

# 丰富的要素层次结构，用于准确的对象检测和语义分割

Tech report (v5)

Ross Girshick

Jeff Donahue Trevor Darrell  
UC Berkeley

Jitendra Malik

frbg,jdonahue,trevor,malikg@eecs.berkeley.edu

## 摘要

在规范的 PASCAL VOC 数据集上测量的物体检测性能在过去几年中已经趋于稳定。表现最佳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。在本文中，我们提出了一种简单且可扩展的检测算法，相对于之前对 VOC 2012 的最佳结果，平均精度 (mAP) 提高了 30%以上 - 实现了 53.3% 的 mAP。我们的方法结合了两个关键知识：

(1) 可以将大容量卷积神经网络 (CNN) 应用于自下而上的区域生成，以便对象进行定位和分割；(2) 当标记的训练数据稀缺时，监督辅助任务的预训练，随后进行特定领域的微调，产生显著的性能提升。由于我们将区域生成与 CNN 结合起来，我们将方法称为 R-CNN：具有 CNN 功能的区域。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 RCNN 在 200 级 ILSVRC2013 检测数据集上大大优于 OverFeat。整个系统的源代码可在以下位置获得：

<http://www.cs.berkeley.edu/~rbg/rcnn>

## 1. 介绍

特征很重要。各种视觉识别任务的最后十年进展基于 SIFT [29] 和 HOG [7] 的使用。但是，如果我们看一下规范视觉识别任务 PASCAL VOC 物体检测 [15] 的表现，人们普遍认为 2010 - 2012 年进展缓慢，因为这一阶段主要建立整体系统和设计成功方法的微小变种。

SIFT 和 HOG 是块状方向直方图，我们可以粗略地与 V1 中的复杂细胞相关联，这是灵长类动物视觉路径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，这表明可能存在多等级的，

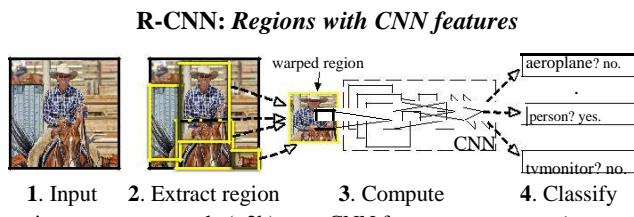


图 1：对象检测系统概述。我们的系统（1）采用输入图像，（2）提取大约 2000 个自下而上区域生成，（3）使用大型卷积神经网络 (CNN) 计算每个生成的特征，然后（4）使用特定的线性分类器 SVM 对每个区域进行分类。R-CNN 在 PASCAL VOC 2010 上实现了 53.7% 的平均精确度 (mAP)。相比之下，[39] 使用相同的区域生成，但是具有空间金字塔和视觉词袋路径，只得到了 35.1% 的 mAP。流行的可变形的组件模型的得分为 33.4%。在 200 级 ILSVRC2013 检测数据集中，RCNN 的 mAP 为 31.4%，比 OverFeat [34] 有很大改进，其中之前的最佳结果为 24.3%。

多阶段的计算特征的过程，这些过程对于视觉识别更具信息性。

福岛的“neocognitron”[19]，一种用于模式识别的生物学启发的分层和移位不变模型，是这种过程的早期尝试。然而，新认知缺乏监督训练算法。以 Rumelhart 等人[33]为基础，LeCun 等人[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) 是有效的，CNN 是一类扩展 neocognitron 的模型。

CNN 在 20 世纪 90 年代得到了大量使用（例如，[27]），但随着支持向量机的兴起，它们已经过时了。2012 年，Krizhevsky 等人[25] 通过在 ImageNet 大规模视觉识别挑战 (ILSVRC) 上显示出更高的图像分类准确度，重新燃起了对 CNN 的兴趣[9,10]。他们的成功是由于荷兰国际列车集团的基于 120 万个带标记图像的超大 CNN，对 LeCun 的 CNN 部分曲折在一起（例如，MAX (X,0) , 修正非线性以及 Dropout 正则化等）。

ImageNet 结果的重要意义引起了非常强烈的讨论

在 ILSVRC 2012 研讨会期间。中心问题可以归结为以下几点: ImageNet 的 CNN 的分类结果在多大程度上推广到 PASCAL VOC 挑战的对象检测结果?

我们通过弥合图像分类和对象检测之间的差距来回答这个问题。本文首次证明, 与基于简单 HOG 特征的系统相比, CNN 可以使 PASCAL VOC 的对象检测性能显著提高。为了实现这一结果, 我们专注于两个问题: 使用深层网络局部提取检测对象, 并仅使用少量带标注的检测数据来训练高容量模型。

与图像分类不同, 检测需要在图像中定位 (可能很多) 对象。一种方法将定位框架化为回归问题。但是, Szegedy 等人的工作[38]与我们自己同时发现, 这种策略在实践中可能表现不佳 (他们报告 VOC 2007 的 mAP 为 30.5%, 而我们的方法实现的 58.5%)。另一种方法是建造一个滑动窗口探测器。CNN 已经以这种方式使用了至少二十年, 通常是在受约束的物体上, 例如面 [32,40] 和行人[35]。为了保持高空间分辨率, 这些 CNN 通常仅具有两个卷积和池化层。我们还考虑采用滑动窗口方法。然而, 在我们的网络中, 有五个卷积层的单位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例内的精确定位成为公开的技术挑战。

相反, 我们通过在“使用区域识别”范例[21]中操作来解决 CNN 定位问题, 该范例已成功用于对象检测 [39] 和语义分割 [5]。在测试时, 我们的方法为输入图像生成大约 2000 个与类别无关的区域生成, 使用 CNN 从每个生成区域中提取固定长度的特征向量, 然后使用类别特定的线性 SVM 对每个区域进行分类。我们使用简单的技术 (仿射图像变幻) 来计算来自每个生成区域的固定大小的 CNN 输入, 而不管区域的形状如何。图 1 概述了我们的方法, 并重点介绍了我们的一些结果。由于我们的系统将区域生成与 CNN 结合起来, 我们称之为 RCNN 方法: 具有 CNN 功能的 Region Proposal。

在本文的更新版本中, 我们通过在 200 级 ILSVRC2013 检测数据集上运行 R-CNN, 提供 R-CNN 与最近提出的 OverFeat [34] 检测系统的头对头比较。OverFeat 使用滑动窗口 CNN 进行检测, 到目前为止, 它是 ILSVRC2013 检测中性能最佳的方法。我们发现 R-CNN 明显优于 OverFeat, 前者 mAP 为 31.4%, 而后者仅为 24.3%。

检测中面临的第二个挑战是标记数据

很少, 目前可用的数量不足以训练大型 CNN。该问题的传统解决方案是使用无监督的预训练, 然后进行有监督的微调 (例如, [35])。本文的第二个主要贡献是要表明, 在一个大的辅助数据集 (ILSVRC) 监督训练前, 先在特定域的小数据集 (PASCAL) 上微调, 是一种数据稀缺时训练高容量 CNN 的有效方法。在我们的实验中, 用于检测的微调可将 mAP 性能提高 8 个百分点。经过微调后, 我们的系统在 VOC 2010 上实现了 54% 的 mAP, 而大幅调整的基于 HOG 的可变形的组件模型 (DPM) 则达到了 33% [17,20]。我们还将向读者介绍 Donahue 等人的同期工作 [12], 他们表明 Krizhevsky 的 CNN 可以作为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表现, 包括场景分类, 细粒度子分类和域适应。

我们的系统也很高效。唯一的分类计算是相当小的矩阵向量乘积和贪婪的非最大抑制。这种计算属性来自于所有类别共享的特征, 并且比先前使用的区域特征也低两维数量级 (参见[39])。

了解我们方法的失效模式对于改进它也是至关重要的, 因此我们报告了 Hoiem 等人的检测分析工具的结果[23]。作为此分析的一个重要结果, 我们证明了一个简单的边界框回归方法可以显著减少错误定位, 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 在区域上运行, 因此将其扩展到语义分割的任务是很自然的。通过微小的修改, 我们还在 PASCAL VOC 分割任务中获得了有竞争力的结果, VOC 2011 测试集的平均分割准确度为 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。第一个生成与类别无关的区域。这些生成区域定义了我们的探测器可用的候选检测集。第二个模块是一个大的循环神经网络, 从每个区域提取固定长度的特征向量。第三个模块是一组用来分类的线性 SVM。在本节中, 我们将介绍每个模块的设计决策, 描述其测试时间和使用情况, 详细说明其参数的学习方式, 并在 PASCAL VOC 2010-12 和 ILSVRC2013 上显示检测结果。

### 2.1. 模块设计

**区域生成提出。**最近的各种论文提供了用于生成与类别无关的区域的方法。



图 2: VOC 2007 列车的扭曲训练样本。

示例包括：对象性[1]，选择性搜索[39]，类别无关的对象提议[14]，约束参数最小分割（CPMC）[5]，多尺度组合分组[3]和 Cires,an 等[6]，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，这是区域生成的一个特例。虽然 R-CNN 对特定区域生成方法是不可知的，但我们使用选择性搜索来实现与先前检测工作的受控比较（例如，[39,41]）。

**特征提取。**我们使用 Krizhevsky 等人描述的 CNN 的 Caffe [24] 实现从每个区域提议中提取 4096 维特征向量 [25]。通过前向传播平均减去的  $227 \times 227$  RGB 图像，用五个卷积层和两个完全连接层来计算特征。我们向读者推荐 [24,25] 以获取更多网络架构细节。

为了计算生成区域的特征，我们必须首先将该区域中的图像数据转换为与 CNN 兼容的形式（其架构需要输入固定的  $227 \times 227$  像素大小）。在我们任意形状区域的许多可能变换中，我们选择最简单的。无论候选区域的大小或纵横比如何，我们都会将围绕它的紧密边界框中的所有像素扭曲到所需的小。在变形之前，我们扩展紧密的边界框，使得在扭曲的尺寸处，在原始框周围存在含有扭曲图像信息的恰好  $p$  个像素（我们使用  $p = 16$ ）。图 2 显示了扭曲训练区域的随机抽样。扭曲的替代方案在附录 A 中讨论。

## 2.2. 测试时间检测

在测试时，我们对测试图像进行选择性搜索以提取大约 2000 个区域生成（我们在所有实验中使用选择性搜索的“快速模式”）。我们对每个区域进行扭曲，然后通过 CNN 将其传播到类的计算当中。也就是，对于每个类，我们使用针对该类训练的 SVM 对每个提取的特征向量进行评分。给定图像中的所有得分区域，我们应用贪婪的非最大抑制（对于每个类独立）算法，如果它与一个得分更高区域的交叉结合重叠（IoU）值大于模型学到的阈值，该得分低的区域就会被抛弃。

**运行时分析。**两个方法使检测效率提高。首先，所有 CNN 参数都在所有类别中共享。第二，由 CNN 计算的特征向量

与其他常见的方法相比，它们是更低维的，例如带有视觉字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 对 4k 维）。

这种共享的结果是，计算区域生成和特征（在 GPU 上为 13s /图像或在 CPU 上为 53s /图像）所花费的时间在所有类上被分摊。唯一的分类计算是特征与 SVM 权重和非最大抑制之间的点积计算。在实践中，图像的所有点积都被分批到单个的矩阵与矩阵的乘积计算中。特征矩阵通常为  $2000 \times 4096$ ，SVM 权重矩阵为  $4096 \times N$ ，其中  $N$  是类的数量。

该分析表明，R-CNN 可以扩展到数千个对象类，而无需采用近似技术，例如散列（hashing）。即使有 100k 类，在现代多核 CPU 上生成的矩阵乘法只需 10 秒。这种效率不仅仅是使用区域生成和共享功能的结果。由于其高维特征，UVA 系统将慢两个数量级，而仅需要 134GB 内存来存储 100k 线性预测器，而我们的低维特征仅需 1.5GB。

将 R-CNN 与 Dean 等人最近的工作进行对比也很有趣。关于使用 DPM 和散列的可扩展检测[8]。他们报告说，当引入 10k 干扰类时，每张图像的运行时间为 5 分钟，VOC 2007 的 mAP 约为 16%。通过我们的方法，10k 检测器可以在 CPU 上运行大约一分钟，并且由于没有进行近似，因此 mAP 将保持在 59%（第 3.2 节）。

## 2.3. 训练

**监督预训练。**我们仅使用带标注图像在大型辅助数据集（ILSVRC2012 分类集）上有条理地预先训练 CNN（此数据不能使用边界框标签）。使用开源 Caffe CNN 库进行预训练[24]。简而言之，我们的 CNN 几乎与 Krizhevsky 等人的表现[25] 相符，在 ILSVRC2012 分类验证集上获得了仅有 2.2% 的错误率的 top-1 成绩。这种差异是由于训练过程的简化。

**特定区域的微调。**为了使我们的 CNN 适应新任务（检测）和新区域（扭曲的生成区域），我们仅使用扭曲生成区域继续 CNN 参数的随机梯度下降（SGD）训练。除了用随机初始化 ( $N + 1$ ) 路分类层替换 CNN 的 ImageNet 专用 1000 路分类层（其中  $N$  是对象类的数量，加上背景为 1），CNN 架构不变。对于 VOC， $N = 20$ ，对于 ILSVRC2013， $N = 200$ 。我们对所有区域进行处理，只要这些区域满足

与完全真实框重叠的  $\text{IoU} \geq 0.5$ , 该区域类就分为正类, 其余为负类。我们以 0.001 的学习率 (初始预训练率的 1/10) 开始 SGD, 这允许微调而不破坏初始化。在每次 SGD 迭代中, 我们单独采样 32 个正窗口 (在所有类别上) 和 96 个背景窗口, 以构建一个 128 的小批量。我们将采样偏向正窗口, 因为它们与背景相比非常罕见。

**对象类别分类器。** 考虑训练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面的例子。同样地, 很明显, 与汽车无关的背景区域应该是一个负面的例子。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 重叠阈值重新解决此问题, 低于该阈值将区域定义为负数。通过验证集上的网格搜索, 我们从  $\{0, 0.1, \dots, 0.5\}$  中不断调整阈值, 最后选定的重叠阈值是 0.3。我们发现, 仔细选择此阈值非常重要。如果将其设置为 0.5, 如[39]中所示, mAP 将降低 5 个点。同样, 将其设置为 0 会使 mAP 降低 4 个点。正例被简单地定义为每个类的基础真值边界框。

一旦提取了特征并应用了训练标签, 我们就会优化每个类的线性分类器 SVM。由于训练数据太大而无法记忆, 我们采用标准的困难负样本挖掘方法(hard negative mining method)[17,37]。困难的负样本挖掘会让模型快速收敛, 并且在实践中, mAP 仅在一次通过所有图像后停止增加。

在附录 B 中, 我们讨论了为什么在微调和 SVM 训练中对正面和负面示例的定义不同。我们还讨论了训练检测 SVM 所涉及的权衡, 而不是简单地使用来自微调 CNN 的最终 softmax 层的输出。

## 2.4. PASCAL VOC 2010-12 上的结果

遵循 PASCAL VOC 最佳实践[15], 我们验证了 VOC 2007 数据集的所有设计决策和超参数 (第 3.2 节)。对于 VOC 2010-12 数据集的最终结果, 我们对 VOC 2012 上训练的 CNN 进行了微调, 并优化了我们在 VOC 2012 train-val 上的检测 SVM。我们仅针对两种主要算法变体 (使用和不使用边界框回归) 将测试结果提交给评估服务器。

表 1 显示了 VOC 2010 的完整结果。我们将我们的方法与四个强基线相比较, 包括 SegDPM [18], 它将 DPM 检测器与语义分割系统的输出结合起来[4], 并使用额外的上下文间检测器和图像分类器进行重评分。最相关的是 Uijlings 等人的 UVA 系统[39], 因为我们的系统使用相同的区域生成算法。为了对区域进行分类, 他们的方法构建了一个四级空间金字塔并用它填充

密集采样的 SIFT, 扩展的 OpponentSIFT 和 RGB-SIFT 描述符, 每个矢量用 4000 字的编码, 实现规范化。使用直方图交叉核 SVM 执行分类。与多特征, 非线性核 SVM 方法相比, 我们在 mAP 方面取得了很大的改进, 从 mAP 的 35.1% 到 53.7%, 同时也更快 (第 2.2 节)。我们的方法在 VOC 2011/12 测试中实现了类似的性能 (53.3% mAP)。

## 2.5. 在 ILSVRC2013 上检测的结果

我们使用与 PASCAL VOC 相同的系统超参数在 200 级 ILSVRC2013 检测数据集上运行 R-CNN。我们遵循相同的协议, 将测试结果仅提交给 ILSVRC2013 评估服务器两次, 一次使用边界框重复, 而另一次没有。

图 3 比较了 R-CNN 与 ILSVRC 2013 竞赛中的参赛作品特别是 OverFeat 的比较结果[34]。R-CNN 的 mAP 达到 31.4%, 远远高于 OverFeat 的 24.3% 的第二好成绩。为了了解类别上的 AP 分布情况, 还提供了箱形图, 并在表 8 的结尾处列出了每类 AP 的表格。大多数竞争提交 (OverFeat, NEC-MU, UvA-Euvision, Toronto-A 和 UIUC-IFP) 使用卷积神经网络, 表明 CNN 如何应用于物体检测存在显著的细微差别, 导致结果差异很大。

在第 4 节中, 我们概述了 ILSVRC2013 检测数据集, 并提供了有关在其上运行 R-CNN 时所做选择的详细信息。

## 3. 可视化, 消融和错误模式

### 3.1. 可视化学习的功能

第一层卷积核可以直接显示, 易于理解[25], 它们捕捉定向边缘和相反的颜色。了解后续层更具挑战性。Zeiler 和 Fergus 在[42]中提出了一种视觉上具有吸引力的反卷积方法。我们提出了一种简单 (和互补) 的非参数方法, 可直接显示网络学到的内容。

我们的想法是在网络中挑出一个特定的单元 (也就是特征) 并使用它就好像它本身就是一个物体探测器。也就是说, 我们计算在大量保留生成区域 (约 1000 万) 上的激活函数, 从最高到最低激活结果对生成区域进行排序, 执行非最大值抑制, 然后显示得分最高的区域。我们的方法通过准确显示它所触发的输入, 让所选单元 “自己说话” 。我们避免平均结果, 是为了看到不同的视觉模式, 并深入了解单元 (特征) 计算的不变性。

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>

表 1: VOC 2010 测试的检测平均精度 (%)。R-CNN 与 UVA 和 Regionlet 最直接可比, 因为所有方法都使用选择性搜索生成区域。边界框回归 (BB) 在 C 部分中描述。在出版时, SegDPM 是 PASCAL VOC 排行榜的最佳表现者。yDPM 和 SegDPM 使用了其他方法未使用的上下文重新绑定。

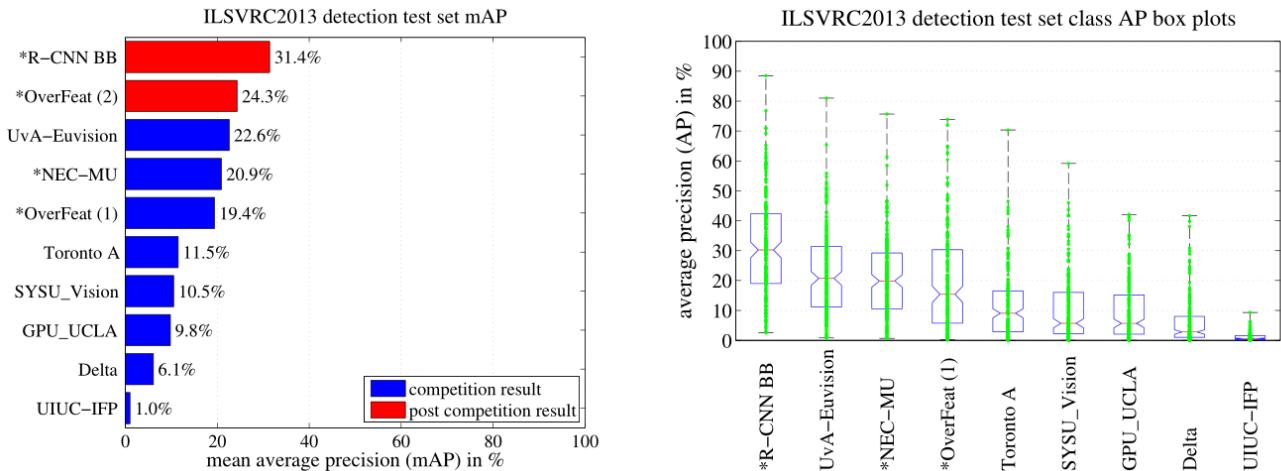


图 3: (左) ILSVRC2013 检测测试装置的 mAP。方法之前\*表示使用外部训练数据 (在所有情况下来自 ILSVRC 分类数据集的图像和标签)。(右) 每个方法的 200 个平均精度值的箱形图。未显示赛后 OverFeat 结果的方框图, 因为每类 AP 尚不可用 (R-CNN 的每类 AP 在表 8 中, 也包含在上传到 arXiv.org 的技术报告源中; 请参阅 R-CNN-ILSVRC2013-APs.txt)。红线表示中位数 AP, 方框底部和顶部是第 25 和第 75 百分位数。晶须延伸到每种方法的最小和最大 AP。每个 AP 在胡须上绘制为绿点 (最好以数字方式查看缩放)。

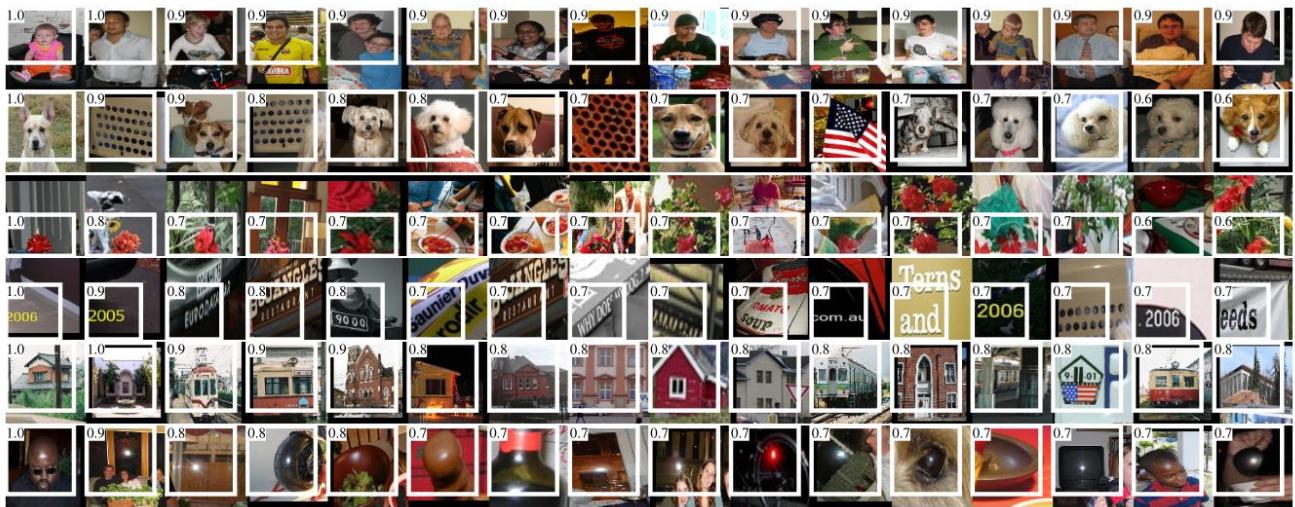


图 4: 六个 pool5 单元的顶部区域。接收字段和激活值以白色绘制。某些单元与概念对齐, 例如人 (第 1 行) 或文本 (4)。其他单位捕获纹理和材质属性, 例如点阵列 (2) 和镜面反射 (6)。

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

表2: VOC 2007 测试的检测平均精度 (%)。第1-3行显示 R-CNN 性能而无需微调。第4-6行显示 CNN 在 ILSVRC 2012 上进行预训练的结果, 然后在 VOC 2007 trainval 上进行微调 (FT)。第7行包括一个简单的边界框回归 (BB) 阶段, 可以减少定位误差 (C部分)。第8-10行将 DPM 方法作为强基线。第一个仅使用 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>

表3: 两种不同 CNN 架构的 VOC 2007 测试的检测平均精度 (%)。前两行是表2中使用 Krizhevsky 等人的架构 (T-Net) 的结果。第三和第四行使用最近提出的 Simonyan 和 Zisserman (O-Net) 的 16 层架构[43]。

我们从图层 pool5 可视化单元, 这是网络的第五个和最后一个卷积层的最大池输出。pool5 特征图是  $6 \times 6 \times 256 = 9216$  维。忽略边界效应, 每个 pool5 单元在原始  $227 \times 227$  像素输入中具有  $195 \times 195$  像素的感受野。中央 pool5 单元几乎具有全局视图, 而靠近边缘的一个单元具有较小的, 裁剪过的支撑。

图4中的每一行显示了我们在 VOC 2007 trainval 上进行微调的 CNN 中的 pool5 单元的前 16 次激活。256 个功能独特单元中的 6 个是可视化的(附录D 包括更多)。选择这些单元以显示网络学习的代表性样本。在第二行中, 我们看到一个在狗脸和点阵列上发射的单位。对应于第三行的单位是红色斑点检测器。还有用于人脸的探测器和更抽象的图案, 例如带有窗户的文本和三角形结构。该网络似乎学习了一种表示, 该表示将少量的类的调整特征与形状, 纹理, 颜色和材料属性的分布式表示相结合。随后的完全连接层 fc6 具有对这些丰富特征的大量组合进行建模的能力。

### 3.2. 消融研究

**性能逐层, 无需微调。**为了解释哪些层对检测性能至关重要, 我们分析了每个 CNN 最后三层的 VOC 2007 数据集的结果。Layer pool5 在 3.1 节中简要描述。最后两层总结如下。

层 fc6 完全连接到 pool5。为了计算特征, 它将  $4096 \times 9216$  权重矩阵乘以 pool5 特征图(重新形成为 9216 维向量), 然后添加偏差向量。该中间矢量是分量半波整流的( $x \leftarrow \max(0, x)$ )。

层 fc7 是网络的最后一层。它是通过将由 fc6 计算的特征乘以  $4096 \times 4096$  权重矩阵, 并类似地添加偏差矢量并应用半波整流来实现的。

我们首先查看 CNN 的结果, 不对 PASCAL 进行微调, 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)显示 fc7 的特征比 fc6 的特征更糟糕。这意味着可以在不降低 mAP 的情况下移除 29% 或约 1680 万个 CNN 参数。更令人惊讶的是, 即使仅使用 6% 的 CNN 参数计算 pool5 特征, 同时移除 fc7 和 fc6 也会产生相当好的结果。CNN 的大部分代表性力量来自其卷积层, 而不是来自更大的密集连接层。该发现揭示了一个潜在效用, 即通过仅使用 CNN 的卷积层, 来计算基于 HOG 特征的任意大小的图像的密集特征图。这种表示, 验证了在 pool5 特征之上进行滑动窗口检测器(包括 DPM)的实验的可行性。

**逐层执行, 具有微调功能。**我们现在看一下 CNN 的调整后的结果

关于 VOC 2007 trainval 的参数。改进正在进行 (表 2 第 4-6 行) : 微调将 mAP 提高 8.0 个百分点至 54.2%。fc6 和 fc7 的微调提升比 pool5 大得多, 这表明从 ImageNet 学到的 pool5 特性是通用的, 并且大多数改进都是从学习特定领域的非线性分类器获得的。

**与最近的特征学习方法的比较。**相对较少的特征学习方法已经尝试过 PASCAL VOC 检测。我们看看最近基于可变形的组件模型的两种方法。作为参考, 我们还包括基于标准 HOG 的 DPM 的结果[20]。

第一个 DPM 特征学习方法 DPM ST [28]利用“草图标记”概率的直方图增强 HOG 特征。直观地说, 草图标记是通过图像补丁中心的轮廓的紧密分布。通过随机森林在每个像素处计算草图标记概率, 该随机森林被训练以将  $35 \times 35$  像素斑块分类为 150 个草图标记或背景之一。

第二种方法 DPM HSC [31]用稀疏码 (HSC) 的直方图替换 HOG。为了计算 HSC, 使用  $100 \times 7 \times 7$  像素 (灰度) 原子的学习字典在每个像素处求解稀疏码激活值。产生的激活以三种方式 (全波和两半波) 进行整流, 空间合并, 单元  $\ell_2$  归一化, 然后进行功率变换 ( $x \leftarrow \text{sign}(x)|x|^\alpha$ )。

所有 R-CNN 变体都强大地优于三个 DPM 基线 (表 2 第 8-10 行), 包括使用特征学习的两个。与仅使用 HOG 功能的 DPM 的最新版本相比, 我们的 mAP 高出 20 多个百分点: 54.2% 对比 33.7%——这意味着 61% 的相对改善。HOG 和草图标记的字段的组合比单独的 HOG 提高 2.5 mAP 点, 而 HSC 比 HOG 提高 4 mAP 点 (内部与其私有 DPM 基线进行比较 - 都使用 DPM 的非公开实现, 其表现不如开源版本[20])。这些方法分别实现 29.1 % 和 34.3% 的 mAP。

### 3.3. 网络架构

本文中的大多数结果都使用了 Krizhevsky 等人的网络架构[25]。但是, 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, 我们显示了使用 Simonyan 和 Zisserman 最近提出的 16 层深层网络的 VOC 2007 测试结果[43]。该网络是最近 ILSVRC 2014 年分类挑战中表现最佳的网络之一。该网络具有均匀的结构, 由 13 层  $3 \times 3$  的卷积核组成, 其中散布着 5 个最大池化层, 并且顶部有三个完全连接层。我们将这个网络称为 OxfordNet 的 “O-Net”, 并将多伦多网的基线称为 “T-Net”。

为了在 R-CNN 中使用 O-Net, 我们从 Caffe Model Zoo 下载了 VGG ILSVRC 16 层模型的公开可预训练的网络权重。然后我们使用相同的协议对网络进行了微调。我们用于 T-Net。唯一的区别是根据需要使用较小的微型计算机 (24 个示例) 以适应 GPU 内存。表 3 中的结果表明, 具有 O-Net 的 R-CNN 基本上优于具有 T-Net 的 R-CNN, 将 mAP 从 58.5% 增加至 66.0%。然而, 在计算时间方面存在相当大的缺点, O-Net 的正向传输大约比 T-Net 长 7 倍。

### 3.4. 检测错误分析

我们应用了 Hoiem 等人的优秀检测分析工具 [23], 为了揭示我们方法的错误模式, 了解微调如何改变它们, 以及了解我们的错误类型与 DPM 的比较。完整的分析工具总结超出了本文的范围, 我们鼓励读者参考[23]来理解一些更精细的细节 (例如 “规范化的 AP”)。由于分析最好在相关图的背景下被吸收, 因此我们在图 5 和图 6 的标题内进行讨论。

### 3.5. 边界框回归

基于错误分析, 我们实施了一种简化方法来减少本地化错误。受 DPM [17] 中使用的边界框回归的启发, 我们训练线性回归模型, 以预测新的检测窗口, 给出选择性搜索区域提供的 pool5 特征。详细信息见附录 C。表 1, 表 2 和图 5 中的结果表明, 这种简单的方法可以修复大量错误定位的检测, 将 mAP 提高 3 到 4 个点。

### 3.6. 定性结果

ILSVRC2013 的定性检测结果在本文末尾的图 8 和图 9 中给出。从 val2 组中随机采样每个图像, 并显示所有检测器的所有检测, 精度大于 0.5。请注意, 这些都没有策划, 并给出了探测器的实际印象。更多的定性结果如图 10 和图 11 所示, 但这些结果已经过策划。我们选择了每个图像, 因为它包含有趣或令人惊讶的结果。此外, 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, 我们提供了 ILSVRC2013 检测数据集的结果。该数据集比 PASCAL VOC 更不均匀,

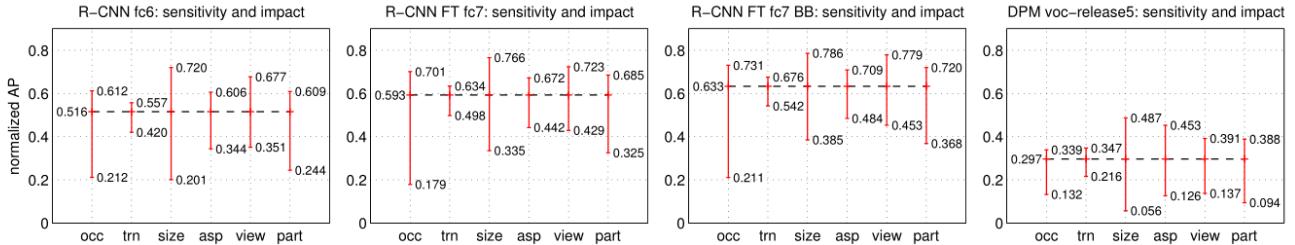


图 6: 对象特征的灵敏度。每个图显示了六个不同对象特征(遮挡, 截断, 边界框区域, 纵横比, 视点, 部分可见性)内最高和最低性能子集的均值(在类别上)归一化 AP(参见[23])。我们显示了我们的方法(R-CNN)的图表, 有和没有微调(FT)和边界框回归(BB)以及DPM voc-release5。总的来说, 微调不会降低灵敏度(最大值和最小值之间的差异), 但几乎可以显著改善几乎所有特性的最高和最低性能子集。这表明微调不仅仅是简单地改进宽高比和边界框区域中性能最低的子集, 因为人们可能会根据我们如何扭曲网络输入进行猜测。相反, 微调可以提高所有特征的鲁棒性, 包括遮挡, 截断, 视点和组件可见性。

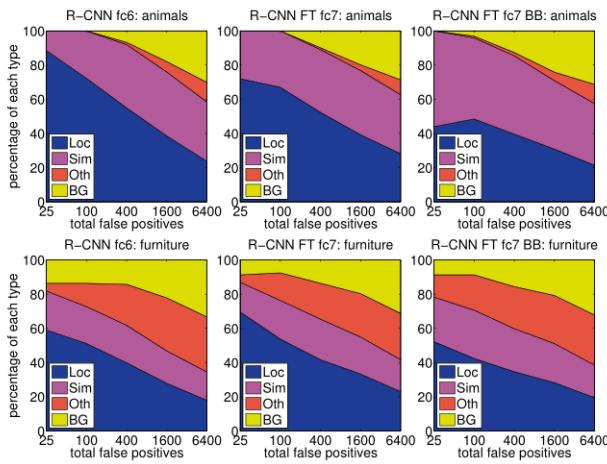


图 5: 排名最高的假阳性(FP)类型的分布。每个图显示 FP 类型的演变分布, 因为按照得分降低的顺序考虑更多的 FP。每个 FP 都分为 4 种类型中的一种: Loc-poor 定位(iou 重叠检测, 正确等级在 0.1 和 0.5 之间, 或者一个重复); 模拟混淆与类似的类别; 与不同对象类别的混淆; BG-a FP 在背景上发射。与 DPM(参见[23])相比, 我们的错误显然更多是由于定位不佳, 而不是与背景或其他对象类混淆, 这表明 CNN 特征比 HOG 更具辨别力。松散的本地化可能是由于我们使用自下而上的区域提议以及从预训练 CNN 进行全图像分类所学到的位置不变性。第三列显示了我们的简单边界框回归方法如何修复许多本地化错误。

需要选择如何使用它。由于这些决定并非无足轻重, 因此我们将在本节中介绍这些决定。

#### 4.1. 数据集概述

ILSVRC2013 检测数据集分为三组: train(395,918), val(20,121) 和 test(40,152), 其中每组中的图像数量在括号中。该

验证和测试分割是从相同的图像分布中提取的。这些图像是类似场景的, 并且与 PASCAL VOC 图像的复杂性(物体数量, 杂波量, 姿势可变性等)相似。验证和测试拆分被完全注释, 这意味着在每个图像中, 来自所有 200 个类的所有实例都用边界框标记。相比之下, 训练组则来自ILSVRC2013 分类图像分布。这些图像具有更多变化的复杂性, 并且偏向于单个对象的图像。与验证和测试不同, 训练图像(由于它们的数量很大)没有被详尽地注释。在任何给定的训练图像中, 来自 200 个类的实例可以标记或不标记。除了这些图像集之外, 每个类还有一组额外的负像。手动检查负图像以验证它们不包含其关联类的任何实例。在这项工作中没有使用负面图像集。有关如何收集和注释ILSVRC 的更多信息可以在[11,36]中找到。

这些分裂的性质为训练 R-CNN 提供了许多选择。训练图像不能用于困难负类挖掘, 因为注释并不是非常耗费精力。负面例子应该从哪里来? 此外, 训练图像具有与验证和测试不同的统计数据。是否应该使用训练图像, 如果使用, 在何种程度上? 虽然我们尚未对大量选择进行全面评估, 但我们根据之前的经验提出了看似最明显的路径。

我们的总体策略是严重依赖验证集并使用一些训练图像作为积极示例的辅助来源。要将验证用于训练和验证, 我们将其分成大致相同大小的“验证 1”和“验证 2”集。由于某些类在验证中的示例非常少(最小的只有 31 个而一半只有不到 110 个), 因此生成近似类平衡的参数非常重要。为此, 生成了大量候选分割并且那些具有最小的最大相对

不平衡的类被选择了。2 通过使用类别计数作为特征聚类验证图像，然后进行可以改善分割平衡的随机本地搜索，生成每个候选分割。这里使用的特定分割具有约 11% 的最大相对不平衡和 4% 的中值相对不平衡。验证 1 / 验证 2 拆分和用于生成它们的代码将公开提供，以允许其他研究人员比较他们在本报告中使用的验证集拆分方法。

## 4.2. 区域生成

我们遵循用于在 PASCAL 上检测的相同区域生成方法。选择性搜索[39]在 val1, val2 和 test 中的每个图像上以“快速模式”运行（但不在训练中的图像上）。需要进行一次小修改以处理选择性搜索不是尺度不变的事实，因此产生的区域数量取决于图像分辨率。ILSVRC 图像大小范围从非常小到几个几百万像素，因此我们在运行选择性搜索之前将每个图像的大小调整为固定宽度（500 像素）。在 val 上，选择性搜索导致每个图像平均有 2403 个区域生成，所有真实边界框的召回率为 91.6%（0.5 IoU 阈值）。此次召回明显低于 PASCAL，其约为 98%，表明该区域生成阶段的改善空间很大。

## 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 val1 的所有选择性搜索和完全真实框以及来自训练的每类最多 N 个真实框（如果一个类的真实实例框少于 N 个）训练，然后我们采取所有这些。我们将这个数据集称为图像和框 val1 + trainN。在消融研究中，我们在 val2 上显示对于  $N \in \{0, 500, 1000\}$  的 mAP（第 4.5 节）。

R-CNN 中的三个程序需要训练数据：

(1) CNN 微调，(2) 检测器 SVM 训练，(3) 边界框回归训练。使用与 PASCAL 中完全相同的设置，在 val1 + trainN 上运行 CNN 微调以进行 50k SGD 迭代。使用 Caffe 对一个单一的 NVIDIA Tesla K20 进行微调需要 13 个小时。对于 SVM 训练，来自 val1 + trainN 的所有完全真实框用作其各自类别的正例。对来自 val1 的 5000 个图像的随机选择子集进行困难负例挖掘。最初的实验表明，从所有 val1 中挖掘负数而不是 5000 个图像子集（大约一半），导致 mAP 仅下降 0.5 个百分点，同时将 SVM 训练时间缩短一半。没有任何负面例子

<sup>2</sup>相对不平衡测量为  $|a - b| / (a + b)$  其中 a 和 b 是分裂的每一半中的类别数。

因为注释并非详尽无遗而进行训练。未使用额外集合的验证负例图像。边界框回归量是在 val1 上训练的。

## 4.4. 验证和评估

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 val2 集的影响。所有系统超参数（例如，SVM C 超参数，区域扭曲中使用的填充，NMS 阈值，边界框回归超级参数）被固定在用于 PASCAL 的相同值。毫无疑问，这些超参数选择中的一些对于 ILSVRC 来说略微不理想，但是这项工作的目标是在没有大量数据集调整的情况下在 ILSVRC 上产生初步的 R-CNN 结果。在 val2 上选择最佳选择后，我们将两个结果文件提交给 ILSVRC2013 评估服务器。第一个子任务没有边界框回归，第二个子任务是边界框回归。对于这些提交，我们扩展了 SVM 和边界框回归训练集，以分别使用 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。

## 4.5. 消融研究

表 4 显示了不同数量的训练数据，微调和边界框回归效应的消融研究。第一个观察结果是 val2 上的 mAP 与测试中的 mAP 非常接近。这让我们相信，val2 上的 mAP 是测试集性能的良好指标。第一个结果，20.9%，是 R-CNN 使用在 ILSVRC2012 分类数据集上预训练的 CNN（无微调）实现的，并且可以访问 val1 中的少量训练数据（回想一下 val1 中的类有 15 到 55 个例子）。将训练集扩展到 val1+trainN 可将性能提高到 24.1%， $N = 500$  和  $N = 1000$  之间基本没有差别。使用 val1 中的示例对 CNN 进行微调可以适度提高到 26.5%，但可能会由于少数正面的训练样例而过度拟合。将微调集扩展到 val1 + train1k，从训练组中每组增加 1000 个正面测试，有助于显著提高 mAP 至 29.7%。边界框回归将结果提高到 31.0%，这是 PASCAL 中观察到的较小的相对增益。

## 4.6. 与 OverFeat 的关系

R-CNN 和 OverFeat 之间存在一种有趣的关系：OverFeat 可以（大致）看作是 R-CNN 的一个特例。如果要替换选择性搜索区域

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>

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

使用常规方形区域的多尺度金字塔的建议, 并将每类边界框回归量更改为单个边界框回归量, 然后系统将非常相似 (模拟它们的一些潜在的显著差异) 训练: CNN 检测微调, 使用 SVM 等。值得注意的是, OverFeat 比 R-CNN 具有明显的速度优势: 基于[34]引用的每张图像 2 秒的速度, 它大约快 9 倍。这种速度来自于 OverFeat 的滑动窗口 (即区域生成) 在图像级别没有扭曲的事实, 因此可以在重叠窗口之间容易地共享计算。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共享。加速 R-CNN 应该可以通过各种方式加速, 并且仍然是未来的工作。

## 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 分段挑战。为了便于与当前领先的语义分段系统 (称为 “二阶池” 的 O2P) 进行直接比较[4], 我们在他们的开源框架内工作。O2P 使用 CPMC 为每个图像生成 150 个区域生成, 然后使用支持向量回归 (SVR) 预测每个类别的每个区域的质量。他们的方法的高性能是由于 CPMC 区域的质量和多种特征类型的强大二阶汇集 (SIFT 和 LBP 的丰富变体)。我们还注意到 Farabet 等人[16]最近在使用 CNN 作为多尺度每像素分类器的几个密集场景标记数据集 (不包括 PAS-CAL) 上展示了良好的结果。

我们遵循 [2,4] 并扩展 PASCAL 分段训练集以包括 Hariharan 等人提供的额外注释[22]。设计决策和超参数在 VOC 2011 验证集上进行了交叉验证。最终测试结果仅评估一次。

**CNN 功能用于细分。** 我们评估了 CPMC 区域计算特征的三个策略, 所有这些策略都是通过将区域周围的矩形窗口扭曲到  $227 \times 227$  开始的。第一个策略 (full) 忽略了

区域的形状和直接在扭曲窗口上计算 CNN 特征, 就像我们检测时一样。但是, 这些功能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, 而重叠非常少。因此, 第二个战略 (fg) 仅在区域的前地掩模上计算 CNN 特征。我们用平均输入替换背景, 以便在平均牵引力之后背景区域为零。第三个策略 (full + fg) 简单地连接 full 和 fg 功能; 我们的实验证明了它们的完美性。

O2P [4]	full R-CNN		fg R-CNN		full+fg R-CNN	
	fc <sub>6</sub>	fc <sub>7</sub>	fc <sub>6</sub>	fc <sub>7</sub>	fc <sub>6</sub>	fc <sub>7</sub>
46.4	43.0	42.5	43.7	42.1	<b>47.9</b>	45.8

表 5: VOC 2011 评估的分段平均准确度 (%)。第 1 列呈现 O2P; 2-7 使用我们在 ILSVRC 2012 上预训练的 CNN。

**关于 VOC 2011 的结果。** 表 5 显示了我们在 VOC 2011 验证集上与 O2P 相比的结果的总结。(有关 full 的每类别结果, 请参阅附录 E) 在每个特征计算策略中, 图层 fc<sub>6</sub> 总是优于 fc<sub>7</sub>, 以下讨论涉及 fc<sub>6</sub> 特征。fg 策略稍微优于 full, 表明屏蔽区域形状提供了更强的信号, 与我们的直觉相匹配。然而, full + fg 实现了 47.9% 的平均准确度, 我们的最佳结果是 4.2% 的边缘 (也略微优于 O2P), 表明即使给出 fg 功能, full 特征提供的文本信息量也很大。值得注意的是, 在我们的 full + fg 功能上训练 20 个 SVR 在单个核心上花费一个小时, 而在 O2P 功能上训练需要 10 个小时。

在表 6 中, 我们提供了 VOC 2011 测试集的结果, 将我们表现最佳的方法 fc<sub>6</sub> (full + fg) 与两个强基线进行了比较。我们的方法在 21 个类别中的 11 个中实现了最高的分割准确度, 并且最高的整体分割准确度为 47.9%, 不同类别的平均值 (但在任何合理的误差范围内可能与 O2P 结果相关)。通过微调可以实现更好的性能。

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>	

表6: VOC 2011 测试的分段准确度 (%)。我们比较两个强基线: [2]的“区域和部分”(R&P)方法和[4]的二阶汇集(O<sub>2</sub>P)方法。没有任何微调,我们的CNN实现了最高的分割性能,优于R&P并且大致匹配O<sub>2</sub>P。

## 6. 结论

近年来,物体检测性能已经停滞不前。表现最佳的系统是复杂的组合,将多个低级图像特征与来自物体探测器和场景分类的高级上下文相结合。本文介绍了一种简单且可扩展的物体检测算法,该算法相对于PASCAL VOC 2012上的最佳结果提供了30%的相对改进。

我们通过两个见解实现了这一表现。第一种是将高容量卷积神经网络应用于自下而上的区域生成,以便对对象进行本地化和分割。第二个是在标记的训练数据稀缺时训练大型CNN的范例。我们表明,对于具有丰富数据(图像分类)的辅助任务,通过监督来预训练网络是非常有效的,然后针对数据稀缺(检测)的目标任务微调网络。我们认为,“监督的预训练/区域特定的微调”范例对于各种数据稀缺的视觉问题将非常有效。

最后,我们指出通过结合使用计算机视觉和深度学习(自下而上区域生成和卷积神经网络)的分类工具来实现这些结果是很重要的。这两者不是反对科学探究的对象,而是自然而然不可避免的合作伙伴。

## 致谢

这项研究得到了DARPA Mind's Eye和MSEE计划的部分支持,NSF颁发了IIS-0905647,IIS-1134072和IIS-1212798,MURI N000014-10-1-0933以及丰田的支持。本研究中使用的GPU由NVIDIA公司慷慨捐赠。

## 附录

### A. 目标生成转换

在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。对于检测,我们将对象生成视为任意图像矩形。我们评估了两种将对象生成转换为有效CNN输入的方法。

第一种方法(“带有上下文的最严格的正方形”)在最紧密的正方形内部关闭每个对象生成



图7:不同的对象生成转换。(A)相对于转换后的CNN输入的实际规模的原始对象生成;(B)具有背景的最严格的正方形;(C)没有背景的最严格的广场;(D)扭曲。在每列和示例生成中,顶行对应于p=0像素的文本填充,而底行对应p=16像素的上下文填充。

然后将该方块中包含的图像(各向同性地)缩放到CNN输入大小。图7栏(B)显示了这种转变。该方法的变体(“没有上下文的最严格的正方形”)排除了围绕原始对象生成的图像内容。图7列(C)显示了这种转变。第二种方法(“扭曲”)各向异性地将每个对象生成缩放到CNN输入大小。图7列(D)显示了扭曲变形。

对于这些转换中的每一个,我们还考虑包括原始对象生成周围的附加图像上下文。上下文填充(p)的量被定义为在变换的输入坐标系中围绕原始对象生成的边界大小。图7示出了每个示例的顶行中的p=0像素,并且底行中的p=16像素。在所有方法中,如果源矩形超出图像,则将缺失的数据替换为图像均值(然后在将图像输入CNN之前将其减去)。一组试验性实验表明,使用上下文填充(p=16像素)的变形优于替代方案(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。对这些替代方案的彻底评估仍然是未来的工作。

## B.正面与负面的例子和 softmax

两种设计选择值得进一步讨论。第一个是：为什么正面和负面的例子被定义为微调 CNN 而不是训练对象检测 SVM？为了简要回顾这些定义，对于微调，我们将每个对象生成映射到具有最大 IoU 重叠（如果有的话）的完全真实实例，并且如果 IoU 至少为 0.5，则将其标记为匹配的真实实例类的正数。所有其他提案都标有“背景”（即所有类别的反面例子）。相反，对于训练 SVM，我们只将真实实例框作为其各自类别和标签提案的正例，其中 IoU 重叠小于一个类的所有实例作为该类的负数。落入灰色区域的建议（超过 0.3 IoU 重叠，但不是基本事实）将被忽略。

从历史上看，我们已经达到了这些定义，因为我们开始通过训练 SVM 来预测由 ImageNet 预训练的 CNN 计算的特征，因此在那个时间点不需要进行微调。在该设置中，我们发现我们用于训练 SVM 的特定标签定义在我们评估的选项集中是最佳的（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，我们最初使用与我们用于 SVM 训练相同的正面和负面示例定义。然而，我们发现结果比使用我们目前的正面和负面定义所获得的结果要糟糕得多。

我们的假设是，如何定义正面和负面的差异并不是从根本上重要的，而是由于微调数据有限这一事实。我们目前的计划引入了许多“抖动”的例子（这些提案的重叠在 0.5 和 1 之间，但不是基本事实），这使得积极的例子数量增加了大约 30 倍。我们猜想在微调整整个网络时需要这个大集合以避免过度拟合。但是，我们还注意到使用这些抖动的示例可能不是最理想的，因为网络没有针对精确定位进行微调。

这导致了第二个问题：为什么在进行微调之后，就完全训练 SVM？简单地应用微调网络的最后一层（一种 21 向软最大回归分类器）作为对象检测器将更加清晰。我们尝试了这一点，发现 VOC 2007 的性能从 mAP 的 54.2% 下降到 50.9%。这种性能下降可能源于几个因素的组合，包括微调中使用的正例的定义不强调精确定位，而 softmax 分类器是在随机抽样的负例上而不是在用于 SVM 培训的“困难负例”的子集上训练的。

该结果表明，在不训练 SVM 的情况下，可以获得接近相同的性能水平

经过微调。我们推测，通过一些额外的调整来微调剩余的性能差距可能会消失。如果为真，这将简化并加速 R-CNN 训练，而不会损失检测性能。

## C.边界框回归

我们使用简单的边界框回归阶段来提高本地化性能。在使用特定于类的检测 SVM 对每个选择性搜索提议进行评分后，我们使用特定于类的边界框回归器来预测用于检测的新边界框。这与可变形的组件模型中使用的边界框回归精神 [17] 相似。两种方法之间的主要区别在于，这里我们从 CNN 计算的特征回归，而不是从推断的 DPM 组件位置计算的几何特征回归。

我们的训练算法的输入是一组  $N$  个训练对  $\{(P^i, G^i)\}_{i=1,\dots,N}$ ，其中  $P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$  是指定中心的像素坐标生成  $P^i$  的边界框以及  $P^i$  的宽度和高度（以像素为单位）。因此，除非需要，否则我们删除上标  $i$ 。每个完全真实边界框  $G$  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拟议的方框  $P$  映射到完全真实框  $G$  的变形。

我们根据四个函数  $d_x(P)$ ,  $d_y(P)$ ,  $d_w(P)$  和  $d_h(P)$  来对变换进行参数化。前两个指定  $P$  的边界框中心的尺度不变的平移，而后两个指定  $P$  的边界框的宽度和高度的对数空间平移。在学习了这些功能之后，我们可以通过应用转换将输入提议  $P$  转换为预测的完全真实框  $G$ 。

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

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

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

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

每个函数  $d_*(P)$ （其中  $*$  是  $x, y, h, w$  之一）被建模为生成的  $P$  的 pool5 特征的线性函数，由  $\phi_5(P)$  表示。（隐含地假设  $\phi_5(P)$  对图像数据的依赖性。）因此我们得到  $d_*(P) = \mathbf{w}_*^\top \phi_5(P)$ ，其中  $\mathbf{w}_*$  是可学习的模型参数的向量。我们学习  $\mathbf{w}_*$  通过优化规范最小二乘目标（岭回归）：

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

回归目标  $t_*$  对于训练对  $(P, G)$  定义为

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

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

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

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

作为标准正则化最小二乘问题，这可以以封闭形式有效地解决。

我们在实现边界框回归时发现了两个微妙的问题。第一个是正则化很重要：我们根据验证集设置  $\lambda = 1000$ 。第二个问题是在选择要使用的训练对  $(P, G)$  时必须小心。直觉上，如果  $P$  远离所有完全真实框，则将  $P$  转换为真实框  $G$  的任务没有意义。使用像  $P$  这样的例子会导致无望的学习问题。因此，我们只从生成  $P$  中学习它是否在至少一个完全真实框附近。我们通过将  $P$  分配给具有最大 IoU 重叠的真实框  $G$ （如果它重叠多于一个）来实现“接近度”，当且仅当重叠大于阈值时（我们使用  $a$  设置为 0.6 的验证集）。所有未分配的提案都将被丢弃。我们为每个对象类执行一次，以便学习一组特定于类的边界框回归量。

在测试时，我们对每个生成进行评分并仅预测其新的检测窗口一次。原则上，我们可以迭代这个过程（即，重新对新预测的边界框进行评分，然后从中预测新的边界框，依此类推）。但是，我们发现迭代不会改善结果。

## D.附加功能可视化

图 12 显示了 20 个 pool5 单元的其他可视化。对于每个单元，我们在所有 VOC 2007 测试中显示了 24 个区域生成，这些生成最大限度地激活了整个大约 1000 万个区域中的单元。

我们通过  $6 \times 6 \times 256$  维 pool5 特征图中的  $(y, x, channel)$  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功能， $(y, x)$  位置仅改变感受野。

## E.按类别划分结果

在表 7 中，除了 O2P 方法[4]之外，我们还显示了我们的六种分割方法中每种类型的 VOC 2011 val 的每类别分割精度[4]。这些结果显示哪些方法在 20 个 PASCAL 类中的每一个中都是最强的，加上背景类。

## F.跨数据集冗余的分析

在辅助数据集上进行训练时，一个问题是它与测试集之间可能存在冗余。尽管对象检测和全图像分类的任务有很大不同，使得这种交叉设置冗余更加令人担忧，但我们仍然进行了彻底的调查，以量化 PASCAL 测试图像在 ILSVRC 2012 训练和验证集中所包含的程度。我们的研究结果可能对有兴趣使用 ILSVRC 2012 作为 PASCAL 图像分类任务的训练数据的研究人员有用。

我们对重复（和近似重复）图像执行了两次检查。第一个测试基于 flickr 图像 ID 的完全匹配，这些匹配包含在 VOC 2007 测试注释中（这些 ID 对于后续的 PASCAL 测试集有意保密）。所有 PASCAL 图像和大约一半的 ILSVRC 都是从 flickr.com 收集的。这项检查在 4952 中出现了 31 次匹配（0.63%）。

第二次检查使用 GIST [30] 描述符匹配，这在[13]中显示，在很大的 ( $> 100$  万) 图像集合中的近似重复图像检测中具有优异的性能。在[13]之后，我们计算了所有 ILSVRC 2012 trainval 和 PASCAL 2007 测试图像的扭曲  $32 \times 32$  像素版本的 GIST 描述符。

GIST 描述符的欧几里德距离最近邻匹配揭示了 38 个近似重复的图像（包括通过 flickr ID 匹配找到的所有 31 个）。匹配在 JPEG 压缩级别和分辨率方面略有不同，并且在较小程度上裁剪。这些发现表明重叠很小，不到 1%。对于 VOC 2012，由于 flickr ID 不可用，我们仅使用 GIST 匹配方法。根据 GIST 比赛，1.5% 的 VOC 2012 测试图像在 ILSVRC 2012 trainval 中。VOC 2012 的比率稍高可能是因为两个数据集的收集时间比 VOC 2007 和 ILSVRC 2012 更接近。

## G.文件更改日志

本文档追踪 R-CNN 的进展情况。为了帮助读者了解它随时间的变化，这里有一个简短的变更日志，描述了修订。

**v1** 初始版本。

**v2** CVPR 2014 相机就绪版。包括通过以下方式实现的检测性能的实质性改进：(1) 从更高的学习速率 (0.001 而不是 0.0001) 开始微调，(2) 在准备 CNN 输入时使用上下文填充，以及 (3) 边界框 回归以修复计算错误。

**v3** ILSVRC2013 检测数据集的结果以及与 OverFeat 的比较被整合到几个部分（主要是第 2 节和第 4 节）。

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

表 7: VOC 2011 验证集上的每类别分割准确度 (%)。

**v4** 附录 B 中的 softmax 与 SVM 结果包含一个已修复的错误。我们感谢 Sergio Guadar-rama 帮助确定这个问题。

**v5** 使用 Simonyan 和 Zisserman [43] 的新的 16 层网络架构到 3.3 节和表 3 添加了结果。

## 参考文献

- [1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the object-ness 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. Malik. Multiscale combinatorial grouping. In CVPR, 2014. 3
- [4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Se-mantic segmentation with second-order pooling. In ECCV, 2012. 4, 10, 11, 13, 14
- [5] J. Carreira and C. Sminchisescu. CPMC: Automatic ob-ject segmentation using constrained parametric min-cuts. TPAMI, 2012. 2, 3
- [6] D. Ciresan, 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. Fei-Fei. 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. Fei-Fei. 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 im-age search. In Proc. of the ACM International Conference on Image and Video Retrieval, 2009. 13
- [14] I. Endres and D. Hoiem. Category independent object pro-posals. 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. Felzenswalb, R. Girshick, D. McAllester, and D. Ra-manan. 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 recogni-tion unaffected by shift in position. Biological cybernetics, 36(4):193–202, 1980. 1
- [20] R. Girshick, P. Felzenswalb, and D. McAllester. Discrimi-natively 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
- [24] Y. Jia. Caffe: An open source convolutional archi-tecture for fast feature embedding. <http://caffe.berkeleyvision.org/>, 2013. 3
- [25] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet clas-sification 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. Gradient-based learning applied to document recogni-tion. 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 de-tection. In CVPR, 2013. 6, 7

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	filings 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 检测测试集上的每类平均精度 (%)。

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

A holistic representation of the spatial envelope. IJCV, 2001. 13

[30] A. Oliva and A. Torralba. Modeling the shape of the scene: [31] X. Ren and D. Ramanan. Histograms of sparse codes for

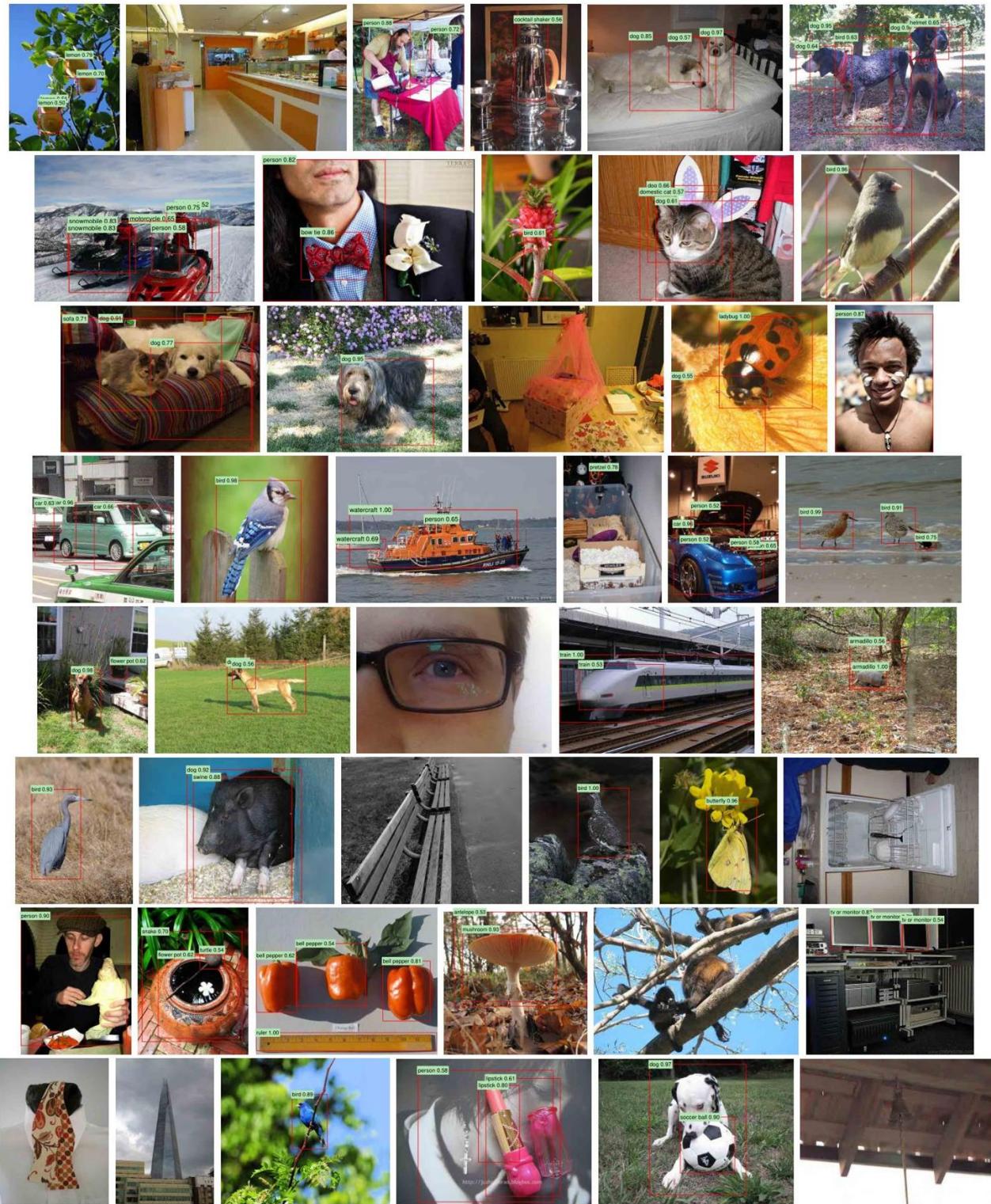


图 8: val2 上的设置检测示例, 该配置在 val2 上实现了 31.0% mAP。每个图像都是随机采样的 (这些图像不是精选的)。显示精度大于 0.5 的所有检测。每个检测都用检测器的精确回忆曲线标记预测的类别和检测的精度值。建议使用缩放以数字方式查看。

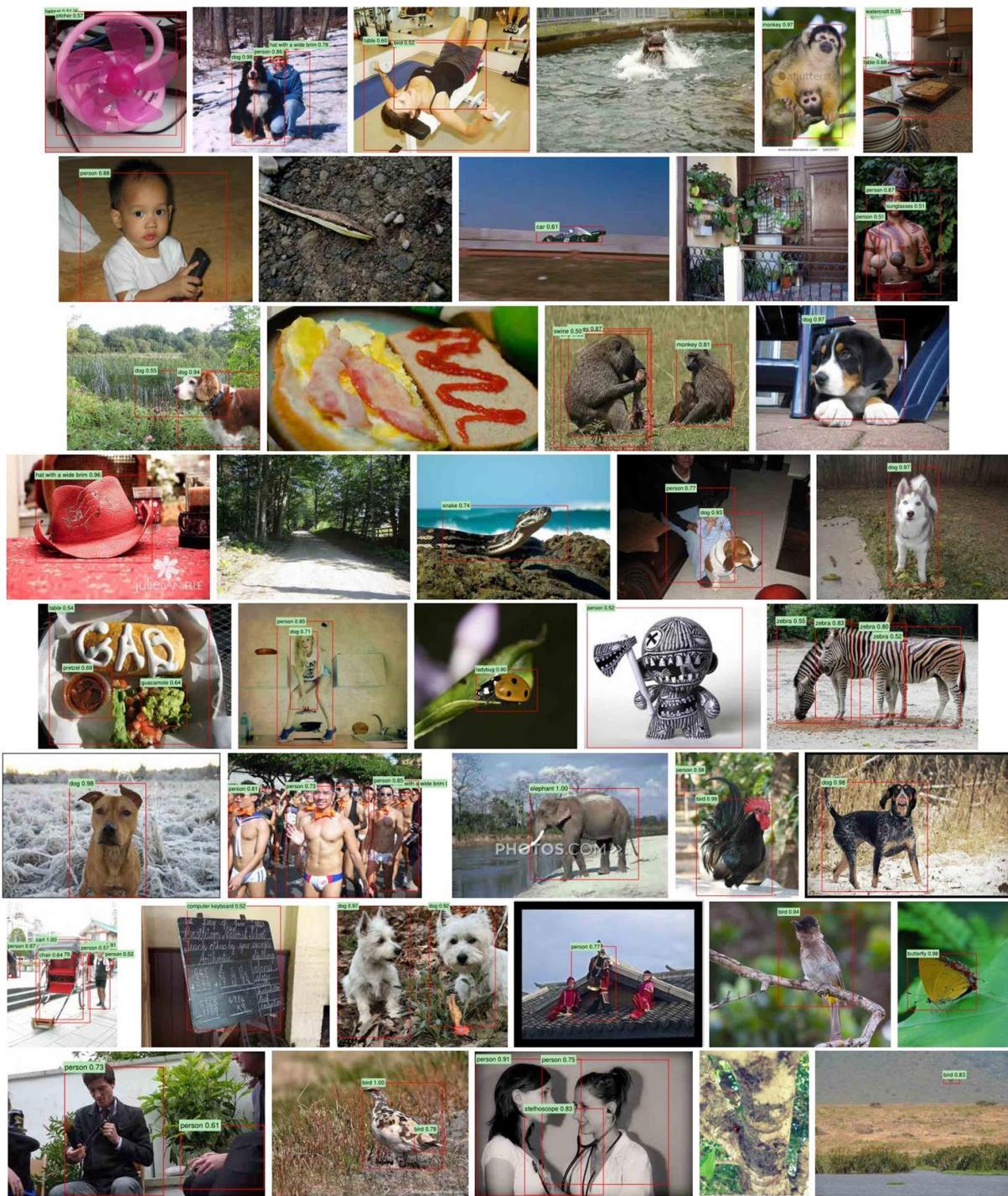


图 9: 更随机选择的示例。有关详细信息,请参见图 8 标题。建议使用缩放以数字方式查看。

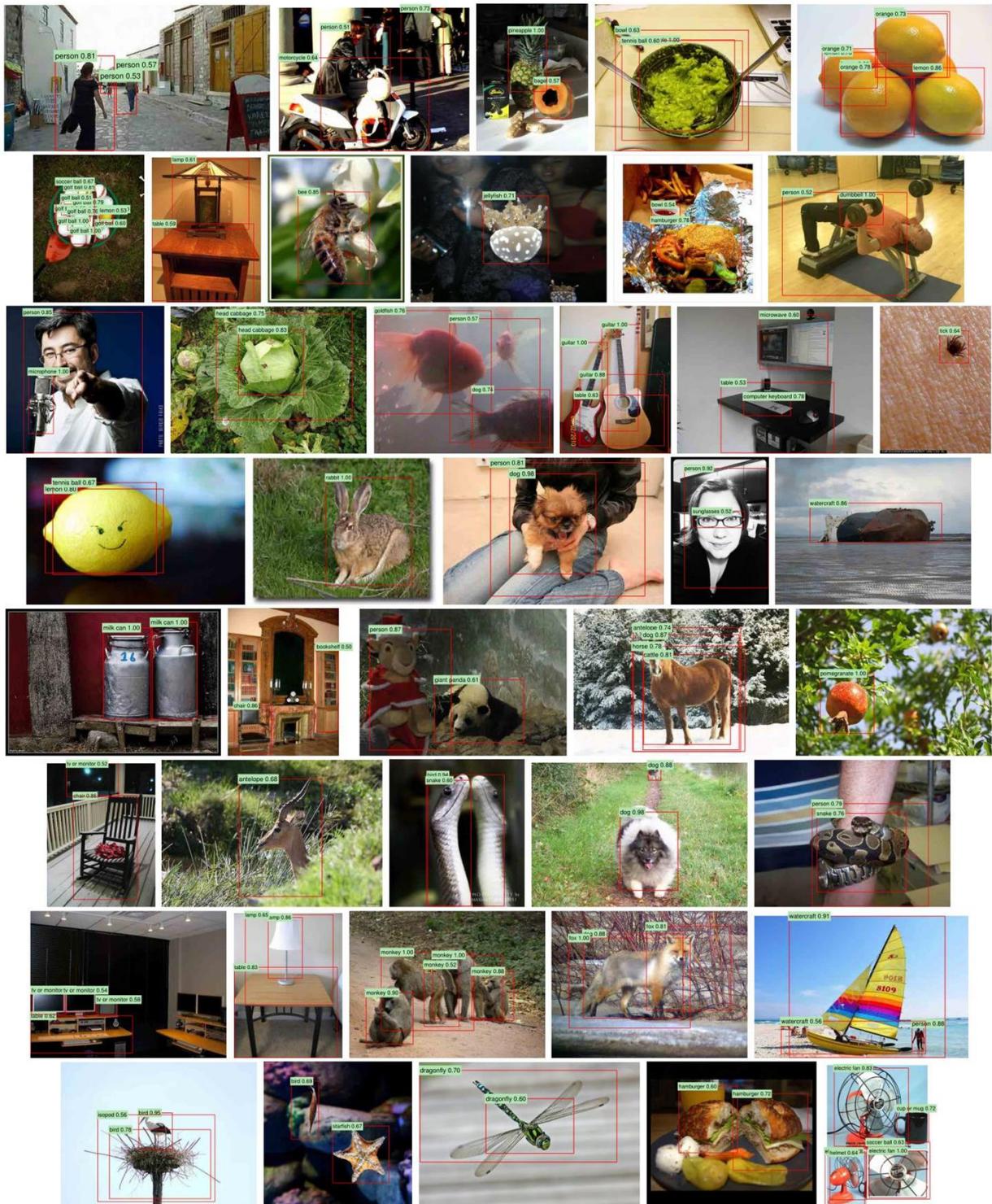


图 10: 策划示例。选择每张图片是因为我们发现它令人印象深刻, 令人惊讶, 或有趣。建议使用缩放以数字方式查看。

- object detection. In CVPR, 2013. 6, 7
- [32] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based 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 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. 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

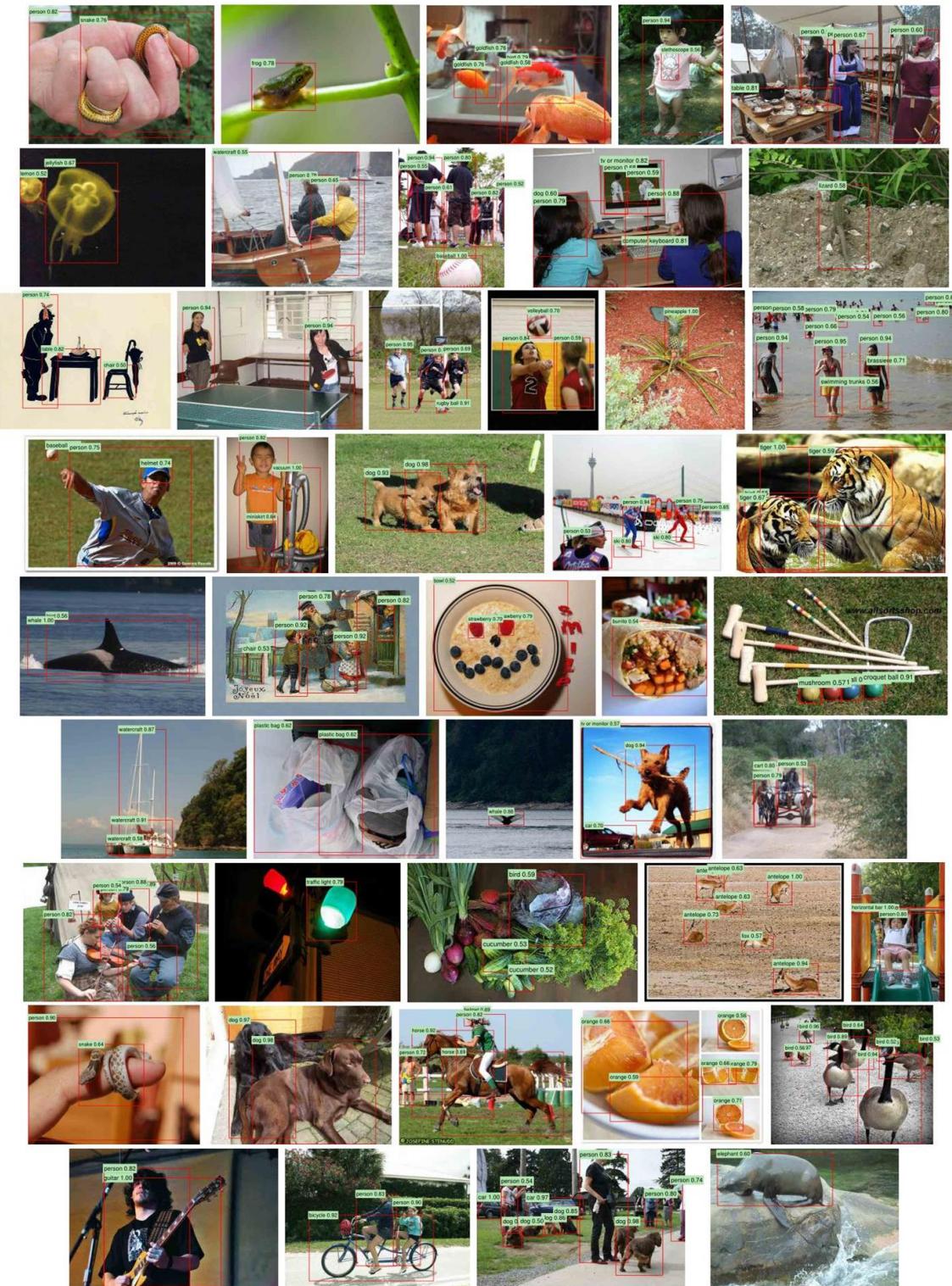


图 11: 更多精选示例。有关详细信息, 请参见图 10 标题。建议使用缩放以数字方式查看。

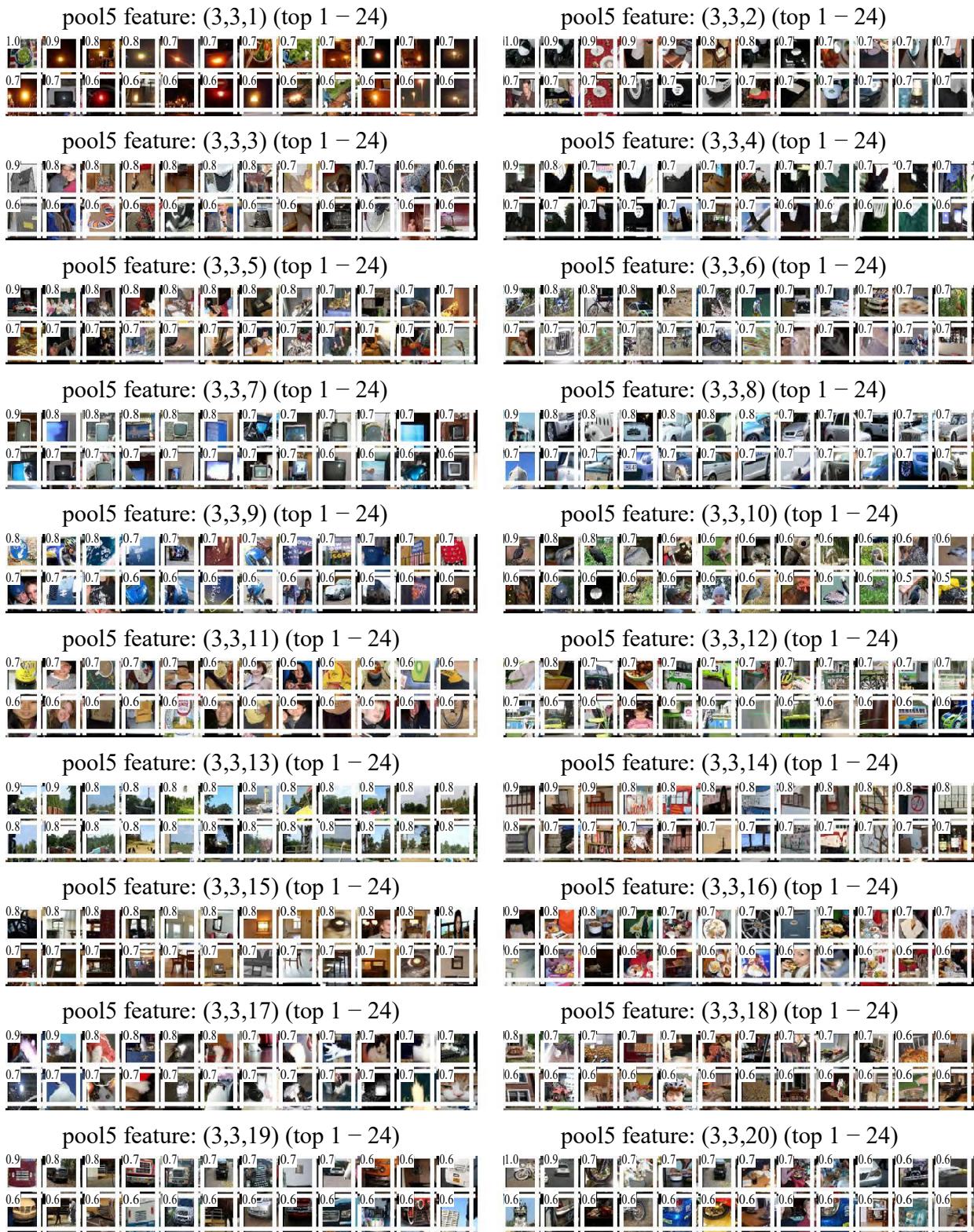


图 12: 我们展示了 VOC 2007 测试中大约 1000 万个区域中的 24 个区域生成, 它们最强烈地激活了 20 个单元中的每一个。每个蒙太奇都由  $6 \times 6 \times 256$  维池 5 特征图中的单位 ( $y, x, \text{channel}$ ) 位置标记。每个图像区域都以白色的单位感受野的覆盖图绘制。激活值 (我们通过除以通道中所有单位的最大激活值进行标准化) 显示在感知区域的左上角。建议以缩放方式以数字方式观看。