

引用格式:李晶,黄山. 基于YOLOv3目标跟踪方法[J]. 电光与控制, 2019, 26(10):87-93. LI J, HUANG S. YOLOv3 based object tracking method [J]. Electronics Optics & Control, 2019, 26(10):87-93.

基于YOLOv3目标跟踪方法

李晶^a, 黄山^b

(四川大学, a. 电气信息学院; b. 计算机学院, 成都 610065)

摘要:提出了一种基于深度学习检测算法YOLOv3的目标跟踪算法。该算法利用深度学习模型在目标特征提取上的优势,采用基于回归的YOLOv3检测模型提取候选目标,同时结合目标颜色直方特征和局部二值模式直方特征进行目标筛选,实现对目标的跟踪。为了提高算法的性能,还提出了一种K邻域搜索方法,可以针对选定目标进行邻域检测。实验结果表明,提出的目标跟踪算法跟踪效果很好,综合表现比4种对比算法提高了80%左右,同时在目标物体光照变化、姿态变化、尺寸变化、旋转变化等复杂情况下有很好的鲁棒性。

关键词:深度学习检测算法; 目标跟踪; K邻域搜索; 鲁棒性

中图分类号: O213.2 **文献标志码:** A **doi:**10.3969/j.issn.1671-637X.2019.10.018

YOLOv3 Based Object Tracking Method

LI Jing^a, HUANG Shan^b

(Sichuan University, a. Electrical Information College; b. Computer College, Chengdu 610065, China)

Abstract: An object tracking algorithm is proposed based on the deep learning detection algorithm of YOLOv3 (YOLOv3: An Incremental Improvement), which utilizes the advantages of deep learning model in target feature extraction, and extracts candidate targets by using regression-based YOLOv3 detection model. The target color histogram feature and Local Binary Pattern (LBP) feature are also used for target screening, thus to implement object tracking. At the same time, a method called K-neighbor searching is presented to improve algorithm performance, which performs neighborhood detection for the selected targets. Experimental results show that the proposed algorithm has a good tracking performance, with an overall performance improved by about 80% in comparison with the four contrast algorithms, and has good robustness in the complex situations of illumination changing, posture changing, size changing and rotation of target object.

Key words: YOLOv3; object tracking; K-neighbor searching; robustness

0 引言

近年来,目标跟踪已经成为计算机视觉研究领域重要的研究热点之一。从经典的MeanShift算法、Kalman Filter等跟踪算法到以KCF为代表的相关滤波跟踪算法,再到最近几年出现的深度学习相关的跟踪算法,目标跟踪研究已取得了很大的进步与发展。一般而言,目标跟踪算法可以分为生成类目标跟踪算法和判别类目标跟踪算法两类^[1]。生成类目标跟踪算法的基本思路是对当前帧的目标区域建模,在后续的帧中找出最可能的目标区域作为跟踪目标;这类跟踪算法的

优势在于速度很快,如基于相关滤波的KCF算法,得益于快速傅里叶变换(FFT)和矩阵计算,其目标跟踪速度可达172帧/s^[2]。判别类目标跟踪算法的基本思路是首先进行选定区域的特征提取,然后采用机器学习算法训练一个分类器将目标与背景区分开来;这类跟踪算法的优势在于适应性强,即使在复杂环境下也能保持一定精度。如多实例学习(Multiple Instance Learning, MIL)算法^[3],其在目标部分遮挡的情况下也能正常工作。

基于深度学习的目标跟踪算法的发展并非十分顺利,这是由于深度学习模型的构建需要大量标注训练数据,才能有效地提高模型的特征表达能力^[4]。目标跟踪往往只能提供第一帧的选定框作为训练数据,这样的训练数据显然是不够的。

针对这个问题,目前的解决方案中大部分算法采

收稿日期:2018-11-26

修回日期:2018-12-09

作者简介:李晶(1993—),男,安徽安庆人,硕士生,研究方向为计算机视觉。

用离线训练结合在线更新的模式,但这需要大量的样本进行模型调整,导致其跟踪速度不理想。本文提供了另外一种解决思路:直接使用 ImageNet 这样的大规模分类数据库上训练出的 CNN 网络获取目标特征表示,再利用观测模型根据目标特征进行分类并获得跟踪结果。

根据这种思路,本文提出了一种基于深度学习检测算法 YOLOv3 的目标跟踪算法,称之为 TOD (Tracking Object based on Detector of YOLOv3)。借助于深度学习检测算法进行目标特征提取,得到同一类别物体的候选框,结合目标全局性的颜色直方图特征和局部性的 LBP 特征,在目标邻域进行搜索,进行目标筛选,达到跟踪的目的。该方法利用了深度学习检测算法在目标特征表达上的优势,将其迁移到目标跟踪上,具有很好的目标跟踪性能。

1 基本概念

深度学习模型在物体检测领域表现十分优秀,检测性能明显优于传统物体检测算法。基于深度学习的目标检测算法一般可以分为两类:基于区域的物体检测和基于回归的物体检测^[5]。前者在检测精度上有更好的表现,但检测速度很慢,基本达不到实时性要求;而后者在检测速度上优势十分明显,且检测精度只比前者稍低一点。考虑到目标跟踪算法对速度要求很高,本文选用基于回归的物体检测算法 YOLOv3 实现对物体的检测,从而得到跟踪目标。

1.1 YOLOv3 介绍

YOLOv3 算法是物体检测算法 YOLO 系列第 3 个版本,可以检测出 person, bicycle 和 car 等 80 个类别物体。借助于 GPU TitanX, YOLOv3 在 COCO 数据上 51 ms 内实现了 57.9 的 AP50^[6],其检测性能表现十分优异。

YOLOv3 采用目标回归的思想,减少了网络的计算量,提升了运行速度;借助于特征金字塔网络 (Feature Pyramid Network, FPN) 的思想,在 3 种不同的尺度上进行特征提取,使算法在小目标的检测上有了更好的表现;采用独立的逻辑回归器取代 Softmax,能够实现使用多标签分类,使检测物体的分类更加准确。YOLOv3 采用的是全卷积神经网络来提取图像特征,该网络由一系列表现良好的 3×3 和 1×1 的卷积层构成,共有 53 个,称之为 Darknet-53^[7]。同时也增加了 Res 层,将深度神经网络的逐层训练改为逐阶段训练,这样可以解决网络的梯度弥散或者梯度爆炸的问题,达到更好的训练效果。Darknet-53 网络结构如图 1 所示。

	Type	Filters	Size	Output
1×	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3/2$	128×128
	Residual	32	1×1	128×128
2×	Convolutional	64	3×3	64×64
	Convolutional	128	3×3	32×32
	Residual	256	$3 \times 3/2$	32×32
8×	Convolutional	128	1×1	32×32
	Convolutional	256	3×3	16×16
	Residual	512	$3 \times 3/2$	16×16
8×	Convolutional	256	1×1	16×16
	Convolutional	512	3×3	8×8
	Residual	1024	$3 \times 3/2$	8×8
4×	Convolutional	512	1×1	8×8
	Convolutional	1024	3×3	8×8
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

图 1 Darknet-53 网络结构

Fig. 1 Structure of Darknet-53 network

1.2 K 邻域搜索

在目标跟踪领域通常有一个共识,即跟踪目标在相邻两帧序列图上的相对位移通常不会非常大。在此基础之上,本文提出了一种 K 邻域搜索方法,以此提高 YOLOv3 检测算法在选择区域的物体检测成功率。 K 邻域搜索以前一帧检测到的目标矩形区域 (序列图第一帧为手动选择矩阵区域) 为基础矩阵,后一帧搜索矩阵区域在该基础矩形周边,同时搜索矩阵区域中心点与该基础矩形区域中心点坐标位置重合,且满足

$$K = \frac{W_{\text{search}}}{W_{\text{object}}} = \frac{H_{\text{search}}}{H_{\text{object}}} \quad (1)$$

式中: W_{search} 和 H_{search} 分别表示矩形搜索区域的宽、高; W_{object} 和 H_{object} 分别表示前一帧目标矩形区域的宽、高。

K 邻域搜索方法如图 2 所示。

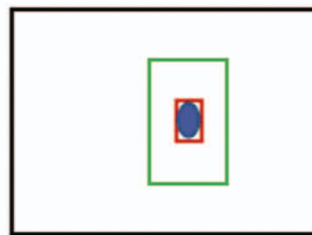


图 2 K 邻域搜索方法示意图

Fig. 2 Schematic of K -neighbor searching method

图 2 中,蓝色椭圆形为目标物体,红色框为前一帧所得目标框,绿色框为当前帧 K 邻域搜索矩阵框,其中 $K=3$ 。

采用 K 邻域搜索方法可以提高 YOLOv3 目标物体检测的成功率。在实际跟踪过程中,目标 K 邻域搜索可在一定程度上增加目标周围物体的检出率,提高目标跟踪效果。

图 3 中显示了同一张图片中不同 K 值下 YOLOv3 的检测结果,其中,白色矩形框均为目标矩形区域。

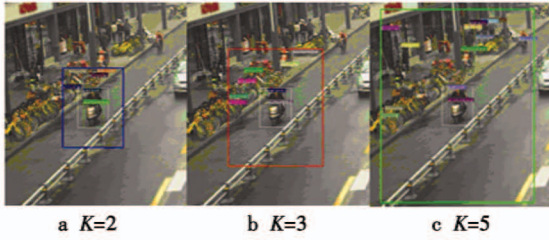


图 3 使用 K 邻域搜索方法的 YOLOv3 检测结果
Fig.3 Detection results of YOLOv3 with K -neighbor searching method

表 1 所示为图 3 中不同 K 值下目标矩形周边区域的物体检出数量。以 $K=2$ 为基准, $K < 2$ 认为是目标周边区域, $K > 2$ 认为是目标周边以外区域。可以看出,使用了不同 K 值的 K 邻域搜索方法的 YOLOv3 在图片相同大小的区域检出物体的数量是不同的, $K=3$ 时在物体周边检出效果较好。

表 1 图 3 中目标矩形周边物体检出数
Table 1 Numbers of objects detected around the destination rectangle in Fig. 3

K 值	类别		
	person	bicycle	motorbike
2	1	4	1
3	1	6	1
5	1	0	1

在实际跟踪过程中, YOLOv3 提取的候选目标框可能分布在图中各个位置, K 邻域搜索方法还能以 K 为阈值,筛选掉那些在 K 邻域搜索框以外的候选目标框,如图 4 所示。

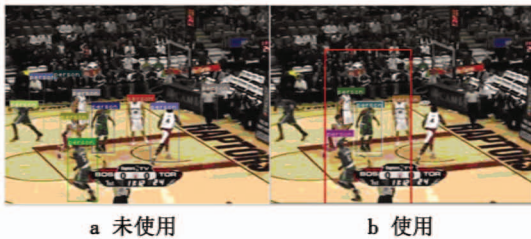


图 4 跟踪过程中是否使用 K 邻域搜索的检测结果对比
Fig.4 Contrast of detection results with/without K -neighbor searching method during object tracking

2 目标跟踪模型

2.1 特征提取

在目标跟踪时,多个目标候选框被检测出来之后,需要做一个筛选,以便在视频序列图像中找到与模板最相似的候选框作为跟踪目标^[8]。本文采用 HSV 颜色直方图特征和 LBP 直方图特征共同进行候选目标的筛选。

图像的颜色特征分为 RGB 和 HSV 空间特征,均

是全局性的特征,其描述子能够较好地表示目标形状的整体信息,对形变较大的目标来说鲁棒性很好^[9]。RGB 颜色空间的均匀性非常差,并且两种颜色之间的直觉差异色差不能表示为颜色空间中两点的距离,因此本文选择 HSV 空间特征。LBP 特征是局部特征,对物体的局部纹理特征具有很好的特征描述,能够对物体的空间边缘及纹理有较好的检测效果,同时 LBP 特征的运算量小,有利于检测速度的提升^[10]。

LBP 特征提取基本公式为

$$P(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c) \quad (2)$$

式中: (x_c, y_c) 是中心像素; i_c 是灰度值; i_p 是相邻像素的灰度值; s 是一个符号函数,即

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (3)$$

此外,采用 LBP 直方图特征而不是 LBP 特征是由于 LBP 特征与其位置信息紧密相关,直接对图片提取 LBP 特征进行判别分析时,提取的特征往往由于位置的不同而产生很大误差。为了避免这种影响,本文采用的是 LBP 直方图特征。

图 5 中,假设序列帧图 Crossing 中红衣女士是候选目标,截取出来命名为 Part of Crossing。图 5b、图 5d、图 5f 分别是序列帧图 Crossing、Crossing 的 HSV 直方图特征、Crossing 的 LBP 直方图特征;而图 5a、图 5c、图 5e 分别是 Part of Crossing 及其对应的 HSV 直方图特征和 LBP 直方图特征。

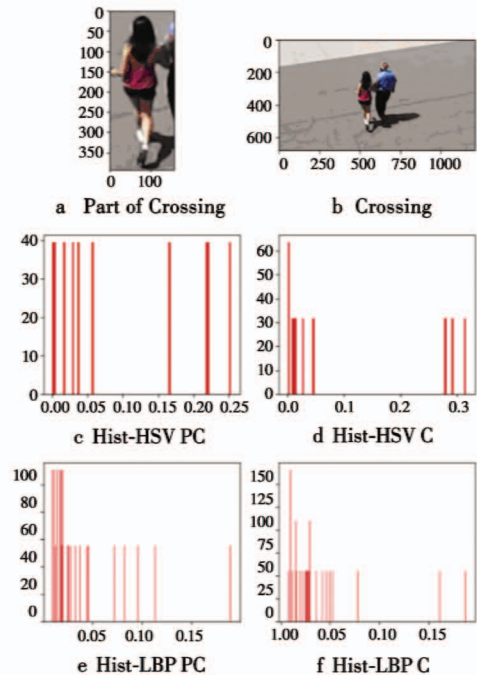


图 5 HSV 直方图和 LBP 直方图特征提取
Fig.5 Feature extraction of HSV and LBP histograms

2.2 目标选择

目标筛选过程中,首先通过检测类别过滤掉一部分非目标类别候选框,然后对剩下的目标候选框依次计算与跟踪目标之间的 HSV 直方特征相似度和 LBP 直方特征相似度,通过合适的权重系数得到相似度得分,本文 HSV 直方特征和 LBP 直方特征对应的权重系数分别为 1 和 2。之后选择相似度得分最大的目标候选框作为跟踪目标。HSV 和 LBP 两个直方特征相似度阈值分别为 6 和 3,若两个相似度没有达到设定的相似度阈值,则判定当前帧上没有要跟踪的目标,选择下一帧继续目标筛选。

HSV 直方图相似度和 LBP 直方图相似度按照巴氏相关系数公式计算,即

$$s(\mathbf{H}_1, \mathbf{H}_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{\mathbf{H}}_1 \bar{\mathbf{H}}_2 N^2}} \sum_I \sqrt{\mathbf{H}_1(I) \mathbf{H}_2(I)}} \quad (4)$$

$$s(\mathbf{L}_1, \mathbf{L}_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{\mathbf{L}}_1 \bar{\mathbf{L}}_2 M^2}} \sum_I \sqrt{\mathbf{L}_1(I) \mathbf{L}_2(I)}} \quad (5)$$

$$Y_i(x) = -\ln x_i \quad (6)$$

式中: $\mathbf{H}_1, \mathbf{H}_2$ 分别表示候选目标和模板目标的 HSV 特征直方图向量; N 表示直方图 bin 数目; $\bar{\mathbf{H}}_k = \frac{1}{N} \sum_J \mathbf{H}_k(J)$, $\mathbf{H}_k(J)$ 表示 \mathbf{H}_k 中序号为 J 的 bin 的 HSV 颜色向量统计值; $\mathbf{L}_1, \mathbf{L}_2$ 分别表示候选目标和模板目标的 LBP 特征直方图向量; M 表示直方图 bin 数; $\bar{\mathbf{L}}_k = \frac{1}{M} \sum_J \mathbf{L}_k(J)$, $\mathbf{L}_k(J)$ 表示 \mathbf{L}_k 中序号为 J 的 bin 的 LBP 特征向量统计值。

每个候选目标在计算出其与模板目标的两个直方图巴氏相似度之后,经过式(6)进行值放大,然后计算该候选目标与模板目标直方图相似度得分,即

$$S_i = W_1 Y(s(h_i, h_o)) + W_2 Y(s(l_i, l_o)) \quad (7)$$

式中: S_i 表示第 i 个目标候选框的得分; W_1, W_2 表示权重系数,取 $W_1 = 1, W_2 = 2$ 。最后根据每个候选目标的直方图相似度得分数值大小进行筛选,选择得分最高的候选目标为跟踪对象。

2.3 模板更新

在目标跟踪过程中,待跟踪目标可能会出现姿态、尺寸、光照等变化,造成跟踪不稳定或者目标丢失的情况,需要及时更新模板目标^[11]。传统的目标更新策略大多是计算当前帧目标与模板目标的相关系数相似度,当其值高于所设阈值时,将模板与最佳位置的图像通过加权得到新的模板^[12]。本文目标模板更新采用帧差阈值更新的策略。

考虑当前帧匹配目标与目标模板的两个直方图相似度特征得分,在当前帧匹配目标与模板目标的 HSV

直方特征及 LBP 直方特征相似度得分均满足阈值条件,同时满足帧差条件时,才更新模板目标。

模板更新条件为

$$\begin{cases} s(\mathbf{H}_i, \mathbf{H}_o) > s(\mathbf{H}_i, \mathbf{H}_o)_{th} \\ s(\mathbf{L}_i, \mathbf{L}_o) > s(\mathbf{L}_i, \mathbf{L}_o)_{th} \\ F_i - F_{i-1} > F_{th} \end{cases} \quad (8)$$

式中: $s(\mathbf{H}_i, \mathbf{H}_o)_{th}, s(\mathbf{L}_i, \mathbf{L}_o)_{th}$ 分别表示 HSV 和 LBP 直方图相似度得分阈值,本文分别为 6 和 3; F_i, F_{i-1} 表示当前匹配目标所在帧数和上一目标模板所在帧数; F_{th} 为帧差阈值,本文为 5。设置帧差条件是为了避免目标模板更新过于迅速,并且可以降低在模糊的帧图中更新目标模板的概率。

同时,在检测过程中可能会出现少量的帧无检测结果的情况,这时跟踪算法 TOD 采用上一帧跟踪目标结果作为当前帧的跟踪结果,且不更新模板目标。

2.4 TOD 算法流程

本文提出的基于 YOLOv3 的目标跟踪算法 TOD 的算法流程如下:

- 1) 在第一帧图像中手动获取跟踪目标,同时将其设为目标模板,提取目标模板的 HSV 直方特征向量和 LBP 直方特征向量;
- 2) 根据前一帧中的跟踪目标坐标位置生成 K 邻域搜索图, K 取 3,并在搜索图中采用 YOLOv3 检测得到检测结果;
- 3) 根据类别标签过滤检测结果,得到同一类别的目标候选框,若没有同类的候选目标,则以上一帧跟踪目标坐标作为当前帧跟踪结果;
- 4) 依次提取候选目标的 HSV 直方特征向量和 LBP 直方特征向量,并根据式(4)~式(7)计算两个直方图的相似度及相似度得分,选择得分最高的候选目标作为跟踪目标,如图 6 所示;
- 5) 判断跟踪目标模板更新条件是否满足,满足则更新模板,同时跳至步骤 2),不满足则不更新模板。

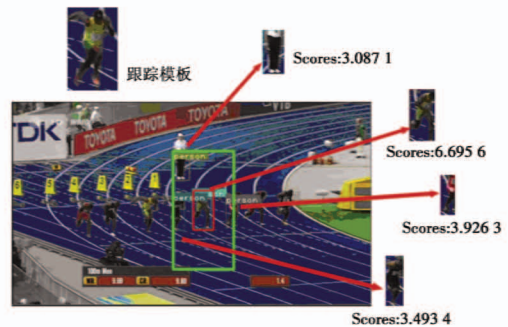


图 6 目标跟踪过程简要示意图

Fig. 6 Sketch of object tracking process

以 VOT-2017 序列图 Bolt1 第 30 帧图像为例,分析

目标跟踪过程中候选目标与模板匹配过程。左上角的小图是目标跟踪模板,检测类别是 person。图中,红色框是上一帧(第 29 帧)匹配的跟踪结果,则本图以此为基础,在 $K=3$ 的 K 邻域搜索同类目标(person),得到 4 个候选目标,图 6 中分别以红色箭头指出;同时计算每个候选目标与模板目标的直方图相似度得分,结果显示在对应小图像下面,选择得分最高的候选目标作为当前帧图像跟踪结果;根据目标模板更新条件是否满足,判断当前帧是否更新目标模板。

3 实验结果与分析

本文实验设备为 MSI 笔记本电脑,CPU 为 Intel-I5 处理器,GPU 为 GTX1050Ti。在 Windows 系统上、Py-Charm 开发平台下开发实现,结合 OpenCV3.4,编程语言选用 Python。

为了对比本文提出的跟踪算法实际性能,选用 4 种 OpenCV3.4 内置跟踪算法 BOOSTING,MIL,KCF 和 TLD(Tracking-Learning-Detection)^[13] 作为对比跟踪算法。测试序列图考虑到要包含目标姿势变化、尺度变化、旋转变化、光照变化等复杂影响因素^[14],选用 VOT-2017 序列数据集上的 Bmx,Crossing,Graduate,Iceskater1,Pedestrian1 和 Singer2 这 6 组具有代表性的序列图测试。

3.1 实验定性分析

实验选用不同颜色的矩形框对不同的跟踪算法进行区分。在每帧图片左上角有“#N”字样,N 代表帧序号。当某一算法丢失跟踪目标而跟踪失败时,该帧图片左侧会显示“XX False”字样,XX 代表跟踪算法名称。

对于姿态变化,在序列图 Iceskater1 中,跟踪目标是图中滑雪运动员,如图 7 所示。

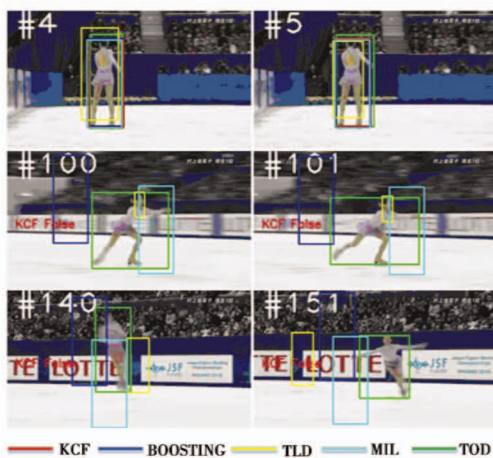


图 7 序列 Iceskater1 跟踪结果

Fig.7 Tracking results of sequences Iceskater1

当运动员仅仅摆动手臂时,如第 4、第 5 帧所示,5 个跟踪算法都能很好地跟踪上;当运动员开始下蹲前倾,四肢有较大变化时,如第 100、第 101 帧所示,对比的其他算法很难有较好的自适应性,本文跟踪算法 TOD 能很好地适应这些变化;当运动员完成复杂的转体动作并落地伸展双臂时,如第 140、第 151 帧所示,对比算法出现较大偏移,KCF 算法甚至跟踪不上,而 TOD 算法仍跟踪良好,体现出强大的适应性。

对于尺度变换,在序列图 Crossing 中跟踪目标是左侧红衣女士。场景中可以看到,当镜头由远及近时,跟踪目标的尺度在逐渐变大。如图 8 中第一行 3 幅图所示,在此过程中,其他对比算法虽然都能跟踪上,但都有不同程度上的偏移。

对于旋转变化,在序列图 Bmx 中,跟踪目标单车运动员在腾空跃起过程中,不仅有很大的姿势变化,同时还有很大的旋转。如第 11、第 29、第 58 帧所示,在所有的 5 种跟踪算法中,KCF 在第 40 帧后出现目标丢失,除本文跟踪算法 TOD 外,其他算法均出现不同程度的跟踪漂移。

对于光照变化,在序列图 Singer2 中可以看出效果。图中跟踪目标歌手在舞台中间,周围的光照环境一直在变化。如第 11、第 41、第 50、第 59 帧所示,每相邻的两帧之间的光照变化很大,在所有的 5 种跟踪算法中,其他各算法都有不同程度的偏移,尤其是在第 59 帧时,光照突然变化加剧,如图 8 中最后 3 幅图所示,仅本文跟踪算法 TOD 和 BOOSTING 算法能跟踪上目标。

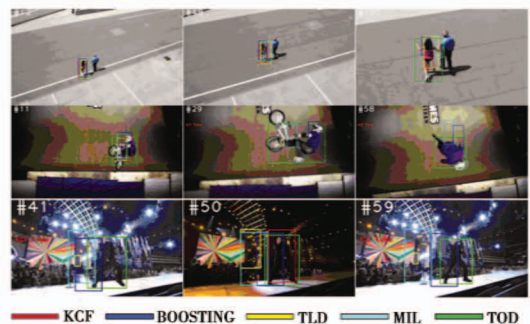


图 8 序列 Crossing,Bmx 和 Singer2 部分跟踪结果
Fig.8 Part of tracking results of sequences Crossing, Bmx and Singer2

3.2 实验定量分析

为了对比出实验中 5 种算法的实际跟踪性能,采用中心点距离误差和覆盖率这两个性能指标来衡量算法性能^[15]。

中心点距离误差是指预测目标中心点和实际目标中心点的欧氏距离,其算式为

$$D(p_1, p_2) = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2} \quad (9)$$

式中, $x_{p1}, y_{p1}, x_{p2}, y_{p2}$ 分别表示预测目标中心点坐标和实际目标中心点坐标。

覆盖率是指预测目标框与实际目标框的相交部分占其合并部分的比率,其算式为

$$P(p, g) = \frac{a(p) \cap a(g)}{a(p) \cup a(g)} \quad (10)$$

式中, $a(p), a(g)$ 分别表示预测目标区域和实际目标中心点区域。

在实际对比测试中,当某一种算法在跟踪过程中出现一些特定帧目标丢失的情形时,用前一帧跟踪结果作为当前帧跟踪结果参与对比。

部分可视化结果如图 9、图 10 所示。

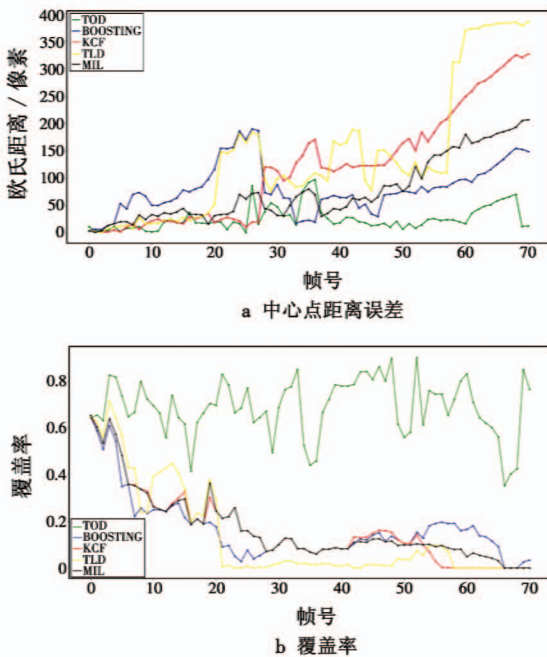


图 9 5 种跟踪算法在序列 Bmx 上的性能对比

Fig.9 Performance of 5 tracking algorithms on sequences Bmx

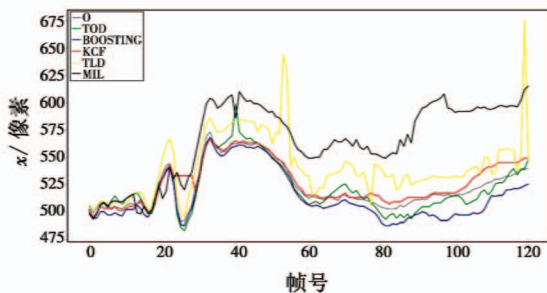


图 10 序列 Crossing 中目标跟踪轨迹(x 坐标)

Fig.10 The trajectory of object tracking on sequences Crossing (x coordinate)

图 9、图 10 中选用不同颜色曲线对不同的跟踪算法进行区分。图 10 中的灰色线表示基准跟踪轨迹。

在 VOT-2017 序列数据集的 Bmx, Crossing, Graduate, Iceskater1, pedestrian1 和 Singer2 这 6 组具有代表性的序列图上测试,定量观测 5 种跟踪算法在这些序列图上的实际性能。计算出各中心点距离误差平均值和平均覆盖率,其中最优值数据用下划线标出。结果如表 2、表 3 所示。

表 2 中心点距离误差平均值

Table 2 Average of center distance error 像素

	TOD	BOOSTING	KCF	TLD	MIL
Bmx	<u>27.568</u>	85.257	133.048	156.943	87.294
Crossing	<u>7.669</u>	41.378	24.356	24.441	19.135
Graduate	<u>7.266</u>	46.138	45.258	73.441	35.462
Iceskater1	<u>11.211</u>	55.955	30.198	78.977	38.251
Pedestrian1	<u>6.59</u>	92.472	80.613	83.365	100.704
Singer2	12.579	30.241	<u>10.546</u>	78.81	14.317

表 3 平均覆盖率

Table 3 Average of coverage %

	TOD	BOOSTING	KCF	TLD	MIL
Bmx	<u>68.379</u>	16.012	15.555	12.840	16.144
Crossing	<u>58.896</u>	51.361	34.664	52.240	44.552
Graduate	<u>69.188</u>	22.925	22.193	18.520	28.272
Iceskater1	<u>63.201</u>	26.266	45.046	16.726	40.187
Pedestrian1	<u>55.252</u>	3.817	4.109	13.718	1.663
Singer2	58.442	49.928	73.711	21.839	67.548

从表 2 和表 3 可以看出,所选的 6 组序列图中,在中心点距离误差指标上,除了 KCF 算法在 Singer2 序列图上的表现,其他组序列图都是本文跟踪算法 TOD 表现最优;而在平均覆盖率指标上,除 Singer2 序列图之外,其他序列图也都是本文跟踪算法 TOD 最优。

对于参与对比的 4 种跟踪算法中表现较好的 KCF 算法,其在 6 组序列图上的中心点距离误差的平均值约为 54.003 像素,覆盖率平均值为 32.546%,而本文跟踪算法 TOD 这两个数据分别为 12.147 和 62.226%,性能分别提高了 77.51% 和 91.94%。

由于本文选用基于深度学习的检测模型 YOLOv3,而该检测算法对实验设备性能要求比较高,故跟踪速度稍显不足。本文跟踪算法 TOD 的平均速度为 11 帧/s。根据 YOLOv3 论文推荐配置,YOLOv3 的检测速度可以达到 45 帧/s 左右^[6],而在本文实验相同配置下,YOLOv3 的实际检测速度在 13 帧/s 左右。因此,可以推断出在硬件配置升级后,本文算法的跟踪速度将会有进一步的提升。

综上所述,对比多组序列图的跟踪性能后可以看出,本文算法 TOD 的实际性能优于其他 4 种跟踪算法,即使是在跟踪目标发生姿势变化、尺度变化、旋转变化、光照变化等复杂情况下仍然能很好地跟踪目标,跟踪性能提升明显。

4 结束语

本文提出了一种以深度学习检测模型 YOLOv3 为基础的目标跟踪算法 TOD。在目标所在区域使用 K 邻域搜索检测出候选目标,同时,结合跟踪目标的颜色直方特征和局部二值特征进行目标筛选达到跟踪目的。实验结果显示,相较于目前几种经典的跟踪算法,本文算法有更好的性能表现,在跟踪目标发生姿态变化、尺度变化、旋转变化和光照变化等复杂情况下也具有很好的鲁棒性。但不足之处在于其跟踪速度没有达到实时要求,在本文实验环境下只有 11 帧/s,其跟踪速度性能有待提高。

参考文献

- [1] 闵召阳,赵文杰. 基于深度学习的目标抗干扰跟踪算法[J]. 红外技术,2018,40(2):176-182.
- [2] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(3):583-596.
- [3] BABENKO B, YANG M H. Visual tracking with online multiple instance learning[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2009:11-17.
- [4] 管皓,薛向阳,安志勇. 深度学习在视频目标跟踪中的应用进展与展望[J]. 自动化学报,2016,42(6):834-847.
- [5] 唐聪,凌永顺,杨华,等. 基于深度学习物体检测的视觉跟踪方法[J]. 红外与激光工程,2018,47(5):0526001-1-0526001-11.
- [6] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. [2018-11-02]. <https://arxiv.org/pdf/1804.02767v1.pdf>.
- [7] 郑志强,刘妍妍,潘长城,等. 改进 YOLO V3 遥感图像飞机识别应用[J]. 电光与控制,2019,26(4):28-32.
- [8] 田健,王开军. 自适应紧致特征的超像素目标跟踪[J]. 中国图象图形学报,2017,22(10):1409-1417.
- [9] 毕威,黄伟国,张永萍,等. 基于图像显著轮廓的目标检测[J]. 电子学报,2017,45(8):1902-1910.
- [10] 吉培培,陈息,刘哲,等. 结合 HOG-LBP 特征及多实例在线学习的随机森林目标跟踪方法[J]. 宁波大学学报:理工版,2015,28(4):42-47.
- [11] 陆惟见,尚振宏,刘辉,等. 基于多模板的鲁棒运动目标跟踪方法[J]. 传感器与微系统,2018,37(2):53-56.
- [12] BAL A, ALUM M S. Automatic target tracking in FLIR image sequences [C]//Automatic Target Recognition, SPIE, 2004, 5423:30-36.
- [13] KALAL Z, MATAS J, MIKOLAJCZYK K. P-N learning: bootstrapping binary classifiers by structural constraints [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2010:49-56.
- [14] 罗海波,许凌云,惠斌,等. 基于深度学习的目标跟踪方法研究现状与展望[J]. 红外与激光工程,2017,46(5):6-12.
- [15] 王海军,葛红娟,张圣燕. 基于 L1 范数和最小软阈值均方的目标跟踪算法[J]. 山东大学学报:工学版,2016,46(3):14-22.