2023 年 第 23 卷 第 28 期
2023, 23(28):12159-09
科 学 技 术 与 工 程
Science Technology and Engineering
ISSN 1671—1815
CN 11—4688/T

引用格式:蒲玲玲, 杨柳. 改进 YOLOv5 的多车辆目标实时检测及跟踪算法[J]. 科学技术与工程, 2023, 23(28): 12159-12167.
Pu Lingling, Yang Liu. Improved real-time detection and tracking algorithm for multi vehicle targets in YOLOv5[J]. Science Technology and Engineering, 2023, 23(28): 12159-12167.

改进 YOLOv5 的多车辆目标实时检测及跟踪算法

蒲玲玲[1,2], 杨柳[2,3*]

(1. 西南交通大学唐山研究生院, 唐山 063000; 2. 西南交通大学综合交通大数据应用技术国家工程实验室, 成都 611756;
3. 西南交通大学信息科学与技术学院, 成都 611756)

摘 要 多车辆目标跟踪时间主要花费在车辆检测模块和对每个车辆表观特征提取模块,一般情况下,车辆检测和车辆表观特征提取是在不同的神经网络中进行的,且一张图中的车辆目标越多,对车辆表观特征提取耗费时间的也越多,推理时间也相应变长。针对这一问题,基于经典的 Tracking-By-Detection 模式,提出一种改进的 YOLO 模型:在 YOLO 网络中添加 ReID(re-identification)特征识别模块,使 YOLO 在输出目标位置信息的同时输出目标特征信息,以提高算法的跟踪速度。针对车辆间彼此覆盖的情况,提出一种基于动态 IOU 阈值的非极大值抑制算法,以提高算法的跟踪精度。最后将 YOLO 输出的信息进行数据匹配,从而实现多目标跟踪。在 UA-DETRAC 数据集上验证改进模型的有效性。实验结果表明:将 YOLOv5 网络进行改进后运用在目标跟踪算法中,相对于经典的 YOLO + DeepSORT 跟踪模型,在车辆密集的情景下平均推理时间减少了 17%;在改进后的网络上添加动态 IOU 阈值非极大值抑制,跟踪精度提高了 3.9 个百分点。改进后的模型有较好的实时性与跟踪准确率。

关键词 YOLOv5; 多目标跟踪; 目标检测; 深度学习; 非极大值抑制
中图法分类号 TP391.41; **文献标志码** A

Improved Real-time Detection and Tracking Algorithm for Multi Vehicle Targets in YOLOv5

PU Ling-ling[1,2], YANG Liu[2,3*]

(1. Graduate School of Tangshan, Southwest Jiaotong University, Tangshan 063000, China;
2. National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu 611756, China;
3. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

[**Abstract**] Multi vehicle target tracking time is mainly spent on vehicle detection module and each vehicle apparent feature extraction module. Generally, vehicle detection and vehicle apparent feature extraction were carried out in different neural networks, and the more vehicle targets in a graph, the more time was spent on vehicle apparent feature extraction, and the reasoning time was correspondingly longer. To solve this problem, based on the classic tracking by detection mode, an improved YOLO model was proposed, adding a re-identification (ReID) feature recognition module to the YOLO network, so that YOLO can output target feature information while outputting target location information, so as to improve the tracking speed of the algorithm. Aiming at the situation that vehicles cover each other, a non maximum suppression algorithm based on dynamic IOU threshold was proposed to improve the tracking accuracy of the algorithm. Finally, the information output by YOLO was matched with data to realize multi-target tracking. The effectiveness of the improved model was verified on the UA-DETRAC data set. The experimental results show that compared with the classic YOLO + DeepSORT tracking model, the average reasoning time in vehicle intensive scenarios is reduced by 17% when the YOLOv5 network is improved and applied to the target tracking algorithm. Adding dynamic IOU threshold non maximum suppression to the improved network, the tracking accuracy is improved by 3.9 percentage points. The improved model has better real-time performance and tracking accuracy.
[**Keywords**] YOLOv5; multi-target tracking; object detection; deep learning; non maximum suppression

对于车辆的跟踪研究属于目标跟踪问题中的多目标跟踪问题,多目标跟踪的解决步骤一般为两步,目标检测和目标数据关联。先使用目标检测算法将感兴趣的目标进行定位和分类,再使用目标数据关联算法,将不同帧之间的相同目标进行关联,以确定不同帧中的目标是否为同一目标。在多目

收稿日期: 2022-10-21 **修订日期**: 2023-07-05
基金项目: 四川省科技创新基地(平台)和人才计划项目(2022JDR0356);四川省科技计划项目(软科学项目)(2021JDR0101);宜宾市双城市校协议专项科研经费科技项目(SWJTU2021020005)
第一作者: 蒲玲玲(1998—),女,汉族,四川遂宁人,硕士研究生。研究方向:深度学习。E-mail:pll01@foxmail.com。
*通信作者: 杨柳(1978—),女,汉族,四川达州人,博士,高级工程师。研究方向:移动通信与工程信息化。E-mail:yangliu@swjtu.edu.cn。

标跟踪问题上,除了目标检测算法的性能对跟踪性能影响大以外,前后帧之间的目标数据关联也同样影响着跟踪的性能。多目标跟踪发展到现在,根据Re-ID(re-identification)模块是否融入目标检测网络,可分为DBT(detection-based tracking)和JDT(joint detection tracking)两类[1],前者将目标检测和Re-ID模块分为两个网络实现,具有较高的准确率,是当前基于深度学习的视觉多目标跟踪的主流方法;后者将DBT两模块联合,具有较高的运行速度,是近两年发展的新趋势。

如何提高多目标跟踪的推理速度一直是近年来研究的热点。Wojke等[2]提出了DeepSORT算法来进行多目标跟踪。毛昭勇等[3]使用EfficientNet作为YOLOv3的骨干网以提高检测速度,将推理耗时的骨干网替换为轻量级网络,减小目标检测网络的推理时间。武明虎等[4]将YOLOv3的损失函数和网络结构进行改进后,与SORT算法相结合进行目标跟踪,跟踪速度最快可达14.39 fps。Zuraimi等[5]将YOLO和DeepSORT用于道路上的车辆检测和跟踪,权衡精度和时间后,证明了使用YOLOv4要好于YOLOv3,在GTX 1600ti上的以YOLOv4为目标检测网络的DeepSORT的总体跟踪速度为14.12 fps。Zhang等[6]提出了FairMOT算法,该算法在大小为输入图像的1/4的高分辨率特征图上使用无锚框的方式进行目标检测和目标外观特征提取,使特征图上的目标中心更准确地对齐到原图上,获取更准确的目标外观特征。Liang等[7]为避免使用JDT方式进行目标跟踪时,两个学习任务在一个网络中出现恶性竞争问题,提出了CSTrack模型,以推动每个分支更好地学习不同的任务。CSTrack模型在单GPU上推理速度为16.4 fps,有较高的多目标跟踪速度。

YOLO发展至今已有多个版本,YOLOv5基于他的灵活性和较低的推理时延,多位学者将其应用在目标检测和跟踪领域。赵桂平等[8]以YOLOv5框架为基础,借鉴了两步法的优点,在边框生成方面进行改进,提高了YOLOv5的检测精度。Wang等[9]使用YOLOv5s进行目标检测,使用SiamRPN进行单目标跟踪,综合跟踪速度为20.43 fps。Neupane等[10]使用YOLOv5进行检测,使用质心算法进行跟踪,跟踪速度最高可达38 fps,但是仅在手工绘制的一小块跟踪框内进行跟踪,不进行长时间跟踪。黄战华等[11]使用YOLOv5m进行检测,使用位置+声源信息进行跟踪,在平均2个目标的情况下跟踪速度达到34.23 fps,但在跟踪时只能定位一个声源,多声源情况下会存在干扰,多目标跟踪时,跟踪精度差。张文龙等[12]使用EfficientNet、D-ECA(DCT-efficient channel attention module)注意力模块、AFN(associative fusion network)改进YOLOv5,平均跟踪速度为10.84 fps。张梦华等[13]为解决多个行人交错运动时出现跟踪错误的问题,在使用YOLOv5+DeepSORT对目标进行跟踪后,再引入ReID(re-identification)技术去纠正行人的运动轨迹,提高跟踪的精度。此方法在YOLOv5+DeepSORT跟踪方式上额外增加了ReID网络,虽然提高了跟踪精度,但由于引入更多的神经网络,故需要消耗更多的推理时间。以上基于YOLOv5的检测+跟踪算法,均有较快的跟踪速度,但可能更适用于少目标或单目标跟踪,用于多目标跟踪时跟踪精度和推理速度还需再提高。

基于上述方法启发,提出ReID特征识别模块,将该模块添加到YOLOv5网络中,使YOLOv5网络在输出目标边界框的同时输出目标的特征信息,以代替DeepSORT中的特征提取网络,从而提高目标跟踪的推理速度。同时提出一种基于动态IOU(intersection over union)阈值的非极大值抑制算法,该算法根据每个边框的置信度为每个边框设置不同的IOU阈值,以提高目标彼此覆盖场景下的跟踪精度,对较多目标进行更准确的实时跟踪。

1 基于YOLO的多目标跟踪算法

对于车辆的多目标跟踪为目标检测和目标跟踪两步。目标跟踪主要是将目标的位置和表观特征等信息进行关联,得出相邻帧之间那些目标是同一个目标。目标检测的工作主要是在图像中找出感兴趣的目标对其进行定位和分类。基于深度学习的目标检测模型发展到现在,可以分为以R-CNN(region-CNN)为代表的基于候选框(two-stage)的算法模型和以YOLO为代表的基于回归(one-stage)的算法模型两类[14],前者算法模型在精度上高于后者,后者在推理时间的表现上好于前者。使用目标检测算法对道路上车辆的检测,除了对算法的精度有要求外,还需对其推理时间有一定要求。故选择基于回归的一步式YOLO目标检测算法来保证较小的推理时间。

1.1 YOLO目标检测算法

YOLO网络发展至今已有多个版本,其结构一般分为骨干网、特征融合层、3个不同大小的预测头。其用3个不同大小预测头分别检测不同大小的目标,以在追求速度的同时保证精度。

YOLOv4基于YOLOv3网络结构进行改进,在提高精度的同时也提升了推理速度。YOLOv4为降低主干网的推理时间,在残差网络的基础上组合成CSP(cross-stage-partial)模块,CSP模块能够在加快

模型推理速度的同时尽可能的保持精度性;使用 SPP(spatial pyramid pooling)结构将不同大小的特征进行融合,增加网络的感受野,有利于检测出图像中不同大小的目标;为将特征进行充分融合,设计 PANet(path aggregation network)特征融合网络,相比于 YOLOv3 采用的 FPN(feature pyramid networks)自顶向下单向特征融合方法,PANet 采用自底向上再自顶向下的双向融合方法,获得更丰富的目标特征信息。

YOLOv5 同 YOLOv4 一样使用 CSPDarknet、Neck 和 3 个输出头的网络结构,模型架构与 YOLOv4 相似。YOLOv5 有 s、m、l、x 4 种大小的结构相同,但宽度和深度不同的模型可供使用,4 种模型推理速度依次降低,推理精度依次升高。在训练时 YOLOv5 使用 Mosaic 数据增强将 4 张图片合并为一张图片进行输入,减小训练花费的时间同时变相增大 batch_size;使用自适应锚框计算,能在小目标的检测上有更好效果;在主干网中使用 focus 结构,起到减少计算量和提高速度的作用。YOLOv5 有两种 CSP 结构,在主干网中使用 CSP1 结构,在 Neck 网络中使用 CSP2 结构,以此加强特征融合能力和减少计算量。且 CSP 结构深度随模型的深度变化而变化。

1.2　改进 YOLOv5 的多目标跟踪网络结构

JDE(joint detection and embedding)算法[15]通过在 YOLOv3 的 3 个预测头中添加特征层,将对目标特征的输出也交给 YOLO 网络,提高了多目标跟踪的推理速度,但在 3 个不同分辨率的特征层上面进行特征提取,当相邻帧中同一目标大小变化明显时,可能检测结果是不同的预测头输出的,由于检测头的分辨率不同,从而导致同一目标在相邻帧获取到的特征相差过大,导致跟踪失败。FairMOT[6]算法为了提取到更准确的目标特征,将输入图片设置为 1 088×608,在 1/4 原图大小的特征图上进行目标位置预测和特征提取,跟踪精度得到了提升,但输入图片分辨率大且在较大特征图上预测和特征提取会非常耗时,导致跟踪速度过慢。基于此,将 YOLOv5 模型进行改进,使 YOLOv5 在输出目标位置信息的同时输出目标的特征信息。改进后的 YOLOv5 网络结构如图 1 所示,在 YOLOv5 网络中添加 ReID 模块,该模块由特征输出模块和 ID(identity)分类模块共同模块组成。

1.2.1　特征输出模块

神经网络学习过程中,低层特征分辨率更高,包含更多的位置、细节、颜色等信息,但由于经过的卷积较少,其语义信息含量低,噪声更多;高层特征包含更多的语义信息,但其分辨率低,对细节的感知能力较差。且如果在较小的特征图(1/8、1/16、1/32)上提取表观特征,会因为特征图分辨率太小,导致特征图上的目标中心不能很好地与原图中心对

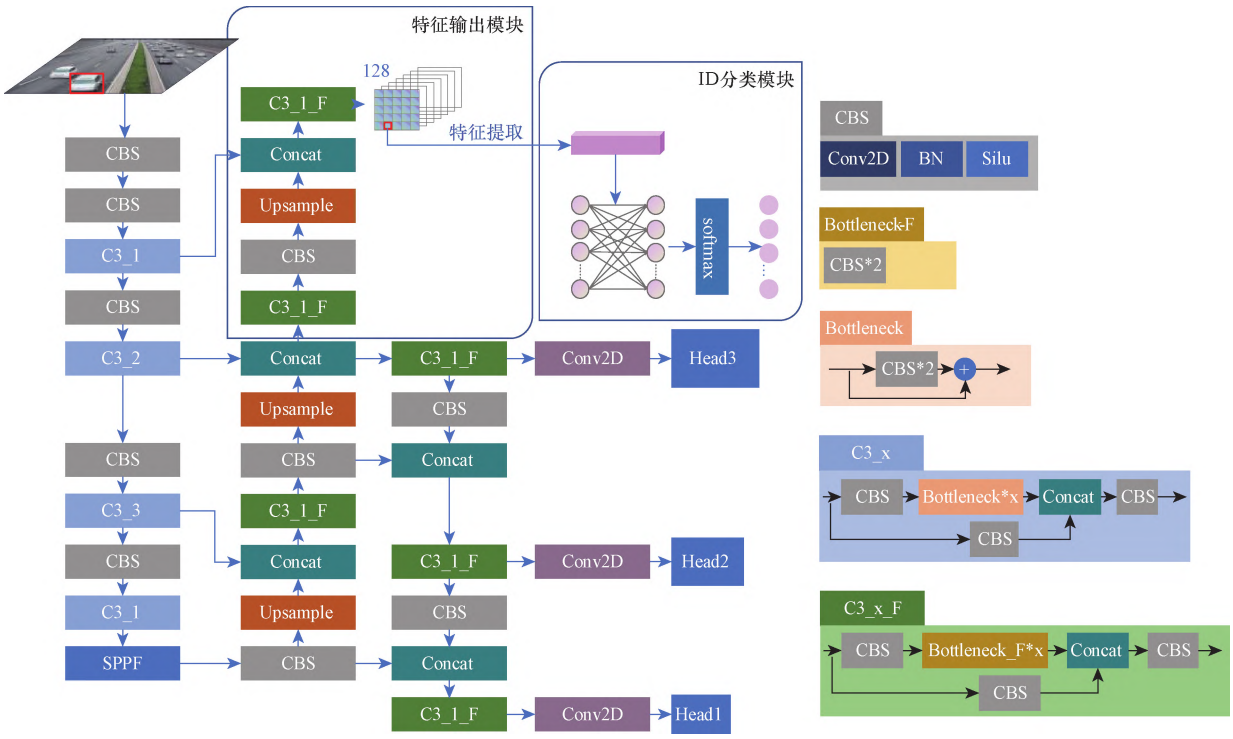


图 1　改进后的 YOLO 网络结构

Fig. 1　Structure of improved YOLO

齐,出现表观特征提取粗糙的问题;如果在较深层特征层上提取特征,会获得更多的语义信息,较少的颜色和细节等浅层信息,导致不同目标的提取到的特征不能很好地区分。

以上两种提取方式均容易造成目标跟踪错误。故特征输出模块为使目标能够提取到较为精细准确的特征,将 PANet 继续自底向上进行上采样到1/4原图大小,再与主干网中 1/4 原图大小的特征进行融合,最后输出原图 1/4 大小的 128 维的特征图,作为提取目标特征的特征池。与浅层 1/4 原图大小的特征层进行融合后,特征池不仅包含丰富的语义信息,也包含丰富的颜色细节等信息,可为不同目标提供有区别的特征。且得益于特征池的高分辨率,能为邻近目标提供精准的特征。

在训练阶段使用标签直接定位到特征池 128 维特征,如图 1 所示的网络框架图中检测图上的用红色框框出的车辆位置,对应到特征池中的红色框框出的单位 128 维特征;在推理阶段提取目标特征时需结合 YOLO 输出的目标位置信息,定位到特征池单位 128 特征。使用目标的位置信息在特征池中定位目标特征,提取到和目标唯一对齐的单位 128 维特征作为该目标的特征。

1.2.2 ID 分类模块

ID 分类模块用以训练网络的目标特征识别能力,使网络能够输出正确的和不同目标有区别的目标特征,其仅在训练模型时使用。ID 分类模块设计为两层的全连接层,设计 128 个节点进行输入,ID 总数个节点进行输出,将特征识别问题转换为 ID 分类问题。在训练模型时,将目标的真实框中心定位到特征池上,从特征池中提取该定位处的 128 维特征,将该 128 维特征作为全连接层的输入,将 ID 总数数量的节点作为全连接层的输出,并且采用 softmax对输出的数据进行归一化,获得该目标的类别概率。然后使用交叉熵损失函数计算 ID 损失。softmax 函数表达式为

$$\mathrm{softmax}(z_i) = \frac{\mathrm{e}^{z_i}}{\sum_{c=1}^{C}\mathrm{e}^{z_c}} \tag{1}$$

式(1)中:z_i 为第 i 个节点的输出值;z_c 为第 c 个节点的输出值;C 为输出节点的总数。

1.3 动态 IOU 阈值的非极大值抑制算法

公路监控安装并非总是正对车辆,如果摄像头安装在公路边侧,离摄像头较远的哪一方公路上的车辆在视频里会很容易出现彼此覆盖的情况。对于彼此覆盖的目标检测,针对不同的问题有不同的解决方法。其中,针对覆盖场景构建对应数据集,直接进行训练,但是由于覆盖的多样性和数据的难收集性这种方式非常困难;如果是神经网络提取的特征不充分,可以针对损失函数和网络结构进行改进;如果是网络特征提取充分但是在覆盖场景下仍表现不好,可以考虑是非极大值抑制(non-maximum suppression,NMS)将相邻的检测框过滤了。

车辆经过目标检测网络检测后,非极大值抑制会基于边框置信度和 IoU(intersection over union)阈值抑制重叠的边框。当相邻目标的检测框彼此重叠面积较大,会出现只留下一个目标的检测框,另一个目标的检测框被抑制的情况,导致对其中一个目标检测失败,进而导致跟踪失败。为此,提出一种基于动态 IoU 阈值的非极大值抑制(dynamic non-maximum suppression,DNMS)算法。DNMS 算法对 YOLO 预测到的边框赋予一定的信任,根据每个边框的置信度得分,对每个边框设置不同的 IoU 抑制阈值。对边框置信度得分越大的置于更多的信任,设置更大的 IoU 过滤阈值,来保留相互大面积覆盖的正确边框。

DNMS 算法中引入两个超参数 supC 和 supT,分别用作过滤掉置信度较低的边框和根据置信度设置 IOU 阈值,两个参数的数值可根据数据集等实验因素自由确定。当计算出的 IOU 阈值小于 0.35 时,设置为 0.35,以避免不同车辆边框覆盖面积很小就被抑制掉。DNMS 算法伪代码如下,其中 N_i 为第 i 个边框的 IOU 阈值。

DNMS 算法

Input: The list of initial detection boxes, $B = \{b_1, b_2, \cdots, b_N\}$; The list contains corresponding detection scores, $S = \{s_1, s_2, \cdots, s_N\}$;
Output: The list of remaining detection boxes after DNMS, $D = \{b_1, b_2, \cdots, b_n\}$; The list contains corresponding detection score $C = \{s_1, s_2, \cdots, s_n\}$
$D = \{\}$; $C = \{\}$
while ($B \neq$ Empty) do
$m = \mathrm{Max}(S)$
$M = b_m$; $N = s_m$
$D = D \cup M$; $C = C \cup N$; $B = B - M$; $S = S - N$
for b_i in B do
$N_i = (s_i - \mathrm{sup}C) * \mathrm{sup}T$
if $0.35 > N_i > 0$ then
$N_i = 0.35$
end if
if IoU (M, b_i) > N_i then
$B = B - b_i$; $S = S - s_i$
end if
end for
end while
return D, C

1.4　损失函数设计

对于 ID 分类模块,采用交叉熵损失函数进行损失计算,其余损失函数采用 YOLOv5 原设计损失函数。

ID 分类模块损失函数为

$$L_{\mathrm{id}} = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic}\ln p_{ic} \tag{2}$$

式(2)中:M 为类别的数量;当样本 i 真实类别为 c 时 y_{ic} 取 1,否则取 0;p_{ic} 为样本 i 属于类别 c 的概率;N 为样本总数。

对于目标检测的置信度损失 L_{conf},为二元交叉熵损失,可表示为

$$\begin{cases} L_{\mathrm{conf}}(o,c) = -\dfrac{\sum_{i}\left[o_i\ln\hat{c}_i + (1-o_i)\ln(1-\hat{c}_i)\right]}{N} \\ \hat{c}_i = \mathrm{Sigmoid}(c_i) \end{cases} \tag{3}$$

式(3)中:$\hat{c}_i$ 为 c_i 经过 Sigmoid 函数后的输出,是损失函数的一个中间变量;Sigmoid 为 Sigmoid 函数;$o_i \in [0,1]$ 为第 i 个预测目标边界框与真实目标边界框的 IOU;o 为真实值;c 为预测值。

对于目标检测的类别损失 L_{cls},为二元交叉熵损失,可表示为

$$\begin{cases} L_{\mathrm{cls}}(o,c) = -\dfrac{\sum_{i\in\mathrm{pos}}\sum_{j\in\mathrm{cls}}\left[o_{ij}\ln\hat{c}_{ij} + (1-o_{ij})\ln(1-\hat{c}_{ij})\right]}{N_{\mathrm{pos}}} \\ \hat{c}_{ij} = \mathrm{Sigmoid}(c_{ij}) \end{cases} \tag{4}$$

式(4)中:$o_{ij} \in \{0,1\}$ 为预测目标边界框 i 中是否存在第 j 类目标;$\hat{c}_{ij}$ 为预测值;N_{pos} 为边框数,下标 pos 为预测的边框;下标 cls 为类别。

对于目标检测的位置损失 L_{loc},为 GIoU 损失,可表示为

$$\begin{cases} \mathrm{GIoU} = \mathrm{IoU} - \dfrac{A^{\mathrm{c}} - U}{A^{\mathrm{c}}} \\ L_{\mathrm{loc}} = 1 - \mathrm{GIoU} \end{cases} \tag{5}$$

式(5)中:IoU 为预测框和真实框的交并比;A^{c} 为同时包含预测框和真实框最小矩形面积;U 为预测框和真实框的并集。

目标检测的损失和可表示为

$$L_{\mathrm{det}} = L_{\mathrm{conf}}(o,c) + L_{\mathrm{cls}}(o,c) + L_{\mathrm{loc}} \tag{6}$$

最后,总的损失函数可表示为

$$L_{\mathrm{total}} = \frac{1}{2}\left[\frac{1}{\mathrm{e}^{w_{\mathrm{det}}}}L_{\mathrm{det}} + \frac{1}{\mathrm{e}^{w_{\mathrm{id}}}}L_{\mathrm{id}} + w_{\mathrm{det}} + w_{\mathrm{id}}\right] \tag{7}$$

式(7)中:w_{det} 和 w_{id} 分别为通过任务的独立不确定性自动学习方案[16]学习得到的目标检测损失权重和 ID 类别损失权重。

1.5　多目标跟踪

基于改进 YOLO 的多车辆场景目标跟踪的跟踪流程图如图 2 所示。视频流输入后,会先使用改进后的 YOLO 算法检测出每帧中目标的位置和特征信息。然后将目标的位置信息采用卡尔曼滤波进行预测,预测目标的下一帧位置,使用当前帧目标位置和上一帧目标预测位置进行马氏距离计算,得到目标和轨迹的位置距离;使用当前检测的特征和轨迹近 100 个特征进行余弦距离计算,取最小的距离作为目标和轨迹之间的特征距离。之后将获得的两个距离进行数据匹配来判断相邻帧的目标是否为同一目标。匹配算法主要使用匈牙利匹配和 IOU 匹配,先使用匈牙利对目标和轨迹之间的位置信息和特征信息进行匹配,若匹配成功则将该目标直接加入到轨迹,若匹配失败,则再进行 IOU 匹配,匹配成功则加入轨迹,匹配失败则创建新轨迹。

提出的多目标跟踪算法未特别说明处均使用多目标跟踪流程进行跟踪。

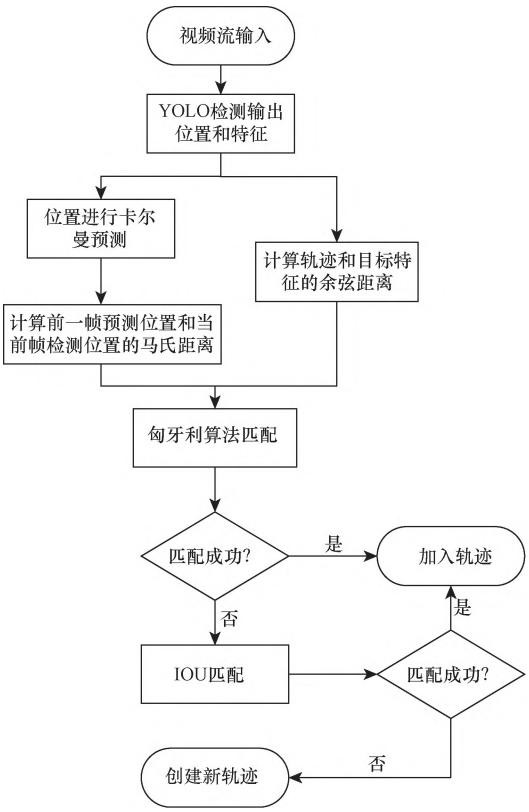


图 2　跟踪流程图

Fig. 2　Flow chart of tracking

2　实验结果与分析

2.1　实验环境

实验操作系统为 64 位 Windows10;硬件环境主要包括:Intel(R) Xeon(R) W-2223 CPU @

3.60 GHz、内存 32 GB、训练环境中显卡型号为 NVIDIA TITAN xp,验证环境显卡型号为 NVIDIA Quadro P2200、深度学习框架为 pytorch。训练的数据集采用 UA-DETRAC[17]公开数据集。在训练改进后的 YOLOv 5 m 时采用在 coco 数据集上训练好的权重结果作为预训练权重,训练批次(batch_size)设置为 16,训练轮数(epoch)设置为 50 个。

2.2 数据集与评判指标

数据集采用 UA-DETRAC 公开数据集,该数据集是车辆检测和跟踪的大规模数据集,数据集主要拍摄于北京和天津的道路过街天桥,并手动标注 8 250 个车辆和 121×10^4 个目标对象外框[17]。车辆分为:轿车、公共汽车、厢式货车和其他车辆。天气情况分为:多云、夜间、晴天和雨天。在使用该数据集前将原始数据的 xml 格式标签转换为 YOLOv5 所需的标签格式,因为需要进行 ID 识别,故需要在标签中包含类别和位置的同时添加目标的 ID。YOLOv5 修改后的标签格式定义为:<目标类别、目标 x 坐标中心、目标 y 坐标中心、目标宽、目标高、ID>。

为评价跟踪的性能,采用 Dendorfer 等[18]提出的多目标跟踪算法评价指标进行评价,主要选取多目标跟踪精度(multiple object tracking accuracy, MOTA)、MT(mostly tracked)、ML(mostly lost)、IDs(ID switch)、假阳性(false positive, FP)、假阴性(false negative,FN)作为评估指标,另添加每秒帧数(frame per second,FPS)评估模型每秒处理的帧数。其中 MOTA 用于评价多目标跟踪的精准度,用以统计在跟踪过程中误差的积累情况,其表达式为

$$\mathrm{MOTA}=1-\frac{\sum_t(\mathrm{FN}_t+\mathrm{FP}_t+\mathrm{IDs}_t)}{\sum_t\mathrm{GT}_t}\times100\% \tag{8}$$

式(8)中:FN_t、FP_t、IDs_t、GT_t分别为第 t 帧时 FN、FP、IDs、GT 指标的数值;t 为帧数。

2.3 结果与分析

2.3.1 YOLOv5 + ReID 跟踪实验

为提高多目标跟踪模型的推理速度,在目标检测网络上选择推理速度更快的 YOLO。YOLO 有多个版本,将 ReID 模块添加到 YOLOv4 和 YOLOv5 上,与使用 YOLOv4 和 YOLOv5 为目标检测网络的 DeepSORT 算法(分别简写为 YOLOv4 和 YOLOv5)对比,实验结果如表 1 所示。结果表明,将 ReID 模块添加到 YOLOv4 上,与以 YOLOv4 为目标检测网络的 DeepSORT 相比,MOTA 指标下降了 4 个百分点,但推理速度有一定的提升。将 ReID 添加到 YOLOv5 上,与以 YOLOv5 为目标检测网络的 DeepSORT 对比,MOTA 指标值不变,但推理速度却明显快于后者。YOLOv5 相对于 YOLOv4 有更快的推理速度,并且在添加 ReID 模块后,得益于 YOLOv5 丰富的特征提取能力,MOTA 指标并没有下降,故选择 YOLOv5 作为多目标跟踪网络。

多目标跟踪算法的推理时间主要花费在目标检测和特征提取上,经过本文方法将目标检测模型进行优化,使目标检测模型在输出目标位置信息的同时输出目标的特征信息,从而提升模型在进行多目标跟踪时的推理速度。特别是在目标多的情况下,与原算法的推理时间对比越明显。以使用 YOLOv5 作为目标检测网络的 DeepSORT[2]和 YOLOv5 + ReID 为例,如表 2 所示的两种模型在不同目标数量下的推理时间对比,可以看出,在 2 ~ 4 个目标的情况下,所提出的 YOLOv5 + ReID 算法和 DeepSORT 算法的推理时间差值为 3.83 ms,在 14 ~ 16 个目标的情况下,推理时间差值达到 17.97 ms,推理时间差随目标的增多而增大。

JDE[15]算法在 3 个不同分辨率的特征层上面进行特征提取,当相邻帧中同一目标大小变化明显时,可能检测结果是不同的预测头输出的,由于检测头的分辨率不同,从而导致同一目标在相邻帧获取到的特征相差过大。如图 3(a)所示,在左侧图像上两车辆 ID 分别为 67 和 68,右侧图片上则为 69 和 70,发生 ID 切换,对同一车辆跟踪失败。所提出的 YOLO + ReID 算法中,不同预测头预测的目标均在同一个特征池中提取特征,在相邻帧中目标大小变化明显时,无论检测结果是那个输出头输出的,都根据输出头输出的目标位置信息在同一个特征池中定位特征。如图 3(b)所示,在左侧图像上两车辆

表 1 ReID 模块应用

Table 1 ReID module application

模型	MOTA/%	FPS
YOLOv4 + ReID	74.8	11.57
YOLOv4	78.8	11.13
YOLOv5 + ReID	86.4	17.91
YOLOv5	86.4	14.87

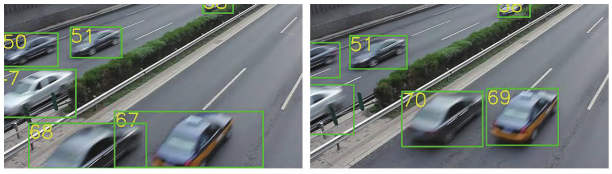
表 2 不同目标数量下的推理时间对比

Table 2 Comparison of reasoning time under different number of targets

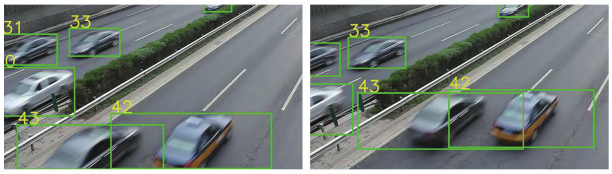
目标数/个	DeepSORT[2]/fps	YOLOv5 + ReID/fps	差值/ms
2 ~ 4	22.42	24.26	3.83
3 ~ 5	20.46	22.90	5.21
6 ~ 8	17.19	20.34	9.01
9 ~ 10	14.76	18.22	12.86
10 ~ 13	14.19	17.58	13.58
14 ~ 16	11.89	15.12	17.97

ID 分别为 42 和 43,右侧图片上仍为 42 和 43,没有发生 ID 切换,对同一车辆跟踪成功。

采用 IDs 指标来评判 JDE 算法和 YOLOv5 + ReID 算法对目标 ID 切换的数量。验证数据采用 UA-DETRAC 数据集中 6 个不同场景的视频段:MVI_20011、MVI_30761、MVI_40192、MVI_40241、MVI_63544、MVI_63563 进行验证。实验结果如表 3 所示,JDE 算法的 IDs 为 399,YOLOv5 + ReID 算法的 IDs 为 65,两者相比,YOLOv5 + ReID 的目标 ID 切换量明显较 JDE 少。



(a) JDE



(b) YOLOv5+ReID

图 3　IDs 对比

Fig. 3　IDs comparison

表 3　IDs 数量对比

Table 3　Comparison of IDs quantity

模型	IDs
JDE[15]	399
YOLOv5 + ReID	65

2.3.2　DNMS 算法检测实验

为验证 DNMS 算法对彼此遮挡目标的检测有效性,选取 UA-DETRAC 数据集中摄像头倾斜于车道布设所获得的视频段:MVI_63544 和 MVI_40241 作为验证数据,其中 MVI_40241 视频段中的车流量相对 MVI_63544 视频段中的车流量大,车辆总数也远高于 MVI_63544 视频段。使用 NMS 算法、DIOU-NMS[19]算法、Soft-NMS[20]算法和 DNMS 算法进行对比,采用 FN(false negative)和 FPS 指标来评判算法在两个视频段上漏检的车辆数量。表 4 为非极大值抑制算法对比结果。在对车辆的漏检上,本文算法在 MVI_40241 上漏检量为 1 021,明显少于对于其他 3 种算法,在 MVI_63544 视频段中的漏检数为 172,与其他算法持平。在推理速度上,所提出的 DNMS 算法相对于 NMS 算法,由于计算量更大,故在推理速度上会比较慢,但相对于 DIoU-NMS 和 Soft-NMS 算法,本文算法更有优势。

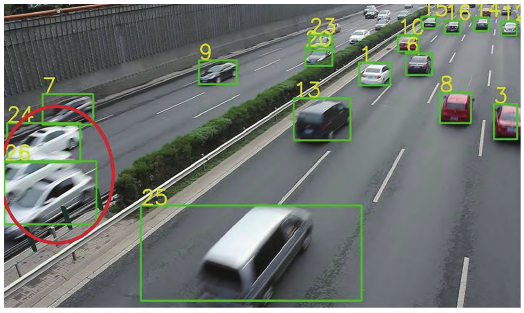
针对彼此覆盖的目标,采用 DNMS 算法代替原

表 4　非极大值抑制算法对比

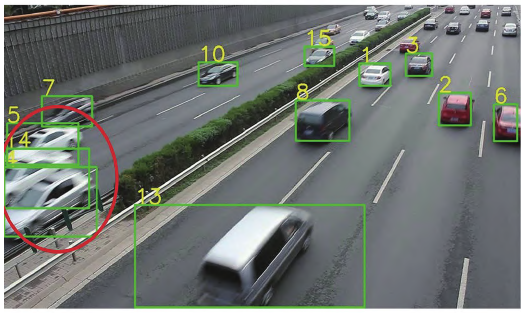
Table 4　Comparison of non maximal suppression algorithms

模型	FN_{MVI_63544}/辆	FN_{MVI_40241}/辆	FPS
YOLOv5 + NMS	178	1 333	19.29
YOLOv5 + DIoU-NMS[19]	171	1 310	4.12
YOLOv5 + Soft-NMS[20]	171	1 310	6.55
YOLOv5 + DNMS	172	1 021	17.78

注:FN_{MVI_40241} 和 FN_{MVI_63544} 分别为模型在 MVI_40241 视频段和 MVI_63544 视频段漏检的车辆数量。



(a) YOLOv5+ReID+NMS



(b) YOLOv5+ReID+DNMS

(c) JDE+NMS

(d) JDE+DNMS

图 4　DNMS 实验结果

Fig. 4　DNMS experimental results

有的 NMS 算法。分别在所提出的 YOLOv5 + ReID 算法和 JDE[15]算法上验证 DNMS 对彼此遮挡目标检测的有效性。实验结果如图 4 所示,其中,图4(a)、图 4(c)使用 NMS 算法进行边框抑制,图4(b)、图4(d)使用本文提出的 DNMS 算法进行抑制。可以看出,在图 4(a)和图 4(c)中的红色椭圆框中,框出的车辆里,有一个车辆无检测框,而在图4(b)和图4(d)中,该车辆的检测框得以出现。实验结果表明,使用本文提出的 DNMS 算法能够检测到 NMS 算法不能检测到的车辆。

采用 FN 指标来评判使用 DNMS 的模型分别在 MVI_40241 视频段和 MVI_63544 视频段,漏检的车辆数量。实验结果如表 5 所示。YOLOv5 + ReID 使用 DNMS 算法后在两个视频段上漏检分别减少 312 和 5;JDE 使用 DNMS 算法后在两个视频段上漏检数量分别减少 22 和 2。实验结果表明,将所提出的 DNMS 算法用在其他模型上也仍然有效。

2.3.3 改进 YOLOv5 的目标跟踪实验

使用以 YOLOv5 为目标检测网络的 DeepSORT[2]和 JDE[15]、FairMOT[6]算法和本文算法,在 UA-DETRAC 数据集上 6 个不同场景的视频段,共 9 389 帧上进行推理精度和速度的对比。

实验结果如表 6 所示。FPS 指标在每张图的跟踪数量为 7 ~ 16 上计算得到。由表 6 可知,将 YOLOv5 网络结构加上 ReID 模块,和基于无锚点预测边框和在 1/4 原图大小上直接进行目标预测的 FairMOT 算法对比,在 UA-DETRAC 数据集上,本文算法的 FPS 为 17.91,明显高于该算法的 5.71。和 JDE 算法相比,所提的 YOLOv5 + ReID 算法的 IDs 为 65,明显小于 JDE 算法的 399。和 DeepSORT 算法相比,本文的 YOLOv5 + ReID 的平均推理时间减少了 11.41 ms。

将 DNMS 算法运用在 YOLOv5 + ReID 模型上,相对于 YOLOv5 + ReID 模型,跟踪精度 MOTA 提升了 3.9 个百分点。实验结果表明,所提出的 YOLOv5 + ReID + DNMS 算法相对于其他算法,在保证推理速度的情况下,在跟踪精度上有明显的优势。

表 5 漏检数量对比

Table 5 Comparison of missing inspection quantity

模型	FN_{MVI_40241}/辆	FN_{MVI_63544}/辆
YOLOv5 + ReID + NMS	1 333	178
YOLOv5 + ReID + DNMS	1 021	172
JDE + NMS	1 437	183
JDE + DNMS	1 415	181

注:FN_{MVI_40241} 和 FN_{MVI_63544} 分别表示模型在 MVI_40241 视频段和 MVI_63544 视频段漏检的车辆数量。

表 6 多目标跟踪算法的性能对比

Table 6 Performance comparison of multi-target tracking algorithms

模型	MOTA ↑/%	MT ↑ /%	ML ↓/%	IDs ↓	FP ↓	FN ↓	FPS ↑
FairMOT[6]	90.0	98.2	0.06	25	5 260	1 866	5.71
DeepSORT[2] + YOLOv5	86.4	94.7	0.05	80	6 284	3 504	14.87
JDE[15]	77.3	91.3	0.97	399	10 194	5 838	10.78
YOLOv5 + ReID	86.4	94.0	0.04	65	6 071	3 697	17.91
YOLOv5 + ReID + DNMS	90.3	94.9	0.04	55	3 598	3 378	17.11

注:↑表示数值越大越好;↓表示数值越小越好;加粗数值为本指标数值最好的。

3 结论

为将多目标跟踪模型应用在多车辆场景下,从而设计低推理时延高精度的跟踪模型。针对多目标跟踪推理时间长的问题,在 YOLOv5 网络结构上进行改进,设计了 ReID 模块,该模块将 YOLOv5 的 PANet 继续上采样,获得一个特征池,使改进后的 YOLOv5 模型在输出目标位置信息的同时输出特征信息。针对车辆间彼此覆盖的情况,为提高跟踪精度,提出一种基于动态 IOU 阈值的非极大值抑制算法,该算法根据每个边框的置信度得分,对每个边框设置不同的 IOU 抑制阈值,以此减少车辆密集场景下对车辆的漏检。实验结果表明,在 YOLO 网络中添加 ReID 模块能明显地减少目标跟踪的推理时间;使用基于动态 IOU 阈值的非极大值抑制能明显的增加目标跟踪精度。将 ReID 和基于动态 IOU 阈值的非极大值抑制用在 YOLOv5 模型中,与 FairMOT、JDE、DeepSORT 算法进行对比,改进后的模型有较好的跟踪精度和实时性。

参 考 文 献

[1] 张瑶, 卢焕章, 张路平, 等. 基于深度学习的视觉多目标跟踪算法综述[J]. 计算机工程与应用, 2021, 57(13): 55-66.
Zhang Yao, Lu Huanzhang, Zhang Luping, et al. Overview of visual multi-object tracking algorithms with deep learning[J]. Computer Engineering and Applications, 2021, 57(13): 55-66.

[2] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//2017 IEEE International Conference on Image Processing (ICIP). New York: IEEE, 2017: 3645-3649.

[3] 毛昭勇, 王亦晨, 王鑫, 等. 面向高速公路的车辆视频监控分析系统[J]. 西安电子科技大学学报, 2021, 48(5): 178-189.
Mao Zhaoyong, Wang Yichen, Wang Xin, et al. Vehicle video surveillance and analysis system for the expressway[J]. Journal of Xidian University, 2021, 48(5): 178-189.

[4] 武明虎, 黄咏曦, 王娟. 基于改进 YOLOv3 的街道行人检测与跟踪方法[J]. 科学技术与工程, 2021, 21(17): 7230-7236.

Wu Minghu, Huang Yongxi, Wang Juan. Pedestrian detection and tracking method on street based on improved YOLOv3[J]. Science Technology and Engineering, 2021, 21(17): 7230-7236.

[5] Zuraimi M A B, Zaman F H K. Vehicle detection and tracking using YOLO and DeepSORT[C]//2021 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). Washington, DC: IEEE, 2021: 23-29.

[6] Zhang Y, Wang C, Wang X, et al. A simple baseline for multi-object tracking[J]. arXiv Preprint, 2020: arXiv: 2004. 01888.

[7] Liang C, Zhang Z, Zhou X, et al. Rethinking the competition between detection and ReID in multiobject tracking[J]. IEEE Transactions on Image Processing, 2022, 31: 3182-3196.

[8] 赵桂平, 邓飞, 王昀, 等. 改进的 YOLOv5-ResNet 相似目标检测方法[J]. 科学技术与工程, 2022, 22(30): 13406-13416.
Zhao Guiping, Deng Fei, Wang Yun, et al. Improved YOLOv5-ResNet method for detecting similar objects[J]. Science Technology and Engineering, 2022, 22(30): 13406-13416.

[9] Wang H, Zhang S, Zhao S, et al. Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN + + [J]. Computers and Electronics in Agriculture, 2022, 192: 106512.

[10] Neupane B, Horanont T, Aryal J. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network[J]. Sensors, 2022, 22(10): 3813.

[11] 黄战华, 陈智林, 张晗笑, 等. 基于音视频信息融合的目标检测与跟踪算法[J]. 应用光学, 2021, 42(5): 867-876.
Huang Zhanhua, Chen Zhilin, Zhang Hanxiao, et al. Object detection and tracking algorithm based on audio-visual information fusion[J]. Journal of Applied Optics, 2021, 42(5): 867-876.

[12] 张文龙, 南新元. 基于改进 YOLOv5 的道路车辆跟踪算法[J]. 广西师范大学学报(自然科学版), 2022, 40(2): 49-57.
Zhang Wenlong, Nan Xinyuan. Road vehicle tracking algowrithm based on improved YOLOv5[J]. Journal of Cuangxi Normal University (Natural Science Edhtion), 2022, 40(2): 49-57.

[13] 张梦华, 陆奎, 高正康. 基于 YOLO 的视频行人检测研究[J]. 忻州师范学院学报, 2022, 38(5): 27-30.
Zhang Minghua, Lu Kui, Gao Zhengkang. Research on video pedestrian detection based on YOLO[J]. Journal of Xinzhou Teachers University, 2022, 38(5): 27-30.

[14] 肖雨晴, 杨慧敏. 目标检测算法在交通场景中应用综述[J]. 计算机工程与应用, 2021, 57(6): 30-41.
Xiao Yuqing, Yang Huimin. Research on application of object detection algorithm in traffic scene[J]. Computer Engineering and Applications, 2021, 57(6): 30-41.

[15] Wang Z, Zheng L, Liu Y, et al. Towards real-time multi-object tracking[C]//European Conference on Computer Vision. Cham: Springer, 2020: 107-122.

[16] Kendall A, Gal Y, Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2018: 7482-7491.

[17] Lü S, Chang M C, Du D, et al. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring [C]//15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). New York: IEEE, 2018: 1-6.

[18] Dendorfer P, Osep A, Milan A, et al. Motchallenge: a benchmark for single-camera multiple target tracking[J]. International Journal of Computer Vision, 2021, 129(4): 845-881.

[19] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: faster and better learning for bounding box regression[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Reston: AAAI, 2020: 12993-13000.

[20] Bodla N, Singh B, Chellappa R, et al. Improving object detection with one line of code[J]. arXiv Preprint, 2007: arXiv: 1704. 04503.