

# 硕士学位论文

(专业学位论文)

基于目标检测与包裹重识别的

智慧物流桌面

**INTELLIGENT LOGISTICS DESKTOP BASED ON  
TARGET DETECTION AND PACKAGE RE-  
IDENTIFICATION**

于彤

哈尔滨工业大学

2023 年 5 月

国内图书分类号：TP391.4

国际图书分类号：004.8

学校代码：10213

密级：公开

## 硕士学位论文

# 基于目标检测与包裹重识别的 智慧物流桌面

硕士研究生：于彤

导师：刘劼教授

申请学位：专业硕士

学科：计算机技术

所在单位：计算学部

答辩日期：2023 年 5 月

授予学位单位：哈尔滨工业大学

Classified Index: TP391.4

U.D.C: 004.8

Dissertation for the Master Degree

**INTELLIGENT LOGISTICS DESKTOP BASED ON  
TARGET DETECTION AND PACKAGE RE-  
IDENTIFICATION**

<b>Candidate:</b>	Yu Tong
<b>Supervisor:</b>	Prof. Liu Jie
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Electronic Information
<b>Affiliation:</b>	Computer Technology
<b>Date of Defence:</b>	May, 2023
<b>Degree-Conferring-Institution:</b>	Harbin Institute of Technology

## 摘 要

随着电子商务的迅速发展，物流行业面临着巨大的挑战，如何提高物流效率和准确性成为了亟待解决的问题。同时物联网+AI 的快速发展使得智慧物流成为了一种新型物流模式。通过运用先进的信息技术和自动化设备，实现物流系统的智能化、高效化运作。本文研究了基于目标检测与包裹重识别的智慧物流桌面，旨在搭建一个实际应用的视觉分拣系统，为物流行业提供一种更高效、准确的分拣方式。

首先，通过分析物流万向桌面的功能需求完成了桌面结构的设计。利用 STM32 开发板、涡轮电机、万向轮和树莓派等组件，搭建了智慧物流万向桌面，并详细介绍了基于 STM32 开发板的底层驱动实现。其次，对视觉系统中的目标检测模型 YOLOv5 进行了改进，提出了一系列优化方案，实验验证表明在速度与精度上均取得了较好的效果。同时，研究了目标跟踪与多摄像头下的包裹重识别问题，介绍了基于 ByteTrack 的快递目标跟踪算法和基于运动建模的包裹重识别算法。最后，进行了实验验证并完善了异常反馈机制，以确保系统在实际应用场景中的稳定性和可靠性。本文所提出的方法在各项指标上表现出较高的性能，为智慧物流领域的发展提供了有益的探讨和实践经验。

**关键词：**智慧物流；STM32；目标检测；目标跟踪；重识别

## Abstract

With the rapid development of e-commerce, the logistics industry is facing enormous challenges, enhancing logistics efficiency and precision has emerged as a pressing issue that requires prompt resolution. Simultaneously, the swift advancements in IoT and AI technologies have given rise to smart logistics as an innovative approach to logistics management. By using advanced information technology and automated equipment, the intelligent and efficient operation of the logistics system can be realized. This paper studies an intelligent logistics desktop based on object detection and package re-identification, aiming to build a practical visual sorting system and provide a more efficient and accurate sorting method for the logistics industry.

Firstly, the design of the desktop structure was completed by analyzing the functional requirements of the logistics omnidirectional desktop. Using components such as STM32 development board, turbo motor, omnidirectional wheel, and Raspberry Pi, the intelligent logistics omnidirectional desktop was built, and the implementation of the underlying driver based on the STM32 development board was described in detail. Secondly, the YOLOv5 object detection model in the visual system was improved, and a series of optimization schemes were proposed. Experimental verification showed that better results were achieved in terms of speed and accuracy. Meanwhile, the problem of object tracking and package re-identification under multiple cameras was studied, and the express object tracking algorithm based on ByteTrack and the package re-identification algorithm based on motion modeling were introduced. Finally, experimental verification was carried out, and the abnormal feedback mechanism was improved to guarantee the steadiness and dependability of the system in real-world implementation settings. The methods proposed in this paper show high performance in various indicators, providing valuable discussion and practical experience for the development of the intelligent logistics field.

**Keywords:** intelligent logistics, STM32, object detection, target tracking, ReID

## 目 录

摘 要 .....	I
ABSTRACT.....	II
第 1 章 绪 论 .....	1
1.1 课题背景及研究的目的和意义 .....	1
1.2 国内外研究现状.....	2
1.2.1 智慧物流研究现状.....	2
1.2.2 目标检测研究现状.....	5
1.2.3 目标跟踪研究现状.....	7
1.2.4 重识别技术研究现状.....	8
1.3 本文的主要研究内容 .....	10
第 2 章 智慧物流万向桌面搭建 .....	12
2.1 引言 .....	12
2.2 物流万向桌面设计 .....	12
2.2.1 系统功能设计 .....	12
2.2.2 桌面结构设计 .....	14
2.3 基于 STM32 开发板的底层驱动 .....	15
2.3.1 指令解析与指令下发.....	16
2.3.2 电机驱动与编码器.....	18
2.4 本章小结 .....	20
第 3 章 基于 YOLOV5 的目标检测算法改进 .....	22
3.1 引言 .....	22
3.2 YOLOv5 目标检测算法研究 .....	22
3.2.1 YOLOv5 目标检测网络.....	23
3.2.2 数据集准备与划分.....	24
3.3 基于 YOLOv5 的算法改进 .....	26
3.3.1 非极大值抑制 NMS 算法改进 .....	26
3.3.2 损失函数改进.....	28
3.3.3 特征融合网络改进.....	30
3.3.4 主干网络轻量化.....	31
3.4 实验结果与对比评价 .....	33

3.4.1 目标检测评价指标.....	33
3.4.2 模型改进结果对比.....	34
3.5 本章小结.....	36
第 4 章 目标跟踪与包裹重识别算法.....	37
4.1 引言.....	37
4.2 基于 BYTETRACK 的快递目标跟踪算法.....	37
4.2.1 轨迹状态定义.....	38
4.2.2 轨迹构建与追踪.....	39
4.2.3 数据关联.....	41
4.2.4 优化策略.....	43
4.3 基于运动建模的包裹重识别算法.....	44
4.3.1 预测运动轨迹.....	45
4.3.2 误差确定.....	45
4.3.3 外观相似度辅助验证.....	46
4.4 本章小结.....	47
第 5 章 实验验证与异常反馈.....	49
5.1 引言.....	49
5.2 实验设置.....	49
5.2.1 实验环境.....	49
5.2.2 实验数据.....	50
5.2.3 网络训练参数.....	50
5.3 实验结果及分析.....	51
5.3.1 评价指标.....	51
5.3.2 目标跟踪结果对比.....	52
5.3.3 包裹重识别结果对比.....	53
5.4 异常反馈机制.....	55
5.4.1 异常类型.....	55
5.4.2 反馈机制.....	56
5.5 本章小结.....	57
结论.....	60
参考文献.....	62
哈尔滨工业大学学位论文原创性声明和使用权限.....	65
致 谢.....	66

# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

智慧物流是指利用现代信息技术手段对物流过程进行全方位、全过程的监控和管理，从而实现物流资源的优化配置和物流效率的提升。智慧物流作为新一代物流发展方向，已经成为各国物流业发展的重要战略目标。其背景和意义主要有以下几点：

**信息技术的快速发展：**随着信息技术的快速发展，智慧物流所需要的基础技术条件已经得到了充分保障。如今物联网、云计算、大数据、人工智能等技术的应用已经成为智慧物流的基础支撑。

**在全球化经济与供应链管理中，**物流系统复杂性与不确定性不断提高，成本和效率成为企业发展瓶颈。智慧物流运用信息技术优化资源配置，提高运作效率，降低支出，为企业带来更大利润。

**环保要求的提高：**现在社会对环保要求越来越高，智慧物流能够通过精细化管理和智能化技术手段实现绿色物流，降低对环境的污染，从而实现可持续发展。

随着全球经济一体化的深化，产业结构不断改变，各行业面临竞争压力，物流业亦然。智能物流作为中国高质量发展的关键，展现出新活力。以互联网信息技术为基础，智能物流在仓储、包装、流通、运输等环节实现系统感知，具备六大核心功能<sup>[1]</sup>。如图 1 所示的智慧物流案例包括仓储、运输、配送等环节，与计算机技术和物联网紧密结合，如路径规划算法和智能分拣技术在仓储物流系统中的应用。

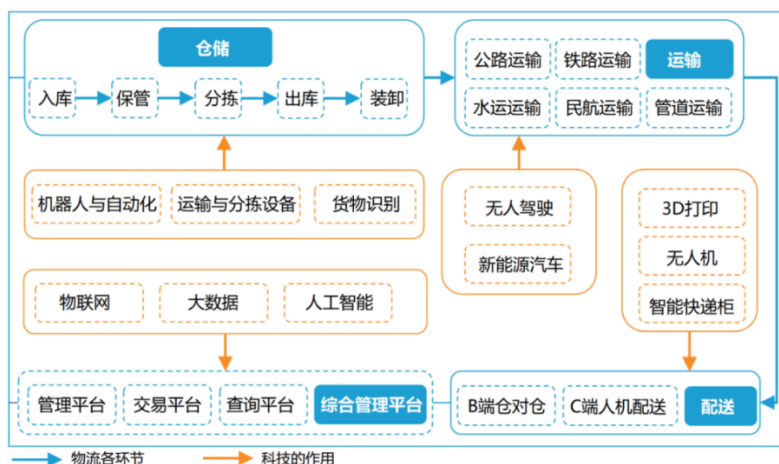


图 1-1 智慧物流案例



在物流运输与配送环节中,分拣作业涉及的内容很多,操作较为复杂,是物流发展中的一个关键环节,需要依据客户信息、货品规格、发送要求来进行分拣,分拣设备需要将货物从相应区域快速、精确地进行分拣,根据货物流向分类并进行装载运输,这个过程在物流中心的日常工作中是最耗费人力、最复杂、设备占用空间最大的环节。在配送中心的搬运成本中,分拣操作的搬运成本大约占据 80%;在劳动力密集型的配送中心,与分拣任务直接相关的人力资源约占 50%;分拣任务所需时间大约占据整个配送中心工作时间的 30%~40%。因此,对分拣任务进行合理规划和管理对于提高配送中心的工作效率和降低作业成本具有显著的成效<sup>[2]</sup>。

但是由于近年来随着快递和电商行业的迅速发展,区域核心物流中心对自动化技术的需求呈现出爆发式增长,它的分拣能力还是不足以满足需求。因此,物流分拣与配送领域急需一种全新的传送分拣技术,来改善当前主流的分拣系统存在的问题。为了更好应对电商和快递行业对智慧物流的强烈需求,在德国最新的智慧物流黑科技——蜂窝输送技术 Celluveyor 基础上,我国也需要研制一款基于目标检测与跟踪的能够满足中小型分拣中心需求的智能物流传送桌面,在保证高而灵活的吞吐量的同时,能有效提升分拣货物的灵活性和效率,降低成本和空间。因此本课题将提出一种基于目标检测与包裹重识别的智慧物流万向桌面,通过来运用于物流系统内的各种应用

## 1.2 国内外研究现状

### 1.2.1 智慧物流研究现状

智慧物流分拣技术是一个全球性的热点研究领域,各个国家和地区都在积极探索和应用相关技术。国内外的技术路线上大体类似,主要包括物品信息采集、智能分拣系统设计、分拣过程监控和分拣数据分析等环节。在具体实现上,国外智慧物流分拣更加注重技术的集成和应用,例如将机器视觉、物联网、云计算等多种技术手段相结合,实现分拣过程的自动化、智能化和信息化。国外智慧物流分拣技术已经在许多国家和地区得到广泛应用,尤其在欧美等发达国家和地区。例如,UPS 推出的“WorldShip”系统,可以通过条形码扫描、RFID 识别等方式采集包裹信息,利用物联网技术实现自动分拣和自动装箱,大大提高了分拣效率和准确度;DHL 推出的“SmartSolutions”系统,可以通过无线传感器实时监测货物状态、温度等信息,保障货物运输的安全和可靠性。还包括美国的康耐视、日本的 MUJIN、德国的 ISRA 等企业均拥有智能分拣机器人系统,如图 1-2 所示。但目前国外的智能分拣机器人系统仍旧存在各种问题,

例如无法解决包裹进入错误轨道的矫正问题等。



(a) 康耐视-物流读码通道

(b) MUJIN-分拣机器人

图 1-2 国外智能分拣系统案例

如图 1-3 所示为德国不莱梅大学生产与物流研究所(BIBA)推出的一款名为 Celluveyor 的智能传输系统，该传输系统是由很多个六边形的模块组成，如图 1-4 所示，并且每个小模组又是由三个可以多向转动的轮子组成<sup>[3]</sup>，每个轮子又可以进行独立转动。整个系统是基于视觉识别系统创建的，它使用 3D 相机进行跟踪，实时识别位于错误轨道上的包裹并纠正其位置，以便它将包裹移动到正确的位置。工作人员可以使用电脑上的程序对任意一个模块上的万向轮进行控制，这些万向轮以不同的排列组合进行滚动，从而可以设定很多种不同的运动路径，从而提高物流仓储的分拣效率<sup>[4]</sup>。

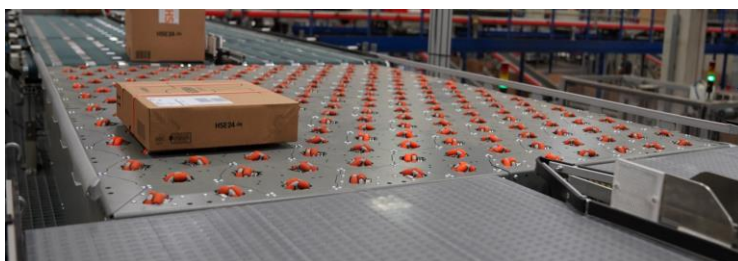


图 1-3 Celluveyor 的智能传输系统



图 1-4 Celluveyor 的智能传输组件

近几年，在货物运输自动化、高效运营和管理等方面，我国物流行业取得

了显著进展。得益于物联网、大数据、云计算和人工智能等新一代信息技术的支持，智能物流得以迅速发展，从而大幅提升了国内物流行业的服务质量、降低了服务成本并减少了资源消耗。尽管我国物流装备企业相较于其他国家起步较晚，但在技术差距方面正逐步缩小，而且在技术难度最高的领域（如自动化立体仓库、AGV 和分拣机器）也实现了重要突破。

德马科技主要研究的是智能运输及分拣的相关系统，包括交叉带、滑块、摆臂分拣机，密集存取穿梭车等设备，如图 1-5 所示，虽然在一定程度上减少了人工搬运，提高了效率，但仍无法解决多方向分拣，分拣路径规划等问题。

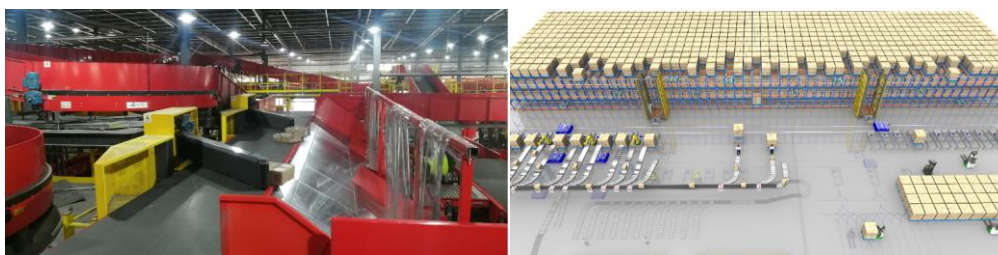


图 1-5 摆臂分拣机及智能穿梭车

艾尔森的 3D 定位系统是国内首款基于 3D 视觉引导的机器人智能分拣系统，如图 1-6 所示，该系统针对杂乱、无序堆放的工件，通过结合 3D 快速成像技术对物体表面进行扫描以生成点云数据、采用深度学习方法对数据进行智能分析处理，以及机器人路径规划以计算拣选目标的实时位置坐标。最后，引导机器人完成拣选任务。然而，在处理点云数据的过程中，深度学习所需的时间较长，因此尚未实现实时拣选。



图 1-6 3D 视觉引导系统

基恩士推出了机械手视觉的相关产品，可以通过图像处理系统和图像传感器，形成准确完善的视觉系统，并匹配合适尺寸的机械手，完成自动化的装箱与分拣，如图 1-7 所示，但是对于实际快递场景的不同尺寸包裹，以及不同重量的大规模数量包裹是否适配，仍有待试验与提高。



图 1-7 机械手视觉装箱

京东的智能小黄人机器人能够根据物流仓库内部的二维码地标进行分拣口的规划。当包裹放置在机器人上时，小黄人能迅速识别包裹信息，并通过最快速度和最佳行驶路线将快递送至相应的分拣口<sup>[5]</sup>，如图 1-8 所示。然而，制造小黄人的成本较高，技术难度大，且对场地要求较为严格，因此不适用于中小型分拣中心。小黄人机器人仅适用于京东本地仓，无法广泛应用于其他场景。



图 1-8 京东小黄人

## 1.2.2 目标检测研究现状

目标识别与检测在计算机视觉领域中是一项至关重要的任务，其目的是在图像或视频中定位和识别特定目标。目标检测可以应用于许多领域，例如自动驾驶、智能监控、无人机等。目标检测的方法和算法也在不断地发展和完善。目标识别与检测的方法主要可分为两个时期：传统的目标识别方法和基于深度学习的目标识别方法。在深度学习兴起之前，传统的目标识别与检测方法<sup>[8]</sup>主要由三个阶段构成，包括区域筛选、特征抽取和分类。，如下图 1-9 所示<sup>[9,10,11]</sup>。代表性的成果包括 Viola-Jones 检测器,HOG 行人检测器<sup>[12,13,14]</sup>等。然而这种方式存在多种问题，主要包括：基于滑动窗口的区域选择具有较大的时间复杂度，



并且定位结果相当粗糙；手工设计的特征鲁棒性较差，难以应对复杂场景。



图 1-9 传统的目标检测方法流程

2013 年由 Ross Girshick 等人提出的 RCNN<sup>[13]</sup>是深度学习在目标检测领域的创始之作，目前的目标检测算法可根据其构架主要划分为单阶段（One-stage）和双阶段（Two-Stage）两个大类。双阶段目标检测算法基本是对 RCNN 的持续改进，沿用 RCNN 的设计思想，双阶段的目标检测算法也被称为基于区域候选的目标检测算法，它通过明确的区域候选将检测问题转换为对生成的候选区域内局部图像的分类问题。这种算法在第一阶段产生区域候选，然后在第二阶段对感兴趣区域的内容进行分类和回归处理。如下图 1-10 所示。代表性算法有 R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN 等。二阶算法的共同特点是检测精度较高，但是复杂的结构限制了它们的检测速度，因此一般实时性较差。

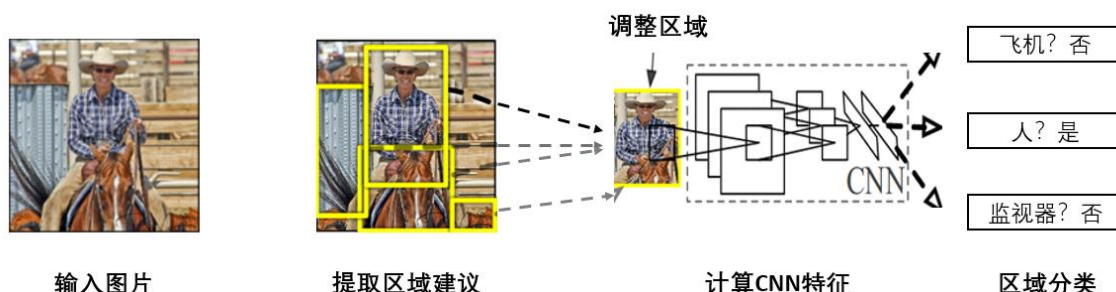


图 1-10 二阶目标检测算法 R-CNN

不同于双阶段算法的分阶段设计思想，单阶段算法使用一个可以端到端训练的网络直接输出目标的边界框和类别，单阶段目标检测算法也被称为基于回归的目标检测方法，这种算法并不会直接生成感兴趣区域，而是将目标检测任务视为对整个图像的回归问题。2016 年，Joseph Redmon 等人提出了著名的 YOLO 算法<sup>[14]</sup>是首个单阶段实时目标检测算法，该算法将图像划分为若干个网格，使用 CNN 对输入图像进行特征提取，通过全连接层输出张量对每个网格中的感兴趣目标进行预测，网络结构如下图 1-11 所示。YOLOv2<sup>[15]</sup>对 YOLO 进行了一系列的改进，包括引入锚框机制、使用更精细的特征、批量归一化<sup>[16]</sup>（Batch Normalization）、多尺度训练等，并使用卷积替代了先前的全连接层，

在速度和精度上都有了较大的提升。YOLOv3<sup>[17]</sup>借鉴了当时多项卷积神经网络的最新研究成果，这包括利用残差网络<sup>[18]</sup>（ResNet）来提取特征映射，运用特征金字塔网络（Feature Pyramid Network）完成多尺度特征的整合。YOLOv4 算法是由 Alexey 等人提出的是对 v3 的持续改进，引入了 SPP 和 PAN，并在数据增强、正则化、损失函数、激活函数等方面进行了改进和调优。YOLOv5 相比于 YOLOv4，为了提升目标检测算法的性能和鲁棒性，数据增强是一项重要的技术手段。其中，缩放、色彩空间调整和马赛克增强是常用的数据增强方法。同时，自适应锚定框的引入也是一种有效的算法优化方法。

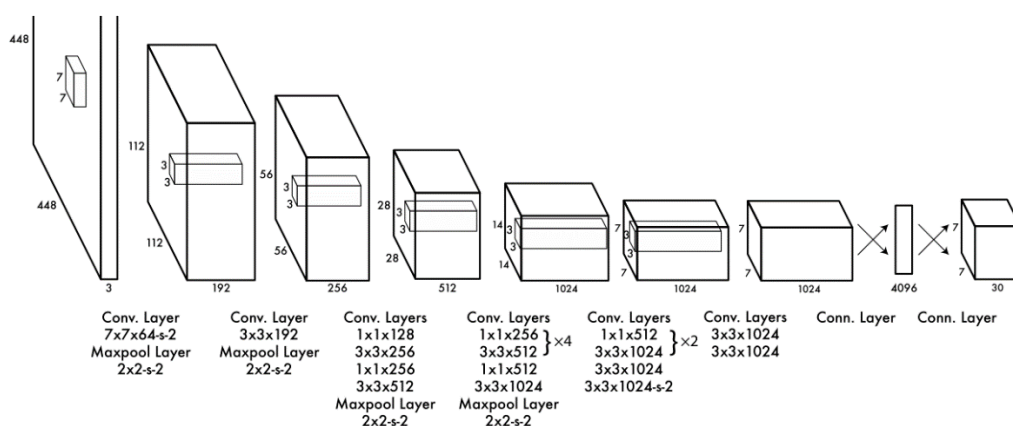


图 1-11 YOLO 的网络结构

### 1.2.3 目标跟踪研究现状

目标跟踪是计算机视觉领域中的一重要研究课题，其目的是在视频序列中跟踪目标的运动轨迹，常见的应用包括视频监控、自动驾驶、无人机导航等。目标跟踪的研究可以分为两个方向：传统方法和深度学习方法。

传统方法主要使用基于特征匹配、背景建模、运动模型等技术来实现目标跟踪。其中，基于特征匹配的方法使用目标的视觉特征（如颜色、纹理、形状等）来进行匹配，以实现目标跟踪。背景建模方法则通过建立背景模型来检测目标的运动，从而进行跟踪。而运动模型方法则是基于目标的运动规律来进行跟踪。

深度学习方法则是利用深度神经网络来学习目标特征表示和运动规律，并根据这些特征来进行目标跟踪。常见的深度学习方法包括基于卷积神经网络（CNN）的方法、基于循环神经网络（RNN）的方法、基于图神经网络（GNN）的方法等。近年来，深度学习技术在目标跟踪领域得到广泛应用。其中，基于卷积神经网络（CNN）的目标跟踪方法在性能和效率上取得了很大的进展。如

Siamese<sup>[19]</sup>网络、Region Proposal-based<sup>[20]</sup>方法和多任务学习等。

近几年目标跟踪算法发展比较成熟后，更多研究者将注意力放在多目标跟踪、跨域目标跟踪及实时目标跟踪上。多目标跟踪是指在一个视频序列中同时跟踪多个目标。多目标跟踪的研究重点包括目标关联、轨迹预测和跟踪策略等。其中，深度学习方法在多目标跟踪中也得到了广泛应用。跨域目标跟踪是指在训练和测试数据集之间存在分布差异的情况下，仍能准确地跟踪目标。跨域目标跟踪的研究包括领域自适应、零样本学习和元学习等。实时目标跟踪是指在保持高精度的前提下，能够实现实时跟踪。实时目标跟踪的研究重点包括目标检测与跟踪的联合优化、轻量化网络设计和硬件加速等。随着目标检测技术的蓬勃发展，基于检测的跟踪（Tracking-By-Detection, TBD）方式逐渐替代了传统方法成为目前多目标跟踪的主要框架。TBD 跟踪器的优劣主要依赖于前置的目标检测方法的效果。目前，TBD 方式最具代表性的算法是 Sort<sup>[21]</sup>（Simple Online and Realtime Tracking）和 DeepSort<sup>[22]</sup>。Sort 算法基于卡尔曼滤波进行预测，数据关联部分采用匈牙利算法，以完成多目标跟踪。该算法的优点是简洁、高效、便于部署，但由于忽略了外观特征，只有背景较为简单的情况下才比较准确。与此同时，DeepSort 算法是在 Sort 算法的基础上进行了改进。DeepSort 加入了目标的外观特性和运动特征，并采用了深度学习方法对目标对应的数据进行训练。该算法增强了对缺失和遮挡情况的关注，但同时也牺牲了速度和空间，增加了部分计算量。

#### 1.2.4 重识别技术研究现状

当前，重识别技术主要被应用于行人领域。该技术可以对不同监控下的行人图像进行特征提取、匹配，判断是否为同一人员。行人重识别（ReID）具有跨摄像头的重要特性，是人脸检测识别的一个拓展领域。随着目前监控的增加，但各地的设备质量参差不齐、成像参数均不相同等因素，监控视频中得到的行人特征往往不足以进行人脸识别，因此提出了基于行人的穿着、体态、发型等信息的 ReID 来识别行人。

与行人检测相比，行人重识别的研究还处于相对不成熟的阶段。然而，早在 1996 年，学术界就开始对行人重识别问题进行探讨。随着时间的推移，该领域的研究日益增多。2006 年，行人重识别概念在会议上首次被提出。随后，越来越多的相关研究出现。2007 年，VIPeR 数据库的提出对于行人重识别研究具有重要意义。如图 1-12 所示<sup>[23, 24, 25, 26, 27]</sup>，这是 ReID 从 1997 年到 2016 年间几个重要事件的发展历程。

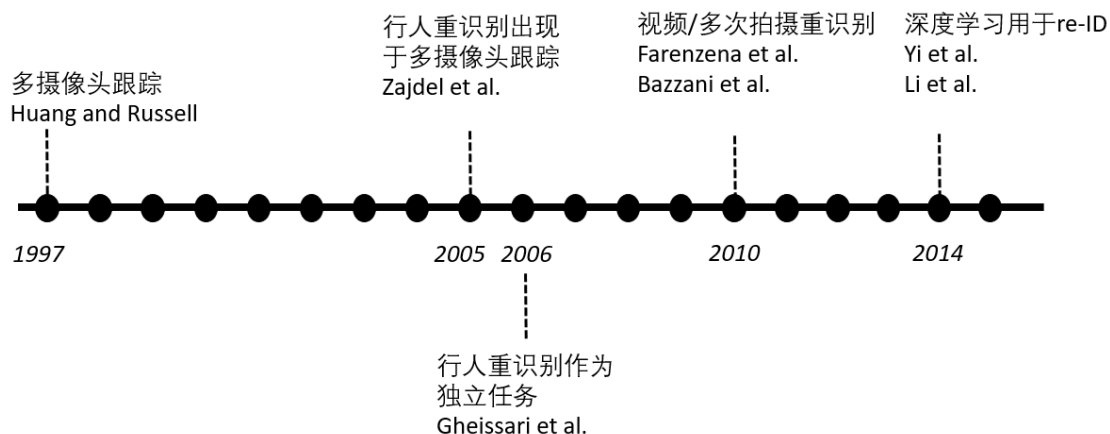


图 1-12 1997-2016 年行人重识别发展

行人重识别研究领域仍旧存在着多个挑战，如视频图像等分辨率低、拍摄角度和距离相差较大、光线变化规律难以寻找、遮挡和背景噪声等。由于监控视频的画面通常不够清晰，依靠仅仅人脸识别等方式进行重识别难度很大，因此只能利用面部之外的人体信息。与此同时，不同行人外形特征也可能相似，这更加增加了行人重识别的难度。此外，实际监控场景下周围环境的背景噪声可能非常大，往往存在遮挡和背景干扰等情况，这使得使用步态等特征进行重识别更加困难。由于重识别图像通常来自于不同监控拍摄，存在拍摄参数差异，导致同一个行人在不同摄像机下也存在较大差别。总之，这些挑战给行人重识别的研究带来了很大的困难，使得目前的研究距离实际应用落地于安防等场景还存在很大差距<sup>[28, 29, 30]</sup>。

行人重识别算法主要包括基于表征的 ReID 方法、基于度量的 ReID 方法、基于局部特征的 ReID 方法、基于视频序列的 ReID 方法四个方面：

**基于表征的 ReID<sup>[31]</sup>方法：**基于表征学习的方法是行人重识别领域中广泛应用的一种方法。这是因为深度学习，尤其是卷积神经网络(CNN)的极速发展。由于 CNN 能够主动提取图像数据中的表征特征，因此有些学者将行人重识别问题依此视为分类或验证问题：分类部分指的是利用行人属性或是身份 ID 等作为训练标签来进行模型的训练；验证部分是指输入一对行人图片，让神经网络学习这两张图片是否属于同一人。

**基于度量的 ReID<sup>[32]</sup>方法：**度量学习是一种常用于图像搜索领域的方法。与特征学习不同，度量学习的目标是通过神经网络学习出两张图片之间的相似之处，并按照统一的相似度指标进行衡量。在 ReID 问题中，这意味着同一人的不同图片之间的相似度应大于不同行人图片之间的相似度。网络的损失函数会最小化同一行人图片（正样本对）之间的距离，同时最大化不同行人图片（负样本对）之间的距离。常用的度量学习损失函数包括对比损失(Contrastive



loss)<sup>[33]</sup>、三元组损失(Triplet loss)<sup>[34,35,36]</sup>、四元组损失(Quadruplet loss)<sup>[37]</sup>、难样本采样三元组损失(Triplet hard loss with batch hard mining, TriHard loss)<sup>[38]</sup>、边界挖掘损失(Margin sample mining loss, MSML)<sup>[39]</sup>。

基于局部特征的 ReID<sup>[40]</sup>方法: 前期的行人重识别研究主要关注于全局特征, 即利用整张图像生成特征向量来进行图像搜索。然而, 随着研究的深入, 人们逐渐认识到全局特征存在瓶颈问题, 因此开始关注局部特征的研究。在提取局部特征方面, 常见的方法包括图像切块、骨骼关键点定位以及姿态校正等。

基于视频序列的 ReID<sup>[41]</sup>方法: 虽然受限于数据集大小和 GPU 问题, 目前单帧的行人重识别研究还是主流。然而, 单帧图像的信息是有限的, 因此越来越多的工作开始关注利用视频序列进行行人重识别的研究。基于视频序列的方法与基于单帧图像的方法最主要的不同点在于, 前者不仅考虑了图像的内容信息, 还考虑了帧与帧之间的运动信息。能够更好的解决遮挡、背景噪声等问题, 同样的也会牺牲一些实时性。

### 1.3 本文的主要研究内容

本课题主要研究内容将包含以下四个个方面, 如图 1-13 所示:



图 1-13 本课题主要研究内容

#### 1) 智慧物流万向桌面搭建:

利用 STM32 开发板, 涡轮电机、万向轮、树莓派完成智慧物流万向桌面的搭建。上游为路径规划算法, 由上游进行实时路径规划, 规划好路径后, 将相关指令解析给树莓派。由树莓派作为中间层, 控制多个相邻的 STM32 开发板, 并判断需要运动的万向轮属于哪块开发板, 并对指定开发板下发万向轮旋转方向、旋转速度、旋转时间的指令。下游利用 STM32 开发板控制涡轮电机带动万向轮转动。其中下游端包括接收到树莓派指令后, 进行旋转方向、旋转速度、旋转时间的解析, 并利用编码器实时反馈具体万向轮的实际运行情况, 便于路线的规划。

#### 2) 基于 YOLOv5 的目标检测算法改进:

由于本课题对于实时性的要求,应采用单阶段目标检测算法,拟采用 YOLO v5 进行目标检测,该模型在灵活性与速度上远强 YOLO v4,且在模型的快速部署上具有极强优势。由于智慧物流桌面中只有快递包裹 box 这一个类别,且考虑到本课题对于实时性的严格要求,最终采用 YOLO v5s 作为快递包裹的检测网络,收集真实数据集训练目标检测网络,并实现目标与其货物信息的对应匹配;并通过实验选取合适的优化方案对 YOLO v5s 进行模型优化:

将 YOLO v5s 网络中的非极大值抑制 NMS 算法改进 Soft-NMS;

将 YOLO v5s 中原有的损失函数 CIOU 改进为 EIOU;

将 YOLO v5s 中的 PANet 层改为 efficientDet 中的 BiFPN;

将 YOLO v5s 的主干网络轻量化为 PP-LCNet。

### 3) 目标跟踪及多摄像头下的包裹重识别:

由于快递包裹尺寸不同,且大小不同的快递箱尺寸存在较大区别,若采取正视图作为视角,将存在大包裹完全遮挡小包裹的问题,导致目标检测出现漏检的情况,因此本课题选取俯视图作为主视角;而摄像头在俯视角的摄像范围相对较小,需要多个摄像头交互才能完整收集物流桌面的视觉信息,因而存在目标跟踪与多摄像头下包裹重识别的问题。在物流桌面起始摄像头位置处进行首次目标检测,检测到包裹目标后,需要为包裹分配唯一的包裹 ID;在后续包裹运行过程中,当包裹进入其他摄像头检测范围内时,再次通过目标检测算法检测到该包裹时应与包裹唯一 ID 绑定,防止同一包裹重复识别分配多个 ID 的情况出现。

### 4) 系统实验验证与包裹异常反馈:

由于上游的路径规划算法到中间层树莓派的指令解析到下游的开发板下发指令过程中可能存在时间延迟,且真实情况下多个快递同时运输时会存在碰撞的情况出现,导致快递包裹无法按照规划路线行进的问题,所以需要通过目标检测及包裹 ID 绑定情况进行异常反馈,便于后续路径规划算法实时更新,规划最新路径。

## 第2章 智慧物流万向桌面搭建

### 2.1 引言

在探究基于目标检测和包裹重识别的智慧物流桌面的研究中，必须配备实际的蜂窝万向轮组件和传感器设备来构建物流桌面，以便进行数据采集、视觉算法设计以及系统测试验证等任务。这些设备可以帮助收集实时物流数据，包括包裹的位置、体积、重量等信息。同时，它们还可以实时监测包裹的状态和运动轨迹，以帮助我们实现自动化的包裹分类和分拣。除此之外，还可以通过这些设备对万向轮组件的性能进行测试和优化，以确保其能够有效地支持智慧物流桌面的应用。

本章将介绍物流万向桌面的设计和开发过程，其中包括系统功能设计和桌面结构设计两个方面。同时，为了实现底层的驱动控制，本章还将介绍基于STM32开发板的底层驱动开发，其中包括指令解析与指令下发、电机驱动与编码器等方面的内容。通过本章的介绍，读者将了解物流万向桌面的设计和开发过程，以及底层驱动的实现原理，为后续的智能化控制和优化奠定基础。

### 2.2 物流万向桌面设计

#### 2.2.1 系统功能设计

当前的智慧物流系统几乎都是在单通道的传送带上对快递进行检测识别和定位，相机及检测只负责传送带上的少部分区间。物流万向桌面希望实现快递分拣从一维到二维的扩展，以此来保障快递可以在二维平台上进行精准、灵活、智能的运输。

为实现快递在物流桌面上的有效分拣，本系统需要满足的功能如图 2-1 所示：

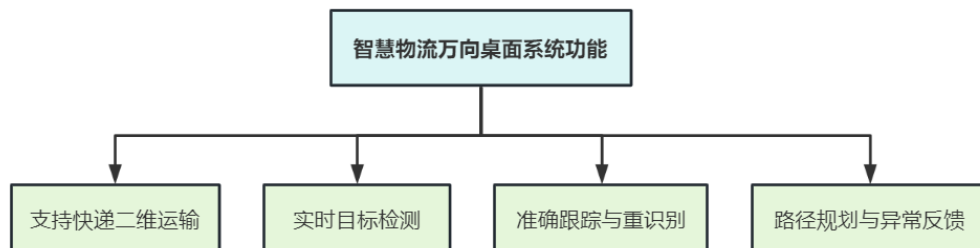


图 2-1 智慧物流万向桌面系统功能

1) 支持快递二维运输: 所设计的智慧物流万向桌面需要支持快递二维运输。传统的传送带物流系统本质上是一维快递分拣, 即快递包裹只支持线性同一方向的前后传送。为实现快递分拣从一维到二维的扩展, 在设计物流桌面的硬件搭建时至少需要支持快递多方向运动。

2) 实时目标检测: 智慧物流万向桌面同时应具备视觉感知系统, 对于每一个进入物流桌面的快递包裹做到实时检测。尽可能减少错检、漏检的情况。

3) 准确跟踪与重识别: 在快递分拣过程中, 视觉系统需要能够追踪和识别每个快递包裹的身份信息, 以确保快递能够被正确地分拣和处理。在处理快递分拣过程中, 由于可能存在快递间的轻微遮挡现象, 因此系统需要具备对快递身份 ID 的高效跟踪和定位能力, 以避免快递身份信息的丢失或切换。为此, 系统需要设计出适当的视觉算法和信号处理模块, 以实现快速对快递身份 ID 的稳定识别和跟踪, 并在遮挡情况下能够快速自适应调整, 以保证快递信息的准确和高效识别。通过这样的方式, 可以确保快递分拣系统的高效稳定运行, 提高快递物流的处理效率和服务质量。

4) 路径规划与异常反馈: 智慧物流桌面应具备路径规划和异常反馈的功能, 可以通过路径规划算法和异常处理模块, 实现对物流运输过程的智能化管理和自动化控制。

针对智慧物流万向桌面所需要的功能进行方案设计, 需要考虑到系统的整体架构和功能模块, 以及各个模块之间的交互和通信。在此基础上, 可以设计出一套完整的方案来实现系统的各种功能和任务。具体而言, 最终方案应该包括三个主要部分: 主机端、树莓派端和硬件端。

主机端是整个系统的核心, 主要涉及到目标检测算法、包裹重识别预测、路径规划算法等方面的内容。目标检测算法可以使用深度学习模型如 YOLO、SSD 等来识别包裹的位置和尺寸信息, 包裹重识别预测则可以利用运动轨迹预测模型来识别包裹信息, 而路径规划算法可以使用 A\* 等算法来规划最优的包裹分拣路径。

树莓派端则是整个系统的控制中心, 负责对万向轮进行实时调控, 同时还需要进行时钟同步以确保各个模块之间的数据同步和协作。时钟同步可以使用 NTP 协议等方式来实现, 以确保所有设备的时钟一致。在树莓派端还需要实现各个模块之间的通信和数据传输, 以保证整个系统的协调性和稳定性。

硬件端则是实现智慧物流万向桌面的重要组成部分, 主要包括 STM32 开发板、涡轮电机和万向轮等硬件设备。STM32 开发板可以作为万向桌面的控制中心, 实现对涡轮电机和万向轮的实时控制和调节, 以及对传感器数据的采集和

处理。涡轮电机和万向轮则可以实现对包裹的运动和位置控制，以及实现万向桌面的各种运动方式和轨迹控制。

### 2.2.2 桌面结构设计

由于所设计的智慧物流万向桌面需要支持快递二维运输，因此在设计物流桌面的硬件搭建时，必须要考虑到支持快递多方向运动的需求。为了实现这一目标，我们共设计了三种智慧物流万向桌面的结构，分别是蜂窝结构、风车结构、和十字结构，如图 2-2 所示。以下是对这三种结构的详细描述：

1) 蜂窝结构：蜂窝结构桌面采用六边形单元作为基本构建模块。每个单元都配备了三个独立驱动的滚轮，可以分别控制滚轮的转速和方向。这种设计使得快递包裹能在一个平面上实现多向移动（方向改变最小角度为  $120^\circ$ ），从而在有限的空间内实现灵活的操控和移动物品。这种结构优势在于其高度紧凑且稳定，适合用于需要高度集成和密集传输的场景。

2) 风车结构：风车结构桌面采用正方形单元作为基本构建模块。每个单元都配备了四个独立驱动的滚轮，可以分别控制滚轮的转速和方向。这种设计使得快递包裹能在一个平面上实现多向移动（方向改变最小角度为  $90^\circ$ ），与蜂窝结构相比，风车结构可以实现更小的方向调整角度，使得快递包裹更容易沿直线运输。这种结构适用于需要高度灵活性和快速响应的应用场景。

3) 十字结构：十字结构桌面同样采用正方形单元作为基本构建模块。每个单元都配备了四个独立驱动的滚轮，可以分别控制滚轮的转速和方向。这种设计使得快递包裹能在一个平面上实现多向移动（方向改变最小角度为  $90^\circ$ ）。与风车结构相比，十字结构减少了因驱动延迟导致快递包裹原地打转的可能性。这种结构在提供良好操控性的同时，降低了误操作的风险，适用于对稳定性和安全性要求较高的场景。

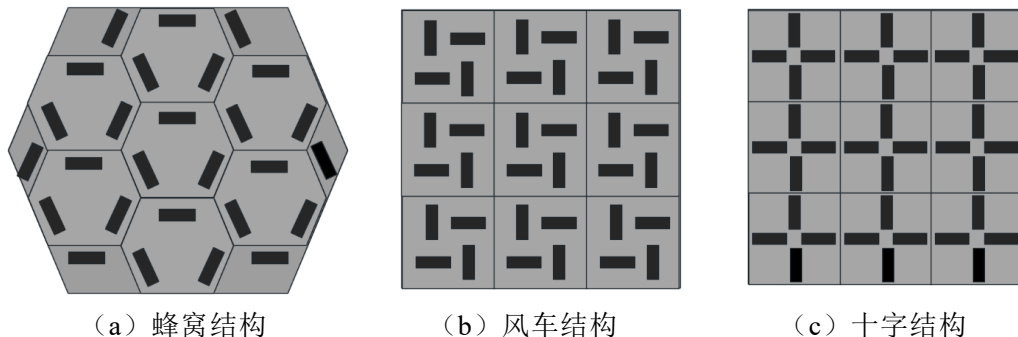


图 2-2 桌面结构设计示意图

综合考虑这三种智慧物流万向桌面的结构特点，并从成本、效果两个维度进行对比，参考了铺设相同面积需要的轮子数量、铺设同面积需要的开发板数量、

运输相同距离驱动电机数量、出现异常原因的数据对比，具体数据见表 2-1。由于开发板成本远高于轮子，且蜂窝结构和风车结构的异常原因更多（包括包裹原地打滑、重量较大的包裹无法运输到相邻单元等），因此最终选择了十字结构作为最终物流万向桌面基本单元。

表 2-1 三种桌面结构对比

桌面结构	铺设相同面积需要的轮子数量 (个/m <sup>2</sup> )	铺设同面积需要的开发板数量 (个/m <sup>2</sup> )	运输相同距离驱动电机数量 (个/m)	出现异常原因数量 (个)
蜂窝结构	3.7	1.2	2.4	3
风车结构	4	1	2	2
十字结构	4	1	2	1

## 2.3 基于 STM32 开发板的底层驱动

本文所使用的开发板为 STM32F103RET6，该开发板是一款高性能的 32 位微控制器，属于 STM32F1 系列产品。该芯片采用了 Cortex-M3 内核，其最高运行频率可达 72MHz，同时还集成了多种外设和接口，包括通用定时器、ADC、DAC、SPI、I2C 和 USART 等。此外，该芯片还具备出色的低功耗特性和完善的保护机制，能够满足各种应用场景的需求。STM32F103RET6 拥有 256KB 的 Flash 存储器和 64KB 的 SRAM，还支持外部存储器接口，能够扩展至 2GB 的存储容量。此芯片采用 LQFP64 封装，具备丰富的引脚功能，方便与其他模块或外设进行连接和通信。



图 2-3 STM32F103RET6 开发板



### 2.3.1 指令解析与指令下发

在这个场景中，STM32 开发板需要接收来自树莓派的指令，然后对这些指令进行解析。解析完成后，开发板会将相应的控制信号发送给各个万向轮电机，从而实现对电机的驱动。这里，使用了一种特定的上游数据格式，即"`{c L t 5 s 100}`"。这个格式的详细含义如下：

1) **cL**：这部分表示一个命令（**cmd-LEFT**），用于控制 STM32 开发板上连接的电机，使得万向轮上的物体向左侧转动。在这里，**cmd** 是一个 **char** 类型的变量，它可以取 12 个不同的值，分别为{1, 2, 3, 4, 5, 6, 7, 8, L, R, F, B}。这些值分别表示：

- 1-4：控制 STM32 开发板上连接的四个轮子正向旋转。
- 5-8：控制 STM32 开发板上连接的四个轮子反向旋转。
- L：表示让万向轮上的物体向左侧转动，实现这个动作需要让 2 号和 4 号轮子反向旋转。
- R：表示让万向轮上的物体向右侧转动，实现这个动作需要让 2 号和 4 号轮子正向旋转。
- F：表示让万向轮上的物体向前侧转动，实现这个动作需要让 1 号和 3 号轮子正向旋转。
- B：表示让万向轮上的物体向后侧转动，实现这个动作需要让 1 号和 3 号轮子反向旋转。

在接收到这些指令之后，STM32 开发板会将它们解析为相应的控制信号，然后将这些信号发送给对应的万向轮电机。通过这种方式，可以实现对各个电机的精确控制，从而让万向轮上的物体按照预期的方式运动。

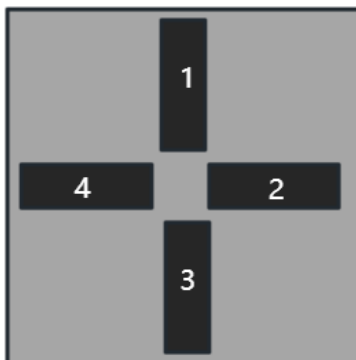


图 2-4 万向轮控制命令示意图

2) **t5**：这部分表示控制对应电机旋转的时间（**time-5s**），其中 **time** 对应的数据类型为 **int**，单位为秒（**s**）。在这个例子中，所选定的时间长度为 5 秒，因

此，在接收到此指令后，对应的电机将持续旋转 5 秒钟。

3)s 100: 这部分表示控制对应电机旋转的速度(speed-100mm/s), 其中 speed 对应的数据类型为 int, 单位为毫米每秒 (mm/s)。在这个例子中, 所选定的速度为 100mm/s, 因此, 在接收到此指令后, 对应的电机将以 100mm/s 的速度旋转。

为了实现这些功能, 我们需要按照图 2-5 所示的解析及下发流程进行操作。首先, 需要对上游数据进行长度检测, 避免接收到错误数据, 其次使用 Split 函数对上游数据进行分割, 然后将 cmd、time、speed 分别赋值。接下来, 需要对 cmd 进行匹配, 以确定需要驱动哪一个电机。最后, 找到对应的电机后, 根据给定的 speed 和 time 参数, 使电机按照指定的速度转动特定的时间。这个过程的关键功能是 split 函数, 其伪代码可以在图 2-6 中找到。



图 2-5 指令解析及下发流程

#### Algorithm 1 Split Function

```

1: procedure SPLIT(src, separator, dest, num)
2:   pNext  $\leftarrow$  NULL
3:   count  $\leftarrow$  0
4:   if src == NULL or strlen(src) == 0 then
5:     return
6:   end if
7:   if separator == NULL or strlen(separator) == 0 then
8:     return
9:   end if
10:  pNext  $\leftarrow$  strtok(src, separator)
11:  while pNext  $\neq$  NULL do
12:    *dest ++  $\leftarrow$  pNext
13:    count  $\leftarrow$  count + 1
14:    pNext  $\leftarrow$  strtok(NULL, separator)
15:  end while
16:  *num  $\leftarrow$  count
17: end procedure
    
```

图 2-6 split 函数实现



### 2.3.2 电机驱动与编码器

电机驱动部分是通过脉宽调制（PWM，Pulse Width Modulation）实现的，脉宽调制（PWM）是一种广泛应用的控制策略，主要用于调节电源信号的有效电压，以便精确地控制电机的速度和位置。PWM 实现电机转速控制的方式是产生一种方波信号，其脉冲宽度可以调节。尽管 PWM 信号的频率通常保持不变，但其占空比（即一个周期内脉冲宽度与总周期时间的比值）可以根据实际需求进行调整。

如图 2-7 是使用 PWM 来控制电机旋转的流程：

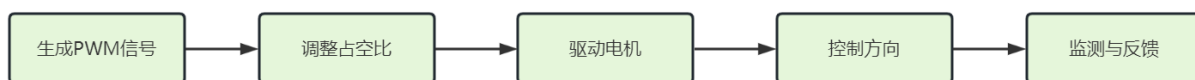


图 2-7 PWM 控制电机旋转流程

1) 生成 PWM 信号：控制器负责生成一个具有固定频率和可调占空比的 PWM 信号。这种信号可以根据具体需求进行调整，从而实现对电机的精确控制。通过改变占空比，可以调整输出功率，从而控制电机的速度和扭矩。

2) 调整占空比：根据所需的电机转速，调整 PWM 信号的占空比。占空比较高时，有效电压较大，电机转速较快；占空比较低时，有效电压较小，电机转速较慢。

3) 驱动电机：将 PWM 信号通过驱动电路传递到电机。驱动电路的主要职责是将 PWM 信号转换成适当的电流和电压，以便有效地驱动电机运转。在这个过程中，驱动电路起到了关键的作用，确保电机根据输入的 PWM 信号进行精确控制，从而实现对电机运动的有效管理。

4) 控制方向：通过改变 PWM 信号的相位或在多个绕组之间切换，可以实现对电机旋转方向的控制。这种调整方式使得电机能够灵活地响应不同的控制需求，实现顺时针或逆时针的旋转。

5) 监测与反馈：实时监测电机的转速和位置，并将反馈信息发送回控制器。控制器可以根据反馈信息调整 PWM 信号的占空比，以达到所需的转速和位置。

这个过程的关键功能是 Mot 函数，其具体实现的伪代码可以在图 2-8 中找到。

**Algorithm 2** MotA Function

```

1: procedure MOTa(val)
2:   if val > 20 then
3:      $TIM2\_CCR1 \leftarrow val + 1000$ 
4:      $TIM2\_CCR2 \leftarrow 0$ 
5:   else if val < -20 then
6:      $val \leftarrow -val$ 
7:      $TIM2\_CCR1 \leftarrow 0$ 
8:      $TIM2\_CCR2 \leftarrow val + 1000$ 
9:   else
10:     $TIM2\_CCR1 \leftarrow 0$ 
11:     $TIM2\_CCR2 \leftarrow 0$ 
12:   end if
13: end procedure
    
```

图 2-8 Mot 函数实现

编码器返回速度部分是通过霍尔编码器实现的。霍尔编码器（Hall Encoder）是利用霍尔效应（Hall Effect）构建的一种定位传感器。霍尔效应描述了当导体中的电荷在磁场中移动时，导体两侧会产生一个横向电压的现象。霍尔编码器运用这个原理来感测磁场的改变，进而确定物体的位置或速度。

如图 2-9 是霍尔编码器测速流程：

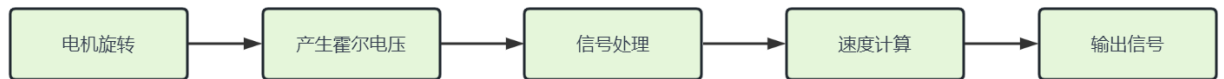


图 2-9 霍尔编码器测速流程

1) 电机旋转：电机运行时，磁场源所产生的磁场会因旋转而发生改变。霍尔元件会捕捉到这些磁场变化。

2) 产生霍尔电压：在磁场的作用下，霍尔元件产生一种与磁场强度成正比的横向电压，称为霍尔电压。磁场源旋转时，霍尔电压也会相应发生变化。

3) 信号处理：霍尔元件生成的霍尔电压被传送至信号处理电路。信号处理电路将霍尔电压转换为数字或模拟信号，这些信号包含磁场源的位置、速度和方向信息。

4) 速度计算：根据处理后的信号，计算电机的转速。这可以通过计算磁场源旋转过程中霍尔电压变化速率或测量磁性齿轮每次旋转的时间间隔来实现。

5) 输出信号：将计算得到的电机速度信息输出到控制系统，以便进行后续分析和控制。

这个过程的关键功能速度计算的 `BeatHandle_Encoder` 函数，其具体实现的伪代码可以在图 2-10 中找到。

---

**Algorithm 3** BeatHandle\_Encoder
 

---

```

1: static uchar Cnt  $\leftarrow$  0
2: long temp  $\leftarrow$  0
3: uchar i  $\leftarrow$  0, k  $\leftarrow$  0
4: Cnt  $\leftarrow$  Cnt + 1
5: for i  $\leftarrow$  0 to 3 do
6:     BUBBLING_LONG(&HallSpeed[i][0], 5)
7:     temp  $\leftarrow$  (HallSpeed[i][1] + HallSpeed[i][2] + HallSpeed[i][3])/3
8:     temp  $\leftarrow$  temp  $\times$  MotMagNumber
9:     temp  $\leftarrow$  10000000/temp
10:    if (i == 0) or (i == 3) then
11:        temp  $\leftarrow$  (HallDir[i])?(temp) : (-temp)
12:    else
13:        temp  $\leftarrow$  (HallDir[i])?(-temp) : (temp)
14:    end if
15:    MotSpeed[i]  $\leftarrow$  temp
16:    if Cnt  $\geq$  4 then
17:        if (Odometer[i] - Odometer_Keep[i])  $\leq$  (ulongMotMagNumber/10) then
18:            MotSpeed[i]  $\leftarrow$  0
19:            for k  $\leftarrow$  0 to 4 do
20:                HallSpeed[i][k]  $\leftarrow$  0
21:            end for
22:        end if
23:        Odometer_Keep[i]  $\leftarrow$  Odometer[i]
24:    end if
25: end for
26: if Cnt  $\geq$  4 then
27:     Cnt  $\leftarrow$  0
28: end if
    
```

---

图 2-10 BeatHandle\_Encoder 函数实现

## 2.4 本章小结

本章主要围绕智能物流万向桌面的设计与实现展开讨论，包括物流万向桌面设计、基于 STM32 开发板的底层驱动等部分。首先在 2.2 物流万向桌面设计部分，从系统功能设计和桌面结构设计两个方面进行了深入阐述。系统功能设计主要介绍了智能物流万向桌面的基本需要满足的功能和方案设计，为桌面结构设计提供了指导。在桌面结构设计中，详细地描述了蜂窝结构、风车结构和十字结构三种设计方案，以及它们的优缺点和适用场景，从成本、效果两个维度进行了对比以及本课题最终选择的基本单元。然后，在 2.3 基于 STM32 开发板的底层驱动部分，对指令解析与指令下发以及电机驱动与编码器进行了详细介绍。这部分工作主要利用了嵌入式系统开发方面的技术，并充分说明了如何

实现智能物流万向桌面的底层控制。综上所述，本章以严谨的论述和清晰的结构为智能物流万向桌面的设计与实现提供了全面的理论支持，并为后续章节的研究工作打下了坚实的基础。

## 第3章 基于 YOLOv5 的目标检测算法改进

### 3.1 引言

快递的精准识别与检测是智慧物流桌面的关键功能，当然在兼顾准确度的同时还需要考虑实时性，因此目标检测是整个视觉识别系统非常重要的一环。近年来深度学习在目标检测领域取得了突破性进展，尤其是基于 YOLO 系列的检测算法，它们在处理实时性要求较高的任务时表现尤为出色。然而，在特定场景下，现有的 YOLOv5 目标检测算法仍然存在一定的局限性，如检测精度、计算效率和模型复杂度等方面。

本章旨在研究和改进基于 YOLOv5 的目标检测算法，以提高其在智慧物流桌面应用中的性能。首先，我们将对 YOLOv5 目标检测网络及其关键组件进行详细的分析和研究。接下来，对数据集的准备和划分策略进行探讨，以便训练出适用于特定场景的检测模型。

在此基础上，本章将提出一系列针对性的算法改进策略，包括非极大值抑制（NMS）算法的优化、损失函数的改进、特征融合网络的优化以及主干网络的轻量化设计。这些改进将有助于提高目标检测算法的性能，实现更高的检测精度、计算效率和模型轻量化。

最后，本章将通过实验验证所提出的改进策略的有效性，并对比评价改进后的算法在智慧物流桌面应用中的性能。通过这一系列的研究和改进，我们期望在智慧物流桌面领域实现更高效、准确和实用的目标检测与包裹重识别方案。

### 3.2 YOLOv5 目标检测算法研究

YOLO（You Only Look Once）是一种实时目标检测算法，于 2016 年首次提出。它的主要优势是速度快、实时性强，适用于需要实时目标检测的应用场景。YOLO 的发展经历了多个版本的迭代和优化，从 YOLOv1 到 YOLOv5，在版本迭代的过程中网络结构、特征提取器和训练策略也在不断优化。目前 YOLOv5 版本在使得模型在保持较高准确性的同时，实现了较低的计算成本和更快的推理速度。YOLOv5 还提供了不同大小的模型版本（YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x），以满足不同应用场景的需求。考虑到本课题所处背景为物流桌面，其中所涉及的目标实体比较单一，且对实时要求较高。因此最终选择采用 YOLOv5s 对物流桌面中的包裹进行目标检测和识别。

### 3.2.1 YOLOv5 目标检测网络

YOLOv5s 是 YOLOv5 系列中最小型的模型，旨在保持较高性能的同时，降低计算成本和内存占用。虽然 YOLOv5s 的精度略低于其他 YOLOv5 版本，但其推理速度更快，适用于本课题所涉及的物流场景。

YOLOv5s 的网络结构主要包括主干网络、连接模块、预测模块、激活函数、数据扩充五部分，具体结构如图 3-1 所示：

**主干网络：**YOLOv5s 以 CSPNet 作为主干网络，该网络采用了交叉阶段层次结构，将特征图划分为两部分，并在各层之间交换信息，以增强特征表达的多样性。CSPNet 的设计有助于提升网络性能的同时，减小计算难度。

**连接模块：**YOLOv5s 将 PANet（路径聚合网络）作为连接模块。PANet 通过自上而下的路径和横向连接实现了多尺度特征的整合。这有利于提高对各种尺度目标的检测性能。

**预测模块：**YOLOv5s 的预测模块负责解码预测结果，包括类别概率、边界框坐标和目标置信度。YOLOv5s 利用三个不同尺度的特征图进行预测，分别对应大、中、小尺寸的目标。预测模块还包括 anchor boxes，以提高边界框预测的准确性。

**激活函数：**YOLOv5s 使用 Leaky ReLU 和 Swish 激活函数。Swish 激活函数是一种自适应激活函数，相较于传统的 ReLU 和 Leaky ReLU，它可以在保持较低计算复杂度的同时，提升网络性能。

**数据扩充：**YOLOv5s 在训练过程中采用了数据扩充策略，如图像翻转、缩放、旋转等，以增强模型的泛化能力。

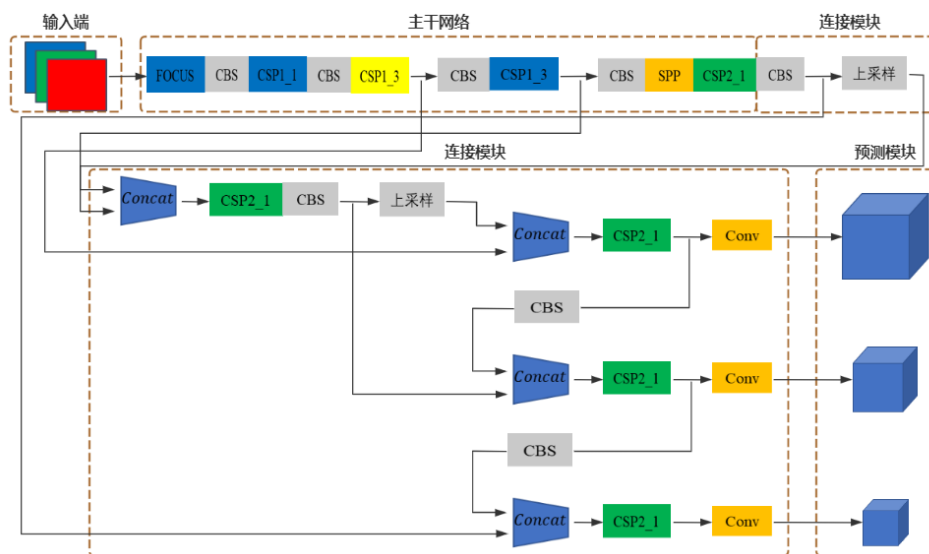


图 3-1 YOLOv5s 网络结构示意图

上图中包括重点模块，如 CBS、CSP1\_X、CSP2\_X、SPP、FOCUS 等模块详细结构图如 3-2 所示：

**CBS 模块：**在 YOLOv5s 中，CBS 模块代表的是 Convolution-BatchNorm-SiLU（卷积-批量归一化-自适应激活函数）模块。这是一个常见的组合，用于构建深度卷积神经网络的基本结构。

**ResUnit 模块（残差单元）：**是一种基于残差连接的网络结构，用于提高神经网络的性能和训练稳定性。

**CSP 模块：**该模块是模型主干网络的一部分。CSP 模块基于 CSPNet 的设计，该网络在传统的残差结构上进行了改进，通过将特征图分为两部分并在不同层级之间进行信息交流，提高特征表示的丰富性和网络性能。

**SPP 模块：**SPP 模块指的是空间金字塔池化模块。SPP 模块用于提取多尺度的特征表示，使网络能够更好地处理具有不同尺寸的目标。空间金字塔池化通过在不同尺度上进行池化操作来获得这些特征表示。

**FOCUS 模块：**FOCUS 模块是一个创新的网络结构，位于模型的输入部分。它的主要目的是从原始输入图像中提取有用的局部特征，减小计算复杂度，同时保持重要的空间。

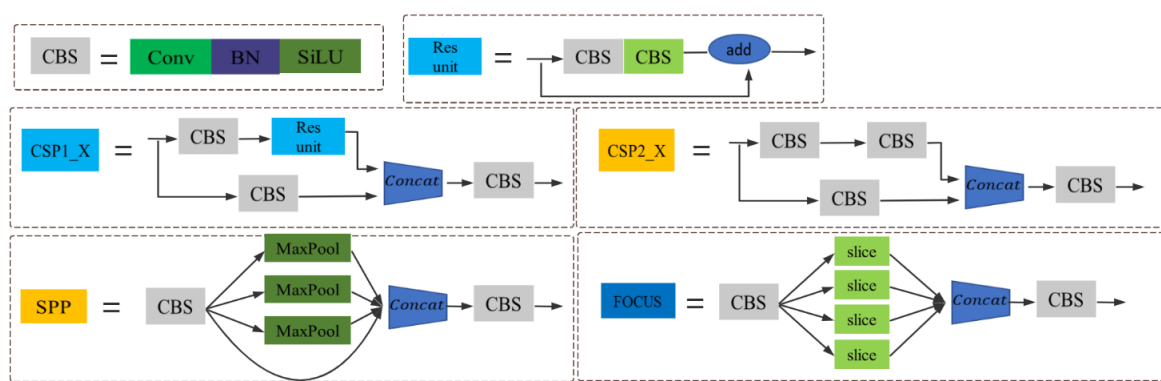


图 3-2 重点模块结构示意图

### 3.2.2 数据集准备与划分

由于本课题所涉及的检测目标为物流万向桌面上的快递包裹，目前网络上并没有完全契合的开源数据集。因此本课题所使用的数据集均为自己创建标注的数据集。相关数据集准备流程主要如下：

1) 收集图像：在尽量确保图像包含各种场景、背景和目标尺寸的前提下，本课题收集的快递包裹图像主要分为两类：静态状态下的不同尺寸快递包装箱及传送带等动态场景下的快递运输，如图 3-3 所示。以达到提高模型的泛化能



力的目的。



图 3-3 所收集的快递包裹图像

2) 标注图像：对于每个图像，需要标注图像中的目标对象。使用矩形边界框（bounding box）来表示目标对象的位置。对于每个边界框，还需要提供一个类别标签，表示所包含对象的类别。本课题使用标注工具为 LabelImg，如图 3-4 所示。

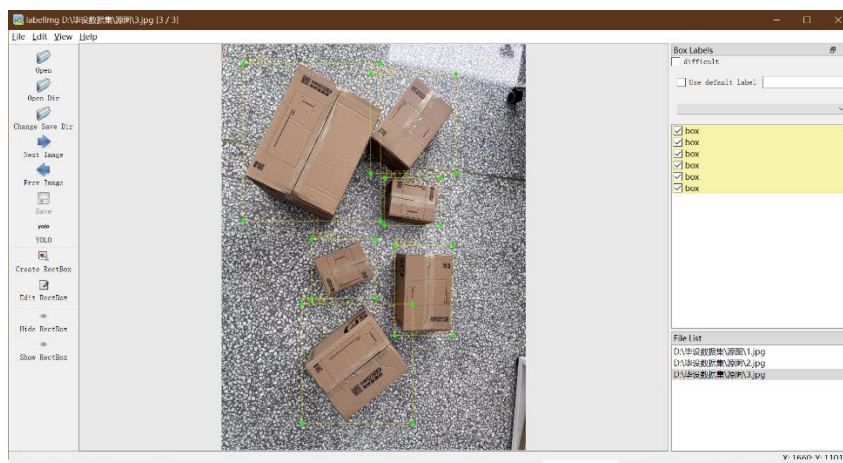


图 3-4 LabelImg 标注界面

3) 划分数据集：将收集到的图像数据集分为训练集、验证集和测试集。训练集用于训练模型，验证集用于调整模型的超参数，测试集用于评估模型的性能。本课题共收集 300 张图像，其中包括 100 张静态包裹和 200 张动态包裹。由于本课题最终的实验场景是动态环境，所以采用部分动态包裹作为验证集&测试集。

表 3-1 快递包裹检测数据集

数据类型	目标类别	数量	百分比	数据集类别
静态包裹	box	100	33.3%	训练集
动态包裹 1	box	120	40.0%	训练集
动态包裹 2	box	80	26.7%	验证集&测试集



4) 转换标注格式: YOLOv5 要求将图像标注转换为特定的格式。对于每个图像, 需要创建一个与图像同名的文本文件(使用.txt 扩展名), 其中包含信息: `<class_id> <x_center> <y_center> <width> <height>`。其中, 所有坐标和尺寸都需要归一化(除以图像的宽度和高度), `<class_id>` 是类别标签的整数索引。一个图像中可能包含多个边界框, 每个边界框应该占据一行。

5) 创建数据清单: 创建一个包含训练集和验证集图像路径的数据清单(.yaml 文件)。这个文件将告诉 YOLOv5 在训练和验证过程中使用哪些图像。

### 3.3 基于 YOLOv5 的算法改进

YOLOv5 作为一种高效和实时的目标检测方法, 在处理目标检测任务时具有显著优势。然而, 它在定位准确性、小目标检测能力以及锚框选择方面仍然存在一定的不足。为了在物流万向桌面背景下的目标检测任务中获得更好的性能, 我们需要对 YOLOv5 进行针对性的改进和优化。本节将从四个方面提出改进策略, 包括对非极大值抑制(NMS)算法的优化、损失函数调整、特征融合网络改进以及主干网络的轻量化设计。

首先, 针对后处理技术, 我们将对非极大值抑制(NMS)算法进行改进, 使其在处理相邻目标时具有更高的鲁棒性。接下来, 在损失函数方面, 我们将尝试调整损失函数, 以便更加关注小目标检测和准确的目标定位。此外, 为了提高特征提取和融合能力, 我们将探索新的特征融合网络结构, 从而在复杂背景下实现更准确的目标检测。最后, 通过对主干网络进行剪枝和轻量化处理, 我们将进一步提高 YOLOv5 的计算效率。

在实现这些改进策略后, 我们将根据实验结果评估各种改进方法的有效性。最终, 我们将选择对物流背景目标检测任务最具适应性的 YOLOv5 改进算法, 以在实际应用中实现更优的检测性能。

#### 3.3.1 非极大值抑制 NMS 算法改进

YOLOv5s 中的非极大值抑制(NMS)算法是一种关键的后处理技术, 旨在降低重叠边界框的数量并保留具有高置信度的预测结果。该算法在消除冗余检测结果的同时, 确保了目标检测的准确性和鲁棒性。

NMS 算法在 YOLOv5s 中的主要步骤如下:

- 1) 将所有边界框按照置信度降序排列。
- 2) 从置信度最高的边界框开始, 遍历所有尚未被抑制的边界框。对于每个当前边界框:

- a. 将当前边界框标记为保留。
- b. 计算当前边界框与其他所有未被抑制的边界框的 IoU（交并比）。
- c. 如果与当前边界框的 IoU 值高于预先设定的阈值，则将其其他边界框标记为抑制，因为它们与当前边界框重叠过多。

3) 继续遍历其他未被抑制的边界框，直至所有边界框均被处理。

经过 NMS 算法处理后会删除其他无用的目标检测框，留下一个最优的目标检测框，具体效果如下图 3-5 所示。

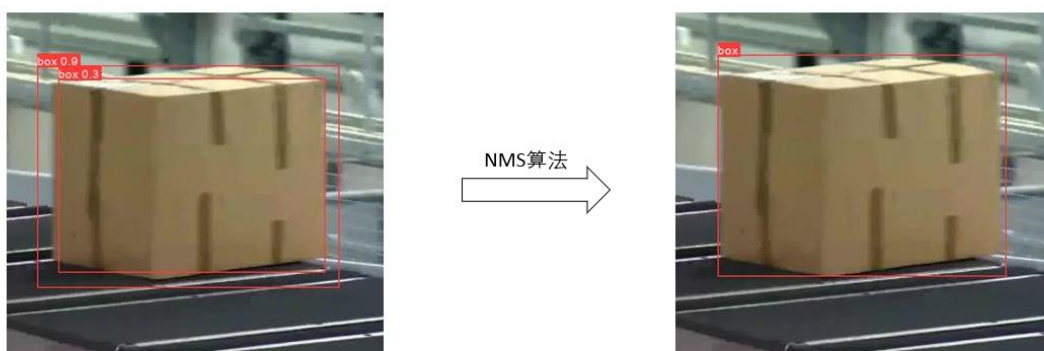


图 3-5 NMS 算法效果

尽管 YOLOv5s 中的 NMS 算法在很多情况下表现良好，但在处理相邻目标时，可能会过于严格地抑制边界框。针对这一问题，将略显粗暴的 NMS 算法改进为 Soft-NMS 算法。通过调整重叠边界框的置信度而非直接删除它们，以提高相邻目标的检测性能。具体实现步骤如下：

- 1) 将所有边界框按照置信度降序排列。
- 2) 从置信度最高的边界框开始，遍历所有尚未被抑制的边界框。对于每个当前边界框：
  - a. 将当前边界框标记为保留。
  - b. 计算当前边界框与其他所有未被抑制的边界框的 IoU（交并比）。
  - c. 根据 IoU 值调整其他边界框的置信度。通常，我们使用下面的公式 3-1 或公式 3-2 进行调整：

$$\text{new}_s\text{core} = \text{original}_s\text{core} * (1 - IoU) \quad (3-1)$$

$$\text{new}_s\text{core} = \text{original}_s\text{core} * \exp(-IoU^2/\sigma^2) \quad (3-2)$$

式中  $\sigma$  ——可调参数。

- 3) 设定一个置信度阈值，将调整后的置信度低于该阈值的边界框删除。
- 4) 继续遍历其他未被抑制的边界框，直至所有边界框均被处理。

Soft-NMS 算法的具体实现的伪代码可以在图 3-6 中找到。

---

**Algorithm 4** Soft-NMS Algorithm

---

```

1: procedure SOFTNMS( $D, \sigma, N_t, s_t$ )
2:    $D \leftarrow$  sort  $D$  by scores in descending order
3:   for  $i = 1$  to  $|D|$  do
4:      $M \leftarrow D[i]$ 
5:     if SCORE( $M$ ) <  $s_t$  then
6:       continue
7:     end if
8:     for  $j = i + 1$  to  $|D|$  do
9:        $N \leftarrow D[j]$ 
10:       $IoU \leftarrow IoU(M, N)$ 
11:      if  $IoU > N_t$  then
12:        SCORE( $N$ )  $\leftarrow$  SCORE( $N$ ) *  $(1 - IoU)$ 
13:      else
14:        SCORE( $N$ )  $\leftarrow$  SCORE( $N$ ) *  $\exp(-IoU^2/\sigma^2)$ 
15:      end if
16:    end for
17:  end for
18:  Remove boxes with scores less than  $s_t$  from  $D$ 
19:  return  $D$ 
20: end procedure

```

---

图 3-6 Soft-NMS 算法实现

### 3.3.2 损失函数改进

YOLOv5 损失函数由若干部分构成，旨在评估预测性能的不同方面。总的来说，YOLOv5 损失函数主要包含以下三个部分：

**边界框回归损失（Box Regression Loss）：**用于量化预测边界框与实际边界框之间的差距。YOLOv5 采用 CIOU（Complete Intersection over Union）损失，它综合了 IOU、中心点距离和宽高比等多种因素，全面评估预测边界框与实际边界框之间的偏差。

**目标可信度损失（Objectness Loss）：**用于衡量预测目标置信度与实际目标置信度之间的误差。这一损失有助于模型识别目标所在区域。YOLOv5 通常使用二元交叉熵损失（Binary Cross-Entropy Loss）来计算目标可信度损失。

**类别损失（Classification Loss）：**用于衡量预测类别概率与实际类别概率之间的误差。这一损失有助于模型对目标进行正确分类。YOLOv5 通常使用交叉熵损失（Cross-Entropy Loss）来计算类别损失。

YOLOv5 的整体损失函数是这三部分损失的加权总和。通过优化该损失函数，YOLOv5 能够实现高效且准确的目标检测任务学习。在本节中主要是通过改进边界框回归损失函数 CIOU 优化目标检测结果。

CIOU 损失是一种更全面的损失函数，旨在解决传统 IOU 损失在某些情况下的不足。CIOU 损失考虑了边界框之间的三个关键因素：重叠区域（IOU）、中心点距离和宽高比。与传统的 IOU 损失相比，CIOU 损失可以更好地衡量预测边界框与实际边界框之间的差异。以下是 CIOU 损失的主要组成部分：

(1) 交并比（IOU）：衡量预测边界框和实际边界框重叠区域的比例。IOU 越大，两个边界框的重叠程度越高。

(2) 中心点距离：衡量预测边界框和实际边界框中心点之间的欧氏距离。中心点距离越小，两个边界框的位置越接近。

(3) 宽高比：衡量预测边界框和实际边界框的宽高比之间的差异。宽高比越接近，两个边界框的形状越相似。

CIOU 损失将这三个因素综合考虑，计算公式如式 3-3：

$$L_{CIOU} = 1 - IOU + d_c^2/c^2 + \alpha \cdot v \quad (3-3)$$

式中  $d_c$ ——预测边界框和实际边界框中心点之间的欧氏距离；

$c$ ——对角线距离，即包含两个边界框的最小矩形的对角线长度；

$v$ ——宽高比误差；

$\alpha$ ——权重参数，用于平衡宽高比误差的影响。

然而，CIOU 损失函数仍存在一些局限性。如对角线距离惩罚，CIOU 损失中包含一个对角线距离项，以惩罚中心点偏移。然而，这种惩罚方式可能对一些情况下的边界框预测效果不佳，例如密集目标检测，在本课题背景中可能出现密集包裹的情况。同时 CIOU 中所涉及的宽高比是相对值，可能会阻碍模型有效的优化相似性。因此将 YOLOv5 的损失函数改进为 EIOU。EIOU 损失的具体计算公式如式 3-4 所示，从公式中可以看到 EIOU 损失函数由三个组成部分构成：预测边界框与实际边界框的重叠损失、中心点距离损失以及宽度和高度损失。EIOU 损失继承了 CIOU 损失中的前两个部分，然而，它针对宽度和高度损失进行了改进。这一改进通过最小化预测边界框与实际边界框之间宽度和高度的差值，进而加速模型收敛速度。

$$L_{EIOU} = 1 - IOU + d_c^2/c^2 + w_c^2/c_w^2 + h_c^2/c_h^2 \quad (3-4)$$

式中  $w_c$ ——预测边界框和实际边界框的宽度之间的欧氏距离；

$h_c$ ——预测边界框和实际边界框的高度之间的欧氏距离；

$c_w$ ——覆盖两个框的最小外接框的宽度；

$c_h$ ——覆盖两个框的最小外接框的高度。

### 3.3.3 特征融合网络改进

YOLOv5 的特征整合网络采用了基于 PANet (Path Aggregation Network) 的架构。PANet 的目标在于高效地整合多尺度特征，从而提高对各种尺寸目标的检测能力。PANet 的核心概念通过自顶向下的通道和侧向连接实现了多尺度特征的融合。自顶向下的通道是指网络从高层特征图传递到低层特征图的方向，以获取更多丰富的语义信息。而侧向连接负责将来自同一层次的特征图整合在一起，从而实现多尺度信息的融合。YOLOv5 的 PANet 架构包含以下几个关键部分：

1) 自顶向下通道：自顶向下通道通过卷积层和上采样操作，将高层特征图传输到低层。这有助于低层特征图获得更丰富的语义信息，有利于小物体的检测。

2) 侧向连接：侧向连接在同一层次的特征图之间建立连接，实现特征整合。这有助于提高特征表示能力，从而检测不同尺度的物体。

3) 逐元素相加和激活函数：自顶向下通道与侧向连接融合后的特征图通过逐元素相加操作，接着使用激活函数（例如 SiLU）进行非线性变换，以增强网络性能。

虽然 YOLOv5 中的 PANet 在特征融合方面取得了一定的成果，但它仍存在一些缺点。例如信息流的局限性，PANet 中，自顶向下路径和横向连接相互独立，可能导致信息流受限的问题。同时在融合效果、灵活性和扩展性方面 PANet 也有待提高的空间。因此在本节中引入了 BiFPN 加权双向金字塔结构，实现自上而下与自下而上的深浅层特征双向融合，增强不同网络层之间特征信息的传递，提升不同尺度的检测效果。BiFPN 的网络结构主要包括以下特点：

1) 双向传播——自顶向下与自底向上：

BiFPN 采用自顶向下和自底向上两种方向的传播策略。自顶向下路径将高层特征向低层特征传递，以便捕获更丰富的语义信息；而自底向上路径从低层特征向高层特征传递，以便捕获更详细的空间信息。这种双向传播结构使得各层级特征能够更有效地互相交流和整合。

## 2) 特征融合的加权方法:

BiFPN 在整合特征时采用加权策略。通过利用可学习的权重对不同层级的特征进行加权，模型可以在训练过程中自动调整各特征图的重要性。这种加权特征整合方法能够提高特征表示能力和计算效率。

## 3) 分层迭代整合:

BiFPN 将特征整合流程分为多个迭代层次，每一层次都包括自顶向下和自底向上的特征传递。通过多次迭代，特征整合过程变得更加充分和深入，从而提升最终特征表示的品质。

## 4) 残差连接的应用:

BiFPN 在特征整合过程中采用残差连接，有助于提高网络的训练稳定性和收敛速度。通过将原始特征图与整合后的特征图相加，残差连接可以更好地保留原始特征的信息。

相比原有的 PANet 结构，BiFPN 部分通过移除只具有单一输入边的节点来简化网络结构。如果原始输入节点和输出节点位于相同层级，将在原始输入节点和输出节点之间增加一条额外连接；同时处理每个双向路径（自顶向下和自底向上）作为一个特征网络层，并在同一层中多次重复，以实现更高级别的特征整合。同时，学习不同输入特征的重要性，实现对各输入特征的区别性融合，具体结构对比如图 3-7 所示。

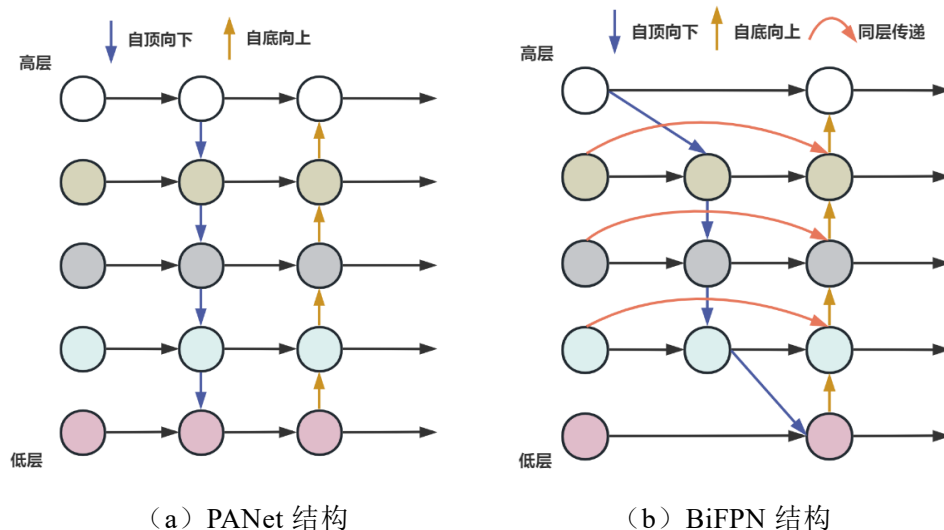


图 3-7 PANet 与 BiFPN 结构对比

### 3.3.4 主干网络轻量化

在 YOLOv5 的主干网络中，C3 模块作为一个包含 3 个连续卷积层的组件，

负责执行特征提取任务。C3 模块的构成如下：

**第一卷积层：**这一卷积层处理输入特征图，其卷积核数量、尺寸和步长因场景而异。紧接着该卷积层是一个 Batch Normalization (BN) 层和一个激活函数层（例如 SiLU）。

**第二卷积层：**与第一卷积层相似，第二个卷积层负责处理第一个卷积层生成的输出特征图。这个卷积层也包括一个 Batch Normalization (BN) 层和一个激活函数层（例如 SiLU）。

**第三卷积层：**这个卷积层处理第二个卷积层产生的输出特征图，接下来是一个 Batch Normalization (BN) 层和一个激活函数层（如 SiLU）。

C3 模块旨在从输入特征图中提取有价值的特征，这些特征将应用于后续的目标检测步骤。在 YOLOv5 的主干网络中，C3 模块与其他模块（如 CSP、ResUnit 等）共同形成一个高效的特征提取器。

C3 结构作为 YOLOv5 中的特征提取组件，虽然相对简单且易于实现，但仍存在一些潜在的缺点，例如网络容量问题。随着卷积层数量的增加，计算复杂性和参数数量相应增加，可能导致更长的训练时间和更高的资源消耗。在资源受限的设备上，这可能成为一个问题。梯度消失/爆炸问题，尽管 C3 结构相对简单，但在训练深层神经网络时，梯度消失或梯度爆炸仍然可能发生。这可能影响网络的训练稳定性和收敛速度。本课题所面对的场景是智慧物流桌面，对实时性要求较高，而 C3 结构会带来较大的参数量。会引起检测速度较慢，应用受限等问题。在本课题所面临的真实场景有嵌入式设备，有如此大而复杂的模型时难以被应用的。因此在本节中，将主干特征提取网络替换为更轻量的 PP-LCNet 网络，以实现网络模型的轻量化，平衡速度和精度。

PP-LCNet (PaddlePaddle-Lite-Computer-Vision-Network) 是一个轻量级神经网络。PP-LCNet 的主要目标是在保持较高精度的同时降低模型大小和计算复杂度，以满足移动端和边缘设备上的实时计算需求。它使用渐进式解析 (Progressively Parsing) 和轻型复合网络结构 (Light Compound Network) 来实现高效的特征提取和特征融合。PP-LCNet 相比 C3 结构的差异可以概括为以下两个方面：

1) 逐步解析：在 PP-LCNet 中，逐步解析策略用于逐级分析输入图像的信息。通过逐渐解析图像，网络能够更有效地关注不同规模的目标，同时降低计算复杂性。

2) 轻量化复合网络架构：PP-LCNet 采用一种轻量级的复合网络结构，通过结合基本的卷积操作和注意力机制进行特征提取。这种结构在保持较低计算量

的同时，有效地提取特征，适应不同规模的目标检测任务。

PP-LCNet 替换为 YOLOv5 主干网络需要实现的算法流程的伪代码可以在图 3-8 中找到。

---

**Algorithm 5** PP-LCNet for Object Detection

---

```

1: Input: Input image  $I$ , anchor boxes  $A$ 
2: Output: Detected objects with class labels and bounding boxes
3: Step 1: Preprocess input image  $I$ 
4: Step 2: Perform feature extraction using Lightweight Compound Network (LCNet)
5:  $F \leftarrow \text{LCNet}(I)$ 
6: Step 3: Implement step-by-step parsing to refine the features
7: for each scale  $s$  in  $F$  do
8:    $F_s \leftarrow \text{StepParsing}(F_s)$ 
9: end for
10: Step 4: Perform multi-scale feature fusion
11:  $F_{\text{fused}} \leftarrow \text{FeatureFusion}(F)$ 
12: Step 5: Predict class probabilities, objectness scores, and bounding box regression parameters
13:  $P_{\text{class}}, P_{\text{obj}}, R \leftarrow \text{PredictionHead}(F_{\text{fused}}, A)$ 
14: Step 6: Post-process predictions
15:  $D \leftarrow \text{PostProcess}(P_{\text{class}}, P_{\text{obj}}, R, A)$ 
16: Step 7: Perform non-maximum suppression (NMS)
17:  $D_{\text{final}} \leftarrow \text{NMS}(D)$ 
    return  $D_{\text{final}}$ 

```

---

图 3-8 PP-LCNet 目标检测算法流程

## 3.4 实验结果与对比评价

### 3.4.1 目标检测评价指标

目标检测性能评价主要涵盖两个关键方面，分别为检测精确性和检测速度。对这两个方面的评估有助于全面地了解目标检测模型的表现。

检测精确性用于衡量模型在正确识别和定位目标上的表现，常见的指标包括精确率（Precision）、召回率（Recall）、F1 分数（F1 score）、IoU（Intersection over Union）、P-R 曲线（Precision-Recall curve）、AP（Average Precision）和 mAP（mean Average Precision）等。这些指标从多个角度评估模型的准确性，如预测边界框与真实边界框的重叠程度、错误报告和遗漏报告的数量等。在实际应用中，通常需要综合考虑这些指标以评价模型的检测精确性。

另一方面，检测速度关注模型在处理图像和执行目标检测任务所需的时间，可以通过帧率（FPS, Frames Per Second）、前传耗时、浮点运算量来衡量。帧率越高，意味着模型的检测速度越快。在本课题所面对的物流实时目标检测场景



中，检测速度至关重要，因为需要在短时间内处理大量图像。

在本课题中主要关注精确度指标精确率(Precision)、召回率(Recall)、mAP<sub>0.5</sub>和速度指标 FPS。

**精确率 (Precision):** 在目标检测任务中，精确率衡量了在所有检测到的正样本中，确实为正样本的占比。精确率越高，意味着模型在预测结果中误报的负样本数量更少

**召回率 (Recall):** 召回率表示在所有实际正样本中，被成功检测出来的比例。召回率越高，说明模型在检测正样本方面表现更优，遗漏的正样本数量较少。

**mAP** 是所有类别 AP 的平均值，通常作为目标检测任务的主要评价指标。较高的 mAP 值表示模型在各个类别的整体性能更佳。由于本课题只有一个 box 类别，所以 mAP 等于 AP。而 AP 是由 Precision 和 Recall 计算得出，具体为在不同置信度阈值下，模型的精确率和召回率之间曲线的面积。本课题中最终使用的 mAP<sub>0.5</sub> 指标表示 IoU 阈值设置为 0.5 时的 mAP 值。

**FPS** 表示每秒处理的帧数，用于衡量目标检测算法的实时性。较高的 FPS 值表示算法能更快地处理图像，更适用于实时目标检测场景。

### 3.4.2 模型改进结果对比

本课题的 YOLOv5 目标检测网络基于 Pytorch 深度学习框架完成。所涉及的计算机参数包括：操作系统——Ubuntu20.04，显卡——RTX3070，显存——8GB。主要的训练参数设置如表 3-2 所示。

表 3-2 YOLO v5s 训练参数

参数	值
Batch size	10
Epoch	48/200
Optimizer	Adam

其中训练周期参数在特征融合网络改进和主干网络轻量化时设置为 200，由于主干网络有所改动，在 200 轮 Epoch 时网络模型才达到收敛。本章中共尝试了四种模型改进，包括 NMS 算法改进为 Soft-NMS 算法、损失函数改进为 EIOU、特征融合网络改为 BiFPN、主干网络轻量化为 PP-LCNet。四种模型改进及原 YOLOv5s 的目标检测结果如表 3-3 所示。

表 3-3 模型改进结果对比

网络模型	精确率	召回率	mAP <sub>0.5</sub>	FPS
YOLOv5s	0.9108	0.88362	0.95369	125.519
Soft-NMS	0.9063	0.85897	0.93849	79.656
EIOU	0.93524	0.84615	0.94397	122.365
BiFPN	0.97089	0.91987	0.98351	97.623
PP-LCNet	0.97182	0.95513	0.98841	144.55

通过表 3-3 的结果，可以分析出以下的相关结论：

1) 将 NMS 算法改进为 Soft-NMS 算法针对本课题的物流包裹目标检测效果较差。由于本课题只有 box 一个类别，虽然 NMS 算法作为后处理技术略微粗暴，但对于单目标还是取得了较好的效果。而 Soft-NMS 算法虽然提高了相邻目标的检测性能，但是由于快递包裹相邻较近的情况比较少，所以在 mAP<sub>0.5</sub> 这一指标并没有得到提升。同时还牺牲了部分后处理时间，导致 FPS 大幅降低。

2) 将损失函数改进为 EIOU 针对本课题的物流包裹目标检测效果区别不大。YOLOv5 原本的损失函数为 CIOU，已经是一种较为全面的损失函数。虽然 EIOU 损失函数在收敛速度、模型有效的优化相似性方面均有提升，但是对于本课题的背景提升效果并不明显。

3) 将特征融合网络改为 BiFPN 针对本课题的物流包裹目标检测效果精确度提高较大，同时速度指标有所降低。由于 BiFPN 是加权双向金字塔结构，引入该结构后显著提高精确率、mAP<sub>0.5</sub> 等精确度指标。同时由于 BiFPN 每个双向路径（自顶向下和自底向上）都作为一个特征网络层，并在同一层中多次重复，使得 FPS 大幅下降。考虑到本课题对于实时性要求较高，最终并未采用该网络模型优化。

4) 将主干网络 C3 轻量化为 PP-LCNet 针对本课题的物流包裹目标检测效果精确度及速度都有显著提高。PP-LCNet 作为一个轻量级网络，主要目标是在保持较高精度的同时降低模型大小和计算复杂度，十分契合本课题的背景诉求。通过评价指标可以看到将 C3 网络替换为 PP-LCNet 后，精确度和速度都有大幅提升。其中 mAP<sub>0.5</sub> 达到了 0.98841，基本不会出现错误检测、遗漏检测的情况，部分快递包裹检测情况如图 3-9 所示。

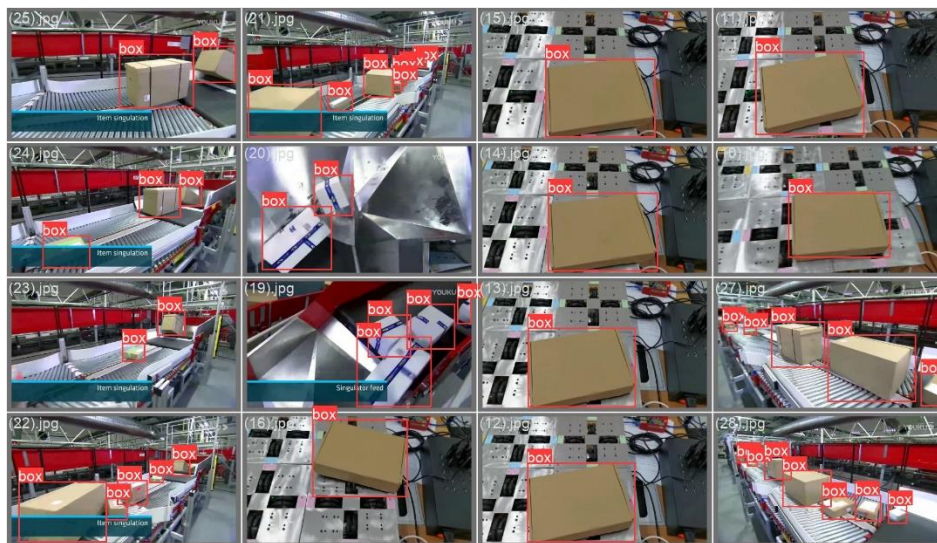


图 3-9 PP-LCNet 改进 YOLOv5 目标检测算法效果

### 3.5 本章小结

本章主要研究了基于 YOLOv5 的目标检测算法改进。首先,通过对 YOLOv5 目标检测算法进行深入研究,探讨了 YOLOv5 目标检测网络的结构及其相关的数据集准备与划分方法。接着,本章提出了针对 YOLOv5 的算法改进策略,涉及非极大值抑制 NMS 算法改进、损失函数改进、特征融合网络改进以及主干网络轻量化等方面。

针对非极大值抑制 NMS 算法改进,本文提出了一种更有效的策略以提高模型的检测性能。同时,通过改进损失函数,本文提出了一种更加适用于目标检测任务的损失函数形式。此外,为了更好地提取特征并适应不同尺度的目标检测任务,本文还对特征融合网络进行了改进。最后,通过对主干网络进行轻量化处理,本文旨在实现在保持较高检测精度的同时,降低模型的计算复杂度和提高检测速度。

为了评估改进策略的有效性,本章设计了一系列实验,并采用了常用的目标检测评价指标进行模型性能评估。实验结果表明,改进后的模型在检测精度和速度方面均取得了显著的提升。通过对比评价,本文证实了基于 YOLOv5 的目标检测算法改进在本课题的物流万向桌面中具有较高的可行性和有效性。

## 第4章 目标跟踪与包裹重识别算法

### 4.1 引言

在第3章中,我们对基于YOLOv5的目标检测算法进行了深入研究和改进,提高了模型的检测精度和速度。本章将在此基础上,进一步探讨目标跟踪与包裹重识别算法的实现。目标跟踪和包裹重识别是目标检测任务中的重要组成部分,对于实时监控和快递自动分拣等场景具有重要的应用价值。

首先,本章将利用改进后的YOLOv5s作为目标检测算法,结合ByteTrack,实现快递目标的跟踪。ByteTrack是一种高效的目标跟踪算法,通过与YOLOv5s相结合,可以实现对快递目标的实时跟踪。在4.2节中,我们将详细介绍基于ByteTrack的快递目标跟踪算法的原理和实现方法。

接下来,本章将探讨基于运动建模的包裹重识别算法。该算法通过分析包裹的运动特征,同时根据外观相似度进行辅助验证,实现对同一物体的重复检测和识别。这对于解决目标检测及跟踪中的遮挡问题 and 提高目标识别的准确性具有重要意义。在4.3节中,我们将详细阐述基于运动建模的包裹重识别算法的原理及其实现过程。

本章旨在深入研究目标跟踪与包裹重识别算法,为智慧物流桌面提供更为高效且准确的目标检测、跟踪与识别方案。

### 4.2 基于ByteTrack的快递目标跟踪算法

在本课题的智慧物流万向桌面系统中,每个快递包裹在桌面上运输的过程中都需要进行实时监控和跟踪,并赋予一个唯一的标识符(ID)。这样可以确保物流过程中包裹的有效管理和实时追踪。因此,在成功检测到快递包裹目标之后,仍需要与目标跟踪算法相结合。配合目标跟踪算法不仅可以提高包裹管理的准确性和效率,还能实时监控包裹的状态,预警潜在的风险,例如丢失、滞留或错误分拣。

ByteTrack是一种基于跟踪-检测范式的跟踪方法,提出了一种简洁且高效的数据关联策略。与先前的跟踪算法最大的不同之处在于,它并不是简单地忽略低分数的检测结果。相反,ByteTrack利用检测框与跟踪轨迹之间的相似性,在保留高分数检测结果的同时,从低分数检测结果中去除背景噪声,并发掘出实际目标(如遮挡、模糊等复杂样本)。这样可以减少漏检并提高轨迹的连续性。

由于 ByteTrack 没有采用外观特征进行匹配, 因此跟踪效果在很大程度上依赖于检测效果。换句话说, 如果检测器效果良好, 则跟踪效果也将相应提高。然而, 如果检测效果不佳, 跟踪效果可能会受到严重影响。目前, 在本课题的改进下 YOLOv5s 目标检测部分的 mAP 达到了 0.98, 这为 ByteTrack 多目标跟踪算法提供了良好的基础, 从而实现了优秀的跟踪结果。

该算法的主要思路就是创建追踪的轨迹, 然后利用追踪的轨迹对每一帧的快递包裹目标进行匹配, 通过数据关联来逐帧匹配目标, 从而形成完整的轨迹, 算法流程图如图 4-1 所示。

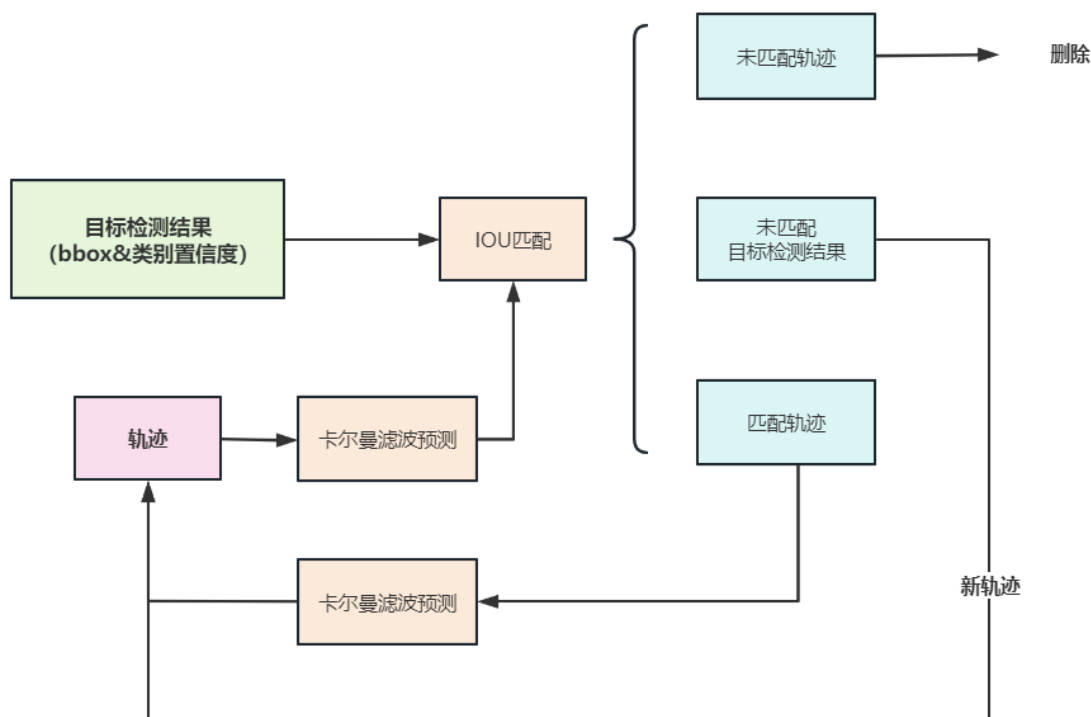


图 4-1 ByteTrack 算法流程图

#### 4.2.1 轨迹状态定义

跟踪轨迹的状态主要可以分为以下六类:

- 1) 激活状态: 指的是已经成功跟踪至少两帧的目标边界框, 包括刚刚创建的轨迹在其初始帧时的目标边界框。
- 2) 未激活状态: 对于在视频序列中间出现的新轨迹, 若暂时还没有找到与其匹配的第二个点, 则将其视为未激活状态。
- 3) 新轨迹: 表示刚刚生成的轨迹, 尚未与其他轨迹进行关联。
- 4) 已追踪轨迹: 这是在前一帧中成功跟踪到的轨迹。

5) 失追轨迹：如果在前  $n$  帧中 ( $n \leq 30$ ) 跟踪失败，这种轨迹会被标记为失追轨迹。

6) 已删除轨迹：如果在前  $n$  帧中 ( $n > 30$ ) 跟踪失败，这种轨迹将被视为已删除轨迹。

这六种状态反映了目标跟踪过程中轨迹的不同阶段和条件，有助于了解跟踪算法的性能表现以及潜在的问题。

#### 4.2.2 轨迹构建与追踪

在初始阶段，当开始处理第一帧图像时，目前还没有任何轨迹产生。此时，算法会针对所有检测到的目标边界框创建对应的轨迹对象，并将其存储起来。需要注意的是，在这个阶段，所有新建的轨迹都将被标记为已追踪轨迹。从第二帧图像开始，算法逐渐构建轨迹，主要步骤如下：

##### 1) 对追踪轨迹和边界框进行分类

所有的追踪轨迹可以分为激活和未激活两类（激活状态指至少追踪了两帧以上的目标框，包括第一帧时新建的轨迹）。同样，当前帧中的边界框可以根据得分阈值（默认为 0.5）分为高分和低分两类。

##### 2) 第一次追踪：针对激活状态轨迹的高分匹配

- a. 将所有已追踪轨迹和失追轨迹合并为初步追踪轨迹。
- b. 使用卡尔曼滤波预测初步追踪轨迹的下一帧边界框可能的位置和大小。
- c. 计算预测的边界框与当前帧高分边界框之间的 IoU 值，构建一个关系损失矩阵。

d. 利用匈牙利算法对初步追踪轨迹和当前帧高分边界框进行匹配，获得已匹配的轨迹与边界框、未成功匹配的轨迹和未成功匹配的当前帧边界框。

e. 最后，用已成功匹配的当前帧边界框更新初步追踪轨迹。

##### 3) 第二次追踪：针对激活状态轨迹的低分匹配

a. 筛选出第一次匹配中未成功匹配的已追踪轨迹。因为这些轨迹之前已经预测过下一帧的边界框，所以不需要再次预测。

b. 计算这些轨迹与当前帧低分边界框之间的 IoU 值，再次使用匈牙利算法进行匹配。

c. 用已成功匹配的当前帧边界框更新这些追踪轨迹。

d. 将未成功追踪的轨迹标记为失追轨迹。

##### 4) 对未激活状态的轨迹进行追踪

a. 筛选出第一步中未成功匹配的当前帧高分边界框，以及未激活的轨迹。

b. 计算这些轨迹与当前帧边界框之间的 IoU 值，并使用匈牙利算法进行匹配。

c. 用已成功匹配的当前帧边界框更新这些追踪轨迹。

d. 未成功追踪的未激活轨迹将被直接标记为已删除轨迹。

#### 5) 创建新轨迹

若此刻仍有未成功匹配的高分边界框，则可将其视为新出现的目标。为此，系统会分配一个新的轨迹和 ID（低分边界框将被视为误判而直接丢弃，不会生成新轨迹）。在此阶段，已不再处理第一帧。新增轨迹均处于未激活状态，若在下一帧中仍无法成功匹配，将会被直接删除。

#### 6) 输出结果

至此，可以返回所有已追踪的轨迹（不包括失追轨迹和已删除轨迹）。所有轨迹都具有唯一的 ID，此结果可用作每一帧追踪结果的依据。

ByteTrack 算法的轨迹构建与追踪具体实现的伪代码可以在图 4-2 中找到。

---

#### Algorithm 6 目标追踪算法

---

**Input:** 当前帧边界框, 已追踪轨迹

---

**Output:** 所有已追踪的轨迹

- 1: **1. 对追踪轨迹和边界框进行分类**
  - 2:   激活和未激活追踪轨迹
  - 3:   高分和低分边界框
  - 4: **2. 对轨迹进行第一次追踪（仅针对激活状态的轨迹的高分匹配）**
  - 5:   预测初步追踪轨迹的下一帧边界框
  - 6:   计算 IoU 损失矩阵
  - 7:   使用匈牙利算法进行匹配
  - 8:   更新初步追踪轨迹
  - 9: **3. 对轨迹进行第二次追踪（仅针对激活状态的轨迹的低分匹配）**
  - 10:   筛选出未匹配的已追踪轨迹
  - 11:   计算 IoU
  - 12:   使用匈牙利算法进行匹配
  - 13:   更新追踪轨迹
  - 14:   标记未成功追踪的轨迹为失追轨迹
  - 15: **4. 对未激活状态的轨迹进行追踪**
  - 16:   计算未成功匹配的当前帧边界框与未激活轨迹的 IoU
  - 17:   使用匈牙利算法进行匹配
  - 18:   更新追踪轨迹
  - 19:   标记未成功追踪的未激活轨迹为已删除轨迹
  - 20: **5. 新建轨迹**
  - 21:   分配新的轨迹和 ID
  - 22: **6. 返回结果**
  - 23:   返回所有已追踪的轨迹
- 

图 4-2 轨迹构建与追踪



### 4.2.3 数据关联

在上述轨迹构建与追踪部分中，涉及到了一些关于数据关联的算法。数据关联过程主要可以分为两个阶段：相似性计算和目标匹配。首先，在相似性计算阶段，我们需要考虑位置信息（通过位置重合度 IOU 来衡量）、运动模型和表观特征模型。ByteTrack 算法主要侧重于结合位置信息和运动模型，它采用卡尔曼滤波器来预测目标在下一帧中的位置。另一方面，目标匹配阶段在 ByteTrack 算法中主要采用匈牙利匹配算法。匈牙利算法是一种高效的数据关联技术，能够在多目标跟踪任务中有效地将目标与观测数据进行匹配。通过这种方法，可以确保在每个时间步长中，目标与其对应的观测数据能够最佳匹配，从而提高轨迹跟踪的准确性和稳定性。

ByteTrack 算法中采用的卡尔曼滤波（Kalman Filter）是一种线性动态系统的递归最佳估计方法，用于估计受噪声影响的系统状态。在 ByteTrack 应用场景下，卡尔曼滤波器主要用以预测目标在接下来的帧中的位置。卡尔曼滤波器的基本原理包含两个阶段：预测和更新。在预测阶段，根据先前的状态以及系统的动态模型对系统状态进行估计。在预测过程中，卡尔曼滤波器采用如下两个公式 4-1，4-2：

$$\mathbf{x}_{\text{pred}} = \mathbf{F} * \mathbf{x} \quad (4-1)$$

$$\mathbf{P}_{\text{pred}} = \mathbf{F} * \mathbf{P} * \mathbf{F}^T + \mathbf{Q} \quad (4-2)$$

式中  $\mathbf{x}$ ——目标的位置和速度；

$\mathbf{F}$ ——状态转移矩阵；

$\mathbf{P}$ ——协方差矩阵；

$\mathbf{Q}$ ——过程噪声矩阵。

更新阶段通过观测数据对预测结果进行校正，从而获得更精确的状态估计。这个阶段同样使用如下公式 4-3，4-4，4-5 所示：

$$\mathbf{K} = \mathbf{P}_{\text{pred}} * \mathbf{H}^T * (\mathbf{H} * \mathbf{P}_{\text{pred}} * \mathbf{H}^T + \mathbf{R})^{-1} \quad (4-3)$$

$$\mathbf{x} = \mathbf{x}_{\text{pred}} + \mathbf{K} * (\mathbf{z} - \mathbf{H} * \mathbf{x}_{\text{pred}}) \quad (4-4)$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K} * \mathbf{H}) * \mathbf{P}_{\text{pred}} \quad (4-5)$$

式中  $\mathbf{K}$ ——卡尔曼增益；

$\mathbf{H}$ ——观测矩阵；

$\mathbf{R}$ ——观测噪声矩阵；

$\mathbf{z}$ ——实际观测值；

$\mathbf{I}$ ——单位矩阵。

ByteTrack 算法所采纳的匈牙利算法（Hungarian Algorithm）是一个组合优化算法，用于解决带权值二部图最优匹配问题。在 ByteTrack 应用场景下，匈牙利算法主要用于在连续视频帧间进行目标匹配，从而形成最终目标轨迹。匈牙利算法的核心原理在于寻找带权二部图中权重总和最小的匹配。在 ByteTrack 算法中，二部图的一部分代表上一帧的目标，另一部分代表当前帧的目标，边的权重表示两个目标的相似程度，通常使用 IoU 来表示。具体流程主要包括以下六步，如图 4-3 所示。

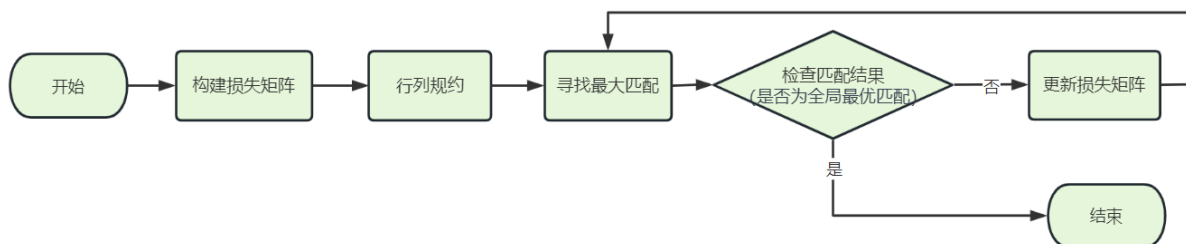


图 4-3 匈牙利算法流程

1) 构建损失矩阵：将二部图中的边权重组成为矩阵，即损失矩阵。在 ByteTrack 场景下，损失矩阵的元素表示上一帧目标与当前帧目标间的 IoU 值。

2) 行列规约：对损失矩阵的每行和每列分别减去其最小值，使矩阵元素规约至零。

3) 寻找最大匹配：在规约后的矩阵中找到一个全零子矩阵，该子矩阵包含的零元素数量最大，同时保证每行和每列最多包含一个零元素。

4) 检查匹配结果：如果找到的匹配已经是全局最优匹配，则算法结束。如果不是，继续执行下一步。

5) 更新损失矩阵：在损失矩阵中，减去未覆盖行的最小值，同时加上覆盖列的最小值。这样，在矩阵行、列规约后，匹配结果将更接近全局最优匹配。

6) 重复执行步骤 3-5，直到找到全局最优匹配。

匈牙利算法的在 ByteTrack 中的具体实现的伪代码可以在图 4-4 中找到。

---

**Algorithm 7** Hungarian Algorithm

---

```

1: function HUNGARIANALGORITHM(cost_matrix)
2:   Input: cost_matrix - an  $n \times m$  cost matrix
3:   Output: assignment - target assignment result

4:   Step 1: Row Reduction
5:   for  $i \in \{1, \dots, n\}$  do
6:      $cost\_matrix_{i,:} \leftarrow cost\_matrix_{i,:} - \min(cost\_matrix_{i,:})$ 
7:   end for

8:   Step 2: Column Reduction
9:   for  $j \in \{1, \dots, m\}$  do
10:     $cost\_matrix_{:,j} \leftarrow cost\_matrix_{:,j} - \min(cost\_matrix_{:,j})$ 
11:  end for

12:  Step 3: Cover all zeros in the reduced matrix with minimum number of lines
13:   $covered\_lines \leftarrow \text{MINCOVER}(cost\_matrix)$ 

14:  while  $|covered\_lines| < \min(n, m)$  do
15:    Step 4: Modify the matrix
16:     $h \leftarrow \min(cost\_matrix_{uncovered\_rows, uncovered\_cols})$ 
17:     $cost\_matrix_{uncovered\_rows, uncovered\_cols} \leftarrow cost\_matrix_{uncovered\_rows, uncovered\_cols} - h$ 
18:     $cost\_matrix_{covered\_rows, covered\_cols} \leftarrow cost\_matrix_{covered\_rows, covered\_cols} + h$ 

19:    Step 5: Repeat Step 3
20:     $covered\_lines \leftarrow \text{MINCOVER}(cost\_matrix)$ 
21:  end while

22:  Step 6: Find the optimal assignment
23:   $assignment \leftarrow \text{FINDASSIGNMENT}(cost\_matrix)$ 
24:  return assignment

25: end function

```

---

图 4-4 匈牙利算法具体实现

#### 4.2.4 优化策略

在 ByteTrack 算法中，为了实现实时多目标追踪的高效和精确，采用了一系列优化措施。其中，卡尔曼滤波和匈牙利算法起到了关键作用。除此之外，还有以下几个优化策略：

1) 轻量化特征抽取模型：ByteTrack 采用了与 YOLOv5s 搭配的轻量化特征抽取模型来执行目标检测和跟踪任务。YOLOv5s 的参数量和计算量相对较小，这使得整个算法能够在较低的硬件配置下实现实时处理，从而确保了高效率。

2) 激活与未激活轨迹管理策略：ByteTrack 引入了激活和未激活轨迹的分类概念，并根据不同状态的轨迹采用相应的处理策略。这种巧妙的管理方法有助

于减少错误匹配和重复轨迹的出现，从而提高了追踪的准确性。

3) 多阶段追踪过程: ByteTrack 实施了多阶段追踪, 该过程包括针对激活轨迹的高分匹配、激活轨迹的低分匹配, 以及未激活轨迹的追踪。这种多阶段追踪策略有效地提高了目标匹配的准确性和完整性, 从而确保了整体追踪性能的优化。

通过这些综合性优化策略, ByteTrack 算法成功实现了在实时多目标追踪任务中的高效和精确表现。这使得算法在各种应用场景中具有更强的适应性和实用性。

### 4.3 基于运动建模的包裹重识别算法

基于 ByteTrack 的快递目标跟踪算法在物流万向桌面环境下实现了同一摄像头下包裹的有效跟踪与识别。然而, 在更为广泛的智能物流万向桌面场景中, 由于多个摄像头共同组成了视觉识别系统, 跨摄像头下的包裹重识别问题便成为了一个新的挑战。为了解决这一问题, 本课题针对跨摄像头环境下的物流包裹重识别问题, 提出了一种基于运动建模的包裹重识别算法。

这种算法利用物流包裹在不同摄像头视野下的运动信息进行建模, 通过分析包裹的运动速度、方向、时间以及万向轮之间的距离等参数, 预测包裹在不同摄像头之间的运动轨迹。通过比较预测的运动轨迹与实际观测到的运动轨迹, 在一定的误差范围内, 可以判断出同一包裹在不同摄像头下的识别结果, 并将其绑定到同一个唯一 ID 上。这种基于运动建模的包裹重识别算法可以在跨摄像头环境下提高包裹识别的准确性和连续性, 为智能物流领域带来更高效的识别和跟踪方案。基于运动建模的包裹重识别算法的具体流程如图 4-5 所示。

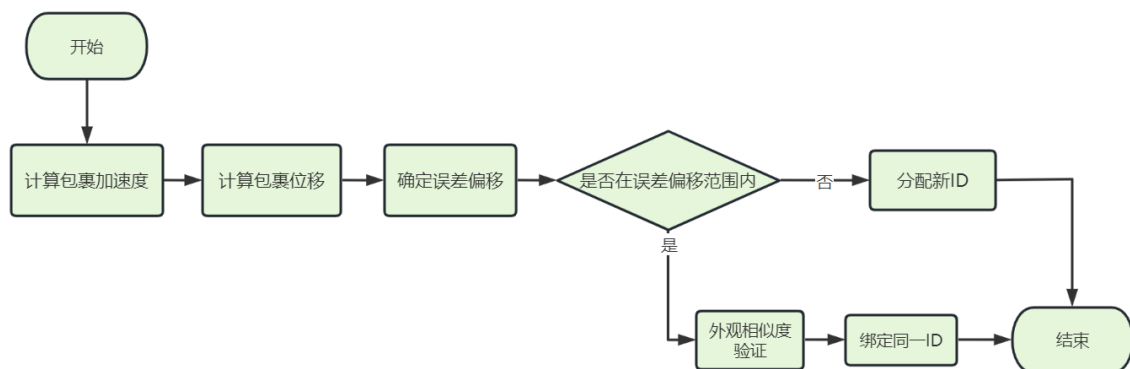


图 4-5 基于运动建模的包裹重识别算法

### 4.3.1 预测运动轨迹

基于运动建模的包裹重识别算法的核心在于运动轨迹的预测。首先需要根据物流包裹的质量和摩擦系数，计算其在万向轮上的加速度和减速度，其次需要通过底层 STM32 开发板的编码器反馈的运动参数，包括万向轮的运动速度、运动方向、运动时间，预测包裹在当前万向轮上的运动位移。由此来完成整体包裹运动轨迹的预测。

根据物流包裹的质量和摩擦系数，我们可以计算其在万向轮上的加速度和减速度。这将有助于更准确地预测包裹的运动轨迹。首先，我们需要确定物流包裹所受的摩擦力。摩擦力公式如式 4-6 所示。接下来，我们可以计算物流包裹在万向轮上的加速度。加速度公式为式 4-7。

$$f = \mu mg \quad (4-6)$$

$$a = f/m \quad (4-7)$$

式中  $f$ ——摩擦力；

$\mu$ ——摩擦系数；

$m$ ——物流包裹的质量；

$g$ ——重力加速度

$a$ ——加速度。

计算出加速度后，我们可以根据物流包裹的初始速度（即编码器此时反馈的万向轮速度）和运动时间来预测包裹在当前万向轮上的位移轨迹。运动位移的公式如式 4-8 所示：

$$s = vt + \frac{1}{2}at^2 \quad (4-8)$$

式中  $s$ ——预测位移；

$v$ ——编码器返回速度，即包裹初速度；

$a$ ——加速度；

$t$ ——编码器返回的运动时间。

通过这两个方面的计算，我们可以预测物流包裹在万向轮上的运动轨迹。通过对加速度和位移的计算，我们可以为基于运动建模的包裹重识别算法提供一个较为准确的预测模型，从而提高跨摄像头环境下包裹识别的准确性和连续性。

### 4.3.2 误差确定

在上述过程中已经基于运动建模对物流包裹的行进轨迹进行了预测，获得

了对于摄像头边界的万向轮上的快递包裹的预测位移。在实际应用中，我们还需要考虑其他因素，如万向轮的转动、物流包裹的形状和质地等，这些因素可能会影响物流包裹在万向轮上的实际运动。因此需要提前确定该模型的偏移误差范围。本课题将通过对比两个相邻摄像头拍摄到的包裹在同一时刻的位置信息来计算它们之间的位置误差。以下是具体的描述和步骤：

1) 首先，我们需要收集两个相邻摄像头在同一时刻拍摄到的包裹的位置信息。这可以通过在每个摄像头下基于轮子的运动速度和方向计算出的位置坐标来实现。

2) 设两个相邻摄像头拍摄到的同一包裹的位置坐标分别为  $(x1, y1)$  和  $(x2, y2)$ 。为了计算两个位置之间的误差，我们可以使用欧几里得距离公式。位置误差计算公式如下：

$$\text{Error} = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (4-9)$$

3) 接下来，我们需要确定一个允许的位置误差范围 ( $E$ )。这个范围将用于判断两个相邻摄像头拍摄到的包裹是否是同一个包裹。允许的位置误差范围在本课题中如式 4-10 所示：

$$E = [D - \text{Error}, D + \text{Error}] \quad (4-10)$$

式中  $D$ ——相邻万向轮间的距离。

4) 最后，我们将比较预测出的快递包裹的位移  $s$  是否在允许的位置误差范围  $E$  内。如果  $s$  在误差偏移范围  $E$  内，我们认为这两个包裹实际上是同一个包裹，并将它们绑定到同一个 ID。否则，它们将被视为不同的包裹，分配给它新的 ID。

### 4.3.3 外观相似度辅助验证

由于智慧物流万向桌面上可能存在多个快递包裹并行运输的情况，因此可能在相同时间段内会有多个包裹满足基于运动建模的包裹重识别的误差范围。为了进一步提高包裹重识别准确率，在基于运动建模的基础上，引入了计算包裹外观相似度的方法进行辅助验证。以下是具体的描述和步骤：

1) 首先，利用卷积神经网络从两个相邻摄像头拍摄到的包裹图像中提取特征向量。这些特征向量能够有效地表示包裹的外观信息，从而有助于判断两个图像是否属于同一个包裹。

2) 接下来，计算两个特征向量之间的余弦相似度。余弦相似度是一种常用的相似度度量方法，可以在  $-1$ （完全相反）到  $1$ （完全相同）之间的范围内反映两个向量的相似程度。余弦相似度计算公式如下：

$$\cos_{sim} = (A \cdot B) / (\|A\| * \|B\|) \quad (4-10)$$

式中  $A$ ——特征向量；

$B$ ——特征向量；

$\|A\|$ ——特征向量  $A$  的模长；

$\|B\|$ ——特征向量  $B$  的模长。

3) 然后，确定外观相似度阈值（设为  $T$ ）。这个阈值将用于判断两个包裹在外观上是否相似。

4) 最后，我们将结合位置误差和外观相似度这两个因素来判断两个包裹是否属于同一个包裹。具体地，如果预测位移在允许的位置误差范围  $E$  内，并且余弦相似度大于或等于外观相似度阈值  $T$ ，我们认为这两个包裹实际上是同一个包裹，并将它们绑定到同一个  $ID$ 。否则，它们将被视为不同的包裹。整体该流程的伪代码如图 4-6 所示。

---

**Algorithm 8** Appearance similarity

---

```

1: procedure COMPUTECOSINESIMILARITY(img1, img2, model, threshold)
2:   feat1  $\leftarrow$  EXTRACTFEATURES(img1, model)
3:   feat2  $\leftarrow$  EXTRACTFEATURES(img2, model)
4:   cos_similarity  $\leftarrow$   $\frac{feat1 \cdot feat2}{\|feat1\| \|feat2\|}$ 
5:   if cos_similarity  $\geq$  threshold then
6:     return True
7:   else
8:     return False
9:   end if
10: end procedure
    
```

---

图 4-6 外观相似度辅助验证

通过综合考虑位置误差和外观相似度，我们可以在跨摄像头环境下对两个包裹是否属于同一个包裹进行更加可靠的判断，从而进一步提高智慧物流万向桌面系统的识别准确率。

## 4.4 本章小结

本章主要介绍了两个关键算法：基于 ByteTrack 的快递目标跟踪算法和基于运动建模的包裹重识别算法。这两个算法在智能物流万向桌面的应用中发挥了重要作用，有效地解决了包裹跟踪和重识别问题。

首先，我们介绍了基于 ByteTrack 的快递目标跟踪算法。该算法通过定义轨迹状态、构建与追踪轨迹、进行数据关联以及采取一系列优化策略，实现了实时的多目标跟踪。在轨迹状态定义中，我们引入了激活和未激活轨迹的概念。



轨迹构建与追踪部分涉及了数据关联,主要包括相似性计算和目标匹配。此外,我们还介绍了 ByteTrack 算法的优化策略,如轻量化特征抽取模型、激活与未激活轨迹管理策略和多阶段追踪过程。

然后,我们详细阐述了基于运动建模的包裹重识别算法。该算法主要包括预测运动轨迹、误差确定和外观相似度辅助验证。通过计算包裹在万向轮上的运动轨迹,我们可以预测包裹到达下一个万向轮的时间。结合误差确定,我们可以判断两个包裹是否为同一个包裹。为了进一步提高识别准确率,我们还引入了计算包裹外观相似度的方法。

综上所述,本章详细介绍了目标跟踪与包裹重识别的相关算法,为智能物流万向桌面提供了有效的技术支持。通过实现这两个算法,我们可以实现对包裹的实时跟踪和重识别,从而提高物流系统的效率和准确性。

## 第5章 实验验证与异常反馈

### 5.1 引言

本章主要是对第 3 章基于 YOLOv5 的目标检测算法改进和第 4 章目标跟踪与包裹重识别算法进行实验验证和结果分析，以确保所提出的方法在实际应用中能够有效地解决智能物流万向桌面的挑战。除此之外，本章还将介绍异常反馈机制，以便在实际运行过程中处理可能出现的问题和不确定性。

首先，我们将在第 5.2 节详细介绍实验设置。实验设置包括了实验环境、实验数据、训练参数等方面。

接下来，在第 5.3 节中，我们将根据实验设置和评价指标对实验结果进行详细的分析。评价指标主要涵盖了目标检测、目标跟踪和包裹重识别三方面的性能度量这将有助于我们了解所提出的算法在不同场景下的表现。

最后，在第 5.4 节中，我们将探讨异常反馈机制并将系统界面进行展示。这一机制的设计和实现是为了确保智能物流万向桌面系统在遇到意外情况或异常状况时能够及时地进行自我调整和纠错。通过有效地利用异常反馈机制，我们可以进一步提高整个系统的稳定性和可靠性。

### 5.2 实验设置

#### 5.2.1 实验环境

主机端涉及的操作系统为 Ubuntu20.04，显卡为 RTX3070，显存为 8GB。硬件端涉及到的实验环境如下表 5-1 所示。其中由一个树莓派控制 6 块 STM32 开发板，每块 STM32 开发板控制 4 个万向轮（十字结构）。并且通过两个相机传感器进行包裹重识别算法的验证。

表 5-1 硬件端实验环境

硬件设备	数量
STM32F103RET6 开发板	6
树莓派 4B	1
相机传感器	2

### 5.2.2 实验数据

本章节的实验数据主要侧重于针对基于 ByteTrack 的快递目标跟踪算法和基于运动建模的包裹重识别算法的视频数据。鉴于当前并没有专门针对快递包裹场景的开源视频数据集，我们通过人工搜索和拍摄方式共收集了 10 段实验视频。这些视频包括 5 段实际场景拍摄的视频和 5 段快递仓储监控视频，如图 5-1 所示。

在实际场景拍摄的视频中，我们尝试捕捉不同的快递包裹类型以及在物流万向桌面上的移动过程，以便更全面地评估我们的算法在实际应用中的表现。而快递仓储监控视频则提供了一个更加真实的物流环境，使我们能够深入了解算法在实际工作条件下的性能。

由于智慧物流万向桌面的面积受限，多目标跟踪部分的数据集主要采用快递仓储监控视频。这些视频可以展示大量包裹在复杂环境中的移动和互动情况，有助于评估算法在处理高度密集的场景时的准确性和鲁棒性。通过这些视频数据的分析，我们将能够更好地评估和优化我们的目标跟踪与包裹重识别算法。

其中实验数据的视频分辨率统一设置为帧宽度 1920p，帧高度 1080p。



(a) 人工拍摄视频



(b) 仓储监控视频

图 5-1 实验测试视频

### 5.2.3 网络训练参数

根据实验数据的多次测试，最终确定在 ByteTrack 目标跟踪器中的相关参数设置如表 5-2 所示。主要包括以下关键参数：

- 1) 跟踪阈值：用于衡量目标检测输出与当前轨迹之间的关联程度。当目标检测输出与当前轨迹的关联分数超过该阈值时，可以认为它们代表同一目标。
- 2) 轨迹缓冲区的大小：用于保存最近几帧的信息。轨迹缓冲区有助于在短

暂的遮挡或丢帧情况下保持目标跟踪。

3) 匹配阈值: 用于判断两个目标间的相似程度。当两个目标间的相似度分数高于该阈值时, 它们被认为是同一个目标。

4) 纵横比阈值: 用于剔除异常的边界框。当目标的纵横比超过该阈值时, 可以认为该边界框可能出错, 因此将其从跟踪过程中排除。

5) 最小边界框面积: 用于排除过小的目标。如果目标的边界框面积低于该阈值, 它们可能是噪声或不相关的物体, 从而被从跟踪过程中移除。

表 5-2 ByteTrack 目标跟踪参数设置

参数	初值
track_thresh (跟踪阈值)	0.25
track_buffer (轨迹缓冲区大小)	30
match_thresh (匹配阈值)	0.8
aspect_ratio_thresh (纵横比阈值)	3.0
min_box_area (最小边界框面积)	1.0

## 5.3 实验结果及分析

### 5.3.1 评价指标

在本课题中, 我们关注的评价指标主要包括多目标跟踪精度 (MOTA)、IDF1 (ID F1 Score)、身份切换次数 (ID Switches)、跟踪成功率、精确率、召回率、mAP 以及 FPS。需要注意的是, 精确率、召回率、mAP 和 FPS 与第三章目标检测相关的评价指标相同, 因此在本章中不再详细解释。以下是多目标跟踪及包裹重识别相关评价指标的具体说明:

1) 多目标跟踪精度 (MOTA): 这是一个综合考虑误报、漏报和身份切换等因素的指标, 用于评估跟踪算法的整体性能。MOTA 值越高, 表示跟踪性能越好。

2) IDF1 (ID F1 Score): 这是一种多目标跟踪评价指标, 主要用于衡量跟踪算法在保持目标身份一致性方面的性能。IDF1 是基于匹配目标的精确率 (Precision) 和召回率 (Recall) 计算得到的 F1 分数。

3) 身份切换次数 (ID Switches): 表示跟踪过程中目标身份错误切换的次数。身份切换次数越低, 表示跟踪算法在保持目标身份一致性方面的性能越好。

4) 跟踪成功率 (Tracking Success Rate): 表示成功跟踪到目标的次数与目标总数的比例。跟踪成功率越高, 表示跟踪算法在保持对目标的持续跟踪方面的性能越好。

### 5.3.2 目标跟踪结果对比

在本课题中选取了主流的多目标跟踪算法 SORT、DeepSORT 作为对比, 基于 ByteTrack 的快递目标跟踪算法的效果如下表 5-3 所示。

表 5-3 ByteTrack 目标跟踪效果对比

跟踪器	MOTA	IDF1	ID Switches	跟踪成功率	FPS
SORT	0.76	0.72	41	0.85	48.65
DeepSORT	0.71	0.69	15	0.79	39.28
ByteTrack	0.78	0.73	18	0.81	50.61

通过表 5-3 的数据可以分析出以下结论:

1) 与当前主流的多目标跟踪算法相比, ByteTrack 在准确性和速度方面具有一定的优势。观察多目标跟踪精度和 IDF1 两项指标, ByteTrack 跟踪器的表现最为出色。尽管在跟踪成功率方面, ByteTrack 略逊于 SORT, 但差距并不明显。ByteTrack 在实时性方面表现也相当优秀。在智慧物流背景下, 实时性是一个至关重要的衡量标准。。

2) 相对于主流的多目标跟踪算法, ByteTrack 更有效地解决了 ID Switches 问题。与 SORT 跟踪器在准确性和实时性方面的差距并不大, 但 ByteTrack 在 ID Switches 指标上取得了显著提升。ByteTrack 通过利用检测框与跟踪轨迹之间的相似性, 在保留高分数检测结果的同时, 从低分数检测结果中剔除背景噪声, 并挖掘出实际目标以降低漏检并提高轨迹连续性。同时, 尽管 DeepSORT 在 ID Switches 方面也大幅降低, 但由于增加了外观特征提取和度量学习, 它在提升物流包裹准确度方面的作用并不显著, 且实时性有所降低。

部分快递包裹的目标跟踪效果如图 5-2 所示, 图中展示了当前视频帧中的包裹总数及向左、向右的包裹数。可以看到在相对复杂的视频中, 目标跟踪效果仍然较为优秀。



图 5-2 ByteTrack 跟踪效果展示

### 5.3.3 包裹重识别结果对比

在本研究中，我们针对给定数据集评估了基于运动建模的包裹重识别算法的性能，具体指标如表 5-4 所示。然而，在通信过程中，由于树莓派缺乏时间同步机制，可能会出现指令延迟问题。这导致在跨摄像头的包裹重识别场景下，mAP 指标表现并不十分理想。尽管如此，在复杂程度较低的环境下，该算法仍然能够正常进行包裹重识别。因此，虽然在实际应用中可能会遇到一定的挑战，但基于运动建模的包裹重识别算法在某些场景下仍具有潜在的应用价值。

表 5-4 基于运动建模的包裹重识别算法评价

精确率	召回率	mAP	FPS
0.7821	0.7410	0.8012	42.19

基于运动建模的包裹重识别算法的效果如图 5-3 所示，图中左侧展示了相机传感器 1 拍摄到的视频序列的最后一帧，而右侧则为相机传感器 2 拍摄到的视频序列的第一帧。观察这两幅图像，我们可以发现同一个快递包裹在两个相机视角下所分配的 ID 是一致的。这说明我们所设计的基于运动建模的包裹重识别算法能够在不同视角下准确地识别和跟踪相同的快递包裹。

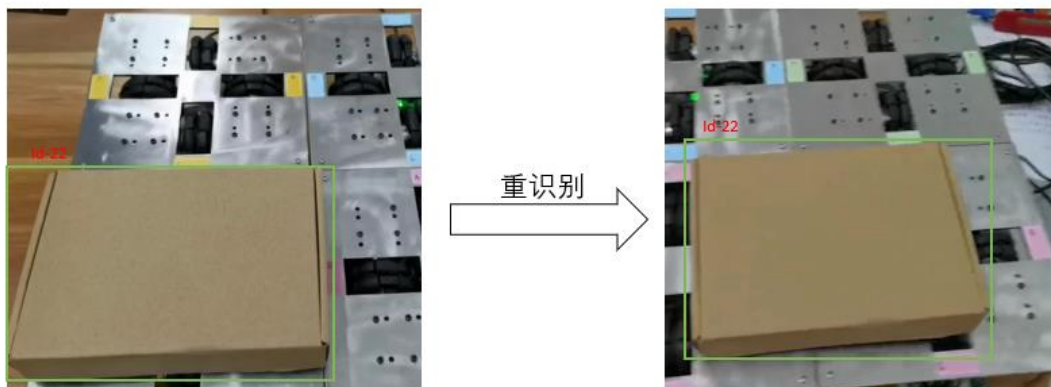


图 5-3 基于运动建模的包裹重识别算法效果

在本课题中，我们设计了一种基于运动建模的包裹重识别算法，并引入了外观相似度辅助验证作为其中一个重要环节。为了证明该辅助验证环节在实际应用中的必要性，我们进行了一组对比实验。在这组对比实验中，我们使用相同的数据集，对比了具有外观相似度辅助验证和不具有该验证环节的两种算法。通过这样的设置，我们可以更好地了解在包裹重识别过程中，外观相似度辅助验证对算法性能的影响。实验结果如图 5-4 所示。

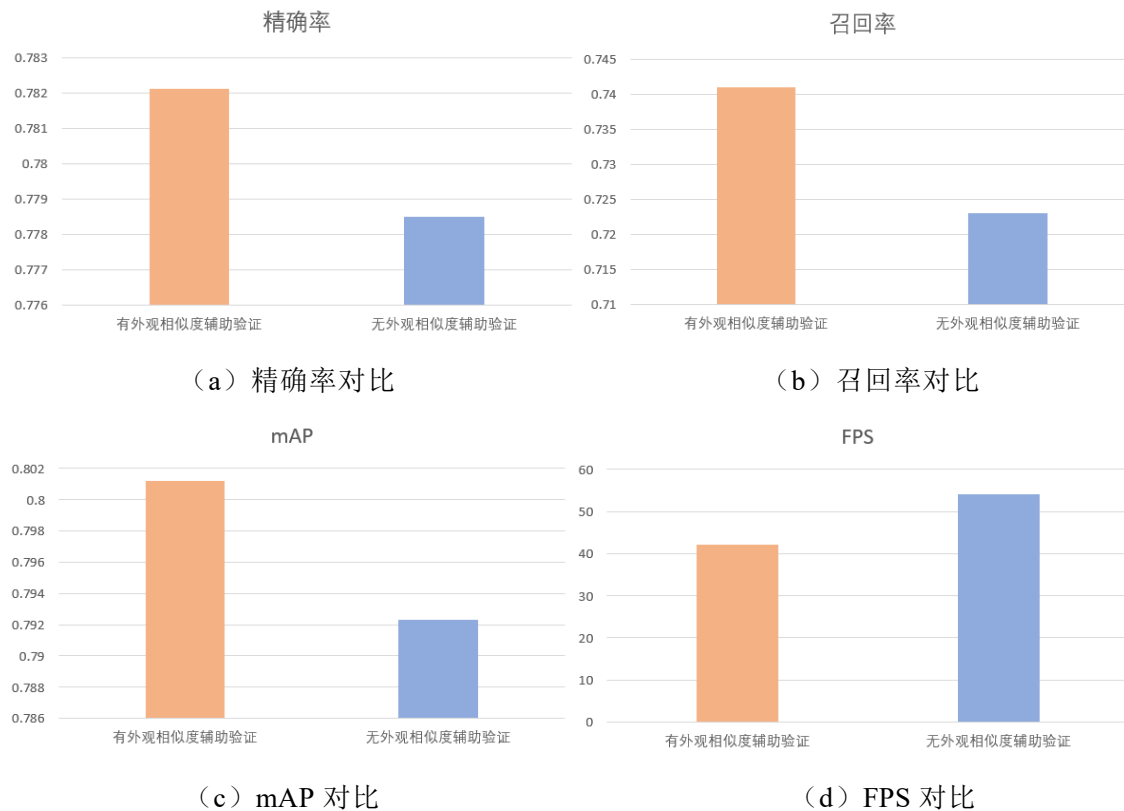


图 5-4 基于运动建模的包裹重识别算法对比结果



图中的结果清晰地展示了，在加入外观相似度辅助验证的情况下，精确率、召回率和 mAP 这三个评价指标都得到了显著提升。尽管引入卷积神经网络导致了 FPS（每秒帧数）指标的降低，但是需要注意的是，前置目标检测算法和目标跟踪算法所消耗的时间相对较短。考虑到目前基于运动建模的包裹重识别算法的准确率并不十分理想，因此，在这种情况下，我们有充分的理由认为，引入外观相似度辅助验证是有助于提高整体准确率的。尽管 FPS 指标略有下降，但在实际应用场景中，提高准确率对于包裹重识别任务的成功执行具有更重要的意义，因此采用外观相似度辅助验证是合理且有益的。

## 5.4 异常反馈机制及界面展示

本节将讨论智慧物流万向桌面中可能遇到的异常情况及其相应的反馈机制。智慧物流万向桌面作为一个高度集成和自动化的系统，涉及多个子系统的协同工作，包括传感器、控制器、执行器、通信网络和软件算法等。由于系统的复杂性，各种原因可能导致系统出现异常，例如硬件故障、软件故障、运输问题、人为干扰和环境因素等。这些异常可能对智慧物流万向桌面的正常运行造成不利影响，甚至导致生产过程中的损失和安全隐患。

为了确保智慧物流万向桌面的稳定运行和高效性能，异常反馈机制显得尤为重要。有效的异常反馈机制可以帮助及时发现和处理异常，预防潜在的故障，从而提高系统的可靠性和安全性。此外，通过对历史数据的分析和预测，异常反馈机制还可以为系统优化和改进提供有益的指导，进一步提高智慧物流万向桌面的整体性能。

在本节中，我们将详细讨论智慧物流万向桌面可能遇到的各种异常类型，以及相应的反馈机制，以期为实现高效、稳定和安全的智慧物流万向桌面提供有力支持。

### 5.4.1 异常类型

由于物流万向桌面所处环境相当复杂，可能会遇到许多不同类型的异常，如人为干扰、环境因素等难以控制和复现的异常。因此，在本课题中，我们仅关注由系统本身引起的异常。这些异常主要可分为以下几类：

- 1) 通信异常：物流万向桌面涉及多个子系统之间的通信，如传感器、控制器、执行器等。通信异常可能由于信号干扰、信号弱、数据包丢失或延迟等原因导致。这种异常可能影响子系统之间的信息传递和协同工作，从而降低整体

系统的性能和稳定性。

2) 硬件故障：物流万向桌面中包含各种硬件设备，如传感器、执行器、控制器等。硬件故障可能由于设备老化、损坏、故障或安装不当等原因导致。这种异常可能导致系统无法正常运行，甚至引发安全隐患。对于硬件故障，需要对设备进行定期检查、维护和更换，以确保系统的正常运行。

3) 重识别异常：物流万向桌面涉及包裹的重识别过程，这是一个关键的环节，以确保包裹在整个物流过程中的正确追踪。重识别异常可能由于算法缺陷、模型泛化能力不足或输入数据质量低等原因导致。这种异常可能导致包裹被误识别或无法识别，从而影响物流过程的准确性和效率。

### 5.4.2 反馈机制

针对上述提到的异常类型，在本课题中主要建立了两种异常反馈机制来应对，具体如图 5-5 所示。

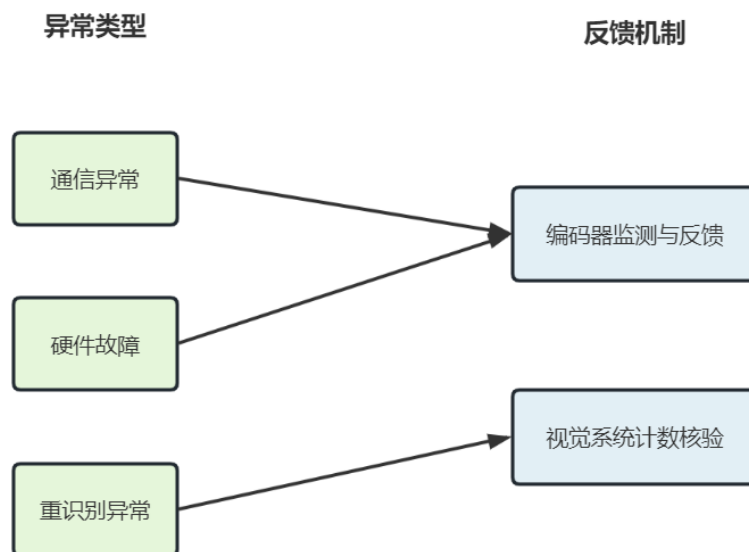


图 5-5 异常类型及反馈机制

1) 编码器监测与反馈：该反馈机制主要针对由通信异常导致的命令下发延迟以及传感器、万向轮等硬件故障无法正常按照指令驱动的异常情况。为了实现在实时监测和快速响应，该反馈机制采用 STM32 开发板上的编码器进行毫秒级别的反馈。通过编码器实时获取电机的转动速度，每 5ms 进行一次速度反馈。将速度信息以数据包的形式发送给树莓派，以便进行实时监测。当转动速度的数据出现异常或未能进行速度反馈时，可以迅速判断出系统存在异常。这种编码器监测与反馈机制的应用能够及时检测到潜在的异常情况，从而帮助系统维

护人员及时采取相应措施，降低故障发生的可能性，提高整个物流万向桌面的可靠性和稳定性。

2) 视觉系统计数核验：该反馈机制主要针对重识别异常。在物流万向桌面的实际运行中，可能会存在多个包裹同时运输的情况。如果路径规划不够合理，可能导致快递包裹发生碰撞，从而改变其外形特征，这将降低重识别准确率。然而，在本课题中，目标检测和目标跟踪的准确率较为理想，因此可以通过对比前后摄像头的数据来应对这一问题。具体而言，可以将前一个摄像头的目标检测和跟踪结果与后一个摄像头重识别后的快递包裹数量进行对比。通过计算数量差异，将这一信息反馈给主机端。这样，操作人员便可以根据反馈信息，对可能出现的重识别异常进行检查和处理。这种视觉系统计数核验的反馈机制有助于确保整个智慧物流万向桌面系统的运行准确性和稳定性，同时也降低了因重识别异常造成的潜在风险。该部分具体实现的伪代码可以在图 5-6 中找到。

**Algorithm 9** 视觉系统计数核验

---

```

1: 初始化前一个摄像头的目标检测和跟踪计数器 counter_prev
2: 初始化后一个摄像头的重识别计数器 counter_reid

3: function DETECTANDTRACK
4:   当前一个摄像头检测到新的目标时:
5:     将目标 ID 添加到跟踪列表 track_list_prev
6:     更新前一个摄像头的计数器  $counter\_prev \leftarrow \text{len}(track\_list\_prev)$ 
7: end function

8: function REIDENTIFY
9:   当前一个摄像头重识别到目标时:
10:    将目标 ID 添加到重识别列表 reid_list
11:    更新后一个摄像头的计数器  $counter\_reid \leftarrow \text{len}(reid\_list)$ 
12: end function

13: function COMPARECOUNTERS
14:   比较前后摄像头的计数器:
15:     计算数量差异  $count\_diff \leftarrow counter\_prev - counter\_reid$ 
16: end function

17: if count_diff  $\neq 0$  then
18:   将 count_diff 反馈给主机端
19:   提醒操作人员进行检查和处理
20: else
21:   系统正常运行，无需额外操作
22: end if

```

---

图 5-6 视觉系统计数核验实现

### 5.4.3 系统界面展示

为了更加直观地展示目标跟踪算法和包裹重识别算法的性能，本研究利用

PyQt5 框架构建了一个用户友好的交互式前端 UI 界面。如图 5-7 所示，这个界面清晰地呈现了系统的各项功能和操作。通过这个界面，用户可以了解和评估算法的实际效果，进一步提高研究的实用价值和可操作性。



图 5-7 UI 界面展示

在本系统的前端界面中主要包括了目标跟踪和包裹重识别两部分，每部分的具体按钮及功能如下：

1) 目标跟踪：该部分主要包括选择视频、播放、暂停、结果显示、结束五个按钮。其中可通过选择本地已有视频进行测试，可对视频进行播放暂停的操作。点击结果显示后会展示该视频进行目标跟踪算法后的视频。点击结束按钮后，该视频流将被清空。

2) 包裹重识别：该部分主要包括选择视频 1、选择视频 2、结果显示、计数

核验四个按钮。由于包裹重识别检测的是不同摄像头间目标跟踪的情况，因此需要选择两个视频。点击结果显示按钮后会展示两个视频的目标跟踪结果及 id 绑定情况。计数核验按钮是反馈机制中所提到的视觉系统计数核验，会反馈两个视频中目标跟踪的数量差。

## 5.5 本章小结

本章主要介绍了实验验证与异常反馈机制。首先，在 5.2 节中，我们详细描述了实验设置，包括实验环境、实验数据以及网络训练参数。实验环境包括硬件设备和软件平台，实验数据则涉及多目标跟踪和包裹重识别任务所需的数据集。网络训练参数包括学习率、迭代次数等超参数的设置。

接着，在 5.3 节中，我们对实验结果进行了分析。首先，我们介绍了评价指标，包括 MOTA、IDF1 等多目标跟踪和包裹重识别任务的评价指标。然后，我们对目标跟踪结果进行了对比，分析了 ByteTrack 跟踪器与其他主流多目标跟踪算法的性能差异。此外，我们还对包裹重识别结果进行了对比，探讨了基于运动建模的包裹重识别算法在不同环境下的表现。

最后，在 5.4 节中，我们阐述了异常反馈机制。首先，我们讨论了可能遇到的异常类型，包括通信异常、硬件故障以及重识别异常。随后，我们介绍了针对这些异常类型的反馈机制，如编码器监测与反馈、视觉系统计数核验等。这些异常反馈机制有助于在复杂的智慧物流万向桌面环境下及时发现并处理异常情况，确保系统的稳定运行。同时还展示了本课题的前端界面，增强了用户交互的性能。

通过本章的实验验证与异常反馈，我们展示了所提出的多目标跟踪和包裹重识别算法在实际应用中的有效性和鲁棒性，同时也为未来在智慧物流万向桌面系统中进一步优化和改进算法提供了有价值的参考。

## 结 论

本课题完成了基于目标检测与包裹重识别的智慧物流桌面，在实际搭建物流万向桌面的基础上进一步完成了视觉分拣系统。主要包括基于 YOLOv5 的目标检测算法改进、目标跟踪与多摄像头下的包裹重识别以及系统实验验证与包裹异常反馈。针对智慧物流万向桌面的实际应用场景，文章从各个角度进行了深入探讨，并针对实际问题提出了相应的解决方案。主要完成的成果如下：

(1) 搭建了智慧物流万向桌面。文章阐述了智慧物流万向桌面的搭建过程，包括系统功能设计和桌面结构设计。通过利用 STM32 开发板、涡轮电机、万向轮、树莓派等组件实现了智慧物流万向桌面的硬件设计。同时，详细介绍了基于 STM32 开发板的底层驱动，包括指令解析、指令下发、电机驱动等关键功能的具体实现。

(2) 对视觉系统中的目标检测模型 YOLOv5 进行了改进。提出了一系列优化方案，包括将 YOLO v5s 网络中的非极大值抑制 NMS 算法改进 Soft-NMS；将 YOLO v5s 中原有的损失函数 CIOU 改进为 EIOU；将 YOLO v5s 中的 PANet 层改为 efficientDet 中的 BiFPN；将 YOLO v5s 的主干网络轻量化为 PP-LCNet。并通过实验验证，所提出的改进方案在速度与精度上均取得了较好的效果。最终目标检测模型的  $mAP_{0.5}$  达到了 0.98841，FPS 达到了 144.55。

(3) 研究了目标跟踪与多摄像头下的包裹重识别问题。详细介绍了基于 ByteTrack 的快递目标跟踪算法，包括其中的轨迹构建与追踪及数据关联的相关问题。同时设计并实现了基于运动建模的包裹重识别算法，包括其中的预测运动轨迹、误差确定和外观相似度辅助验证等关键功能。最终成功解决了在多摄像头情况下的目标识别与跟踪问题。

(4) 进行了实验验证并完善了异常反馈机制及系统界面展示。为了验证目标跟踪与多摄像头下的包裹重识别算法的可行性，进行了针对性实验，并对实验结果进行了详尽分析。分析了 ByteTrack 跟踪器与其他主流多目标跟踪算法的性能差异，最终 MOTA 指标数值达到了 0.78。并验证了包裹重识别算法的外观相似度辅助验证的必要性，最终基于运动建模的包裹重识别算法的  $mAP$  达到了 0.8012。并且完善了异常反馈机制，针对实际应用场景中可能出现的异常情况，提出了异常反馈方案。同时还展示了本课题的前端界面，增强了用户交互的性能。

虽然本课题已面对真实场景设计并实现了智慧物流桌面，并取得部分成果。

但是仍存在需要进一步研究的问题，主要如下：

（1）实验数据有限：由于实验数据的收集较为困难，本文所使用的数据集规模相对较小，可能影响了目标检测算法的泛化性能。在未来研究中，可以通过扩大数据集规模以提高模型的鲁棒性。

（2）对异常情况的处理不够完善：本文提出了针对部分异常情况的反馈机制，但仍存在一定的局限性。例如，面对人为干扰、环境因素等不可控制、复现困难的异常，当前反馈机制可能无法提供有效的解决方案。未来可以考虑对这些异常情况进行更详细的研究，提出更为完善的异常处理方案。

（3）能耗与性能权衡：在实际应用中，通常需要在能耗与性能之间进行权衡。本文虽然在一定程度上优化了目标检测算法，但在提高算法性能的同时，可能会增加能耗。未来研究可以探讨如何在保证性能的基础上降低能耗，以实现更高效的智慧物流万向桌面系统。



## 参考文献

- [1] 黄彬.大数据时代传统物流产业智慧化转型路径研究[ J ] .技术经济与管理研究, 2021,(12):118-121.
- [2] 白振成, 智能分拣系统的种类及发展趋势. 物流技术与应用, 2020. 25(09): 第 114-116 页.
- [3] Uriarte, C., et al., Flexible Automatisierung logistischer prozesse durch modulare roboter. Logistics Journal: Proceedings, 2016. 2016(05).
- [4] 李沛澜, 全向物流运输模块化设备设计. 电子制作, 2019(16): 第 20-22 页.
- [5] 全球首个由机器人完成配送任务的智能配送站正式投入使用. 中国储运, 2019(01): 第 136 页.
- [6] 李祥兵, 陈炼.基于改进 Faster-RCNN 的自然场景人脸检测[J].计算机工程.2021,47(1):210-216.
- [7] Turk M A, Pentland A P. Recognition in face space[C].Intelligent Robots and Computer Vision IX: Algorithms and Techniques. International Society for Optics and Photonics,1991,1381:43-54.
- [8] Viola P, Jones M. Robust real-time face detection[J]. International journal of computer vision,2004,57(2):137-154.
- [9] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features.Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001.Ieee,2001,1: I-I.
- [10] Girshick R, Doahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C].Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition.2014:580-587.
- [11] Lienhart R, Maydt J. An extended set of haar-like features for rapid object detection[C].Proceedings international conference on image processing. IEEE,2002,1:I-I.
- [12] Kuang H L,Chan L L H, Yan H. Multi-class fruit detection based on multiple color channels[C].2015 International Conference on Wavelet Analysis and Pattern Recognition(ICWAPR).IEEE,2015:1-7
- [13] Girshick, R., et al. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.
- [14] Redmon, J., et al. You only look once: Unified, real-time object detection. 2016.

- [15] Redmon, J. and A. Farhadi. YOLO9000: better, faster, stronger. 2017.
- [16] Ioffe, S. and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015: PMLR.
- [17] Redmon, J. and A. Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
- [18] He, K., et al. Deep residual learning for image recognition. 2016.
- [19] Li B, Yan J, Wu W, et al. High performance visual tracking with siamese region proposal network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8971-8980.
- [20] Wang J, Chen K, Yang S, et al. Region proposal by guided anchoring[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 2965-2974.
- [21] 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.
- [22] Chen L, Ai H, Zhuang Z, et al. Real-time multiple people tracking with deeply learned candidate selection and person re-identification[C]//2018 IEEE international conference on multimedia and expo (ICME). IEEE, 2018: 1-6.
- [23] S. Liao, Y. Hu, Xiangyu Zhu and S. Z. Li, Person re-identification by Local Maximal Occurrence representation and metric learning[C]. CVPR, 2015: 2197-2206.
- [24] MA B, SU Y, JURIE F. Local descriptors encoded by fisher vectors for person re-identification[C]. ICCV, 2012: 413–422.
- [25] CHEN D, YUAN Z, HUA G, 等. Similarity learning on an explicit polynomial kernel feature map for person re-identification[C]. CVPR, 2015: 1565–1573.
- [26] SUGIYAMA. Local fisher discriminant analysis for supervised dimensionality reduction[J]. Machine learning, 2010, 78(1/2): 35-61.
- [27] MATSUKAWA T, OKABE T, SUZUKI E, 等. Hierarchical gaussian descriptor for person re-identification[C]. CVPR, 2016: 1363–1372.
- [28] PEDAGADI S, ORWELL J, VELASTIN S, 等. Local fisher discriminant analysis for pedestrian re-identification[C]. CVPR, 2013: 3318–3325.
- [29] KOESTINGER M, HIRZER M, WOHLHART P, 等. Large scale metric learning from equivalence constraint[C]. CVPR, 2012: 2288–2295.
- [30] ENGEL C, BAUMGARTNE P, HOLZMANN M, 等. Person re-identification by support vector ranking[C]. British Machine Vision Conference, 2010: 1–11.
- [31] SCOVANNER P, ALI S, SHAH M. A 3-dimensional sift descriptor and its

- application to action recognition[C]. 15th ACM International Conference on Multimedia, 2007: 357–360.
- [32] MAN J, BHANU B. Individual recognition using gait energy image[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(2): 316–322.
- [33] Rahul Rama Varior, Mrinal Haloi, Gang Wang. Gated siamese convolutional neural network architecture for human re-identification[C]//European Conference on Computer Vision. Springer, 2016:791-808.
- [34] Florian Schroff, Dmitry Kalenichenko, James Philbin. Facenet: A unified embedding for face recognition and clustering[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015:815-823.
- [35] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, Shuicheng Yan. End-to-end comparative attention networks for person re-identification[J]. IEEE Transactions on Image Processing, 2017.
- [36] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, Nanning Zheng. Person re-identification by multichannel parts-based cnn with improved triplet loss function[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:1335-1344.
- [37] Weihua Chen, Xiaotang Chen, Jianguo Zhang, Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification[J]. arXiv preprint arXiv:1704.01719, 2017.
- [38] Alexander Hermans, Lucas Beyer, Bastian Leibe. In defense of the triplet loss for person reidentification[J]. arXiv preprint arXiv:1703.07737, 2017
- [39] Xiao Q, Luo H, Zhang C. Margin Sample Mining Loss: A Deep Learning Based Method for Person Re-identification[J]. 2017.
- [40] Song G, Leng B, Liu Y, et al. Region-based quality estimation network for large-scale person re-identification[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).
- [41] Zheng Z, Zheng L, Yang Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro[C]//Proceedings of the IEEE international conference on computer vision. 2017: 3754-3762.

## 哈尔滨工业大学学位论文原创性声明和使用权限

### 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于目标检测与包裹重识别的智慧物流桌面》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：



日期： 2023 年 5 月 20 日

### 学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1)学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2)学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3)研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。


本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：



日期： 2023 年 5 月 20 日

导师签名：



日期： 2023 年 5 月 20 日

## 致 谢

在这篇论文完成之际，我想表达我最诚挚的感激之情。首先，我要向我的导师刘劼教授表示衷心的感谢。在整个论文研究过程中，您一直以严谨的治学态度和渊博的学识给予我无尽的启示与支持。在每一个关键时刻，您都为我提供了宝贵的建议，使我能够克服困难，突破自己。在您的悉心指导下，我不仅学到了专业知识，更在研究过程中学会了如何解决问题、独立思考和批判性思维。您的无私奉献和坚定信念，将永远成为我人生中的宝贵财富。

同时，我要对实验室的师兄师姐表示感激。特别是感谢殷明睿师姐在学术和生活中给予我的关心与帮助。在这个充满智慧和活力的团队里，每一个成员都以他们的独特方式为我提供了宝贵的支持。还有同届的小伙伴李景旭、文诗慧，我们共同努力、携手前行，每一个欢声笑语的瞬间都成为我珍贵的回忆。我们彼此鼓励，共同成长，让我深知团队力量的伟大。

我还要感谢身边给予我最多支持的小伙伴林之浩和何明睿，在我遇到困难时是你们的鼓励与帮助让我度过难关。和你们创造的每一个美好回忆都将值得我终身铭记。还有给予我很多快乐和力量的室友刘诗琦、杜佳琪。虽然最后一个学期我没能能在宿舍陪伴你们，但是我们会是一辈子的家人，北京、杭州、哈尔滨终点虽不在一处，但我很庆幸路途中有你们陪我走过 6 年！

此外，我要感谢篮球队的教练和队友们，我们在球场上共同拼搏，锻炼意志，培养团队精神。是你们让我在紧张的学术生活中找到了释放压力的方式，更为我提供了难忘的友谊和陪伴。

我还要向我的母校表示由衷的感谢。在这所学校里，我度过了无数美好时光，收获了知识、友情与成长。感谢学校提供了良好的学术环境和丰富的资源，让我能够在这里不断进步，实现梦想。

最后，我要向我的妈妈表达最深的谢意。是您的关爱和支持，让我能够追求梦想，勇敢面对挑战。在我遇到困难时，您们总是给予我鼓励和信心。您的爱如阳光般温暖，伴我成长。谢谢您！