丰富的特征层次结构可实现精确的物体检测和语义分割 技术报告 (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

根据 PASCAL VOC 数据集的标准测量，物体检测性能在过去几年中已趋于稳定。性能最好的方法是复杂的集合系统，通常将多个低级图像特征与高级上下文结合在一起。在本文中，我们提出了一种简单、可扩展的检测算法，与 2012 年 VOC 数据集上之前的最佳结果相比，平均精度 (mAP) 提高了 30% 以上，达到了 53.3% 的 mAP。我们的方法结合了两个关键见解：(1) 可以将大容量卷积神经网络（CNN）应用于自下而上的区域建议，以定位和分割对象；(2) 当标记的训练数据稀缺时，针对辅助任务进行监督预训练，然后再进行特定领域的微调，可显著提高性能。由于我们将区域建议与 CNN 相结合，因此我们称这种方法为 R-CNN：具有 CNN 特征的区域。我们还将 R-CNN 与最近提出的基于类似 CNN 架构的滑动窗口检测器 OverFeat 进行了比较。我们发现，在 200 级 ILSVRC2013 检测数据集上，R-CNN 远远优于 OverFeat。完整系统的源代码可从 http://www.cs.berkeley.edu/˜rbg/rcnn 获取。

1. 导言

特征很重要。过去十年中，各种视觉识别任务的进展在很大程度上是基于 SIFT [29] 和 HOG [7]的使用。但如果我们看一下典型视觉识别任务--PASCAL VOC 物体检测[15]--的表现，人们普遍认为，2010-2012 年期间进展缓慢，通过建立集合系统和采用成功方法的小变体，只取得了微小的进步。

SIFT 和 HOG 是顺时针方向的直方图，我们可以将其与灵长类视觉通路第一皮层区域 V1 中的复杂细胞粗略地联系起来。但我们也知道，识别发生在下游的几个阶段，这表明可能存在分层、多阶段的特征计算过程，这些过程对视觉识别的信息量更大。

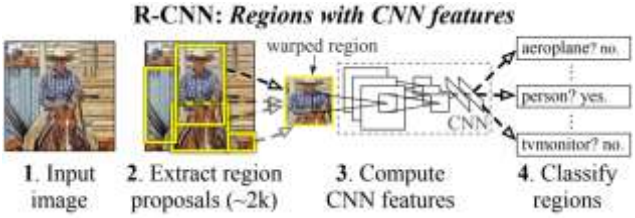

图 1：物体检测系统概览。我们的系统 (1) 接收输入图像，(2) 提取约 2000 个自下而上的区域建议，(3) 使用大型卷积神经网络 (CNN) 计算每个建议的特征，然后 (4) 使用特定类别的线性 SVM 对每个区域进行分类。R-CNN 在 PASCAL VOC 2010 上取得了 53.7% 的平均精度 (mAP)。相比之下，文献[39]使用相同的区域建议，但采用空间金字塔和视觉词袋方法，得出的 mAP 为 35.1%。流行的可变形部件模型的表现为 33.4%。在 ILSVRC2013 的 200 级检测数据集上，R-CNN 的 mAP 为 31.4%，比 OverFeat[34]有了很大提高，OverFeat 之前的最好成绩是 24.3%。

福岛的“新认知机”是一种受生物启发的分层和平移不变的模式识别模型，是早期尝试这一过程的一个例子。然而，新认知机缺乏监督训练算法。在 Rumelhart 等人的基础上，LeCun 等人表明，通过反向传播的随机梯度下降对于训练卷积神经网络（CNNs）是有效的，这类模型扩展了新认知机。

CNN 在 20 世纪 90 年代得到了广泛应用（例如 [27]），但随着支持向量机的兴起而逐渐过时。2012 年，Krizhevsky 等人[25] 在 ImageNet 大规模视觉识别挑战赛（ILSVRC）[9, 10]上展示了大幅提高的图像分类准确率，重新点燃了人们对 CNN 的兴趣。他们的成功源于在 120 万张标注图像上训练了一个大型

CNN，并对 LeCun 的 CNN 进行了一些改进（例如，max(x,0) 修正非线性和 "dropout "正则化）。

在 ILSVRC 2012 研讨会上，与会者对 ImageNet 结果的意义进行了激烈的讨论。核心问题可归纳为以下几点：CNN 在 ImageNet 上的分类结果能在多大程度上推广到 PASCAL VOC 挑战赛的物体检测结果？

我们通过缩小图像分类与物体检测之间的差距来回答这个问题。本文首次证明，与基于更简单的 HOG 类特征的系统相比，CNN 可以显著提高 PASCAL VOC 的物体检测性能。为了取得这一成果，我们重点解决了两个问题：使用深度网络定位物体，以及仅使用少量注释检测数据训练高容量模型。

与图像分类不同，检测需要定位图像中的物体（可能有很多）。有一种方法将定位作为一个回归问题。然而，与我们同时进行的 Szegedy 等人[38]的研究表明，这种策略在实践中可能效果不佳（他们报告的 VOC 2007 mAP 为 30.5%，而我们的方法达到了 58.5%）。另一种方法是建立滑动窗口检测器。以这种方式使用 CNN 已有至少二十年的历史，通常用于受限对象类别，如人脸 [32, 40] 和行人 [35]。为了保持高空间分辨率，这些 CNN 通常只有两个卷积层和池化层。我们还考虑过采用滑动窗口方法。然而，我们网络的高层单元有五个卷积层，在输入图像中具有非常大的感受野（195 × 195 像素）和步长（32 × 32 像素），这使得在滑动窗口范例中进行精确定位成为一项公开的技术挑战。

相反，我们通过 "使用区域识别 "范例 [21]来解决 CNN 定位问题，该范例已成功用于物体检测 [39] 和语义分割 [5]。在测试时，我们的方法会为输入图像生成约 2000 个与类别无关的区域建议，使用 CNN 从每个建议中提取固定长度的特征向量，然后使用特定类别的线性 SVM 对每个区域进行分类。我们使用一种简单的技术（仿射图像扭曲）从每个区域建议中计算出固定大小的 CNN 输入，而不管该区域的形状如何。图 1 概述了我们的方法，并重点介绍了我们的一些成果。由于我们的系统结合了区域建议和 CNN，因此我们将该方法命名为 R-CNN：具有 CNN 特征的区域。

在本文的更新版中，我们通过在 200 级 ILSVRC2013 检测数据集上运行 R-CNN，对 R-CNN 和最近提出的 OverFeat [34] 检测系统进行了正面比较。OverFeat 使用滑动窗口 CNN 进行检测，是迄今为止在 ILSVRC2013 检测中表现最好的方法。我们的研究表明，R-CNN 的性能明显优于 OverFeat，mAP 为 31.4% 对 24.3%。

检测面临的第二个挑战是标注数据稀缺，目前可用的数据量不足以训练大型 CNN。解决这一问题的传统方法是使用无监督预训练，然后进行有监督微调（如 [35]）。本文的第二个主要贡献是证明，在数据稀缺的情况下，在大型辅助数据集（ILSVRC）上进行监督预训练，然后在小型数据集（PASCAL）上进行特定领域的微调，是学习大容量 CNN 的有效范例。在我们的实验中，对检测进行微调可将 mAP 性能提高 8 个百分点。经过微调后，我们的系统在 2010 年 VOC 上实现了 54% 的 mAP，而经过高度微调、基于 HOG 的可变形部件模型 (DPM) [17, 20] 的 mAP 只有 33%。我们还向读者介绍了 Donahue 等人的同期研究成果[12]，他们的研究表明，Krizhevsky 的 CNN 可以用作黑盒特征提取器（无需微调），在场景分类、细粒度子分类和领域适应等多项识别任务中表现出色。

我们的系统也相当高效。唯一针对特定类别的计算是相当小的矩阵向量乘积和贪婪的非最大抑制。这一计算特性源于所有类别共享的特征，而且这些特征比以前使用的区域特征低两个数量级（参见 [39]）。

了解我们方法的失效模式对于改进它也至关重要，因此我们报告了 Hoiem 等人的检

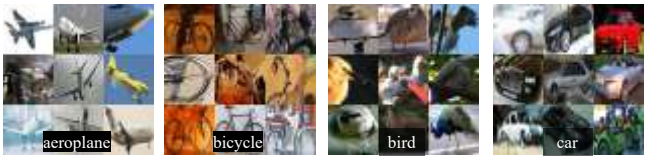

图 2：来自 2007 年 VOC 训练的扭曲训练样本。

测分析工具[23]的结果。作为分析的直接结果，我们证明了一种简单的边界框回归方法可以显著减少定位错误，而定位错误是主要的错误模式。

在介绍技术细节之前，我们要指出的是，由于 R-CNN 是在区域上运行的，因此自然可以将其扩展到语义分割任务中。稍作修改后，我们在 PASCAL VOC 分割任务中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%。

2. 利用 R-CNN 进行物体检测

我们的物体检测系统由三个模块组成。第一个模块生成与类别无关的区域建议。这些建议定义了可供我们的检测器使用的候选检测集合。第二个模块是一个大型卷积神经网络，从每个区域提取固定长度的特征向量。第三个模块是一组特定类别的线性 SVM。在本节中，我们将介绍每个模块的设计决策，描述它们在测试时的使用情况，详细说明如何学习它们的参数，并展示 PASCAL VOC 2010-12 和 ILSVRC2013 的检测结果。

2.1. 模块设计

区域提议。例如：对象性[1]、选择性搜索[39]、与类别无关的对象提议[14]、受限参数最小切分（CPMC）[5]、多尺度组合分组[3]，以及 Cires, an 等人[6]，他们通过将 CNN 应用于有规律间隔的方形作物来检测有丝分裂细胞，这是区域提议的一种特殊情况。虽然 R-CNN 与特定的区域提议方法无关，但我们使用了选择性搜索，以便与之前的检测工作（如 [39, 41]）进行有控制的比较。

特征提取我们使用 Krizhevsky 等人[25] 所描述的 CNN 的 Caffe [24] 实现从每个区域提案中提取 4096 维特征向量。特征的计算方法是通过五个卷积层和两个全连接层向前传播平均减缩后的 227×227 RGB 图像。更多网络架构详情，请读者参阅 [24, 25]。

为了计算一个区域提案的特征，我们必须首先将该区域的图像数据转换成与 CNN 兼容的形式（CNN 的架构要求输入固定的 227 × 227 像素大小）。在任意形状区域的多种可能转换中，我们选择了最简单的一种。无论候选区域的大小或长宽比如何，我们都会将其周围严格边界框中的所有像素翘曲到所需的大小。在翘曲之前，我们先扩张狭小的边界框，这样在翘曲后的大小上，原始边界框周围正好有 p 个像素的翘曲图像上下文（我们使用 p = 16）。图 2 显示了翘曲训练区域的随机样本。附录 A 讨论了翘曲的替代方法。

2.2. 测试时间检测

测试时，我们在测试图像上运行选择性搜索，以提取约 2000 个区域建议（我们在所有实验中都使用了选择性搜索的 "快速模式"）。我们对每个提议进行翘曲，并通过 CNN 进行前向传播，以计算特征。然后，对于每个类别，我们使用针对该类别训练的 SVM 对每个提取的特征向量进行评分。给定图像中的所有得分区域，我们应用贪婪的非最大抑制（针对每个类别独立应用），如果某个区域与得分较高的选定区域的交集-重叠（IoU）大于学习阈值，则剔除该区域。

运行时分析。有两个特性使检测变得高效。首先，所有类别的 CNN 参数都是共享的。其次，与其他常见方法相比，CNN 计算出的特征向量维度较低，例如具有视觉词袋编码的空间金字塔。例如，UVA 检测系统 [39] 中使用的特征比我们的大两个数量级（360k 对 4k）。

这种共享的结果是，计算区域提案和特征所花费的时间（在 GPU 上为 13 次/图像，在 CPU 上为 53 次/图像）被分摊到所有类别中。唯一针对类别的计算是特征与 SVM 权重之间的点乘和非最大抑制。在实际操作中，

图像的所有点积都会合并为一个矩阵-矩阵乘积。特征矩阵通常为 2000×4096，SVM 权重矩阵为 4096×N，其中 N 为类别数。

这项分析表明，R-CNN 可以扩展到数千个对象类别，而无需采用哈希等近似技术。即使有 10 万个类别，在现代多核 CPU 上进行矩阵乘法运算也只需 10 秒钟。这种效率不仅仅是使用区域提案和共享特征的结果。UVA 系统的高维特征需要 134GB 的内存来存储 100k 个线性预测器，而我们的低维特征只需要 1.5GB，因此速度要慢两个数量级。

将 R-CNN 与 Dean 等人最近在使用 DPM 和散列进行可扩展检测方面所做的工作[8]进行对比也很有趣。他们报告说，在 VOC 2007 上，当引入 10k 个分心类别时，每个图像的 mAP 约为 16%，运行时间为 5 分钟。而采用我们的方法，10k 个检测器在 CPU 上的运行时间约为 1 分钟，而且由于没有进行近似处理，mAP 将保持在 59%（第 3.2 节）。

2.3. 训练

监督预训练。我们在一个大型辅助数据集（ILSVRC2012 分类）上，仅使用图像级注释（该数据没有边框标签）对 CNN 进行了判别预训练。预训练使用开源的 Caffe CNN 库[24]。简而言之，我们的 CNN 几乎与 Krizhevsky 等人[25]的性能相当，在 ILSVRC2012 分类验证集上获得的最高-1 错误率高出 2.2 个百分点。这种差异是由于训练过程的简化造成的。

特定领域微调。为了让我们的 CNN 适应新任务（检测）和新领域（翘曲提议窗口），我们继续仅使用翘曲区域提议对 CNN 参数进行随机梯度下降（SGD）训练。除了用随机初始化的 (N + 1)-way 分类层（其中 N 为对象类别数，1 为背景）替换 CNN 的 ImageNets 特定 1000-way 分类层外，CNN 架构保持不变。对于 VOC，N = 20，对于 ILSVRC2013，N = 200。我们将所有与地面实况方框重叠度≥ 0.5 IoU 的区域提案视为该方框类别的阳性提案，其余的视为阴性提案。我们以 0.001 的学习率（初始预训练率的 1/10）开始 SGD，这样既可以进行微调以取得进展，又不会破坏初始化。在 SGD 的每次迭代中，我们对 32 个正向窗口（覆盖所有类别）和 96 个背景窗口进行均匀采样，以构建大小为 128 的迷你批次。我们偏向于正向窗口采样，因为与背景窗口相比，正向窗口极为罕见。

物体类别分类器。考虑训练一个二元分类器来检测汽车。很明显，紧紧包围汽车的图像区域应该是一个正例。同样，与汽车无关的背景区域显然应该是负面示例。但如何标注与汽车部分重叠的区域就不那么清楚了。我们通过一个 IoU 重叠阈值来解决这个问题，低于该阈值的区域将被定义为负例。重叠阈值 0.3 是通过在验证集上对{0,0.1,...,0.5}进行网格搜索而选出的。我们发现，仔细选择这个阈值非常重要。如文献[39]所述，将阈值设为 0.5 会使 mAP 下降 5 个点。同样，将阈值设为 0 会使 mAP 下降 4 个百分点。正面例子被简单定义为每个类别的地面实况边界框。

提取特征并应用训练标签后，我们对每个类别优化一个线性 SVM。由于训练数据太大，内存无法容纳，我们采用了标准的硬负挖掘方法 [17，37]。硬负挖掘收敛速度很快，实际上只需对所有图像进行一次扫描，mAP 就会停止增加。

在附录 B 中，我们将讨论为什么在微调与 SVM 训练中对正例和负例的定义不同。我们还讨论了在训练检测 SVM 而非简单使用微调 CNN 最后 softmax 层的输出时所涉及的权衡问题。PASCAL VOC 2010-12 测试结果

按照 PASCAL VOC 最佳实践[15]，我们在 VOC 2007 数据集上验证了所有设计决策和超参数（第 3.2 节）。为了获得 VOC 2010-12 数据集的最终结果，我们在 VOC 2012 train 上对 CNN 进行了微调，并在 VOC 2012 trainval 上优化了检测 SVM。对于两种主要算法变体（有边框回归和无边框回归），我们只向评估服务器提交了一次测试结果。

表 1 显示了 VOC 2010 的完整结果。我们将我们的方法与四种强大的基准进行了比较，其中包括 SegDPM [18]，它将 DPM 检测器与语义分割系统 [4] 的输出相结合，并使用了额外的检测器间上下文和图像分类器重新评分。与 Uijlings 等人的 UVA 系统[39]进行比较最有意义，因为我们的系统使用相同的区域建议算法。为了对区域进行分类，他们的方法建立了一个四层空间金字塔，并用密集采样的 SIFT、Extended OpponentSIFT 和 RGBSIFT 描述符填充，每个向量都用 4000 字的编码本量化。使用直方图交集核 SVM 进行分类。与他们的多特征非线性核 SVM 方法相比，我们的 mAP 有了很大提高，从 35.1% 提高到 53.7%，同时速度也快得多（第 2.2 节）。我们的方法在 VOC 2011/12 测试中取得了类似的性能（53.3% mAP）。

2.5. Results on ILSVRC2013 detection

我们使用与 PASCAL VOC 相同的系统超参数，在 ILSVRC2013 200 级检测数据集上运行了 R-CNN。我们遵循相同的协议，只向 ILSVRC2013 评估服务器提交了两次测试结果，一次有边界框回归，一次没有。

图 3 比较了 R-CNN 与 ILSVRC 2013 竞赛中的参赛作品以及赛后 OverFeat 的结果 [34]。R-CNN 的 mAP 高达 31.4%，大幅领先于 OverFeat 第二名的 24.3%。为了解 AP 在不同类别中的分布情况，本文还给出了箱形图，并在表 8 中列出了每个类别的 AP 表。大多数参赛者（OverFeat、NEC-MU、UvAEuvision、Toronto A 和 UIUC-IFP）都使用了卷积神经网络，这表明在如何将 CNN 应用于物体检测方面存在很大的细微差别，导致结果大相径庭。

在第 4 节中，我们将概述 ILSVRC2013 检测数据集，并详细介绍我们在该数据集上运行 R-CNN 时所作的选择。

3. 可视化、烧蚀和误差模式

3.1. 学习特征可视化

第一层滤波器可以直接可视化，而且易于理解 [25]。它们能捕捉到有方向的边缘和对立的颜色。理解后续层则更具挑战性。Zeiler 和 Fergus 在[42]中提出了一种具有视觉吸引力的去卷积方法。我们提出了一种简单（且互补）的非参数方法，可直接显示网络学习到了什么。

我们的想法是将网络中的一个特定单元（特征）单列出来，将其作为一个物体检测器来使用。也就是说，我们计算该单元在一大组保留区域提案（约 1 千万个）上的激活度，将提案从激活度最高到最低排序，执行非最大抑制，然后显示得分最高的区域。我们的方法通过准确显示所选单元对哪些输入起作用，让所选单元 "为自己代言"。我们避免了平均化，以便看到不同的视觉模式，并深入了解该单元计算的不变量。

我们将第 5 层池中的单元可视化，它是网络第五层也是最后一层卷积层的最大池输出。池 5 的特征图为 $6 \times 6 \times 256 = 9216$ 维。忽略边界效应，在 227×227 像素的原始输入中，每个池 5 单元的感受野为 195×195 像素。池 5 中心单元的视角几乎是全局的，而靠近边缘的单元则有一个较小的剪切支持。

图 4 中的每一行都显示了我们在 VOC 2007 trainval 上微调的 CNN 的 pool5 单元的前 16 个激活。图中显示了 256 个功能独特的单元中的 6 个单元（附录 D 中包含更多单元）。选择这些单元是为了展示网络学习的代表性样本。在第二行中，我们看到的是一个对狗脸和点阵列起作用的单元。第三行对应的单元是一个红色圆球检测器。此外，还有人脸和更抽象图案的检测器，如文本和带窗口的三角形结构。该网络似乎在学习一种表征方式，它将少量经过类别调整的特征与形状、纹理、颜色和材料属性的分布式表征方式结

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

表 1：VOC 2010 测试的平均检测精度（%）。R-CNN 与 UVA 和 Regionlets 具有最直接的可比性，因为所有方法都使用选择性搜索区域建议。SegDPM 是 PASCAL VOC 排行榜上的佼佼者。†DPM 和 SegDPM 使用了其他方法没有使用的上下文重评分。

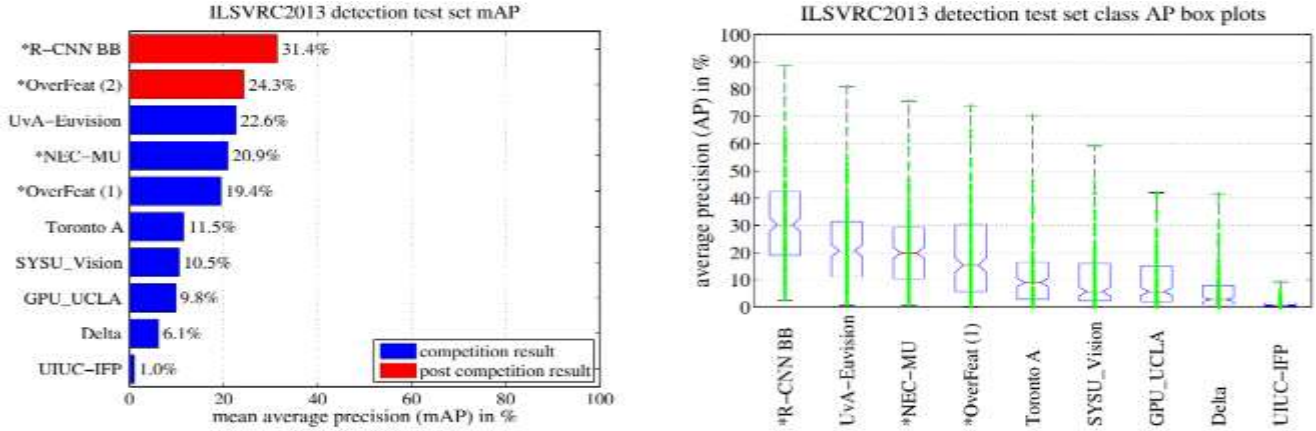

图 3：（左）ILSVRC2013 检测测试集的平均精度。带 * 的方法使用外部训练数据（所有情况下，图像和标签均来自 ILSVRC 分类数据集）。(右图）每种方法的 200 个平均精度值的方框图。由于尚未提供每类 AP，因此未显示赛后 OverFeat 结果的方框图（R-CNN 的每类 AP 见表 8，也包含在上传到 arXiv.org 的技术报告源中；见 R-CNN-ILSVRC2013-APs.txt）。红线表示 AP 中位数，方框底部和顶部表示第 25 和 75 百分位数。晶须延伸至每种方法的最小和最大 AP。每个 AP 在晶须上绘制为一个绿点（最好放大数字查看）。

图 4：六个 pool5 单元的顶部区域。感知区域和激活值以白色绘制。有些单元与概念对齐，如人物（第 1 行）或文本（第 4 行）。其他单元则捕捉纹理和材料属性，如点阵列（2）和镜面反射（6）。

合在一起。随后的全连接层 fc6 能够为这些丰
富特征的大量组合建模。

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN $pool_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc_6	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc_7	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT $pool_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc_6	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc_7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc_7 BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

表 2：2007 年 VOC 测试的平均检测精度（%）。第 1-3 行显示的是未经微调的 R-CNN 性能。第 4-6 行显示的是 CNN 在 ILSVRC 2012 上进行预训练，然后在 VOC 2007 trainval 上进行微调 (FT) 的结果。第 7 行包括一个简单的边界框回归 (BB) 阶段，可减少定位误差（C 节）。第 8-10 行将 DPM 方法作为强基线。第一行只使用 HOG，而后两行则使用不同的特征学习方法来增强或替代 HOG。

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	**35.7**	62.1	64.0	66.5	**71.2**	62.2
R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

表 3：两种不同 CNN 架构在 VOC 2007 测试中的平均检测精度（%）。前两行是表 2 中使用 Krizhevsky 等人的架构（T-Net）得出的结果。第三行和第四行使用的是 Simonyan 和 Zisserman 最近提出的 16 层架构（O-Net）[43]。

3.2. 消融研究

逐层性能，无微调。为了了解哪些层对检测性能至关重要，我们分析了 VOC 2007 数据集上 CNN 最后三层中每一层的检测结果。第 3.1 节简要介绍了第 5 层。最后两层总结如下。

层 fc6 与池 5 完全连接。为计算特征，它将 4096×9216 权重矩阵与池 5 特征图（重塑为 9216 维向量）相乘，然后添加一个偏置向量。这个中间向量经过分量半波整流（x ← max(0,x)）。

fc7 层是网络的最后一层。其实现方法是将 fc6 计算出的特征乘以 4096 × 4096 权重矩阵，然后加入一个偏置向量并进行半波整流。

我们首先查看未在 PASCAL 上进行微调的 CNN 的结果，即所有 CNN 参数仅在 ILSVRC 2012 上进行了预训练。逐层分析性能（表 2 第 1-3 行）发现，fc7 的特征泛化效果比 fc6 差。这意味着，在不降低 mAP 的情况下，可以移除 29% 的 CNN 参数，即大约 1,680 万个参数。更令人惊讶的是，尽管池 5 特征的计算只使用了 CNN 6% 的参数，但同时移除 fc7 和 fc6 会产生相当好的结果。CNN 的大部分表征能力来自卷积层，而不是更大的密集连接层。这一发现表明，只使用 CNN 的卷积层计算任意大小图像的 HOG 意义上的密集特征图具有潜在的实用性。通过这种表示方法，可以在池 5 特征的基础上尝试使用滑动窗口检测器（包括 DPM）。.

微调后的逐层性能现在我们来看看在 VOC 2007 trainval 上微调 CNN 参数后的结果。改进非常明显（表 2 第 4-6 行）：微调后 mAP 增加了 8.0 个百分点，达到 54.2%。对 fc6 和 fc7 进行微调后，mAP 的提升幅度远大于对 pool5 的提升幅度，这表明从 ImageNet 中学习到的 pool5 特征是通用的，大部分改进都是通过在其基础上学习特定领域的非线性分类器获得的。

与最新特征学习方法的比较。在 PASCAL VOC 检测中尝试过的特征学习方法相对较少。我们研究了基于可变形部件模型的两种最新方法。作为参考，我们还包括基于 HOG 的标准 DPM [20] 的结果。

第一种 DPM 特征学习方法 DPM ST [28]，利用 "草图标记 "概率直方图增强 HOG 特征。从直观上讲，草图标记是通过图像补丁中心的轮廓线的紧密分布。草图标记概率由随机森林在每个像素上计算得出，该随机森林经过训练可将 35×35 像素的斑块分为 150 个草图标记或背景之一。

第二种方法是 DPM HSC [31]，用稀疏代码直方图（HSC）取代 HOG。为计算 HSC，使用由 100 个 7 × 7 像素（灰度）原子组成的学习字典求解每个像素的稀疏代码激活。由此产生的激活以三种方式进行整流（全波和半波）、空间汇集、单位 `2 归一化，然后进行幂变换($x \leftarrow \text{sign}(x)|x|^{\alpha}$).

所有 R-CNN 变体都大大优于三个 DPM 基线（表 2 第 8-10 行），包括使用特征学习的两个变体。与只使用 HOG 特征的最新版 DPM 相比，我们的 mAP 高出 20 多个百分点：54.2% 对 33.7%，相对提高 61%。HOG 和草图标记的组合比单独使用 HOG 提高了 2.5 个 mAP 点，而 HSC 比 HOG 提高了 4 个 mAP 点（在内部与它们的私有 DPM 基线进行比较时--两者都使用了 DPM 的非公开实现，但其性能低于开源版本 [20]）。这些方法的 mAP 分别为 29.1% 和 34.3%。

3.3. 网络架构

本文中的大多数结果都使用了 Krizhevsky 等人[25]的网络架构。然而，我们发现架构的选择对 R-CNN 的检测性能有很大影响。在表 3 中，我们展示了使用 Simonyan 和 Zisserman [43] 最近提出的 16 层深度网络进行 VOC 2007 测试的结果。该网络是最近 ILSVRC 2014 分类挑战赛中表现最好的网络之一。该网络具有同构结构，由 13 层 3 × 3 卷积核组成，中间穿插了 5 层最大池化层，顶部有 3 层全连接层。我们将该网络称为 "O-Net"（牛津网络），基线网络称为 "T-Net"（多伦多网络）。

为了在 R-CNN 中使用 O-Net，我们从 Caffe Model Zoo 下载了 VGGILSVRC16layers 模型的公开预训练网络权重。然后，我们使用与 T-Net 相同的协议对网络进行了微调。唯一的区别是根据需要使用较小的迷你批次（24 个示例），以适应 GPU 内存。表 3 中的结果显示，使用 O-Net 的 RCNN 明显优于使用 TNet 的 R-CNN，mAP 从 58.5% 提高到 66.0%。不过，在计算时间方面存在相当大的缺陷，O-Net 的前向传递时间大约是 T-Net 的 7 倍。

3.4. 检测误差分析

我们应用了 Hoiem 等人[23]的优秀检测分析工具，以揭示我们方法的误差模式，了解微调如何改变误差模式，以及我们的误差类型与 DPM 的比较。对分析工具的全面总结超出了本文的范围，我们鼓励读者参阅 [23]，以了解一些更精细的细节（如 "归一化 AP"）。由于分析最好结合相关图表进行，因此我们在图 5 和图 6 的标题中进行了讨论。

3.5. 边界框回归

基于误差分析，我们采用了一种简单的方法来减少定位误差。受 DPM [17] 中采用的边界框回归法的启发，我们训练了一个线性回归模型，以预测一个新的检测窗口，并给出选择性搜索区域建议的池 5 特征。详情见附录 C。表 1、表 2 和图 5 中的结果表明，这种简单的方法修复了大量定位错误的检测，将 mAP 提升了 3 到 4 个点。

3.6. 定性结果

ILSVRC2013 的定性检测结果见本文末尾的图 8 和图 9。每幅图像都是从 val2 图像集中随机抽取的，图中显示了所有检测器检测到的精度大于 0.5 的图像。需要注意的是，这些数据并不是经过精心策划的，而是检测器工作时的真实情况。图 10 和图 11 显示了更多定性结果，但这些结果都经过了整理。我们之所以选择每张图片，是因为它包含了

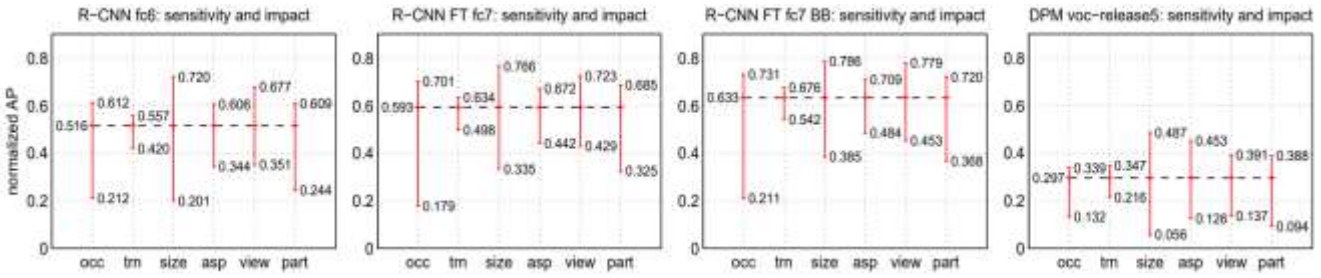

图 6：对物体特征的敏感性。每幅图都显示了在六种不同的对象特征（遮挡、截断、边界框面积、长宽比、视角、部分可见度）下，表现最好和表现最差的子集的平均（跨类）归一化 AP（见 [23]）。我们展示了我们的方法（R-CNN）在进行微调（FT）和边界框回归（BB）以及 DPM voc-release5 的情况下的曲线图。总体而言，微调并没有降低灵敏度（最大值与最小值之间的差异），但却大大改善了几乎所有特征的最高和最低性能子集。这表明，微调不只是简单地改善了长宽比和边界框面积方面性能最低的子集，这可能是基于我们如何扭曲网络输入的猜想。相反，微调提高了所有特征的鲁棒性，包括遮挡、截断、视角和部件可见性。

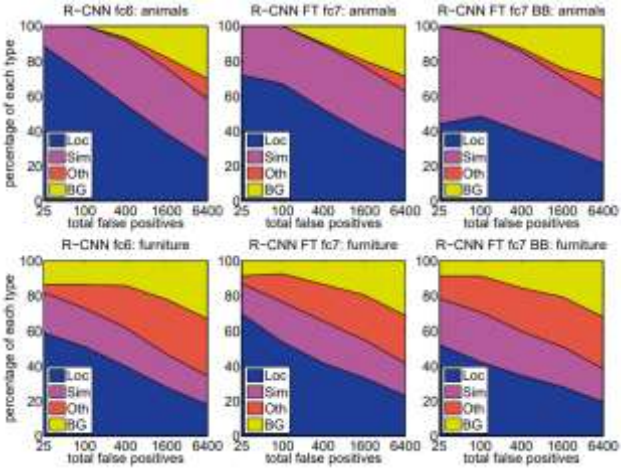

图 5：排名靠前的假阳性 (FP) 类型分布。每幅图都显示了随着考虑的 FP 越多，FP 类型的分布也在不断变化，得分依次递减。每个 FP 可分为 4 种类型中的 1 种：Loc-定位能力差（与正确类别的 IoU 重叠度在 0.1 和 0.5 之间的检测，或重复检测）；Sim-与相似类别相混淆；Oth-与不相似的物体类别相混淆；BG-在背景上发射的 FP。与 DPM（见 [23]）相比，我们的错误更多是由于定位不准确造成的，而不是与背景或其他物体类别混淆，这表明 CNN 特征比 HOG 更具有区分性。定位松散的原因可能是我们使用了自下而上的区域建议，以及通过对 CNN 进行全图像分类预训练而获得的位置不变性。第三列显示了我们的简单边界框回归方法如何修正了许多定位错误。

有趣、令人惊讶或令人捧腹的结果。这里还显示了精度大于 0.5 的所有检测结果。

4. ILSVRC2013 检测数据集

在第 2 节中，我们介绍了 ILSVRC2013 检测数据集的结果。与 PASCAL VOC 相比，该数据集的同质性较低，因此需要选择如何使用该数据集。由于这些决定并不简单，我们将在本节中介绍。

4.1. 数据集概览

ILSVRC2013 检测数据集分为三组：训练集（395,918 张）、验证集（20,121 张）和测试集（40,152 张），括号内为每组中的图像数量。val 和 test 两组图像来自相同的图像分布。这些图像与 PASCAL VOC 图像具有相似的场景和复杂性（物体数量、杂乱程度、姿势变化等）。val 和 test 分割图像都有详尽的注释，这意味着在每幅图像中，所有 200 个类别的所有实例都标有边框。训练集则取自 ILSVRC2013 分类图像分布。这些图像的复杂度更加多变，偏向于单个居中物体的图像。与 val 和 test 不同的是，训练图像（由于数量众多）并没有进行详尽的注释。在任何给定的训练图像中，200 个类别中的实例可能被标注，也可能未被标注。除了这些图像集，每个类别还有一组额外的负图像。阴性图像会经过人工检查，以确认其中不包含任何相关类别的实例。本研究未使用负图像集。有关如何收集和注释 ILSVRC 的更多信息，请参阅 [11, 36]。

这些分割的性质为训练 R-CNN 提供了多种选择。训练图像不能用于硬性负面挖掘，因为注释并不详尽。负面示例从何而来？此外，训练图像的统计信息与 val 和 test 不同。是否应该使用训练图像？虽然我们还没有对大量选择进行彻底评估，但根据以往的经验，我们提出了看起来最明显的路径。

我们的一般策略是主要依靠 val 集，并使用部分训练图像作为正面示例的辅助来源。为了将 val 用于训练和验证，我们将其分为大小大致相同的 "val1 "和 "val2 "集。由于一些类别在 val 中的例子很少（最小的只有 31 个，一半的少于 110 个），因此产生一个近似类别平衡的分区非常重要。为此，我们生成了大量候选分割集，并选择了相对类不平衡最大值最小的一个。 每个候选分区都是通过使用 val 图像的类计数作为特征对其进行聚类，然后进行随机局部搜索，从而提高分区的平衡性。这里使用的特定分割的最大相对不平衡度约为 11%，中位相对不平衡度为 4%。val1/val2 分割和用于生成它们的代码将被公开，以便其他研究人员将他们的方法与本报告中使用的 val 分割进行比较。

4.2. 区域提议

我们采用了与 PASCAL 检测相同的区域建议方法。在 val1、val2 和 test 中的每幅图像（但不包括 train 中的图像）上以 "快速模式 "运行选择性搜索 [39]。由于选择性搜索不具有比例不变性，因此产生的区域数量取决于图像的分辨率，因此需要对其稍作修改。ILSVRC 图像的尺寸从非常小到几百万像素不等，因此我们在运行选择性搜索之前将每幅图像调整为固定宽度（500 像素）。选择性搜索的结果是，每幅图像平均有 2403 个区域提案，所有地面实况边界框的召回率为 91.6%（阈值为 0.5 IoU）。这一召回率明显低于 PASCAL，PASCAL 的召回率约为 98%，这表明在区域建议阶段还有很大的改进空间。

4.3. 训练数据

在训练数据方面，我们建立了一个图像和方框集，其中包括 val1 中的所有选择性搜索和地面实况方框，以及 train 中每个类别的最多 N 个地面实况方框（如果一个类别在 train 中的地面实况方框少于 N 个，那么我们就取全部）。我们称这个图像和方框数据集为 val1+trainN。在一项消融研究中，我们展示了 $N \in \{0,500,1000\}$ 时 val2 的 mAP（第 4.5 节）。

R-CNN 的三个过程都需要训练数据：(1) CNN 微调，(2) 检测器 SVM 训练，以及 (3) 边框回归器训练。使用与 PASCAL 完全相同的设置，在 val1+trainN 上对 CNN 进行了 50k 次 SGD 迭代微调。使用 Caffe 在一台 NVIDIA Tesla K20 上进行微调耗时 13 小时。在 SVM 训练中，val1+trainN 中的所有地面实况盒都被用作各自类别的正例。从 val1 中随机选取的 5000 张图片子集进行了硬负片挖掘。最初的实验表明，从 val1 的全部图像中挖掘底片与从 5000 张图像子集（约占一半）中挖掘底片相比，mAP 只下降了 0.5 个百分点，而 SVM 的训练时间却减少了一半。由于注释并不详尽，因此没有从 train 中提取负面示例。没有使用额外的已验证负图像集。边界框回归器在 val1 上进行了训练。

4.4. 验证和评估

在向评估服务器提交结果之前，我们使用上述训练数据验证了数据使用选择以及在 val2 集上进行微调和边界框回归的效果。所有系统超参数（如 SVM C 超参数、区域扭曲中使用的填充、NMS 阈值、边界框回归超参数）都固定为 PASCAL 使用的相同值。毫无疑问，对于 ILSVRC 而言，其中一些超参数选择略微次优，但这项工作的目标是在不对数据集进行大量调整的情况下，在 ILSVRC 上得出初步的 R-CNN 结果。在 Val2 上选出最佳选择后，我们向 ILSVRC2013 评估服务器提交了两个结果文件。第一次提交的文件没有边界框回归，第二次提交的文件有边界框回归。在提交这些文件时，我们扩展了 SVM 和边界框回归训练集，分别使用了 val+train1k 和 val。我们使用了在 val1+train1k 上进行微调的 CNN，以避免重新运行微调和特征计算。

test set	val_2	val_2	val_2	val_2	val_2	val_2	test	test
SVM training set	val_1	val_1+$train_{.5k}$	val_1+$train_{1k}$	val_1+$train_{1k}$	val_1+$train_{1k}$	val_1+$train_{1k}$	val+$train_{1k}$	val+$train_{1k}$
CNN fine-tuning set	n/a	n/a	n/a	val_1	val_1+$train_{1k}$	val_1+$train_{1k}$	val_1+$train_{1k}$	val_1+$train_{1k}$
bbox reg set	n/a	n/a	n/a	n/a	n/a	val_1	n/a	val
CNN feature layer	fc_6	fc_6	fc_6	fc_7	fc_7	fc_7	fc_7	fc_7
mAP	20.9	24.1	24.1	26.5	29.7	**31.0**	30.2	**31.4**
median AP	17.7	21.0	21.4	24.8	29.2	**29.6**	29.0	**30.3**

表 4：ILSVRC2013 数据使用选择、微调和边界框回归的消融研究。

4.5. 消融研究

表 4 显示了对不同训练数据量、微调和边界框回归效果的消融研究。第一个观察结果是，val2 的 mAP 与 test 的 mAP 非常接近。这让我们相信，val2 的 mAP 是测试集性能的良好指标。第一个结果（20.9%）是 R-CNN 在 ILSVRC2012 分类数据集上使用预先训练好的 CNN（未进行微调），并访问 val1 中的少量训练数据（请注意，val1 中一半类别的示例数在 15 到 55 之间）所取得的结果。将训练集扩大到 val1+trainN 后，性能提高到 24.1%，N = 500 和 N = 1000 之间基本没有差别。仅使用 val1 中的示例对 CNN 进行微调后，性能略有提高，达到 26.5%，但由于正向训练示例数量较少，可能存在明显的过拟合。将微调集扩大到 val1+train1k，即从训练集中为每个类别添加多达 1000 个正面示例，会有很大帮助，使 mAP 提高到 29.7%。边框回归将结果提高到 31.0%，与 PASCAL 中观察到的结果相比，相对增益较小。

4.6. 与 OverFeat 的关系

R-CNN 和 OverFeat 之间有一种有趣的关系：OverFeat（大致）可以看作是 R-CNN 的一个特例。如果将选择性搜索区域建议替换为规则方形区域的多尺度金字塔，并将每类边界框回归器替换为单个边界框回归器，那么这两个系统将非常相似（只是在训练方式上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，与 R-CNN 相比，OverFeat 在速度上有显著优势：根据 [34] 中引用的每张图像 2 秒的数据，OverFeat 的速度约为 R-CNN 的 9 倍。之所以能达到这样的速度，是因为 OverFeat 的滑动窗口（即区域建议）没有在图像层面进行扭曲，因此重叠窗口之间可以轻松共享计算。共享是通过在任意大小的输入上以卷积方式运行整个网络来实现的。加快 R-CNN 的速度可以通过多种方式实现，这也是未来的工作方向。

5. 语义分割

区域分类是语义分割的标准技术，因此我们可以轻松地将 R-CNN 应用于 PASCAL VOC 分割挑战。为了便于与当前领先的语义分割系统（称为 O2P，意为 "二阶池化"）[4] 进行直接比较，我们在其开源框架内工作。O2P 使用 CPMC 为每幅图像生成 150 个区域建议，然后使用支持向量回归 (SVR) 预测每个区域、每个类别的质量。他们的方法之所以性能卓越，是因为 CPMC 区域的质量以及多种特征类型（SIFT 和 LBP 的丰富变体）的强大二阶池化功能。我们还注意到，Farabet 等人[16]最近使用 CNN 作为多尺度每像素分类器，在几个密集场景标注数据集（不包括 PASCAL）上取得了良好的效果。

我们遵循文献[2, 4]，扩展了 PASCAL 分割训练集，以包括 Hariharan 等人[22]提供的额外注释。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一次。

用于分割的 CNN 特征。我们对计算 CPMC 区域特征的三种策略进行了评估，所有策略都是从将区域周围的矩形窗口调整为 227 × 227 开始。第一种策略（完全）忽略了区域的形状，直接在翘曲窗口上计算 CNN 特征，这与我们的检测方法完全相同。但是，这些特征忽略了区域的非矩形形状。两个区

域的边界框可能非常相似，但重叠的部分却很少。因此，第二种策略 (fg) 只在区域的前景掩膜上计算 CNN 特征。我们用平均输入替换背景，这样背景区域在平均值减去后为零。第三种策略（full+fg）简单地将 full 和 fg 特征合并；我们的实验验证了它们的互补性。

	full R-CNN		*fg* R-CNN		*full+fg* R-CNN	
O_2P [4]	fc_6	fc_7	fc_6	fc_7	fc_6	fc_7
46.4	43.0	42.5	43.7	42.1	**47.9**	45.8

表 5：VOC 2011 验证的平均分割准确率（%）。第 1 列显示的是 O2P；第 2-7 列使用的是我们在 2012 年 ILSVRC 上预先训练的 CNN。

VOC 2011 的结果。表 5 显示了我们在 2011 年 VOC 验证集上与 O2P 的对比结果汇总。(在每种特征计算策略中，fc6 层始终优于 fc7 层，以下讨论将针对 fc6 特征。fg 策略略微优于 full，这表明屏蔽区域的形状提供了更强的信号，符合我们的直觉。然而，full+fg 实现了 47.9% 的平均准确率，以 4.2% 的优势成为我们的最佳结果（也略微优于 O2P），这表明即使考虑到 fg 特征，full 特征提供的上下文信息也非常丰富。值得注意的是，在我们的完整+fg 特征上训练 20 个 SVR 仅需一个小时，而在 O2P 特征上训练则需要 10 多个小时。

表 6 列出了 VOC 2011 测试集的结果，将我们表现最好的方法 fc6（full+fg）与两个强大的基线进行了比较。在 21 个类别中，我们的方法在 11 个类别中获得了最高的分割准确率，在所有类别中平均获得了 47.9% 的最高总体分割准确率（但在任何合理的误差范围内都可能与 O2P 的结果持平）。通过微调，可能会取得更好的性能。

6. 结论

近年来，物体检测性能停滞不前。性能最好的系统是将多个低级图像特征与来自物体检测器和场景分类器的高级上下文相结合的复杂集合。本文提出了一种简单、可扩展的物体检测算法，与 PASCAL VOC 2012 上的最佳结果相比，相对提高了 30%。

我们通过两种方法实现了这一性能。首先是将大容量卷积神经网络应用于自下而上的区域建议，以定位和分割对象。其次是在标注训练数据稀缺的情况下训练大型 CNN 的范例。我们的研究表明，针对数据丰富的辅助任务（图像分类），在有监督的情况下预先训练网络，然后针对数据稀缺的目标任务（检测）对网络进行微调是非常有效的。我们推测，"有监督的预训练/特定领域的微调 " 模式将对各种数据稀缺的视觉问题非常有效。

最后，我们要指出，重要的是，我们结合使用了计算机视觉和深度学习的经典工具（自下而上的区域建议和卷积神经网络），从而取得了这些成果。这两者不是科学探索的对立面，而是天然的、不可避免的合作伙伴。

致谢。本研究部分得到了 DARPA Mind's Eye 和 MSEE 计划、NSF IIS-0905647、IIS-1134072 和 IIS-1212798 奖项、MURI N000014-10-1-0933 以及丰田公司的支持。本研究中使用的 GPU 由英伟达公司慷慨捐赠。

附录

A. 目标提议变换

这项工作中使用的卷积神经网络需要 227 × 227 像素的固定大小输入。在检测时，我们考虑的对象建议是任意图像矩形。我们评估了两种将物体提案转化为有效卷积神经网络输入的方法。

第一种方法（"带上下文的最密正方形"）将每个对象提案包围在最密正方形内，然后将该正方形所包含的图像按 CNN 输入大小进行（等比例）缩放。图 7 (B) 列显示了这种变换。这种方法的一个变体（"无上下文的最密正方形"）排除了原始对象建议周围的图像内容。图 7（C）列显示了这种变换。第二种方法（"翘曲"）是将每个对象建议按 CNN 输入大小进行各向异性缩放。图 7 (D) 列显示了翘曲变换。

对于每一种转换，我们都会考虑在原始对象提案周围添加额外的图像上下文。上下文填充量（p）被定义为在变换后的输入坐标

test set	val_2	val_2	val_2	val_2	val_2	val_2	test	test
SVM training set	val_1	$val_1+train_{.5k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val+train_{1k}$	$val+train_{1k}$
CNN fine-tuning set	n/a	n/a	n/a	val_1	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$
bbox reg set	n/a	n/a	n/a	n/a	n/a	val_1	n/a	val
CNN feature layer	fc_6	fc_6	fc_6	fc_7	fc_7	fc_7	fc_7	fc_7
mAP	20.9	24.1	24.1	26.5	29.7	**31.0**	30.2	**31.4**
median AP	17.7	21.0	21.4	24.8	29.2	**29.6**	29.0	**30.3**

表 4：ILSVRC2013 数据使用选择、微调和边界框回归的消融研究。

VOC 2011 test	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
R&P [2]	83.4	46.8	18.9	36.6	31.2	42.7	57.3	47.4	44.1	8.1	39.4	**36.1**	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O_2P [4]	**85.4**	**69.7**	22.3	45.2	**44.4**	46.9	66.7	57.8	56.2	**13.5**	**46.1**	32.3	41.2	**59.1**	55.3	51.0	**36.2**	50.4	**27.8**	46.9	**44.6**	47.6
ours (*full+fg* R-CNN fc_6)	84.2	66.9	**23.7**	**58.3**	37.4	**55.4**	**73.3**	**58.7**	**56.5**	9.7	45.5	29.5	**49.3**	40.1	**57.8**	**53.9**	33.8	**60.7**	22.7	**47.1**	41.3	**47.9**

表 6：2011 年 VOC 测试的分割准确率（%）。我们与两个强大的基准进行了比较：[2] 的 "区域和部分"（R&P）方法和[4] 的二阶池化（O2P）方法。在不做任何微调的情况下，我们的 CNN 实现了最高的分割性能，优于 R&P，与 O2P 大致相当。

框架中原始对象提案周围的边框大小。图 7 显示，每个示例中最上面一行的 p = 0 像素，最下面一行的 p = 16 像素。在所有方法中，如果源矩形超出了图像的范围，缺失的数据将用图像平均值代替（然后在将图像输入 CNN 之前减去图像平均值）。一组先导实验表明，使用上下文填充（p = 16 像素）的翘曲效果远远优于其他方法（3-5 mAP 点）。显然，还可以有更多的替代方案，包括使用复制代替平均填充。对这些替代方案的详尽评估将作为未来工作的一部分。

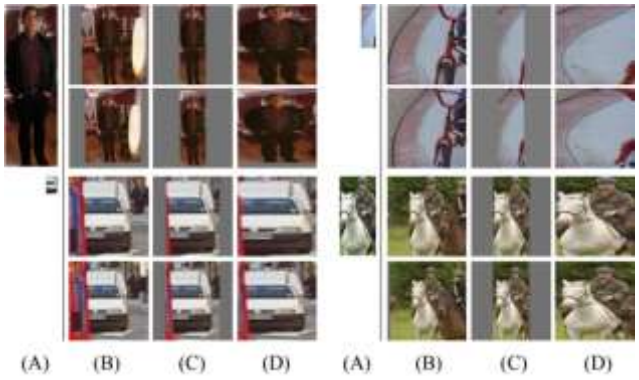

图 7：不同的对象提议转换。(A) 相对于转换后的 CNN 输入，原始对象建议的实际比例；(B) 有上下文的最密正方形；(C) 无上下文的最密正方形；(D) 翘曲。在每一列和示例方案中，最上面一行对应 p = 0 像素的上下文填充，而最下面一行对应 p = 16 像素的上下文填充。

B. 正例与反例和软最大值

有两个设计方案值得进一步讨论。第一个问题是为什么在微调 CNN 和训练对象检测 SVM 时，正例和负例的定义不同？简单回顾一下定义，在微调时，我们将每个对象建议映射到与之有最大 IoU 重叠（如果有）的地面实况实例，如果 IoU 至少为 0.5，则将其标记为匹配地面实况类别的正例。所有其他建议都被标记为 "背景"（即所有类别的负例）。相反，在训练 SVM 时，我们只将地面实况框作为各自类别的正例，而将与某一类别所有实例的 IoU 重合度小于 0.3 的建议标记为该类别的负例。属于灰色区域（重合度超过 0.3 IoU，但不是地面实况）的建议将被忽略。

从历史上看，我们之所以得出这些定义，是因为我们一开始是根据 ImageNet 预训练 CNN 计算出的特征来训练 SVM 的，因此当时并没有考虑微调。在这一设置中，我们发现用于训练 SVM 的特定标签定义在我们评估的选项集（包括我们现在用于微调的设置）中是最优的。当我们开始使用微调时，我们最初使用了与 SVM 训练相同的正负示例定义。然而，我们发现结果比我们目前使用的正例和负例定义差很多。

我们的假设是，正负定义方式的这种差异并不是根本性的，而是由于微调数据有限这一事实造成的。我们目前的方案引入了许多 "抖动 "示例（那些重叠度在 0.5 和 1 之间的建议，但不是地面实况），这将正面示例的数量扩大了约 30 倍。我们推测，在对整个网络进行微调以避免过度拟合时，需要使用这

一大型集合。不过，我们也注意到，使用这些抖动示例很可能是次优的，因为网络微调并不是为了精确定位。

这就引出了第二个问题：微调之后，为什么还要训练 SVM？如果简单地将微调后网络的最后一层（即 21 路 softmax 回归分类器）用作对象检测器，会更简洁。我们尝试了这种方法，发现 VOC 2007 的 mAP 性能从 54.2% 下降到 50.9%。性能下降可能是多种因素共同作用的结果，其中包括微调中使用的正面示例定义并不强调精确定位，而且 softmax 分类器是在随机抽样的负面示例上进行训练的，而不是在 SVM 训练中使用的 "硬负面 "子集上进行训练。

这一结果表明，经过微调后，无需训练 SVM 也能获得接近相同水平的性能。我们推测，如果对微调进行一些额外的调整，剩余的性能差距可能会缩小。果真如此，这将简化并加快 R-CNN 的训练，而不会降低检测性能。

C. 边界框回归

我们使用简单的边界框回归阶段来提高定位性能。在使用特定类别的检测 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为检测预测一个新的边界框。这与可变形部件模型中使用的边界框回归法[17]在精神上相似。这两种方法的主要区别在于，这里我们使用 CNN 计算出的特征进行回归，而不是根据推断出的 DPM 零件位置计算出的几何特征进行回归。

The input to our training algorithm is a set of N training pairs $\{(P_i,G_i)\}_{i=1,\ldots,N}$, where $P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$ specifies the pixel coordinates of the center of proposal P^i's bounding box together with P^i's width and height in pixels. Hence forth, we drop the superscript i unless it is needed. Each ground-truth bounding box G is specified in the same way: $G = (G_x,G_y,G_w,G_h)$. Our goal is to learn a transformation that maps a proposed box P to a ground-truth box G.

We parameterize the transformation in terms of four functions $d_x(P)$, $d_y(P)$, $d_w(P)$, and $d_h(P)$. The first two specify a scale-invariant translation of the center of P's bounding box, while the second two specify log-space translations of the width and height of P's bounding box. After learning these functions, we can transform an input proposal P into a predicted ground-truth box $\hat{G}$ by applying the transformation

$$\hat{G}_x = P_w d_x(P) + P_x \quad (1)$$

$$\hat{G}_y = P_h d_y(P) + P_y \quad (2)$$

$$\hat{G}_w = P_w \exp(d_w(P)) \quad (3)$$

$$\hat{G}_h = P_h \exp(d_h(P)). \quad (4)$$

Each function $d_\star(P)$ (where $\star$ is one of x,y,h,w) is modeled as a linear function of the pool_5 features of proposal P, denoted by $\varphi_5(P)$. (The dependence of $\varphi_5(P)$ on the image data is implicitly assumed.) Thus we have $d_\star(P) = \mathbf{w}_\star^{\mathrm{T}}\varphi_5(P)$, where $\mathbf{w}_\star$ is a vector of learnable model parameters. We learn $\mathbf{w}_\star$ by optimizing the regularized least squares objective (ridge regression):

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}}\varphi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2. \quad (5)$$

The regression targets $t_\star$ for the training pair (P,G) are defined as

$$t_x = (G_x - P_x)/P_w \quad (6)$$

$$t_y = (G_y - P_y)/P_h \quad (7)$$

$$t_w = \log(G_w/P_w) \quad (8)$$

$$t_h = \log(G_h/P_h). \quad (9)$$

As a standard regularized least squares problem, this can be solved efficiently in closed form.

We found two subtle issues while implementing bounding-box regression. The first is that regularization is important: we set $\lambda = 1000$ based on a validation set. The second issue is that care must be taken when selecting which training pairs (P,G) to use. Intuitively, if P is far from all ground-truth boxes, then the task of transforming P to a ground-truth box G does not make sense.

Using examples like P would lead to a hopeless learning problem. Therefore, we only learn from a proposal P if it is *nearby* at least one ground-truth box. We implement "nearness" by assigning P to the ground-truth box G with which it has maximum IoU overlap (in case it overlaps more than one) if and only if the overlap is greater than a threshold (which we set to 0.6 using a validation set). All unassigned proposals are discarded. We do this once for each object class in order to learn a set of class-specific bounding-box regressors.

At test time, we score each proposal and predict its new detection window only once. In principle, we could iterate this procedure (i.e., re-score the newly predicted bounding box, and then predict a new bounding box from it, and so on). However, we found that iterating does not improve results.

D. 其他功能可视化

图 12 显示了 20 个池 5 单元的其他可视化效果。对于每个单元，我们显示了在所有 VOC 2007 测试的约 1,000 万个区域中，最大程度激活该单元的 24 个区域提案。

我们用每个单元在 6 × 6 × 256 维 pool5 特征图中的 (y, x, 通道) 位置来标记它。在每个通道中，CNN 计算的输入区域函数完全相同，（y、x）位置只改变感受野。

E. 每个类别的细分结果

表 7 显示了除 O2P 方法 [4] 之外的六种分割方法在 VOC 2011 val 上的每类分割准确率。这些结果表明，在 20 个 PASCAL 类别中的每个类别中，哪种方法的准确率最高，另外还有背景类别。

F. 跨数据集冗余分析

在辅助数据集上进行训练时，一个令人担忧的问题是辅助数据集与测试集之间可能存在冗余。尽管物体检测和整体图像分类的任务有很大不同，使得这种跨集冗余的担忧大大降低，但我们仍然进行了彻底的调查，量化了 PASCAL 测试图像包含在 ILSVRC 2012 训练集和验证集中的程度。我们的研究结果可能会对有意使用 ILSVRC 2012 作为 PASCAL 图像分类任务训练数据的研究人员有所帮助。

我们对重复（和接近重复）的图像进行了两次检查。第一项检查基于 Flickr 图像 ID 的精确匹配，这些 ID 已包含在 VOC 2007 测试注释中（这些 ID 在后续的 PASCAL 测试集中有意保密）。所有 PASCAL 图像和大约一半的 ILSVRC 图像都是从 flickr.com 收集的。这项检查在 4952 张图片中找到了 31 张匹配图片（0.63%）。

第二项检查使用 GIST [30] 描述符匹配，[13] 中显示该描述符在大型（大于 100 万）图像集合中的近乎重复图像检测方面具有出色的性能。按照文献[13]，我们在所有 ILSVRC 2012 训练值和 PASCAL 2007 测试图像的 32 × 32 像素翘曲版本上计算了 GIST 描述符。

GIST 描述符的欧氏距离近邻匹配发现了 38 幅近乎重复的图像（包括通过 flickr ID 匹配发现的全部 31 幅图像）。这些匹配图像在 JPEG 压缩级别和分辨率方面略有不同，在裁剪方面也略有不同。这些发现表明，重叠率很小，不到 1%。对于 VOC 2012，由于没有 Flickr ID，我们只使用了 GIST 匹配方法。根据 GIST 匹配结果，1.5% 的 VOC 2012 测试图像出现在 ILSVRC 2012 trainval 中。VOC 2012 的匹配率稍高，可能是因为这两个数据集的收集时间比 VOC 2007 和 ILSVRC 2012 更近。

G. 文件更新日志

本文档跟踪 R-CNN 的进展情况。为了帮助读者了解它随着时间的推移发生了哪些变化，这里有一个简短的更新日志，描述了修订的内容。

v1 初始版本。

v2 CVPR 2014 相机就绪修订版。包括检测性能的大幅提升，其原因是：（1）从更高的学习率（0.001 而不是 0.0001）开始微调；（2）在

VOC 2011 val	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
O_2P [4]	**84.0**	**69.0**	21.7	47.7	42.2	42.4	**64.7**	**65.8**	57.4	**12.9**	37.4	20.5	43.7	35.7	52.7	51.0	**35.8**	**51.0**	28.4	59.8	49.7	46.4
full R-CNN fc_6	81.3	56.2	23.9	42.9	40.7	38.8	59.2	56.5	53.2	11.4	34.6	16.7	48.1	37.0	51.4	46.0	31.5	44.0	24.3	53.7	51.1	43.0
full R-CNN fc_7	81.0	52.8	**25.1**	43.8	40.5	42.7	55.4	57.7	51.3	8.7	32.5	11.5	48.1	37.0	50.5	46.4	30.2	42.1	21.2	57.7	**56.0**	42.5
fg R-CNN fc_6	81.4	54.1	21.1	40.6	38.7	**53.6**	59.9	57.2	52.5	9.1	36.5	**23.6**	46.4	38.1	53.2	51.3	32.2	38.7	**29.0**	53.0	47.5	43.7
fg R-CNN fc_7	80.9	50.1	20.0	40.2	34.1	40.9	59.7	59.8	52.7	7.3	32.1	14.3	48.8	42.9	54.0	48.6	28.9	42.6	24.9	52.2	48.8	42.1
full+fg R-CNN fc_6	83.1	60.4	23.2	48.4	**47.3**	52.6	61.6	60.6	**59.1**	10.8	**45.8**	20.9	**57.7**	43.3	**57.4**	**52.9**	34.7	48.7	28.1	60.0	48.6	**47.9**
full+fg R-CNN fc_7	82.3	56.7	20.6	**49.9**	44.2	43.6	59.3	61.3	57.8	7.7	38.4	15.1	53.4	**43.7**	50.8	52.0	34.1	47.8	24.7	**60.1**	55.2	45.7

表 7：VOC 2011 验证集上的每类分割准确率（%）。

准备 CNN 输入时使用上下文填充；以及（3）使用边界框回归来修复定位错误。

v3 ILSVRC2013 检测数据集的结果以及与 OverFeat 的比较被纳入多个章节（主要是第 2 节和第 4 节）。

v4 附录 B 中的 softmax vs. SVM 结果包含一个错误，现已修复。感谢 Sergio Guadarrama 帮助我们发现了这个问题。

v5 在第 3.3 节和表 3 中添加了使用 Simonyan 和 Zisserman [43] 的新 16 层网络架构得出的结果。

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 2012. 2

[2] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and´ J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 10, 11

[3] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Ma-´ lik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 4, 10, 11, 13, 14

[5] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 2, 3

[6] D. Cires¸an, A. Giusti, L. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *MICCAI*, 2013. 3

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[8] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013. 3

[9] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. FeiFei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). http://www.image-net.org/ challenges/LSVRC/2012/. 1

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[11] J. Deng, O. Russakovsky, J. Krause, M. Bernstein, A. C. Berg, and L. Fei-Fei. Scalable multi-label annotation. In *CHI*, 2014. 8

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *ICML*, 2014. 2

[13] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and´ C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proc. of the ACM International Conference on Image and Video Retrieval*, 2009. 13

[14] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 3

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1, 4

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013. 10

[17] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 2, 4, 7, 12

[18] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 4, 5

[19] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 1

[20] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://www.cs.berkeley.edu/~rbg/latent-v5/. 2, 5, 6, 7

[21] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik.´ Recognition using regions. In *CVPR*, 2009. 2

[22] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik.´ Semantic contours from inverse detectors. In *ICCV*, 2011. 10

[23] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*. 2012. 2, 7, 8

class	AP	class	AP	class	AP	class	AP	class	AP
accordion	50.8	centipede	30.4	hair spray	13.8	pencil box	11.4	snowplow	69.2
airplane	50.0	chain saw	14.1	hamburger	34.2	pencil sharpener	9.0	soap dispenser	16.8
ant	31.8	chair	19.5	hammer	9.9	perfume	32.8	soccer ball	43.7
antelope	53.8	chime	24.6	hamster	46.0	person	41.7	sofa	16.3
apple	30.9	cocktail shaker	46.2	harmonica	12.6	piano	20.5	spatula	6.8
armadillo	54.0	coffee maker	21.5	harp	50.4	pineapple	22.6	squirrel	31.3
artichoke	45.0	computer keyboard	39.6	hat with a wide brim	40.5	ping-pong ball	21.0	starfish	45.1
axe	11.8	computer mouse	21.2	head cabbage	17.4	pitcher	19.2	stethoscope	18.3
baby bed	42.0	corkscrew	24.2	helmet	33.4	pizza	43.7	stove	8.1
backpack	2.8	cream	29.9	hippopotamus	38.0	plastic bag	6.4	strainer	9.9
bagel	37.5	croquet ball	30.0	horizontal bar	7.0	plate rack	15.2	strawberry	26.8
balance beam	32.6	crutch	23.7	horse	41.7	pomegranate	32.0	stretcher	13.2
banana	21.9	cucumber	22.8	hotdog	28.7	popsicle	21.2	sunglasses	18.8
band aid	17.4	cup or mug	34.0	iPod	59.2	porcupine	37.2	swimming trunks	9.1
banjo	55.3	diaper	10.1	isopod	19.5	power drill	7.9	swine	45.3
baseball	41.8	digital clock	18.5	jellyfish	23.7	pretzel	24.8	syringe	5.7
basketball	65.3	dishwasher	19.9	koala bear	44.3	printer	21.3	table	21.7
bathing cap	37.2	dog	76.8	ladle	3.0	puck	14.1	tape player	21.4
beaker	11.3	domestic cat	44.1	ladybug	58.4	punching bag	29.4	tennis ball	59.1
bear	62.7	dragonfly	27.8	lamp	9.1	purse	8.0	tick	42.6
bee	52.9	drum	19.9	laptop	35.4	rabbit	71.0	tie	24.6
bell pepper	38.8	dumbbell	14.1	lemon	33.3	racket	16.2	tiger	61.8
bench	12.7	electric fan	35.0	lion	51.3	ray	41.1	toaster	29.2
bicycle	41.1	elephant	56.4	lipstick	23.1	red panda	61.1	traffic light	24.7
binder	6.2	face powder	22.1	lizard	38.9	refrigerator	14.0	train	60.8
bird	70.9	fig	44.5	lobster	32.4	remote control	41.6	trombone	13.8
bookshelf	19.3	filing cabinet	20.6	maillot	31.0	rubber eraser	2.5	trumpet	14.4
bow tie	38.8	flower pot	20.2	maraca	30.1	rugby ball	34.5	turtle	59.1
bow	9.0	flute	4.9	microphone	4.0	ruler	11.5	tv or monitor	41.7
bowl	26.7	fox	59.3	microwave	40.1	salt or pepper shaker	24.6	unicycle	27.2
brassiere	31.2	french horn	24.2	milk can	33.3	saxophone	40.8	vacuum	19.5
burrito	25.7	frog	64.1	miniskirt	14.9	scorpion	57.3	violin	13.7
bus	57.5	frying pan	21.5	monkey	49.6	screwdriver	10.6	volleyball	59.7
butterfly	88.5	giant panda	42.5	motorcycle	42.2	seal	20.9	waffle iron	24.0
camel	37.6	goldfish	28.6	mushroom	31.8	sheep	48.9	washer	39.8
can opener	28.9	golf ball	51.3	nail	4.5	ski	9.0	water bottle	8.1
car	44.5	golfcart	47.9	neck brace	31.6	skunk	57.9	watercraft	40.9
cart	48.0	guacamole	32.3	oboe	27.5	snail	36.2	whale	48.6
cattle	32.3	guitar	33.1	orange	38.8	snake	33.8	wine bottle	31.2
cello	28.9	hair dryer	13.0	otter	22.2	snowmobile	58.8	zebra	49.6

表 8：ILSVRC2013 检测测试集的每类平均精度（%）。

[24] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe. berkeleyvision.org/, 2013. 3

[25] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3, 4, 7

[26] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comp.*, 1989. 1

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. *Proc. of the IEEE*, 1998. 1

[28] J. J. Lim, C. L. Zitnick, and P. Dollar.´ Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013. 6, 7

[29] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[30] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 13

[31] X. Ren and D. Ramanan. Histograms of sparse codes for

[32] H. A. Rowley, S. Baluja, and T. Kanade. Neural networkbased face detection. *TPAMI*, 1998. 2

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing*, 1:318–362, 1986. 1

[34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *ICLR*, 2014. 1, 2, 4, 10

[35] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, 2013. 2

[36] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI Technical Report, 4th Human Computation Workshop*, 2012. 8

[37] K. Sung and T. Poggio. Example-based learning for viewbased human face detection. Technical Report A.I. Memo No. 1521, Massachussets Institute of Technology, 1994. 4

[38] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013. 2

[39] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 2, 3, 4, 5, 9

[40] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc on Vision, Image, and Signal Processing*, 1994. 2

[41] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 3, 5

[42] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *CVPR*, 2011. 4

[43] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint*, arXiv:1409.1556, 2014. 6, 7, 14