

数据稀缺，目前可用的数据量不足以训练大型卷积神经网络（CNN）。解决这个问题的传统方法是使用无监督预训练，随后进行有监督微调（例如，[35]）。本文的第二个主要贡献是展示了在大量辅助数据集（ILSVRC）上进行有监督预训练，然后在小规模特定领域数据集（PASCAL）上进行领域特定微调，是一种在数据稀缺情况下学习高容量CNN的有效范式。在我们的实验中，检测任务的微调使平均精度（mAP）提高了8个百分点。经过微调后，我们的系统在VOC 2010上的mAP达到了54%，相比之下，高度优化的基于HOG的可变形部件模型（DPM）[17, 20]的mAP为33%。我们还提请读者参阅Donahue等人的同期工作 [12]，他们表明Krizhevsky的CNN可以在不进行微调的情况下作为黑盒特征提取器，在包括场景分类、细粒度子分类和领域适应在内的多个识别任务中表现出色。

我们的系统也非常高效。唯一的类别特定计算是一个相对较小的矩阵向量乘法和贪婪非极大值抑制。这种计算特性源于所有类别共享的特征，并且这些特征比之前使用的区域特征低两个数量级（参见[39]）。理解我们方法的失败模式对于改进它至关重要，因此我们报告了Hoiem等人[23]的检测分析工具的结果。作为这一分析的直接结果，我们证明了一种简单的边界框回归方法显著减少了误定位，这是主要的错误模式。

在展开技术细节之前，我们注意到由于R-CNN操作于区域，因此自然可以将其扩展到语义分割任务。经过一些小的修改，我们在PASCAL VOC分割任务上也取得了具有竞争力的结果，在VOC 2011测试集上的平均分割准确率为47.9%。



特征提取。我们使用Krizhevsky等人[25]描述的CNN的Caffe[24]实现从每个区域提议中提取一个4096维的特征向量。特征是通过前向传播一个均值减去后的 227×227

RGB图像，经过五个卷积层和两个全连接层计算得到的。有关网络架构的更多细节，请参阅[24, 25]。为了计算区域提议的特征，我们必须首先将该区域内的图像数据转换为与CNN兼容的形式（其架构要求输入固定大小的 227×227 像素）。在许多可能的形状任意区域变换中，我们选择了最简单的。无论候选区域的大小或宽高比如何，我们都将其周围的紧致边界框内的所有像素变形到所需尺寸。在变形之前，我们会扩展紧致边界框，以便在变形后，在原始框周围恰好有 p 个像素的变形图像上下文（我们使用 $p=16$ ）。图2显示了一些随机采样的变形训练区域。关于变形的替代方案，请参见附录A。

目标类别分类器。考虑训练一个二元分类器来检测汽车。

显然，紧密包围一辆汽车的图像区域应该是一个正例。类似地，显然一个与汽车无关的背景区域应该是一个负例。不太清楚的是如何标注部分重叠于汽车的区域。我们通过交并比 (IoU) 重叠阈值解决了这个问题，在该阈值之下，区域被定义为负例。重叠阈值0.3是通过对{0, 0.1, ...,

0.5}进行网格搜索在验证集上选定的。我们发现仔细选择这个阈值很重要。将其设置为0.5（如[39]），会导致mAP下降5个百分点。同样，将其设置为0会导致mAP下降4个百分点。正例简单地定义为每个类别的真实边界框。

一旦提取了特征并应用了训练标签，我们针对每类优化一个线性支持向量机 (SVM)。由于训练数据过大无法全部放入内存，我们采用了标准的难负样本挖掘方法[17,

37]。难负样本挖掘收敛得很快，实际上mAP在仅遍历所有图像一次后就停止增加。

在附录B中，我们讨论了为什么在微调和SVM训练中正负样本的定义不同。我们还讨论了训练检测SVM而不是简单使用微调CNN的最终softmax层输出所涉及的权衡。

我们在200类的ILSVRC2013检测数据集上运行了R-CNN，使用与PASCAL VOC相同的系统超参数。我们按照同样的协议，仅两次提交测试结果到ILSVRC2013评估服务器，一次使用边界框回归，一次不使用。

图3比较了R-CNN与ILSVRC 2013竞赛中的其他参赛作品以及赛后OverFeat的结果【34】。R-CNN达到了31.4%的mAP，显著优于第二好的OverFeat的24.3%的结果。

为了展示各类别的AP分布情况，还提供了箱线图，并在本文末尾的表8中列出了每类别的AP值。大多数参赛作品 (OverFeat、NEC-MU、UvA-Euvision、Toronto A 和 UIUC-IFP) 都使用了卷积神经网络，表明CNN在目标检测中的应用存在显著差异，导致了不同的结果。

在第4节中，我们将概述ILSVRC2013检测数据集，并提供我们在其上运行R-CNN时所做选择的详细信息。

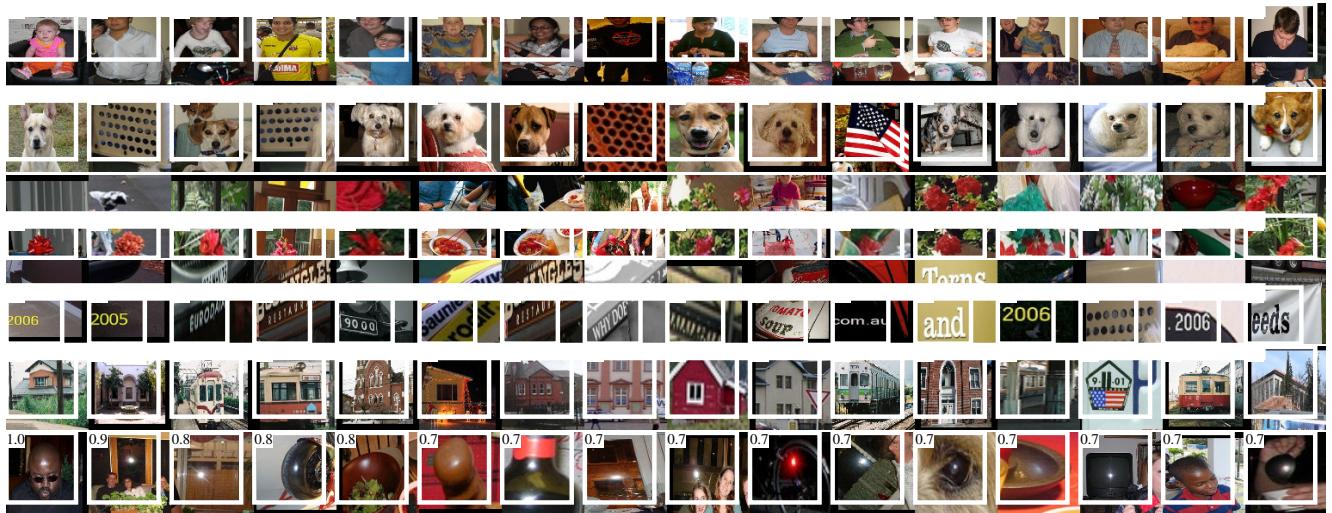
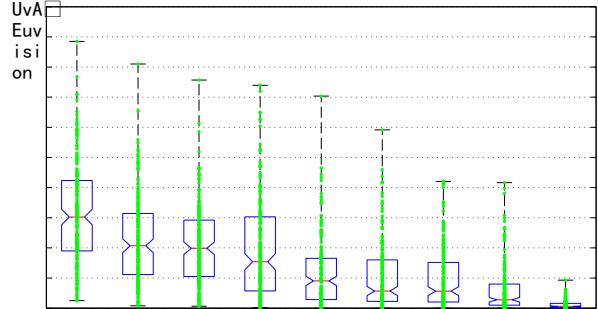
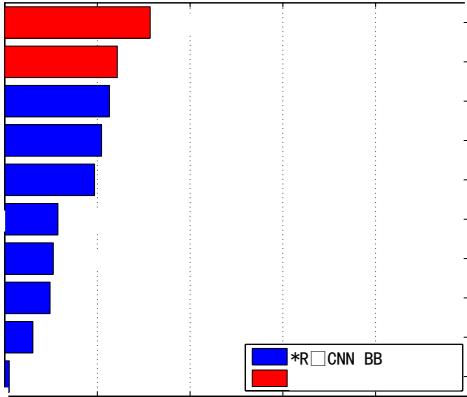


Figure 4: Top regions for six pool₅ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

为了在R-CNN中使用O-Net，我们从Caffe Model Zoo下载了VGG ILSVRC

16层模型的公开预训练网络权重。¹然后，我们按照与T-Net相同的协议对网络进行了微调。唯一的不同是使用较小的小批量（24个样本），以适应GPU内存的要求。表3的结果显示，使用O-Net的R-CNN显著优于使用T-Net的R-CNN，mAP从58.5%提高到了66.0%。然而，在计算时间方面存在相当大的缺点，O-Net的前向传递大约比T-Net慢7倍。

本文中的大多数结果使用了来自Krizhevsky等人的网络架构[25]。然而，我们发现架构的选择对R-CNN检测性能有很大影响。在表3中，我们展示了使用Simonyan和Zisserman最近提出的16层深度网络在VOC 2007测试集上的结果[43]。该网络是近期ILSVRC 2014分类挑战中的顶级表现者之一。该网络具有同质结构，包含13层 3×3 卷积核，并夹杂五层最大池化层，顶部有三层全连接层。我们将此网络称为“O-Net”（OxfordNet），并将基准网络称为“T-Net”（TorontoNet）。

¹<https://github.com/BVLC/caffe/wiki/Model-Zoo>

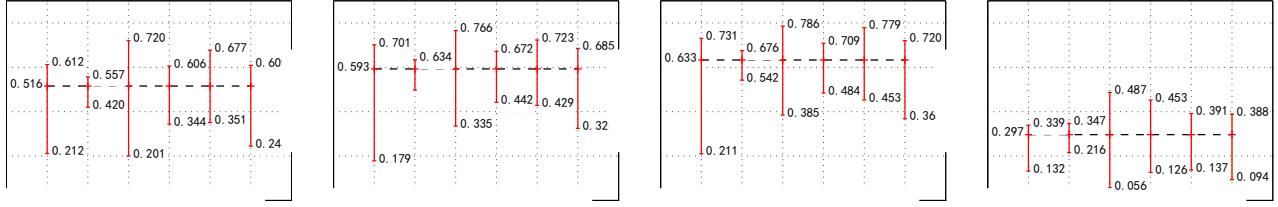


图6：对物体特征的敏感度。每个图表展示了在六种不同的物体特征（遮挡、截断、边界框面积、长宽比、视角、部分可见性）中表现最好和最差的子集的平均（按类别计算）归一化AP（参见[23]）。我们展示了我们的方法（R-CNN）在有和没有微调（FT）以及边界框回归（BB）的情况下的图表，以及DPM voc-release5的结果。总体而言，微调并没有降低敏感度（最大值与最小值之间的差异），但几乎在所有特征上都显著提升了表现最好的和最差的子集。这表明微调不仅仅是简单地改善了长宽比和边界框面积这两个方面的最低性能子集，就像根据我们对网络输入的扭曲方式所推测的那样。相反，微调提高了包括遮挡、截断、视角和部分可见性在内的所有特征的鲁棒性。

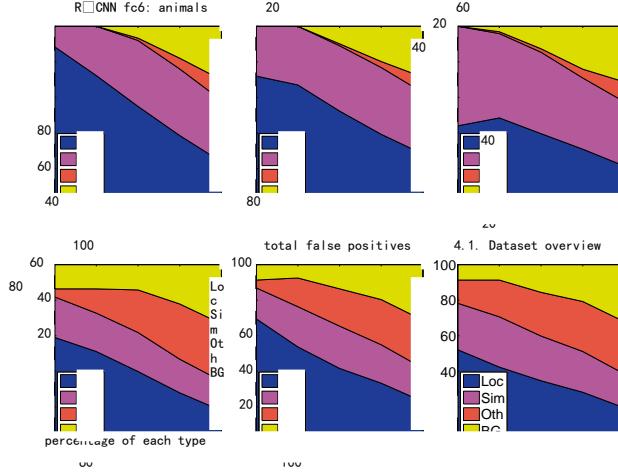


Figure 5: Distribution of top-ranked false positive (FP) types. Each plot shows the evolving distribution of FP types as more FPs are considered in order of decreasing score. Each FP is categorized into 1 of 4 types: Loc—poor localization (a detection with an IoU overlap with the correct class between 0.1 and 0.5, or a duplicate); Sim—confusion with a similar category; Oth—confusion with a dissimilar object category; BG—a FP that fired on background. Compared with DPM (see [23]), significantly more of our errors result from poor localization, rather than confusion with background or other object classes, indicating that the CNN features are much more discriminative than HOG. Loose localization likely results from our use of bottom-up region proposals and the positional invariance learned from pre-training the CNN for whole-image classification. Column three shows how our simple bounding-box regression method fixes many localization errors.

PASCAL VOC, requiring choices about how to use it. Since these decisions are non-trivial, we cover them in this section.

4.1. Dataset overview

The ILSVRC2013 detection dataset is split into three sets: train (395,918), val (20,121), and test (40,152), where the number of images in each set is in parentheses. The

val and test splits are drawn from the same image distribution. These images are scene-like and similar in complexity (number of objects, amount of clutter, pose variability, etc.) to PASCAL VOC images. The val and test splits are exhaustively annotated, meaning that in each image all instances from all 200 classes are labeled with bounding boxes. The train set, in contrast, is drawn from the ILSVRC2013 *classification* image distribution. These images have more variable complexity with a skew towards images of a single centered object. Unlike val and test, the train images (due to their large number) are not exhaustively annotated. In any given train image, instances from the 200 classes may or may not be labeled. In addition to these image sets, each class has an extra set of negative images. Negative images are manually checked to validate that they do not contain any instances of their associated class. The negative image sets were not used in this work. More information on how ILSVRC was collected and annotated can be found in [11, 36].

The nature of these splits presents a number of choices for training R-CNN. The train images cannot be used for hard negative mining, because annotations are not exhaustive. Where should negative examples come from? Also, the train images have different statistics than val and test. Should the train images be used at all, and if so, to what extent? While we have not thoroughly evaluated a large number of choices, we present what seemed like the most obvious path based on previous experience.

Our general strategy is to rely heavily on the val set and use some of the train images as an auxiliary source of positive examples. To use val for both training and validation, we split it into roughly equally sized “val₁” and “val₂” sets. Since some classes have very few examples in val (the smallest has only 31 and half have fewer than 110), it is important to produce an approximately class-balanced partition. To do this, a large number of candidate splits were generated and the one with the smallest maximum relative

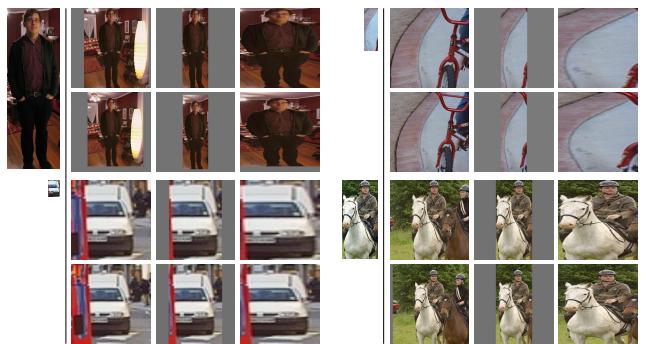
²Relative imbalance is measured as $|a - b|/(a + b)$ where a and b are class counts in each half of the split.

使用多尺度金字塔的正方形区域提议，并将每个类别的边界框回归器改为单一的边界框回归器，那么这些系统会非常相似（除了它们的训练方式可能存在一些显著差异：CNN检测微调、使用SVM等）。值得注意的是，OverFeat在速度上相对于R-CNN具有显著优势：它大约快9倍，根据引用[34]中每张图像2秒的数据。这种速度提升来自于OverFeat的滑动窗口（即区域提议）在图像级别上没有变形，因此可以在重叠窗口之间轻松共享计算。通过以卷积的方式在整个网络上运行任意大小的输入来实现共享。加速R-CNN的方法有很多，这仍然是未来的工作。

区域分类是一种标准的技术，用于语义分割，使我们能够轻松地将R-CNN应用于PASCAL VOC分割挑战中。为了与当前领先的语义分割系统（称为O2P，即“二级池化”）[4]进行直接比较，我们在他们的开源框架内工作。O2P使用CPMC为每张图像生成150个区域提案，然后针对每个类别使用支持向量回归（SVR）预测每个区域的质量。他们方法的高性能归功于CPMC区域的质量以及多种特征类型的强大二级池化（SIFT和LBP的丰富变体）。我们还注意到，Farabet等人[16]最近在几个密集场景标注数据集（不包括PASCAL）上使用CNN作为多尺度逐像素分类器取得了良好结果。我们遵循[2, 4]，并将PASCAL分割训练集扩展为包含Hariharan等人[22]提供的额外标注。设计决策和超参数在VOC 2011验证集上进行了交叉验证。最终测试结果只评估了一次。

结果在VOC 2011上。表5展示了我们在VOC 2011验证集上的结果，与O2P进行比较。（详见附录E中的完整类别结果。）在每种特征计算策略中，层fc6始终优于fc7，因此以下讨论均基于fc6特征。fg策略略优于full，表明掩膜区域形状提供了更强的信号，符合我们的直觉。然而，full+fg达到了47.9%的平均准确率，这是我们的最佳结果，比第二好的结果高出4.2%（也略微优于O2P），这表明即使有了fg特征，完整的特征所提供的上下文信息仍然非常丰富。值得注意的是，在单核处理器上，使用我们的full+fg特征训练20个支持向量机需要一个小时，而使用O2P特征则需要10多个小时。

表6展示了我们在VOC 2011测试集上的结果，对比了我们表现最好的方法fc6（full+fg）与其他两个强基准方法。我们的方法在21个类别中的11个类别中取得了最高的分割准确率，并且总体上实现了47.9%的最高平均分割准确率（跨类别平均，但在合理的误差范围内可能与O2P的结果持平）。通过微调，性能可能会进一步提高。



有两个设计选择值得进一步讨论。

第一个问题是：为什么在微调CNN和训练目标检测SVM时，正例和负例的定义不同？简要回顾一下定义，对于微调，我们将每个对象提议映射到与其具有最大IoU重叠的真实实例（如果有），如果IoU至少为0.5，则将其标记为匹配的真实类别的正例。所有其他提议都被标记为“背景”（即所有类别的负例）。而在训练SVM时，我们只取真实框作为各自类别的正例，并将与某一类的所有实例IoU重叠小于0.3的提议标记为该类的负例。落在灰色区域（IoU重叠大于0.3但不是真实值）的提议被忽略。

从历史角度来看，我们之所以采用这些定义是因为我们最初是在ImageNet预训练的CNN计算的特征上训练SVM，因此当时并未考虑微调。在这种设置下，我们发现我们所使用的特定标签定义在我们评估的一组选项中是最佳的（包括我们现在用于微调的设置）。当我们开始使用微调时，最初使用了与SVM训练相同的正例和负例定义。然而，我们发现结果远不如使用当前的正例和负例定义好。

我们的假设是，这种正例和负例定义的不同并不是根本重要的，而是由于微调数据有限导致的。目前的方案引入了许多“抖动”示例（那些IoU在0.5到1之间的提议，但不是真实值），这使正例的数量大约增加了30倍。我们推测，在微调整个网络以避免过拟合时，需要这样一个大集合。不过我们也注意到，使用这些“抖动”示例可能是次优的，因为网络没有被微调以实现精确定位。

这引出了第二个问题：为什么在微调之后还要训练SVM？更简洁的方法是直接应用微调后的网络的最后一层，这是一个21路softmax回归分类器，作为目标检测器。我们尝试了这种方法，发现VOC 2007上的性能从54.2% mAP下降到了50.9% mAP。这一性能下降可能源于几个因素的结合，包括微调中使用的正例定义没有强调精确定位，以及softmax分类器是在随机采样的负例而不是用于SVM训练的“难负例”子集上训练的。

这个结果显示，即使不训练SVM也可以获得接近相同水平的性能。

我们使用一个简单的边界框回归阶段来提高定位性能。在用特定类别的检测SVM对每个选择性搜索建议进行评分后，我们使用特定类别的边界框回归器预测一个新的检测边界框。

这与可变形部件模型[17]中使用的边界框回归在精神上相似。两种方法的主要区别在于，这里我们从CNN计算出的特征进行回归，而不是从推断出的DPM部件位置计算出的几何特征。

我们的训练算法的输入是一组N个训练样本对 $\{(P_i, G_i)\}_{i=1, \dots, N}$ ，其中 $P_i = (P_{ix}, P_{iy}, P_{iw},$

$P_{ih})$ 指定了提议 P_i 的边界框中心的像素坐标以及 P_i 的宽度和高度（以像素为单位）。接下来，除非需要，我们将省略上标 i 。每个真实边界框 G 也以同样的方式指定： $G = (G_x, G_y, G_w, G_h)$ 。我们的目标是学习一种将提议框 P 映射到真实边界框 G 的变换。

我们用四个函数 $dx(P)$ ， $dy(P)$ ， $dw(P)$ 和 $dh(P)$ 来参数化这种变换。前两个函数指定 P 的边界框中心的尺度不变平移，而后两个函数指定 P 的边界框宽度和高度的对称空间平移。学习这些函数后，我们可以通过对输入提议 P 应用该变换将其转换为预测的真实边界框 (\hat{G}) 。

