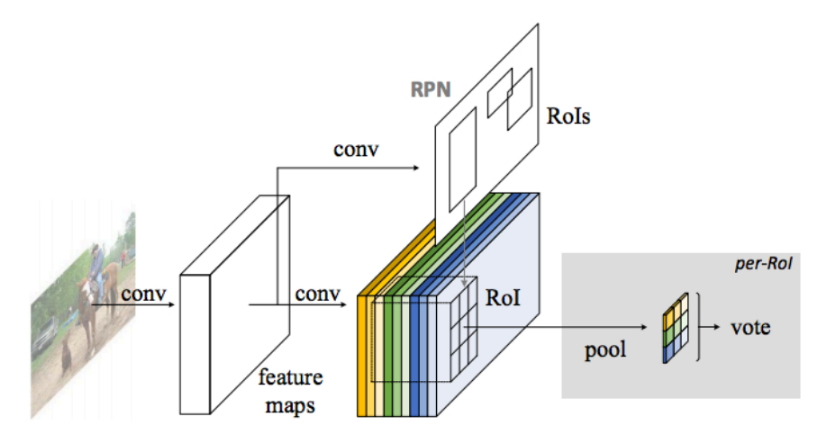
1. R-FCN算法步骤



首先，我们选择一张需要处理的图片，并对这张图片进行相应的预处理操作；

接着，我们将预处理后的图片送入一个预训练好的分类网络中（这里使用了ResNet-101网络的Conv4之前的网络），固定其对应的网络参数；

接着，在预训练网络的最后一个卷积层获得的feature map上存在3个分支，第1个分支就是在该feature map上面进行RPN操作，获得相应的ROI；第2个分支就是在该feature map上获得一个K\*K\*（C+1）维的位置敏感得分映射（position-sensitive score map），用来进行分类；第3个分支就是在该feature map上获得一个4\*K\*K维的位置敏感得分映射，用来进行回归；

最后，在K\*K\*（C+1）维的位置敏感得分映射和4\*K\*K维的位置敏感得分映射上面分别执行位置敏感的ROI池化操作（Position-Sensitive Rol Pooling，这里使用的是平均池化操作），获得对应的类别和位置信息。

**2. Position-Sensitive Score Map解析**

我们将RoI划分为K\*K个子区域是希望这个RoI在其中的每一个子区域都应该含有该类别C的物体的各个部位。**一个RoI必须是K\*K个子区域都含有该物体的相应部位，我们才能判断该RoI属于该物体，如果该物体的很多部位都没有出现在相应的子区域中，那么就该RoI判断为背景类别**

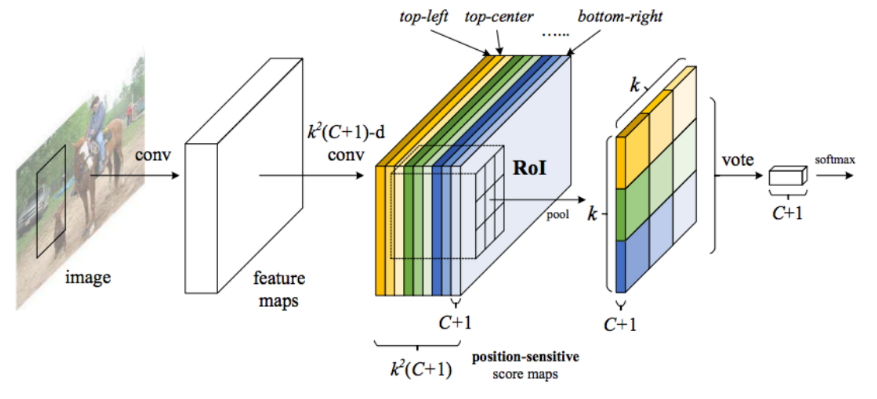
如何判断一个RoI的 K\*K个子区域都含有相应部位呢？

R-FCN会在共享卷积层的最后一层网络上接上一个卷积层，而该卷积层就是**位置敏感得分图position-sensitive score map，该score map的含义如下所述，首先它就是一层卷积层，它的height和width和共享卷积层的一样（即具有同样的感受野），但是它的通道个数为K\*K\*(C+1)**。其中C表示物体类别种数，再加上1个背景类别，所以共有(C+1)类，而每个类别都有 K\*K个score maps

现在我们只针对其中的一个类别来进行说明，假设我们的目标属于人这个类别，那么其有 K\*K 个score maps，**每一个score map表示原始图像中的哪些位置含有人的某个部位，该score map会在含有对应的人体的某个部位的位置有高的响应值**，也就是说每一个score map都是用来描述人体的其中一个部位出现在该score map的何处，而在出现的地方就有高响应值”。既然是这样，那么我们只要将RoI的各个子区域对应到属于人的每一个score map上然后获取它的响应值就好了。**但是要注意的是，由于一个score map都是只属于一个类别的一个部位的，所以RoI的第 i个子区域一定要到第i张score map上去寻找对应区域的响应值，因为RoI的第i个子区域需要的部位和第i张score map关注的部位是对应的**。那么现在该RoI的K\*K个子区域都已经分别在属于人的K\*K个score maps上找到其响应值了，那么如果这些响应值都很高，那么就证明该RoI是人呀

**3. Position-Sensitive Rol Pooling解析**

通过RPN提取出来的RoI区域，其是包含了x,y,w,h的4个值，也就是说不同的RoI区域能够对应到score map的不同位置上，而一个RoI会被划分成K\*K个bins（也就是子区域。每个子区域bin的长宽分别是 h/k 和 w/k ），每个bin都对应到score map上的某一个区域。既然该RoI的每个bin都对应到score map上的某一个子区域，那么池化操作就是在该bin对应的score map上的子区域执行，且执行的是平均池化。我们在前面已经讲了，**第i个bin应该在第i个score map上寻找响应值，那么也就是在第i个score map上的第i个bin对应的位置上进行平均池化操作**。由于我们有(C+1)个类别，所以每个类别都要进行相同方式的池化操作。



对于每个类别，它都有K\*K个score maps，那么按照上述的池化方式，**ROI可以针对该类别可以获得K\*K个值，那么一共有(C+1)个类别，那么一个RoI就可以得到K\*K\*(C+1)个值，就是上图的特征图。那么对于每个类别，该类别的K\*K个值都表示该RoI属于该类别的响应值，那么将这K\*K个数相加就得到该类别的score**，那么一共有(C+1)个scores，那么在这(C+1)个数上面使用简单的softmax函数就可以得到各个类别的概率了（注意，这里不需要使softmax分类器了，只需要使用简答的softmax函数，因为这里就是通过简单的比大小来判断最终的类别的）。

**4. Position-Sensitive Regression解析**

前面的position-sensitive score map和Position-sensitive RoI pooling得到的值是用来分类的，那么自然需要相应的操作得到对应的值来进行回归操作。

按照position-sensitive score map和Position-sensitive RoI pooling思路，其会让每一个RoI得到(C+1)个数作为每个类别的score，那么现在每个RoI还需要 4个数作为回归偏移量，也就是x,y,w,h的偏移量，所以仿照分类设计的思想，我们还需要一个类似于position-sensitive score map的用于回归的score map。

那么应该如何设置这个score map呢，论文中给出了说明：**即在ResNet的共享卷积层的最后一层上面连接一个与position-sensitive score map并行的score maps，该score maps用来进行regression操作，我们将其命名为regression score map，而该regression score map的维度应当是 4\*K\*K** ，然后经过Position-sensitive RoI pooling操作后，每一个RoI就能得到4个值作为该RoI的x,y,w,h的偏移量了，其思路和分类完全相同。