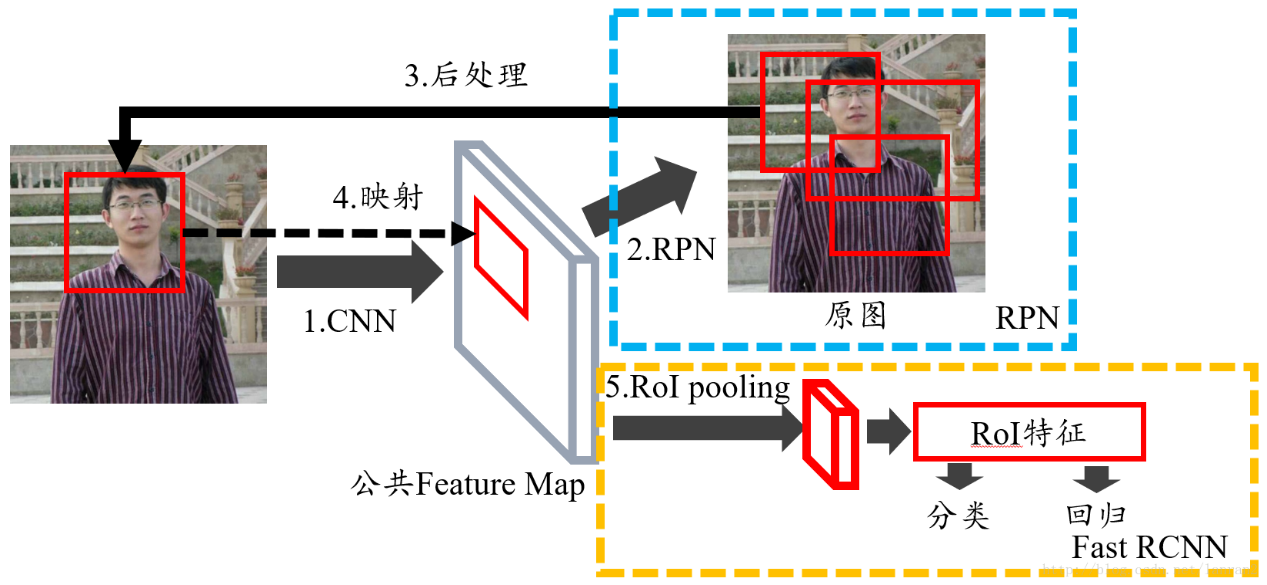
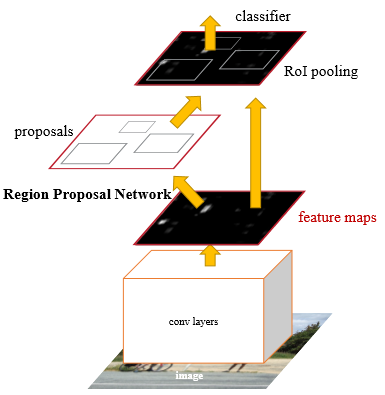
**Mask R-CNN**

1. **Faster R-CNN**

Faster R-CNN是两阶段目标检测算法，包括第一阶段的RPN（Region Proposal Network），以及第二阶段的RoI Pooling（Classification + Regression），下图是Faster R-CNN的整个流程：



Faster R-CNN使用卷积神经网络（CNN）提取图像特征，然后使用RPN提取RoI（Region of Interest），接着使用RoI Pooling将这些RoI全部变成固定尺寸，再输入最终的分类器（一般为CNN+Linear）进行边界框物体的坐标回归与类别预测。

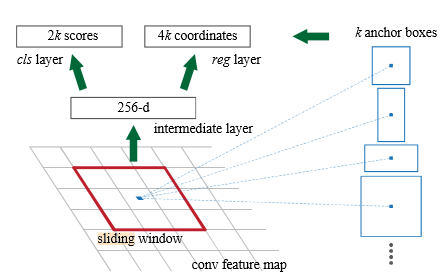
1. **BackBone**

一般来说，用于提取图片特征的那部分CNN称为BackBone。BackBone可以理解为一个简单的卷积神经网络，它几乎可以是任何结构的CNN，比如ResNet、ResNeXt、AlexNet等等都可以充当BackBone。Faster R-CNN一般使用ResNet及其变体作为BackBone。

1. **RPN（Region Proposal Network）**

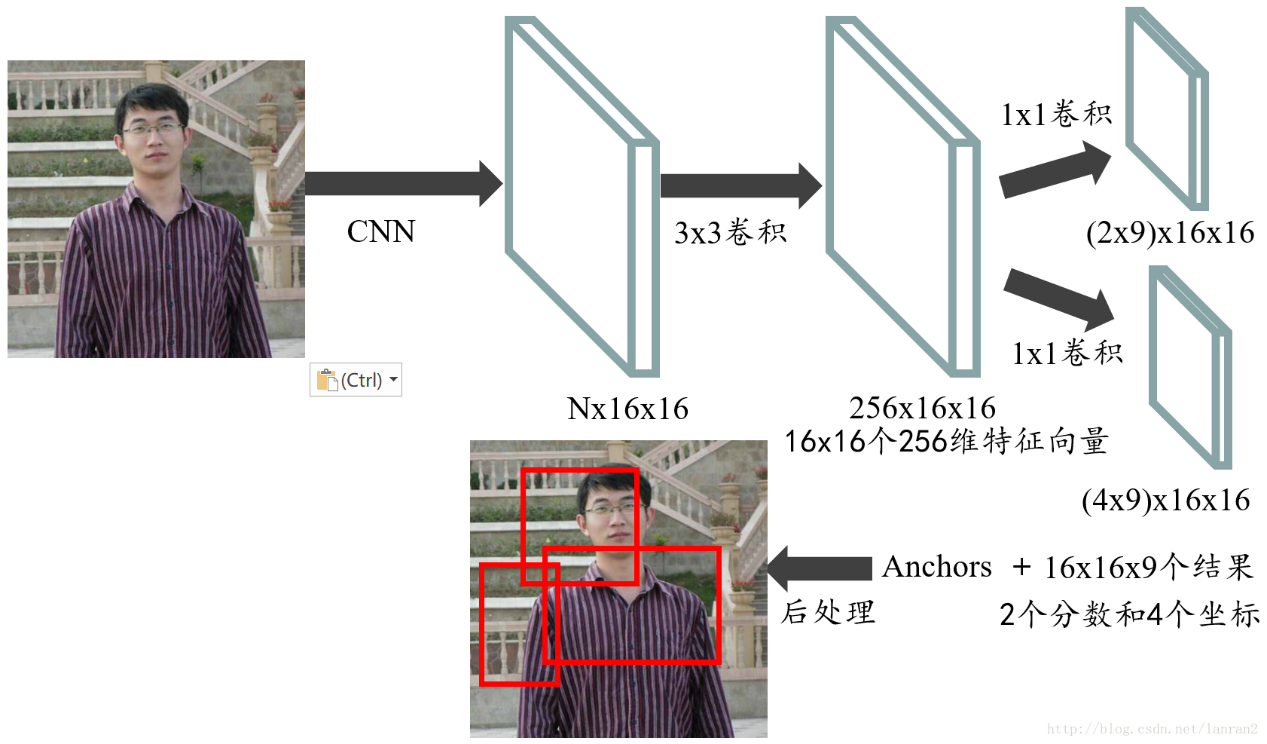
有关于RPN的介绍可以参考这篇博客【<https://www.cnblogs.com/Terrypython/p/10584384.html>】，这里对RPN作简要介绍。

在Faster R-CNN中，输入原始图像经过一系列卷积后得到公共特征图Shared Feature Map，在公共特征图上利用一个n\*n的滑动窗口对其进行特征提取，映射到一个更低维的特征空间。**实际上，这里的滑动窗口是用n\*n的卷积来实现的。也就是说卷积完的特征图对于Shared Feature Map的感受野是n\*n，这样特征图上的每一点就可以映射到原图上一个感受野大小的区域。**最后，再利用特征图分别经过两个1\*1的卷积输出特征图上每个点代表置信度与坐标的特征向量（这里输出的坐标是相对于原图的坐标），然后将坐标映射回公共特征图，生成RoI区域。



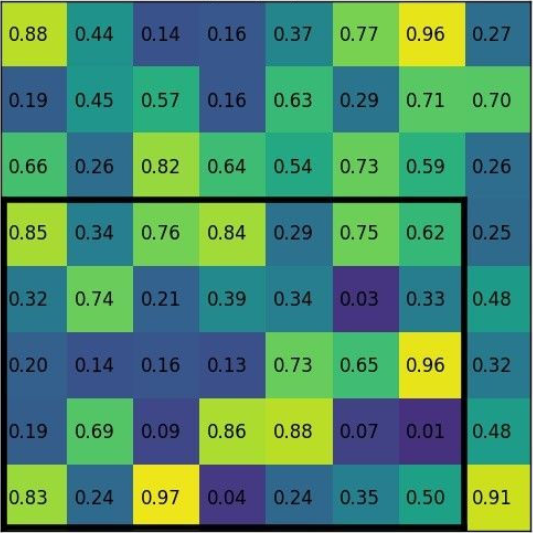
这里之所以是用n\*n的滑动窗口目的是为了进一步结合公共特征图上每一点周围像素的信息（注意公共特征图对于原图的感受野，这里的n\*n的卷积其实进一步扩大了感受野）

RPN只负责生成RoI区域的坐标以及区域内是否有物体的置信度，不负责输出分类结果。RPN只是第一阶段，第二阶段的RoI Pooling负责生成分类结果。

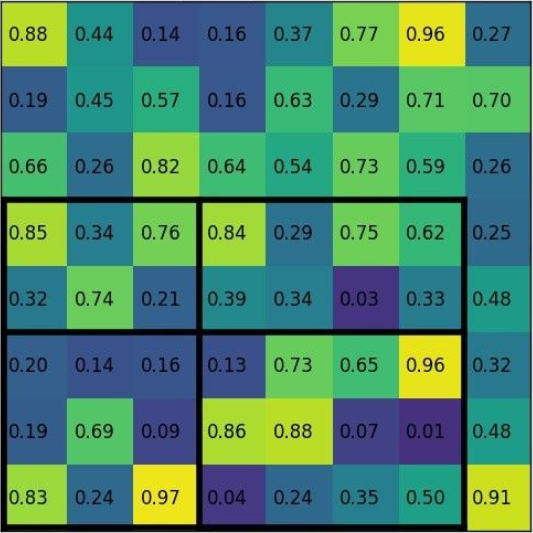


1. **RoI Pooling**

在Faster R-CNN中，在Shared Feature Map上通过RPN后，生成了2k个置信度与4k个坐标。这些坐标是相对于原图的，将这些坐标映射到公共特征图上即形成RoI区域。RoI Pooling就是对这些RoI区域进行操作。

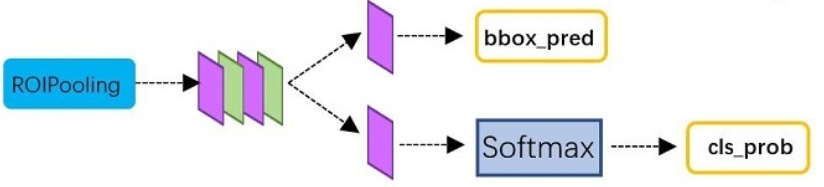


假设需要输出$2\*2$的特征图，那么就对RoI区域进行划分4大块，在每块内进行常规Pooling操作。



实际上RoI Pooling会对RoI区域平均划分k×k个bins，一般来说这些bins的坐标都为浮点数，需要进行取整量化（向下取整），然后对这些bins进行max pooling操作。例如一个7×7的RoI区域需要划分2×2的bins，那么每个bin的长宽为3.5，向下取整到3，每个bin的尺寸就变为3×3，只利用到了RoI区域的一部分，并不会像上图那样用到全部。

1. **Regression & Classification**



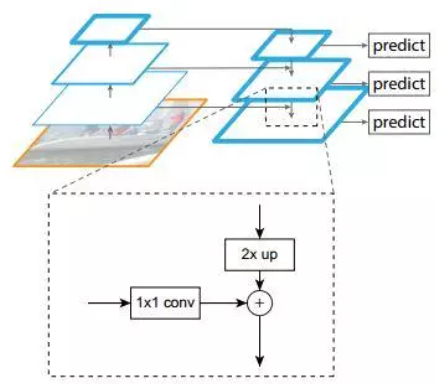
本部分将RoI Pooling得到的特征向量输入几层简单的CNN，最后用两个分类器（全连接层，Linear）对图片进行最后的边界框坐标预测以及类别分类

1. **Mask R-CNN**

Mask R-CNN整体结构是在Faster R-CNN基础上进行细微改动的。下面介绍Mask R-CNN区别于FasterR-CNN的特性。

1. **FPN（Featurre Pyramid Network）**

在前述过程中，我们都是只在网络输出的最后一张特征图上进行产生锚框、生成预测值这一系列操作的。这种方式有一个弊端：模型最后一层输出的特征图尺寸较小，每个像素点的感受野很大，具有非常丰富的高层级语义信息，容易检测到较大的物体，但对小物体的检测效果不尽如人意。如果我们参照高层级特征图的特性，采用一些低层级的特征图进行预测，这些特征图虽然感受野较小，尺寸较大，但它们没有经过充分的特征提取，像素点包含的语义信息不够丰富，有可能很难提取到有效的特征模式。所以，在Mask R-CNN中，借鉴了FPN（Feature Pyramid Network）特征金字塔的Botton-Up以及Top-Down的多层级结构，将高层语义信息丰富的特征图经过上采样后与底层特征图相融合，从而形成多个层级的特征图进行预测输出。FPN的基本结构如下：

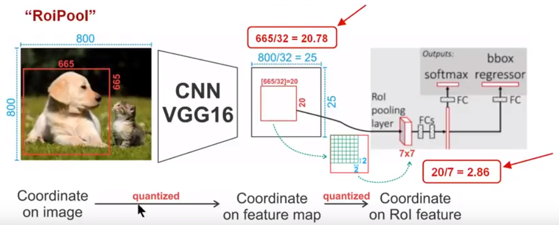


1. **RoI Align**

在实例分割任务中，RoI区域与原图是不对齐的（**Mis-Alignment**），这对于实例分割任务来说是致命的，毕竟实例分割需要精确分割出物体的边界。

**RoI Pooling的局限性：**由RPN产生的回归预测坐标是浮点数，而在RoI Pooling中，要求映射到特征图上的RoI区域尺寸是整数，故会存在两次取整量化过程：

1. 原图坐标映射回特征图上时会将候选框边界量化为整数坐标值
2. 将量化后的候选区域平均分割成k×k个单元（bin），对每一个单元进行量化取整



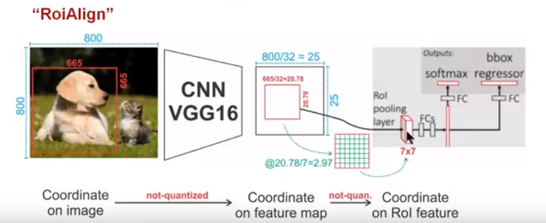
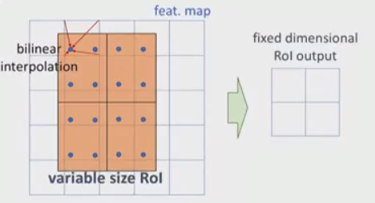
如上图所示，原图尺寸800×800，候选区域在原图上尺寸665×665，公共特征图与原图之间的步长为32。因此，将候选区域映射到特征图上时，需要进行缩放，即：

**RoI Pooling直接将其量化为20，这一步的误差0.78还原到原图上就是0.78 × 32 = 25，接近25个像素点的偏差**。接下来对量化后的RoI区域平均分割成7×7个bins，那么每个bin的尺寸大小为：

**RoI Pooling直接将其量化为2，这一步的误差还原到原图也接近30个像素点了**。

**RoI Align的改进方案：**RoI Align取消了RoI Pooling的所有量化操作，保持浮点数不变。在对RoI区域划分bins的过程中，由于bins的尺寸均为浮点数，故RoI Align对每个bin作如下处理：

1. **选择bin的采样点的数量**。若为1，则采样点就为该bin的中心坐标；若为4，则将bin内部平均划分2×2的区域，每个采样点位于区域中心坐标处
2. **根据采样点周围四个固定的整数坐标值，采用双线性插值（BiLinear Interpolation）计算每个采样点的值**。然后在bin内对采样点值进行max pooling操作，计算出该bin最终的值



1. **Mask Head**

Mask R-CNN在Faster R-CNN的基础上加了Mask Head，用于输出每个像素点是否应该进行Mask的矩阵。

