

layout: post

title: "RCNN系列文章之Cascade RCNN详解"

date: 2020-07-18

description: "目标检测"

## tag: 目标检测

RCNN系列的文章主要是RCNN，Fast RCNN，Faster RCNN，Mask RCNN, Cascade RCNN,这一系列的文章是目标检测two-stage算法的代表，这系列的算法精度高，效果好，是一类重要的方法。

论文地址：[Cascade R-CNN](#)

## 简要介绍

在目标检测中，IOU阈值被用来定义正样本（positive）与负样本（negative）

1. 如果使用较低的IOU阈值，那么会学习到大量的背景框，产生大量的噪声预测。
2. 但是 如果采用较高的阈值，这个检测器的表现往往会变得很差，两个主要的原因，第一就是随着IOU阈值的增加，正样本的数量会呈指数级的减小，因此产生过拟合。第二就是推理过程中出现于IOU的误匹配，也就是在训练优化感知器的过程中的最优IOU与输入proposal的IOU不相同，出现误匹配，这样很大程度上降低了检测精度。
  - mismatch：detector通常在proposal自身的IOU值与detector训练的IOU阈值较为接近的时候才会有更好的结果，如果二者差异较大那么很难产生良好的检测效果
3. 因此为了解决这个问题，多阶段（multi-stage）的Cascade RCNN横空出世。它由多个感知器构成，这些感知器通过递增的IOU阈值分级段训练。一个感知器输出一个良好的数据分布来作为输入训练下一个高质量感知器，这样对于假阳性的问题会好很多，在inference阶段使用同样的网络结构合理的提高了IOU的阈值而不会出现之前所说的mismatch问题

网络结构图：

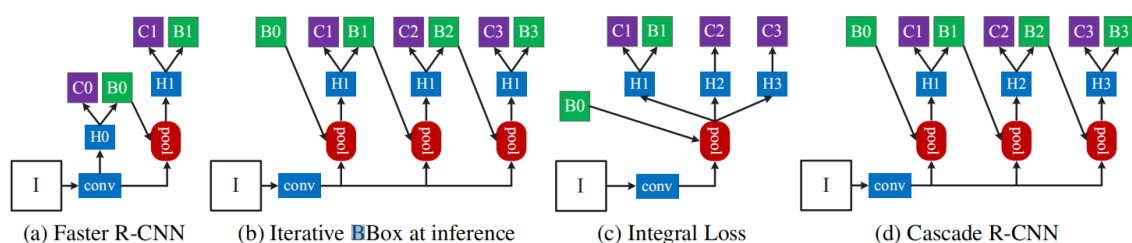
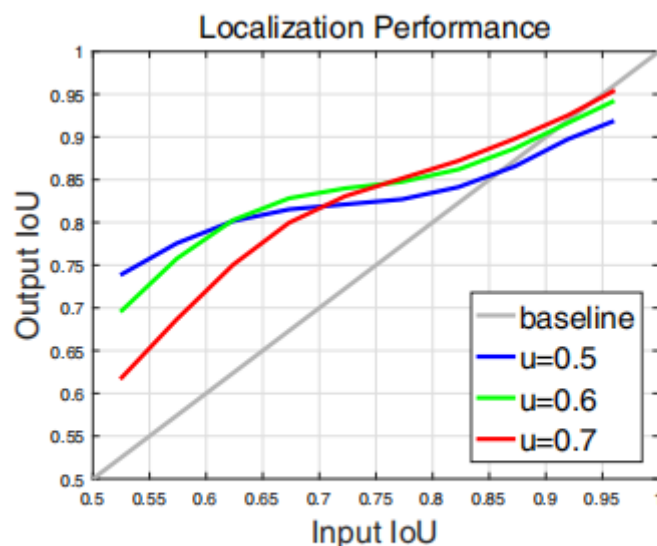


Figure 3. The architectures of different frameworks. "I" is input image, "conv" backbone convolutions, "pool" region-wise feature extraction, "H" network head, "B" bounding box, and "C" classification. "B0" is proposals in all architectures.

其中三个stage的IOU阈值分别为0.5,0.6,0.7。

对比cascade rcnn与Iterative BBox的区别，cascade rcnn的每个stage采用了不同的head，这样cascade rcnn不同的stage可以适应不同的分布，效果更好。

## 网络结构的一些分析



(c) Regressor

1. 一个检测器通常只在一个小范围的IOU阈值内(a single quality level)性能最好，从上图中可以发现，在0.55-0.6的范围内阈值为0.5的detector性能最好，在0.6~0.75阈值为0.6的detector性能最佳，而到了0.75之后就是阈值为0.7的detector了，比IOU阈值过高过低的proposal都会导致检测器性能下降，因此保证检测器训练时的IOU阈值与输入proposal的IOU相近是十分重要并且有必要的，否则就会出现前文所说的mismatch的情况，这种由很多的实验证明，并没有严格的理论证明。
2. 通过观察上图可以发现，几乎所有检测器输出的检测框的IOU都好于输入proposal的IOU（曲线几乎都在灰色对角线之上），因此这保证了我们通过一个检测器输出的检测框整体IOU相对输入的proposal的IOU都会提高，可以作为下一个使用更高IOU阈值训练检测器一个很好的数据输入。因此每个检测器输出的检测框质量都会变高，阈值的提高其实也相当于一个resample的过程，一些异常值也可以去掉，提高了模型的鲁棒性。
3. 那么持续的提高阈值，又如何避免之前提到的正样本数目呈现指数级的消失，导致过拟合呢？作者通过详细的实验证明了每个阶段大于对应IOU阈值的proposal数量基本没有改变，甚至还有所提升，实验结果如下：

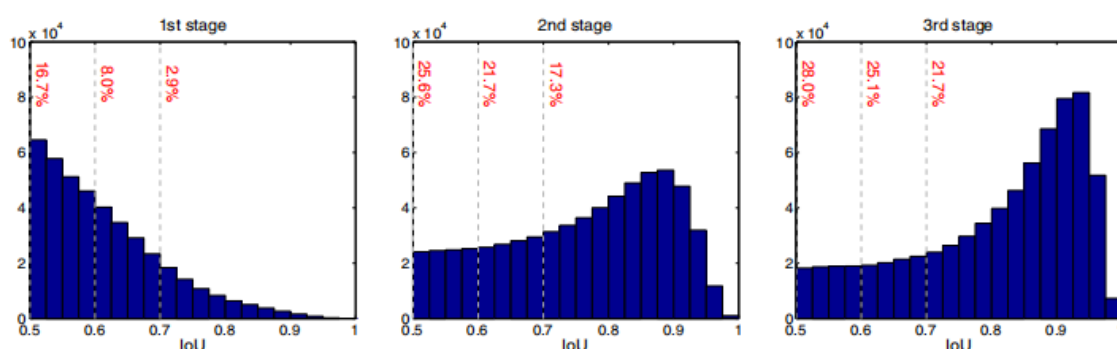
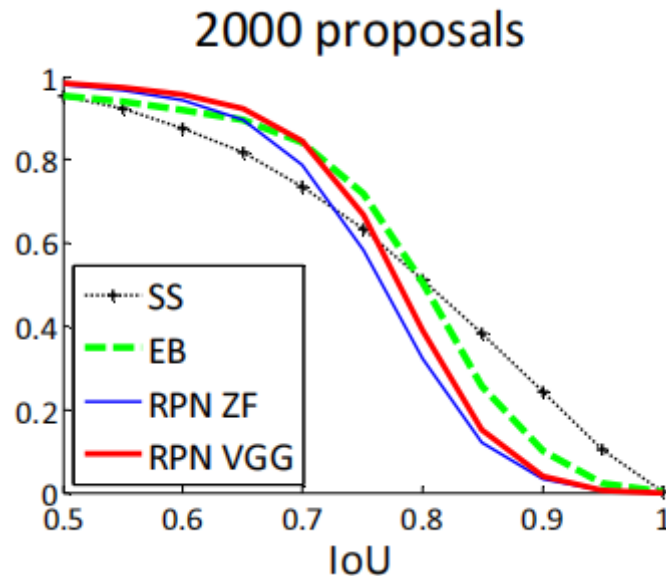


Figure 4. The IoU histogram of training samples. The distribution at 1st stage is the output of RPN. The red numbers are the positive percentage higher than the corresponding IoU threshold.

从图中可以看出，casacde随着stage的加深，相应区域的依然拥有大量的proposal，因此不会出现严重的过拟合的现象。

参考知乎大佬的一个关于faster rcnn的分析：<https://zhuanlan.zhihu.com/p/42553957>

- training阶段，RPN网络提出了2000左右的proposals，这些proposals被送入到Fast R-CNN结构中，在Fast R-CNN结构中，首先计算每个proposal和gt之间的iou，通过人为的设定一个IoU阈值（通常为0.5），把这些Proposals分为正样本（前景）和负样本（背景），并对这些正负样本采样，使得他们之间的比例尽量满足（1:3，二者总数量通常为128），之后这些proposals（128个）被送入到Roi Pooling，最后进行类别分类和box回归。
- inference阶段，RPN网络提出了300左右的proposals，这些proposals被送入到Fast R-CNN结构中，和training阶段不同的是，inference阶段没有办法对这些proposals采样（inference阶段肯定不知道gt的，也就没法计算iou），所以他们直接进入Roi Pooling，之后进行类别分类和box回归。



faster rcnn论文给出了RPN的proposal的关系，横轴表示Proposals和gt之间的iou值，纵轴表示满足当前iou值的Proposals数量。

- 在training阶段，由于我们知道gt，所以可以很自然的把与gt的iou大于threshold（0.5）的Proposals作为正样本，这些正样本参与之后的bbox回归学习。
- 在inference阶段，由于我们不知道gt，所以只能把所有的proposal都当做正样本，让后面的bbox回归器回归坐标。

我们可以明显的看到training阶段和inference阶段，bbox回归器的输入分布是不一样的，training阶段的输入proposals质量更高(被采样过， $\text{IoU} > \text{threshold}$ )，inference阶段的输入proposals质量相对较差（没有被采样过，可能包括很多 $\text{IoU} < \text{threshold}$ 的），这就是论文中提到**mismatch**问题，这个问题是固有存在的，通常threshold取0.5时，mismatch问题还不会很严重。

cascade使用级联的detector，每个stage采用递增的阈值，实验证明，每个stage均有足够数量的proposal，并不会出现严重的过拟合的现象。

虽然在测试的时候RPN得到的proposal依然质量不高，但是经过几个级联的stage之后，proposal的质量逐步提高，不会出现严重的mismatch现象。

其他的结构及loss等各个其他的部分均类似于Faster RCNN，此处不再赘述。

[更多深度学习论文点击查看](#)

[github地址](#)