跨域环境下特定多目标跟踪算法的改进

穆晓芳[1]，李毫[2]，刘嘉骥[3]，刘振宇[4]，李越[5]

（1. 太原师范学院 计算机科学与技术学院，晋中 030619；2. 太原师范学院 计算机科学与技术学院，晋中 030619；3. 太原师范学院 计算机科学与技术学院，晋中 030619；4. 太原师范学院 计算机科学与技术学院，晋中 030619；5. 天津工业大学 电子信息学院，天津 300387；）

摘　要：监控视频跨域环境下的多目标跟踪，在智慧安防中是十分重要且具有挑战性的任务。该任务的难度在于视频画面中目标之间的频繁遮挡、轨迹开始终止时刻未知、目标太小、目标间交互、表观相似以及摄像头视角变化等问题。本算法改进针对频繁遮挡、表观相似问题，提出一种改进的多目标跟踪算法，最大化利用低分检测对象，将未匹配的低分对象进行二次匹配，目标跨域后，依据摄像头拓扑排序规则，以及相邻摄像头的未匹配跟踪轨迹，同时对检测器 YOLOv5 算法进行优化改进，通过信息流的层层递进，有效解决多尺度问题和小目标信息提取不充分等问题，在相邻的摄像头中快速匹配到跟踪对象，以提高跨域环境下特定多目标跟踪的精度。在进行对比消融试验，本改进算法 MOTA 达到了 62.8%，IDswitch 也显著降低。

关键词：多目标跟踪；YOLO；计算机视觉；深度学习

中图分类号：TP181　　**文献标识码**：A

Improvement of Specific Multi-target Tracking Algorithm in Cross-domain Environment

MU Xiaofang[1]，LI Hao[2]，LIU Jiaji[3]，LIU Zhenyu[4]，LI Yue[5]

(1. Department of Computer Science and Technology, 2. Department of Computer Science and Technology, 3. Department of Computer Science and Technology, 4. Department of Computer Science and Technology, Jinzhong 030619, China 5. School of Electronic Information Tianjin Polytechnic University, Tianjin 300387, China)

Abstract：Multi-target tracking in the cross-domain environment of surveillance video is a very important and challenging task in intelligent security. The difficulty of this task lies in the frequent occlusion between the objects in the video frame, the unknown start and end time of the trajectory, the target is too small, the interaction between the objects, the apparent similarity and the camera Angle change. This algorithm is improved in view of the frequent occlusions and apparent similar problems, put forward an improved multiple target tracking algorithm, maximum use low detection object, object will not match the low secondary matching, target cross-domain, topological sort rules, according to the camera and adjacent camera not matching tracking trajectory, optimize the detector YOLOv5 algorithm is improved at the same time, In order to improve the accuracy of multi-target tracking in cross-domain environment, the tracking object can be quickly matched from adjacent cameras. In the comparative ablation test, the MOTA of the improved algorithm reached 62.8%, and the IDswitch was also significantly reduced.

Keywords：multi-target tracking；YOLO；computer vision；deep learning

随着科学技术的发展，人们的生活中充满着各种便捷，伴随而生的安全问题也引起了人们的着重关注，在新时代，人们对安全保障也有了新的要求。目前越来越多的场所也利用视频摄像头用于监控目的，国家发展改革委提出 2020 年实现公共安全视频监控

收稿日期：2022-10-28.

基金项目：山西省重点研发计划(202102010101008)；山西省基础研究计划(自由探索)（20210302123334）.

作者简介：穆晓芳（1974— ），女，教授，主要从事计算机视觉的研究，(E-mail) mu_xiao_fang@163.com.

通讯作者：穆晓芳，女，教授，（E-mail）mu_xiao_fang@163.com.

的全域覆盖和全网共享。到 2020 年，重点公共区域视频监控覆盖率达到 100%。目前视觉监控系统主要有单摄像头系统和多摄像头系统。但是单摄像头系统所能够监控的区域是十分有限的，所以选择多摄像头系统来扩大监控场景的范围，在多摄像头的安防监控系统当中，首先是要将各个摄像头下的目标能够检测出来并能实现有效跟踪，其次是目标在不同的多个摄像头出现时，要将属于同一目标的相对应起来。目标检测是实现跨域环境下特定多目标跟踪的关键更是基础，还是在计算机视觉领域中如行为分析、异常检测[1]、目标检测[2]等众多问题的基础所在。

目前已有的多目标跟踪方法有很多，主要可以分为两大类：第一类是 DFT（Detection Free Tracking）[3]不需要检测的跟踪，需要人工来标注第一帧图像中的目标，之后跟踪在第一帧中给定的初始检测框，这一算法有很明显的缺点如果有些新目标在后续帧中出现，这些新出现的目标并不会被跟踪。第二类是 TBD（Tracking By Detection）基于检测的跟踪，即在跟踪之前每一幅图像中的目标都事先经过检测算法得到。它首先检测目标，然后将检测到的目标链接到已有的轨迹中。这一类算法的跟踪数量，跟踪目标的类型全部由检测算法的结果来决定，即检测算法的好坏决定了跟踪器的效果。

近年来，国内外学者提出了许多关于目标跟踪的算法。如 2015 年 Boser[4]提出了一种在线的多目标检测与跟踪算法，这一算法使用卷积神经网络（CNN）在每一个检测器当中进行数据关联，通过深度学习技术将数据关联问题等效为 CNN 中的推理，但其速度过于缓慢且帧率也不是太高。2016 年 Bewley[5]等提出了 SORT 算法，但这一算法只使用 Kalman 滤波器和匈牙利算法等基本组合构建跟踪器，但是该算法目标的标签交换次数过多，2017 年，在 SORT 算法的基础上又提出了 DeepSORT[6]算法，该算法在 SORT 算法的基础上整合了外观信息，并且在进行轨迹关联时使用了视觉外观空间中的最近邻数据关联算法，使用马氏距离和和余弦距离计算外观信息和运动信息，降低了目标标签交换的次数。但对于跟踪精度依旧偏低。2020 年清华大学提出 Joint Detection and Embedding (JDE)[7]基于 anchor-based 的目标检测，将目标检测和嵌入学习融合在同一个网络中，适合低维度的特征，速度较快，准确率有所下降。2020 年，华中科技大学提出了 FairMot[8]，使用了 Anchor-Free 目标检测范式来进行目标跟踪，DLA(Deep Layer Aggregation)的网络进行特征提取，跟踪的精度提高了但速度有所下降。2021 年字节跳动提出了 ByteTrack[9]算法，优化了低分检测框的使用，跟踪精度有所提高，但推理速度有所下降。

1 YOLOv5 算法

1.1 YOLOv5 算法原理

到目前 YOLO 系列检测算法已经更新到了 YOLOv7[10]，但 YOLOv5 在其输入端采用 Mossaic 数据增强、自适应锚框计算，在 Backbone 部分引入了 Focus 结构和 CSP 结构，Neck 部分采用 FPN+PAN 结构。YOLOv5 的这些改进使得其对小目标检测效果显著提升，并且既保证了推理速度和准确率，又减小了模型尺寸。因此本文选用 YOLOv5 作为目标检测算法。

YOLOv5s[11]的网络结构如图 1 所示，该网络由骨干网络 Backbone、颈部 Neck、输出 Output 组成。

1.2 改进 YOLOv5 算法

YOLOv5 的 Backbone 部分的第一层为 focus 结构，如图 1.2 所示，输入图像为 640*640*3，对图像在进入 Backbone 之前先进行切片计算，再通过一个 32 通道，大小为 1*1 的卷积计算，得到 320*320*32 大小的特征图，这样使得 W、H 的信息集中到通道上，使得特征提取的更加充分，实现降维，但对精度的提升并不大，对此本文采用 DenseNet[12]的核心组件 DenseBlock 模块来替换 Focus 模块，该卷积神经网络具有紧密连接性，DenseBlock 模块的具体连接方式如公式 3.3 所示：

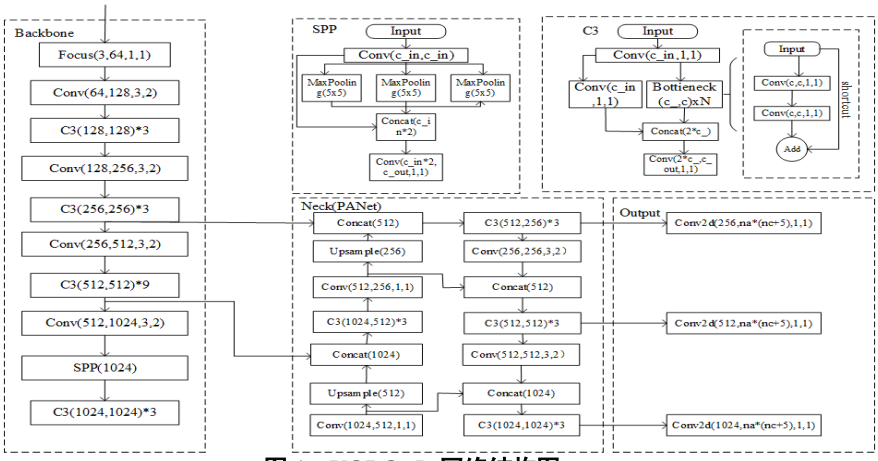

图 1　YOLOv5s 网络结构图

Fig. 1　Network structure diagram of YOLOv5s

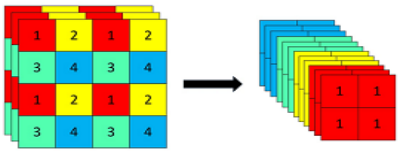

图 2　focus 结构图

Fig. 2　Structure diagram of focus

$$X_k = H_k(X_{k-1}) \quad (1)$$

$$X_k = H_k(X_{k-1}) + X_k \quad (2)$$

$$X_k = H_k([X_0, X_1, X_2, \ldots, X_{k-1}]) \quad (3)$$

公式 1 是传统的网络第 k 层的输出，即第 k 层的输出为 k-1 层的输出做非线性变换的结果。

公式 2 是 ResNet 的,其中 k 表示层，Xk 表示第 k 层的输出，H_k 表示一个非线性变换函数，即在 ResNet 当中，第 k 层的输出为第 k-1 层的输出再加上 k-1 层的非线性变换。

公式 3 是 DenseNet 的核心 DenseBlock，$[X_0,X_1,X_2,\ldots,X_{k-1}]$表示将第 0 到 k-1 层的输出 feature map 做 concatenation。其中 concatenation 是做通道的合并。其中 k 表示层，X_k 表示第 k 层的输出，H_k 表示一个非线性变换。

由三个公式对比可以看出 ResNet[13]网络只是做了单纯的值相加，并没有改变每一层通道数，而 DenseBlock 则是将 0~(k-1)层输出的特征图进行合并拼接后，在进行了非线性变换，通过这样的操作便可以保证网络中层与层之间最大程度的信息传输前提下，直接将每一层都链接了起来。

在 denseblock 中每个卷积层输出的 feature map 的数量 k 都很小(小于 100，其他的网络会有几百上千的宽度)，denseblock 的结构如图 3 所示，它含有四个模块且每个模块均为 4 层，将前面多层网络的输出作为自身的输入，每一层都会经过非线性变换函数，会经过归一化(Batch Normalization)、激活函数(Activation Function)、卷积(Convolution)等操作后传输至后面的网络。在 Denseblock 模块中将输入的任意两层的 feature map 直接相连，并且可作为独立的输入，上一层的 feature map 会传递给下一层，这样就减轻了梯度消失(vanishing-gradient)的问题，就可以用来替换 focus 操作进行对输入图像的降维，实现更加完整的特征提取。

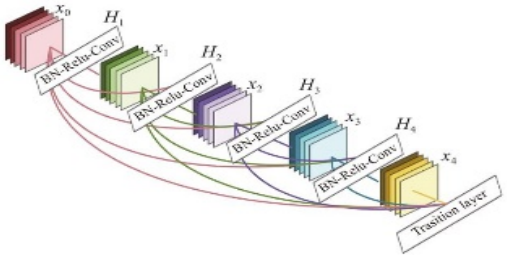

图 3　DenseBlock 结构图

Fig. 3　Structure diagram of DenseBlock

采用 denseblock 模块替换 focus 结构后的骨干网络 backbone 如图 4 所示，在替换后检测平均精确率 mAP 比原始模型提高了 2.3%。mAP 对比如图 5 所示。

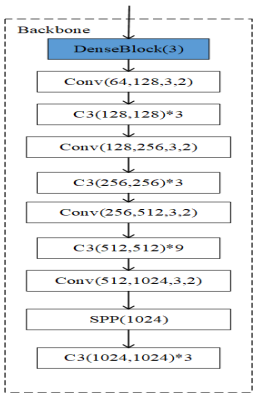

图 4　替换后的 backbone

Fig. 4　Replaced backbone network

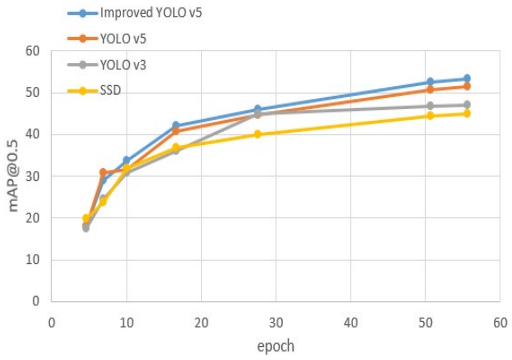

图 5　mAP 对比图

Fig. 5　Comparison of mAP

2　DeepSORT 算法

2.1　DeepSORT 算法原理

DeepSORT 算法是在 SORT 算法的基础上改进而来的，SORT 算法基本原理为在跟踪开始之前，首先对所有的目标进行检测，完成特征建模，在第一帧视频开始时，对检测到的目标完成初始化并标注 ID。然后后面的帧进来，先到卡尔曼滤波器[14]中得到由前面帧 box 产生的预测状态和协方差预测，再对跟踪器的所有目标预测状态和本帧检测到的目标计算 IOU，通过匈牙利匹配算法得到最大唯一匹配，去掉匹配值小于 IOU_Threshold 的匹配对。最后用当前帧匹配到的目标检测的 box 去更新 kalman 跟踪器，计算卡尔曼增益、状态更新、协方差更新，将更新后的状态值输出，作为本帧跟踪的 box 框，对于当前帧中未匹配到的目标重新初始化跟踪器。以往的一些算法先使用匈牙利匹配算法对相邻帧间的目标进行匹配生成大量的跟踪片段(tracklets)，之后使用这些片段做二次匹配，来解决遮挡引发的轨迹中断，不能达到一个实时的效果。SORT 算法将这种两阶段的匹配算法改进为一阶段的方法达到了在线跟踪。

DeepSORT 算法在整体框架上没有大改，在 SORT 算法的基础上增加了 Deep Association Metric，加入了目标的外观信息来实现当发生较长时间遮挡的目标跟踪，在 kalman 滤波预测结果的基础上，使用匈牙利匹配算法来进行目标匹配，在这一过程加入了运动信息和外观信息，对于运动信息采用运动匹配度，采用 detection 和 track 在 kalman 滤波器预测的位置之间的马氏距离来进行运动匹配程度的描述。

2.2　DeepSORT 算法改进

低分检测框包含许多损害跟踪性能的背景，许多被遮挡的物体可以被正确的检测到，但分数低，为了减少丢失的检测并保持轨迹的持久性，保留所有检测框并在每个框之间进行关联。

针对 DeepSORT 级联匹配中两个跟踪器竞争同一个检测结果的匹配权，以及对二次匹配时，判定跟踪框和检测框如何相交的问题，本文采用 GIOU 来替换 IOU,在进行跟踪预测之前先经过 IOU 匹配来排除那些不可能匹配的目标，在匹配过程中充分利用从高分到低分的检测框，对于低分的检测框，不会直接地丢掉，如果目标因为遮挡而导致检测分数低并且又被舍弃，就会带来缺失和碎片化的轨迹，对低分的检测框，利用它们与轨迹的相似性，来恢复真实对象并过滤掉背景检测，当视频帧序列发生遮挡或运动模糊时，利用前一帧的信息来增强视频检测性能，保留每一个检测框，分为高分低分。首先将高分框进行轨迹关联，然后将低分框和剩余下来不匹配的轨迹利用 GIOU 匹配关联起来，同时恢复低分检测框中的对象。

为降低严重遮挡情况下，发生目标转换的情况，做出的具体改进算法的流程图如图 6 所示。输入是一个视频序列 V，以及一个目标检测器 Det 和卡尔曼滤波器 KF。阈值

T_{high}、T_{low}，T_{high}、T_{low} 是检测分数阈值（设置为 0.8,0.2），输出为视频的轨迹 T，每个轨迹包含每一帧中对象的边界框和身份。对于视频中的每一帧，用检测器 Det 预测检测框和分数，根据检测分数阈值将所有检测框分成两部分 D_{high} 和 D_{low}。在将低分检测框和高分检测框分离后，使用卡尔曼滤波器 KF 来预测 T 中每个轨道的新位置。在高分检测框 D_{high} 和所有轨迹 T 之间进行第一次关联。相似度由检测框 D_{high} 和轨迹 T 的预测框之间的 IoU 计算。然后，使用匈牙利算法[15]完成基于相似度的匹配。如果检测框和轨迹框之间的 GIoU 小于 0.2，匹配失败。将未匹配的检测框保留在 D_{remain} 中，将未匹配的轨迹保留在 T_{remain} 中，经过低分检测框 D_{low} 与第一次关联后的剩余轨迹 T_{remain} 之间进行第二次关联。将未匹配的轨迹保留在 $T_{re\text{-}remain}$ 中，检测对象可能离开了当前摄像头区域，最后在第一次关联后从未匹配的高分检测框 D_{remain} 中初始化新轨迹。对于 D_{remain} 中每个检测框，如果它的检测分数高于阈值并连续存在两帧，初始化一个新的轨迹。

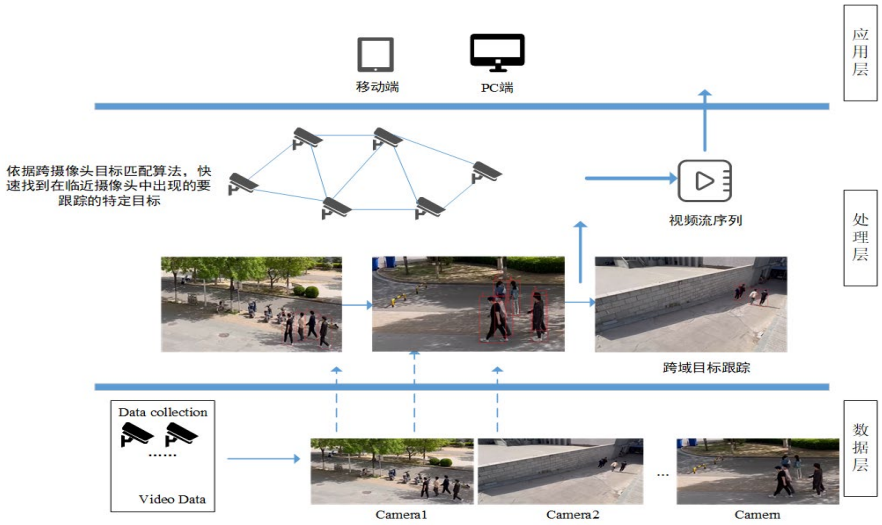

图 6　改进后的算法流程图

Fig. 6　Flowchart of the improved algorithm

当目标离开当前区域后，依据摄像头的排序规则，在相邻摄像头下对在当前摄像头 D_{remain} 中的检测框与相邻摄像头下 $T_{re\text{-}remain}$ 的跟踪轨迹进行匹配，并结合其表观信息一同进行度量，实现跨域目标匹配，进而达到了跨域特定多目标跟踪的目的。

3　实验结果与分析

3.1　实验平台

实验使用 kaggle-Safe Helmet Detection 安全帽识别数据集都对改进的检测器 YOLOv5 进行精度和速度评估，使用自制数据集对多目标跟踪算法进行测试，实验平台配置为：操作系统：Windows10；CPU:AMD Ryzen 7 3700X @3.59GHz 八核处理器；GPU:NVIDIA GeForce RTX 3080 11G 显存；深度学习框架：Pytorch；编程语言：python。

3.2　数据集描述

kaggle-Safe Helmet Detection 安全帽识别数据集含有 5000 张图片，训练和测试图像尺寸均为 416*416，数据集标注为三个类别：helmet、head、person。这一数据集涵盖各个佩戴安全帽的应用场景，包含各种施工现场、高空作业、矿业开采、设备检修等多个应用场景。自制数据集以实验室场景为基础，使用两个摄像头，每段视频以 25 帧/秒的速度进行录制，分辨率为 960*540 像素。

多摄像头多人跟踪(MMPTRACK)数据集包含大约 5 小时的训练视频和 1.5 小时的验证视频。该数据集使用人员边界框和相应

的人员 ID 进行了完全注释。所有视频都是使用在现场以不同角度放置的摄像机录制的，并保证所有摄像机的视野都是连接的(一台摄像机与至少一台其他摄像机的 FoV 重叠)。视野和校准良好。

3.3 执行细节

训练数据的过程中，在 VOC 预训练权重的基础上对 YOLOv5 算法进行超参数设置，损失函数采用随机梯度下降法初始学习率为 0.01，终止学习率为 0.2，学习率调整轮数设定为 5，一次传入图片数设定为 16，训练轮数设定为 500，训练数据如图 7 所示。

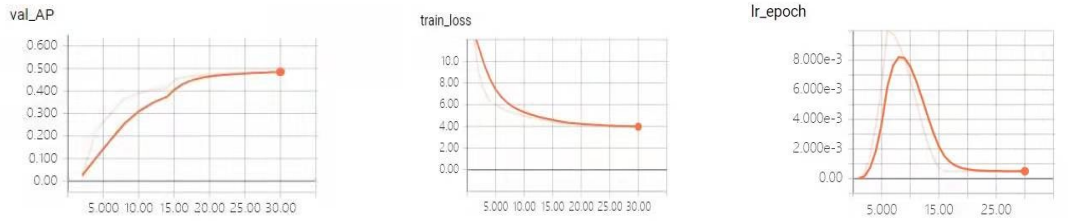

图 7 训练数据

Fig. 7 Training data

3.4 算法性能测试

3.4.1 评价指标

在多目标跟踪当中，使用跟踪准确度 MOTA(Multi-Object Tracking Accuracy)和跟踪精确度 MOTP(Multi-Object Tracking Precision)这两个评价指标，此外还有 Recall、Precision、IDS。一般来说 MOTA 值越高，说明模型检测精度越高，检测效果越好。

MOTA、MOTP 计算公式如下：

$$\mathrm{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t GT_t}$$

$$\mathrm{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$$

上式中，FNt 代表第 t 帧中漏检个数，FPt 代表第 t 帧中的虚检个数，IDS 代表第 t 帧当中轨迹的 id 号发生转变的个数。C_t 代表第 t 帧成功与 GT 匹配的检测框数目，$d_{t,i}$ 代表匹配对之间的距离度量。

3.4.2 实验结果与分析

为验证改进后跨域多目标跟踪算法的性能，首先将改进的 YOLOv5 算法与原始的 YOLOv5 算法和改进前后的多目标跟踪算法随机组合后，进行消融实验，跨域多目标跟踪结果如图 8 所示，在 T-n 帧、T 帧、T+n 帧，都能对视频中的目标进行有效的跟踪。

3.4.3 消融实验对比

对改进后的算法进行消融实验对比，消融实验对比如表 1 所示。在未改进的 DeepSORT 算法进行试验其 IDS 达到了 251，使用 YOLOV5+DeepSORT 进行多目标跟踪后，IDS 显著降低，并且 MOTA 也有所提高，当时用改进后的 YOLOv5 和 DeepSORT 进行结合时，跟踪的准确度 MOTA 达到了 61.6%，当使用 YOLOv5 和改进后的 DeepSORT 算法结合时，虽然 FPS 有所降低，但跟踪的精度 MOTP 达到了 72.9%，最后使用改进后的 YOLOv5 算法和改进后的 DeepSORT 的算法相结合，虽然 FPS 相较于其他有所下降，但 IDS 显著降低，达到了 159，并且跟踪准确度显著提高，达到了 62.8%，效果最优。

表 1 消融实验对比

Table1 Comparison of ablation experiments

Algorithms	MOTA	MOTP	IDS	FPS
DeepSORT	59.8	78.8	251	23.8
YOLOV5+DeepSORT	60.2	79.2	229	26.3
Improved YOLOv5+DeepSORT	61.6	78.9	235	26.5
YOLOv5+ Improved DeepSORT	60.5	79.2	183	23.6
Improved YOLOv5+ Improved DeepSORT	62.8	79.2	159	24.5

图 8　跨域多目标跟踪结果

Fig. 8　Cross-domain multi-target tracking results

3.4.4　性能对比

本文设计了两组实验，对比分析本文算法的性能。选用 MOT-17 作为基准数据集。

实验 1 本文改进算法在不同视频序列上进行多目标跟踪测试，分析在不同背景下的实验结果。

本文算法在MOT-17训练集的视频序列上进行对比实验，实验结果如表 1 所示，从实验数据可以看出本文改进算法在 MOT-17-04-DPM上实验效果最优，在MOT-17-02-DPM 上性能较差，这是由于 MOT17-02-DPM 视频序列当中行人目标太小，背景较为昏暗并且相机晃动导致跟踪效果差；在 MOT17-04-DPM 视频序列当中跟踪目标明显，视频分辨率较高使得跟踪效果最优；在其他视频跟踪序列当中由于图像分辨率低、相机晃动、目标太小等问题导致跟踪效果一般。

表 2　本算法在 MOT-17 训练集不同序列上的量化跟踪结果

Table2 The quantization tracking results of this algorithm on different sequences of MOT-17 training set

视频序列	IDF1/%	IDP/%	MT	ML	MOTA/%	MOTP%	IDS
MOT17-02-DPM	66.4	78.5	24	13	47.3	71	176
MOT17-04-DPM	83.4	94.3	77	2	63.9	73.5	34
MOT17-05-DPM	81.0	91.1	48	18	55.0	72.8	60
MOT17-09-DPM	76.8	83.3	17	1	56.5	75.6	37
MOT17-10-DPM	69.2	77.5	28	3	61.3	68.1	134
MOT17-11-DPM	89.9	94.0	47	9	63.7	74.3	37
MOT17-13-DPM	81.5	86.5	78	9	59.6	68.3	76

实验 2 在 MOT-17 训练集上，本文算法与其他跟踪算法进行对比测试，对比算法为 OC-SORT[16]、StrongSORT[17]和 DeepSORT[6]。实验结果如表 3 所示，由表 3 可见：本文算法的跟踪准确度为 65.8%，比第二位的 StrongSORT 算法（准确度为 62.6%）高出 3.2%，本文算法的 IDF1 为 78.5%，且本文算法在MOT-17数据集中的跟踪目标切换次数为 120，是对比算法中最低的，可以看出本算法有效解决了多目标跟踪中目标遮挡以及 ID 切换的问题，证明了本算法具有良好的跟踪准确度。

实验 3 在 MOT-20 训练集上，本文算法与 OC-SORT、StrongSORT 和 DeepSORT 实验结果如表 4 所示，由表 4 可见：本文算法的跟踪精度为 73.2%，比第二位的 StrongSORT 算法（跟踪精度为 70.9%）高出 2.3%，本文算法在 MOT-20 数据集中目标切换次数为 136，是对比四种算法中最低的。

经过在数据集 MOT-17、MOT-20 数据集中的对比，可以看出本算法具有较好的性能，在不同的数据集中都能有较好的表现。

表 3　不同算法在 MOT-17 训练集上的量化跟踪结果对比

Table3 Comparison of quantized tracking results of different algorithms on MOT-17 training set

算法	IDF1/%	IDP/%	MT	ML	MOTA/%	MOTP	IDS
本文	78.5	73.9	54	16	65.8	78.5	120
OC-SORT	65.1	71.2	41	15	58.2	67.6	141
StrongSORT	69.3	76.9	56	13	62.6	71.2	139
DeepSORT	70.6	76.2	32	18	61.4	79.1	206

表 4　不同算法在 MOT-20 训练集上的量化跟踪结果对比

Table4 Comparison of quantized tracking results of different algorithms on MOT-20 training set

算法	IDF1/%	IDP/%	MT	ML	MOTA/%	MOTP	IDS
本文	75.2	70.8	48	16	60.5	73.2	136
OC-SORT	66.8	69.6	46	12	59.7	65.3	156
StrongSORT	72.6	76.8	49	19	63.8	70.9	142
DeepSORT	66.2	67.1	30	22	52.3	62.1	218

为直观展示本文改进算法在处理目标较小时和目标发生遮挡时的优势，在视频序列 MOT-17-13-DPM 上进行不同算法处理效果的对比分析，实验结果如图 9 所示，在图中第一行为视频序列 MOT-17-13-DPM 中的初始帧，本算法检测出两个小目标，其余算法未检出；第二行为视频序列的第 18 帧，本算法检测出两个小目标，其余算法未检出；第三行为视频序列的第 36 帧，本算法检测出一个小目标，其余算法未检出；第四行为视频序列的第 44 帧，本算法检测出一个小目标，其余算法未检出并还存在误检漏检现象。

在图 10 中，显示了不同算法在目标被遮挡时的跟踪效果，在 MOT-17-09-DPM 视频序列上进行实验，第一行为 397 帧目标被遮挡前，第二行为 415 帧目标正在被遮挡，第三行为427帧遮挡结束，从图中可以看出，本文改进算法可以在目标发生遮挡后准确跟踪到原有目标且不发生 IDSwitch，OC-SORT、StrongSORT 和 DeepSORT 三个算法都框选到了目标，但都存在 IDSwitch 的问题。

综上本算法首先通过改进检测器的骨干网络，有效提高了对目标的检测性能（一个好的检测器对跟踪结果至关重要），在输入跟踪网络之前，排除那些不可能的符合关联的目标，其次通过把得分框按照一定阈值划分为高分框和低分框。对于高分框来说按照正常的方法送入跟踪器，并使用 IOU 计算代价矩阵进行预处理，排除不可能匹配的框，再对目标进行级联匹配，然后利用匈牙利算法进行分配。对于低分框，则利用未匹配上的框（未匹配上就说明上一帧是匹配上的）和低分的框进行 GIoU 匹配，然后同样利用匈牙利算法进行分配，有效解决了目标突然加速移动时跟踪失败的情况以及目标遮挡、目标较小时跟踪目标切换的情况。

其时间复杂度满足卷积神经网络整体的时间复杂度

$$\text{Time}\sim O\left(\sum_{l=1}^{2} M_l^2 * K_l^2 * C_{l-1} * C_l\right)$$

其中 D 表示网络所具有的的卷积层数，即为网络的深度，l 表示神经网络第 l 个卷积层，C_l 表示神经网络第 l 个卷积层的输出通道数。

实验结果表明，本文算法的跟踪准确率比其他对比算法都高，且在目标发生遮挡时有较好的跟踪结果，显著减少了目标标签交换的次数，对小目标的跟踪也有一定的效果。在处理实时视频时基本可以达到实时处理的效果。

图 9　不同算法在目标较小时的跟踪效果

Fig. 9　Tracking effect of different algorithms when targets are too small

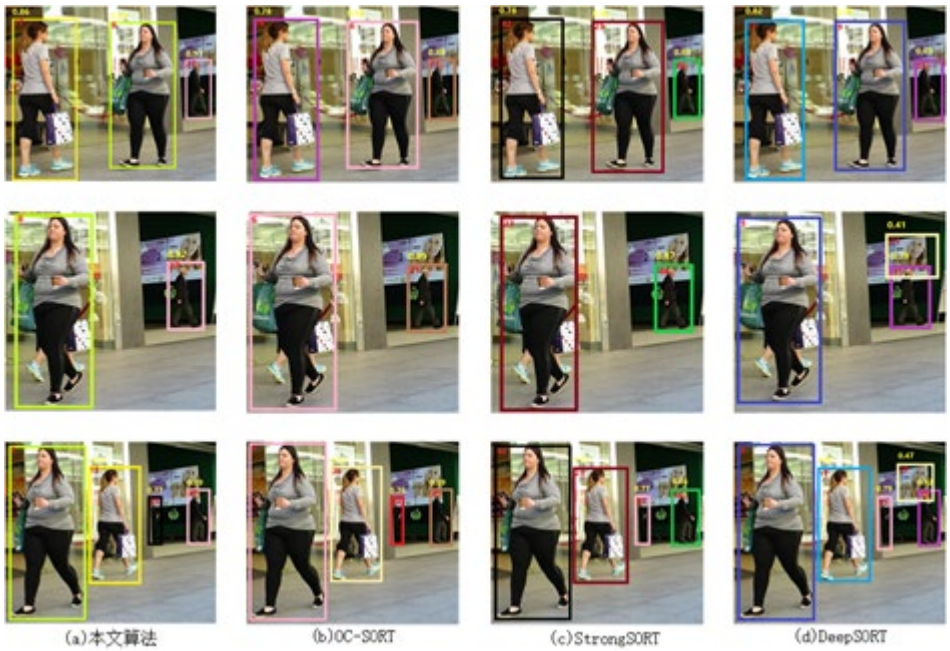

图 10　不同算法在目标被遮挡时的跟踪效果

Fig. 10　Tracking effect of different algorithms when the target is occluded

4　结论

本文通过对检测器 YOLOv5 主干网络以及跟踪器 DeepSORT 对低分框的利用部

分进行改进，在一定程度上能改善单摄像头下多目标跟踪目标频繁变换、识别精度低的问题。通过对检测器 YOLOv5 主干网络头部的替换，采用 DenseNet 的核心 DenseBlock 结构快，减轻了梯度消失的问题，以及低分框的二次匹配，大大降低了跟踪目标频繁切换的频率。改进后的 YOLOv5 和改进的 DeepSORT 算法结合，使得跨域特定多目标跟踪的检测精度有不同程度的提高。但是，本文算法仍存在一些缺陷，在复杂场景下仍会出现跟踪失败的情况。在之后的工作中，会针对发现的问题继续改进本算法，进一步提高跨域特定多目标跟踪的性能。

参考文献：

[1] 王杰 张雪英 李凤莲 杜海文 于丽君 马秀.改进 DM-SVDD 算法的异常检测研究及应用[J].太原理工大学学报,2021,52(05):764-768.

[2] 韩强 张喆 续欣莹 谢新林.基于 FF R-CNN 钢材表面缺陷检测算法[J].太原理工大学学报,2021,52(05):754-763.

[3] Lin L, Lu Y, Li C, et al. Detection-free multiobject tracking by reconfigurable inference with bundle representations[J]. IEEE transactions on cybernetics, 2015, 46(11): 2447-2458.

[4] Boser B E, Guyon I M, Vapnik V N. A training algorithm for optimal margin classifiers[C]//Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. 144-152.

[5] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]//2016 IEEE international conference on image processing (ICIP). IEEE, 2016: 3464-3468.

[6] 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). IEEE, 2017: 3645-3649.

[7] Wan J, Deng J, Qiu X, et al. Body-Face Joint Detection via Embedding and Head Hook[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 2959-2968.

[8] Zhang Y, Wang C, Wang X, et al. Fairmot: On the fairness of detection and re-identification in multiple object tracking[J]. International Journal of Computer Vision, 2021, 129(11): 3069-3087.

[9] Zhang Y, Sun P, Jiang Y, et al. Bytetrack: Multi-object tracking by associating every detection box[J]. arXiv preprint arXiv:2110.06864, 2021.

[10] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[J]. arXiv preprint arXiv:2207.02696, 2022.

[11] Wu W, Liu H, Li L, et al. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image[J]. PloS one, 2021, 16(10): e0259283.

[12] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[13] Targ S, Almeida D, Lyman K. Resnet in resnet: Generalizing residual architectures[J]. arXiv preprint arXiv:1603.08029, 2016.

[14] 石龙伟, 邓欣, 王进, 等. 基于光流法和卡尔曼滤波的多目标跟踪[J]. 计算机应用, 2017, 37(A01): 131-136.

[15] Shopov V K, Markova V D. Application of Hungarian Algorithm for Assignment Problem[C]//2021 International Conference on Information Technologies (InfoTech). IEEE, 2021: 1-4.

[16] Cao J, Weng X, Khirodkar R, et al. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking[J]. arXiv preprint arXiv:2203.14360, 2022.

[17] Du Y, Song Y, Yang B, et al. Strongsort: Make deepsort great again[J]. arXiv preprint arXiv:2202.13514, 2022.