

基于改进 YOLOv5+DeepSort 算法模型的 交叉路口车辆实时检测^{*}

贾 志, 李茂军, 李婉婷

(长沙理工大学电气与信息工程学院, 湖南 长沙 410114)

摘 要:针对传统目标检测跟踪算法检测精度低、鲁棒性差的缺点,以及交叉路口图像视频资源冗余的现象和车辆密集程度高的特点,提出了一种基于改进 YOLOv5 和 DeepSort 算法模型的交叉路口实时车流量检测方法,在 MS COCO 和 BDD100k 相结合的数据集上,采用改进的 YOLOv5 算法模型实现视频小目标车辆检测,利用深度学习多目标跟踪算法 DeepSort 对检测的车辆进行实时跟踪计数,实现了交叉路口监控端对端的实时车流量检测。通过分析比较不同参数的模型,最终选定了 YOLOv5m 模型。实验结果表明,该方法在复杂环境、车辆遮挡和目标密集程度高等环境下检测速度更加快,对车辆的检测效果更好,平均准确度达到 96.6%。该方法完全满足目标实时性检测的要求,能充分满足交叉路口车辆检测的有效性,满足实际需要的使用需求。

关键词:YOLOv5 算法;车辆检测;DeepSort 算法;目标检测;实时检测

中图分类号:TP301.6

文献标志码:A

doi:10.3969/j.issn.1007-130X.2023.04.014

Real-time vehicle detection at intersections based on improved YOLOv5+DeepSort algorithm model

JIA Zhi, LI Mao-jun, LI Wan-ting

(School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China)

Abstract: Aiming at the characteristics of low detection accuracy and poor robustness of traditional target detection and tracking algorithm, as well as the phenomenon of image and video resource redundancy and high vehicle density at the intersection, a real-time traffic flow detection method based on improved YOLOv5 and DeepSort algorithm model is proposed. This experiment uses a data set combining MS COCO and BDD100k, and uses the improved YOLOv5 algorithm model to realize the small target vehicle detection in video. Then, the deep learning multi-target tracking algorithm (DeepSort algorithm) is used to carry out real-time tracking and counting of the detected vehicles, and the real-time traffic flow detection of the intersection monitoring end-to-end is realized. By analyzing and comparing models with different parameters, the YOLOv5m model is finally selected. Experimental results show that the proposed method has a faster detection speed and better detection effect for vehicles in complex environments, vehicle occlusion and high target density environments, with an average accuracy of 96.6%. This method can fully meet the requirements of real-time detection of targets, and fully meet the effectiveness of vehicle detection at intersections, and meet the actual requirements of use.

Key words: YOLOv5 algorithm; vehicle detection; DeepSort algorithm; target detection; real-time detection

^{*} 收稿日期:2021-08-30;修回日期:2022-01-03

通信地址:410114 湖南省长沙市长沙理工大学电气与信息工程学院

Address: School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, Hunan, P. R. China

1 引言

近年来,随着中国智能交通和城市化进程的快速发展,汽车检测与跟踪已成为智能交通研究的重要课题之一。截至 2020 年底,全国机动车保有量达近 4 亿辆,已有 13 个城市机动车数量超百万辆^[1]。汽车数量的急剧增加对于交通管理部门和道路资源来说是一项严重的挑战。如何准确识别车辆、合理分配车辆及道路资源是智能交通研究的重中之重。近几年,随着计算机算力的增加,图像中场景目标的检测^[2-4]和视频检测^[5]已成为计算机视觉研究领域的研究热点。该目标检测模型不仅能够识别图像中的多个目标,还能判断目标的类别与位置^[6]。将目标检测应用于生活等场景中一直是研究人员的研究方向^[7-9]。车辆作为交通道路环境的主要元素之一,对保障大家的出行效率和安全有着重要的意义。然而由于车辆的急剧增加和驾驶人员的操作问题,时常会导致交通拥堵或陷入瘫痪,因此车流量的实时检测与跟踪计数就成为解决交通问题最基本的手段之一,特别是对于城市地区。

基于交叉路口监控视频的车流量检测通常分为 2 个部分:车辆检测和车辆跟踪。传统的目标检测方法采用的是基于滑动窗口的区域选择方法^[10],该方法不具备针对性,且鲁棒性较差,具有较高的复杂性。近年来,研究人员提出了许多基于深度学习的目标检测算法。2012 年出现的 Alex-Net 网络开启了深度学习的研究热潮,2014 年 R-CNN(Region based Convolutional Neural Network)^[11]将深度学习真正地引入目标检测领域。目前,主流和成熟的检测方法分为 2 类:把预测目标作为一个盒子基于先验框的方法和将目标预测为点的无锚定方法。目标检测经典算法根据网络是否提前生成区域,可分为 R-CNN、Fast R-CNN^[3]和 Faster R-CNN^[4]等检测算法,以及 YOLO(You Only Look Once)系列^[12-16]、SSD(Single Shot Detector)系列^[17,18]和 RetinaNet^[19]等单目标检测算法。在 2015 年,Redmon 等^[12]提出了 YOLO 算法,它比其它类目标检测算法要快很多。YOLO 系列算法与其他深度学习检测算法在 Microsoft COCO(Microsoft Common Objects in Context)数据集上的测试精度与速度对比如表 1 所示。

YOLO 系列算法由 Ultralytics LLC 公司在

Table 1 Comparison of deep learning detectors

表 1 深度学习检测器对比

检测器	mAP/%	FPS	检测速度/(ms/img)
SSD	31.2	8	125
DSSD	33.2	6.4	156
R-FCN	29.9	11.7	85
FPN FRCN	36.2	5.8	172
RetinaNet	37.8	5	200
YOLOv3	33.0	19.6	51
YOLOv4	43.5	62	16
PP-YOLO	45.9	72.9	14
EfficientDet	38.9	125	8
YOLOv5s	36.4	278	3.6
YOLOv5m	43.4	250	4
YOLOv5l	46.6	167	6
YOLOv5x	48.2	117.6	8.5

2020 年更新到 YOLOv5,相比 YOLOv4,YOLOv5 检测速度更快,大约每秒可处理 140 帧图像,满足视频图像实时检测需求,体积结构更为小巧,图像检测平均准确度更高。从表 1 综合比较得出,YOLOv5m 模型可以更好地保证交叉路口监控视频检测的实时性和准确度,其网络模型结构如图 1 所示。

2 目标检测算法的研究

2.1 目标检测算法

目标检测算法现已在实际中得到了广泛的应用。目标检测是为了查找图像中感兴趣的对象,它包括 2 个子任务:确定对象的类别和位置。本文将目标检测算法分为传统的算法^[20-22]和基于卷积神经网络的算法。

传统目标检测方法主要有基于颜色、基于纹理、基于形状和一些中高级语义特征的方法,检测的最终目标就是准确地对检测目标进行检测与识别。检测步骤主要包括以下 3 个:区域选择、特征提取和分类器分类。首先将输入图像作为候选区域,使用不同大小、比例的滑动窗口以一定步长滑动;然后对每个候选区域的局部信息进行特征提取;最后使用分类器对检测的目标进行识别分类。在判断出检测目标预测框之后,可能会有一系列的目标预测框,并且这些预测框可能会有一些重叠遮挡问题。因此,需要使用非极大值抑制 NMS(Non-Maximum Suppression)的方法来对这些预

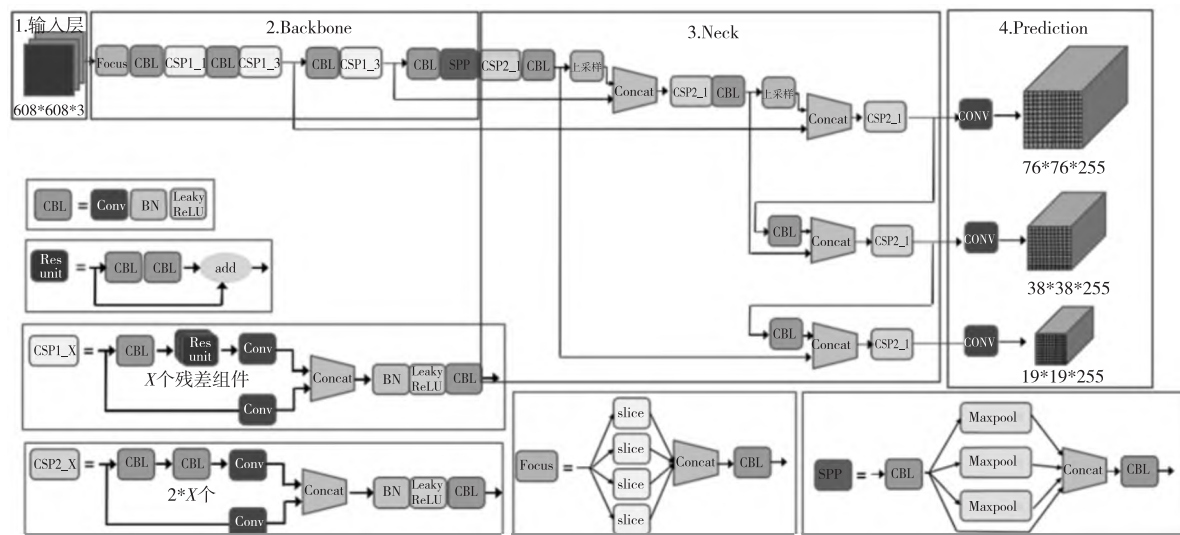


Figure 1 Structure of YOLOv5 network model
图 1 YOLOv5 网络模型结构

测框进行筛选与合并。

虽然传统的检测算法可以在特定的情况下取得较好的效果,但其主要存在 2 个方面的问题:一是滑窗选择策略没有针对性,时间复杂度高,窗口冗余,无法满足实时监控的要求;二是手工设计的特征鲁棒性较差,在天气变化、物体分布不均匀等条件下,其准确度难以保证,泛化能力较差。此外,传统的手工设计特性还需要大量的先验知识。基于卷积神经网络的对象检测算法主要分为 2 类:(1)由 RCNN^[1] 表示的两阶段算法;(2)由 YOLO 表示的基于回归的目标检测算法。由 YOLO 表示的基于回归的目标检测算法真正实现了端到端训练,一次完成目标类别的确定和定位。整个网络结构只由卷积层和输入图像组成。卷积操作后,直接返回目标类别和位置。因此,单阶段目标检测算法快于两阶段目标检测算法,特别是 YOLOv5,已达到先进的速度和精度水平。

2.2 车辆检测算法

车辆检测由于其独特的应用吸引了许多研究人员。Wang 等^[9]提出了一种基于卷积神经网络并结合颜色和深度图像的车辆检测算法。Muslu 等^[23]提出了一种基于 Haar 级联分类器和基于规则的图像处理相结合的夜间车辆尾灯检测算法,用于夜间车辆的检测。He 等^[24]提出了一种自适应检测算法,在不手动调整阈值的情况下实现自适应阴影抑制。

但是,上述基于传统目标检测的算法只能适用于特定场景,且准确度不高。随着目标检测技术的发展,Wang^[25]改进了 YOLOv3 算法,通过改进损

失函数提高了对小车区域的检测能力,通过对多尺度检测的改进,提高了对实时车辆的检测效果。陈佳倩等^[26]提出一种基于 YOLOv3 和 DeepSort 的车流量检测方法,实现了对快速移动的车辆和受环境光照影响的车辆实时监测与跟踪计数。

3 YOLOv5 算法和 DeepSort 算法模型

3.1 YOLOv5 算法模型

YOLO (You Only Look Once)的意思是神经网络只需要看 1 次图像,就能输出结果。YOLO 系列算法是单阶段目标检测的代表算法,其检测方法就是将输入图像划分为 $S \times S$ 个网格(Grid),如果目标的中心落入网格单元,则该网格单元负责检测该目标。最新的单阶段算法 YOLOv5 实际上是一个模型集群,它是 4 种模型 YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x 的通用名称。这 4 种模型的基本结构是相同的,除了瓶颈层和卷积核的数量分别由 2 个不同的参数控制,这使得这 4 种模型具有不同的深度和宽度。表 2 给出了这 4 种模型中 2 个参数的不同设置。

Table 2 Parameter settings of depth and width in the four models
表 2 4 种模型中深度和宽度的参数设置

模型	参数	
	深度	宽度
YOLOv5s	0.33	0.50
YOLOv5m	0.67	0.75
YOLOv5l	1.00	1.00
YOLOv5x	1.33	1.25

YOLOv5 的网络结构分为输入、Backbone、Neck 和输出 4 个部分。输入端包括 Mosaic 数据增强、自适应锚框计算和图像尺寸处理 3 部分。Mosaic 数据增强方式是 4 幅图像通过随机缩放、随机裁剪和随机排布的方式进行拼接。通过这种方式,增强对小目标检测的效果,使得网络的鲁棒性更好。

3.2 改进的 YOLOv5 算法模型

尽管 YOLOv5 因其体积结构小巧、处理速度快,在目标检测领域中得到大量研究与应用,但对于交叉路口中的车辆检测,因车辆密集程度高,环境复杂,其对小目标的检测效果并不太理想,故需要对 YOLOv5 进行适当改进,以满足在交叉路口密集程度下小目标的实时性检测要求。

在 YOLO 算法中,针对不同的数据集,有初始设定长宽的锚框。在网络训练中,网络在初始锚框的基础上输出预测框,进而和真实框进行对比,计算两者差距,再反向更新,迭代网络参数。YOLOv5 一般初始设定的锚框为 [10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119] 和 [116, 90, 156, 198, 373, 326]。本文使用改进的 K-means 聚类方法,针对深度学习大数据集的特点,采用最新的 Mini Batch Kmeans 来确定初始锚框,从而加速聚类过程,在车辆图像检测中更快地寻找到合适的预测框。

传统的 K-means 聚类方法是随机初始化 K 个聚类中心,然后将样本点分配到最近的中心,分为 K 个类别,再更新中心点,如此循环直到 K 个中心点不再改变或达到迭代次数。随着深度学习数据集的不断变大,传统的 K-means 聚类会更加耗费时间,可以用更加有效的 Mini Batch Kmeans 来弥补,其具体步骤为:

(1) 将数据集随机拆分为 n 份,首先从中抽取 1 份数据集,使用 K-means 聚类方法构建出 K 个聚类模型;

(2) 抽取下一份训练集中的数据,将其添加到聚类模型中,且分配给最近的聚类中心;

(3) 更新聚类中心点;

(4) 如此循环,直到中心点不再变化或达到迭代次数,停止计算。

在目标检测中,为更好地提取融合特征,通常在 Backbone 和输出层之间插入一些层,这部分称为 Neck。YOLOv5 中 Neck 采用了 FPN+PAN (Feature Pyramid Network+Pixel Aggregation

Network) 的结构,FPN 是自上而下的,利用上采样的方式对信息进行传递融合,获得预测特征图。PAN 采用的是自底向上的特征金字塔,每个阶段都将前一阶段的特征映射作为输入,并用 3×3 卷积层对其进行处理。输出通过横向连接被添加到自上而下通路的同一阶段特征图中,这些特征图为下一阶段提供信息。本文对其检测层进行改进,增加小目标检测层,改进后的结构如图 2 所示。

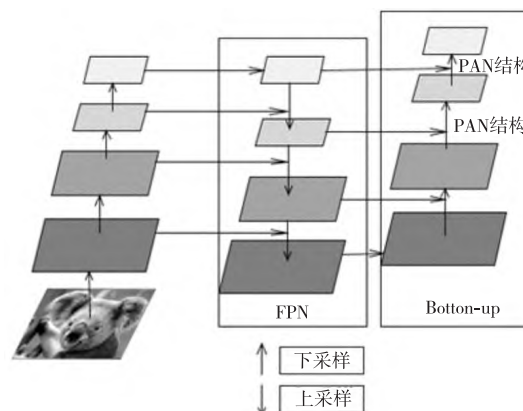


Figure 2 Improved FPN+PAN structure

图 2 改进的 FPN+PAN 结构

Prediction 也被称为输出端,其包括边界框 (Bounding Box) 损失函数和非极大值抑制 (NMS)。YOLOv5 中损失函数共包含分类损失 (Classification Loss)、定位损失 (Localization Loss) 和置信度损失 (Confidence Loss) 3 种,总体损失为三者的加权和,可通过改变权值来调整对三类损失的关注度。其中,各类损失函数分别如式 (1)~式 (3) 所示:

$$L_{\text{class}} = - \sum_{i=1}^N y_i^* \log(y_i^*) + (1 - y_i^*) \log(1 - y_i^*) \quad (1)$$

$$L_{\text{local}} = (x - x^*)^2 + (y - y^*)^2 + (w - w^*)^2 + (h - h^*)^2 \quad (2)$$

$$L_{\text{conf}} = - \sum_{i=1}^N p_i^* \log(p_i^*) + (1 - p_i^*) \log(1 - p_i^*) \quad (3)$$

式 (1)~式 (3) 中各参数信息详见文献 [14]。如式 (1) 和式 (3) 所示,分类损失和置信度损失均采用的是二元交叉熵损失函数;而定位损失 (式 (2)) 采用的是真实坐标与预测值坐标误差的平方。由于定位预测需要着重于该框与真实框的重叠区域面积,且重叠面积与两者并集面积之比越大越好,但仅使用平方损失无法很好地衡量这点,因此衍生出多个基于交并比 IOU (Intersection Over U-

nion)的损失计算公式。YOLOv5 中使用 GIOU_Loss 作为损失函数,有效解决了边界框不重合的问题,但其未解决真实框包含预测框的问题,为此衍生出 DIOU_Loss 损失函数。与 GIOU_Loss 不同,DIOU_Loss 中使用真实框和预测框中心点距离的平方与两者最小框对角线长度平方的比值作为衡量标准的一部分。DIOU 计算方法和损失函数分别如式(4)和式(5)所示:

$$DIOU(B, B_{gt}) = IOU(B, B_{gt}) - \frac{\rho^2(B, B_{gt})}{c^2} \quad (4)$$

$$L_{DIOU(B, B_{gt})} = 1 - DIOU(B, B_{gt}) = 1 - IOU(B, B_{gt}) + \frac{\rho^2(B, B_{gt})}{c^2} \quad (5)$$

其中, B 和 B_{gt} 表示真实框和预测框的中心点, ρ^2 表示两者之间的欧氏距离, c 表示 2 个框最小包围框的最短对角线长度。

考虑到预测框与真实框的长宽比例及多维度问题,研究人员提出了升级版的 CIOU_Loss 损失函数,其中又添加 2 个参数,计算方法及损失函数分别如式(6)和式(7)所示:

$$CIOU(B, B_{gt}) = IOU(B, B_{gt}) - \frac{\rho^2(B, B_{gt})}{c^2} - \alpha v \quad (6)$$

$$L_{CIOU(B, B_{gt})} = 1 - CIOU(B, B_{gt}) = 1 - IOU(B, B_{gt}) + \frac{\rho^2(B, B_{gt})}{c^2} + \alpha v \quad (7)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (8)$$

$$\alpha = \frac{v}{1 - IOU(B, B_{gt}) + v} \quad (9)$$

其中, v 为预测框和真实框长宽比例差值的归一化, $\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2$ 的值在 $0 \sim \pi^2/4$, 乘以 $4/\pi^2$ 就可以转化为 $0 \sim 1$ 。 α 为权衡长宽比例造成的损失和 IOU 部分造成的损失平衡因子。本文将多个 IOU 的计算方式相结合,首先计算 IOU,当存在 GIOU、DIOU 和 CIOU 时,进一步计算闭包区域;当存在 DIOU 和 CIOU 时,进一步计算中心点距离;当存在 CIOU 时,进一步计算参数 v 和 α ;最后取最优值作为相应的损失。在目标检测预测结果处理阶段,采用加权 NMS 操作筛选目标框,使得被遮挡的物品也可以被检测出,从而获得最优目标框。

3.3 DeepSort 算法

为实时获取运动车辆的信息,需要对机动车进

行目标跟踪定位。本文使用了 Wojke 等^[27]提出的基于检测的多目标跟踪算法 DeepSort 负责车辆目标的跟踪定位。DeepSort 是多目标跟踪算法,基本思想是 tracking-by-detection,利用运动模型和外观信息进行数据关联,使用一个 8 维空间变量 $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ 来刻画轨迹在某时刻的状态,其中, (u, v) 表示边界框的中心; γ 表示边界框的纵横比; h 表示边界框的高度; $(\dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ 表示边界框的运动信息,即它们在图像坐标中沿各自方向的速度信息。然后使用一个卡尔曼(Kalman)滤波器预测更新轨迹,该卡尔曼滤波器采用匀速模型和线性观测模型,其观测变量为 (u, v, γ, h) 。DeepSort 算法带有深度关联特征,是基于 SORT (Simple Online and Realtime Tracking)算法进行的改进,解决了在相机运动或抖动情况下 SORT 算法运动信息失效的问题。DeepSort 算法通过集成表观信息来提升 SORT 的表现,通过这个扩展,模型能够更好地处理目标被长时间遮挡的情况,将身份切换 ID_s (ID switch) 指标降低了 45%。该算法以检测结果 Bounding Box、置信度分数 (Confidence) 和特征信息 (Feature) 作为输入。其中, Confidence 主要用于检测框的筛选, Bounding Box 与 Feature 用于与跟踪器的匹配计算。该算法需要对输入的视频进行图像归一和目标检测等预处理,然后逐帧输入,其 2 大核心算法是卡尔曼滤波和匈牙利算法。通过使用标准卡尔曼滤波器对目标进行运动状态预测,更新模块部分利用 IOU 进行匈牙利算法的关联匹配,跟踪流程如图 3 所示。

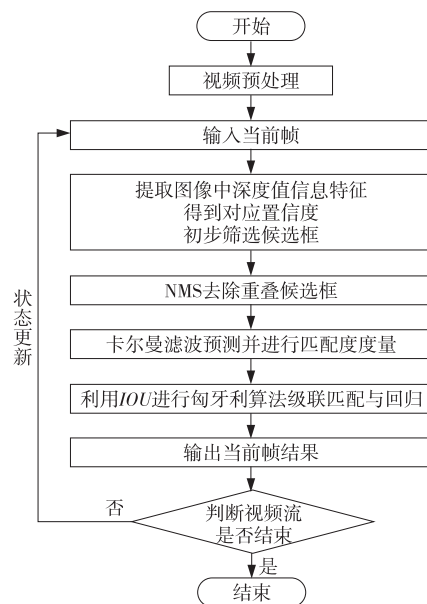


Figure 3 Flowchart of DeepSort algorithm

图 3 DeepSort 算法流程图

目标检测框与跟踪器预测框之间的匹配程度用马氏距离来描述,如式(10)和式(11)所示:

$$d^{(1)}(i, j) = (\mathbf{d}_j - \mathbf{y}_i)^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y}_i) \quad (10)$$

$$b_{i,j}^{(1)} = 1 \text{ if } d^{(1)}(i, j) \leq t^{(1)} \quad (11)$$

其中, $d^{(1)}(i, j)$ 表示第 j 个预测框结果与第 i 条轨迹的匹配度,其使用检测框和跟踪框在卡尔曼滤波器预测下的目标状态与检测值来刻画运动匹配程度,因此其也代表马氏距离; \mathbf{d}_j 代表第 j 个检测框的位置; \mathbf{y}_i 表示第 i 个目标预测框的位置; \mathbf{S}_i^{-1} 表示第 i 个目标的检测位置与预测位置之间的协方差矩阵,即通过检测位置与预测位置之间标准差的形式来考查状态测量的稳定性。式(11)是一个指示器,也可以称为门限函数,其比较的是马氏距离和卡方分布的阈值。其中, $t^{(1)}$ 由卡方分布中概率 P 和自由度 m 共同决定,本文算法中设 $P = 0.05$, 自由度 $m = 4$, 则阈值 $t^{(1)} = 9.49$ 。如果马氏距离小于该阈值,则 $b_{i,j}^{(1)}$ 为 1, 代表匹配成功。使用重识别网络提取目标的一个 128 维的特征向量作为外观特征信息,缓存匹配成功的每一帧作为特征向量集,对于第 i 个跟踪框和第 j 个检测框使用最小余弦距离来度量特征之间的距离,如式(12)和式(13)所示:

$$d^{(2)}(i, j) = \min\{1 - \mathbf{r}_j^T \mathbf{r}_k^{(i)} \mid \mathbf{r}_k^{(i)} \in R_i\} \quad (12)$$

$$b_{i,j}^{(2)} = 1 \text{ if } d^{(2)}(i, j) \leq t^{(2)} \quad (13)$$

其中, $d^{(2)}(\cdot)$ 表示外观匹配度; \mathbf{r}_j 表示对每一个检测框 \mathbf{d}_j 计算出的外观特征描述符,并且 $\|\mathbf{r}_j\| = 1$; 对于每一个跟踪轨迹 k 设置特征仓库 $R_k = \{\mathbf{r}_k^{(i)}\}_{i=1}^{L_k}$, 用来保存最近 100 条目标成功关联的特征描述符, $L_k = 100$; R_i 则表示第 i 个跟踪框特征仓库的集合; $\mathbf{r}_j^T \mathbf{r}_k^{(i)}$ 计算的是余弦相似度,而余弦距离 $= 1 - \text{余弦相似度}$,通过余弦距离来度量其特征向量和所有追踪器特征向量集,以更加准确地预测跟踪轨迹改变目标标号的次数 ID ; $b_{i,j}^{(2)}$ 也是一门限函数, $t^{(2)}$ 为阈值,一般在代码中设为 0.2 (由参数 max_dist 控制),不同的物体识别要根据实际情况进行调整。最后的综合匹配度 $c_{i,j}$ 由运动模型和外观模型加权得到,如式(14)和式(15)所示:

$$c_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j) \quad (14)$$

$$b_{i,j} = \prod_{l=1}^2 b_{i,j}^{(l)} \quad (15)$$

其中, λ 为权重系数,在代码中默认为 0,这并不是说马氏距离在 DeepSort 中毫无用处,而是马氏距离会对外观模型得到的距离矩阵进行限制,忽视掉明显不可行的分配。只有 $b_{i,j}$ 为 1 时,才认为是初步匹配(2 个顶点之间有连线),代表成功匹配。

为了将 DeepSort 与其他多目标跟踪算法进行对比分析,选取标号转换后的准确率 $MOTA$ 、平均边框重叠率 $MOTP$ 、被跟踪到的轨迹占比 MT 、跟丢的轨迹占比 ML 、跟踪轨迹改变目标标号的次数 ID 、真实轨迹被打断的次数 FM 、误报总数 FP 、未命中目标总数 FN 和运行频率 HZ 作为评价指标进行评价。NOMT (Near-Online Multi-Target Tracking) 是一种近似在线多目标跟踪算法,引入了一种新颖的聚合局部流描述符 ALFD (Aggregated Local Flow Descriptor),将跟踪问题描述为时间窗口中目标与检测目标之间的数据关联,通过使用长期兴趣点轨迹 IPT (long term Interest Point Trajectories) 来编码 2 个时间帧中检测框之间的相对运动模式。MCMOT (Multi-Class Multi-Object Tracking) 是一种新颖的多类多目标跟踪算法,通过组合检测响应和变点检测 CPD (Changing Point Detection) 算法来进行无限类多目标跟踪。EAMTT (Early Association Multi-Task Tracking) 跟踪算法是在粒子滤波框架下,将目标跟踪作为多任务稀疏学习问题,其原理是粒子模型与动态更新的字典模板线性组合,而学习问题则使用一个加速梯度的方法解决,在预测阶段之后会马上进行数据关联(EA),以降低在标签阶段的巨大计算开销。LMP (Lifted Multicut Problem) 算法主要是针对遮挡影响及行人重识别提出一种新结构,结合深度网络中提取的整体表示特征和从最先进的姿态估计模型中抽取的身体姿态进行判断,从而提高准确度。SORT (Simple Online and Realtime Tracking) 算法是一种简单的在线实时多目标跟踪算法,主要利用卡尔曼滤波将目标物体传播到未来帧中,再通过度量指标 IOU 来建立关系。DeepSort 算法和以上算法在 Mot2016 数据集上的跟踪结果对比如表 3 所示。

4 实验与结果

为验证改进的 YOLOv5 与 DeepSort 算法模型相结合算法对车辆检测跟踪计数的可行性、实时性及准确性,在 Windows 64 位操作系统下实现了该算法,选用 PyTorch 架构,对拍摄视频进行实验仿真,具体实验配置如表 4 所示。

4.1 数据集

本实验采用的是 MS COCO 和 BDD100k (a large-scale diverse driving video database) 相结合的数据集,MS COCO 起源于微软 2014 年出资标

Table 3 Comparison between DeepSort and other multi-target tracking algorithms on Mot2016 dataset

表 3 DeepSort 与其他多目标跟踪算法在 Mot2016 数据集上的对比

算法	MOTA/%	MOTP/%	MT/%	ML/%	ID	FM	FP	FN	HZ
NOMT	46.4	76.6	18.3	41.4	359	504	9 753	87 565	2.6
MCMOT_HDM	62.4	78.3	31.5	24.2	1 394	1 318	9 855	57 257	35.0
EAMTT	52.5	78.8	19.0	34.9	910	1321	4 407	81 223	12.0
LMP	48.8	79.0	18.2	40.1	481	595	6 654	86 245	0.5
SORT	59.8	79.6	25.4	22.7	1 423	1 835	8 698	63 245	60.0
DeepSort	61.4	79.1	32.8	18.2	781	2 008	12 852	56 668	40.0

Table 4 Experimental configuration parameters

表 4 实验配置参数

参数	配置
CPU	Intel(R)Core(TM)i7-9700K CPU @ 3.60 GHz
系统环境	Windows 64 位系统
语言	Python 3.7
模型	改进 YOLOv5+DeepSort 算法
GPU	NVIDIA GeForce RTX 2080
加速环境	CUDA 11.1

注的 Microsoft COCO 数据集。MS COCO 数据集是一个丰富的物体检测、分割和字幕数据集,以 scene understanding 为目标,主要是从复杂的日常场景中截取的,图像中的目标通过精确的分割进行位置标定。而 BDD100k 是 2018 年 5 月由伯克利大学 AI 实验室(BAIR)发布的,是目前规模最大、内容最具多样性的公开驾驶数据集。该数据集包括不同的天气、场景、时间、目标有无遮挡、截断、高清和模糊等多种情况,更加适应于路口车辆检测,可增强模型对车辆检测的鲁棒性。

本文通过 LabelImg 软件对数据集进行图像目标标注,标注后生成 xml 文件,再将标注文件转换为 YOLO 格式的 txt 文件,文件与图像命名一致,并将 yolov5m.yaml 文件中识别类别设置为 3,分别代表数据集内的 bus、truck 和 car。改进初始锚框,对检测部分进行改进,增加了操作层,在第 17 层后,继续对特征图进行上采样等处理,使得特征图继续变大;在第 20 层时,将获取到的大小为 160 * 160 的特征图与骨干网络中第 2 层特征图进行拼接融合,以此获取更大的特征图进行小目标检测。用训练好的检测器对不同测试数据集进行测试,从测试效果来看,在复杂环境中,检测器对数据集中的车辆检测准确度很高,具体测试效果如图 4 所示。



Figure 4 Vehicle detection effect

图 4 车辆检测效果

4.2 车辆预测跟踪及计数

在检测的基础之上,本文结合 DeepSort 算法对检测框质心进行级联匹配,完成车辆的运动轨迹检测,实现预测跟踪及计数,完成车流量检测,检测效果如图 5 所示。

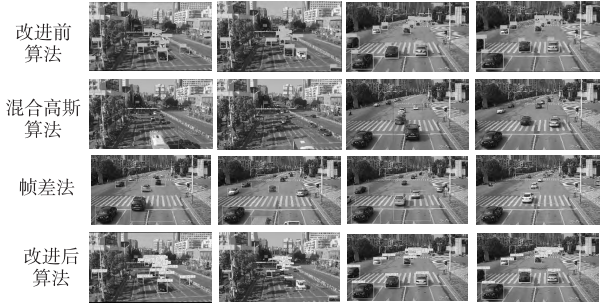


Figure 5 Comparison of detection effects of different algorithms

图 5 不同算法检测效果对比

分别采用带虚拟线圈的混合高斯法、帧差法、YOLOv5+DeepSort 算法及改进后的算法进行仿真对比,选取一段实际路口真实视频进行实验,不同算法的车流量检测性能对比如表 5 所示。

由表 5 可以看出,传统的混合高斯算法车流量检测,由于光照等原因会影响车辆检测,同时检测框的大小也会受影响,对车辆数量的检测不准确;

Table 5 Comparison of traffic flow
detection performance of different algorithms

表 5 不同算法车流量检测性能对比

算法	实际车辆数量	检测车辆数量	准确度/%
混合高斯算法	387	297	76.8
帧差法	387	234	60.5
改进前算法	387	357	92.3
改进后算法	387	374	96.6

帧差法在交叉路口红灯车辆进行等待时不能有效检测车辆,因此会影响车流量检测;而本文算法(改进后算法)在光照等复杂条件下仍可以准确地检测与跟踪车辆,可以更好地识别小目标,对远处车辆识别效果大大增强,相比原算法(改进前算法)准确度提升了 4.3%,且不影响对视频监控的实时性处理,具有较好的实用性。

5 结束语

本文详细介绍了基于改进 YOLOv5+DeepSort 算法模型的车辆检测跟踪计数方法,包括 YOLOv5 的网络结构、分类器设置、数据集处理、改进策略及 DeepSort 跟踪算法。此外,还对比分析了不同算法的车辆检测计数效果,结果表明,基于 YOLOv5m 和 DeepSort 算法的车辆检测算法在准确性和实时性方面都取得了良好的效果,这说明将这 2 种算法相结合进行车流量的统计,能有效排除其他遮挡、光照和阴影等环境影响,提高检测准确度和跟踪的稳定性。

参考文献:

[1] Huang Lin-quan, Jiang Liang-wei. Self-extending algorithm of driving area based on trajectory feature learning[J]. Modern Computer, 2020(32):27-32. (in Chinese)

[2] Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1):142-158.

[3] Girshick R. Fast R-CNN[C]//Proc of the IEEE International Conference on Computer Vision, 2015:1440-1448.

[4] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6):1137-1149.

[5] Kang K, Li H, Yan J, et al. T-CNN: Tubelets with convolutional neural networks for object detection from videos[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(10):2896-2907.

[6] Mark E, Luc G, Christopher K, et al. The pascal visual object classes (VOC) challenge[J]. International Journal of Computer Vision, 2010, 88(2):303-338.

[7] Boyuk M, Duvar R, Urhan O. Deep learning based vehicle detection with images taken from unmanned air vehicle[C]//Proc of 2020 Innovations in Intelligent Systems and Applications Conference, 2020:1-4.

[8] Li Ze-chen, Li Heng-chao, Hu Wen-shuai, et al. Masked face detection model based on multi-scale attention-driven Faster R-CNN[J]. Journal of Southwest Jiaotong University, 2021, 56(5):1002-1010. (in Chinese)

[9] Wang D, Zhou Y, Wu L, et al. Research on vehicle detection algorithm based on convolutional neural network and combining color and depth images[C]//Proc of the 2nd International Conference on Information Systems and Computer Aided Education, 2019:274-277.

[10] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Proc of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005:886-893.

[11] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition, 2014:580-587.

[12] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proc of 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015:779-788.

[13] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger [C]//Proc of 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017:6517-6525.

[14] Redmon J, Farhadi A. YOLOv3: An incremental improvement[J]. arXiv:1804.02767, 2018.

[15] Bochkovskiy A, Wang C Y, Liao H. YOLOv4: Optimal speed and accuracy of object detection [J]. arXiv, 2004. 19037, 2020.

[16] Yang Xiao-li, Duan Min, Yu Han-nian, et al. Research on real-time vehicle detection based on YOLO algorithm [J]. Instrumentation & Analysis & Monitoring, 2021(1):7-10. (in Chinese)

[17] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multi-box detector[C]//Proc of European Conference on Computer Vision, 2016:21-37.

[18] Fu C Y, Liu W, Ranga A, et al. DSSD: Deconvolutional single shot detector[J]. arXiv:1701.06659, 2017.

[19] Lin T-Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proc of 2017 IEEE International Conference on Computer Vision, 2017:2999-3007.

[20] Pawar B, Humbe V T, Kundnani L. Morphology based moving vehicle detection[C]//Proc of 2017 International Conference on Big Data Analytics and Computational Intelligence, 2017:217-223.

- [21] Bhaskar P K, Yong S P. Image processing based vehicle detection and tracking method[C]//Proc of the International Conference on Computer & Information Sciences, 2014: 1-5.
- [22] Kalyan S S, Pratyusha V, Nishitha N, et al. Vehicle detection using image processing[C]//Proc of 2020 IEEE International Conference for Innovation in Technology, 2020: 1-5.
- [23] Muslu G, Bolat B. Nighttime vehicle tail light detection with rule based image processing[C]//Proc of 2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, 2019: 1-4.
- [24] He Y F, Li J, Hua W, et al. Adaptive vehicle shadow detection algorithm in highway[C]//Proc of the 5th International Symposium on Computational Intelligence & Design, 2012: 240-243.
- [25] Wang A H. Vehicle recognition algorithm based on improved YOLOV3[C]//Proc of 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference, 2020: 2301-2305.
- [26] Chen Jia-qian, Jin Xuan-hong, Wang Wen-yuan, et al. Vehicle flow detection based on YOLOv3 and DeepSort[J]. Acta Metrologica Sinica, 2021, 42(6): 718-723. (in Chinese)
- [27] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//Proc of 2017 IEEE International Conference on Image Processing, 2017: 3645-3649.

附中文参考文献:

- [1] 黄林泉, 蒋良卫. 基于运动轨迹特征学习的行驶区域自扩充算法[J]. 现代计算机, 2020(32): 27-32.
- [8] 李泽琛, 李恒超, 胡文帅, 等. 多尺度注意力学习的 Faster R-

CNN 口罩人脸检测模型[J]. 西南交通大学学报, 2021, 56(5): 1002-1010.

- [16] 杨晓丽, 段敏, 虞汉年, 等. 基于 YOLO 算法的车辆实时检测研究[J]. 仪器仪表与分析监测, 2021(1): 7-10.
- [26] 陈佳倩, 金恒宏, 王文远, 等. 基于 YOLOv3 和 DeepSort 的车流量检测[J]. 计量学报, 2021, 42(6): 718-723.

作者简介:



贾志(1995-), 男, 河北衡水人, 硕士生, 研究方向为图像处理、深度学习和智能算法。E-mail: 1761191684@qq.com

JIA Zhi, born in 1995, MS candidate, his research interests include image processing, deep learning, and intelligent algorithm.



李茂军(1964-), 男, 湖南宁乡人, 博士, 教授, 研究方向为智能控制与智能计算、电力系统自动化。E-mail: 591338413@qq.com

LI Mao-jun, born in 1964, PhD, professor, his research interests include intelligent control & intelligent computing and power system automation.



李婉婷(1997-), 女, 湖南常德人, 硕士生, 研究方向为智能算法。E-mail: 1505429647@qq.com

LI Wan-ting, born in 1997, MS candidate, her research interest includes intelligent algorithm.