

ROI Pooling 和 ROIAlign 的特点和区别

ROI 操作: ROI Pooling 和 ROIAlign

目标: 为了使得检测网络可以输入任意 size 的图片, 使用 ROI Pooling 在网络中某一个阶段将不同尺度的图片 ROI pooling 成相同的尺度, 使得 fc 的存在也无法写死输入图片的 size。

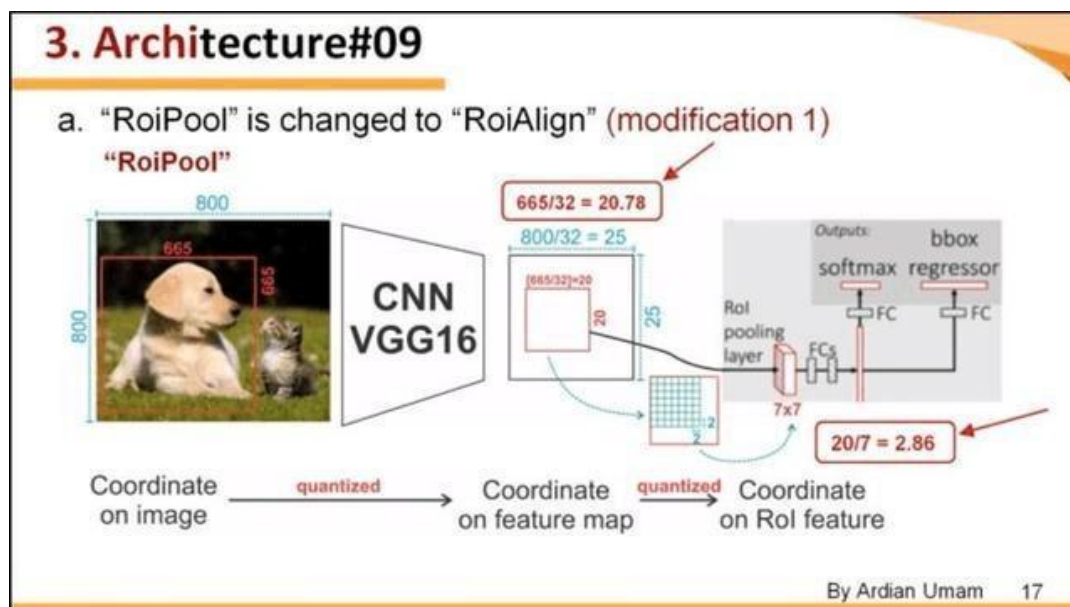
ROI Pooling:

方法: 直接 Pooling, 若尺寸不对应则直接取整

缺点:

由于 ROI Pooling 采用的是 INTER_NEAREST (即最近邻插值), 在 resize 时, 对于 缩放后坐标不能刚好为整数 的情况, 采用了 粗暴的舍去小数, 相当于选取离目标点最近的点, 损失一定的空间精度。

例如: 一张图片输入 VGG16, 则得到的尺寸缩小为 $1/32$, 如下图中, 图片大小为 800, 则恰好为 25, 但区域块大小为 665 经过 VGG 后则为 20.78, 则直接处理为 20, 再通过 ROI Pooling (即, 将所有不同的尺寸全部 Pooling 处理为相同的大小) 处理为 7×7 , 则 $20/7 \rightarrow 2$, 由下图可知, Pooling 之后存在一定的位置偏移。



具体计算:

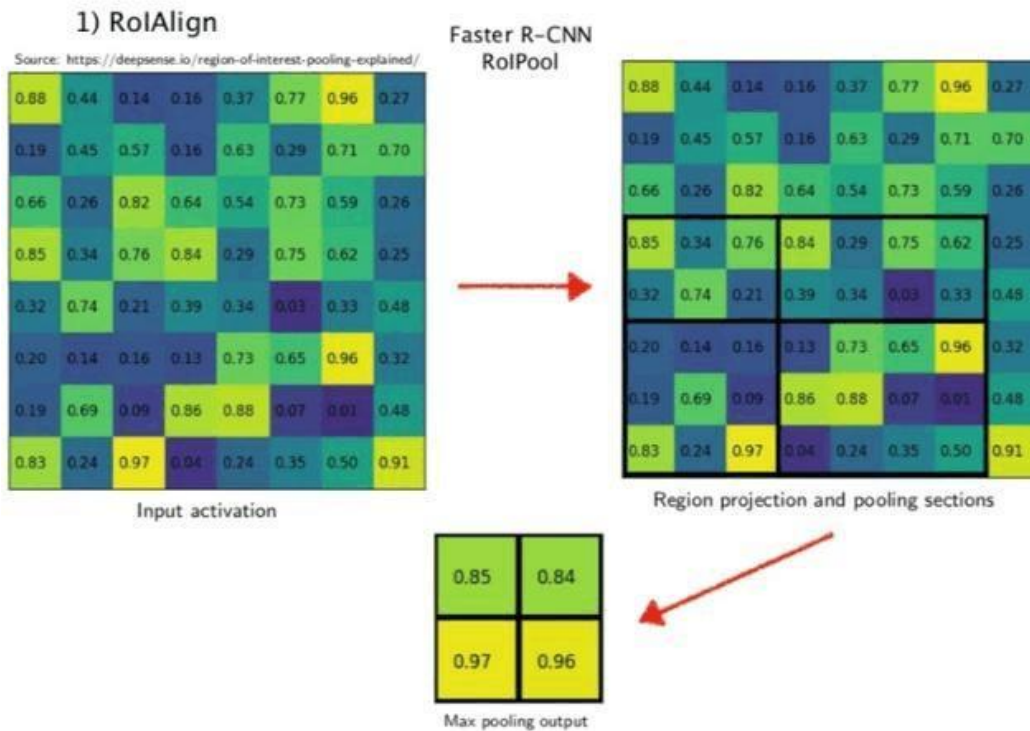
下图为一张 8×8 的 feature map, 选取其中一个 5×7 的 region 输入 ROI Pooling 输出 2×2 的结果。

(1) 划分为 $2 \times 2 = 4$ 块区域

1) $5/2 = 2.5 \rightarrow 2$, 剩下的为 3, 则 $2+3$

2) $7/2 = 3.5 \rightarrow 3$, 剩下的为 4, 则 $3+4$

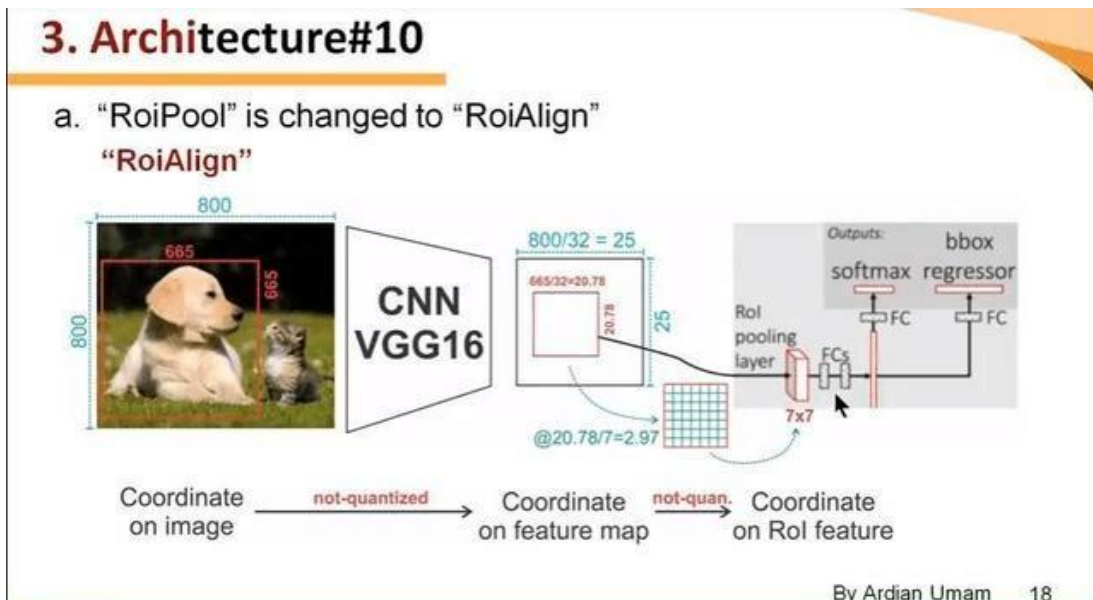
(2) 取每个小区域的最大值为 pooling 值



RoIAlign:

改进：将最近邻插值换为双线性插值，使得即使缩放后坐标不能刚好为整数，也能通过插值得到浮点数处的值处理得到 pooling 后的值。

如下图所示：RoIAlign 中池化的块允许为浮点数，通过双线性插值得到尺寸为浮点数的块池化后的结果，保证了空间精度。



具体计算：

下图为一张 8*8 的 feature map，选取其中一个 5*7 的 region 输入 ROI Pooling 输出 2*2 的结果。

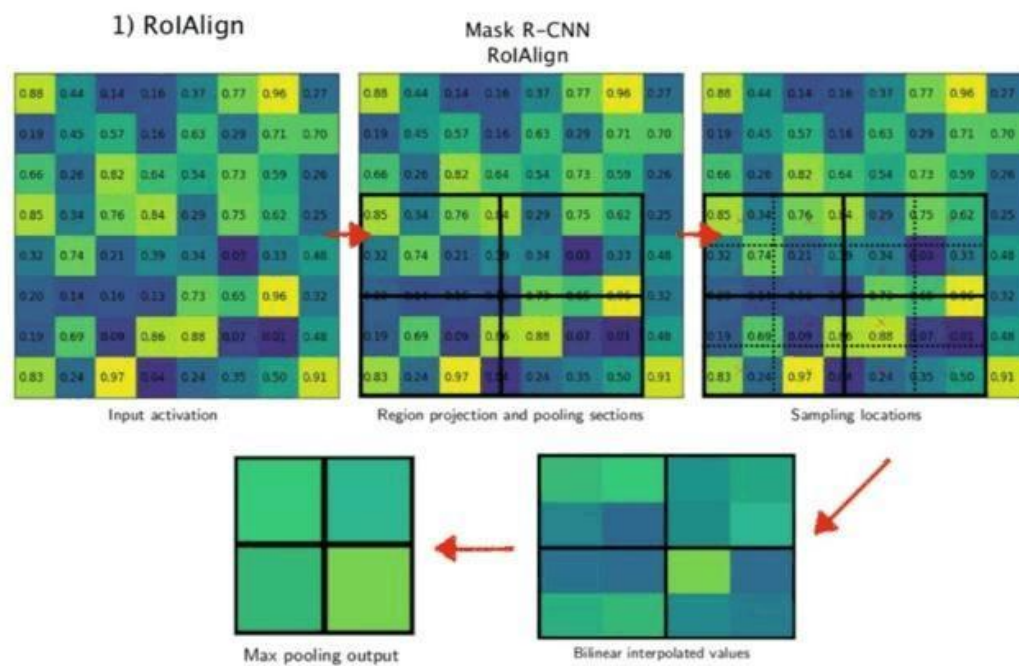
(1) 划分为 $2 \times 2 = 4$ 块区域

1) $5/2 = 2.5 \rightarrow 2.5$ ，不再取整，则 $2.5 + 2.5$

2) $7/2 = 3.5 \rightarrow 3.5$ ，不再取整，则 $3.5 + 3.5$

(2) 将每小块再分为 4 个小区域，使用双线性插值的方法求取这四个小区域的中心点处的值

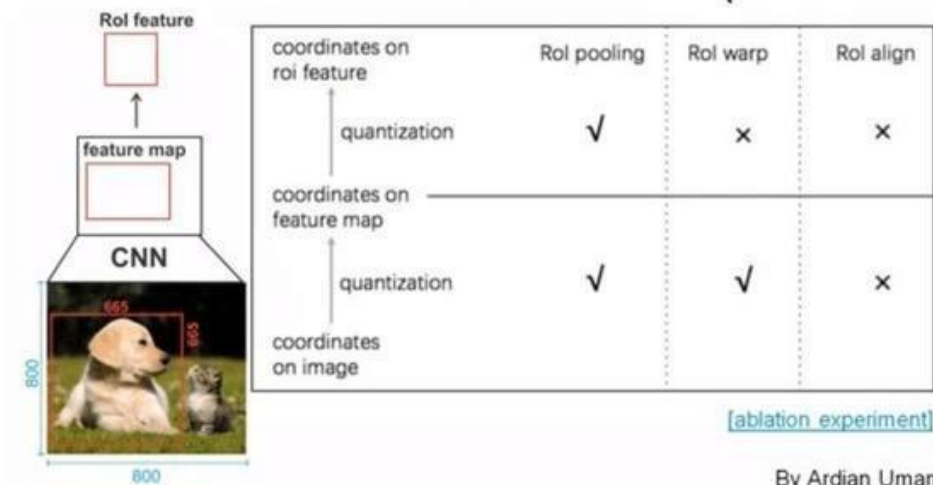
(3) 取每个区域的最大值为 pooling 值



ROI Pooling 换为 RoIAlign 所做的改动：

3. Architecture#12

a. "RoiPool" is changed to "RoiAlign"



By Ardian Umam 20

3. Architecture#13

Modification 2

b. Decouple mask and class prediction

- predict binary mask for each class.
- mask branch can predict K (K = number of classes) masks per RoI, only use the k-th mask, where k is the predicted class by the classification branch

