

# Faster R-CNN: 基于区域建议网络的实时目标检测

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

**Abstract**—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

**Index Terms**—Object Detection, Region Proposal, Convolutional Neural Network.

## 1 简介

目标检测的最新进展是由区域建议方法(*e.g.*, [4])和基于区域的卷积神经网络(*r - cnn*) [5]的成功推动的。尽管最初在[5]中开发的基于区域的cnn在计算上很昂贵,但由于在不同提案之间共享卷积,它们的成本已大大降低[1], [2]。最新的化身Fast R-CNN [2], 在忽略在区域建议上花费的时间时, 使用非常深的网络[3]实现了接近实时的速率。目前, 在最先进的检测系统中, 提案是测试时的计算瓶颈。

区域建议方法通常依赖于廉价的特征和经济的推理方案。选择性搜索[4], 最流行的方法之一, 基于工程的低级特征贪婪地合并超像素。然而, 与高效的检测网络[2]相比, 选择性搜索要慢一个数量级, 在CPU实现中每图像需要2秒。EdgeBoxes [6]目前提供了提案质量和速度之间的最佳权衡, 每幅图像0.2秒。然而, 区域候选步骤的运行时间仍然与检测网络相同。

有人可能会注意到, 快速的基于区域的cnn利用了gpu, 而研究中使用的区域建议方法是在CPU上实现的, 这使得这种运行时比较不公平。加速建议计算的一个明显方法是为GPU重新实现它。这可能是一个有效的工程解决方案, 但重新实现忽略了下游检测网络, 因此错过了共享计算的重要机会。

本文展示了一种算法变化——用深度卷积神经网络计算建议——产生了一个优雅而有效的解决方案, 考虑到检测网络的计算, 建议计算几乎是免费的。本文提出新

的区域建议网络(RPNs), 与最先进的目标检测网络共享卷积层[1], [2]。通过在测试时共享卷积, 计算建议的边际成本很小(*e.g.*, 每张图像10ms)。

我们的观察是, 基于区域的检测器(如Fast R-CNN)使用的卷积特征图也可以用于生成区域建议。在这些卷积特征之上, 通过添加一些额外的卷积层来构建RPN, 这些层同时回归规则网格上每个位置的区域边界和目标性分数。因此, RPN是一种全卷积网络(FCN) [7], 可以针对生成检测建议的任务进行端到端的训练。

RPNs旨在有效地预测具有广泛尺度和宽高比的区域建议。与使用图像金字塔(图1, a)或滤波器金字塔(图1, b)的流行方法[8], [9], [1], [2]相比, 我们引入了新颖的“锚”框, 在多个尺度和宽高比上作为参考。我们的方案可以被认为是回归参考的金字塔(图1, c), 它避免了枚举多个尺度或长宽比的图像或滤波器。该模型在使用单尺度图像进行训练和测试时表现良好, 因此有利于运行速度。

为了将RPNs与Fast R-CNN [2]目标检测网络统一起来, 本文提出了一种训练方案, 在保持建议框固定的同时, 在对区域建议任务进行微调和对目标检测进行微调之间交替进行。该方案快速收敛, 并产生一个统一的网络, 其中的卷积特征在两个任务之间共享。<sup>1</sup>

在PASCAL VOC检测基准[11]上综合评估了该方法, 其中具有快速r - cnn的RPNs产生的检测精度优于使用快速r - cnn的选择性搜索的强基线。同时, 该方法免除了测试时选择性搜索的几乎所有计算负担——建议的有效运行时间为10毫秒。使用[3]昂贵的very deep模型, 我们的检测方法在GPU上仍然具有5fps的帧率(包括所有步骤), 因此在速度和精度方面都是一个

- S. Ren is with University of Science and Technology of China, Hefei, China. This work was done when S. Ren was an intern at Microsoft Research. Email: sqren@mail.ustc.edu.cn
- K. He and J. Sun are with Visual Computing Group, Microsoft Research. E-mail: {kahe,jiansun}@microsoft.com
- R. Girshick is with Facebook AI Research. The majority of this work was done when R. Girshick was with Microsoft Research. E-mail: rbg@fb.com

1. 自本文会议版[10]发表以来, 我们还发现RPNs可以与Fast R-CNN网络联合训练, 从而减少训练时间。

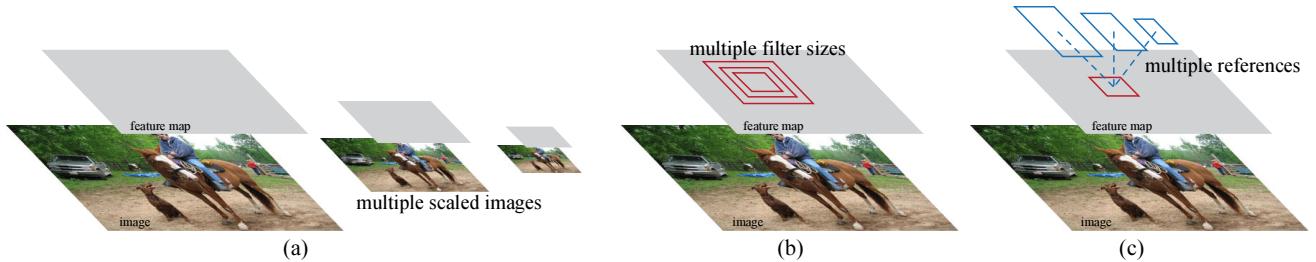


Figure 1: 不同的方案解决多个规模和大小。(a)建立图像金字塔和特征图，并在所有尺度上运行分类器。(b)在特征图上运行具有多个尺度/大小的滤波器金字塔。(c)我们在回归函数中使用参考框的金字塔。

实用的目标检测系统。我们还报告了MS COCO数据集[12]上的结果，并研究了使用COCO数据对PASCAL VOC的改进。代码已在[https://github.com/shaoqingren/faster\\_rcnn](https://github.com/shaoqingren/faster_rcnn) (MATLAB)和<https://github.com/rbgirshick/py-faster-rcnn> (Python)上公开。

该手稿的初步版本之前发表过[10]。从那时起，RPN和Faster R-CNN的框架被采用并推广到其他方法，如3D目标检测[13]，基于部件的检测[14]，实例分割[15]和图像描述[16]。我们快速有效的目标检测系统也已经构建在商业系统中，如at Pinterests [17]，据报道用户参与度有所提高。

在ILSVRC和COCO 2015比赛中，Faster R-CNN和RPN是几个在ImageNet检测、ImageNet定位、COCO检测和COCO分割的轨道上获得第一名的作品[18]的基础。RPNs完全学会从数据中提出区域，因此可以很容易地从更深和更有表现力的特征中获益(例如[18]中采用的101层残差网络)。Faster R-CNN和RPN也被这些比赛中的其他几个领先项目使用<sup>2</sup>。这些结果表明，该方法不仅是一种实用的经济有效的解决方案，而且是提高目标检测精度的有效途径。

## 2 相关工作

目标提案。有大量关于目标建议方法的文献。对对象建议方法的全面调查和比较可以在[19], [20], [21]中找到。广泛使用的目标建议方法包括基于分组超像素的方法(*e.g.*, 选择性搜索[4], CPMC [22], MCG [23])和基于滑动窗口的方法(*e.g.*, windows中的objectness [24], EdgeBoxes [6])。目标建议方法作为独立于检测器的外部模块(*e.g.*、选择性搜索[4]目标检测器、R-CNN [5]和Fast R-CNN [2])。

用于目标检测的深度网络。R-CNN方法[5]对cnn进行端到端的训练，以将建议区域分类为目标类别或背景。R-CNN主要用作分类器，它不预测对象边界(除了通过边界框回归进行细化)。它的准确性取决于区域建议模块的性能(见[20]中的比较)。几篇论文提出了使用深度网络预测对象边界框的方法[25], [9], [26], [27]。在OverFeat方法[9]中，训练一个全连接层来预测假设单个目标的定位任务的框坐标。然后将全连接层转换为卷积层，用于检测多个特定类的对象。多盒方法[26], [27]从一个网络中生成区域建议框，该网络的最后一个全连接层同时预测多个类别无关的盒，泛化了“单盒”的过度风格。这些与类别无关的框

2. <http://image-net.org/challenges/LSVRC/2015/results>

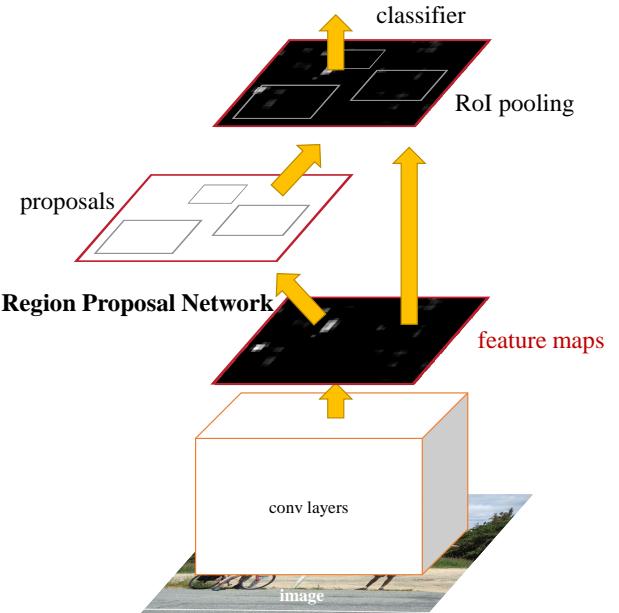


Figure 2: Faster R-CNN是一个用于目标检测的单一统一网络。RPN模块充当这个统一网络的“注意力”。

被用作R-CNN [5]的提案。与我们的全卷积方案相比，多框建议网络应用于单个图像作物或多个大图像作物(*e.g.*, 224 × 224)。MultiBox在建议网络和检测网络之间不共享特征。稍后将在所提出方法的上下文中更深入地讨论OverFeat和MultiBox。与我们的工作同时，DeepMask方法[28]被开发用于学习分割建议。

卷积的共享计算[9], [1], [29], [7], [2]因高效而准确的视觉识别而受到越来越多的关注。OverFeat论文[9]从图像金字塔计算卷积特征，用于分类、定位和检测。基于共享卷积特征图的自适应大小池化(SPP) [1]是为了高效的基于区域的目标检测[1], [30]和语义分割[29]而开发的。Fast R-CNN [2]实现了对共享卷积特征的端到端检测器训练，并显示了令人信服的准确性和速度。

## 3 FASTER R-CNN

我们的目标检测系统称为Faster R-CNN，由两个模块组成。第一个模块是提出区域的深度全卷积网络，第二个模块是使用建议区域的Fast R-CNN检测器[2]。整个系统是一个用于目标检测的单一、统一的网络(图2)。使用最近流行的带有“注意力”[31]机制的神经网络术语，RPN模块告诉Fast R-CNN模块去哪里找。在3.1部

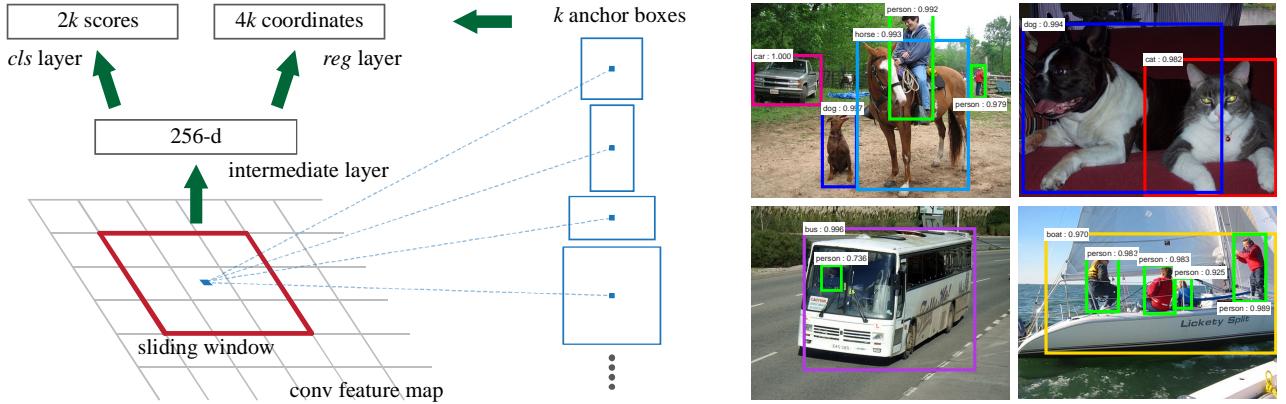


Figure 3: 左:区域建议网络(RPN)。右:PASCAL VOC 2007测试中使用RPN建议的示例检测。该方法在广泛的尺度和纵横比下检测物体。

分，我们介绍了区域规划网络的设计和特点。在3.2节中，我们开发了用于训练具有共享特征的两个模块的算法。

### 3.1 区域建议网络

区域建议网络(RPN)以图像(任意大小)为输入，输出一组矩形建议目标，每个目标都有一个目标性分数。<sup>3</sup> 我们用一个全卷积网络[7]对这个过程进行建模，我们在本节中描述。因为我们的最终目标是与快速R-CNN目标检测网络[2]共享计算，我们假设两个网络共享一组公共的卷积层。在实验中，我们研究了具有5个可共享卷积层的Zeiler和Fergus模型[32] (ZF)和具有13个可共享卷积层的Simonyan和Zisserman模型[3] (VGG-16)。

为了生成区域建议，我们在最后一个共享卷积层输出的卷积特征图上滑动一个小网络。这个小网络将输入卷积特征图的 $n \times n$ 空间窗口作为输入。每个滑动窗口都映射到低维特征(ZF的256-d和VGG的512-d，后面是ReLU [33])。该功能被馈送到两个兄弟的全连接层——盒回归层(reg)和盒分类层(cls)。我们在本文中使用 $n = 3$ ，注意到输入图像上的有效感受野很大(ZF和VGG分别为171和228像素)。这个微型网络在图3(左)中有一个单独的位置。请注意，因为微型网络以滑动窗口方式运行，所以全连接层在所有空间位置上共享。该体系结构自然是由一个 $n \times n$ 卷积层和两个同胞 $1 \times 1$ 卷积层(分别用于reg和cls)实现的。

#### 3.1.1 锚

在每个滑动窗口位置，同时预测多个区域建议框，其中每个位置的最大可能建议框数量表示为 $k$ 。因此，reg层有 $4k$ 输出编码 $k$ 框的坐标，cls层输出 $2k$ 分数，估计每个建议的对象或非对象的概率<sup>4</sup>。 $k$ 建议相对于 $k$ 参考框进行参数化，我们称之为锚。一个锚点位于所述滑动窗口的中心，并与缩放和宽高比相关联(图3，左)。默认情况下，我们使用3个比例和3个纵横比，在每个滑动位置产生 $k = 9$ 锚。对于大小为 $W \times H$ (通常为 $\sim 2400$ )的卷积特征图，总共有 $W H k$ 锚点。

3. “区域”是一个通用术语，在本文中，我们只考虑矩形区域，这是许多方法的共同之处(*e.g.*, [27], [4], [6])。“对象性”衡量的是对一组对象类*vs.*背景的隶属度。

4. 为了简单起见，我们将cls层实现为一个两类softmax层。或者，可以使用逻辑回归来生成 $k$ 分数。

#### 平移不变锚点

该方法的一个重要属性是，无论是从锚点还是相对于锚点计算建议框的函数而言，它都是平移不变的。如果在图像中转换对象，建议框应该转换，相同的函数应该能够在任何位置预测建议框。这种平移不变性是由我们的方法<sup>5</sup>保证的。作为比较，MultiBox方法[27]使用k-means生成800个锚点，它们不是平移不变量。因此，MultiBox不能保证在转换对象时生成相同的建议框。

平移不变特性也减少了模型大小。MultiBox具有 $(4+1) \times 800$ 维的全连接输出层，而我们的方法在 $k = 9$ 锚点的情况下具有 $(4+2) \times 9$ 维的卷积输出层。因此，我们的输出层有 $2.8 \times 10^4$ 参数(对于VGG-16  $512 \times (4+2) \times 9$ )，比MultiBox的输出层少两个数量级，MultiBox的输出层有 $6.1 \times 10^6$ 参数( $1536 \times (4+1) \times 800$ 对于GoogleNet [34]在MultiBox [27])。如果考虑特征投影层，我们的建议层仍然比MultiBox<sup>6</sup>少一个数量级的参数。我们希望我们的方法在小型数据集(如PASCAL VOC)上过拟合的风险更小。

#### 多尺度锚点作为回归参考

锚点的设计提出了一种解决多尺度(和长宽比)的新方案。如图1所示，有两种流行的多尺度预测方法。第一种方法是基于图像/特征金字塔*e.g.*，在DPM [8]和基于cnn的方法[9], [1], [2]。在多个尺度上调整图像的大小，并为每个尺度计算特征图(HOG [8]或深度卷积特征[9], [1], [2])(图1 (a))。这种方法通常很有用，但很耗时。第二种方法是在特征图上使用多尺度(和/或纵横比)的滑动窗口。例如，在DPM [8]中，使用不同的滤波器大小(如 $5 \times 7$ 和 $7 \times 5$ )分别训练不同长宽比的模型。如果使用这种方法来解决多个尺度，它可以被认为是一个“滤波器金字塔”(图1 (b))。第二种方式通常与第一种方式联合使用[8]。

作为比较，基于锚点的方法是建立在锚点金字塔上的，具有更高的成本效率。该方法参考多种尺度和长宽比的锚框对边界框进行分类和回归。它只依赖于单一尺

5. 就像FCNs [7]的情况一样，我们的网络直到网络的总步长都是平移不变的。

6. 考虑到特征投影层，建议层的参数计数为 $3 \times 3 \times 512 \times 512 + 512 \times 6 \times 9 = 2.4 \times 10^6$ ; MultiBox的建议层的参数计数是 $7 \times 7 \times (64 + 96 + 64 + 64) \times 1536 + 1536 \times 5 \times 800 = 27 \times 10^6$ 。

度的图像和特征图，并使用单一尺寸的滤波器(特征图上的滑动窗口)。我们通过实验展示了该方案在处理多个尺度和大时的效果(表8)。

由于这种基于锚点的多尺度设计，我们可以简单地使用在单尺度图像上计算的卷积特征，这也是由Fast R-CNN检测器完成的[2]。多尺度锚点的设计是共享特征而不增加寻址尺度成本的关键组成部分。

### 3.1.2 损失函数

对于训练RPNs，我们为每个锚点分配一个二分类标签(是否为对象)。为两种锚点分配了正标签:(i)具有最高交并比(IoU)的锚点/锚点与真实值框重叠，或(ii)与任何真实值框的IoU重叠高于0.7的锚点。请注意，单个ground-truth框可能会为多个锚点分配正标签。通常情况下，第二个条件足以确定正样本;但我们仍然采用第一种情况，因为在一些罕见的情况下，第二种情况可能没有发现正样本。如果非正锚点的IoU比率低于0.3，则为其分配负标签。既不积极也不消极的锚点对培训目标没有帮助。

通过这些定义，最小化了Fast R-CNN中基于多任务损失的目标函数[2]。图像的损失函数定义如下：

$$\begin{aligned} L(\{p_i\}, \{t_i\}) = & \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ & + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \end{aligned} \quad (1)$$

在这里， $i$ 是小批量中锚点的索引， $p_i$ 是锚点*i*成为对象的预测概率。如果锚点为正，则基本事实标签 $p_i^*=1$ ，如果锚点为负，则为0。 $t_i$ 是表示预测边界框的4个参数化坐标的向量， $t_i^*$ 是与正锚点相关联的ground-truth框的向量。分类损失 $L_{cls}$ 是两个类别的日志损失(对象vs.而不是对象)。对于回归损失，我们使用 $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ ，其中 $R$ 是在[2]中定义的鲁棒损失函数(光滑L<sub>1</sub>)。术语 $p_i^* L_{reg}$ 意味着回归损失仅对正锚点( $p_i^* = 1$ )激活，否则禁用( $p_i^* = 0$ )。 $cls$ 层和 $reg$ 层的输出分别由 $\{p_i\}$ 和 $\{t_i\}$ 组成。

这两项用 $N_{cls}$ 和 $N_{reg}$ 进行归一化，并用平衡参数 $\lambda$ 进行加权。在我们当前的实现中(如在发布的代码中)，Eqn中的 $cls$ 术语。(1)归一化为小批量大小(*i.e.*,  $N_{cls} = 256$ )， $reg$ 术语归一化为锚点位置的数量(*i.e.*,  $N_{reg} \sim 2,400$ )。默认情况下，我们设置 $\lambda = 10$ ，因此 $cls$ 和 $reg$ 词条的权重大致相等。我们通过实验表明，该结果在很大范围内对 $\lambda$ 的值不敏感(表9)。我们还注意到，上面的规范化不是必需的，可以简化。

对于边界框回归，我们采用以下4个坐标的参数化[5]:

$$\begin{aligned} t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\ t_w &= \log(w/w_a), \quad t_h = \log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a), \end{aligned} \quad (2)$$

其中 $x$ 、 $y$ 、 $w$ 和 $h$ 表示框的中心坐标及其宽度和高度。变量 $x$ 、 $x_a$ 和 $x^*$ 分别代表预测值框、锚点框和真实值框( $y$ 、 $w$ 、 $h$ 也是如此)。这可以被认为是从锚框到附近的ground-truth框的边界框回归。

然而，所提出方法通过一种不同于以前基于roi(感兴趣区域)的方法[1], [2]的方式实现了边界框回归。在[1],

[2]中，对从任意大小的兴趣区域汇集的特征执行边界框回归，并且回归权重由所有区域大小共享。在我们的公式中，用于回归的特征在特征图上具有相同的空间大小( $3 \times 3$ )。为了解释不同的大小，学习了一组 $k$ 边界框回归器。每个回归器负责一个尺度和一个纵横比， $k$ 回归器不共享权重。因此，由于锚点的设计，即使特征具有固定的大小/尺度，仍然可以预测各种尺寸的盒子。

### 3.1.3 训练RPNs

RPN可以通过反向传播和随机梯度下降(SGD)进行端到端的训练[35]。我们遵循来自[2]的“以图像为中心”的采样策略来训练这个网络。每个小批量都来自一个包含许多正例和负例锚点的图像。可以对所有锚点的损失函数进行优化，但这将偏向负样本，因为它们占主导地位。相反，我们在图像中随机采样256个锚点以计算mini-batch的损失函数，其中采样的正锚点和负锚点的比例高达1:1。如果图像中少于128个正样本，我们用负样本填充小批量。

我们通过从标准差为0.01的零均值高斯分布中提取权重来随机初始化所有新层。所有其他层(*i.e.*, 共享卷积层)通过预训练ImageNet分类模型[36]来初始化，这是标准实践[5]。我们调整ZF网络的所有层，并conv3\_1和VGG网络以节省内存[2]。我们对PASCAL VOC数据集的60k小批次使用0.001的学习率，对接下来的20k小批次使用0.0001的学习率。我们使用动量0.9和权重衰减0.0005 [37]。我们的实现使用Caffe [38]。

## 3.2 共享RPN和Fast R-CNN的特征

到目前为止，我们已经描述了如何训练一个用于生成区域建议框的网络，而没有考虑将利用这些建议框的基于区域的目标检测CNN。对于检测网络，我们采用Fast R-CNN [2]。接下来，我们描述学习由RPN和具有共享卷积层的Fast R-CNN组成的统一网络的算法(图2)。

独立训练的RPN和Fast R-CNN都将以不同的方式修改它们的卷积层。因此，我们需要开发一种技术，允许在两个网络之间共享卷积层，而不是学习两个单独的网络。我们将讨论三种共享特征的网络训练方法。

(i)交替训练。在这个解决方案中，我们首先训练RPN，并使用这些建议来训练Fast R-CNN。然后使用Fast R-CNN调整的网络初始化RPN，并迭代该过程。这是本文中所有实验中使用的解决方案。

(ii)近似联合训练。在这个解决方案中，RPN和Fast R-CNN网络在训练期间被合并为一个网络，如图2所示。在每次SGD迭代中，前向传递生成区域建议框，在训练快速R-CNN检测器时，将其视为固定的、预先计算的建议框。反向传播照常进行，其中对于共享层，来自RPN损失和Fast R-CNN损失的反向传播信号被结合了。这个解决方案很容易实现。但是这个解决方案忽略了导数w.r.t.提议框的坐标也是网络响应，所以是近似的。在实验中，根据经验发现，该求解器产生了接近的结果，但与交替训练相比，训练时间减少了约25-50%。这个求解器包含在我们发布的Python代码中。

(iii)非近似联合训练。如上所述，RPN预测的边界框也是输入的函数。Fast R-CNN中的RoI池化层[2]接受卷积特征和预测的边界框作为输入，因此理论上有效的反向传播求解器还应该涉及梯度w.r.t.框坐标。在上

Table 1: 使用ZF网络学习到的每个锚点的平均建议大小(数字为 $s = 600$ )。

anchor	$128^2, 2:1$	$128^2, 1:1$	$128^2, 1:2$	$256^2, 2:1$	$256^2, 1:1$	$256^2, 1:2$	$512^2, 2:1$	$512^2, 1:1$	$512^2, 1:2$
proposal	$188 \times 111$	$113 \times 114$	$70 \times 92$	$416 \times 229$	$261 \times 284$	$174 \times 332$	$768 \times 437$	$499 \times 501$	$355 \times 715$

Table 2: 在PASCAL VOC 2007测试集上的检测结果(在VOC 2007 trainval上进行训练)。检测器是带有ZF的Fast R-CNN, 但使用各种建议方法进行训练和测试。

train-time region proposals		test-time region proposals		mAP (%)
method	# boxes	method	# proposals	
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	<b>59.9</b>
<i>ablation experiments follow below</i>				
RPN+ZF, unshared	2000	RPN+ZF, unshared	300	58.7
SS	2000	RPN+ZF	100	55.1
SS	2000	RPN+ZF	300	56.8
SS	2000	RPN+ZF	1000	56.3
SS	2000	RPN+ZF (no NMS)	6000	55.2
SS	2000	RPN+ZF (no cls)	100	44.6
SS	2000	RPN+ZF (no cls)	300	51.4
SS	2000	RPN+ZF (no cls)	1000	55.8
SS	2000	RPN+ZF (no reg)	300	52.1
SS	2000	RPN+ZF (no reg)	1000	51.3
SS	2000	RPN+VGG	300	59.2

述近似联合训练中, 这些梯度被忽略。在一个非近似的联合训练解决方案中, 我们需要一个可微的RoI池化层w.r.t. 盒子坐标。这是一个不平凡的问题, 解决方案可以通过在[15]中开发的“RoI扭曲”层给出, 这超出了本文的范围。

**4步交替训练。**本文采用一种实用的四步训练算法, 通过交替优化来学习共享特征。在第一步中, 我们训练RPN, 如3.1.3节所述。该网络使用imagenet预训练模型进行初始化, 并针对区域建议任务进行端到端的微调。第二步, 我们使用step-1 RPN生成的建议, 通过Fast R-CNN训练一个单独的检测网络。该检测网络也由imagenet预训练模型初始化。在这一点上, 两个网络不共享卷积层。在第三步中, 我们使用检测器网络来初始化RPN训练, 但我们固定共享卷积层, 并只微调RPN特有的层。现在, 两个网络共享卷积层。最后, 在保持共享卷积层固定的情况下, 对Fast R-CNN的独特层进行微调。因此, 两个网络共享相同的卷积层, 并形成统一的网络。类似的交替训练可以运行更多的迭代, 但我们观察到的改进微不足道。

### 3.3 实现细节

在单一尺度的图像上训练和测试区域建议网络和目标检测网络[1], [2]。我们重新缩放图像, 使其较短的边为 $s = 600$ 像素[2]。多尺度特征提取(使用图像金字塔)可以提高精度, 但没有表现出良好的速度-精度权衡[2]。在重新缩放的图像上, 最后一个卷积层上的ZF和VGG网络的总步长是16像素, 因此在调整大小之前, 在典型的PASCAL图像上是 $\sim 10$ 像素( $\sim 500 \times 375$ )。即使如此大的步幅也可以提供很好的结果, 尽管较小的步幅可能会进一步提高准确性。

对于锚点, 我们使用 $128^2$ ,  $256^2$ 和 $512^2$ 像素框区域的3个比例, 3个纵横比为1:1, 1:2和2:1。这些超参数并

不是针对特定数据集精心选择的, 我们在下一节中提供了关于它们影响的消融实验。如前所述, 我们的解决方案不需要图像金字塔或滤波器金字塔来预测多个尺度的区域, 节省了大量的运行时间。图3(右)显示了我们的方法在各种尺度和长宽比下的能力。表1显示了使用ZF网络学习到的每个锚点的平均建议大小。我们注意到, 该算法允许预测大于基础感受野。这样的预测并非不可能——如果只看到物体的中间部分, 人们仍然可以粗略地推断出物体的范围。

跨越图像边界的锚点框需要小心处理。在训练过程中, 我们忽略所有跨界锚点, 因此它们不会对损失做出贡献。对于一个典型的 $1000 \times 600$ 图像, 总共大约有20000 ( $\approx 60 \times 40 \times 9$ )个锚点。在忽略跨界锚点的情况下, 每个图像大约有6000个锚点用于训练。如果在训练中不忽略跨越边界的离群点, 它们会在目标中引入较大的、难以纠正的误差项, 训练不会收敛。然而, 在测试过程中, 我们仍然将全卷积RPN应用于整个图像。这可能会生成跨边界的建议框, 我们将其裁剪到图像边界。

一些RPN提案彼此高度重叠。为了减少冗余, 我们根据候选区域的cls分数对其进行非极大值抑制(NMS)。将NMS的IoU阈值固定在0.7, 这使每张图像大约有2000个建议区域。正如我们将展示的, NMS不会损害最终的检测精度, 但大大减少了建议的数量。NMS之后, 我们使用top-  $N$ 排名的候选区域进行检测。下面, 我们使用2000个RPN建议框训练Fast R-CNN, 但在测试时评估不同数量的建议框。

## 4 实验

### 4.1 PASCAL VOC数据集实验

在PASCAL VOC 2007检测基准[11]上对该方法进行了综合评估。该数据集由20个对象类别的约5k训练图像

Table 3: 在PASCAL VOC 2007测试集上的检测结果检测器是Fast R-CNN和VGG-16。训练数据:“07”:VOC 2007 trainval, “07+12”: VOC 2007 trainval和VOC 2012 trainval的并集。对于RPN, Fast R-CNN的训练时间为2000。<sup>†</sup>:这个数字是在[2]上报道的;使用本文提供的存储库,这个结果更高(68.1)。

method	# proposals	data	mAP (%)
SS	2000	07	66.9 <sup>†</sup>
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	<b>73.2</b>
RPN+VGG, shared	300	COCO+07+12	<b>78.8</b>

Table 4: PASCAL VOC 2012测试集上的检测结果检测器是Fast R-CNN和VGG-16。训练数据:“07”:VOC 2007 trainval, “07++12”: VOC 2007 trainval+test和VOC 2012 trainval的并集。对于RPN, Fast R-CNN的训练时间为2000。<sup>†</sup>: <http://host.robots.ox.ac.uk:8080/anonymous/HZJTQA.html>. <sup>‡</sup>: <http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html>. <sup>§</sup>: <http://host.robots.ox.ac.uk:8080/anonymous/XEDH10.html>.

method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared <sup>†</sup>	300	12	67.0
RPN+VGG, shared <sup>‡</sup>	300	07++12	<b>70.4</b>
RPN+VGG, shared <sup>§</sup>	300	COCO+07++12	<b>75.9</b>

Table 5: 时间(ms)在K40 GPU上,除了SS建议在CPU上进行评估。‘区域级’包括NMS、池化、全连接和softmax层。有关运行时间的分析,请参阅我们发布的代码。

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	<b>10</b>	47	<b>198</b>	<b>5 fps</b>
ZF	RPN + Fast R-CNN	31	<b>3</b>	25	<b>59</b>	<b>17 fps</b>

和5k测试图像组成。我们还提供了一些模型在PASCAL VOC 2012基准上的结果。对于ImageNet预训练网络,我们使用具有5个卷积层和3个全连接层的ZF net [32]的“快速”版本,以及具有13个卷积层和3个全连接层的公共VGG-16模型<sup>7</sup>[3]。我们主要评估检测平均精度(mAP),因为这是对象检测的实际指标(而不是关注对象建议代理指标)。

表2(顶部)显示了使用各种区域建议方法进行训练和测试时的Fast R-CNN结果。这些结果使用了ZF网络。对于选择性搜索(SS) [4], 我们通过“快速”模式生成了大约2000个建议。对于EdgeBoxes (EB) [6], 我们通过调整为0.7 IoU的默认EB设置生成建议。在Fast R-CNN框架下, SS的mAP为58.7%, EB的mAP为58.6%。使用Fast R-CNN的RPN取得了有竞争力的结果, mAP为59.9%, 同时使用多达300个建议<sup>8</sup>。由于共享卷积计算, 使用RPN比使用SS或EB产生更快的检测系统;更少的提案也减少了区域全连接层的成本(表5)。

**RPN**的消融实验。为了研究RPNs作为一种建议方法的行为, 我们进行了几个消融研究。首先, 展示了在RPN和Fast R-CNN检测网络之间共享卷积层的效果。为了做到这一点, 我们在4步训练过程的第二步之后停止。使用单独的网络将结果略微降低

7. [www.robots.ox.ac.uk/~vgg/research/very\\_deep/](http://www.robots.ox.ac.uk/~vgg/research/very_deep/)

8. 对于RPN, 提案数量(e.g., 300)是图像的最大数量。RPN可能在NMS之后产生更少的提案, 因此平均提案数量更少。

到58.7%(RPN+ZF, 未共享, 表2)。这是因为在第三步中, 当使用经过检测器调整的特征来微调RPN时, 建议的质量得到了提高。

接下来, 解析RPN对训练Fast R-CNN检测网络的影响。为此, 使用2000个SS建议和ZF net训练一个快速的R-CNN模型。修正这个检测器, 并通过改变测试时使用的建议区域来评估检测图。在这些烧蚀实验中, RPN与检测器不共享特征。

在测试时用300个RPN提案替换SS, 可以得到56.8%的mAP。mAP中的损失是由于训练/测试建议之间的不一致。这个结果作为后续比较的基准。

有些令人惊讶的是, 在测试时使用排名前100的建议时, RPN仍然可以得到有竞争力的结果(55.1%), 这表明排名前100的RPN建议是准确的。在另一个极端, 使用排名最高的6000个RPN建议(不使用NMS)具有可比较的mAP(55.2%), 表明NMS不会损害检测mAP, 并可能减少误报。

通过在测试时关闭其中一个, 分别研究了RPN的cls和reg输出的作用。当cls层在测试时被移除时(因此没有使用NMS/ranking), 我们从未评分的区域中随机抽样N建议。N = 1000的mAP几乎没有变化(55.8%), 但N = 100的mAP下降到44.6%。这表明, cls分数说明了排名最高的提案的准确性。

另一方面, 当reg层在测试时被删除时(因此提案变成锚框), mAP下降到52.1%。这表明高质量的建议框主要是由于回归的框边界。锚框虽然具有多个尺度和纵横比, 但不足以进行精确检测。

还评估了更强大的网络对单独RPN建议质

Table 6: 使用Fast R-CNN检测器和VGG-16在PASCAL VOC 2007测试集上进行实验。对于RPN, Fast R-CNN的训练时间为2000。RPN \*表示反共享功能版本。

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
SS	2000	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
RPN*	300	07	68.5	74.1	77.2	67.7	53.9	51.0	75.1	79.2	78.9	50.7	78.0	61.1	79.1	81.9	72.2	75.9	37.2	71.4	62.5	77.4	66.4
RPN	300	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
RPN	300	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
RPN	300	COCO+07+12	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9

Table 7: 在PASCAL VOC 2012测试集上使用Fast R-CNN检测器和VGG-16进行测试, 结果显示。对于RPN, Fast R-CNN的训练时间为2000。

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
RPN	300	COCO+07++12	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2

量的影响。我们使用VGG-16来训练RPN, 仍然使用上述SS+ZF的检测器。mAP从56.8%(使用RPN+ZF)提高到59.2%(使用RPN+VGG)。实验结果表明, RPN+VGG算法的候选结果质量优于RPN+ZF算法。因为RPN+ZF的建议与SS有竞争力(当持续用于训练和测试时, 两者都是58.7%), 我们可以预计RPN+VGG比SS更好。以下实验证明了这一假设。

**VGG-16**的性能。表3显示了VGG-16的候选框和检测结果。使用RPN+VGG, 非共享特征的结果为68.5%, 略高于SS基线。如上所示, 这是因为RPN+VGG生成的建议比SS更准确。与预定义的SS不同, RPN是主动训练的, 并从更好的网络中获益。对于特征共享的变体, 结果是69.9%——比强大的SS基线更好, 但几乎是免费的建议。在PASCAL VOC 2007 trainval和2012 trainval的联合集上进一步训练RPN和检测网络。地图是73.2%。图5显示了PASCAL VOC 2007测试集上的一些结果。在PASCAL VOC 2012测试集上(表4), 我们的方法在VOC 2007 trainval+test和VOC 2012 trainval的联合集上训练的mAP为70.4%。具体数字见表6和表7。

在表5中, 我们总结了整个目标检测系统的运行时间。SS需要1-2秒, 这取决于内容(平均约1.5秒), 使用VGG-16的Fast R-CNN在2000个SS建议上需要320毫秒(如果在全连接层上使用SVD, 则需要223毫秒[2])。该系统使用VGG-16编码, 生成和检测时间共为**198ms**。在共享卷积特征的情况下, 仅RPN计算额外的层只需要10ms。由于建议框更少(每张图像300个), 我们的区域计算也更低。通过ZF网络, 我们的系统的帧率为17 fps。

超参数敏感性。在表8中, 我们研究了锚点的设置。默认情况下, 我们使用3个比例和3个宽高比(表8中的69.9%的mAP)。如果在每个位置只使用一个锚点, 地图会下降3-4%。如果使用3个尺度(1个宽高比)或3个宽高比(1个尺度), 地图会更高, 这表明使用多个尺寸的锚点作为回归参考是一个有效的解决方案。在这个数据集上, 仅使用3个具有1个长宽比的尺度(69.8%)与使

Table 8: 使用不同锚点设置的Faster R-CNN在PASCAL VOC 2007测试集上的检测结果。这个网络是VGG-16。训练数据为VOC 2007 trainval。默认设置使用3比例和3宽高比(69.9%)与表3相同。

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128 <sup>2</sup>	1:1	65.8
	256 <sup>2</sup>	1:1	66.7
1 scale, 3 ratios	128 <sup>2</sup>	{2:1, 1:1, 1:2}	68.8
	256 <sup>2</sup>	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	{128 <sup>2</sup> , 256 <sup>2</sup> , 512 <sup>2</sup> }	1:1	69.8
3 scales, 3 ratios	{128 <sup>2</sup> , 256 <sup>2</sup> , 512 <sup>2</sup> }	{2:1, 1:1, 1:2}	69.9

Table 9: 使用方程(1)中 $\lambda$ 的不同值对Faster R-CNN在PASCAL VOC 2007测试集上的检测结果。这个网络是VGG-16。训练数据为VOC 2007 trainval。默认使用 $\lambda = 10$ (69.9%), 与表3中相同。

$\lambda$	0.1	1	10	100
mAP (%)	67.2	68.9	69.9	69.1

用3个具有3个长宽比的尺度一样好, 这表明尺度和长宽比并不是检测精度的解缠维度。但我们仍然在设计中采用这两个维度来保持系统的灵活性。

在表9中, 我们比较了公式(1)中 $\lambda$ 的不同值。默认情况下, 我们使用 $\lambda = 10$ , 这使得方程(1)中的两个项在归一化后的权重大致相等。表9显示, 当 $\lambda$ 在大约两个数量级(1到100)的范围内时, 我们的结果只受到轻微影响(由~1%)。这表明该方法在很大范围内对 $\lambda$ 不敏感。

欠条回收分析。接下来, 我们使用ground-truth框计算不同IoU比率下的建议的召回率。值得注意的是, Recall-to-IoU度量仅与最终检测精度大致相关[19], [20], [21]。使用这个指标来诊断建议方法比评估它更合适。

在图4中, 我们展示了使用300、1000和2000个建议的结果。我们将这些方法与SS和EB方法进行了比较, N方法的置信度最高, 其中N方法的置信度最高。图中显示, 当建议数量从2000下降到300时, RPN方

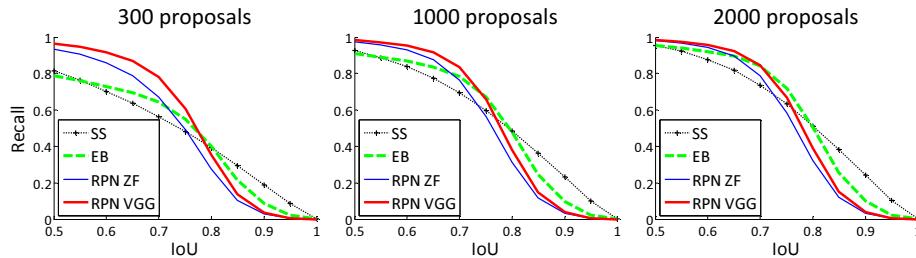


Figure 4: 回想一下PASCAL VOC 2007测试集上的vs. IoU重叠率。

Table 10: 单阶段检测vs.两阶段提议+检测。使用ZF模型和Fast R-CNN在PASCAL VOC 2007测试集上获得检测结果。RPN使用非共享特性。

	proposals	detector	mAP (%)
Two-Stage	RPN + ZF, unshared	Fast R-CNN + ZF, 1 scale	58.7
One-Stage	dense, 3 scales, 3 aspect ratios	Fast R-CNN + ZF, 1 scale	53.8
One-Stage	dense, 3 scales, 3 aspect ratios	Fast R-CNN + ZF, 5 scales	53.9

法表现良好。这解释了为什么RPN在使用仅300个建议时具有良好的最终检测图。正如我们之前分析的，这个性质主要归因于RPN的cls项。当建议数较少时，SS和EB的召回率下降速度比RPN快。

单级检测vs.两阶段提议+检测。OverFeat论文[9]提出了一种在卷积特征图上的滑动窗口上使用回归器和分类器的检测方法。OverFeat是一个单级、特定于类检测管道，而我们的是一个两级叶栅，由与类无关的建议和特定类的检测组成。在OverFeat中，区域特征来自尺度金字塔上一个宽高比的滑动窗口。利用这些特征同时确定物体的位置和类别。在RPN中，特征来自正方形( $3 \times 3$ )滑动窗口，并预测相对于具有不同尺度和长宽比的锚点的建议。虽然这两种方法都使用滑动窗口，但区域建议任务只是Faster R-CNN的第一阶段——下游的Fast R-CNN检测器出席对建议进行改进。在级联的第二阶段，从更忠实地覆盖区域特征的建议框中自适应汇集区域特征[1], [2]。我们相信这些特征会导致更准确的检测。

为了比较单阶段和两阶段系统，用单阶段Fast R-CNN模拟OverFeat系统(从而规避实现细节的其他差异)。在这个系统中，“建议”是3个尺度(128、256、512)和3个长宽比(1:1、1:2、2:1)的密集滑动窗口。Fast R-CNN经过训练，可以预测特定类别的分数，并从这些滑动窗口回归框位置。由于OverFeat系统采用了图像金字塔，我们还使用从5个尺度提取的卷积特征进行评估。我们使用[1], [2]中的5个刻度。

表10比较了两级系统和两种单级系统的变体。使用ZF模型，单阶段系统的mAP为53.9%。这比两阶段制(58.7%)低4.8%。实验证明了级联区域建议和目标检测的有效性。在[2], [39]上也报告了类似的观察结果，其中用滑动窗口替换SS区域建议在两篇论文中都会导致~6%的退化。我们还注意到，单阶段系统较慢，因为它有更多的建议需要处理。

## 4.2 MS COCO实验

我们在Microsoft COCO目标检测数据集[12]上展示了更多结果。该数据集涉及80个对象类别。我们在训

练集上使用80k张图像，在验证集上使用40k张图像，在测试-开发集上使用20k张图像进行实验。我们评估了IoU  $\in [0.5 : 0.05 : 0.95]$  (COCO的标准度量，简记为mAP@[。5, .95])和mAP@0.5 (PASCAL VOC的度量标准)。

我们的系统对这个数据集做了一些小改动。在一个8-GPU实现上训练模型，RPN的有效小批量大小为8(每个GPU 1)，Fast R-CNN为16(每个GPU 2)。RPN步骤和Fast R-CNN步骤都以0.003的学习率训练240k次迭代，然后以0.0003的学习率训练80k次迭代。我们修改了学习率(从0.003开始，而不是0.001)，因为小批量大小发生了变化。对于锚点，我们使用3个纵横比和4个尺度(添加 $64^2$ )，主要是受此数据集上处理小对象的启发。此外，在我们的快速R-CNN步骤中，负样本被定义为在[0, 0.5)的区间内具有最大IoU的ground truth，而不是在[1], [2]中使用的[0.1, 0.5)。我们注意到，在SPPnet系统[1]中，[0.1, 0.5)中的负样本被用于网络微调，但在进行硬负挖掘的SVM步骤中，仍然会访问[0, 0.5)中的负样本。但是Fast R-CNN系统[2]放弃了SVM步骤，因此[0, 0.1)中的负样本永远不会被访问。包括这些[0, 0.1)样本对COCO数据集的mAP@0.5的Fast R-CNN和Faster R-CNN系统都有改进(但对PASCAL VOC的影响可以忽略不计)。

其余的实现细节与PASCAL VOC相同。特别是，我们继续使用300个建议和单尺度( $s = 600$ )测试。在COCO数据集上，每张图像的测试时间仍然约为200ms。

在表11中，我们首先报告了使用本文实现的Fast R-CNN系统[2]的结果。我们的Fast R-CNN基线在测试开发集上有39.3% mAP@0.5，高于[2]报告的值。我们推测这一差距的原因主要是由于负样本的定义以及小批量大小的变化。我们还注意到mAP@[。5.95]是可以比较的。

接下来，我们评估我们的Faster R-CNN系统。使用COCO训练集进行训练，Faster R-CNN有42.1% mAP@0.5和21.5%的mAP@[。5, .95]在COCO测试开发集上。mAP@0.5和mAP@[分 别 高 出 2.8% 和 2.2%]。.5, .95]比相同协议下的Fast R-CNN对

Table 11: 在MS COCO数据集上的目标检测结果(%)。模型是VGG-16。

method	proposals	training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	42.7	21.9

Table 12: Faster R-CNN在PASCAL VOC 2007测试集和2012测试集上使用不同的训练数据的检测mAP(%)。模型是VGG-16。'COCO'表示使用COCO训练集进行训练。参见表6和表7。

training data	2007 test	2012 test
VOC07	69.9	67.0
VOC07+12	73.2	-
VOC07++12	-	70.4
COCO (no VOC)	76.1	73.0
COCO+VOC07+12	<b>78.8</b>	-
COCO+VOC07++12	-	<b>75.9</b>

应版本(表11)。这表明RPN算法在较高IoU阈值下具有较高的定位精度。使用COCO trainval集进行训练，Faster R-CNN有42.7%mAP@0.5和21.9%的mAP@[.5, .95]在COCO测试开发集上。图6显示了MS COCO测试开发集上的一些结果。

在ILSVRC和COCO 2015比赛中的Faster R-CNN 我们已经证明，更快的R-CNN从更好的特征中获益更多，这是因为RPN完全通过神经网络学习提出区域。即使深度大大增加到100层以上[18]，这种观察仍然有效。仅通过将VGG-16替换为101层残差网络(ResNet-101) [18]，Faster R-CNN系统在COCO val集上的mAP从41.5%/21.2%(VGG-16)提高到48.4%/27.2%(ResNet-101)。通过与Faster R-CNN正交的其他改进，He *et al.* [18]在COCO test-dev集上获得了55.7%/34.9%的单模型结果和59.0%/37.4%的集成结果，在COCO 2015目标检测竞赛中获得了第一名。同样的系统[18]也在ILSVRC 2015目标检测竞赛中获得了第一名，以绝对8.5%的优势超过了第二名。RPN也是ILSVRC 2015本地化和COCO 2015细分比赛的第一名获奖作品的组成部分，详情分别在[18]和[15]。

### 4.3 从COCO女士到PASCAL VOC

大规模数据对于改进深度神经网络至关重要。研究了MS COCO数据集如何帮助提高PASCAL VOC的检测性能。

作为一个简单的基线，直接在PASCAL VOC数据集上评估COCO检测模型，而不需要在任何PASCAL VOC数据上进行微调。这种评估是可能的，因为COCO上的类别是PASCAL VOC上的类别的超集。在本实验中忽略COCO上独占的类别，只对加背景的20个类别进行softmax层。在PASCAL VOC 2007测试集上，此设置下的mAP为76.1%(表12)。即使没有利用PASCAL VOC数据，这个结果也比在VOC07+12(73.2%)上训练的结果要好很多。

然后在VOC数据集上对COCO检测模型进行微调。在这个实验中，COCO模型取代了imagenet预训练模型(用于初始化网络权重)，并对Faster R-CNN系统进行了微调，如3.2节所述。这样做可以在PASCAL VOC 2007测试集上获得78.8%的mAP。来自COCO集合的额外数据使mAP增加了5.6%。表6显示了在COCO+VOC上训练的模型在PASCAL VOC 2007上对每个单独类别都有最好的AP。在PASCAL VOC 2012测试集上也观察到了类似的改进(表12和表7)。获得这些强大结果的测试时间速度仍然约为每张图像200毫秒。

## 5 结论

本文提出RPNs，用于高效和准确的区域建议生成。通过与下游检测网络共享卷积特征，区域建议步骤几乎是免费的。该方法使一个统一的、基于深度学习的目标检测系统能够以接近实时的帧率运行。学到的RPN还提高了区域建议的质量，从而提高了整体目标检测的精度。

## REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision (ECCV)*, 2014.
- [2] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.
- [4] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision (IJCV)*, 2013.
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [6] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision (ECCV)*, 2014.
- [7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.
- [9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2014.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Neural Information Processing Systems (NIPS)*, 2015.
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," 2007.

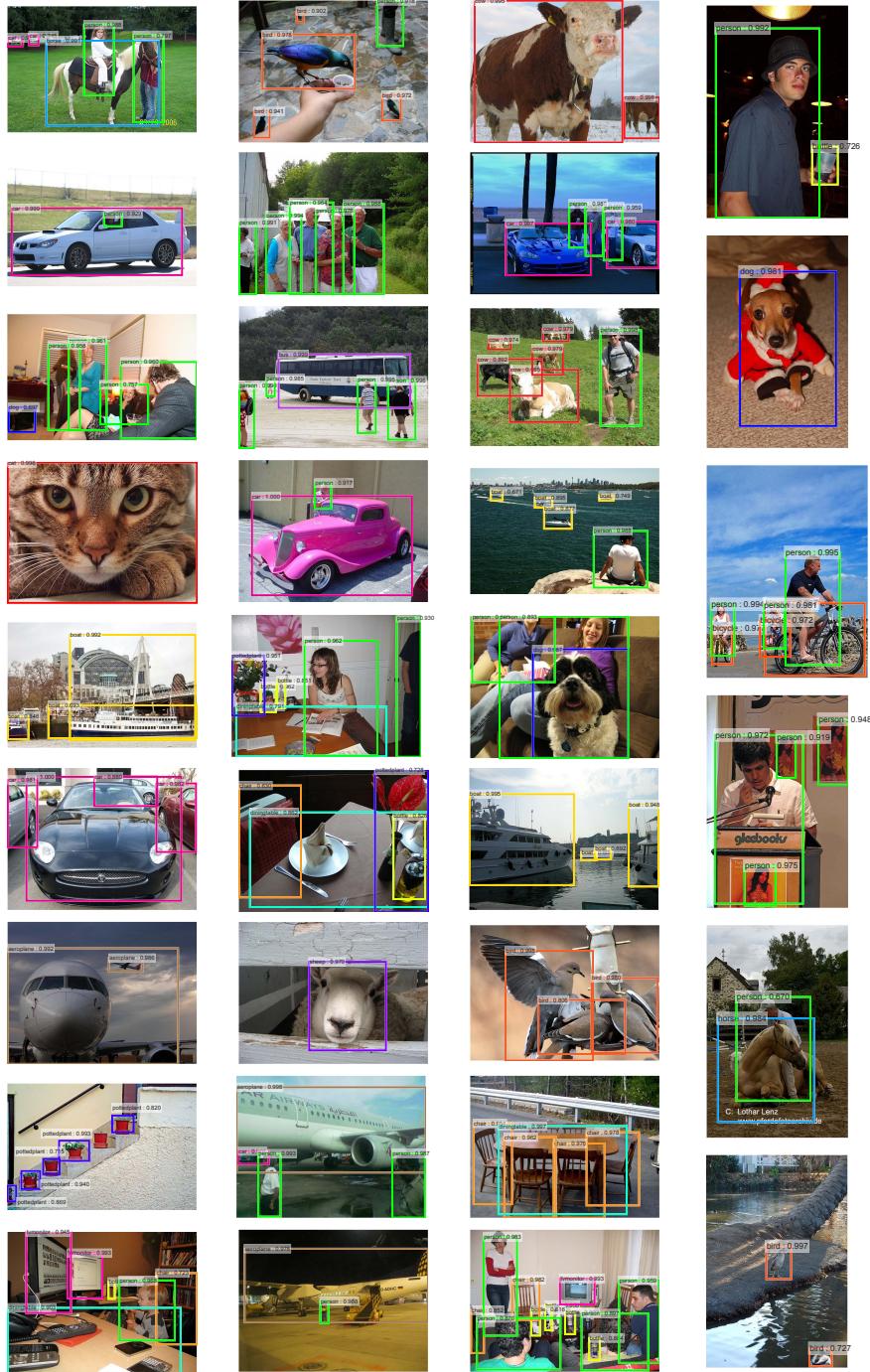


Figure 5: 使用Faster R-CNN系统在PASCAL VOC 2007测试集上的目标检测结果的选择示例。该模型是VGG-16，训练数据为07+12 trainval(在2007测试集上的mAP为73.2%)。该方法可以检测具有广泛尺度和宽高比的物体。每个输出框都与一个类别标签和[0, 1]中的softmax分数相关联。分数阈值0.6用于显示这些图像。获得这些结果的运行时间为每张图像**198ms**，包括所有步骤。

- [12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision (ECCV)*, 2014.
- [13] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," *arXiv:1511.02300*, 2015.
- [14] J. Zhu, X. Chen, and A. L. Yuille, "DeePM: A deep part-based model for object detection and semantic part localization," *arXiv:1511.07131*, 2015.
- [15] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmenta- tion via multi-task network cascades," *arXiv:1512.04412*, 2015.
- [16] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," *arXiv:1511.07571*, 2015.
- [17] D. Kislyuk, Y. Liu, D. Liu, E. Tzeng, and Y. Jing, "Human curation and convnets: Powering item-to-item recommendations on pinterest," *arXiv:1511.04003*, 2015.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.
- [19] J. Hosang, R. Benenson, and B. Schiele, "How good are de-

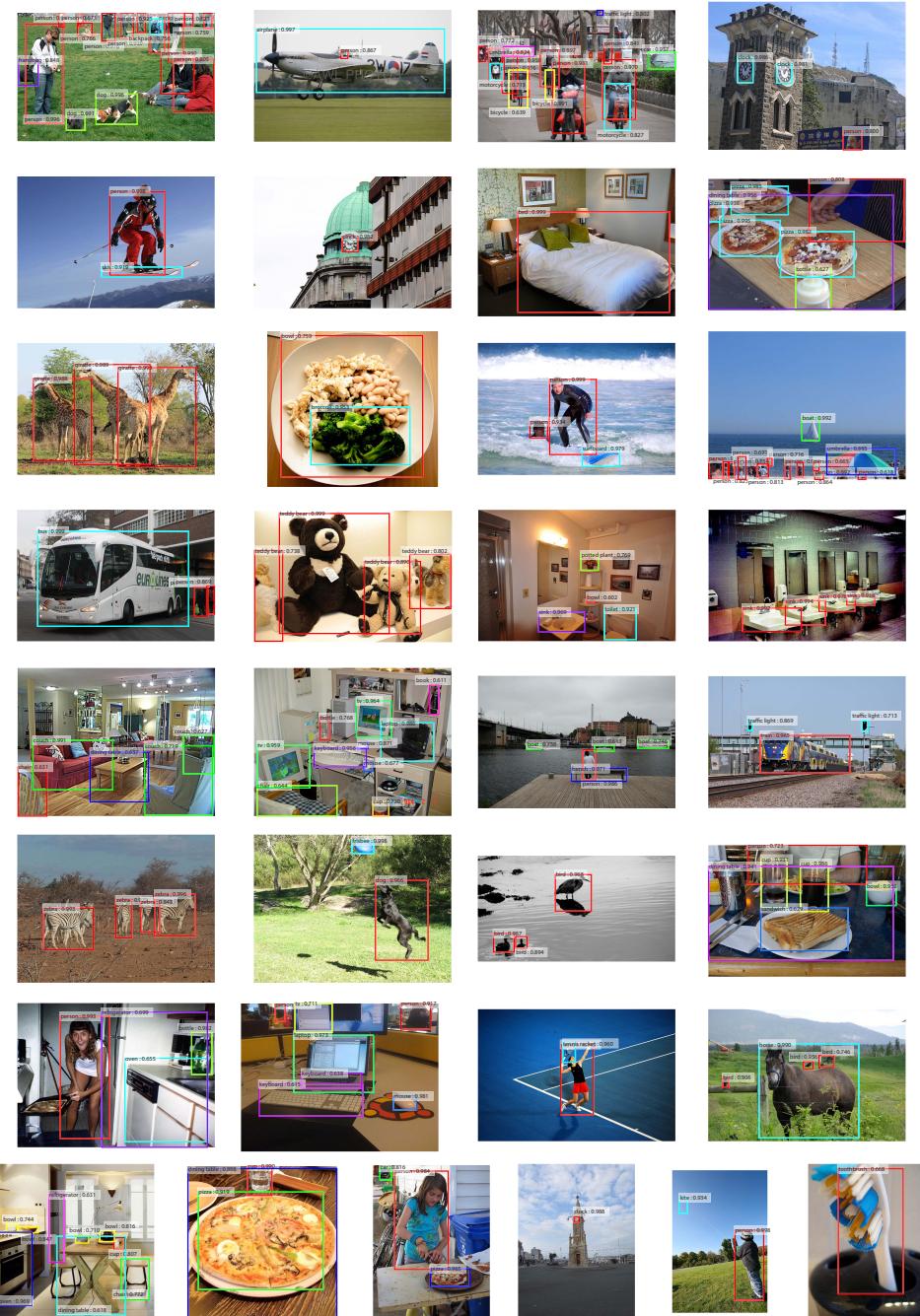


Figure 6: 使用Faster R-CNN系统在MS COCO测试dev集上的目标检测结果示例。模型是VGG-16，训练数据是COCO trainval(在test-dev集上42.7% mAP@0.5)。每个输出框都与一个类别标签和[0, 1]中的softmax分数相关联。分数阈值0.6用于显示这些图像。对于每个图像，一种颜色代表该图像中的一个对象类别。

- tection proposals, really?" in *British Machine Vision Conference (BMVC)*, 2014.
- [20] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.
  - [21] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra, "Object-Proposal Evaluation Protocol is 'Gameable,'" *arXiv: 1505.05836*, 2015.
  - [22] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.
  - [23] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
  - [24] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.
  - [25] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Neural Information Processing Systems (NIPS)*, 2013.
  - [26] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
  - [27] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," *arXiv:1412.1441 (v1)*, 2015.

- [28] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," in *Neural Information Processing Systems (NIPS)*, 2015.
- [29] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [30] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *arXiv:1504.06066*, 2015.
- [31] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Neural Information Processing Systems (NIPS)*, 2015.
- [32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2014.
- [33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010.
- [34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [35] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, 1989.
- [36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *International Journal of Computer Vision (IJCV)*, 2015.
- [37] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012.
- [38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014.
- [39] K. Lenc and A. Vedaldi, "R-CNN minus R," in *British Machine Vision Conference (BMVC)*, 2015.