# 阅读纲要

## 1 自己的总结、评价以及应用

R-FCN相对于R-CNN、Fast(er) R-CNN的改进在于摒弃了FC layer，实现了Fully Convolutional，提高效率。

本论文为了实现将整个网络卷积化Fully Convolutional，提出了deeper ROI-wise这一概念和操作，摒弃costly Fully connected layer，同时整个network architecture的backbone采用ResNet。

本论文提出了几个比较新的概念：

①deeper RoI-wise：the RoI pooling layer of the Faster R-CNN detector [18] is unnaturally inserted between two sets of convolutional layers — this creates a deeper RoI-wise subnetwork.

②a dilemma between translation-invariance in image classification and　translation-variance in object detection.

参考博客：<https://blog.csdn.net/ibunny/article/details/79397694>

③position-sensitive score map（对相对空间位置信息进行编码）：是②的解决方案。

## 2 文章的主要问题（abstract、疑问句中）

为了提高accurancy引入F-RCN，提出了deep RoI-wxie的概念，从而实现Fully Convolutional，接着为了解决translation-invariance in image classification和translation-variance in object detection之间的矛盾，提出了position-sensitive score map（对相对空间位置信息进行编码），从而将classification和localization结合到一起，实现对feature map进行share computation.

## 3 结论（abstract以及conclusion中）

R-FCN相对于R-CNN、Fast(er) R-CNN的改进在于摒弃了FC layer，实现了Fully Convolutional，提高效率。

本论文为了实现将整个网络卷积化Fully Convolutional，提出了deeper ROI-wise这一概念和操作，摒弃costly Fully connected layer，同时整个network architecture的backbone采用ResNet。

## 4 思路脉络（小标题中的关键句）

1 Introduction

2 Our approach

3 Related Work

4 Experiments

4.1 Experiments on PASCAL VOC

4.2 Experiments on MS COCO

5 Conclusion and Future Work

## 5 理解难点

理解难点：R-FCN结构的优点

　　R-FCN要解决的根本问题是Faster R-CNN检测速度慢的问题，速度慢是因为ROI层后的结构对不同的proposal是不共享的，试想下如果有300个proposal，ROI后的全连接网络就要计算300次（每一个RoI都要进行一次“特殊的卷积运算”，得到FC层的一个对应的神经元），这个耗时就太吓人了。所以作者把ROI后的结构往前挪来提升速度，但光是挪动下还不行，ROI在conv5后会引起上节提到的平移可变性问题，必须通过其他方法加强结构的可变性，所以作者就想出了通过添加Position-sensitive score map来达到这个目的。

理解难点：Position-sensitive score map的结构（理解不了）

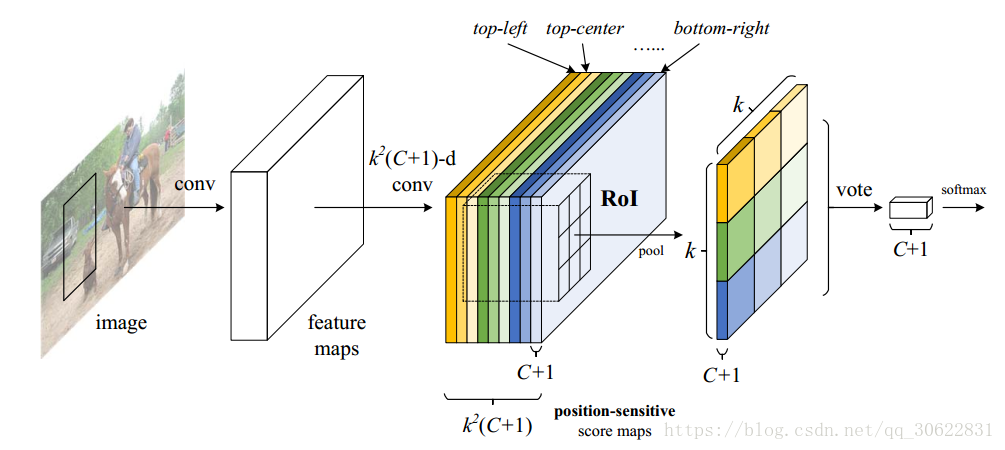
图1的ResNet-101应用到R-FCN时会把最后的average pool和1000-d fc全连接层都去掉了，仅保留前100层，再新加一个1x1x1024的卷积层用来降维（从2048维降到1024维），和一个很特殊的卷积来生成k2 \* (C+1)维的Position-sensitive score map。其中的C是要分类的类别数，比如PASCAL VOC类别就是20，加上1表示加上一个背景分类；k是之后的ROI Pooling中对ROI区域要划分的小格数，比如论文中k=3就是对ROI在长宽方向各三等分形成9个小区域(如图2)。Position-sensitive score map的值对小区域相对于ROI中的位置很敏感，为什么这么说后面会解释。  


图2中最后一个特殊卷积输出Position-sensitive score map后，就要做ROI Pooling了，和Faster R-CNN中的ROI Pooling一样要对9个小区域分别进行pooling，要注意的是R-FCN中９个小区域并不是在所有k2 \* (C+1)维度上都做pooling，每个小区域只会在对应的(C+1)个维度上作pooling，比如ROI左上角的区域就在前C+1个维度上pooling，左中位置的区域就在C+２到２C+２间的维度上作pooling，以此类推。pooling后输出的是C+1维度的k\*k数据，每个维度上的k\*k个数据再加到一起(图２的vote过程)形成C+1个单点数据，就代表了C+1个类别的分类概率。对于目标定位的输出和上面的分类输出过程类似，只是维度不再是k2 \* (C+1)，而是k2\*4（不理解，怎么操作？），表示９个小区域的［dx,dy,dw,dh］4个偏移坐标。