

Faster R-CNN 中 RPN 原理

1.RPN 原理

RPN 的用途在于, 判断需要处理的图片区域(where), 以降低推断时的计算量.

RPN 快速有效的扫描图片中每一个位置, 以判断给定区域是否需要进一步处理. 其产生 k 个 bounding-box proposals, 每一个 box proposal 有两个分数, 分别表示该 box 中是 object 的概率.

anchor 用于寻找 boxes proposals.

anchor boxes 是参考 boxes, 所选择的 anchors 具有不同的长宽比(aspect ratios) 和尺度(scale), 以囊括不同类型的 objects.

细长的 objects, 如 buses, 则不能用方形 square bounding box 来合适的表示.

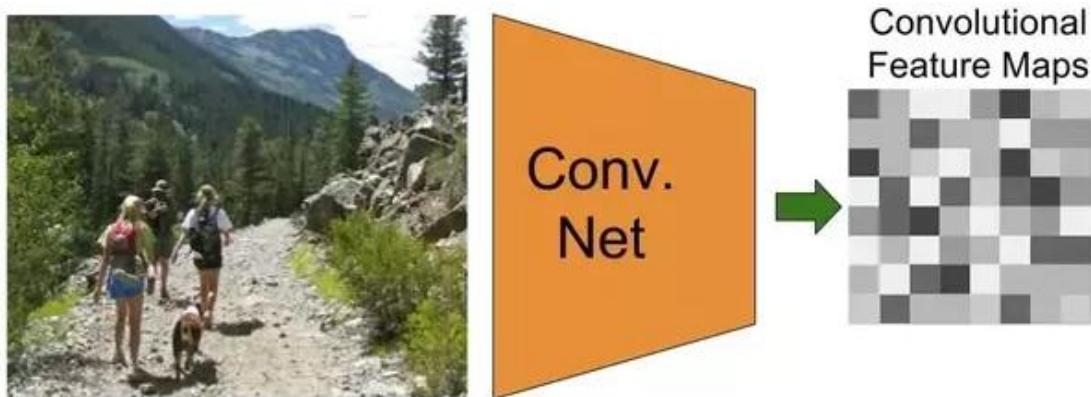
Faster R-CNN 采用了 $k=9$ 个 anchors, 分别为 3 aspect ratios 和 3 scales.

RPN 的每个 regressor 只计算与对应参考 anchor box 的 4 个偏移值 (w, h, x, y).

RPN 采用 $3 \times 3 \times 3$ 的滑窗, 其有效的接受野实际上是 $177 \times 177 \times 177$. 因此, RPN 在生成 proposals 时用到了大量的内容信息.

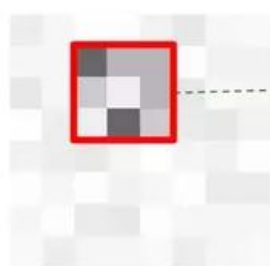
RPN 主要可以包括三步:

1. 输入图片经卷积网络(如 VGGNet 和 ResNet)处理后, 会输出最后一个卷积层的 feature maps;



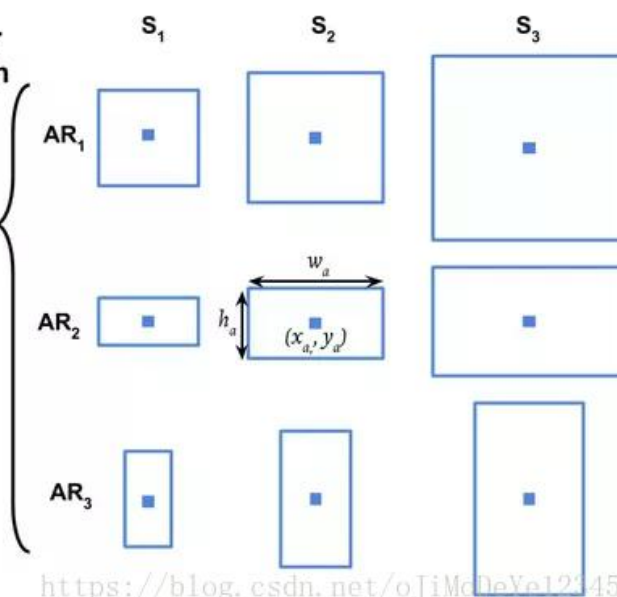
2. 在 feature maps 上进行滑窗操作(sliding window). 滑窗尺寸为 $n \times n \times n$, 如 $3 \times 3 \times 3$. 对于每个滑窗, 会生成 9 个 anchors, anchors 具有相同的中心 $center=x_a, y_a$, 但 anchors 具有 3 种不同的长宽比(aspect ratios) 和 3 种不同的尺度(scales), 计算是相对于原始图片尺寸的, 如下图:

Generate 9 anchors for each **sliding window** on conv. feature map



w_a : anchor's width
 h_a : anchor's height
 x_a, y_a : anchor's center

@vmirly



<https://blog.csdn.net/oJiMcDeVe12345>

对于每个 anchor, 计算 anchor 与 ground-truth bounding boxes 的重叠部分(overlap) 值 $p^* = \text{IoU}$ (intersection over union):

如果 $\text{IoU} > 0.7$, 则 $p^* = 1$

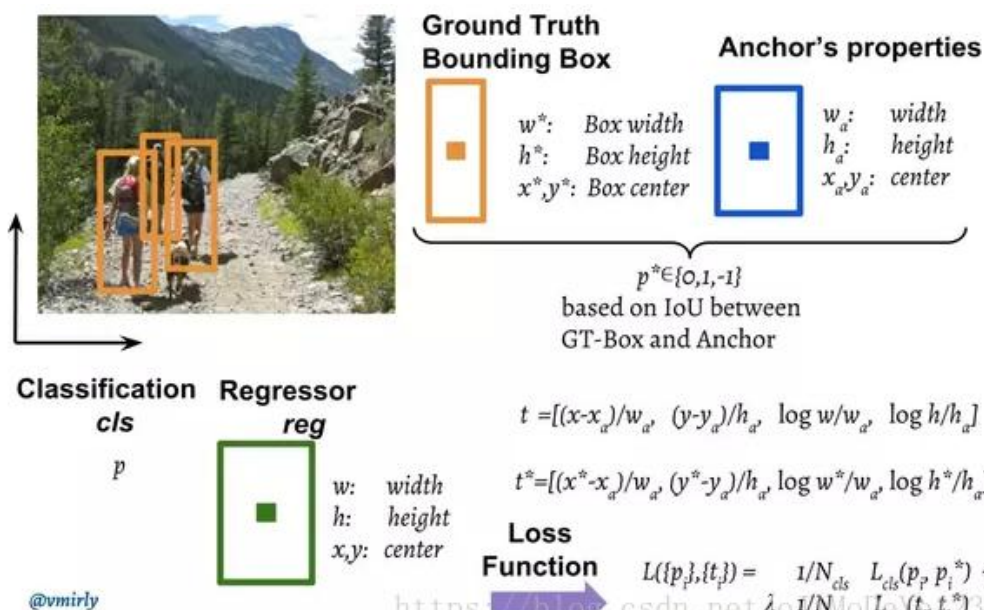
如果 $\text{IoU} < 0.3$, 则 $p^* = -1$

其它, $p^* = 0$

3. 从 feature maps 中提取 $3 \times 3 \times 3$ 的空间特征(上图中红色方框部分), 并将其送入一个小网络. 该网络具有两个输出任务分支: classification(cls) 和 regression(reg).

regression 分支输出预测的边界框 bounding-box: (x, y, w, h).

classification 分支输出一个概率值, 表示 bounding-box 中是否包含 object (classid = 1), 或者是 background (classid = 0), no object.



@vmirly

<https://blog.csdn.net/oJiMcDeVe12345>

2. Anchors 生成示例

Detectron 中 `generate_anchors.py` 给出了 anchors 的实现.

主要包括两步:

保持 anchor 面积固定不变, 改变长宽比(aspect ratio)

`_ratio_enum(anchor, ratios)`

保持 anchor 长宽比固定不变, 缩放尺度 scale

`_scale_enum(anchor, scales)`

最终生成 $5 \times 3 = 15$ 个 anchors.