

论文题目： Cascade R-CNN: Delving into High Quality Object Detection

----from CVPR2018

一、背景

作者提出在目标检测中,需要交叉联合(IoU)阈值来定义positive和negative目标检测器用低IoU阈值训练,例如0.5,通常会产生*noisy detections*.但是,随着IoU阈值的增加,检测性能会下降,主要是受到以下影响：

- 1) 训练期间由于样本减少出现过拟合问题
- 2) 探测器最佳的IoU与输入假设的IoU之间的推理时间不匹配

二、创新点

- 1、作者提出了一种多级目标检测架构 Cascade R-CNN，它由一系列通过增加IoU 阈值训练的探测器组成，依次对近距离误报具有更强的选择性。
- 2、探测器 stage by stage(逐步)进行训练,利用观察器的输出是良好的分布来训练下一个更高质量的探测器。
- 3、逐步改进的假设的重新采样保证了所有检测器都具有相同大小的一组正实例，从而减少了过度拟合问题。
- 4、在推理中应用相同的级联过程,使得假设与每个级的检测器质量之间能够更紧密地匹配。

三、Result

Cascade R-CNN超越了具有挑战性的COCO数据集上的所有单模型物体探测器。

实验还表明， Cascade R-CNN可广泛应用于探测器架构， 无需baseline探测器强度即可实现一致的增益。

四、Cascade R-CNN

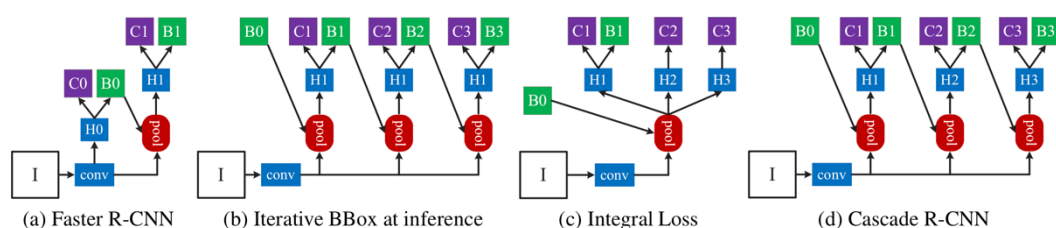


Figure 3. The architectures of different frameworks. “I” is input image, “conv” backbone convolutions, “pool” region-wise feature extraction, “H” network head, “B” bounding box, and “C” classification. “B0” is proposals in all architectures.

Cascade R-CNN扩展了Faster-RCNN的two-stage架构，如上图（a）所示.第一阶段是sub-network (提议子网) (“H0”), 应用于整个图像，以产生初步检测假设，称为对象提议。在第二阶段，这些假设然后由region-of-interest(感兴趣区域)检测子网络(“H1”)处理，表示为检测头.最终的分类分数(“C”)和边界框(“B”)分配给每个假设。作者专注于对多阶段检测子网络进行建模，并采用（但不限于）用于提案检测的RPN。

1、Cascaded Bounding Box Regression

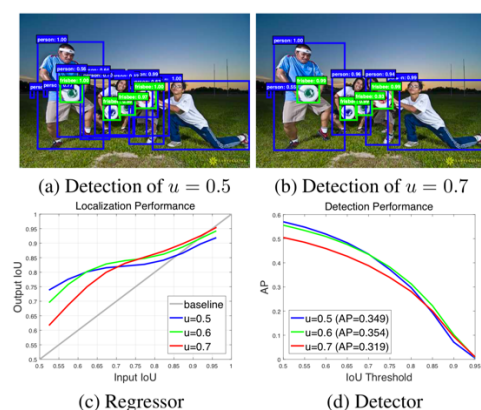


Figure 1. The detection outputs, localization and detection performance of object detectors of increasing IoU threshold u .

如上图C所示要求单个回归器在所有质量水平上均匀地执行是非常困难的。作者在cascade pose regression(级联姿势回归)和face alignment(面部对齐)受到启发，在Cascade R-CNN中，它被构建为级联回归问题，如上图D所示。这依赖于一系列专门的回归量：

$$f(x,b)=f_T \circ f_{T-1} \cdots \circ f_1(x,b),$$

T 是级联阶段的总数。注意每个回归者 f_t 到达相应阶段的分布 $\{b^t\}$ ，而不是 $\{b^1\}$ 的初始分布,这个级联逐渐改进了假设。

级联回归特点：（1）不同于上上图3（b）的迭代BBBox架构，虽然迭代BBBox是用于改进边界框的后处理过程，但是级联回归是一种重新采样过程，它改变了由不同阶段处理的假设的分布。

（2）因为它在训练和推理中都使用，所以训练和推理分布之间没有差异

（3）多个专用回归器 $\{f_T, f_{T-1}, \dots, f_1\}$ 针对不同阶段的重采样分布进行了优化

2、Cascaded Detection

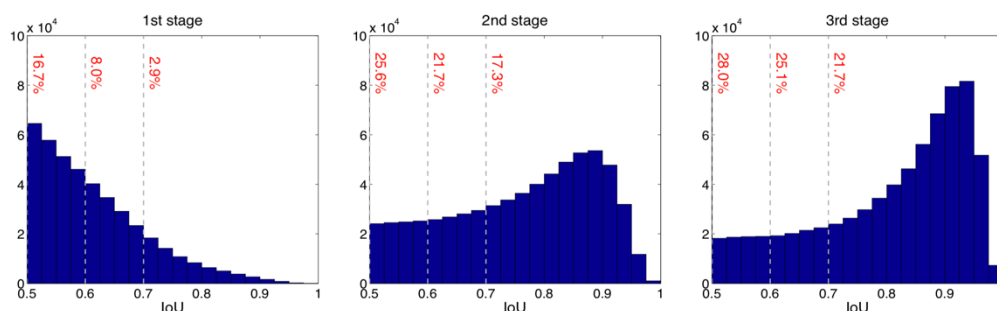


Figure 4. The IoU histogram of training samples. The distribution at 1st stage is the output of RPN. The red numbers are the positive percentage higher than the corresponding IoU threshold.

Cascade R-CNN依靠级联回归作为重采样机制来解决对高质量分类器的学习效率低下的问题。R-CNN包括分类器 h_t 和针对IoU阈值 u^t 优化的回归器 f_t ，其中 $u^t > u^{t-1}$ 。通过最小化损失来保证这一点

$$L(x^t, g) = L_{cls}(h_t(x^t), y^t) + \lambda[y^t \geq 1]L_{loc}(f_t(x^t, \mathbf{b}^t), \mathbf{g}), \quad (8)$$

其中 $\mathbf{b}^t = f(x^{t-1}, \mathbf{b}^{t-1})$,对于 x^t, \mathbf{g} 是基础事实对象, $\lambda=1$,是权衡系数, $[\cdot]$ 指标函数, y^t 是由 (5) 给出的 x^t 的标签。

五、实验

Baseline Models : Resnet

Pretrained Model: detnet59

数据集 : VOC2007

实验环境

- Python 2.7 or 3.6

- Pytorch 0.2.0 or higher (not support pytorch version $\geq 0.4.0$)
- CUDA 8.0 or higher
- tensorboardX

由于代码经常运行一段时间出现 **out of memory** (后来找到了问题, 当时跑代码的人比较多, 显卡内存不够), 训练时间很长, 最后设置只跑两个批次。耗时 2 个小时。结果如下:

1、设置不同的 Overthresh 的结果

Overthresh: [0.5 , 0.55, 0.6 , 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]

mAPs: [0.49306032613836004, 0.44324736404155096,
0.3841301218260791, 0.3149583616326112, 0.24752565751188493,
0.18052016697688344, 0.1133202322684336, 0.04720416135506957,
0.011468298437253832, 0.0016651352308196528]

mean mAP: 0.22370998254189461

2、Test Result

测试部分图片结果 :



作者给出的结果

| | backbone | AP | AP ₅₀ | AP ₇₅ | AP _S | AP _M | AP _L |
|----------------------------|--------------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| YOLOv2 [26] | DarkNet-19 | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD13 [23] | ResNet-101 | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| RetinaNet [22] | ResNet-101 | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| Faster R-CNN+++ [16]* | ResNet-101 | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPN [21] | ResNet-101 | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Faster R-CNN w FPN+ (ours) | ResNet-101 | 38.8 | 61.1 | 41.9 | 21.3 | 41.8 | 49.8 |
| Faster R-CNN by G-RMI [17] | Inception-ResNet-v2 | 34.7 | 55.5 | 36.7 | 13.5 | 38.1 | 52.0 |
| Deformable R-FCN [5]* | Aligned-Inception-ResNet | 37.5 | 58.0 | 40.8 | 19.4 | 40.1 | 52.5 |
| Mask R-CNN [14] | ResNet-101 | 38.2 | 60.3 | 41.7 | 20.1 | 41.1 | 50.2 |
| AttractionNet [10]* | VGG16+Wide ResNet | 35.7 | 53.4 | 39.3 | 15.6 | 38.0 | 52.7 |
| Cascade R-CNN | ResNet-101 | 42.8 | 62.1 | 46.3 | 23.7 | 45.5 | 55.2 |

Table 5. Comparison with the state-of-the-art *single-model* detectors on COCO test-dev. The entries denoted by "*" used bells and whistles at inference.

六、总结

在本文中，作者提出了一个多级对象检测框架，Cascade R-CNN，用于设计高质量的物体探测器。该结构被证明可以避免在训练中过度拟合和推理质量不匹配的问题。并且在具有挑战性的 COCO 数据集上的可靠且一致的检测改进表明，需要对各种同时因素进行建模和理解以推进物体检测。Cascade R-CNN 被证明适用于许多物体检测架构。论文其实没怎么看懂，但是代码还是能理解，基于 detnet 网络进行的改进。