用于精确物体定位和语义分割的丰富特征层次结构

技术报告（第5版）

Ross Girshick　Jeff Donahue　Trevor Darrell　Jitendra Malik

加州大学伯克利分校

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

摘要

过去几年，在经典数据集PASCAL上，物体检测的效果已经达到一个稳定水平。效果最好的方法是融合了多种低维图像特征和高维上下文环境的复杂集成系统。在这篇论文里，我们提出了一种简单并且可扩展的检测算法，可以在VOC2012最好结果的基础上将mAP值提高30%以上——达到了53.3%。我们的方法结合了两个关键的思想：（1）在候选区域上自下而上使用大型卷积神经网络(CNNs)，用以定位和分割物体。（2）当带标签的训练数据不足时，先针对辅助任务进行有监督预训练，再进行特定任务的调优，就可以产生明显的性能提升。

因为我们把region proposal和CNNs结合起来，所以该方法被称为R-CNN：Regions with CNN features。我们也把R-CNN效果跟OverFeat比较了下，OverFeat是最近提出的基于类似的CNN特征并采用滑动窗口进行目标检测的一种方法，结果发现R-CNN在200个类别的ILSVRC2013检测数据集上的性能明显优于OVerFeat。系统完整的源代码见：http://www.cs.berkeley.edu/~rbg/rcnn。（译者注：网址已失效，github新地址：https://github.com/rbgirshick/rcnn）

1. 前言

特征的重要性。在过去十年，各类视觉识别任务基本都建立在对SIFT[29]和HOG[7]特征的使用。但如果我们关注一下PASCAL VOC对象检测[15]这个经典的视觉识别任务，就会发现2010-2012年进展缓慢，取得的微小进步都是通过构建一些集成系统和采用一些成功方法的变种才达到的。

SIFT和HOG是块方向直方图(blockwise orientation histograms)，一种类似大脑初级皮层V1层复杂细胞的表示方法。但我们知道识别发生在多个下游阶段，（我们是先看到了一些特征，然后才意识到这是什么东西）也就是说对于视觉识别来说，更有价值的信息，是层次化的，多个阶段的特征。

Fukushima的“neocognitron，一种受生物学启发用于模式识别的层次化、移动不变性模型，算是这方面最早的尝试。然而，neocognitron缺乏监督训练算法。基于Rumelhart等人的研究，Lecun等人提出反向传播的随机梯度下降(SGD)对训练卷积神经网络（CNNs）非常有效，CNNs被认为是neocognitron的一种扩展。

CNNs在1990年代被广泛使用，但之后随着SVM的崛起便淡出研究主流。2012年，Krizhevsky等人在ImageNet大规模视觉识别挑战赛(ILSVRC)上的出色表现重新燃起了对CNNs的兴趣。他们的成功在于在120万标注的图像上使用了一个大型的CNN，并且对LeCUN的CNN进行了一些改造（比如ReLU和Dropout正则化）。

这个ImangeNet的结果的重要性在ILSVRC2012 workshop上得到

了热烈的讨论。提炼出来的核心问题是：ImageNet上的CNN分类结果在何种程度上能够应用到PASCAL VOC挑战的物体检测任务上？

我们通过弥合图像分类和目标检测差别，回答了这个问题。本论文是第一个说明在PASCAL VOC的物体检测任务上CNN比基于简单的类似HOG特征的系统有大幅的性能提升。我们主要关注了两个问题：使用深度网络定位目标和在小规模的标注数据集上进行大型网络模型的训练。

与图像分类不同，目标检测需要定位图像中的目标（可能有多个）。一个方法是将框定位看做是回归问题。但Szegedy等人的研究以及我们自己的研究表明这种策略在实际应用中并不可行（在VOC2007上他们的mAP是30.5%，而我们的达到了58.5%）。另一种方法是使用滑动窗口检测器。通过这种方法使用CNNs至少已经有20年的时间了，通常用于一些特定目标种类的检测，例如人脸检测、行人检测等。为了获得较高的空间分辨率，这些CNNs普遍采用了两个卷积层和两个池化层。我们本来也考虑过使用滑动窗口的方法。但是由于我们网络有5个卷积层，具有更深的层和更多的神经元，使得输入图片有非常大的感受野（195×195）和步长（32×32），这使得采用滑动窗口的精确定位方法充满挑战。

为了解决CNN的定位，我们是通过操作”recognition using regions”范式，这种方法已经成功用于目标检测和语义分隔。测试时，每张图片产生了接近2000个与类别无关的region proposal，然后分别通过CNN提取了一个固定长度的特征向量，最后使用特定类别的线性

SVM对每个region进行分类。不论region的形状，我们使用一种简单的方法（仿射图像变形）将每个region proposal转换成固定尺寸的大小作为CNN的输入。图1展示了我们方法的全貌并突出展示了一些实验结果。由于我们的模型结合了Region proposals和CNNs，所以把这种方法称为R-CNN，即Regions with CNN features。

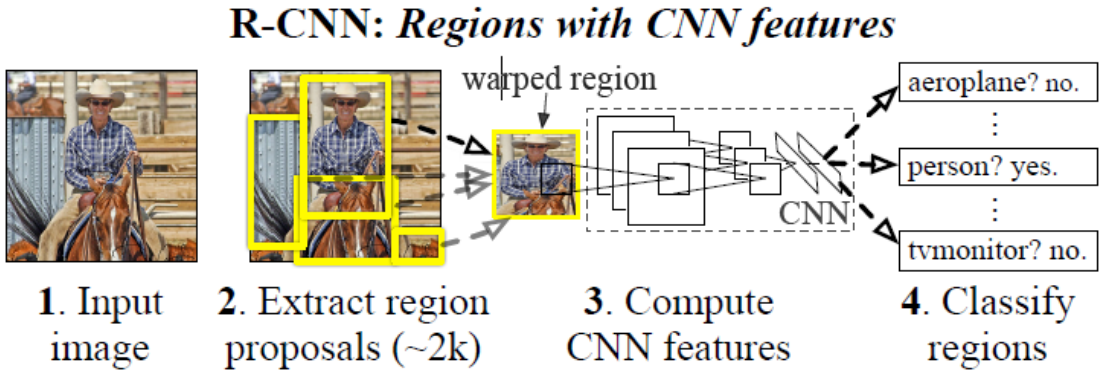

图 1：物体检测系统概述。我们的系统（1）输入一张图像、（2）提取大约 2000 个自下而上的 region proposals、（3）使用大型卷积神经网络（CNN）计算每个 region proposals 的特征向量、（4）使用特定类别的线性 SVM 对每个 region 进行分类。R-CNN 在 PASCAL VOC 2010 上的 mAP 为 53.7%。对比[39]文献中使用相同 region proposals 方法，并用使用采用空间金字塔和 bag-of-visual-words 方法的模型，其 mAP 只有 35.1%。流行的可变部件的模型的性能也只有 33.4%。在 200 个类别的 ILSVRC2013 检测数据集上，R-CNN 的 mAP 为 31.4%，比最佳结果 24.3%OverFeat 有了很大改进。

本篇论文的更新版本中，我们提供了R-CNN和最近提出的OverFeat检测系统在ILSVRC2013的200分类检测数据集上对比结果。OverFeat使用了滑动窗口CNN做检测，目前为止是ILSVRC2013检测集上表现最好的方法。我们的结果显示，R-CNN的mAP达到31.4%，显著超越了OverFeat的24.3%的结果。

检测任务第二个挑战是标签数据太少，手头可用于训练一个大型卷积网络的数据往往很不充足。传统解决方法通常采用无监督预训练，再进行有监督调优的方法（如[35]）。本文的第二个核心贡献是在辅助数据集（ILSVRC）上进行有监督预训练，再在小数据集（PASCAL）

上针对特定问题进行调优，这种方法在训练数据稀少的情况下训练大型卷积神经网络是非常有效的。我们的实验中，针对检测的调优将mAP提高了8个百分点。调优后，我们的系统在VOC2010上达到了54%的mAP，远远超过高度优化的基于HOG的可变性部件模型（deformable part model，DPM）[17, 20]。另外提醒读者朋友们关注Donahue等人同时期的工作，他们的研究表明Krizhevsky的CNN（译者注：Krizhevsky的CNN指AlexNet网络）可以用来作为一个黑盒特征提取器，（没有调优的情况下）在多个识别任务上包括场景分类、细粒度的子分类和领域自适应（domain adaptation）方面均表现很出色。

同时，我们的模型系统也很高效。都是相当小型的矩阵向量相乘和贪婪的非极大值抑制这些特定类别的计算。这个计算特性源自于所有类别都共享的特征，同时这些特征比之前使用的区域特征（[39]）少了两个数量级的维度。

分析我们方法的失败案例对进一步改进和提升很有帮助，所以我们借助Hoiem等人的定位分析工具[23]做实验结果的报告和分析。作为本次分析的直接结果，我们发现一个简单的边界框回归的方法会明显地降低错误定位问题，而错误定位是我们的模型系统的主要误差。

开发技术细节之前，我们注意到由于R-CNN是在推荐区域上进行操作，所以可以很自然地扩展到语义分割任务上。经过微小的改动，我们就在PASCAL VOC语义分割任务上达到了很有竞争力的结果，在VOC 2011测试集上平均语义分割精度达到了47.9%。

2. 使用 R-CNN 做物体检测

我们的物体检测系统有三个模块构成。第一个模块产生类别无关的region proposals。这些proposals组成了一个模型可用的候选检测区域的集合。第二个模块是一个大型卷积神经网络，从每个region提取固定长度的特征向量。第三个模块是特定类别线性SVM的集合。这一节将展示每个模块的设计，并介绍它们的测试阶段的用法，以及一些参数学习的细节，并得出在PASCAL VOC 2010-12和ILSVRC2013上的检测结果。

2.1 模块设计

区域推荐（Region Proposals）。近来有很多研究都提出了生成类别无关的区域推荐的方法。比如objectness [1]、selective search [39]、category-independent object proposals [14]、constrained parametric min-cuts (CPMC) [5]、multi-scale combinatorial grouping [3]，以及Ciresan等人提出的将CNN用在规律空间块裁剪上以检测有丝分裂细胞的方法，也算是一种特殊的区域推荐类型。由于R-CNN对特定区域算法是不关心的，所以我们采用了选择性搜索以方便和之前的工作[39, 41]进行可控的比较。

特征提取。我们使用Krizhevsky等人[25]所描述的CNN（译者注：AlexNet）的一个Caffe[24]实现版本对每个推荐区域提取一个4096维度的特征向量。减去像素均值的277×277大小的RGB输入图像通过五个卷积层和两个全连接层，最终计算得到特征向量。读者可以参考[24, 25]获得更多的网络架构细节。

为了计算推荐区域的特征，首先需要将输入的图像数据进行转变，使得推荐的区域变成CNN可以接受的方式（我们架构中的CNN只能接受像素宽高比为227*227固定大小的图像）。有很多种方法可以对我们任意形状的区域进行变换，我们选择了最简单的一种。无论候选区域是什么尺寸或者任意长宽比，我们将区域放入无缝的边框内变形到希望的尺寸。变形之前，先放大紧边框以便在新的变形后的尺寸上保证变形图像上下文的p的像素都围绕在原始框上（我们使用p=16）。图2展示了一些变形训练图像的例子。其它的变形方法可以参考附录A。

图 2：VOC 2007 训练集部分图像变形成训练样本。

2.2 测试阶段的物体检测

在测试阶段，在测试图像上使用selective search提取2000个推荐区域（所有实验中我们使用了selective search的加速版本）。对每一个推荐区域变形后通过CNN前向传播计算出特征。然后我们使用训练过特定类别的SVM给特征向量中的每个类别单独打分。然后给出一张图像中所有的打分区域，然后使用贪婪非最大值抑制算法（每个类别是独立进行的）舍弃那些与大于学习阈值更高得分的推荐区域有重叠（intersection-overunion (IoU)）的区域。

运行时分析。两个特性让检测变得很高效。首先，所有的CNN参数都是跨类别共享的。其次，通过CNN计算的特征向量维度相比其他

常见方法（比如spatial pyramids with bag-of-visual-word encodings）计算特征的维度是很低的。例如，UVA检测系统[39]中使用的特征比我们的要多两个数量级(360k维相比于4k维)。

这种共享的结果就是计算推荐区域和特征的耗时可以分摊到所有类别上（GPU：每张图13s，CPU：每张图53s）。唯一和类别有关的计算都是特征和SVM权重以及最大化抑制之间的点积。实际应用中，所有的点积都可以批量化成一个单独矩阵间运算。特征矩阵的通常是2000×4096（译者注：通常产生2000个左右的推荐区域，每个推荐区域经过CNN产生4096维的向量），SVM权重的矩阵是4096*N，其中N是类别数目。

分析表明R-CNN可以扩展到上千个类别，而不需要诉诸近似技术，例如hashing。即使有10万个类别，计算矩阵乘法在现代多核CPU上只需要10s而已。这种高效不仅仅是因为使用了区域推荐和共享的特征。，UVA系统由于其高维特征需要134GB的内存来存10万个线性预测器，而我们只要1.5GB，这使得UVA系统比我们慢了两个数量级。

更有趣的是R-CCN和最近Dean等使用了DPMs和hashing[8]进行大规模检测任务对比。当他们用了1万个干扰类时每五分钟可以处理一张图片，在VOC2007上的mAP能达到16%。我们的方法1万个检测器由于没有做近似，可以在CPU上一分钟跑完，达到59%的mAP（3.2节）。

2.3 训练

有监督预训练。我们仅使用图像级注释的大型辅助数据集

（ILSVRC2012分类任务）上有区别地预训练了CNN（该数据集没有边界框标签）。预训练采用了开源的Caffe CNN库[24]。简单地说，我们的CNN十分接近krizhevsky等人网络的性能，在ILSVRC2012分类验证集上top-1错误率比他们高2.2%。差异主要来自于训练过程的简化。

特定领域的参数调优。为了让我们的CNN适应新的任务（即检测任务）和新的领域（变形后的推荐窗口），我们只使用变形后的推荐区域对CNN参数进行SGD训练。我们替掉了ImageNet特定的1000类分类层，换成了一个随机初始化的(N+1)类的分类层（其中N是目标类别数目，1代表背景），而卷积层架构没有改变。对于VOC，N=20，对于ILSVRC2013，N=200。对于所有的推荐区域，如果与真实标注框的IoU重叠大于等于0.5就认为该推荐区域代表的类是正例，否则就是负例。SGD初始学习率为0.001（初始化预训练时的十分之一），这使得调优得以有效进行而不会破坏初始化的成果。每轮SGD迭代，我们统一使用32个正例窗口（跨所有类别）和96个背景窗口组成大小为128的mini-batch。另外我们倾向于采样正例窗口，因为和背景相比他们很稀少。

目标类别分类器。思考一下检测汽车的二分类器。很显然，一个图像区域紧紧包裹着一辆汽车应该就是正例。相似的，背景区域应该看不到任何汽车，就是负例。较为不明晰的是怎样标注哪些只和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，低于这个阈值的就是负例。这个阈值我们选择了0.3，是在验证集上基于{0, 0.1, … 0.5}通过网格搜索得到的。我们发现认真选择这个阈值很重要。

如果设置为0.5，如[39]，可以提升mAP5个点，设置为0，就会降低4个点。正例就严格的是标注的框。

一旦特征提取出来，就应用标签数据，然后优化每个类的线性SVM。由于训练数据太大，难以装进内存，我们选择了标准的hard negative mining method（高难负例挖掘算法？用途就是正负例数量不均衡，而负例分散代表性又不够的问题）[17, 37]。 高难负例挖掘算法收敛很快，实践中只要经过一轮mAP就可以基本停止增加了。

附录B中，我们讨论了，正例和负例在调优和SVM训练两个阶段的为什么定义得如此不同。我们也会讨论训练检测SVM的平衡问题，而不只是简单地使用来自调优后的CNN的最终softmax层的输出。

2.4. 在 PASCAL VOC 2010-12 上的结果

按照PASCAL VOC的最佳实践步骤，我们在VOC2007的数据集上验证了我们所有的设计思想和参数处理，对于在2010-2012数据库中，我们在VOC2012上训练和优化了我们的支持向量机检测器，我们一种方法（带BBox和不带BBox）只提交了一次评估服务器。

表1展示了（本方法）在VOC2010的结果，我们将自己的方法同四种先进基准方法作对比，其中包括SegDPM，这种方法将DPM检测子与语义分割系统相结合并且使用附加的内核的环境和图片检测器打分。更加恰当的比较是同Uijling的UVA系统比较，因为我们的方法同样基于候选框算法。对于候选区域的分类，他们通过构建一个四层的金字塔，并且将之与SIFT模板结合，SIFT为扩展的OpponentSIFT和RGB-SIFT描述子，每一个向量被量化为4000词的codebook。分类任务

由一个交叉核的支持向量机承担，对比这种方法的多特征方法，非线性内核的SVM方法，我们在mAP达到一个更大的提升，从35.1%提升至53.7%，而且速度更快。我们的方法在VOC2011/2012数据达到了相似的检测效果mAP53.3%。

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

表 1：VOC 2010 测试集上的检测平均精度（%）。R-CNN 与 UVA 和 Regionlet 直接比较，因为所有方法都使用 selective search 的 region proposals 方法。Bounding-box 回归（BB）在 C 节中进行了描述。本文发布时，SegDPM 是 PASCAL VOC 排行榜上性能最优的算法。DPM 和 SegDPM 使用其他方法未使用的 context rescoring。

2.5. 在 ILSVRC2013 上检测任务结果

我们使用与用于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，仅将测试结果提交给ILSVRC2013评估服务器两次，一次带有边界框回归，一次带没有边界框回归。

图3将R-CNN与ILSVRC 2013竞赛中的参赛作品以及竞赛后的OverFeat结果进行了比较[34]。 R-CNN的mAP达到31.4%，大大超过了OverFeat的第二佳结果24.3%。 为了让您了解AP在各个类别中的分布情况，还提供了箱形图，并在表8的末尾列出了每个类别的AP。大多数竞争者（OverFeat，NEC-MU，UvAEuvision，Toronto A， 和UIUC-IFP）使用了卷积神经网络，这表明CNN如何应用于目标检测有很大的细微差别，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。

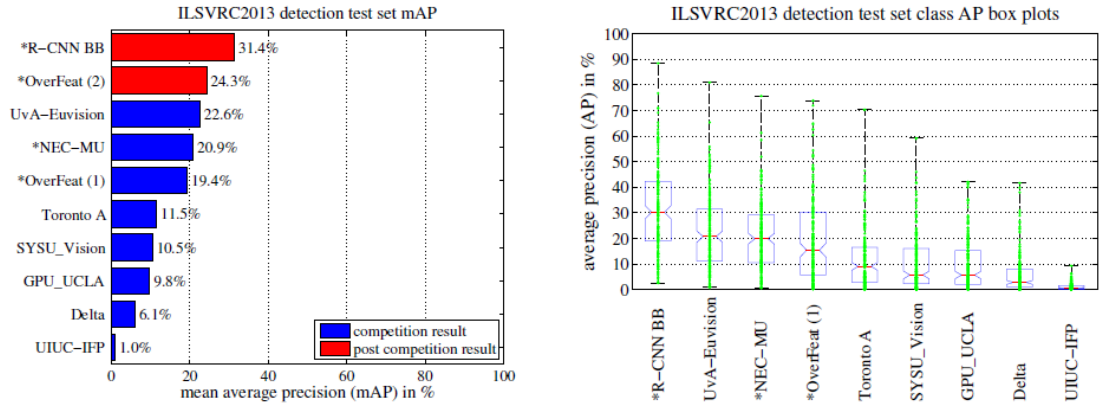

图 3 ：（左图）ILSVRC2013 检测测试集的 mAP。*开头的方法使用外部训练数据（所有方法都使用 ILSVRC 分类数据集中的图像和标签）。**（右图）每种方法的 200 个平均精度值的箱形图。**竞赛后的 OverFeat 结果的箱形图未显示，因为无法获得按类别的 AP（R-CNN 按类别的 AP 在表 8 中，并且也包含在上传到 arXiv.org 的技术报告资源中；详细见：*R-CNN-ILSVRC2013-APs.txt*）。红线标记 AP 的中位数，方框的底部和顶部分别是第 25 个和第 75 个百分点。whiskers 扩展到每种方法的 AP 最小值和最大值。将每个 AP 绘制为 whiskers 上的绿点（最好通过电子版缩放进行查看）（译者注：需要将右图放大看，否则可能看不清楚）。

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

表 8：ILSVRC 2013 检测测试集上的每类平均精度（%）。

3. 可视化、消融研究和模型误差

3.1. 可视化学习到的特征

直接可视化第一层特征过滤器非常容易理解[25]，它们主要捕获方向性边缘和对比色。难以理解的是后面的层。Zeiler and Fergus提出了一种可视化的很棒的反卷积办法[42]。我们则使用了一种简单的非参数化方法，直接展示网络学到的东西。这个想法是单一输出网络中

一个特定单元（特征），然后把它当做一个正确类别的物体检测器来使用。

方法是这样的，先计算所有抽取出来的推荐区域（大约1000万），计算每个区域所导致的对应单元的激活值，然后按激活值对这些区域进行排序，然后进行最大值抑制，最后展示分值最高的若干个区域。这个方法让被选中的单元在遇到他想激活的输入时“自己说话”。我们避免平均化是为了看到不同的视觉模式和深入观察单元计算出来的不变性。

我们可视化了第五层的池化层pool5，是卷积网络的最后一层，feature_map(卷积核和特征数的总称)的大小是6 x 6 x 256 = 9216维。忽略边界效应，每个pool5单元拥有195×195的感受野，输入是227×227。pool5中间的单元，几乎是一个全局视角，而边缘的单元有较小的带裁切的支持。

图4的每一行显示了对于一个pool5单元的最高16个激活区域情况，这个实例来自于VOC 2007上我们调优的CNN，这里只展示了256个单元中的6个（附录D包含更多）。我们看看这些单元都学到了什么。第二行，有一个单元看到狗和斑点的时候就会激活，第三行对应红斑点，还有人脸，当然还有一些抽象的模式，比如文字和带窗户的三角结构。这个网络似乎学到了一些类别调优相关的特征，这些特征都是形状、纹理、颜色和材质特性的分布式表示。而后续的fc6层则对这些丰富的特征建立大量的组合来表达各种不同的事物。

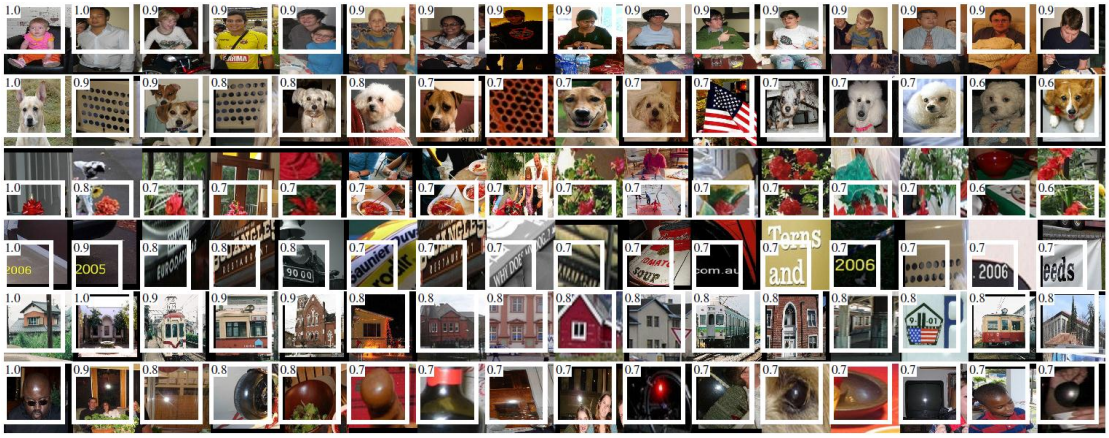

图 4：六个 pool5 神经元的顶部区域。感受野和激活值以白线绘制。有些神经元与概念保持一致，例如人（第 1 行）或文本（第 4 行）。 其他单元捕获纹理和材料属性，例如点阵列（第 2 行）和镜面反射（第 6 行）。

3.2. 消融研究

没有调优的各层性能。为了理解哪一层对于检测的性能十分重要，我们分析了CNN最后三层的每一层在VOC2007上面的结果。Pool5在3.1中做过剪短的表述。最后两层下面来总结一下。

fc6是一个与pool5连接的全连接层。为了计算特征，它和pool5的feature map（reshape成一个9216维度的向量）做了一个4096×9216的矩阵乘法，并添加了一个bias向量。中间的向量是逐个组件的半波整流（component-wise half-wave rectified）ReLU(x <– max(0,="" x))。<="" p="">

fc7是网络的最后一层。跟fc6之间通过一个4096×4096的矩阵相乘。也是添加了bias向量和应用了ReLU。

我们先来看看没有调优的CNN在PASCAL上的表现，没有调优是指所有的CNN参数就是在ILSVRC2012上训练后的状态。分析每一层的性能显示来自fc7的特征泛化能力不如fc6的特征。这意味29%的CNN参数，也就是1680万的参数可以移除掉，而且不影响mAP。更多

的惊喜是即使同时移除fc6和fc7，仅仅使用pool5的特征，只使用CNN参数的6%也能有非常好的结果。可见CNN的主要表达力来自于卷积层，而不是全连接层。这个发现提醒我们也许可以在计算一个任意尺寸的图片的稠密特征图（dense feature map）时使仅仅使用CNN的卷积层。这种表示可以直接在pool5的特征上进行滑动窗口检测的实验。

调优后的各层性能。我们来看看调优后在VOC2007上的结果表现。提升非常明显，mAP提升了8个百分点，达到了54.2%。fc6和fc7的提升明显优于pool5，这说明pool5从ImageNet学习的特征通用性很强，在它之上层的大部分提升主要是在学习领域相关的非线性分类器。

对比其他特征学习方法。相当少的特征学习方法应用与VOC数据集。我们找到的两个最近的方法都是基于固定探测模型。为了参照的需要，我们也将基于基本HOG的DFM方法的结果加入比较。

第一个DPM的特征学习方法，DPM ST,将HOG中加入略图表征的概率直方图。直观的，一个略图就是通过图片中心轮廓的狭小分布。略图表征概率通过一个被训练出来的分类35*35像素路径为一个150略图表征的的随机森林方法计算。

第二个方法，DPM HSC，将HOG特征替换成一个稀疏编码的直方图。为了计算HSC（HSC的介绍略）

所有的RCNN变种算法都要强于这三个DPM方法（表2 8-10行），包括两种特征学习的方法（特征学习不同于普通的HOG方法？）与最新版本的DPM方法比较，我们的mAP要多大约20个百分点，61%的相对提升。略图表征与HOG现结合的方法比单纯HOG的性能高出2.5%，

而HSC的方法相对于HOG提升四个百分点（当内在的与他们自己的DPM基准比价，全都是用的非公共DPM执行，这低于开源版本）。这些方法分别达到了29.1%和34.3%。

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

表 2：VOC 2007 测试集上的检测平均精度（%）。第 1-3 行显示了没有进行 fine-tuning 的 R-CNN 性能。第 4-6 行显示了在 ILSVRC 2012 上进行了预训练并在 VOC 2007 上进行了 fine-tuning（FT）的 CNN 的结果。第 7 行包括一个简单的 bounding-box 回归（BB）阶段，可减少定位错误（详见 C 节）。第 8-10 行显示将 DPM 方法作为强 baseline 的结果。前一种仅使用 HOG，而后两种使用不同的特征学习方法来增强或替代 HOG。

3.3. 网络架构

本文中的大部分结果所采用的架构都来自于Krizhevsky等人的工作[25]。然后我们也发现架构的选择对于R-CNN的检测性能会有很大的影响。表3中我们展示了VOC2007测试时采用了16层的深度网络，由Simonyan和Zisserman[43]刚刚提出来。这个网络在ILSVRC2014分类挑战上是最佳表现。这个网络采用了完全同构的13层3×3卷积核，中间穿插了5个最大池化层，顶部有三个全连接层。我们称这个网络为O-Net表示OxfordNet，将我们的基准网络称为T-Net表示TorontoNet。

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

表 3：两种不同 CNN 架构在 VOC 2007 测试集上的检测平均精度（%）。前两行源于表 2 中使用 Krizhevsky 等人架构（T-Net）的结果。第三和第四行使用 Simonyan 和 Zisserman（O-Net）[43]最近提出的 16 层架构。

为了使用O-Net，我们从Caffe模型库中下载了他们训练好的权重

VGG_ILSVRC_16_layers。然后使用和T-Net上一样的操作过程进行调优。唯一的不同是使用了更小的Batch Size（24），主要是为了适应GPU的内存。表3中的结果显示使用O-Net的R-CNN表现优越，将mAP从58.5%提升到了66.0%。然后它有个明显的缺陷就是计算耗时。O-Net的前向传播耗时大概是T-Net的7倍。

3.4. 检测误差分析

为了揭示出我们方法的错误之处，我们使用Hoiem提出的优秀的检测分析工具，来理解fine-tuning是怎样改变他们，并且观察相对于DPM方法，我们的错误形式（译者注：即错误的各种来源）。这个分析方法全部的介绍超出了本篇文章的范围，我们建议读者查阅文献23来了解更加详细的介绍（例如归一化AP的介绍），由于这些分析是不太有关联性，所以我们放在图4和图5的题注中讨论。

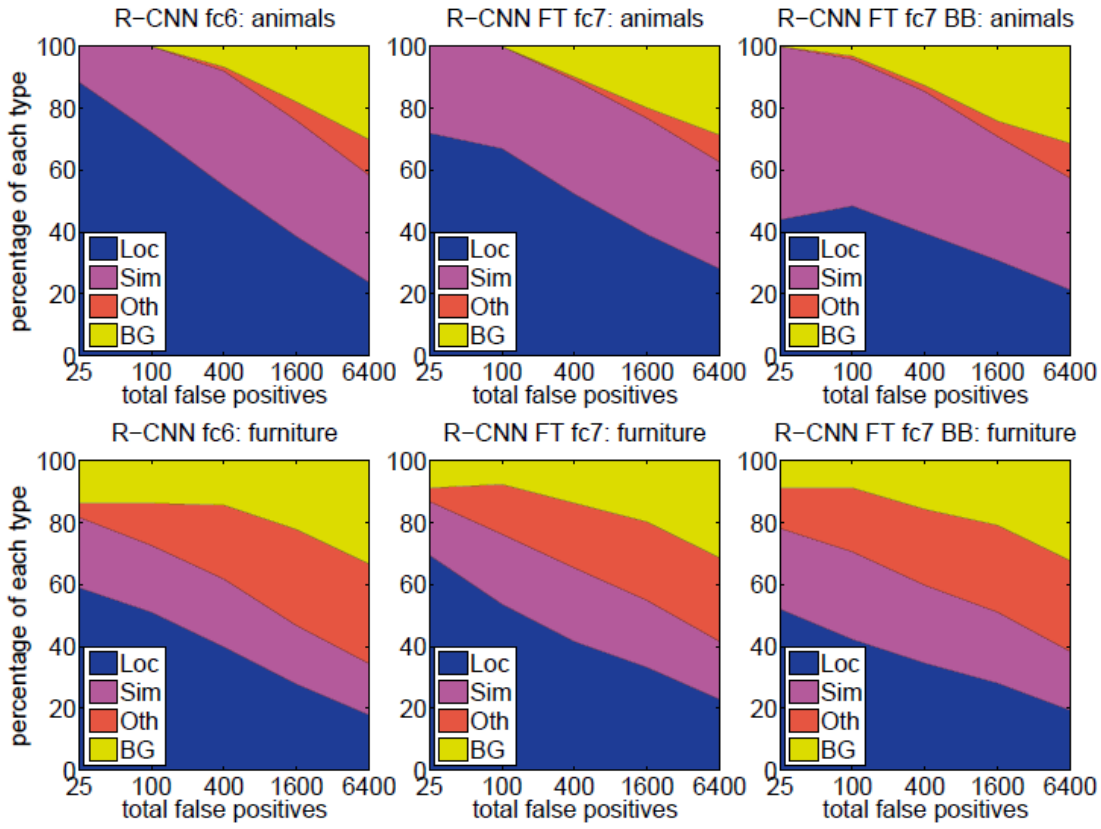

图 5：排名最高的假阳性（FP）类型分布。每个图都显示了 FP 类型的演变分布，因为按照得分递减的顺序考虑了更多 FP。每个 FP 分为以下 4 种类型：Loc——定位较差（检测结果与正确类别的 IoU 重叠在 0.1 和 0.5 之间，或重复）；Sim——与相似类别混淆；Oth——与非相似类别混淆；BG——将背景当作检测目标的假阳性。与 DPM 相比（参见[23]），我们的错误明显更多是由于错误定位所致，而不是与背景或其他对象类别造成的混淆，这表明 CNN 特征比 HOG 具有更大的判别力。宽松的定位很可能是由于我们使用了自下而上的 region proposals，以及从对神经网络进行整个图像分类的预训练中学到的位置不变性。第三列显示了我们的简单 bounding-box 回归方法如何解决许多定位错误。

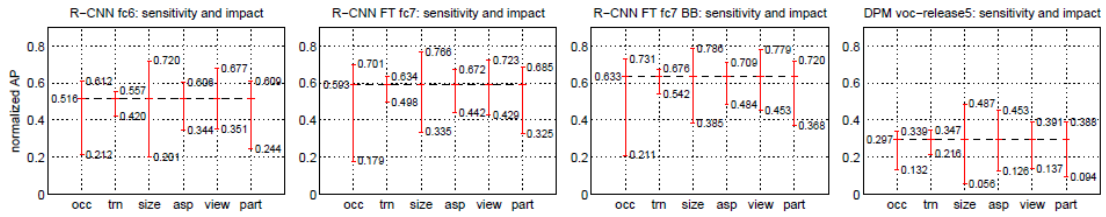

图6：目标特性的敏感性。每个图都显示了六个不同目标特征（遮挡、截断、bounding-box区域、长宽比、viewpoint、部分可见性）子集性能的最高和最低均值（基于每个类别）标准化AP（详见[23]）。我们展示了进行和没有进行fine-tuning（FT）、bounding-box回归（BB）以及DPM voc-release5的方法（R-CNN）的结果图。总体而言，fine-tuning没有降低灵敏度（最大值和最小值之间的差异），但会实质上改善几乎所有特性子集的最高和最低性能。这表明，fine-tuning不仅可以改善长宽比和bounding-box区域的性能最低的子集，还可以根据我们扭曲网络输入的方式来推测。相反，fine-tuning可以提高所有特性的鲁棒性，包括遮挡、截断、viewpoint,和部分可见性。

3.5. Bounding-box 回归

基于对误差的分析，我们使用了一种简单的方法减小定位误差。受到 DPM[17]中使用的 bounding-box 回归的启发，我们训练了一个线性回归模型在给定一个选择区域的 $pool_5$ 特征时去预测一个新的检测窗口。详细的细节参考附录 C。表 1、表 2 和图 5 的结果说明这个简单的方法，改善了大量的错误定位检测结果，mAP 提升了 3-4 个百分点。

3.6. 定性结果

ILSVRC2013 数据集上的定性检测结果如文章末尾图 8 和图 9 所示。从 val2 集中随机采样每个图像，并显示所有检测器的所有检测结

果，其精度均大于 0.5。请注意这些不是经过精心策划的，它们会给实际使用中的检测器以真实的印象。图 10 和图 11 给出了更多的定性结果，但是这些结果已经得到确认。我们选择每张图像是因为它包含有趣，令人惊讶或有趣的结果。此处，还显示了所有精度大于 0.5 的检测。

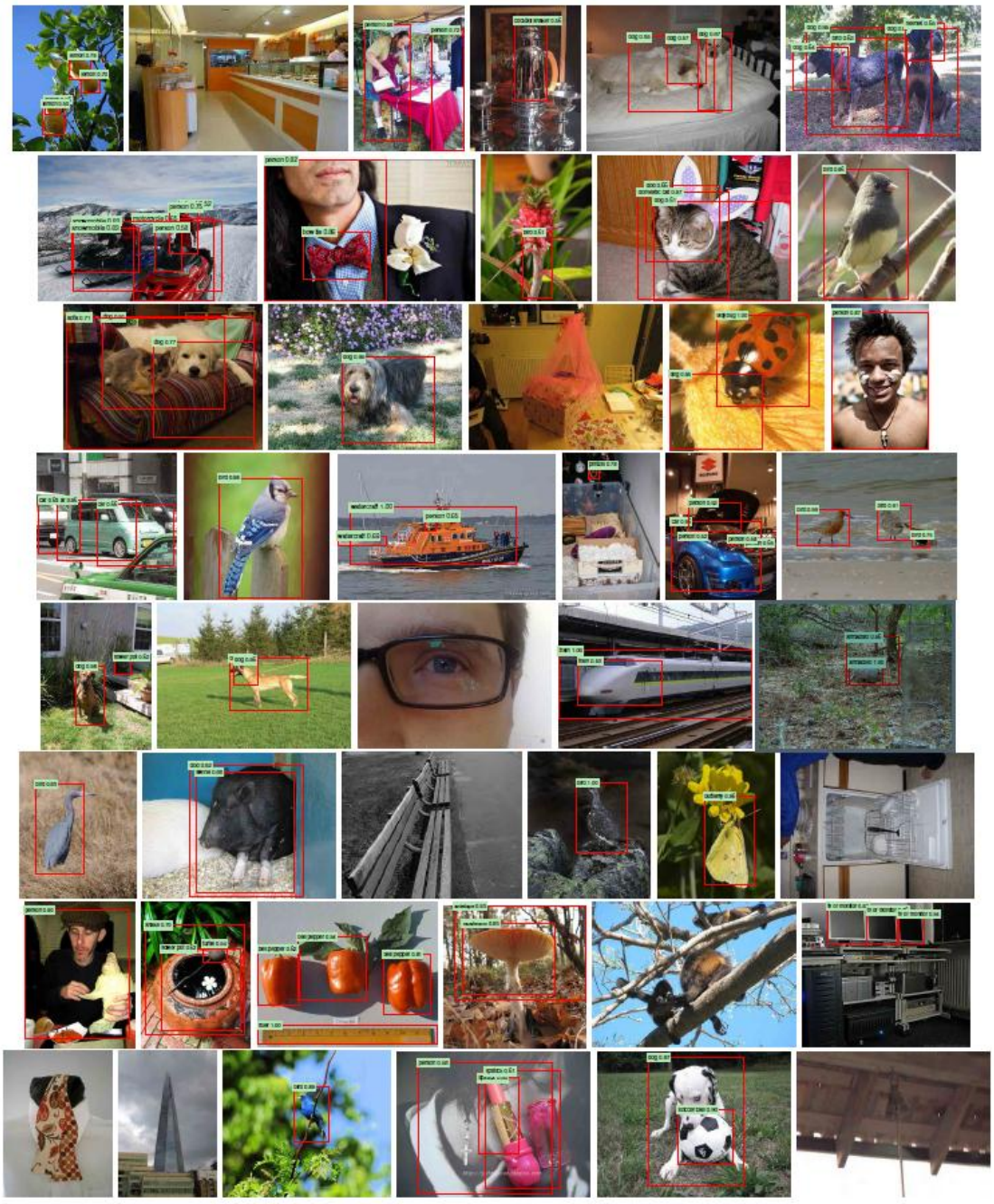

图 8：通过在 val2 上达到 31.0％ mAP 的配置对 val2 数据子集进行检测的结果。

每个图像都是随机采样的（这些图像不是精心挑选的）。显示了所有精度大于 0.5 的检测结果。每个检测器都标记有预测类别和检测器的 precision-recall 曲线中的检测值的精确度。建议数字化放大观看。

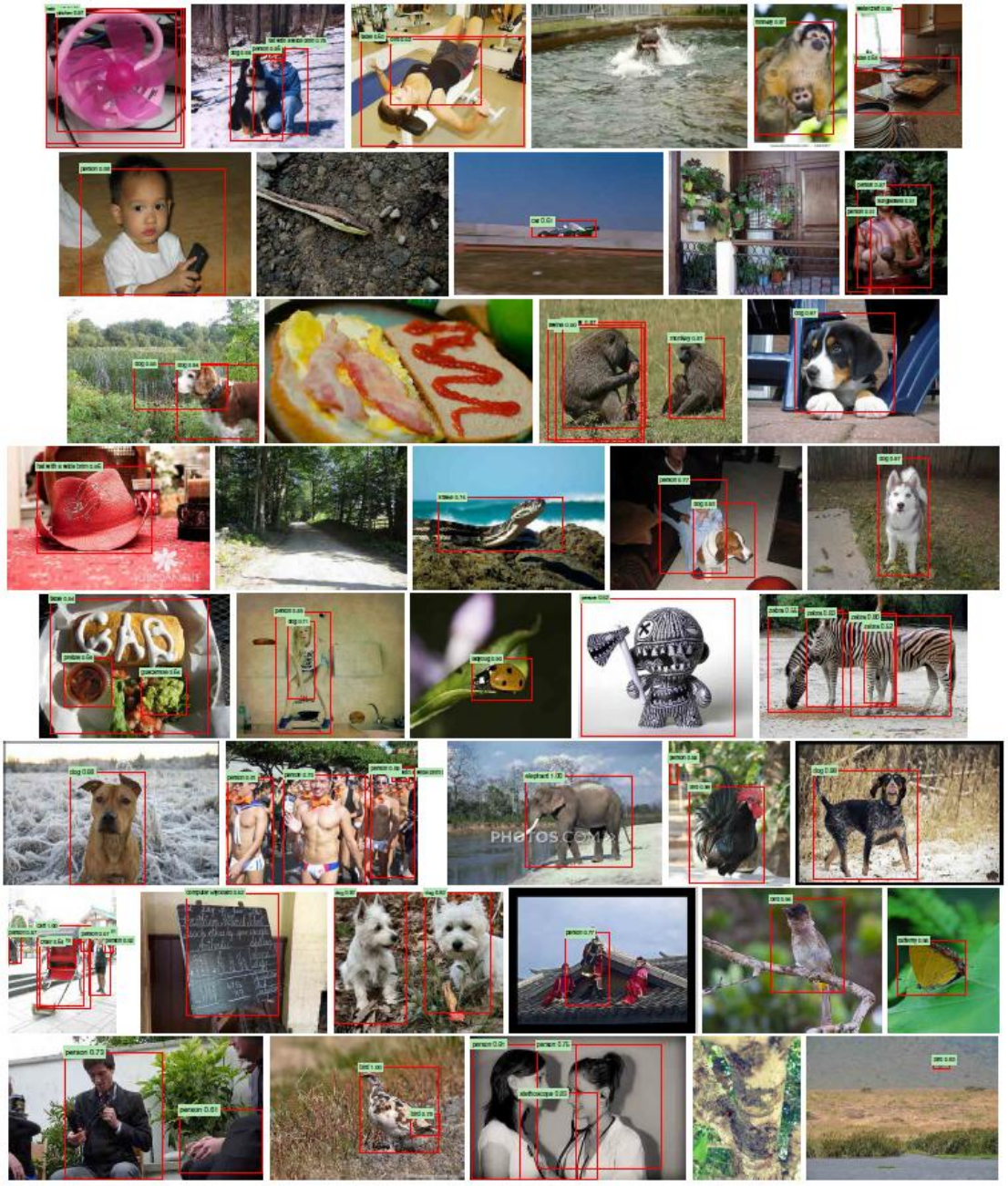

图 9：更多随机选择的示例。详细介绍信息见图 8 标题。建议数字化放大观看。

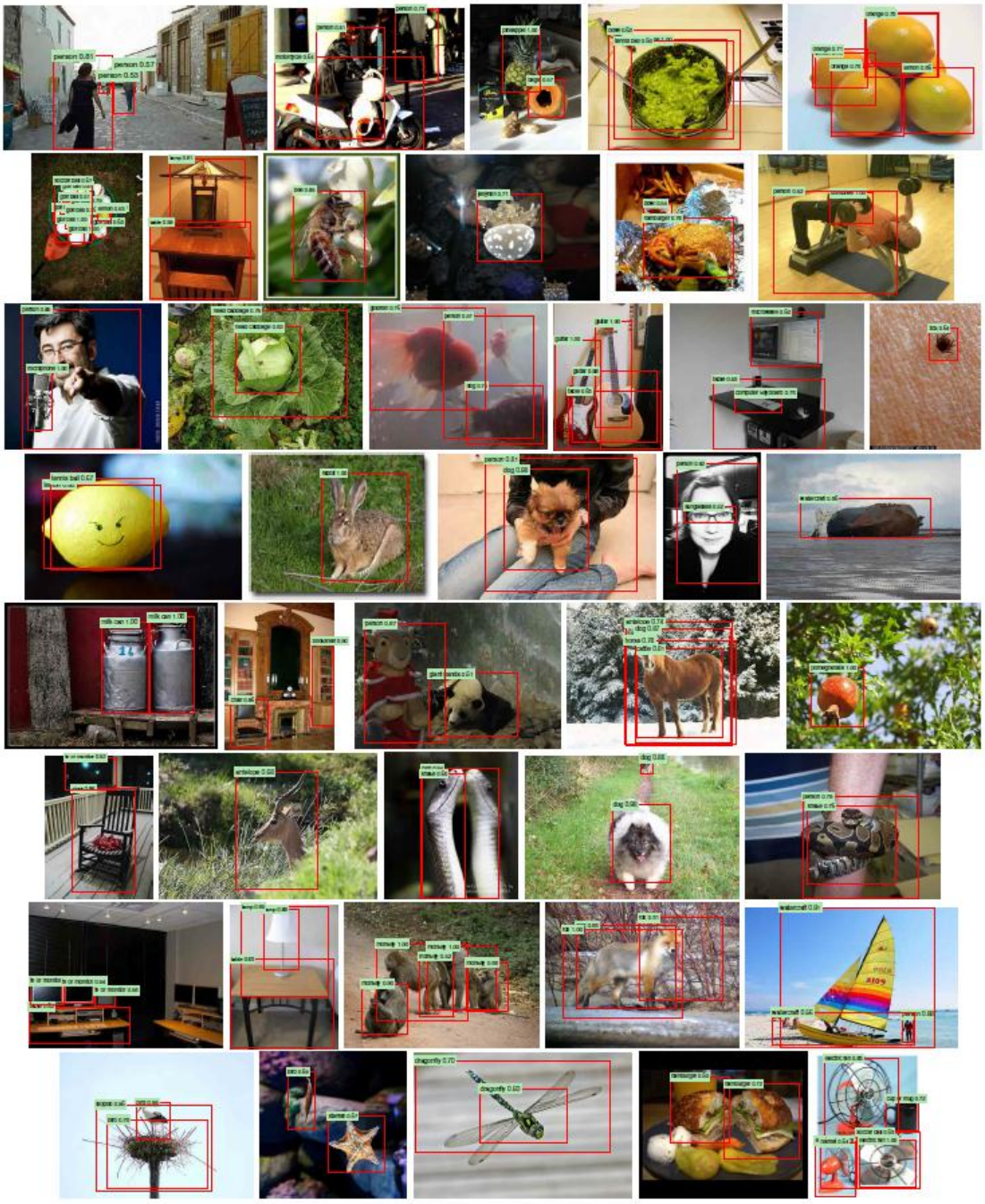

图 10：精心挑选的结果示例。选择每张图片是因为我们发现它令人印象深刻、令人惊讶、有趣。建议数字化放大观看。

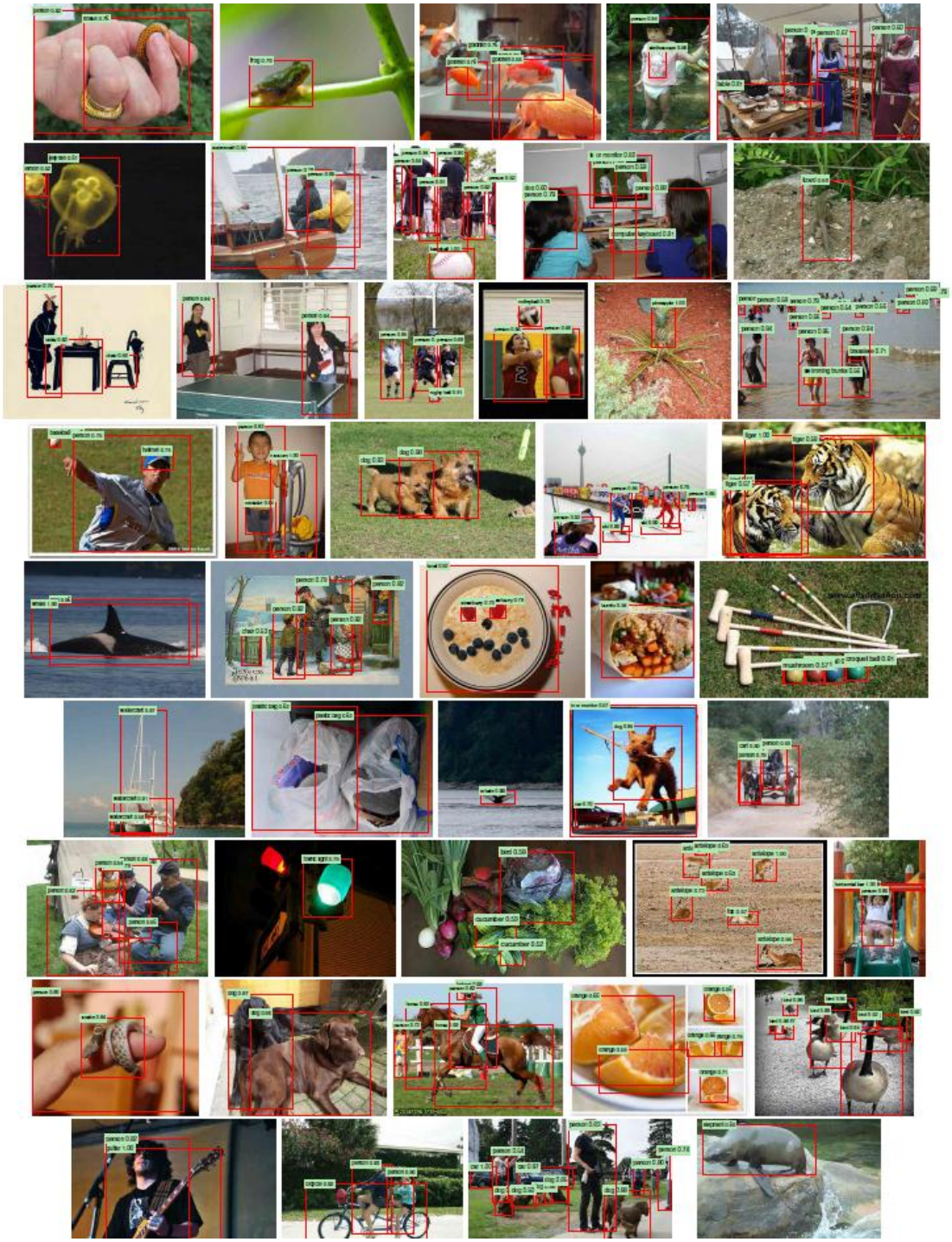

Figure 11: 更多精心挑选的例子。详细介绍见图 10 标题。建议数字化放大观看。

4. ILSVRC2013 检测数据集

在第 2 节中，我们展示了 ILSVRC2013 检测数据集上的结果。该数据集的同质性不如 PASCAL VOC 数据集，需要选择使用方式。由于这些决定是重要的，因此在本节中将对它们进行介绍。

4.1. 数据集概述

ILSVRC2013 检测数据集分为三个子集：训练集（395,918）、验证集（20,121）和测试集（40,152），括号中数字表示每个子集中的图像个数。验证集和测试集源于相同的图像分布。这些图像类似于场景，并且在复杂性（检测对象的数量、混杂程度、姿势多样性等）方面与 PASCAL VOC 数据集中的图像相似。验证集和测试集拆分均进行了详尽注释，即在每个图像中所有 200 类的所有实例都用边框标出。相反，训练集是源于 ILSVRC 2013 分类图像分布。这些图像具有更多的可变复杂性，并且大部分检测对象位于图像中央。与验证集和测试集不同，训练集图像（由于数量众多）未得到详尽注释。任何训练集中的图像可能标记了 200 个类别的实例，也有可能没有被标记。除了这些图像集之外，每个类别还具有一组负样本图像。手动检查负样本图像以确认它们不包含其关联类的任何检测对象。负样本图像集未在这项工作中使用。有关如何收集和注释 ILSVRC 的更多信息，请参见论文[11，36]。

这些拆分的性质为训练 R-CNN 提供了许多选择。训练集图像不能用于硬负样本挖掘，因为注释并不详尽。负样本应该从哪里来？而且，训练集图像的统计信息不同于验证集和测试集（译者注：也就是说训练集与验证集和测试集源于不同的样本分布）。是否应该完全使用训练集图像，如果可以，使用程度如何？尽管我们尚未彻底评估大量选择，但根据以前的经验，我们提出了最合适的选择。

我们的总体策略是严重依赖验证集，并使用一些训练集图像作为

正样本的辅助来源。为了将验证集用于训练和验证，我们将验证集粗略分为大致相等的“val1”和“val2”子集。由于某些类别在验证集中的样本很少（最小的实例只有 31 个，而一半的实例少于 110 个），因此产生一个近似类别平衡的拆分方法是很重要的。为此，生成了大量候选拆分方法，并选择了最小最大相对类别不平衡的拆分方法。每个候选拆分方法都是通过将验证集图像以其类别计数为特征进行聚类而成的，然后进行随机局部搜索以使改善拆分平衡性。此处使用的特定拆分方法的最大相对不平衡约为 11%，中值相对不平衡为 4%。val1/val2 拆分方法和用于生成它们的代码将公开可用，以便其他研究人员将他们的方法和本报告中使用的验证集拆分方法进行比较。

4.2. Region proposals

我们关注了用于 PASCAL 目标检测的类似 region proposal 的方法。论文[39]中对 val1、val2 和 test 中的每个图像上以“快速模式”运行了 Selective search 的方法（但在训练集图像上没有）。需要进行一次较小的修改来处理 selective search 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 图像的大小从很小到几百万像素不等，因此我们在运行 selective search 之前将每个图像的大小调整为固定宽度（500 像素）。在训练集上，selective search 平均每幅图像产生 2403 个 region proposals，所有真值 bounding boxes 的召回率达到 91.6%（阈值为 0.5 IoU）。这个召回率明显低于在 PASCAL 数据集，后者约为 98%，表明在 region proposal 阶段仍有很大的改进空间。

4.3. 训练数据

对于训练数据，我们形成了图像和框的集合，其中包括 val1 中的所有 selective search 和真值 bounding boxes，以及训练集上每个类别最多 N 个真值 bounding boxes（如果训练集类别中少于 N 个真值 bounding boxes，我们全部放到上面的集合中）。我们将此图像和框的数据集称为 val1＋trainN。在消融研究中，我们展示了当 N 属于｛0，500，1000｝时 val2 上的 mAP 值（第 4.5 节）。

R-CNN 中的三个过程需要训练数据：（1）CNN fine-tuning，（2）检测器 SVM 训练，（3）bounding-box 回归器训练。CNN fine-tuning 使用与 PASCAL 完全相同的设置，对 val1+trainN 进行 5 万轮 SGD 迭代。使用 Caffe 在单个 NVIDIA Tesla K20 上进行 fine-tuning 需要 13 个小时。对于 SVM 训练，来自 val1+trainN 的所有真值 bounding-box 均用作各自类别的正样本。对 val1 中 5000 个图像随机选择的子集进行了负样本挖掘。最初的实验表明，从 val1 的所有样本中提取负样本，而不是 5000 个图像子集（大约占一半），最终结果 mAP 仅下降 0.5 个百分点，然而将 SVM 训练时间缩短了一半。由于注释不够详尽，因此没有从训练集中产生任何负样本。未使用额外的验证负样本图像。bounding-box 回归在 val1 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，我们根据上述训练数据的介绍评估了数据使用选择以及在 val2 子集上的 fine-tuning 和 bounding-box 回归的效果。所有系统超参数（例如 SVM C 超参数，区域变形中使

用的填充，NMS 阈值，bounding-box 回归超参数）都固定为与 PASCAL 相同的值。毫无疑问，对于 ILSVRC 这些超参数选择中有些是次优化的，但是这项工作的目标是在 ILSVRC 上产生初步的 R-CNN 结果，而无需进行大量的数据集调整。在 val2 上选择最佳选择之后，我们将两个结果文件提交给 ILSVRC2013 评估服务器。第一个提交没有 bounding-box 回归，第二个提交有 bounding-box 回归。对于这些提交，我们将 SVM 和 bounding-box 回归训练集扩展为分别使用 val + train1k 和 val。我们使用已经在 val1 + train1k 上 fine-tuning 的 CNN 来避免重新运行 fine-tuning 和特征计算。

4.5. 消融研究

表 4 显示了对不同数量的训练数据，微调和边界框回归的影响的消融研究。第一个观察结果是 val2 上的 mAP 与测试中的 mAP 非常接近。这使我们相信 val2 上的 mAP 是测试集性能的良好指标。第一个结果为 20.9%，是 R-CNN 使用在 ILSVRC2012 分类数据集上进行预训练的 CNN 所实现的结果（无微调），并且可以访问 val1 中的少量训练数据（回想一下，val1 中一半类别只有 15 到 55 个样本）。将训练集扩展为 val1 + trainN 可以将性能提高到 24.1%，而 N = 500 和 N = 1000 之间基本上没有差异。仅使用 val1 的样本对 CNN 进行微调就可以将其略微提高到 26.5%，但是可能会存在严重的过拟合，因为训练数据正样本太少。将 fine-tuning 数据集扩展为 val1 + train1k，即从训练集中获取样本将每个类别增加至 1000 个正样本，这将有明显的帮助作用，可以将 mAP 提升至 29.7%。Bounding-box 回归将结果

提高到 31.0%，这是一个较小的相对提升，比在 PASCAL 中观察到的要小。

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

表 4：ILSVRC2013 消融研究：数据使用选择，fine-tuning 和 bounding-box 回归的。

4.6. 与 OverFeat 的关系

R-CNN 与 OverFeat 之间存在有趣的关系：OverFeat 可以（大致）视为 R-CNN 的特例。如果要用规则正方形区域的多尺度金字塔替换 selective search region proposals，并将每类 bounding-box 回归更改为单个 bounding-box 回归，则这两个系统将非常相似（在训练方式上取一些潜在的显着差异取模：CNN 检测 fine-tuning，使用 SVM 等）。值得注意的是，OverFeat 相对于 R-CNN 具有显着的速度优势：基于[34]引用的每幅图像 2 秒的数据，它的速度快约 9 倍。这种速度的提高是因为 OverFeat 的滑动窗口（例如 region proposals）不会在图片级别发生扭曲，因此可以轻松地在重叠的窗口之间共享计算。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。可以通过多种方式加快 R-CNN，这将是未来研究的内容。

5.语义分割

区域分类是语义分割的一种标准技术，使我们能够轻松地将 R-CNN 应用于 PASCAL VOC 分割挑战赛。为了便于与当前领先的语义分割系统（称为“二阶池化”的 O2P）[4]进行直接的比较，我们在其开

源框架开展研究工作。O2P 使用 CPMC 为每个图像生成 150 个 region proposals，然后使用支持向量机回归（SVR）预测每个类别的每个区域的**性质**。其方法的高性能归因于 CPMC 区域的质量以及强大的多种特征类型（SIFT 和 LBP 的多种变体）的二阶池化。我们还注意到，Farabet 等[16]最近在使用 CNN 作为多尺度逐像素分类器的几个密集场景标记数据集（不包括 PASCAL）上得到了良好的结果。

我们根据[2，4]并扩展了 PASCAL 分隔训练集，以包括 Hariharan 等人[22]提供的额外注释。设计决策和超参数在 VOC 2011 验证集中进行了交叉验证。最终测试结果仅评估一次。

使用 CNN 进行语义分隔特征提取。我们评估了三种用于在 CPMC 区域上计算特征的策略，所有策略均从将区域周围的矩形窗口变形为 227*227 开始。**第一个策略（full）**忽略区域的形状并直接在变形窗口上计算 CNN 特征，这与我们做检测时完全一样。但是，这些特征忽略了该区域的非矩形形状。两个区域可能具有非常相似的边界框，而几乎没有重叠。因此，**第二种策略**（fg）仅在区域的前景蒙版上计算 CNN 特征。我们用均值输入替换背景，以便减去均值后背景区域为零。**第三种策略（full + fg）**简单地将 full 和 fg 特征串联起来；我们的实验证明了它们的互补性。

VOC 2011 上的结果。表 5 总结了我们与 O2P 相比在 VOC 2011 验证集中的结果（有关完整的逐类别的结果请参阅附录 E）。在每种特征计算策略中，fc6 层始终胜过 fc7，下面的讨论涉及 fc6 特征。fg 策略略胜于 full 策略，表明被遮罩的区域形状提供了更强的信号，与

我们的直觉相符。但是，full + fg 策略的平均准确度达到 47.9%，我们的最佳结果与其相差 4.2%（也略微胜过 O2P），这表明即使使用 fg 特征，full 特征所提供的上下文信息也非常有用。值得注意的是，在我们的 full + fg 特征上训练 20 个 SVR 在单核 CPU 上花费一个小时，而在 O2P 特征上训练则需要 10 多个小时。

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

表 5：VOC 2011 验证集上的（语义）分隔平均准确度（%）。 第 1 列表示 O2P（模型）；2-7 列（模型）使用我们在 ILSVRC 2012 上进行预训练的 CNN。

在表 6 中，我们给出了 VOC 2011 测试集的结果，并将我们表现最佳的方法 fc6（full + fg）与两个性能较强 baseline 进行了比较。我们的方法在 21 个类别中的 11 个类别中实现了最高的分隔准确度，在各个类别之间平均获得了最高的整体分隔准确度 47.9%（但在任何合理的误差范围内，可能与 O2P 结果相关）。通过 fine-tuning 仍可能获得更好的性能。

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	**36.1**	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O_2P [4]	**85.4**	**69.7**	22.3	45.2	**44.4**	46.9	66.7	57.8	56.2	**13.5**	**46.1**	32.3	41.2	**59.1**	55.3	51.0	**36.2**	50.4	**27.8**	46.9	**44.6**	47.6
ours (*full+fg* R-CNN fc_6)	84.2	66.9	**23.7**	**58.3**	37.4	**55.4**	**73.3**	**58.7**	**56.5**	9.7	45.5	29.5	**49.3**	40.1	**57.8**	**53.9**	33.8	**60.7**	22.7	**47.1**	41.3	**47.9**

表 6：VOC 2011 测试集上的（语义）分隔准确度（%）。我们对比了两个强大的 baseline 模型：论文[2]中提到的“Regions and Parts”（R&P）方法和论文[4]中提到的 second-order pooling（O2P）方法。无需进行任何 fine-tuning，我们的 CNN 就能达到最佳的语义分隔性能，胜过 R&P，与 O2P 基本一致。

6. 结论

近年来，物体检测性能停滞不前。表现最佳的系统是复杂的集成系统，将来自对象检测器和场景分类器的多个低级图像特征与高级上下文结合在一起。本文提出了一种简单且可扩展的对象检测算法，与 PASCAL VOC 2012 上的最佳往期结果相比，性能相对提升了 30%。

我们通过两个思路达到了这一性能。首先是将大容量卷积神经网络应用于自下而上的 region proposals 议，主要为了定位和分割对象。第二个是在缺少标签的训练数据上训练大型 CNN 的方法。我们表明，在有监督的情况下对具有大量数据的辅助任务（图像分类）进行网络预训练，然后针对数据稀缺（检测）的目标任务 fine-tune 网络是非常有效的。我们猜想“有监督的预训练/特定领域的 fine-tuning”方法将对多种数据稀缺的视觉问题非常有效。

最后,我们注意到能得到这些结果，将计算机视觉中经典的工具和深度学习(自底向上的 region proposals 和卷积神经网络）组合是非常重要的。而不是违背科学探索的主线，这两个部分是自然而且必然的结合。

Acknowledgments. This research was supported in part by DARPA Mind's Eye and MSEE programs, by NSF awards IIS-0905647, IIS-1134072, and IIS-1212798, MURI N000014-10-1-0933, and by support from Toyota. The GPUs used in this research were generously donated by the NVIDIA Corporation.

Appendix

A. Object proposal transformations

The convolutional neural network used in this work requires a fixed-size input of 227 227 pixels. For detection, we consider object proposals that are arbitrary image rectangles. We evaluated two approaches for

transforming object proposals into valid CNN inputs.

The first method ("tightest square with context") encloses each object proposal inside the tightest square and then scales (isotropically) the image contained in that square to the CNN input size. Figure 7 column (B) shows this transformation. A variant on this method ("tightest square without context") excludes the image content that surrounds the original object proposal. Figure 7 column (C) shows this transformation. The second method ("warp") anisotropically scales each object proposal to the CNN input size. Figure 7 column (D) shows the warp transformation.

Figure 7: Different object proposal transformations. (A) the original object proposal at its actual scale relative to the transformed CNN inputs; (B) tightest square with context; (C) tightest square without context; (D) warp. Within each column and example proposal, the top row corresponds to p = 0 pixels of context padding while the bottom row has p = 16 pixels of context padding.

For each of these transformations, we also consider including

additional image context around the original object proposal. The amount of context padding (p) is defined as a border size around the original object proposal in the transformed input coordinate frame. Figure 7 shows p = 0 pixels in the top row of each example and p = 16 pixels in the bottom row. In all methods, if the source rectangle extends beyond the image, the missing data is replaced with the image mean (which is then subtracted before inputing the image into the CNN). A pilot set of experiments showed that warping with context padding (p = 16 pixels) outperformed the alternatives by a large margin (3-5 mAP points). Obviously more alternatives are possible, including using replication instead of mean padding. Exhaustive evaluation of these alternatives is left as future work.

B. Positive vs. negative examples and softmax

Two design choices warrant further discussion. The first is: Why are positive and negative examples defined differently for fine-tuning the CNN versus training the object detection SVMs? To review the definitions briefly, for finetuning we map each object proposal to the ground-truth instance with which it has maximum IoU overlap (if any) and label it as a positive for the matched ground-truth class if the IoU is at least 0.5. All other proposals are labeled “background” (i.e., negative examples for all classes). For training SVMs, in contrast, we take only the ground-truth boxes as positive examples for their respective classes and label proposals

with less than 0.3 IoU overlap with all instances of a class as a negative for that class. Proposals that fall into the grey zone (more than 0.3 IoU overlap, but are not ground truth) are ignored.

Historically speaking, we arrived at these definitions because we started by training SVMs on features computed by the ImageNet pre-trained CNN, and so fine-tuning was not a consideration at that point in time. In that setup, we found that our particular label definition for training SVMs was optimal within the set of options we evaluated (which included the setting we now use for fine-tuning). When we started using fine-tuning, we initially used the same positive and negative example definition as we were using for SVM training. However, we found that results were much worse than those obtained using our current definition of positives and negatives.

Our hypothesis is that this difference in how positives and negatives are defined is not fundamentally important and arises from the fact that fine-tuning data is limited. Our current scheme introduces many "jittered" examples (those proposals with overlap between 0.5 and 1, but not ground truth), which expands the number of positive examples by approximately 30x. We conjecture that this large set is needed when fine-tuning the entire network to avoid overfitting. However, we also note that using these

jittered examples is likely suboptimal because the network is not being fine-tuned for precise localization.

This leads to the second issue: Why, after fine-tuning, train SVMs at all? It would be cleaner to simply apply the last layer of the fine-tuned network, which is a 21-way softmax regression classifier, as the object detector. We tried this and found that performance on VOC 2007 dropped from 54.2% to 50.9% mAP. This performance drop likely arises from a combination of several factors including that the definition of positive examples used in fine-tuning does not emphasize precise localization and the softmax classifier was trained on randomly sampled negative examples rather than on the subset of "hard negatives" used for SVM training.

This result shows that it's possible to obtain close to the same level of performance without training SVMs after fine-tuning. We conjecture that with some additional tweaks to fine-tuning the remaining performance gap may be closed. If true, this would simplify and speed up R-CNN training with no loss in detection performance.

C. Bounding-box regression

We use a simple bounding-box regression stage to improve localization performance. After scoring each selective search proposal with

a class-specific detection SVM, we predict a new bounding box for the detection using a class-specific bounding-box regressor. This is similar in spirit to the bounding-box regression used in deformable part models [17]. The primary difference between the two approaches is that here we regress from features computed by the CNN, rather than from geometric features computed on the inferred DPM part locations.

The input to our training algorithm is a set of N training pairs f(Pi;Gi)gi=1;:::;N , where Pi = (Pix ; Pi y; Piw ; Pih) specifies the pixel coordinates of the center of proposal Pi's bounding box together with Pi's width and height in pixels. Hence forth, we drop the superscript i unless it is needed. Each ground-truth bounding box G is specified in the same way: G = (Gx;Gy;Gw;Gh). Our goal is to learn a transformation that maps a proposed box P to a ground-truth box G.

We parameterize the transformation in terms of four functions dx(P), dy(P), dw(P), and dh(P). The first two specify a scale-invariant translation of the center of P's bounding box, while the second two specify log-space translations of the width and height of P's bounding box. After learning these functions, we can transform an input proposal P into a predicted ground-truth box ^G by applying the transformation

^G x = Pwdx(P) + Px (1)

^G y = Phdy(P) + Py (2)

^G w = Pw exp(dw(P)) (3)

^G h = Ph exp(dh(P)): (4)

Each function d?(P) (where ? is one of x; y; h;w) is modeled as a linear function of the pool5 features of proposal P, denoted by -5(P). (The dependence of -5(P) on the image data is implicitly assumed.) Thus we have d?(P) = wT ?-5(P), where w? is a vector of learnable model parameters. We learn w? by optimizing the regularized least squares objective (ridge regression):

w? = argmin ^w? XN I (ti ? □ ^wT?-5(Pi))2 + k ^w?k2 : (5)12

The regression targets t? for the training pair (P;G) are defined as

tx = (Gx □ Px)=Pw (6)

ty = (Gy □ Py)=Ph (7)

tw = log(Gw=Pw) (8)

th = log(Gh=Ph): (9)

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

We found two subtle issues while implementing bounding-box

regression. The first is that regularization is important: we set = 1000 based on a validation set. The second issue is that care must be taken when selecting which training pairs (P;G) to use. Intuitively, if P is far from all ground-truth boxes, then the task of transforming P to a ground-truth box G does not make sense. Using examples like P would lead to a hopeless learning problem. Therefore, we only learn from a proposal P if it is nearby at least one ground-truth box. We implement "nearness" by assigning P to the ground-truth box G with which it has maximum IoU overlap (in case it overlaps more than one) if and only if the overlap is greater than a threshold (which we set to 0.6 using a validation set). All unassigned proposals are discarded. We do this once for each object class in order to learn a set of class-specific bounding-box regressors.

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

D. Additional feature visualizations

Figure 12 shows additional visualizations for 20 pool5 units. For each unit, we show the 24 region proposals that maximally activate that unit out

of the full set of approximately 10 million regions in all of VOC 2007 test.

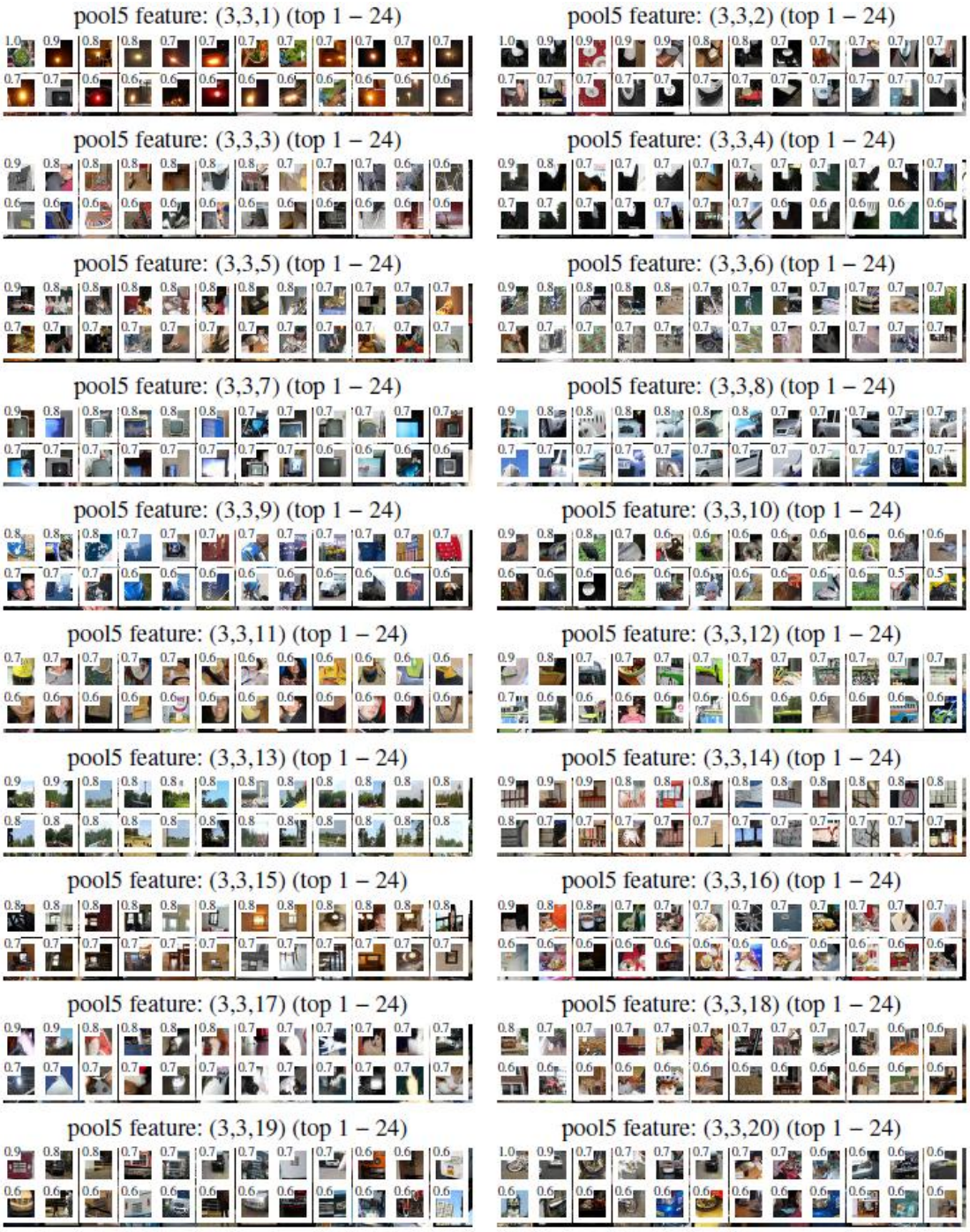

Figure 12: We show the 24 region proposals, out of the approximately 10 million regions in VOC 2007 test, that most strongly activate each of 20 units. Each montage is labeled by the unit's (y, x, channel) position in the 66256 dimensional pool5 feature map. Each image region is drawn with an overlay of the unit's receptive field in white. The activation value (which we normalize by dividing by the max activation value over all units in a channel) is shown in the receptive field's upper-left corner. Best viewed digitally with zoom.

We label each unit by its (y, x, channel) position in the 6 6 256 dimensional pool5 feature map. Within each channel, the CNN computes exactly the same function of the input region, with the (y, x) position changing only the receptive field.

E. Per-category segmentation results

In Table 7 we show the per-category segmentation accuracy on VOC 2011 val for each of our six segmentation methods in addition to the O2P method [4]. These results show which methods are strongest across each of the 20 PASCAL classes, plus the background class.

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

Table 7: Per-category segmentation accuracy (%) on the VOC 2011 validation set.

F. Analysis of cross-dataset redundancy

One concern when training on an auxiliary dataset is that there might be redundancy between it and the test set. Even though the tasks of object detection and whole-image classification are substantially different, making such cross-set redundancy much less worrisome, we still conducted a thorough investigation that quantifies the extent to which PASCAL test images are contained within the ILSVRC 2012 training and validation sets. Our findings may be useful to researchers who are

interested in using ILSVRC 2012 as training data for the PASCAL image classification task.

We performed two checks for duplicate (and nearduplicate) images. The first test is based on exact matches of flickr image IDs, which are included in the VOC 2007 test annotations (these IDs are intentionally kept secret for subsequent PASCAL test sets). All PASCAL images, and about half of ILSVRC, were collected from flickr.com. This check turned up 31 matches out of 4952 (0.63%).

The second check uses GIST [30] descriptor matching, which was shown in [13] to have excellent performance at near-duplicate image detection in large (> 1 million) image collections. Following [13], we computed GIST descriptors on warped 32 32 pixel versions of all ILSVRC 2012 trainval and PASCAL 2007 test images.

Euclidean distance nearest-neighbor matching of GIST descriptors revealed 38 near-duplicate images (including all 31 found by flickr ID matching). The matches tend to vary slightly in JPEG compression level and resolution, and to a lesser extent cropping. These findings show that the overlap is small, less than 1%. For VOC 2012, because flickr IDs are not available, we used the GIST matching method only. Based on GIST

matches, 1.5% of VOC 2012 test images are in ILSVRC 2012 trainval. The slightly higher rate for VOC 2012 is likely due to the fact that the two datasets were collected closer together in time than VOC 2007 and ILSVRC 2012 were.

G. Document changelog

This document tracks the progress of R-CNN. To help readers understand how it has changed over time, here's a brief changelog describing the revisions.

v1 Initial version.

v2 CVPR 2014 camera-ready revision. Includes substantial improvements in detection performance brought about by (1) starting fine-tuning from a higher learning rate (0.001 instead of 0.0001), (2) using context padding when preparing CNN inputs, and (3) bounding-box regression to fix localization errors.

v3 Results on the ILSVRC2013 detection dataset and comparison with OverFeat were integrated into several sections (primarily Section 2 and Section 4).

v4 The softmax vs. SVM results in Appendix B contained an error, which has been fixed. We thank Sergio Guadarrama for helping to identify this issue.

v5 Added results using the new 16-layer network architecture from

Simonyan and Zisserman [43] to Section 3.3 and Table 3.

References

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

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

[3] P. Arbel´aez, 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. Semantic segmentation with second-order pooling. In ECCV, 2012. 4, 10, 11, 13, 14

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

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

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

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

[9] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. 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. J´egou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In Proc. of the ACM International Conference on Image and Video Retrieval, 2009. 13

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

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

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

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

[18] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-

down detection. In CVPR, 2013. 4, 5
[19] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4):193–202, 1980. 1
[20] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http: //www.cs.berkeley.edu/~rbg/latent-v5/. 2, 5, 6, 7
[21] C. Gu, J. J. Lim, P. Arbel´aez, and J. Malik. Recognition using regions. In CVPR, 2009. 2
[22] B. Hariharan, P. Arbel´aez, 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 architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 3
[25] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3, 4, 7
[26] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comp., 1989. 1
[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. Proc. of the IEEE, 1998. 1
[28] J. J. Lim, C. L. Zitnick, and P. Doll´ar. Sketch tokens: A learned mid-level representation for contour and object detection. In CVPR, 2013. 6, 7
[29] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004. 1
[30] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV, 2001. 13
[31] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In CVPR, 2013. 6, 7
[32] H. A. Rowley, S. Baluja, and T. Kanade. Neural networkbased face detection. TPAMI, 1998. 2
[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Parallel Distributed Processing, 1:318–362, 1986. 1
[34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In ICLR, 2014. 1, 2, 4, 10
[35] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In CVPR, 2013. 2
[36] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In AAAI Technical Report, 4th Human Computation Workshop, 2012. 8
[37] K. Sung and T. Poggio. Example-based learning for viewbased human face detection. Technical Report A.I. Memo No. 1521, Massachussets Institute of Technology, 1994. 4

[38] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In NIPS, 2013. 2
[39] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013. 1, 2, 3, 4, 5, 9
[40] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. IEE Proc on Vision, Image, and Signal Processing, 1994. 2
[41] X.Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In ICCV, 2013. 3, 5
[42] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In CVPR, 2011. 4
[43] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint, arXiv:1409.1556, 2014. 6, 7, 14