用于精准物体检测和语义分割的丰富特征层次结构

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik

摘要

在过去的几年里，在典型的 PASCAL VOC 数据集上测量的物体检测性能已经趋于平稳。表现最好的方法是复杂的组合系统，通常将多个低层次的图像特征与高层次的背景相结合。在本文中，我们提出了一种简单的、可扩展的检测算法，相对于 VOC 2012 上的最佳结果，该算法的平均精度（mAP）提高了 30% 以上，达到了 53.3%。我们的方法结合了两个关键的见解：(1) 我们可以将大容量卷积神经网络（CNN）应用于自下而上的候选区域，来对物体进行定位和分割；(2) 当标记的训练数据不足时，对辅助任务进行监督预训练，然后再进行特定领域的微调，可以产生显著的性能提升。由于我们将区域候选与 CNN 结合起来，我们将我们的方法称为 R-CNN：具有 CNN 特征的区域。我们还将 R-CNN 与 OverFeat 进行了比较，后者是最近提出的基于类似 CNN 架构的滑动窗口检测器。我们发现，在 200 类 ILSVRC2013 检测数据集上，R-CNN 比 OverFeat 的性能高出很多。系统源代码可见http://www.cs.berkeley.edu/~rbg/rcnn。

1 简介

特征很重要。在过去的十年里，各种视觉识别任务的进展主要基于 SIFT[2] 和 HOG[3] 的使用。但是，如果我们看一下它们在典型的视觉识别任务的表现，即 PASCAL VOC 物体检测 [15]，人们普遍承认在 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获得了小的收益。

SIFT 和 HOG 是顺时针方向的直方图，这种表示方法我们可以大致与 V1 的复杂细胞联系起来，V1 是灵长类动物视觉通路的第一个皮质区域。但我们也知道，识别发生在下游的几个阶段，这表明可能有分层的、多阶段的过程来计算对视觉识别更有参考价值的特征。

Fukushima 的"neocognitron"[19]，一个受生物启发的分层并具有移位不变性的模式识别模型，就是对这样一个过程的早期尝试。然而，neocognitron 缺乏一个监督训练算法。在 Rumelhart 等人 [33] 的基础上，LeCun 等人 [26] 表明，通过反向传播的随机梯度下降对训练卷积神经网络（CNN）是有效的，CNN 是扩展 neocognitron 的一类模型。

CNN 在 20 世纪 90 年代得到了大量使用（例如，[1]），但随后随着支持向量机的兴起而逐渐淡出人们的视野。2012 年，Krizhevsky 等人 [8] 在 ImageNet 大规模视觉识别挑战赛（ILSVRC）[4] 上展示了大幅提高的图像分类准确性，重新点燃了人们对 CNN 的兴趣。他们的成功来自于在 120 万张标记图像上训练一个大型的 CNN，以及对 LeCun 的 CNN 的一些改变（例如，ReLU 非线性和 dropout 正则化）。

在 ILSVRC 2012 研讨会上，大家对 ImageNet 结果的意义进行了激烈的辩论。核心问题可以提炼为以下内容。ImageNet 上的 CNN 分类结果在多大程度上可以推广到 PASCAL VOC 挑战赛的物体检测结果？

我们通过弥合图像分类和物体检测之间的差距来回答这个问题。本文首次表明，与基于更简单的类似 HOG 的特征的系统相比，CNN 可以使 PASCAL VOC 的物体检测性能大幅提高。为了实现这一结果，我们重点研究了两个问题：用深度网络定位物体，以及用少量的注释检测数据训练一个大容量的模型。

与图像分类不同，检测需要对图像中的（可能是许多）物体进行定位。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] 的工作表明，这种策略在实践中可能并不理想（他们报告说，2007 年 VOC 的 mAP 为 30.5%，而我们的方法为 58.5%）。另一个选择是建立一个滑动窗口检测器。CNN 已经以这种方式使用了至少 20 年，通常用于受限的物体类别，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。我们也考虑过采用滑动窗口的方法。然而，在我们的网络中，有五个卷积层的高位单元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），这使得滑动窗口范式中的精确定位成为一个开放的技术挑战。

相反，我们通过“使用区域识别”范式 [21] 来解决 CNN 的定位问题，该范式在物体检测 [39] 和语义分割 [5] 中都很成功。在测试时，我们的方法为输入图像生成大约 2000 个与类别无关的候选区域，使用 CNN 从每个候选中提取一个固定长度的特征向量，然后用特定类别的线性 SVM 对每个区

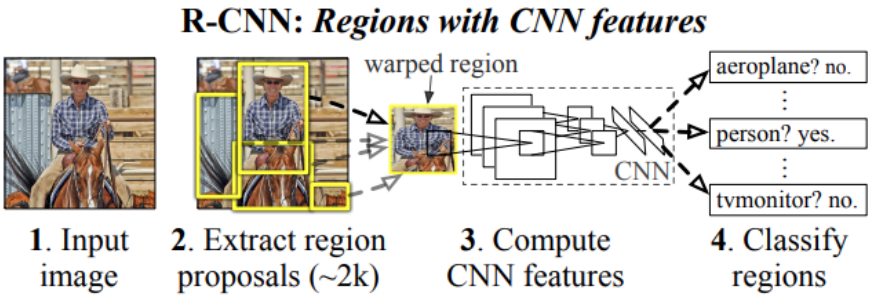

图 1: **物体检测系统概述**。我们的系统 (1) 接受一个输入图像，(2) 提取大约 2000 个自下而上的区域候选，(3) 使用一个大型卷积神经网络（CNN）计算每个候选的特征，然后 (4) 使用类别特定的线性 SVM 对每个区域进行分类。R-CNN 在 PASCAL VOC 2010 上达到了 53.7% 的平均精度（mAP）。作为比较，[39] 报告说，使用相同的区域建议，但采用空间金字塔和视觉词汇袋的方法，达到了 35.1% 的 mAP。流行的可变形部分模型的表现为 33.4%。在 200 级 ILSVRC2013 检测数据集上，R-CNN 的 mAP 为 31.4%，比 OverFeat[34] 有了很大的改进，它之前的最佳结果是 24.3%。

域进行分类。我们使用一种简单的技术（仿生图像扭曲）为每个候选区域计算得到一个固定大小的 CNN 输入，而不必考虑该区域的形状。图1展示了我们方法的概况，并强调了我们的一些结果。由于我们的系统将区域候选与 CNN 结合起来，我们将该方法称为 R-CNN：具有 CNN 特征的区域。

在本文的更新版本中，我们通过在有 200 类的 ILSVRC2013 检测数据集上运行 R-CNN，对 R-CNN 和最近提出的 OverFeat[9] 检测系统进行了正面的比较。OverFeat 使用滑动窗口 CNN 进行检测，到目前为止是 ILSVRC2013 检测中表现最好的方法。我们表明，R-CNN 明显优于 OverFeat，其 mAP 为 31.4% 而不是 24.3%。

检测中面临的第二个挑战是，标注的数据很少，目前可用的数据量不足以训练一个大型 CNN。这个问题的传统解决方案是使用无监督的预训练，然后再进行有监督的微调（例如，[35]）。本文的第二个主要贡献是表明，在一个大的辅助数据集（ILSVRC）上进行有监督的预训练，然后在一个小的数据集（PASCAL）上进行特定领域的微调，是在数据匮乏时学习大容量 CNN 的一个有效范式。在我们的实验中，检测的微调使 mAP 性能提高了 8 个百分点。经过微调，我们的系统在 2010 年的 VOC 上实现了 54% 的

mAP，而高度调整的、基于 HOG 的可变形部件模型（DPM）的 mAP 为 33%[17, 20]。我们还向读者指出 Donahue 等人 [12] 的同期工作，他们表明 Krizhevsky 的 CNN 可以作为一个黑盒特征提取器使用（不需要微调），在几个识别任务上产生出色的性能，包括场景分类、细粒度的子分类和领域适应。

我们的系统也是相当高效的。唯一针对类别的计算是一个相当小的矩阵-向量乘积和贪心非最大抑制。这一计算特性来自于所有类别共享的特征，这些特征也比以前使用的区域特征低两个数量级（参见 [39]）。

了解我们方法的失败模式对于改进它也很关键，因此我们报告了来自 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，我们证明了一个简单的边界盒回归方法大大减少了错误定位，而这是最主要的错误模式。

在发展技术细节之前，我们注意到，由于 R-CNN 是在区域上操作的，因此很自然地将其扩展到语义分割的任务中。经过细微的修改，我们在 PASCAL VOC 分割任务上也取得了有竞争力的结果，在 VOC 2011 测试集上的平均分割精度为 47.9%。

2 使用 R-CNN 进行物体检测

我们的物体检测系统由三个模块组成。第一个模块产生独立于类别的候选区域。这些候选定义了可供我们的检测器使用的候选检测的集合。第二个模块是一个从每个区域提取一个固定长度的特征向量的大型卷积神经网络。第三个模块是一组类别特定的线性 SVM。在本节中，我们将介绍我们对每个模块的设计决定，描述它们在测试时的使用情况，详细说明它们的参数是如何习得的，并展示在 PASCAL VOC 2010-12 和 ILSVRC2013 上的检测结果。

2.1 模块设计

区域候选。 最近的论文提供了各种生成与类别无关的区域候选的方法。例子包括：objectness[1]、selective search[39]、category-independent object proposals[14]、constrained parametric min-cuts（CPMC）[5]、multi-scale combinatorial grouping[3] 和 Cires¸an 等人 [6] 通过将 CNN 应用于规则间隔的方形作物，检测有丝分裂细胞，这是区域建议的一个特殊案例。虽然

R-CNN 与特定的区域提议方法无关，但我们使用选择性搜索，以便与先前的检测工作（例如，[39，41]）进行控制变量比较。

特征提取。 我们使用 Krizhevsky 等人 [8] 描述的 CNN 的 Caffe[10] 实现，从每个候选区域中提取 4096 维的特征向量。特征的计算通过将减去均值的 227×227RGB 图像通过五个卷积层和两个全连接层进行前向传播。关于更多的网络结构细节，我们请读者参考 [8, 10]。

图 2: 来自 VOC 2007 训练集中的**扭曲训练样本**。

为了计算一个候选区域的特征，我们必须首先将该区域的图像数据转换为与 CNN 兼容的格式（其架构要求输入大小为固定的 227×227 像素）。在对于任意形状区域的许多可能的转换中，我们选择了最简单的。无论候选区域的大小或长宽比如何，我们都会将其周围的紧缩边界框中的所有像素扭曲到所需的大小。在变形之前，我们扩张紧缩边界框，以便在变形的尺寸下，原始框周围正好有 p 个像素的变形图像背景（我们使用 $p = 16$）。图2显示了一个随机抽样的被包裹的训练区域。附录A中讨论了可替代的扭曲方法。

2.2 检测的测试阶段

在测试时间，我们对测试图像进行选择性搜索，以提取大约 2000 个候选区域（我们在所有实验中使用选择性搜索的” 快速模式”）。我们对每个候选进行扭曲，并通过 CNN 进行前向传播，以计算出特征。然后，对于每个类别，我们使用为该类别训练的 SVM 对每个提取的特征向量进行评分。得到图像中的所有被打分的区域后，我们使用一个贪婪的非最大抑制（对每个类别独立），如果一个区域与一个更高得分的选定区域的交叉重叠（IoU）大于习得的阈值，则拒绝该区域。

运行分析。 有两个特性使检测变得高效。首先，所有的 CNN 参数在所有类别中都是共享的。第二，与其他常见的方法相比，CNN 计算的特征向量是低维的，如带有视觉词包编码的空间金字塔。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 vs. 4k-维）。

这种共享的结果是，计算区域候选和特征的时间（GPU 上的 13s/图像或 CPU 上的 53s/图像）被所有类别分摊。唯一针对类的计算是特征和 SVM 权重之间的点乘和非最大抑制。在实践中，一个图像的所有点积都被打包成一个单一的矩阵-矩阵乘积。特征矩阵通常为 2000×4096，SVM 权重矩阵为 4096×N，其中 N 为类的数量。

这一分析表明，R-CNN 可以在不借助近似的技术，如散列，的情况下扩展到成千上万的类别。即使有 10 万个类，所产生的矩阵乘法在现代多核 CPU 上只需要 10 秒。这种效率不仅仅是使用区域候选和共享特征的结果。UVA 系统，由于其高维特征，会慢两个数量级，同时需要 134GB 的内存来存储 10 万个线性预测器，而我们的低维特征只需要 1.5GB。

将 R-CNN 与 Dean 等人 [8] 最近关于使用 DPM 和散列的可扩展检测工作进行对比也很有意思。他们报告说，在引入 10k 个干扰物类别时，VOC 2007 的 mAP 约为 16%，每幅图像的运行时间为 5 分钟。使用我们的方法，10k 个检测器可以在 CPU 上运行约 1 分钟，而且由于没有进行近似，mAP 将保持在 59%（3.2 节）。

2.3 训练

监督预训练。 我们在一个大型的辅助数据集（ILSVRC2012 分类）上对 CNN 进行判别性预训练，只使用图像级别的注释（该数据没有边界框标签）。预训练是使用开源的 Caffe CNN 库 [10] 进行的。简而言之，我们的 CNN 几乎达到了与 Krizhevsky 等人 [8] 相匹配的性能，在 ILSVRC2012 分类验证集上的 top-1 错误率比之高 2.2%。这种差异训练过程简化造成的。

特定领域微调。 为了使我们的 CNN 适应新的任务（检测）和新的领域（扭曲的候选窗口），我们仅使用扭曲的区域候选来继续对 CNN 参数进行随机梯度下降（SGD）训练。除了用一个随机初始化的（N+1）分类层（其中 N 是物体类别的数量，加上 1 是背景）取代 CNN 的 ImageNet 特定的 1000 路分类层外，CNN 的架构没有变化。对于 VOC，$N = 20$，对于 ILSVRC2013，$N = 200$。我们将所有与 ground-truth 框重叠 ≥ 0.5 的区域候选作为该框类

别的正例，其余的作为负例。我们以 0.001 的学习率（初始预训练率的 1/10）开始 SGD，这允许微调在不使初始化失效的情况下取得进展。在 SGD 的每次迭代中，我们均匀地对 32 个正例窗口（所有类别）和 96 个背景窗口进行采样，以构建一个 128 大小的迷你批。我们之所以偏向于对正例窗口进行抽样，是因为与背景相比，正例窗口是非常罕见的。

物体类别分类器。 考虑训练一个检测汽车的二分类器。很明显，一个紧密包围着汽车的图像区域应该是一个正例。同样，很明显，一个与汽车无关的背景区域应该是一个负例。不太清晰的是如何标记一个与汽车部分重叠的区域。我们用 IoU 重叠阈值来解决这个问题，低于这个阈值的区域被定义为负例。重叠阈值，0.3，是通过在验证集上对 $\{0, 0.1, \ldots, 0.5\}$ 进行网格搜索得到的。我们发现，精心选择这个阈值是很重要的。如同在 [39] 中，将其设置为 0.5，使 mAP 减少了 5 个点。同样地，将其设置为 0 会使 mAP 减少 4 个点。正例被简单地定义为每个类别的 ground-truth 边界框。

一旦提取了特征并应用了训练标签，我们就为每个类别优化一个线性 SVM。由于训练数据太大，无法装入内存，我们采用了标准的难负例挖掘方法 [17, 37]。难负例挖掘方法收敛很快，在实践中，mAP 仅在对所有图像进行一次处理后就不再增加。

在附录B中，我们讨论了为什么在微调与 SVM 训练中对正负样本的定义是不同的。我们还讨论了训练检测 SVM 所涉及的权衡，而不是简单地使用微调 CNN 最后的 softmax 层的输出。

2.4 在 PASCAL VOC 2010-12 上的结果

按照 PASCAL VOC 的最佳实践 [15]，我们在 VOC 2007 数据集上验证了所有的设计决策和超参数（3.2 节）。对于 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 上优化了我们的检测 SVMs。对于两种主要的算法变体（有边界盒回归和无边界盒回归），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 上的完整结果。我们将我们的方法与四个强大的基线进行比较，包括 SegDPM[18]，它将 DPM 检测器与语义分割系统 [4] 的输出结合起来，并使用额外的检测器之间的背景和图像分类器的重新评分。最有意义的是与 Uijlings 等人 [39] 的 UVA 系统的比较，因为它与我们的系统使用了相同的区域候选算法。为了对区域进行分类，他们的方法建立

了一个四级空间金字塔，并用密集采样的 SIFT、Extended OpponentSIFT 和 RGBSIFT 描述符填充它，每个向量用 4000 字的编码簿量化。分类是用直方图相交核 SVM 进行的。与他们的多特征、非线性核 SVM 方法相比，我们在 mAP 方面取得了很大的改进——从 35.1% 到 53.7% 的 mAP，同时速度也更快（3.2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 在 ILSVRC2013 检测上的结果

我们使用了与 PASCAL VOC 相同的系统超参数，在 200 类 ILSVRC2013 检测数据集上运行 R-CNN。我们遵循同样的协议，向 ILSVRC2013 评估服务器只提交了两次测试结果，一次有边界框回归，一次没有。

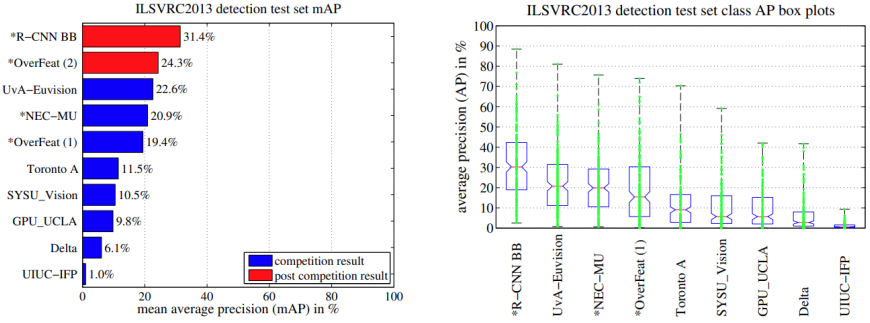

图 3: **(左) ILSVRC2013 检测测试集的平均精度**。前面带 * 的方法表示使用了外部训练数据 (在所有情况下，图像和标签来自 ILSVRC 分类数据集)。**(右) 每个方法的 200 个平均精度值的箱形图**。没有显示 OverFeat 赛后结果的箱形图，因为还没有可用的每类的 AP（R-CNN 的每类 AP 见表 8。R-CNN 的每级 AP 在表 8 中，也包括在上传到 arXiv.org 的技术报告源中；见 R-CNN-ILSVRC2013-APs.txt）。红色的线标志着 AP 的中位数，方框底部和顶部是第 25 和 75 个比例。胡须延伸至每种方法的最小和最大 AP。每个 AP 在晶须上被绘制成一个绿色的点（最好用数字放大查看）。

图3将 R-CNN 与 ILSVRC2013 比赛中的参赛模型以及 OverFeat[9] 的赛后结果进行了比较。R-CNN 实现了 31.4% 的 mAP，明显领先于 OverFeat 的第二好成绩 24.3%。为了让大家了解 AP 在不同类别中的分布情况，我们还展示了箱形图，并在文末的表 8 中列出了每类 AP 的表格。大多数竞争

者（OverFeat、NEC-MU、UvAEuvision、Toronto A 和 UIUC-IFP）都使用了卷积神经网络，这表明在将 CNN 应用于物体检测的细微差别将导致结果的大不相同。

在第 4 节中，我们概述了 ILSVRC2013 检测数据集，并提供了我们在其上运行 R-CNN 时所作选择的细节。

3 可视化、消融实验和误差类型

3.1 可视化习得特征

第一层过滤器可以直接可视化，而且很容易理解 [8]。它们能捕捉到定向的边缘和对手的颜色。理解后续层则更具挑战性。Zeiler 和 Fergus 在 [6] 中提出了一种视觉上有吸引力的去卷积方法。我们提出了一个简单的（和互补的）非参数方法，直接显示网络学到了什么。

我们的想法是在网络中挑出一个特定的单元（特征），并将其作为一个物体检测器来使用。也就是说，我们计算出该单元在一大组被搁置的区域候选（大约 1000 万）上的激活，将这些建议从最高激活到最低激活进行排序，进行非最大抑制，然后显示出得分最高的区域。我们的方法让所选的单元”为自己说话”，准确地显示它在哪些输入上开火。我们避免平均化，以便看到不同的视觉模式，并深入了解该单元计算的不变性。

我们将 pool5 层的单元可视化，它是网络第五层也是最后一层卷积层的最大池化输出。pool5 的特征图是 $6 \times 6 \times 256 = 9216$ 维的。忽略边界效应，每个 pool5 单元在原始 227×227 像素的输入中具有 195×195 像素的感受野。一个中央的 pool5 单元有着几乎全局的视野，而靠近边缘的单元有较小的、被剪切的视野。

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。在 256 个功能独特的单元中，有 6 个是做了可视化的（附录 D 包括更多）。选择这些单元是为了显示网络学习的代表性样本。在第二行，我们看到一个单元在狗脸和点阵上发射。第三行对应的单元是一个红色的圆球检测器。还有一些检测器用于检测人脸和更抽象的图案，如文字和带窗口的三角形结构。该网络似乎在学习一种表征，将少量的类调整特征与形状、纹理、颜色和材料属性的分布式表征结合在一起。随后的全连接层 fc6 有能力对这些丰富特征的大量组合进行建模。

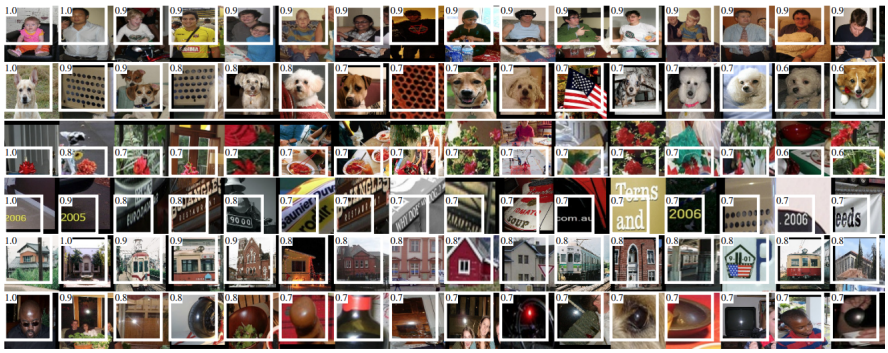

图 4: **六个 pool5 单元的顶部区域**。感受视野和激活值以白色绘制。一些单元与概念相一致，如人（第 1 行）或文本（4）。其他单元捕捉纹理和材料属性，如点阵（2）和镜面反射（6）。

3.2 消融实验

在无微调设置下的逐层性能。 为了了解哪些层对检测性能至关重要，我们分析了 CNN 最后三层中每一层在 VOC 2007 数据集上的结果。第3.1节中简要介绍了 pool5。下面总结了最后两层的情况。

fc6 层与 pool5 完全连接。为了计算特征，它用一个 4096×9216 的权重矩阵乘以 pool5 的特征图（重塑为 9216 维的向量），然后加上一个偏置向量。这个中间向量的每一个分量都要经过半波整流（$x = \max(0; x)$）。

fc7 层是网络的最后一层。它是通过将 fc6 计算出的特征乘以 4096×4096 的权重矩阵来实现的，同样地，加入一个偏置矢量并应用半波整流。

我们首先看一下没有在 PASCAL 上进行微调的 CNN 的结果，即所有的 CNN 参数都只在 ILSVRC 2012 上进行了预训练。逐层分析性能（表 2 第 1-3 行），发现 fc7 的特征比 fc6 的特征泛化性能差。这意味着 29%，即大约 1680 万个 CNN 的参数可以被移除而不降低 mAP。更令人惊讶的是，尽管 pool5 的特征只用了 CNN 的 6% 的参数计算，但去除 fc7 和 fc6 都会产生相当好的结果。CNN 的大部分表征能力来自其卷积层，而不是来自更大的密集连接层。这一发现表明，只用 CNN 的卷积层就能计算出任意大小图像的密集特征图，即 HOG 的意义。这种表示方法将使人们能够在 pool5 特征的基础上进行滑动窗口检测器的实验，包括 DPM。

在微调设置下的逐层性能。 我们现在看看我们的 CNN 在 VOC 2007 trainval 上微调了参数后的结果。改进是惊人的（表 2 第 4-6 行）：微调使 mAP 增加了 8.0 个百分点，达到 54.2%。微调对 fc6 和 fc7 的提升比对 pool5 的提升要大得多，这表明从 ImageNet 学到的 pool5 特征是通用的，大部分的改进是通过在它们上面学习特定领域的非线性分类器获得的。

与近期特征学习方法的比较。 相对来说，在 PASCAL VOC 检测上尝试的特征学习方法很少。我们看一下最近两种建立在可变形部件模型上的方法。作为参考，我们也包括基于 HOG 的标准 DPM[5] 的结果。

第一种 DPM 特征学习方法，DPM ST [28]，用" 草图标记" 概率直方图来增强 HOG 特征。直观地说，草图标记是通过图像斑块中心的轮廓线的紧密分布。草图标记概率是由一个随机森林在每个像素上计算出来的，该随机森林被训练成将 35×35 像素的斑块分类为 150 个草图标记或背景之一。

第二种方法，DPM HSC[31]，用稀疏编码直方图（HSC）取代了 HOG。为了计算 HSC，使用 100 个 7×7 像素（灰度）原子的学习字典来解决每个像素的稀疏代码激活问题。得到的激活以三种方式（全波和半波）进行整顿，空间汇集，单位 ℓ_2 归一化，然后进行功率变换（$x \leftarrow \text{sign}(x)|x|^\alpha$）。

所有的 R-CNN 变体都强烈地超越了三个 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 DPM 的最新版本相比，我们的 mAP 高出 20 多个百分点：54.2% 对 33.7%——61% 的相对改进。HOG 和草图标记的组合比单独的 HOG 产生了 2.5 个 mAP 点，而 HSC 比 HOG 提高了 4 个 mAP 点（当与他们的私有 DPM 基线进行内部比较时–两者都使用了 DPM 的非公开实现，性能低于开源版本 [20]）。这些方法的 mAPs 分别为 29.1% 和 34.3%。

3.3 网络结构

本文中的大多数结果都使用了 Krizhevsky 等人 [8] 的网络架构。然而，我们发现，架构的选择对 R-CNN 的检测性能有很大的影响。在表 3 中，我们展示了使用 Simonyan 和 Zisserman[11] 最近提出的 16 层深度网络对 VOC 2007 测试的结果。这个网络是最近 ILSVRC 2014 分类挑战中表现最好的网络之一。该网络有一个同质的结构，由 13 层 3×3 卷积核组成，中间穿插着 5 个最大池化层，顶部是三个全连接层。我们把这个网络称为"O-Net"，即 OxfordNet，把基线称为"T-Net"，即 TorontoNet。

为了在 R-CNN 中使用 O-Net，我们从 Caffe Model Zoo1 中下载了公开可用的VGG_ILSVRC_16_layers模型的预训练网络权重，然后使用与 T-Net 相同的协议对网络进行微调。唯一的区别是使用较小的迷你批（24 个例子），以适应 GPU 内存的需要。表 3 中的结果显示，使用 O-Net 的 R-CNN 大大优于使用 T-Net 的 R-CNN，将 mAP 从 58.5% 提高到 66.0%。然而，在计算时间方面有一个相当大的缺点，O-Net 的前向传递大约比 T-Net 长 7 倍。

3.4 检测误差分析

我们应用了 Hoiem 等人 [7] 的优秀检测分析工具，以揭示我们方法的错误模式，了解微调如何改变它们，并查看我们的错误类型与 DPM 的比较。对分析工具的全面总结超出了本文的范围，我们鼓励读者查阅 [7] 以了解一些更精细的细节（如" 规范化 AP"）。由于分析最好是在相关图表的背景下进行，我们在图5和图6的字幕中进行了讨论。

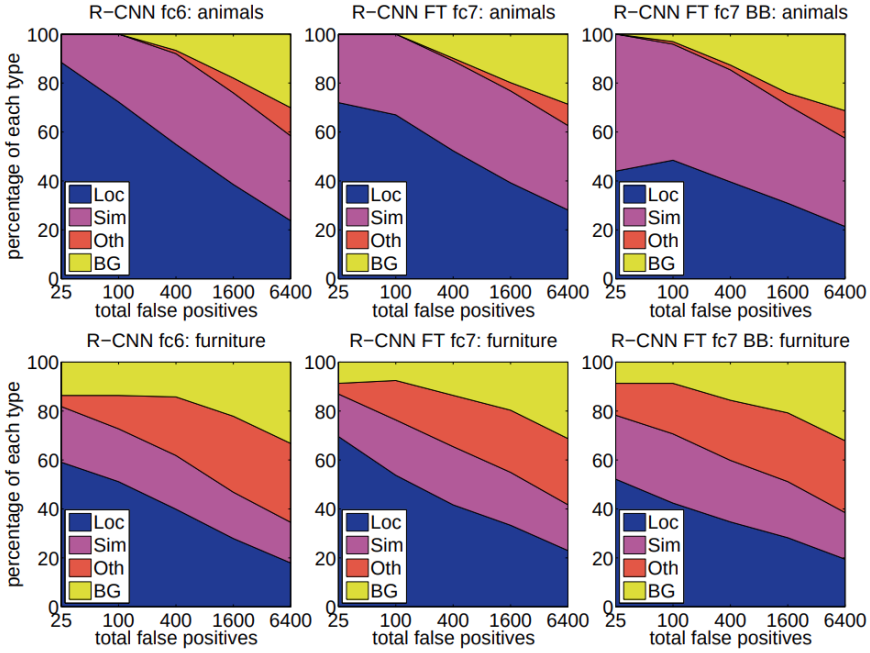

图 5: **排名靠前的误报（fasle positive, FP）类型的分布**。每幅图显示了随着分数递减产生更多误报，误报类型的演变分布。每个 FP 被分为 4 种类型中的一种：Loc-差的定位（与正确类别的 IoU 重叠在 0.1 和 0.5 之间的检测，或重复）；Sim-与相似类别的混淆；Oth-与不相似物体类别的混淆；BG-在背景上发射的误报。与 DPM 相比（见 [5]），我们的错误中明显有更多是由于定位不准确造成的，而不是与背景或其他物体类别的混淆，这表明 CNN 的特征比 HOG 更具有分辨力。松散的定位可能是由于我们使用了自下而上的区域候选和从预先训练整个图像分类的 CNN 中学到的位置不变性。第三栏显示了我们简单的边界盒回归方法如何解决了许多定位错误。

3.5 边界框回归

基于误差分析，我们实现了一个简单的方法来减少定位误差。受 DPM[5] 中采用的边界盒回归的启发，对于一个选择性搜索区域，给定 pool5 的特征，我们训练了一个线性回归模型，以预测一个新的检测窗口。完整的细节在附录 C 中给出。表 1、表 2 和图5中的结果显示，这种简单的方法修复了大量的错误定位的检测，使 mAP 提高了 3 到 4 个点。

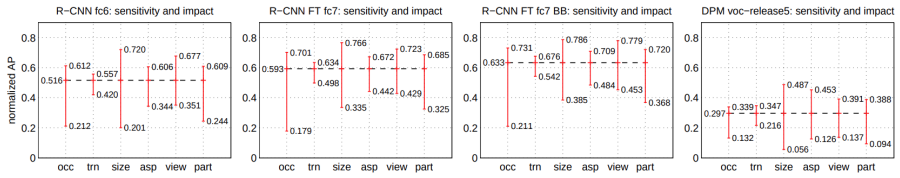

图 6: **对物体特征的敏感性**。每张图都显示了在六个不同的物体特征（遮挡、截断、边界框面积、长宽比、视角、部分可见）中，最高和最低性能子集的平均（所有类别）归一化 AP（见 [5]）。我们展示了我们的方法（R-CNN）在有微调（FT）和无微调（BB）以及 DPM voc-release5 的情况下的图表。总的来说，微调并没有降低灵敏度（最大和最小之间的差异），但对于几乎所有的特征来说，微调确实大大改善了最高和最低性能子集。这表明，微调不仅仅是简单地改善了长宽比和边界盒面积的最低性能子集，正如人们根据我们如何扭曲网络输入所猜想的那样。相反，微调提高了所有特征的鲁棒性，包括遮挡、截断、视角和部件的可见性。

A 物体候选变换

这项工作中使用的卷积神经网络需要 227×227 像素的固定输入尺寸。对于检测，我们考虑的物体候选是任意的矩形图像。我们评估了两种将物体候选转化为有效的 CNN 输入的方法。

第一种方法（”带上下文的最紧密正方形”）将每个物体候选包围在最紧密正方形内，然后将该正方形所包含的图像（各向同性）缩放到 CNN 输入尺寸。图7（B）列显示了这种转换。这种方法的一个变种（”无上下文的最紧密正方形”）排除了围绕原始物体候选的图像内容。图7（C）列显示了这种转换。第二种方法（”扭曲”）以各向异性的方式将每个物体候选扩展到 CNN 的输入尺寸。图7(D) 列显示了扭曲的变换。

图 7: **不同的物体候选转换**。(A) 相对于转换后的 CNN 输入的实际比例的原始物体候选；(B) 带上下文的最紧密正方形；(C) 无上下文的最紧密正方形；(D) 扭曲。在每一列和例子的候选中，最上面一行对应的是 $p = 0$ 像素的上下文填充，而最下面一行则是 $p = 16$ 像素的上下文填充。

对于每一个转换，我们也考虑在原始物体候选周围包括额外的图像背景。上下文填充量（p）被定义为转换后的输入坐标帧中原始对象建议周围的边界大小。图7顶部展示了每个例子中 $p = 0$ 像素的情况，底部展示了 $p = 16$ 像素的情况。在所有的方法中，如果源矩形超出了图像，缺失的数据将被替换为图像的平均值（然后在将图像输入到 CNN 之前将其减去）。一组试验表明，带有上下文填充的扭曲（$p = 16$ 像素）在很大程度上胜过了其他方法（3-5 个 mAP 点）。显然，更多的替代方案是可能的，包括使用复制而不是平均填充。对这些替代方案的详尽评估将作为未来的工作。

B 正例 vs. 负例以及 softmax

有两个设计选择值得进一步讨论。第一个是：为什么在微调 CNN 和训练物体检测 SVM 的过程中，对正反例子的定义不同？简要回顾一下定义，对于微调，我们将每个候选物体映射到与之有最大 IoU 重叠的 ground-truth 实例（如果有的话），如果 IoU 至少为 0.5，则将其标记为匹配 ground-truth 类的正例。所有其他候选都被标记为” 背景”（即所有类别的负例）。相比之

下，在训练 SVM 时，我们只把 ground-truth 框作为其各自类别的正例，并把与对应类别的所有实例重合度低于 0.3 IoU 的候选标记为该类别的负例。属于灰色区域的候选（IoU 超过 0.3 的重叠，但不属于 ground-truth）将被忽略。

从历史上看，我们之所以得出这些定义，是因为我们一开始就在 ImageNet 预训练的 CNN 计算的特征上训练 SVM，所以微调在当时并不是一个考虑因素。在那个设置中，我们发现我们训练 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）中是最佳的。当我们开始使用微调时，我们最初使用了与 SVM 训练相同的正负样例定义。然而，我们发现，结果比使用我们目前的正负定义所得到的结果要差得多。

我们的假设是，在如何定义正例和负例方面的这种差异，从根本上说并不重要，而是源于微调数据有限的事实。我们目前的方案引入了许多" 抖动" 的例子（那些重合度在 0.5 和 1 之间但不是 ground truth 的候选），这使得正例的数量扩大了大约 30 倍。我们猜想，在对整个网络进行微调时，需要这个大集合以避免过度拟合。然而，我们也注意到，使用这些抖动的例子很可能是次优的，因为网络不是为了精确定位而被微调的。

这就导致了第二个问题。为什么在微调之后还要训练 SVM? 简单地应用微调网络的最后一层，也就是一个 21 路 softmax 回归分类器，作为物体检测器会更干净。我们尝试了这个方法，发现 VOC 2007 的性能从 54.2% 下降到 50.9% mAP。这种性能下降可能是由几个因素共同造成的，包括在微调中使用的正例定义并不强调精确的定位，而且 softmax 分类器是在随机抽样的负例上训练的，而不是在用于 SVM 训练的 “难负例” 子集上训练的。

这一结果表明，在微调后不训练 SVM 也能获得接近相同水平的性能。我们猜想，通过对微调的一些额外调整，性能差距可能会被缩小。如果是真的，这将简化和加快 R-CNN 的训练，而在检测性能上没有损失。

C 边界框回归

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打分后，我们使用特定类别的边界框回归器预测检测的新边界框。这与可变形部件模型 [5] 中使用的边界盒回归相似。这两种方法的主要区别是，在这里我们从 CNN 计算的特征进行回

归，而不是从推断的 DPM 零件位置上计算的几何特征进行回归。

我们的训练算法的输入是一组 N 个训练对 $\{(P^i, G^i)\}_{i=1,\ldots,N}$，其中 $P^i = \left(P^i_x, P^i_y, P^i_w, P^i_h\right)$ 指定候选 P^i 的边界框中心的像素坐标，以及 P^i 的宽度和高度（像素）。因此，除非有必要，我们放弃上标 i。每个真实边界盒 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个将候选框 P 映射到 ground-truth 框 G 转换。

我们用四个函数 $d_x(P), d_y(P), d_w(P)$ 和 $d_h(P)$ 来确定变换的参数。前两个函数指定了 P 的边界框中心的尺度不变的平移，而后两个函数指定了 P 的边界框的宽度和高度的对数空间转换。在学习了这些函数之后，我们可以通过将转换在应用输入的候选 P 来得到预测的 ground-truth 框 $\hat{G}$

$$\begin{aligned}
\hat{G}_x &= P_w d_x(P) + P_x \\
\hat{G}_y &= P_h d_y(P) + P_y \\
\hat{G}_w &= P_w \exp\left(d_w(P)\right) \\
\hat{G}_h &= P_h \exp\left(d_h(P)\right)
\end{aligned}$$

每个函数 $d_\star(P)$(其中 $\star$ 是 x, y, h, w 中的一个) 被建模为候选 P 的 pool5 特征的线性函数，用 $\phi_5(P)$ 表示。($\phi_5(P)$ 对图像数据的依赖性是隐含的假设)。因此，我们有 $d_\star(P) = \mathbf{w}_\star^\top \phi_5(P)$，其中 $\mathbf{w}_\star$ 是一个可学习的模型参数的向量。我们通过优化正则化最小二乘法目标（脊回归）来学习 $\mathbf{w}_\star$:

$$\mathbf{w}_\star = \arg\min_{\hat{\mathbf{w}}_\star} \sum_i^N \left(t^i_\star - \hat{\mathbf{w}}_\star^\top \phi_5(P)\right)^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2$$

训练对 (P, G) 的回归目标 $t_\star$ 定义为

$$\begin{aligned}
t_x &= (G_x - P_x)/P_w \\
t_y &= (G_y - P_y)/P_h \\
t_w &= \log(G_w/P_w) \\
t_h &= \log(G_h/P_h)
\end{aligned}$$

作为一个标准的正则化最小二乘问题，这可以通过闭合形式有效解决。

我们在实现边界框回归时发现了两个微妙的问题。第一个问题是正则化很重要：我们在验证集的基础上设置了 $\lambda = 1000$。第二个问题是，在选择使用哪些训练对 (P, G) 时必须小心。直观地说，如果 P 远离所有的 ground-truth 框，那么将 P 转化为 ground-truth 框 G 的任务就没有意义

了。使用像 P 这样的例子会导致一个无望的学习问题。因此，仅当某个候选至少靠近一个 ground-truth 框时，我们才从这个候选 P 中学习。我们通过当且仅当 P 与与它有最大 IoU 重合 ground-truth 框 G（如果它重合了一个以上）的重合度大于阈值（我们使用验证集设定为 0.6）时, 才将 P 分配到与它有最大 IoU 重合的 ground-truth 框 G，来实现 “接近性”。所有未分配的候选都被丢弃。我们为每个对象类别做一次，以便学习一组特定类别的边界箱回归器。

在测试时，我们对每个提议进行评分，并只预测其新的检测窗口一次。原则上，我们可以迭代这个程序（即对新预测的边界盒重新打分，然后再从中预测一个新的边界盒，如此反复）。然而，我们发现，迭代并不能改善结果。

References

[1] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[2] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[3] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[4] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[5] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.

[6] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. “Adaptive deconvolutional networks for mid and high level feature learning”.

In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 2018–2025.

[7] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. "Diagnosing error in object detectors". In: *European conference on computer vision*. Springer. 2012, pp. 340–353.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[9] Pierre Sermanet et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[10] Yangqing Jia et al. "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 675–678.

[11] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).