处理的像素数量来加快处理时间, 但是这样会导致精度的大幅度下降; 在全局性上, CornerNet 缺乏对物体全局信息的考虑, 即因为每个目标物都有两个角点, 算法对识别目标物的边界框很敏感, 同时又无法确定哪两个角点属于同一个物体, 所以会产生一些错误的边框. 因此, 对于这些问题, Law 等[8] 提出 CornerNet-Lite 的高效目标检测和 Duan 等[9] 提出 CenterNet 的关键点三元组进行解决.

1.2 CornerNet-Lite

CornerNet-Lite 的高效目标检测是在 CornerNet 的基础上进行的优化, 是 Cornernet 的两个高效变体的组合, 优化包括两点: (1) CornerNet-Saccade 使用类似于人眼的注意力机制消除了对图像的所有像素进行处理的需要, 通过减少像素个数来提高检测速度. CornerNet-Saccade 可以用于线下处理, 不用牺牲精度也可提升速度. CornerNet-Saccade 是第一个在基于关键点的目标检测方法中使用 Saccade 的; (2) 引入了新的紧凑骨干架构 CornerNet-Squeeze, 通过减少每个像素的处理量来加速图像处理. 它融合了 SqueezeNet[10] 和 MobileNet[11] 的思想, 并引入了一种新的紧凑型沙漏骨干 (54 层, CornerNet 的沙漏骨干 104 层), 广泛使用 1×1 卷积, 瓶颈层和深度可分离卷积[12]. CornerNet-Squeeze 可用于实时处理, 提升精确度而无需牺牲速度, 是第一篇把 Squeeze 和沙漏网络组合用于目标检测的文章.

CornerNet-Saccade 的算法步骤: 第一步是获取图像中可能的目标位置. (1) 先把原图进行缩小和裁剪; (2) 将缩小的完整图像输入到骨架网络 (沙漏网络: 卷积、下采样和卷积、上采样) 中预测 attention maps 和检测缩小后的图像中的目标并生成粗边框 (两者都提出可能的对象位置). 通过使用不同尺度的特征图来预测 3 个 attention maps, 用于小、中、大物体; (3) 从预测的 attention maps 和粗边框中得到可能的目标位置. 第二步是检测目标. (1) 对第一步 (3) 的可能位置中选取前 k 个位置, 把这 k 个位置与对原图裁剪得到的图片进行对应, 得到在可能的位置处检测到目标; (2) 对检测的结果基于 soft-NMS 进行处理, 处理方式与 CornerNet[4] 一样, 从而得到目标物的边界框; (3) 利用得到的边界框的尺寸来确定目标所在图像的缩放大小, 进行目标的合并且大小与原图一致. 在训练时, 采用与 CornerNet 相似的训练损失来训练网络以预测角点热图、嵌入和偏置. CornerNet-Saccade 的流程图如图 2 所示.

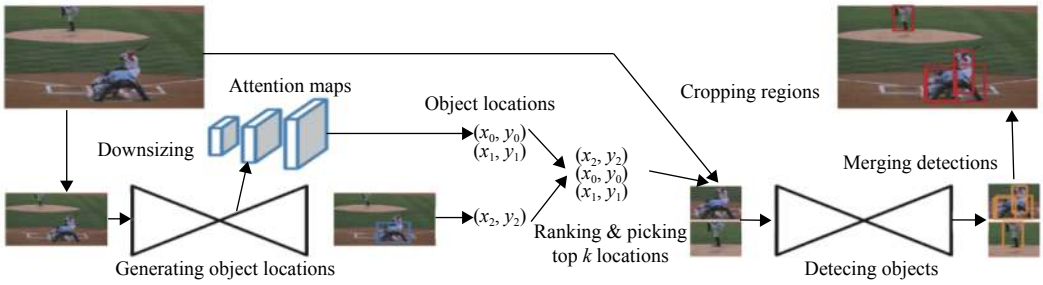

图 2 CornerNet-Saccade 的流程图[8]

CornerNet-Squeeze 是 CornerNet-Lite 研究的另一个方案, 降低每个像素点上处理成本. 在 CornerNet 中, 大多数的计算都耗费在 Hourglass-104[4,5] 上, 非常耗时. 为了降低 Hourglass-104 网络的复杂度, 引入 Squeeze 和 MobileNet 的思想[10,11], 设计了一个轻量级的 Hourglass-52. 训练 CornerNet-Squeeze 使用了与 CornerNet 中一样的损失函数和超参数, 唯一不同的是 batch size.

Squeeze 提出了 3 种降低网络复杂策略: (1) 用 1×1 卷积替换 3×3 卷积, 减少输入通道的数量; (2) 减少 3×3 卷积的输入通道; (3) 下采样后延 (此文没用到), 沙漏网络是对称的, 延迟下采样会在上采样期间产生更高分辨率的特征图, 再执行卷积, 会增加计算量.

由文献 [6] 实验结果可知: (1) 在 COCO 数据集上 CornerNet-Saccade 的计算速度比 CornerNet 提升了 6 倍, 且精度提高了 1.0%, 这说明 CornerNet-Saccade 在追求高准确率的同时, 速度也得到了很大的提高;

(2) 在 COCO 数据集上 CornerNet-Squeeze 的 34 ms 时有 34.4% 的准确度, 而 YOLOv3 的 33 ms 时有 33.0% 的准确度, 提高了当前流行的实时检测器 YOLOv3 的速度和准确度.

2 基于中心点的 anchor free 目标检测算法

基于中心点的目标检测方法是对特征图的每个位置预测它是目标中心点的概率, 并且在没有锚框先验的情况下进行边框的预测. 基于中心点的 Anchor free 目标检测模型主要有 CenterNet (中心点和角点) 和 CenterNet (中心点).

2.1 CenterNet (使用中心点和角点)

由于 CornerNet 缺乏对物体全局信息的考虑, 通常会遭受大量不正确的目标边框的困扰, 为解决该问题, Duan 等[9] 提出了 CenterNet 的用于目标检测的关键点三元组方法, 即在 CornerNet 的基础上加入一个中心关键点的热图, 并预测了中心点的偏置. 相较于 CornerNet, CenterNet 关键点三元组方法进行了如下优化: (1) 确定中心区域的大小, 能更准确的定位目标; (2) 为了增强中心点和角点信息, 提出了中心池化和级联角点池化 (丰富了边界和内部信息, 而 CornerNet 的角点池化只有边界信息). 该方法提高了准确性和查全率. 在 COCO 数据集上, CenterNet 的 AP 达到了 47.0%, 比现有的一级检测器至少好 4.9%.

算法步骤: 该方法的检测步骤与 CornerNet[4] 的步骤类似, (1) 在 CornerNet 的步骤 (1) 中再预测一个中心热图, 在热图上采用非极大值抑制, 选择前 k 个中心点. 该方法中引入的是级联角点池化和中心点池化; (2) 在 CornerNet 的步骤 (2) 中添加一个中心偏置; (3) 两个角点用来检测潜在的目标框, 然后用中心区域判断中心点是否在中心区域内, 最后确定最终的边界框的位置. CenterNet 的流程图如图 3 所示.

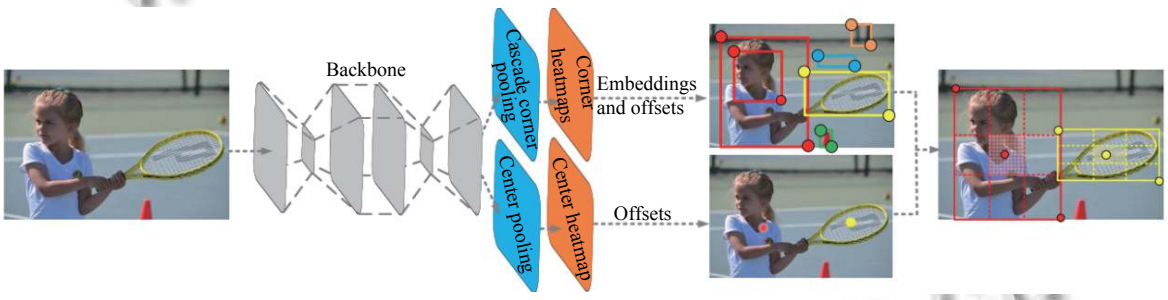

图 3 CenterNet 的流程图[9]

由文献 [9] 实验结果可知, 该算法的 AP 相较于 CornerNet 有很大的提高. 但是, 基本上所有的目标检测器都会把潜在的目标位置列举出来, 并对每一个物体进行分类. 这样做是非常浪费时间、低效的, 而且还需要额外的后处理, 如非极大值抑制等, 使网络变得很复杂. 而 Zhou 等提出的 CenterNet[13], 即将目标当作中心点来检测不需要这些步骤, 极大提高了检测速度. 虽然不需要进行非极大抑制等操作, 但 CornerNet 在完成关键点检测后, 还需要将左下角关键点和右上角关键点进行两两匹配, 导致检测速度有所下降.

2.2 CenterNet (只使用中心点)

Zhou 等[13] 提出的基于 CenterNet 的将目标作为点的方法规避了低效和额外的后处理等缺点. 该方法通过其边界框中心的单个点来表示所检测的目标, 然后可以直接从中心位置的图像特征回归其他属性, 如目标大小、尺寸、3D 位置、方向甚至姿势. 因此, 基于中心点的方法相对于其他基于边界框的目标检测器来说, 其具有更简单、速度更快和准确度更高的特点. 图 4 所示为在 COCO 数据集上该方法与基于框的几种方法在速度和准确度上的比较.

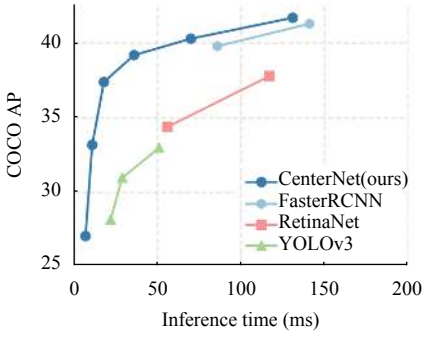

图 4 几种方法比较[13]

该算法的流程如下: 首先, 将图像输入完全卷积网络生成热图, 该热图中的峰值对应于目标中心. 其次, 将输出的热图中每个类别的峰值点单独提取出来, 即热图上的所有点与其相连的 8 的临近点比较, 若该点的值不小于其 8 个临近点, 则该点保留, 最后保留满足所有要求的前 100 个峰值点. 最后, 对每个峰值处的图像特征预测得到目标框的高度和宽度信息 (根据不同的任务预测不同的属性, 如对于 3D 边框估计, 需要预测目标绝对深度、边框尺寸和方向[14]). 该训练网络是单个网络前向传递的, 因为每个目标仅有一个中心点, 所以没有非极大值抑制等后处理.

将目标作为点的方法对于上述几个基于关键点估计[15] 的目标检测器来说, 不同之处在于: (1) 该方法不需要在关键点检测之后进行组合分组, 这明显提高了算法的计算速度; (2) 该方法仅仅需要为每个目标预测单个中心点, 而不需要进行非极大值抑制后处理; (3) 损失函数中不仅有关键点损失 (焦点损失[3]) 和偏置损失, 还加入了目标大小损失, 这样可以提高算法的准确度.

由文献 [8] 实验可得, 该方法在 COCO[16] 数据集上实现了速度与准确度的权衡, 其中 AP 为 28.1% 的模型处理速度达到了 142 FPS, AP 为 37.4% 的的模型处理速度达到了 52 FPS, AP 为 45.1% 的模型处理速度达到了 1.4 FPS. 该方法通过预测一个中心点解决了 2D 和 3D 目标检测, 以及姿态估计, 把这 3 个任务很完美的统一到了一起 (在这里主要综述了 2D 目标检测). 虽然该方法简单、快速、准确, 但是在训练过程中, 如果两个不同的目标重叠了, 共享同一个中心点, 在这种情况下, CenterNet 会把这两个目标当成一个目标来训练, 只会检测其中一个, 这样就会存在大量的误检, 这是该方法的一个不足.

3 基于全卷积的 anchor free 目标检测

Tian 等[17] 提出的 FCOS 是一种基于全卷积的单级目标检测器, 是像素级别的目标检测, 其主要思想类似于语义分割. 该方法不需要锚框, 因此其完全避开了锚框的缺点. FOCS 仅凭借后处理非极大值抑制, 该方法优于之前的基于锚框的一级探测器, 其优势在于更简单、灵活, 可以提高检测的精度. 算法步骤如下: 首先, 对输入的原始图像进行预处理操作; 然后将预处理之后的数据送入主干网络中获取输入数据的 feature map, 对获得的 feature map 上的每一像素点进行回归操作, 对网络进行训练以获得网络模型; 再将得到的模型用于测试, 利用特征金字塔网络进行多级预测, 从而得到多个 head, 从多个 head 中可以获得预测的结果; 最后使用非极大值抑制后处理获得最终的检测结果. FCOS 的流程图如图 5 所示.

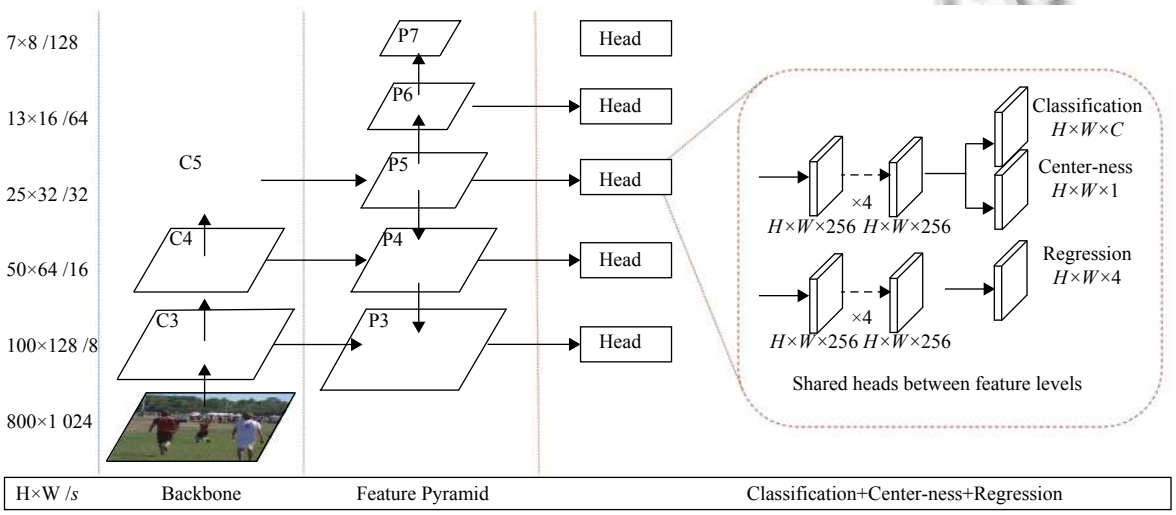

图 5 FCOS 的流程图[17]

FCOS 的新颖之处: (1) 在远离目标中心的位置上会产生一些不好的框, FCOS 中为了降低这些不好的结果, 引入了 Center-ness 分支, 即利用中心分支来抑制低质量预测边界框; (2) 在网络中, 除了最后的预测层, 在卷积层中加入了组标准化 GN, 使得训练更加稳定; (3) FCOS 中, 特征金字塔网络中利用 P5 生成 P6 和

P7, 而不是用骨干网络的 C5, 这使得网络的性能得到略微的提高.

根据文献 [10] 实验结果可知, 相比于 CornerNet[4], FCOS[17] 在 mAP 上有 0.5% 的提高. 虽然 FCOS 性能有略微的提高, 但是相对于 CornerNet 来说是有优势的, 如其主干网络是 ResNet-101 而不是 Hourglass-104[5], 该网络更快更简单; 除了非极大值抑制, 不需要其他的后处理, 而 CornerNet 还需要将具有嵌入向量的角点分组等. 实验也表明了, FCOS 可以检测到各种各样的物体, 包括拥挤、遮挡、高度重叠、极小和极大的物体.

4 算法性能比较

基于深度学习的目标检测算法可分为两大类: 基于锚框的目标检测和基于 Anchor free 的目标检测. 目标检测中非常重要的两个性能: 精度和速度. 本文在 VOC2007 和 COCO 数据集上, 分别对这两类目标检测算法进行了性能上的比较和分析.

4.1 基于锚框的目标检测算法比较

基于锚框的目标检测算法分为两种: two stage 检测算法和 one stage 检测算法. two stage 检测算法中典型的目标检测算法有: R-CNN、Fast R-CNN、Faster R-CNN、Mask R-CNN 等. 从 R-CNN 算法到 Mask R-CNN 算法, 它们依次不断地在检测的准确度和速度上进行改进. 但是, 这些算法的实时性很差, 难以满足实际的需求. 表 1 总结了 two stage 的目标检测算法在 VOC2007 和 COCO 数据集上的性能, “—”表示没有相关的数据.

表 1 Two stage 的目标检测算法性能比较

网络结构	VOC2007 mAP (%)	COCO mAP (%)	检测速度(FPS)
R-CNN	58.5	—	—
SPP-Net	59.2	—	2
Fast R-CNN	70.0	19.7	3
Faster R-CNN	73.2	21.9	5
R-FCN	79.5	29.9	6
Mask R-CNN	—	37.1	5

由表 1 数据可以得出, 在 two stage 目标检测算法中, 每个算法在精确度上都是从前往后不断地提高, 而检测的速度没有达到这样的特性, 也就是说在保证精度的前提下, 实时性没有得到显著提高. 因此, 研究者们提出了基于回归的目标检测算法, 即 one stage 目标检测算法.

One stage 检测算法中典型的算法有 YOLO 系列、SSD 及 RetinaNet 等. 该类算法虽然在实际中的应用成为可能, 但是实时性和准确度还有待提高. 表 2 罗列了 one stage 算法在 VOC2007 和 COCO 数据集上的性能, “—”表示没有相关的数据.

表 2 One stage 的目标检测算法性能比较

网络结构	VOC2007 mAP (%)	COCO mAP (%)	检测速度 (FPS)
YOLOv1	66.4	—	45
SSD	76.8	31.2	19
YOLOv2	78.6	21.6	40
YOLOv3	—	33.0	45.4
RetinaNet	—	39.1	5

从表 2 的数据可以得到, one stage 算法不但在精确度上有所提高, 而且检测的速度也有明显的提升. 由此可见, 相比于 two stage 算法, 其更有实用性. 但是, 在实际应用中, one stage 模型的实时性还满足不了需求. 因此, 研究者们提出了基于 anchor free 的目标检测, 此种方法的提出, 极大的满足了目标检测在实际应用中的实时性需求.

4.2 基于关键点的 anchor free 目标检测算法比较

第 1~3 节介绍的算法是基于关键点的 anchor free 的目标检测算法, 其基本思路是输入图像、提取关键点、尺度预测及位置回归的所有过程在一个卷积神经网络中实现, 不需要提前设置锚框, 极大地改善了检测的实时性, 在很大程度上满足实际应用的需求. 表 3 对本文中所列举的各类目标检测模型的机制、优点、缺点、适用范围及实现成本进行了总结. 表 4 总结了各个算法在 COCO 数据集上的性能.

从表 4 各个基于关键点的 anchor free 目标检测算法的性能比较可见, 通过逐步改进网络, 相比于基于锚框的目标检测算法, 由表 1 和表 2 所示, anchor free 目标检测算法不仅在精确度上有很大提升, 检测速度也得到明显提高. 例如 CornerNet 模型, 该算法在 COCO 数据集上准确度达到 42.1%, 超过所有基于锚框的检测算法. 在 CornerNet 上改进的 CornerNet-Lite 模型中引入 Saccade、Squeeze 来减少像素个数和降低网络复杂度, 因此精确度提高了 1% 和运算速度提高了 6 倍. 而 CenterNet 模型中, 为更好的检测中心点和角点, 该算法中提出中心点池化 (center pooling) 和级联角点池化 (cascade corner pooling), 使得在

COCO 数据集上达到 47% mAP, 超过目前所有的 one stage 检测算法, 并大幅度领先于基于关键点的 anchor free 检测模型, 其领先幅度至少为 4.9%, 但是检测时间略慢. 基于 CenterNet 的将目标作为点的模型, 相比于其他模型来说, 由于没有额外的后处理, 使得该模型更简单, 在 COCO 数据集上 mAP 达到 45.1%, 高于 YOLO、SSD、RetinaNet 等基于锚框的 one stage 模型. FCOS 模型通过对不同特征级别的像素点进行回归, 共享不同特征层之间信息, 后处理仅用非极大值抑制, 使得该模型参数效率更高、更灵活, 相比于锚框的检测, 检测精度有很大提高, 其 mAP 高达 44.7%. 由此可见, 基于关键点的 anchor free 目标检测算法具有更高的检测精确度和检测速度, 因此其更具有实用性.

表 3 各类目标检测模型总结

模型	机制	优点	缺点	适用范围	实现成本
CornerNet	一对角点(左上角点和右下角点)表示一个目标	角点池化、嵌入向量等提高了精确度.	(1)角点分组增加计算难度; (2)对边缘敏感, 忽略内部信息, 误检率高.	多目标检测	网络层数多, 导致推理速度不高
CornerNet-Lite	一对角点(左上角点和右下角点)表示一个目标	相比于CornerNet精确度和实时性有所提高.	小物体的误检率高.	多目标检测	网络层少, 内存消耗低, 运算速度快
CenterNet (中心点和角点)	中心点、左上角点和右下角点3个关键点表示一个目标	(1)级联角点池化增强了点的表征能力; (2)中心点池化表示了更多的内部信息, 消除误检.	(1)级联角点池化虽增强了点的表征能力, 但内部信息表义不明; (2)使用到的内部信息仍然不够, 特别是回归框的内部信息.	小目标检测	网络层数多, 推理时间略慢
CenterNet (只有中心点)	一个中心点+长宽值表示一个目标	简单、快速、高效、没有NMS后处理.	只使用中心点进行回归, 可获得的信息过少.	可用于2D、3D目标检测及人体姿态识别	推理仅需单个前向传播网络, 没有后处理
FCOS	点+点到框的4个距离表示一个目标	仅凭唯一后处理极大值抑制达到了更高的检测性能.	(1)召回率较低; (2) centerness虽然提高了准确率, 但是缺乏理论可解释性.	可用于实力分割、关键点检测	设计复杂度低, 内存占用少

表 4 基于关键点的 anchor free 目标检测性能比较

网络结构	COCO mAP(%)	检测速度(FPS)
CornerNet	42.1	1
CornerNet-Squeeze	34.4	33
CornerNet-Saccade	43.1	6
CenterNet(中心点和角点)	47.0	3
CenterNet(中心点)	45.1	1.4
FCOS	44.7	—

5 总结与展望

近年来, 卷积神经网络正在不断的被运用到计算机视觉领域, 尤其是在目标检测方向已经有很多基于卷积神经的目标检测模型, 极大地提高了检测的精度和运算时间. 本文主要对基于关键点的 anchor free 目标检测方法进行了综述, 根据检测关键点个数的不同, 将基于关键点的 anchor free 目标检测模型进一步分为基于角点的目标检测模型、基于中心点的目标检测模型以及基于全卷积神经网络的目标检测模型. 对每类模型的方法进行了研究、分析和对比, 总结了每类模型的思路及其优缺点, 并对基于关键点的 anchor free 目标检测算法与基于锚框的检测算法性能做了对比, 由此可以看出 anchor free 目标检测算法不论是在速度还是精度上都有非常大的改善.

目前, 在交通等各领域的发展中, 目标检测是一大研究课题, 速度和精度检测仍是重中之重, 在现有研究的基础上, 我认为该领域中研究热点及发展趋势如下:

(1) 从专注精度的 CornerNet、CenterNet、FCOS 和专注速度的 CornerNet-Squeeze, anchor free 目标检测未来的方向更加专注于精度和速度的结合. 同时为了提高目标的检测精度, 目前大多算法只是对单一时间、单一空间的信息进行融合, 研究者可以从多维度, 对多层级的信息进行融合以提升算法的准确度与鲁棒性.

(2) 目标检测依赖大量的训练样本, 在数据标注上消费很大成本, anchor free 目标检测在精度上已经有所提升, 那么如何对小规模数据的监督学习进行更有效的训练, 使其检测精度大幅度提高将会促进目标检测

检测领域的进一步发展.

(3) Anchor free 目标检测对已知类的检测已日趋成熟, 如何实现未知目标类的检测, 即从已知类别迁移到对未知类别的目标进行检测也将成为未来研究的热点.

参考文献

1 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 580–587.

2 Fu CY, Liu W, Ranga A, *et al*. DSSD: Deconvolutional single shot detector. arXiv: 1701.06659, 2017.

3 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318–327. [doi: 10.1109/TPAMI.2018.2858826]

4 Law H, Deng J. Cornernet: Detecting objects as paired keypoints. Proceedings of the 15th European Conference on Computer Vision. Munich, Germany. 2018. 734–750.

5 Newell A, Yang KY, Deng J. Stacked hourglass networks for human pose estimation. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, the Netherlands. 2016. 483–499.

6 Newell A, Huang ZA, Deng J. Associative embedding: End-to-end learning for joint detection and grouping. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, CA, USA. 2017. 2274–2284.

7 Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.

8 Law H, Teng Y, Russakovsky O, *et al*. CornerNet-Lite: Efficient keypoint based object detection. arXiv: 1904.08900, 2019.

9 Duan KW, Bai S, Xie LX, *et al*. CenterNet: Keypoint triplets for object detection. arXiv: 1904.08189, 2019.

10 Iandola FN, Han S, Moskewicz MW, *et al*. Squeezenet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv: 1602.07360, 2016.

11 Howard AG, Zhu ML, Chen B, *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861, 2017.

12 Chollet F. Xception: Deep learning with depthwise separable convolutions. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 1251–1258.

13 Zhou XY, Wang DQ, Krahenbuhl P. Objects as points. arXiv: 1904.07850, 2019.

14 Mousavian A, Anguelov D, Flynn J, *et al*. 3D bounding box estimation using deep learning and geometry. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 5632–5640.

15 Cao Z, Hidalgo G, Simon T, *et al*. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. [doi: 10.1109/TPAMI.2019.2929257]

16 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. Proceedings of the 13th European Conference on Computer Vision. Zurich, Switzerland. 2014. 740–755.

17 Tian Z, Shen CH, Chen H, *et al*. FCOS: Fully convolutional one-stage object detection. arXiv: 1904.01355, 2019.