基于深度学习物体检测的视觉跟踪方法

唐 聪[1,2],凌永顺[1,2],杨 华[1,2],杨 星[1,2],郑 超[1,2]

(1. 国防科技大学,安徽 合肥 230037; 2. 脉冲功率激光技术国家重点实验室,安徽 合肥 230037)

摘 要:提出了一种基于深度学习物体检测的视觉跟踪方法。该方法利用深度学习在特征表达上的优势,采用基于回归的深度检测模型 SSD(Single Shot Multibox Detector)提取候选目标,并结合颜色直方图特征和 HOG(Histogram of Oriented Gradient)特征进行目标筛选,实现目标跟踪。为了提升深度检测模型的物体检测性能,文中构建了多尺度目标搜索图,可在一张图上实现不同尺度的目标检测。在标准跟踪测试库上选取八个具有代表性的跟踪视频序列,并选取六种具有代表性的跟踪方法进行了对比测试。结果表明,文中所提方法在跟踪效果上,整体优于参与对比的其他算法,且对于物体姿态变化、尺寸变化、旋转变化、光照变化、复杂背景杂波等影响因素具有较好的鲁棒性。

关键词:视觉跟踪; 深度学习; SSD; 非在线更新

中图分类号:TP391.4 **文献标志码**:A **DOI**:10.3788/IRLA201847.0526001

A visual tracking method via object detection based on deep learning

Tang Cong[1,2], Ling Yongshun[1,2], Yang Hua[1,2], Yang Xing[1,2], Zheng Chao[1,2]

(1. National University of Defense Technology, Hefei 230037, China;
2. State Key Laboratory of Pulsed Power Laser Technology, Hefei 230037, China)

Abstract: A visual tracking method via object detection based on deep learning was proposed. In consideration of the advantages of deep learning in feature representation, deep model SSD (Single Shot Multibox Detector) was used as the candidate object extractor in the tracking model. Simultaneously, the color histogram feature and HOG (Histogram of Oriented Gradient) feature were combined to select the tracking object. In the process of tracking, multi-scale object searching map, which was applied to implement the object detection in different scales, was built to improve the detection performance of deep learning model. In the experiment of eight respective tracking video sequences in the baseline dataset, compared with six typical tracking methods, the proposed method has better performance in tracking effect, and has better robustness in the tracking challenging factors, such as deformation, scale variation, rotation variation, illumination variation, and background clutters.

Key words: visual tracking; deep learning; SSD; non-online updating

收稿日期:2017-12-05; **修订日期**:2018-01-03

基金项目:国家自然科学基金(61405248,61503394);安徽省自然科学基金(1708085MF137)

作者简介:唐聪(1989-),男,博士生,主要从事计算机视觉、深度学习、模式识别等方面的研究。Email:tangcong_eei@163.com

导师简介:凌永顺(1937-),男,中国工程院院士,教授,博士生导师,主要从事光电工程等方面的研究。Email:lys@126.com

0 引 言

目标跟踪已经成为计算机视觉领域重要的研究方向和研究热点。军事上,可应用于精确制导武器、无人机侦察监视等领域;民用上,可应用于机器人导航、人机交互,行人与车辆的视频监控等领域[1-3]。经过多年的发展,目标跟踪技术已经取得了长足的进步,但依然面临多方面的挑战,如目标外观改变、光照变化、遮挡、相似目标,这些因素都将导致目标漂移甚至跟踪失败。一般地,跟踪方法可以分为两类:生成类跟踪和判别类跟踪。生成类跟踪一般是在当前帧对目标区域建模,然后在后续帧中找出最可能是目标的候选区域作为跟踪目标,正常情况下,具有较高的精度。比较典型的例子有基于稀疏编码[4],主成分方法[5],字典学习[6]等方法。若采用相关滤波,凭借快速傅里叶变换和矩阵操作将具有较好的实时性,如 KCF(Kernerlized Correlation Tracking),其跟踪速度可达 172 帧/s[7]。判别类跟踪一般是先进行特征提取,再采用一个分类器对目标和背景进行区分,这类方法多采用机器学习的方法,比如多事例学习(Multiple Instance Learning,MIL)[8],boosting 变种[9],支持向量机(Support Vector Machine,SVM)[10],因其具有较强的特征提取和区分能力,这类方法对于复杂环境的适应性强。

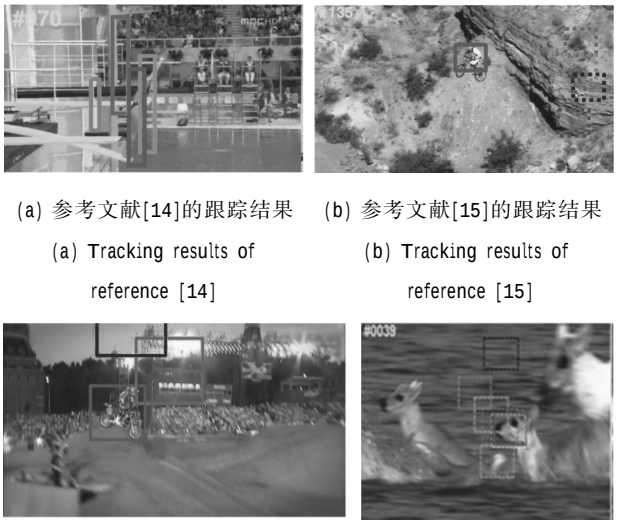
目前,基于深度学习的目标跟踪方法大多采用离线训练结合在线更新的模式,但该模式需实时采用大量正负样本进行模型修正,通常速度较慢。比如,第一个将深度学习引入目标跟踪的跟踪方法 DLT(Deep Learning Tracking)[11],其采用粒子滤波的方法,每一个代表粒子的图像块都需要通过深度模型,数据运算量较大,使其跟踪速度受限。再比如 VOT2015 的冠军算法 MDNet(Muti-Domain Networks)[12],其同样采用在线更新模板的方法,在跟踪精度上取得了当年的最好成绩,但其跟踪速度却 1s 不及 1 帧。而目前仅采用离线训练而不进行在线更新的方法很少,GOTURN(Generic Object Tracking Using Regression Networks)[13] 是其典型代表,其速度能达到 100 fps 以上,但是精度相比于 MDNet 相差较多。

文中提出了一种基于深度学习物体检测的物体跟踪方法,称之为 TDLD(Tracking via Deep Learning Detector)。TDLD 采用基于深度学习的物体检测算法进行候选目标提取,同时结合全局性的颜色直方图特征和局部性的 HOG 特征进行目标选择,以实现跟踪。该方法利用了目前深度学习在物体检测上的强大优势,顺利将其迁移到目标跟踪上,提升了方法的鲁棒性,具有较强的理论价值和参考意义。

1 物体检测迁移至目标跟踪的可行性

1.1 传统跟踪的跟踪漂移分析

传统的跟踪方法,跟踪过程中,在目标跟踪失败之后,多数漂移到不可知的位置,如图 1 所示。



(a) 参考文献[14]的跟踪结果 (b) 参考文献[15]的跟踪结果

(a) Tracking results of reference [14] (b) Tracking results of reference [15]

(c) 参考文献[16]的跟踪结果 (d) 参考文献[17]的跟踪结果

(c) Tracking results of reference [16] (d) Tracking results of reference [17]

图 1 传统跟踪方法中的跟踪漂移

Fig.1 Tracking drift in the traditional tracking methods

如图 1 所示,上述跟踪结果中,除去所引用文献自身所提出的方法外,其他参与对比的跟踪方法均发生了不同程度的跟踪漂移,且这些位置处的框选目标与模板目标并非相似。从跟踪的效果上来看,在视觉跟踪的过程中,跟踪目标应该与模板目标在外观上近似,通常地,目标多为一个物体。假如在跟踪的过程中,可以首先根据上一帧目标位置检测出当前帧中一定范围内的物体,再以这些物体为候选目标进行识别,进而可实现目标跟踪。

目前,深度学习在物体检测上性能非常出色,明显优于传统物体检测方法。一般地,基于深度学习的物体检测方法可分为两类:基于区域的物体检测和

基于回归的物体检测。从精度上讲,基于区域的物体检测方法相对来说更好一些,但实时性比较差。而基于回归的物体检测虽然精度稍有一定的下降, 但实时性较好。综合考虑之下,文中选用基于回归的物体检测方法。在基于回归的物体检测方法中,主要有两种代表方法:YOLO[18]、SSD[19],相比之下,SSD 相比于 YOLO 精度更高一些,因此,文中选用 SSD 作为物体跟踪的前端,实现第一步的物体检测,从而提取出候选目标。

1.2 SSD

SSD 是一种单次检测深度神经网络, 同时结合了 YOLO 的回归思想和 Faster R-CNN 的 anchors 机制。采用回归的思想,可以简化神经网络的计算复杂度,提高算法的实时性;采用 anchors 机制,可以提取不同宽高比尺寸的特征。另外,SSD 针对不同尺度的特征表达不同这一特点,采取了多尺度目标特征提取的方法[20],该设计有助于提升检测不同尺度目标的鲁棒性。

SSD 的架构主要分为两部分:一部分是位于前端的深度卷积神经网络,采用的是去除分类层的图像分类网络,如 VGG[21],用于目标初步特征提取;另一部分是位于后端的多尺度特征检测网络,是一组级联的卷积神经网络,将前端网络产生的特征层进行不同尺度条件下的特征提取。SSD 框架如图 2 所示。

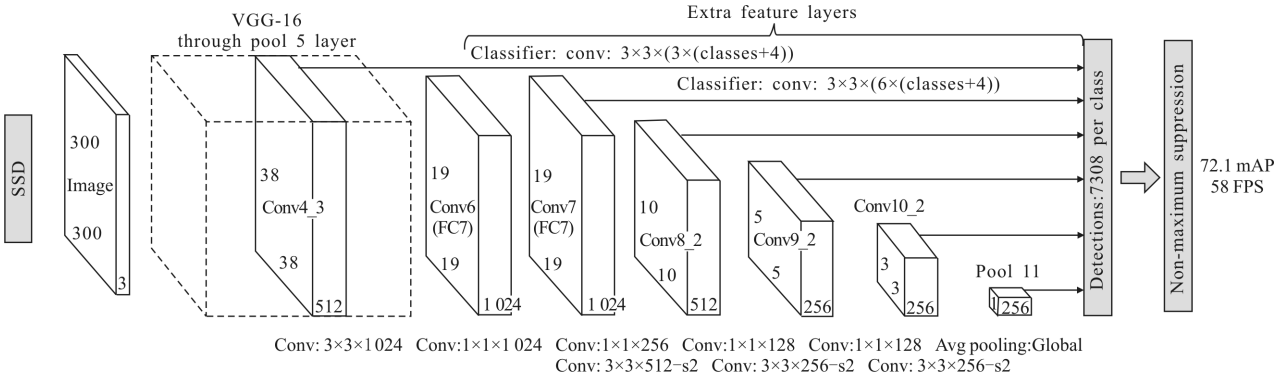


图 2 SSD 框架

Fig.2 Framework of SSD

一般地,物体检测过程中,图像尺寸一般较大,而在实际跟踪过程中, 目标只是来源于图像的一部分,通常尺寸较小,使得跟踪的搜索区域一般不会太大,实验发现,以跟踪的搜索区域作为物体检测的输入,可对搜索区域内的目标实现高效检测。例如,对于同样一个检测场景,物体检测阈值设为 0.6,分别将整个场景作为模型输入和将局部区域作为模型输入,对比 SSD 模型检测结果,如图 3 所示。为体现问题的针对性, 这里的局部区域特别选择了以整体场景作为输入时未检测出的物体区域。

如图 3 所示,同等条件下,以局部场景作为物体检测模型的输入相比于以整体场景作为输入, 模型的物体检测能力有较大的提升, 可实现对以整体场景作为输入时未检测出物体的检测。虽然物体检测类别标签与真实类别标签存在不一致的现象, 比如图 3 中在局部检测出的五只鸟中, 三只鸟的标签已发生变化,但这并不影响物体跟踪的应用。在目标跟踪的过程中, 其跟踪结果只依赖于物体框的准确定位, 而与物体的类别无关。因此, 基于深度学习的 SSD 物体检测模型在物体检测方面将很好地契合物体跟踪的需要。

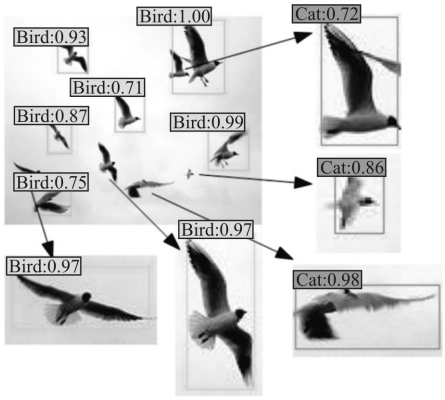


图 3 全局物体检测与局部物体检测对比

Fig.3 Comparison between global object detection and local object detection

2 基于深度学习物体检测的视觉跟踪模型

2.1 多尺度目标搜索图

为提高检测成功率，文中设计了一种多尺度目标搜索图，用以进行目标的高效检测。多尺度目标搜索图以前一帧目标模板为基础，同时涵盖了六种不同尺度的搜索场景。其示意图如图 4 所示。

(a) 实时场景 (b) 多尺度目标搜索图

(a) Real-time scene (b) Multi-scale object searching map

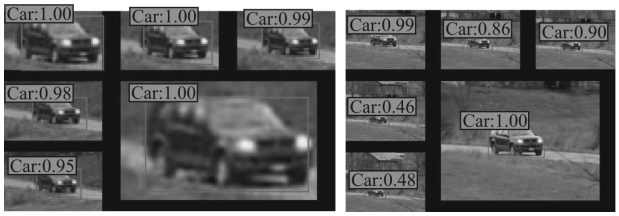
图 4 多尺度目标搜索图生成

Fig.4 Generation of multi-scale object searching map

如图 4 所示，区域Ⅰ为无任何尺度变换的目标搜索区域，其大小等于上一帧目标的大小，区域Ⅱ、区域Ⅲ、区域Ⅳ、区域Ⅴ分别代表不同尺度条件下的目标区域，具体地，参数设置可根据实际需要进行设置，文中选择 1.25、1.5、1.75、2。区域Ⅵ为区域Ⅰ的插值放大区域，目的是增强模板区的目标信息，以提高模板区的检测成功率，这里为二倍插值放大。在实际跟踪过程中，目标丢失后，为了提升目标的搜索成功率，区域Ⅱ、区域Ⅲ、区域Ⅳ、区域Ⅴ的参数可进一步更新，比如更新为 1.5、2、2.5、3，增大检测目标的搜索区域。

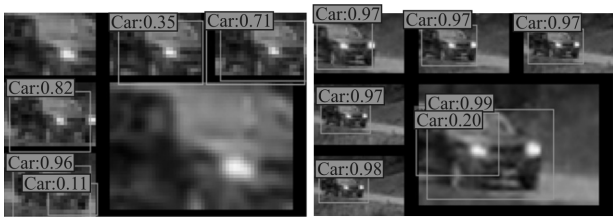
在多尺度目标搜索图条件下，目标检测对目标的尺寸变化，位置偏移等因素具有较强的鲁棒性。如图 5 所示，图 5(a)为正常检测模式，该模式下，各个视窗在给定置信度阈值条件下均能实现对目标的检测；图 5(b)为大视窗检测模式，各个视窗在给定置信度阈值条件下均能实现对目标的检测，同时，在下一帧更新模板尺寸的过程中，切换到正常检测模式(图5(a))；图 5(c)为小视窗检测，该模式下，区域Ⅰ和区域Ⅵ只能对物体的局部信息进行检测，在给定置信度阈值条件下不能很好地完成物体检测，而其余四个区域可实现互补；图 5(d)为偏移视窗检测，该模式下，区域Ⅰ、Ⅱ、Ⅲ、Ⅵ因为模板框中心偏移物体中心，只能实现局部检测，与图 5(c)一样不能很好地完成物体检测，而区域Ⅳ、Ⅴ则可实现较好地物体检测。

图 5 中黑色边框是为了隔断不同区域之间的联系，否则在非极大值限制(Non-maximum Suppression，NMS)操作的过程中，容易出现粘连的情况，造成不同区域之间相互干扰，检测出错误的物体区域。进行物体检测时，为了提高物体检测的准确性，一般阈值设置较高，而文中为了提高检测灵敏度，阈值一般设置较低，以提高检测候选区域的数量。这里检测置信度阈值设置为 0.1。



(a) 正常模式 (b) 大视窗模式

(a) Normal mode (a) Large view mode



(c) 小视窗模式 (d) 视窗偏移模式

(c) Small view mode (d) View shifting mode

图 5 基于多尺度目标搜索图的目标检测

Fig.5 Object detection based on multi-scale object searching image

2.2 目标选择

当物体被检索出来之后，对于多个物体候选框进行物体挑选，挑选出与模板目标最相似的物体作为跟踪目标。这里选用颜色直方图和 HOG 两个特征共同进行目标的挑选。颜色直方图提取的全局特征，对于目标的形变鲁棒性较强，而 HOG 提取的是局部特征，对于物体的空间边缘等检测效果较好。结合这两种特征，可以更好地描述候选目标与模板目标之间的相似性。

对于候选目标，首先通过设置检测置信度阈值，去除一部分候选目标，此时再对候选目标按照检测置信度进行从高到低的排名，进而计算排名后的候选目标与模板目标之间的颜色直方图相似度和 HOG 相似度，分别进行阈值条件判断，若两者均满

足阈值条件则选出该候选目标作为相似目标，并挑选出相似目标中具有最佳颜色直方图相似度的候选目标作为目标。若无候选目标或者所有候选目标都不满足阈值条件,则认为当前视场内无目标。

颜色直方图、HOG 直方图的相似度均采用计算相关系数的方法。其公式如下：

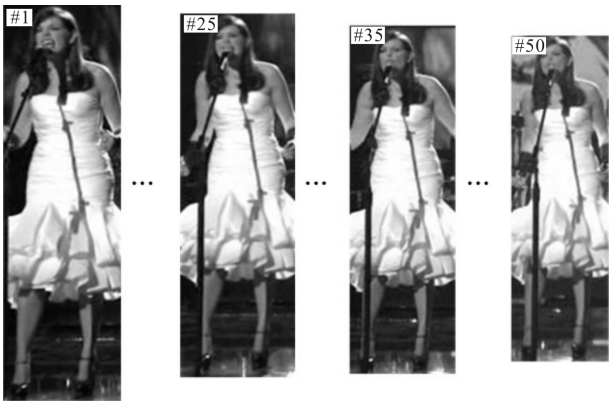
$$C(H_1,H_2)=\frac{\sum_I(H_1(I)-\bar{H}_1)(H_2(I)-\bar{H}_2)}{\sqrt{\sum_I(H_1(I)-\bar{H}_1)^2\sum_I(H_2(I)-\bar{H}_2)^2}} \quad (1)$$

$$C(G_1,G_2)=\frac{\sum_I(G_1(I)-\bar{G}_1)(G_2(I)-\bar{G}_2)}{\sqrt{\sum_I(G_1(I)-\bar{G}_1)^2\sum_I(G_2(I)-\bar{G}_2)^2}} \quad (2)$$

式中：H_1,H_2 分别为模板目标图像块与候选目标图像块的颜色直方图向量；$\bar{H}_k=\frac{1}{N}\sum_J H_k(J)$，$H_k(J)$为 H_k 中序号为 J 的 bin 的颜色向量统计值；N 代表直方图的 bin 数。G_1,G_2 为模板目标图像块与候选目标图像块的 HOG 直方图向量，为多维数组拼接成的向量；$\bar{G}_k=\frac{1}{M}\sum_J G_k(J)$，$G_k(J)$为向量 G_k 中序号为 J 的元素中的梯度信息统计值；M 代表向量中元素的数量。

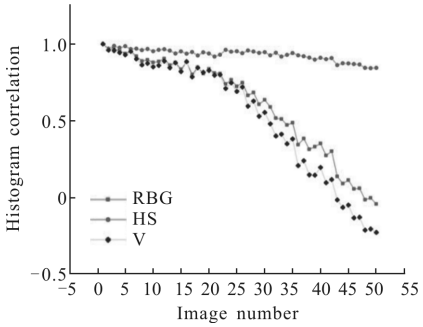
关于颜色直方图相似度的计算，这里采用 HSV 色彩空间,并只保留 H 通道、S 通道,去除 V 通道。由于 H 通道表征的是色调信息,S 通道表征的是饱和度信息,V 通道主要表征亮度信息,因此,采用 H、S 通道进行目标选择，可一定程度上减弱光照变化对目标判断的影响。图 6(a)展示了某视频序列中目标 50 帧光照变化,同时,其附带一定的尺度变化,但尺度变化对于颜色直方图的计算影响不大，图 6(b)为后 49 帧与第 1 帧的 RGB 直方图相似度,HS 直方图相似度,V 直方图相似度随帧图像变化的曲线。其中,RGB 直方图相似度为 R、G、B 三通道的直方图相似度均值,HS 直方图相似度为 H、S 通道的直方图相似度均值。

如图 6 所示，可以看出,RGB 直方图相关系数曲线与 V 直方图相关系数曲线趋势较一致，同时,RGB 直方图相关系数曲线随帧图像光照的增强而下降较快,表明 RGB 颜色空间直方图相关系数对光照变化较敏感，而 HS 直方图相关系数曲线随帧图像光照增强的变化则相对较平缓。因此，文中选用 HS 直方图相关系数来评价候选目标图像和模板目标图像之间的相似度。



(a) 光照变化(第 1 帧至第 50 帧)

(a) Illumination variation (from frame 1 to frame 50)



(b) RGB、HS、V 直方图相关系数曲线

(b) Histogram correlation curves of RGB, HS and V

图 6 RGB、HS 颜色空间受光照变化的影响

Fig.6 Influence of illumination variation on RGB and HS color space

与颜色直方图的计算不同,HOG 特征的计算对于图像的尺寸有一定的要求，而且计算目标图像与候选图像的 HOG 特征相似度时，两个 HOG 特征向量的维度应该一样，即要求目标图像与候选目标图像的尺寸应该一致。一般地,计算 HOG 特征时,图像尺寸应为单元格(cell)尺寸的整数倍,方可进行有效计算,而 cell 的边长为 8 pixel,一个块(block)一般为 2×2 个 cell,即 block 的边长为 168 pixel,而 HOG 特征向量是由若干 block 向量按照一定的顺序排列组成，计算后的 HOG 特征向量至少应包含 2×2 个 block 向量，因此，调整之后图像的尺寸可设置为 32 pixel×32 pixel。将目标图像和候选目标图像均尺度变换到该尺寸,进而可计算 HOG 特征。HOG 特征计算过程如下：

(1) 归一化处理。对图像进行归一化处理,可有效抑制图像局部的阴影和光照变化的影响，提升特

征描述符对光照的鲁棒性。由于 HOG 的计算对颜色信息不依赖,可变换到灰度图像空间,再进行Gamma校正,实现归一化处理。Gamma 校正公式如下:

$$I(x,y)=I(x,y)^{Gamma} \tag{3}$$

式中:**Gamma** 一般取 0.5。

(2) 计算图像梯度。梯度幅值和梯度方向公式如下:

$$g(x,y)=\sqrt{g_x(x,y)^2+g_y(x,y)^2} \tag{4}$$

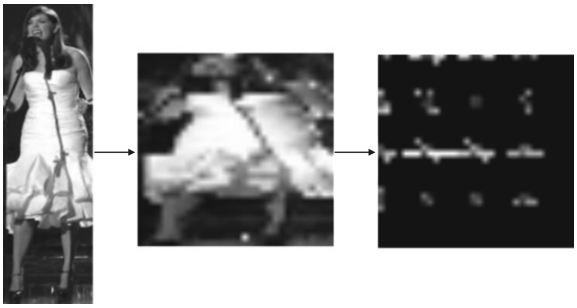
$$\alpha(x,y)=\arctan\left(\frac{g_y(x,y)}{g_x(x,y)}\right) \tag{5}$$

式中:$g_x(x,y)$、$g_y(x,y)$分别为(x,y)在 x 方向上和 y 方向上的梯度。

(3) 计算 cell 梯度方向直方图,bin 设置为 8。

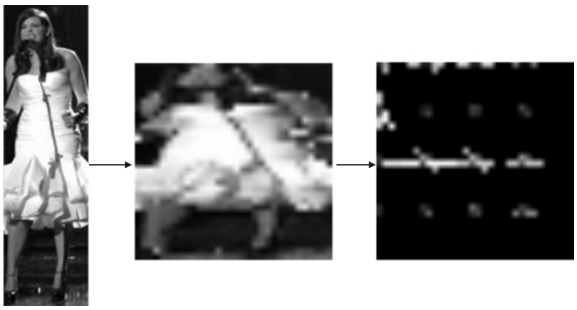
(4) 归一化块 block 梯度直方图,按顺序拼接得到 HOG 特征向量。

图 7 展示了目标图像与候选目标图像的 HOG 特征可视化过程。



(a) 模板目标

(a) Template target



(b) 候选目标

(b) Candidate target

图 7 模板目标与候选目标的 HOG 特征可视化过程

Fig.7 Visualization process of HOG feature of template target and candidate target

如图 7 所示, 首先将模板目标图像与候选目标图像变换到 32×32 的固定尺寸,分别计算其 HOG 特征,进而根据公式(2)计算两者的相关系数。经计算,目标图像与候选图像的 HOG 相关系数为 0.925。

跟踪过程中,可能出现检测模型在少数特定帧无检测结果的现象,针对这种情况,文中以模板在当前帧进行相关匹配的结果作为跟踪结果。这里采用快速相关跟踪的方法[22],其计算相关系数的公式如下:

$$\rho(x,y)=\frac{R_1(x,y)-\bar{T}R_2(x,y)}{\sqrt{R_3(x,y)-\frac{1}{mn}R_2^2(x,y)}\sqrt{\sum_{k=1}^{m}\sum_{l=1}^{n}T^2(k,l)-mn\bar{T}^2}} \tag{6}$$

$$R_1(x,y)=\sum_{k=1}^{m}\sum_{l=1}^{n}I(x+k,y+l)T(k,l) \tag{7}$$

$$R_2(x,y)=\sum_{k=1}^{m}\sum_{l=1}^{n}I(x,y) \tag{8}$$

$$R_3(x,y)=\sum_{k=1}^{m}\sum_{l=1}^{n}I^2(x+k,y+l) \tag{9}$$

式中:$R_1(x,y)$为 $I_{x,y}$ 与模板 T 的相关,可通过傅里叶变换和反傅里叶变换快速求出,$R_2(x,y)$和 $R_3(x,y)$分别是 $I_{x,y}$ 的灰度值积分和能量积分,可通过求解整幅图的积分图的方法快速求得。由于整个运算基于快速傅里叶变换和积分图,计算速度相当于传统相关跟踪的 100 多倍,对于 576×432 的场景图片,匹配大小为 59×114 目标,Python 运行环境下,速度可达 0.01 s。

同时,在跟踪过程中,通过检测与挑选最后得到的目标, 包括利用相关匹配得到的目标均可能出现尺度的突变,造成跟踪目标的突变,为了减小候选目标尺寸突变带来的跟踪不稳定, 这里选用候选目标与模板的交并比(Intersection-over-union,IoU)作为阈值条件,当其小于一定阈值,也认为该候选目标无效。这里,IoU 设置为 0.4。

2.3 模板更新策略

传统模板更新在相关系数低于某个阈值时将模板与最佳匹配位置的图像加权得到新的模板[23],而文中的模板更新在相关系数高于指定阈值时将最佳匹配位置的图像作为新的模板。模板更新时兼顾颜色直方图相似度和 HOG 相似度,其更新条件是在候选目标与模板目标同时满足所设定的高颜色直方图相似度阈值和较高 HOG 相似度阈值时,或者同时满足所设定高 HOG 相似度阈值和较高颜色直方图相似度阈值时。模板更新,具体步骤如下:

(1) 根据公式(1)、(2)计算颜色直方图相似度 $\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)$和 HOG 相似度 $\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)$。

(2) 求出候选目标中满足阈值条件中颜色直方图相似度最高的候选目标。阈值条件:

$$\begin{cases}\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)>\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)_{\text{threshold}}\\ \mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)>\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)_{\text{threshold}}\end{cases}\tag{10}$$

式中:$\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)_{\text{threshold}}$,$\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)_{\text{threshold}}$ 分别为设置的颜色直方图相似度阈值、HOG 相似度阈值。

(3) 判断候选目标是否满足模板更新条件,若满足则更新模板。模板更新条件:

$$\begin{cases}\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)>\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)^{\mathbf{H}}_{\text{updating}}\\ \mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)>\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)^{\mathbf{L}}_{\text{updating}}\end{cases}\tag{11}$$

或

$$\begin{cases}\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)>\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)^{\mathbf{L}}_{\text{updating}}\\ \mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)>\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)^{\mathbf{H}}_{\text{updating}}\end{cases}\tag{12}$$

式中:$\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)^{\mathbf{H}}_{\text{updating}}$、$\mathbf{C}(\mathbf{H}_1,\mathbf{H}_2)^{\mathbf{L}}_{\text{updating}}$ 分别为设定的高、较高的颜色直方图相似度更新阈值;$\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)^{\mathbf{H}}_{\text{updating}}$、$\mathbf{C}(\mathbf{G}_1,\mathbf{G}_2)^{\mathbf{L}}_{\text{updating}}$ 分别为设定的高、较高的 HOG 相似度更新阈值。

为了加强模型的鲁棒性,这里采用长时模板、短时模板两种模板。长时模板,其更新条件相对于短时模板相对苛刻,添加了如下条件:

$$\mathbf{N}_2-\mathbf{N}_1>\mathbf{N}_{\text{threshold}}$$

式中:$\mathbf{N}_1$ 代表前一次更新时的图像帧数;$\mathbf{N}_2$ 代表当前图像帧数;$\mathbf{N}_{\text{threshold}}$ 代表长时模板更新的最小帧差数。

同时使用两种模板,一方面可以让模板及时更新,充分利用帧与帧之间的连续性,同时可以纠正短时模型更新过快带来的目标丢失。

在跟踪过程中,更新多尺度搜索图的视窗时,同时加入运动信息,使视窗与目标运动适应性更佳,其运动信息为临近两帧的目标中心偏移量。

2.4 算法流程

基于上述描述,文中提出了一种基于深度学习物体检测的目标跟踪算法 TDLD,整个算法的具体步骤如下:

(1) 根据第一帧图像中目标的初始状态,获取模板,生成多尺度目标搜索图;

(2) 对多尺度目标搜索图进行物体检测,生成候选目标;

(3) 根据置信度阈值滤除低置信度的候选目标;

(4) 将剩余候选目标根据置信度大小从高到低排序,得到 Top-K 候选目标集合;

(5) 根据 Top-K 候选目标集合,逐次计算候选目标与目标模板(包含长时模板、短时模板)的颜色直方图相似度和 HOG 相似度,并判断阈值条件,选出满足阈值条件的颜色直方图相似度最高的候选目标,若不存在,则进行相关匹配,进而判断是否满足 IoU 条件,若满足,则认为该候选目标为当前跟踪目标,输出跟踪结果,若不满足,切换多尺度目标搜索图模式,增大视窗区域,在下一帧继续搜索目标;

(6) 判断模板更新条件,结合与上次更新模板时的帧差,更新长时模板、短时模板,进而根据模板尺寸和多尺度目标搜索图模式在下一帧图像上生成新多尺度目标搜索图,并跳转至步骤(2)。

3 实验结果与分析

文中挑选了 OTB-100[24]数据集中八个跟踪视频序列(Basketball,Bolt2,Car24,Gym,Singer1,Human2,CarScale,MountainBike),主要涵盖了姿态变化、尺度变化、旋转变化、光照变化、复杂背景五种影响因素。而对比算法则选取了具有代表性的六种跟踪算法,包括四种非基于深度学习的跟踪方法和二种基于深度学习的跟踪方法,以评价其算法性能。四种非基于深度学习的跟踪方法分别是 TLD(Tracking-Learning-Detection)[25]、基于压缩感知的 CT(Compressive Tracking)[26]、DFT(Distribution Fields for Tracking)[27]和 MIL,二种基于深度学习的跟踪方法是 DLT(Deep Learning Tracking)和 GOTURN。六种方法中,前五种均是在线学习的方法,只有 GOTURN 是非在线学习的方法。

3.1 定性分析

首先,对比八个视频序列下不同跟踪算法的跟踪结果,并进行定性分析。图 8 给出了八个视频序列下六种对比算法与文中算法的部分跟踪效果图展示,并以不同颜色的矩形框对不同的算法进行区分,同时在场景左上角标注该视频图像的帧序号。

姿态变化:以视频序列 Human2 为参考,检测不同算法对姿态变化的适应能力。视频 Human2 的场景是行人在室内环境以不同姿态行走,如第 313 帧与第 1 061 帧分别代表行人侧身行走和正面侧弯腰

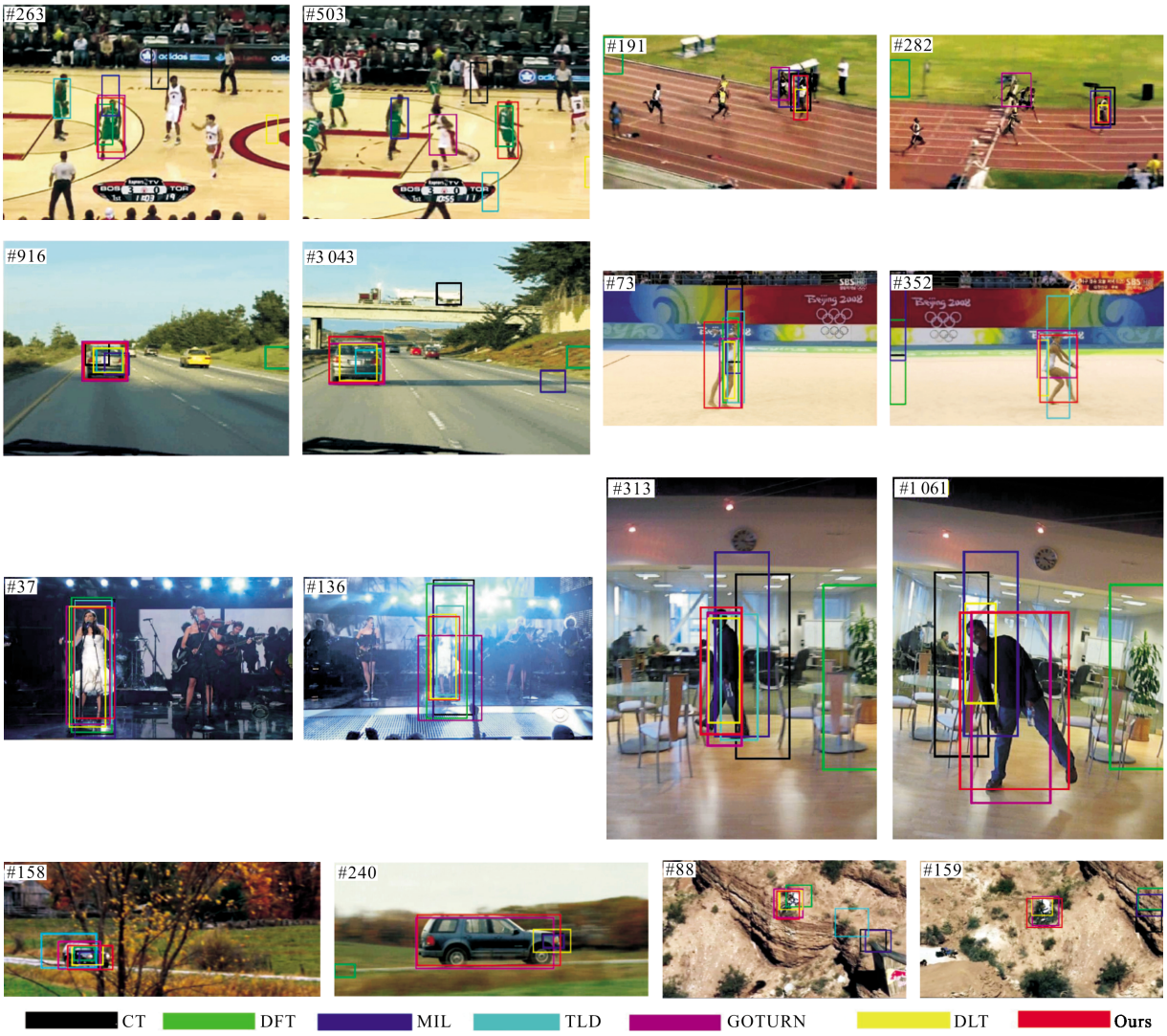


图 8 八个视频序列的跟踪结果定性比较(从左至右,从上到下分别是 Basketball, Bolt2, Car24, Gym, Singer1, Human2, CarScale, MountainBike)

Fig.8 Qualitative evaluation in the tracking results on eight challenging sequences (from left to right and top to bottom are Basketball, Bolt2, Car24, Gym, Singer1, Human2, CarScale, MountainBike)

两种姿态,对比不同算法的结果,文中提出的方法能准确地捕捉行人姿态的变化,而其他算法则出现了不同程度的偏移。

尺度变化:以视频序列 CarScale 为参考,检测不同算法对尺度变化的适应能力。视频 CarScale 的场景是汽车从远景进入到近景的过程,在尺度变化的过程中,GOTURN 和文中提出的方法随着其尺度变化而及时地变化,其他几种算法则不能及时准确地做出响应,以致在第 240 帧时,只有 GOTURN 和文中提出的方法能精准地跟踪目标。

旋转变化:以视频序列 Gym 为参考,检测不同算法对目标旋转变化的适应能力。Gym 存在着在同一个平面内的旋转变化和不在同一个平面内的旋转变化。如第 73 帧、第 353 帧所示,当运动员做出扭动和旋转动作时,文中方法相比于其他方法具有较好的适应能力,甚至于 TLD 和 DFT 已经出现了目标丢失。

光照变化:以视频序列 Singer1、Car24 为参考,检测不同算法对光照变化的适应能力。其中,视频序列 Singer1 的光照变化主要来自于后方探照灯的光照变化,如第 136 帧,当光照逐渐增强时,文中所提算法依旧能实现稳定的跟踪。视频序列 Car24 中的光照变化主要来自于在汽车行驶过程中附近遮挡物所形成阴影的交替,如 916 帧时,当目标车辆驶出阴影时,CT 和 MIL 发生了一定的位置偏移,而 TLD、

DLT、GOTURN 及文中所提方法无影响。这主要与文中方法采用的 HS 颜色空间对光照变化不敏感有直接关系。

背景杂波:以视频 Basketball、Bolt2、MountainBike 为参考，检测不同算法对背景杂波的适应能力。其中,Basketball、Bolt2 除了目标所处的背景比较复杂外,还有一定数量的类似目标,这对算法的跟踪性能提出了更高的要求。文中方法利用颜色特征可区分与目标外观相近而颜色特征不同的类似目标，而结合 HOG 特征,可区分颜色特征相近,空间边缘特征不同的类似目标，从而能对目标与类似目标进行较好地区分。

3.2 定量分析

这里主要采用中心位置误差和覆盖率[24]来进行算法的定量评价。其中,中心位置误差是指跟踪框与真实目标框的中心偏差，而覆盖率则是跟踪框与真实目标框的相交部分占其合并部分的比重。为了评价不同算法在整个视频序列上的跟踪性能，这里将采用平均中心位置偏差和平均覆盖率。

计算不同算法在八种场景下的平均中心位置偏差和平均覆盖率,结果如表 1、表 2 所示,其中,参与对比的六种算法的跟踪结果是根据官方开源程序测试所得。

表 1 平均中心位置偏差值(单位:像素)

Tab.1 Average centre location error (Unit: pixel)

	CT	DFT	TLD	MIL	DLT	GOTURN	Proposed
Basketball	121.6	18.0	-	103.8	267.6	62.1	**12.0**
Bolt2	9.9	276.7	-	**7.0**	8.5	40.3	10.9
Car24	86.7	165.4	-	82.4	**1.9**	3.0	3.5
CarScale	27.8	75.6	-	30.5	26.9	5.5	**3.6**
Singer1	18.9	18.8	10.4	19.4	**5.1**	14.2	13.0
Gym	134.6	104.3	14.3	125.3	16.9	20.4	**11.6**
Human2	75.1	181.6	-	90.3	27.8	**12.8**	16.0
MountainBike	212.3	154.8	-	217.2	13.1	**7.0**	9.7

备注:粗体标定代表该跟踪视频序列下单项指标最好的算法;下划线字体标定代表次好的算法,"_"表示算法在该视频序列跟踪失败。

如表 1、表 2 所示,在八个跟踪视频序列中,文中提出的方法在平均中心位置偏差指标上，三组最佳,两组次好,在平均覆盖率上,两组最佳,四组次好,其余也都接近次好。其中,TLD 因为自身算法的原因,在除视频序列 Singer1、Gym 之外,其余六组视频序列无完整跟踪结果。为进一步量化评价算法的性能,这里采用打分的方法对除 TLD 之外的六种算法的两项指标分别进行评价，打分规则是将两项指标根据性能从高到低进行排序,分为六个等级,依次打分为 6、5、4、3、2、1 分。首先，对每一种视频序列进行打分,最后对八个视频序列进行求和,作为其最终的打分结果,如图 9 所示。

表 2 平均覆盖率

Tab.2 Average overlap score

	CT	DFT	TLD	MIL	DLT	GOTURN	Proposed
Basketball	20.8%	**60.3%**	-	20.8%	6.7%	35.4%	50.9%
Bolt2	62.2%	1.0%	-	**69.1%**	40.5%	37.2%	49.6%
Car24	22.5%	7.5%	-	20.5%	**77.0%**	76.3%	77.3%
CarScale	41.2%	41.2%	-	34.2%	55.5%	72.3%	**85.3%**
Singer1	32.4%	**35.5%**	72.0%	32.1%	80.7%	52.7%	60.8%
Gym	4.1%	10.7%	41.6%	5.5%	28.7%	48.0%	59.6%
Human2	24.1%	10.2%	-	30.5%	52.5%	**74.6%**	71.5%
MountainBike	13.5%	29.4%	-	12.1%	55.4%	**73.7%**	60.4%

备注:粗体标定代表该跟踪视频序列下单项指标最好的算法;下划线字体标定代表次好的算法,"_"表示算法在该视频序列跟踪失败。

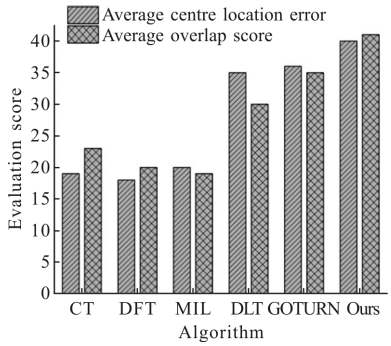


图 9 不同算法平均中心位置偏差与平均覆盖率之间的性能对比

Fig.9 Performance comparison in average centre location error and average overlap score among the algorithms

从图 9 中可以看出，文中提出的方法在平均中心位置偏差与平均覆盖率两项指标的打分上均优于其他对比算法,同时,可看出,所列举的几种算法中,基于深度学习的跟踪方法相比于传统方法更佳。

时效性上，在 Ubuntu 操作平台上,GPU 采用 Quadro K4000,采用 Python、C++混合编程,整个跟踪速度为 6 帧/s。根据 SSD 官方硬件配置，采用

TITAN X 显卡,SSD 的检测速度可达 58 帧/s[18],而文中采用的显卡,SSD 的检测速度大概在 10 帧/s,因此,若更新硬件环境,将进一步提升算法的实时性。但与 GOTURN 相比,TDLD 的实时性仍不及,主要原因在于 GOTURN 是一个为跟踪专门设计的深度神经网络,其实现了端到端的模型训练,应用于目标跟踪时,其模型计算量较低。而 TDLD 是结合基于深度学习的目标检测模型与计算机视觉方法设计的跟踪方法,其模型相对较复杂,耗时相对较多,但该设计随着基于深度学习的目标检测方法的速度升级,其实时性将进一步提升,同时,相比于 GOTURN,其模型可基于其他深度学习的目标检测方法,普适性更好。

综合上述结果,从跟踪效果上看,文中所提出的方法强于与所列举的六种具有代表性的跟踪算法,特别是,其对于物体的尺度变化和旋转变化的鲁棒性更好,可随着物体的姿态变化、尺度变化、旋转变化而及时准确的调整,充分展现了将基于深度学习的目标检测方法运用于跟踪的优势。

4 结 论

文中提出了一种基于深度学习物体检测的视觉跟踪方法 TDLD,属于一种非在线更新模板的跟踪方法,其前端基于深度学习的 SSD 模型,后端采用的是传统的计算机视觉方法,文中主要结合颜色特征和 HOG 特征对前端深度学习模型的检测结果进行目标选择。该方法成功将基于深度学习的物体检测迁移到目标跟踪上,为物体检测和目标跟踪实现融合提供了参考。经过实验表明,TDLD 相比于几种目前较成熟和经典的跟踪方法,在物体姿态变化、尺寸变化、旋转变化、光照变化、复杂背景杂波等影响因素具有更好的鲁棒性。但是,从跟踪的实时性来说,文中方法尚不能达到实时,在给定实验环境下,其跟踪速度大概在 6 帧/s,若升级硬件环境,其实时性将有效提高。

参考文献:

[1] Sivanantham S, Paul N N, Iyer R S. Object tracking algorithm implementation for security applications[J]. Far East Journal of Electronics and Communications, 2016, 16(1): 1-13.

[2] Kwak S, Cho M, Laptev I, et al. Unsupervised object discovery and tracking in video collections [C]//IEEE International Conference on Computer Vision, 2015: 3173-3181.

[3] Luo Haibo, Xu Lingyun, Hui Bin, et al. Status and prospect of target tracking based on deep learning [J]. Infrared and Laser Engineering, 2017, 46(5): 0502002. (in Chinese)

[4] Mei X, Ling H. Robust visual tracking using l1 minimization [C]//IEEE International Conference on Computer Vision, 2010: 1436-1443.

[5] Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2008, 77(1-3): 125-141.

[6] Wang N, Wang J, Yeung D Y. Online robust non-negative dictionary learning for visual tracking[C]//IEEE International Conference on Computer Vision, 2013: 657-664.

[7] Henriques J F, Rui C, Martins P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(3): 583-596.

[8] Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011, 33(8): 1619-1632.

[9] Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting [C]//British Machine Vision Conference, 2006: 47-56.

[10] Hare S, Saffari A, Torr P H S. Struck: structured output tracking with kernels [C]//IEEE International Conference on Computer Vision, 2011: 263-270.

[11] Wang N, Yeung D Y. Learning a deep compact image representation for visual tracking[C]//International Conference on Neural Information Processing Systems, 2013: 809-817.

[12] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016: 4293-4302.

[13] Held D, Thrun S, Savarese S. Learning to track at 100 FPS with deep regression networks [C]//European Conference on Computer Vision, 2016: 749-765.

[14] Ma C, Huang J B, Yang X, et al. Hierarchical convolutional features for visual tracking[C]//IEEE International Conference on Computer Vision, 2015: 3074-3082.

[15] Wang L, Liu T, Wang G, et al. Video tracking using learned hierarchical features [J]. IEEE Transactions on Image Processing, 2015, 24(4): 1424-1435.

[16] Wang N, Li S, Gupta A, et al. Transferring rich feature hierarchies for robust visual tracking [J]. Computer Science, 2015, arXiv: 1501.0458.

[17] Wang X, Hou Z, Yu W, et al. Robust visual tracking via multiscale deep sparse networks [J]. Optical Engineering, 2017, 56(4): 043107.

[18] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[19] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C]//European Conference on Computer Vision, 2016: 21-37.

[20] Cai Z, Fan Q, Feris R S, et al. A unified multi-scale deep convolutional neural network for fast object detection [C]// European Conference on Computer Vision, 2016: 354-370.

[21] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [C]//ICLR, 2015: arXiv: 1409.1556.

[22] Yin S F, Wang Y C, Cao L C, et al. Fast correlation matching based on fast fourier fransform and integral image [J]. Acta Photonia Sinica, 2010, 39 (12): 2246-2250. (in Chinese)

[23] Bal A, Alum M S. Automatic target tracking in FLIR image sequences[C]//SPIE, 2004, 5426: 30-36.

[24] Wu Y, Lim J, Yang M H. object tracking benchmark [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9): 1834-1848.

[25] 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, 238(6): 49-56.

[26] Zhang K, Zhang L, Yang M H. Real-time compressive tracking [C]//European Conference on Computer Vision, 2012: 864-877.

[27] Learnedmiller E, Sevillalara L. Distribution fields for tracking [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2012: 1910-1917.