

# 基于四摄像头的全景目标检测系统<sup>\*</sup>

尹睿昀 仝傲 姚新阳 袁观妙 王佳茹  
(西安工程大学电子信息学院 西安 710048)

**摘要:**目前基于图像的检测系统多以单摄像头为主,但是单幅图像的视野有限,往往存在检测盲区,为此,设计了一种基于四摄像头的全景目标检测系统。首先搭建了由3B+树莓派、摄像头、LCD液晶屏、移动电源联合组成的系统硬件,基于QT设计了可进行检测结果显示与参数设置的用户交互软件界面;系统通过四部摄像头实现对周边360°场景进行取景,然后对采集到的图像进行预处理与拼接,最后通过基于EfficientDet深度学习模型检测器进行目标检测与结果显示。实验结果表明,该系统可便携移动,能实现360°全景取景与目标检测功能且目标检测准确率达94%以上。

**关键词:**全景检测系统;图像拼接;目标检测;深度学习

**中图分类号:** TP399;TN98 **文献标识码:** A **国家标准学科分类代码:** 520.604

## Panoramic target detection system based on quad camera

Yin Ruiyun Tong Ao Yao Xinyang Yuan Guanmiao Wang Jiuru  
(Electronics and Information College, Xi'an Polytechnic University, Xi'an 710048, China)

**Abstract:** At present, most of the image-based detection systems are mainly single-camera, but the field of view of a single image is limited, and there are often blind spots, so a panoramic object detection system based on four cameras is designed in this paper. Firstly, the system hardware composed of 3B+ Raspberry Pi, camera, LCD screen and mobile power supply was built, and a user interactive software interface for detection result display and parameter setting was designed based on QT. The system uses four cameras to frame the surrounding 360-degree scene, then preprocess and stitch the collected images, and finally perform object detection and result display through the EfficientDet-based deep learning model detector. Experimental results show that the system can be portable and mobile, and can realize 360-degree panoramic framing and target detection functions, and the accuracy of target detection is more than 94%.

**Keywords:** panoramic detection system; image processing; object detection; deep learning

### 0 引言

近年来,在自动化和智能化技术快速发展的背景下,目标检测技术得到了充分关注。目标检测<sup>[1]</sup>是一种与计算机视觉和图像处理相关的计算机技术,处理的是数字图像或者视频中特定类别的对象<sup>[2]</sup>。该技术目前在,工业监控、行人跟踪、无人驾驶<sup>[3]</sup>等领域得到了广泛应用,如王敏等<sup>[4]</sup>于2019年研究制备高分辨率遥感影像的舰船目标样本,采用成像过程模型辅助的样本参数空间统计插值技术,显著提高了目标检测的准确率。梁继然等<sup>[5]</sup>结合注意力机制和密集连接网络以保留更多的车辆特征信息,实现

复杂交通场景下的车辆检测任务。朱焕宇等<sup>[6]</sup>于2023年提出一种基于YOLOv5神经网络的检测方法,并引入交叉注意力机制和深度可分离卷积机制来增强特征表达和降低参数量,提高了轮胎胎体钢丝帘线缺陷的检测性能。随着研究的深入,各类检测的性能逐步提升,但是对检测图像和数据处理器的要求也越来越高<sup>[7]</sup>。而赵羽等<sup>[8]</sup>提出了一种部署在边缘端的视频处理方法,利用轻量级的神经网络和联邦学习分场景训练模型。Yang等<sup>[9]</sup>提出了基于检测的跟踪方法,适用于边缘端高分辨率视频的分析。Gao等<sup>[10]</sup>提出了一种针对云边缘视频流的智能视频处理架构。但是,这些研究只针对对单个摄像头采集的视频进

收稿日期:2024-05-18

<sup>\*</sup> 基金项目:西安工程大学创新创业训练项目(S202310709036)资助

行分析,而未涉及多摄像头的数据处理。由于在图像采集时使用的为单摄像头,而单幅图像的视野有限,以及遮挡的问题,会导致最终图像存在视野盲区<sup>[11]</sup>。在目前实际应用场景中,因单摄像头无法大范围地跟踪目标,多摄像头多目标跟踪<sup>[12-13]</sup>在应用中逐渐得到关注。

本文设计的基于图像拼接的全景检测系统。以树莓派3B+作为主控器,通过搭载4个USB摄像头,实现实时捕获周围环境的信息,能有效避免视野盲区问题。并且通过对图像帧进行处理,降低数据处理压力。最后通过基于

深度学习的目标检测算法<sup>[14-15]</sup>对处理后的图像进行准确识别。本文会对系统的软硬件设计方法进行介绍,并对最终实验测试结果进行分析总结。

## 1 系统总体设计

本文采用了基于树莓派的图像处理模块和摄像头组合,结合 Python 语言、OpenCV 库和 EfficientDet 等技术,实现了实时的图像处理与拼接,进而在树莓派上完成了全景目标检测功能。系统的总体设计框图如图 1 所示。

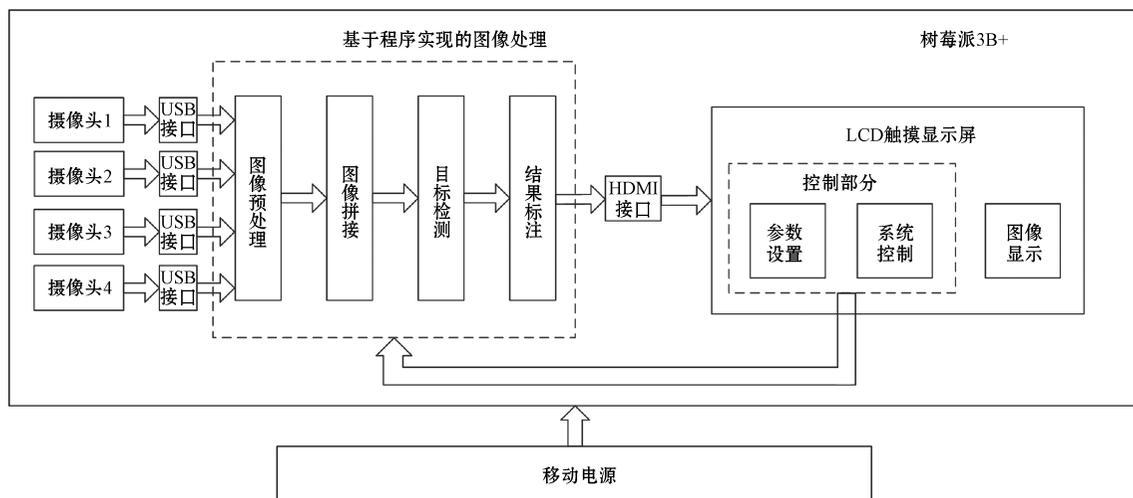


图 1 基于四摄像头的全景检测系统总体设计框图

Fig. 1 Block diagram of the overall design of a panoramic inspection system based on four cameras

系统首先通过4部摄像头进行图像采集,能解决单摄像头采集图像视野有限,以及遮挡的问题。对采集到的图像进行处理,转变为具有 $600 \times 300$ 的像素分辨率的图像,优化后续系统处理速度。其次系统运用 OpenCV 和 PIL 库,对采集到的图像进行拼接处理。拼接过程中,系统根据设定好的排列格式,保证图像质量的前提下对全方位的图像进行拼接。接着系统采用 EfficientDet-lite0 轻量化网络对拼接后的图像进行目标检测并对检测到的目标进行框图标注。保证使用轻量化系统,也能有良好的准确率。最后为系统配备可视化界面,用户通过这一界面可以调整系统参数、控制运行模式,以及查看摄像头捕获的原始图像和经过处理并标识出检测结果的拼接图像。

## 2 硬件设计

本文提出的基于图像拼接的全景巡检系统的硬件部分包括树莓派3B+、USB摄像头、和LCD触摸显示屏,其硬件连接设计如图2所示。

系统采用树莓派3B+作为开发平台。树莓派作为一种搭载 Linux 内核的单片微型计算机,被广泛应用于智能需求场景<sup>[16]</sup>。通过体积更小、功耗和成本更低的树莓派微型主机完成算法的移植,非常适用于移动设备和现场

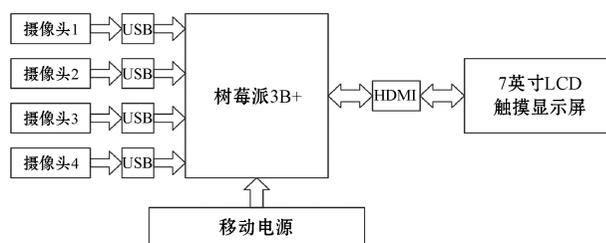


图 2 硬件方案设计框图

Fig. 2 Hardware solution design block diagram

实时监测等场景<sup>[17]</sup>。

系统采用 10 000 mAh 的移动电源。此电源能够使整个系统工作约 3 h。

系统配置了 7 in 的 LCD 触摸显示屏作为其显示和操作设备。为了稳定信号传输,选用 HDMI 接口作为数据交互通道。

系统使用 4 部分辨率为  $1\,280 \times 720$  的无畸变 USB 摄像头。通过标准 USB 接口,摄像头与树莓派之间实现数据传输,确保实时视频流的流畅传输与处理。系统硬件实物如图 3 所示。

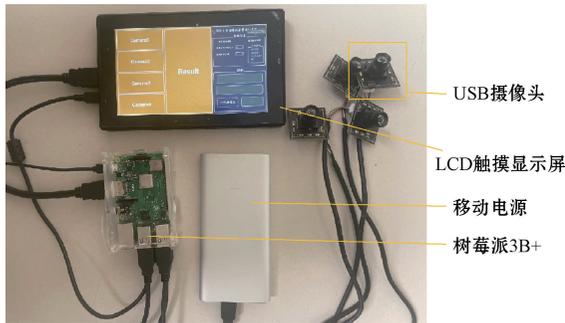


图3 系统硬件实物

Fig. 3 Physical diagram of system hardware

### 3 软件设计

软件部分使用 Python 语言进行开发,系统所需的软件环境为适配树莓派的 Linux 操作系统,安装 OpenCV、TensorFlow Lite、PyQt5 等依赖库。

#### 3.1 UI 界面设计

系统采用 Python 与 Qt 相结合来设计 UI 界面。根据本次开发项目的需求,围绕图像显示区域、参数设置区域,系统控制区域 3 个功能模块设计了一套专属的系统 UI 界面。设计的界面如图 4 所示。



图4 系统 UI 界面

Fig. 4 System UI

此界面实现的功能主要包含显示和控制两个部分,通过线程的调度可以显示 4 个摄像头拍摄到的画面,并在运行拼接检测程序之后显示检测结果。此外,用户可以通过设置参数来调整程序的功能以及系统的工作模式。

使用 QtDesigner 设计好界面,将生成的 ui 文件转为 python 文件后,通过代码绑定信号与槽。具体实现代码如下:

```
self.start.clicked.connect(self.start_camera) # 启动 4 个线程
```

```
self.state1.clicked.connect(self.stiching_detect) # 单次拍摄
```

#### 3.2 软件主流程

系统主要实现如下 4 个功能:用户通过 UI 界面设置

参数,以及按键操控实现对系统的调度;图像采集部分通过调用 4 个摄像头来捕获图像,并将采集到的图像传输给图像拼接模块以及显示在 UI 界面;图像拼接部分接收到摄像头采集的图像后,通过 Pillow 库生成全景图像,根据使用者需求决定是否保存所生成的图像;目标检测部分通过 EfficientDet\_lite 网络对全景图像进行目标检测,并对最终的检测结果进行框图标注,最后将结果返回给 UI 界面进行显示,如图 5 所示。

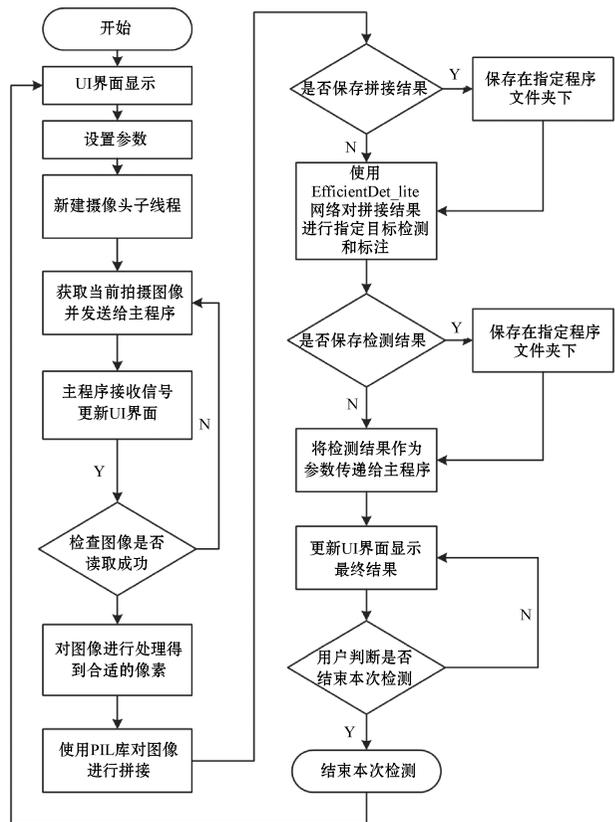


图5 系统软件总流程

Fig. 5 Total flow chart of the system software

#### 3.3 图像采集

采集程序分为摄像头初始化和图像捕获与显示两部分,首先使用 OpenCV 库的 VideoCapture 函数初始化摄像头,其次循环调用 VideoCapture 对象的 read 方法获取帧图像,最后使用 OpenCV 的 imshow 函数在 UI 界面的 Label 控件中显示图像。其运行流程如图 6 所示。

在 run 函数中,首先使用 cv2.VideoCapture 初始化摄像头,并通过 set 方法设置摄像头的属性,如视频流格式和帧大小。然后进入一个循环,不断地使用 read 方法从摄像头读取图像帧。如果读取成功,就通过 emit 函数发射当前帧图像。如果在读取过程中遇到错误(例如摄像头未连接或帧为空),则打印错误信息并退出循环。最后,确保在线程结束时调用 cap.release() 来释放摄像资源。通过 run 函数调用摄像头的部分代码如下:

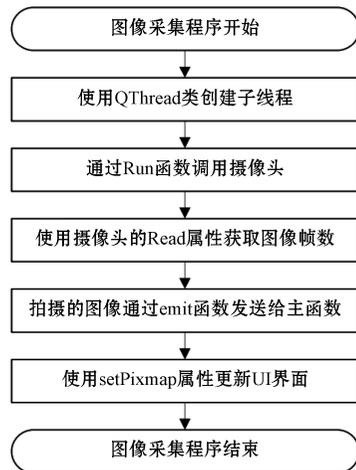


图6 图像采集流程

Fig. 6 Image acquisition flowchart

```
cap = cv2.VideoCapture(0) # 创建对象
ret, frame1 = cap.read() # 从摄像头读取一帧图像
self.sinOut.emit(frame1) # 通过自定义信号发射当前帧图像
```

更新界面的部分代码如下：

```
self.camera1.setPixmap(QtGui.QPixmap.fromImage(coverqt_img)) # 更新显示窗口
```

### 3.4 图像拼接

图像拼接部分主要对采集后的图像进行拼接处理，利用了 OpenCV、PIL 和 NumPy 等库对采集到的图像进行，如图 7 所示。在拼接过程中，每个图像首先会从 BGR 格式转换为 paste 函数能处理的 RGB 格式，然后使用 resize 函数调整至所需的像素大小。之后创建一个新的白色背景图像，将调整后的图片按照需求使用 paste 函数粘贴到新图像的正确位置，最后将结果图格式从 RGB 格式转换为方便后续呈现的 BGR 格式，并返回结果图。

拼接相关的核心操作写在 stitching\_image 这个函数中。首先主函数通过发射停止信号来告知所有摄像头线程结束它们的图像捕获工作。当所有图像帧都被成功读取后，就会调用 stitch\_image 函数来执行实际的图像拼接操作。拼接完成后，如果用户界面上的保存选项被选中，方法会将结果图像保存到磁盘上。最后，方法返回拼接后的图像，这个图像可以用于后续的处理步骤，比如目标检测或显示给用户。主函数调控的部分代码如下：

```
self.signal.stop_signal.emit() # 发射停止信号，让所有摄像头线程停止捕捉新的图像帧
```

```
result_img = camera.stitch_image(imag1, imag2, imag3, imag4) # 调用 stitch_image 函数，将摄像头捕获的图像帧进行拼接
```

```
return result_img # 返回拼接后的图像，供后续图像检测使用
```

stiching\_image 函数中的部分代码如下：

```
image4 = Image.fromarray(cv2.cvtColor(img4, cv2.COLOR_BGR2RGB)) # 读取图像并转换格式
result_img.paste(image1, (0, 0)) # 粘贴处理后的图像
```

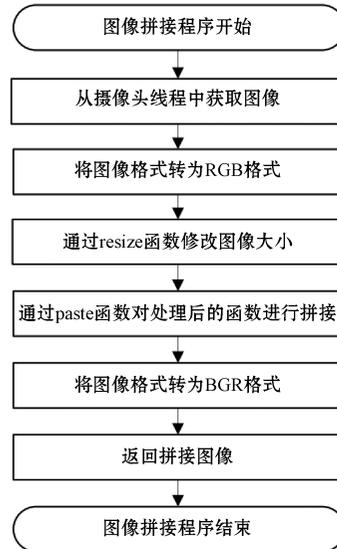


图7 图像拼接流程

Fig. 7 Image stitching flowchart

### 3.5 目标检测

目标检测部分通过创建检测器，对拼接后得到的图像按照设置参数进行目标检测，并将检测结果展示在显示界面上，其过程如图 8 所示。

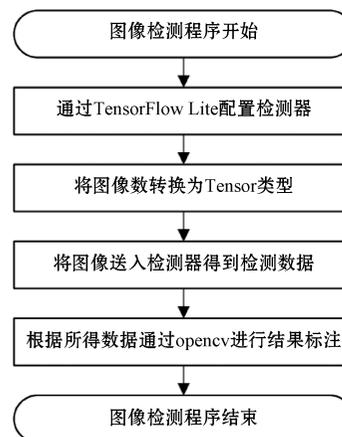


图8 目标检测流程

Fig. 8 Object detection flowchart

该部分使用了深度学习模型 EfficientDet<sup>[18]</sup> 和 Tensor Flow\_lite 架构。EfficientDet 通过集成 EfficientNet 的主干网络，级联双向特征金字塔网络<sup>[19]</sup>，再通过平衡网络的深度、宽度和分辨率，实现了跨尺度特征的高效融合<sup>[20]</sup>。其结构模型如图 9 所示。

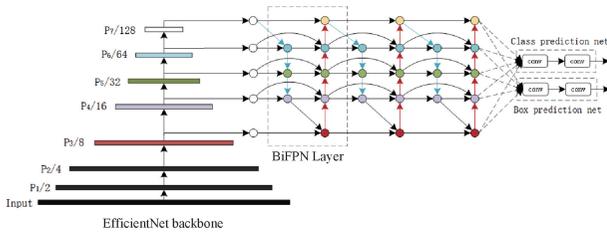


图9 EfficientDet 模型结构

Fig.9 EfficientDet model structure

EfficientDet-Lite0 作为该系列中的一个成员,是经过特别优化的,牺牲了一部分准确性以换取更小的模型尺寸和更快的推理速度。

本文首先从 tflite\_support 工具包中导入 core、processor 和 vision 模块。定义 core 模块的 BaseOptions 类的参数来初始化目标检测模型。定义 processor 模块的最大结果、置信度阈值和目标种类 3 个参数。使用 vision 模块将先前生成的对象合并为“options”对象,最后根据“options”对象建立检测器对象“detector”。

为了更好地处理数据,TensorFlow 架构引入了 Tensor 类型数据,使用时需要将图像数据转换为 Tensor 类型。将图像预处理,即将拼接后的图像转换为模型所需的输入格式,得到“TensorImage”对象后,将此对象送入检测器检测,得到包含检测目标大小、种类等信息的数据,根据这些数据使用 OpenCV 库将检测结果标注在图像上。

配置检测器的代码如下:

```
detection_options = processor. DetectionOptions
(max_results=result_num, score_thresh=0.3, category_name_allowlist=category) # 初始化检测类型,设定了返回结果数,最小置信阈值以及允许检测类别
```

```
detector = vision. ObjectDetector. create_from_options(options) # 建立检测器
```

对图像进行检测的部分代码如下:

```
detection_result = detector. detect(input_tensor) # 将数据送入检测器进行目标检测
```

```
image=utils. visualize(image,detection_result) # 对检测结果进行标注。
```

## 4 系统测试与结果分析

### 4.1 系统功能测试

为了全面评估本文系统性能,对系统进行总体测试,查看图像的全景拼接以及目标检测的效果。通过多次实验并设置不同检测环境,不同检测目标以及不同目标数量的多个对照组,测试本系统的实际效果。具体包括将检测目标数量设为 1 个,并指定目标类别为“person”;将检测目标数量增至 5 个,同时不特别指定目标类别;设定检测目标为 2 个,同样不指定目标类别。当不指定目标时系统默认检测所有目标,并根据置信度来显示目标结果,经过

这一系列的测试,所得结果如图 10 所示。



(a) 楼道内环境

(a) The environment in the corridor



(b) 室内桌面环境

(b) Indoor desktop environment



(c) 室外环境

(c) Outdoor environment

图 10 不同场景下的图像拼接与检测结果

Fig.10 Image stitching and detection results in different scenarios

### 4.2 系统检测准确率测试

为了验证系统方法检测的准确性和可重复性,对系统检测能力进行测试,分别为简单背景下的单目标检测精度测试、复杂背景下的单目标检测精度测试。根据可接受的误差范围将每组测试数据设计为 100 次检测结果构成,主要检测目标对象包括人、杯子、鼠标和电脑等常见物品,这些结果覆盖了多种角度和环境下的观测以模拟实际应用中的多样性。测试数据如表 1 所示。

表 1 针对不同样本的测试结果

Table 1 Results for different samples

样本	人	杯子	电脑	鼠标
样本数	50	50	50	50
正确识别数	49	48	47	49
准确率/%	98	96	94	98

由检测结果可知,系统能够稳定获取多个摄像头的视频流并将多个视频流截取并拼接成全景图像,拼接效果好;目标检测部分能够准确检测全景图像中的目标,并输

出检测结果。在部分复杂场景下,系统的检测准确率仍有提升空间。通过实验结果可知最后准确率能达到94%以上,在保证系统整体轻量化的前提下,对常见各类物品进行检测仍有较高准确率,达到设计预期目的。但是,由于树莓派性能有限,当处理大量数据时,系统可能会出现一定的延迟或卡顿现象。

### 4.3 其他检测算法对比

为了验证本文所选网络的优势,本文选择与目标检测领域比较具有代表性的几种算法 SSD、YOLOv2 和 YOLOv3 进行比较,结果如表 2 所示。本文所使用的 EfficientDet 在 COCO 数据集下的表现不仅保证了高精度,还兼顾了模型的轻量化,在二者之间保持了良好的平衡。

表 2 针对不同训练集下各网络的性能比较

算法	VOC07 (mAP@0.5)	VOC12 (mAP@0.5)	COCO (mAP@0.5:0.95)	来源
SSD	76.8 (07+12)	74.9 (07++12)	31.2	ECCV'16 <sup>[21]</sup>
YOLO v2	78.6 (07+12)	73.4 (07++12)	—	CVPR'16 <sup>[22]</sup>
YOLO v3	—	—	33.0	arXiv'18 <sup>[23]</sup>
EfficientDet	—	—	52.6	CVPR'20 <sup>[24]</sup>

## 5 结 论

本文设计的基于图像拼接的全景目标检测系统,以树莓派 3B+ 和多摄像头搭建的硬件平台为载体,结合高效的图像拼接与目标检测算法,实现对目标物体的图像采集、图像拼接以及目标检测。因其轻量化与多摄像头的设计,使系统可便携移动的同时还能对周围环境进行全方位监控和目标检测,有效处理视觉盲区问题,且综合准确率能够达到94%以上。

目前,此系统能在无人机连续图像拍摄监控、自动化生产线进行自动检测和分类、机器人感知周围环境进行判断有广泛应用前景。

### 参 考 文 献

- [1] 尹宏鹏,陈波,柴毅,等. 基于视觉的目标检测与跟踪综述[J]. 自动化学报,2016,42(10):1466-1489.  
YIN H P, CHEN B, CHAI Y, et al. Vision-based object detection and tracking: A review [J]. Acta Automatica Sinica, 2016, 42(10): 1466-1489.
- [2] 史朋飞,韩松,倪建军,等. 结合数据增强和改进 YOLOv4 的水下目标检测算法[J]. 电子测量与仪器学报,2022,36(3):113-121.  
SHI P F, HAN S, NI J J, et al. Underwater object detection algorithm combining datae-nhancement and improved YOLOv4 [J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(3): 113-121.
- [3] 方路平,何杭江,周国民. 目标检测算法研究综述[J]. 计算机工程与应用,2018,54(13):11-18,33.  
FANG L P, HE H J, ZHOU G M. Research overview of object detection methods [J]. Computer Engineering and Applications, 2018, 54(13): 11-18, 33.
- [4] 王敏,陈金勇,王港,等. 面向深度学习的遥感图像目  
标样本生成方法[J]. 国外电子测量技术, 2019, 38(4):60-65.  
WANG M, CHEN J Y, WANG G, et al. Remote sensing image object sample generation method for deep learning [J]. Foreign Electronic Measurement Technology, 2019, 38(4): 60-65.
- [5] 梁继然,陈壮,董国军,等. 结合注意力机制和密集连接网络的车辆检测方法[J]. 电子测量与仪器学报, 2022,36(3):210-216.  
LIANG J R, CHEN ZH, DONG G J, et al. Vehicle detection method combining attention mechanism and dense connection network [J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(3): 210-216.
- [6] 朱焕宇,王明泉,李磊磊,等. 胎体钢丝帘线缺陷的 CD-YOLO 分割算法应用[J]. 国外电子测量技术, 2023,42(7):175-180.  
ZHU H Y, WANG M Q, LI L L, et al. Application of CD-YOLO segmentation algorithm for carcass steel cord defects [J]. Foreign Electronic Measurement Technology, 2023, 42(7): 175-180.
- [7] 蓝金辉,王迪,申小盼. 卷积神经网络在视觉图像检测的研究进展[J]. 仪器仪表学报,2020,41(4):167-182.  
LAN J H, WANG D, SHEN X P. Research progress of convolutional neural network in visual image detection [J]. Chinese Journal of Scientific Instrument, 2020, 41(4): 167-182.
- [8] 赵羽,杨洁,刘森,等. 面向视频监控基于联邦学习的智能边缘计算技术[J]. 通信学报,2020,41(10):109-115.  
ZHAO Y, YANG J, LIU M, et al. Federated learning based intelligent edge computing technique for video surveillance [J]. Journal on Communications, 2020, 41(10): 109-115.

- [9] YANG K, YI J, LEE K, et al. FlexPatch: Fast and accurate object detection for on-device high-resolution live video analytics [C]. Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 1898-1907.
- [10] GAO C S, WANG Y, HAN Y H, et al. IVP: An intelligent video processing architecture for video streaming [J]. IEEE Transactions on Computers, 2023, 72(1): 264-277.
- [11] 叶一飞, 王建中. 基于点云的复杂环境下楼梯区域识别[J]. 电子测量与仪器学报, 2020, 34(4): 124-133.  
YE Y F, WANG J ZH. Stair area recognition in complex environment based on point cloud[J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(4): 124-133.
- [12] 刘畅, 金立左, 费树岷, 等. 固定多摄像头的视频拼接技术[J]. 数据采集与处理, 2014, 29(1): 126-133.  
LIU CH, JIN L Z, FEI SH M, et al. Video stitching technology based on fixed multi-camera [J]. Journal of Data Acquisition and Processing, 2014, 29(1): 126-133.
- [13] 杜运亮, 王明甲. 基于半监督域适应的微弱光环境下行人检测研究[J]. 电子测量与仪器学报, 2024, 38(1): 106-113.  
DU Y L, WANG M J. Research on pedestrian detection in low-light conditions based on semi-supervised domain adaptation [J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(1): 106-113.
- [14] 赵永强, 饶元, 董世鹏, 等. 深度学习目标检测方法综述[J]. 中国图象图形学报, 2020, 25(4): 629-654.  
ZHAO Y Q, RAO Y, DONG SH P, et al. Survey on deep learning object detection[J]. Journal of Image and Graphics, 2020, 25(4): 629-654.
- [15] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1): 1-17.  
LU H T, ZHANG Q CH. Applications of deep convolutional neural network in computer vision [J]. Journal of Data Acquisition and Processing, 2016, 31(1): 1-17.
- [16] 薛海彬, 彭小苗, 颜锦奎. 基于树莓派的频率合成器系统设计[J]. 电子测量技术, 2018, 41(16): 125-131.  
XUE H B, PENG X M, YAN J K. Frequency synthesizer system design based on Raspberry Pi [J]. Electronic Measurement Technology, 2018, 41(16): 125-131.
- [17] 汪乐章, 林娟, 唐伊文, 等. 基于树莓派与计算机视觉的家庭火灾报警系统的设计与研究[J]. 电子测量技术, 2019, 42(8): 83-87.  
WANG L ZH, LIN X, TANG Y W, et al. Design and research of home fire alarm system based on Raspberry Pie and computer vision [J]. Electronic Measurement Technology, 2019, 42(8): 83-87.
- [18] MIAO L N, ZHOU D, LI X. Spatial non-cooperative object detection based on deep learning [C]. China Automation Congress, 2022: 116-121.
- [19] ZHAO Z Q, ZHENG P, XU S T, et al. Object detection with deep learning: A review [J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(11): 3212-3232.
- [20] 朱登柯, 侯兴松, 要晓迪. 基于 EfficientDet 网络的输电线路故障检测[J]. 国外电子测量技术, 2021, 40(6): 144-151.  
ZHU D K, HOU X S, YAO X D. Research on power line fault detection based on EfficientDet network [J]. Foreign Electronic Measurement Technology, 2021, 40(6): 144-151.
- [21] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector [C]. Proceedings of the European Conference on Computer Vision (ECCV), 2016.
- [22] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]. 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.
- [23] REDMON J, FARHADI A. YOLOv3: An incremental improvement [J]. ArXiv preprint arXiv: 1804.02767, 2018.
- [24] TAN M, PANG R, LE Q V. EfficientDet: Scalable and efficient object detection [C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

## 作者简介

尹睿昀, 本科, 主要研究方向为图像处理、嵌入式系统应用。

E-mail: 1937847362@qq.com

全傲, 本科, 主要研究方向为图像处理、嵌入式系统应用。

姚新阳, 硕士研究生, 主要研究方向为图像处理、嵌入式系统应用。

袁观妙, 本科, 主要研究方向为图像处理、嵌入式系统应用。

王佳茹, 本科, 主要研究方向为图像处理、嵌入式系统应用。