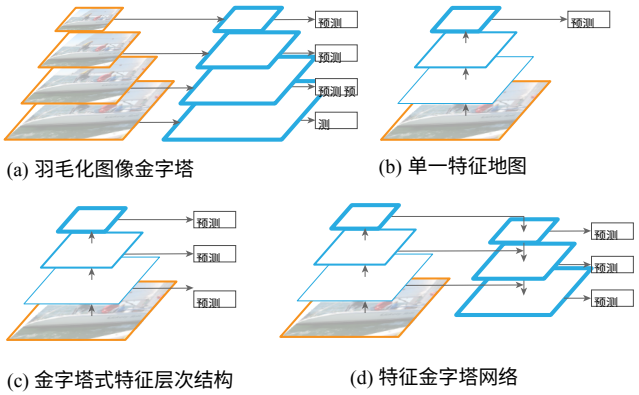
用于物体检测的特征金字塔网络

Tsung-Yi Lin[1,2] , Piotr Dolla´r[1] , Ross Girshick[1] , Kaiming He[1] , Bharath Hariharan[1] , and Serge Belongie[2]

1Facebook人工智能研究（FAIR）
2康奈尔大学和康奈尔理工大学

摘要

特征金字塔是识别系统中检测不同尺度物体的一个基本组成部分。但是最近的深度学习物体检测器避免了金字塔的表现，部分原因是它们是计算和内存密集型的。在本文中，我们利用深度卷积网络固有的多尺度、金字塔式的层次结构，以微弱的额外成本构建特征金字塔。我们开发了一个自上而下的横向连接的架构，用于构建所有尺度的高级语义特征图。这个架构被称为特征金字塔网络（FPN）、
在一些应用中，FPN作为一种通用的特征外延显示出明显的改进。在一个基本的Faster R-CNN系统中使用FPN，我们的方法在COCO检测基准上取得了最先进的单模型结果，没有任何花哨的东西，超过了所有现有的单模型能力，包括来自COCO 2016挑战赛的获胜者。此外，我们的方法可以在GPU上以6FPS的速度运行，因此是一个实用而准确的多尺度物体检测解决方案。代码将被公开提供。



(a) 羽毛化图像金字塔　(b) 单一特征地图

(c) 金字塔式特征层次结构　(d) 特征金字塔网络

1. 简介

识别不同尺度的物体是计算机视觉中一个有趣的挑战。*建立在图像金字塔基础上的特征金字塔*（我们简称为*特征化图像金字塔*）构成了标准解决方案的基础。

[1]（图1（a））。这些金字塔是尺度不变的，也就是说，物体的尺度变化会被其在金字塔中的级别所抵消。直观地说，这一特性使模型能够通过扫描模型的位置和金字塔级别来检测大范围内的物体。在手工设计特征的时代，羽化图像金字塔被大量使用[5, 25]。它们是如此关键，以至于像DPM[7]这样的物体检测器需要密集的尺度采样来达到良好的效果（*例如*，每八度10个尺度）。对于识别任务，工程化特征有

图1. (a) 使用图像金字塔来构建特征金字塔。
特征是在每个图像尺度上独立计算的，这很慢。 (b) 最近的检测系统选择只使用单一尺度的特征，以加快检测。(c) 另一种方法是重新使用由ConvNet计算的金字塔特征层次，就像它是一个特征化的图像金字塔。(d) 我们提出的特征金字塔网络（FPN）与(b)和(c)一样快速，但更准确。在该图中，特征图由蓝色的轮廓表示，较粗的轮廓表示语义上更强的特征。

大部分被深度卷积网络（ConvNets）计算的特征所取代[19, 20]。除了能够代表更高层次的语义外，ConvNets对比例的变化也更加稳健，因此便于从单一输入比例计算的特征中进行识别[15, 11, 29]（图1（b））。但是，即使有这种鲁棒性，仍然需要pyra-mids来获得最准确的结果。在ImageNet[33]和COCO[21]的检测挑战中，所有排名靠前的条目都使用了对特征图像金字塔的多尺度测试（*例如*，[16，35]）。对图像金字塔的每个层次进行特征化的主要优势在于，它能产生一个多尺度的特征表示，其中*所有层次的语义都很强*，包括高分辨率的层次。

然而，对图像 pyra-mid 的每一层进行featurizing有明显的局限性。推理时间大大增加（*例如*，四倍[11]），使得这种方法在实际应用中不实用。此外，训练深度

在图像金字塔上端到端的网络在内存方面是不可行的，因此，如果利用的话，图像金字塔只在测试时使用[15, 11, 16, 35]，这造成了训练/测试时间推理的不一致。由于这些原因，Fast和Faster R-CNN[11, 29]选择在默认设置下不使用fea-turized image pyramids。

然而，图像金字塔并不是计算多尺度特征表示的唯一方法。深度ConvNet逐层计算*特征层次*，通过子采样层，特征层次有一个固有的多尺度、金字塔形状。这种网络内的特征层次产生了不同空间分辨率的特征图，但却引入了不同深度造成的巨大语义差距。高分辨率的地图具有低层次的特征，损害了它们对物体识别的表示能力。

单次检测器（SSD）[22]是首次尝试使用ConvNet的金字塔特征层次，就好像它是一个特征化的图像金字塔（图1（c））。理想情况下，SSD风格的金字塔会重复使用前向计算的不同层的多尺度特征图，因此是免费的。但是为了避免使用低层次的特征，SSD放弃了重复使用已经计算过的层，而是从网络的高层开始建立金字塔（*例如*，VGG网络的conv4 3[36]），然后增加几个新的层。因此，它错失了重用特征层次的高分辨率地图的机会。我们表明，这些地图对检测小物体很重要。

本文的目标是自然地利用ConvNet特征层次的pyra-midal形状，同时创建一个在所有尺度上都具有强语义的特征金字塔。为了实现这一目标，我们依靠一种架构，通过自上而下的途径和横向连接，将低分辨率、语义强的特征与高分辨率、语义弱的特征相结合（图1（d））。其结果是一个特征金字塔，该金字塔在各个层面都有丰富的语义，并能从单一的输入图像比例中快速构建。换句话说，我们展示了如何创建网络内的特征金字塔，这些特征金字塔可以用来替代特征化的图像金字塔，而不会牺牲表示能力、速度或内存。

采用自上而下和跳过连接的类似架构在最近的研究中很受欢迎[28, 17, 8, 26]。他们的目标是产生一个精细分辨率的单一高层特征图，在此基础上进行预测（图2顶部）。相反，我们的方法利用结构作为一个特征金字塔，预测（*例如*，物体检测）是在每一层独立进行的（图2底部）。我们的模型与特征金字塔相呼应，在这些工作中还没有探讨过这个问题。

我们评估了我们的方法，称为特征金字塔网（FPN），在各种系统中进行检测和分离[11, 29, 27]。在没有任何附加条件的情况下，我们在具有挑战性的COCO检测基准[21]上，仅仅基于FPN就重新得到了最先进的单模型结果。

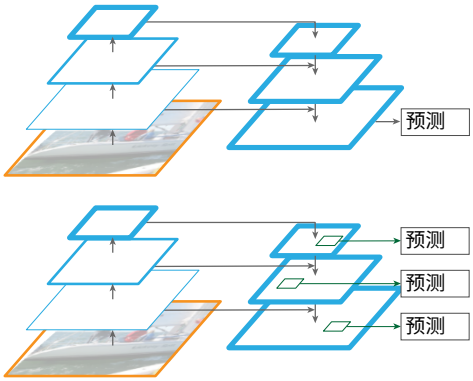


图2.上图：自上而下的结构，有跳过的连接，在最细的层次上进行预测（*例如*，[28]）。底部：我们的模型具有类似的结构，但将其作为一个*特征金字塔来*利用，在所有层面上独立进行预测。

一个基本的Faster R-CNN检测器[29]，超过了所有现有的、经过大量工程设计的竞争获胜者的单一模型作品。在消融实验中，我们发现，对于边界盒建议，FPN将平均召回率（AR）显著提高了8.0分；对于物体检测，它将COCO式的平均精度（AP）提高了2.3分，将PASCAL式的AP提高了3.8分，超过了ResNets上Faster R-CNN的强大单尺度基线[16]。我们的方法也很容易扩展到掩码建议，并比严重依赖图像金字塔的最先进的方法提高了实例分割的AR和速度。

此外，我们的金字塔结构可以对所有尺度进行端到端的训练，并在训练/测试时一致使用，这在使用图像金字塔时是不可行的。因此，FPN能够达到比所有现有最先进的方法更高的精度。此外，这种提高是在不增加测试时间的情况下实现的，而不是在单尺度基线上。我们相信这些进展将促进未来的研究和应用。我们的代码将被公开提供。

2. 相关工作

手工设计的特征和早期的神经网络。SIFT特征[25]最初是在标度空间极值处提取的，用于特征点匹配。HOG特征[5]，以及后来的SIFT特征，是在整个图像金字塔上密集计算的。这些HOG和SIFT金字塔已被用于图像分类、物体检测、人体姿势估计等众多工作中。人们对快速计算特征化图像金字塔也有很大兴趣。Dolla´r等人[6]展示了快速金字塔的计算，首先计算一个稀疏采样（按比例）的金字塔，然后插值缺失的层次。在HOG和SIFT之前，用ConvNets[38, 32]进行的早期人脸检测工作是在图像金字塔上计算浅层网络，以检测不同尺度的人脸。

深度ConvNet物体检测器。随着现代深度ConvNets[19]的发展，像Over- Feat[34]和R-CNN[12]这样的物体检测器在准确性方面有了极大的提高。OverFeat采用了类似于早期神经网络人脸检测器的策略，将ConvNet作为滑动窗口检测器应用于图像金字塔。R-CNN采用了一种基于区域提议的策略[37]，在用ConvNet进行分类之前，每个提议都被尺度化了。SPPnet[15]表明，这种基于区域的检测器可以更有效地应用于在单一图像尺度上提取的图像图。最近更准确的检测方法，如快速R-CNN[11]和快速R-CNN[29]，主张使用从单一比例计算的特征，因为它在准确性和速度之间提供了一个良好的权衡。然而，多尺度检测仍然表现得更好，特别是对于小物体。

使用多层的方法。最近的一些方法通过在ConvNet中使用不同的层来改善检测和分割。FCN[24]将每个类别的部分分数在多个标度上相加以计算语义分割。Hypercolumns[13]使用类似的方法对物体实例进行分割。其他一些方法（HyperNet[18]、ParseNet[23]和ION[2]）在计算预测之前将多层的特征连接起来，这相当于将转换的特征相加。SSD [22]和MS-CNN[3]在不结合特征或分数的情况下预测特征层次的多层对象。最近有一些方法利用横向/跳跃连接，在不同的分辨率和语义层次上关联低层次的特征图，包括用于分割的U-Net[31]和Sharp- Mask[28]，用于人脸检测的Recombinator网络[17]，以及用于关键点估计的Stacked Hourglass网络[26]。Ghiasi等人[8]提出了一个用于FCN的Lapla- cian金字塔演示，以逐步完善分割。尽管这些方法采用了具有金字塔形状的架构，但它们与featurized图像金字塔[5, 7, 34]不同，在那里，预测是在所有层次上独立进行的，见图2。事实上，对于图2中的金字塔结构（顶部），仍然需要图像金字塔。来识别多尺度的物体[28]。

3. 特征金字塔网络

我们的目标是利用ConvNet的金字塔式特征层次，其语义从低到高，并建立一个具有高层次语义的特征金字塔。由此产生的*特征金字塔网络*是通用的，在本文中，我们专注于滑动窗口程序（Region Proposal Network，简称RPN）[29]和基于区域的检测器（Fast R-CNN）[11]。我们还在第6节中对实例分割建议生成了FPN。

我们的方法将任意大小的单尺度图像作为输入，并输出按比例大小的特征图

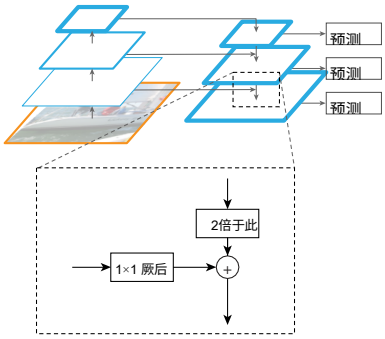


图3.一个说明横向联系和自上而下途径的构件，通过加法合并。

在多个层次上，以完全卷积的方式。这个过程独立于骨干卷积结构（*例如*，[19，36，16]），在本文中，我们使用ResNets[16]提出结果。我们的金字塔的构造包括一个自下而上的通路，一个自上而下的通路，以及后面的连接，如下所述。

自下而上的途径。自下而上的途径是骨干ConvNet的前馈计算，它以2的缩放步长，对由七种尺度的特征图组成的特征层次进行组合。通常有许多层产生相同大小的输出图，我们说这些层处于同一网络*阶段*。对于我们的特征金字塔，我们为每个阶段定义一个金字塔层。我们选择每个阶段最后一层的输出作为我们的特征图参考集，我们将充实这些特征图来创建我们的金字塔。这种选择是自然的，因为每个阶段的最深层应该有最强的特征。

具体来说，对于ResNets[16]，我们使用每个阶段的最后一个剩余块输出的特征激活。我们表示这些最后的剩余块的输出为C_2, C_3, C_4, C_5为conv2、conv3、conv4和conv5的输出，并注意到它们相对于输入图像有4、8、16、32像素的跨度。由于conv1的内存占用较大，我们没有将其纳入金字塔中。

自上而下的途径和横向连接。自上而下的途径通过对来自较高金字塔层的空间上较粗、但语义上较强的图谱进行上采样，幻化出更高分辨率的特征。然后，这些特征通过横向连接与来自自下而上途径的特征一起增强。每个侧向连接合并来自自下而上途径和自上而下途径的相同空间大小的特征图。自下而上的特征图是低层次的语义，但它的激活是更尖锐的定位，因为它被子采样的次数更少。

图3显示了构建我们自上而下的特征图的结构块。对于较粗分辨率的特征图，我们将空间分辨率提高2倍（为简单起见，使用近邻提高取样）。扩增后的

{ }

{ }

然后，该图与相应的自下而上的图（经历了1 1卷积层以减少通道尺寸）通过元素相加的方式合并。这个过程反复进行，直到生成最精细的分辨率地图。为了开始迭代，我们只需将一个1.1卷积层与一个1.2卷积层合并。
C上的层$_5$，以产生最粗的分辨率地图。最后，我们在每个合并的地图上附加一个3.3卷积，以便
生成最终的特征图，这是为了减少别名------。
的影响。这组最终的特征图被称为P_2，P，P_{34}，P_5，
分别对{应于空间大小}相同的C_2，C_3，C{C。$_{45}$ }

由于金字塔的所有层次都使用共享的分类器/回归器，就像传统的特征图像金字塔一样，我们固定了所有特征图的特征维度（通道数，表示为d）。在这个金字塔中，我们设定d=256，因此所有额外的卷积层都有256通道的输出。在这些额外层中没有非线性因素、
我们根据经验发现，这一点影响不大。

简洁性是我们设计的核心，我们发现我们的模型对许多设计选择是稳健的。我们对更复杂的模块进行了实验（*例如*，使用多层剩余模块[16]作为连接），观察到的结果稍好一些。设计更好的连接模块不是本文的重点，所以我们选择了上述的简单设计。

4. 应用

我们的方法是在深度ConvNets中建立特征金字塔的通用解决方案。在下文中，我们在RPN[29]中采用我们的方法进行边界盒提议生成，在Fast R-CNN[11]中采用我们的方法进行物体检测。为了证明我们方法的简单性和有效性，我们在将[29, 11]的原始系统适应于我们的特征金字塔时对其进行了最小的修改。

4.1. 用于RPN的特征金字塔网络

RPN[29]是一个滑动窗口的类别诊断物体检测器。在最初的RPN设计中，一个小的子网络在密集的3 3个滑动窗口上进行评估，在单尺度卷积特征图的基础上，进行物体/非物体二元分类和边界盒回归。这是由一个3 3卷积层和两个1 1卷积层实现的，用于分类和回归，我们将其称为网络*头*。物体/非物体标准和边界盒回归目标是相对于一组称为*锚*的参考盒进行的[29]。锚点具有多种预定的比例和长宽比，以涵盖不同形状的物体。

我们通过用我们的FPN取代单尺度特征图来调整RPN。我们在特征金字塔的每一层都附加一个相同设计的头部（3 3 conv和两个同级的1 1 conv）。由于头部在金字塔所有层级的所有位置上密集滑动，因此不需要多尺度的

在一个特定层次上的锚。相反，我们分配给锚的每个级别都有一个单一的刻度。形式上，我们将锚点定义为在P_2、P_3、P_4、P_5、P_6上分别具有32^2、64^2、128^2、256^2、512^2像素的区域。[1]与[29]一样，我们也在每一级使用1:2、1:1、2:1的多个长宽比的锚点。因此，在金字塔上总共有15个锚点。

我们按照[29]中的方法，根据锚点与地面实况框的交集比（IoU）为其分配训练标签。从形式上看，如果一个锚对于给定的地面实况框具有最高的IoU，或者与任何地面实况框的IoU超过0.7，则被分配为正标签；如果它对于所有地面实况框的IoU低于0.3，则被分配为负标签。请注意，地面实证方块的尺度并不明确用于将其分配到金字塔的各个层次；相反，地面实证方块与锚点相关，而锚点已经被分配到金字塔的各个层次。因此，除了[29]中的规则外，我们没有引入额外的规则。我们注意到，头的参数在所有特征金字塔层中是共享的；我们也评估了不共享参数的替代方案，并观察到类似的准确性。共享参数的良好表现表明，我们的金字塔的所有层次都有类似的语义层次。这一优势类似于使用一个特征金字塔，一个共同的头部分类器可以可以应用于在任何图像比例下计算的特征。

通过上述调整，RPN可以自然地与我们的FPN进行训练和测试，其方式与[29]相同。我们在实验中详细说明了实施细节。

4.2. 用于快速R-CNN的特征金字塔网络

快速R-CNN[11]是一种基于区域的物体检测器，其中兴趣区域（RoI）集合被用来提取特征。快速R-CNN最常用于单尺度特征图。为了将其用于我们的FPN，我们需要将不同尺度的RoI分配给金字塔层。

我们把我们的特征金字塔看作是由图像金字塔产生的。因此，我们可以调整基于区域的检测器[15, 11]的分配策略，以适应它们在图像金字塔上运行的情况。从形式上看，我们将一个宽度为w、高度为h的RoI（在网络的输入图像上）分配给我们的特征金字塔的级别P_k：

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor \quad (1)$$

这里224是典型的ImageNet预训练规模，而k_0是一个具有$w\,h = 224^2$的RoI应该被映射到的目标层次。与基于ResNet的Faster R-CNN系统[16]类似，该系统使用C_4作为单尺度特征图，我们将k_0设为4。直观地说，公式（1）意味着如果RoI的规模变小（例如224的1/2），它应该被映射到一个更精细的分辨率级别（例如，$k = 3$）。

[1] 这里我们引入P_6，只是为了覆盖更大的锚点规模512^2。P_6仅仅是P_5的两个步长的子采样。P_6不被下一节中的快速R-CNN检测器使用。

我们将预测器头（在Fast R-CNN中，头是特定类别的分类器和边界盒回归器）附加到所有级别的RoI上。同样，这些头都是共享参数的，与它们的级别无关。在[16]中，ResNet的conv5层（一个9层的深度子网络）被作为conv4特征之上的头部，但我们的方法已经使用conv5来构建特征金字塔。因此，与[16]不同的是，我们只是采用RoI池提取7个特征，并在最后的分类层和边界盒回归层之前附加两个隐藏的1,024-d全连接（*fc*）层（每个层后都有ReLU）。这些层是随机初始化的，因为ResNets中没有预训练的*fc*层。请注意，与标准的conv5头相比，我们的*2*-fc MLP头重量更轻，速度更快。

基于这些调整，我们可以在特征金字塔的顶部训练和测试快速R-CNN。实施细节在实验部分给出。

5. 关于物体探测的实验

我们在80个类别的COCO de- tection数据集[21]上进行了实验。我们使用8万张训练图像和3.5万张估值图像子集（`trainval35k`[2]）进行训练，并在5千张估值图像子集（`minival`）上报告消融情况。我们还报告了在标准测试集（`test-std`）[21]上的最终结果，该测试集没有披露标签。

按照惯例[12]，所有的网络骨干都在ImageNet1k分类集[33]上进行了预训练，然后在检测数据集中进行了微调。我们使用公开的预训练的ResNet-50和ResNet-101模型。[2]我们的代码是使用Caffe2对`py-faster-rcnn`进行的重新实现。[3]的重新实现，使用Caffe2。[4]

5.1. 使用RPN的区域提案

我们评估了COCO式的平均召回率（AR）和对小、中、大物体的AR（AR_s , AR_m , and AR_l ）按照[21]中的定义。我们报告了每个图像100和1000个提案的结果（AR^{100} 和 AR^{1k} ）。

实施细节。表1中的所有架构都是端到端的训练。输入的图像被调整大小，使其短边有800像素。我们在8个GPU上采用同步的SGD训练。一个小型批次涉及每个GPU的2张图像和每个图像的256个锚。我们使用0.0001的权重衰减和0.9的动量。前3万个迷你批次的学习率为0.02，后1万个为0.002。对于所有的RPN实验（包括基线），我们将图像外的锚定框纳入训练，这与[29]不同，在那里这些锚定框被忽略了。其他实施细节与[29]相同。在8个GPU上用FPN训练RPN，在COCO上大约需要8小时。

[2] https://github.com/kaiminghe/deep-residual-networks [3] https://github.com/rbgirshick/py-faster-rcnn
[4] https://github.com/caffe2/caffe2

5.1.1 消融实验

与基线的比较。 为了公平地与原始的RPNs [29]，我们运行了两条基线（表1（a，b））），使用C_4（与[16]相同）或C_5的单尺度图，两者都使用与我们相同的超参数，包括使用32^2、64^2、128^2、256^2、512^2五个尺度锚点。表1(b)显示与(a)相比没有优势，表明单一的高阶级别的特征图是不够的，因为在更粗的分辨率和更强的语义之间存在着权衡。

将FPN置于RPN中，将AR^{1k}，提高到56.3（表1）。(c))，比单一尺度的RPN增加了**8.0**分。表1(a)）。此外，在小物体上的表现(AR^{1k})得到了12.9分的大幅提升。我们的金字塔表示法极大地提高了RPN的鲁棒性------。
对物体尺度变化的重要性。

自上而下的富集有多重要？表1(d)显示了我们的特征金字塔的结果，没有自上而下的途径。通过这种修改，1 1个横向连接和3 3个卷积被连接到自下而上的金字塔。这种结构模拟了重复使用金字塔特征层次的效果（图1（b））。

表1(d)中的结果只是与RPN基线持平，远远落后于我们的结果。我们猜测这是因为自下而上的金字塔（图1(b)）的不同层次之间存在很大的语义差距，特别是对于非常深的ResNets。我们还评估了不共享头的参数的Ta- ble 1(d)的一个变体，但观察到类似的性能下降。这个问题不能简单地通过特定级别的头来补救。

横向连接有多重要？表1(e)显示了一个没有11个侧向连接的自上而下的特征金字塔的消融结果。这个自上而下的金字塔有很强的语义特征和精细的分辨率。但我们认为，这些特征的位置并不精确，因为这些地图已经被下采样和上采样多次了。更精确的特征位置可以从自下而上地图的更精细层次中通过
的侧向连接到自上而下的地图。结果，FPN的AR^{1k}，比表1（e）高10分。

金字塔表征有多重要？与其求助于金字塔表征，不如将头部附在分辨率最高、语义强烈的特征上
P_2（*即*我们金字塔中最细的一层）的地图。Simi-
与单尺度基线相比，我们将所有锚点分配给P_2特征图。这个变体（表1（f））比基线好，但比我们的方法差。RPN是一个滑动的赢

缩小的检测器有固定的窗口大小，所以扫描时要注意金字塔级别可以增加其对规模变化的稳健性。此外，我们注意到，单独使用P_2，会导致更多的锚点（750k，表1(f)），这是由其大的空间重构引起的。谨慎。这一结果表明，更多的锚点本身并不足以提高准确性。 { }

s

×

×

×

RPN	特征	# 锚	侧面?	自上而下?	AR100	AR^{1k}	AR^{1k}_s	AR^{1k}_m	AR^{1k}_l
(a) conv4的基线	C_4	47k			36.1	48.3	32.0	58.7	62.2
(b) conv5的基线	C_5	12k			36.3	44.9	25.3	55.5	64.2
(c)**FPN**	$\{P_k\}$	200k	C	C	**44.0**	**56.3**	**44.9**	**63.4**	66.2
消融实验随后进行:									
(d) 自下而上的金字塔	$\{P_k\}$	200k	C		37.4	49.5	30.5	59.9	**68.0**
(e) 自上而下的金字塔，没有横向的	$\{P_k\}$	200k		C	34.5	46.1	26.5	57.4	64.7
(f) 只有最好的级别	P_2	750k	C	C	38.4	51.3	35.1	59.7	67.6

表1.使用RPN[29]的边界盒提议结果，在COCO迷你集上进行了评估。所有模型都是在`trainval35k`上训练的。其中 "横向 "和 "自上而下 "两栏分别表示存在横向和自上而下的连接。"特征 "一栏表示连接头的特征图。所有的结果都是基于ResNet-50的，并且共享相同的超参数。

快速R-CNN	建议	特征	头	侧面?	自上而下?	AP@0.5	美联社	AP_s	AP_m	美联社$_l$
(a) 关于定罪4的基线	RPN, $\{P_k\}$	C_4	conv5			54.7	31.9	15.7	36.5	45.5
(b) 关于定罪的基线5	RPN, $\{P_k\}$	C_5	*2fc*			52.9	28.8	11.9	32.4	43.4
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	*2fc*	C	C	**56.9**	**33.9**	**17.8**	**37.7**	**45.8**
消融实验随后进行:										
(d) 自下而上的金字塔	RPN, $\{P_k\}$	$\{P_k\}$	*2fc*	C		44.9	24.9	10.9	24.4	38.5
(e) 自上而下的金字塔，没有横向的	RPN, $\{P_k\}$	$\{P_k\}$	*2fc*		C	54.0	31.3	13.3	35.2	45.3
(f) 只有最好的水平	RPN, $\{P_k\}$	P_2	*2fc*	C	C	56.3	33.4	17.3	37.3	45.6

表2.使用**快速R-CNN**[11]对*一组固定的提议*（RPN, P_k, 表1(c)）进行的物体检测结果，在COCO上进行了评估。`minival`集。模型是在`trainval35k`集上训练的。所有的结果都是基于ResNet-50的，并且共享相同的超参数。

更快的R-CNN	建议	特征	头	横向?	自上而下?	AP@0.5	美联社	AP_s	AP_m	美联社$_l$
(*) 基线来自He*等人*[16]†。	RPN, C_4	C_4	conv5			47.3	26.3	-	-	-
(a) conv4的基线	RPN, C_4	C_4	conv5			53.1	31.6	13.2	35.6	**47.1**
(b) conv5的基线	RPN, C_5	C_5	*2fc*			51.7	28.0	9.6	31.9	43.1
(c)**FPN**	RPN, $\{P_k\}$	$\{P_k\}$	*2fc*	C	C	**56.9**	**33.9**	**17.8**	**37.7**	45.8

表3.使用**Faster R-CNN**[29]对COCO `minival` set进行评估的物体检测结果。*RPN的骨干网络与Fast R-CNN一致。*模型是在`trainval35k`集上训练的，并使用ResNet-50。†由[16]的作者提供。

5.2. 用快速/更快的R-CNN进行物体检测

接下来我们研究基于区域（非滑动窗口）的检测器的FPN。我们通过COCO式的平均精度（AP）和PASCAL式的AP（在0.5的单一IoU阈值下）评估物体检测。我们还按照[21]中的定义，对小、中、大尺寸的物体（即AP_s，AP_m，AP_l）报告了COCO AP。

实施细节。输入的图像被调整大小，使其短边有800像素。同步SGD被用来在8个GPU上训练模型。每个迷你批次在每个GPU上有2个图像，每个图像有512个RoI。我们使用0.0001的权重衰减和0.9的动量。前6万个小批次的学习率为0.02，接下来的2万个小批次为0.002。我们用每幅图像2000个RoI进行训练，1000个用于测试。在COCO数据集上，用FPN训练快速R-CNN大约需要10个小时。

5.2.1 快速的R-CNN（关于固定的建议）。

为了更好地研究FPN对基于区域的检测器的影响，我们对*一组固定的提案进行了*快速R-CNN的消融。我们选择将这些提案冻结在一个固定的区域内。

表1(c))，因为它在检测器要识别的小物体上具有良好的性能。为了简单起见，我们不在快速R-CNN和RPN之间共享特征，除非有特别说明。

作为基于ResNet的快速R-CNN基线，按照[16]，我们采用输出大小为14 14的RoI池，并将所有的conv5层作为头部的隐藏层。这使得表2(a)中的AP为31.9。表2(b)是一个基准线，利用一个具有2个隐藏*fc*层的MLP头，类似于我们架构中的头。它得到的AP为28.8，表明*2*-fc头没有给我们带来任何比表2(a)中的基线的正交优势。

表2(c)显示了我们在快速R-CNN中的FPN的结果。与表2(a)中的基线相比，我们的方法使AP提高了2.0分，小物体AP提高了2.1分。与同样采用*2fc*头的基线（表2（b））相比，我们的方法提高了5.1分的AP。[5]这些比较表明，对于*基于区域的*物体检测器，我们的特征金字塔优于单尺度特征。

表2(d)和(e)显示，去掉了自上而下的con-...

[5] 我们期望一个更强大的头部结构[30]将改善我们的结果，这超出了本文的重点。

方法	骨干力量	竞争	形象金字塔	测试-开发					测试-STD				
				AP@.5	美联社	APs	ĀP̄Ā	美联社,	AP@.5	美联社	APs	ĀP̄Ā	美联社,
我们的，在**FPN**上更快的R-CNN	里斯网-101	-		**59.1**	**36.2**	**18.2**	**39.0**	48.2	**58.5**	**35.8**	**17.5**	**38.7**	47.8
*赢得比赛的**单一模型**结果如下：*													
G-RMI†	初始化-ResNet	2016		-	34.7	-	-	-	-	-	-	-	-
AttractioNet‡ [10]	VGG16+宽幅ResNet§	2016	C	53.4	35.7	15.6	38.0	**52.7**	52.9	35.3	14.7	37.6	**51.9**
更快的R-CNN +++ [16]	里斯网-101	2015	C	55.7	34.9	15.6	38.7	50.9	-	-	-	-	-
多路径[40]（关于minival）。	VGG-16	2015		49.6	31.5	-	-	-	-	-	-	-	-
ION‡ [2]	VGG-16	2015		53.4	31.2	12.8	32.9	45.2	52.9	30.7	11.8	32.8	44.8

表4.COCO检测基准的**单一模型**结果的比较。有些结果在test-std集上没有，所以我们也包括test-dev的结果（以及minival上的Multipath [40]）。†：http://image-net.org/challenges/ talks/2016/GRMI-COCO-slidedeck.pdf。‡：http://mscoco.org/dataset/#detections-leaderboard。§：AttractioNet [10] 的这个条目采用VGG-16进行提议，Wide ResNet [39] 进行对象检测，所以严格来说不是单模型结果。

分享功能?	里斯网-50		里斯网-101	
	AP@0.5	美联社	AP@0.5	美联社
不	56.9	33.9	58.0	35.0

与我们在上面关于RPN的小节中观察到的情况类似，删除横向连接会导致较差的结果。值得注意的是，去掉自上而下的连接（表2(d)）会显著降低RPN的准确性。

这表明快速R-CNN因在高分辨率地图上使用低水平特征而受到影响。

在表2(f)中，我们在P_2的单一最细尺度特征图上采用了快速R-CNN。其结果（33.4AP）比使用所有金字塔级别的结果（33.9AP，Ta-）略差。

详见图2(c))。我们认为这是因为RoI pooling是一个类似翘曲的操作，对re-gion的尺度不太敏感。尽管这个变体有很好的准确性，但它是基于P_k的RPN建议，因此已经从金字塔表示法中受益。

5.2.2 更快的R-CNN（关于一致的建议）。

在上面的例子中，我们使用了一组固定的建议来对检测器进行检测。但是在Faster R-CNN系统中[29]，RPN和Fast R-CNN必须使用*相同的网络骨干，以*使特征共享成为可能。表3显示了我们的方法和两个基线之间的比较，它们都使用了*一致的*RPN和快速R-CNN的骨干架构。表3(a)显示了我们对[16]中描述的基线Faster R-CNN系统的再现。在受控设置下，我们的FPN（表3（c））比这个强大的基线好**2.3**分AP和**3.8**分AP@0.5。

请注意，表3(a)和(b)是基线，比Ta-ble 3(*)中He等人[16]提供的基线强很多。我们发现以下实现方式导致了差距：（i）我们使用800像素的图像比例，而不是[11, 16]中的600像素；（ii）我们用每幅图像512个RoI进行训练，这加速了收敛，而[11, 16]中只有64个RoI；(iii) 我们使用5个比例锚，而不是[16]中的4个（增加了32个²）；(iv) 在测试时，我们在每幅图像上使用1000个提议，而不是[16]中的300个。因此，与He等人的ResNet-50个更快的R-CNN基线在表3(*)中，我们的方法将AP提高了7.6分，AP@0.5 提高了9.6分。

共享特征。在上面的例子中，为了简单起见，我们不在RPN和Fast R-CNN之间分享特征。在Ta-

表5.使用Faster R-CNN和我们的FPN的更多物体检测结果，在minival上进行了评估。共享特征使训练时间增加1.5*倍*（使用4步训练[29]），但减少了测试时间。

在第5章中，我们按照[29]中描述的4步训练法评估了共享特征。与[29]类似，我们发现共享特征能以很小的幅度提高准确性。特征共享也减少了测试时间。

运行时间。通过特征共享，我们基于FPN的Faster R-CNN系统在单个NVIDIA M40 GPU上对ResNet-50的推理时间为每幅图像0.148秒，对ResNet-101为0.172秒。[6]作为比较，表3(a)中的单规模ResNet-50基线运行时间为0.32秒。我们的方法在FPN中引入了少量的额外成本，但有一个较轻的重量头。总的来说，我们的系统比基于ResNet的Faster R-CNN对应的系统要快。我们相信我们的方法的效率和简单性将有利于未来的研究和应用。

5.2.3 与COCO竞赛获奖者的比较

我们发现，表5中的ResNet-101模型在默认的学习率安排下训练得不够充分。所以我们在训练快速R-CNN步骤时，在每个学习速率下增加了2个小批的数量。这样，在不共享特征的情况下，minival上的AP增加到35.6。这个模型是我们提交给COCO检测排行榜的模型，如表4所示。由于时间有限，我们没有对其特征共享版本进行评估，正如表5所暗示的那样，它应该略胜一筹。

表4将我们的方法与COCO竞赛获奖者的*单模型*再结果进行了比较，包括2016年的冠军G-RMI和2015年的冠军Faster R-CNN+++。*在没有添加任何花哨的东西的情况下*，我们的单模型作品已经超过了这些强大的、经过大量设计的竞争对手。

[6] 这些运行时间是从本文的早期版本中更新的。

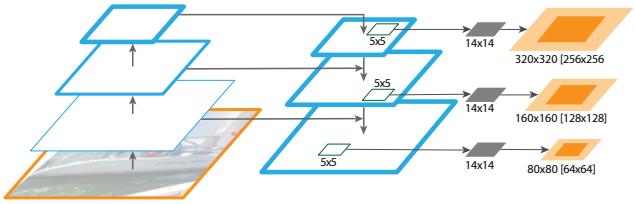


图4.对象段建议的FPN。特征金字塔的构造与物体检测的结构相同。我们在5 5个窗口上应用一个小型MLP，生成输出维度为14 14的密集的物体分割。橙色显示的是遮罩对应于每个pyra-mid级别（这里显示的是$P3$-$_5$级别）的图像区域的大小。相应的图像区域大小（浅橙色）和典型的物体大小（深色
橙色）所示。半个八度是由7x7窗口的MLP处理的（7≈5 2），这里没有显示。详情见附录。

在测试开发集上，我们的方法比以前的最佳结果增加了**0.5**个点的AP（36.2*对*35.7）和
3.4分的AP@0.5（59.1*对*55.7）。值得注意的是，我们的方法不依赖图像金字塔，只使用单一的输入图像比例，但在小尺寸物体上仍有突出的AP。这只有在以前的方法中通过高分辨率的图像输入才能实现。

此外，我们的方法*没有*利用许多流行的改进方法，如迭代回归[9]、硬否定挖掘[35]、上下文建模[16]、更强大的数据增强[22]等。这些改进是对FPN的补充，应该能进一步提高准确性。

最近，FPN在COCO竞赛的*所有*赛道上都取得了新的顶尖成绩，包括检测、实例分割和关键点估计。详见[14]。

6. 延伸：分割建议

我们的方法是一种通用的金字塔表示法，可以用于物体检测以外的应用。在本节中，我们使用FPN来生成分割建议，遵循DeepMask/SharpMask框架[27, 28]。

DeepMask/SharpMask是在图像作物上训练的，用于预测实例段和物体/非物体的分数。在推理时，这些模型以卷积方式运行，在图像中生成密集的建议。为了生成多尺度的片段，图像金字塔是必要的[27, 28]。

调整FPN以产生掩码建议是很容易的。我们在训练和推断中都使用了完全卷积的设置。我们按照第5.1节的规定构建特征金字塔，并设置d=128。在特征金字塔的每一层之上，我们应用一个小型的5 5 MLP来预测14 14个面具和物体。

在一个完全卷积的方式下，见图4。另外，受[27, 28]图像金字塔中每个八度空间使用2个刻度的启发，我们使用第二个输入大小为7 7的MLP来处理半八度空间。这两个MLP在RPN中起到了类似于锚的作用。该架构是端到端的训练；完整的实施细节在附录中给出。

	图像金字塔	AR	AR_s	AR_m	AR_l	时间 (s)
DeepMask [27]	C	37.1	15.8	50.1	54.9	0.49
锐利面具[28]	C	39.8	17.4	53.1	59.1	0.77
InstanceFCN [4]	C	39.2	-	-	-	1.5†
FPN掩码结果：						
单个MLP[5×5]		43.4	32.5	49.2	53.7	**0.15**
单个MLP[7×7]		43.5	30.0	49.6	57.8	0.19
双MLP[5×5，7×7]		45.7	31.9	51.5	60.8	0.24
2倍的掩膜分辨率		46.5	34.7	52.3	62.2	0.40

表6.在前5千张COCO val图像上评估的实例分割建议。所有模型都是在训练集上训练的。DeepMask、SharpMask和FPN使用ResNet-50，而Instance- FCN使用VGG-16。DeepMask和SharpMask的性能是用来自https://github.com/facebookresearch/deepmask的模型计算的（两者都是 "缩放 "变体）。†运行时间是在NVIDIA M40 GPU上测量的，除了InstanceFCN的时间是基于较慢的K40。

6.1. 分段建议的结果

结果显示在表6中。我们报告了小、中、大物体上的分段AR和分段AR，总是针对1000个提案。我们的基线FPN模型有一个单一的
5 5 MLP实现了43.4的AR。换成一个稍大的7 7 MLP，准确率基本没有变化。同时使用这两个MLP会使准确率提高到45.7 AR。将掩码输出大小从14 14增加到28 28，使AR又增加了一个点（更大的大小开始降低准确性）。最后，将训练迭代次数增加一倍，使AR增加到48.1。我们还报告了与DeepMask[27]、Sharp- Mask[28]和InstanceFCN[4]的比较，这些都是以前在掩码建议生成方面的最先进方法。我们比这些方法的准确率高出**8.3**个点AR。特别是，我们在小物体上的精确度几乎翻了一番。

现有的掩码建议方法[27, 28, 4]是基于密集采样的图像金字塔（*例如，*在[27, 28]中按2的比例$\{-2:0.5:1\}$），使其计算成本很高。我们的
基于FPN的方法，速度要快得多（我们的模型以6到7FPS运行）。这些结果表明，我们的模型是一个通用的特征提取器，可以取代图像金字塔用于其他多尺度检测问题。

7. 总结

我们提出了一个简洁的框架，用于在ConvNets内构建特征金字塔。我们的方法比几个强大的基线和竞争优胜者都有明显的改进。因此，它为特征金字塔的研究和应用提供了一个实用的解决方案，而不需要计算图像金字塔。最后，我们的研究表明，尽管深度ConvNets具有强大的表示能力和对尺度变化的隐性鲁棒性，但使用金字塔表示明确地解决多尺度问题仍然是至关重要的。

A.分类建议的实施

我们使用我们的特征金字塔网络来有效地生成物体段建议，采用流行于物体检测的以图像为中心的训练策略[11, 29]。我们的FPN掩码生成模型继承了DeepMask/SharpMask[27, 28]的许多想法和动机。然而，与这些在图像裁剪上进行训练并使用密集采样的图像金字塔进行推理的模型相比，我们在特征金字塔上进行完全卷积训练来预测面具。虽然这需要改变许多细节，但我们的实现在精神上仍然与DeepMask相似。具体来说，为了定义每个滑动窗口的遮罩实例的标签，我们认为这个窗口是对输入图像的裁剪，使我们能够从DeepMask中继承正/负的定义。接下来我们会给出更多的细节，也请看图4的可视化。

我们使用与第5.1节所述相同的架构，用P_{2-6}，构建特征金字塔。我们设定d=128。我们的特征金字塔的每一层都被用来预测
在不同的比例下的蒙版。与DeepMask一样，我们将遮罩的比例定义为其宽度和高度的最大值。
比例为32、64、128、256、512像素的面具分别映射到P_2、P_3、P_4、P_5、P_6，并由一个处理。
5×5 MLP。由于DeepMask使用的是半八度的金字塔，我们使用了第二个稍大的7 7（7 5 2）的MLP来处理我们模型中的半八度（*例如*，在P_4上的7 7 MLP预测了一个128 2比例的面具）。处于间歇期的物体
中间的尺度被映射到对数空间中最接近的尺度。

由于MLP必须预测每个金字塔级别的音阶范围内的物体（特别是半个八度的范围），所以必须在典型物体的大小周围给予一些填充。我们使用25%的填充。这意味着，掩码输出超过
$\{P_2, P_3, P_4, P_5, P_6\}$映射到{40, 80, 160, 320, 640}的大小。

5×5的MLP的图像区域（以及对7×7的MLP的对应尺寸）。2个较大的相关

特征图中的每个空间位置都被用于预
在不同的位置上决定一个面具。具体来说，在比例尺P_k，特征图中的每个空间位置都被用来预测其中心落在该位置2^k像素范围内的掩码。
阶段（对应于特征图中的1个单元偏移）。如果
没有任何物体中心在这个范围内，位置就会被确定。
在DeepMask中，它只用于训练分数分支，而不是掩码分支。

我们用来预测面具和分数的MLP是相当简单的。我们应用一个有512个输出的5.5核，然后由同胞全连接层来预测14 14的分数。
掩码（14个^2 输出）和物体得分（1个输出）。该模型是以完全卷积的方式实现的（使用1 1
用卷积来代替全连接层）。用于处理半倍频程的物体的7 7 MLP与5 5 MLP相同，只是其输入区域更大。

在训练过程中，我们对每个小批次的2048个例子进行随机抽样（每幅图像有128个例子，来自16幅图像）正/负抽样比例为1：3。掩码损失的权重比分数损失高10。这个模型在8个GPU上使用同步SGD进行端到端训练（每个GPU有2幅图像）。我们从0.03的学习率开始，训练了8万个小批次，在6万个小批次后将学习率除以10。在训练和测试期间，图像比例被设置为800像素（我们不使用比例抖动）。在推理过程中，我们的完全卷积模型预测所有位置和比例的得分，并在1000个最高得分的位置进行掩码。我们不进行任何非最大限度的抑制或后处理。

参考文献

[1] E.H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J.M. Ogden.图像处理中的金字塔方法。*RCA engineer*, 1984.

[2] S.Bell, C. L. Zitnick, K. Bala, and R. Girshick.内-外网：用跳过集合 和递归神经网络检测上下文中的物体。In *CVPR*, 2016.

[3] Z.Cai, Q. Fan, R. S. Feris, and N. Vasconcelos.一个统一的多尺度深度卷积神经网络用于快速检测物体 。In *ECCV*, 2016.

[4] J.Dai, K. He, Y. Li, S. Ren, and J. Sun.Instance-sensitive 全卷积网络。In *ECCV*, 2016.

[5] N.Dalal and B. Triggs.面向梯度的直方图用于 人类检测。In *CVPR*, 2005.

[6] P.Dolla´r, R. Appel, S. Belongie, and P. Perona.用于物体检测的快速特征 金字塔。*TPAMI*, 2014.

[7] P.F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan.基于判别性训练的部件的物体检测 模型。*TPAMI*, 2010.

[8] G. Ghiasi and C. C. Fowlkes.Laplacian pyramid reconstruction and refinement for semantic segmentation.In *ECCV*, 2016.

[9] S.Gidaris and N. Komodakis.通过一个多区域和语义分割感知的CNN模型
进行物体检测。

，其中有

ICCV, 2015.

[10] S.Gidaris and N. Komodakis.Attend refine repeat: Active box proposal generation via in-out localization.在*BMVC*，2016。

[11] R.Girshick.快速R-CNN.In *ICCV*, 2015.

[12] R.Girshick, J. Donahue, T. Darrell, and J. Malik.用于精确物体检测和语义 分割的丰富图像层次结构。In *CVPR*, 2014.

[13] B.Hariharan, P. Arbela´ez, R. Girshick, and J. Malik.用于物体分割和细粒度定位的超列 tion。在*CVPR*, 2015.

[14] K.He, G. Gkioxari, P. Dolla´r, and R. Girshick.掩码r-cnn。*arXiv:1703.06870*, 2017.

[15] K.He, X. Zhang, S. Ren, and J. Sun.用于视觉识别的深度卷积网络中的空间金字塔集合。在 *ECCV*。2014.

[16] K.He, X. Zhang, S. Ren, and J. Sun.用于图像识别的深度残差学习。In *CVPR*, 2016.

[17] S.Honari, J. Yosinski, P. Vincent, and C. Pal.重组者网络：学习从粗到细的特征聚合。在 *CVPR*, 2016。

[18] T.Kong, A. Yao, Y. Chen, and F. Sun.Hypernet：朝着accurate区域建议生成和联合物体检测的方向发展。In *CVPR*, 2016.

[19] A.Krizhevsky, I. Sutskever, and G. Hinton.用深度卷积神经网络进行ImageNet分类。在*NIPS*, 2012.

[20] Y.LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.应用于手写邮政编码识别的反向传播法。*Neural compu- tation*, 1989.

[21] T.-Y.Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, and C. L. Zitnick.Microsoft COCO: Common objects in context.在*ECCV*，2014年。

[22] W.Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed.SSD：单枪多盒检测器。在*ECCV*，2016年。

[23] W.Liu, A. Rabinovich, and A. C. Berg.ParseNet：，看得更清楚。在*ICLR研讨会上*，2016年。

[24] J.Long, E. Shelhamer, and T. Darrell.用于语义分割的完全卷积 网络。在*CVPR*, 2015.

[25] D.G. Lowe.Distinctive image features from scale-invariant keypoints.*IJCV*, 2004.

[26] A.Newell, K. Yang, and J. Deng.叠加沙漏网，用于人体姿势估计。In *ECCV*, 2016.

[27] P.O. Pinheiro, R. Collobert, and P. Dollar.学习分离 ment对象候选人。In *NIPS*, 2015.

[28] P.O. Pinheiro, T.-Y.Lin, R. Collobert, and P. Dolla´r.学习-完善物体段。在*ECCV*，2016年。

[29] S.Ren, K. He, R. Girshick, and J. Sun.更快的R-CNN：用区域建议网（ works）进行实时物体检测。In *NIPS*, 2015.

[30] S.Ren, K. He, R. Girshick, X. Zhang, and J. Sun.卷积特征图上的物体检测网络。*PAMI*, 2016.

[31] O.Ronneberger, P. Fischer, and T. Brox. U-Net：用于生物医学图像分割的Convolu- tional网络。在*MIC- CAI*，2015。

[32] H.Rowley, S. Baluja, and T. Kanade.视觉场景中的人脸识别。技术报告CMU-CS-95-158R， 卡耐基梅隆大学，1995。

[33] 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大规模视觉 识别挑战。*IJCV*, 2015.

[34] P.Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun.Overfeat：使用卷积网络的综合识别、定位 和检测。In *ICLR*, 2014.

[35] A.Shrivastava, A. Gupta, and R. Girshick.用在线硬例挖掘训练基于区域的物体检测器。在 *CVPR*, 2016。

[36] K.Simonyan and A. Zisserman.用于大规模图像识别的非常深入的卷积 网络。In *ICLR*, 2015.

[37] J.R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders.对象识别的选择性搜索。*IJCV*, 2013.

[38] R.Vaillant, C. Monrocq, and Y. LeCun.图像中物体定位的原始方法。*IEE Proc. on Vision, Image, and Signal Processing*, 1994.

[39] S.Zagoruyko and N. Komodakis.广义的剩余网络。In *BMVC*, 2016.

[40] S.Zagoruyko, A. Lerer, T. -Y.Lin, P. O. Pinheiro, S. Gross、
S.Chintala, and P. Dolla´r.一个用于物体检测的多路径网络。在*BMVC*，2016年。