doi:10.3969/j.issn.1672-1292.2021.04.005

基于改进 YOLOv3 的运动目标分类检测算法研究

梁秦嘉,刘　怀,陆　飞

(南京师范大学电气与自动化工程学院,江苏 南京 210023)

[摘要]　提出一种基于改进 YOLOv3 算法的一类运动目标检测算法. 为进一步提高 YOLOv3 的检测精度,采用基于 DIoU 优化的边界框回归损失函数进行计算;优化非极大值抑制,有效减少了目标框重叠的现象,提高检测精度;针对运动目标检测,提出一种基于目标框多中心点位移的检测算法. 经 UA-DETRAC 数据集上的实验表明,改进后的算法在提高检测精度的同时保证了较快的速度,准确率和召回率相比原始 YOLOv3 分别提高了 8.07%和 3.87%,对运动目标的检测速度可达 20 fps/s,可满足实时检测的要求.

[关键词]　交通监控,卷积神经网络,运动目标检测

[中图分类号]TP391.4　[文献标志码]A　[文章编号]1672-1292(2021)04-0027-06

Moving Target Classification and Detection Algorithm Based on Improved YOLOv3

Liang Qinjia, Liu Huai, Lu Fei

(School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China)

Abstract: A kind of moving target detection algorithm based on improved YOLOv3 algorithm is proposed in this paper. In order to improve the detection accuracy of YOLOv3, the boundary box regression loss function based on DIoU optimization is used. Non-maximum suppression is optimized to effectively reduce the overlap of target boxes and improve the detection accuracy. Aiming at moving target detection, a multi-center displacement detection algorithm based on target frame is proposed. The experimental results on UA-DETRAC dataset show that the detection accuracy and the fast speed can be improved by the improved algorithm. Compared with the original YOLOv3, the accuracy and recall rate are increased by 8.07% and 3.87%, respectively. The detection speed of moving target can reach 20 fps/s, which can meet the requirements of real-time detection.

Key words: traffic monitoring, convolutional neural networks, moving object detection

随着计算机视觉的发展,研究人员越来越重视与运动目标检测相关的课题研究[1]. 伴随着城市智能交通的不断发展,针对交通视频中的运动车辆进行检测已成为当前一个重要的研究课题,该研究可为解决交通拥堵、提取交通违法证据等各种交通智能控制应用提供有效数据支持.

传统的运动目标检测算法主要是基于视频帧的[2],通过帧与帧之间的差异来判断目标是否运动,常用方法有帧间差分法[3]、背景减除法[4]、光流法[5]等. 此类方法极易受到背景信息的影响,导致检测精度较低,易造成误检和漏检[6]. 随着机器学习的不断发展,为了能够更好地完成检测工作,专业人员深入分析大图像的特点,结合实际需要进行方法多样化拓展. 目标检测工作首先需确认检测区域,全面分析检测目标属性,提取相应的特征,再进行类别划分[7]. 传统方法受较多条件限制,尤其在目标特征设置方面,只有按需妥善进行特征设计才能更精准地完成模型的建立. 此外,特征提取的准确与否直接影响目标的准确定位,多数情况下传统方法无法提取出目标高层特征,所表达的语义仅仅停留在低层范围内.

近年来,随着图形处理单元(graphics processing unit,GPU)硬件的快速发展,深度学习在目标检测领域取得了显著的进步[8]. 在特征提取过程中利用手工模式得到的结果不够精确,存在很多不足之处,这也

收稿日期:2021-03-08.

基金项目:国家自然科学基金项目(61603194).

通讯作者:刘怀,博士,副教授,研究方向:数字图像处理、实时控制系统. E-mail:liuhuai@njnu.edu.cn

是传统机器学习方法的弊端. 卷积神经网络的应用可弥补这一缺陷,提高研究结果的准确性. 近来在研究运动目标检测的过程中,更多专业人士注意到深度学习的积极影响,并以此为基础进行了运动目标检测算法的创新[9],提出了 Fast-RCNN、Faster-RCNN 等目标检测算法,通过新的算法所得到的计算结果精度更高,但仍有不足之处,如实时检测效率低,这主要是由于这些算法属于穷举法的范围[10]. 基于回归的检测算法可直接利用卷积神经网络的全局特征预测目标位置和类别,检测速度较快,常用的基于回归的算法有 SSD[11]、YOLOv2[12]、YOLOv3[13]等. 相较于 SSD,YOLOv3 采用特征金字塔(FPN)的思想,在精度上比 SSD 有了很大提高;同时,YOLOv3 采用残差结构,其运算速度超过了 SSD. 但由于 YOLOv3 对视频进行检测时会检测出所有目标,并不适用于目标检测.

本文提出一种基于改进 YOLOv3 的交通视频运动目标检测算法. 首先,为进一步提高 YOLOv3 的检测精度,针对损失函数进行改进;其次,对非极大值抑制进行优化,减少同一目标的目标框重叠;最后,针对运动目标,提出一种基于目标框多中心点位移的检测算法.

1　改进的 YOLOv3 算法

1.1　YOLOv3 算法

YOLOv3 算法将原始输入图像划分为 $S\times S$ 个网格单元格,如图 1 所示. 计算公式为:

$$\begin{cases} b_x=\sigma(t_x)+c_x, \\ b_y=\sigma(t_y)+c_y, \\ b_w=p_w e^{t_w}, \\ b_h=p_h e^{t_h}. \end{cases} \tag{1}$$

式中,c_x 和 c_y 表示每个网格的左上角坐标;(b_w,b_h) 为预测的边界框的宽度和高度;(b_x,b_y) 作为一个中心坐标,可表明边界框的位置.

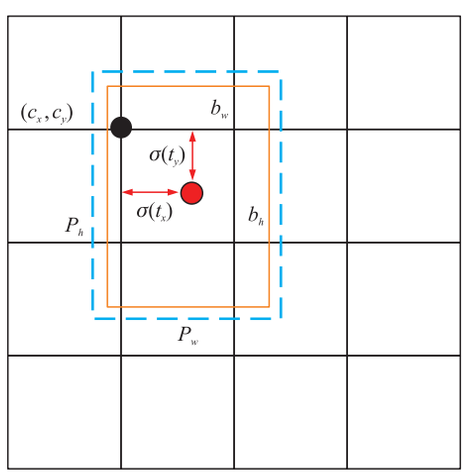


图 1　YOLOv3 检测框与预测框的关系

Fig. 1　The relationship between the Yolov3 check box and the prediction box

YOLOv3 使用逻辑回归计算每个先验框的置信度为:

$$\mathrm{Conf}=\mathrm{Pre}(\mathrm{object})\times \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}, \tag{2}$$

式中,Pre(object)用来判断目标对象是否出现在网格中,若出现,设置为 1;若未出现,设置为 0. $\mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}$ 是预测框与真实框的交集面积与并集面积的比值:

$$\mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}=\frac{\mathrm{area}(\mathrm{box}_{\mathrm{pred}}\cap \mathrm{box}_{\mathrm{truth}})}{\mathrm{area}(\mathrm{box}_{\mathrm{pred}}\cup \mathrm{box}_{\mathrm{truth}})}. \tag{3}$$

当目标出现时,需预测目标出现的类别,定义为:

$$C(M)=\mathrm{Pre}(\mathrm{class}_M\mid \mathrm{object})\times \mathrm{Pre}(\mathrm{object})\times \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}=\mathrm{Pre}(\mathrm{class}_M)\times \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}. \tag{4}$$

模型预测值并不一定能够始终保持与真实值保持一致,这种情况可通过损失函数来进行描述. 损失函数作为重要参数,会对网络性能造成影响. YOLOv3 算法的设计是围绕损失函数展开的,通过预测误差和边界框的置信误差及分类误差来设计,定义为:

$$\begin{aligned} L=&\lambda_{\mathrm{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\mathrm{obj}}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]+\lambda_{\mathrm{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\mathrm{obj}}\left[(\sqrt{\omega_i}-\sqrt{\hat{\omega}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2\right]+ \\ &\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\mathrm{obj}}\left[(C_i-\hat{C}_i)^2\right]+\lambda_{\mathrm{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\mathrm{noobj}}\left[(C_i-\hat{C}_i)^2\right]+\sum_{i=0}^{S^2} l_i^{\mathrm{obj}}\sum_{c\in \mathrm{classes}}(p_i(c)-\hat{p}_i(c))^2. \end{aligned} \tag{5}$$

1.2　基于改进 YOLOv3 的运动目标检测算法

1.2.1　优化边界框回归损失函数

交并比(intersection over union,IoU)是目标检测中一个非常重要的参数,是产生的预测框与原标记框的交叠率,即其交集与并集的比值. 现阶段 IoU 的应用范围越来越广,但作为目标检测任务的一种仍存在一些不足:(1)当预测边界框与目标边界框不相交,由 IoU 定义可得,IoU = 0,此时 IoU 不能反映两个边界框之间的距离,同时,无法按照需求完成梯度回传,这是位置误差和置信度的特点所决定的,从而降低了网

络学习效率;(2)若边界框的目标值和预测值在距离存在差异的情况下相交面积相同,最终得到的 IoU 结果也一致,就无法对两者重合度进行准确描述,从而影响网络性能.

为了改善这些不足,Rezatofighi 等[14]提出了一种改进的 GIoU 方法,计算方法为:

$$\mathrm{GIoU}=\mathrm{IoU}-\frac{C-(A\cup B)}{C}, \tag{6}$$

式中,A 和 B 分别表示预测边界框和目标边界框,C 表示 A 和 B 的最小凸集. 相较于 IoU,GIoU 除了对重叠区域比较关注之外,还关注其他的非重合区域,能更好地反映两者的重合度. 当 A 与 B 处于不同点时,GIoU 的值会随着其间距的增加而与-1 无限接近. 1-GIoU 代表损失函数,这也能够证明 A 与 B 之间的重合情况. 此外,还有一种情况会导致 GIoU 退化为 IoU,即 B 包含 A.

针对以上问题,本文对 YOLOv3 进行改进,采用 DIoU 作为边界框回归损失函数[15],其原理如图 2 所示,其计算过程为:

$$L_{\mathrm{DIoU}}=1-\mathrm{IoU}+\frac{\rho^2(b,b^{gt})}{d_1^2}, \tag{7}$$

$$\mathrm{IoU}=\left|\frac{B\cap B^{gt}}{B\cup B^{gt}}\right|. \tag{8}$$

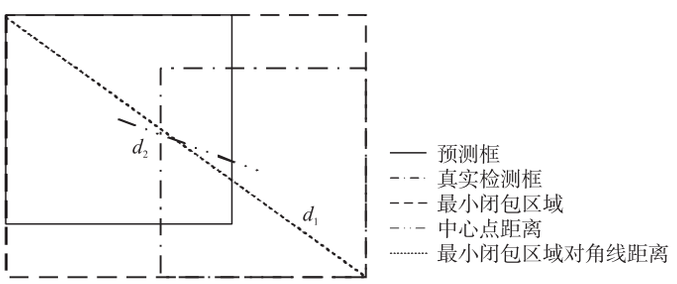


图 2　DIoU 原理示意图

Fig. 2　The DIoU schematic

相较于目前广泛应用的 IoU 和 GIoU 函数,DIoU 更加符合目标框回归机制,综合考虑了目标与目标框之间的距离、重叠率及尺度,提高了目标框回归的稳定性,不会像 IoU 和 GIoU 一样出现训练过程中的发散等问题,可使检测精度更高.

1.2.2　非最大值抑制优化算法

非最大抑制(NMS)是目标检测算法中一个必要的后处理步骤. 非最大抑制算法的传统应用模式中检测框 B 是最早被确定的,对于被检测图片来说其与对应分数 S 都属于能够最先确定的值. 将分数最高的检测框标记为 M,当 M 被确定时,就会被归属于检测结果集合 D,离开集合 B. 这种算法通过强制归零的方式来处理相邻检测框分数存在严重的弊端,若重叠区域内存在真实物体,就会影响检测结果.

针对以上问题,本文采用软化非极大值抑制(Soft-NMS)算法[16],通过设置衰减函数来解决重叠部分检测框的问题. 分数的高低会随着 M 和其余检测框之间重叠面积的大小而发生变化,重叠越大,分数越低,若所得检测分数影响不大则说明重叠面积很小. Soft-NMS 实现便捷,节省了额外训练所消耗的时间和经济成本. 其算法流程如下:

Soft-NMS

Input: $B=\{b_1,\cdots,b_N\}$, $S=\{s_1,\cdots,s_N\}$

1. $D\leftarrow\{\ \}$

2. while $B\neq$ empty do

3. $\quad m\leftarrow$argmax S

4. $\quad M\leftarrow b_m$

5. $\quad D\leftarrow D\cup M$; $B\leftarrow B-M$

6. $\quad$ for b_i in B do

7. $\quad\quad S_i\leftarrow S_i\cdot f(\mathrm{IoU}(M,b_i))$

8.　end
9.end
10.return D,　S

其中,B 集合是检测到的所有建议框,S 集合是各个建议框得分,函数 $f(\mathrm{IoU}(M,b_i))$ 定义为:

$$f(\mathrm{IoU}(M,b_i))=\begin{cases}1, & \mathrm{IoU}(M,b_i)<N_t;\\ \mathrm{e}^{-\frac{\mathrm{IoU}(M,b_i)^2}{\sigma}}, & \mathrm{IoU}(M,b_i)\geqslant N_t.\end{cases}\tag{9}$$

式中,b_i 为边界框的序号;M 为最高分;N_t 为设定的阈值;σ 为超参数.

与 NMS 算法相比,Soft-NMS 算法增加了一个惩罚函数. 若一个预测框和 M 计算出的 IoU 超过了一定阈值,预测框不会被删除,但其分数会相应减少.

1.2.3　基于中心点位移的运动目标检测算法

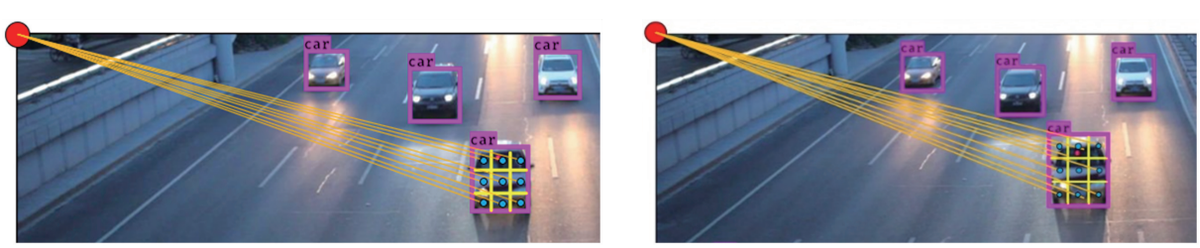
在目标检测的最后阶段,检测层对所有检测出的目标框进行非最大值抑制. 通常,在进行非极大值抑制后,多余的目标框将被删除,剔除静止的目标,只保留运动的目标,从而实现运动目标检测. 为了判断目标是否发生了移动,需判断目标框是否发生了一定程度的位移. 针对检测过程中视频中每一帧的相同目标所生成的目标框位置不尽相同,本文提出一种基于目标框均值中心点位移的运动目标检测算法.

在进行非极大值抑制前,有多个目标框出现,这是由算法的特性决定的. 每个目标框的中心点坐标可由式(1)得出. 在得到每个目标框的中心点坐标后,对这些坐标在两个方向上分别求取加权平均值,得到新的坐标点(x,y),定义为目标框的均值中心点. 在对视频进行运动目标检测时,首先检测出当前帧的所有目标,并对检测出的目标求取均值中心点. 之后对视频下一帧进行目标检测,同样求出检测到目标的均值中心点. 对下一帧目标完成检测后,将下一帧所检测到的均值中心点与当前帧同一位置目标进行比较,若中心点坐标在 x 方向和 y 方向的位移偏移量超过一定阈值时,则判断该目标发生了运动. 由于每个目标在不同帧检测时得到的目标框位置可能会发生一定的变化,因此,通过一个中心点无法准确判断目标的运动情况. 当阈值设置较大时,若目标移动速度较慢,会导致目标被误认为未运动;当阈值设置较小时,由于每一帧生成目标框位置不固定,静止目标又可能会被误认为发生了运动. 对此,本文提出一种多点位移变化的方法,以准确判断目标是否发生了运动.

在检测到目标之后,将目标的多个检测框划分为 3×3 的网格,对每一个网格分别求取均值中心点. 实验表明,3×3 的网格可准确判断出运动目标,同时不会增加过多的计算量. 为方便计算偏移量,本文将视频左上角设为原点建立坐标系,以中心点到原点之间的距离作为衡量指标. 以其中一个 3×3 网格为例,以视频图像的左上角为原点(0,0),在视频当前帧中求出 9 个中心点分别为$(x_1,y_1),\cdots,(x_9,y_9)$,之后,求出 9 个中心点相对于左上角原点之间的距离 $y_1,\cdots,y_9$,计算公式为:

$$l_n=\sqrt{(x_n-0)^2+(y_n-0)^2},\tag{10}$$

其中,$n=1,2,\cdots,9$. 针对下一帧进行同样操作,下一帧中此目标的 9 个中心点分别为$(x_1',y_1'),\cdots,(x_9',y_9')$,并计算出 9 个中心点相对于左上角原点之间的距离 $l_1',\cdots,l_9'$. 当 9 个中心点相对于原点的位移偏移量超过设置的阈值时,则判断该目标发生了变化. 判定过程如图 3 所示.



(a)　　(b)

(a)、(b)分别为视频中连续的两帧

图 3　多中心点位移偏移示意图

Fig. 3　Schematic diagram of multi-center point displacement offset

2 实验结果及分析

2.1 数据集

在网络改进后,为了能够对其性能和方法进行评测,本文以车辆目标为例,采用 UA-DETRAC 数据集在深度学习框架 keras 下对算法进行训练. 实验环境配置为:CPU 为 Intel i5-9400,主频 2.90 GHz,16 GB 内存,GPU 为 NVIDIA 1070,8 GB 显存,操作系统为 Windows 10.

本文对数据集进行重新标注训练,检测目标包含小型汽车(car)、公共汽车(bus)、大型货车(truck)3 类. 为了提高训练效果,使用了不同角度旋转图像和改变图像的饱和度、曝光和色调等数据增强方法. 在训练阶段,初始学习率为 0.001,权值衰减为 0.000 5. 当训练批次为 60 000 和 70 000 时,学习率分别降至 0.000 1 和 0.000 01,使损失函数进一步收敛.

2.2 评价指标

本文中精度和召回率分别定义为:

$$R_{\text{Precision}}=\frac{TP}{FP+TP}, \tag{11}$$

$$R_{\text{Recall}}=\frac{TP}{FP+TN}. \tag{12}$$

式中,TP 为检测正确的在运动的小型汽车数量;FP 为将其他类型如卡车、公共汽车、行人及其他静止目标误检为运动的小型汽车的数量;FN 为将小型汽车错误识别为其他类型的数量. 帧率为每秒检测的帧数.

2.3 结果分析

利用优化后 YOLOv3 模型对 UA-DETRAC 数据集中的目标进行测试. 在所有目标检测阶段,针对所有车辆目标,采用不同算法进行实验,测试改进后算法的性能. 测试结果如表 1 所示.

表 1 不同算法实验结果对比

Table 1 Comparison of experimental results of different algorithms

网络	mAP/%	Recall/%	平均时间/ms
Faster-R-CNN(resnet101)	86.14	92.15	250.21
Faster-R-CNN(VGG16)	86.42	92.27	163.52
YOLOv3-tiny	73.25	90.65	6.38
YOLOv3	76.68	92.36	29.67
YOLOv3-DIoU	78.27	93.14	29.85
YOLOv3-Soft-NMS	79.52	94.33	29.31
改进 YOLOv3	84.75	97.23	31.35

从表 1 可知,YOLOv3 模型经优化后目标精度可达 84.75%,召回率为 97.23%,各项数据均得到明显提升,检测速度也显著提升. Faster-R-CNN 无论是加载 ResNet101 还是 VGG16 模型,速度都更慢. 只改进损失函数的 YOLOv3-DIoU 平均精度为 78.27%,只改进非极大值抑制的 YOLOv3-Soft-NMS 的平均精度为 79.52%,相比原 YOLOv3 算法均提升不大. 改进后的 YOLOv3 算法检测一帧图像的时间为 31.35 ms,与同系列其他算法相比无明显增加,可满足实际应用时的实时性需求.

为了更加准确地验证检测有效性,本文进行了实践检验. 通过大量训练与检测的实验表明,当距离偏移量的阈值设置为 7 时,检测效果最好.

(1) 第1段视频第163帧 (2) 第2段视频第78帧 (3) 第2段视频第268帧 (4) 第3段视频第166帧 (5) 第3段视频第308帧

图 4 改进 YOLOv3 算法运动目标检测效果图

Fig. 4 Moving object detection effect diagram of improved YOLOV3 algorithm

在第一个视频的第 163 帧,可以看到,本文算法准确检测到所有的运动目标. 在第二段视频中,在第 78 帧,本文检测到小型汽车与公共汽车共 5 个运动目标;在第 268 帧,公共汽车由于到站而停下,因此,本

文算法将其排除,仅标注小型汽车一个运动目标. 在第三个视频中,中央的卡车停在路边,第 166 帧,上方与下方的车辆均在等红灯,只有中央三辆小型汽车运动;在第 308 帧,上方与下方的车辆开始通行,运动的车辆均正确检测,而静止的车辆被排除. 本文算法对视频的检测速度平均为 20.35 fps/s. 由此可见,本文的算法可以实现对运动目标的检测.

3　结论

本文以改进的 YOLOv3 检测算法测定运动目标. 首先,为进一步提高 YOLOv3 的检测精度,采用基于 DIoU 优化的损失函数进行计算;其次,对非极大值抑制进行优化,减少目标框重叠现象,提高了检测精度;最后,针对运动目标,提出一种基于目标框中心点位移的检测算法. 通过在 UA-DETRAC 数据集上与原始 YOLOv3 进行对比实验,本文所提出的改进算法不仅使检测结果更准确,同时也能够提高检测速度,准确率和召回率相比原始 YOLOv3 分别提高了 8.07%和 3.87%,对运动目标的检测速度可达 20.35 fps/s,能够满足实时检测的要求.

[参考文献](References)

[1] PAN M Y,SUN J,YANG Y H,et al. Improved TQWT for marine moving target detection[J]. Journal of Systems Engineering and Electronics,2020,31(3):470-481.

[2] HU H B,XU L,ZHAO H. A spherical codebook in YUV color space for moving object detection[J]. Sensor Letters,2012,10(1-2):177-189.

[3] DU B,SUN Y J,CAI S H,et al. Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm[J]. IEEE Geoscience and Remote Sensing Letters,2018,15(2):168-172.

[4] WEI P C,HE F,LI J. Fast detection of moving objects based on sequential images processing[J]. Journal of Intelligent and Fuzzy Systems,2020,39(4):5037-5044.

[5] 李成美,白宏阳,郭宏伟,等. 一种改进光流法的运动目标检测及跟踪算法[J]. 仪器仪表学报,2018,39(5):249-256.

[6] MANE S,MANGALE S. Moving object detection and tracking using convolutional neural networks[C]//2018 Second International Conference on Intelligent Computing and Control Systems(ICICCS). Madurai,India:IEEE,2018:1809-1813.

[7] ZOU Z X,SHI Z W,GUO Y H,et al. Object detection in 20 years:a survey[J/OL]. Computer Vision and Pattern Recognition,2019 [2021-03-08]. https://arxiv.org/abs/1905.05055.

[8] LI X,LIU Y,ZHAO Z F,et al. A deep learning approach of vehicle multitarget detection from traffic video[J/OL]. Journal of Advanced Transportation,2018(11):1-11 [2021-03-08]. https://doi.org/10.1155/2018/7075814.

[9] JU M,LUO H B,WANG Z B,et al. The application of improved YOLO V3 in multi-scale target detection[J]. Applied Sciences,2019,9(18):3775.

[10] GIRSHICK R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision(ICCV). Santiago,Chile:IEEE,2015:1440-1448.

[11] CHEN W P,QIAO Y T,LI Y J. Inception-SSD:An improved single shot detector for vehicle detection[J/OL]. Journal of Ambient Intelligence and Humanized Computing,2020 [2021-03-08]. https://doi.org/10.1007/S12652-020-02085W.

[12] SANG J,WU Z Y,GUO P,et al. An improved YOLOv2 for vehicle detection[J]. Sensors,2018,18(12):4272.

[13] REDMON J,FARHADI A. YOLOv3:an incremental improvement[EB/OL]. [2020-07-27]. https://arxiv.org/abs/1804.02767.

[14] REZATOFIGHI H,TSOI N,GWAK J,et al. Generalized intersection over union:a metric and a loss for bounding box regression[C]//2019 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Long Beach,CA,USA:IEEE,2019:658-666.

[15] ZHENG Z H,WANG P,LIU W,et al. Distance-IoU loss:faster and better learning for bounding box regression[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2000,34(7):12993-13000.

[16] BODLA N,SINGH B,CHELLAPPA R,et al. Soft-NMS — improving object detection with one line of code[C]//2017 IEEE International Conference on Computer Vision(ICCV). Venice,Italy:IEEE,2017:5562-5570.

[责任编辑:严海琳]