

+ 3pt - 9pt + 3pt + 3pt - 9pt + 3pt - 4pt

# 丰富的特征层次，用于精确的目标检测和语义分割

## 技术报告(v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik  
UC Berkeley

{rgb, jdonahue, trevor, malik}@eecs.berkeley.edu

### Abstract

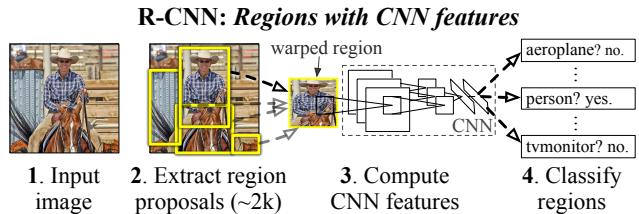
在标准PASCAL VOC数据集上测量的目标检测性能，在过去几年中已经稳定下来。表现最好的方法是复杂的集成系统，通常将多个低级图像特征与高级上下文相结合。本文提出了一种简单且可扩展的检测算法，相对于之前在VOC 2012上的最佳结果，平均精度均值(mAP)提高了30%以上——实现了53.3%的mAP。该方法结合了两个关键见解：(1)可以将大容量卷积神经网络(cnn)应用于自底向上的区域建议，以定位和分割对象；(2)当标记训练数据稀缺时，辅助任务的监督预训练，然后进行特定领域的微调，产生显著的性能提升。由于我们将区域建议框与CNN相结合，因此将该方法称为R-CNN：具有CNN特征的区域。本文还将R-CNN与OverFeat进行了比较，OverFeat是最近提出的一种基于类似CNN架构的滑动窗口检测器。在200类ILSVRC2013检测数据集上，R-CNN的表现大大超过了超值。完整系统的源代码可在此处获得<http://www.cs.berkeley.edu/~rgb/rcnn>。

### 1. 简介

特征很重要。在过去的十年中，各种视觉识别任务的进展都是基于SIFT [29]和HOG [7]的使用。但是，如果我们看看规范的视觉识别任务PASCAL VOC目标检测[15]的性能，一般认为在2010-2012年期间进展缓慢，通过构建集成系统和使用成功方法的小变体获得了小的增益。

SIFT和HOG是分块方向直方图，我们可以将这种表示大致与V1中的复杂细胞联系起来，V1是灵长类动物视觉通路中的第一个皮层区域。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特征，这些特征对视觉识别来说信息量更大。

福岛的“neocognitron”[19]，一个受生物启发的分层和移位不变的模式识别模型，就是这样一个过程的早期尝试。然而，neocognitron缺乏监督训练算法。基于Rumelhart et al. [33]，LeCun et al. [26]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(cnn)是有效



**Figure 1:** 目标检测系统概述。我们的系统(1)获取输入图像，(2)提取大约2000个自下而上的候选区域，(3)使用大型卷积神经网络(CNN)计算每个候选区域的特征，(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上的平均精度均值(mAP)达到53.7%。作为比较，[39]报告了35.1%的地图使用相同的区域建议，但使用空间金字塔和视觉词袋方法。流行的可变形部件模型的性能为33.4%。在200类的ILSVRC2013检测数据集上，R-CNN的mAP为31.4%，比OverFeat [34]有很大的提升，后者拥有之前最好的结果24.3%。

的，卷积神经网络是一类扩展了neocognitron的模型。

cnn在20世纪90年代被大量使用(e.g., [27])，但随后随着支持向量机的兴起而过时。2012年，Krizhevsky et al. [25]重新点燃了人们的兴趣在cnn中，通过在ImageNet大规模视觉识别挑战(ILSVRC) [9, 10]上显示出更高的图像分类精度。他们的成功源于在120万张标记图像上训练一个大型CNN，以及LeCun的CNN(e.g.,  $\max(x, 0)$ 纠正非线性和“dropout”正则化)上的一些改变。

在ILSVRC 2012研讨会上，对ImageNet结果的重要性进行了激烈的辩论。核心问题可以归结为以下几点：做什么在ImageNet上的CNN分类结果泛化程度PASCAL VOC挑战上的目标检测结果？

本文通过弥合图像分类和目标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hog特征的系统相比，CNN可以使PASCAL VOC上的目标检测性能显著提高。为了实现这一结果，我们关注了两个问题：用深度网络定位目标和只用少量标注检测数据训练高容量模型。

与图像分类不同，检测需要在图像中定位(可能有很多)目标。一种方法是将帧定位作为回归问题。然而，Szegedy et al. [38]的工作，与我们自己的

同时，表明此策略在实践中可能表现不太好(他们报告2007年VOC的mAP为30.5%，而我们的方法实现了58.5%)。另一种方法是构建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。我们还考虑过采用滑动窗口方法。然而，在我们的网络中，具有五个卷积层的高层单元，在输入图像中具有非常大的感受野( $195 \times 195$ 像素)和步幅( $32 \times 32$ 像素)，这使得在滑动窗口范式内的精确定位成为一个开放的技术挑战。

相反，我们通过在“使用区域识别”范式[21]内操作来解决CNN定位问题，这在目标检测[39]和语义分割[5]方面都取得了成功。在测试时，该方法为输入图像生成了约2000个类别独立的候选区域，使用CNN从每个候选区域中提取固定长度的特征向量，然后用特定类别的线性svm对每个区域进行分类。我们使用一种简单的技术(仿射图像变形)从每个区域建议中计算固定大小的CNN输入，而不管区域的形状。Figure 1提供了我们方法的概述并突出了我们的一些结果。由于该系统将区域建议框与CNN相结合，因此将该方法称为R-CNN：具有CNN特征的区域。

在本文的这个更新版本中，通过在200类ILSVRC2013检测数据集上运行R-CNN，对R-CNN和最近提出的OverFeat [34]检测系统进行了针锋相对的比较。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。R-CNN的表现明显优于OverFeat，mAP为31.4%，而不是24.3%。

检测面临的第二个挑战是标记数据稀缺，目前可用的数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，然后是监督的微调(e.g., [35])。本文的第二个主要贡献是表明，在大型辅助数据集(ILSVRC)上进行有监督的预训练，然后在小型数据集(PASCAL)上进行特定领域的微调，是在数据稀缺时学习高容量cnn的有效范式。在我们的实验中，对检测进行微调使mAP性能提高了8个百分点。经过微调，该系统在VOC 2010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) [17, 20]为33%。我们还向读者介绍了Donahue的同代工作et al. [12]，他表明Krizhevsky的CNN可以用作黑箱特征提取器(无需微调)，在几个识别任务上产生优异的性能，包括场景分类、细粒度子分类和域自适应。

我们的系统也非常高效。唯一的类相关计算是较小的矩阵向量乘积和贪婪的非极大值抑制。这种计算特性源于所有类别共享的特征，而且这些特征的维度比以前使用的区域特征低两个数量级(cf. [39])。

理解我们方法的故障模式对于改进它也至关重要，因此我们报告了来自Hoiem的检测分析工具et al.的结果[23]。这种分析的一个直接结果是，一个简单的边界框回归方法大大减少了错误定位，这是主要的错误模式。

在开发技术细节之前，我们注意到，由于R-CNN对



Figure 2: 来自VOC 2007训练的扭曲训练样本。

区域进行操作，因此很自然地将其扩展到语义分割任务。稍加修改，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进行目标检测

我们的目标检测系统由三个模块组成。第一个生成独立于类别的候选区域。这些建议定义了我们的检测器可用的候选检测集。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，描述它们的测试时使用情况，详细说明它们的参数是如何学习的，并展示在PASCAL VOC 2010-12和ILSVRC2013上的检测结果。

### 2.1. 模块设计

**区域建议。** 最近的各种论文提供了生成类别无关的方法区域建议。例如：objectness [1]，选择性搜索[39]，独立类别目标建议[14]，约束参数最小割(CPMC) [5]、多尺度组合分组[3]和Cire §和et al. [6]，他们通过将CNN应用于规则间隔的方形作物来检测有丝分裂细胞，这是区域建议的一个特殊情况。虽然R-CNN与特定的区域建议方法无关，但我们使用选择性搜索来实现与之前的检测工作的对照比较(e.g., [39, 41])。

**特征提取。** 我们使用Caffe [24] Krizhevsky描述的CNN实现et al.从每个区域建议中提取4096维特征向量[25]。特征通过正向传播减去均值的 $227 \times 227$ 来计算RGB图像通过5个卷积层和2个全连接层。我们参考读者[24, 25]了解更多的网络架构细节。

为了计算候选区域的特征，我们必须首先将该区域中的图像数据转换为兼容的形式对于CNN(其架构需要固定的输入 $227 \times 227$ 像素大小)。在任意形状区域的许多可能的变换中，我们选择了最简单的。无论候选区域的大小或长宽比如何，我们将其周围的紧密边界框中的所有像素扭曲为所需的大小。在扭曲之前，我们膨胀的紧密边界框，以便在扭曲的大小，有正好 $p$ 像素的扭曲图像上下文周围的原始框(我们使用 $p = 16$ )。Figure 2显示了扭曲训练区域的随机抽样。在Appendix A中讨论了变形的替代方案。

### 2.2. 测试时检测

在测试时，我们对测试图像进行选择性搜索，以提取大约2000个建议区域(在所有实验中我们使用选择性搜索的“快速模式”)。我们扭曲每个建议框，并

通过CNN向前传播，以计算特征。然后，对于每个类别，我们使用为该类训练的SVM对每个提取的特征向量进行评分。给定图像中所有得分的区域，我们应用贪婪的非最大抑制(独立于每个类)，如果一个区域与大于学习阈值的更高得分选择区域有交并(IoU)重叠，则拒绝该区域。

运行时分析。两个特性提高了检测效率。首先，所有CNN参数在所有类别中共享。第二，通过比较，CNN计算的特征向量是低维的到其他常见的方法，例如使用视觉词袋的空间金字塔编码。例如，在UVA检测系统[39]中使用的功能比我们的大两个数量级(360k vs. 4k维)。

这种共享的结果是计算region所花费的时间建议和功能(13s/图像在GPU或53s/图像在CPU上)摊销到所有类上。唯一的特定于类的计算是特征和的点积SVM权重与非极大值抑制。实际上，图像的所有点积都是批量处理的转化成一个矩阵-矩阵乘积。特征矩阵是典型的 $2000 \times 4096$ , SVM权重矩阵为 $4096 \times N$ , 其中 $N$ 为类别数。

分析表明R-CNN可以扩展到数千个对象类，而无需求助于近似技术，比如散列。即使有100 000个类，结果矩阵乘法在现代多核CPU上只需要10秒。这种效率不仅仅是结果使用区域建议和共享特征。UVA系统，由于其高维特征，将是两阶当需要时，幅度变慢相比之下，仅存储100k线性预测变量就需要134GB内存对于我们的低维特征，只有1.5GB。

将R-CNN与Dean最近的工作et al.进行对比也很有趣关于使用DPMs和散列的可扩展检测[8]。他们报告说，在每个图像运行5分钟时，VOC 2007的mAP约为16%介绍10k类干扰物时。用我们的方法，10k 检测器可以在CPU上运行大约一分钟，因为没有近似值，地图将保持在59%(Section 3.2)。

### 2.3. 培训

监督式预训练。我们仅使用图像级注释在一个大型辅助数据集(ILSVRC2012 classification)上对CNN进行了判别性预训练(此数据没有边界框标签)。预训练使用开源的Caffe CNN库[24]进行。简而言之，我们的CNN几乎可以媲美Krizhevsky的表现et al. [25]，错误率top-1在ILSVRC2012分类验证上提高了2.2个百分点集合。这种差异是由于训练过程中的简化。

特定领域的微调。为了使CNN适应新任务(检测)和新域(扭曲建议框窗口)，继续使用仅扭曲区域建议框对CNN参数进行随机梯度下降(SGD)训练。除了将CNN的imagenet特定的1000路分类层替换为随机初始化的 $(N + 1)$ 路分类层(其中 $N$ 是对象类的数量，加1为背景)，CNN架构没有改变。VOC为 $N = 20$ , ILSVRC2013为 $N = 200$ 。我们将 $\geq 0.5$  IoU与基本事实重叠的所有区域建议处理方框为该方框所属类别的正类，其余为负类。SGD的学习率为0.001(初始预训练速率的十分之一)，这样可以在不影响初始化的情况下进行微调。在每次SGD迭代中，我们均匀采样32个正

窗口(所有类别)和96个背景窗口，以构建大小为128的小批量。我们将采样偏向于正窗口，因为与背景相比，它们非常罕见。

对象类别分类器。考虑训练一个二分类器来检测汽车。很明显，紧紧包围汽车的图像区域应该是一个积极的例子。同样，很明显，与汽车无关的背景区域应该是一个负面例子。不太清楚的是如何标记与汽车部分重叠的区域。我们使用IoU重叠阈值来解决这个问题，低于该阈值的区域被定义为负数。重叠阈值0.3是在验证集 $\{0, 0.1, \dots, 0.5\}$ 上通过网格搜索选择的。我们发现仔细选择这个阈值很重要。将其设置为0.5，如[39]，将减少5点的mAP。类似地，将其设置为0会减少4点的mAP。正例被定义为每个类的真实边界框。

一旦提取特征并应用训练标签，我们就为每个类优化一个线性SVM。由于训练数据太大，内存无法容纳，我们采用标准的硬负挖掘方法[17, 37]。硬负挖掘收敛很快，在实践中，仅对所有图像进行一次遍历后，mAP就停止增长。

在Appendix B 中，我们讨论了为什么在微调和SVM训练中正负示例的定义不同。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的最终softmax层的输出。

### 2.4. PASCAL VOC 2010-12测试结果

根据PASCAL VOC最佳实践[15]，我们进行了验证VOC 2007数据集上的所有设计决策和超参数(Section 3.2)。对于VOC 2010-12数据集的最终结果，我们进行了微调VOC 2012上的CNN在VOC 2012上训练并优化了我们的检测svm。对于两种主要的算法变体(使用和不使用边界框回归)，我们只向评估服务器提交了一次测试结果。

Table 1 显示VOC 2010的完整结果。将所提出方法与四个强大的基线进行了比较，包括SegDPM [18]，它将DPM检测器与语义分割系统的输出[4]相结合，并使用额外的检测器间上下文和图像分类器重新评分。最相关的比较是从UVA系统Uijlings et al. [39]，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级分类空间金字塔，并用密集采样的SIFT进行填充，扩展的entsift和RGB-SIFT描述子，用4000字的码本对每个向量进行量化。分类是通过直方图交集进行的核SVM。与它们的多特征非线性核SVM进行比较实验结果表明，该方法在mAP上取得了较大的提升，从35.1%提升到53.7%的地图，同时也更快(Section 2.2)。该方法在VOC 2011/12测试中取得了类似的性能(mAP为53.3%)。

### 2.5. ILSVRC2013检测结果

我们使用与PASCAL VOC相同的系统超参数在200类ILSVRC2013检测数据集上运行R-CNN。我们遵循相同的协议，只向ILSVRC2013评估服务器提交两次测试结果，一次使用边界框回归，一次不使用边界框回归。

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] <sup>†</sup>	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] <sup>†</sup>	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	<b>71.8</b>	<b>65.8</b>	<b>53.0</b>	<b>36.8</b>	<b>35.9</b>	<b>59.7</b>	<b>60.0</b>	<b>69.9</b>	<b>27.9</b>	<b>50.6</b>	<b>41.4</b>	<b>70.0</b>	<b>62.0</b>	<b>69.0</b>	<b>58.1</b>	<b>29.5</b>	<b>59.4</b>	<b>39.3</b>	<b>61.2</b>	<b>52.4</b>	<b>53.7</b>

**Table 1: VOC 2010**测试的平均精度(%)。R-CNN与UVA和Regionlets最直接相似，因为所有方法都使用选择性搜索区域建议。边界框回归(Bounding-box regression, BB)的描述见Section C。在本文发表时，SegDPM是PASCAL VOC排行榜上表现最好的。<sup>†</sup> DPM和SegDPM使用其他方法不使用的上下文重新评分。

Figure 3 将R-CNN与ILSVRC 2013比赛中的参赛作品和比赛后的超成绩进行比较[34]。R-CNN取得了31.4%的mAP，明显领先于第二好的结果OverFeat的24.3%。为了直观地了解AP在类上的分布，还提供了箱线图，并在本文的末尾列出了每个类的AP表，见Table 8。大多数参赛作品(OverFeat、NECMU、UvA-Euvision、Toronto A和UIUC-IFP)都使用了卷积神经网络，这表明cnn如何应用于目标检测存在显著的细微差别，导致了不同的结果。

在Section 4中，我们给出了ILSVRC2013检测数据集的概述，并提供了关于我们在其上运行R-CNN时所做选择的详细信息。

### 3. 可视化、消融和误差模式

#### 3.1. 可视化学习到的特征

第一层滤波器可以直接可视化并且很容易理解[25]。他们捕获有方向的边缘和对立的颜色。理解后续的层更具有挑战性。泽勒和费格斯提出了一个视觉上有吸引力的反卷积请访问[42]。本文提出一种简单的(互补的)非参数方法，直接显示网络学习到的内容。

其想法是在网络中挑选出一个特定的单元(特征)，并将其作为一个对象来使用探测器本身。也就是说，我们计算单元在a上的激活值大量保留的区域提案(约1000万个)，对提案进行排序从最高激活到最低激活，执行非最大值抑制，然后显示得分最高的区域。我们的方法让选中的单位‘不言自明’，通过显示它确切地触发了哪些输入。我们避免平均，以看到不同的视觉模式和增益洞察单位计算的不变性。

我们可视化来自pool<sub>5</sub>层的单元，这是的最大池化输出网络的第五层也是最后一层卷积层。pool<sub>5</sub>特征图为 $6 \times 6 \times 256 = 9216$ 维度。忽略边界效应，每个pool<sub>5</sub>单元在原始 $227 \times 227$ 像素输入中具有 $195 \times 195$ 像素的感受野。一个中央pool<sub>5</sub>单元具有几乎全局的视图，而一个靠近边缘的单元具有较小的、剪切的支持。

Figure 4中的每一行显示了我们在VOC 2007 trainval上微调的CNN的pool<sub>5</sub>单元的前16个激活。256个功能独特的单元中有6个被可视化(Appendix D包含更多)。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，我们看到一个针对狗的脸和点数组的单元。第三行对应的单元是一个红色斑点检测器。还有用于人脸和更抽象模式(如文本和带窗口的三角形

结构)的检测器。该网络似乎学习了一种表示，将少量类别调整的特征与形状、纹理、颜色和材料属性的分布式表示相结合。随后的全连接层fc<sub>6</sub>能够对这些丰富特征的大量组合进行建模。

#### 3.2. 消融研究

逐层性能优化，无需微调。为了了解哪些层对检测性能至关重要，我们分析了CNN最后三层在VOC 2007数据集上的结果。层pool<sub>5</sub>在Section 3.1中有简要描述。下面总结了最后两层。

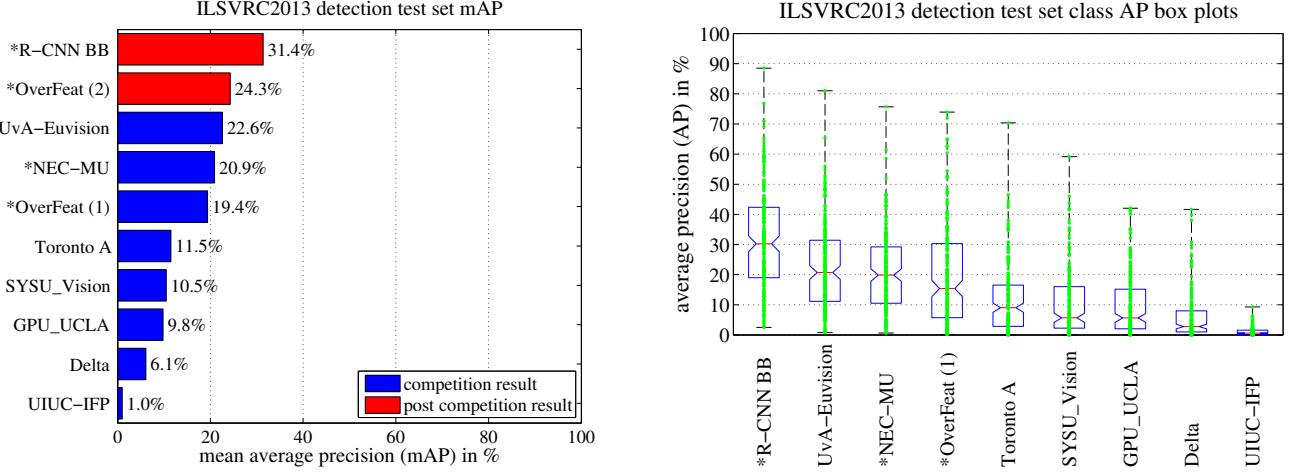
层fc<sub>6</sub>与pool<sub>5</sub>完全连接。为了计算特征，它将 $4096 \times 9216$ 权重矩阵乘以pool<sub>5</sub>特征映射(重塑为一个9216维向量)，然后添加一个向量偏见。这个中间向量是分量半波整流( $x \leftarrow \max(0, x)$ )。

fc<sub>7</sub>层是网络的最后一层。它被实现了通过将fc<sub>6</sub>计算出的特征乘以 $4096 \times 4096$ 权重矩阵，并类似地添加一个偏置和向量应用半波整流。

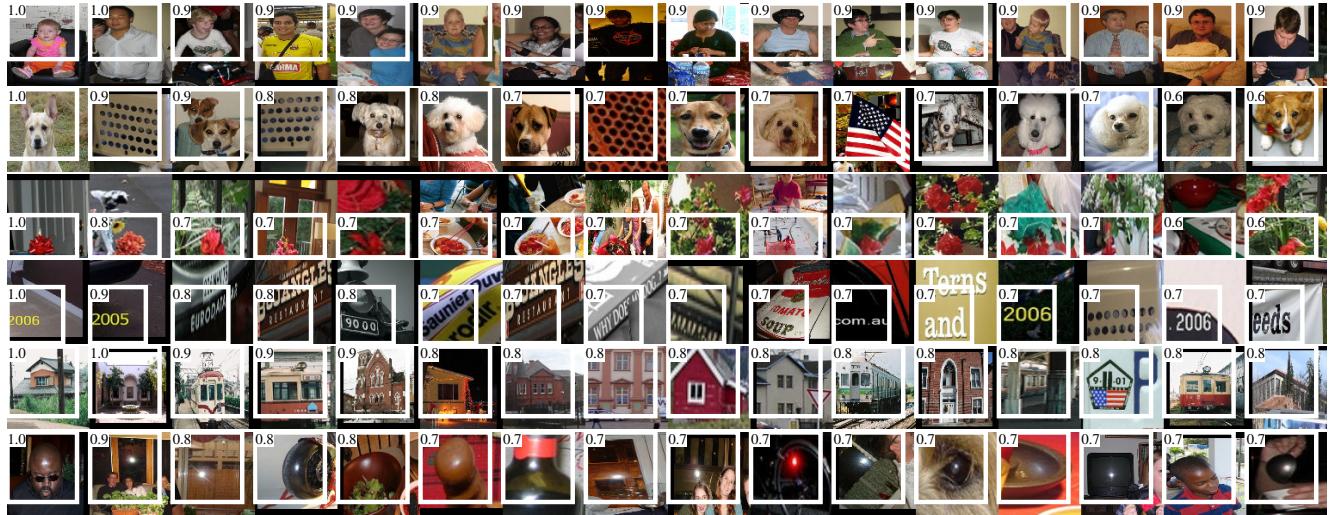
我们首先查看CNN在PASCAL上的结果无需微调，i.e.所有CNN参数仅在ILSVRC 2012上进行了预训练。逐层分析性能(Table 2行1-3)揭示了这一点来自fc<sub>7</sub>的功能概括起来比功能来自fc<sub>6</sub>。这意味着29%，约1680万，可以在不降低mAP的情况下删除CNN的参数。更令人惊讶的是，删除都有fc<sub>7</sub>和fc<sub>6</sub>产生相当好的结果，尽管pool<sub>5</sub>使用只有6%计算特征CNN的参数。大部分的CNN的表征能力来自于它的卷积层，而不是来自更大的紧密连接的层。这个发现表明计算密集特征图的潜在效用，在HOG的意义上，通过仅使用卷积层来实现任意大小的图像CNN的新闻。这种表示法将使滑动窗口检测器的实验成为可能，包括DPM，在pool<sub>5</sub>功能之上。

性能逐层进行微调。在对VOC 2007 trainval的参数进行微调后，我们现在查看CNN的结果。改进是惊人的(Table 2第4-6行)：微调将mAP提升8.0个百分点至54.2%。微调对fc<sub>6</sub>和fc<sub>7</sub>的提升要比pool<sub>5</sub>大得多，这表明，从ImageNet学习到的pool<sub>5</sub>特征是通用的，大多数改进是通过在其基础上学习特定领域的非线性分类器来获得的。

与最近的特征学习方法的比较。在PASCAL VOC检测上尝试的特征学习方法相对较少。本文介绍了两种建立在可变形部件模型上的最新方法。为了参考，我们还包括了标准的基于hog的DPM [20]的结果。



**Figure 3:** (左)ILSVRC2013检测测试集上的平均精度均值。\*之前的方法使用外部训练数据(所有情况下来自ILSVRC分类数据集的图像和标签)。(右)每种方法200个平均精度值的箱线图。由于尚未提供每类ap(R-CNN的每类ap在Table 8中,也包含在上传到arXiv.org的技术报告来源中;参见R-CNN-ILSVRC2013-APs.txt)。红线表示AP的中位数,盒子底部和顶部分别表示第25和第75百分位数。须延伸到每种方法的最小和最大AP。每个AP都被绘制为胡须上的一个绿色点(以缩放方式进行数字化查看效果最好)。



**Figure 4:** 六个最受欢迎的地区pool<sub>5</sub> 单元。感受野和激活值用白色表示。有些单元与概念对齐,例如人(第1行)或文本(第4行)。其他单元用于捕获纹理和材质属性,如点阵(2)和镜面反射(6)。

第一种DPM特征学习方法DPM ST [28], 使用“sketch”的直方图增强HOG特征象征性的“概率”。直观地说, sketch token是通过中心的轮廓线的紧密分布一个图像补丁。草图标记的概率计算在每个像素由一个随机森林训练分类 $35 \times 35$ 像素补丁到150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HOG。计算HSC, 稀疏代码使用学习到的字典解决每个像素的激活 $100 \times 7 \times 7$ 像素(灰度)原子。结果是通过三种方式进行校正(全波和半波), 空间汇集, 单元 $\ell_2$ 归一化, 然后权力转换( $x \leftarrow \text{sign}(x)|x|^\alpha$ )。

所有R-CNN变体都明显优于三个DPM基

线(Table 2行8-10), 包括两个使用特征学习。与最新版本的DPM, 仅使用HOG特征, 我们的mAP提高了20多个百分点:54.2%vs. 33.7%——相对提升了61%。HOG和sketch token的结合比HOG单独产生2.5个mAP点, 而HSC比HOG提高4个地图点(当内部比较时到他们的私有DPM基线——都使用非公开的DPM实现表现不如开源版本[20])。这些方法的map分别达到29.1%和34.3%。

### 3.3. 网络体系结构

本文的大部分结果使用了Krizhevsky的网络架构et al. [25]。然而, 我们发现架构的选择对R-CNN的

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 <sub>5</sub>	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 <sub>6</sub>	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 <sub>7</sub>	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 <sub>5</sub>	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 <sub>6</sub>	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 <sub>7</sub>	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 <sub>7</sub> BB	<b>68.1</b>	<b>72.8</b>	<b>56.8</b>	<b>43.0</b>	<b>36.8</b>	<b>66.3</b>	<b>74.2</b>	<b>67.6</b>	<b>34.4</b>	<b>63.5</b>	<b>54.5</b>	<b>61.2</b>	<b>69.1</b>	<b>68.6</b>	<b>58.7</b>	<b>33.4</b>	<b>62.9</b>	<b>51.1</b>	<b>62.5</b>	<b>64.8</b>	<b>58.5</b>
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

**Table 2:** VOC 2007测试的平均精度(%)。第1-3行显示不进行微调的R-CNN性能。第4-6行显示了在ILSVRC 2012上预训练的CNN的结果然后在VOC 2007训练集上进行微调(FT)。第7行包括一个简单的边界框回归(BB)阶段，可以减少定位错误(Section C)。第8-10行将DPM方法作为一个强大的基线。第一个只使用HOG，而接下来的两个使用不同的特征学习方法来增强或替换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	<b>73.4</b>	<b>77.0</b>	<b>63.4</b>	<b>45.4</b>	<b>44.6</b>	<b>75.1</b>	<b>78.1</b>	<b>79.8</b>	<b>40.5</b>	<b>73.7</b>	<b>62.2</b>	<b>79.4</b>	<b>78.1</b>	<b>73.1</b>	<b>64.2</b>	<b>35.6</b>	<b>66.8</b>	<b>67.2</b>	<b>70.4</b>	<b>71.1</b>	<b>66.0</b>

**Table 3:** 在VOC 2007上对两种不同的CNN架构进行检测平均精度(%)。前两行是使用Krizhevsky et al.的架构(T-Net)从Table 2得到的结果。第三行和第四行使用了Simonyan和Zisserman (O-Net) [43]最近提出的16层架构。

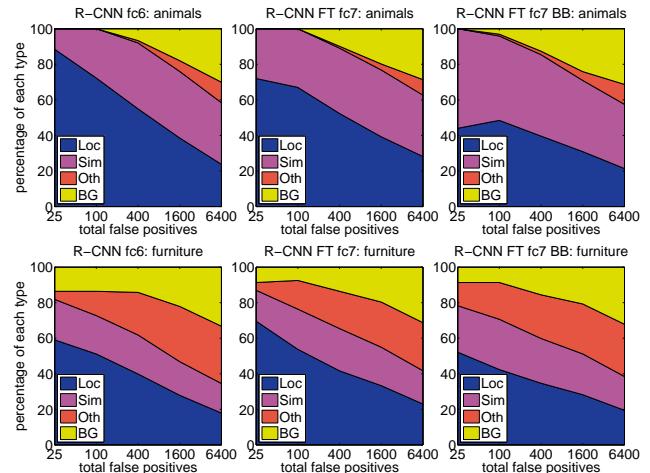
检测性能有很大的影响。在Table 3中，我们展示了使用Simonyan和Zisserman最近提出的16层深度网络进行VOC 2007测试的结果[43]。该网络是最近的ILSVRC 2014分类挑战中表现最好的网络之一。该网络具有由13层 $3 \times 3$ 卷积核组成的同质结构，其中穿插了五个最大池化层，顶部是三个全连接层。我们称这个网络为‘O-Net’，代表牛津网，基线为‘T-Net’，代表多伦多网。

为了在R-CNN中使用O-Net，我们从Caffe模型库中下载了VGG\_ILSVRC\_16\_layers模型的公开可用的预训练网络权重。<sup>1</sup>然后，我们使用与T-Net相同的协议对网络进行微调。唯一的区别是根据需要使用较小的minibatch(24个示例)，以便适合GPU内存。在Table 3上的结果表明，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的前向传递时间大约是T-Net的7倍。

### 3.4. 检测误差分析

我们应用了来自Hoiem et al.的优秀检测分析工具[23]为了揭示我们的方法的错误模式，了解微调如何改变它们，以及看看我们的错误类型与DPM的比较。一个完整的总结分析工具超出了本文的范围，我们鼓励读者咨询[23]理解一些更详细的细节(例如‘规范化AP’)。自从我们认为，分析最好是在相关情节的背景下进行在Figure 5的标题中展示讨论和Figure 6。

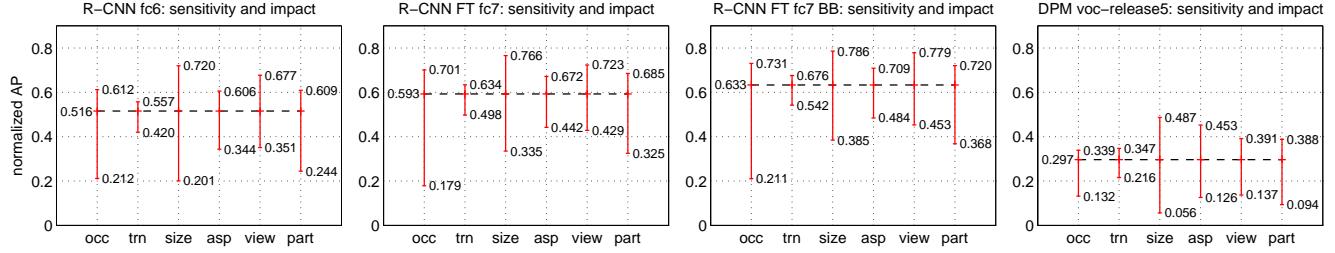
<sup>1</sup><https://github.com/BVLC/caffe/wiki/Model-Zoo>



**Figure 5: top-ranking**假阳性(FP)分布类型。每张图显示了FP类型的演变分布按照分数递减的顺序考虑更多FPs。每个FP都是分为4种类型中的1种:Loc—较差的定位(带有IoU重叠的检测将正确的类别设置在0.1到0.5之间，即重复);模拟—混淆类似的类别; Oth—与不相似的对象类别混淆;BG—一个被解雇的FP背景介绍。与DPM相比(参见[23])，我们的错误明显更多是由于本地化不好，而不是与背景混淆或其他对象类，表明CNN特征很多比HOG更具辨别性。松散的本地化可能源于我们的使用自下而上的区域建议和位置不变性通过预训练CNN学到的全图像分类。第三列展示了我们简单的边界框回归方法如何修复许多定位错误。

### 3.5. 边界框回归

在误差分析的基础上，实现了一种降低定位误差



**Figure 6:** 对物体特征的敏感性。每张图显示了平均值(超过类别)归一化AP(见[23])在六个不同的子集中获得最高和最低的表现目标特征(遮挡, 截断, 边界框区域, 高宽比, 视点, 部分可见性)。展示了所提出方法(R-CNN)在有和没有微调(FT)和边界框回归(BB)以及DPM voc-release5的情况下的图。总的来说, 微调不会降低灵敏度(max和min之间的差异), 但会显著提高灵敏度几乎所有特征的最高和最低表现的子集。这表明微调可以不仅仅是简单地改进性能最低的高宽比和边界框区域子集基于我们如何扭曲网络输入的猜想。相反, 微调提高了鲁棒性所有特征, 包括遮挡、截断、视点和部分可见性。

的简单方法。受DPM [17]中采用的边界框回归的启发, 训练了一个线性回归模型, 以预测一个新的检测窗口, 给定pool<sub>5</sub>特征的选择性搜索区域建议。详情请见Appendix C。Table 1、Table 2和Figure 5的结果表明, 这种简单的方法修复了大量的错误定位检测, 将mAP提升了3到4个点。

### 3.6. 定性结果

本文最后给出了ILSVRC2013的定性检测结果Figure 8和Figure 9。每个图像都是从val<sub>2</sub>集合中随机采样的, 显示了所有精度大于0.5的检测器的所有检测结果。请注意, 这些不是精心策划的, 并给人一种实际运行探测器的印象。更多的定性结果在Figure 10和Figure 11中呈现, 但这些都是经过策划的。我们选择每一张图片, 因为它们包含有趣的、令人惊讶的或有趣的结果。这里也显示了精度大于0.5的所有检测结果。

## 4. ILSVRC2013检测数据集

在Section 2中, 我们展示了ILSVRC2013检测数据集的结果。这个数据集不像PASCAL VOC那么同质, 需要选择如何使用它。由于这些决策非常重要, 我们将在本节中介绍它们。

### 4.1. 数据集概览

ILSVRC2013检测数据集分为三个集合:train (395,918)、val (20,121)和test (40,152), 其中每个集合中的图像数量在括号中。val和test分割来自相同的图像分布。这些图像像场景一样, 在复杂性(对象数量、杂波数量、姿态可变性, etc.)上与PASCAL VOC图像相似。val和test分割进行了详尽的注释, 这意味着在每个图像中, 来自所有200个类别的所有实例都用边界框标记。相比之下, train集合来自ILSVRC2013分类图像分布。这些图像具有更多的可变复杂性, 并且倾向于单个中心物体的图像。与val和test不同, train图像(由于数量众多)没有进行详尽的注释。在任何给定的train图像中, 来自200个类别的实例可能被标记, 也可能不被标记。除了这些图像集之外, 每个类别都有一个额外的负图像集。负面图像需要手动检查, 以确认

它们不包含相关类的任何实例。在这项工作中没有使用负图像集。有关如何收集和注释ILSVRC的更多信息, 请参见[11, 36]。

这些划分的性质为训练R-CNN提供了许多选择。train图像不能用于困难的负挖掘, 因为注释不是详尽的。负例应该从哪里来? 此外, train图像的统计数据与val和test不同。应该使用train图像吗? 如果可以, 使用到什么程度? 虽然我们没有彻底评估大量的选择, 但根据以往的经验, 我们提出了看似最明显的路径。

我们的一般策略是严重依赖val集, 并使用一些train图像作为积极示例的辅助来源。为了使用val进行训练和验证, 我们将其划分为大致相同大小的“val<sub>1</sub>”和“val<sub>2</sub>”集。由于有些类在val中只有很少的示例(最小的类只有31个, 一半的类小于110), 因此生成近似类平衡的分区很重要。为此, 生成大量候选划分, 并选择具有最小的最大相对类不平衡的划分。<sup>2</sup>每个候选分割都是通过将val图像的类计数作为特征进行聚类生成的, 其次是可以改善分割平衡的随机局部搜索。这里使用的特殊分割的最大相对不平衡度约为11%, 中位相对不平衡度为4%。val<sub>1</sub>/ val<sub>2</sub>分割和用于产生它们的代码将公开, 以允许其他研究人员在本报告中使用的val分割上比较他们的方法。

### 4.2. 区域建议

我们采用了与PASCAL检测相同的区域建议方法。选择性搜索[39]在“快速模式”运行在val<sub>1</sub>, val<sub>2</sub>和test的每个图像(但不是在train的图像)。一个小的修改是为了处理选择性搜索不是尺度不变的这一事实, 因此产生的区域数量取决于图像分辨率。ILSVRC图像的大小从非常小到几个几百万像素不等, 因此我们在运行选择性搜索之前将每个图像调整为固定宽度(500像素)。在val上, 选择性搜索导致每个图像平均有2403个建议区域, 所有地面真实边界框的召回率为91.6%(在0.5 IoU阈值下)。这一召回率明显低于PASCAL的约98%, 表明在区域建议阶段还有很大的改进空间。

<sup>2</sup>相对不平衡用 $|a - b|/(a + b)$ 来衡量, 其中a和b是分割的每一半的类计数。

### 4.3. 训练数据

对于训练数据，我们形成了一组图像和方框，其中包括来自 $\text{val}_1$ 的所有选择性搜索和ground-truth方框，以及来自 $\text{train}$ 的每个类最多 $N$  ground-truth方框(如果一个类在 $\text{train}$ 中少于 $N$  ground-truth方框，则我们获取所有它们)。我们将这个图像和盒子的数据集称为 $\text{val}_1 + \text{train}_N$ 。在消融研究中，我们在 $\text{val}_2$ 上为 $N \in \{0, 500, 1000\}$  (Section 4.5)显示了mAP。

R-CNN中需要三个过程的训练数据:(1)CNN微调，(2)检测器SVM训练，和(3)边界框回归器训练。CNN微调在 $\text{val}_1 + \text{train}_N$ 上进行50k SGD迭代，使用与PASCAL完全相同的设置。使用Caffe对单个NVIDIA Tesla K20进行微调需要13个小时。对于SVM训练，来自 $\text{val}_1 + \text{train}_N$ 的所有基础事实框都被用作各自类别的正例。对从 $\text{val}_1$ 中随机选择的5000张图像的子集进行硬负挖掘。最初的实验表明，与5000个图像子集(大约一半)相比，从 $\text{val}_1$ 中挖掘所有的负数，只导致mAP下降了0.5个百分点，同时将SVM训练时间减少了一半。没有从 $\text{train}$ 中摘取负面示例，因为注释并不详尽。没有使用额外的经过验证的负面图像集。边界框回归器在 $\text{val}_1$ 上进行训练。

### 4.4. 验证与评估

在将结果提交给评估服务器之前，我们使用上述训练数据验证了数据使用选择以及微调和边界框回归对 $\text{val}_2$ 集的影响。所有系统超参数(e.g., SVM C超参数，区域弯曲中使用的填充，NMS阈值，边界框回归超参数)都固定在PASCAL的相同值。毫无疑问，其中一些超参数的选择对于ILSVRC来说略有次优，然而这项工作的目标是在不进行大量数据集调优的情况下在ILSVRC上产生初步的R-CNN结果。在 $\text{val}_2$ 上选择最佳方案后，我们将两个结果文件提交到ILSVRC2013评估服务器。第一篇文章没有使用边界框回归，第二篇文章使用了边界框回归。对于这些提交，我们扩展了SVM和bounding-box回归器训练集，分别使用 $\text{val} + \text{train}_{1k}$ 和 $\text{val}$ 。我们使用了在 $\text{val}_1 + \text{train}_{1k}$ 上进行微调的CNN，以避免重新运行微调和特征计算。

### 4.5. 消融实验

Table 4 对不同数量的训练数据、微调和边界框回归的影响进行了消融研究。第一个发现是 $\text{val}_2$ 上的地图与 $\text{test}$ 上的地图非常接近。这让我们相信 $\text{val}_2$ 上的mAP是测试集性能的一个很好的指示器。第一个结果是20.9%，这是R-CNN使用在ILSVRC2012分类数据集上预训练的CNN实现的(没有微调)，并允许访问 $\text{val}_1$ 中的少量训练数据( $\text{val}_1$ 中的一半类有15到55个示例)。将训练集扩展到 $\text{val}_1 + \text{train}_N$ 将性能提高到24.1%， $N = 500$ 和 $N = 1000$ 之间基本上没有区别。使用来自 $\text{val}_1$ 的示例对CNN进行微调，使其达到26.5%，但由于正训练示例数量较少，可能存在严重的过拟合。将微调集扩展到 $\text{val}_1 + \text{train}_{1k}$ ，从训练集中每个类别增加1000个正例，可以显著帮助，将mAP提高到29.7%。边界框回归将结果提升

到31.0%，与PASCAL相比，这是一个较小的相对提升。

### 4.6. 与过度的关系

R-CNN和OverFeat之间存在着一种有趣的关系：OverFeat可以(粗略地)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正方形区域的多尺度金字塔，并将每个类的边界框回归器更改为单个边界框回归器，则系统将非常相似(对它们在训练方式方面的一些潜在显著差异取模：CNN检测微调，使用svm等)。值得注意的是，OverFeat比R-CNN有显著的速度优势：根据从[34]引用的每张图像2秒的数字，它大约快9倍。这种速度来自OverFeat的滑动窗口(i.e.，区域建议)在图像级别上不变形，因此可以很容易地在重叠窗口之间共享计算。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。加速R-CNN应该可以通过各种方式实现，这仍然是未来的工作。

## 5. 语义分割

区域分类是语义分割的标准技术，使我们能够轻松地将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(“二阶池化”称为O<sub>2</sub>P) [4]进行直接比较，我们在他们的开源框架内工作。O<sub>2</sub>P使用CPMC为每张图像生成150个建议区域，然后使用支持向量回归(SVR)预测每个类的每个区域的质量。他们的方法的高性能归功于CPMC区域的质量和多种特征类型(丰富的SIFT和LBP变体)的强大的二阶池化。我们还注意到，Farabet et al. [16]最近在几个密集场景标记数据集(不包括PASCAL)上展示了良好的结果，使用CNN作为多尺度的每像素分类器。

我们关注[2, 4]并扩展PASCAL分割训练集，以包括Hariharan提供的额外注释et al. [22]。设计决策和超参数在VOC 2011验证集上进行交叉验证。最终测试结果只评估一次。

用于分割的CNN特征。我们评估了在CPMC区域上计算特征的三种策略，所有这些策略都是从将区域周围的矩形窗口弯曲到 $227 \times 227$ 开始的。第一种策略(*full*)忽略了区域的形状，并直接在弯曲窗口上计算CNN特征，就像我们在检测时所做的那样。然而，这些特征忽略了区域的非矩形形状。两个区域可能具有非常相似的边界框，但重叠很少。因此，第二种策略(*fg*)仅在一个区域的前景掩码上计算CNN特征。我们用均值输入替换背景，使得背景区域减去均值后为零。第三种策略(*full+fg*)只是简单地连接*full*和*fg*特性；实验验证了它们的互补性。

**VOC 2011**测试结果。Table 5 显示了我们在VOC 2011验证集上的结果摘要，并与O<sub>2</sub>P进行了比较。(参见Appendix E查看每个类别的完整结果。)在每个特征计算策略中，层fc<sub>6</sub>总是优于fc<sub>7</sub>，下面的讨论是指fc<sub>6</sub>特征。*fg*策略略优于全策略，表明掩码区域形状提供了

test set	val <sub>2</sub>	val <sub>2</sub>	val <sub>2</sub>	val <sub>2</sub>	val <sub>2</sub>	val <sub>2</sub>	test	test
<b>SVM training set</b>	val <sub>1</sub>	val <sub>1</sub> +train <sub>.5k</sub>	val <sub>1</sub> +train <sub>1k</sub>	val+train <sub>1k</sub>	val+train <sub>1k</sub>			
<b>CNN fine-tuning set</b>	n/a	n/a	n/a	val <sub>1</sub>	val <sub>1</sub> +train <sub>1k</sub>			
<b>bbox reg set</b>	n/a	n/a	n/a	n/a	n/a	val <sub>1</sub>	n/a	val
<b>CNN feature layer</b>	fc <sub>6</sub>	fc <sub>6</sub>	fc <sub>6</sub>	fc <sub>7</sub>				
<b>mAP</b>	20.9	24.1	24.1	26.5	29.7	<b>31.0</b>	30.2	<b>31.4</b>
<b>median AP</b>	17.7	21.0	21.4	24.8	29.2	<b>29.6</b>	29.0	<b>30.3</b>

Table 4: ILSVRC2013数据使用选择、微调和边界框回归的消融研究。

	full R-CNN	fg R-CNN	full+fg R-CNN
O <sub>2</sub> P [4]	fc <sub>6</sub>	fc <sub>7</sub>	fc <sub>6</sub>
	46.4	43.0	42.5

Table 5: VOC 2011验证上的分割平均精度(%)。第一栏介绍O<sub>2</sub>P; 2-7在ILSVRC 2012上使用我们的CNN预训练。

更强的信号，与我们的直觉相匹配。然而，full+fg达到了47.9%的平均准确率，我们的最佳结果为4.2%(也略微优于O<sub>2</sub>P)，这表明即使给定fg特征，全功能提供的上下文也是信息量很大的。值得注意的是，在我们的full+fg功能上训练20个svr需要在单个核心上花费一个小时，而在O<sub>2</sub>P功能上训练10多个小时。

在Table 6中，我们展示了VOC 2011测试集的结果，将我们表现最好的方法fc<sub>6</sub>(full+fg)与两个强大的基线进行了比较。我们的方法在21个类别中的11个达到了最高的分割精度，在各个类别中平均达到了47.9%的最高整体分割精度(但在任何合理的误差范围内可能与O<sub>2</sub>P结果一致)。通过微调可能会获得更好的性能。

## 6. 结论

近年来，目标检测性能停滞不前。表现最好的系统是将多个低级图像特征与目标检测器和场景分类器的高级上下文相结合的复杂集成。本文提出了一种简单且可扩展的目标检测算法，与PASCAL VOC 2012上之前的最佳结果相比，有30%的相对改进。

我们通过两种见解实现了这一性能。第一种是将大容量卷积神经网络应用于自底向上的区域建议，以定位和分割目标。第二种是在标记训练数据稀缺时训练大型cnn的范式。为具有丰富数据的辅助任务(图像分类)预训练网络——有监督的，然后为数据稀缺的目标任务(检测)微调网络，是非常有效的。本文推测，“有监督的预训练/特定领域的微调”范式对各种数据稀缺的视觉问题将非常有效。

最后指出，通过结合使用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)来实现这些结果是有意义的。两者不是科学探究的对立路线，而是自然而必然的伙伴关系。

致谢 这项研究得到了DARPA Mind’s Eye和MSEE的部分支持项目，由美国国家科学基金会颁发的IIS-0905647, IIS-1134072, 和IIS-1212798, MURI N000014-10-1-0933，以及丰田的支持。这项研究中使用

的gpu由NVIDIA公司慷慨捐赠。

## 附录

### A. 对象建议变换

这项工作中使用的卷积神经网络需要 $227 \times 227$ 像素的固定大小的输入。在检测方面，我们考虑任意图像矩形作为目标建议。我们评估了两种将目标建议转换为有效CNN输入的方法。

第一种方法(‘最紧的上下文正方形’)将每个对象建议框包围在最紧的正方形中，然后将该正方形中包含的图像按(各向同性)缩放到CNN输入大小。Figure 7列(B)显示了这种转换。此方法的变体(‘tightest square without context’)排除了围绕原始对象建议框的图像内容。Figure 7列(C)显示了这种转换。第二种方法(‘warp’)向各向异性地将每个对象建议框扩展到CNN输入大小。Figure 7列(D)显示warp转换。

对于每个变换，我们还考虑在原始目标建议框周围添加额外的图像上下文。上下文内边距的数量( $p$ )被定义为转换后的输入坐标框架中围绕原始对象建议框的边框大小。Figure 7在每个示例的顶部行显示 $p = 0$ 像素，在底部行显示 $p = 16$ 像素。在所有方法中，如果源矩形扩展到图像之外，则将缺失的数据替换为图像的均值(然后在将图像输入CNN之前减去均值)。一组实验表明，使用上下文填充( $p = 16$ 像素)的变形比其他方法的性能高出很多(3-5个mAP点)。显然还有更多的选择，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留给未来的工作。

### B. 正vs.负例和softmax

有两个设计选择值得进一步讨论。第一个是：为什么微调CNN和训练目标检测svm的正例和负例的定义不同？为了进行微调，我们将每个对象建议映射到具有最大IoU重叠(如果有)的基础真实例，如果IoU至少为0.5，则将其标记为匹配的基础真类的正例。所有其他提案都被标记为“背景”(i.e., 所有类别的负面示例)。相比之下，对于训练svm，我们只将基础真值框作为各自类别的正例，而与某个类别的所有实例重叠小于0.3 IoU的标签建议方案作为该类别的负例。落入灰色地带(超过0.3 IoU重叠，但不是基本事实)的建议会

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	<b>36.1</b>	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O <sub>2</sub> P [4]	<b>85.4</b>	<b>69.7</b>	22.3	45.2	<b>44.4</b>	46.9	66.7	57.8	56.2	<b>13.5</b>	<b>46.1</b>	32.3	41.2	<b>59.1</b>	55.3	51.0	<b>36.2</b>	50.4	<b>27.8</b>	46.9	<b>44.6</b>	47.6
ours (full+fg R-CNN fc <sub>6</sub> )	84.2	66.9	<b>23.7</b>	<b>58.3</b>	37.4	<b>55.4</b>	<b>73.3</b>	<b>58.7</b>	<b>56.5</b>	9.7	45.5	29.5	<b>49.3</b>	40.1	<b>57.8</b>	<b>53.9</b>	33.8	<b>60.7</b>	22.7	<b>47.1</b>	41.3	<b>47.9</b>

**Table 6: VOC 2011**上的分割精度(%)。我们比较了两个强大的基线:[2]的“区域和部分”(R&P)方法和[4]的二阶池化(O<sub>2</sub>P)方法。在没有任何微调的情况下，我们的CNN实现了顶级的分割性能，超过了R&P和粗略匹配O<sub>2</sub>P。



**Figure 7:** 不同的对象建议变换。(A)相对于变换后的CNN输入的实际尺度的原始目标建议;(B)最紧的具有上下文的正方形;(C)最紧的没有上下文的正方形;(D)扭曲。在每个列和示例建议中，顶部行对应于 $p = 0$ 像素的上下文填充，而底部行对应于 $p = 16$ 像素的上下文填充。

被忽略。

从历史上讲，我们得到这些定义是因为我们一开始是在ImageNet预训练CNN计算的特征上训练svm，所以当时没有考虑微调。在这种设置中，我们发现用于训练svm的特定标签定义在我们评估的一组选项中是最佳的(其中包括我们现在用于微调的设置)。当我们开始使用微调时，我们最初使用与SVM训练相同的正例和负例定义。然而，我们发现结果比使用我们目前的正负定义得到的结果要差得多。

我们的假设是，这种正负定义的差异从根本上来说并不重要，原因是微调数据是有限的。我们当前的方案引入了许多“抖动”的示例(那些在0.5和1之间重叠的建议，但不是真实值)，这将正示例的数量扩展了约30倍。我们猜测，在微调整个网络时需要这个大集合，以避免过度拟合。然而，我们也注意到，使用这些抖动的示例可能是次优的，因为网络没有进行微调以进行精确定位。

这就引出了第二个问题:为什么在微调之后还要训练svm呢？简单地应用微调网络的最后一层(即21路softmax回归分类器)作为目标检测器将更清晰。我们尝试了一下，发现VOC 2007上的性能mAP从54.2%下降到了50.9%。这种性能下降可能是由几个因素的组合引起的，包括微调中使用的正例定义不强调精确定位，并且softmax分类器是在随机采样的负例上进行训练的，而不是用于SVM训练的“硬负例”子集。

这一结果表明，在微调后无需训练svm就可以获得接近相同水平的性能。我们推测，通过一些额外的微

调，剩余的性能差距可能会缩小。如果为真，这将简化和加快R-CNN训练，而不会损失检测性能。

## 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。在用特定类的检测SVM对每个选择性搜索建议框进行评分后，我们使用特定类的边界框回归器预测用于检测的新边界框。这在精神上类似于可变形部件模型中使用的边界框回归[17]。两种方法的主要区别在于，这里我们从CNN计算的特征中回归，而不是从推断的DPM部件位置上计算的几何特征中回归。

我们训练算法的输入是一组 $N$ 训练对 $\{(P^i, G^i)\}_{i=1,\dots,N}$ ，其中 $P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$ 指定建议框 $P^i$ 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)。因此，除非有必要，否则我们将去掉上标 $i$ 。每个真实值边界框 $G$ 以相同的方式指定: $G = (G_x, G_y, G_w, G_h)$ 。我们的目标是学习将建议框 $P$ 映射到真实框 $G$ 的转换。

我们使用四个函数对转换进行参数化 $d_x(P)$ 、 $d_y(P)$ 、 $d_w(P)$ 和 $d_h(P)$ 。前两个指定 $P$ 边界框中心的尺度不变转换，而后两个指定 $P$ 边界框的宽度和高度的对数空间转换。在学习这些函数之后，我们可以通过应用转换将输入建议框 $P$ 转换为预测的ground-truth框 $\hat{G}$

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

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

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

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

每个函数 $d_*(P)$ (其中 $*$ 是 $x, y, h, w$ 中的一个)被建模为提案 $P$ 的pool<sub>5</sub>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 $d_*(P) = \mathbf{w}_*^\top \phi_5(P)$ ，其中 $\mathbf{w}_*$ 是可学习的模型参数的向量。我们通过优化正则化最小二乘目标(岭回归)来学习 $\mathbf{w}_*$ :

$$\mathbf{w}_* = \underset{\hat{\mathbf{w}}_*}{\operatorname{argmin}} \sum_i^N (t_*^i - \hat{\mathbf{w}}_*^\top \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_*\|^2. \quad (5)$$

训练对 $(P, G)$ 的回归目标 $t_\star$ 定义为

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

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

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

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

作为一个标准的正则化最小二乘问题，它可以以闭合形式有效地求解。

在实现边界框回归时，我们发现了两个微妙的问题。首先，正则化很重要：我们基于验证集设置 $\lambda = 1000$ 。第二个问题是，在选择使用哪些训练对 $(P, G)$ 时必须谨慎。直观地说，如果 $P$ 远离所有的基础事实框，那么将 $P$ 转换为基础事实框 $G$ 的任务就没有意义。使用 $P$ 这样的例子会导致无望的学习问题。因此，我们只有在提案 $P$ 附近至少有一个ground-truth框时才会从该提案中学习。我们通过将 $P$ 分配给ground-truth框 $G$ 来实现“接近”，当且仅当重叠大于阈值（我们使用验证集将其设置为0.6）时，它具有最大的IoU重叠（在情况下，它重叠超过一个）。所有未分配的提案都会被丢弃。我们对每个对象类做一次，以便学习一组特定于类的边界框回归器。

在测试时，我们对每个候选框进行评分，并只预测其新的检测窗口一次。原则上，我们可以迭代这个过程（i.e.，重新对新预测的边界框进行评分，然后从中预测一个新的边界框，以此类推）。然而，我们发现迭代并不能改善结果。

## D. 额外的功能可视化

Figure 12 显示 $20 \text{ pool}_5$ 单位的额外可视化。对于每个单元，展示了24个区域建议，这些建议在VOC 2007测试的全部约1000万个区域中最大限度地激活了该单元。

我们通过其在 $6 \times 6 \times 256$ 维度 $\text{pool}_5$ 特征图中的(y, x, 通道)位置标记每个单元。在每个通道内，CNN计算输入区域的完全相同的函数，(y, x)位置只改变感受野。

## E. 每类别分割结果

在Table 7中，我们展示了我们的六种分割方法以及O<sub>2</sub>P方法[4]在VOC 2011 val上的每个类别的分割精度。这些结果显示了20个PASCAL类中的最强方法，以及背景类。

## F. 跨数据集冗余分析

在使用辅助数据集进行训练时，一个问题是它与测试集之间可能存在冗余。尽管目标检测和全图像分类的任务有本质上的不同，使得这种跨集冗余不那么令人担忧，但我们仍然进行了彻底的调查，量化了PASCAL测试图像在ILSVRC 2012训练和验证集中的包含程度。我们的发现可能对有兴趣使用ILSVRC

2012作为PASCAL图像分类任务训练数据的研究人员有用。

我们对重复（和近似重复）的图像进行了两次检查。第一个测试是基于flickr图像id的精确匹配，这些id包含在VOC 2007测试注释中（这些id有意为后续的PASCAL测试集保密）。所有PASCAL图像和大约一半的ILSVRC图像都是从flickr.com上收集的。在4952个匹配项中有31个匹配项（0.63%）。

第二项检查使用GIST [30]描述符匹配，在[13]中可以看到，它在大型（> 100万）图像集的近重复图像检测中具有出色的性能。在[13]之后，我们在所有ILSVRC 2012 trainval和PASCAL 2007测试图像的扭曲 $32 \times 32$ 像素版本上计算GIST描述符。

对GIST描述子的欧氏距离最近邻匹配发现了38张近似重复的图像（包括flickr ID匹配发现的全部31张）。匹配往往在JPEG压缩级别和分辨率上略有不同，裁剪程度较小。这些发现表明，重叠度很小，小于1%。对于VOC 2012，因为没有flickr id，所以我们只使用GIST匹配方法。基于GIST匹配，1.5%的VOC 2012测试图像在ILSVRC 2012训练中。VOC 2012的比率略高，可能是因为这两个数据集的收集时间比VOC 2007和ILSVRC 2012更接近。

## G. 文档变更日志

本文跟踪R-CNN的进展。为了帮助读者了解它是如何随着时间的推移而变化的，这里有一个简要的更改日志，描述了这些修订。

**v1** 初始版本。

**v2** CVPR 2014相机准备修订。包括检测性能的实质性改进，带来的是：(1)从更高的学习率开始微调(0.001而不是0.0001)，(2)在准备CNN输入时使用上下文填充，以及(3)边界框回归以修复定位错误。

**v3** 在ILSVRC2013检测数据集上的实验结果与OverFeat进行了对比，主要集中在Section 2和Section 4两个部分。

**v4** softmax vs. SVM的结果Appendix B包含一个错误，已被修复。我们感谢Sergio guadarama帮助确定这个问题。

**v5** 从Simonyan和Zisserman [43]到Section 3.3和Table 3增加了使用新的16层网络架构的结果。

## References

- [1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 2012. 3
- [2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 9, 11
- [3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3

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 <sub>2</sub> P [4]	<b>84.0</b>	<b>69.0</b>	21.7	47.7	42.2	42.4	<b>64.7</b>	<b>65.8</b>	57.4	<b>12.9</b>	37.4	20.5	43.7	35.7	52.7	51.0	<b>35.8</b>	<b>51.0</b>	28.4	59.8	49.7	46.4
full R-CNN fc <sub>6</sub>	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 <sub>7</sub>	81.0	52.8	<b>25.1</b>	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	<b>56.0</b>	42.5
fg R-CNN fc <sub>6</sub>	81.4	54.1	21.1	40.6	38.7	<b>53.6</b>	59.9	57.2	52.5	9.1	36.5	<b>23.6</b>	46.4	38.1	53.2	51.3	32.2	38.7	<b>29.0</b>	53.0	47.5	43.7
fg R-CNN fc <sub>7</sub>	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 <sub>6</sub>	83.1	60.4	23.2	48.4	<b>47.3</b>	52.6	61.6	60.6	<b>59.1</b>	10.8	<b>45.8</b>	20.9	<b>57.7</b>	43.3	<b>57.4</b>	<b>52.9</b>	34.7	48.7	28.1	60.0	48.6	<b>47.9</b>
full+fg R-CNN fc <sub>7</sub>	82.3	56.7	20.6	<b>49.9</b>	44.2	43.6	59.3	61.3	57.8	7.7	38.4	15.1	53.4	<b>43.7</b>	50.8	52.0	34.1	47.8	24.7	<b>60.1</b>	55.2	45.7

Table 7: VOC 2011验证集上的每个类别分割精度(%)。

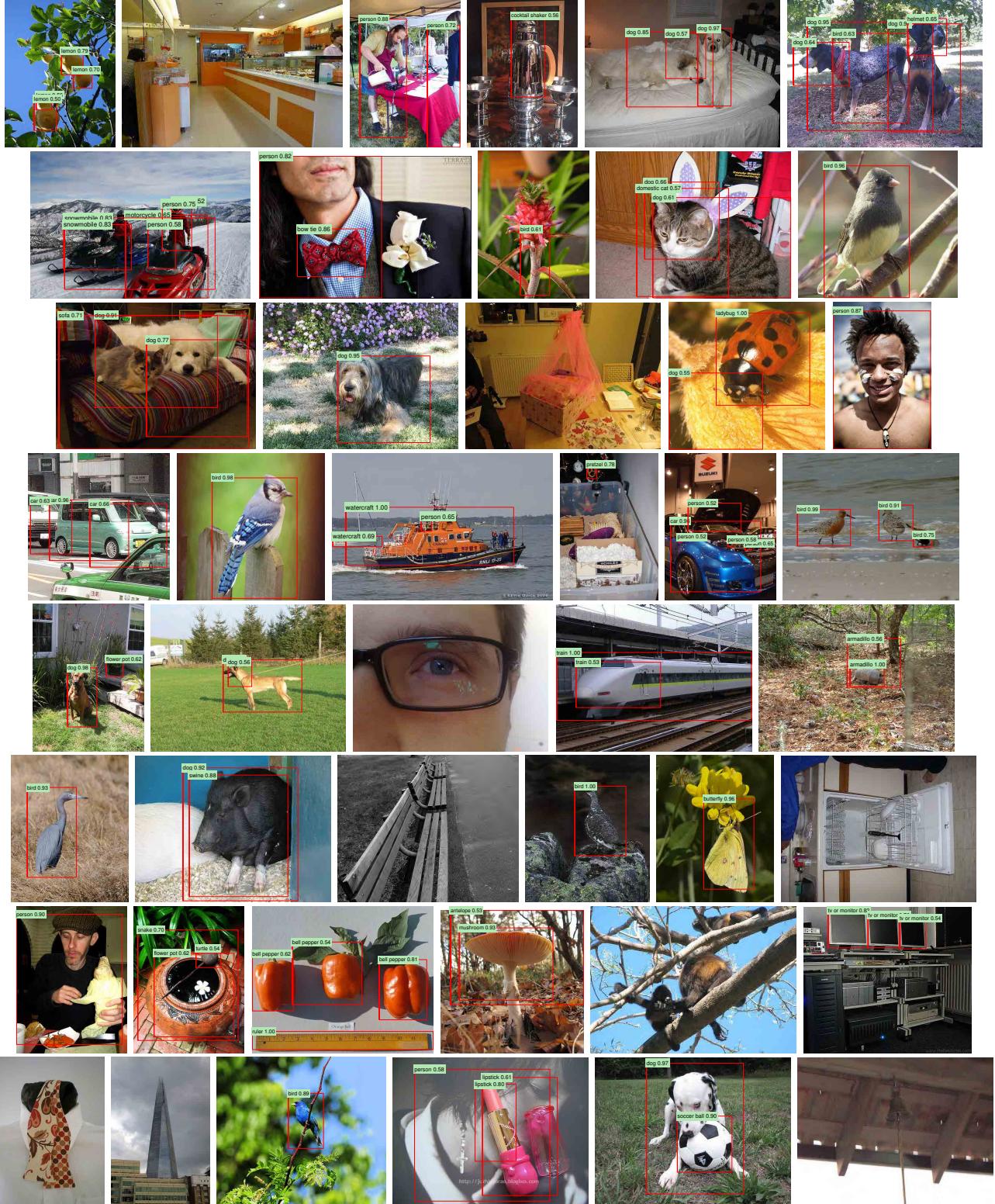
- [4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 4, 9, 10, 11, 12, 13
- [5] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 3
- [6] D. Cireşan, A. 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. 2
- [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. 4
- [9] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). <http://www.image-net.org/challenges/LSVRC/2012/>. 2
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 2
- [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. 3
- [13] M. Douze, H. Jégou, 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. 12
- [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. 2, 4
- [16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013. 9
- [17] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 3, 4, 8, 11
- [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. 2
- [20] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. <http://www.cs.berkeley.edu/~rbg/latent-v5/>. 3, 5, 6, 7
- [21] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 3
- [22] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 9
- [23] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*. 2012. 3, 7, 8
- [24] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. <http://caffe.berkeleyvision.org/>, 2013. 3, 4
- [25] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 3, 4, 5, 6
- [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. 2
- [27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998. 2
- [28] J. J. Lim, C. L. Zitnick, and P. Dollár. 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. 2
- [30] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 12
- [31] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013. 6, 7
- [32] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *TPAMI*, 1998. 3
- [33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing*, 1:318–362, 1986. 2
- [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. 2, 3, 5, 9

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	filng 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	golfeart	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

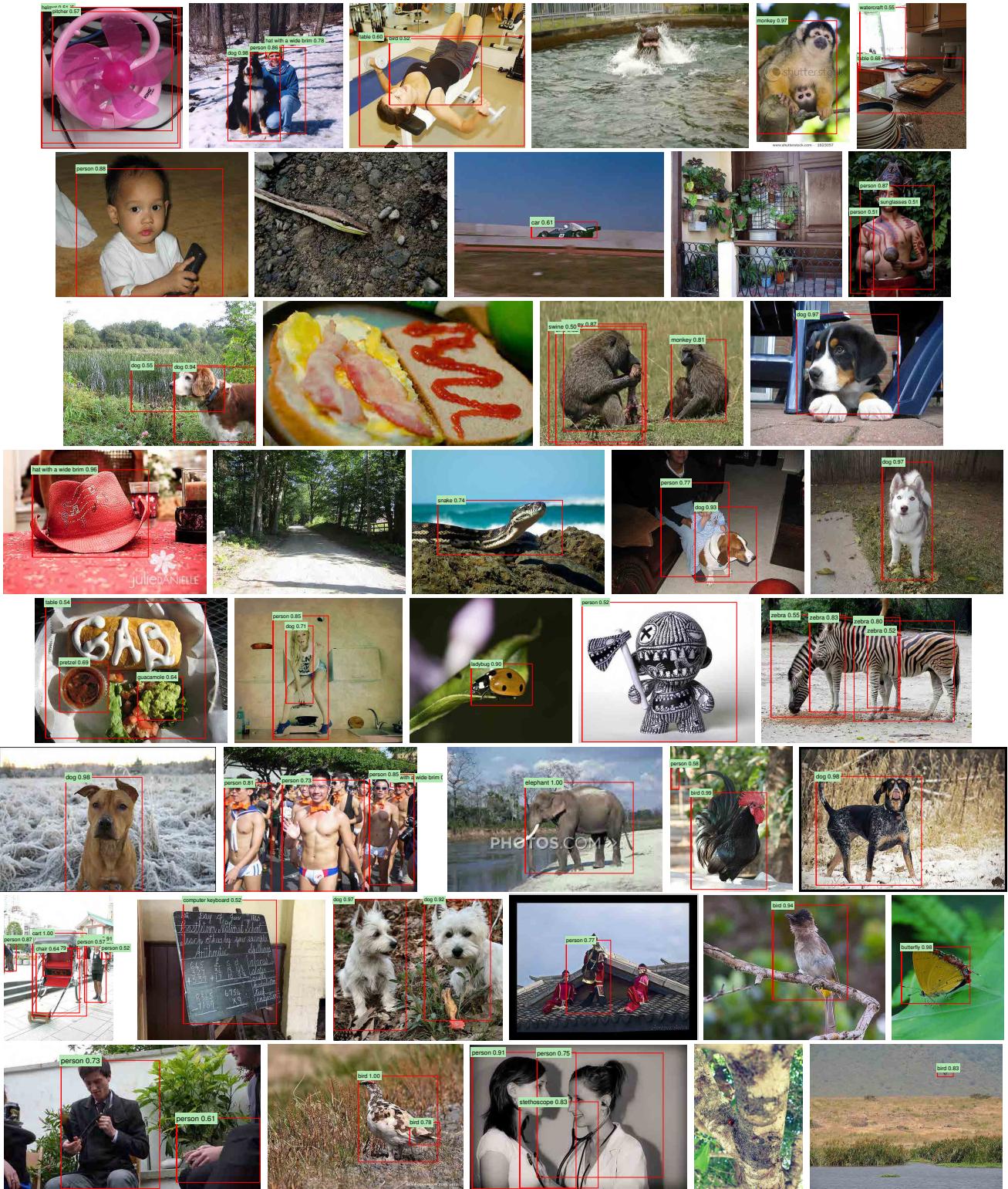
**Table 8:** 在ILSVRC2013检测test集上的类平均精度(%)。

[35] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, 2013. 3

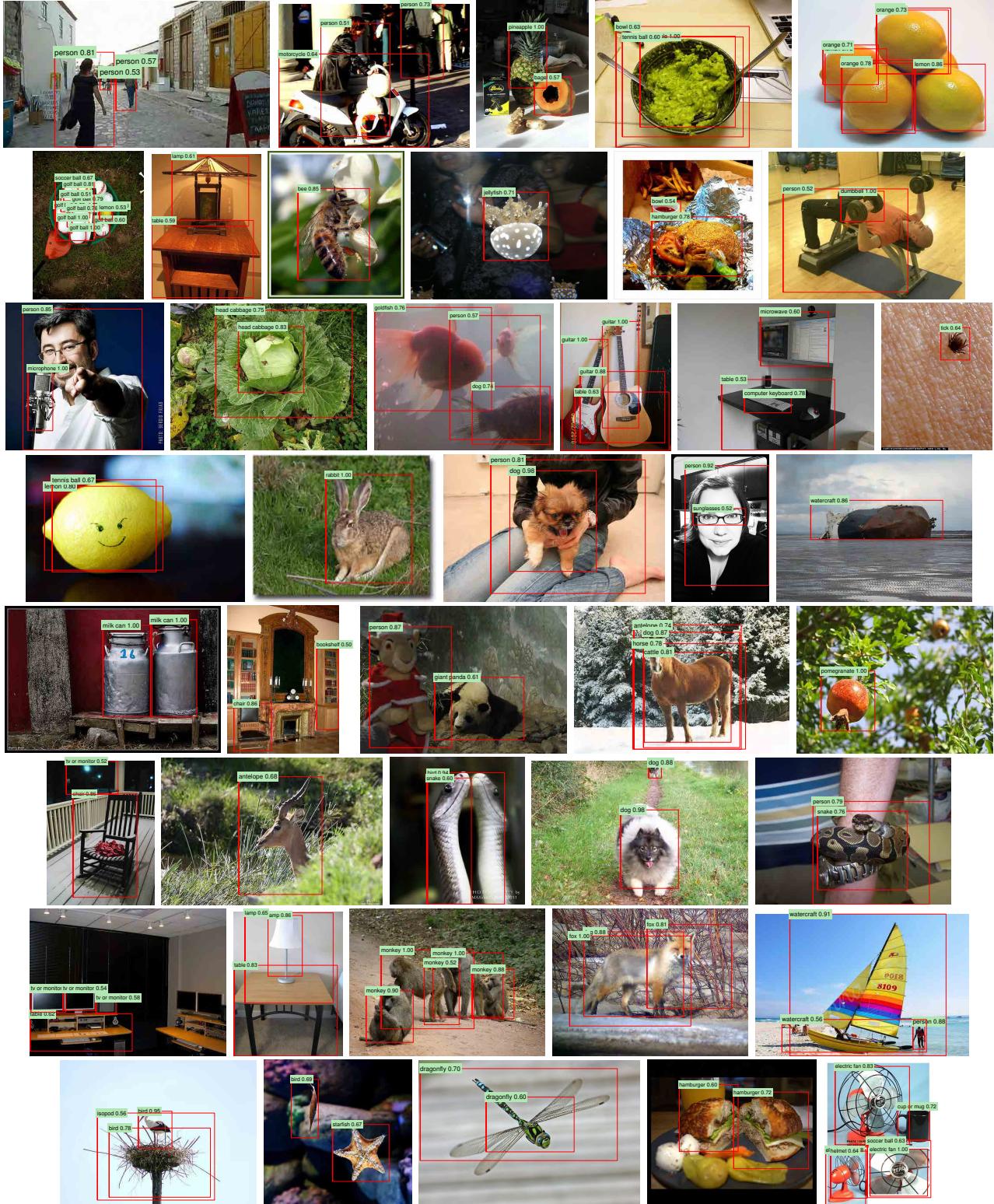
[36] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI Technical Report, 4th Human Computation Workshop*, 2012. 8



**Figure 8:** 从val<sub>2</sub>上实现31.0%mAP的配置在val<sub>2</sub>上的示例检测。每个图像都是随机采样的(这些不是精心策划的)。图中显示了精度大于0.5的所有检测结果。每个检测都用预测的类别和检测器的准确率-召回率曲线中的精度值标记。建议使用数字缩放观看。

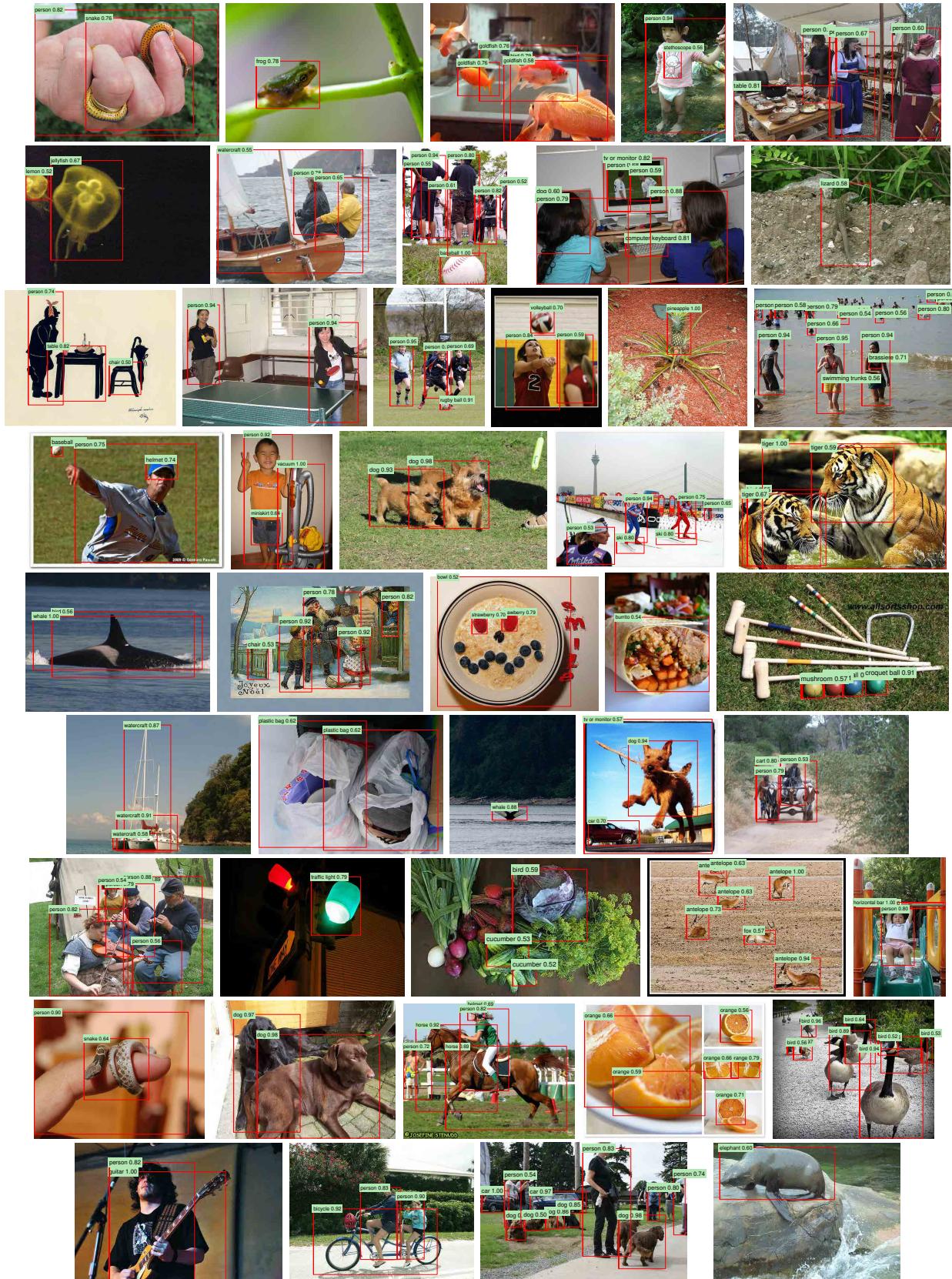


**Figure 9:** 更多随机选择的例子。详情见Figure 8标题。建议使用数字缩放观看。

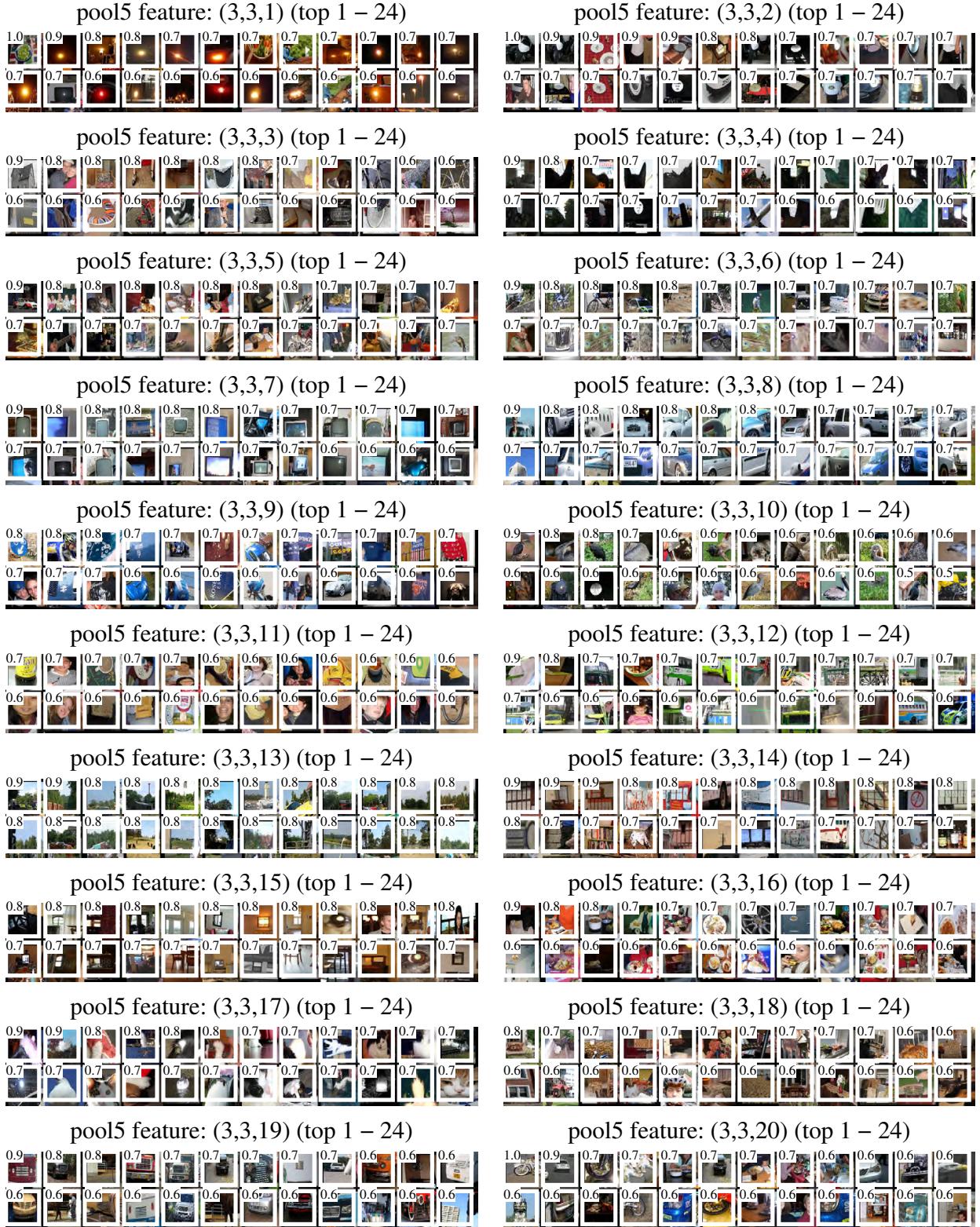


**Figure 10:** 精选的例子。选择每一张图片是因为我们发现它令人印象深刻、令人惊讶、有趣或有趣。建议使用数字缩放观看。

- [37] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical Report A.I. Memo No. 1521, Massachusetts 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. 2, 3, 4, 5, 8
- [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. 3
- [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. 5
- [43] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint*, arXiv:1409.1556, 2014. 7, 12



**Figure 11:** 更多精心策划的例子。详情见Figure 10标题。建议使用数字缩放观看。



**Figure 12:** 展示了24个区域建议，在VOC 2007测试的大约1000万个区域中，最强烈地激活了20个单元中的每个。每个蒙太奇由单元(y, x, 通道)在 $6 \times 6 \times 256$ 维度pool<sub>5</sub>特征图中的位置标记。每个图像区域都被绘制为白色的单元感受野覆盖。激活值(我们通过除以通道中所有单元的最大激活值进行归一化)显示在感受野的左上角。最佳数字观看与zoom。