



相信这样看过去就一目了然了，但是需要说明的还是：YOLO可能不应该放在这里，但是为了和SSD进行比较还是放了。另外，YOLO出了第二版本了，所以放在这边也没有问题。

个人觉得，分析比较Faster Yolo SSD这几种算法，有一个问题要先回答，Yolo SSD为什么快？

最主要的原因还是提proposal（最后输出将全连接换成全卷积也是一点）。其实总结起来我认为有两种方式：1.RPN，2. 暴力划分。RPN的设计相当于是一个sliding window 对最后的特征图每一个位置都进行了估计，由此找出anchor上面不同变换的proposal，设计非常经典，代价就是sliding window的代价。相比较yolo比较暴力，直接划为7*7的网格，估计以网格为中心两个位置也就是总共98个"proposal"。快的很明显，精度和格子的大小有关。SSD则是结合：不同layer输出的输出的不同尺度的 Feature Map提出来，划格子，多种尺度的格子，在格子上提"anchor"。结果显而易见。

需要说明一个核心：目前虽然已经有更多的RCNN，但是Faster RCNN当中的RPN仍然是一个经典的设计。下面来说一下RPN：

在Faster RCNN当中，一张大小为224*224的图片经过前面的5个卷积层，输出256张大小为13*13的特征图（你也可以理解为一张13*13*256大小的特征图，256表示通道数）。接下来将其输入到RPN网络，输出可能存在目标的reign WHk个（其中WH是特征图的大小，k是anchor的个数）。

实际上，这个RPN由两部分构成：一个卷积层，一对全连接层分别输出分类结果（cls layer）以及 坐标回归结果（reg layer）。卷积层：stride为1，卷积核大小为3*3，输出256张特征图（这一层实际参数为3*3*256*256）。相当于一个sliding window 探索输入特征图的每一个3*3的区域位置。当这个13*13*256特征图输入到RPN网络以后，通过卷积层得到13*13个 256特征图。也就是169个256维的特征向量，每一个对应一个3*3的区域位置，每一个位置提供9个anchor。于是，对于每一个256维的特征，经过一对全连接网络（也可以是1*1的卷积核的卷积网络），一个输出 前景还是背景的输出2D；另一个输出回归的坐标信息（x,y,w, h,4*9D，但实际上是一个处理过的坐标位置）。于是，在这9个位置附近求到了一个真实的候选位置。

作者：xiaoiker
来源：CSDN
原文：<https://blog.csdn.net/ikerpeng/article/details/54316814>
版权声明：本文为博主原创文章，转载请附上博文链接！

