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

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

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

文献走读：

摘要：现如今较好的目标检测网络，依托于region proposal算法来假定目标的位置。
较好的有SPPnet[1]和Fast R-CNN方法，均大幅减少了检测的时间，也指出了region proposal 的计算是检测的瓶颈。

本文中，我们引入了**Region Proposal 网络(RPN)**，它与detection网络共享整幅图像的卷积特征，这使得region proposal成本几乎为0。RPN是一种完全卷积网络，它同时预测了目标的位置（bounds）并计算每一个位置的检测分数（成为目标的可能性）。RPN采用端到端的方式训练，以生成高质量的region proposal，Fast R-CNN正是用此来检测目标的。

进一步，我们将RPN和Fast R-CNN合并为一个网络——通过共享他们的卷积特征，换句近期很流行的术语来讲，即神经网络的"注意力"机制（attention mechanism）。RPN部分告诉整个网络往哪里“看”。

对于文献3中的超深VGG-16模型，我们的检测系统在GPU上达到了5fps的帧率，同时在PASCAL VOC2007，2012上实现了极佳的目标检测准确率，在MS COCO数据库上每张图片仅提出300个候选区域。在ILSVRC和COCO 2015竞赛中，冠军分别是使用Faster R-CNN和RPN来完成的。并且本方法代码公开了。

1 介绍

近期的目标检测的进步主要是基于region proposal方法和基于region的卷积神经网络方法的成功。尽管在最初的方法【5】中，基于region的CNN的计算代价很高，但得益于【1，2】中提出的对proposal区域的卷积共享，这部分代价大大减少了、最近的衍生方法中，忽略region proposal花费的时间后，Fast R-CNN使用很深的网络实现了实时检测。现如今，proposal成为各个顶级方法中的计算瓶颈。

Region proposal方法主要依托于廉价的特征和快速经济的推测方案。选择性搜索（Selective Search[4]）就是一种流行RP方法，它基于工程性的低水平特征进行了大量的像素合并。如今与【2】提出的高效检测网络相比，选择性搜索在CPU实现中，每张图片需要2s，就显得相当慢了。[6]中的EdgeBoxes(边缘boxes)方法，在proposal质量和速度之间给出了很好的平衡，达到了0.2s/image。尽管如此，region proposal step仍然在检测网络中耗费了大量的运行时间。

注意到基于region的快速CNN的方法利用到了GPU的优势，然而以往研究中的region proposal方法是CPU实现的，这显然造成了运行时间的不相容（proposal太慢）。一种明显的加速proposal计算的方法是使用GPU再次实现之。这或许是一个高效工程方案，但是重新师兄忽略了下游检测网络并且也因此漏掉了重要的机会进行共享计算。

本文中，我们进行了算法改进，我们使用深层的卷积神经网络来计算proposals、这是一种优雅且高效的解决方案，因为在已有的检测网络计算的基础上，proposal的计算几乎是不花费时间的。最后，我们引出了新颖的RPN,它与【1，2】中的目标检测网络共享卷积层。通过在test-time阶段共享卷积特征，计算proposals的边际成本变得很小。（10ms/image）

我们发现基于region的检测器（如Fast R-CNN）使用的卷积特征maps，也可以用于生成region proposals。在这些卷积特征的顶层，我们通过增加一点额外的卷积层来构造了一个RPN，这些卷积层对region bounds进行回归，同时对网格中每个位置给出目标分数。因此，RPN是一种完全卷积网络，可以进行端到端的训练，以生成检测的proposals。

RPN网络是为了对大范围的尺寸和宽高比的场景，进行高效的region proposals。与普通的使用图像金字塔或金字塔滤波器的方法相比，我们引入了新颖的**anchor boxes**。anchor boxes像是一种固定点，在多尺度和多宽高比时，它可以作为一种参照物。我们的方案可以看成是对回归参照的金字塔设计，这避免了对多尺度和多宽高比情况下的图像枚举或滤波器遍历。当使用单尺度图像进行训练和测试时，这种模型性能很好，两者相得益彰。

为了统一RPN和Fast R-CNN网络，我们提出了一种训练方案——时而进行对region proposal任务进行微调，或者在proposal固定时对目标检测任务微调。这种方案收敛很快，并且产生了一个统一的网络——在两个任务之间，卷积特征被共享了。

我们在PASCAL VOC上进行测试，本方案（RPN结合Fast R-CNN）比基线方案（Selective Search结合Fast R-CNN）准确率高很多。同时，我们的方法几乎释放了Selective Search方案在test-time阶段的所有计算负担——我们的proposals时间仅需10ms。使用【3】中非常庞大深层的模型时，我们在GPU模式下仍能达到5fps的速度，从速度和准确度角度来看，这是一个极具实用价值的目标检测系统。在MS COCO数据库上有评测结果，也改进了PASCAL VOC的结果。MATLAB和Python版本的实现参考github.

这种方案的手稿版曾发表于【10】，从那时起，RPN+Faster R-CNN的方案就被其他方案采用和推广，比如3D 目标检测【13】，基于部件的检测【14】，实例分割【15】，图像字幕【16】。我们的快速高效的目标检测系统也已经被商用了。

在ILSVRC和COCO2015竞赛中，Faster R-CNN和RPN是进入ImageNet检测，ImageNet定位，COCO检测和COCO分割比赛前茅者的方案的基础。RPN完全是从数据中学习proposal regions,因此可以很容易从深层和expressive 特征中获益（深度特征的品质比raw数据好），例如【18】中采用的101层的resudual 网络。
Faster R-CNN和RPN也被用在其他几个竞赛的优秀方案中。它们的结果均显示我们的方法不仅是一个实用的成本效益好的方案，也是一个提高目标检测准确度的高效方式。（性能有，准确度好。）

2 RELATED WORK相关工作

目标Proposal。在目标proposal 方法领域有很多文献。【19，20，21】对目标proposal方法进行了综述和对比。广泛使用的目标proposal方法包括：基于像素分组（如Selective Search【4】，PMC【22】,MCG）,以及基于滑窗的方法（【24】，EdgeBoxes【6】）在其他的检测器（【4】Selective Search目标检测器，【5】R-CNN和【2】Fast R-CNN）中，目标proposal方法是作为外部模块使用的。

用于目标检测的深度网络。【5】R-CNN方法训练了端到端的CNN网络，用来在目标类别和背景中对区域进行proposal归类。R-CNN主要扮演着分类器的角色，(除了进行边界框回归的精炼的做法之外)它没有预测目标的边界（bound）,因此这类网络的精度取决于region proposal模块的性能（参考【20】中的对比）。好几篇文章提出使用深度网络预测目标的边界框（【25，9，26，27】）。在【9】的OverFeat方法中，训练了一个全连接层来预测框的坐标，以处理单一场景下的定位任务。（只能进行单目标的定位）。这种全连接层随后就变成了卷积层，以解决多类指定目标的检测。【26，27】中的MultiBox方法，可以从一个网络中产生region proposal——该网络最后的全连接层同时预测多个未知类别的框，并产生Overfeat中的单类。这些未知类别的框用作【5】中R-CNN中的proposal。与我们的完全卷积方案有所不同，MultiBox proposal 网络应用在单一图块或者多个大图块（如224*224）上。MultiBox没有在proposal和检测网络之间共享特征。我们结合我们的方案，稍后深入讨论OverFeat和MultiBox。与我们同时开展的【28】DeepMask方法可以学习分段proposal(segmentation proposal)

3 FASTER R-CNN

我们的目标检测系统Faster R-CNN由两个模块组成。第一个模块是proposal region的深层全卷积网络。第二个模块是使用proposaled区域的Fast R-CNN检测器【2】。整个系统是一个单一的统一的目标检测网络。
用时下流行的“注意力”说法，RPN模块告诉Fast R-CNN模块“看”向哪里。在3.1节，我们介绍了区域proposal网络的设计和属性。在3.2节中，我们实现了同时训练两个共享特征模块的算法。

3.1 Region Proposal Networks

RPN网络输入一个图像（任何尺寸），输出目标proposal的矩形框集合，每一个框都带有一个目标分数。我们使用一个完全卷积网络来对此过程进行建模，本节主要阐述此内容。因为我们最终的目标是与Fast R-CNN目标检测网络共享计算（computation），所以我们假设两个网络共享一个共同的卷积层集合（多个全卷积层）。我们在实验中也研究了ZF模型【32】和VGG-16模型【3】，它们分别有5个和13个共享卷积层。

为了产生区域proposal,我们考虑最后一个共享卷积层的输出，在输出的卷积特征图上设计了一个滑动的小网络。这个小网络输入是卷积特征图的n*n的空间窗口。每一个滑动窗口都映射到一个低维的特征上（ZF是256d,VGG是512，并在后面加上ReLU【33】）。这个特征送入两个姊妹全连接层——框回归层（reg）和框分类层(cls)。注意到输入图片的感受野很大（ZF是171，VGG是228），本文中我们使用n=3.这个微型网络在图3中给出。注意到因为这个微型网络在滑窗上进行运算，那么全连接层在每个空间位置均共享。自然而然，我们将此结构设计为n*n的卷积层，并跟上两个1*1的姊妹卷积层（reg和cls）。

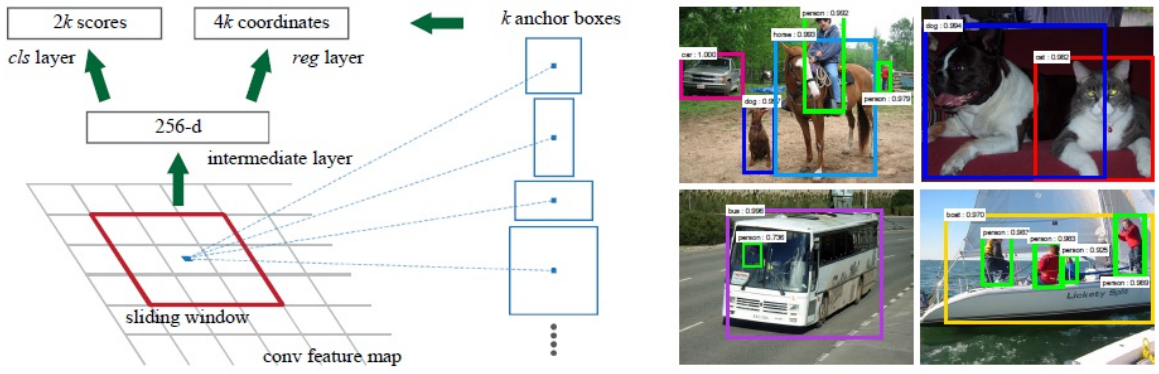

Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

3.1.1 Anchors

在每一个滑窗位置处，我们同时预测多个region proposal，将每个位置最大的可能的proposal的个数记为k。因此reg层有4k个参数输出（才能表示k个框的坐标），cls层有2k个参数输出，表示对每个proposal能否成为目标的可能性分数。这k个proposal(建议值)是对k个参照框的参数化，我们称之为anchor(锚点，参照点)。anchor是当前滑窗的中心，并且与尺寸比例和宽高比有关系。在每个滑动位置，我们使用3个尺寸比例和3个宽高比，产生k=9个锚点anchor。一般的卷积特征图的尺寸W*H（典型值为2400），就会有WHk个锚点。

Translation-Invariant Anchors（平移不变的anchor）
本方法是平移不变的，对锚点以及根据锚点计算proposal来讲，都具有平移不变性。
何谓平移不变性呢，比如将目标在图像中平移到另一个位置，那么所得到的proposal 区域也应该有相应的平移，并且同一个预测proposal的函数可以正确预测其位置。我们的方法5保证了这种平移不变性。作为比较，MultiBox方法【27】使用K-means方法产生800个anchor，他们的方法不具有平移不变性。所以MultiBox不保证当目标平移时能够产生相同的proposal。

这种平移不变性也降低了模型的大小。MB的全连接输出层是（4+1）*800维，然而我们的方法中，取anchor个数为k=9时，卷积输出层的维度为（4+2）*9。结果我们的输出层仅有2.8*10^4个参数（VGG-16有512*（4+2）*9个），比MB方法输出层参数少两个数量级，MB输出层参数为6.1*10^6个（【34】使用GooleNet的MB，1536 * (4 + 1) * 800）。如果仅考虑特征投影层，我们的proposal层的参数比MB少一个数量级。我们期望本方法在小数据集上过拟合的风险较小，比如PASCAL VOC。

Multi-Scale Anchors as Regression References
我们的anchor设计为解决多尺度多宽高比问题提供了一种新颖的方法。如图1所示，有两种流行的方法可用于多尺度的预测。第一种方式基于图像/特征 金字塔，比如【8】DPM和【9，1，2】基于CNN的方法。图像按照不同的尺寸比例进行缩放，每一种比例下都计算特征图（【8】是HOG,【9，1，2】是深度卷积特征），如图1a所示。然而这种方式比较有用，但是很耗时间。第二种方式是在特征图上使用多尺寸比例多宽高比的滑窗方法。比如，在【8】DPM中，不同宽高比的模型是使用不同滤波器大小（5*7和7*5）单独训练的。如果采用这种方式解决多尺寸比例问题，可以看出是一种“滤波器金字塔”，如图1b。第二种方式通常与第一种方法结合使用。

对比来看，我们基于anchor的方法建立在anchor金字塔的基础上，这是更加高效的方法。我们根据不同尺寸比例和不同宽高比情况下的anchor框，对边界框进行分类和回归。这仅仅依赖一个单一尺寸比例的图像和特征，且只需要使用单一尺寸的滤波器（即特征图上的滑窗）。我们通过实验证明了这种方案对解决多尺寸比例和大小问题的作用，如表8.

因为这个多尺度的设计是基于anchor的，我们可以简单地，就使用在单尺度图像上计算出的卷积特征，正如Fast R-CNN检测器【2】做的那样。这种多尺度anchor的设计，是在多尺度问题下，保证无需其他计算成本即可共享特征的关键因素。

3.1.2 Loss Function
训练RPN网络阶段，我们为每个anchor分配了一个二类标签（是目标或者不是目标）。以下两种anchor赋值为正样本标签：1，该anchor/anchors 与真实目标框具有高度重合；2，该anchor与任意真实目标框的重合度高于0.7。注意到，单一真实目标框可能给多个anchor都赋予了正标签。通常情况下，第二种情况已经足够进行判别了，但我们仍采用情况一，因为在某些罕见情况下，情况二也许找不到任何正样本标签（即使存在真实目标时）。对于那些与所有真实目标框的重合度均小于0.3的anchor，我们将其划分为负样本。那些既非正样本也非负样本的anchor样本，不参与训练，也不会造成影响。

按照Fast R-CNN【2】中多任务loss的目标函数，进行最小化。

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

i 是mini-batch的第i个anchor，
pi是预测的第i个anchor成为目标的可能性。
真实的标签pi*是1或者0，1表示正标签的anchor，0表示负标签的anchor。
ti是预测到的框的向量化坐标值，ti*是正样例anchor的真实框的坐标。

分类损失Lcls是在两个类别上的log 损失函数。
对于回归loss,我们使用Lreg(ti,ti*)=R(ti-ti*)来量化，其中R是【2】中定义的鲁棒loss函数（L1平滑）。并且给Lreg带上pi*因子，表示只有正标签的anchor（pi*=1）才会激活该损失。

cls层和reg层的输出分别由{pi}和{ti}组成。即cls层输出预测到的{pi}，reg层输出预测到的{ti}。