Rich feature hierarchies for accurate object detection and semantic segmentation

Tech report (v5)

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

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: *Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset. Source code for the complete system is available at* http://www.cs.berkeley.edu/~rbg/rcnn.

1. Introduction

Features matter. The last decade of progress on various visual recognition tasks has been based considerably on the use of SIFT [29] and HOG [7]. But if we look at performance on the canonical visual recognition task, PASCAL VOC object detection [15], it is generally acknowledged that progress has been slow during 2010-2012, with small gains obtained by building ensemble systems and employing minor variants of successful methods.

SIFT and HOG are blockwise orientation histograms, a representation we could associate roughly with complex cells in V1, the first cortical area in the primate visual pathway. But we also know that recognition occurs several stages downstream, which suggests that there might be hierarchical, multi-stage processes for computing features that are even more informative for visual recognition.

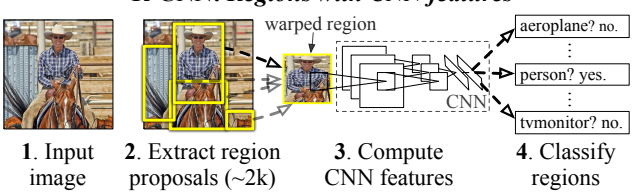


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [39] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%. On the 200-class **ILSVRC2013 detection dataset, R-CNN's mAP is 31.4%**, a large improvement over OverFeat [34], which had the previous best result at 24.3%.

Fukushima's "neocognitron" [19], a biologically-inspired hierarchical and shift-invariant model for pattern recognition, was an early attempt at just such a process. The neocognitron, however, lacked a supervised training algorithm. Building on Rumelhart et al. [33], LeCun et al. [26] showed that stochastic gradient descent via back-propagation was effective for training convolutional neural networks (CNNs), a class of models that extend the neocognitron.

CNNs saw heavy use in the 1990s (e.g., [27]), but then fell out of fashion with the rise of support vector machines. In 2012, Krizhevsky et al. [25] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9, 10]. Their success resulted from training a large CNN on 1.2 million labeled images, together with a few twists on LeCun's CNN (e.g., $\max(x, 0)$ rectifying non-linearities and "dropout" regularization).

The significance of the ImageNet result was vigorously

用于精确目标检测和语义分割的丰富特征层次结构技术报告（v5）

罗斯·吉尔希克 杰夫·多纳休 特雷弗·达雷尔 吉滕德拉·马利克 加州大学伯克利分校

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

摘要

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: *Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset. Source code for the complete system is available at* http://www.cs.berkeley.edu/rbg/rcnn.

1. 引言

特征至关重要。过去十年在各种视觉识别任务上的进展，很大程度上基于SIFT[29]和HOG[7]的使用。但如果我们观察经典视觉识别任务——PASCAL VOC目标检测[15]的性能表现，普遍认为2010年至2012年间进展缓慢，仅通过构建集成系统和采用成功方法的微小变体获得了有限的提升。

SIFT和HOG是分块方向直方图，这种表征我们可以大致关联到灵长类视觉通路的第一皮层区域V1中的复杂细胞。但我们也知道识别发生在下游数个阶段，这意味着可能存在层次——

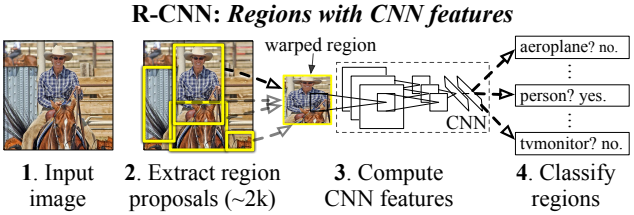


图1：目标检测系统概览。我们的系统（1）接收输入图像，（2）提取约2000个自底向上的区域建议，（3）使用大型卷积神经网络（CNN）计算每个建议的特征，随后（4）使用类别特定的线性支持向量机对每个区域进行分类。R-CNN在PASCAL VOC 2010数据集上实现了53.7%的平均精度均值（mAP）。作为对比，文献[39]使用相同的区域建议，但采用空间金字塔和视觉词袋方法，报告的mAP为35.1%。流行的可变形部件模型性能为33.4%。在200个类别的ILSVRC2013检测数据集上，R-CNN的mAP达到31.4%，较之前最佳结果OverFeat[34]的24.3%有显著提升。

层次化、多阶段的过程，用于计算对视觉识别更具信息量的特征。

福岛邦彦的“神经认知机”[19]是一种受生物学启发的、用于模式识别的分层且平移不变的模型，正是这一过程的早期尝试。然而，神经认知机缺乏监督训练算法。基于Rumelhart等人[33]的研究，LeCun等人[26]证明了通过反向传播进行随机梯度下降能有效训练卷积神经网络（CNNs）——一类扩展了神经认知机的模型。

卷积神经网络在20世纪90年代被大量使用（例如[27]），但随着支持向量机的兴起而逐渐失宠。2012年，Krizhevsky等人[25]通过在ImageNet大规模视觉识别挑战赛（ILSVRC）[9, 10]上展示显著更高的图像分类准确率，重新点燃了对CNN的兴趣。他们的成功源于在120万张标注图像上训练大型CNN，并对LeCun的CNN进行了一些改进（例如使用max({v*} 0)整流非线性激活函数和“dropout”正则化方法）。

ImageNet结果的重要性被激烈地

debated during the ILSVRC 2012 workshop. The central issue can be distilled to the following: To what extent do the CNN classification results on ImageNet generalize to object detection results on the PASCAL VOC Challenge?

We answer this question by bridging the gap between image classification and object detection. This paper is the first to show that a CNN can lead to dramatically higher object detection performance on PASCAL VOC as compared to systems based on simpler HOG-like features. To achieve this result, we focused on two problems: localizing objects with a deep network and training a high-capacity model with only a small quantity of annotated detection data.

Unlike image classification, detection requires localizing (likely many) objects within an image. One approach frames localization as a regression problem. However, work from Szegedy et al. [38], concurrent with our own, indicates that this strategy may not fare well in practice (they report a mAP of 30.5% on VOC 2007 compared to the 58.5% achieved by our method). An alternative is to build a sliding-window detector. CNNs have been used in this way for at least two decades, typically on constrained object categories, such as faces [32, 40] and pedestrians [35]. In order to maintain high spatial resolution, these CNNs typically only have two convolutional and pooling layers. We also considered adopting a sliding-window approach. However, units high up in our network, which has five convolutional layers, have very large receptive fields (195×195 pixels) and strides (32×32 pixels) in the input image, which makes precise localization within the sliding-window paradigm an open technical challenge.

Instead, we solve the CNN localization problem by operating within the "recognition using regions" paradigm [21], which has been successful for both object detection [39] and semantic segmentation [5]. At test time, our method generates around 2000 category-independent region proposals for the input image, extracts a fixed-length feature vector from each proposal using a CNN, and then classifies each region with category-specific linear SVMs. We use a simple technique (affine image warping) to compute a fixed-size CNN input from each region proposal, regardless of the region's shape. Figure 1 presents an overview of our method and highlights some of our results. Since our system combines region proposals with CNNs, we dub the method R-CNN: Regions with CNN features.

In this updated version of this paper, we provide a head-to-head comparison of R-CNN and the recently proposed OverFeat [34] detection system by running R-CNN on the 200-class ILSVRC2013 detection dataset. OverFeat uses a sliding-window CNN for detection and until now was the best performing method on ILSVRC2013 detection. We show that R-CNN significantly outperforms OverFeat, with a mAP of 31.4% versus 24.3%.

A second challenge faced in detection is that labeled data is scarce and the amount currently available is insufficient for training a large CNN. The conventional solution to this problem is to use *unsupervised* pre-training, followed by supervised fine-tuning (e.g., [35]). The second principle contribution of this paper is to show that *supervised* pre-training on a large auxiliary dataset (ILSVRC), followed by domain-specific fine-tuning on a small dataset (PASCAL), is an effective paradigm for learning high-capacity CNNs when data is scarce. In our experiments, fine-tuning for detection improves mAP performance by 8 percentage points. After fine-tuning, our system achieves a mAP of 54% on VOC 2010 compared to 33% for the highly-tuned, HOG-based deformable part model (DPM) [17, 20]. We also point readers to contemporaneous work by Donahue et al. [12], who show that Krizhevsky's CNN can be used (without fine-tuning) as a blackbox feature extractor, yielding excellent performance on several recognition tasks including scene classification, fine-grained sub-categorization, and domain adaptation.

Our system is also quite efficient. The only class-specific computations are a reasonably small matrix-vector product and greedy non-maximum suppression. This computational property follows from features that are shared across all categories and that are also two orders of magnitude lower-dimensional than previously used region features (*cf.* [39]).

Understanding the failure modes of our approach is also critical for improving it, and so we report results from the detection analysis tool of Hoiem et al. [23]. As an immediate consequence of this analysis, we demonstrate that a simple bounding-box regression method significantly reduces mislocalizations, which are the dominant error mode.

Before developing technical details, we note that because R-CNN operates on regions it is natural to extend it to the task of semantic segmentation. With minor modifications, we also achieve competitive results on the PASCAL VOC segmentation task, with an average segmentation accuracy of 47.9% on the VOC 2011 test set.

2. Object detection with R-CNN

Our object detection system consists of three modules. The first generates category-independent region proposals. These proposals define the set of candidate detections available to our detector. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region. The third module is a set of class-specific linear SVMs. In this section, we present our design decisions for each module, describe their test-time usage, detail how their parameters are learned, and show detection results on PASCAL VOC 2010-12 and on ILSVRC2013.

2.1. Module design

Region proposals. A variety of recent papers offer methods for generating category-independent region proposals.

在ILSVRC 2012研讨会上进行了辩论。核心问题可以归结为：CNN在ImageNet上的分类结果能在多大程度上推广到PASCAL VOC挑战赛的目标检测结果?

我们通过弥合图像分类与目标检测之间的鸿沟来回答这个问题。本文首次证明，相较于基于简单HOG类特征的系统，CNN能在PASCAL VOC数据集上实现显著更高的目标检测性能。为实现这一成果，我们聚焦于两个核心问题：利用深度网络定位目标，以及仅使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（可能多个）物体。一种方法将定位问题构建为回归任务。然而，Szegedy等人[38]的研究（与我们的工作同期进行）表明，这种策略在实际应用中可能效果不佳（他们在VOC 2007数据集上报告的mAP为30.5%，而我们的方法达到了58.5%）。另一种方案是构建滑动窗口检测器。至少二十年来，CNN一直以这种方式被使用，通常用于受限的目标类别，如人脸[32, 40]和行人[35]。为了保持高空间分辨率，这些CNN通常仅包含两个卷积层和池化层。我们也曾考虑采用滑动窗口方法。但我们的网络包含五个卷积层，其高层单元在输入图像中具有极大的感受野（195 ×××195像素）和步长（32×××32像素），这使得在滑动窗口框架内实现精确定位成为一个开放的技术挑战。

相反，我们通过在“使用区域进行识别”的范式[21]内操作来解决CNN定位问题，该范式在目标检测[39]和语义分割[5]方面均取得了成功。在测试阶段，我们的方法为输入图像生成约2000个与类别无关的区域建议，使用CNN从每个建议中提取固定长度的特征向量，然后用特定类别的线性SVM对每个区域进行分类。我们采用一种简单技术（仿射图像变形）从每个区域建议中计算固定尺寸的CNN输入，而不考虑区域的形状。图1展示了我们方法的概览，并突出了一些结果。由于我们的系统将区域建议与CNN相结合，我们称该方法为R-CNN：具有CNN特征的区域。

在本文的更新版本中，我们通过在200类别的ILSVRC2013检测数据集上运行R-CNN，对R-CNN与近期提出的OverFeat[34]检测系统进行了直接比较。OverFeat采用滑动窗口CNN进行检测，此前一直是ILSVRC2013检测任务中性能最佳的方法。我们的实验表明，R-CNN以31.4%的mAP显著优于OverFeat的24.3%。

检测面临的第二个挑战是标记数据稀缺且目前可用的数据量不足以训练大型CNN。解决这一问题的常规方法是采用*unsupervised*预训练，随后进行监督微调（例如[35]）。本文的第二个核心贡献在于证明：当数据稀缺时，在大规模辅助数据集（ILSVRC）上进行*supervised*预训练，再针对小规模数据集（PASCAL）进行领域特定微调，是学习高容量CNN的有效范式。在我们的实验中，针对检测任务的微调使mAP性能提升了8个百分点。经微调后，我们的系统在VOC 2010上实现了54%的mAP，而基于HOG的高度调优可变形部件模型（DPM）[17,20]仅为33%。我们同时提请读者关注Donahue等人[12]同期发表的研究，他们证明Krizhevsky的CNN可作为黑盒特征提取器直接使用（无需微调），在场景分类、细粒度子类别划分和领域自适应等多个识别任务中取得了优异性能。

我们的系统也相当高效。唯一的类别特定计算是一个合理的小型矩阵-向量乘积和贪心非极大值抑制。这种计算特性源于所有类别共享的特征，并且这些特征的维度也比之前使用的区域特征（*cf.* [39]）低两个数量级。

理解我们方法的失败模式对于改进它同样至关重要，因此我们报告了Hoiem等人[23]的检测分析工具的结果。作为该分析的一个直接成果，我们证明了一种简单的边界框回归方法能显著减少定位错误，这是最主要的误差模式。

在展开技术细节之前，我们注意到，由于R-CNN基于区域进行操作，将其扩展至语义分割任务是自然而然的。通过少量修改，我们在PASCAL VOC分割任务上也取得了具有竞争力的结果，在VOC 2011测试集上达到了47.9%的平均分割准确率。

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。第一个模块生成与类别无关的区域建议，这些建议定义了检测器可用的候选检测集合。第二个模块是一个大型卷积神经网络，从每个区域提取固定长度的特征向量。第三个模块是一组特定类别的线性支持向量机。在本节中，我们将介绍每个模块的设计决策，描述它们在测试阶段的使用方式，详细说明其参数的学习方法，并展示在PASCAL VOC 2010-12和ILSVRC2013数据集上的检测结果。

2.1. 模块设计

区域提议。最近的多篇论文提出了生成类别无关区域提议的方法。

Figure 2: Warped training samples from VOC 2007 train.

Examples include: objectness [1], selective search [39], category-independent object proposals [14], constrained parametric min-cuts (CPMC) [5], multi-scale combinatorial grouping [3], and Cireşan et al. [6], who detect mitotic cells by applying a CNN to regularly-spaced square crops, which are a special case of region proposals. While R-CNN is agnostic to the particular region proposal method, we use selective search to enable a controlled comparison with prior detection work (e.g., [39, 41]).

Feature extraction. We extract a 4096-dimensional feature vector from each region proposal using the Caffe [24] implementation of the CNN described by Krizhevsky et al. [25]. Features are computed by forward propagating a mean-subtracted 227×227 RGB image through five convolutional layers and two fully connected layers. We refer readers to [24, 25] for more network architecture details.

In order to compute features for a region proposal, we must first convert the image data in that region into a form that is compatible with the CNN (its architecture requires inputs of a fixed 227×227 pixel size). Of the many possible transformations of our arbitrary-shaped regions, we opt for the simplest. Regardless of the size or aspect ratio of the candidate region, we warp all pixels in a tight bounding box around it to the required size. Prior to warping, we dilate the tight bounding box so that at the warped size there are exactly p pixels of warped image context around the original box (we use $p = 16$). Figure 2 shows a random sampling of warped training regions. Alternatives to warping are discussed in Appendix A.

2.2. Test-time detection

At test time, we run selective search on the test image to extract around 2000 region proposals (we use selective search's "fast mode" in all experiments). We warp each proposal and forward propagate it through the CNN in order to compute features. Then, for each class, we score each extracted feature vector using the SVM trained for that class. Given all scored regions in an image, we apply a greedy non-maximum suppression (for each class independently) that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

Run-time analysis. Two properties make detection efficient. First, all CNN parameters are shared across all categories. Second, the feature vectors computed by the CNN are low-dimensional when compared to other common approaches, such as spatial pyramids with bag-of-visual-word encodings. The features used in the UVA detection system [39], for example, are two orders of magnitude larger than ours (360k vs. 4k-dimensional).

The result of such sharing is that the time spent computing region proposals and features (13s/image on a GPU or 53s/image on a CPU) is amortized over all classes. The only class-specific computations are dot products between features and SVM weights and non-maximum suppression. In practice, all dot products for an image are batched into a single matrix-matrix product. The feature matrix is typically 2000×4096 and the SVM weight matrix is $4096 \times N$, where N is the number of classes.

This analysis shows that R-CNN can scale to thousands of object classes without resorting to approximate techniques, such as hashing. Even if there were 100k classes, the resulting matrix multiplication takes only 10 seconds on a modern multi-core CPU. This efficiency is not merely the result of using region proposals and shared features. The UVA system, due to its high-dimensional features, would be two orders of magnitude slower while requiring 134GB of memory just to store 100k linear predictors, compared to just 1.5GB for our lower-dimensional features.

It is also interesting to contrast R-CNN with the recent work from Dean et al. on scalable detection using DPMs and hashing [8]. They report a mAP of around 16% on VOC 2007 at a run-time of 5 minutes per image when introducing 10k distractor classes. With our approach, 10k detectors can run in about a minute on a CPU, and because no approximations are made mAP would remain at 59% (Section 3.2).

2.3. Training

Supervised pre-training. We discriminatively pre-trained the CNN on a large auxiliary dataset (ILSVRC2012 classification) using *image-level annotations* only (bounding-box labels are not available for this data). Pre-training was performed using the open source Caffe CNN library [24]. In brief, our CNN nearly matches the performance of Krizhevsky et al. [25], obtaining a top-1 error rate 2.2 percentage points higher on the ILSVRC2012 classification validation set. This discrepancy is due to simplifications in the training process.

Domain-specific fine-tuning. To adapt our CNN to the new task (detection) and the new domain (warped proposal windows), we continue stochastic gradient descent (SGD) training of the CNN parameters using only warped region proposals. Aside from replacing the CNN's ImageNet-specific 1000-way classification layer with a randomly initialized $(N + 1)$-way classification layer (where N is the number of object classes, plus 1 for background), the CNN architecture is unchanged. For VOC, $N = 20$ and for ILSVRC2013, $N = 200$. We treat all region proposals with

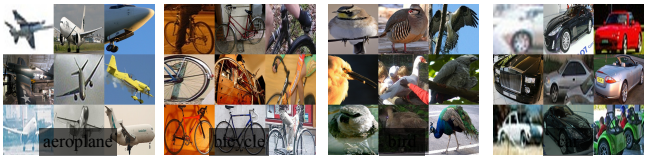


图2：来自VOC 2007训练集的扭曲训练样本。

例子包括：物体性 [1]、选择性搜索 [39]、类别无关物体提议 [14]、约束参数最小割 (CPMC) [5]、多尺度组合分组 [3]，以及 Ciresan 等人 [6]——他们通过将 CNN 应用于规则间隔的方形裁剪区域（这是区域提议的一种特例）来检测有丝分裂细胞。虽然 R-CNN 对具体的区域提议方法并无偏好，但我们采用选择性搜索，以便与先前的检测工作（如 [39, 41]）进行可控的比较。

特征提取。我们使用Krizhevsky等人[25]提出的CNN的Caffe[24]实现，从每个区域建议中提取一个4096维的特征向量。特征计算通过将减去均值的227×227 RGB图像前向传播，经过五个卷积层和两个全连接层完成。更多网络架构细节请参阅[24, 25]。

为了计算区域提议的特征，我们首先必须将该区域内的图像数据转换为与CNN兼容的形式（其架构要求输入为固定的227×227像素尺寸）。在我们任意形状区域众多可能的变换方式中，我们选择了最简单的一种。无论候选区域的尺寸或宽高比如何，我们都会将其紧密边界框内的所有像素变形至所需尺寸。在进行变形前，我们会扩展紧密边界框，使得变形后的图像在原始边界框周围恰好保留p像素的上下文信息（我们采用p ==16）。图2展示了变形训练区域的随机采样示例。关于变形替代方案的讨论详见附录A。

2.2. 测试时检测

在测试阶段，我们对测试图像运行选择性搜索以提取约2000个候选区域（所有实验均使用选择性搜索的“快速模式”）。我们将每个候选区域进行形变处理，并通过CNN前向传播以计算特征。接着，针对每个类别，使用为该类别训练的SVM对提取的特征向量进行评分。在获得图像中所有评分区域后，我们采用贪心非极大值抑制方法（对每个类别独立处理），若某区域与更高评分的已选区域之间的交并比（IoU）超过学习得到的阈值，则拒绝该区域。

运行时分析。两个特性使得检测高效。首先，所有CNN参数在所有类别间共享。其次，由CNN计算出的特征向量与其他常见方法（如使用词袋视觉编码的空间金字塔）相比，其维度较低。例如，UVA检测系统[39]中使用的特征维度比我们的方法高出两个数量级（36万维对比4千维）。

这种共享的结果是，计算区域提议和特征所花费的时间（GPU上13秒/图像或CPU上53秒/图像）被分摊到所有类别上。唯一针对特定类别的计算是特征与SVM权重之间的点积以及非极大值抑制。在实践中，图像的所有点积计算会被批量处理为单个矩阵-矩阵乘积。特征矩阵通常为2000×4096，SVM权重矩阵为4096×N，其中N为类别数量。

分析表明，R-CNN能够扩展到数千个物体类别，而无需借助哈希等近似技术。即使存在10万个类别，在现代多核CPU上进行矩阵乘法也仅需10秒。这种效率不仅源于使用区域提议和共享特征。UVA系统由于其高维特征，速度会慢两个数量级，仅存储10万个线性预测器就需要134GB内存，而我们的低维特征仅需1.5 GB。

将R-CNN与Dean等人近期采用DPM和哈希技术进行可扩展检测的研究[8]进行对比也颇具意义。他们在引入1万个干扰类别时，在VOC 2007数据集上实现了约16%的mAP，每张图像处理耗时约5分钟。而我们的方法在CPU上运行1万个检测器仅需约1分钟，且由于未采用近似处理，mAP可保持在59%（见第3.2节）。

2.3. 训练

监督式预训练。我们仅使用*image-level annotations*在一个大型辅助数据集（ILSVRC2012分类任务）上对CNN进行了判别式预训练（该数据未提供边界框标注）。预训练采用开源Caffe CNN库[24]完成。简而言之，我们的CNN性能与Krizhevsky等人[25]的工作基本相当，在ILSVRC2012分类验证集上的top-1错误率仅高出2.2个百分点。这一差异源于训练流程中的简化处理。

领域特定微调。为了使我们的CNN适应新任务（检测）和新领域（扭曲的候选窗口），我们仅使用扭曲的区域候选继续对CNN参数进行随机梯度下降（SGD）训练。除了将CNN的ImageNet特定1000路分类层替换为随机初始化的（$N+1$）路分类层（其中N是对象类别数量，加上背景类为1），CNN架构保持不变。对于VOC，N =为20；对于ILSVRC2013，N =为200。我们将所有区域候选视为

≥ 0.5 IoU overlap with a ground-truth box as positives for that box's class and the rest as negatives. We start SGD at a learning rate of 0.001 (1/10th of the initial pre-training rate), which allows fine-tuning to make progress while not clobbering the initialization. In each SGD iteration, we uniformly sample 32 positive windows (over all classes) and 96 background windows to construct a mini-batch of size 128. We bias the sampling towards positive windows because they are extremely rare compared to background.

Object category classifiers. Consider training a binary classifier to detect cars. It's clear that an image region tightly enclosing a car should be a positive example. Similarly, it's clear that a background region, which has nothing to do with cars, should be a negative example. Less clear is how to label a region that partially overlaps a car. We resolve this issue with an IoU overlap threshold, below which regions are defined as negatives. The overlap threshold, 0.3, was selected by a grid search over $\{0, 0.1, \ldots, 0.5\}$ on a validation set. We found that selecting this threshold carefully is important. Setting it to 0.5, as in [39], decreased mAP by 5 points. Similarly, setting it to 0 decreased mAP by 4 points. Positive examples are defined simply to be the ground-truth bounding boxes for each class.

Once features are extracted and training labels are applied, we optimize one linear SVM per class. Since the training data is too large to fit in memory, we adopt the standard hard negative mining method [17, 37]. Hard negative mining converges quickly and in practice mAP stops increasing after only a single pass over all images.

In Appendix B we discuss why the positive and negative examples are defined differently in fine-tuning versus SVM training. We also discuss the trade-offs involved in training detection SVMs rather than simply using the outputs from the final softmax layer of the fine-tuned CNN.

2.4. Results on PASCAL VOC 2010-12

Following the PASCAL VOC best practices [15], we validated all design decisions and hyperparameters on the VOC 2007 dataset (Section 3.2). For final results on the VOC 2010-12 datasets, we fine-tuned the CNN on VOC 2012 train and optimized our detection SVMs on VOC 2012 trainval. We submitted test results to the evaluation server only once for each of the two major algorithm variants (with and without bounding-box regression).

Table 1 shows complete results on VOC 2010. We compare our method against four strong baselines, including SegDPM [18], which combines DPM detectors with the output of a semantic segmentation system [4] and uses additional inter-detector context and image-classifier rescoring. The most germane comparison is to the UVA system from Uijlings et al. [39], since our systems use the same region proposal algorithm. To classify regions, their method builds a four-level spatial pyramid and populates it with densely sampled SIFT, Extended OpponentSIFT, and RGB-SIFT descriptors, each vector quantized with 4000-word codebooks. Classification is performed with a histogram intersection kernel SVM. Compared to their multi-feature, non-linear kernel SVM approach, we achieve a large improvement in mAP, from 35.1% to 53.7% mAP, while also being much faster (Section 2.2). Our method achieves similar performance (53.3% mAP) on VOC 2011/12 test.

2.5. Results on ILSVRC2013 detection

We ran R-CNN on the 200-class ILSVRC2013 detection dataset using the same system hyperparameters that we used for PASCAL VOC. We followed the same protocol of submitting test results to the ILSVRC2013 evaluation server only twice, once with and once without bounding-box regression.

Figure 3 compares R-CNN to the entries in the ILSVRC 2013 competition and to the post-competition OverFeat result [34]. R-CNN achieves a mAP of 31.4%, which is significantly ahead of the second-best result of 24.3% from OverFeat. To give a sense of the AP distribution over classes, box plots are also presented and a table of per-class APs follows at the end of the paper in Table 8. Most of the competing submissions (OverFeat, NEC-MU, UvA-Euvision, Toronto A, and UIUC-IFP) used convolutional neural networks, indicating that there is significant nuance in how CNNs can be applied to object detection, leading to greatly varying outcomes.

In Section 4, we give an overview of the ILSVRC2013 detection dataset and provide details about choices that we made when running R-CNN on it.

3. Visualization, ablation, and modes of error

3.1. Visualizing learned features

First-layer filters can be visualized directly and are easy to understand [25]. They capture oriented edges and opponent colors. Understanding the subsequent layers is more challenging. Zeiler and Fergus present a visually attractive deconvolutional approach in [42]. We propose a simple (and complementary) non-parametric method that directly shows what the network learned.

The idea is to single out a particular unit (feature) in the network and use it as if it were an object detector in its own right. That is, we compute the unit's activations on a large set of held-out region proposals (about 10 million), sort the proposals from highest to lowest activation, perform non-maximum suppression, and then display the top-scoring regions. Our method lets the selected unit "speak for itself" by showing exactly which inputs it fires on. We avoid averaging in order to see different visual modes and gain insight into the invariances computed by the unit.

≥ 0.5 与一个真实边界框的IoU重叠作为该框类别的正样本，其余为负样本。我们以0.001的学习率（初始预训练学习率的1/10）开始SGD，这样既能使微调取得进展，又不会破坏初始化权重。在每次SGD迭代中，我们统一采样32个正样本窗口（覆盖所有类别）和96个背景窗口，构建一个大小为128的小批量。我们倾向于采样正样本窗口，因为与背景窗口相比，它们极为罕见。

物体类别分类器。考虑训练一个二元分类器来检测汽车。显然，紧密包围汽车的图像区域应作为正例。同样，与汽车无关的背景区域显然应作为负例。较不明确的是如何标记与汽车部分重叠的区域。我们通过设定一个IoU重叠阈值来解决此问题，低于该阈值的区域被定义为负例。重叠阈值0.3是通过在验证集上对$\{0, 0.1, \ldots, 0.5\}$进行网格搜索选定的。我们发现谨慎选择该阈值非常重要。若将其设为0.5（如文献[39]所示），会使mAP下降5个百分点；若设为0，则会使mAP下降4个百分点。正例则直接定义为每个类别的真实标注边界框。

一旦特征被提取并应用训练标签，我们为每个类别优化一个线性支持向量机。由于训练数据过大无法全部载入内存，我们采用了标准的难负样本挖掘方法[17, 37]。难负样本挖掘收敛迅速，实践中平均精度均值仅需遍历全部图像一次便停止增长。

在附录B中，我们讨论了为何在微调与SVM训练中正负样本的定义方式不同。同时，我们也探讨了训练检测SVM而非直接使用微调CNN最终softmax层输出所涉及的权衡。

2.4. PASCAL VOC 2010-12 数据集上的结果

遵循PASCAL VOC最佳实践[15]，我们在VOC 2007数据集上验证了所有设计决策和超参数（第3.2节）。针对VOC 2010-12数据集的最终结果，我们在VOC 2012训练集上对CNN进行了微调，并在VOC 2012训练验证集上优化了检测SVM。对于两种主要算法变体（使用和不使用边界框回归），我们各自仅向评估服务器提交了一次测试结果。

表1展示了在VOC 2010上的完整结果。我们将本方法与四种强基准方法进行比较，包括SegDPM[18]——该方法将DPM检测器与语义分割系统[4]的输出相结合，并使用了额外的检测器间上下文和图像分类器重评分机制。最具相关性的比较对象是Uijlings等人[39]提出的UVA系统，因为我们的系统采用了相同的区域提议算法。在对区域进行分类时，他们的方法构建了一个四层空间金字塔，并在其中填充密集采样的SIFT、扩展OpponentSIFT和RGB-SIFT描述符，每个均使用4000词码本进行向量量化。分类通过直方图交叉核SVM实现。相较于他们采用的多特征、非线性核SVM方法，我们在mAP上实现了大幅提升——从35.1%提高至53.7%，同时处理速度显著加快（第2.2节）。我们的方法在VOC 2011/12测试集上取得了相近的性能表现（53.3% mAP）。

2.5. ILSVRC2013检测结果

我们在200类的ILSVRC2013检测数据集上运行了R-CNN，使用的系统超参数与PASCAL VOC实验相同。我们遵循了相同的提交协议，仅向ILSVRC2013评估服务器提交了两次测试结果：一次使用边界框回归，一次未使用。

图3将R-CNN与ILSVRC 2013竞赛的参赛方法以及赛后公布的OverFeat结果[34]进行了比较。R-CNN实现了31.4%的mAP，显著领先于第二名OverFeat的24.3%。为展示各类别AP的分布情况，图中同时提供了箱线图，而各类别的具体AP数值则以表格形式列于文末表8。多数参赛方案（OverFeat、NEC-MU、UvA-Euvision、Toronto A和UIUC-IFP）均采用了卷积神经网络，这表明CNN在目标检测中的具体应用方式存在显著差异，从而导致结果产生巨大变化。

在第4节中，我们概述了ILSVRC2013检测数据集，并详细说明了在该数据集上运行R-CNN时所做的选择。

3. 可视化、消融研究与错误模式

3.1. 可视化已学习特征

第一层滤波器可以直接可视化且易于理解[25]。它们捕捉定向边缘和对比色。理解后续层更具挑战性。Zeiler和Fergus在[42]中提出了一种视觉上吸引人的反卷积方法。我们提出了一种简单（且互补）的非参数方法，能直接展示网络学习到的内容。

其思路是挑选出网络中的一个特定单元（特征），并将其视为一个独立的对象检测器。具体来说，我们计算该单元在大量预留区域提议（约1000万个）上的激活值，将提议按激活值从高到低排序，执行非极大值抑制，然后展示得分最高的区域。我们的方法让所选单元“为自己发声”，通过精确展示它对哪些输入产生响应。我们避免使用平均化处理，以便观察不同的视觉模式，并深入理解该单元所计算的不变性。

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]†	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**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in Section C. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

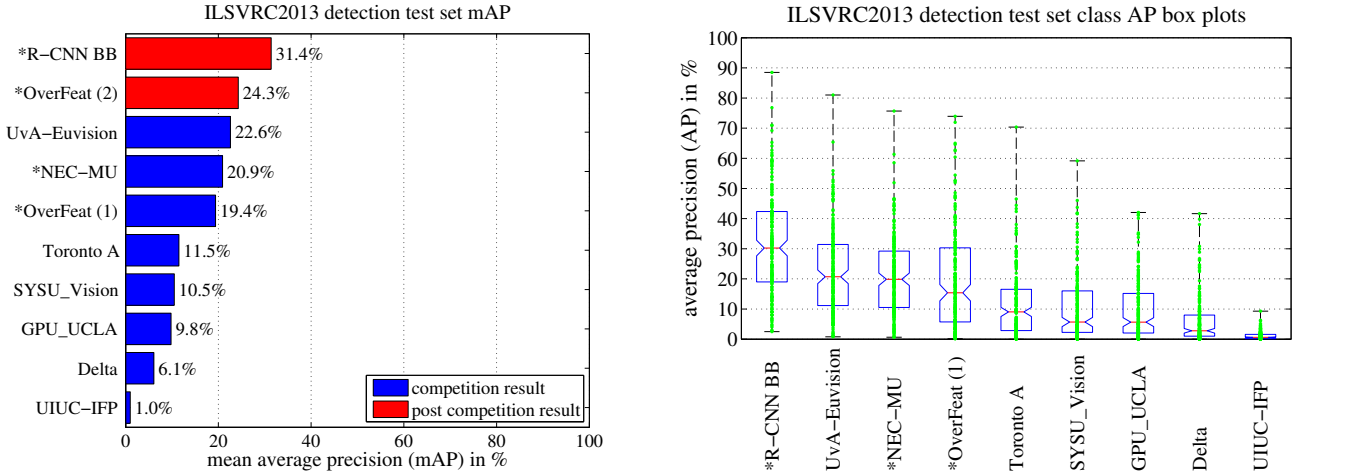


Figure 3: (Left) Mean average precision on the ILSVRC2013 detection test set. Methods preceeded by * use outside training data (images and labels from the ILSVRC classification dataset in all cases). **(Right) Box plots for the 200 average precision values per method.** A box plot for the post-competition OverFeat result is not shown because per-class APs are not yet available (per-class APs for R-CNN are in Table 8 and also included in the tech report source uploaded to arXiv.org; see `R-CNN-ILSVRC2013-APs.txt`). The red line marks the median AP, the box bottom and top are the 25th and 75th percentiles. The whiskers extend to the min and max AP of each method. Each AP is plotted as a green dot over the whiskers (best viewed digitally with zoom).

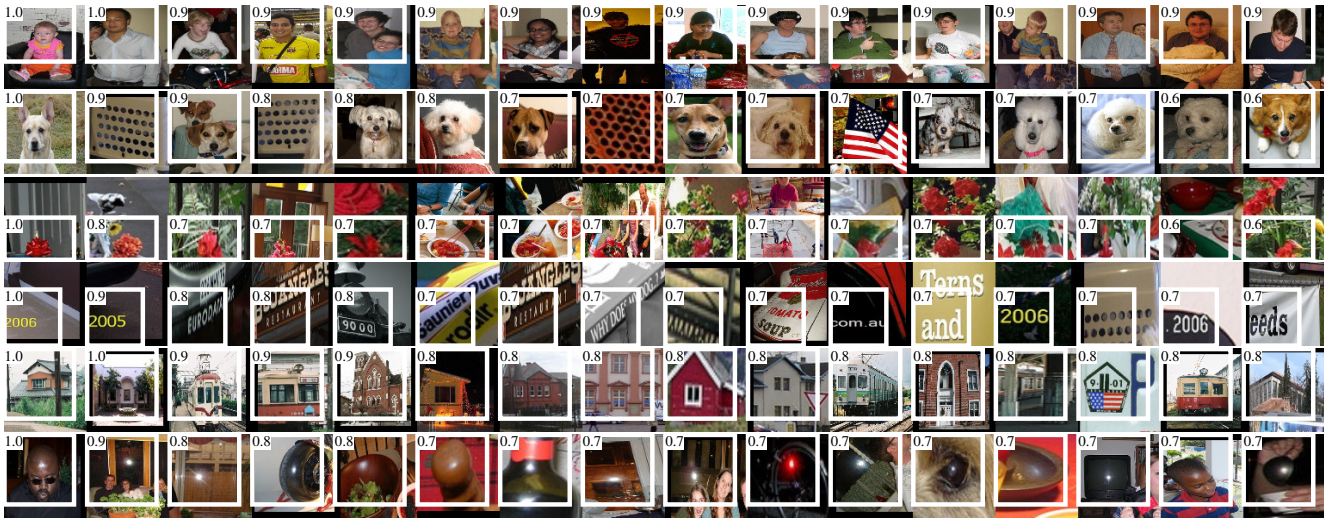


Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

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（%）。由于所有方法均采用选择性搜索区域建议，R-CNN与UVA及Regionlets最具直接可比性。边界框回归（BB）详见C节。在论文发表时，SegDPM是PASCAL VOC排行榜上性能最佳的方法。{v*}DPM和SegDPM使用了其他方法未采用的上下文重评分技术。

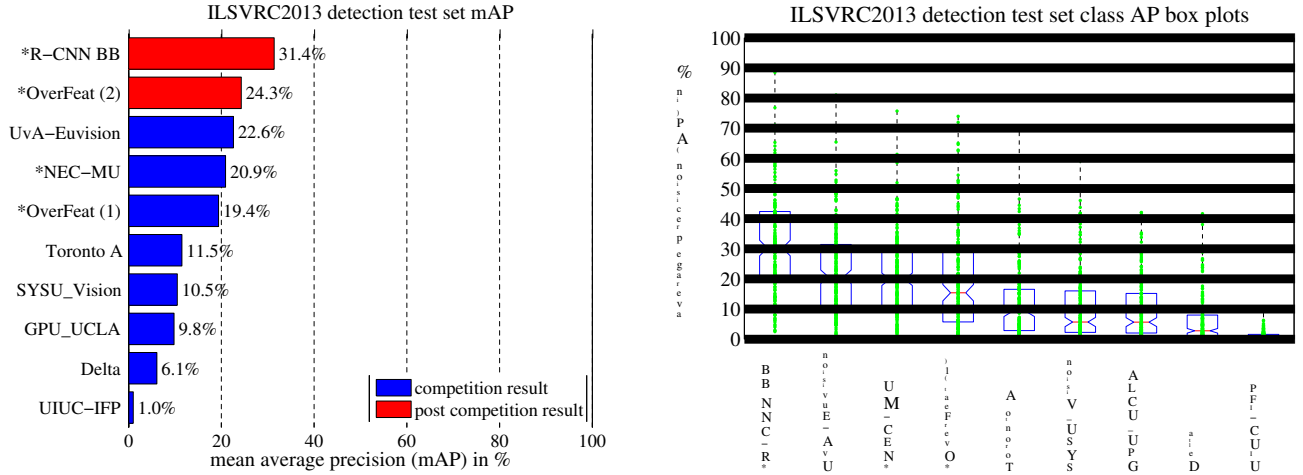


图3：（左）ILSVRC2013检测测试集的平均精度均值。标有*的方法使用了外部训练数据（所有情况均使用了ILSVRC分类数据集的图像和标签）。（右）每种方法对应的200个平均精度值的箱线图。未展示赛后OverFeat结果的箱线图，因为其每类AP尚未公布（R-CNN的每类AP见表8，同时已包含在提交至arXiv.org的技术报告源文件中，详见R-CNN-ILSVRC2013-APs.txt）。红线表示中位数AP，箱体底部和顶部分别为第25和第75百分位数。须线延伸至每种方法AP的最小值和最大值。每个AP以绿点形式绘制在须线上方（建议在数字设备上缩放查看）。

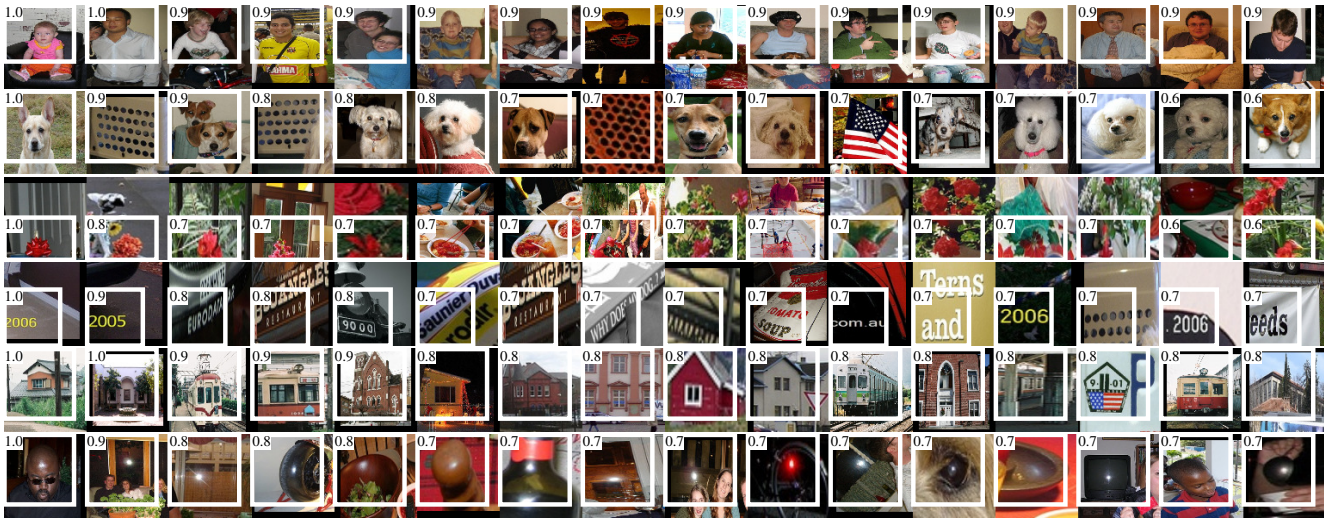


图4：六个池化$_5$单元的顶部区域。感受野和激活值以白色绘制。一些单元与概念对齐，例如人物（第1行）或文本（第4行）。其他单元捕捉纹理和材质属性，例如点阵（第2行）和镜面反射（第6行）。

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

Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show R-CNN performance without fine-tuning. Rows 4-6 show results for the CNN pre-trained on ILSVRC 2012 and then fine-tuned (FT) on VOC 2007 trainval. Row 7 includes a simple bounding-box regression (BB) stage that reduces localization errors (Section C). Rows 8-10 present DPM methods as a strong baseline. The first uses only HOG, while the next two use different feature learning approaches to augment or replace HOG.

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

Table 3: Detection average precision (%) on VOC 2007 test for two different CNN architectures. The first two rows are results from Table 2 using Krizhevsky et al.'s architecture (T-Net). Rows three and four use the recently proposed 16-layer architecture from Simonyan and Zisserman (O-Net) [43].

We visualize units from layer $pool_5$, which is the max-pooled output of the network's fifth and final convolutional layer. The $pool_5$ feature map is $6 \times 6 \times 256 = 9216$-dimensional. Ignoring boundary effects, each $pool_5$ unit has a receptive field of 195×195 pixels in the original 227×227 pixel input. A central $pool_5$ unit has a nearly global view, while one near the edge has a smaller, clipped support.

Each row in Figure 4 displays the top 16 activations for a $pool_5$ unit from a CNN that we fine-tuned on VOC 2007 trainval. Six of the 256 functionally unique units are visualized (Appendix D includes more). These units were selected to show a representative sample of what the network learns. In the second row, we see a unit that fires on dog faces and dot arrays. The unit corresponding to the third row is a red blob detector. There are also detectors for human faces and more abstract patterns such as text and triangular structures with windows. The network appears to learn a representation that combines a small number of class-tuned features together with a distributed representation of shape, texture, color, and material properties. The subsequent fully connected layer fc_6 has the ability to model a large set of compositions of these rich features.

3.2. Ablation studies

Performance layer-by-layer, without fine-tuning. To understand which layers are critical for detection performance, we analyzed results on the VOC 2007 dataset for each of the CNN's last three layers. Layer $pool_5$ was briefly described in Section 3.1. The final two layers are summarized below.

Layer fc_6 is fully connected to $pool_5$. To compute features, it multiplies a 4096×9216 weight matrix by the $pool_5$ feature map (reshaped as a 9216-dimensional vector) and then adds a vector of biases. This intermediate vector is component-wise half-wave rectified ($x \leftarrow \max(0, x)$).

Layer fc_7 is the final layer of the network. It is implemented by multiplying the features computed by fc_6 by a 4096×4096 weight matrix, and similarly adding a vector of biases and applying half-wave rectification.

We start by looking at results from the CNN *without fine-tuning* on PASCAL, i.e. all CNN parameters were pre-trained on ILSVRC 2012 only. Analyzing performance layer-by-layer (Table 2 rows 1-3) reveals that features from fc_7 generalize worse than features from fc_6. This means that 29%, or about 16.8 million, of the CNN's parameters can be removed without degrading mAP. More surprising is that removing *both* fc_7 and fc_6 produces quite good results even though $pool_5$ features are computed using *only 6%* of the CNN's parameters. Much of the CNN's representational power comes from its convolutional layers, rather than from the much larger densely connected layers. This finding suggests potential utility in computing a dense feature map, in the sense of HOG, of an arbitrary-sized image by using only the convolutional layers of the CNN. This representation would enable experimentation with sliding-window detectors, including DPM, on top of $pool_5$ features.

Performance layer-by-layer, with fine-tuning. We now look at results from our CNN after having fine-tuned its pa-

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行展示了未经微调的R-CNN性能。第4-6行展示了在ILSVRC 2012上预训练、随后在VOC 2007训练验证集上微调（FT）的CNN结果。第7行加入了一个简单的边界框回归（BB）阶段以减少定位误差（C节）。第8-10行展示了作为强基线的DPM方法。第一种仅使用HOG特征，后两种则采用不同的特征学习方法以增强或替代HOG。

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

表3：在VOC 2007测试集上两种不同CNN架构的检测平均精度（%）。前两行是采用Krizhevsky等人架构（T-Net）的表2结果，第三、四行则使用了Simonyan与Zisserman近期提出的16层架构（O-Net）[43]。

我们可视化来自层$pool_5$的单元，这是网络第五个也是最后一个卷积层的最大池化输出。$pool_5$特征图是6××6××256=维的，即9216维。忽略边界效应，每个$pool_5$单元在原始227××227像素输入中具有195××195像素的感受野。位于中心的$pool_5$单元具有近乎全局的视野，而靠近边缘的单元则具有较小且被裁剪的感受范围。

图4中的每一行展示了一个来自我们在VOC 2007训练验证集上微调过的CNN的池化$_5$单元的前16个激活。图中可视化了256个功能独特单元中的六个（附录D包含更多示例）。选择这些单元是为了展示网络学习内容的代表性样本。在第二行中，我们看到一个对狗脸和点阵响应的单元。第三行对应的单元是一个红色斑点检测器。此外还有针对人脸以及更抽象模式（如文本和带窗户的三角形结构）的检测器。网络似乎学会了一种表示方法，该方法将少量类别调谐特征与形状、纹理、颜色和材质属性的分布式表示相结合。随后的全连接层fc_6能够对这些丰富特征的大量组合进行建模。

3.2. 消融研究

逐层性能表现，无需微调。为了理解哪些层对检测性能至关重要，我们分析了VOC 2007数据集上CNN最后三层的各自结果。第$pool_5$层已在3.1节简要描述。最后两层的总结如下。

fc_6层与$pool_5$层全连接。为计算特征，它将一个4096××9216的权重矩阵与$pool_5$特征图（重塑为9216维向量）相乘，随后加上偏置向量。该中间向量会进行逐分量半波整流（$x \leftarrow \max(0, x)$）。

fc_7层是网络的最后一层。它通过将fc_6计算的特征乘以一个4096×4096的权重矩阵来实现，同样地加上偏置向量并应用半波整流。

我们首先查看CNN *without fine-tuning*在PASCAL数据集上的结果，即所有CNN参数仅在ILSVRC 2012上进行预训练。通过逐层分析性能（表2第1-3行）发现，fc_7层的特征比fc_6层的特征泛化能力更差。这意味着可以移除CNN中29%（约1680万）的参数而不降低mAP。更令人惊讶的是，移除*both* fc_7和fc_6层仍能产生相当好的结果，尽管$pool_5$特征仅使用了CNN*only 6%*的参数进行计算。CNN的大部分表征能力来自其卷积层，而非参数量大得多的全连接层。这一发现表明，通过仅使用CNN的卷积层计算任意尺寸图像的密集特征图（类似于HOG的思路）具有潜在应用价值。这种表征方式能够在$pool_5$特征之上实现滑动窗口检测器（包括DPM）的实验。

逐层性能表现，经过微调。我们现在观察CNN在对其参数进行微调后的结果——

rameters on VOC 2007 trainval. The improvement is striking (Table 2 rows 4-6): fine-tuning increases mAP by 8.0 percentage points to 54.2%. The boost from fine-tuning is much larger for fc_6 and fc_7 than for $pool_5$, which suggests that the $pool_5$ features learned from ImageNet are general and that most of the improvement is gained from learning domain-specific non-linear classifiers on top of them.

Comparison to recent feature learning methods. Relatively few feature learning methods have been tried on PASCAL VOC detection. We look at two recent approaches that build on deformable part models. For reference, we also include results for the standard HOG-based DPM [20].

The first DPM feature learning method, DPM ST [28], augments HOG features with histograms of "sketch token" probabilities. Intuitively, a sketch token is a tight distribution of contours passing through the center of an image patch. Sketch token probabilities are computed at each pixel by a random forest that was trained to classify 35×35 pixel patches into one of 150 sketch tokens or background.

The second method, DPM HSC [31], replaces HOG with histograms of sparse codes (HSC). To compute an HSC, sparse code activations are solved for at each pixel using a learned dictionary of 100 7×7 pixel (grayscale) atoms. The resulting activations are rectified in three ways (full and both half-waves), spatially pooled, unit ℓ_2 normalized, and then power transformed ($x \leftarrow \text{sign}(x)|x|^\alpha$).

All R-CNN variants strongly outperform the three DPM baselines (Table 2 rows 8-10), including the two that use feature learning. Compared to the latest version of DPM, which uses only HOG features, our mAP is more than 20 percentage points higher: 54.2% vs. 33.7%—*a 61% relative improvement*. The combination of HOG and sketch tokens yields 2.5 mAP points over HOG alone, while HSC improves over HOG by 4 mAP points (when compared internally to their private DPM baselines—both use non-public implementations of DPM that underperform the open source version [20]). These methods achieve mAPs of 29.1% and 34.3%, respectively.

3.3. Network architectures

Most results in this paper use the network architecture from Krizhevsky et al. [25]. However, we have found that the choice of architecture has a large effect on R-CNN detection performance. In Table 3 we show results on VOC 2007 test using the 16-layer deep network recently proposed by Simonyan and Zisserman [43]. This network was one of the top performers in the recent ILSVRC 2014 classification challenge. The network has a homogeneous structure consisting of 13 layers of 3×3 convolution kernels, with five max pooling layers interspersed, and topped with three fully-connected layers. We refer to this network as "O-Net" for OxfordNet and the baseline as "T-Net" for TorontoNet.

To use O-Net in R-CNN, we downloaded the publicly available pre-trained network weights for the `VGG_ILSVRC_16_layers` model from the Caffe Model Zoo.[1] We then fine-tuned the network using the same protocol as we used for T-Net. The only difference was to use smaller minibatches (24 examples) as required in order to fit within GPU memory. The results in Table 3 show that R-CNN with O-Net substantially outperforms R-CNN with T-Net, increasing mAP from 58.5% to 66.0%. However there is a considerable drawback in terms of compute time, with the forward pass of O-Net taking roughly 7 times longer than T-Net.

3.4. Detection error analysis

We applied the excellent detection analysis tool from Hoiem et al. [23] in order to reveal our method's error modes, understand how fine-tuning changes them, and to see how our error types compare with DPM. A full summary of the analysis tool is beyond the scope of this paper and we encourage readers to consult [23] to understand some finer details (such as "normalized AP"). Since the analysis is best absorbed in the context of the associated plots, we present the discussion within the captions of Figure 5 and Figure 6.

3.5. Bounding-box regression

Based on the error analysis, we implemented a simple method to reduce localization errors. Inspired by the bounding-box regression employed in DPM [17], we train a linear regression model to predict a new detection window given the $pool_5$ features for a selective search region proposal. Full details are given in Appendix C. Results in Table 1, Table 2, and Figure 5 show that this simple approach fixes a large number of mislocalized detections, boosting mAP by 3 to 4 points.

3.6. Qualitative results

Qualitative detection results on ILSVRC2013 are presented in Figure 8 and Figure 9 at the end of the paper. Each image was sampled randomly from the val_2 set and all detections from all detectors with a precision greater than 0.5 are shown. Note that these are not curated and give a realistic impression of the detectors in action. More qualitative results are presented in Figure 10 and Figure 11, but these have been curated. We selected each image because it contained interesting, surprising, or amusing results. Here, also, all detections at precision greater than 0.5 are shown.

4. The ILSVRC2013 detection dataset

In Section 2 we presented results on the ILSVRC2013 detection dataset. This dataset is less homogeneous than

[1] https://github.com/BVLC/caffe/wiki/Model-Zoo

在VOC 2007训练验证集上的参数调整。改进效果显著（表2第4-6行）：微调使mAP提升了8.0个百分点，达到54.2%。fc_6和fc_7通过微调获得的提升远大于$pool_5$，这表明从ImageNet学习到的$pool_5$特征具有通用性，而主要改进来源于在其基础上学习领域特定的非线性分类器。

与近期特征学习方法的比较。相对而言，在PASCAL VOC检测任务上尝试过的特征学习方法还不多。我们考察了两种基于可变形部件模型的最新方法。作为参照，我们也包含了基于标准HOG的DPM方法的结果[20]。

首个DPM特征学习方法，DPM ST [28]，通过“草图标记”概率直方图增强了HOG特征。直观来说，草图标记是穿过图像块中心的轮廓紧密分布。每个像素点的草图标记概率由随机森林计算得出，该随机森林经过训练，能将35×35像素块分类为150种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）替代了HOG。为计算HSC，需使用一个包含100个7×7像素（灰度）原子的学习字典，在每个像素处求解稀疏编码激活值。得到的激活值通过三种方式（全波与两个半波）进行整流，经空间池化、单元ℓ_2归一化，再进行幂变换（$x \leftarrow \text{sign}(x)|x|^{\alpha}$）。

所有R-CNN变体均显著优于三种DPM基线方法（表2第8-10行），其中包含两种采用特征学习的方法。与仅使用HOG特征的DPM最新版本相比，我们的mAP高出20多个百分点：54.2%对比33.7%——*a 61% relative improvement*。HOG与草图标记的组合相比单独使用HOG提升了2.5个mAP点，而HSC相比HOG则提升了4个mAP点（此数据基于其内部非公开DPM基线的对比结果——两者均使用未公开的DPM实现方案，其性能低于开源版本[20]）。这些方法分别实现了29.1%和34.3%的mAP。

3.3. 网络架构

本文中的大多数结果采用了Krizhevsky等人[25]提出的网络架构。然而，我们发现架构选择对R-CNN检测性能有显著影响。表3展示了使用Simonyan和Zisserman [43]近期提出的16层深度网络在VOC 2007测试集上的结果。该网络在近期ILSVRC 2014分类挑战赛中位列前茅，其采用同质化结构：包含13层3×3卷积核，穿插5个最大池化层，顶部连接三个全连接层。我们将该网络称为“O-Net”（牛津网络），基线网络称为“T-Net”（多伦多网络）。

为了在R-CNN中使用O-Net，我们从Caffe Model Zoo下载了VGG ILSVRC 16层模型的公开预训练网络权重[1]。随后，我们采用与T-Net相同的流程对网络进行微调。唯一的区别是使用了更小的最小批次（24个样本），以满足GPU内存的限制。表3的结果表明，采用O-Net的R-CNN显著优于采用T-Net的R-CNN，将mAP从58.5%提升至66.0%。然而，这在计算时间上存在明显缺陷——O-Net的前向传播耗时约为T-Net的7倍。

3.4. 检测误差分析

我们采用了Hoiem等人[23]提出的优秀检测分析工具，以揭示我们方法的错误模式、理解微调如何改变这些模式，并比较我们的错误类型与DPM的差异。该分析工具的完整概述超出了本文的讨论范围，我们建议读者查阅文献[23]以了解更精细的细节（例如“归一化AP”）。由于结合相关图表能更有效地理解分析结果，我们将具体讨论置于图5和图6的说明文字中。

3.5. 边界框回归

基于误差分析，我们实施了一种简单方法来减少定位误差。受DPM[17]中采用的边界框回归启发，我们训练了一个线性回归模型，利用选择性搜索区域建议的池化$_5$特征来预测新的检测窗口。完整细节见附录C。表1、表2和图5中的结果表明，这种简单方法修正了大量定位错误的检测，将mAP提升了3到4个百分点。

3.6. 定性结果

ILSVRC2013上的定性检测结果在论文末尾的图8和图9中展示。每张图像均从val_2集中随机抽取，并显示了所有检测器中精度大于0.5的全部检测结果。请注意，这些结果未经筛选，真实反映了检测器的实际运行情况。更多定性结果见图10和图11，但后者经过人工筛选。我们选择每张图像是因为其包含有趣、令人惊讶或有趣的检测结果。此处同样展示了所有精度大于0.5的检测结果。

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。该数据集的同质性低于

[1] https://github.com/BVLC/caffe/wiki/Model-Zoo

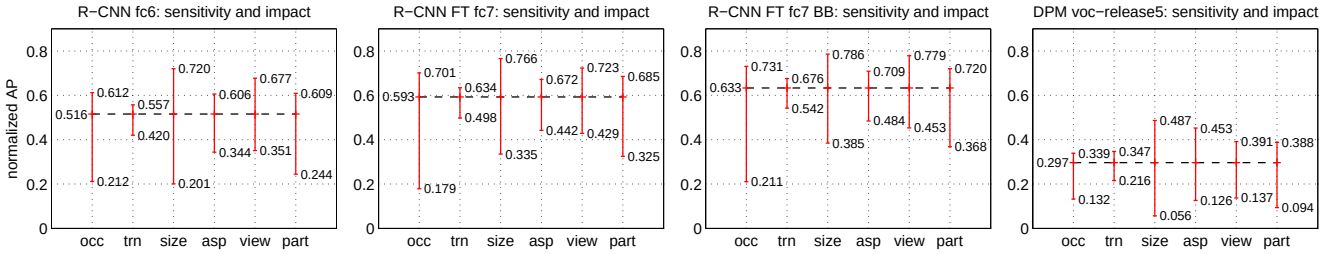


Figure 6: Sensitivity to object characteristics. Each plot shows the mean (over classes) normalized AP (see [23]) for the highest and lowest performing subsets within six different object characteristics (occlusion, truncation, bounding-box area, aspect ratio, viewpoint, part visibility). We show plots for our method (R-CNN) with and without fine-tuning (FT) and bounding-box regression (BB) as well as for DPM voc-release5. Overall, fine-tuning does not reduce sensitivity (the difference between max and min), but does substantially improve both the highest and lowest performing subsets for nearly all characteristics. This indicates that fine-tuning does more than simply improve the lowest performing subsets for aspect ratio and bounding-box area, as one might conjecture based on how we warp network inputs. Instead, fine-tuning improves robustness for all characteristics including occlusion, truncation, viewpoint, and part visibility.

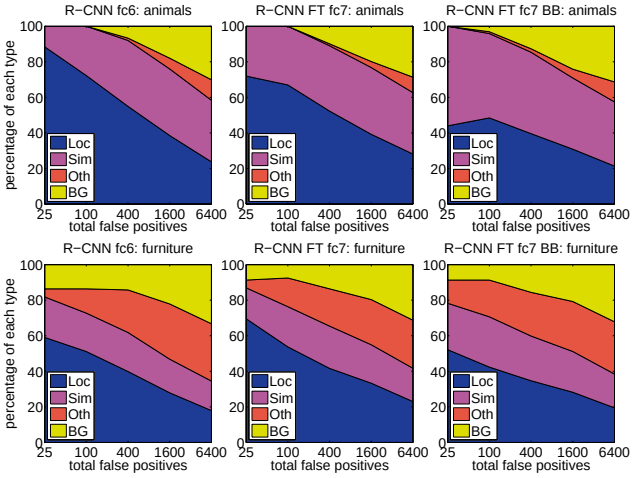


Figure 5: Distribution of top-ranked false positive (FP) types. Each plot shows the evolving distribution of FP types as more FPs are considered in order of decreasing score. Each FP is categorized into 1 of 4 types: Loc—poor localization (a detection with an IoU overlap with the correct class between 0.1 and 0.5, or a duplicate); Sim—confusion with a similar category; Oth—confusion with a dissimilar object category; BG—a FP that fired on background. Compared with DPM (see [23]), significantly more of our errors result from poor localization, rather than confusion with background or other object classes, indicating that the CNN features are much more discriminative than HOG. Loose localization likely results from our use of bottom-up region proposals and the positional invariance learned from pre-training the CNN for whole-image classification. Column three shows how our simple bounding-box regression method fixes many localization errors.

PASCAL VOC, requiring choices about how to use it. Since these decisions are non-trivial, we cover them in this section.

4.1. Dataset overview

The ILSVRC2013 detection dataset is split into three sets: train (395,918), val (20,121), and test (40,152), where the number of images in each set is in parentheses. The val and test splits are drawn from the same image distribution. These images are scene-like and similar in complexity (number of objects, amount of clutter, pose variability, etc.) to PASCAL VOC images. The val and test splits are exhaustively annotated, meaning that in each image all instances from all 200 classes are labeled with bounding boxes. The train set, in contrast, is drawn from the ILSVRC2013 *classification* image distribution. These images have more variable complexity with a skew towards images of a single centered object. Unlike val and test, the train images (due to their large number) are not exhaustively annotated. In any given train image, instances from the 200 classes may or may not be labeled. In addition to these image sets, each class has an extra set of negative images. Negative images are manually checked to validate that they do not contain any instances of their associated class. The negative image sets were not used in this work. More information on how ILSVRC was collected and annotated can be found in [11, 36].

The nature of these splits presents a number of choices for training R-CNN. The train images cannot be used for hard negative mining, because annotations are not exhaustive. Where should negative examples come from? Also, the train images have different statistics than val and test. Should the train images be used at all, and if so, to what extent? While we have not thoroughly evaluated a large number of choices, we present what seemed like the most obvious path based on previous experience.

Our general strategy is to rely heavily on the val set and use some of the train images as an auxiliary source of positive examples. To use val for both training and validation, we split it into roughly equally sized "val_1" and "val_2" sets. Since some classes have very few examples in val (the smallest has only 31 and half have fewer than 110), it is important to produce an approximately class-balanced partition. To do this, a large number of candidate splits were generated and the one with the smallest maximum relative

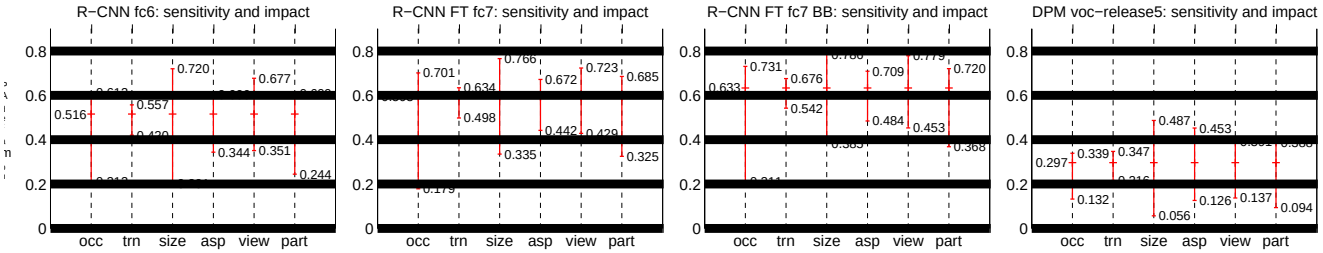


图6：对物体特征的敏感性。每个图表展示了在六种不同物体特征（遮挡、截断、边界框面积、宽高比、视角、部件可见性）中，表现最佳和最差子集的平均（按类别）归一化AP（参见[23]）。我们展示了本方法（R-CNN）在有无微调（FT）和边界框回归（BB）下的结果，以及DPM voc-release5的结果。总体而言，微调并未降低敏感性（最大值与最小值之间的差异），但几乎对所有特征都显著提升了表现最佳和最差子集的性能。这表明，微调不仅仅改善了宽高比和边界框面积方面表现最差的子集——正如人们可能根据我们对网络输入的变形方式所推测的那样。相反，微调提升了包括遮挡、截断、视角和部件可见性在内的所有特征的鲁棒性。

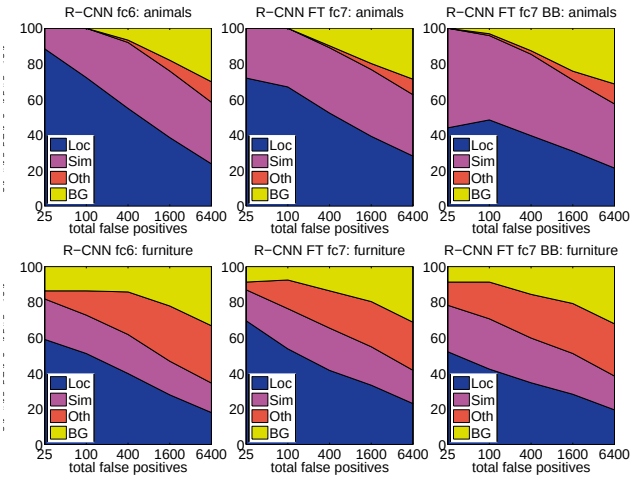


图5：排名靠前的假阳性（FP）类型分布。每个子图展示了随着按分数递减顺序考虑更多FP时，FP类型的动态分布情况。每个FP被归类为以下4种类型之一：定位错误（Loc）——定位不佳（与正确类别的IoU重叠度在0.1到0.5之间的检测结果，或重复检测）；相似类别混淆（Sim）——与相似类别混淆；其他类别混淆（Oth）——与不相似物体类别混淆；背景误判（BG）——在背景上触发的FP。与DPM相比（参见[23]），我们的错误更多源于定位不佳，而非背景或其他物体类别的混淆，这表明CNN特征比HOG更具区分性。定位松散可能源于我们采用自下而上的区域提议方法，以及通过全图像分类预训练CNN所学习到的位置不变性。第三列展示了我们简单的边界框回归方法如何修正许多定位错误。

PASCAL VOC，需要决定如何使用它。由于这些决策并非无关紧要，我们将在本节中讨论它们。

4.1. 数据集概述

ILSVRC2013检测数据集被划分为三个部分：训练集（395,918张）、验证集（20,121张）和测试集（40,152张），括号内为各集合的图像数量。

验证集和测试集来自相同的图像分布。这些图像具有场景感，其复杂度（物体数量、杂乱程度、姿态变化等）与PASCAL VOC图像相似。验证集和测试集均经过详尽标注，即每张图像中所有200个类别的实例均用边界框进行了标注。相比之下，训练集则来自ILSVRC 2013 *clas- sification*图像分布。这些图像的复杂度变化更大，且倾向于呈现单一居中的物体。与验证集和测试集不同，训练图像（因其数量庞大）未进行详尽标注。在任何给定的训练图像中，200个类别的实例可能被标注，也可能未被标注。除了这些图像集外，每个类别还有一组额外的负样本图像。负样本图像经过人工检查，确认不包含其对应类别的任何实例。本工作中未使用负样本图像集。关于ILSVRC数据收集和标注方式的更多信息可在[11, 36]中找到。

这些分割的性质为训练R-CNN提出了若干选择。由于标注并非详尽无遗，训练图像不能用于难负例挖掘。那么负样本应从何而来？此外，训练图像与验证集和测试集具有不同的统计特性。是否应该使用训练图像？如果使用，又该用到何种程度？虽然我们尚未对大量选择进行全面评估，但基于以往经验，我们提出了看似最直接的路径。

我们的总体策略是严重依赖验证集，并将部分训练图像作为正样本的辅助来源。为了将验证集同时用于训练和验证，我们将其大致等分为“val_1”和“val_2”两个子集。由于某些类别在验证集中的样本数量极少（最少的仅有31个，半数类别不足110个），因此生成一个近似类别均衡的划分至关重要。为此，我们生成了大量候选划分方案，并选择了最大相对差异最小的方案。

class imbalance was selected.[2] Each candidate split was generated by clustering val images using their class counts as features, followed by a randomized local search that may improve the split balance. The particular split used here has a maximum relative imbalance of about 11% and a median relative imbalance of 4%. The val_1/val_2 split and code used to produce them will be publicly available to allow other researchers to compare their methods on the val splits used in this report.

4.2. Region proposals

We followed the same region proposal approach that was used for detection on PASCAL. Selective search [39] was run in "fast mode" on each image in val_1, val_2, and test (but not on images in train). One minor modification was required to deal with the fact that selective search is not scale invariant and so the number of regions produced depends on the image resolution. ILSVRC image sizes range from very small to a few that are several mega-pixels, and so we resized each image to a fixed width (500 pixels) before running selective search. On val, selective search resulted in an average of 2403 region proposals per image with a 91.6% recall of all ground-truth bounding boxes (at 0.5 IoU threshold). This recall is notably lower than in PASCAL, where it is approximately 98%, indicating significant room for improvement in the region proposal stage.

4.3. Training data

For training data, we formed a set of images and boxes that includes all selective search and ground-truth boxes from val_1 together with up to N ground-truth boxes per class from train (if a class has fewer than N ground-truth boxes in train, then we take all of them). We'll call this dataset of images and boxes $\text{val}_1{+}\text{train}_N$. In an ablation study, we show mAP on val_2 for $N \in \{0, 500, 1000\}$ (Section 4.5).

Training data is required for three procedures in R-CNN: (1) CNN fine-tuning, (2) detector SVM training, and (3) bounding-box regressor training. CNN fine-tuning was run for 50k SGD iteration on $\text{val}_1{+}\text{train}_N$ using the exact same settings as were used for PASCAL. Fine-tuning on a single NVIDIA Tesla K20 took 13 hours using Caffe. For SVM training, all ground-truth boxes from $\text{val}_1{+}\text{train}_N$ were used as positive examples for their respective classes. Hard negative mining was performed on a randomly selected subset of 5000 images from val_1. An initial experiment indicated that mining negatives from all of val_1, versus a 5000 image subset (roughly half of it), resulted in only a 0.5 percentage point drop in mAP, while cutting SVM training time in half. No negative examples were taken from train because the annotations are not exhaustive. The extra sets of verified negative images were not used. The bounding-box regressors were trained on val_1.

[2] Relative imbalance is measured as $|a - b|/(a + b)$ where a and b are class counts in each half of the split.

4.4. Validation and evaluation

Before submitting results to the evaluation server, we validated data usage choices and the effect of fine-tuning and bounding-box regression on the val_2 set using the training data described above. All system hyperparameters (e.g., SVM C hyperparameters, padding used in region warping, NMS thresholds, bounding-box regression hyperparameters) were fixed at the same values used for PASCAL. Undoubtedly some of these hyperparameter choices are slightly suboptimal for ILSVRC, however the goal of this work was to produce a preliminary R-CNN result on ILSVRC without extensive dataset tuning. After selecting the best choices on val_2, we submitted exactly two result files to the ILSVRC2013 evaluation server. The first submission was without bounding-box regression and the second submission was with bounding-box regression. For these submissions, we expanded the SVM and bounding-box regressor training sets to use $\text{val}{+}\text{train}_{1\text{k}}$ and val, respectively. We used the CNN that was fine-tuned on $\text{val}_1{+}\text{train}_{1\text{k}}$ to avoid re-running fine-tuning and feature computation.

4.5. Ablation study

Table 4 shows an ablation study of the effects of different amounts of training data, fine-tuning, and bounding-box regression. A first observation is that mAP on val_2 matches mAP on test very closely. This gives us confidence that mAP on val_2 is a good indicator of test set performance. The first result, 20.9%, is what R-CNN achieves using a CNN pre-trained on the ILSVRC2012 classification dataset (no fine-tuning) and given access to the small amount of training data in val_1 (recall that half of the classes in val_1 have between 15 and 55 examples). Expanding the training set to $\text{val}_1{+}\text{train}_N$ improves performance to 24.1%, with essentially no difference between $N = 500$ and $N = 1000$. Fine-tuning the CNN using examples from just val_1 gives a modest improvement to 26.5%, however there is likely significant overfitting due to the small number of positive training examples. Expanding the fine-tuning set to $\text{val}_1{+}\text{train}_{1\text{k}}$, which adds up to 1000 positive examples per class from the train set, helps significantly, boosting mAP to 29.7%. Bounding-box regression improves results to 31.0%, which is a smaller relative gain that what was observed in PASCAL.

4.6. Relationship to OverFeat

There is an interesting relationship between R-CNN and OverFeat: OverFeat can be seen (roughly) as a special case of R-CNN. If one were to replace selective search region

类别不平衡被选中。[2] 每个候选分割都是通过将验证集图像按其类别计数作为特征进行聚类生成的，随后进行可能改善分割平衡的随机局部搜索。此处使用的特定分割最大相对不平衡约为11%，中位数相对不平衡为4%。用于生成验证集$_1$/验证集$_2$分割的代码将公开提供，以便其他研究人员能基于本报告使用的验证集分割比较其方法。

4.2. 区域提案

我们采用了与PASCAL检测任务相同的区域提议方法。在val$_1$、val$_2$和测试集（但不包括训练集图像）的每张图像上，我们以“快速模式”运行选择性搜索[39]。由于选择性搜索不具备尺度不变性，其生成的区域数量会受图像分辨率影响，因此我们进行了一项微调：ILSVRC数据集的图像尺寸差异极大，从极小尺寸到数百万像素不等，为此我们在运行选择性搜索前将所有图像统一缩放至固定宽度（500像素）。在验证集上，选择性搜索平均每张图像产生2403个区域提议，对全部真实边界框的召回率为91.6%（IoU阈值为0.5）。该召回率明显低于PASCAL数据集约98%的水平，表明区域提议阶段仍有显著改进空间。

4.3. 训练数据

对于训练数据，我们构建了一个包含图像和边界框的集合，该集合囊括了来自val$_1$的所有选择性搜索框和真实标注框，以及来自训练集的每个类别最多N个真实标注框（若某类别在训练集中的真实标注框少于N个，则全部取用）。我们将这个图像与边界框的数据集称为val$_1$+train$_N$。在一项消融实验中，我们展示了在val$_2$上使用$N \in \{0, 500, 1000\}$ (第4.5节)的mAP结果。

R-CNN 需要训练数据用于三个步骤：(1) CNN 微调，(2) 检测器 SVM 训练，以及 (3) 边界框回归器训练。CNN 微调在 val$_1$+train$_N$ 上运行了 50k 次 SGD 迭代，使用的设置与 PASCAL 完全相同。在单块 NVIDIA Tesla K20 上使用 Caffe 进行微调耗时 13 小时。对于 SVM 训练，来自 val$_1$+train$_N$ 的所有真实标注框被用作各自类别的正样本。难负样本挖掘是在从 val$_1$ 中随机选取的 5000 张图像子集上进行的。一项初步实验表明，从整个 val$_1$ 中挖掘负样本与从 5000 张图像子集（约占其一半）中挖掘相比，仅导致 mAP 下降 0.5 个百分点，同时将 SVM 训练时间缩短了一半。负样本未从训练是因为标注并非详尽无遗。额外的已验证负图像集未被使用。边界框回归器是在val$_1$上训练的。

4.4. 验证与评估

在将结果提交至评估服务器之前，我们使用上述训练数据验证了数据使用选择、微调及边界框回归在val$_2$集上的效果。所有系统超参数（例如SVM C超参数、区域变形中使用的填充方式、非极大值抑制阈值、边界框回归超参数）均保持与PASCAL实验相同的设定值。尽管其中部分超参数选择对ILSVRC而言可能略欠优化，但本研究的目标是在未进行大量数据集调优的情况下，为ILSVRC提供初步的R-CNN结果。在val$_2$集上确定最佳方案后，我们向ILSVRC2013评估服务器提交了两份结果文件：第一份未使用边界框回归，第二份则包含边界框回归。针对此次提交，我们分别将SVM和边界框回归器的训练集扩展至val+train$_{1k}$和val集。为避免重复运行微调与特征计算，我们采用了在val$_1$+train$_{1k}$集上微调后的CNN模型。

4.5. 消融研究

表4展示了不同数量训练数据、微调以及边界框回归效果的消融研究。首先观察到在val$_2$上的mAP与测试集上的mAP高度吻合，这使我们确信val$_2$上的mAP能有效反映测试集性能。首个结果20.9%是R-CNN使用ILSVRC2012分类数据集预训练的CNN（未微调）并仅利用val$_1$ (少量训练数据达到的性能——需注意val$_1$中半数类别仅包含15至55个样本)。将训练集扩展至val$_1$+train$_N$后性能提升至24.1%，且N =500与N =1000的设置基本无差异。仅使用val$_1$样本对CNN进行微调可小幅提升至26.5%，但由于正训练样本数量有限，可能存在显著过拟合。将微调集扩展至val$_1$+train$_{1k}$（每类从训练集增加至1000个正样本）带来显著改善，使mAP跃升至29.7%。边界框回归进一步将结果提升至31.0%，其相对增益小于在PASCAL数据集中观察到的效果。

4.6. 与OverFeat的关系

R-CNN与OverFeat之间存在一个有趣的关系：OverFeat可被（粗略地）视为R-CNN的特例。若将选择性搜索区域替换为

[2] Relative imbalance is measured as $|a-b|/(a+b)$ where a and b are class counts in each half of the split.

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**

Table 4: ILSVRC2013 ablation study of data usage choices, fine-tuning, and bounding-box regression.

proposals with a multi-scale pyramid of regular square regions and change the per-class bounding-box regressors to a single bounding-box regressor, then the systems would be very similar (modulo some potentially significant differences in how they are trained: CNN detection fine-tuning, using SVMs, etc.). It is worth noting that OverFeat has a significant speed advantage over R-CNN: it is about 9x faster, based on a figure of 2 seconds per image quoted from [34]. This speed comes from the fact that OverFeat's sliding windows (i.e., region proposals) are not warped at the image level and therefore computation can be easily shared between overlapping windows. Sharing is implemented by running the entire network in a convolutional fashion over arbitrary-sized inputs. Speeding up R-CNN should be possible in a variety of ways and remains as future work.

5. Semantic segmentation

Region classification is a standard technique for semantic segmentation, allowing us to easily apply R-CNN to the PASCAL VOC segmentation challenge. To facilitate a direct comparison with the current leading semantic segmentation system (called O_2P for "second-order pooling") [4], we work within their open source framework. O_2P uses CPMC to generate 150 region proposals per image and then predicts the quality of each region, for each class, using support vector regression (SVR). The high performance of their approach is due to the quality of the CPMC regions and the powerful second-order pooling of multiple feature types (enriched variants of SIFT and LBP). We also note that Farabet et al. [16] recently demonstrated good results on several dense scene labeling datasets (not including PASCAL) using a CNN as a multi-scale per-pixel classifier.

We follow [2, 4] and extend the PASCAL segmentation training set to include the extra annotations made available by Hariharan et al. [22]. Design decisions and hyperparameters were cross-validated on the VOC 2011 validation set. Final test results were evaluated only once.

CNN features for segmentation. We evaluate three strategies for computing features on CPMC regions, all of which begin by warping the rectangular window around the region to 227×227. The first strategy (*full*) ignores the region's shape and computes CNN features directly on the warped window, exactly as we did for detection. However, these features ignore the non-rectangular shape of the region. Two regions might have very similar bounding boxes while having very little overlap. Therefore, the second strategy (*fg*) computes CNN features only on a region's foreground mask. We replace the background with the mean input so that background regions are zero after mean subtraction. The third strategy (*full*+*fg*) simply concatenates the *full* and *fg* features; our experiments validate their complementarity.

	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

Table 5: Segmentation mean accuracy (%) on VOC 2011 validation. Column 1 presents O_2P; 2-7 use our CNN pre-trained on ILSVRC 2012.

Results on VOC 2011. Table 5 shows a summary of our results on the VOC 2011 validation set compared with O_2P. (See Appendix E for complete per-category results.) Within each feature computation strategy, layer fc_6 always outperforms fc_7 and the following discussion refers to the fc_6 features. The *fg* strategy slightly outperforms *full*, indicating that the masked region shape provides a stronger signal, matching our intuition. However, *full*+*fg* achieves an average accuracy of 47.9%, our best result by a margin of 4.2% (also modestly outperforming O_2P), indicating that the context provided by the *full* features is highly informative even given the *fg* features. Notably, training the 20 SVRs on our *full*+*fg* features takes an hour on a single core, compared to 10+ hours for training on O_2P features.

In Table 6 we present results on the VOC 2011 test set, comparing our best-performing method, fc_6 (*full*+*fg*), against two strong baselines. Our method achieves the highest segmentation accuracy for 11 out of 21 categories, and the highest overall segmentation accuracy of 47.9%, averaged across categories (but likely ties with the O_2P result under any reasonable margin of error). Still better performance could likely be achieved by fine-tuning.

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中数据使用选择、微调及边界框回归的消融研究。

提出一种多尺度金字塔结构的规则方形区域建议，并将每个类别的边界框回归器改为单一的边界框回归器，那么这两个系统将非常相似（除了在训练方式上可能存在一些显著差异：如CNN检测的微调、使用支持向量机等）。值得注意的是，OverFeat在速度上相比R-CNN具有显著优势：根据[34]引用的每张图像2秒的数据，其速度大约快9倍。这一速度优势源于OverFeat的滑动窗口（即区域建议）在图像层面未进行扭曲处理，因此计算可以在重叠窗口间轻松共享。这种共享是通过以卷积方式在整个网络上对任意尺寸输入进行运算来实现的。通过多种方式加速R-CNN应当是可行的，这仍是未来的研究方向。

5. 语义分割

区域分类是语义分割的标准技术，它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（称为O_2P，意为“二阶池化”）[4]进行直接比较，我们在其开源框架内展开工作。O_2P使用CPMC为每张图像生成150个区域建议，然后利用支持向量回归（SVR）为每个类别预测每个区域的质量。该方法的高性能得益于CPMC区域的质量以及多种特征类型（SIFT和LBP的增强变体）的强大二阶池化。我们还注意到，Farabet等人[16]最近使用CNN作为多尺度逐像素分类器，在多个密集场景标注数据集（不包括PASCAL）上取得了良好结果。

我们遵循[2, 4]的方法，扩展了PASCAL分割训练集，以包含Hariharan等人[22]提供的额外标注。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分割特征。我们评估了在CPMC区域上计算特征的三种策略，所有这些策略都始于将区域周围的矩形窗口变形为227×227。第一种策略（*full*）忽略了区gion的形状，并直接在扭曲的窗口上计算CNN特征，正如我们在检测中所做的那样。然而，这些特征忽略了区域非矩形的形状。两个区域可能具有非常相似的边界框，但重叠部分却很少。因此，第二种策略（*fg*）仅在区域的前景掩码上计算CNN特征。我们将背景替换为平均输入，这样在减去均值后背景区域就变为零。第三种策略（*full*+*fg*）简单地将*full*和*fg*特征连接起来；我们的实验验证了它们的互补性。

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

表5：在VOC 2011验证集上的分割平均准确率（%）。第1列展示O_2P；第2-7列使用我们在ILSVRC 2012上预训练的CNN。

在VOC 2011上的结果。表5总结了我们在VOC 2011验证集上的结果，并与O_2P进行了比较。（完整的分类别结果见附录E。）在每种特征计算策略中，层fc_6始终优于fc_7，因此后续讨论将围绕fc_6特征展开。*fg*策略略优于*full*，表明掩码区域形状提供了更强的信号，这与我们的直觉相符。然而，*full*+*fg*取得了47.9%的平均准确率，这是我们的最佳结果，领先4.2%（同时也小幅超越了O_2P），这表明即使已有*fg*特征，*full*特征所提供的上下文信息仍具有很高的价值。值得注意的是，在单核处理器上，基于我们的*full*+*fg*特征训练20个SVR仅需一小时，而在O_2P特征上训练则需要10+小时。

在表6中，我们展示了在VOC 2011测试集上的结果，将我们性能最佳的方法fc_6 (*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个上取得了最高的分割精度，并以47.9%的整体分割精度位居首位，该数值为跨类别平均值（但在任何合理的误差范围内，很可能与O_2P的结果持平）。通过微调，仍有可能实现更好的性能。

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	**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**

Table 6: Segmentation accuracy (%) on VOC 2011 test. We compare against two strong baselines: the "Regions and Parts" (R&P) method of [2] and the second-order pooling (O_2P) method of [4]. Without any fine-tuning, our CNN achieves top segmentation performance, outperforming R&P and roughly matching O_2P.

6. Conclusion

In recent years, object detection performance had stagnated. The best performing systems were complex ensembles combining multiple low-level image features with high-level context from object detectors and scene classifiers. This paper presents a simple and scalable object detection algorithm that gives a 30% relative improvement over the best previous results on PASCAL VOC 2012.

We achieved this performance through two insights. The first is to apply high-capacity convolutional neural networks to bottom-up region proposals in order to localize and segment objects. The second is a paradigm for training large CNNs when labeled training data is scarce. We show that it is highly effective to pre-train the network—*with supervision*—for a auxiliary task with abundant data (image classification) and then to fine-tune the network for the target task where data is scarce (detection). We conjecture that the "supervised pre-training/domain-specific fine-tuning" paradigm will be highly effective for a variety of data-scarce vision problems.

We conclude by noting that it is significant that we achieved these results by using a combination of classical tools from computer vision *and* deep learning (bottom-up region proposals and convolutional neural networks). Rather than opposing lines of scientific inquiry, the two are natural and inevitable partners.

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.

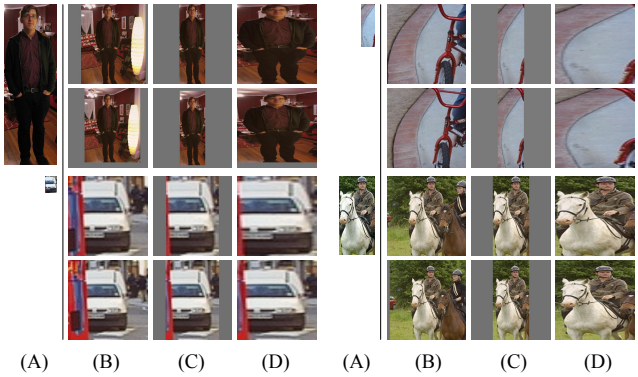


Figure 7: Different object proposal transformations. (A) the original object proposal at its actual scale relative to the transformed CNN inputs; (B) tightest square with context; (C) tightest square without context; (D) warp. Within each column and example proposal, the top row corresponds to $p = 0$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

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：在VOC 2011测试集上的分割准确率（%）。我们与两个强基线方法进行了比较：[2]提出的“区域与部件”（R&P）方法，以及[4]提出的二阶池化（O_2P）方法。在未经任何微调的情况下，我们的CNN取得了最佳分割性能，超越了R&P方法，并与O_2P方法大致相当。

6. 结论

近年来，物体检测性能一度停滞不前。表现最佳的系统是复杂的集成模型，它们将多种低层图像特征与来自物体检测器和场景分类器的高层上下文信息相结合。本文提出了一种简单且可扩展的物体检测算法，在PASCAL VOC 2012数据集上相比先前最佳结果实现了30%的相对提升。

我们通过两个洞见实现了这一性能。首先，将高容量卷积神经网络应用于自下而上的区域建议，以定位和分割物体。其次，提出了一种在标注训练数据稀缺时训练大型CNN的范式。我们证明，先在数据充足的任务（图像分类）上对网络——*with supervision*——进行预训练，再针对数据稀缺的目标任务（检测）对网络进行微调，这种方法极为有效。我们推测，“监督预训练/领域特定微调”的范式将对各类数据稀缺的视觉问题非常有效。

最后需要指出的是，我们通过结合计算机视觉领域的经典工具与深度学习技术（自下而上的区域建议和卷积神经网络）取得了这些成果，这一点具有重要意义。这两者并非对立的研究路径，而是天然且必然的合作伙伴。

致谢。本研究部分得到了DARPA的“心灵之眼”和MSEE项目、NSF奖项IIS-0905647、IIS-1134072和IIS-1212798、MURI N000014-10-1-0933以及丰田公司的支持。本研究中使用的GPU由英伟达公司慷慨捐赠。

附录

A. 物体提议变换

本工作中使用的卷积神经网络需要227×227像素的固定尺寸输入。对于检测任务，我们考虑任意图像矩形区域作为候选目标框。我们评估了两种将候选目标框转换为有效CNN输入的方法。

第一种方法（“带上下文的最紧正方形”）将每个对象提议包围在最紧的正方形内，并且

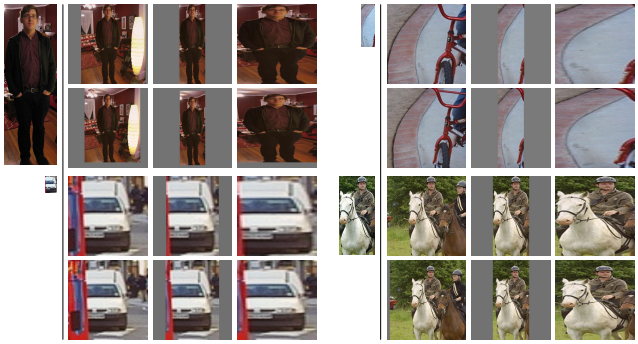


图7：不同的物体候选框变换方式。(A) 原始物体候选框相对于变换后CNN输入的实际尺度；(B) 带上下文的最紧凑正方形；(C) 无上下文的最紧凑正方形；(D) 形变处理。每列示例候选框中，顶行对应$p = 0$像素的上下文填充，底行对应$p = 16$像素的上下文填充。

然后将该正方形内的图像（各向同性地）缩放至CNN输入尺寸。图7列(B)展示了这一变换。此方法的一个变体（“无上下文最紧凑正方形”）排除了原始物体提议周围的图像内容。图7列(C)展示了这一变换。第二种方法（“拉伸”）将每个物体提议各向异性地缩放至CNN输入尺寸。图7列(D)展示了拉伸变换。

对于每一种变换，我们同样考虑在原物体提议周围加入额外的图像上下文。上下文填充量（p）定义为在变换后的输入坐标框架中，原物体提议周围的边框大小。图7展示了每个示例顶部行中$p = 0$像素和底部行中$p = 16$像素的情况。在所有方法中，若源矩形超出图像范围，缺失数据将以图像均值替代（随后在将图像输入CNN前减去该均值）。初步实验表明，采用上下文填充（$p = 16$像素）的形变方法显著优于其他方案（提升3-5个mAP点）。显然还存在更多可行方案，例如使用复制填充替代均值填充。对这些方案的全面评估将留待未来工作。

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

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

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

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

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

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

Each function $d_\star(P)$ (where $\star$ is one of x, y, h, w) is modeled as a linear function of the pool_5 features of proposal P, denoted by $\boldsymbol{\phi}_5(P)$. (The dependence of $\boldsymbol{\phi}_5(P)$ on the image data is implicitly assumed.) Thus we have $d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P)$, where $\mathbf{w}_\star$ is a vector of learnable model parameters. We learn $\mathbf{w}_\star$ by optimizing the regularized least squares objective (ridge regression):

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 . \tag{5}$$

B. 正例与负例及softmax

两个设计选择值得进一步讨论。第一个是：为何在微调CNN与训练目标检测SVM时，正负样本的定义方式不同？简要回顾定义：在微调阶段，我们将每个候选区域映射到与其IoU重叠度最高的真实标注实例（若存在），若IoU至少达到0.5，则将其标记为该匹配真实类别的正样本。其余所有候选区域均标记为“背景”（即所有类别的负样本）。相比之下，在训练SVM时，我们仅将真实标注框作为其对应类别的正样本，并将与某类别所有实例IoU重叠度均低于0.3的候选区域标记为该类别的负样本。落入灰色区域（IoU重叠度高于0.3但非真实标注）的候选区域则被忽略。

从历史角度来看，我们之所以得出这些定义，是因为最初使用ImageNet预训练CNN提取的特征训练支持向量机，当时并未考虑微调。在该实验设置下，我们发现所采用的特定标签定义在评估的所有方案中（包括当前用于微调的设置）是最优的。当我们开始采用微调方法时，最初沿用了与支持向量机训练相同的正负样本定义。然而，我们发现其结果远不如采用当前正负样本定义所获得的效果。

我们的假设是，正负样本定义方式的这种差异并非根本性重要，而是源于微调数据有限这一事实。我们当前的方案引入了许多“抖动”样本（即重叠度在0.5到1之间但非真实标注的候选框），这使正样本数量扩大了约30倍。我们推测，在微调{v*}网络时需要如此大量的样本以避免过拟合。但我们也注意到，使用这些抖动样本可能并非最优选择，因为网络并未针对精确定位进行微调。

这就引出了第二个问题：为什么在微调之后，还要训练SVM呢？更简洁的做法是直接使用微调网络的最后一层——一个21路softmax回归分类器——作为目标检测器。我们尝试了这种方法，发现在VOC 2007数据集上的性能从54.2% mAP下降到了50.9%。这种性能下降可能源于几个因素的综合作用，包括微调中使用的正样本定义并未强调精确定位，以及softmax分类器是在随机采样的负样本上训练的，而非使用SVM训练时所用的“难负样本”子集。

这一结果表明，无需重新训练支持向量机，也有可能获得接近相同水平的性能——经过精细调整后，我们推测，若对微调过程进行一些额外调整，或许能弥合剩余的性能差距。若此推测成立，这将能在不损失检测性能的前提下，简化并加速R-CNN的训练过程。

C. 边界框回归

我们采用一个简单的边界框回归阶段来提升定位性能。在对每个选择性搜索建议使用类别特定的检测SVM进行评分后，我们利用类别特定的边界框回归器为检测预测一个新的边界框。这在理念上与可变形部件模型中使用的边界框回归相似[17]。两种方法的主要区别在于，这里我们基于CNN计算的特征进行回归，而非基于推断出的DPM部件位置所计算的几何特征。

我们训练算法的输入是一组N训练对$\{(P^i, G^i)\}_{i=1,\ldots,N}$，其中$P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$指定了候选框$P^i$边界框中心的像素坐标及其像素宽度和高度$P^i$。此后除非必要，我们将省略上标$i$。每个真实边界框$G$以相同方式指定：$G = (G_x, G_y, G_w, G_h)$。我们的目标是学习一个将候选框$P$映射到真实框$G$的变换。

我们将变换参数化为四个函数$d_x(P)$、$d_y(P)$、$d_w(P)$和$d_h(P)$。前两个函数指定了P边界框中心的尺度不变平移，而后两个函数则指定了P边界框宽度和高度的对数空间平移。学习这些函数后，我们可以通过应用该变换将输入提议P转换为预测的真实框$\hat{G}$。

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

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

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

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

每个函数 $d_\star(P)$ (，其中 $\star$ 是 x, y, h, w) 之一，被建模为提案 P 的池化特征 $_5$ 的线性函数，记作 $\boldsymbol{\phi}_5(P)$。（这里隐含假设了 $\boldsymbol{\phi}_5(P)$ 对图像数据的依赖性。）因此我们有 $d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P)$，其中 $\mathbf{w}_\star$ 是可学习的模型参数向量。我们通过优化正则化最小二乘目标（岭回归）来学习 $\mathbf{w}_\star$：

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 . \tag{5}$$

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

$$t_x = (G_x - P_x)/P_w \quad (6)$$
$$t_y = (G_y - P_y)/P_h \quad (7)$$
$$t_w = \log(G_w/P_w) \quad (8)$$
$$t_h = \log(G_h/P_h). \quad (9)$$

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

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

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

D. Additional feature visualizations

Figure 12 shows additional visualizations for 20 pool_5 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.

We label each unit by its (y, x, channel) position in the $6 \times 6 \times 256$ dimensional pool_5 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 O_2P method [4]. These results show which methods are strongest across each of the 20 PASCAL classes, plus the background class.

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 near-duplicate) 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).

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，这可以通过闭式解高效求解。

在实现边界框回归时，我们发现了两个细微的问题。首先是正则化很重要：我们基于验证集将 $\lambda =$ 设为1000。第二个问题是在选择训练对 (P, G) 时必须谨慎。直观地说，如果 P 远离所有真实标注框，那么将 P 转换到真实标注框 G 的任务就没有意义。使用像 P 这样的样本会导致学习问题无法解决。因此，我们仅当一个提议 P 至少与一个真实标注框 *nearby* 时，才从中学习。我们通过以下方式实现“邻近性”：当且仅当重叠度大于阈值（我们使用验证集将其设为 0.6）时，将 P 分配给与其具有最大 IoU 重叠度的真实标注框 G（若其与多个框重叠）。所有未分配的提议均被丢弃。我们对每个对象类别执行一次此操作，以学习一组特定类别的边界框回归器。

在测试时，我们对每个提议进行评分，并仅预测一次其新的检测窗口。原则上，我们可以迭代此过程（即重新评分新预测的边界框，然后从中预测新的边界框，依此类推）。然而，我们发现迭代并不会改善结果。

D. 附加特征可视化

图12展示了20个池化$_5$单元的额外可视化结果。对于每个单元，我们从VOC 2007测试集全部约1000万个区域提案中，选取了最能激活该单元的24个区域提案进行展示。

我们将每个单元按其位于$6 \times 6 \times 256$维池化$_5$特征图中的（y, x, 通道）位置进行标注。在每个通道内，CNN对输入区域计算完全相同的函数，仅（y, x）位置的变化会改变感受野。

E. 按类别分割结果

在表7中，我们展示了除O$_2$P方法[4]外，我们六种分割方法在VOC 2011验证集上每个类别的分割准确率。这些结果显示了在20个PASCAL类别及背景类别中，每种方法在不同类别上的最强表现。

F. 跨数据集冗余分析

在使用辅助数据集进行训练时，一个值得关注的问题是它可能与测试集之间存在冗余。尽管目标检测和全图分类任务存在显著差异，使得这种跨数据集冗余的担忧大为减轻，我们仍进行了深入调查，量化了PASCAL测试图像在ILSVRC 2012训练集和验证集中的包含程度。我们的研究结果或许能为有意使用ILSVRC 2012作为PASCAL图像分类任务训练数据的研究者提供参考。

我们对重复（及近似重复）图像进行了两项检查。首项测试基于flickr图像ID的精确匹配，这些ID包含在VOC 2007测试标注中（后续PASCAL测试集有意对这些ID保密）。所有PASCAL图像及约半数ILSVRC图像均采集自flickr.com。此项检查在4952张图像中发现31组匹配（占比0.63%）。

第二次检查采用GIST[30]描述符匹配方法，该方法在[13]中被证明在大型（约100万张）图像集合中进行近重复图像检测时具有优异性能。依照[13]的方法，我们对所有ILSVRC 2012训练验证集图像和PASCAL 2007测试图像进行扭曲处理后的32×32像素版本计算了GIST描述符。

基于GIST描述符的欧几里得距离最近邻匹配揭示了38张近似重复图像（包括通过flickr ID匹配找到的全部31张）。这些匹配图像在JPEG压缩级别和分辨率上往往略有差异，裁剪程度则相对较小。这些发现表明重叠部分很小，不足1%。对于VOC 2012数据集，由于无法获取flickr ID，我们仅采用GIST匹配方法。根据GIST匹配结果，VOC 2012测试图像中有1.5%出现在ILSVRC 2012训练验证集中。VOC 2012的重叠率略高，很可能是因为这两个数据集的收集时间间隔比VOC 2007与ILSVRC 2012的间隔更接近。

G. 文档变更日志

本文档追踪R-CNN的进展。为帮助读者了解其随时间的变化，以下简要记录了修订版本的变化日志。

v1 初始版本。

v2 CVPR 2014 最终修订版。检测性能得到显著提升，主要得益于以下改进：(1) 从更高的学习率（0.001 而非 0.0001）开始微调，(2) 在准备 CNN 输入时使用上下文填充，以及 (3) 采用边界框回归来修正定位误差。

v3在ILSVRC2013检测数据集上的结果以及与OverFeat的比较已整合至多个章节（主要为第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_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.

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. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 10, 11

[3] P. Arbeláez, 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. 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. 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. 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. 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. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 2

[22] B. Hariharan, P. Arbeláez, 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. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998. 1

[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

VOC 2011 验证集	背景	飞机	自行车	鸟	船	瓶子	巴士	汽车	猫	椅子	牛	桌子	狗	马	摩托车	人	植物	羊	沙发	火车	电视	平均值
O_2P [4]	84.0	69.0	21.7	47.7	42.2	42.4	64.7	65.8	57.4	12.9	37.4	20.5	43.7	35.7	52.7	51.0	35.8	51.0	28.4	59.8	49.7	46.4
full R-CNN fc_6	81.3	56.2	23.9	42.9	40.7	38.8	59.2	56.5	53.2	11.4	34.6	16.7	48.1	37.0	51.4	46.0	31.5	44.0	24.3	53.7	51.1	43.0
full R-CNN fc_7	81.0	52.8	25.1	43.8	40.5	42.7	55.4	57.7	51.3	8.7	32.5	11.5	48.1	37.0	50.5	46.4	30.2	42.1	21.2	57.7	56.0	42.5
fg R-CNN fc_6	81.4	54.1	21.1	40.6	38.7	53.6	59.9	57.2	52.5	9.1	36.5	23.6	46.4	38.1	53.2	51.3	32.2	38.7	29.0	53.0	47.5	43.7
fg R-CNN fc_7	80.9	50.1	20.0	40.2	34.1	40.9	59.7	59.8	52.7	7.3	32.1	14.3	48.8	42.9	54.0	48.6	28.9	42.6	24.9	52.2	48.8	42.1
full+fg R-CNN fc_6	83.1	60.4	23.2	48.4	47.3	52.6	61.6	60.6	59.1	10.8	45.8	20.9	57.7	43.3	57.4	52.9	34.7	48.7	28.1	60.0	48.6	47.9
full+fg R-CNN fc_7	82.3	56.7	20.6	49.9	44.2	43.6	59.3	61.3	57.8	7.7	38.4	15.1	53.4	43.7	50.8	52.0	34.1	47.8	24.7	60.1	55.2	45.7

表7：VOC 2011验证集上各类别的分割准确率（%）。

v4 附录B中的softmax与SVM对比结果存在一处错误，现已修正。感谢Sergio Guadarrama协助发现此问题。

v5 在3.3节和表3中增加了使用Simonyan和Zisserman [43]提出的新16层网络架构的实验结果。

参考文献

[1] B. Alexe、T. Deselaers 和 V. Ferrari。测量图像窗口的物体性。*TPAMI*，2012年。2 [2] P. Arbeláez、B. Hariharan、C. Gu、S. Gupta、L. Bourdev 和 J. Malik。使用区域和部件进行语义分割。收录于 *CVPR*，2012年。10、11 [3] P. Arbeláez、J. Pont-Tuset、J. Barron、F. Marques 和 J. Ma- lik。多尺度组合分组。收录于 *CVPR*，2014年。3 [4] J. Carreira、R. Caseiro、J. Batista 和 C. Sminchisescu。基于二阶池化的语义分割。收录于 *ECCV*，2012年。4、10、11、13、14 [5] J. Carreira 和 C. Sminchisescu。CPMC：使用约束参数最小割的自动物体分割。*TPAMI*，2012年。2、3 [6] D. Ciresan、A. Giusti、L. Gambardella 和 J. Schmidhu- ber。使用深度神经网络检测乳腺癌组织图像中的有丝分裂。收录于 *MICCAI*，2013年。3 [7] N. Dalal 和 B. Triggs。用于人体检测的定向梯度直方图。收录于 *CVPR*，2005年。1 [8] T. Dean、M. A. Ruzon、M. Segal、J. Shlens、S. Vijaya- narasimhan 和 J. Yagnik。在单台机器上快速准确检测10万个物体类别。收录于 *CVPR*，2013年。3 [9] J. Deng、A. Berg、S. Satheesh、H. Su、A. Khosla 和 L. Fei- Fei。ImageNet 2012大规模视觉识别挑战赛 (ILSVRC2012)。http://www.image-net.org/challenges/LSVRC/2012/。1 [10] J. Deng、W. Dong、R. Socher、L.-J. Li、K. Li 和 L. Fei- Fei。ImageNet：一个大规模分层图像数据库。收录于 *CVPR*，2009年。1 [11] J. Deng、O. Russakovsky、J. Krause、M. Bernstein、A. C. Berg 和 L. Fei-Fei。可扩展的多标签标注。收录于 *CHI*，2014年。8 [12] J. Donahue、Y. Jia、O. Vinyals、J. Hoffman、N. Zhang、E. Tzeng 和 T. Darrell。DeCAF：用于通用视觉识别的深度卷积激活特征。收录于 *ICML*，2014年。2

[13] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, 与 C. Schmid。面向网络规模图像搜索的Gist描述子评估。载于 *Proc. of the ACM International Conference on Image and Video Retrieval*，2009年。13[14] I. Endres 与 D. Hoiem。类别无关的目标提议。载于 *ECCV*，2010年。3[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, 与 A. Zisserman。PASCAL视觉目标类别（VOC）挑战赛。*IJCV*，2010年。1, 4[16] C. Farabet, C. Couprie, L. Najman, 与 Y. LeCun。用于场景标注的层次特征学习。*TPAMI*，2013年。10[17] P. Felzenszwalb, R. Girshick, D. McAllester, 与 D. Ramanan。基于判别性训练部件模型的目标检测。*TPAMI*，2010年。2, 4, 7, 12[18] S. Fidler, R. Mottaghi, A. Yuille, 与 R. Urtasun。自底向上分割用于自顶向下检测。载于 *CVPR*，2013年。4, 5[19] K. Fukushima。新认知机：一种不受位置偏移影响的模式识别机制的自组织神经网络模型。*Biological cybernetics*，36(4):193–202，1980年。1[20] R. Girshick, P. Felzenszwalb, 与 D. McAllester。判别性训练的可变形部件模型，第5版。http: //www.cs.berkeley.edu/rbg/latent-v5/。2, 5, 6, 7[21] C. Gu, J. J. Lim, P. Arbeláez, 与 J. Malik。基于区域的识别。载于 *CVPR*，2009年。2[22] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, 与 J. Malik。来自逆向检测器的语义轮廓。载于 *ICCV*，2011年。10[23] D. Hoiem, Y. Chodpathumwan, 与 Q. Dai。目标检测器中的错误诊断。载于 *ECCV*，2012年。2, 7, 8[24] Y. Jia。Caffe：一种用于快速特征嵌入的开源卷积架构。http://caffe.berkeleyvision.org/，2013年。3[25] A. Krizhevsky, I. Sutskever, 与 G. Hinton。使用深度卷积神经网络进行ImageNet分类。载于 *NIPS*，2012年。1, 3, 4, 7[26] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, 与 L. Jackel。反向传播应用于手写邮政编码识别。*Neural Comp.*，1989年。1[27] Y. LeCun, L. Bottou, Y. Bengio, 与 P. Haffner。基于梯度的学习应用于文档识别。*Proc. of the IEEE*，1998年。1[28] J. J. Lim, C. L. Zitnick, 与 P. Dollár。草图标记：一种用于轮廓和目标检测的学习中层表示。载于 *CVPR*，2013年。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	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

Table 8: Per-class average precision (%) on the ILSVRC2013 detection test set.

[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

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

表8：ILSVRC2013检测测试集上各类别的平均精度（%）。

[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: 空间包络的整体表示。*IJCV*，2001年。13 [31] X. Ren与D. Ramanan。稀疏编码直方图用于

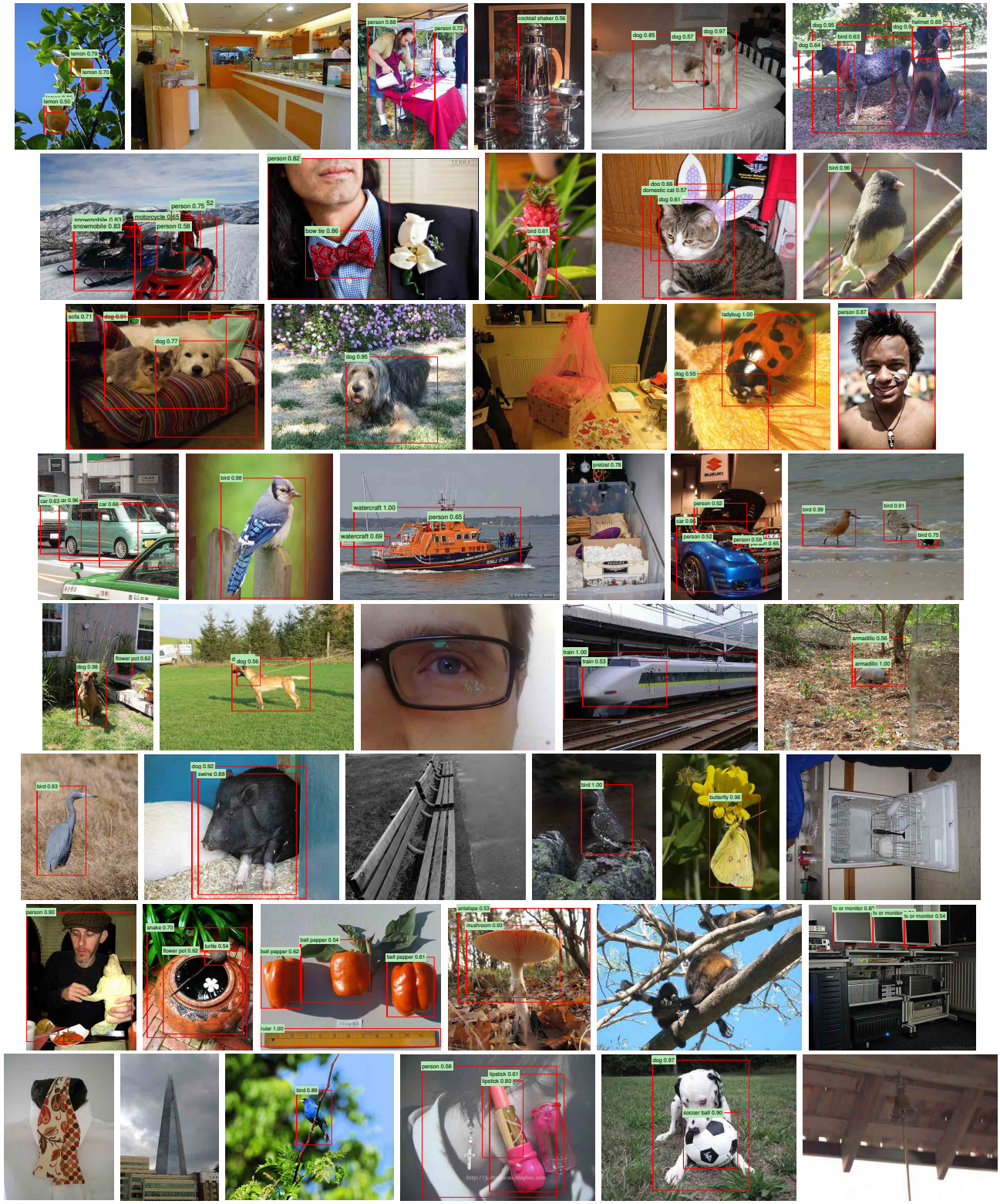


Figure 8: Example detections on the val_2 set from the configuration that achieved 31.0% mAP on val_2. Each image was sampled randomly (these are *not* curated). All detections at precision greater than 0.5 are shown. Each detection is labeled with the predicted class and the precision value of that detection from the detector's precision-recall curve. Viewing digitally with zoom is recommended.

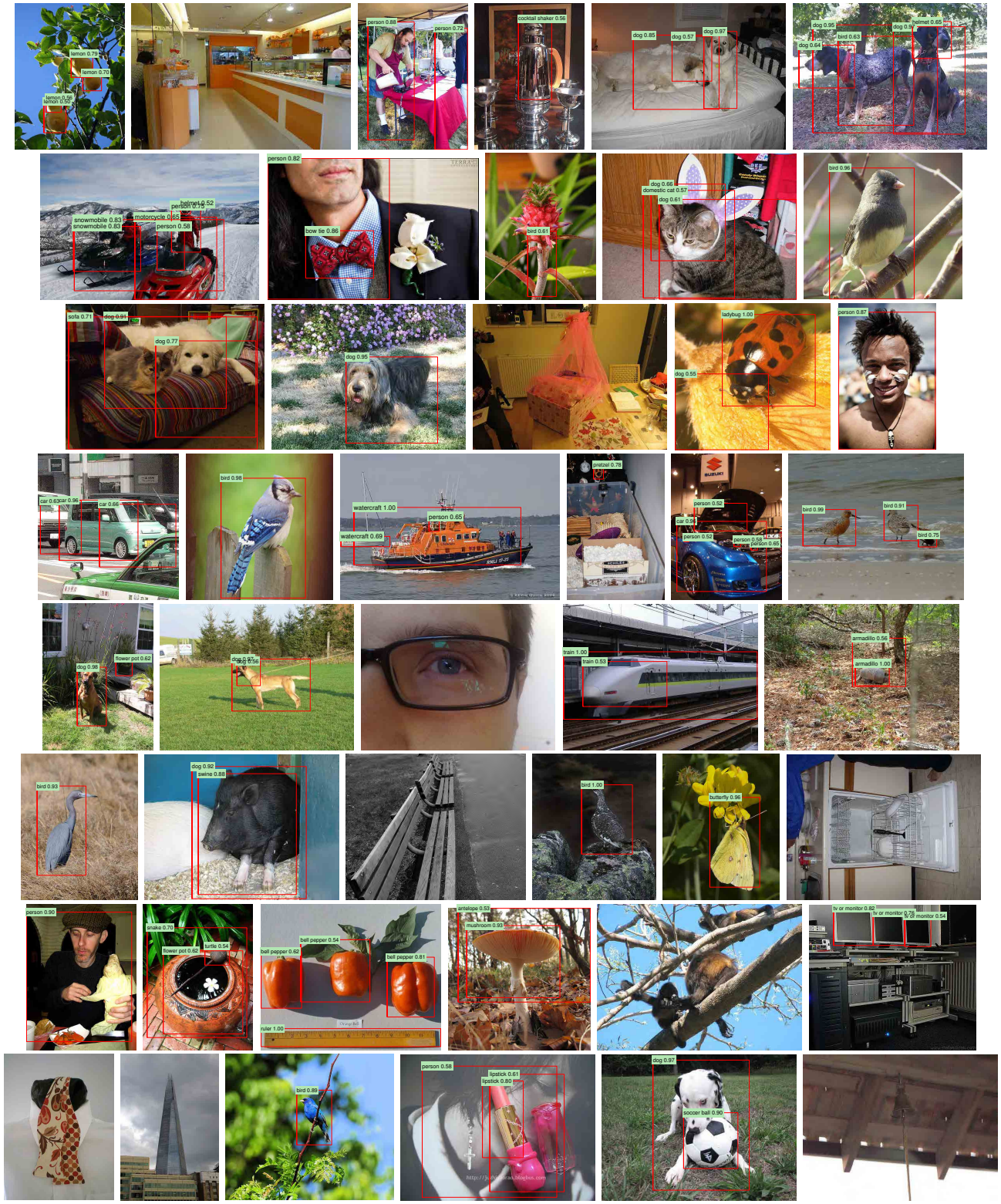


图8：在val_2集上的检测示例，来自在val_2上达到31.0% mAP的配置。每张图像均为随机采样（这些图像经过*not*筛选）。所有精度大于0.5的检测结果均已显示。每个检测结果均标注了预测类别及该检测在检测器精度-召回曲线中的精度值。建议使用数字设备缩放查看。

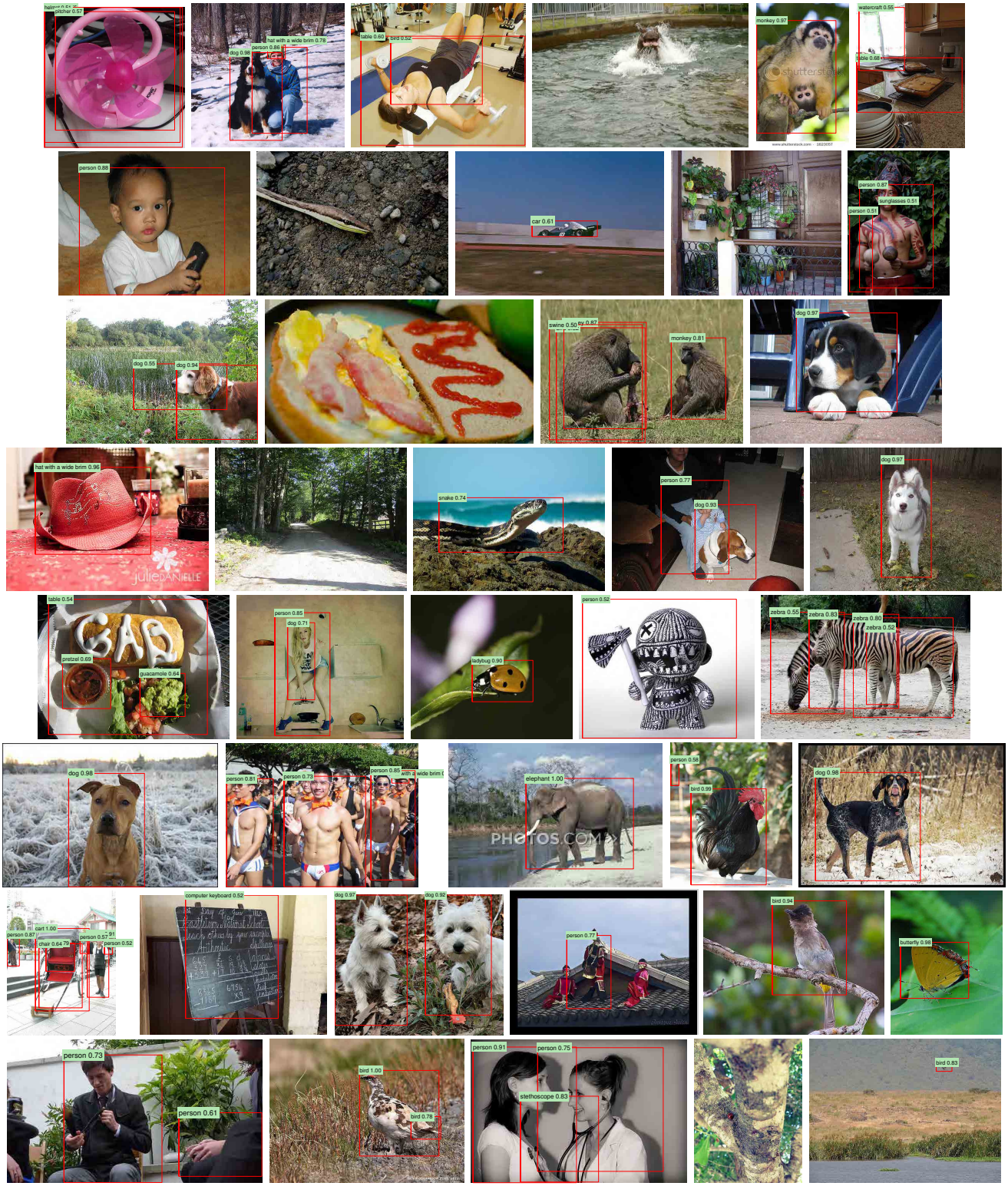


Figure 9: More randomly selected examples. See Figure 8 caption for details. Viewing digitally with zoom is recommended.

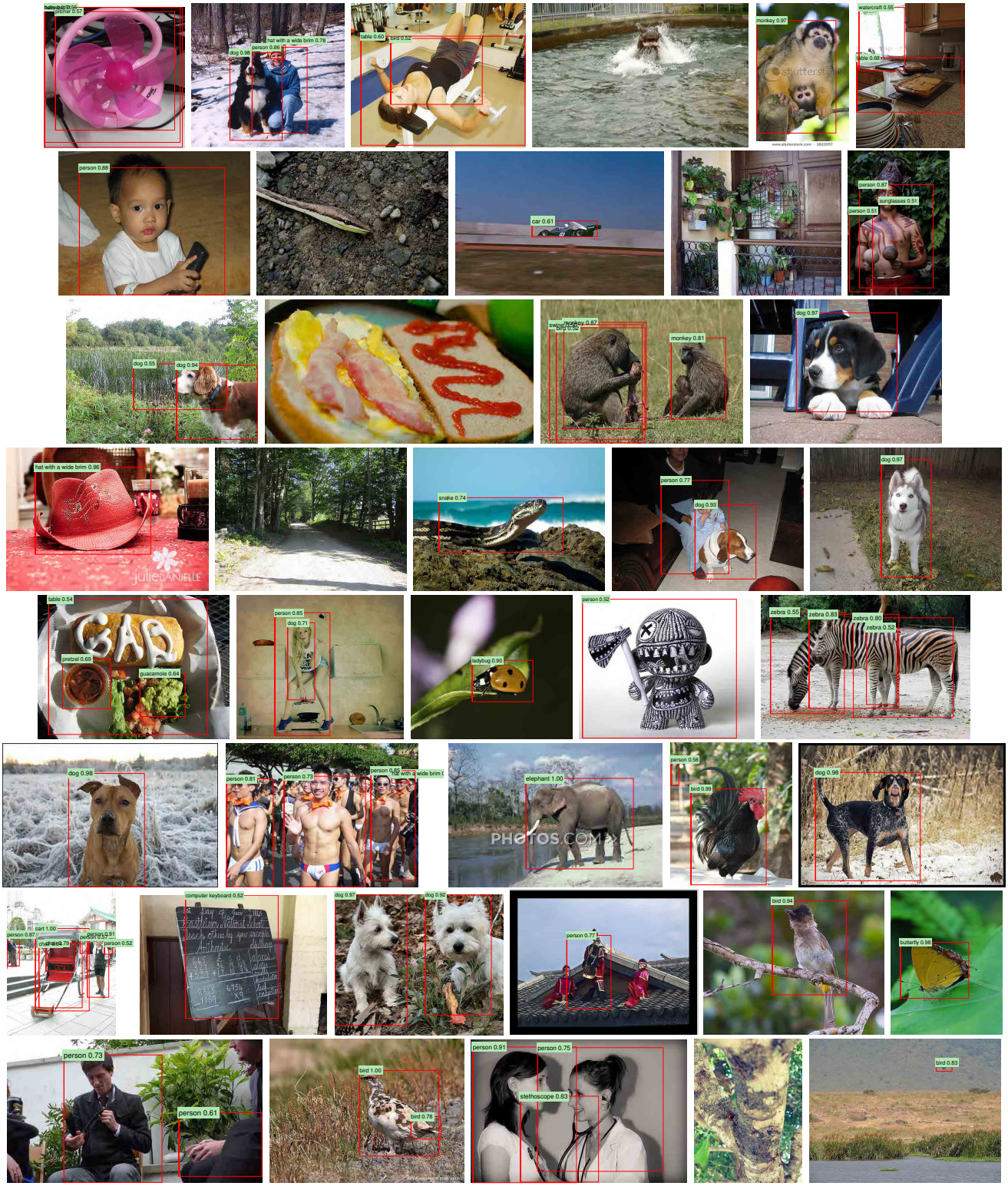


图9：更多随机选取的示例。详情见图8说明。建议数字缩放查看。

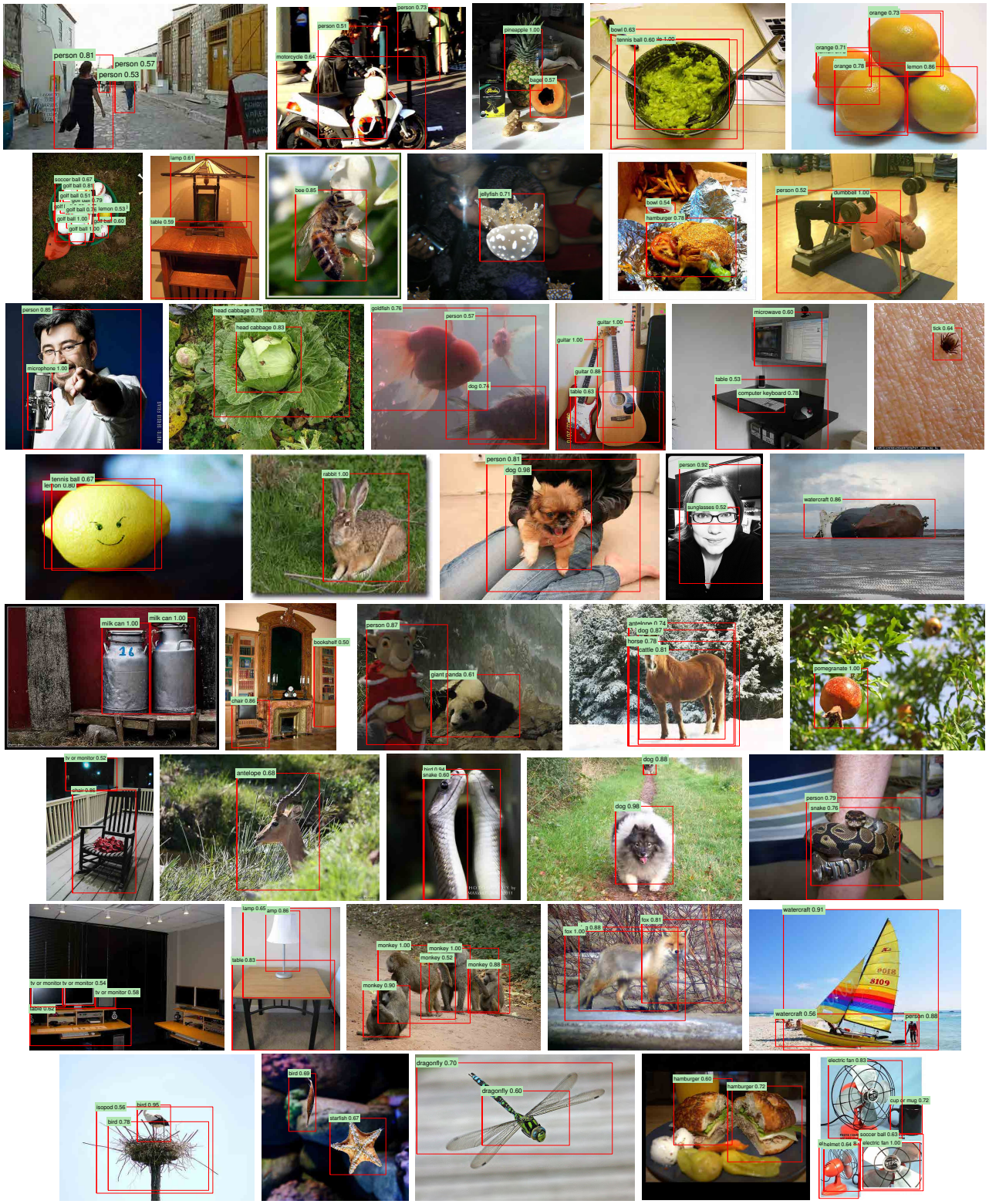


Figure 10: Curated examples. Each image was selected because we found it impressive, surprising, interesting, or amusing. Viewing digitally with zoom is recommended.

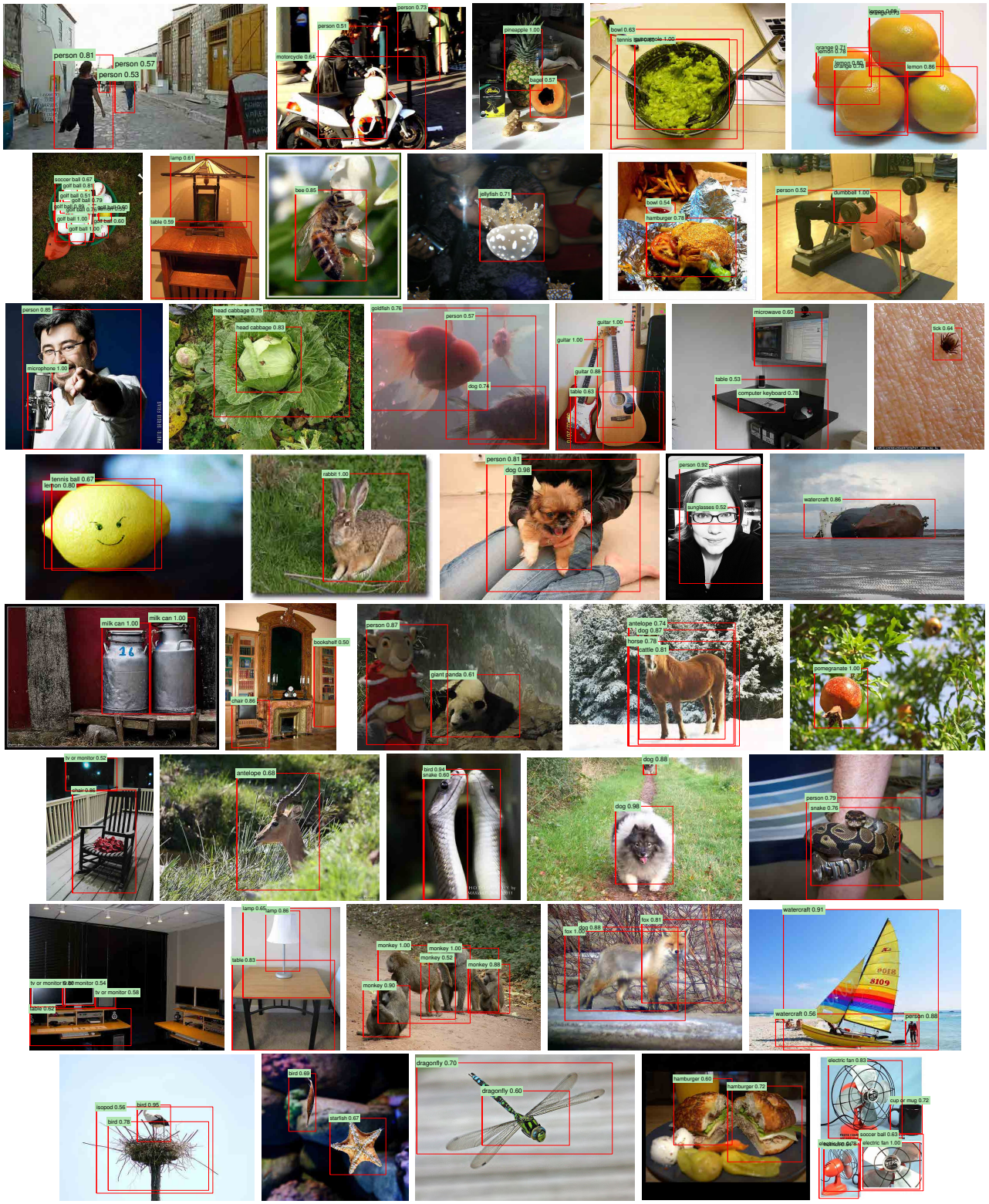


图10：精选示例。每张图片的选取皆因我们觉得其令人印象深刻、出乎意料、有趣或引人发笑。建议在数字设备上放大查看。

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, 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

目标检测。在*CVPR*，2013年。6, 7 [32] H. A. Rowley, S. Baluja, 和 T. Kanade。基于神经网络的人脸检测。*TPAMI*，1998年。2 [33] D. E. Rumelhart, G. E. Hinton, 和 R. J. Williams。通过误差传播学习内部表示。*Parallel Distributed Processing*，1:318–362，1986年。1 [34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, 和 Y. LeCun。OverFeat：使用卷积网络进行集成识别、定位与检测。在*ICLR*，2014年。1, 2, 4, 10 [35] P. Sermanet, K. Kavukcuoglu, S. Chintala, 和 Y. LeCun。使用无监督多阶段特征学习的行人检测。在*CVPR*，2013年。2 [36] H. Su, J. Deng, 和 L. Fei-Fei。用于视觉目标检测的众包标注。在*AAAI Technical Report, 4th Human Computation Workshop*，2012年。8 [37] K. Sung 和 T. Poggio。基于示例的视角人脸检测学习。技术报告 A.I. Memo No. 1521，麻省理工学院，1994年。4 [38] C. Szegedy, A. Toshev, 和 D. Erhan。用于目标检测的深度神经网络。在*NIPS*，2013年。2 [39] J. Uijlings, K. van de Sande, T. Gevers, 和 A. Smeulders。用于目标识别的选择性搜索。*IJCV*，2013年。1, 2, 3, 4, 5, 9 [40] R. Vaillant, C. Monrocq, 和 Y. LeCun。图像中目标定位的原始方法。*IEE Proc on Vision, Image, and Signal Processing*，1994年。2 [41] X. Wang, M. Yang, S. Zhu, 和 Y. Lin。用于通用目标检测的Regionlets。在*ICCV*，2013年。3, 5 [42] M. Zeiler, G. Taylor, 和 R. Fergus。用于中高层特征学习的自适应反卷积网络。在*CVPR*，2011年。4 [43] K. Simonyan 和 A. Zisserman。用于大规模图像识别的极深度卷积网络。*arXiv preprint*，arXiv:1409.1556，2014年。6, 7, 14

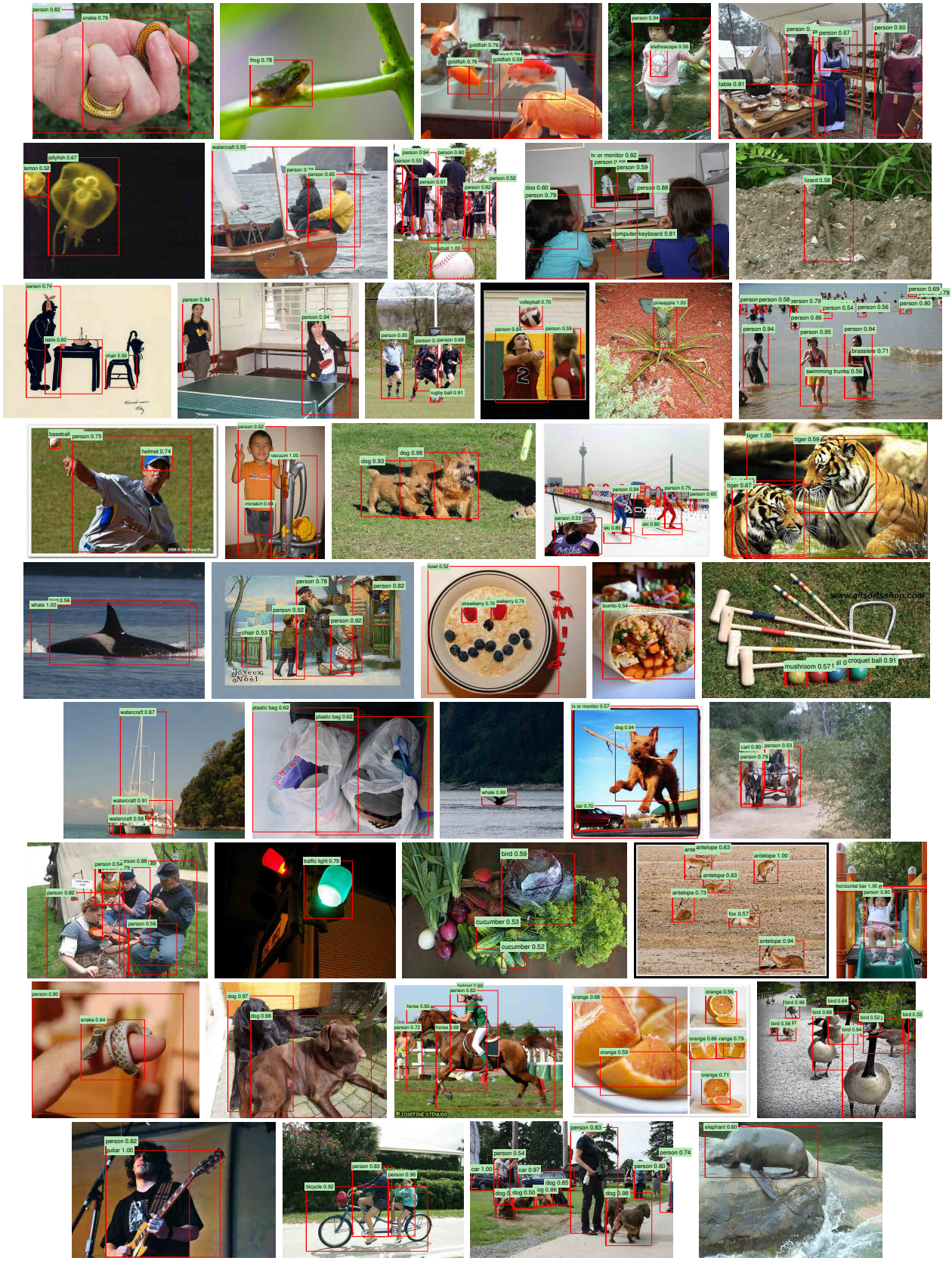


Figure 11: More curated examples. See Figure 10 caption for details. Viewing digitally with zoom is recommended.

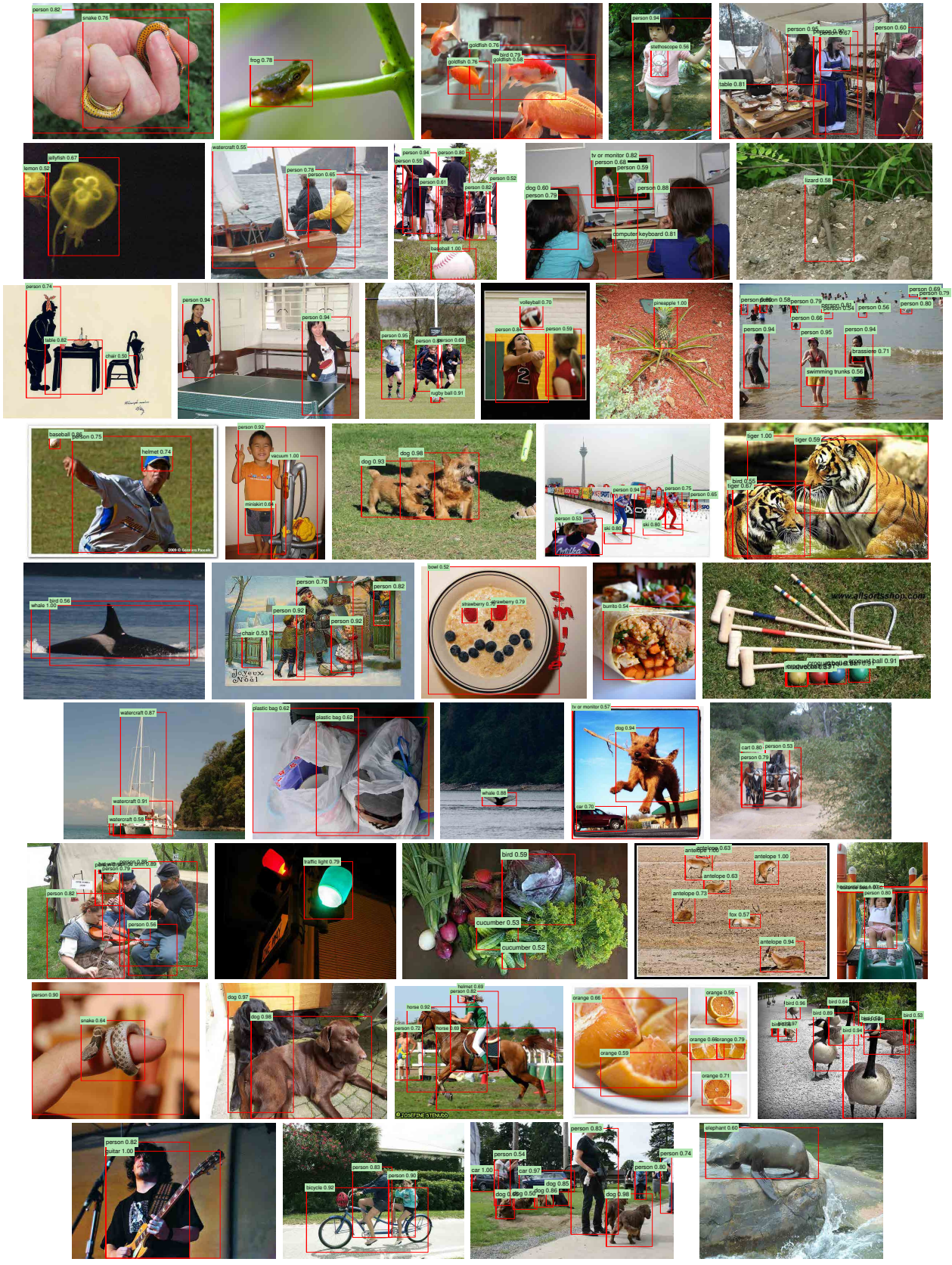


图11：更多精选示例。详情请参见图10的说明。建议在数字设备上使用缩放功能查看。

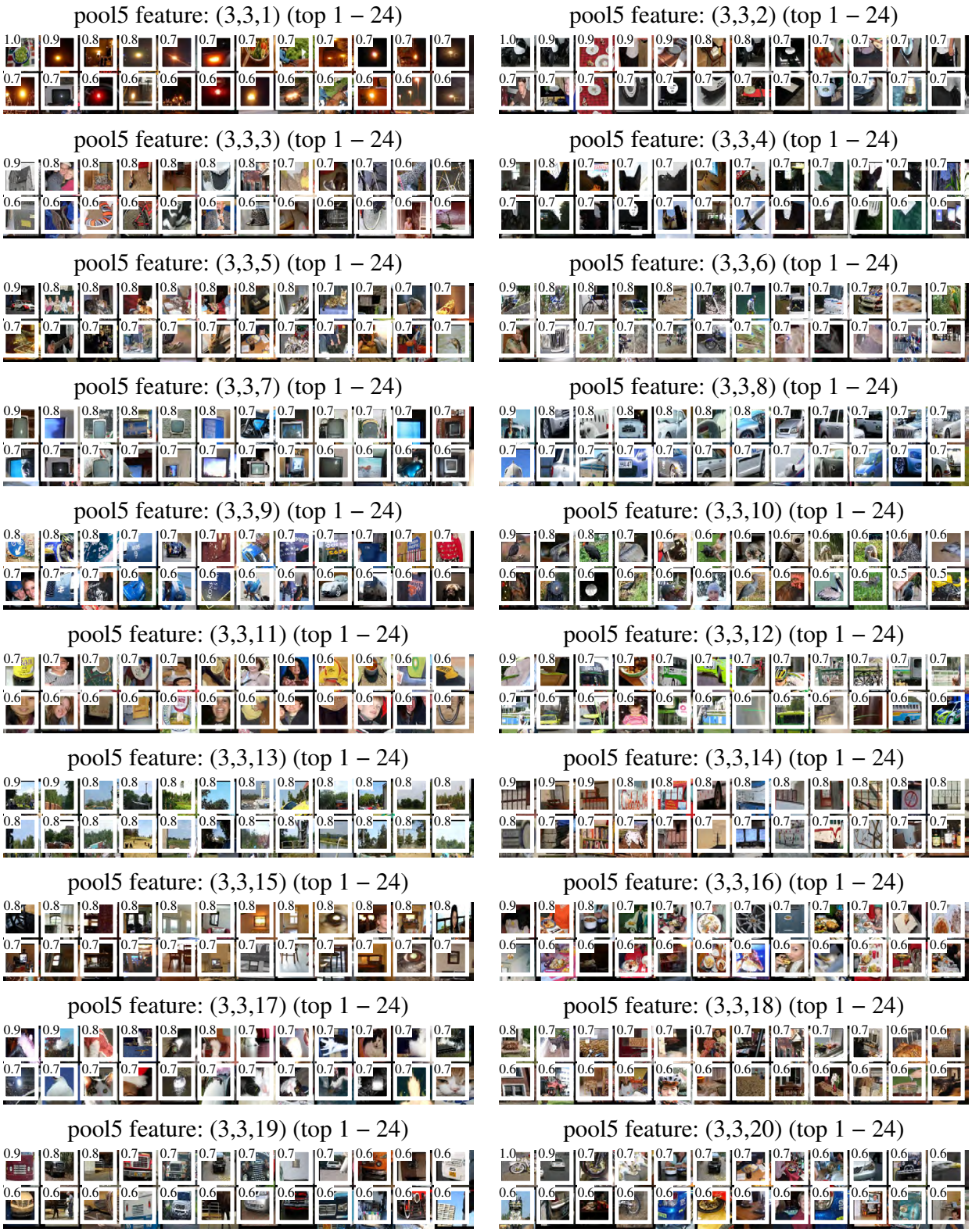


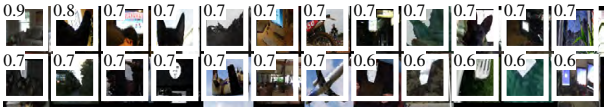
Figure 12: We show the 24 region proposals, out of the approximately 10 million regions in VOC 2007 test, that most strongly activate each of 20 units. Each montage is labeled by the unit's (y, x, channel) position in the $6 \times 6 \times 256$ dimensional pool_5 feature map. Each image region is drawn 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.

pool5 feature: (3,3,1) (top 1 – 24)

pool5 feature: (3,3,2) (top 1 – 24)

pool5 feature: (3,3,3) (top 1 – 24)

pool5 feature: (3,3,4) (top 1 – 24)



pool5 feature: (3,3,5) (top 1 – 24)

pool5 feature: (3,3,6) (top 1 – 24)

pool5 feature: (3,3,7) (top 1 – 24)

pool5 feature: (3,3,8) (top 1 – 24)

pool5 feature: (3,3,9) (top 1 – 24)

pool5 feature: (3,3,10) (top 1 – 24)

pool5 feature: (3,3,11) (top 1 – 24)

pool5 feature: (3,3,12) (top 1 – 24)

pool5 feature: (3,3,13) (top 1 – 24)

pool5 feature: (3,3,14) (top 1 – 24)

pool5 feature: (3,3,15) (top 1 – 24)

pool5 feature: (3,3,16) (top 1 – 24)

pool5 feature: (3,3,17) (top 1 – 24)

pool5 feature: (3,3,18) (top 1 – 24)

pool5 feature: (3,3,19) (top 1 – 24)

pool5 feature: (3,3,20) (top 1 – 24)

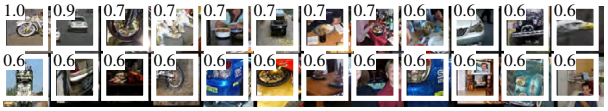


图12：我们展示了在VOC 2007测试集约1000万个区域中，对20个单元各自激活最强的24个候选区域。每个蒙太奇图像均以该单元在6×6×256维池化特征图中的（y, x, 通道）位置进行标注。每个图像区域均以白色叠加显示该单元的接收野。激活值（已通过除以通道内所有单元的最大激活值进行归一化）显示在接收野的左上角。建议使用数字缩放功能以获得最佳观看效果。