

Anchor-free第三篇-CornerNet-Lite 论文总结

作者：小哲

微信公众号：小哲AI

github: <https://github.com/lxztju/notes>

Anchor-free第三篇-CornerNet-Lite 论文总结

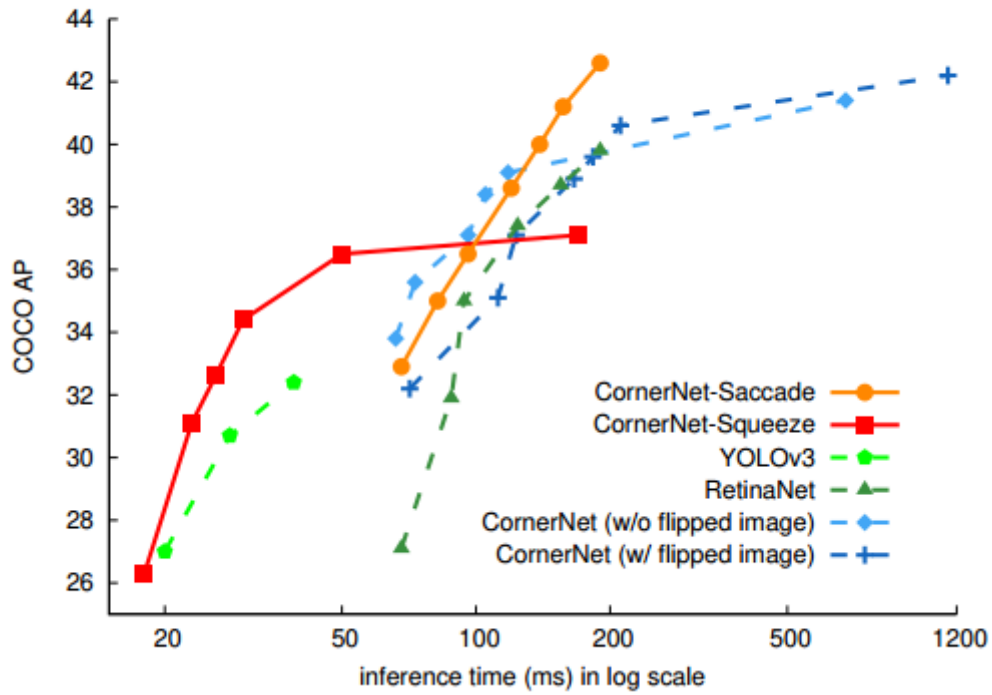
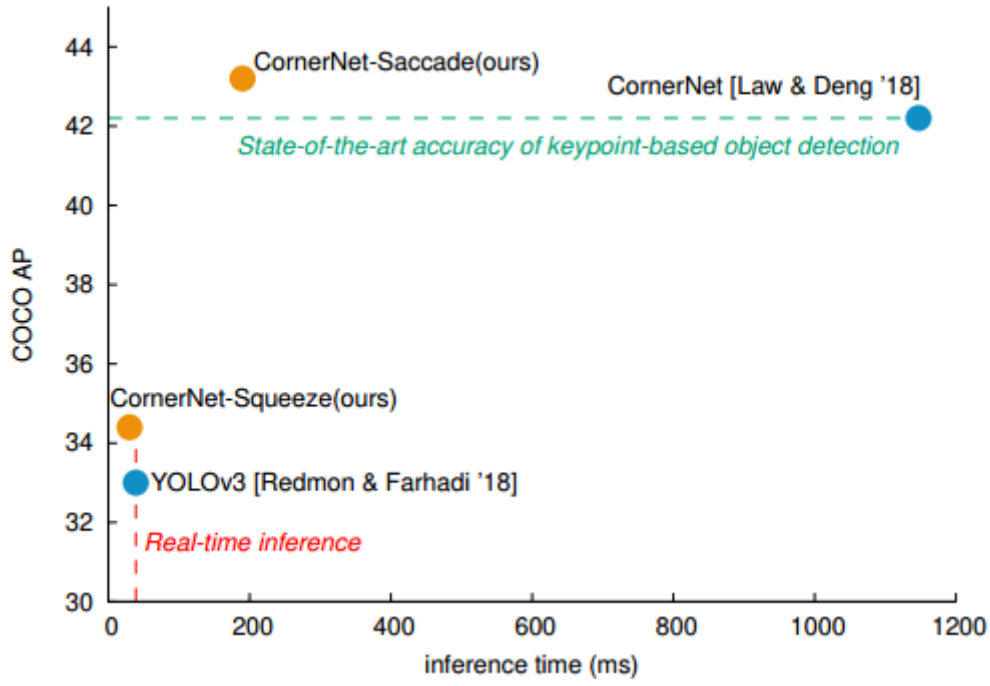
1. 论文摘要
2. 算法实现效果
3. 论文主要思想及创新点
4. CornerNet-Saccade
5. CornerNet-Squeeze

1. 论文摘要

基于关键点的方法是一个新型的目标检测范式，这种方法移除了anchor boxes的设计并且提供了一个简单的目标检测框架。基于吧关键点检测的目标检测算法已经实现了目前最优的单步法检测结果。可是对于cornernet来说搞得准确率需要很高的计算价值。在本文的文章中，我们解决了这个问题并且提出了一个高效的目标检测方法——cornernet-lite: CornerNet-Saccade, CornerNet-Squeeze.其中前者采用注意及机制来限制图像中所有像素的处理的大量消耗。后者引入了新的backbone架构。利用这两个变种解决了两种关键的使用场景：不损失准确率的情况下提升高效性，在实时高效的情况下提升准确率。

CornerNet-Saccade适合应用于离线的处理。效率相较于cornernte提升了6倍，Ap提升了1%，CornerNet-Squeeze适用于实时的目标检测，相较于YOLOv3在准确率与效率上均有提升。

2. 算法实现效果



	Time	AP	AP^s	AP^m	AP^l
YOLOv3	39ms	33.0	18.3	35.4	41.9
CornerNet-Squeeze	30ms	34.4	13.7	36.5	47.4
CornerNet (single)	211ms	40.6	19.1	42.8	54.3
CornerNet (multi)	1147ms	42.2	20.7	44.8	56.6
CornerNet-Saccade	190ms	43.2	24.4	44.6	57.3

Table 8: CornerNet-Lite versus CornerNet and YOLOv3 on COCO test set.

3. 论文主要思想及创新点

论文主要考虑提升基于关键点目标检测的效率，提出了两种cornernet的变体：

1. CornerNet-Saccade，主要考虑提升准确率，同时保证高效性，离线情况下使用，采用注意力机制减少图像中所要处理的像素的个数
2. CornerNet-Squeeze，主要提升检测速度，实时在线推理使用。采用新的backbone来减少同一个像素的处理次数

4. CornerNet-Saccade

cornernet-saccade的主要的网络架构：

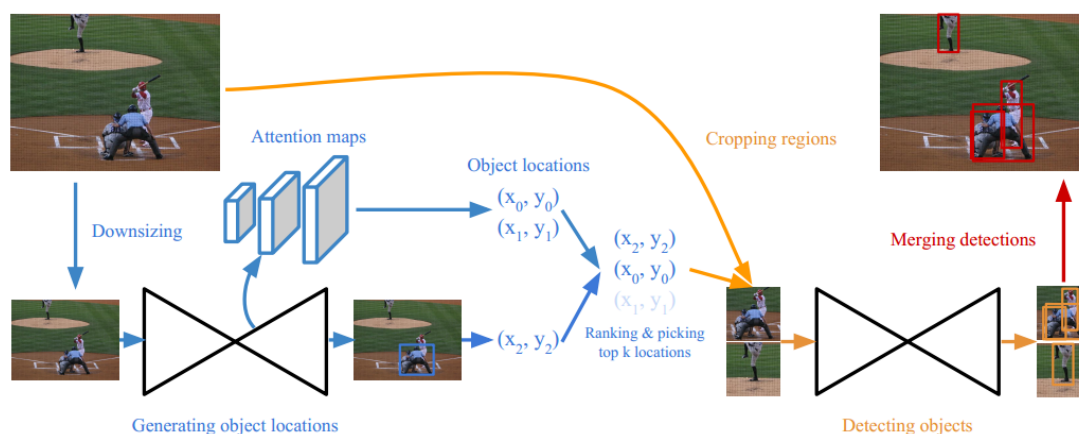


Figure 2: Overview of CornerNet-Saccade. We predict a set of possible object locations from the attention maps and bounding boxes generated on a downsized full image. We zoom into each location and crop a small region around that location. Then we detect objects in each region. We control the efficiency by ranking the object locations and choosing top k locations to process. Finally, we merge the detections by NMS.

人类视觉中的 Saccades（扫视运动）是快速的眼部移动来确定不同的图像区域。在目标检测算法中，我们广义地使用该术语来表示在推理期间选择性地裁剪（crop）和处理图像区域（顺序地或并行地，像素或特征）。

R-CNN 系列论文中的 saccades 机制为 single-type and single-object，也就是产生 proposal 的时候为单类型（前景类）单目标（每个 proposal 中仅含一个物体或没有），AutoFocus 论文中的 saccades 机制为 multi-type and mixed（产生多种类型的 crop 区域）

CornerNet-Saccade 中的 saccades 是 single type and multi-object，也就是通过 attention map 找到合适大小的前景区域，然后 crop 出来作为下一阶段的精检图片。CornerNet-Saccade 检测图像中可能的目标位置周围的小区域内的目标。它使用缩小后的完整图像来预测注意力图和粗边界框；两者都提出可能的对象位置，然后，CornerNet-Saccade 通过检查以高分辨率为中心的区域来检测目标。它还可以通过控制每个图像处理的较大目标位置数来提高效率，主要分为两个阶段估计目标位置和检测目标。

首先将输入图片通过resize下采样操作2次，分别得到2个图像，分别为长边255像素和长边192像素。然后类似faster rcnn一样，将192长边的图像上下左右补黑色像素至255大小，这样这2个图就可以并行处理，当然和faster rcnn还有一点区别，faster中是直接从左上角开始对齐，不足的地方补充黑色像素。

然后基于一个沙漏模块的编码解码模块，在解码模块的不同层上可以分别得到3个不同大小的特征图，分别预测小物体（少于32像素），中物体（32到96像素），大物体（大于96像素），最终得到3个预测的attention特征图。而在测试的时候只处理阈值大于0.3的特征图区域。

基于这3个特征图，可以得到在这3个特征图上的物体坐标和缩放尺度，基于此，还原回原图的坐标。而训练的时候是在每一个物体的中心位置点作为attention点，并使用focal loss进行训练。

然后基于得到的尚不精确的边框，根据边框得分进行排序，取前top-k的边框。

最后基于这些边框从原图crop出这些区域，再经过2个沙漏结构的网络，得到最终的预测的精确的边框坐标。最后进行soft-nms操作，得到最终的结果。

5. CornerNet-Squeeze

改进思想源自于squeezeNet与mobilenet。

与专注于 subset of the pixels 以减少处理量的 CornerNet-Saccade 相比，而 CornerNet-Squeeze 探索了一种减少每像素处理量的替代方法。在 CornerNet 中，大部分计算资源都花在了 Hourglass-104 上。Hourglass-104 由残差块构成，其由两个 3×3 卷积层和跳连接（skip connection）组成。尽管 Hourglass-104 实现了很强的性能，但在参数数量和推理时间方面却很耗时。为了降低 Hourglass-104 的复杂性，本文将来自 SqueezeNet 和 MobileNets 的想法融入到轻量级 hourglass 架构中。

主要操作是：

1. 使用SqueezeNet中的firemodule替换CornerNet中的残差模块。包括squeeze layer中的1x1卷积替代3x3卷积进行通道的降维。expand layer中的3x3+1x1卷积替代3*3卷积。
2. 使用3x3的depthwise conv替换firemodule中的传统3*3 conv

论文地址：<https://arxiv.org/abs/1904.08900>