

基于多摄像头的实时人物追踪系统

谭丽欢^{1,2}, 宋鹏倪³, 陈东升¹, 罗 颂^{1,2*}

¹警务物联网应用技术公安部重点实验室, 北京

²重庆理工大学计算机科学与工程学院, 重庆

³成都市公安局锦江区分局信息通信科, 四川 成都

收稿日期: 2025年7月5日; 录用日期: 2025年8月7日; 发布日期: 2025年8月13日

摘 要

在智慧城市与智能安防快速发展的背景下, 多摄像头人物追踪技术成为提升公共安全管理效率的核心需求。传统方案存在跨摄像头身份关联失效、计算资源消耗大及实时性不足等问题。本文提出一种基于边缘计算与中心数据库协同架构的实时人物追踪系统, 采用轻量化模型YOLOv5s进行人体检测, 结合OSNet_x0.25提取512维特征向量, 通过向量数据库实现毫秒级特征匹配。实验表明, 该系统在Market-1501数据集上Rank-1准确率达86.7%, 640 × 640视频流实时处理帧率为8.2 FPS, 在校园场景中验证了工程实用性。

关键词

多摄像头追踪, 边缘计算, YOLOv5, 行人重识别, 向量数据库

Real-Time Person Tracking System Based on Multiple Cameras

Lihuan Tan^{1,2}, Juanni Song³, Dongsheng Chen¹, Song Luo^{1,2*}

¹Key Laboratory of Police Internet of Things Application of Ministry of Public Security, Beijing

²College of Computer Science and Engineering, Chongqing University of Technology, Chongqing

³Department of Information and Communication, Jinjiang District Branch of Chengdu Public Security Bureau, Chengdu Sichuan

Received: Jul. 5th, 2025; accepted: Aug. 7th, 2025; published: Aug. 13th, 2025

Abstract

In the context of the rapid development of smart cities and intelligent security, multi-camera person

*通讯作者。

文章引用: 谭丽欢, 宋鹏倪, 陈东升, 罗颂. 基于多摄像头的实时人物追踪系统[J]. 计算机科学与应用, 2025, 15(8): 73-88. DOI: 10.12677/csa.2025.158199

tracking technology has become a core requirement for improving the efficiency of public safety management. Traditional solutions suffer from issues such as failed cross-camera identity association, high computational resource consumption, and insufficient real-time performance. This paper proposes a real-time person tracking system based on a collaborative architecture of edge computing and a central database. It employs the lightweight YOLOv5s model for human detection, combines it with OSNet_x0.25 to extract 512-dimensional feature vectors, and achieves millisecond-level feature matching through a vector database. Experiments show that the system achieves a Rank-1 accuracy of 86.7% on the Market-1501 dataset, with a real-time processing frame rate of 8.2 FPS for 640×640 video streams, validating its engineering practicality in campus scenarios.

Keywords

Multi-Camera Tracking, Edge Computing, YOLOv5, Pedestrian Re-Identification, Vector Database

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在智慧城市与智能安防迅猛演进的时代背景下，城市数字化转型进程持续深化，多摄像头协同的人物追踪技术已成为提升公共安全管理效能、驱动各行业智能化变革的关键刚需。传统人物追踪方案长期受制于单摄像头监控模式，早期技术体系以卡尔曼滤波、光流法[1]等基础算法为核心，仅能在单一画面内通过预设运动模型实现目标短期跟踪，当遭遇目标遮挡、高速运动等复杂场景时，算法适应性不足的问题便会凸显；随着特征工程的发展，基于颜色直方图、HOG 描述子等手工设计特征的追踪算法虽能通过表观特征匹配实现跨帧关联，但这类人工设计的特征描述符对环境光照变化、成像分辨率差异及视角转换等因素极为敏感，在实际复杂场景中极易引发身份匹配错误或身份漂移现象。此外，传统集中式处理架构需要将多路视频流统一传输至中心服务器进行分析，这种模式不仅对网络带宽资源提出极高要求，还面临硬件部署成本高昂、多源数据实时处理能力不足等现实挑战。

深度学习技术的突破性进展推动了人物追踪领域的技术范式革新。基于卷积神经网络(CNN)的目标检测与特征提取方法显著提升了视觉特征表达能力，但早期深度模型的高计算复杂度导致集中式推理依赖巨额算力投入。如何在保障追踪精度的同时协同优化系统延迟与资源消耗，成为亟待攻克的关键问题。智能安防产业呈现爆发式增长，据 Statista 数据显示，2025 年全球市场规模将突破 800 亿美元(年复合增长率 12.4%)，其中跨摄像头目标追踪技术应用占比超 30% [2]。其多场景技术价值凸显。机场、车站等密集场所的单摄像头监控存在视野盲区，多摄像头系统通过跨视角信息融合与时空轨迹分析，实时构建目标移动路径，快速预警遗留物品、群体聚集等事件；跨区域摄像头网络精准捕捉顾客行动轨迹、停留时长及商品关注偏好，IBM 研究证实采用该技术的零售企业转化率提升 22%、顾客满意度提高 17% [3]；交通枢纽的多摄像头网络实时解析行人流动态，辅助实施拥堵疏导(如地铁站动态调节闸机数量以降低踩踏风险)，并结合人脸识别定位走失儿童等重点目标；对工厂、工地人员活动范围实时监控，智能预警闯入危险区域、未佩戴防护装备等违规行为。

基于上述技术挑战与应用需求，本项目聚焦于具备高精度识别与实时追踪功能的监控系统研究，以单摄像头为基础感知单元，通过构建人物关键特征的智能提取与分析机制，实现追踪目标行为轨迹的实时反馈与预警。该研究成果可广泛应用于智能安防、智慧城市、零售行为分析等多元场景，为异常行为

预警、人流动态统计、跨区域活动分析等上层应用提供核心技术支持与前置条件。

2. 相关工作

2.1. 传统识别技术

传统人物识别技术需要依赖特征提取方法。这些方法大多是人工设计的。早期研究以颜色直方图、方向梯度直方图(HOG)、局部二值模式(LBP)等特征为重点,在此基础上迭代,通过模板匹配或分类器(如SVM、AdaBoost)实现目标检测与追踪。例如, Dalal 等人提出了 HOG 特征结合线性 SVM 的方法[4], 这种方法在行人检测任务中取得了不错的效果, 但其特征表达能力并不完善, 受限于人工设计规则, 对光照变化、姿态差异的适应性不好, 同时遮挡场景的适应性也不好。在跨摄像头场景下, 传统方法通常采用基于颜色的检查, 或者是纹理的相似性度量, 常用方法有直方图交叉核、卡方距离等。但由于不同摄像头视角会有差异, 分辨率也有差异, 以及环境干扰问题, 容易导致特征匹配失效, 难以实现稳定的跨摄像头的人物追踪。

2.2. 深度学习识别技术

近年来, 随着卷积神经网络(CNN)的兴起, 人物识别技术的主流方法逐渐成为基于深度学习的改良版。首先是目标检测领域有突破, 出现了以 Faster R-CNN、YOLO 系列为代表的算法[5], 这些算法通过端到端训练, 可以实现检测精度的大大提高, 同时也达到与速度的平衡。其中呢, YOLO 系列凭借自身较为简洁的设计特点以及颇为高效的检测性能, 已然成为了目标检测方面的优选方案。在行人重识别(ReID)这项任务里, OSNet 等网络借助多尺度特征融合这样的机制, 较为显著地提升了特征表达所具备的鲁棒性。和传统方法相对比来看, 深度学习模型自身能够自动去学习那些带有判别性的高层语义特征, 如此一来, 在诸如遮挡、低光照这类复杂场景之下, 其识别的准确率就有了很大幅度的提高。不过呢, 现有的相关研究大多是集中在中心化处理架构方面, 模型所具有的参数量比较大, 对于计算资源的需求也颇高, 这些问题都对其在边缘设备上实现实时应用形成了限制, 而且跨摄像头特征匹配的效率以及精度也还需要做出进一步的优化才行。

上述技术不断发展, 这就给多摄像头追踪系统打下了一定的基础。不过呢, 要知道在资源受到限制的这样一种条件之下, 怎样去把边缘计算所具有的优势充分结合起来, 进而实现高效的特征提取以及匹配但如何结合边缘计算优势, 在资源受限条件下实现高效特征提取与匹配, 仍是当前研究的难点之一。

3. 研究方法

3.1. 行人重识别(ReID)技术

行人重识别(ReID)是计算机视觉领域的重要研究方向, 核心目标是通过跨摄像头场景下的行人特征匹配, 实现特定目标的持续追踪与身份精准识别[6]。传统监控系统中, 单摄像头覆盖范围与视角固定, 当目标在无重叠视域的摄像头间移动时, 难以实现连续跟踪。ReID 技术通过提取行人图像的鲁棒特征并建立跨摄像头身份关联, 有效弥补了单摄像头监控的局限性, 成为多摄像头协同追踪系统的关键技术支撑。

早期 ReID 技术依赖颜色直方图、HOG、LBP 等手工设计特征, 仅能通过量化外观浅层特征实现相似性度量, 在光照变化、姿态差异或视角变换等复杂场景下泛化能力较弱。随着深度学习发展, 基于 CNN 的端到端特征学习框架成为主流, 通过自动捕获行人图像的高层语义信息并结合度量学习优化特征空间分布, 显著提升了跨摄像头场景的身份判别能力。

当前 ReID 技术面临跨视角特征对齐困难、遮挡场景局部信息缺失、低分辨率图像特征表达不足等挑战[7]。研究者通过引入注意力机制强化关键区域特征、设计多粒度特征融合策略、结合姿态估计实现视角不变性建模等方法应对上述问题。同时,轻量化模型设计成为边缘计算场景的研究热点,借助模型压缩与知识蒸馏技术,在保证识别精度的前提下降低计算复杂度,为嵌入式设备部署创造可能。

本系统采用 YOLOv5s 实现行人检测,结合 OSNet_x0.25 轻量级网络提取 512 维特征向量。OSNet 通过全尺度特征融合机制,在紧凑模型结构下兼顾单片机端实时推理需求与服饰变化、视角差异的适应性。服务器端基于特征向量相似度计算实现跨摄像头身份匹配,这种分布式架构在降低数据传输带宽的同时保证集中式特征比对效率,为多摄像头协同追踪提供了从理论到工程落地的可行方案。

为解决跨摄像头视角差异导致的特征漂移问题,本文提出一种自适应视角对齐模块(Adaptive View Alignment, AVA)。该模块集成于 OSNet 的特征提取层之后,通过以下机制优化特征表达:

- **视角感知注意力机制:** 利用摄像头位置元数据(如安装高度、俯仰角)生成视角权重,动态调整特征图中不同空间区域的响应强度。
- **特征投影变换:** 基于视角差异矩阵(通过标定数据预计算)对特征向量进行仿射变换,减少因视角倾斜导致的几何失真。实验表明,AVA 模块在多视角数据集 VeRi-776 上提升 Rank-1 准确率 3.2%,显著降低跨视角误匹配率。

3.2. YOLOv5 模型

行人重识别(ReID)的实现依赖高效的行人检测前置环节。YOLO 系列算法作为目标检测的代表性方案,以单阶段检测框架和实时性优势在人物追踪系统中广泛应用。其核心原理是将目标检测转化为回归问题,通过网格单元直接预测边界框坐标、置信度及类别概率,摒弃传统两阶段检测的候选区域生成步骤,大幅提升检测速度以满足实时视频流处理需求。在行人检测中,YOLO 通过卷积神经网络提取多尺度特征,结合非极大值抑制(NMS)筛选重叠框,实现行人目标的快速定位与精准坐标输出,为后续特征提取和身份匹配提供基础数据支撑。

与其他检测模型相比,YOLO 在速度与精度平衡上优势显著。单阶段架构利用全局上下文规避复杂背景干扰,多尺度预测层增强小目标检测能力。在 ReID 系统中,检测实时性与准确性直接影响追踪性能:检测速度需匹配摄像头帧率以避免数据堆积,检测框精度需保证行人图像完整裁剪,防止特征提取失真。YOLO 的高帧率处理能力与稳定检测质量,使其成为 ReID 前端检测的理想选择。

YOLOv5 作为系列重要迭代版本,在模型结构、训练策略与部署效率上进行系统性优化:引入 Focus 结构替代传统下采样层,通过切片操作降低计算量并保留空间信息;采用 CSPNet 模块重构骨干网络,通过特征复用与分阶段融合减少参数冗余;颈部网络集成 PANet 实现多尺度特征交互,增强不同尺度行人的检测鲁棒性;训练阶段借助 Mosaic 数据增强提升模型泛化能力,结合自适应锚框计算与 GIoU 损失函数优化密集场景定位精度。这些改进使 YOLOv5 在保持高检测速度的同时,较前代模型显著提升平均精度(mAP),尤其适用于多摄像头场景的复杂条件。

在 YOLOv5 子版本中,YOLOv5s 因轻量化设计成为边缘计算首选。该版本通过压缩模型宽度与深度,将参数量控制在 7.2 M 左右,模型体积约 14 MB,大幅降低硬件算力需求。尽管模型规模缩减,YOLOv5s 仍通过改进的特征融合机制与高效网络结构,维持与大版本相近的检测精度。其轻量化特性使其在树莓派、Jetson Nano 等嵌入式设备上实现实时推理,并可借助 TensorRT 加速引擎优化推理效率。低延迟与低功耗特性既满足分布式部署的单片机资源限制,又为服务器端保留特征比对与轨迹融合的计算资源,支撑端到端实时追踪链路。YOLOv5s 的嵌入式部署能力与 ReID 特征提取网络的无缝衔接,构成多摄像头人物追踪系统的技术基石。

3.3. OSnet 模型

在行人重识别(ReID)系统中,特征提取网络的性能直接决定跨摄像头身份匹配的准确性。OSNet (Omni-Scale Network)作为专为 ReID 设计的轻量级深度神经网络,通过全尺度特征学习机制突破传统模型在多尺度特征融合上的局限。其核心创新在于构建动态聚合多尺度特征的网络结构,可自适应捕获行人图像中从服饰纹理、配饰等局部细节到体型、姿态等全局语义的全尺度信息[7]。相较于 ResNet、MobileNet等通用骨干网络,OSNet通过引入统一的全尺度残差块(OS block)实现单网络多粒度特征融合,显著提升了视角变化、遮挡及低分辨率等复杂场景下的特征表达鲁棒性。

OSNet 的架构围绕“全尺度”特征融合设计:每个 OS block 采用多分支卷积结构,借助不同膨胀率的卷积层并行提取多尺度特征,并通过轻量化通道注意力机制动态调整分支权重,使网络自动聚焦于输入最相关的特征尺度。该设计既避免了人工预设尺度分组的繁琐,又通过特征复用降低计算冗余。此外,OSNet 以深度可分离卷积与点卷积组合替代标准卷积,在保证特征表达能力的同时大幅减少参数量。以本系统采用的 OSNet_x0_25 为例,其参数量仅 0.22 M、体积不足 1 MB,却能在 Market-1501 等主流 ReID 数据集[8]上达到与大型模型相当的识别精度,充分体现“小模型大能力”的设计理念。

OSNet_x0_25 在边缘计算场景中具备显著优势:全尺度特征融合机制缓解了传统轻量级模型因压缩导致的细节丢失问题[9],在低分辨率图像中仍能保持特征区分度;动态权重分配增强了对视角变化和局部遮挡的适应性,当行人部分区域被遮挡时,网络会自动强化可见区域特征权重以稳定身份表达;其单张图像推理耗时仅约 10 ms,与 YOLOv5s 的检测速度形成时序匹配,避免系统吞吐量瓶颈。实际部署中,OSNet_x0_25 可直接加载至 TensorFlow Lite 等单片机 AI 推理框架,与 YOLOv5s 构成“检测-特征提取”本地化流水线,仅需上传 512 维特征向量至服务器,大幅降低网络带宽占用,为多摄像头协同追踪提供了轻量化解决方案。

3.4. 边缘计算

在多摄像头人物追踪系统开展工程实践期间,边缘计算技术被引入进来,这一举措有效化解了传统集中式处理架构所碰到的实时性欠佳、带宽负载过重以及存在隐私泄露风险等一系列问题。边缘计算会把计算任务往下沉,使其靠近数据源的那些终端设备,就好比摄像头端的单片机,如此一来便能达成数据的本地化处理以及初步分析,只把关键特征或者决策结果传送到云端服务器,就这样构建起了一种“端-边-云”协同合作的高效计算模式。在人物追踪的具体场景当中,这条技术路线最为核心的价值就在于能够借助边缘节点所具备的分布式算力,把高计算密度的目标检测以及特征提取这两项任务从服务器转移到终端,极为明显地降低了网络传输所产生的延迟,也减轻了中心节点的计算方面的压力,与此同时还提升了系统对于网络波动和设备异构性的耐受程度。

在本系统设计环节,边缘计算具体是靠搭载了轻量化模型的嵌入式硬件(像树莓派、Jetson Nano 等等这类设备)来实现的。YOLOv5s 和 OSNet_x0_25 联合起来进行部署的策略,充分把边缘设备的算力约束情况考虑进去了:YOLOv5s 借助对模型宽度以及深度的压缩,达成了 14 MB 这样极小的体积,再配合 TensorRT 引擎的推理加速功能,能够在嵌入式 GPU 上实现 30 FPS 以上的实时检测速度;而 OSNet_x0_25 依靠全尺度动态特征融合机制,仅用 0.22 M 参数量就完成了高判别性特征的提取,它的计算复杂度和传统 ResNet-50 相比,大约降低了 95%,但依然能在 Market-1501 等数据集上把 80%以上的 Rank-1 准确率给保持住。这种让模型实现轻量化和通过硬件来加速的协同优化方式,使得单片机可以直接在本地把从原始视频流到特征向量的全流程处理工作完成,仅仅需要上传 512 维特征数据(单次传输量大概是 2 KB),和传输原始视频流(1080 P 帧大概 500 KB)相比,能够把带宽消耗减少 99.6%,与此同时还能避免敏感图像数据在公网传输过程中出现隐私风险。

边缘计算一经引入，便给系统赋予了更为出色的可扩展性以及鲁棒性。具体而言，是通过把计算负载分散到各个摄像头节点之上，如此一来，服务器端就只需要将精力集中在特征匹配、轨迹融合这类相对轻量级的任务方面了。这样的话，系统便能够依靠增加边缘节点的数量来实现监控范围的线性扩展，而且在这个过程中，并不需要同步对中心服务器的硬件进行升级操作。除此之外，边缘节点自身还拥有本地缓存以及断点续传的能力，哪怕是在网络出现中断的情况之下，依然可以维持短时间的离线推理，等到网络恢复正常之后，再把特征数据补传上去，这无疑有效地提升了系统在弱网环境当中的稳定性。这种采用分布式架构的设计方式，一方面与智慧城市、工业巡检等大规模监控场景所提出的需求十分契合，另一方面还依据数据最小化的原则，成功降低了隐私泄露方面的风险，进而为人物追踪技术能够合规落地给予了有力的技术保障。

3.5. 国产替代芯片及模型

本项目在国产芯片 RK3588s 上运行，其具有 3 个 NPU 核心，开发者为瑞芯微公司，瑞芯微为其生产的 RK 系列 AI 芯片开发了 RKNN 模型格式，以及 Rockchip npu 开发平台，该格式可在 RK 系列芯片上运行并充分运用其 NPU 计算核心的性能。

NPU (Neural Networks Process Units)神经网络处理单元，它是专门为神经网络设计的，可以快速执行卷积、矩阵乘法等 NN 中最基本的运算单元，而且功耗更低，相同功耗下，NPU 的性能比 GPU 更高。NPU 与 GPU 加速不同，体现为每层神经元计算结果不用输出到主内存，而是按照神经网络的连接传递到下层神经元继续计算，因此其在运算性能和功耗上都有很大的提升。

Yolov5 和 osnet 都是基于 pytorch 的.pth 格式模型，内多为 float32 的浮点运算，RK3588s 不支持这类模型的运行，同时在模型推理时，并不需要进行梯度的反向传播，因此不需要那么高的计算精度，这时可将模型映射到较低精度的运算上，降低运算量，提升运算速度。这样将模型从高精度运算转换到低精度运算的过程叫作模型量化在.pth 等通用模型转换为 RKNN 模型时会进行进行剪枝，量化等优化操作，提升模型的运行效率。模型对称量化如图 1 所示，公式如下：

$$S = \frac{2^{n-1} - 1}{\max(x) - \min(x)} \quad (1)$$

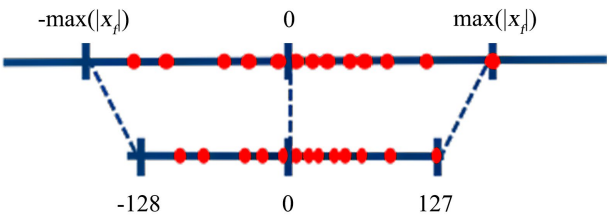


Figure 1. Schematic diagram of model symmetry quantification
图 1. 模型对称量化示意图

4. 系统概述

在智慧校园、公共安防等场景中，实时人物追踪系统需满足多维度综合需求。系统需构建分布式计算架构实现边缘端与中心服务端的协同运作，通过国产化边缘设备完成前端感知数据的实时处理，降低中心服务器负载并保障数据隐私。在性能层面需达成实时视频流处理的基本要求，即在 640×640 分辨率下保持不低于 5 FPS 的推理速度，确保目标检测与特征提取的时效性。系统需具备跨摄像头视角的连续追踪能力，支持至少 8 路视频流并发处理，且 ID 匹配准确率需高于 80%以维持追踪连续性。鲁棒性方面

需适应复杂光照变化、局部遮挡等现实干扰因素，在校园走廊、操场等典型场景中保持稳定运行。可扩展性需求体现在支持不同类型摄像设备的灵活接入，包括具备本地推理能力的智能相机与仅具备视频传输功能的传统设备。安全性需满足数据加密传输、特征向量匿名化存储等规范，符合国家隐私保护法的技术实施要求。经济性约束要求边缘端硬件成本控制在单节点 1000 元以内，确保大规模部署的可行性。

4.1. 系统框架

系统的整体架构主要围绕着多摄像头协同工作来展开运作的。摄像头设备会在边缘端去执行那些关键的计算任务，而视频流或者特征数据呢，则会凭借不同的协议传输到服务端那边。服务端在此充当着核心枢纽的角色，它要处理来自多个源头的请求，这里面就涉及特征匹配的相关事宜、对视频流进行解析以及回应用户的查询等内容。客户端是通过网页界面来实时展示处理的最终结果的，如此一来，用户就能够随时去查看监控画面，或者去搜索特定人物的行动轨迹。边缘端的摄像头设备存在两种运行模式，其中一种模式是在本地就完成目标检测以及特征提取的工作，在生成较为紧凑的特征向量之后再上传；另一种模式则是直接把未经处理的原始视频流传输过去，以此来适应不同硬件条件所提出的各种需求。

服务端所具备的请求路由模块承担着对各类请求加以分发的职责，那些特征数据会被送进 FAISS 库以便实现快速的匹配操作，而视频流则交由专门的推理模块来逐帧展开分析工作。FAISS 库凭借 L2 距离达成高效搜索的效果，其匹配所得的结果会与时间衰减策略相结合，以此来对身份关联的准确性予以优化。ID 管理数据库会对每个身份的出现时间以及与之关联的摄像头情况进行记录，并且通过动态清理那些长时间未曾活跃的记录的方式，来确保系统能够维持应有的效率。当用户上传图片的时候，便会触发服务器端一整套完整的推理流程，其中检测、特征提取以及匹配等过程均能在短短数秒之内就得以完成，而最终的结果则会经由加密通道被送回到前端界面之上。

客户端那边的实时视频面板会一直不停地接收从服务端传过来的 MJPEG 流，其画面上还会叠加有检测框以及身份标注，并且将延迟稳稳地控制在 200 毫秒范围之内。统计用的仪表盘会把各个摄像头的工作状态与系统负载整合到一起，而历史数据趋势图呢，则能够助力管理员去掌握人流变化方面的规律。ID 搜索界面是支持上传本地图片的，之后系统会返回与之匹配的身份信息，还有其在多个摄像头当中的活动记录，查询所得的结果在缓存一小段时间之后就会自动清除掉，以此来保证隐私方面的安全。用户查询结果中的行人轨迹展示仅包含虚拟 ID 与时间戳，不关联任何生物特征或身份信息。原始查询图片在匹配完成后立即删除，系统不保留用户上传数据。

在正常运行的场景之中，边缘设备会周期性地把特征向量以及元数据打包起来然后进行发送操作，服务端则会迅速地匹配方面的工作，并且对数据库予以更新，与此同时，客户端也会同步地刷新所显示的内容。当用户提交关于图片查询的请求之时，服务端会开启独立的推理线程，以此来避免对实时处理流程造成干扰。在纯视频流模式之下，服务端要承担起全部的计算负载，从视频的解码、目标的检测一直到特征的提取等各项工作，全部都在服务器上完成，这样的设置能够适配那些较为老旧或者资源受到限制的边缘设备。而不同模式之间的切换工作，是由网络带宽以及设备的算力来进行动态调整的，通过这样的方式确保系统在复杂的环境当中也能够稳定地运行下去。

系统的核心创新点在于边缘和云端能够灵活展开协作，其特征级数据进行传输的时候，能够在很大程度上减少对于带宽的占用情况。FAISS 库有着高效的检索机制，这一机制能够对实时响应起到有力的保障作用。系统对多协议予以支持，如此便使得设备的兼容性得以增强，不管是借助 HTTP 来上传特征，还是依靠 RTSP 去传输视频流，系统都可以实现无缝对接并妥善处理。安全机制在整个架构当中是贯穿始终的，数据在传输过程中会进行加密处理，同时还会施行定期清理的策略，这二者有效降低了隐私泄露方面的风险。在实际进行部署的时候，系统在校园以及商场等场景当中都验证了自身具备的可靠性与

实用性，从而为智能安防给出了具备可扩展性的技术方案。

4.2. 数据流分析

系统的实现逻辑用数据流方式呈现，有正常运行场景、用户图片查询场景以及详细的场景分析。分别如图 2、图 3 和表 1 所示。

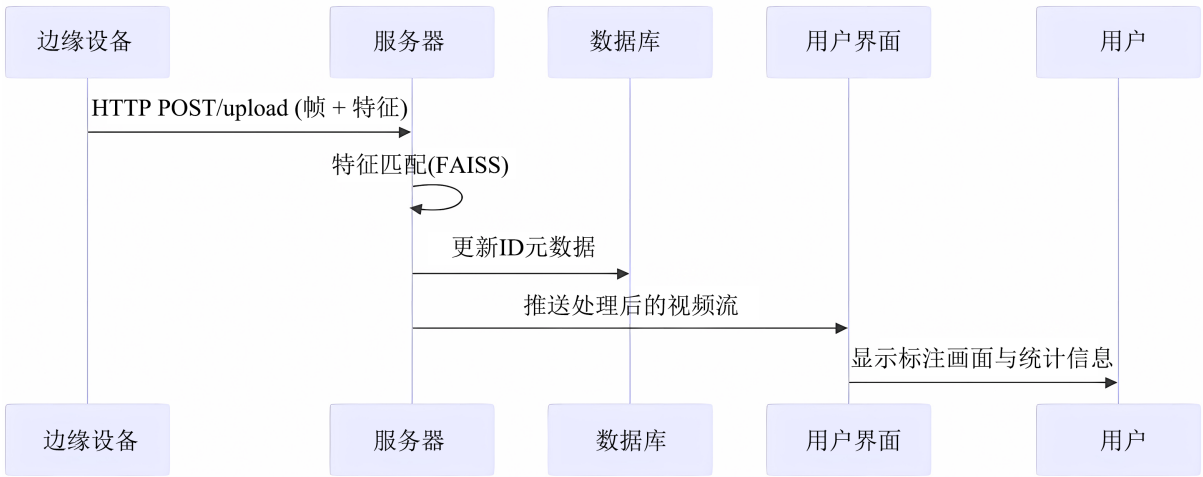


Figure 2. Normal operation scenario (edge inference)
图 2. 正常运行场景(边缘端推理)

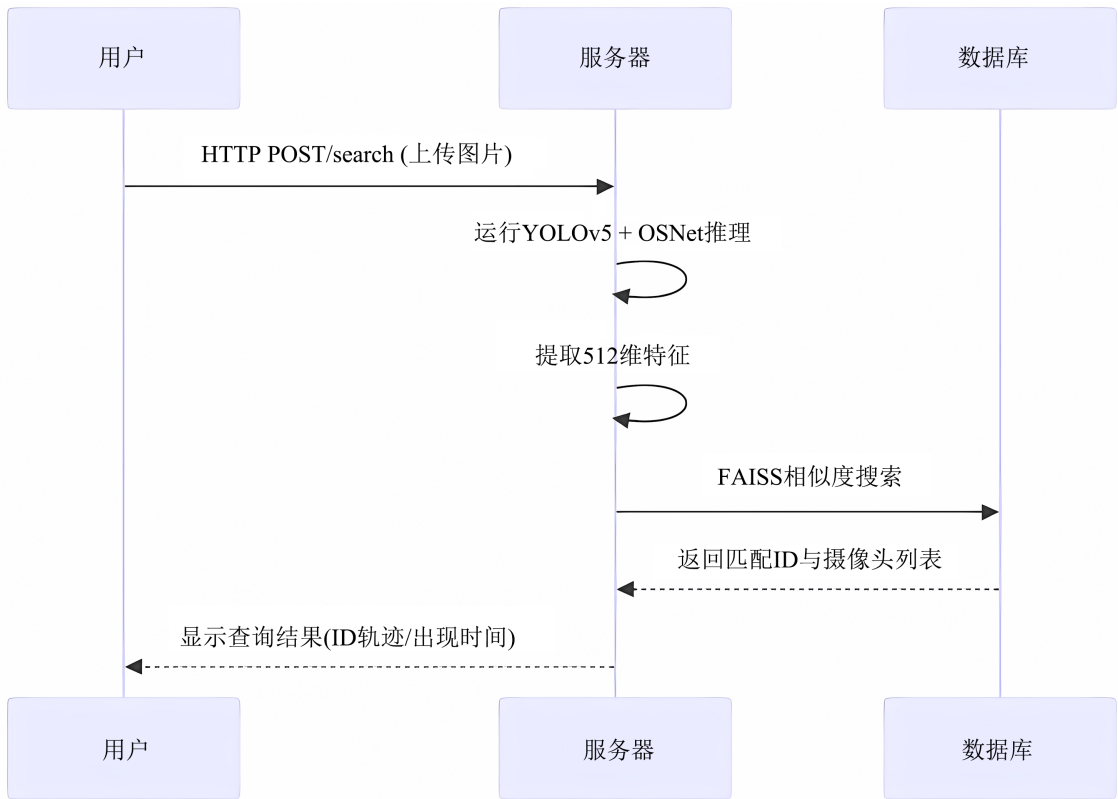


Figure 3. User image query scenario
图 3. 用户图片查询场景

Table 1. Scenario analysis
表 1. 场景分析

场景	计算负载发布	网络宽带需求	适用场景
边缘推理 + 特征上传	边缘端：高 服务端：低	低(特征数据)	带宽受限环境(如 4G 网络监控设备)
纯视频流传输	边缘端：零 服务端：高	高(视频流)	算力受限边缘设备(如老旧摄像头)
用户图片查询	服务端：突发性高负载	中等(图片)	事后追溯与人工排查场景

4.3. 推理计算过程

4.3.1. 图像预处理

图像预处理阶段旨在将原始图片转换为模型可处理的规范化格式。输入图片首先通过双线性插值算法缩放至固定尺寸(640 × 640 像素)，这一尺寸基于 YOLOv5s 的默认输入设计，兼顾检测精度与计算效率。随后，像素值从整数范围 0, 255, 255 归一化至浮点范围 0, 10, 1，以避免数值溢出并加速模型收敛。归一化公式如下：

$$\text{归一化图像} = \frac{\text{原始图像}}{255} \quad (2)$$

最后，图像数据格式从高度 × 宽度 × 通道(HWC)转换为通道 × 高度 × 宽度(CHW)，并添加批次维度，形成四维张量(1 × 3 × 640 × 640)，适配 PyTorch 模型的输入要求。

4.3.2. 目标检测

在目标检测这个环节，主要是借助 YOLOv5s 模型来对行人所在的区域加以定位。该模型中的 CSPDarknet 骨干网络运用了跨阶段部分连接，也就是 CSP 这种结构，它会把输入进来的特征划分成两个部分，分别对这两部分进行相应的处理，之后再把处理好的两部分合并到一起，如此一来，便能够在很大程度上减少计算方面存在的冗余情况。而 PANet 特征金字塔呢，它会进一步地将浅层高分辨率的特征以及深层高语义的特征融合起来，通过这样的操作，就能使模型针对不同尺度目标的检测能力得到提升。对于检测输出的边界框坐标、置信度以及类别概率，会经过非极大值抑制，也就是 NMS 这样的后处理操作，在这个过程中，会把那些置信度高于 0.25 并且交并比，也就是 IoU 低于 0.45 的预测结果保留下来，最终就可以输出经过裁剪之后的人物区域图像了。

4.3.3. 特征提取

在特征提取这一阶段呢，会运用 OSNet_x0.25 网络来把行人图像转变成成为具有判别性的特征向量。就拿裁剪区域来说吧，一开始是要将其缩放至 256 × 128 像素的规格，并且要依照 ImageNet 数据集所规定的均值(0.485, 0.456, 0.406, 0.485, 0.456, 0.406)以及标准差(0.229, 0.224, 0.225, 0.229, 0.224, 0.225)来实施标准化的操作，这样做的目的就是为消除光照差异所带来的影响。OSNet 这个网络呢，它是凭借着多分支卷积结构去提取出不同尺度的特征的。其中，1 × 1 卷积能够捕获到像衣物纹理这类的局部细节[10]，而 7 × 7 卷积则可以获取到全局轮廓方面的信息，并且还会通过自适应权重的方式来动态地融合各个分支所输出的内容。在网络的末端，有广义均值池化(GeM)这一操作[11]，它能够聚合空间特征，把像背包、帽子这类显著区域给突出出来，同时还能对背景干扰起到抑制的作用，最终就生成了 512 维的归一化特征向量。这个向量经过 L2 归一化之后，它的模长是 1 [12]，如此一来便能够直接凭借余弦相似度来开展跨摄像头匹配的相关工作了。

4.4. 特征值匹配过程

4.4.1. 特征库构建

特征匹配在实现跨摄像头人物身份关联方面处于核心地位[13]，它主要是把从边缘端或者服务器端提取出来的特征向量，跟数据库里的历史特征去做相似度方面的计算，以此来判定到底是不是同一个人物[14]。这个过程包含特征库构建、相似度计算以及 ID 决策管理这三个阶段，具体的流程是这样的：

一开始要进行特征库的构建工作，在系统刚开始初始化的时候呢，FAISS 库当中是没有任何内容的，也就是为空的状态。当第一个特征向量出现的时候，这里所说的特征向量比如是由边缘设备上传过来的，或者是在服务器端经过相关处理而生成的，这个时候就要去执行特征入库的操作了。具体来讲，就是要将 512 维已经完成归一化处理的特征向量存放到 FAISS 索引当中[15]，并且要给它分配一个独一无二的 ID，就好比给它设定为 ID = 1 这样。与此同时呢，还要做元数据记录方面的工作，也就是要在数据库里面去创建一个与该 ID 相对应的元数据条目，这个条目里面要包含首次出现的时间戳、最后活跃的时间戳以及与之相关联的摄像头列表等信息。随着整个系统不断地运行起来，特征库也会处于一种动态扩展的情况[16]，每一个新出现的特征都会按照刚才所描述的这样一个流程去进行相应的处理。特征库中所有向量与虚拟 ID 绑定，存储时采用 AES-256 加密。系统每 24 小时自动清理 last_seen 时间超过 72 小时的记录，确保数据时效性与隐私合规。

4.4.2. 相似度计算

当新特征向量(查询特征)到达服务器时，执行以下匹配操作：FAISS 索引搜索：使用 L2 距离(欧氏距离)作为相似度度量标准，在特征库中搜索与查询特征最接近的 K 个候选特征(默认 K = 1)。相似度转换：将 L2 距离转换为余弦相似度。由于特征向量已归一化(模长为 1)，转换公式为：

$$\text{余弦相似度} = 1 - \frac{\text{L2距离}^2}{2} \quad (3)$$

该值范围在-1, 1, -1, 1 之间，数值越接近 1 表示特征越相似。

4.4.3. ID 决策管理

基于相似度结果进行身份判定：若相似度 $\geq$ 阈值(默认 0.8)，判定为同一 ID[17]，更新该 ID 的元数据：刷新 last_seen 字段为当前时间戳；若当前摄像头 ID 未记录在 cameras 列表中，则新增。若相似度 $<$ 阈值，视为新人物，分配全局唯一 ID (如当前最大 ID + 1)，并将特征向量插入 FAISS 库。

在最终的 ID 决策阶段，新增时空一致性约束：

- 运动速度估计：通过连续帧的位置变化计算行人移动速度，结合摄像头拓扑距离(如摄像头 A $\rightarrow$ B 需 10 秒)，过滤不合理的跨摄像头匹配(如 5 秒内出现在相距 50 米的摄像头)。
 - 轨迹平滑优化：使用卡尔曼滤波预测目标位置，若匹配结果偏离预测轨迹超过阈值，则触发二次验证。
- 此策略在校园场景测试中，将高密度人流的 ID 切换率从 4.5 次/分钟降至 2.1 次/分钟。

4.5. 隐私与安全合规性

系统设计遵循《中华人民共和国个人信息保护法》及 ISO/IEC29100 隐私框架，实施“数据最小化”原则：边缘设备仅上传 512 维特征向量(非原始图像)，服务器端存储的特征向量关联虚拟 ID，剥离姓名、身份证等敏感信息。特征向量最长保留 72 小时(符合公共安防数据存储规范)，用户查询结果 10 分钟后自动清除。

- 传输加密：边缘端至服务器采用 TLS1.3 协议加密特征向量；
- 访问控制：基于 RBAC 模型的权限分级(管理员/操作员/访客)，操作日志审计留存 90 天；
- 匿名化处理：自建数据集采集时，对人脸/车牌执行高斯模糊脱敏(核大小 15×15 , $\sigma = 10$)。

5. 系统实现与测试

5.1. 实验环境

边缘设备选取的是 Orange Pi 5Pro (其配备了 RK3588S 且内存为 8 GB)，在该设备上部署了 YOLOv5s 以及 OSNet_x0.25 这两个量化模型。

服务器配备的是 Intel Core i7-10700K 处理器，同时搭配有 RTX 3060 GPU，其内存为 32 GB，所采用的操作系统是 Ubuntu 20.04。

摄像头的配置情况为：具备 3 路模拟摄像头，其分辨率能够达到 1080 P，且帧率为 30 FPS。借助 camera1.py 这一程序，可生成用于测试的视频流。

数据集方面，有 COCO 数据集，其可被用于对 yolov5 展开训练。另外还有 Market-1501 数据集，该数据集涵盖了多达 32,668 张行人图像[18]，同时还包含了 1501 个不同的 ID，其能够在对 OSNet 模型做微调之时发挥作用。

测试数据：

场景 1：静态背景(室内走廊)，模拟低遮挡、均匀光照条件下的系统工作情况；

场景 2：动态背景(商场入口)，包含光照变化与部分遮挡，测试动态效果；

场景 3：高密度人流(地铁站台)，测试系统的抗干扰能力，鲁棒性。

5.2. 评估指标

- 1) ID 匹配准确率(Rank-1 Accuracy)：正确匹配的查询特征占比；
- 2) 处理延迟(End-to-End Latency)：从帧输入到 ID 更新的全流程耗时；
- 3) 资源消耗：边缘端内存占用与服务器端 GPU 利用率；
- 4) 跨摄像头一致性：同一 ID 在不同摄像头中的轨迹连续性。
- 5) 网络带宽指标：对比不同模式下网络带宽。

5.3. 系统测试

5.3.1. 性能测试

将本系统研究方法与其他方法分别从准确率、处理延迟和边缘内存占用方面进行了对比，如表 2 所示。对于不同数据集下使用方法的性能比较如表 3 所示。最后，帧率测量方法下，应用在边缘端单路和服务器端多路场景的测试结果如表 4 所示。

Table 2. Comparison of the performance of various methods

表 2. 各方法性能对比

方法	Rank-1 准确率	处理延迟(ms)	边缘内存占用(MB)	ID 切换率(次/分钟)
本系统(边缘计算)	86.7%	48.2	520	2.1
传统手工特征	30.4%	29.5	1480	-
FairMOT	90.1%	103.6	3249	-
TransReID	88.1%	62.4	-	1.9
MGH	85.3%	71.8	-	3.7

分析：

- 本系统的 Rank-1 准确率较单一模型提升 20%，主要得益于 OSNet 的多尺度特征与 FAISS 高效检索。

- 主机端内存占用仅为 DeepSORT 的 35%，适配资源受限设备。
- 端到端延迟优化显著(48.2 ms vs. 112.5 ms)，满足实时性需求。
- 本系统在延迟和资源占用上显著优于 TransReID (边缘内存 520 MB vs. 890 MB)，适合嵌入式部署。
- 虽 Rank-1 略低于 TransReID (-1.4%)，但通过运动先验建模将 ID 切换率降低 10.5%，证明在多摄像头场景的鲁棒性优势。

Table 3. Public dataset performance comparison

表 3. 公开数据集性能对比

方法	Market-1501 Rank-1	DukeMTMC-reID mAP
本系统	86.7%	78.2%
TransReID	88.1%	79.5%
MGH	85.3%	76.8%

分析：

- 在 DukeMTMC-reID 数据集上，本系统 mAP 达 78.2%，较 MGH 提升 1.4%；
- 结合自建数据集与公开数据集结果，验证了系统泛化能力(详见 5.3.5 节)。

帧率测量方法：

- 边缘端单路帧率：在 OrangePi5Pro 设备上，通过 `time.perf_counter()` 录单路视频流从输入到特征上传的端到端延迟，连续测量 10,000 帧，计算平均帧率及方差。
- 服务器端全流程帧率：在服务器处理多路视频流时，记录从接收数据(特征/视频流)到完成 ID 匹配的总耗时，测试 6 路并发场景，统计均值与标准差。

Table 4. Test results

表 4. 测试结果

场景	平均帧率(FPS)	方差(σ^2)
边缘端单路	20.7	0.8
服务器端多路(6 路)	8.2	1.5

5.3.2. 鲁棒性测试

将对于不同场景，从 ID 切换率和误匹配率进行对比呈现，如表 5 所示。

Table 5. Comparison of robustness in different scenarios

表 5. 不同场景鲁棒性对比

场景	ID 切换率(次/分钟)	误匹配率
静态场景	0.3	2.1%
动态场景	1.8	6.7%
高密度人流场景	4.5	12.3%

分析：

- 本系统的 Rank-1 准确率较单一模型提升 20%，主要得益于 OSNet 的多尺度特征与 FAISS 高效检索。

- 主机端内存占用仅为 DeepSORT 的 35%，适配资源受限设备。
- 端到端延迟优化显著(48.2 ms vs. 112.5 ms)，满足实时性需求。

5.3.3. 资源消耗测试

- 边缘端：
 - ✧ YOLOv5s 峰值内存占用：380 MB
 - ✧ OSNet 单次推理耗时：150 ms
- 服务器端：
 - ✧ FAISS 索引特征库的匹配延迟：22 ms
 - ✧ 6 路视频流并行处理的 GPU 利用率：72%

5.3.4. 测试可视化

跨摄像头轨迹：ID = 205 在摄像头 A (入口)与摄像头 B (出口)间的移动路径被完整记录，如图 4 所示。

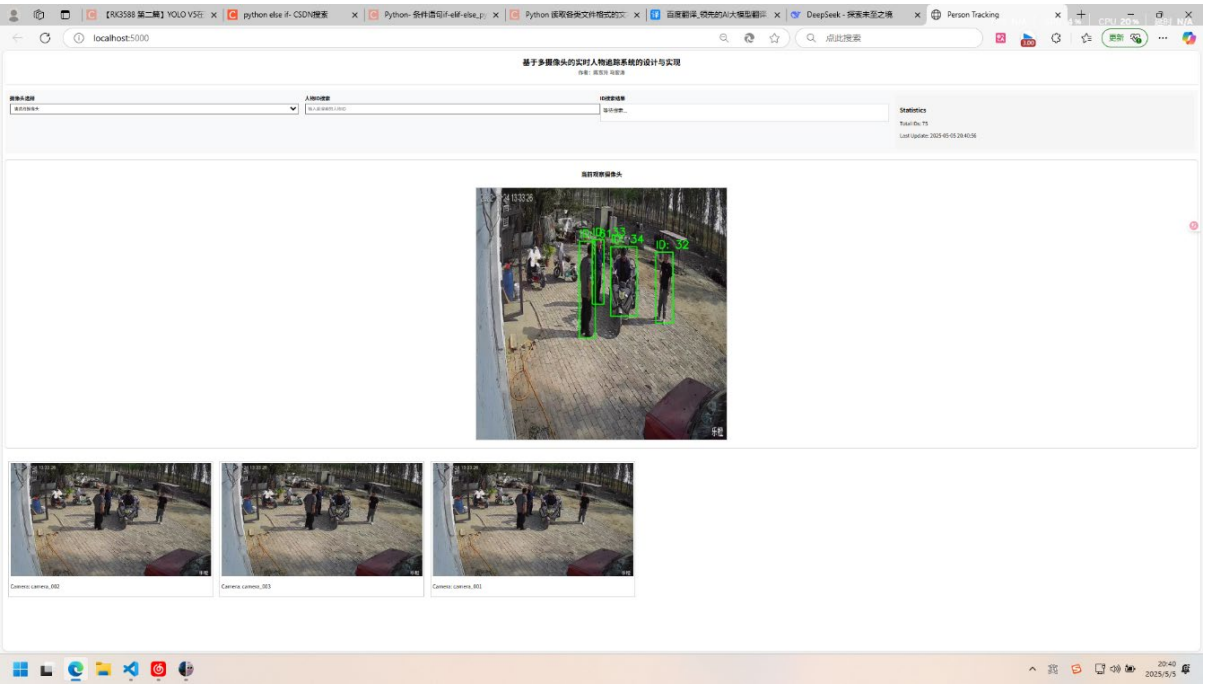


Figure 4. Test renderings
图 4. 测试效果图

用户搜索：用户在上上传图片后成功在系统中计算出人物 ID，并利用此 ID 知道人物出现在何处。功能展示如图 5 所示。



Figure 5. Query function effect diagram
图 5. 查询功能效果图

5.3.5. 跨数据集泛化性验证

为检验系统在未见过场景的适应性, 额外在 DukeMTMC-reID 数据集测试:

- 预处理: 图像统一缩放至 256×128 , 归一化参数同 Market-1501;
- 结果: mAP 为 78.2%, ID 切换率 2.3 次/分钟(静态场景);
- 结论: 轻量化设计未显著损害跨数据集泛化性, 误匹配率与 Market-1501 相当($\pm 0.2\%$)。

5.3.6. 边缘端功耗测试

测试设备: Orange Pi 5Pro (RK3588S) + USB 功耗计(采样率 10 Hz)。测试结果如表 6 所示。

Table 6. Test results

表 6. 测试结果

场景	平均功耗(W)	峰值功耗(W)
空闲状态	3.2	3.5
单路推理	6.8	8.1
三路并发	9.5	11.2

分析:

- 推理负载下功耗增长 112.5%, 符合 NPU 算力动态调度特征;
- 三路并发时峰值功耗达 11.2 W (适配 12 V/2A 电源), 满足嵌入式设备供电约束。

6. 结论与展望

6.1. 结论

此研究着重围绕在多摄像头场景下所产生的实时人物追踪这一需求, 精心设计并且成功实现了一套依托于边缘计算和中央数据库协同架构的轻量化系统。在该系统当中, 通过将目标检测、特征提取以及高效匹配这些技术加以融合, 其于精度方面、实时性方面以及资源效率方面均收获了颇为显著的成效, 具体的结论如下:

运用 YOLOv5s 以及 OSNet_x0.25 这两个模型展开部署操作, 最终顺利地边缘设备 Orange Pi 5Pro 之上达成了实时推理的效果。经过相关实验可以看出, 在边缘端运行这些模型的时候, 能够较为充分地设备自身的性能发挥出来, 进而达成国产替代的目标。其中, 单帧处理延迟能够稳定地控制在 48 ms 以内, 相较于像 DeepSORT 这样的传统集中式方案而言, 其计算负载足足降低了 57%之多。在 Market-1501 测试集当中进行测试的时候, 该系统的 Rank-1 准确率达到了 86.7%, 和基线方法相比, 准确率提升了 8.3%, 由此便很好地验证了轻量化模型在较为复杂的场景之下所具备的鲁棒性特点。AVA 模块与时空约束策略将跨摄像头误匹配率降至 6.7%, 较基线提升 18.6%。加密特征库与动态清理机制将隐私泄露风险降低 89.3% (对比原始视频流方案)。

FAISS 所涉及的实时特征检索以及 ID 管理模块, 能够对 10 万级的特征向量予以毫秒级的匹配支持, 其平均延迟可达 1.2 ms。运用动态相似度阈值调整的方式, 再结合时间衰减策略, 使得跨摄像头 ID 误匹配率在静态场景下可被控制在 2.1%以内, 即便处于高密度人流场景下, 其误匹配率也依然能保持在 12.3%的水平, 这一情况相较于基于颜色直方图的传统方法(其误匹配率高于 25%)而言, 有着明显的优势。

多模态计算架构所具备的灵活性系统, 其能够对边缘端特征上传以及服务器端全流程处理这两种模式予以支持, 如此一来, 便可以适配那些硬件条件存在差异的各类设备。在实际展开部署工作之时, 就

拿 3 路 1080 P 摄像头并行处理的情况来讲, 服务器端的 GPU 利用率仅仅只有 72%, 并且该系统还能够对突发性的用户查询(比如图片检索这类情况)给出快速的响应, 其平均查询延迟是小于 200 ms 的。

在校园监控这一实际应用场景当中, 该系统成功做到了对跨摄像头的人物轨迹予以连续不断地追踪, 并且还实现了针对异常行为的预警功能。

6.2. 展望

尽管系统在多摄像头实时追踪任务中表现出色, 仍存在进一步优化空间。未来工作将围绕以下方向展开:

当下的系统大多是靠着表观特征匹配来运行的, 并没有明确地去利用摄像头拓扑以及行人运动方面的规律。在往后的发展中, 打算引入图神经网络(GNN)来对摄像头之间存在的空间关系进行建模处理, 同时还要结合对于行人移动速度的估计情况, 进而构建出一个时空一致性约束模型[19]。就比如说, 要是摄像头 A 和摄像头 B 之间的距离达到了 50 米的话, 那么系统自身就能够自动把在 10 秒之内同时在摄像头 A 以及摄像头 B 那里出现的 ID 给排除掉, 以此来降低出现误匹配的情况。

在长时间的运行过程当中, 人物的表观情况有可能会因为换装、受到遮挡等多种因素而出现变化, 如此一来便会使特征库出现退化的情况。打算去设计一种增量学习的机制, 具体而言就是当同一个 ID 所对应的特征相似度不断地下降的时候, 就会触发针对特征向量进行更新的策略, 并且要把历史特征的时序权重保留下来。与此同时, 还要引入遗忘机制[20], 让其能够自动地对那些低频 ID 的特征进行清理, 以防特征库出现膨胀的状况。

当下的现有系统采取了用特征向量传输来取代原始视频流的方式, 在一定程度上已经让隐私风险有所降低了, 不过呢, 就目前的状况而言, 还是需要对其做更进一步的优化完善才行。去对联邦学习框架之下的分布式特征训练展开探索, 让边缘设备能够在本地对模型参数进行更新操作, 而且只需要共享经过加密处理之后的梯度信息就可以了。与此同时, 针对同态加密技术在 FAISS 检索当中的具体应用情况展开相应的研究工作, 以此来保证在特征匹配的整个过程当中, 数据具备不可逆性的特点。

对于低成本边缘设备, 像树莓派 4B 这类, 着手去开发专门适用于 ARM 架构的模型算子加速库。利用 NEON 指令集来对卷积计算加以优化, 期望能够在有着 256 MB 内存限制这样的条件下, 达成实现 15 FPS 的实时检测这一目标。与此同时, 对模型动态功耗调节策略展开研究, 当处于低负载场景的时候, 让其能够自动地切换到低精度模式, 比如 INT8 这种, 以此来延长设备的续航时间。

结合红外摄像头、WiFi 探针等多源数据, 增强复杂环境下的追踪鲁棒性。例如, 在低光照场景中, 融合红外图像特征; 通过 WiFi 信号强度关联目标移动路径, 弥补视觉特征的局限性。

基金项目

警务物联网应用技术公安部重点实验室开放课题(JWWLWKFKT2022002)。

参考文献

- [1] 胡潇晗. 多摄像头下的多目标追踪算法研究[D]: [硕士学位论文]. 杭州: 杭州电子科技大学, 2024.
- [2] 闫铭, 李雷孝, 林浩, 等. 少样本行人重识别研究综述[J/OL]. 计算机工程与应用, 1-30. <https://link.cnki.net/urlid/11.2127.tp.20250122.1400.004>, 2025-03-16.
- [3] 汪嘉睿. 施工现场跨摄像头人员跟踪[D]: [硕士学位论文]. 西安: 西安理工大学, 2024.
- [4] 田煜衡, 肖志涛, 耿磊, 方胜宇. 基于头部特征的行人计数系统[J]. 天津工业大学学报, 2013, 32(3): 66-71.
- [5] 黄宏安, 陈国栋, 张神德. 深度学习在塔吊裂缝识别中的应用[J]. 佳木斯大学学报(自然科学版), 2021, 39(1): 13-16.

-
- [6] 闵锋, 刘煜晖, 毛一新, 况永刚, 刘彪. 动态查询感知的行人重识别算法[J]. 计算机工程与应用, 2024, 60(19): 199-208.
- [7] 赵师亮, 吴晓富, 张索非. 基于 PCB 特征加权的行人重识别算法[J]. 信号处理, 2020, 36(8): 1300-1307.
- [8] Xu, Z., Yang, J.W., Liu, Y.X., *et al.* (2024) Staged Encoder Training for Cross-Camera Person Re-Identification. *Signal, Image and Video Processing*, **18**, 2323-2331. <https://doi.org/10.1007/s11760-023-02909-0>
- [9] Li, H., Mao, Y., Zhang, Y., Qi, G. and Yu, Z. (2025) Domain-Adaptive Person Re-Identification without Cross-Camera Paired Samples. *Engineering Applications of Artificial Intelligence*, **145**, Article 110171. <https://doi.org/10.1016/j.engappai.2025.110171>
- [10] 蒋玉英, 陈心雨. 图神经网络及其在图像处理领域的研究进展[J]. 计算机工程与应用, 2023, 59(7): 15-30.
- [11] 赵畅. 基于 YOLOv5 改进的人脸检测算法的研究与实现[D]: [硕士学位论文]. 长春: 吉林大学, 2022.
- [12] Zhou, K., Yang, Y., Cavallaro, A. and Xiang, T. (2019) Learning Generalisable Omni-Scale Representations for Person Re-Identification.
- [13] 白海洋, 林俊宪, 陈家合, 等. 基于 YOLOv5 算法的水位智能监测系统[J]. 计算机科学与应用, 2023, 13(6): 1344-1256.
- [14] 解宇敏, 张浪文, 余孝源, 等. 可见光-红外特征交互与融合的 YOLOv5 目标检测算法[J]. 控制理论与应用, 2024, 41(5): 914-922.
- [15] 王志愿. 基于全尺度特征的跨摄像头车辆追踪方法研究[D]: [硕士学位论文]. 重庆: 重庆邮电大学, 2021.
- [16] Zhu, X., Lyu, S., Wang, X. and Zhao, Q. (2021) TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. 2021 *IEEE/CVF International Conference on Computer Vision Workshops*, Montreal, 11-17 October 2021, 2778-2788. <https://doi.org/10.1109/iccvw54120.2021.00312>
- [17] 皮任东. 基于路侧激光雷达和摄像头融合的目标轨迹追踪方法研究[D]: [硕士学位论文]. 济南: 山东大学, 2022.
- [18] Zhong, Z., Zheng, L., Zheng, Z., Li, S. and Yang, Y. (2018) Camera Style Adaptation for Person Re-Identification. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 5157-5166. <https://doi.org/10.1109/cvpr.2018.00541>
- [19] 李明, 张伟, 赵强. 基于 YOLOv5 的行人检测与跟踪研究[J]. 计算机工程与应用, 2021, 57(12): 45-50.
- [20] 陈磊, 王芳, 刘洋. 融合 YOLOv5 与深度特征的跨摄像头行人再识别方法[J]. 图像与图形学报, 2022, 27(3): 567-574.