Faster R-CNN：利用区域提案网络实现实时目标检测

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

摘要——现有技术的对象检测网络依赖于区域提案算法来假设对象位置。像 SPPnet [1]和 Fast R-CNN [2]这样的进步减少了这些检测网络的运行时间，将区域提案计算推向了瓶颈。在这项工作中，我们引入了一个区域提案网络（RPN），它与检测网络共享全图像卷积特征，从而实现了几乎无成本的区域提案。 RPN 是一个完全卷积网络，可同时预测每个位置的对象边界和对象分数。 RPN 经过端到端的训练，可以生成高质量的区域提案，并由 Fast R-CNN 用于检测。我们通过共享其卷积特征进一步将 RPN 和 Fast R-CNN 合并到一个网络中 - 使用最近流行的具有”注意力机制”(Attention)的神经网络术语，RPN 组件告诉全局网络在哪里查看。对于非常深的 VGG-16 模型[3]，我们的检测系统在 GPU 上具有 5fps 的帧速率（包括所有步骤），同时在 PASCAL VOC 2007,2012 上实现了最先进的物体检测精度，并且在 MS COCO 数据集，每个图像只有 300 个提案。在 ILSVRC 和 COCO 2015 比赛中，Faster R-CNN 和 RPN 是多个赛道中获得第一名的参赛作品的基础。代码已公开发布。

索引项——物体检测，区域提案，卷积神经网络。

1 介绍

物体检测的最新进展是由区域提议方法（例如，[4]）和基于区域的卷积神经网络（R-CNN）[5]的成功推动的。尽管基于区域的 CNNs 在[5]中最初开发的时候代价极高，但由于各提案之间存在相互矛盾，它们的成本已经大大降低[1]，[2]。最新版本的 Fast R-CNN [2]使用非常深的网络[3]实现接近实时的速率，忽略了在区域提案上花费的时间。现在，提案是最先进的检测系统中测试时的计算瓶颈。

区域提案方法通常依赖于简易的的特征和经济的推理方案。选择性搜索[4]是最受欢迎的方法之一，它基于工程化的低级特征贪婪地合并超像素。然而，与有效的检测网络[2]相比，选择性搜索速度慢了一个数量级，在 CPU 实现中每个图像花费 2 秒。 EdgeBoxes [6]目前提供了提案质量和速度之间的最佳权衡，每张图像 0.2 秒。然而，区域提案步骤仍然消耗与检测网络一样多的运行时间。

S. Ren 来自中国科学技术大学，中国合肥。 当 S. Ren 是 Microsoft Research 的实习生时，这项工作就完成了。
电子邮件：sqren@mail.ustc.edu.cn
K. He 和 J. Sun 与微软研究院的 Visual Computing Group 合作。
电子邮件：fkahe，jiansung @ microsoft.com
R. Girshick 与 Facebook AI Research 合作。 大部分工作是在 R. Girshick 与 Microsoft Research 合作时完成的.
电子邮件: rbg@fb.com

人们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提案方法是在 CPU 上实现的，这使得这种运行在比较时不公平。加速提案计算的一种有效方法是为 GPU 重新实现它。这可能是一种有效的工程解决方案，但重新实施会忽略下游检测网络，因此错过了利用共享计算的重要机会。

在本文中，我们展示了具有深度卷积神经网络的算法变更计算方案 – 推导出优雅有效的解决方案，其中提案计算在检测网络的计算下几乎是无成本的。为此，我们引入了新颖的区域提案网络（RPN），它们与最先进的物体检测网络共享卷积层[1]，[2]。通过在测试时共享卷积，计算提案的边际成本很小（例如，每个图像 10ms）。

我们的观察结果是，基于区域的探测器（如 Fast R-CNN）使用的卷积特征图也可用于生成区域提案。在这些卷积特征之上，我们通过添加一些额外的卷积层来构建 RPN，这些层同时在常规网格上的每个位置处回归区域边界和对象得分。因此，RPN 是一种完全融合的网络（FCN）[7]，可以专门针对生成检测提议的任务进行端到端的训练。

RPN 旨在有效地预测具有各种尺度和纵横比的区域提案。与流行方法[8]，[9]，[1]，[2]相反，

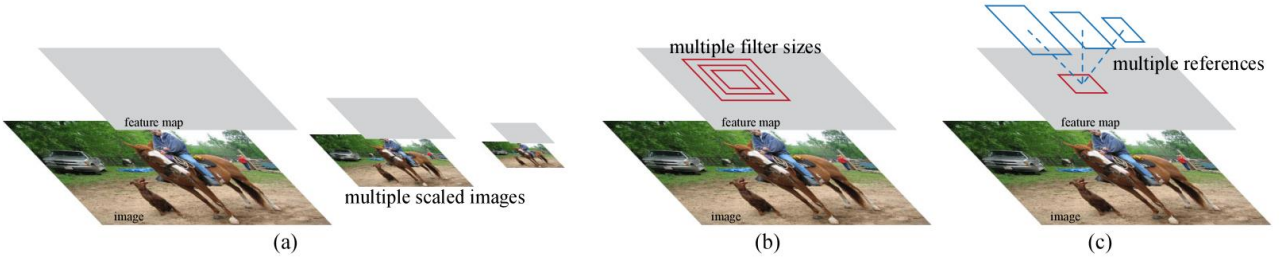

图 1：用于处理多种尺度和尺寸的不同方案。 （a）构建图像和特征图的金字塔，并且分类器在所有尺度上运行。 （b）在特征图上运行具有多个尺度/尺寸的滤波器的金字塔。 （c）我们在回归函数中使用参考框的金字塔。

这些流形方法使用图像的金字塔（图 1，a）或滤波器的金字塔（图 1，b），而我们引入了新的“锚”框，作为多尺度和纵横比的参考。我们的方案可以被认为是回归参考的金字塔（图 1，c），它避免了枚举多尺度或纵横比的图像或滤波器。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 RPN 与 Fast R-CNN [2]对象检测网络统一起来，我们提出了一种训练方案，该方案在区域提案任务的微调和对象检测的微调之间交替进行，同时保持提案的固定。该方案快速收敛并产生具有在两个任务之间共享的轮廓特征的统一网络（见页底 1）。

我们在 PASCAL VOC 检测基准[11]上全面评估了我们的方法，其中具有 Fast R-CNN 的 RPN 产生的检测精度优于具有 Fast R-CNN 的选择性搜索的强基线。同时，我们的方法在测试时放弃了几乎所有选择性搜索的计算负担 - 提案的有效运行时间仅为 10 毫秒。使用[3]的昂贵的非常深的模型，我们的检测方法在 GPU 上仍然具有 5fps 的帧速率（包括所有步骤），因此在速度和准确度方面是实用的物体检测系统。我们还报告了 MS COCO 数据集[12]的结果，并使用 COCO 数据研究了 PASCAL VOC 的改进。代码已在 https://github.com/shaoqingren/faster_rcnn(在 MATLAB 中）和 https://github.com/rbgirshick/py-faster-rcnn （在 Python 中）公开发布。

该手稿的初步版本之前已被公布[10]。从那时起，RPN 和 Faster R-CNN 框架已被采用并普及到其他方法，如 3D 物体检测[13]，基于部件的检测[14]，实例分割[15]和图像字幕[16]。我们还内置了快速有效的物体检测系统

1.自本文件会议版本发布以来[10]，我们还发现 RPN 可以与 Fast RCNN 网络联合训练，从而缩短训练时间。

在诸如 Pinterests [17]等商业系统中，报告了用户参与度改进。

在 ILSVRC 和 COCO 2015 竞赛中，Faster R-CNN 和 RPN 是 ImageNet 检测，ImageNet 定位，COCO 检测和 COCO 分段的几个第一名[18]的基础。 RPN 完全学会从数据中提出区域，因此可以轻松地从更深层次和更具表现力的特征中受益（例如[18]中采用的 101 层残差网络）。Faster R-CNN 和 RPN 也被这些竞赛中的其他几个主要参赛者使用(见页底 2)。 这些结果表明，我们的方法不仅是实用的经济高效的解决方案，而且是提高物体检测精度的有效方法。

2 相关工作

对象提案。关于对象提议方法的文献很多。对象提议方法的综合调查和比较可以在[19]，[20]，[21]中找到。广泛使用的对象提议方法包括基于分组超像素的方法（例如，选择性搜索[4]，CPMC [22]，MCG [23]）和基于滑动窗口的方法（例如，窗口中的对象[24]，EdgeBoxes [6]）。采用对象提议方法作为独立于检测器的外部模块（例如，选择性搜索[4]对象检测器，R-CNN [5]和 Fast R-CNN [2]）。

用于对象检测的深度网络。 R-CNN 方法[5]端到端地训练 CNN 以将提议区域分类为对象类别或背景。 R-CNN 主要作为分类器使用，并且它不预测对象边界（除了通过边界框回归进行精炼）。其准确性取决于区域提案模块的性能（参见[20]中的比较）。有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，训练完全连接层以预测，比如单个对象，这一局部任务的边框坐标。然后转动完全连接层

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

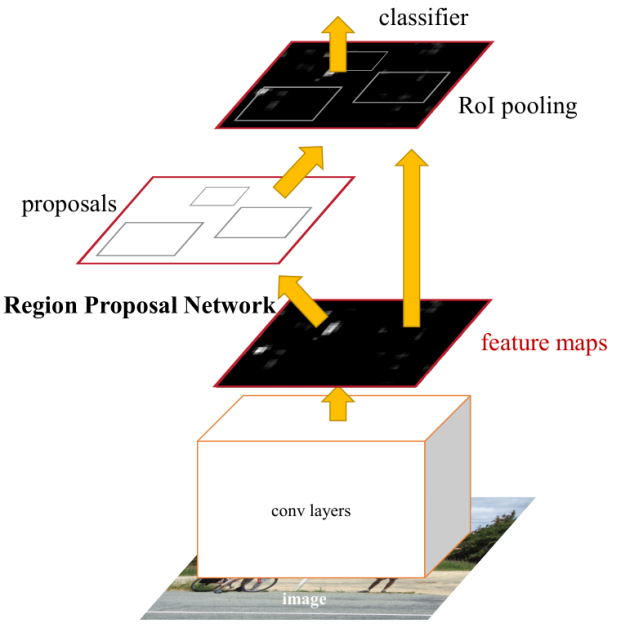

图 2：Faster R-CNN 是用于对象检测的单个统一网络。 RPN 模块是该统一网络的“关注点”。

进入卷积层以检测多个特定类的对象。 MultiBox 方法[26]，[27]从网络生成区域提案，其最后一个完全连接层同时预测多个不可知类框，推广 OverFeat 的“单盒”方式。这些与类无关的框用作 R-CNN 的提议[5]。与我们的完全卷积方案相比，MultiBox 提议网络应用于单个图像裁剪或多个大图像裁剪（例如，224×224）。 MultiBox 不会在提案和检测网络之间共享功能。我们将在后面的方法中更深入地讨论 OverFeat 和 MultiBox。在我们的工作的同时，DeepMask 方法[28]被开发用于学习细分提议。

卷积[9]，[1]，[29]，[7]，[2]的共享计算已经引起了越来越多的关注，以获得有效且准确的视觉识别。 OverFeat 论文[9]计算了图像金字塔中的卷积特征，用于分类，定位和检测。共享卷积特征图上的自适应大小池（SPP）[1]被开发用于有效的基于区域的对象检测[1]，[30]和语义分割[29]。Fast R-CNN [2]实现了对共享卷积特征的端到端检测器训练，并显示出引人注目的准确性和速度。

3 FASTER R-CNN

我们的物体检测系统称为 Faster R-CNN，由两个模块组成。 第一个模块是生成区域提案的深度完全卷积网络，第二个模块是基于提案区域的 Fast R-CNN 检测器[2]。 整个系统是一个用于对象检测的单一统一网络（图 2）。 使用最近流行的具有“注意力”(Attention)[31]机制的神经网络术语，RPN 模块告诉 Fast R-CNN 模块在哪里查看。 在 3.1 节中，我们介绍了区域提案网络的设计和属性。 在 3.2 节中，我们开发了用于训练具有共享功能的两个模块的算法。

3.1 区域提案网络

区域提案网络（RPN）将图像（任意大小）作为输入并输出一组矩形对象提案，每个提案都具有对象性得分。正如我们在本章描述的，我们使用完全卷积网络[7]对此过程进行建模，因为我们的最终目标是与 Fast R-CNN 对象检测网络共享计算[2]，我们假设两个网络共享一组共同的卷积层。在我们的实验中，我们研究了 Zeiler 和 Fergus 模型[32](ZF)，有 5 个可共享的卷积层和 Simonyan 和 Zisserman 模型[3]（VGG-16），它有 13 个可共享的卷积层。

为了生成区域提案，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。该小网络将输入卷积特征映射的 n×n 空间窗口作为输入。每个滑动窗口都映射到一个较低维度的特征（ZF 为 256-d，VGG 为 512-d，后面是 ReLU [33]）。此功能被送入两个兄弟完全连接层 - 一个盒子回归层（reg）和一个盒子分类层（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）。这个迷你网络在图 3（左）中示出了单个位置。请注意，由于迷你网络以滑动窗口方式运行，因此全连接层在所有空间位置共享。该架构自然地实现了 n ×n 卷积层，随后是两个兄弟 1×1 卷积层（分别用于 reg 和 cls）。

3.1.1 锚点（Anchors）

在每个滑动窗口位置，我们同时预测多个区域提案，其中每个位置的最大可能提案的数量表示为 k。 因此，reg 层具有 4k 输出，其编码 k 个盒子的坐标，并且 cls 层输出 2k 得分，其估计每个提案的对象或非对象的概率(见页底 4)。 相对于我们称之为 k 个参考框，k 个提案是参数化的

3. “区域”是一个通用术语，在本文中我们只考虑矩形区域，这是许多方法常见的（例如，[27]，[4]，[6]）。 “对象”测量一组对象类与背景的集合的成员。

4. 为简单起见，我们将 cls 层实现为两级 softmax 层。 或者，可以使用逻辑回归来产生 k 分数。

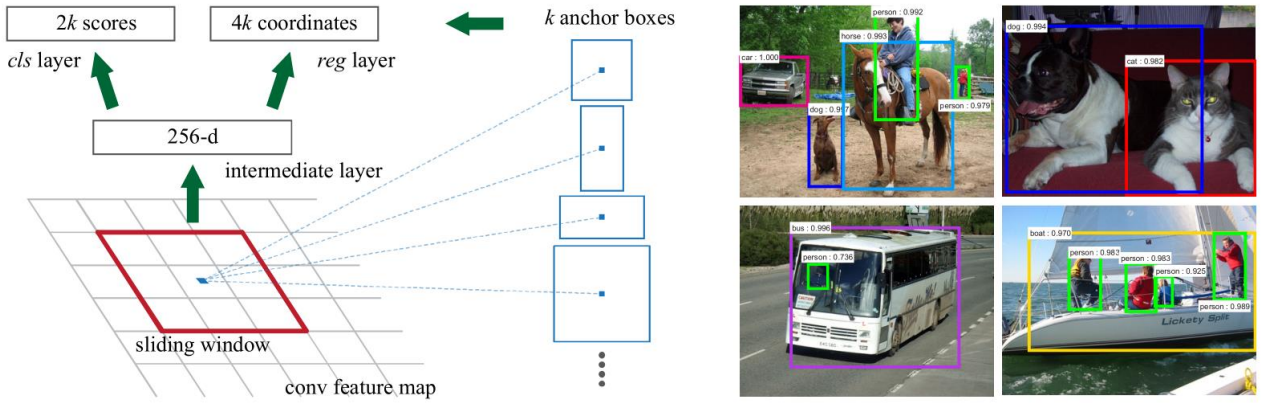

图 3：**左图：**区域提案网络（RPN）。 **右图：**在 PASCAL VOC 2007 测试中使用 RPN 提案的示例检测。 我们的方法可以检测各种比例和纵横比的物体。

锚点。 锚点位于所讨论的滑动窗口的中心，并且与比例和纵横比相关联（图 3，左）。 默认情况下，我们使用 3 个刻度和 3 个纵横比，在每个滑动位置产生 k = 9 个锚点。 对于大小为 W×H（通常为 2,400）的卷积特征图，总共有 W·H·k 个锚点。

移动不变性的锚点

我们方法的一个重要特性是它在平移和计算提案相对于锚点的函数方面都是平移不变的。如果需要移动图像中的对象，则提案应该移动，并且相同的功能应该能够预测任何一个方案中的提议。这种移动不变的属性由我们的方法（见页底 5）保证。作为比较，MultiBox 方法[27]使用 k-means 生成 800 个锚点，这些锚点不是平移不变的。因此，如果转换对象，MultiBox 不保证生成相同的提议。

平移不变属性还会减小模型大小。 MultiBox 具有（4 + 1）×800 维全连接输出层，而在 k = 9 个锚点的情况下，我们的方法具有（4 + 2）×9 维卷积输出层。因此，我们的输出层有 2.8×10^4个参数（VGG-16 为 512×(4+2)×9），比具有 6.1×10^6个参数的 MultiBox 的输出层少 2 个数量级，(MultiBox [27]中的 GoogleNet [34]为 1536×(4+1)×800)。如果考虑特征投影图层，我们的提案图层的参数仍然比 MultiBox（见页底 6）少一个数量级。我们希望我们的方法在 PASCAL VOC 等小数据集上的过度拟合风险较小。

5.与 FCN [7]的情况一样，依据网络的总体步伐，我们的网络是平移不变的。

6.考虑到特征投影层，我们的提案图层的参数计数为 3×3×512×512 + 512×6×9 = 2.4 $\times10^6$; MultiBox 的提案图层的参数计数为 7×7×(64 + 96 + 64 + 64) ×1536 + 1536×5×800 = 27 × 10^6。

多尺度锚点作为回归参考

我们的锚设计提出了一种解决多尺度（和纵横比）的新方案。如图 1 所示，有两种流行的多尺度预测方法。第一种方式基于图像/特征金字塔，例如，在 DPM [8]和基于 CNN 的方法[9]，[1]，[2]中。图像在多个尺度上调整大小，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））。这种方式通常很有用但很耗时。第二种方法是在特征图上使用多个尺度（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（例如 5×7 和 7×5）分别训练不同宽高比的模型。如果这种方式用于解决多个尺度，可以将其视为“过滤金字塔”（图 1（b））。第二种方式通常与第一种方式共同采用[8]。

作为比较，我们基于锚的方法建立在锚的金字塔上，这更具成本效益。我们的方法参考多尺度和纵横比的锚框对边界框进行分类和回归。它仅依赖于单个尺度的图像和特征图，并使用单个尺寸的滤镜（在特征图上滑动窗口）。我们通过实验证明了该方案对于解决多种尺度和尺寸的影响（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，如同 Fast R-CNN 检测器[2]所做的那样。多尺度锚的设计是共享特征的关键组件，无需额外成本来解决尺度问题。

3.1.2 损失函数

为了训练 RPN，我们为每个锚分配一个二维类标签（对象是或否）。 我们为两种锚点签署了一个正面标签：（i）与完全真实框具有的重叠交并比（IoU）最高的锚点（们），或（ii）与任何真实的盒子具有 IoU 重叠高于 0.7 的锚点们

请注意，单个完全真实框可以为多个锚点分配正面标签。 通常第二个条件足以确定正样本; 但我们仍采用第一个条件，因为在极少数情况下，第二个条件可能找不到正样本。 如果所有完全真实框的 IoU 比率低于 0.3，我们会为非正例锚分配负标签。 既不是正面也不是负面的锚点对训练目标没有贡献。

通过这些定义，我们可以在 Fast R-CNN 中将多任务损失的目标函数最小化[2]。 我们对图像的损失函数定义为：

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

这里，i 是 mini-bach 中锚点的索引，pi 是锚点 i 作为对象的预测概率。如果锚为正，则完全真实标签p_i^*为 1，如果锚为负，则为 0。 ti 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完全真实框的向量。分类损失 Lcls 是两类（对象与非对象）的对数损失。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[2]中定义的鲁棒损失函数（光滑 L1）。术语$p_i^* L_{reg}$表示仅对正锚（p_i^* = 1）激活回归损失，否则禁用（p_i^* = 0）。 cls 和 reg 层的输出分别由{pi}和{ti}组成。

这两个项由 Ncls 和 Nreg 归一化，并由平衡参数λ加权。在我们当前的实现中（如在已发布的代码中），方程（1）中的 cls 项由小批量大小（即，Ncls = 256）标准化，并且 reg 项由锚位置的数量标准化（即，Nreg ~ 2,400）。默认情况下，我们设置λ = 10，因此 cls 和 reg 术语大致相等。我们通过实验表明，结果对宽范围的值不敏感（表 9）。我们还注意到，上述标准化不是必需的，可以简化。

对于边界框回归，我们采用[5]之后的 4 个坐标的参数化：

$$\begin{aligned} t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}), \\ t_{\mathrm{x}}^* &= (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}}^* &= \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}), \end{aligned} \quad (2)$$

其中 x，y，w 和 h 表示盒子的中心坐标及其宽度和高度。变量 x，x_{a}和x^*分别用于预测框，锚框和完全真实框（同样适用于 y, w, h）。 这个可以被认为是从锚到附近完全真实盒的边界框回归。

然而，我们的方法通过与先前基于 RoI（感兴趣区域）的方法[1]，[2]不同的方式实现了边界框回归。 在[1]，[2]中，对从任意大小的 RoI 汇集的特征执行边界框回归，并且回归权重由所有大小区域共享。 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（3×3）。 为了考虑不同的大小，学习了一组 k 个边界框回归量。 每个回归量负责一个比例和一个纵横比，并且 k 个回归量不共享权重。 因此，由于锚的设计，即使特征具有固定尺寸/比例，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）进行端到端训练[35]。我们遵循[2]中的“以图像为中心”的采样策略来训练这个网络。每个小批量产生于包含许多正面和负面示例锚点的单个图像。有可能优化所有锚的损失函数，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 个锚点来计算小批量的损失函数，其中采样的正和负锚点的比率高达 1：1。如果图像中的正样本少于 128 个，我们将小批量填充为负数。

我们通过从标准偏差为 0.01 的零均值高斯分布中绘制权重来随机初始化所有新图层。通过预训练 ImageNet 分类模型[36]来初始化所有其他层（即共享卷积层），这是标准的实践[5]。我们调整 ZF 网络的所有层，并将 conv3_1 和更高版本用于 VGG 网络以节省内存[2]。我们对 60k 小批量使用 0.001 的学习率，而对 PASCAL VOC 数据集的下一个 20k 小批量使用 0.0001 的学习率。我们使用 0.9 的动量和 0.0005 的重量衰减[37]。我们的实现使用 Caffe [38]。

3.2 与 RPN 和 Fast R-CNN 共享特征

到目前为止，我们已经描述了如何训练用于区域提案生成的网络，而不考虑将利用这些基于提案区域的对象检测 CNN。 对于检测网络，我们采用 Fast R-CNN [2]。 接下来，我们描述了学习由 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 都将以不同方式修改其卷积层。 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

表 1：使用 ZF 网络学习的每个锚点的平均提案大小（数值 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	188×111	113×114	70×92	416×229	261×284	174×332	768×437	499×501	355×715

而不是学习两个单独的网络。我们讨论了三种训练具有共享特征的网络的方法：

（i）**交替训练。**在此解决方案中，我们首先训练 RPN，并使用提案来训练 Fast R-CNN。然后，使用由 Fast R-CNN 调谐的网络来初始化 RPN，并且迭代该过程。这是本文所有实验中使用的解决方案。

（ii）**近似的联合训练。**在该解决方案中，RPN 和 Fast R-CNN 网络在训练期间被合并到一个网络中，如图 2 所示。在每个 SGD 迭代中，前向传递生成区域提案，其在训练时被视为固定的，预先计算的部分。Fast R-CNN 探测器。反向传播如常进行，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组合。该解决方案易于实施。但是这个解决方案忽略了衍生情况 w.r.t 提案框的坐标也是网络响应，因此是近似值。在我们的实验中，我们通过实验发现，该解算器产生了接近的结果，但与交替训练相比，训练时间缩短了约 25-50％。此解算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，由 RPN 预测的边界框也是输入的函数。 Fast R-CNN 中的 RoI 池层[2]接受卷积特征以及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也应该涉及梯度 w.r.t.框坐标。在上述近似联合训练中忽略这些梯度。在非近似联合训练解决方案中，我们需要一个可区分的 RoI 池层 w.r.t.框坐标。这是一个非常重要的问题，并且可以通过[15]中开发的“RoI 翘曲”层给出解决方案，这超出了本文的范围。

四步交替训练。 在本文中，我们采用实用的 4 步训练算法来通过交替优化来学习共享特征。 在第一步中，我们按照 3.1.3 节的描述训练 RPN。 该网络使用 ImageNet 预先训练的模型进行初始化，并针对生成提案区域的任务进行端对端微调。 在第二步中，我们使用由步骤 1 RPN 生成的提案通过 Fast R-CNN 训练单独的检测网络。 该检测网络也由 ImageNet 预训练模型初始化。 此时，两个网络不共享卷积层。 在第三步中，我们使用探测器网络初始化 RPN 训练，但我们修复共享卷积层并仅微调 RPN 特有的层。 现在这两个网络共享卷积层。 最后，保持共享卷积层固定，我们微调 Fast R-CNN 的独特层。 因此，两个网络共享相同的卷积层并形成统一的网络。 可以进行类似的交替训练以进行更多迭代，但我们观察到近乎可以忽略的提升。

3.3 实现细节

我们在单一尺度的图像上训练和测试区域提案和物体检测网络[1]，[2]。我们重新缩放图像，使其短边为 s = 600 像素[2]。多尺度特征提取（使用图像金字塔）可以提高准确性，但不会表现出良好的速度 - 准确性权衡[2]。在重新缩放的图像上，最后一个卷积层上的 ZF 和 VGG 网络的总步幅是 16 个像素，因此在调整大小之前在典型的 PASCAL 图像上是 10 个像素（500 × 375）。即使是如此大的步幅也能提供良好的效果，尽管可以通过更小的步幅进一步提高精度。

对于锚点，我们使用 3 个刻度，框区域为128^2，256^2和512^2像素，3 个宽高比为 1：1,1：2 和 2：1。对于特定数据集，这些超参数并未仔细选择，我们将在下一节中提供有关其效果的消融实验。如上所述，我们的解决方案不需要图像金字塔或滤波金字塔来预测多个尺度的区域，从而节省了大量的运行时间。图 3（右）显示了我们的方法适用于各种比例和纵横比的能力。表 1 显示了使用 ZF 网络的每个锚点的平均学习提案大小。我们注意到，我们的算法允许预测大于潜在的感受野。这样的预测并非不可能 - 如果只有对象的中间可见，人们仍然可以大致推断出对象的范围。

需要小心处理跨越图像边界的锚。在训练期间，我们忽略所有跨境锚点，因此它们不会造成损失。对于典型的 1000 × 600 图像，总共将有大约 20000（约等于 60 × 40 × 9）个锚。由于忽略了跨界锚点，每个图像大约有 6000 个锚点用于训练。如果在训练中不忽略过境异常值，则会在目标中引入大的，难以纠正的误差项，并且训练不会收敛。然而，在测试期间，我们仍然将完全卷积 RPN 应用于整个图像。这可能会生成跨界提案框，我们将其剪切到图像边界。

表 2：**PASCAL VOC 2007 测试集**的检测结果（在 VOC 2007 trainval 上训练）。 探测器是带有 ZF 的 Fast R-CNN，但使用各种提案方法进行训练和测试。

train-time region proposals		test-time region proposals		
method	# boxes	method	# proposals	mAP (%)
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**
ablation experiments follow below				
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

一些 RPN 提案彼此高度重叠。 为了减少冗余，我们根据其 cls 分数对提案区域采用非最大值抑制（NMS）。 我们将 NMS 的 IoU 阈值修正为 0.7，这使得每个图像大约有 2000 个提案区域。 正如我们将要展示的那样，NMS 不会损害最终的检测准确性，但会大大减少提案的数量。 在 NMS 之后，我们使用排名前 N 的提案区域进行检测。 在下文中，我们使用 2000 RPN 提议训练 Fast R-CNN，但在测试时评估不同数量的提议。

4 实验

4.1 在 PASCAL VOC 上的实验

我们全面评估了 PASCAL VOC 2007 检测基准的方法[11]。 该数据集由大约 5k 个 trainval 图像和 20 个对象类别的 5k 个测试图像组成。 我们还针对少数型号提供 PASCAL VOC 2012 基准测试的结果。 对于 ImageNet 预训练网络，我们使用具有 5 个卷积层和 3 个完全连接层的 ZF 网络[32]的“快速”版本，以及具有 13 个卷积层的公共 VGG-16 模型（见页底 7） [3] 和 3 个完全连接的层。 我们主要评估检测平均精度（mAP），因为这是对象检测的实际度量（而不是关注对象提议的代理度量）。

表 2（上图）显示了使用各种区域提案方法进行训练和测试时的 Fast R-CNN 结果。 这些结果使用 ZF 网络。 对于选择性搜索（SS）[4]，我们通过“快速”模式生成大约 2000 个提案。 对于 EdgeBoxes（EB）[6]，我们通过调整 IoU 为 0.7 的默认 EB 设置生成提案。

7. www.robots.ox.ac.uk/~vgg/research/very_deep/

在 Fast R-CNN 框架下，SS 的 mAP 为 58.7%，EB 的 mAP 为 58.6%。 具有 Fast R-CNN 的 RPN 实现了有竞争力的结果，mAP 为 59.9%，同时使用多达 300 个提案（见页底 8）。 由于共享卷积计算，使用 RPN 产生了比使用 SS 或 EB 快得多的检测模型; 较少的提案也减少了区域完全连接层的计算成本（表 5）。

RPN 的消融实验。为了研究 RPN 作为一种提案方法的可行性，我们进行了几项消融研究。首先，我们展示了在 RPN 和 Fast R-CNN 检测网络之间共享卷积层的效果。为此，我们在 4 步训练过程的第二步后停止。使用单独的网络将结果略微降低至 58.7%（RPN + ZF，非共享，表 2）。我们观察到这是因为在第三步中，当使用检测器调谐的特征来微调 RPN 时，提案质量得到改善。

接下来，我们将解开 RPN 对训练 Fast R-CNN 检测网络的影响。为此，我们使用 2000 SS 提案和 ZF 网络训练 Fast R-CNN 模型。我们通过更改在测试时使用的提案区域来修复此检测器并评估检测 mAP。在这些消融实验中，RPN 不与检测器共享特征。

在测试时用 300 个 RPN 提议替换 SS 导致 mAP 为 56.8%。 mAP 的损失是由于训练/测试提案之间的不一致。该结果作为后续比较的基线。

有点令人惊讶的是，当使用排名靠前的 RPN 时，RPN 仍然会带来竞争结果（55.1%）

8. 对于 RPN，提议的数量（例如，300）是图像的最大数量。 在 NMS 之后，RPN 可能会产生较少的提案，因此平均提案数量较少。

表 3：**PASCAL VOC 2007 测试集**的检测结果。 探测器是 Fast R-CNN 和 VGG-16。 训练数据："07"：VOC 2007 trainval，"07 + 12"：VOC 2007 trainval 和 VOC 2012 trainval 的联合组合。 对于 RPN，Fast R-CNN 的训练时间建议是 2000。 †：这个数字在[2]中提及; 使用本文提供的存储库，此结果更高（68.1）。

method	# proposals	data	mAP (%)
SS	2000	07	66.9†
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	**73.2**
RPN+VGG, shared	300	COCO+07+12	**78.8**

表 4：**PASCAL VOC 2012 测试集**的检测结果。 探测器是 Fast R-CNN 和 VGG-16。 训练数据："07"：VOC 2007 trainval，"07 ++ 12"：VOC 2007 trainval + test 和 VOC 2012 trainval 的联合组合。 对于 RPN，Fast R-CNN 的训练时间建议是 2000。 †：http://host.robots.ox.ac.uk:8080/anonymous/HZJTQA.html.
‡:http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html. §: 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†	300	12	67.0
RPN+VGG, shared‡	300	07++12	**70.4**
RPN+VGG, shared§	300	COCO+07++12	**75.9**

表 5：K40 GPU 上的**时间**（ms），但 SS 提案在 CPU 中评估。 "区域性"包括 NMS，池化层，完全连接层和 softmax 层。 请参阅我们发布的运行时分析代码。

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	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

在测试时提交了 100 个提案，表明排名靠前的 RPN 提案是准确的。另一方面，使用排名靠前的 6000 个 RPN 提案（没有 NMS）具有有竞争力的 mAP（55.2%），提案部分的 NMS 不会损害检测 mAP 并可能减少误报。

接下来，我们通过在测试时关闭其中任何一个来分别调查 RPN 的 cls 和 reg 输出的作用。当在测试时移除 cls 层（因此不使用 NMS/排名）时，我们从未划分的区域中随机抽样 N 个提案。 mAP 几乎没有变化，N = 1000（55.8%），但是当 N = 100 时，mAP 显着降低到 44.6%。这表明 cls 分数考虑了排名最高的提案的准确性。

另一方面，当在测试时移除 reg 层（因此提案成为锚盒）时，mAP 降至 52.1%。这表明高质量的提案主要是由于回归的边界限制。锚盒虽然具有多个刻度和纵横比，但不足以进行精确检测。

我们还评估了更强大的网络作用对 RPN 的提案质量的影响。我们使用 VGG-16 训练 RPN，仍然使用上面的 SS + ZF 探测器。 mAP 从 56.8%（使用 RPN + ZF）提高为 59.2%（使用 RPN + VGG）。这是一个很有希望的结果，因为它表明 RPN + VGG 的提案质量优于 RPN + ZF。由于 RPN + ZF 的提案与 SS 竞争（一直用于训练和测试时均为 58.7%），我们可能期望 RPN + VGG 优于 SS。以下实验证明了这种假设是正确的。

VGG-16 的表现。表 3 显示了 VGG-16 对提案和检测的结果。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 生成的提案比 SS 更准确。与预先定义的 SS 不同，RPN 受到积极训练并受益于更好的网络。对于特征共享变体，结果是 69.9%，比强 SS 基线更好，但提案几乎没有成本。我们在 PASCAL VOC 2007 trainval 和 2012 trainval 的数据集上进一步训练 RPN 和检测网络， mAP 为 73.2%。图 5 显示了 PASCAL VOC 2007 测试集的一些结果。在 PASCAL VOC 2012 测试集（表 4）中，我们的方法在 VOC 2007 trainval + test 和 VOC 2012 trainval 的联合数据集上进行了训练，mAP 达到了 70.4%。表 6 和表 7 显示了详细的数字。

表 6：使用 Fast R-CNN 检测器和 VGG-16 的 PASCAL VOC 2007 测试集的结果。 对于 RPN，Fast R-CNN 的训练时间建议是 2000。RPN 取消共享特征版本。

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**

表 7：使用 Fast R-CNN 检测器和 VGG-16 的 PASCAL VOC 2012 测试集的结果。 对于 RPN，Fast R-CNN 的训练时间建议是 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**

表 8：使用不同锚设置在 PASCAL VOC 2007 测试集上 Faster R-CNN 的检测结果。 该网络是 VGG-16。 训练数据是 VOC 2007 trainval。 使用 3 个刻度和 3 个纵横比（69.9%）的默认设置与表 3 中的相同。

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128^2	1:1	65.8
	256^2	1:1	66.7
1 scale, 3 ratios	128^2	{2:1, 1:1, 1:2}	68.8
	256^2	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	$\{128^2, 256^2, 512^2\}$	1:1	**69.8**
3 scales, 3 ratios	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}	**69.9**

表 9：使用等式（1）中的不同 λ 值在 PASCAL VOC 2007 测试集上 Faster R-CNN 的检测结果。 该网络是 VGG-16。 训练数据是 VOC 2007 trainval。 使用 λ= 10（69.9%）的默认设置与表 3 中的相同。

λ	0.1	1	10	100
mAP (%)	67.2	68.9	69.9	69.1

在表 5 中，我们总结了整个物体检测系统的运行时间。SS 取决于内容需要 1-2 秒（平均约 1.5 秒），而带有 VGG-16 的 Fast R-CNN 在 2000 SS 提案上需要 320 毫秒（如果在完全连接层上使用 SVD，则需要 223 毫秒[2]）。 我们的 VGG-16 系统在提案和检测方面总共需要 198 毫秒。 由于共享卷积特征，仅使用 RPN 计算附加层需要 10ms。 由于提案较少（每张图像 300 张），我们的区域计算次数也较少。 我们的系统的 ZF 网络帧速率为 17 fps。

对超参数的敏感性。 在表 8 中，我们研究了锚的设置。默认我们使用 3 个刻度和 3 个纵横比（表 8 中的 69.9%mAP）。如果在每个位置仅使用一个锚点，则 mAP 下降 3-4%。如果使用 3 个刻度（具有 1 个纵横比）或 3 个纵横比（具有 1 个刻度），则 mAP 更高，证明使用多个尺寸的锚作为回归参考是有效的解决方案。仅使用 3 个具有 1 个纵横比（69.8%）的比例与在该数据集上使用具有 3 个纵横比的 3 个比例一样好，表明比例和纵横比不是解开检测精度的关键。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，我们比较了方程（1）中的不同值。默认情况下，我们使用 λ = 10，这使得等式（1）中的两个项在归一化后大致相等地加权。表 9 显示，当在约两个数量级（1 到 100）的范围内时，我们的结果仅略微受影响（减少 1%）。这表明结果在很大范围内不敏感。

对 Recall-to-IoU 的分析。 接下来，我们使用完全真实框计算不同 IoU 比率的提案召回率。 值得注意的是，Recall-to-IoU 指标与最终检测精度有关[19]，[20]，[21]。使用此度量标准来诊断提案方法比评估提案方法更合适。

在图 4 中，我们显示了使用 300,1000 和 2000 提案的结果。 我们与 SS 和 EB 进行比较，并且基于这些方法产生的置信度，N 个提案是指排名前 N 的提案。 这些图显示，当提案数量从 2000 下降到 300 时，RPN 方法表现得很好。这解释了为什么 RPN 在使用少至 300 个提案时具有良好的最终检测 mAP。 正如我们之前分析的那样，这个属性主要归因于 RPN 的 cls 术语。 当提案较少时，SS 和 EB 的召回比 RPN 下降得更快。

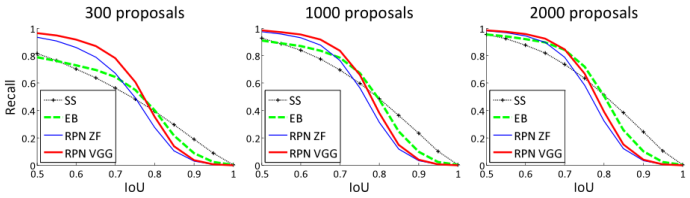

图 4：PASCAL VOC 2007 测试集的召回率与 IoU 重叠率。

表 10：**一阶段检测与两阶段提案+检测对比。** 检测结果使用 ZF 模型和 Fast R-CNN 在 PASCAL VOC 2007 测试集上进行。 RPN 不使用共享特征。

	proposals		detector	mAP (%)
Two-Stage	RPN + ZF, unshared	300	Fast R-CNN + ZF, 1 scale	58.7
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 1 scale	53.8
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 5 scales	53.9

一阶段检测与两阶段提案+检测。 OverFeat 论文[9]提出了一种检测方法，该方法在卷积特征映射上的滑动窗口上使用回归量和分类器。 OverFeat 是一个一阶段，特定于类的检测管道，我们的是一个两阶段级联，由不可知类的提议和特定类的检测组成。在 OverFeat 中，区域特征来自于比例金字塔上的一个纵横比的滑动窗口。这些功能用于同时确定对象的位置和类别。在 RPN 中，这些特征来自方形（3×3）滑动窗口，并预测相对于具有不同尺度和纵横比的锚点的提案。虽然这两种方法都使用滑动窗口，但区域提案任务只是 Fast R-CNN 的第一阶段 - 下游 Fast R-CNN 探测器参与了改进它们的提案。在我们的级联的第二阶段，区域性特征被自适应地汇集[1]，[2]来自提案框，更加充分地覆盖了区域的特征。我们相信这些特征可以提供更准确的检测。

为了比较单级和两级系统，我们通过一级 Fast R-CNN 模拟 OverFeat 系统（从而也避免了实现细节的其他差异）。在该系统中，“提案”是 3 个刻度（128,256,512）和 3 个纵横比（1：1,1：2,2：1）的密集滑动窗口。训练 Fast R-CNN 以预测特定类分数并从这些滑动窗口回归框位置。由于 OverFeat 系统采用图像金字塔，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]中的那 5 个尺度。

表 10 比较了两级系统和一级系统的两个变体。使用 ZF 模型，单级系统的 mAP 为 53.9％。这比两阶段系统（58.7%）低 4.8%。该实验证明了级联区域提案和对象检测的有效性。在[2]，[39]中报告了类似的观察结果，它们取代了 SS 具有滑动窗口的区域提案导致两种论文的结果均降低了约 6%。 我们还注意到，单阶段系统较慢，因为它需要处理更多的提案。

4.2 在 MS COCO 上的实验

我们在 Microsoft COCO 对象检测数据集[12]上提供了更多结果。该数据集涉及 80 个对象类别。我们在训练集上测试 80k 图像，在验证集上测试 40k 图像，在 test-dev 集上测试 20k 图像。我们评估 IoU $\in$ [0.5：0.05：0.95]的平均 mAP（COCO 的标准度量，简称为 mAP@[.5，.95]）和 mAP@0.5（PASCAL VOC 的度量标准）。

我们的系统对此数据集进行了一些细微的更改。我们在 8-GPU 上实现训练我们的模型，RPN（每个 GPU 1 个）和 16 个 Fast R-CNN（每个 GPU 2 个）的有效小批量大小为 8。RPN 步骤和 Fast R-CNN 步骤都训练为 240k 次迭代，学习率为 0.003，然后进行 80k 次迭代，学习率为 0.0003。我们修改了学习率（从 0.003 开始而不是 0.001），因为小批量大小已经改变。对于锚点，我们使用 3 个纵横比和 4 个比例（添加64^2），主要是通过处理此数据集上的小对象来实现的。此外，在我们的 Fast R-CNN 步骤中，负样本被定义为具有最大 IoU 和完全真实的那些[0，0.5），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）用于网络微调，但负样本在使用困难负例挖掘的 SVM 步骤中仍然访问[0，0.5）。但是 Fast R-CNN 系统[2]放弃了 SVM 步骤，所以[0，0.1）从未访问过。包括这些[0，0.1）对于 Fast R-CNN 和 Fast R-CNN 系统，样本在 COCO 数据集上提高了 mAP@0.5 的值（但对 PASCAL VOC 的影响可以忽略不计）。

表 11：**MS COCO 数据集**上的对象检测结果（%）。 模型为 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**

其余的实施细节与 PASCAL VOC 相同。特别是，我们继续使用 300 个提案和单一规模（s = 600）测试。 COCO 数据集上的每个图像的测试时间仍然约为 200ms。

在表 11 中，我们首先使用本文中的实现报告了 Fast R-CNN 系统[2]的结果。我们的 Fast R-CNN 基线在测试开发组中具有 39.3%mAP@0.5，高于[2]中报告的基线。我们推测这种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]只有可比性但不具竞争力。

接下来我们评估 Faster R-CNN 系统。使用 COCO 训练集进行训练，Faster R-CNN 在 COCO 测试集中具有 42.1% mAP@0.5 和 21.5%mAP@[.5，.95]。对于 mAP@0.5，这比同样的协议下的 Fast R-CNN 对应物的 mAP@0.5 高 2.8%，比 mAP@[.5，.95]高出 2.2%（表 11）。这表明 RPN 在更高的 IoU 阈值上表现出色，可以提高定位精度。使用 COCO 训练集进行训练，Faster R-CNN 在 COCO 测试集上具有 42.7%mAP@0.5 和 21.9%mAP@[.5，.95]。图 6 显示了 MS COCO test-dev 集的一些结果。

ILSVRC 和 COCO 2015 竞赛。Faster R-CNN 我们已经证明，由于 RPN 完全学会通过神经网络生成提案区域，Faster R-CNN 从更好的特征中获益更多。即使将深度基本上增加到超过 100 层，这种观察仍然有效[18]。只有将 VGG-16 替换为 101 层剩余网络（ResNet-101）[18]，Faster R-CNN 在 COCO val 集上将 mAP 从 41.5%/ 21.2%（VGG-16）增加到 48.4%/ 27.2%（ResNet-101）。通过与 Faster R-CNN 正交的其他改进措施，He 等人[18]获得了单一模型结果 55.7％/ 34.9％，COCO test-dev 集的综合结果为 59.0%/ 37.4%，在 COCO 2015 物体检测竞赛中获得第一名。同样的系统[18]也在 ILSVRC 2015 物体检测竞赛中获得第一名，超过第二名绝对值 8.5％。 RPN 也是 ILSVRC 2015 本地化和 COCO 2015 细分竞争中第一名获奖作品的构建模块，详情分别见[18]和[15]。

表 12：在 PASCAL VOC 2007 测试集和 2012 测试集上使用不同的训练数据检测 Faster R-CNN 的 mAP（%）。 该模型为 VGG-16。 “COCO”表示 COCO 训练集用于训练。 另见表 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	**78.8**	-
COCO+VOC07++12	-	**75.9**

4.3 从 MS COCO 到 PASCAL VOC

大规模数据对于改进深度神经网络至关重要。接下来，我们将研究 MS COCO 数据集如何帮助 PASCAL VOC 的检测性能。

作为一个简单的基线，我们直接评估 PASCAL VOC 数据集上的 COCO 检测模型，而无需对任何 PASCAL VOC 数据进行微调。此评估是可行的，因为 COCO 上的类别是 PASCAL VOC 上的类别的超集。在此实验中忽略了 COCO 专用的类别，而 softmax 层仅在 20 个类别和背景上执行。在 PASCAL VOC 2007 测试集下，mAP 达到了 76.1%（表 12）。尽管 PASCAL VOC 数据未被利用，但这一结果优于 VOC 07 + 12（73.2%）的训练。

然后我们微调 VOC 数据集上的 COCO 检测模型。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），并且 Faster R-CNN 系统进行了微调，如第 3.2 节所述。这样做可以使 PASCAL VOC 2007 测试集的 mAP 达到 78.8%。来自 COCO 集的额外数据使 mAP 增加 5.6%。表 6 显示，在 PASCAL VOC 2007 上，使用 COCO + VOC 训练的模型具有每个类别的最佳 AP。在 PASCAL VOC 2012 测试集上观察到类似的改进（表 12 和表 7）。我们注意到，获得这些强大结果的测试时间仍然是每张图像大约 200 毫秒。

5 结论

我们已经提出了 RPN，以便有效和准确地生成区域提案。通过共享

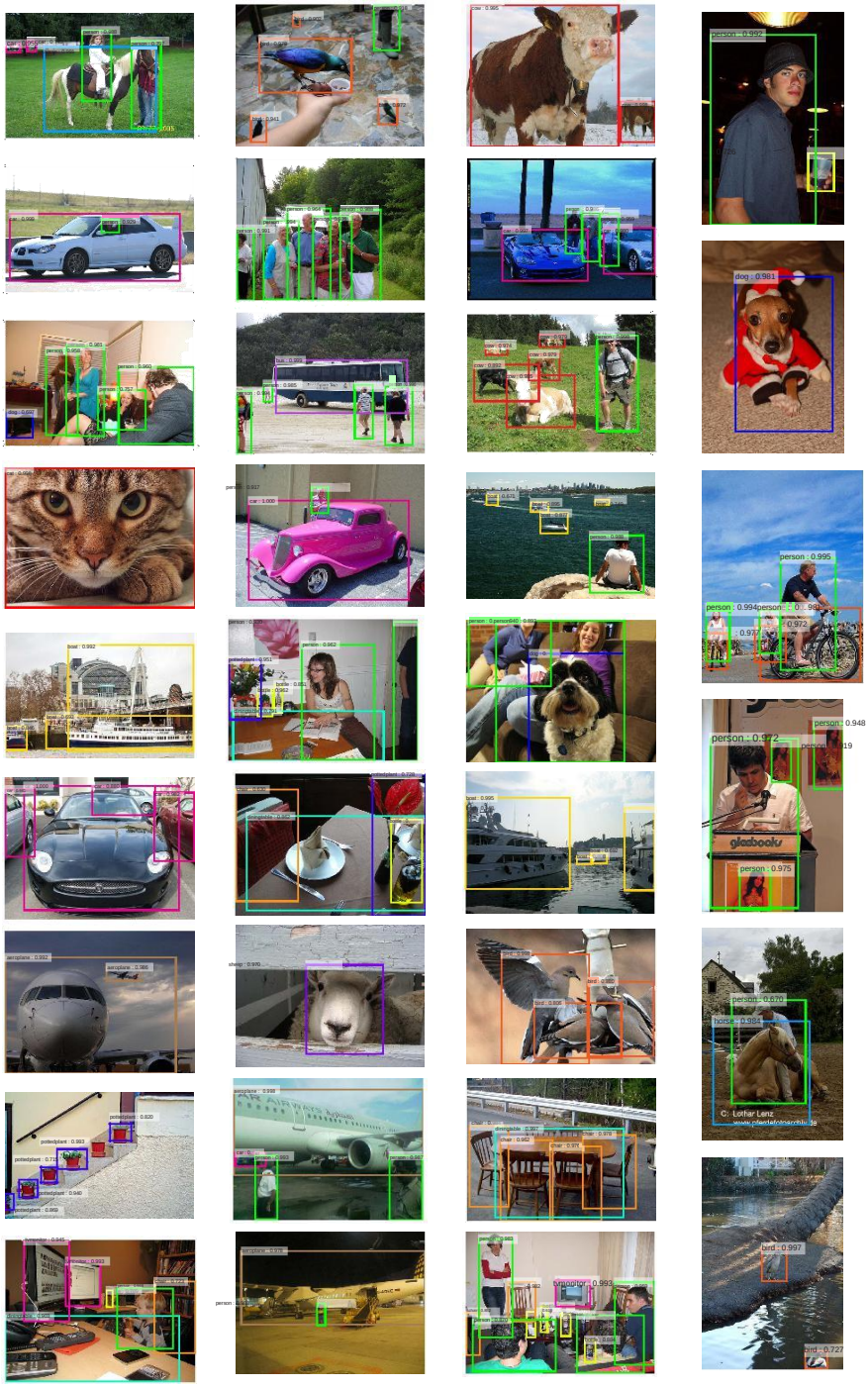

图 5：使用 Faster R-CNN 系统在 PASCAL VOC 2007 测试集上选择物体检测结果的例子。 该模型为 VGG-16，训练数据为 07 + 12 trainval（2007 年测试集上为 73.2%mAP）。 我们的方法检测各种尺度和纵横比的物体。 每个输出框与类别标签和 softmax 分数$\in$ [0，1]相关联。 得分阈值 0.6 用于显示这些图像。 获得这些结果的运行时间为每个图像 198ms，包括所有步骤。

具有下游检测网络的卷积特征，区域提案步骤几乎是免费的。 我们的方法使基于深度学习的统一对象检测系统能够以接近实时的帧速率运行。 学习到的 RPN 还改善了区域提案质量，从而提高了整体物体检测精度。

参考文献

[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

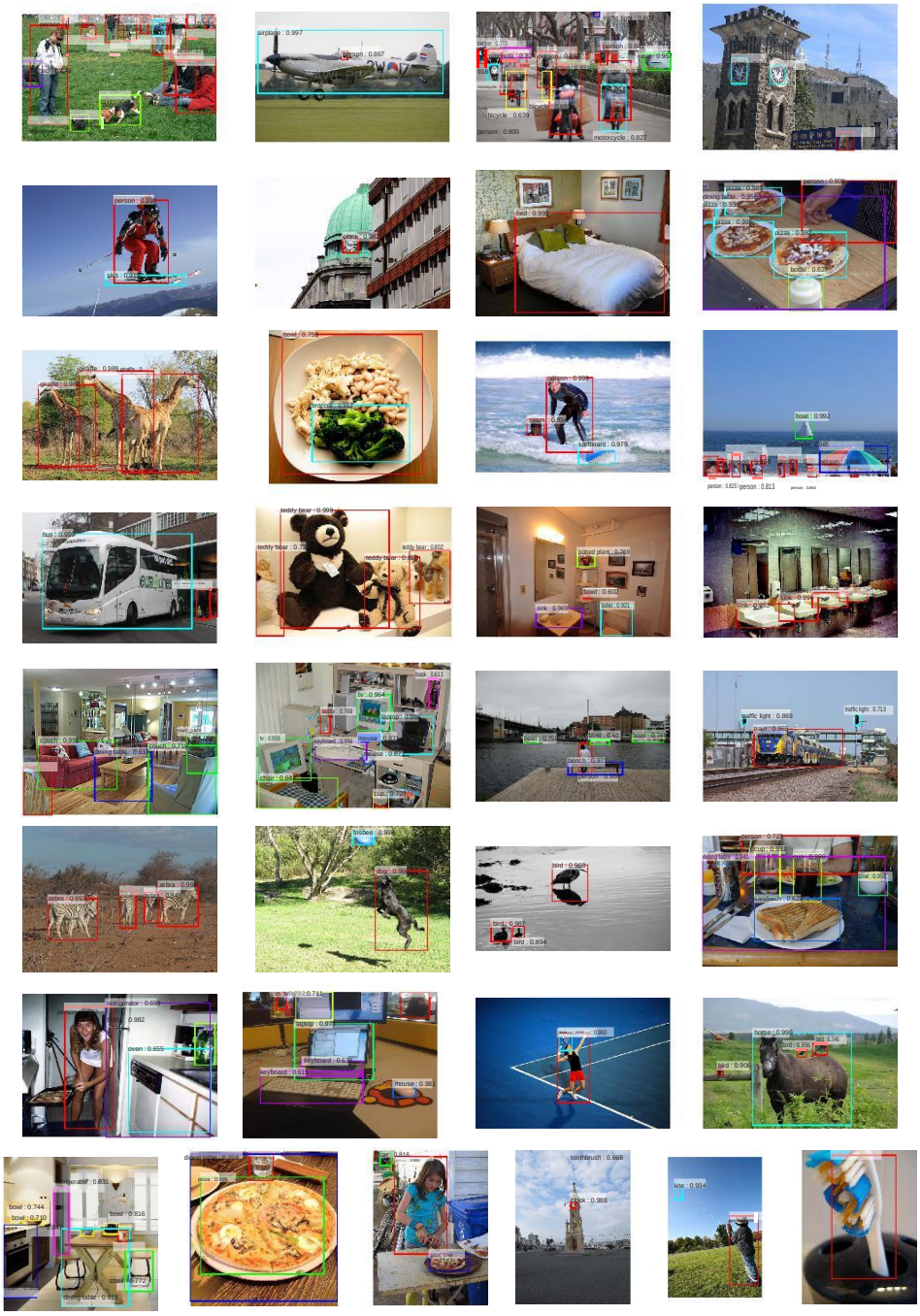

图 6：使用 Faster R-CNN 系统在 MS COCO test-dev 集上设置的对象检测结果的选定示例。 该模型为 VGG-16，训练数据为 COCO trainval（测试集 42.7%mAP@0.5）。 每个输出框与类别标签和 softmax 分数$\in$ [0，1]相关联。 得分阈值 0.6 用于显示这些图像。 对于每个图像，一种颜色代表该图像中的一个对象类别。

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. Smeul-ders, “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. Dollar,´ “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 Ma-chine 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.
[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar,´ 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 segmentation 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 detection proposals, really?" in British Machine Vision Conference (BMVC), 2014.
[20] J. Hosang, R. Benenson, P. Dollar,´ 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 ob-ject segmentation using constrained parametric min-cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2012.
[23] P. Arbelaez,´ 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, "Ob-ject 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.