Anchor-free第二篇-CerterNet 论文总结

> 作者： 小哲
>
> 微信公众号： 小哲AI
>
> github： https://github.com/lxztju/notes

1. 论文摘要

在目标检测中，由于缺少对于裁剪框中的内容作进一步的校验，基于关键点的方法通常会有大量的不正确目标框的问题。论文介绍了基于最小代价情况下探索裁剪框中视觉模式的方法。本文所采用的方法是基于一步法的关键点检测的目标检测方法。CenterNet检测每一个目标为一个三元组（triplet）而不是一对关键点（cornernet）。这种方法提升了精确率与召回率。相对应的，设计了两个自定义的模块，分别为cascade corner pooling和center pooling。这两个模块的作用分别为丰富左上角与右下角的信息， 提升更多中心区域的识别信息。

在MSCOCO数据集上，CenterNet实现了47.0%AP， 远远超出了之前提出的所有的一步法检测器，与此同时，CenterNet与两步法的检测器也有一定的可比性并且拥有较大的检测速度。

2. 算法实现效果

CenterNet在coco数据集上的实验效果如下表所示：

Method	Backbone	Train input	Test input	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
Two-stage:															
DeNet [40]	ResNet-101 [14]	512×512	512×512	33.8	53.4	36.1	12.3	36.1	50.8	29.6	42.6	43.5	19.2	46.9	64.3
CoupleNet [47]	ResNet-101	ori.	ori.	34.4	54.8	37.2	13.4	38.1	50.8	30.0	45.0	46.4	20.7	53.1	68.5
Faster R-CNN by G-RMI [16]	Inception-ResNet-v2 [39]	∼ 1000×600	∼ 1000×600	34.7	55.5	36.7	13.5	38.1	52.0	-	-	-	-	-	-
Faster R-CNN +++ [14]	ResNet-101	∼ 1000×600	∼ 1000×600	34.9	55.7	37.4	15.6	38.7	50.9	-	-	-	-	-	-
Faster R-CNN w/ FPN [23]	ResNet-101	∼ 1000×600	∼ 1000×600	36.2	59.1	39.0	18.2	39.0	48.2	-	-	-	-	-	-
Faster R-CNN w/ TDM [37]	Inception-ResNet-v2	-	-	36.8	57.7	39.2	16.2	39.8	52.1	**31.6**	**49.3**	**51.9**	**28.1**	**56.6**	**71.1**
D-FCN [7]	Aligned-Inception-ResNet	∼ 1000×600	∼ 1000×600	37.5	58.0	-	19.4	40.1	52.5	-	-	-	-	-	-
Regionlets [43]	ResNet-101	∼ 1000×600	∼ 1000×600	39.3	59.8	-	21.7	43.7	50.9	-	-	-	-	-	-
Mask R-CNN [12]	ResNeXt-101	∼ 1300×800	∼ 1300×800	39.8	62.3	43.4	22.1	43.2	51.2	-	-	-	-	-	-
Soft-NMS [2]	Aligned-Inception-ResNet	∼ 1300×800	∼ 1300×800	40.9	62.8	-	23.3	43.6	53.3	-	-	-	-	-	-
Fitness R-CNN [41]	ResNet-101	512×512	1024×1024	41.8	60.9	44.9	21.5	45.0	57.5	-	-	-	-	-	-
Cascade R-CNN [4]	ResNet-101	-	-	42.8	62.1	46.3	23.7	45.5	55.2	-	-	-	-	-	-
Grid R-CNN w/ FPN [28]	ResNeXt-101	∼ 1300×800	∼ 1300×800	43.2	63.0	46.6	25.1	46.5	55.2	-	-	-	-	-	-
D-RFCN + SNIP (multi-scale) [38]	DPN-98 [5]	∼ 2000×1200	∼ 2000×1200	45.7	**67.3**	51.1	29.3	48.8	57.1	-	-	-	-	-	-
PANet (multi-scale) [26]	ResNeXt-101	∼ 1400×840	∼ 1400×840	**47.4**	67.2	**51.8**	**30.1**	**51.7**	**60.0**	-	-	-	-	-	-
One-stage:															
YOLOv2 [32]	DarkNet-19	544×544	544×544	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
DSOD300 [34]	DS/64-192-48-1	300×300	300×300	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0
GRP-DSOD320 [35]	DS/64-192-48-1	320×320	320×320	30.0	47.9	31.8	10.9	33.6	46.3	28.0	42.1	44.5	18.8	49.1	65.0
SSD513 [27]	ResNet-101	513×513	513×513	31.2	50.4	33.3	10.2	34.5	49.8	28.3	42.1	44.4	17.6	49.2	65.8
DSSD513 [8]	ResNet-101	513×513	513×513	33.2	53.3	35.2	13.0	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4
RefineDet512 (single-scale) [45]	ResNet-101	512×512	512×512	36.4	57.5	39.5	16.6	39.9	51.4	-	-	-	-	-	-
CornerNet511 (single-scale) [20]	Hourglass-52	511×511	ori.	37.8	53.7	40.1	17.0	39.0	50.5	33.9	52.3	57.0	35.0	59.3	74.7
RetinaNet800 [24]	ResNet-101	800×800	800×800	39.1	59.1	42.3	21.8	42.7	50.2	-	-	-	-	-	-
CornerNet511 (multi-scale) [20]	Hourglass-52	511×511	≤1.5×	39.4	54.9	42.3	18.9	41.2	52.7	35.0	53.5	57.7	36.1	60.1	75.1
CornerNet511 (single-scale) [20]	Hourglass-104	511×511	ori.	40.5	56.5	43.1	19.4	42.7	53.9	35.3	54.3	59.1	37.4	61.9	76.9
RefineDet512 (multi-scale) [45]	ResNet-101	512×512	≤2.25×	41.8	62.9	45.7	25.6	45.1	54.1						
CornerNet511 (multi-scale) [20]	Hourglass-104	511×511	≤1.5×	42.1	57.8	45.3	20.8	44.8	56.7	36.4	55.7	60.0	38.5	62.7	77.4
CenterNet511 (single-scale)	Hourglass-52	511×511	ori.	41.6	59.4	44.2	22.5	43.1	54.1	34.8	55.7	60.1	38.6	63.3	76.9
CenterNet511 (single-scale)	Hourglass-104	511×511	ori.	44.9	62.4	48.1	25.6	47.4	57.4	36.1	58.4	63.3	41.3	67.1	80.2
CenterNet511 (multi-scale)	Hourglass-52	511×511	≤1.8×	43.5	61.3	46.7	25.3	45.3	55.0	36.0	57.2	61.3	41.4	64.0	76.3
CenterNet511 (multi-scale)	Hourglass-104	511×511	≤1.8×	**47.0**	**64.5**	**50.7**	**28.9**	**49.9**	**58.9**	**37.5**	**60.3**	**64.8**	**45.1**	**68.3**	**79.7**

Table 2: Performance comparison (%) with the state-of-the-art methods on the MS-COCO test-dev dataset. CenterNet outperforms all existing one-stage detectors by a large margin and ranks among the top of state-of-the-art two-stage detectors.

3. 论文主要思想及创新点

基于anchor的方法的不足

1. 需要大量的anchor导致负样本的数量远远大于正样本的数目，导致正负样本的极度不均衡，这是目标检测相关方法的一个极大的痛点
2. anchor的使用引入了大量的超参数，例如anchor的数目，尺寸大小，长宽比等需要手动设计
3. anchor通常与ground-truth不能对齐，这不利于分类任务的进行。

cornernet方法的缺陷

1. cornernet使用左上右下两个关键点来检测物体，虽然解决了基于anchor搭的弊端，但是依然存在大量的不足
2. 由于cornernet缺少对物体全局信息的观察，检测出大量不正确的bbox，尤其是当iou较小时，这种情况更加严重。也即是说每个物体由一对关键点检测得到，导致算法对bbox目标框边界比较敏感。

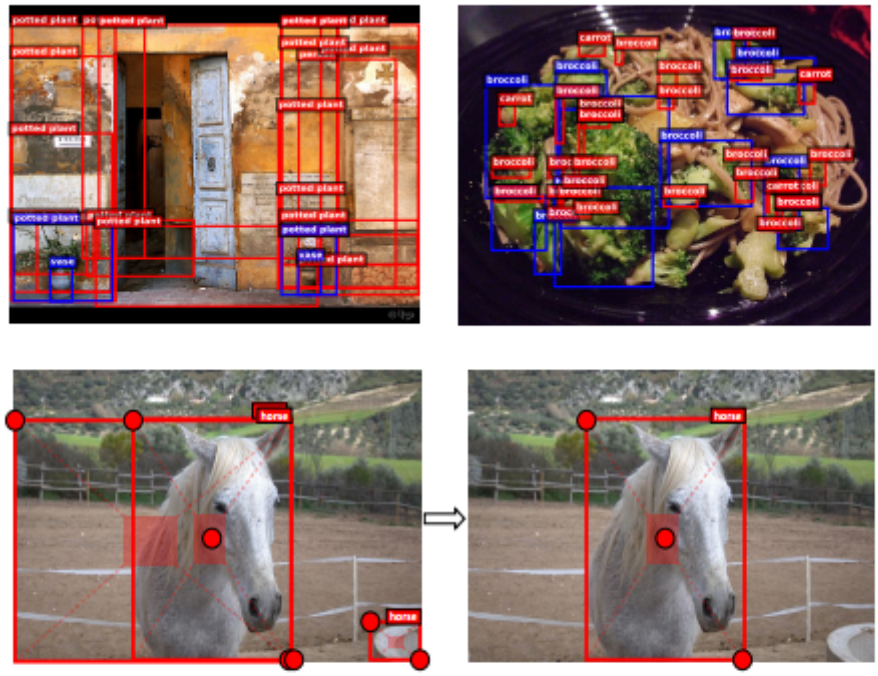


Figure 1: In the first row, we visualize the top 100 bounding boxes (according to the MS-COCO dataset standard) of CornerNet. Ground-truth and predicted objects are marked in blue and red, respectively. In the second row, we show that correct predictions can be determined by checking the central parts.

创新点：

1. 使用三元组的三个关键点（左上角右下角中心点）来解决在cornernet中出现的大量不正确的边界框的问题。利用一个额外的关键点来获取建议区域的中心区域，思想就是如果一个预测的bbox与GT有大的iou那么其中心点在预测框的中心区域预测同样类别的得分就会很高。通过检测是否有同类别的中心点出现在器中信区域。

2. centerpooling： 在预测中心关键点的分支网络中使用。Center pooling帮助中心关键点获取更多目标中可识别的视觉信息，对proposal中心部分的感知会更容易。通过在预测中心关键点的特征图上，对关键点响应值纵向和横向值的求和的最大值来获取最大响应位置。
3. cascade corner pooling使得原始corner pooling 模块具有感知内部信心的能力。我们通过在特征图上获得目标边界和内部方向最大响应值的和来预测角点。

4. 论文架构

论文整体的架构：

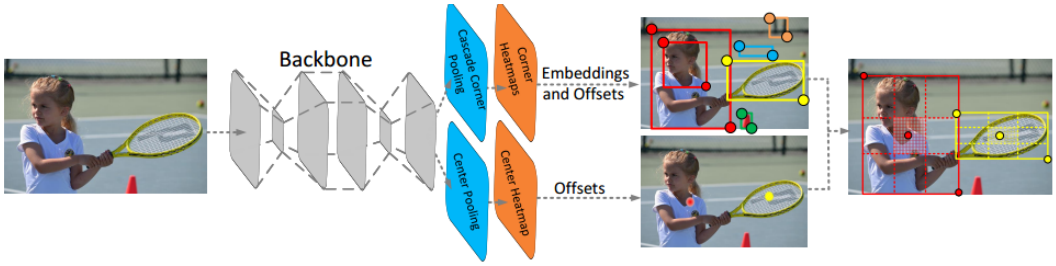


Figure 2: Architecture of CenterNet. A convolutional backbone network applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Similar to CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes.

论文使用corner net的方法作为baseline， cornernet会检测出左上与右下两个heatmaps，embedding来评估两个关键点是否属于同一个object，offset来将角点从heatmap映射回原始图像。根据得分值大小，**从热力图上分别获得左上角点和右下角点的前top-k个来生成目标框。计算一对角点的向量（embedding vector）距离来确定这对角点是否来自同个目标。当一对角点的距离小于一个特定阈值，即生成一个目标框，目标框的置信度是这对角点热力值的平均值**。

centernet利用一个中心关键点与一对角点来检测目标物体， 整合center keypoint的heatmap到cornernet的baseline中， 然后生成topk个bbox，采用如下的过程来滤除大量的不正确的bbox。

1. 按照得分的高低，选择topk的center 关键点
2. 使用对应的offsets将中心点映射回原图
3. 为每一个bbox定义一个中心区域并检查这个区域是否包含一个中心点（中心区域与中心点的类别一致）。如果一个区域包含一个中心点，那么这个bbox就会被保存，得分就是三个关键点的得分平均值。如果该区域不包含中心点，那么该bbox就会被移除。如下图：

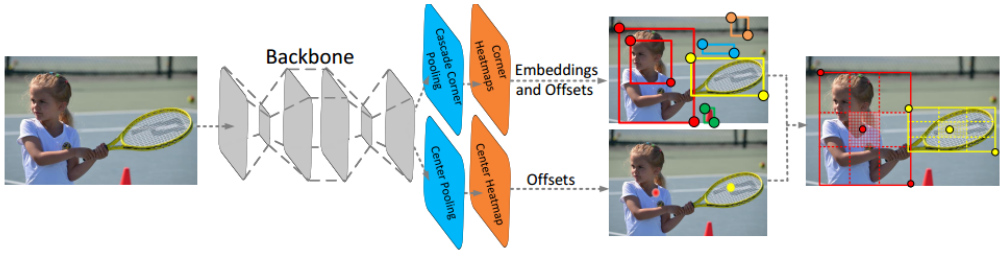


Figure 2: Architecture of CenterNet. A convolutional backbone network applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Similar to CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes.

5. 论文的细节

5.1 自适应的中心区域的大小

bbox中选取的中心区域大小影响了检测的结果，如果中心区域小，那么检测结果就回更加准确那么就会有更低的召回率， 如果中心区域较大，那么召回率就回更高，但是准确率就会下降，因此对于这个问题采用自适应的中心区域尺寸。

1. 对于小物体生成较大的中心区域（相对应的大小）
2. 对于大物体生成较小的中心区域

公式如下：

$$
\begin{cases}
\mathrm{ctl_x} = \dfrac{(n+1)\mathrm{tl_x} + (n-1)\mathrm{br_x}}{2n} \\
\mathrm{ctl_y} = \dfrac{(n+1)\mathrm{tl_y} + (n-1)\mathrm{br_y}}{2n} \\
\mathrm{cbr_x} = \dfrac{(n-1)\mathrm{tl_x} + (n+1)\mathrm{br_x}}{2n} \\
\mathrm{cbr_y} = \dfrac{(n-1)\mathrm{tl_y} + (n+1)\mathrm{br_y}}{2n}
\end{cases}
$$

其中t表示top-left， b表示bottom-right， c表示中心区域的左上与右下角，在论文实验中n=3（bbox小于150）和5（大于150）.

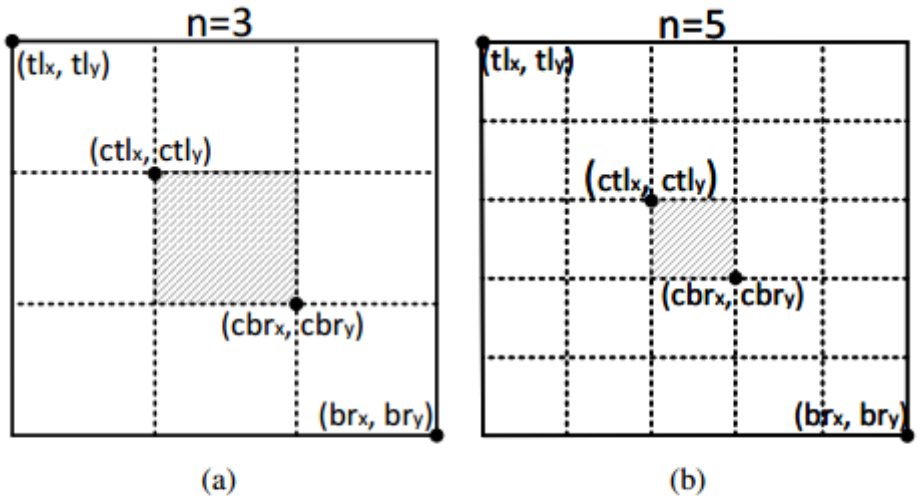
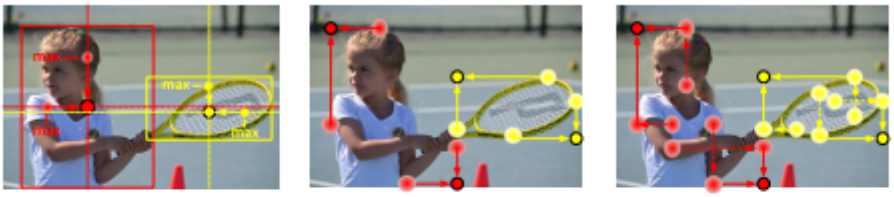


Figure 3: (a) The central region when $n = 3$. (b) The central region when $n = 5$. The solid rectangles denote the predicted bounding boxes and the shaded regions denote the central regions.

5.2 center pooling

物体的几何中心通常不会传递视觉可识别的较强的视觉模式（visual pattern），例如一个人的几何中心是在身体上，而头部具有较强的可是别视觉信息。为了解决这个问题，采用center pooling， 处理过程为：首先backbone会输出一个特征图，为了确定特征图上的某个像素是否为中心关键点，需要在水平和垂直方向寻找最大值，并且将最大值相加。 center pooling有利于更好检测中心关键点。

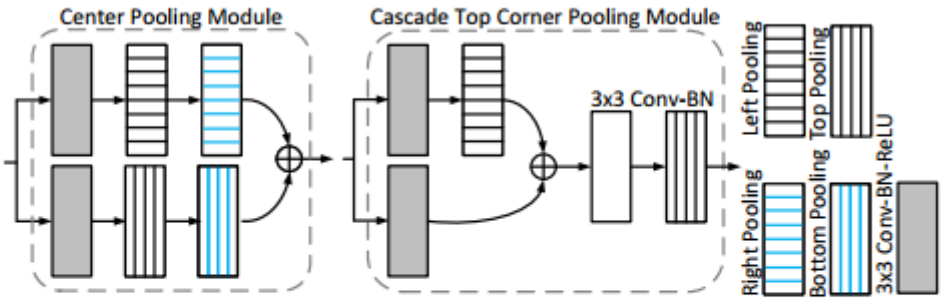


(a) (b) (c)

Figure 4: (a) Center pooling takes the maximum values in both horizontal and vertical directions. (b) Corner pooling only takes the maximum values in boundary directions. (c) Cascade corner pooling takes the maximum values in both boundary directions and internal directions of objects.

5.3 cascade corner pooling

通常情况下，角点存在于物体之外，缺乏局部外观特性。CornerNet用corner Pooling来解决此问题，通过沿边界方向找到最大值从而确定角点。但是其使得角点对边界特别敏感（因为是针对边界的特征信息做的池化操作，受边界信息变化影响较大）。为此，**本文作者提出让角点可看到更多的目标视觉模式信息（即获取物体内部的信息，而不仅仅是边界的），见图4（c），原理是沿边界寻找最大值，根据边界沿物体内部方向寻找最大值，将两个最大值相加。该方法获取的角点 带有边界信息以及物体内部的数据模式信息**。



(a) (b)

Figure 5: The structures of the center pooling module (a) and the cascade top corner pooling module (b). We achieve center pooling and the cascade corner pooling by combining the corner pooling at different directions.

论文地址：https://arxiv.org/pdf/1904.08189.pdf