

引用格式:胡永利,武剑,孙艳丰.一种角度自适应的椭圆模板检测器[J].中国传媒大学学报(自然科学版),2022,29(06):36-42.

文章编号:1673-4793(2022)06-0036-07

一种角度自适应的椭圆模板检测器

胡永利*,武剑,孙艳丰

(北京工业大学人工智能与自动化学院,北京 100124)

摘要:提出了一种新颖的摆脱锚框的椭圆模板检测器 EllipseNet,其能够直接预测椭圆框的圆心,长短轴长以及旋转角度。由于椭圆模板检测网络能够自适应的调整旋转角度故能够更加精确的检测出物体的轮廓。另外,为了提高模型的检测精度,设计了一种高效的椭圆交并比损失函数。实验结果表明,由于没有引入复杂的锚框,该模型结构简单,且取得了和基于矩形框目标检测算法相近的性能。

关键词:目标检测;旋转椭圆框;损失函数

中图分类号:TP751.1 **文献标识码:**A

An angle-adaptive elliptic template detector

HU Yongli*, WU Jian, SUN Yanfeng

(Artificial Intelligence Institute, Beijing University of Technology, Beijing 100124, China)

Abstract: This paper proposes a novel anchor-free object detector, namely EllipseNet. Instead of using the rectangular box, EllipseNet detects an elliptical box by predicting the center, axis and tilt angle of the ellipse. As the ellipse model can capture the object rotation and match the target adaptively by adjusting tilt angle, EllipseNet has more accurate detection boundary. In addition, we introduce an Elliptical IOU Loss to improve the detection performance of the model. The experimental results show that the model has simple structure and achieves similar performance to the object detection methods based on rectangular boxes because there is no complex anchor boxes introduced.

Keywords: object detection; oriented elliptical box; loss function

1 引言

近年来,随着深度学习的兴起,基于卷积神经网络^{[1][2]}(Convolutional Neural Networks, CNN)的深度学习算法不断刷新诸如图像识别、目标检测、语义分割、姿态估计等各类视觉任务的性能上限。相较于传统的特征提取器,卷积神经网络能够自动从图像数据中学习到各个层次的视觉特征,浅层的卷积神经网络可以提取图像的边缘、纹理等低级特征,深层的卷积

神经网络则可以提取出图像的高级抽象特征。整个网络的学习过程无需手工设计参数,与此同时,随着硬件性能的不提高,加快了卷积神经网络算法的落地和应用。

自2014年以来,涌现出了许多基于卷积神经网络的目标检测算法,这类算法大体上都包含三个部分:首先利用卷积神经网络提取图像特征,之后处理图像特征信息从而完成对物体的类别预测和定位。目前已有的优秀目标检测算法包括,基于锚框的目标检测

基金项目:国家自然科学基金(U19B2039,61906011,61632006,61772048,61672071,U1811463,61806014)

作者简介(*为通讯作者):胡永利(1973-),男,教授,主要从事计算机视觉,交通预测。Email:huyongli@bjut.edu.cn;武剑(1996-),男,硕士,主要从事目标检测研究。Email:wulele@bjut.emails.edu.cn

算法 Fast RCNN^[3]、Faster RCNN^[4]、SSD^[5]、YOLO^{[6][7][8]}等,以及摆脱锚框的目标检测算法 CenterNet-Keypoint^[9]、CenterNet^[10]、FCOS^[11]、CornerNet^[12]等。这些优秀的方法为目标检测的应用及落地创造了可能,具有很高的应用价值。

虽然上述目标检测算法均采用了不同的模型结构,但受限于数据集的影响,这些算法通常使用矩形框来进行检测。例如,现有常用的两个公开目标检测数据集 VOC^[13]、COCO^[14]均采用矩形框进行标注。如图 1(a)所示,如果检测对象存在倾斜的角度,那么矩形框将会包含很多背景区域。在提取图像特征的过程中,由于目标物体包含了大量的背景区域,势必会对网络的分类性能造成影响。于是近些年在遥感和文本图像检测方面,出现了一些基于旋转矩形框的目

标检测算法,例如 R3Det^[15]、RoI Transformer^[16]、GlidingVertex^[17]、RRPN^[18]等。旋转矩形框改善了矩形框的问题,但在面对自然图像中复杂的物体时候,依然会包含较多的背景区域,如图 1(b)所示。另外,这类算法由于需要额外学习一个倾斜角度参数,导致锚框的使用成倍增加,进而使计算复杂度大幅上升。因此,设计一种能够在不同场景、姿态和方向上均能准确检测物体边界的模型是非常价值和现实意义的工作。如图 1(c)所示,若采用旋转椭圆框进行检测则能够更好地适配物体的轮廓。另外据我们所知,目前并没有以椭圆框进行标注的自然场景下的数据集。因此,我们希望通过构建一个以旋转椭圆框标注的自然场景下的数据集,推动越来越多的研究者研究椭圆框目标检测器。



图 1 不同形式检测框

2 椭圆模板检测器

2.1 符号表示

以 $E = [cx, cy, la, sa, \theta, n^{(i)}] \in \mathbb{R}^6$ 来表示旋转椭圆框。其中令 $C = [cx, cy]$ 表示椭圆的圆心, $A = [la, sa]$

表示椭圆的长短轴长, θ 表示椭圆的旋转角度, $n^{(i)}$ 表示当前的物体属于第 i 个类别。模型总体结构如图 2 所示,总共包括四个部分:骨架网络 ResNet^[19]主要提取图像的多级特征 C3, C4 和 C5。然后通过特征金字塔塔结构^[20](Feature Pyramid Network, FPN)获得融合后

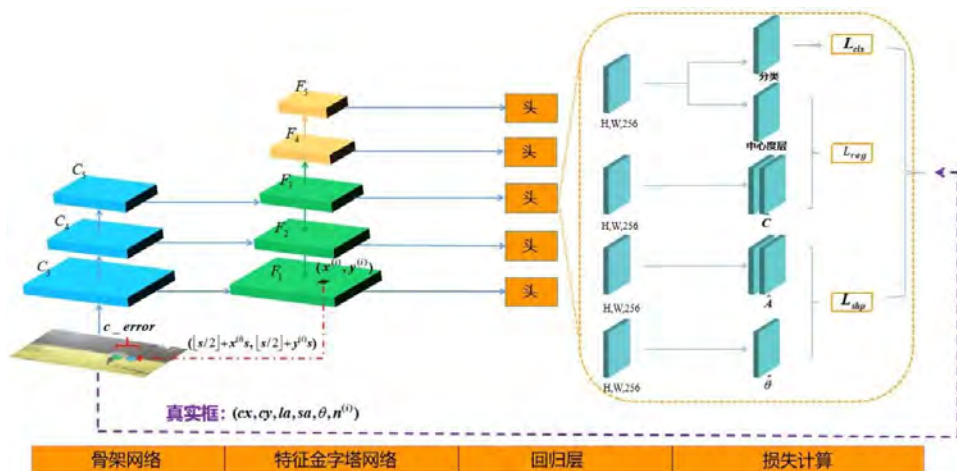


图 2 EllipseNet 网络结构图

的特征图 F1 至 F5, 回归层主要完成对椭圆属性的预测, 包括椭圆圆心的偏移量 \hat{C} , 长短轴长 \hat{A} 以及椭圆的旋转角度 $\hat{\theta}$ 以及类别, 最后再计算预测椭圆框和真实椭圆框之间的损失。

2.2 网络结构

如图 2 所示, 原始图像首先经过 ResNet50 来提取出不同尺度大小的特征图 C3-C5。其中 C5 包含了丰富的语义信息, 但随着下采样率的增加导致 C5 上包含的小物体的有效信息逐渐较少。而浅层特征图 C3 包含了小物体的边缘信息但语义信息不足。为了融合不同特征层之间的语义信息, 本文采用特征金字塔网络 (Feature Pyramid Network, FPN), FPN 首先使用一个 1×1 的卷积核来将 C2-C5 的通道数目统一变成 256, 之后相邻特征层借助上采样来自上而下的融合不同特征层之间语义信息, 最终经过 FPN 得到了四张融合后的特征图 F1-F5, 后续分别用于完成图像的分类和定位。

在得到 F1-F5 之后, EllipseNet 的回归层部分则是在各个特征层的基础上用以预测椭圆的类别及属性 (圆心位置、长短轴长和旋转角度)。然而在实际实验过程中, 如图 3(a) 所示, 观察到 FCOS 预测出物体的中心位置会跟实际的中心发生偏移, 究其原因是由于特征图下采样丢失精度以及 FCOS 会在中心区域附近选择正样本。为了弥补圆心偏移的损失, 在预测层中额外回归了圆心偏移量。假设第 i 个特征图 F_i 中的某个位置坐标为 $(x^{(i)}, y^{(i)})$, 则该位置和圆心之间的真实偏移量如公式 (1) 所示:

$$c_error = (cx - x^{(i)}, cy - y^{(i)}) \quad (1)$$

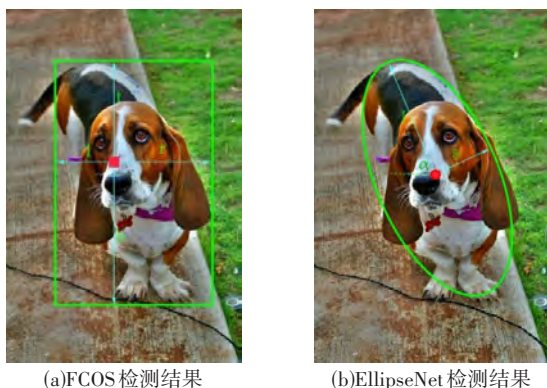


图 3 FCOS 和 EllipseNet 检测效果图

故在模型的输出层中, \hat{C} 负责回归椭圆圆心的误

差偏移量, \hat{A} 负责预测椭圆的长短轴长以及 $\hat{\theta}$ 负责预测椭圆的旋转角度。另外, 由于椭圆框的长短轴长以及旋转角度均为正数, 因此在每个输出分支后面经过 $\exp(x)$ 函数将其映射成正数以便于模型的收敛。

2.3 正负样本选择

EllipseNet 正样本的选择采用了 FCOS 算法的思想, 即首先将每个特征图的像素位置映射回原图尺度, 若特征图的某个位置落在了真实框内, 则将其视为正样本的一个必要条件; 之后如公式 (2) 所示, 设计了一个中心采样比例系数 $center_sampling_ratio$, 分别为每个特征层 F_i 根据其下采样步长设定了一个半径 r , 将特征图上的像素位置在半径范围内的视为正样本的另一个必要条件;

$$r = center_sampling_ratio \times stride \quad (2)$$

但此时并没有考虑 FPN 结构按照各自特征图尺度分配正样本的原则, 故又额外设计了 $regree_ranges$ 参数, 该参数为每个特征图 F_i 设计了一个尺度范围 $scale$, 若正样本的最长边落在某个 $scale$ 内, 则该特征层负责预测该物体, 其余特征层则视为负样本。



图 4 正样本点匹配两个真实框示意图

在完成上述正样本分配后, 如图 4 所示, 此时依然存在一个正样本点可能同时匹配两个真实框的情况, 而本文做法则是直接按照面积匹配最小原则将当前正样本点匹配给面积最小的真实框。

2.4 损失函数

EllipseNet 整体损失函数如公式 (3) 所示:

$$L = L_{cls} + L_{ellipse_iou} \quad (3)$$

其中 L_{cls} 表示类别损失, 而 $L_{ellipse_iou}$ 表示椭圆的交并比损失。考虑到任意形状的两个椭圆的交并比存在运算强度大的缺点。因此, 为了加速网络的训

练,重新定义的损失函数为:

$$L = \frac{1}{N_{pos}} L_{cls} + \frac{\alpha}{N_{pos}} L_{reg} + \frac{\beta}{N_{pos}} L_{shp} \quad (4)$$

其中 L_{cls} , L_{reg} 和 L_{shp} 分别表示类别损失,椭圆中心位置损失以及形状损失。 N_{pos} 表示正样本像素点的数量, α 和 β 是平衡上述损失的权重。

如公式(5)所示,类别损失 L_{cls} 使用的是 Focal Loss^[21],定义如下:

$$L_{cls} = -\eta(1 - p_t)^\gamma \log(p_t) \quad (5)$$

其中 η 和 γ 是两个超参数, p_t 表示物体属于类别 t 的概率。

如公式(6)所示,椭圆中心位置损失 L_{reg} 由两部分组成,其中第一部分 $BCE(P_{centerness}, G_{centerness})$ 采用的是FCOS算法的中心度损失,旨在为每个正样本像素点根据距离圆心的远近自适应分配一个权重,距离越近则该正样本点质量越高;而 \hat{C} 和 C 分别表示预测椭圆和真实椭圆之间的误差偏移量。

$$L_{reg} = BCE(P_{centerness}, G_{centerness}) + Smooth_{L1}(\hat{C}, C) \quad (6)$$

最后椭圆形状损失 L_{shp} 如公式(7)所示,其中 \hat{A} , A 分别表示预测椭圆和真实椭圆的长短轴长; $\hat{\theta}$ 和 θ 分别表示预测旋转角度和真实旋转角度。

$$L_{shp} = Elliptical_IOU(\hat{A}, A) + Smooth_{L1}(\hat{A}, A) + Smooth_{L1}(\hat{\theta}, \theta) \quad (7)$$

另外,本文还额外引入了一种椭圆交并比损失函数,如图5所示,令 $G = [g_1, g_2]$ 和 $P = [p_1, p_2]$ 分别表示椭圆的真实长短轴长和预测长短轴长,则

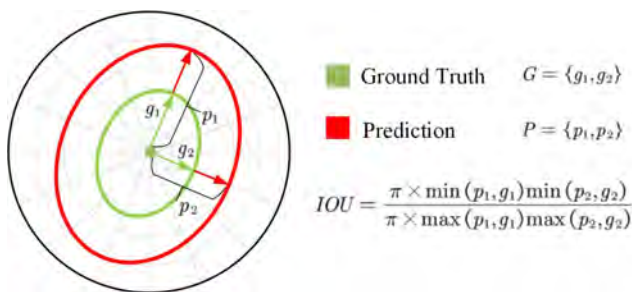


图5 椭圆交并比损失函数

两个椭圆的近似交并比如公式(8)所示:

$$IOU = \frac{\pi \times \min(p_1, g_1) \times \min(p_2, g_2)}{\pi \times \max(p_1, g_1) \times \max(p_2, g_2)} \quad (8)$$

由于交并比需要优化到1,最终的椭圆交并比损失为:

$$Elliptical\ IOU\ Loss = \log \frac{\prod_{i=1}^2 \max(p_i, g_i)}{\prod_{i=1}^2 \min(p_i, g_i)} \quad (9)$$

当然在模型训练初始阶段,网络还不能够准确的预测椭圆的旋转角度和中心偏移量,因此椭圆交并比损失是在模型训练稳定后才被添加进去的。

2.5 推理过程

给定一张输入图像后,经过FPN得到五张特征图,之后经过回归层之后每张特征图的每个位置会得到五个预测值:类别置信度分数、中心度分数、椭圆中心偏移量、长短轴长以及旋转角度。然后将特征图所有像素点映射回原图,将类别置信度分数和中心度分数相乘得到最终每个像素点的置信度分数,另外,本文设定一张图最多只检测 M 个物体,并通过一个分数阈值 δ 来过滤掉低于阈值的像素点。在得到预测点之后,再通过公式(1)来修正这些点的位置便得到了椭圆圆心的位置,另外在结合对应位置其他预测值(长短轴长,旋转角度)便得到了预测椭圆框,最终经过椭圆的NMS便得到了最终的输出结果。

3 实验结果

3.1 数据集

据本文所知,目前没有用旋转椭圆框标注的自然场景下的数据集。为了测试EllipseNet在自然场景数据集上的性能,本文在PASCAL VOC2007数据集基础上构造了一个旋转数据集,其中原始的PASCAL VOC2007数据集包含了丰富的自然场景,该数据集总共标注了20个类别的物体。但由于PASCAL VOC2007数据集中物体大多数垂直于地面,为了增加角度的多样性需要对该数据集进行旋转。本文通过使用传统的图像处理算法等比例的旋转图像并人为的用旋转椭圆框对物体进行了标注。本文将构造完成的数据集称为Rotate VOC2007数据集,该数据集总共包含了5,747张图像,其中4,981张图像用于训练,766张图像用于测试,总共包含了隶属于20个类别的18,445个旋转的物体。

3.2 对比方法

本文选择了目前性能优异的旋转检测算法和EllipseNet做对比,包括一阶段旋转检测算法和二阶段旋转检测算法。其中,一阶段检测算法包括:

(1)RetinaNet-OB^[21],一种借助水平锚框的高效旋转检测算法。

(2)FCOS-Poly^[11]:在FCOS算法基础上结合Poly-IoU-Loss损失函数进行旋转检测的算法。

(3)Gliding Vertex^[17]:一种在预测水平矩形框基础上额外再预测四个角点偏移量的旋转检测算法。

(4)R3Det^[15]:一种通过引入特征微调模块来克服旋转候选框特征不对齐的旋转检测算法。

与上述一阶段旋转检测算法相比,二阶段旋转检测算法由于额外增加了一个RPN模块其性能普遍较好,但随之带来的缺点就是检测速度相对较慢。本文选择的二阶段检测算法包括:

(1)R2CNN-OB^[4]:一个在FasterRCNN基础上改造的直接预测旋转框的二阶段检测算法。

(2)RoI Transformer^[16]:一种借用水平锚框并借助RoI Learner模块学习水平锚框到旋转锚框变换的二阶段检测算法。

(3)Oriented RCNN^[22]:一种借助Oriented RPN模块学习水平锚框到旋转锚框变换的二阶段目标检测算法。

3.3 超参数设置

本文将数据集R-VOC2007中图像统一变换成

[512, 800, 3]大小。另外,在模型训练过程中,使用了随机上下、左右翻转的图像增强方法。模型总共迭代了100轮,并将初始学习率设置为5e-5。使用Adam^[23]优化器并以批次2对模型进行训练。在正负样本选择过程中,令中心采样率 $center_sampling_ratio = 1.5$,各特征图下采样率为 $stride = [8, 16, 32, 64, 128]$ 。在损失函数计算过程中,将Focal Loss的两个超参分别设置为 $\eta = 0.25$ 和 $\gamma = 2.0$,将权重系数分别设置为 $\alpha = 1, \beta = 1$ 。

本文采用的硬件平台为英特尔的i9-10900X CPU和一块显存为24GB的英伟达3090显卡,Pytorch采用的版本为1.7.1。

3.4 实验结果分析

表1展示了EllipseNet和其他旋转检测算法的性能比较。从表中可以看出,二阶段目标检测算法RoI Transformer和一阶段目标检测算法Gliding Vertex分别以48.27%和46.06%的mAP值取得了最优的性能。虽然EllipseNet仅取得了38.71%的mAP值,但在汽车和自行车两个类别上分别击败了其余所有的方法,这也在一定程度上验证了本文方法在面对复杂的自然场景时也具有一定的性能优势。

表1 RVOC2007数据集实验结果对比

骨架网络	二阶段检测算法			一阶段检测算法				
	ResNet50+FPN			ResNet50+FPN				
类别	R2CNN-OB	RoI Transformer	Oriented RCNN	RetinaNet-OB	FCOS-Poly	Gliding Vertex	R3Det	EllipseNet
飞机	79.66	75.98	64.98	65.42	48.93	78.03	66.25	51.32
自行车	47.73	46.75	41.77	47.74	43.53	48.09	40.28	36.41
鸟	37.20	27.00	31.76	35.22	17.89	31.33	40.21	31.20
船	30.66	36.45	33.23	28.06	15.80	23.87	26.00	8.23
杯子	20.08	19.87	22.40	13.77	7.09	17.41	17.28	8.56
公交车	65.77	66.69	70.86	61.43	47.82	62.02	62.60	58.69
汽车	58.32	59.02	60.10	55.32	48.83	59.62	55.69	62.84
猫	64.90	66.56	61.66	62.84	43.17	66.29	66.44	57.63
椅子	19.03	17.75	21.62	18.81	13.03	17.50	17.61	11.56
牛	50.65	49.12	44.66	39.01	19.59	49.93	36.32	36.84
桌子	41.91	40.51	42.16	29.80	15.93	46.16	29.41	21.65
狗	55.53	54.04	54.82	56.53	32.27	45.94	54.08	43.85
马	64.42	60.77	64.13	63.81	32.44	61.04	58.76	51.46
自行车	52.35	57.81	60.23	56.01	39.78	55.83	51.91	68.52
人	49.25	49.38	55.08	44.34	30.43	47.76	52.62	41.62
花盆	20.60	25.23	22.37	20.56	7.24	20.44	25.38	14.62
绵羊	54.50	56.67	55.47	49.57	33.35	55.63	45.82	54.26
沙发	40.72	44.11	44.98	33.09	13.43	41.29	34.35	24.84
火车	57.47	56.54	51.83	56.27	43.49	47.82	53.07	43.81
电视	51.01	55.05	52.40	51.89	29.01	45.23	54.43	46.28
mAP	48.09	48.27	47.82	44.47	29.15	46.06	44.43	38.71

3.5 消融实验

(1)椭圆中心回归:一个椭圆的中心对确定一个椭圆框的位置非常重要。虽然FCOS能够预测物体的中心区域,但在一定程度上,如图3(a)所示,依然会存在一定的偏差。表2展示了是否添加椭圆中心偏差分支的消融实验,从中可以看出,在额外回归椭圆圆心偏差后 mAP有 1.4% 的提升,从而验证了椭圆中心对预测椭圆位置的重要性。

(2)椭圆交并比损失函数:表3展示了不同损失函数对 EllipseNet 性能的影响,从表中可以看出,相较于仅使用 Smooth L1 Loss^[4],在添加了椭圆交并比损失后 EllipseNet 的 mAP 涨了 0.9%,从而验证了椭圆交并比损失函数能够进一步的提升模型的性能。

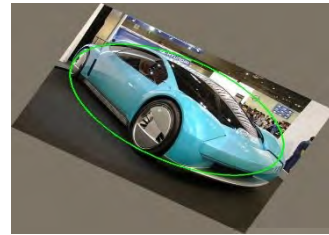


图6 RVOC2007可视化

4 结论

本文提出了一种能够更加精确检测物体的旋转椭圆目标检测器 EllipseNet,设计网络结构使其预测椭圆的圆心偏移量、长短轴长以及旋转角度来完成旋转椭圆的预测。另外在损失函数部分,考虑到椭圆框本质上应该作为一个整体来进行损失计算,而不应该用 Smooth L1 Loss 来单独回归椭圆框的各个属性,因此本文额外又提出了一种椭圆交并比损失函数来进一步提升了模型的性能。最终实验表明, EllipseNet 具有一定的应用价值。

参考文献 (References):

- [1] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.
- [2] 员娇娇,胡永利,孙艳丰,尹宝才.基于深度学习的小目标检测方法综述[J].北京工业大学学报,2021,47(03): 293-302.
- [3] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [4] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks

表2 椭圆中心偏差分支消融实验

EllipseNet	不回归圆心偏差	回归圆心偏差
mAP	35.9	37.3

表3 损失函数消融实验

EllipseNet	Smooth L1 Loss	椭圆交并比损失
mAP	37.3	38.2

3.6 可视化分析

如图6所示,本文可视化了 EllipseNet 在 RVOC2007 数据集上的部分检测结果,可以看出,在复杂的自然场景下 EllipseNet 能够较好的检测各个角度姿态下的物体。

[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6): 1137-1149.

- [5] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multi-box detector [C]//European Conference on Computer Vision, 2016:21-37.
- [6] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016:779-788.
- [7] Redmon J, Farhadi A. Yolo9000: better, faster, stronger [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 7263-7271.
- [8] Redmon J, Farhadi A. Yolov3: An incremental improvement [DB/OL]. arXiv:1804.02767, 2018.
- [9] Duan K, Bai S, Xie L. et al. Centernet: keypoint triplets for object detection [C]//Proceedings of the IEEE International Conference on Computer Vision, 2019:6569-6578.
- [10] Zhou X, Wang D, Krähenbühl P. Objects as points [DB/OL]. arXiv:1904.07850, 2019.
- [11] Tian Z, Shen C, Chen H, et al. FCOS: Fully convolutional one-stage object detection [C]//Proceedings of the IEEE International Conference on Computer Vision, 2019: 9627-9636.

- [12] Law H, Deng J. Cornernet: detecting objects as paired keypoints [C]//Proceedings of the European Conference on Computer Vision, 2018: 734-750.
- [13] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge [J]. International Journal of Computer Vision, 2010, 88(2): 303-338.
- [14] Lin TY, Maire M, Belongie S, et al. Microsoft coco: Common objects in context [C]//European Conference on Computer Vision, 2014:740-755.
- [15] Yang X, Yan J, Feng Z, et al. R3det: Refined single-stage detector with feature refinement for rotating object [DB/OL]. arXiv:1908.05612, 2019.
- [16] Ding J, Xue N, Long Y, et al. Learning roi transformer for oriented object detection in aerial images [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 2849-2858.
- [17] Xu Y, Fu M, Wang Q, et al. Gliding vertex on the horizontal bounding box for multi-oriented object detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(4): 1452-1459.
- [18] Ma J, Shao W, Ye H, et al. Arbitrary-oriented scene text detection via rotation proposals [J]. IEEE Transactions on Multimedia, 2018, 20(11): 3111-3122.
- [19] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.
- [20] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2117-2125.
- [21] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection [C]//Proceedings of the IEEE International Conference on Computer Vision, 2017: 2980-2988.
- [22] Xie X, Cheng G, Wang J, et al. Oriented r-cnn for object detection [C]//Proceedings of the IEEE International Conference on Computer Vision, 2021: 3520-3529.
- [23] Kingma D P, Ba J. Adam: A method for stochastic optimization [DB/OL]. arXiv:1412.6980, 2014.

编辑:王谦

(上接第8页)

- [23] Liu Y, Fan B, Xiang S, et al. Relation-shape convolutional neural network for point cloud analysis [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 8895-8904.
- [24] Liu Y, Fan B, Meng G, et al. Denspoint: Learning densely contextual representation for efficient point cloud processing [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 5239-5248.
- [25] Wu W, Qi Z, Fuxin L. Pointconv: Deep convolutional networks on 3d point clouds [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 9621-9630.
- [26] Hermosilla P, Ritschel T, Vázquez P P, et al. Monte carlo convolution for learning on non-uniformly sampled point clouds [J]. ACM Transactions on Graphics, 2018, 37(6): 1-12.
- [27] Qi C R, Su H, Mo K, et al. Pointnet: deep learning on point sets for 3d classification and segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 652-660.
- [28] Qi C R, Yi L, Su H, et al. Pointnet++: deep hierarchical feature learning on point sets in a metric space [C]//NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 5105-5114.
- [29] Jiang M, Wu Y, Zhao T, et al. Pointsift: a sift-like network module for 3d point cloud semantic segmentation [DB/OL]. arXiv:1807.00652, 2018.
- [30] Cheraghian A, Petersson L. 3dcapsule: extending the capsule architecture to classify 3d point clouds [C]//IEEE Winter Conference on Applications of Computer Vision (WACV), 2019: 1194-1202.
- [31] Yang Y, Feng C, Shen Y, et al. Foldingnet: point cloud auto-encoder via deep grid deformation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 206-215.
- [32] Yuan W, Khot T, Held D, et al. Pcn: point completion network [C]//International Conference on 3D Vision (3DV), 2018: 728-737.
- [33] Li R, Li X, Fu C W, et al. Pu-gan: a point cloud upsampling adversarial network [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 7203-7212.

编辑:王谦