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

**1 引言**

SPPnet和Fast RCNN的提出，极大地降低了目标检测任务中对ROI区域的检测时间，但是区域建议计算仍然非常缓慢，越来越成为目标检测任务时间和性能的瓶颈。常用的区域建议算法例如selective search算法专注于将低级特征（例如纹理，颜色等）像素进行聚合，以此实现对原始图像的分割，提取ROI。但这一算法与诸如Fast RCNN这样有效的目标检测网络相比，慢了一个数量级。就算是像Edgeboxes这样对建议质量和速度实现极好平衡的区域建议算法，也需要消耗和目标检测相同的时间。

此外，当时常用的区域建议算法都是在CPU上进行实现的，而目标检测网络是在GPU上进行训练的，据我所知在15年那会儿GPU在大多数情况下要强于CPU，所以作者认为实现一个在GPU上训练的区域建议算法能够解决这一瓶颈问题。

基于上述想法，本文提出了新的区域提议网络（Region Proposal Network），基于全卷积神经网络架构，实现和目标检测网络的特征共享，从而实现几乎零成本实现ROI的提取。

为了将RPN和当前性能最好的一些目标检测网络（例如Fast-RCNN）结合，本文提出了一种训练方案，即交替进行两个阶段的微调，具体细节在正文中进行介绍。此外，使用了当时比较流行的“注意力机制”，让RPN告诉Fast-RCNN模块应该在哪里寻找，以提高目标检测的效率以及性能。

本文将这一新架构命名为Faster-RCNN，不仅在多项目标检测比赛中获得第一名，也应用到了Pinterest的商业系统中，提高了用户参与度。

**2 RPN**

**2.1 RPN介绍**

Faster-RCNN中最核心的部分就是区域建议网络，下面就从图2-1和2-2介绍具体的细节。

Faster-RCNN与Fast-RCNN最大的不同就是用区域建议网络RPN取代了selective search。在Faster-RCNN中，先将图像经过RPN进行特征提取，得到对应特征图，接下来的过程比较关键，即使用*anchor*提取候选框。在全卷积网络提取出特征图后，对特征图进行一个3\*3的卷积操作，3\*3的感受野中心我们称之为*anchor*，也就是“锚点”,加下来我们以锚点为中心，提取9个候选框，即anchor box，这9个框是由三种尺寸和三种宽高比组合而来的。这些anchor box就是我们需要进行目标检测的候选框。显然，这一操作会产生大量的候选框，并不是把所有的候选框都喂给目标检测网络进行预测，而是选取其中的256张作为目标检测网络的输入。首先，删去超过图像边界的候选框；接着计算候选框与真实边界框的IoU，将IoU大于0.7的设置为正样本，IoU小于0.3的设置为负样本（因为我们需要模型不仅能够检测出一张图像中有目标，也要检测出一张图像中没有目标），选取正样本IoU最大的128张，负样本随机选取128张，总共256张进行后续训练。

对特征图进行3\*3的卷积以及图像选取之后，可以看到线路一分为二，其中上方的路线是分类分支，经过一个1\*1的卷积用于通道数的调整，图中的“18”其实是2\*9，其中2表明是二分类问题，9是anchor box的数量。下方的路线是预测框回归分支，同样经过一个1\*1的卷积用于通道数的调整，图中的“36是4\*9，其中4是调整后的边界框，即(x,y,w,h)。

上述工作完成后，将提取到的边界框映射到原始图像上进行ROI池化，后续工作和Fast-RCNN相同。

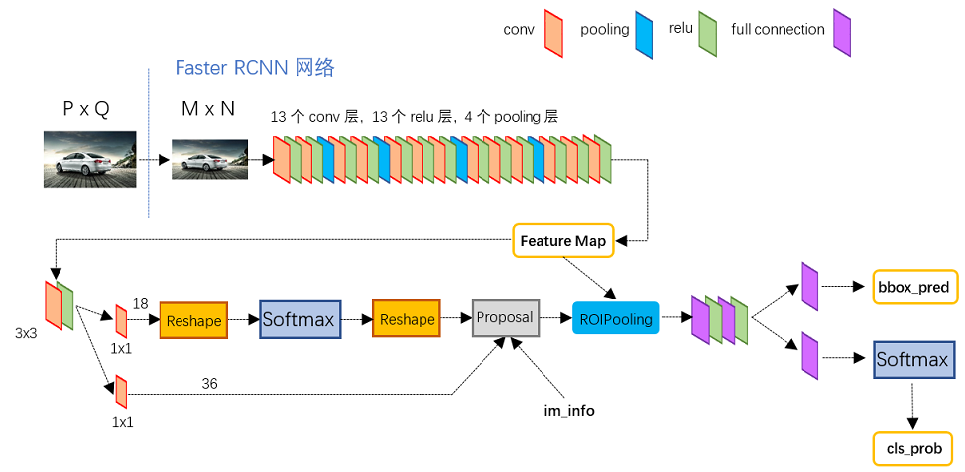


图2-1 Faster RCNN网络架构

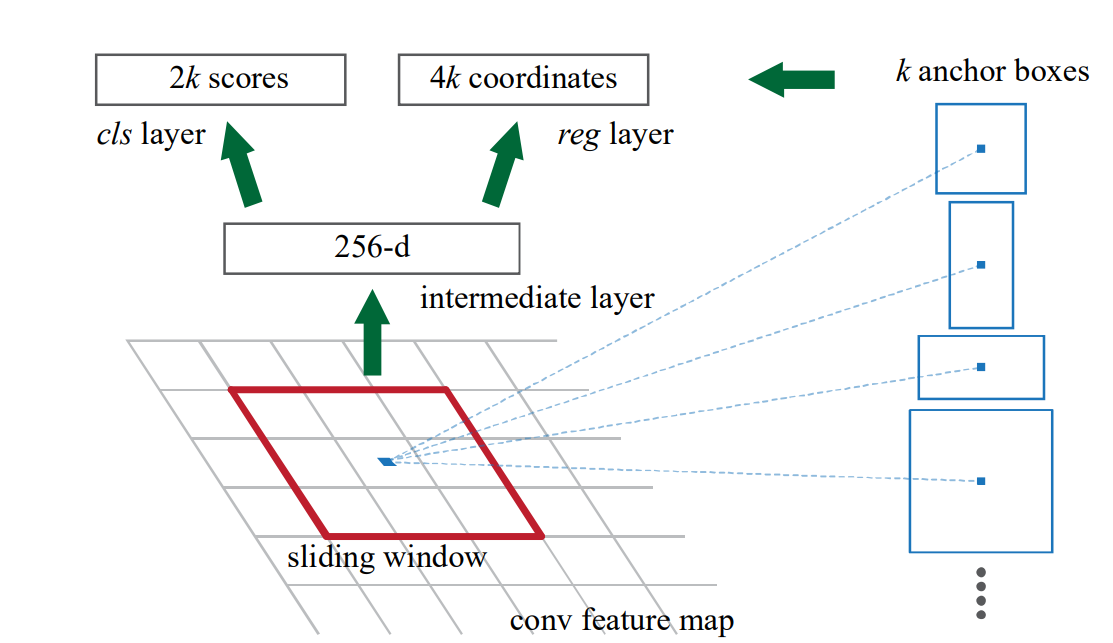
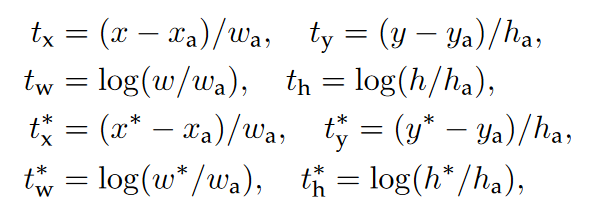


图2-2 anchor box示例

为了训练RPN，本文定义了如下所示的损失函数



其中，是mini-batch中锚点的索引，是当前锚框中存在目标的预测概率。是真实概率。是边界框预测参数的向量，是边界框真实参数的向量。是两个类别（目标或不是目标）的对数损失。用于计算回归损失，，其中是Fast-RCNN论文中提到的鲁棒损失函数。的存在使得只计算正样本的损失。



**2.2RPN与Fast-RCNN联合训练**

为了实现RPN和Fast-RCNN之间的特征共享，且实现完全的端到端训练，作者实现了一种四步交替训练的算法(4-Step Alternating Training)，将整个Faster-RCNN的训练分为了如下4步：

1、训练RPN

在第一步中，RPN被独立训练。使用预训练的ImageNet模型作为初始参数，然后通过对RPN进行端到端的微调，来学习生成高质量的候选目标框。

2、训练独立的Fast R-CNN

在第二步中，独立的Fast R-CNN网络被训练。同样使用预训练的ImageNet模型作为初始参数。在这个阶段，RPN生成的候选目标框用于训练Fast R-CNN，但是两个网络的卷积层参数仍然是独立的，没有共享。

3、初始化RPN共享参数

在第三步中，使用第二步训练得到的Fast R-CNN网络来初始化RPN的训练。在这一步中，RPN的共享卷积层参数被固定，只微调RPN特有的层。这样，RPN开始共享部分特征表示。

4、微调Fast R-CNN共享参数

在第四步中，保持共享的卷积层参数固定，只微调Fast R-CNN特有的层。这样，Fast R-CNN也开始共享与RPN相同的卷积特征。

通过这四步的交替训练过程，RPN和fast RCNN逐步共享特征，形成了一个整体的联合网络。这种共享特征的方式使得整个网络可以更好地理解图像，并有效地检测目标。这个训练算法在Faster R-CNN中起到了至关重要的作用，使得该方法在目标检测任务中取得了显著的性能提升。

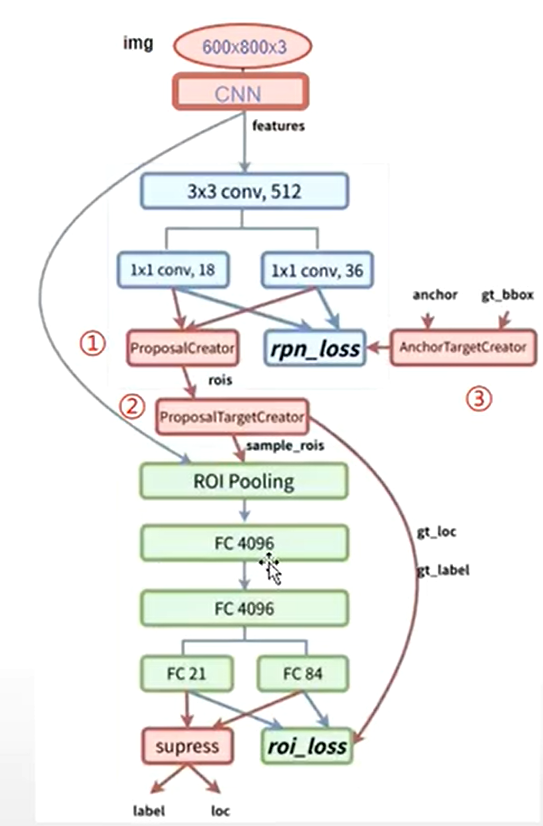


图2-3 交替训练示意图

**3 实验**

本文进行了一系列实验来验证所提出的 Faster R-CNN 方法在目标检测任务中的有效性和优越性。

Region Proposal Networks（RPN）效果验证：首先，作者在PASCAL VOC 2007数据集上验证了RPN的效果。RPN作为独立模块，在不使用Fast R-CNN的情况下，它可以快速且准确地生成候选目标框。实验结果表明，RPN在生成高质量候选框方面取得了显著的性能提升。

Faster R-CNN 整体性能验证：接着，作者将RPN与Fast R-CNN整合为一个联合网络，即Faster R-CNN。在PASCAL VOC 2007和2012数据集以及MS COCO数据集上，Faster R-CNN相比之前的目标检测方法，取得了更高的目标检测准确率。特别是在目标检测速度方面，Faster R-CNN显著地加快了检测过程。

不同骨干网络的比较：作者还在不同的骨干网络上比较了Faster R-CNN的性能。通过使用VGG-16和ResNet作为特征提取网络，实验结果表明ResNet在Faster R-CNN中表现更好，特别是对于更复杂和大规模的数据集，ResNet能够提供更好的性能。

超参数敏感性分析：作者还进行了超参数敏感性分析，对Faster R-CNN的超参数进行了调整和验证。实验结果表明，Faster R-CNN对一些超参数相对较稳定，对其他超参数较为敏感。