

# 融合 YOLOV5+DSST+KCF 的目标检测跟踪算法在激光通信中的应用

李悦<sup>1</sup>, 马志勇<sup>1</sup>, 俞建杰<sup>2,3</sup>, 许博玮<sup>1</sup>, 郭佳杰<sup>2</sup>

(1. 湖州师范学院 工学院, 浙江 湖州 313000;

2. 浙江大学 湖州研究院, 浙江 湖州 313299;

3. 浙江大学 生物医学工程与仪器科学学院, 杭州 310027)

**摘要:** 在空间激光通信中, 两通信设备实现激光通信需要经过粗瞄准与精瞄准的过程, 而粗瞄准过程在大部分场景下需借助光学望远镜和信标激光等辅助实现, 实施不易; 针对在粗瞄准过程中如何快速、准确识别出激光通信设备, 并辅助完成激光通信, 提出将不同目标检测算法与长时跟踪 KCF 算法相结合; 通过对比发现, YOLOV5+DSST+KCF (YDK) 算法的平均帧率为 12 帧, 在帧率满足系统要求情况下, YDK 算法的响应峰值要比原 KCF 算法、HAAR+DSST+KCF (HDK) 算法分别提高了 43.5%、31.4%, 且有效解决了目标尺度变化问题。

**关键词:** 激光通信; YOLOV5 算法; KCF 算法; 目标检测; 长时追踪

## Application of Target Detection and Tracking Algorithm Combining YOLOV5+DSST+KCF in Laser Communication

LI Yue<sup>1</sup>, MA Zhiyong<sup>1</sup>, YU Jianjie<sup>2,3</sup>, XU Bowei<sup>1</sup>, WU Jiajie<sup>2</sup>

(1. School of Engineering, Huzhou University, Huzhou 313000, China;

2. Huzhou Research Institute of Zhejiang University, Huzhou 313299, China;

3. College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310027, China)

**Abstract:** In the filed of space laser communication, the realization of laser communication by two communication devices needs to go through the process of coarse and fine aiming, and the process of coarse aiming needs to be assisted by tools such as optical telescopes and beacon laser in most scenarios, which is not easy to implement. Aiming at how to quickly and accurately identify laser communication devices in the rough aiming process, and assist in the completion of laser communication, a combination of different target detection algorithms and long-term tracking KCF algorithm is proposed. Through comparison, it is found that the average frame rate of the YOLOV5+DSST+KCF (YDK) algorithm is 12 frames per second, when the frame rate meets the system requirements, the response peak value of the YDK algorithm is 43.5% and 31.4% higher than the original KCF algorithm and HAAR+DSST+KCF (HDK) algorithm, respectively, and effectively solved the problem of target scale change.

**Keywords:** laser communication; Yolov5 algorithm; KCF algorithm; target detect; long-term tracking

## 0 引言

在空间激光通信领域中, 两激光通信设备能够互相准确识别出对方所在位置并接收彼此发射的激光, 完成通信是实现激光通信的关键所在。并且空间激光通信具有传输速率高、通信容量大、功率消耗低、保密性强、安全性能好且结构轻便等优势, 使其应用领域变得越来越广泛。其中, 将飞机、高纬度浮空平台、卫星等应用系统获得的海

量数据实时下传到地面终端是其主要应用领域<sup>[1-2]</sup>, 从系统框架构成上考虑, 可将空间激光通信分为有信标和无信标两种结构, 本文提到的空间激光通信使用的是有信标光。空间激光通信中的 APT 系统 (捕获、对准、跟踪) 是实现高概率、快速捕获、高动态跟踪的前提与保障<sup>[3]</sup>, 通常使用的是复合控制系统机构, 可将其工作的过程划分为 4 个主要阶段: 分别是初始指向、快速捕获、粗精跟踪、动态

收稿日期: 2022-12-20; 修回日期: 2023-01-28。

作者简介: 李悦 (1997-), 女, 硕士研究生。

通讯作者: 马志勇 (1979-), 男, 博士, 副教授, 硕士生导师。

俞建杰 (1980-), 男, 博士, 副研究员, 硕士生导师。

引用格式: 李悦, 马志勇, 俞建杰, 等. 融合 YOLOV5+DSST+KCF 的目标检测跟踪算法在激光通信中的应用[J]. 计算机测量与控制, 2023, 31(10): 49-53, 75.

通信。捕获指在不确定区域内对目标进行判断和识别,是粗略的跟踪,大约在  $1 \sim 20^\circ$  的角度范围内发现目标,即“捕获目标”的意思。跟踪是指对发现的目标展开追踪,对准是指精准的跟踪。跟踪的精度一般与具体的使用场景有关,为了保证能够实现激光通信,通常要求捕获的概率要达到 95% 以上。传统的捕获、对准、跟踪这一过程往往需要人工辅助,操作起来比较麻烦,本文通过加入视觉目标检测与跟踪算法来取代捕获与粗跟踪这一过程。使用目标检测算法识别激光通信设备来取代激光通信中 APT 系统中的初始指向与快速捕获,正确识别到通信设备后利用步进电机辅助视觉目标跟踪算法对被跟踪目标展开跟踪,这一过程对应的是激光通信过程中的粗跟踪。接下来分别对通信设备的目标检测与目标跟踪部分展开叙述。

针对激光通信设备识别模块采用的方法是使用视觉目标检测方法,视觉目标检测是环境感知的重要内容<sup>[4]</sup>,但是目标检测算法中有一个困难是目标特征的提取。目标检测算法共分为传统目标检测算法和基于深度学习的目标检测算法两大类。传统的目标识别算法都是通过手工进行特征提取,通过这种方式提取到的特征一般适用于特征较为明显、背景单一的场景下,但是在大多数场景下,背景通常较为复杂、特征多变且存在干扰,通过手工进行特征提取来实现目标检测较为困难,且传统的目标检测算法识别准确率低、计算量大、有时还会出现多个正确识别效果。随着科学技术的发展以及对深度学习理论的不断研究,直到深度神经网络的出现,目标检测算法得到了进一步的提升。基于深度学习的目标检测算法借助于卷积神经网络的特征提取以及分类性能自动提取图像中的高维信息,对目标识别有较强的泛化能力<sup>[5]</sup>。目前基于深度学习的目标检测算法可分为两大类:一类是基于分类的“两步式”目标检测算法,代表性算法有 R-CNN<sup>[6]</sup>、Fast R-CNN<sup>[7]</sup>,另一类是基于回归的“一步式”目标检测算法,代表性算法有 YOLO<sup>[8-10]</sup>、SSD<sup>[11]</sup>,基于分类的目标检测算法的发展使得检测精度越来越高,但是检测速度上的提升空间很大,与基于分类的目标检测算法相比,基于回归的目标检测算法将目标检测问题转化为回归问题,真正意义上实现了端到端的实时目标检测,通过利用 CNN 卷积特征获取目标类别的概率与坐标信息,有着更快的检测速度,并且随着对 YOLO 算法的改进,检测精度也达到了惊人的效果,适用于实时检测应用场景。因此本文采用的是第二类方法中的 YOLOV5 算法。

由于两激光通信设备实现激光通信情况下,需要检测时间尽可能短、能够稳定有效地对另一激光通信设备展开跟踪。上述基于回归的目标检测算法虽然精度高,且也能够实现视觉上的跟踪效果,但是它是基于每一帧进行检测且当该帧没有检测到通信设备时,则不会出现目标框,无法做到连续稳定地对通信设备展开跟踪,且 YOLOV5 目标检测算法基于每一帧图像耗费的运算量大,容易造成资源

的浪费,而基于相关滤波的目标跟踪算法如 MOSSE<sup>[12]</sup>、Meanshift<sup>[13]</sup>、DSST、KCF<sup>[14]</sup>等等可以弥补这一缺陷。本文采用的是 KCF 跟踪算法,这种算法主要思想是将视频目标跟踪问题抽象为模板图像与候选区域的相似度匹配问题<sup>[15]</sup>,即通过将目标模型作为正样本,背景区域作为负样本,利用正负样本之间的差异训练一个目标分类器,在下一帧图像中,通过利用该分类器计算图像中像素点的响应值,找出最大响应值的位置点作为当前帧图像的目标位置。但是此类算法在跟踪过程中存在一定的弊端:1) 由于目标框大小不会随着目标大小的改变而改变,导致容易将背景信息引入到跟踪模型中;2) 目标跟踪的初始帧需要人为进行框选,即若第一帧中没有目标则需要切换到下一帧甚至需要移动相机模块将目标移动到拍摄画面中才可以进行后续视觉上的跟踪;3) 在跟踪过程中容易受到遮挡物的干扰;实施起来较为麻烦且容易出现跟踪漂移的现象。针对上述问题中提到的原 KCF 算法目标框大小是不随目标大小变化而发生变化的问题,本文加入 DSST<sup>[16]</sup>算法中的尺度估计滤波器算法,使得在跟踪过程中目标框大小随着目标大小改变而做出相应改变,从而避免跟踪过程中目标漂移导致跟踪失败现象的发生;针对目标跟踪第一帧需人为框选这一问题,前文中提到的 YOLOV5 目标检测算法刚好可以弥补这一不足,使用 YOLOV5 检测到被跟踪目标后,作为 KCF 跟踪算法的初始帧展开后续的跟踪;针对若拍摄画面中没有出现被跟踪目标问题,提出在检测过程中,进行区域扫描来辅助搜索被跟踪目标。

## 1 融合目标检测与跟踪算法的目标跟踪

### 1.1 KCF 目标跟踪算法

KCF 目标跟踪算法是一种基于核相关滤波的跟踪算法,属于判别式跟踪<sup>[17-18]</sup>方法中的一种。它通过在跟踪的初始帧中给出的目标框进行填充,获取一个填充框,在下一帧图像中将填充框进行循环移位获取循环矩阵,来收集大量的正负样本,将有目标的区域作为正样本,其它区域作为负样本,使用收集到的这些正负样本来训练目标检测器,在下一帧图像中,使用训练好的目标检测器检测预测位置是否是目标出现位置,并使用上一帧的填充框继续进行循环移位,将得到的样本框内的图像再次进行分类,作为正负样本,并选择这些样本框中响应最强的作为当前帧目标的填充框,使用当前帧获取的样本来更新目标检测分类器,在此过程中,利用循环矩阵在傅里叶空间可对角化的性质以及快速傅里叶变换对这些样本进行计算,提升其运算速度,用公式表示该过程为:首先设一组训练样本为  $(x_i, y_i)$ ,  $x_i$  为样本,  $y_i$  为样本标签,则回归函数为  $f(x_i) = \omega^T x_i$ ,  $\omega$  为列向量表示权重系数,可通过最小二乘法求解,用式 (1) 表示:

$$\min_{\omega} \sum_i (f(x_i) - y_i)^2 + \lambda \|\omega\|^2 \quad (1)$$

式中,  $\lambda$  为正则化系数,用于控制系统的结构复杂性。

通过引入非线性映射函数  $\varphi(x_i)$ , 可将低维非线性不可分问题映射到高维空间, 从而转化为线性求解, 如式 (2) 所示:

$$w = \sum_i a_i \varphi(x_i) \quad (2)$$

在核空间下将式 (2) 代入式 (1) 并进行离散傅里叶变化, 如式 (3) 所示:

$$\hat{\alpha} = \frac{\hat{y}}{K^{xx} + \lambda} \quad (3)$$

此时将  $w$  的优化问题转化为对  $\alpha$  求解问题, 式中:  $\hat{\alpha}$  为  $\alpha$  的离散傅里叶变换,  $K^{xx}$  为核函数矩阵的第一行元素。

得到非线性滤波器后, 对视频中的图像块进行检测, 从而求出目标所在的位置, 待检测图像块用  $z$  表示, 则在检测过程中的输出响应方程如式 (4) 所示:

$$f(z) = (K^z)^T \alpha \quad (4)$$

式中,  $K^z$  表示训练样本与候选样本之间的核矩阵, 它为非对称阵,  $K^z$  的每个元素定义为  $K_{ij}^z = K(P^{j-1}z, P^{i-1}x)$ , 其中  $P$  为置换矩阵, 作用是实现位移。 $\alpha$  是由各个系数  $\alpha^i$  组成的向量。

原 KCF 跟踪算法在对激光通信设备进行视觉上的跟踪过程的工作如图 1 所示, 其原理是: 首先, 人为框选出拍摄画面中的激光通信设备, 作为跟踪的初始帧, 并根据初始帧框选出来的激光通信设备的特征来初始化相关滤波器的模型; 其次, 利用该模板在后续帧的拍摄画面中找到响应最大的位置, 作为当前帧跟踪到的激光通信设备的位置; 最后, 利用每帧中确定的激光通信设备的位置来更新滤波器模型的参数。

### 1.2 YOLOV5 目标检测算法

YOLOV5 算法是 YOLO 系列算法中的一种, 用到的模型是深度神经网络, 以整张图像作为输入, 通过卷积、池化、上采样、张量拼接等操作, 最后输出图像中目标的类别以及图像中的位置信息。YOLOV5 在检测精度、运行效率、训练难易程度等方面相较于 YOLO 的其他版本具有优秀的表现, YOLOV5 又包括不同的网络模型, 其中, YOLOV5s 网络在不同阶段使用的卷积核数量都是最小的,

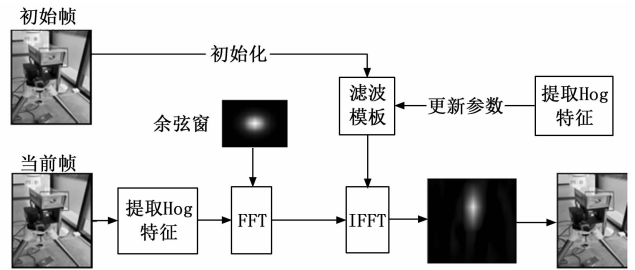


图 1 基于原 KCF 算法的通信设备跟踪框图

因此检测速度也是最快的, 其它 3 种网络是在 YOLOV5s 的基础上, 不断加深加宽网络, 但检测速度随着神经网络的加深、加宽而下降。综合考虑本文选用的是 yolov5s 模型。其框架结构如图 2 所示, 主要包括 3 个模块: 特征提取层、特征池化层以及目标预测层<sup>[19]</sup>。

### 1.3 DSST 尺度估计滤波器

DSST 跟踪算法中提出了一种三维尺度相关的滤波器<sup>[20]</sup>, 由两部分组成: 一维尺度滤波器和二维位置滤波器。一维尺度滤波器用于评估目标的尺度变换问题, 又称为尺度滤波器 (scale filter), 二维滤波器用于评估目标的位置变换问题, 又称为位置滤波器 (translation filter)。这两个滤波器是相互独立的, 因此, 其中的尺度滤波器具有可移植性, 可与其它具有尺度不变性的跟踪算法相结合, 尺度估计滤波器可分为训练与检测两个过程, 在训练过程中, 其滤波器的大小为  $M * N * S$ , 其中  $M * N$  为目标所在区域的图像块的高和宽,  $S$  为以图像块正中心位置开始, 截取不同尺度图片的数量,  $f$  为不同尺度上的图像块的特征描述子,  $g$  表示大小为  $1 * S$  的高斯函数构建的三维滤波响应输出, 通过将  $g, f$  代入式 (5) 可获取滤波器的模板  $h$ 。

$$H^l = \frac{\overline{GF^l}}{\sum_{k=1}^d \overline{F^k F^k} + \lambda}, \quad l = 1, \dots, d \quad (5)$$

式中,  $H, G, F$  为对应小写字母的傅里叶变换,  $l$  表示  $1 \sim d$  维其中的一种,  $\overline{\phantom{x}}$  表示负数的共轭,  $\lambda$  表示正则项。

在检测过程中, 在新一帧的图像中, 已经确定目标所

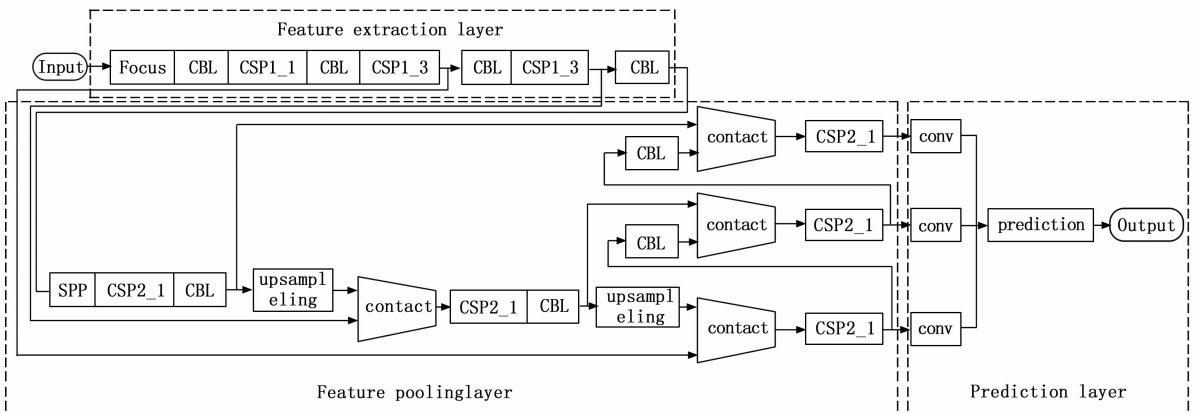


图 2 YOLOV5 网络框架图

在位置情况下,以目标位置为中心,截取  $S$  个不同尺度的图像块,并分别求出每个图像块的特征描述子,组成一个新的特征金字塔  $z$ ,与训练过程中迭代更新过的模板  $h$ ,通过图 3 所示的方式分别获取它们的响应输出  $g$ ,并根据响应输出  $g$  中最大值来确定新一帧图像中目标的尺度信息。

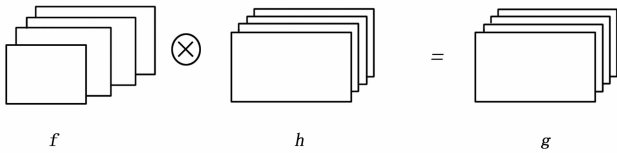


图 3 DSST 尺度估计滤波器原理图

#### 1.4 YDK 检测跟踪

KCF 跟踪算法的优点是不需要事先对大量目标样本进行训练,只需要对初始帧中人为框选的目标位置以及后续跟踪过程中不断更新的目标位置进行训练,得到一个目标检测器,使用该目标检测器确定后续帧中目标的位置,进而实现实时跟踪,但当目标的大小发生变化,导致引入过多背景信息,或者出现遮挡时间过长情况下,则会最终导致跟踪失败。

YOLOV5 目标检测算法的优点在于在对目标进行检测之前,通过神经网络对大量的目标样本进行训练,使用训练好的模型对目标进行检测,由于该算法是通过提取到的目标的特征信息在图像中通过滑动框的方法对多尺度的目标进行检测,一定程度上避免目标大小发生变化时或者出现过长时间遮挡后再次出现无法检测出目标状况的发生;但是 YOLOV5 目标检测算法对训练样本有很高的要求,如果待检测画面中的目标与背景信息与训练样本中的差异较大,则不能稳定有效的检测出目标位置,且 YOLOV5 检测算法的帧率要低于 KCF 跟踪算法。

针对两种算法的不足与优点,提出将两种算法相结合。首先,在实时拍摄画面中,使用 YOLOV5 算法进行视觉上的目标检测,确定被跟踪激光通信设备在当前帧的位置后,作为 KCF 跟踪算法的初始帧展开后续的跟踪,并在 KCF 跟踪算法中加入 DSST 算法中的尺度估计滤波器算法来对通信设备进行实时检测与追踪,具体流程如图 4 所示。

由图 4 可知,YDK 算法的大致流程是:开启程序,控制步进电机进行区域扫描,即在水平方向与垂直方向各放一个步进电机,发送指令控制这两个步进电机交替运行实现矩形螺旋扫描,同时,使用 YOLOV5 算法在实时拍摄画面中识别出被跟踪激光通信设备所在位置信息。当检测到被跟踪激光通信设备的置信度达到设定阈值时,则说明检测成功,检测成功后,电机停止扫描,并将当前帧被跟踪激光通信设备的坐标信息传入 KCF 跟踪算法,作为初始帧中的被跟踪对象,从而展开后续跟踪;在后续帧的跟踪过程中,确定每帧图像中的目标位之后,使用 DSST 算法中的尺度估计滤波器算法确定每一帧中目标框大小,并根据

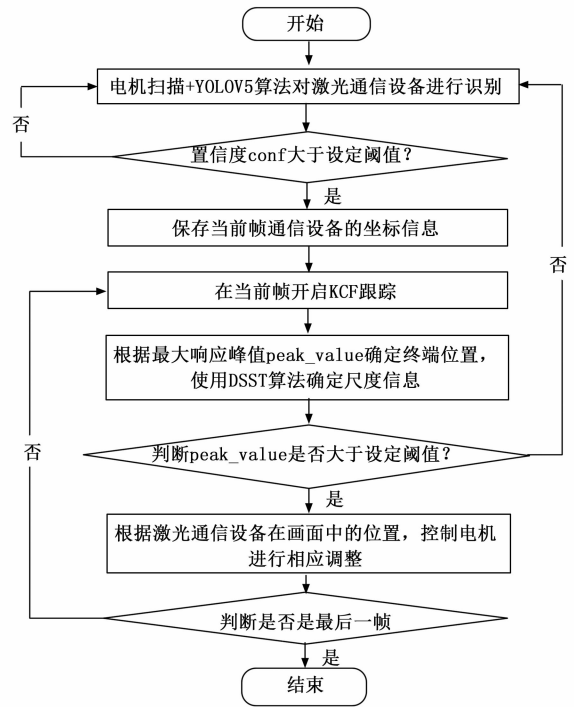


图 4 YDK 算法流程图

每一帧的响应峰值来判断跟踪是否正确,若跟踪正确,则发送指令控制控制步进电机运行,从而对拍摄画面中目标的位置做出相应的调整,若跟踪失败则调用 YOLOV5 算法重新进行 YOLOV5 目标检测,直到跟踪结束,停止程序运行。

## 2 实验与分析

### 2.1 实验平台环境

激光通信设备图片采集主要使用海康威视的 DS-2CD3T86FWDV2-I3S 相机,拍摄范围为 5 m 内,焦距为 2.8 mm;训练网络模型使用的硬件平台为 Core i7-12700F @2.1 GHz 处理器,软件平台为 Win11, CUDA10.2, CUDNN8.0, Pycharm 2021, Pytorch1.8.0, Torchvision 0.9.0, Opencv4.4.46 (CUDA 编译);实时拍摄测试硬件平台使用的是 Jetson Xavier NX,开发板,软件平台是 Jetpack 5.0.2, python 3.8.10, CUDA11.4, CUDN8.4.3, Pytorch 1.12.0, Torchvision0.13.0, Opencv4.4.45 (非 CUDA 编译)。系统硬件搭建如图 5 所示。

根据图 5 可知,单个激光通信设备用到的模块有三脚架、相机、步进电机、电机驱动板,电机驱动板供电电源、转台,开发板。

### 2.2 激光通信设备图像采集

分别在不同场景、不同距离下采集激光通信设备图片作为训练数据集,共收集数据集 3 811 张图片,使用 Labelimg 工具对数据集每张图片中的激光通信设备进行标注,标注完成后,将其中 3 028 张作为训练集,783 张作为测试集,通过使用 YOLOV5 自带的网络训练模型对这些样本集进行训练,为了缩短网络的训练时间以及达到较好的检测

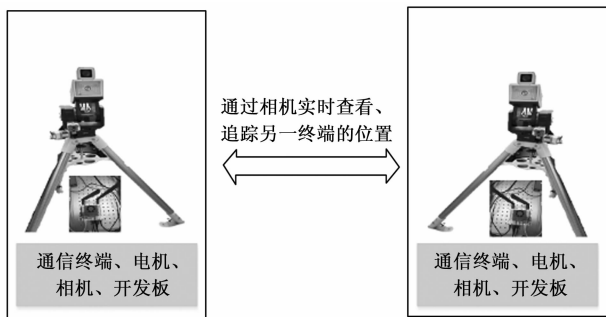


图 5 系统总体设计

精度, 采用的是 yolov5s.pt 预训练权重。数据训练完成后, 会在项目文件夹下产生一个 best.pt 和 last.pt 文件, 前者为训练最好的权重文件, 后者为最后一轮的权重文件。本文选用的是训练效果最好的权重文件, 将训练好的模型分别在拍摄完成的一段视频, 以及实时拍摄视频上进行测试。数据集的部分样本如图 6 所示。

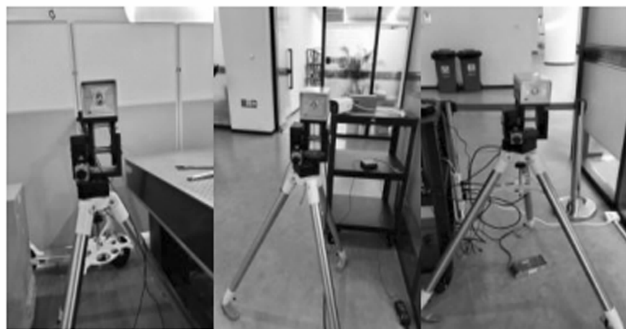
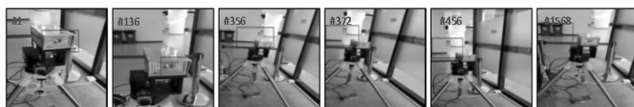


图 6 数据集样本

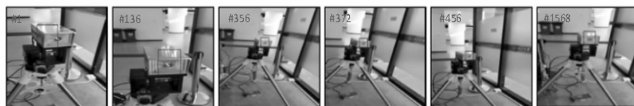
### 2.3 激光通信设备跟踪结果分析

在视频上进行测试时, 是使用手机拍摄一段视频, 其中视频中要体现出激光通信设备由远及近, 以及不同角度转变; 将原 KCF、YDK、HDK 算法在这段视频上进行测试, 观察它们的跟踪效果, 该过程不涉及电机模块的控制, 如图 7 所示。在进行实时测试的过程中, 则是在检测过程中, 控制步进电机运行从而带动相机模块进行区域扫描; 在跟踪过程中, 控制步进电机运行从而调整被跟踪激光通信设备在拍摄画面中的位置信息。

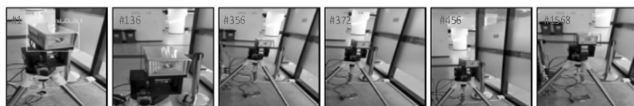
根据图 7 可知, 图 (a) 第一帧为人为框选的结果, 在第 136 帧时, 目标框偏离了被跟踪激光通信设备所在位置, 在第 356 帧时引入了过多的背景信息, 导致后续帧逐渐完全偏离激光通信设备所在位置, 致使无法辅助两激光通信设备实现激光通信; 图 (b) 第一帧为 Haar 级联分类检测效果, 只对局部进行框选导致后续跟踪的帧中只是对通信设备的局部进行跟踪, 但可以发现跟踪的效果要优于图 (a) 跟踪效果; 图 (c) 第一帧为使用 YOLOV5 算法实现对被跟踪激光通信设备自动检测的效果, 其置信度达到了 0.91, 且目标框与被跟踪激光通信设备所在位置重合, 最



(a) A组为原KCF算法跟踪效果图



(b) B组为HDK算法跟踪效果图



(c) C组为YDK算法跟踪效果

图 7 跟踪效果图

大程度上减少了背景引入量, 并且在图 (c) 后续的跟踪画面中, 目标框完美的框选出了被跟踪激光通信设备所在位置, 这是由于 DSST 算法中尺度估计滤波器算法的加入, 使得每帧目标框的大小在正确框选出被跟踪激光通信设备所在位置情况下, 随着目标的大小做出相应的调整; 通过这三组跟踪效果对比可发现 YDK 算法的跟踪效果要明显优于图 (a) 与图 (b) 的跟踪效果。原 KCF、HDK、YDK 算法在跟踪过程中的平均响应峰值如表 1 所示。

表 1 不同跟踪算法响应峰值与帧率

测试平台	GPU		
	跟踪算法	响应峰值(均值)	帧率
视频	原 KCF	0.432	38.800
	HDK	0.472	11.893
	YDK	0.620	12.041
实时	YDK	0.710	13.338

根据表 1 可知, 在一段视频上进行测试时, 原 KCF 算法的平均响应峰值为 0.432, 平均帧率为 39 帧; HDK 算法的平均响应峰值为 0.472, 比原 KCF 算法提高了 9.26%, 平均帧率为 12 帧, 比原 KCF 算法下降了 69.2%; 通过结合图 7 中不同算法的跟踪效果展示与表 1 中不同跟踪算法的响应峰值以及帧率的数据发现: HDK 算法虽然有加入 DSST 尺度自适应算法, 但是由于在跟踪的初始帧中, 识别到被跟踪激光通信设备后, 框选的是被跟踪激光通信设备的局部信息, 导致在后续帧中一直是针对激光通信设备的局部进行的, 随着被跟踪目标渐远有丢失目标的风险; YDK 算法的平均响应峰值为 0.620, 平均帧率为 12 帧, 平均响应峰值比原 KCF 算法提高了 43.5%, 比 HDK 算法提高了 31.4%, 平均帧率比原 KCF 算法下降了 68.97%, 比 HDK 算法提高了 1.2%; 随着帧数的增加, YDK 算法比 HDK 算法能够更好的检测出通信设备所在位置, 并且目标

(下转第 75 页)