VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman+**
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

1 INTRODUCTION

Convolutional networks (ConvNets) have recently enjoyed a great success in large-scale image and video recognition (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014) which has become possible due to the large public image repositories, such as ImageNet (Deng et al., 2009), and high-performance computing systems, such as GPUs or large-scale distributed clusters (Dean et al., 2012). In particular, an important role in the advance of deep visual recognition architectures has been played by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2014), which has served as a testbed for a few generations of large-scale image classification systems, from high-dimensional shallow feature encodings (Perronnin et al., 2010) (the winner of ILSVRC-2011) to deep ConvNets (Krizhevsky et al., 2012) (the winner of ILSVRC-2012).

With ConvNets becoming more of a commodity in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky et al. (2012) in a bid to achieve better accuracy. For instance, the best-performing submissions to the ILSVRC-2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014) utilised smaller receptive window size and smaller stride of the first convolutional layer. Another line of improvements dealt with training and testing the networks densely over the whole image and over multiple scales (Sermanet et al., 2014; Howard, 2014). In this paper, we address another important aspect of ConvNet architecture design – its depth. To this end, we fix other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3×3) convolution filters in all layers.

As a result, we come up with significantly more accurate ConvNet architectures, which not only achieve the state-of-the-art accuracy on ILSVRC classification and localisation tasks, but are also applicable to other image recognition datasets, where they achieve excellent performance even when used as a part of a relatively simple pipelines (e.g. deep features classified by a linear SVM without fine-tuning). We have released our two best-performing models[1] to facilitate further research.

The rest of the paper is organised as follows. In Sect. 2, we describe our ConvNet configurations. The details of the image classification training and evaluation are then presented in Sect. 3, and the

*current affiliation: Google DeepMind +current affiliation: University of Oxford and Google DeepMind
[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep/

用于大规模图像识别的极深卷积网络

Karen Simonyan* 与 Andrew Zisserman+ 牛津大学工程科学系视觉几何组 {karen,az}@robots.ox.ac.uk

摘要

在本工作中，我们研究了卷积网络深度在大规模图像识别场景下对其准确率的影响。我们的主要贡献是：通过采用极小（3×3）卷积核的架构，对增加网络深度进行了全面评估，结果表明将深度提升至16–19个权重层时，可以在现有先进配置的基础上实现显著改进。这些发现为我们参与2014年ImageNet挑战赛奠定了基础，我们的团队分别在定位和分类任务中获得了第一名和第二名的成绩。我们还证明了我们的表征能够很好地泛化到其他数据集，并在这些数据集上取得了领先水平的结果。我们已公开两个性能最佳的卷积网络模型，以促进深度视觉表征在计算机视觉领域的进一步研究。

1 引言

卷积网络（ConvNets）近年来在大规模图像与视频识别领域取得了巨大成功（Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014），这得益于大型公共图像库（如ImageNet（Deng et al., 2009））以及高性能计算系统（如GPU或大规模分布式集群（Dean et al., 2012））的发展。其中，ImageNet大规模视觉识别挑战赛（ILSVRC）（Russakovsky et al., 2014）在推动深度视觉识别架构进步方面发挥了关键作用，该赛事已成为多代大规模图像分类系统的测试平台——从高维浅层特征编码（Perronnin et al., 2010）（ILSVRC-2011冠军）到深度卷积网络（Krizhevsky et al., 2012）（ILSVRC-2012冠军）均在此得到验证。

随着卷积神经网络在计算机视觉领域日益普及，许多研究尝试改进Krizhevsky等人（2012）提出的原始架构，以期获得更高的准确率。例如，在ILSVRC-2013中表现最佳的提交方案（Zeiler & Fergus, 2013; Sermanet et al., 2014）采用了更小的感受野窗口尺寸和第一卷积层更小的步长。另一类改进方法涉及在整个图像和多种尺度上对网络进行密集的训练与测试（Sermanet et al., 2014; Howard, 2014）。本文中，我们探讨卷积神经网络架构设计的另一个重要维度——网络深度。为此，我们固定架构的其他参数，通过增加更多卷积层来稳步加深网络深度，这一设计得以实现是因为在所有层中均使用了极小（3×3）的卷积滤波器。

因此，我们提出了显著更准确的卷积网络架构，这些架构不仅在ILSVRC分类和定位任务上达到了最先进的准确率，而且适用于其他图像识别数据集，即使作为相对简单流程的一部分（例如，未经微调的线性SVM分类深度特征），也能取得优异的性能。我们已经发布了两个性能最佳的模型[1]，以促进进一步的研究。

本文的其余部分组织如下。在第2节中，我们描述了ConvNet的配置。随后在第3节中介绍了图像分类训练与评估的细节，

*current affiliation: Google DeepMind +current affiliation: University of Oxford and Google DeepMind
[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep/

configurations are compared on the ILSVRC classification task in Sect. 4. Sect. 5 concludes the paper. For completeness, we also describe and assess our ILSVRC-2014 object localisation system in Appendix A, and discuss the generalisation of very deep features to other datasets in Appendix B. Finally, Appendix C contains the list of major paper revisions.

2 CONVNET CONFIGURATIONS

To measure the improvement brought by the increased ConvNet depth in a fair setting, all our ConvNet layer configurations are designed using the same principles, inspired by Ciresan et al. (2011); Krizhevsky et al. (2012). In this section, we first describe a generic layout of our ConvNet configurations (Sect. 2.1) and then detail the specific configurations used in the evaluation (Sect. 2.2). Our design choices are then discussed and compared to the prior art in Sect. 2.3.

2.1 ARCHITECTURE

During training, the input to our ConvNets is a fixed-size 224×224 RGB image. The only preprocessing we do is subtracting the mean RGB value, computed on the training set, from each pixel. The image is passed through a stack of convolutional (conv.) layers, where we use filters with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations we also utilise 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU (Krizhevsky et al., 2012)) non-linearity. We note that none of our networks (except for one) contain Local Response Normalisation (LRN) normalisation (Krizhevsky et al., 2012): as will be shown in Sect. 4, such normalisation does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time. Where applicable, the parameters for the LRN layer are those of (Krizhevsky et al., 2012).

2.2 CONFIGURATIONS

The ConvNet configurations, evaluated in this paper, are outlined in Table 1, one per column. In the following we will refer to the nets by their names (A–E). All configurations follow the generic design presented in Sect. 2.1, and differ only in the depth: from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

In Table 2 we report the number of parameters for each configuration. In spite of a large depth, the number of weights in our nets is not greater than the number of weights in a more shallow net with larger conv. layer widths and receptive fields (144M weights in (Sermanet et al., 2014)).

2.3 DISCUSSION

Our ConvNet configurations are quite different from the ones used in the top-performing entries of the ILSVRC-2012 (Krizhevsky et al., 2012) and ILSVRC-2013 competitions (Zeiler & Fergus, 2013; Sermanet et al., 2014). Rather than using relatively large receptive fields in the first conv. layers (e.g. 11×11 with stride 4 in (Krizhevsky et al., 2012), or 7×7 with stride 2 in (Zeiler & Fergus, 2013; Sermanet et al., 2014)), we use very small 3×3 receptive fields throughout the whole net, which are convolved with the input at every pixel (with stride 1). It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an effective receptive field of 5×5; three

配置在ILSVRC分类任务上的比较见第4节。第5节对本文进行总结。为求完整，我们还在附录A中描述并评估了我们的ILSVRC-2014目标定位系统，并在附录B中讨论了极深度特征在其他数据集上的泛化能力。最后，附录C列出了论文的主要修订记录。

2 种卷积网络配置

为了在公平环境下衡量增加ConvNet深度带来的改进，我们所有的ConvNet层配置均遵循由Ciresan等人（2011）和Krizhevsky等人（2012）启发的相同设计原则。本节中，我们首先描述ConvNet配置的通用布局（第2.1节），随后详述评估中使用的具体配置（第2.2节）。最后在第2.3节中讨论我们的设计选择，并与现有技术进行比较。

2.1 架构

在训练过程中，我们ConvNets的输入是固定尺寸224×224的RGB图像。我们唯一的预处理操作是从每个像素中减去在训练集上计算得到的RGB均值。图像会经过一系列卷积层，其中我们使用感受野极小的滤波器：3×3（这是能够捕捉左/右、上/下、中心概念的最小尺寸）。在某个配置中，我们还使用了1×1卷积滤波器，这可以视为对输入通道的线性变换（随后进行非线性处理）。卷积步长固定为1像素；卷积层输入的空间填充设置使得卷积后空间分辨率得以保持，例如对于3×3卷积层，填充为1像素。空间池化由五个最大池化层执行，这些池化层位于部分卷积层之后（并非所有卷积层后都跟随最大池化）。最大池化在2×2像素窗口上以步长2进行。

一系列卷积层（在不同架构中深度不同）之后是三个全连接层：前两层各有4096个通道，第三层执行1000类ILSVRC分类任务，因此包含1000个通道（每个类别对应一个通道）。最后一层是soft-max层。所有网络中的全连接层配置均保持一致。

所有隐藏层均配备了修正线性单元（ReLU，Krizhevsky等人，2012）非线性激活函数。我们注意到，除一个网络外，我们的所有网络均未包含局部响应归一化（LRN，Krizhevsky等人，2012）：如第4节所示，此类归一化并未提升ILSVRC数据集上的性能，反而增加了内存消耗与计算时间。在适用的情况下，LRN层的参数均遵循Krizhevsky等人（2012）的设置。

2.2 配置

本文评估的ConvNet配置如表1所示，每列对应一种配置。下文将以名称（A–E）指代这些网络。所有配置均遵循第2.1节提出的通用设计，仅深度存在差异：从网络A的11个权重层（8个卷积层和3个全连接层）到网络E的19个权重层（16个卷积层和3个全连接层）。卷积层的宽度（通道数）设置较小，第一层为64通道，随后每个最大池化层后通道数翻倍，直至达到512通道。

在表2中，我们报告了每种配置的参数数量。尽管网络深度很大，但我们网络中的权重数量并未超过那些卷积层宽度更大、感受野更广的较浅网络（(Sermanet et al., 2014) 中的网络拥有1.44亿权重）。

2.3 讨论

我们的卷积网络配置与ILSVRC-2012（Krizhevsky等人，2012）和ILSVRC-2013竞赛（Zeiler & Fergus，2013；Sermanet等人，2014）中表现最佳的参赛模型所使用的配置有很大不同。我们没有在第一个卷积层使用相对较大的感受野（例如（Krizhevsky等人，2012）中使用的11×11、步幅4，或（Zeiler & Fergus，2013；Sermanet等人，2014）中使用的7×7、步幅2），而是在整个网络中始终使用非常小的3×3感受野，并以每个像素（步幅1）对输入进行卷积。可以明显看出，两个3×3卷积层（中间没有空间池化）堆叠后的有效感受野为5×5；三个

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

such layers have a 7×7 effective receptive field. So what have we gained by using, for instance, a stack of three 3×3 conv. layers instead of a single 7×7 layer? First, we incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Second, we decrease the number of parameters: assuming that both the input and the output of a three-layer 3×3 convolution stack has C channels, the stack is parametrised by $3\left(3^2C^2\right) = 27C^2$ weights; at the same time, a single 7×7 conv. layer would require $7^2C^2 = 49C^2$ parameters, i.e. 81% more. This can be seen as imposing a regularisation on the 7×7 conv. filters, forcing them to have a decomposition through the 3×3 filters (with non-linearity injected in between).

The incorporation of 1×1 conv. layers (configuration C, Table 1) is a way to increase the non-linearity of the decision function without affecting the receptive fields of the conv. layers. Even though in our case the 1×1 convolution is essentially a linear projection onto the space of the same dimensionality (the number of input and output channels is the same), an additional non-linearity is introduced by the rectification function. It should be noted that 1×1 conv. layers have recently been utilised in the "Network in Network" architecture of Lin et al. (2014).

Small-size convolution filters have been previously used by Ciresan et al. (2011), but their nets are significantly less deep than ours, and they did not evaluate on the large-scale ILSVRC dataset. Goodfellow et al. (2014) applied deep ConvNets (11 weight layers) to the task of street number recognition, and showed that the increased depth led to better performance. GoogLeNet (Szegedy et al., 2014), a top-performing entry of the ILSVRC-2014 classification task, was developed independently of our work, but is similar in that it is based on very deep ConvNets

表1：ConvNet配置（按列显示）。配置的深度从左（A）到右（E）逐渐增加，因为添加了更多层（添加的层以粗体显示）。卷积层参数表示为“conv⟨感受野大小⟩-⟨通道数⟩”。为简洁起见，未显示ReLU激活函数。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64
	LRN	**conv3-64**	conv3-64	conv3-64	conv3-64
maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
		conv3-128	conv3-128	conv3-128	conv3-128
maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
			conv1-256	**conv3-256**	conv3-256
					conv3-256
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
			conv1-512	**conv3-512**	conv3-512
					conv3-512
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
			conv1-512	**conv3-512**	conv3-512
					conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

表2：参数数量（以百万计）。

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

这样的层具有7×7的有效感受野。那么，我们通过使用例如三个3×3卷积层的堆叠，而不是单个7×7层，获得了什么？首先，我们引入了三个非线性整流层，而非仅一个，这使得决策函数更具判别力。其次，我们减少了参数数量：假设三层3×3卷积堆叠的输入和输出都有C个通道，则该堆叠由$3\left(3^2C^2\right) = 27C^2$个权重参数化；同时，单个7×7卷积层将需要$7^2C^2 = 49C^2$个参数，即多出81%。这可以视为对7×7卷积滤波器施加了正则化，强制其通过3×3滤波器进行分解（并在其间注入非线性）。

引入1×1卷积层（配置C，表1）是在不影响卷积层感受野的前提下，增强决策函数非线性表达能力的一种方法。尽管在我们的案例中，1×1卷积本质上是在相同维度空间（输入与输出通道数相同）上的线性投影，但整流函数仍为其引入了额外的非线性特性。值得注意的是，1×1卷积层近期已被应用于Lin等人（2014）提出的“Network in Network”架构中。

小尺寸卷积滤波器先前已被Ciresan等人（2011年）使用，但他们的网络深度明显低于我们的网络，且未在大型ILSVRC数据集上进行评估。Goodfellow等人（2014年）将深度卷积网络（11个权重层）应用于街景门牌号识别任务，并证明增加深度能提升性能。GoogLeNet（Szegedy等人，2014年）是ILSVRC-2014分类任务中表现最佳的模型，其开发工作独立于我们的研究，但相似之处在于它同样基于极深的卷积网络。

(22 weight layers) and small convolution filters (apart from 3×3, they also use 1×1 and 5×5 convolutions). Their network topology is, however, more complex than ours, and the spatial resolution of the feature maps is reduced more aggressively in the first layers to decrease the amount of computation. As will be shown in Sect. 4.5, our model is outperforming that of Szegedy et al. (2014) in terms of the single-network classification accuracy.

3 CLASSIFICATION FRAMEWORK

In the previous section we presented the details of our network configurations. In this section, we describe the details of classification ConvNet training and evaluation.

3.1 TRAINING

The ConvNet training procedure generally follows Krizhevsky et al. (2012) (except for sampling the input crops from multi-scale training images, as explained later). Namely, the training is carried out by optimising the multinomial logistic regression objective using mini-batch gradient descent (based on back-propagation (LeCun et al., 1989)) with momentum. The batch size was set to 256, momentum to 0.9. The training was regularised by weight decay (the L_2 penalty multiplier set to $5 \cdot 10^{-4}$) and dropout regularisation for the first two fully-connected layers (dropout ratio set to 0.5). The learning rate was initially set to 10^{-2}, and then decreased by a factor of 10 when the validation set accuracy stopped improving. In total, the learning rate was decreased 3 times, and the learning was stopped after 370K iterations (74 epochs). We conjecture that in spite of the larger number of parameters and the greater depth of our nets compared to (Krizhevsky et al., 2012), the nets required less epochs to converge due to (a) implicit regularisation imposed by greater depth and smaller conv. filter sizes; (b) pre-initialisation of certain layers.

The initialisation of the network weights is important, since bad initialisation can stall learning due to the instability of gradient in deep nets. To circumvent this problem, we began with training the configuration A (Table 1), shallow enough to be trained with random initialisation. Then, when training deeper architectures, we initialised the first four convolutional layers and the last three fully-connected layers with the layers of net A (the intermediate layers were initialised randomly). We did not decrease the learning rate for the pre-initialised layers, allowing them to change during learning. For random initialisation (where applicable), we sampled the weights from a normal distribution with the zero mean and 10^{-2} variance. The biases were initialised with zero. It is worth noting that after the paper submission we found that it is possible to initialise the weights without pre-training by using the random initialisation procedure of Glorot & Bengio (2010).

To obtain the fixed-size 224×224 ConvNet input images, they were randomly cropped from rescaled training images (one crop per image per SGD iteration). To further augment the training set, the crops underwent random horizontal flipping and random RGB colour shift (Krizhevsky et al., 2012). Training image rescaling is explained below.

Training image size. Let S be the smallest side of an isotropically-rescaled training image, from which the ConvNet input is cropped (we also refer to S as the training scale). While the crop size is fixed to 224×224, in principle S can take on any value not less than 224: for $S = 224$ the crop will capture whole-image statistics, completely spanning the smallest side of a training image; for $S \gg 224$ the crop will correspond to a small part of the image, containing a small object or an object part.

We consider two approaches for setting the training scale S. The first is to fix S, which corresponds to single-scale training (note that image content within the sampled crops can still represent multi-scale image statistics). In our experiments, we evaluated models trained at two fixed scales: $S = 256$ (which has been widely used in the prior art (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014)) and $S = 384$. Given a ConvNet configuration, we first trained the network using $S = 256$. To speed-up training of the $S = 384$ network, it was initialised with the weights pre-trained with $S = 256$, and we used a smaller initial learning rate of 10^{-3}.

The second approach to setting S is multi-scale training, where each training image is individually rescaled by randomly sampling S from a certain range $[S_{min}, S_{max}]$ (we used $S_{min} = 256$ and $S_{max} = 512$). Since objects in images can be of different size, it is beneficial to take this into account during training. This can also be seen as training set augmentation by scale jittering, where a single

(22个权重层)和小卷积核（除了3××3，他们还使用了1××1和5××5卷积）。然而，他们的网络拓扑结构比我们的更复杂，并且特征图的空间分辨率在初始层被更激进地降低以减少计算量。如第4.5节所示，在单网络分类准确率方面，我们的模型优于Szegedy等人（2014）的模型。

3 分类框架

在上一节中，我们详细介绍了网络配置的具体细节。本节将阐述分类卷积网络的训练与评估细节。

3.1 训练

卷积网络的训练过程大体上遵循Krizhevsky等人（2012）的方法（除了从多尺度训练图像中采样输入裁剪区域，这一点将在后文说明）。具体而言，训练通过小批量梯度下降（基于反向传播（LeCun等人，1989））结合动量法来优化多项逻辑回归目标。批量大小设为256，动量设为0.9。训练通过权重衰减（L_2惩罚乘数设为$5 \cdot 10^{-4}$）和前两个全连接层的Dropout正则化（Dropout比例设为0.5）进行正则化。学习率初始设为10^{-2}，当验证集准确率停止提升时，学习率以10倍因子降低。学习率总共降低了3次，训练在37万次迭代（74个周期）后停止。我们推测，尽管我们的网络参数数量更多、深度比（Krizhevsky等人，2012）的网络更大，但由于以下原因，网络收敛所需的周期更少：（a）更大深度和更小卷积滤波器尺寸带来的隐式正则化；（b）特定层的预初始化。

网络权重的初始化至关重要，因为糟糕的初始化会因深度网络中梯度的不稳定性而阻碍学习。为解决这一问题，我们首先训练了配置A（表1），该网络结构较浅，足以通过随机初始化进行训练。随后，在训练更深层架构时，我们用网络A的层参数初始化了前四个卷积层和最后三个全连接层（中间层仍采用随机初始化）。对于预初始化的层，我们并未降低其学习率，允许它们在训练过程中持续调整。在随机初始化（适用情况下）中，我们从均值为零、方差为10^{-12}的正态分布中采样权重，偏置项则初始化为零。值得注意的是，在论文提交后我们发现，采用Glorot & Bengio（2010）的随机初始化方法，无需预训练即可完成权重初始化。

为了获得固定尺寸的224×224 ConvNet输入图像，这些图像是从重新缩放的训练图像中随机裁剪得到的（每次SGD迭代每张图像裁剪一次）。为了进一步扩充训练集，裁剪后的图像还进行了随机水平翻转和随机RGB色彩偏移处理（Krizhevsky等人，2012年）。训练图像的重新缩放方法将在下文说明。

训练图像尺寸。设S为各向同性重缩放后训练图像的最小边，ConvNet的输入从此图像中裁剪（我们也将S称为训练尺度）。虽然裁剪尺寸固定为224×224，但原则上S可以取不小于224的任意值：当$S = 224$时，裁剪区域将覆盖完整图像统计信息，完全跨越训练图像的最小边；当$S \gg 224$时，裁剪区域将对应图像的一小部分，包含小物体或物体局部。

我们考虑两种设定训练尺度S的方法。第一种是固定S，这对应于单尺度训练（需注意采样裁剪区域内的图像内容仍可体现多尺度图像统计特性）。在实验中，我们评估了两种固定尺度下训练的模型：$S = 256$（该尺度在现有技术中已被广泛采用（Krizhevsky等人，2012；Zeiler & Fergus，2013；Sermanet等人，2014））与$S = 384$。对于给定的卷积网络配置，我们首先使用$S = 256$训练网络。为加速$S = 384$网络的训练，我们采用经$S = 256$预训练的权重进行初始化，并设置了较小的初始学习率10^{-3}。

设置S的第二种方法是多尺度训练，其中每个训练图像通过从特定范围$[S_{min}, S_{max}]$中随机采样S进行单独重新缩放（我们使用了$S_{min} = 256$和$S_{max} = 512$）。由于图像中的物体可能具有不同的大小，在训练过程中考虑这一点是有益的。这也可以被视为通过尺度抖动进行训练集增强，其中单个

model is trained to recognise objects over a wide range of scales. For speed reasons, we trained multi-scale models by fine-tuning all layers of a single-scale model with the same configuration, pre-trained with fixed $S = 384$.

3.2 TESTING

At test time, given a trained ConvNet and an input image, it is classified in the following way. First, it is isotropically rescaled to a pre-defined smallest image side, denoted as Q (we also refer to it as the test scale). We note that Q is not necessarily equal to the training scale S (as we will show in Sect. 4, using several values of Q for each S leads to improved performance). Then, the network is applied densely over the rescaled test image in a way similar to (Sermanet et al., 2014). Namely, the fully-connected layers are first converted to convolutional layers (the first FC layer to a 7×7 conv. layer, the last two FC layers to 1×1 conv. layers). The resulting fully-convolutional net is then applied to the whole (uncropped) image. The result is a class score map with the number of channels equal to the number of classes, and a variable spatial resolution, dependent on the input image size. Finally, to obtain a fixed-size vector of class scores for the image, the class score map is spatially averaged (sum-pooled). We also augment the test set by horizontal flipping of the images; the soft-max class posteriors of the original and flipped images are averaged to obtain the final scores for the image.

Since the fully-convolutional network is applied over the whole image, there is no need to sample multiple crops at test time (Krizhevsky et al., 2012), which is less efficient as it requires network re-computation for each crop. At the same time, using a large set of crops, as done by Szegedy et al. (2014), can lead to improved accuracy, as it results in a finer sampling of the input image compared to the fully-convolutional net. Also, multi-crop evaluation is complementary to dense evaluation due to different convolution boundary conditions: when applying a ConvNet to a crop, the convolved feature maps are padded with zeros, while in the case of dense evaluation the padding for the same crop naturally comes from the neighbouring parts of an image (due to both the convolutions and spatial pooling), which substantially increases the overall network receptive field, so more context is captured. While we believe that in practice the increased computation time of multiple crops does not justify the potential gains in accuracy, for reference we also evaluate our networks using 50 crops per scale (5×5 regular grid with 2 flips), for a total of 150 crops over 3 scales, which is comparable to 144 crops over 4 scales used by Szegedy et al. (2014).

3.3 IMPLEMENTATION DETAILS

Our implementation is derived from the publicly available C++ Caffe toolbox (Jia, 2013) (branched out in December 2013), but contains a number of significant modifications, allowing us to perform training and evaluation on multiple GPUs installed in a single system, as well as train and evaluate on full-size (uncropped) images at multiple scales (as described above). Multi-GPU training exploits data parallelism, and is carried out by splitting each batch of training images into several GPU batches, processed in parallel on each GPU. After the GPU batch gradients are computed, they are averaged to obtain the gradient of the full batch. Gradient computation is synchronous across the GPUs, so the result is exactly the same as when training on a single GPU.

While more sophisticated methods of speeding up ConvNet training have been recently proposed (Krizhevsky, 2014), which employ model and data parallelism for different layers of the net, we have found that our conceptually much simpler scheme already provides a speedup of 3.75 times on an off-the-shelf 4-GPU system, as compared to using a single GPU. On a system equipped with four NVIDIA Titan Black GPUs, training a single net took 2–3 weeks depending on the architecture.

4 CLASSIFICATION EXPERIMENTS

Dataset. In this section, we present the image classification results achieved by the described ConvNet architectures on the ILSVRC-2012 dataset (which was used for ILSVRC 2012–2014 challenges). The dataset includes images of 1000 classes, and is split into three sets: training (1.3M images), validation (50K images), and testing (100K images with held-out class labels). The classification performance is evaluated using two measures: the top-1 and top-5 error. The former is a multi-class classification error, i.e. the proportion of incorrectly classified images; the latter is the

模型经过训练，能够识别各种尺度下的物体。出于速度考虑，我们通过微调单尺度模型的所有层来训练多尺度模型，该单尺度模型采用相同配置，并已使用固定的$S = 384$进行预训练。

3.2 测试

在测试阶段，给定一个训练好的卷积网络和一张输入图像，其分类方式如下。首先，将图像各向同性地缩放到预定义的最小图像边长，记为Q (，我们亦称之为测试尺度)。需要说明的是，Q不一定等于训练尺度S (——正如第4节将展示的，对每个S使用多个Q值能提升性能)。随后，以类似于（Sermanet等人，2014）的方式在缩放后的测试图像上密集应用网络。具体而言，全连接层首先被转换为卷积层（第一个全连接层转为7×7卷积层，最后两个全连接层转为1×1卷积层）。将得到的全卷积网络应用于整张（未裁剪的）图像后，会生成通道数等于类别数的类别得分图，其空间分辨率取决于输入图像尺寸。最后，通过对类别得分图进行空间平均（求和池化），得到图像对应的固定维度类别得分向量。我们还通过水平翻转图像来增强测试集：将原始图像与翻转图像的softmax类别后验概率取平均，从而得到图像的最终得分。

由于全卷积网络应用于整个图像，因此在测试时无需采样多个裁剪区域（Krizhevsky等人，2012），因为为每个裁剪区域重新计算网络效率较低。同时，如Szegedy等人（2014）所做，使用大量裁剪区域可以提高准确性，因为与全卷积网络相比，它能对输入图像进行更精细的采样。此外，多裁剪评估与密集评估因卷积边界条件不同而具有互补性：将卷积神经网络应用于裁剪区域时，卷积特征图会以零填充；而在密集评估中，同一裁剪区域的填充自然来自图像的相邻部分（由于卷积和空间池化的共同作用），这显著增加了网络的整体感受野，从而捕获更多上下文信息。尽管我们认为在实践中，多裁剪带来的计算时间增加并不足以证明其可能带来的精度提升，但为便于参考，我们仍使用每个尺度50个裁剪区域（5×5规则网格加2次翻转）评估网络，总计3个尺度共150个裁剪区域，这与Szegedy等人（2014）使用的4个尺度144个裁剪区域具有可比性。

3.3 实现细节

我们的实现基于公开可用的C++ Caffe工具箱（Jia，2013）（2013年12月分支），但包含若干重大修改，使我们能够在单个系统中安装的多块GPU上进行训练和评估，并支持在多尺度下对完整尺寸（未裁剪）图像进行训练和评估（如上所述）。多GPU训练采用数据并行机制，通过将每批训练图像分割为多个GPU批次实现，各GPU并行处理这些批次。计算完GPU批次梯度后，将其平均以获得完整批次的梯度。梯度计算在GPU间同步进行，因此结果与单GPU训练完全一致。

尽管最近已经提出了更复杂的加速卷积网络训练的方法（Krizhevsky, 2014），这些方法对网络的不同层采用了模型并行和数据并行策略，但我们发现，我们概念上简单得多的方案，在现成的4-GPU系统上，相比使用单GPU，已经能带来3.75倍的加速。在配备四块NVIDIA Titan Black GPU的系统上，训练单个网络需要2到3周，具体时间取决于网络架构。

4 分类实验

数据集。在本节中，我们展示了所描述的卷积网络架构在ILSVRC-2012数据集（该数据集用于ILSVRC 2012–2014挑战赛）上取得的图像分类结果。该数据集包含1000个类别的图像，并分为三个部分：训练集（1.3M张图像）、验证集（50K张图像）和测试集（100K张图像，附带保留的类别标签）。分类性能通过两个指标进行评估：top-1和top-5错误率。前者是多类别分类错误率，即错误分类图像的比例；后者是

main evaluation criterion used in ILSVRC, and is computed as the proportion of images such that the ground-truth category is outside the top-5 predicted categories.

For the majority of experiments, we used the validation set as the test set. Certain experiments were also carried out on the test set and submitted to the official ILSVRC server as a "VGG" team entry to the ILSVRC-2014 competition (Russakovsky et al., 2014).

4.1 SINGLE SCALE EVALUATION

We begin with evaluating the performance of individual ConvNet models at a single scale with the layer configurations described in Sect. 2.2. The test image size was set as follows: $Q = S$ for fixed S, and $Q = 0.5(S_{min} + S_{max})$ for jittered $S \in [S_{min}, S_{max}]$. The results of are shown in Table 3.

First, we note that using local response normalisation (A-LRN network) does not improve on the model A without any normalisation layers. We thus do not employ normalisation in the deeper architectures (B–E).

Second, we observe that the classification error decreases with the increased ConvNet depth: from 11 layers in A to 19 layers in E. Notably, in spite of the same depth, the configuration C (which contains three 1×1 conv. layers), performs worse than the configuration D, which uses 3×3 conv. layers throughout the network. This indicates that while the additional non-linearity does help (C is better than B), it is also important to capture spatial context by using conv. filters with non-trivial receptive fields (D is better than C). The error rate of our architecture saturates when the depth reaches 19 layers, but even deeper models might be beneficial for larger datasets. We also compared the net B with a shallow net with five 5×5 conv. layers, which was derived from B by replacing each pair of 3×3 conv. layers with a single 5×5 conv. layer (which has the same receptive field as explained in Sect. 2.3). The top-1 error of the shallow net was measured to be 7% higher than that of B (on a center crop), which confirms that a deep net with small filters outperforms a shallow net with larger filters.

Finally, scale jittering at training time ($S \in [256; 512]$) leads to significantly better results than training on images with fixed smallest side ($S = 256$ or $S = 384$), even though a single scale is used at test time. This confirms that training set augmentation by scale jittering is indeed helpful for capturing multi-scale image statistics.

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

4.2 MULTI-SCALE EVALUATION

Having evaluated the ConvNet models at a single scale, we now assess the effect of scale jittering at test time. It consists of running a model over several rescaled versions of a test image (corresponding to different values of Q), followed by averaging the resulting class posteriors. Considering that a large discrepancy between training and testing scales leads to a drop in performance, the models trained with fixed S were evaluated over three test image sizes, close to the training one: $Q = \{S - 32, S, S + 32\}$. At the same time, scale jittering at training time allows the network to be applied to a wider range of scales at test time, so the model trained with variable $S \in [S_{min}; S_{max}]$ was evaluated over a larger range of sizes $Q = \{S_{min}, 0.5(S_{min} + S_{max}), S_{max}\}$.

ILSVRC中使用的主要评估标准，计算方式为真实类别不在预测的前五个类别之内的图像比例。

在大多数实验中，我们使用验证集作为测试集。部分实验也在测试集上进行，并作为“VGG”团队参赛作品提交至ILSVRC-2014竞赛的官方服务器（Russakovsky等人，2014年）。

4.1 单尺度评估

我们首先评估单个ConvNet模型在单一尺度下的性能，使用第2.2节所述的层配置。测试图像尺寸设置如下：固定*S*时采用$Q = S$，抖动$S \in [S_{min}, S_{max}]$时采用$Q = 0.5(S_{min} + S_{max})$。相关结果展示在表3中。

首先，我们注意到使用局部响应归一化（A-LRN网络）并未对未包含任何归一化层的模型A带来改进。因此，在更深的架构（B–E）中我们未采用归一化处理。

其次，我们观察到分类误差随着ConvNet深度的增加而减小：从A的11层到E的19层。值得注意的是，尽管深度相同，配置C（包含三个1×1卷积层）的表现却逊于配置D，后者在整个网络中均使用3×3卷积层。这表明，虽然额外的非线性确实有帮助（C优于B），但通过使用具有非平凡感受野的卷积滤波器来捕捉空间上下文同样重要（D优于C）。当深度达到19层时，我们架构的错误率趋于饱和，但对于更大的数据集，更深的模型可能仍有益处。我们还将网络B与一个浅层网络进行了比较，该浅层网络包含五个5×5卷积层，它是通过将B中每对3×3卷积层替换为单个5×5卷积层而得到的（如第2.3节所述，其感受野相同）。经测量，该浅层网络的top-1误差比B高出7%（基于中心裁剪测试），这证实了采用小滤波器的深层网络优于使用大滤波器的浅层网络。

最后，在训练时进行尺度抖动（$S \in [256; 512]$）相比在固定最小边长的图像上训练（$S = 256$ 或 $S = 384$）能带来显著更好的结果，即便在测试时仅使用单一尺度。这证实了通过尺度抖动进行训练集增强确实有助于捕捉多尺度的图像统计特征。

表3：ConvNet在单一测试尺度下的性能。

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
	256	256	28.1	9.4
C	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
	256	256	27.0	8.8
D	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
	256	256	27.3	9.0
E	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

4.2 多尺度评估

在评估了单尺度下的ConvNet模型后，我们现在评估测试时尺度抖动的影响。该方法包括对测试图像的多个缩放版本（对应不同的Q值）运行模型，然后对得到的类别后验概率进行平均。考虑到训练与测试尺度间的巨大差异会导致性能下降，使用固定*S*训练的模型在三个接近训练尺度的测试图像尺寸上进行了评估：$Q = \{S - 32, S, S + 32\}$。同时，训练时的尺度抖动使得网络能够在测试时应用于更广泛的尺度范围，因此使用可变$S \in [S_{min}; S_{max}]$训练的模型在更大的尺寸范围$Q = \{S_{min}, 0.5(S_{min} + S_{max}), S_{max}\}$上进行了评估。

The results, presented in Table 4, indicate that scale jittering at test time leads to better performance (as compared to evaluating the same model at a single scale, shown in Table 3). As before, the deepest configurations (D and E) perform the best, and scale jittering is better than training with a fixed smallest side S. Our best single-network performance on the validation set is $24.8\%/7.5\%$ top-1/top-5 error (highlighted in bold in Table 4). On the test set, the configuration E achieves 7.3% top-5 error.

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	$[256; 512]$	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	$[256; 512]$	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	$[256; 512]$	256,384,512	**24.8**	**7.5**

4.3 MULTI-CROP EVALUATION

In Table 5 we compare dense ConvNet evaluation with mult-crop evaluation (see Sect. 3.2 for details). We also assess the complementarity of the two evaluation techniques by averaging their soft-max outputs. As can be seen, using multiple crops performs slightly better than dense evaluation, and the two approaches are indeed complementary, as their combination outperforms each of them. As noted above, we hypothesize that this is due to a different treatment of convolution boundary conditions.

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

4.4 CONVNET FUSION

Up until now, we evaluated the performance of individual ConvNet models. In this part of the experiments, we combine the outputs of several models by averaging their soft-max class posteriors. This improves the performance due to complementarity of the models, and was used in the top ILSVRC submissions in 2012 (Krizhevsky et al., 2012) and 2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014).

The results are shown in Table 6. By the time of ILSVRC submission we had only trained the single-scale networks, as well as a multi-scale model D (by fine-tuning only the fully-connected layers rather than all layers). The resulting ensemble of 7 networks has 7.3% ILSVRC test error. After the submission, we considered an ensemble of only two best-performing multi-scale models (configurations D and E), which reduced the test error to 7.0% using dense evaluation and 6.8% using combined dense and multi-crop evaluation. For reference, our best-performing single model achieves 7.1% error (model E, Table 5).

4.5 COMPARISON WITH THE STATE OF THE ART

Finally, we compare our results with the state of the art in Table 7. In the classification task of ILSVRC-2014 challenge (Russakovsky et al., 2014), our "VGG" team secured the 2nd place with

表4所示结果表明，测试时采用尺度抖动能带来更好的性能（相较于表3中单一尺度的模型评估）。与之前情况相同，最深层的配置（D和E）表现最佳，且尺度抖动优于采用固定最小边S的训练方式。我们在验证集上的最佳单网络性能为24.8%/7.5%的top-1/top-5错误率（表4中加粗显示）。在测试集上，配置E取得了7.3%的top-5错误率。

表4：ConvNet在多种测试尺度下的性能。

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
	256	224,256,288	27.7	9.2
C	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
	256	224,256,288	26.6	8.6
D	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
	256	224,256,288	26.9	8.7
E	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

4.3 多作物评估

在表5中，我们将密集卷积网络评估与多裁剪评估进行比较（详见第3.2节）。我们还通过平均两种评估技术的softmax输出来评估它们的互补性。可以看出，使用多裁剪方法的表现略优于密集评估，且两种方法确实具有互补性——它们的组合效果优于任一单独方法。如前所述，我们推测这是由于对卷积边界条件的不同处理方式所导致的。

表5：ConvNet评估技术对比。在所有实验中，训练尺度S从[256; 512]中采样，并考虑了三个测试尺度Q：{256, 384, 512}。

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
	dense	24.8	7.5
D	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
	dense	24.8	7.5
E	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

4.4 卷积网络融合

到目前为止，我们评估了单个ConvNet模型的性能。在本部分实验中，我们通过平均多个模型的soft-max类别后验概率来融合它们的输出。由于模型间的互补性，这一方法提升了整体性能，并在2012年（Krizhevsky等人，2012）和2013年（Zeiler & Fergus，2013；Sermanet等人，2014）的ILSVRC顶级参赛方案中得到应用。

结果如表6所示。在提交ILSVRC时，我们仅训练了单尺度网络，以及一个多尺度模型D（仅微调全连接层而非所有层）。由此得到的7个网络集成在ILSVRC测试集上错误率为7.3%。提交后，我们尝试仅集成两个表现最佳的多尺度模型（配置D和E），通过密集评估将测试错误率降至7.0%，结合密集与多裁剪评估后错误率进一步降至6.8%。作为参考，我们表现最佳的单模型错误率为7.1%（模型E，见表5）。

4.5 与现有技术的比较

最后，我们在表7中与现有技术进行了比较。在ILSVRC-2014挑战赛（Russakovsky等人，2014）的分类任务中，我们的“VGG”团队以{v*}的成绩获得了第二名。

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

7.3% test error using an ensemble of 7 models. After the submission, we decreased the error rate to 6.8% using an ensemble of 2 models.

As can be seen from Table 7, our very deep ConvNets significantly outperform the previous generation of models, which achieved the best results in the ILSVRC-2012 and ILSVRC-2013 competitions. Our result is also competitive with respect to the classification task winner (GoogLeNet with 6.7% error) and substantially outperforms the ILSVRC-2013 winning submission Clarifai, which achieved 11.2% with outside training data and 11.7% without it. This is remarkable, considering that our best result is achieved by combining just two models – significantly less than used in most ILSVRC submissions. In terms of the single-net performance, our architecture achieves the best result (7.0% test error), outperforming a single GoogLeNet by 0.9%. Notably, we did not depart from the classical ConvNet architecture of LeCun et al. (1989), but improved it by substantially increasing the depth.

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

5 CONCLUSION

In this work we evaluated very deep convolutional networks (up to 19 weight layers) for large-scale image classification. It was demonstrated that the representation depth is beneficial for the classification accuracy, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture (LeCun et al., 1989; Krizhevsky et al., 2012) with substantially increased depth. In the appendix, we also show that our models generalise well to a wide range of tasks and datasets, matching or outperforming more complex recognition pipelines built around less deep image representations. Our results yet again confirm the importance of depth in visual representations.

ACKNOWLEDGEMENTS

This work was supported by ERC grant VisRec no. 228180. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

表6：多ConvNet融合结果。

Combined ConvNet models	Error top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

使用7个模型的集成实现了7.3%的测试误差。提交后，我们通过2个模型的集成将错误率降低至6.8%。

从表7可以看出，我们极深的ConvNet显著超越了前几代模型——这些模型曾在ILSVRC-2012和ILSVRC-2013竞赛中取得最佳成绩。我们的结果在分类任务上也与冠军模型（GoogLeNet，错误率6.7%）具有竞争力，并大幅超越了ILSVRC-2013的获胜提交方案Clarifai（该方案在使用外部训练数据时错误率为11.2%，未使用时为11.7%）。值得注意的是，我们的最佳结果仅通过组合两个模型实现，这远少于大多数ILSVRC参赛方案所使用的模型数量。在单网络性能方面，我们的架构取得了最佳结果（测试错误率7.0%），比单GoogLeNet模型高出0.9%。需要强调的是，我们并未脱离LeCun等人（1989）提出的经典ConvNet架构，而是通过大幅增加深度对其进行了改进。

表7：与ILSVRC分类领域最新技术的比较。我们的方法标记为“VGG”。仅报告未使用外部训练数据获得的结果。

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

5 结论

在本工作中，我们评估了用于大规模图像分类的极深度卷积网络（多达19个权重层）。研究表明，表征深度有利于提升分类精度，并且通过显著增加传统ConvNet架构（LeCun等人，1989；Krizhevsky等人，2012）的深度，可以在ImageNet挑战数据集上达到最先进的性能。在附录中，我们还展示了我们的模型能够很好地泛化到各种任务和数据集，其表现与基于较浅图像表征构建的、更复杂的识别流程相当甚至更优。我们的结果再次证实了深度在视觉表征中的重要性。

致谢

本工作由欧洲研究理事会视觉识别项目（ERC grant VisRec no. 228180）资助。我们衷心感谢英伟达公司为本研究捐赠GPU设备。

REFERENCES

Bell, S., Upchurch, P., Snavely, N., and Bala, K. Material recognition in the wild with the materials in context database. *CoRR*, abs/1412.0623, 2014.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC.*, 2014.

Cimpoi, M., Maji, S., and Vedaldi, A. Deep convolutional filter banks for texture recognition and segmentation. *CoRR*, abs/1411.6836, 2014.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In *IJCAI*, pp. 1237–1242, 2011.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. Large scale distributed deep networks. In *NIPS*, pp. 1232–1240, 2012.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. The Pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision*, 2004.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524v5, 2014. Published in Proc. CVPR, 2014.

Gkioxari, G., Girshick, R., and Malik, J. Actions and attributes from wholes and parts. *CoRR*, abs/1412.2604, 2014.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proc. AISTATS*, volume 9, pp. 249–256, 2010.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *Proc. ICLR*, 2014.

Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729v2, 2014.

Hoai, M. Regularized max pooling for image categorization. In *Proc. BMVC.*, 2014.

Howard, A. G. Some improvements on deep convolutional neural network based image classification. In *Proc. ICLR*, 2014.

Jia, Y. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/`, 2013.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306, 2014.

Kiros, R., Salakhutdinov, R., and Zemel, R. S. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *NIPS*, pp. 1106–1114, 2012.

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

Lin, M., Chen, Q., and Yan, S. Network in network. In *Proc. ICLR*, 2014.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proc. CVPR*, 2014.

Perronnin, F., Sánchez, J., and Mensink, T. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.

Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features off-the-shelf: an Astounding Baseline for Recognition. *CoRR*, abs/1403.6382, 2014.

参考文献

Bell, S., Upchurch, P., Snavely, N., and Bala, K. 在真实场景下的材料识别：基于Materials in Context数据库。*CoRR*, abs/1412.0623, 2014. Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. 细节中的魔鬼回归：深入探究卷积网络。In *Proc. BMVC.*, 2014. Cimpoi, M., Maji, S., and Vedaldi, A. 用于纹理识别与分割的深度卷积滤波器组。*CoRR*, abs/1411.6836, 2014. Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. 用于图像分类的灵活高性能卷积神经网络。In *IJCAI*, pp. 1237–1242, 2011. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. 大规模分布式深度网络。In *NIPS*, pp. 1232–1240, 2012. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet：一个大规模分层图像数据库。In *Proc. CVPR*, 2009. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf：用于通用视觉识别的深度卷积激活特征。*CoRR*, abs/1310.1531, 2013. Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. Pascal视觉对象分类挑战：回顾。*IJCV*, 111(1):98–136, 2015. Fei-Fei, L., Fergus, R., and Perona, P. 从少量训练样本中学习生成式视觉模型：在101个对象类别上测试的增量贝叶斯方法。In *IEEE CVPR Workshop of Generative Model Based Vision*, 2004. Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. 用于精确目标检测与语义分割的丰富特征层次结构。*CoRR*, abs/1311.2524v5, 2014. 发表于Proc. CVPR, 2014. Gkioxari, G., Girshick, R., and Malik, J. 从整体与部分中获取动作与属性。*CoRR*, abs/1412.2604, 2014. Glorot, X. and Bengio, Y. 理解训练深度前馈神经网络的困难。In *Proc. AISTATS*, volume 9, pp. 249–256, 2010. Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. 使用深度卷积神经网络从街景图像中识别多位数数字。In *Proc. ICLR*, 2014. Griffin, G., Holub, A., and Perona, P. Caltech-256 对象类别数据集。技术报告 7694, 加州理工学院, 2007. He, K., Zhang, X., Ren, S., and Sun, J. 深度卷积网络中用于视觉识别的空间金字塔池化。*CoRR*, abs/1406.4729v2, 2014. Hoai, M. 用于图像分类的正则化最大池化。In *Proc. BMVC.*, 2014. Howard, A. G. 基于深度卷积神经网络的图像分类的一些改进。In *Proc. ICLR*, 2014. Jia, Y. Caffe：一个用于快速特征嵌入的开源卷积架构。http://caffe.berkeleyvision.org/, 2013. Karpathy, A. and Fei-Fei, L. 用于生成图像描述的深度视觉-语义对齐。*CoRR*, abs/1412.2306, 2014. Kiros, R., Salakhutdinov, R., and Zemel, R. S. 用多模态神经语言模型统一视觉-语义嵌入。*CoRR*, abs/1411.2539, 2014. Krizhevsky, A. 并行化卷积神经网络的一个巧妙技巧。*CoRR*, abs/1404.5997, 2014. Krizhevsky, A., Sutskever, I., and Hinton, G. E. 使用深度卷积神经网络进行ImageNet分类。In *NIPS*, pp. 1106–1114, 2012. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. 反向传播应用于手写邮政编码识别。*Neural Computation*, 1(4):541–551, 1989. Lin, M., Chen, Q., and Yan, S. 网络中的网络。In *Proc. ICLR*, 2014. Long, J., Shelhamer, E., and Darrell, T. 用于语义分割的全卷积网络。*CoRR*, abs/1411.4038, 2014. Oquab, M., Bottou, L., Laptev, I., and Sivic, J. 使用卷积神经网络学习和迁移中层图像表示。In *Proc. CVPR*, 2014. Perronnin, F., Sánchez, J., and Mensink, T. 改进用于大规模图像分类的Fisher核。In *Proc. ECCV*, 2010. Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. 现成的CNN特征：一个惊人的识别基线。*CoRR*, abs/1403.6382, 2014.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *Proc. ICLR*, 2014.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014. Published in Proc. NIPS, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. CNN: Single-label to multi-label. *CoRR*, abs/1406.5726, 2014.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. Published in Proc. ECCV, 2014.

A LOCALISATION

In the main body of the paper we have considered the classification task of the ILSVRC challenge, and performed a thorough evaluation of ConvNet architectures of different depth. In this section, we turn to the localisation task of the challenge, which we have won in 2014 with 25.3% error. It can be seen as a special case of object detection, where a single object bounding box should be predicted for each of the top-5 classes, irrespective of the actual number of objects of the class. For this we adopt the approach of Sermanet et al. (2014), the winners of the ILSVRC-2013 localisation challenge, with a few modifications. Our method is described in Sect. A.1 and evaluated in Sect. A.2.

A.1 LOCALISATION CONVNET

To perform object localisation, we use a very deep ConvNet, where the last fully connected layer predicts the bounding box location instead of the class scores. A bounding box is represented by a 4-D vector storing its center coordinates, width, and height. There is a choice of whether the bounding box prediction is shared across all classes (single-class regression, SCR (Sermanet et al., 2014)) or is class-specific (per-class regression, PCR). In the former case, the last layer is 4-D, while in the latter it is 4000-D (since there are 1000 classes in the dataset). Apart from the last bounding box prediction layer, we use the ConvNet architecture D (Table 1), which contains 16 weight layers and was found to be the best-performing in the classification task (Sect. 4).

Training. Training of localisation ConvNets is similar to that of the classification ConvNets (Sect. 3.1). The main difference is that we replace the logistic regression objective with a Euclidean loss, which penalises the deviation of the predicted bounding box parameters from the ground-truth. We trained two localisation models, each on a single scale: $S = 256$ and $S = 384$ (due to the time constraints, we did not use training scale jittering for our ILSVRC-2014 submission). Training was initialised with the corresponding classification models (trained on the same scales), and the initial learning rate was set to 10^{-3}. We explored both fine-tuning all layers and fine-tuning only the first two fully-connected layers, as done in (Sermanet et al., 2014). The last fully-connected layer was initialised randomly and trained from scratch.

Testing. We consider two testing protocols. The first is used for comparing different network modifications on the validation set, and considers only the bounding box prediction for the ground truth class (to factor out the classification errors). The bounding box is obtained by applying the network only to the central crop of the image.

The second, fully-fledged, testing procedure is based on the dense application of the localisation ConvNet to the whole image, similarly to the classification task (Sect. 3.2). The difference is that instead of the class score map, the output of the last fully-connected layer is a set of bounding box predictions. To come up with the final prediction, we utilise the greedy merging procedure of Sermanet et al. (2014), which first merges spatially close predictions (by averaging their coordinates), and then rates them based on the class scores, obtained from the classification ConvNet. When several localisation ConvNets are used, we first take the union of their sets of bounding box predictions, and then run the merging procedure on the union. We did not use the multiple pooling

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet大规模视觉识别挑战赛。*CoRR*, abs/1409.0575, 2014. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat：使用卷积网络进行集成识别、定位与检测。于 *Proc. ICLR*, 2014. Simonyan, K. and Zisserman, A. 用于视频动作识别的双流卷积网络。*CoRR*, abs/1406.2199, 2014. 发表于 Proc. NIPS, 2014. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. 深入探索卷积网络。*CoRR*, abs/1409.4842, 2014. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. CNN：从单标签到多标签。*CoRR*, abs/1406.5726, 2014. Zeiler, M. D. and Fergus, R. 卷积网络的可视化与理解。*CoRR*, abs/1311.2901, 2013. 发表于 Proc. ECCV, 2014.

本地化

在论文主体部分，我们探讨了ILSVRC挑战赛的分类任务，并对不同深度的卷积网络架构进行了全面评估。本节我们将转向该挑战赛的定位任务——我们曾于2014年以25.3%的错误率赢得该任务。该任务可视为目标检测的特殊案例：无论实际存在多少个目标实例，都需要为每个top-5类别预测单个目标边界框。为此，我们借鉴了ILSVRC-2013定位挑战赛冠军Sermanet等人（2014）的方法，并进行了若干改进。具体方法详见A.1节，评估结果见A.2节。

A.1 局部化卷积网络

为了实现目标定位，我们采用了一个非常深的卷积网络，其中最后一个全连接层用于预测边界框的位置，而非类别分数。边界框由一个4维向量表示，存储其中心坐标、宽度和高度。边界框预测可以选择在所有类别间共享（单类别回归，SCR（Sermanet等人，2014）），或是针对每个类别独立预测（每类别回归，PCR）。在前一种情况下，最后一层为4维；而在后一种情况下，由于数据集中包含1000个类别，最后一层为4000维。除了最后的边界框预测层外，我们使用了卷积网络架构D（表1），该架构包含16个权重层，在分类任务中表现最佳（第4节）。

训练。定位卷积网络的训练与分类卷积网络的训练类似（见第3.1节）。主要区别在于，我们将逻辑回归目标替换为欧几里得损失，该损失会惩罚预测边界框参数与真实值之间的偏差。我们训练了两个定位模型，每个模型均采用单一尺度：$S = 256$ 和 $S = 384$（由于时间限制，我们在ILSVRC-2014提交中未使用训练尺度抖动）。训练以相应的分类模型（在相同尺度上训练）进行初始化，初始学习率设置为10^{-3}。我们探索了微调所有层以及仅微调前两个全连接层两种方式，如（Sermanet等人，2014）所述。最后一个全连接层被随机初始化并从头开始训练。

测试。我们考虑两种测试协议。第一种用于在验证集上比较不同的网络修改，并且只考虑真实类别的边界框预测（以排除分类错误的影响）。边界框是通过仅对图像的中心裁剪区域应用网络来获得的。

第二种成熟的测试程序基于将定位卷积网络密集应用于整个图像，类似于分类任务（第3.2节）。不同之处在于，最后一个全连接层的输出不是类别分数图，而是一组边界框预测。为了得出最终预测，我们采用了Sermanet等人（2014）的贪婪合并程序：首先合并空间上接近的预测（通过平均其坐标），然后根据从分类卷积网络获得的类别分数对其进行评级。当使用多个定位卷积网络时，我们首先取其边界框预测集合的并集，然后在该并集上运行合并程序。我们未使用多重池化

offsets technique of Sermanet et al. (2014), which increases the spatial resolution of the bounding box predictions and can further improve the results.

A.2 LOCALISATION EXPERIMENTS

In this section we first determine the best-performing localisation setting (using the first test protocol), and then evaluate it in a fully-fledged scenario (the second protocol). The localisation error is measured according to the ILSVRC criterion (Russakovsky et al., 2014), i.e. the bounding box prediction is deemed correct if its intersection over union ratio with the ground-truth bounding box is above 0.5.

Settings comparison. As can be seen from Table 8, per-class regression (PCR) outperforms the class-agnostic single-class regression (SCR), which differs from the findings of Sermanet et al. (2014), where PCR was outperformed by SCR. We also note that fine-tuning all layers for the localisation task leads to noticeably better results than fine-tuning only the fully-connected layers (as done in (Sermanet et al., 2014)). In these experiments, the smallest images side was set to $S = 384$; the results with $S = 256$ exhibit the same behaviour and are not shown for brevity.

Table 8: **Localisation error for different modifications** with the simplified testing protocol: the bounding box is predicted from a single central image crop, and the ground-truth class is used. All ConvNet layers (except for the last one) have the configuration D (Table 1), while the last layer performs either single-class regression (SCR) or per-class regression (PCR).

Fine-tuned layers	regression type	GT class localisation error
1st and 2nd FC	SCR	36.4
	PCR	34.3
all	PCR	**33.1**

Fully-fledged evaluation. Having determined the best localisation setting (PCR, fine-tuning of all layers), we now apply it in the fully-fledged scenario, where the top-5 class labels are predicted using our best-performing classification system (Sect. 4.5), and multiple densely-computed bounding box predictions are merged using the method of Sermanet et al. (2014). As can be seen from Table 9, application of the localisation ConvNet to the whole image substantially improves the results compared to using a center crop (Table 8), despite using the top-5 predicted class labels instead of the ground truth. Similarly to the classification task (Sect. 4), testing at several scales and combining the predictions of multiple networks further improves the performance.

Table 9: **Localisation error**

smallest image side		top-5 localisation error (%)	
train (S)	test (Q)	val.	test.
256	256	29.5	-
384	384	28.2	26.7
384	352,384	27.5	-
fusion: 256/256 and 384/352,384		**26.9**	**25.3**

Comparison with the state of the art. We compare our best localisation result with the state of the art in Table 10. With 25.3% test error, our "VGG" team won the localisation challenge of ILSVRC-2014 (Russakovsky et al., 2014). Notably, our results are considerably better than those of the ILSVRC-2013 winner Overfeat (Sermanet et al., 2014), even though we used less scales and did not employ their resolution enhancement technique. We envisage that better localisation performance can be achieved if this technique is incorporated into our method. This indicates the performance advancement brought by our very deep ConvNets – we got better results with a simpler localisation method, but a more powerful representation.

B GENERALISATION OF VERY DEEP FEATURES

In the previous sections we have discussed training and evaluation of very deep ConvNets on the ILSVRC dataset. In this section, we evaluate our ConvNets, pre-trained on ILSVRC, as feature

Sermanet等人（2014）提出的偏移量技术，该技术提高了边界框预测的空间分辨率，并能进一步改善结果。

A.2 本地化实验

在本节中，我们首先确定性能最佳的定位设置（使用第一个测试协议），然后在完整场景（第二个协议）中对其进行评估。定位误差根据ILSVRC标准（Russakovsky等人，2014）进行衡量，即当预测边界框与真实边界框的交并比高于0.5时，该预测被视为正确。

设置对比。从表8可以看出，按类别回归（PCR）优于类别无关的单类别回归（SCR），这与Sermanet等人（2014）的研究结果不同，后者发现SCR优于PCR。我们还注意到，为定位任务微调所有层比仅微调全连接层（如Sermanet等人（2014）所做）能带来明显更好的结果。在这些实验中，图像最小边长设置为$S = 384$；使用$S = 256$的结果表现出相同趋势，为简洁起见未予展示。

表8：采用简化测试协议时不同修改方案的定位误差：边界框由单个中心图像裁剪预测，并使用真实类别。所有ConvNet层（除最后一层外）均采用配置D（表1），而最后一层执行单类别回归（SCR）或每类别回归（PCR）。

Fine-tuned layers	regression type	GT class localisation error
1st and 2nd FC	SCR	36.4
	PCR	34.3
all	PCR	**33.1**

全面评估。在确定了最佳定位设置（PCR，全层微调）后，我们现在将其应用于完整场景中：使用我们性能最佳的分类系统（第4.5节）预测前5个类别标签，并采用Sermanet等人（2014）的方法融合多个密集计算的边界框预测结果。从表9可以看出，尽管使用了预测的前5类别标签而非真实标签，但将定位卷积神经网络应用于整张图像的结果相比使用中心裁剪（表8）仍有显著提升。与分类任务（第4节）类似，在多尺度下进行测试并融合多个网络的预测结果能进一步提升性能。

表9：定位误差

smallest image side		top-5 localisation error (%)	
train (S)	test (Q)	val.	test.
256	256	29.5	-
384	384	28.2	26.7
384	352,384	27.5	-
fusion: 256/256 and 384/352,384		**26.9**	**25.3**

与现有技术的比较。我们在表10中将我们最佳的定位结果与现有技术进行了比较。以25.3%的测试误差，我们的“VGG”团队赢得了ILSVRC-2014的定位挑战赛（Russakovsky等人，2014年）。值得注意的是，即使我们使用了更少的尺度且未采用其分辨率增强技术，我们的结果仍显著优于ILSVRC-2013的获胜者Overfeat（Sermanet等人，2014年）。我们预计，若将此项技术融入我们的方法中，可以实现更好的定位性能。这表明了我们极深卷积网络带来的性能提升——我们以更简单的定位方法但更强大的表征能力获得了更好的结果。

B 极深特征的泛化

在前面的章节中，我们已经讨论了在ILSVRC数据集上训练和评估极深卷积网络的方法。本节中，我们将评估这些在ILSVRC上预训练的卷积网络作为特征提取器的表现。

Table 10: **Comparison with the state of the art in ILSVRC localisation**. Our method is denoted as "VGG".

Method	top-5 val. error (%)	top-5 test error (%)
VGG	**26.9**	**25.3**
GoogLeNet (Szegedy et al., 2014)	-	26.7
OverFeat (Sermanet et al., 2014)	30.0	29.9
Krizhevsky et al. (Krizhevsky et al., 2012)	-	34.2

extractors on other, smaller, datasets, where training large models from scratch is not feasible due to over-fitting. Recently, there has been a lot of interest in such a use case (Zeiler & Fergus, 2013; Donahue et al., 2013; Razavian et al., 2014; Chatfield et al., 2014), as it turns out that deep image representations, learnt on ILSVRC, generalise well to other datasets, where they have outperformed hand-crafted representations by a large margin. Following that line of work, we investigate if our models lead to better performance than more shallow models utilised in the state-of-the-art methods. In this evaluation, we consider two models with the best classification performance on ILSVRC (Sect. 4) – configurations "Net-D" and "Net-E" (which we made publicly available).

To utilise the ConvNets, pre-trained on ILSVRC, for image classification on other datasets, we remove the last fully-connected layer (which performs 1000-way ILSVRC classification), and use 4096-D activations of the penultimate layer as image features, which are aggregated across multiple locations and scales. The resulting image descriptor is L_2-normalised and combined with a linear SVM classifier, trained on the target dataset. For simplicity, pre-trained ConvNet weights are kept fixed (no fine-tuning is performed).

Aggregation of features is carried out in a similar manner to our ILSVRC evaluation procedure (Sect. 3.2). Namely, an image is first rescaled so that its smallest side equals Q, and then the network is densely applied over the image plane (which is possible when all weight layers are treated as convolutional). We then perform global average pooling on the resulting feature map, which produces a 4096-D image descriptor. The descriptor is then averaged with the descriptor of a horizontally flipped image. As was shown in Sect. 4.2, evaluation over multiple scales is beneficial, so we extract features over several scales Q. The resulting multi-scale features can be either stacked or pooled across scales. Stacking allows a subsequent classifier to learn how to optimally combine image statistics over a range of scales; this, however, comes at the cost of the increased descriptor dimensionality. We return to the discussion of this design choice in the experiments below. We also assess late fusion of features, computed using two networks, which is performed by stacking their respective image descriptors.

Table 11: **Comparison with the state of the art in image classification on VOC-2007, VOC-2012, Caltech-101, and Caltech-256**. Our models are denoted as "VGG". Results marked with * were achieved using ConvNets pre-trained on the *extended* ILSVRC dataset (2000 classes).

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	$\mathbf{93.4 \pm 0.5}$	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	$\mathbf{86.2 \pm 0.3}$

Image Classification on VOC-2007 and VOC-2012. We begin with the evaluation on the image classification task of PASCAL VOC-2007 and VOC-2012 benchmarks (Everingham et al., 2015). These datasets contain 10K and 22.5K images respectively, and each image is annotated with one or several labels, corresponding to 20 object categories. The VOC organisers provide a pre-defined split into training, validation, and test data (the test data for VOC-2012 is not publicly available; instead, an official evaluation server is provided). Recognition performance is measured using mean average precision (mAP) across classes.

Notably, by examining the performance on the validation sets of VOC-2007 and VOC-2012, we found that aggregating image descriptors, computed at multiple scales, by averaging performs sim-

表10：与ILSVRC定位领域最新技术的比较。我们的方法表示为
作为“VGG”。

Method	top-5 val. error (%)	top-5 test error (%)
VGG	**26.9**	**25.3**
GoogLeNet (Szegedy et al., 2014)	-	26.7
OverFeat (Sermanet et al., 2014)	30.0	29.9
Krizhevsky et al. (Krizhevsky et al., 2012)	-	34.2

在其他较小数据集上，由于从头训练大型模型容易导致过拟合而不可行时，特征提取器便显得尤为重要。最近，此类应用场景引起了广泛关注（Zeiler & Fergus, 2013; Donahue et al., 2013; Razavian et al., 2014; Chatfield et al., 2014），因为研究发现，在ILSVRC上学到的深度图像表征能够很好地迁移到其他数据集，并且大幅超越了手工设计的特征表示。沿着这一研究方向，我们探讨了我们的模型是否能够比现有先进方法中使用的较浅层模型带来更好的性能。在此评估中，我们采用了在ILSVRC上分类性能最佳的两个模型（第4节）——即我们已公开的“Net-D”和“Net-E”配置。

为了利用在ILSVRC上预训练的卷积神经网络（ConvNets）进行其他数据集的图像分类，我们移除了最后一个全连接层（该层执行1000类ILSVRC分类），并将倒数第二层的4096维激活值作为图像特征，这些特征在多个位置和尺度上进行聚合。生成的图像描述符经过L_2归一化处理，并与在线性SVM分类器结合，该分类器在目标数据集上进行训练。为简化流程，预训练的卷积神经网络权重保持固定（不进行微调）。

特征聚合的方式与我们的ILSVRC评估流程（第3.2节）类似。具体而言，首先将图像重新缩放，使其最短边等于Q，然后在图像平面上密集应用网络（当所有权重层均被视为卷积层时，此操作可行）。接着对生成的特征图进行全局平均池化，得到一个4096维的图像描述符。该描述符随后与水平翻转图像对应的描述符进行平均。如第4.2节所示，多尺度评估具有优势，因此我们在多个尺度Q上提取特征。生成的多尺度特征可通过跨尺度堆叠或池化处理。堆叠方式允许后续分类器学习如何在一系列尺度上最优地组合图像统计信息，但这会以增加描述符维度为代价。我们将在后续实验中重新讨论这一设计选择。此外，我们还评估了使用两个网络计算特征的后期融合方法，该方法通过堆叠各自网络的图像描述符实现。

表11：在VOC-2007、VOC-2012、Caltech-101和Caltech-256数据集上与图像分类领域先进技术的比较。我们的模型标记为“VGG”。标有*的结果是使用在*extended* ILSVRC数据集（2000个类别）上预训练的ConvNets实现的。

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	**93.4** ± **0.5**	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	**86.2** ± **0.3**

在VOC-2007和VOC-2012上进行图像分类。我们首先评估PASCAL VOC-2007和VOC-2012基准测试（Everingham等人，2015）中的图像分类任务。这些数据集分别包含10K和22.5K张图像，每张图像标注有一个或多个标签，对应20个物体类别。VOC组织方提供了预定义的训练集、验证集和测试集划分（VOC-2012的测试集未公开提供，而是通过官方评估服务器进行评测）。识别性能采用各类别平均精度均值（mAP）进行衡量。

值得注意的是，通过检查在VOC-2007和VOC-2012验证集上的表现，我们发现通过平均聚合多个尺度下计算出的图像描述符，其性能表现相

ilarly to the aggregation by stacking. We hypothesize that this is due to the fact that in the VOC dataset the objects appear over a variety of scales, so there is no particular scale-specific semantics which a classifier could exploit. Since averaging has a benefit of not inflating the descriptor dimensionality, we were able to aggregated image descriptors over a wide range of scales: $Q \in \{256, 384, 512, 640, 768\}$. It is worth noting though that the improvement over a smaller range of $\{256, 384, 512\}$ was rather marginal (0.3%).

The test set performance is reported and compared with other approaches in Table 11. Our networks "Net-D" and "Net-E" exhibit identical performance on VOC datasets, and their combination slightly improves the results. Our methods set the new state of the art across image representations, pre-trained on the ILSVRC dataset, outperforming the previous best result of Chatfield et al. (2014) by more than 6%. It should be noted that the method of Wei et al. (2014), which achieves 1% better mAP on VOC-2012, is pre-trained on an extended 2000-class ILSVRC dataset, which includes additional 1000 categories, semantically close to those in VOC datasets. It also benefits from the fusion with an object detection-assisted classification pipeline.

Image Classification on Caltech-101 and Caltech-256. In this section we evaluate very deep features on Caltech-101 (Fei-Fei et al., 2004) and Caltech-256 (Griffin et al., 2007) image classification benchmarks. Caltech-101 contains 9K images labelled into 102 classes (101 object categories and a background class), while Caltech-256 is larger with 31K images and 257 classes. A standard evaluation protocol on these datasets is to generate several random splits into training and test data and report the average recognition performance across the splits, which is measured by the mean class recall (which compensates for a different number of test images per class). Following Chatfield et al. (2014); Zeiler & Fergus (2013); He et al. (2014), on Caltech-101 we generated 3 random splits into training and test data, so that each split contains 30 training images per class, and up to 50 test images per class. On Caltech-256 we also generated 3 splits, each of which contains 60 training images per class (and the rest is used for testing). In each split, 20% of training images were used as a validation set for hyper-parameter selection.

We found that unlike VOC, on Caltech datasets the stacking of descriptors, computed over multiple scales, performs better than averaging or max-pooling. This can be explained by the fact that in Caltech images objects typically occupy the whole image, so multi-scale image features are semantically different (capturing the whole object *vs*. object parts), and stacking allows a classifier to exploit such scale-specific representations. We used three scales $Q \in \{256, 384, 512\}$.

Our models are compared to each other and the state of the art in Table 11. As can be seen, the deeper 19-layer Net-E performs better than the 16-layer Net-D, and their combination further improves the performance. On Caltech-101, our representations are competitive with the approach of He et al. (2014), which, however, performs significantly worse than our nets on VOC-2007. On Caltech-256, our features outperform the state of the art (Chatfield et al., 2014) by a large margin (8.6%).

Action Classification on VOC-2012. We also evaluated our best-performing image representation (the stacking of Net-D and Net-E features) on the PASCAL VOC-2012 action classification task (Everingham et al., 2015), which consists in predicting an action class from a single image, given a bounding box of the person performing the action. The dataset contains 4.6K training images, labelled into 11 classes. Similarly to the VOC-2012 object classification task, the performance is measured using the mAP. We considered two training settings: (i) computing the ConvNet features on the whole image and ignoring the provided bounding box; (ii) computing the features on the whole image and on the provided bounding box, and stacking them to obtain the final representation. The results are compared to other approaches in Table 12.

Our representation achieves the state of art on the VOC action classification task even without using the provided bounding boxes, and the results are further improved when using both images and bounding boxes. Unlike other approaches, we did not incorporate any task-specific heuristics, but relied on the representation power of very deep convolutional features.

Other Recognition Tasks. Since the public release of our models, they have been actively used by the research community for a wide range of image recognition tasks, consistently outperforming more shallow representations. For instance, Girshick et al. (2014) achieve the state of the object detection results by replacing the ConvNet of Krizhevsky et al. (2012) with our 16-layer model. Similar gains over a more shallow architecture of Krizhevsky et al. (2012) have been ob-

类似于通过堆叠进行的聚合。我们假设这是由于在VOC数据集中，物体出现在多种尺度上，因此没有特定的尺度特定语义可供分类器利用。由于平均法具有不增加描述符维度的优势，我们能够在广泛的尺度范围内聚合图像描述符：$Q \in \{256, 384, 512, 640, 768\}$。值得注意的是，与较小范围{256, 384, 512}相比，改进幅度相当有限（0.3%）。

测试集性能在表11中进行了报告并与其他方法进行了比较。我们的网络“Net-D”和“Net-E”在VOC数据集上表现出相同的性能，二者的组合略微提升了结果。我们的方法在使用ILSVRC数据集预训练的图像表征方面创造了新的最优性能，比Chatfield等人（2014）先前的最佳结果高出6%以上。需要注意的是，Wei等人（2014）的方法在VOC-2012上实现了高出1%的mAP，但其使用了扩展的2000类ILSVRC数据集进行预训练，该数据集额外包含了1000个与VOC数据集语义相近的类别。该方法还受益于与目标检测辅助分类流程的融合。

在Caltech-101和Caltech-256上的图像分类。本节中，我们在Caltech-101（Fei-Fei等人，2004年）和Caltech-256（Griffin等人，2007年）图像分类基准上评估了非常深的特征。Caltech-101包含9K张图像，标记为102个类别（101个对象类别和一个背景类别），而Caltech-256更大，包含31K张图像和257个类别。这些数据集的标准评估协议是生成多个随机划分，将数据分为训练集和测试集，并报告各划分间的平均识别性能，该性能通过平均类别召回率来衡量（这补偿了每个类别测试图像数量的不同）。遵循Chatfield等人（2014年）、Zeiler & Fergus（2013年）、He等人（2014年）的方法，在Caltech-101上，我们生成了3个随机划分，每个划分包含每个类别30张训练图像，以及最多每个类别50张测试图像。在Caltech-256上，我们也生成了3个划分，每个划分包含每个类别60张训练图像（其余用于测试）。在每个划分中，20%的训练图像被用作超参数选择的验证集。

我们发现，与VOC不同，在Caltech数据集上，通过多尺度计算得到的描述符堆叠比平均池化或最大池化表现更佳。这可以解释为，在Caltech图像中，物体通常占据整个图像，因此多尺度图像特征在语义上有所不同（捕捉整个物体*vs*或物体部分），而堆叠能让分类器利用这种特定尺度的表征。我们使用了三种尺度$Q \in \{256,、384,和512\}$。

我们的模型在表11中进行了相互比较，并与现有最佳技术进行了对比。可以看出，更深的19层Net-E比16层Net-D表现更好，而它们的组合进一步提升了性能。在Caltech-101数据集上，我们的表征方法与He等人（2014）的方法具有可比性，但该方法在VOC-2007数据集上的表现明显逊于我们的网络。在Caltech-256数据集上，我们的特征以较大优势（8.6%）超越了现有最佳技术（Chatfield等人，2014）。

在VOC-2012上的动作分类。我们还在PASCAL VOC-2012动作分类任务（Everingham等人，2015）上评估了我们表现最佳的图像表示（Net-D和Net-E特征的堆叠），该任务旨在给定执行动作人物的边界框，从单张图像预测动作类别。该数据集包含4.6K训练图像，标注为11个类别。与VOC-2012物体分类任务类似，性能使用mAP进行衡量。我们考虑了两种训练设置：（i）在整个图像上计算ConvNet特征，忽略提供的边界框；（ii）在整个图像和提供的边界框上分别计算特征，并将它们堆叠以获得最终表示。结果与其他方法的比较见表12。

我们的表示方法即使在未使用提供的边界框的情况下，也在VOC动作分类任务上达到了当前最优水平，而当同时使用图像和边界框时，结果得到了进一步提升。与其他方法不同，我们未引入任何任务特定的启发式规则，而是依赖于极深度卷积特征的表征能力。

其他识别任务。自我们的模型公开发布以来，研究界已将其广泛应用于各类图像识别任务中，其表现持续优于较浅层的表征方法。例如，Girshick等人（2014年）通过用我们的16层模型替换Krizhevsky等人（2012年）的卷积网络，实现了当时最先进的物体检测结果。相较于Krizhevsky等人（2012年）的较浅层架构，类似的性能提升也已得到验证——

Table 12: **Comparison with the state of the art in single-image action classification on VOC-2012**. Our models are denoted as "VGG". Results marked with * were achieved using ConvNets pre-trained on the *extended* ILSVRC dataset (1512 classes).

Method	VOC-2012 (mean AP)
(Oquab et al., 2014)	70.2*
(Gkioxari et al., 2014)	73.6
(Hoai, 2014)	76.3
VGG Net-D & Net-E, image-only	**79.2**
VGG Net-D & Net-E, image and bounding box	**84.0**

served in semantic segmentation (Long et al., 2014), image caption generation (Kiros et al., 2014; Karpathy & Fei-Fei, 2014), texture and material recognition (Cimpoi et al., 2014; Bell et al., 2014).

C PAPER REVISIONS

Here we present the list of major paper revisions, outlining the substantial changes for the convenience of the reader.

v1 Initial version. Presents the experiments carried out before the ILSVRC submission.

v2 Adds post-submission ILSVRC experiments with training set augmentation using scale jittering, which improves the performance.

v3 Adds generalisation experiments (Appendix B) on PASCAL VOC and Caltech image classification datasets. The models used for these experiments are publicly available.

v4 The paper is converted to ICLR-2015 submission format. Also adds experiments with multiple crops for classification.

v6 Camera-ready ICLR-2015 conference paper. Adds a comparison of the net B with a shallow net and the results on PASCAL VOC action classification benchmark.

表12：在VOC-2012单图像动作分类任务上与前沿技术的比较。我们的模型标记为“VGG”。标有*的结果是使用在*extended* ILSVRC数据集（1512个类别）上预训练的ConvNets实现的。

Method	VOC-2012 (mean AP)
(Oquab et al., 2014)	70.2*
(Gkioxari et al., 2014)	73.6
(Hoai, 2014)	76.3
VGG Net-D & Net-E, image-only	**79.2**
VGG Net-D & Net-E, image and bounding box	**84.0**

在语义分割（Long等人，2014）、图像描述生成（Kiros等人，2014；Karpathy与Fei-Fei，2014）、纹理与材质识别（Cimpoi等人，2014；Bell等人，2014）等领域中得到应用。

C 论文修订

在此我们列出论文的主要修订内容，为方便读者概述实质性改动。

v1 初始版本。展示了在ILSVRC提交之前进行的实验。

v2 在提交后增加了使用尺度抖动进行训练集增强的ILSVRC实验，这提升了性能。

v3在PASCAL VOC和Caltech图像分类数据集上增加了泛化实验（附录B）。这些实验所使用的模型均已公开提供。

v4 论文已转换为ICLR-2015投稿格式，并增加了多裁剪分类实验。

v6 ICLR-2015会议最终版论文。增加了网络B与浅层网络的比较，以及在PASCAL VOC动作分类基准上的结果。