进行分类时，我们采用了6个ConvNet组成的集成模型，将准确率从40%提升至43.9%。需要注意的是，由于时间限制，我们未像R-CNN那样使用边界框回归方法。

我们首先报告了最佳检测结果，并展示了自首届检测任务举办以来的进展。与2013年的结果相比，准确率几乎翻了一番。表现最佳的团队均采用了卷积网络。我们在表4中列出了官方评分及各团队的常用策略：使用外部数据、集成模型或上下文模型。外部数据通常指ILSVRC12分类数据，用于预训练模型，随后再基于检测数据进行微调。部分团队还提及使用了定位数据。由于定位任务中相当一部分边界框未包含在检测数据集中，研究者可借此数据预训练通用边界框回归器，其方式与使用分类数据进行预训练类似。GoogLeNet参赛方案未使用定位数据进行预训练。

在表5中，我们仅对比了使用单一模型的结果。表现最佳的模型来自Deep Insight，令人惊讶的是，其采用3个模型集成后仅提升了0.3分，而GoogLeNet通过集成方法获得了显著更强的结果。

9 结论

我们的结果似乎提供了坚实的证据，表明通过现成的密集构建模块来近似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。该方法的主要优势在于，与较浅且较窄的网络相比，它在计算需求适度增加的情况下实现了显著的质量提升。还需注意的是，尽管我们的检测工作既未利用上下文也未执行边界框回归，但仍具有竞争力。

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].

10 Acknowledgements

We would like to thank Sanjeev Arora and Aditya Bhaskara for fruitful discussions on [2]. Also we are indebted to the DistBelief [4] team for their support especially to Rajat Monga, Jon Shlens, Alex Krizhevsky, Jeff Dean, Ilya Sutskever and Andrea Frome. We would also like to thank to Tom Duerig and Ning Ye for their help on photometric distortions. Also our work would not have been possible without the support of Chuck Rosenberg and Hartwig Adam.

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] Ümit V. Çatalyürek, Cevdet Aykanat, and Bora Uçar. 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, Léon 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, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

回归，这一事实进一步证明了Inception架构的强大。尽管预期通过更昂贵、深度和宽度相似的网络可以获得类似质量的结果，但我们的方法提供了坚实的证据，表明转向更稀疏的架构总体上是可行且有益的想法。这为未来基于[2]以自动化方式创建更稀疏、更精细的结构提出了有前景的研究方向。

10 致谢

我们要感谢Sanjeev Arora和Aditya Bhaskara就文献[2]进行的富有成果的讨论。同时，我们感谢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在光度失真方面提供的帮助。此外，如果没有Chuck Rosenberg和Hartwig Adam的支持，我们的工作也无法完成。

参考文献

[1] 了解你的梗：我们需要更深入。http://knowyourmeme.com/memes/we-need-to-go-deeper。访问日期：2014-09-15。[2] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. 学习某些深度表示的可证明边界。*CoRR*, abs/1310.6343, 2013.[3] Ümit V. Çatalyürek, Cevdet Aykanat, and Bora Uçar. 论二维稀疏矩阵划分：模型、方法与方案。*SIAM J. Sci. Comput.*, 32(2):656–683, 2010年2月。[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. 大规模分布式深度网络。载于 P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger 编辑的 *Advances in Neural Information Processing Systems 25*, 第1232–1240页。2012年。[5] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. 使用深度神经网络的可扩展目标检测。载于 *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014年。[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 用于精确目标检测和语义分割的丰富特征层次结构。载于 *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014年。[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 通过防止特征检测器的共适应来改进神经网络。*CoRR*, abs/1207.0580, 2012年。[8] Andrew G. Howard. 基于深度卷积神经网络的图像分类的一些改进。*CoRR*, abs/1312.5402, 2013年。[9] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. 使用深度卷积神经网络进行ImageNet分类。载于 *Advances in Neural Information Processing Systems 25*, 第1106–1114页, 2012年。

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, 与 L. D. Jackel。反向传播应用于手写邮政编码识别。*Neural Comput.*, 1(4):541–551, 1989年12月。

[11] Yann LeCun, Léon Bottou, Yoshua Bengio 和 Patrick Haffner。基于梯度的学习应用于文档识别。*Proceedings of the IEEE*, 86(11):2278–2324, 1998。

[12] Min Lin、Qiang Chen 和 Shuicheng Yan。Network in network。*CoRR*，abs/1312.4400，2013。

[13] B. T. Polyak 与 A. B. Juditsky。通过平均加速随机逼近。*SIAM J. Control Optim.*，第30卷第4期，第838–855页，1992年7月。

[14] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann Le-Cun. Overfeat: 使用卷积网络进行集成识别、定位与检测。*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, Léon 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, Tomás 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.

[15] Thomas Serre, Lior Wolf, Stanley M. Bileschi, Maximilian Riesenhuber, 与 Tomaso Poggio。采用类皮层机制的鲁棒物体识别。*IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):411–426, 2007。[16] Fengguang Song 与 Jack Dongarra。在具有1000个CPU核心的共享内存众核系统上扩展矩阵计算。收录于 *Proceedings of the 28th ACM International Conference on Supercomputing*, ICS '14, 第333–342页, 美国纽约, 2014。ACM。[17] Ilya Sutskever, James Martens, George E. Dahl, 与 Geoffrey E. Hinton。论深度学习中初始化和动量的重要性。收录于 *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 第28卷 *JMLR Proceedings*, 第1139–1147页。JMLR.org, 2013。[18] Christian Szegedy, Alexander Toshev, 与 Dumitru Erhan。用于物体检测的深度神经网络。收录于 Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, 与 Kilian Q. Weinberger 编辑的 *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.*, 第2553–2561页, 2013。[19] Alexander Toshev 与 Christian Szegedy。Deeppose：通过深度神经网络进行人体姿态估计。*CoRR*, abs/1312.4659, 2013。[20] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, 与 Arnold W. M. Smeulders。分割作为物体识别的选择性搜索。收录于 *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 第1879–1886页, 美国华盛顿特区, 2011。IEEE计算机学会。[21] Matthew D. Zeiler 与 Rob Fergus。可视化和理解卷积网络。收录于 David J. Fleet, Tomás Pajdla, Bernt Schiele, 与 Tinne Tuytelaars 编辑的 *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, 第8689卷 *Lecture Notes in Computer Science*, 第818–833页。Springer, 2014。