Faster R-CNN论文翻译

原创 mengduanhonglou 2017-11-07 19:15:49 2341 收藏 1 版权

分类专栏: 人工智能 文章标签: 深度学习 神经网络 物体检测

首发地址
原文：SPPNet论文翻译
译者：邓范鑫

Faster R-CNN是互怼完了的好基友一起合作出来的巅峰之作，本文翻译的比例比较小，主要因为本paper是前述paper的一个简单改进，方法清晰，想法自然。什么想法？就是把那个一直明明应该换掉却一直被几位大神挤牙膏般地拖着不换的选择性搜索算法，即区域推荐算法。在Fast R-CNN的基础上将区域推荐换成了神经网络，而且这个神经网络和Fast R-CNN的卷积网络一起复用，大大缩短了计算时间。同时mAP又上了一个台阶，我早就说过了，他们一定是在挤牙膏。

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

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

摘要

最新的检测网络都依赖区域推荐算法来推测物体位置。像SPPnet[1]和Fast R-CNN[2]已经大幅削减了检测网络的时间开销，但区域推荐的计算却变成了瓶颈。本作将引入一个区域推荐网络（RPN）和检测网络共享全图像卷积特征，使得区域推荐的开销几近为0。一个RPN是一个全卷积网络技能预测物体的边框，同时也能对该位置进行物体打分。RPN通过端到端的训练可以产生高质量的推荐区域，然后再用Fast R-CNN进行检测。通过共享卷积特征，我们进一步整合RPN和Fast R-CNN到一个网络，用近期流行的“术语”说，就是一种“注意力”机制。RPN组件会告诉整合网络去看哪个部分。对于非常深的VGG-16模型[3]。我们的检测系统在GPU上达到了5fps的检测帧率（包括所有步骤），同时也在PASCAL VOC2007,2012和MS COCO数据集上达到了最好的物体检测精度，而对每张图片只推荐了300个区域。在ILSVRC和COCO 2015竞赛中，Faster R-CNN和RPN是多个赛道都赢得冠军的基础。代码已经公开。

1. 介绍

区域推荐方法(比如[4])和基于区域的卷及神经网络（RCNNs）[5]的成功推动了物体检测水平的进步。尽管RCNNs刚开发出来时[5]十分费时，经过[1][2]的跨推荐区域的共享卷积的改进，已经大幅消减了开销。近期大作Fast R-CNN[2]，如果不考虑区域推荐的耗时，使用超深度网络[3]已经达到几乎实时的处理速度。但推荐显然是最先进检测系统的瓶颈。区域推荐算法主要依赖简单的特征和经济的推理机制。最受欢迎的方法——选择性搜索[4]是基于低层次的人工特征贪婪地进行超级像素合并。而跟有效的检测网络[2]相比，选择性搜索的就慢了一个数量级，CPU上每张图片耗时2秒。EdgeBoxes[6]当前做到了速度和推荐质量的最佳平衡。然而，在整个检测网络中，区域推荐这一步仍然是主要耗时阶段。

你也许会注意到快速的基于推荐的CNNs充分利用了GPU，而区域推荐算法都是CPU中实现的。所以进行这个时间比较是不公平的。如果想加速它，用GPU实现就好了呀。这也许是个有效的工程化解决方案，但重新实现仍然会忽略下游的检测网络，因而也失去了共享计算的好机会。

本大作将向您展示一个算法上的改变——使用深度卷积神经网络计算推荐区域——将引出一个优雅而高效的解决方案，在给定检测网络完成的计算的基础上，让区域的计算近乎为0。鉴于此，我们向大家隆重介绍这个新型的区域推荐网络（Region Proposal Networks，RPNs），它和当今世界最棒的检测网络[1][2]共享卷积层。通过在测试阶段共享卷积，让计算推荐区域的边际成本变得很低（比如每张图片10ms）。

我们观察到像Fast R-CNN这样的基于区域的检测器锁使用的卷积特征图也可以用来生成推荐区域。在这些卷积层的特征之上，我们通过添加一些额外的卷积网络引入一个RPN，可以和回归约束框、物体打分相并列。RPN是一种完全卷积网络(FCN)[7]，可以为特定任务进行端到端的训练来产生检测推荐。

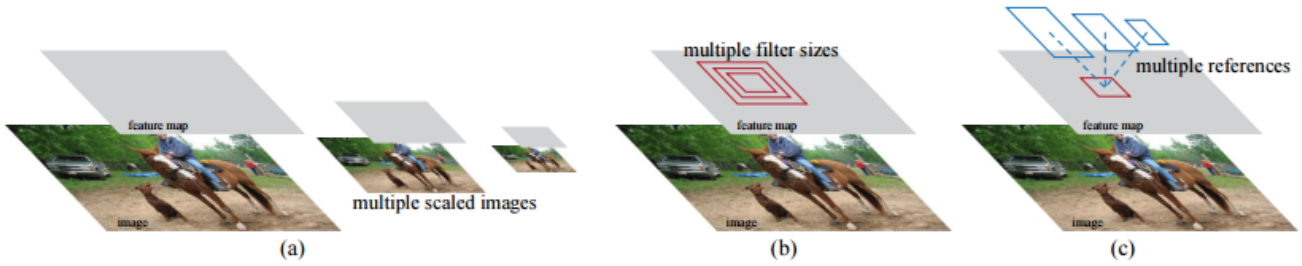

Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

RPNs被设计用来高效地预测各种尺度和宽高比的区域推荐。对称之前的[8][9][1][2]，他们均使用图像金字塔（图1，a）或特征的金字塔（图1，b），我们则使用“锚点”盒（“anchor” boxes）作为不同尺度和宽高比的参照物。我们的模式可以看做是一个回归参照物的金字塔（图1，c），这避免了穷举各种尺度和宽高比的图像或过滤器。这个模型在单一尺度图像的训练和测试时表现优异，因而运行速度大为受益。

为了统一RPNs和Fast R-CNN[2]物体检测网络，我们提出一种介于区域推荐任务调优和之后的物体检测调优之间的训练方法，同时还能保证固定的推荐。这个方法可以很快收敛，并产生一个统一的网络，该网络在两个任务上共享卷积特征。

我们在PASCAL VOC检测benchmarks[11]上全面评估了我们的方法，RPNs结合Fast R-CNNs可以比选择性搜索结合Fast R-CNN有更高的准确度。于此同时我们的方法摒弃了选择性搜索在测试阶段几乎所有的计算负担，有效推荐的运行时间只有区区的10毫秒。使用十分耗时的超深度模型[3]，我们的检测方法仍然可以在GPU上达到5fps的速度，这使得物体检测系统在速度和精度上都变得更加使用。我们也报告了在MS COCO数据集[12]上的结果，探究了PASCAL VOS上使用COCO数据集带来的提升。代码现在开放在 https://github.com/shaoqingren/faster_rcnn (in MATLAB) 和https://github.com/rbgirshick/py-faster-rcnn (in Python)。

本文的一个早期版本发布在[10]上。从那时起，RPN和Faster R-CNN的框架就已经被采用，并应用到其他的方法中，比如3D物体检测[13]，基于组件的检测[14]，实力分割[[13]和图像字幕[16]。我们的快速而有效的物体检测系统已经构建在想Pinterests[17]这样的商业系统中，提升了用户交互。

在ILSVRC和COCO 2015竞赛中，Faster R-CNN和RPN是多项分赛长的第一名[18]，包括ImageNet 检测，ImageNet定位，COCO检测和COCO分割。RPNs从数据中完全学会了推荐区域，而且使用更深或更有表达力的特征（比如101层的Resnet[18]）效果会更好。Faster R-CNN和RPN也用于多个其他领先名词的团队所使用。这些结果都说明我们的方法不仅实用省时，而且有效精准。

2 相关工作

物体推荐。有大量的推荐方法。有一些综述和这些方法的比较可见于[19], [20], [21]。广泛使用的方法很多基于grouping super-pixels (比如,Selective Search [4], CPMC [22], MCG [23])，还有一些基于滑动窗口(e.g., 比如窗口的物体属性objectness in windows[24], EdgeBoxes [6])。物体推荐方法也经常独立于它的检测器而被很多外部的模块使用 (比如, Selective Search [4] object detectors, RCNN [5], 和Fast R-CNN [2])。

用于物体检测的深度网络。R-CNN方法[5]端到端地训练CNNs，用于将推荐区域分类成物体类别或背景。R-CNN主要扮演了分类器的角色，它并不预测物体的边框（除了用于约束框回归的净化模块）。他的精度依赖于区域推荐模块的性能（见[20]中的比较）。多篇论文推荐是用深度网络预测物体约束框 [25], [9], [26], [27]。OverFeat方法中，一个全连接网络用于训练预测定位任务的单一物体的框坐标。为了检测多个特定类的物体又将全连接层转变成卷积层。MultiBox方法[26][27]也使用网络产生推荐，它的最后一个全连接层可以同时预测多个未知类的框，推广了OverFeat的“单框”风格。这些未知类方框也被R-CNN[5]所使用。MuiltiBox推荐网络应用于单张图片的一个裁切，或者一个大型图片的多个裁切（比如224×224），和我们的全卷积模式完全不同。MultiBox并不在推荐和检测网络之间共享特征。后面结合我们的方法，我们将深入讨论OverFeat和MultiBox。和我们的工作同时进行的DeepMask方法[28]也被开发出来用于语义推荐。

卷积计算的共享 [9], [1], [29],[7], [2]，已经越来越受关注。OverFeat[9]中针对分类、定位、检测时会只从一个图像金字塔计算卷积特征。尺寸自适应的SPP[1]也是建立在共享卷积特征图智商的，在基于区域的物体检测[1][30]和语义分割[29]上很有效。Fast R-CNN[2]使得端到端的检测器训练全部建立在共享卷积特征之上，表现出了有引人注目的精度和速度。

3 FASTER R-CNN

我们的物体检测系统，成为Faster R-CNN有两个模块组成。第一个模块是深度卷积网络用于生成推荐区域，第二个模块是Fast R-CNN检测器[2]，使用推荐的区域。整个系统是一个单一的统一的网络（图2）。使用近期流行的属于“注意力”[31]机制，RPN模块告知Fast R-CNN看向哪里。3.1节我们介绍网络的设计和特性。3.2节，我们开发算法用于训练模块和特征共享。

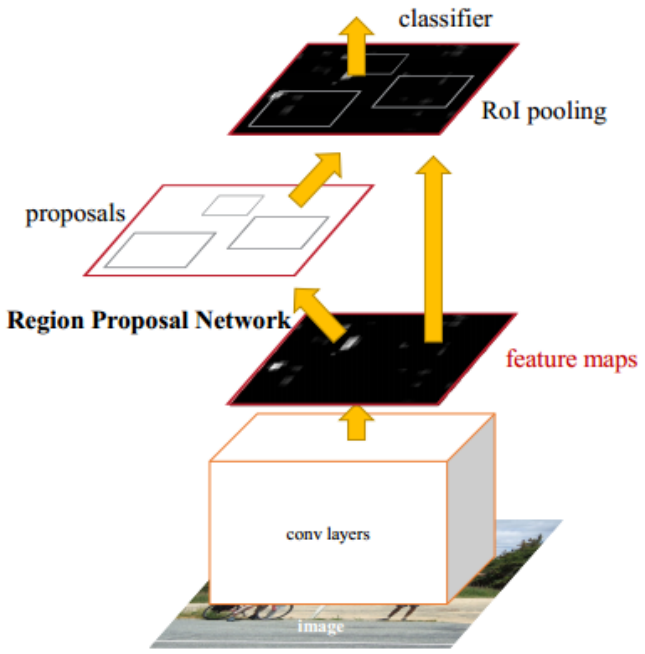

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

3.1 区域推荐网络

特征推荐网络接收任意尺寸的图像输入，输出一组矩形框代表物体推荐区域，每个区域都会有一个物体性的打分。我们使用完全卷积网络（FCN）[7]构建这个过程，本节将详细表述它。由于我们的终极目标是共享Fast R-CNN和物体检测网络[2]之间的计算力，我们假定两个网络可以共享一套卷积层。在实验中，我们研究了Zeiler和Fergus模型[32](ZF)，他们就共享了5个卷积层，还研究了Simonyan 和Zisserman模型[3] (VGG-16)，他们共享了13个卷积层。
为了产生区域推荐，我们用一个小网络在最后一个卷积层的卷积特征图上滑动。每个滑动窗口都映射到一个更加低维度的特征（对ZF使用256，对VGG使用512，后面跟一个ReLU[33]）。这个特征再喂给两个并列的全连接层，一个框回归层（reg）和一个框分类层（cls）。本文中，我们使用n=3,一个在大图片（对于ZF和VGG来说，分别是171和228像素）十分有效的感受野大小。这个迷你网络在单一位置的示意如图3（左）。注意，由于迷你网络以滑动窗口的方式进行操作，全连接层是在全部空间位置共享的。这个架构很自然就就实现成一个nxn的卷积网络跟两个1×1的卷积网络层（分别是reg和cls）。

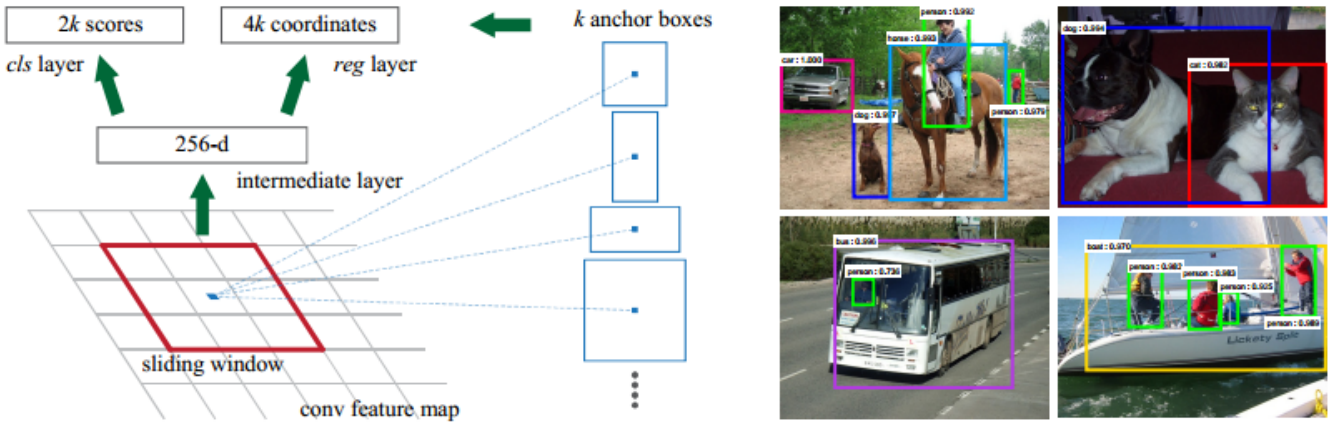

Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

3.1.1 锚点

在每个滑窗位置，我们同时预测多个区域推荐，每个位置的最大滑窗推荐数量定位为k。这样reg层就有4k的输出编码k个框的坐标，cls就有2k的预测对象还是非对象的概率的打分。k个推荐是针对k个参考框进行参数化的，这个参考框我们称之为锚点。一个锚点就是正在关注的滑窗的中心，并和缩放比例、宽高比想关联（图3左）（译者注：就是滑窗中心坐标、缩放比例、宽高比形成的三元组决定一个锚点）。默认我们使用3个缩放尺度和3个宽高比，在每个滑动位置产生k=9个锚点。对于一个WxH（通常是2400）大小的卷积特征图，总共有WHk个锚点。

平移不变性锚点

我们方法有一个重要特性就是平移不变性。无论是锚点还是相对锚点计算推荐的函数都有这个特性。如果在一涨图片上移动一个物体，推荐也应该平移并且相同的函数应该能够在新的位置也计算出推荐来。我们的方法可以保证这种平移不变性。作为对比，MultiBox方法[27]使用k-means产生了800个锚点，却不能保持平移不变性。因此MultiBox不能保证在物体平移后产生同样的推荐。
平移不变性可以缩减模型的大小。MultiBox有（4+1）x 800维的全链接输出层，而我们的方法只有（4+2）x9的卷积输出层，锚点数是k=9。结果，我们的输出层有2.8 x 10^4个参数（对于VGG-16而言是512 x （4 + 2）x 9），比MultiBox的输出层的6.1×10^6个参数（对GoogleNet[34]为1536x（4 + 1）x800）少了两个数量级。如果考虑特征映射层，我们的推荐层也还是少一个数量级。我们预期这个方法可以在PASCAL VOC这样的小数据集上有更小的过拟合风险。

多尺度锚点作为回归参照物

我们的锚点设计是解决多尺度问题的一种新颖形式。如图1所示，有两种流行的多尺度预测形式。第一种是基于图像/特征金字塔，也就是DPM[8]和基于CNN的方法[9][1][2]。图像被缩放到各种尺度，特征图(HOG[8]或深度卷积特征[9][1][2])也在每个尺度进行计算，参见图1（a）。这种方式通常很有用，但是很耗时。第二种方式是在特征图的多个尺度上使用滑窗。例如，在DPM[8]中，不同缩放比例的模型分开训练，使用了不同的过滤器尺寸（诸如5×7,7×5）。如果这种方式解决多尺度问题，可以看作是过滤器的金字塔，图1（b）。第二种方式通常和第一种方式联合使用[8]。作为比较，我们的基于锚点的方法是建立在锚点金字塔上的，是否高效。我们的方法使用不同尺度和不同宽高比的锚点作为参考分类和回归约束框。他之和单一尺度的图像和特征图有关，并且使用单一尺寸的过滤器，这些过滤器在特征图上进行滑动。我们通过实验显示了我们这个方法解决多尺度和多尺寸问题的效果（表8）。由于基于锚点的多尺度设计，我们可以和Fast R-CNN检测器[2]一样，只在单一尺度的图像上计算卷积特征。多尺度锚点的设计是不用额外计算开销共享特征解决多尺度问题的关键。

3.1.2 损失函数

为了训练RPNs，我们设计了针对每个锚点的二分类标签（是否是一个物体）。我们给两类锚点标记位正例：（i）和标注框最大重合的锚点 （ii）和任何标注框IoU重叠度超过0.7的。对于一个真实标注可能会产生多个正例锚点。通常第二类情况就足够确定正例了，但我们仍然采用第一类的原因是一些特别极端的案例里面没有正例。对于与标注框重叠度低于0.3的都标注为负例。既正且负的锚点对训练没有帮助。结合这些定义，我们参照Fast R-CNN中的多任务损失函数的定义我们的损失函数是：

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

【略】
对于约束框回归，我们对四个坐标参数化[5]：

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

【略】

3.1.3 训练RPNs

【略】

3.2 RPN and Fast R-CNN之间共享特征

【略】

3.3 实现细节

【由于faster r-cnn的设计十分简洁，后续的英文原文十分易懂，感兴趣的可以直接阅读原文了，略】

4 EXPERIMENTS

【略】

5 CONCLUSION

【略】