题 目：校园安防监测系统的设计与实现Design and Implementation of Campus Security Monitoring System

学院： 软件学院 专业： 软件工程 组号： 1-6组

注：1表示8点课堂，2表示10点课堂

**分工明细表（全文文字共计 19364字，不算参考文献和图；**有效图个数12**）**

**小组自行填写A和B的值**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 成员 | 组长 | 组员1 | 组员2 | 组员3 | 组员4 |
| 学号 | 22301061 | 22301103 | 22301114 | 22301030 | 22301154 |
| 姓名 | 陈曦红 | 谢雨桐 | 周英 | 赵雨滔 | 马海博 |
| 章节分工（可修改分工） | 中文摘要、  第5章详细设计 | 第1章引言  （研发现状）参考文献、  目录 | 第3章  需求分析 | 第4章  概要设计  第6章  系统测试 | 第2章  校园安防监控系统相关理论及技术综述  第7章  总结与展望 |
| 撰写文字的  字数 | 3582 | 3056 | 5940 | 3103 | 3683 |
| **绘图编号（自己绘制的，copy的不算）** | 图5-1，图5-2 | 图5-18，图5-19，图5-20，图5-12 |  | 图4-1，图4-2，图4-3，图5-11 | 图5-13， 图5-3 |
| 占比 | 18.2% | 18.5% | 25.8% | 18.7% | 18.6% |

**注：占比=（个人撰写的有效文字数+个人绘制的有效图个数\*300）/（全文有效的文字数+全文有效图的个数\*300）**

# 总分：30分

**小组互评打分明细 (每人至少指出2处问题)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **指出问题所在**  **（**什么地方，违反了哪一条规范？） | **修改**  **（**应该怎么改，给出修改后的结果） | **该项满分** | **打分** | **修改人** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |



本科毕业设计（论文）

**校园安防监测系统的设计与实现**

**Design and Implementation of Campus Security Monitoring System**

学 院：

专 业：

学生姓名：

学 号：

指导教师：

**北京交通大学**

2024年11月

中文摘要【撰写人姓名：陈曦红】

本项目设计并实现了一个校园安防监测系统，以满足当今校园对于高效安全管理的需求。随着校园开放性增强，外来人员频繁出入会带来一定的安全风险，因此，通过引入计算机视觉和深度学习技术，提升校园安防智能化水平成为亟待解决的问题。本系统结合Dlib库与预训练的ResNet模型，完成了人脸识别与身份验证，保证了人员出入的高效管理。同时，为了防止身份冒用行为，系统加入了基于眼动检测的活体检测功能，能有效分辨真实用户与照片、视频等虚假验证。此外，系统集成了YOLOv5深度学习算法，通过对实时视频流的多帧处理与目标检测，精准识别校园内的异常事件，例如持刀、火灾、黑名单人员进入及在人员在危险区域逗留等情况，并在异常发生时触发报警。系统还通过行人重识别技术支持多摄像头协同，针对同一用户在不同摄像头下的识别与追踪进行优化，实现跨区域的行人重识别。实验结果显示，系统在身份验证、异常事件检测及行人重识别方面均达到了80%以上的准确率和召回率，有效提高了校园安全管理的实时性与稳定性，验证了该系统在智慧校园建设中的重要应用价值。

**关键词：校园安防、人脸识别、行人重识别、活体检测、YOLOv5**

目 录【撰写人姓名：谢雨桐】

[中文摘要 i](#_Toc1225336790)

[目 录 ii](#_Toc2014738447)

[1 引言 1](#_Toc186932833)

[1.1 项目背景 1](#_Toc11548670)

[1.2 安防监测产品研发现状 1](#_Toc824968460)

[1.3 项目主要内容 2](#_Toc1108690218)

[1.4 本文组织结构 2](#_Toc40888907)

[2 校园安防监控系统相关理论及技术综述 4](#_Toc25092909)

[2.1 SpringBoot框架 4](#_Toc829726751)

[2.2 Flask框架 4](#_Toc1606184086)

[2.3 Dlib人脸识别技术 5](#_Toc1266490612)

[2.3.1 Dlib的特性 5](#_Toc49806820)

[2.3.2 基础算法模块 5](#_Toc1732085057)

[2.3.3 人脸检测与识别模块 5](#_Toc2012717914)

[2.4 EAR活体检测技术 6](#_Toc587573054)

[2.5 Yolov5模型 6](#_Toc1210509672)

[2.5.1 输入层 6](#_Toc1923469273)

[2.5.2 Backbone 7](#_Toc1676733020)

[2.5.3 Neck 7](#_Toc1571451206)

[2.5.4 Head 7](#_Toc1626528436)

[2.6 本章小结 7](#_Toc1744081189)

[3 校园安全监测系统中的需求分析 8](#_Toc1768245620)

[3.1 需求分析综述 8](#_Toc2025428154)

[3.2 功能性需求分析 9](#_Toc1607695681)

[3.2.1 身份验证模块 9](#_Toc902064013)

[3.2.2 异常事件识别模块 13](#_Toc1902802318)

[3.2.3 行人重识别模块 18](#_Toc393869206)

[3.3 非功能性需求分析 21](#_Toc1215145188)

[3.4 本章小结 22](#_Toc375691746)

[4 校园安全监测系统中的概要设计 23](#_Toc649252842)

[4.1 系统体系结构 23](#_Toc628105087)

[4.2 系统功能结构 24](#_Toc1680072204)

[4.3 系统数据库设计 25](#_Toc1858541872)

[4.4 本章小结 29](#_Toc1363597089)

[5 校园安全监测系统中的详细设计与实现 30](#_Toc30794039)

[5.1基于人脸识别的身份验证模块 30](#_Toc11854546)

[5.1.1身份验证模块流程设计 30](#_Toc1670859098)

[5.1.2身份验证模块类设计 31](#_Toc1632691914)

[5.1.3身份验证模块时序设计 32](#_Toc106957232)

[5.1.4人脸识别模块实现   33](#_Toc186385685)

[5.2异常事件识别模块 37](#_Toc1553050469)

[5.2.1异常事件识别模块流程设计 37](#_Toc1602986845)

[5.2.2异常事件识别模块类设计 38](#_Toc1217552300)

[5.2.3异常事件识别模块时序设计 39](#_Toc29833837)

[5.2.4异常事件识别模块实现 40](#_Toc1053608708)

[5.3行人重识别模块 42](#_Toc1998885841)

[5.3.1行人重识别模块流程设计 42](#_Toc40156019)

[5.3.2行人重识别模块类设计 43](#_Toc592346175)

[5.3.3行人重识别模块时序设计 43](#_Toc1975459380)

[5.3.4行人重识别模块实现 44](#_Toc1448617040)

[5.4本章小结 44](#_Toc884485241)

[6 系统测试 45](#_Toc661640953)

[6.1 测试目标及准备 45](#_Toc529172905)

[6.2 功能测试 45](#_Toc1079232108)

[6.3 非功能测试 46](#_Toc1007156594)

[6.4 本章小结 47](#_Toc814769704)

[7 总结与展望 47](#_Toc1478681856)

[7.1 全文总结 48](#_Toc1525190708)

[7.2 系统展望 48](#_Toc1515418764)

[参考文献 50](#_Toc487113128)

1. 引言【撰写人：谢雨桐】

目前，安防智能化水平不断提高。然而，校园安全防范工作较薄弱，传统的安防技术已不能满足现在的需求，使用新技术来完善校园安防建设显得尤为重要，通过人脸识别和目标检测等功能，提高校园安防水平。本章着重介绍本项目的应用背景、发展现状；同时介绍本项目的工作内容、实现的技术方法以及全文组织结构。

## 1.1 项目背景

随着校园开放程度的提高，外来人员频繁进入校园，给管理带来了巨大的安全挑战。传统的安防监测方式依赖于人工巡逻和单一的监控设备，不仅效率低下，还难以满足突发事件的快速响应需求。同时，近年来以人脸识别与目标检测为核心的智能监控技术逐步普及，为校园安全管理提供了新的技术路径。

人脸识别是一种基于人类面部生物特征进行身份识别的一种方法，具有唯一性和稳定性，已经广泛应用于金融、医疗、游戏、交通运输、刑侦查案、安保系统等领域[1]。将人脸识别技术应用于校园安防监测系统中，能够实现快速识别人员身份。目标检测是对目标进行定位和分类的技术，可以准确高效地找出给定图像中所有感兴趣的目标，并用矩形边界框来定位被检测目标的位置和大小[2]。

本项目设计的系统能够通过身份验证和行为识别技术，显著提高校园安全管理水平，减少安全事故的发生，保护师生的生命财产安全，具有重要的社会意义与经济价值。

## 1.2 安防监测产品研发现状

随着信息技术的发展，基于人脸识别与目标检测的智能监测系统已在校园管理和安防领域中获得广泛应用。目前多数学校安防设施仅建立了功能单一的视频监控系统，无法有效识别安全隐患。此外，该监控方式对安保人员执勤要求也较高，必须24小时关注监控画面，已不能满足当前校园安防的建设要求。

董新颖[3]研究出一种以机器视觉技术为基础的校园安防智能识别系统。该系统可以对出入校门的人脸影像图进行动态采集，并根据OpenCv等算法，对这些采集到的影像图进行动态分析，然后对这些影像图的人脸特征进行检测。韩等人[4]借助领先的人工智能技术，将人脸识别、智能分析技术与传统的校园安防模式相结合，将语音识别、图像识别与传统的教学模式相结合，提出了校园信息化建设整体解决方案。刘[5]等人提出了一种高校校园智能安防体系框架，具有多维感知、异构互联、智能高效、综合管控、集中呈现和云端管理的特点。

总体来看，现有产品仍存在以下问题：

(1)现有的产品在识别身份冒用行为的效果不佳，难以有效区分真实用户和使用图片或视频冒充他人的虚假用户。

1. 现有系统缺乏对异常事件的实时监测和预警功能，无法有效识别安全隐患。
2. 现有系统在多摄像机场景下，行人重识别的预测精度较低。

本文通过集成YOLOv5（You Only Look Once version5，你只看一次第五版）模型和ResNet（Residual Network，残差神经网络）模型以及活体检测技术，为校园安全提供了更加全面的解决方案。

## 1.3 项目主要内容

本项目在现有的安防系统基础上，实现基于人脸识别的身份验证、异常事件识别报警以及行人重识别的智能安防监测系统。

系统主要内容包括：

1. 基于人脸识别的身份验证功能。在校园门禁处，系统采用Dlib库[11]和ResNet模型进行高精度人脸识别，同时通过基于眼动检测的活体检测技术有效防范身份冒用行为。识别完成后，根据识别结果显示用户的姓名、学工号等信息。
2. 异常事件识别功能。系统中的异常事件分为人员持刀、火灾、在危险区域内停留3秒以上以及黑名单用户进入校园这四类事件。本系统基于YOLOv5模型，对视频流中的异常行为进行实时监测，并且异常事件的监测响应速度在1s以内。
3. 行人重识别功能[12]。系统接入多个摄像头实时捕捉动态视频流，对出现在不同摄像头中的同一用户进行有效重识别，同时识别的准确率和召回率有80%以上。

## 1.4 本文组织结构

本文共分为七章，具体内容安排如下：

第一章引言，主要介绍项目背景和意义，以及校园安防检测产品研究现状，同时阐述该项目的主要内容。

第二章相关理论与技术综述。介绍项目中所涉及的理论和技术，包括SpringBoot框架、Flask框架、Dlib人脸识别技术、EAR活体检测技术、YOLOv5模型。

第三章校园安防监测系统的需求分析。介绍系统的总体架构与功能模块设计，包括身份验证、异常事件检测和行人重识别模块。

第四章校园安防监测系统的概要设计与分析。介绍系统总体的概要设计，包括系统总体架构、功能模块设计方案和数据库设计方案。

第五章校园安防监测系统的详细设计与实现。在概要设计的基础上，对各模块进行了详细描述，展示模块设计的详细类图、顺序图。

第六章系统测试，以需求分析为基础，对系统功能进行验证。

第七章结论与展望。总结本文工作的主要成果，并对未来发展进行展望。

1. 校园安防监控系统相关理论及技术综述 【撰写人：马海博】

本章对校园安防监控系统中使用到的理论和技术基础进行简要介绍。系统后端架构结合了SpringBoot框架和Flask框架，前端采用vue框架，数据库使用Mysql搭建。身份验证通过Dlib进行人脸识别。活体检测利用眼动特征计算EAR值(Eye Aspect Ratio，眼睛长宽比)判断用户是否为真人。系统采用 YOLOv5 模型实时检测多路摄像头视频流，识别行人特征并实现跨摄像头匹配与轨迹追踪，同时检测异常事件的发生。

## 2.1 SpringBoot框架

Spring Boot 是一种简化 Java Web 应用开发的轻量级框架，通过约定优于配置的原则、依赖注入，以及去 XML 配置等特性，显著提升了开发效率。相较于传统 JavaEE 项目开发，Spring Boot 通过自动化配置（如starter POMs）、内嵌的 Web 容器等特性，实现了模块化，简化了开发与部署。

此外，Spring Boot 还集成了丰富的监控和测试工具。 Actuator 提供多种端点以监控应用的运行状态和系统信息，并结合 JMX 实现细致管理。MockMvc则支持便捷的HTTP请求模拟，并进行自动化测试。

目前，Springboot正被广泛地应用于Web项目的开发，并逐渐成为主流技术之一[6]。

## 2.2 Flask框架

Flask 是一种广泛应用的 Python 微框架，以轻量级和扩展性著称，专为 Web 应用和服务的快速开发而设计。作为微框架，Flask 保持核心简洁，没有强制的数据库和模板引擎要求，允许灵活选择并添加所需的扩展功能。

Flask主要依赖的两个核心库是 Werkzeug 和 Jinja2。Werkzeug 提供 URL 路由、请求管理、会话和身份认证支持等 Web 基础功能，Jinja2 则提供模板渲染和 HTML 转义功能，有效防护脚本攻击。此外，Flask 提供丰富的插件库，可以轻松实现表单验证、文件上传、数据库集成等多种功能。凭借模块化和高效的优势，Flask逐渐成为轻量级 Web 项目开发中的首选技术[7]。

## 2.3 Dlib人脸识别技术

人脸识别身份验证功能的核心是Dlib库，Dlib库可以完成对人脸特征点的提取以及进行人脸检测与识别。其具有可移植性和鲁棒性，并且由基础算法模块、人脸检测与识别模块等模块组成。下面依次对Dlib库的特性与模块进行介绍。

### 2.3.1 Dlib的特性

Dlib 是一个基于 C++ 开发的开源数据工具库，以其在图像处理和人脸识别中的优越性能而著称。Dlib提供强大的特征提取和识别功能，具有高可移植性和鲁棒性，广泛应用于多种应用场景。

（1）Dlib 代码高度封装，无需第三方依赖，可轻松集成至各种项目。

（2）Dlib 提供详细的文档说明，每个功能模块和文件均带有注释，方便开发者调用。

### 2.3.2 基础算法模块

Dlib库提供机器学习和图像处理的常见方法，包括支持向量机、递归算法、大规模分类降维方法以及贝叶斯相关向量机等[8]。这些算法有助于高效实现从特征提取到模型训练的关键步骤。在人脸识别方面，Dlib 支持特征点检测、图像对齐等处理步骤，能够快速生成标准化的特征点图像，实现高精度人脸识别。

### 2.3.3 人脸检测与识别模块

Dlib进行人脸识别的关键为ResNet（Residual Neural Network，残差神经网络）是一种基于残差学习的深度神经网络。其核心结构是残差模块，其中通过“恒等映射”将输入变量 x 直接绕过部分网络层，通过“跳跃连接”与“残差映射”的输出 F(x) 进行相加，形成最终的输出 F(x) + x。如图2-1所示。

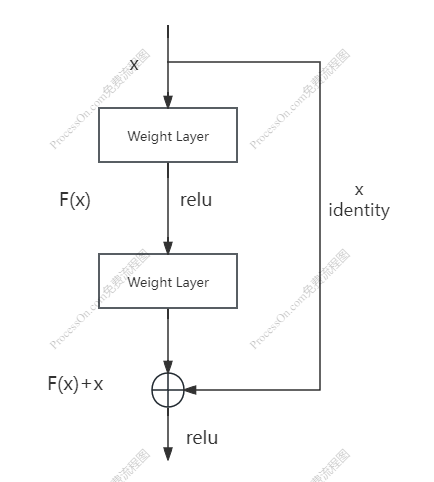


图2-1 Resnet模块结构

残差模块的设计有效解决了深度网络中的梯度消失问题，确保数据在传输过程中的完整性，从而提升了网络的训练效率。依靠这种结构，ResNet 可以在加深网络深度的同时保持训练速度，并不断优化模型的准确性。这种结构不仅降低了模型在分析问题时的复杂度，还能够实现比传统神经网络更快的训练速度和更高的准确性，使其在深度学习任务中表现优异。

在本项目中采用的Resnet模型为Resnet34。通过 ResNet34 将人脸特征向量转换为128维向量；最后，计算目标人脸与已知人脸数据集的特征向量间的欧式距离，若距离低于设定的阈值，便可认定为同一个人。

## 2.4 EAR活体检测技术

EAR是一项基于眼睛开合状态变化的生物特征参数，广泛应用于视频监控和人脸识别的活体检测中。通过分析眼动变化，EAR能够有效识别人眼在自然状态下的闭合与张开过程，从而实现虚假身份信息的检测[9]。

首先通过普通的摄像头捕捉视频帧并检测人脸区域,获取人眼关键点后计算EAR值。计算基于眼睛的六个关键点，通过两条垂直距离与水平距离的比值得出。具体公式为：

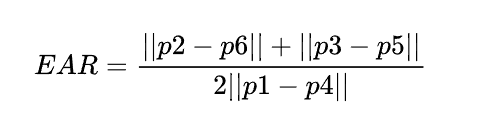


图2-2 EAR计算公式

其中p1,p2,p3,p4,p5,p6为眼睛关键点，利用该公式可以测量眼睛的开闭程度。通过EAR反映眼睛开闭程度的动态变化，以区分真人与假体。对于真人，眨眼会导致EAR值显著降低，而在视频或照片中，EAR值通常无自然波动。设定一个EAR阈值，若EAR值低于此阈值则可视为闭眼。若在一定时间内检测到频繁的EAR变化，即判定为眨眼行为，进而判定为活体。反之，如检测到的EAR值无显著变化，则判定为非活体。

通过EAR值的变化识别虚假身份信息验证，在需要保证安全性和实时性的人脸识别系统中应用广泛，能够有效降低欺骗攻击的风险，同时保持较高的用户体验。

## 2.5 Yolov5模型

YOLOv5是一种高效的单阶段目标检测算法，以其实时性能和高检测精度被广泛应用。其架构包括输入层、Backbone、Neck和Head四个主要模块[10]。下面依次对这四个模块进行介绍。

### 2.5.1 输入层

输入层的功能是对输入图像进行预处理，增强数据的多样性并优化锚框的适应性。通过Mosaic数据增强，YOLOv5将四张图像拼接在一起，增强训练数据的多样性。同时，YOLOv5通过计算不同目标的宽度和高度的比值，自动选择最佳的锚框尺寸，对锚框进行自适应调整，使得锚框更好地适配目标的大小和形状，提高识别的精度。

### 2.5.2 Backbone

Backbone部分负责从输入图像中提取特征。Focus模块通过将输入图像下采样以更好地学习图像中的细节特征。结合CSP结构（Cross-Stage Partial，跨阶段部分结构），骨干网络能够共享和分离部分特征图来提高计算效率和减少梯度消失问题。SPP模块（Spatial Pyramid Pooling，空间金字塔池化）通过多尺度池化，使得YOLOv5在检测不同大小的目标时能够更好地处理尺度变化，从而增强模型对多尺度目标的适应能力。

### 2.5.3 Neck

Neck部分负责将不同尺度的特征进行融合，以增强模型对多尺度目标的识别能力。FPN（Feature Pyramid Network，特征金字塔网络）通过在不同尺度上提取特征并进行融合，提高了模型对大尺度目标的检测精度。而PAN（Path Aggregation Network，路径聚合网络）提高了网络对于低层特征的利用，使得模型在检测小目标时具有更高的鲁棒性。

### 2.5.4 Head

Head部分负责将提取的特征转化为最终的目标检测结果。通过多个损失函数计算目标的分类、位置和置信度，使得模型能够准确地预测目标的类别和位置。同时结合非极大值抑制（NMS，Non-Maximum Suppression）用来去除重复检测框，确保最终的检测结果稳定且准确。

## 2.6 本章小结

本章围绕校园安防监控系统的核心理论与技术展开综述，详细介绍了系统采用的后端技术框架、数据库管理及前端框架。在人脸识别与活体检测技术方面，深入分析了 Dlib 库的工作原理及 EAR 值的计算方法，阐述了通过眼动特征进行身份验证的优势与实现细节。此外，针对多路摄像头视频流的目标检测与追踪任务，全面解读了 YOLOv5 模型的架构及其在系统中的应用场景。本章内容为系统设计与实现奠定了坚实的理论与技术基础，接下来将在下一章详细阐述系统的功能实现与模块设计。

1. 校园安全监测系统中的需求分析【撰写人：周英】

本章主要对本项目的需求进行分析，从用户角度梳理基本需求，并按照功能模块对产品进行划分，针对每一个模块，通过用例图和用例描述的方式对该模块具体功能进行阐述。同时从产品使用和用户需求角度，对系统非功能性需求进行梳理。

* 1. 需求分析综述

校园安全监测系统的诞生是为了解决校园安全管理中的需求，通过实时监控、风险预警和精准识别等技术手段，提升校园安全保障能力。要设计出一款符合需求、易于使用的产品，首先必须明确系统的使用者是谁，了解他们的具体需求，并据此构建出合适的系统功能。

该系统的主要使用者分为普通用户和管理员两类，以下是针对不同角色的需求分析：

（1）普通用户：普通用户的需求相对简单，主要集中在通过人脸匹配技术进行身份验证，以及接收来自系统的安全告警信息。普通用户要求人脸匹配操作简单易用，能够快速完成身份识别，并且在系统发现异常情况时，能够及时接收到告警信息，以便采取必要的应对措施。

（2）管理员：管理员是系统的主要操作用户，其需求涉及系统的几乎所有功能。首先，管理员需要进行用户的登录和注册操作，并能够执行人脸录入和匹配操作。系统应提供人脸访问记录查询功能，以便管理员查看用户的访问历史。为进一步提升安全性，管理员需要管理访客的黑白名单，通过添加或删除名单中的用户来防止潜在的安全威胁。此外，管理员还需要能够划定危险区域，进行实时目标追踪，及时发现并应对异常情况。对于行人重识别技术的应用，管理员可以识别出校园内重要人物的行为模式，以便于安全管理。同时，管理员需启用实时行为检测功能，及时分析并警告任何可疑行为。为了方便事后分析，管理员还需要查看视频回放功能，了解事件发生的经过，辅助调查。系统的告警信息查询和数据大屏展示功能也可以帮助管理员进行全局监控和分析，为决策提供依据。

* 1. 功能性需求分析

校园安防监测系统是一个综合性的监控平台，旨在保障校园安全，实时监控并预警各种异常事件。该系统利用视频监控技术，实现身份验证、异常事件识别和行人重识别功能。通过系统的核心功能，管理员可以高效地管理校园安全，及时处理潜在风险。主要功能包括身份验证、异常事件识别、行人重识别等。根据功能划分，该系统可分为三个主要功能模块，分别为身份验证、异常事件识别、行人重识别。下面将详细介绍这三个功能模块的具体功能性需求。

身份验证模块：该模块主要用于通过人脸识别技术对进入校园的用户进行身份验证，确保校内人员的安全性。功能包括对用户进行身份验证，判断是否存在虚假信息欺诈行为，并在验证成功后展示用户的相关信息。

异常事件识别模块：该模块旨在通过实时监控和分析校园内的异常事件，提供即时的预警。包括黑名单用户监控、持刀行为监测、火灾监测以及危险区域停留监测。系统通过摄像头获取的视频流实时分析，及时发出预警信息。

行人重识别模块：该模块利用多个摄像头捕捉校园内的动态视频流，确保同一用户在不同摄像头下的识别结果一致，实现精准的行人重识别。通过重识别技术，系统能够追踪每一位用户的轨迹，提高监控的精准性与安全性。

* + 1. 身份验证模块

身份验证模块主要实现对校园进入人员的身份验证和监控。该部分与校园门禁系统配合使用，利用人脸识别技术对进入校园的人员进行实时身份验证。系统需从摄像头获取图像或视频流，通过图像处理和识别算法对人员进行身份比对，确保进入人员的身份合法性。同时，系统需要对使用虚假信息（如照片或视频）进行欺诈的行为进行检测，并将该行为记录在案。验证成功后，系统会显示用户的姓名、学工号、权限等信息，供管理员查看。若识别失败或存在欺诈行为，系统将触发警报，及时通知管理员处理。管理员还可在系统中查看历史验证记录，分析验证过程中可能出现的问题。该功能模块与校园门禁系统、安防监控系统紧密集成，通过实时监控和身份验证保证校园安全。该功能用例图如图3-1所示。

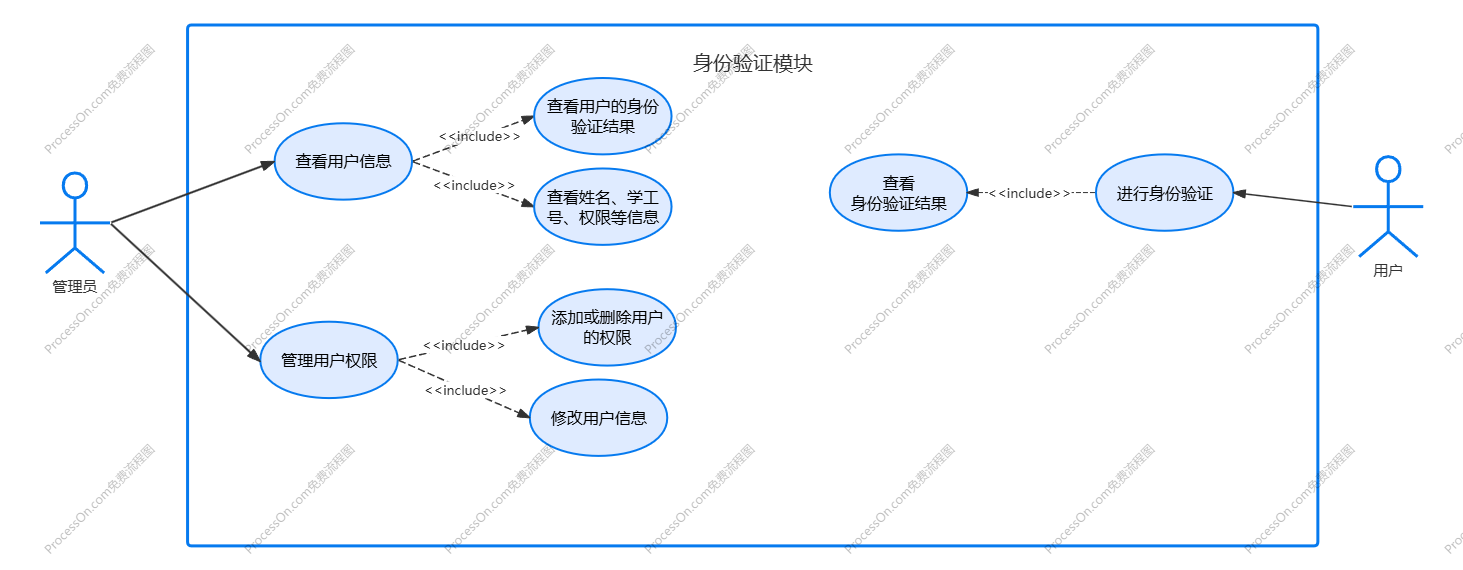


图3-1身份验证用例图

下面对该部分中的两个主要用例查看用户信息、管理用户权限进行详细描述。

表3-1 查看用户信息用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC1 | **用例名称** | 查看用户信息 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 管理员在身份验证后，通过系统查看验证用户的详细信息，包括姓名、学工号、权限等，用于核实身份并进一步处理或授权。 | | |
| **前置条件** | 1. 用户已经通过身份验证。 2. 系统已成功识别用户身份并生成验证结果。 3. 管理员已经登录系统，并具备查看用户信息的权限。 | | |
| **基本流程** | 1. 管理员登录系统并进入身份验证管理界面。 2. 管理员选择“查看用户信息”选项。 3. 系统展示用户信息查询界面，管理员输入或选择需要查看的用户身份信息。 4. 系统从数据库中检索相关用户信息并展示给管理员，内容包括用户姓名、学工号等。 5. 管理员核对信息，如确认无误则进行后续操作，如授权或进行其他管理操作。 6. 管理员查看完毕后，选择退出查看界面，返回到主界面或其他操作界面。 | | |
| **扩展流程** | 3a.如果管理员查询的用户信息不存在或无效，系统会提示“未找到该用户信息”，管理员可以选择重新输入用户信息或取消操作。 | | |
| **异常流程** | 无。 | | |
| **后置条件** | 无。 | | |
| **补充说明** | 无。 | | |

表3-2 管理用户权限用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC2 | **用例名称** | 管理用户权限 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 管理员能够通过系统添加、删除或修改用户的权限，并能够更新用户的相关信息。此功能确保管理员能够有效地控制用户的访问权限，并维护校园安防系统的正常运作。 | | |
| **前置条件** | 1. 管理员已登录到系统，且具备管理用户权限的权限。 2. 系统中已存在需要管理权限的用户。 | | |
| **基本流程** | 1. 管理员登录系统并进入“用户管理”模块。 2. 系统显示当前用户的列表。 3. 管理员选择一个用户，点击“修改权限”或“删除权限”按钮，进入用户权限管理界面。 4. 在权限管理界面，管理员可以：  * 添加权限：选择用户可访问的功能模块并为其分配权限。 * 删除权限：取消用户已有的某些权限。 * 修改权限：更新用户的权限设置，改变用户能够访问的模块或功能。  1. 管理员也可以修改用户的基本信息，如姓名、学工号、权限组等。 2. 管理员点击“保存”按钮，系统保存对用户权限及信息的更改。 3. 系统返回用户列表页面，显示更新后的用户权限信息。 | | |
| **扩展流程** | 4a.如果管理员查询的用户信息不存在或无效，系统会提示“未找到该用户信息”，管理员可以选择重新输入用户信息或取消操作。  4b.如果管理员选择删除用户权限，系统会弹出确认框询问管理员是否确定删除，以防误操作。 | | |
| **异常流程** | 1A.管理员没有权限进行权限管理操作。系统提示“权限不足”并阻止管理员继续操作。  6A.管理员输入了无效的用户信息（例如，学工号重复）。系统提示“用户信息无效，请重新输入”，并要求管理员更正信息。  6B.在保存用户权限或信息时，系统发生异常或无法连接到数据库。系统提示“保存失败，请稍后重试”，并记录错误日志。 | | |
| **后置条件** | 1. 用户的权限和信息已更新或删除，系统已保存最新的权限数据。 2. 系统能够根据更新后的权限信息对用户的访问进行有效控制。 | | |
| **补充说明** | 该用例涉及管理员对用户权限的管理，因此需要确保只有授权的管理员能够执行此操作。在权限修改过程中，系统需要确保权限数据的完整性和准确性，防止误操作或错误权限配置导致的安全隐患。 | | |

* + 1. 异常事件识别模块

异常事件识别模块主要实现对校园内潜在安全风险的实时监控与预警。该部分需要与校园内的各类监控系统（如摄像头、传感器等）协同工作，实时分析视频流与传感器数据，以识别可能发生的异常事件。系统将根据预设的规则识别并分类四种主要异常事件：黑名单用户进入校园、人员持刀、火灾发生以及用户在危险区域内停留超过3秒。对于每一类异常事件，系统将自动进行实时监控，并根据监控数据触发相应的预警。

系统将维护一份黑名单，实时比对进入校园的人员是否在黑名单中，一旦识别到黑名单用户进入，系统将自动发出预警；同时，系统会对持刀事件、火灾等危险行为进行自动检测，一旦发生异常，系统会立即报警；另外，系统还允许管理员预先设定危险区域，若用户在该区域停留超过设定时间（如3秒），系统也会触发警报。

为了确保异常事件的高效处理，系统将实时将事件监测数据传送给管理员，管理员可以在系统中查看所有异常事件的详细记录，并进行后续的处理与调查。该模块的目标是通过自动化的监控和报警功能，大大提高校园安全管理的效率与响应速度。该功能用例图如图3-2所示。



图3-2异常事件识别用例图

下面对于该部分中的三个主要用例配置危险区域、管理黑名单用户、查看异常事件记录进行详细描述。

表3-3 配置危险区域用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC3 | **用例名称** | 配置危险区域 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 管理员在系统中配置校园内的危险区域，并设置相应的停留时间阈值（如3秒）。当用户进入这些区域并停留超过阈值时，系统会触发预警。 | | |
| **前置条件** | 1. 管理员已成功登录到系统并具有相应的权限。 2. 系统已集成监控摄像头或传感器，能够实时监测校园内的活动情况。 3. 系统已提供危险区域配置功能，管理员能够访问该功能页面。 | | |
| **基本流程** | 1. 管理员登录系统，进入“异常事件识别”模块。 2. 管理员选择“配置危险区域”功能。 3. 系统显示校园地图或危险区域配置界面。 4. 管理员选择需要标记为危险区域的区域，并在地图上进行绘制或选择。 5. 管理员设置该区域的停留时间阈值（例如3秒）。 6. 管理员确认配置并保存危险区域设置。 7. 系统更新危险区域配置，并在系统中生效。系统开始实时监控该区域，检测用户的停留时间。 8. 若用户在危险区域内停留超过设定阈值，系统触发预警。 | | |
| **扩展流程** | 4A. 如果管理员在配置区域时需要调整区域的大小或位置，可以通过拖动地图中的区域边界进行修改。  5A.系统支持批量配置多个危险区域，管理员可以通过选择不同的区域并设置不同的停留阈值来完成配置。 | | |
| **异常流程** | 6A. 管理员未配置危险区域的停留时间阈值，点击保存后系统提示“停留时间阈值不能为空”。 | | |
| **后置条件** | 1. 校园内的危险区域已成功配置并保存到系统中。 2. 系统开始对配置的危险区域进行实时监控。 3. 系统在识别到用户在危险区域内停留超过设定时间阈值时，能够触发预警并通知管理员。 | | |
| **补充说明** | 无。 | | |

表3-4 管理黑名单用户用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC4 | **用例名称** | 管理黑名单用户 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 此用例描述了管理员如何管理黑名单用户，包括添加、删除黑名单用户以及查看黑名单用户列表。管理员通过此功能对校园安全进行动态管理，确保系统实时有效地监控黑名单用户的进入并触发相应的预警。 | | |

续表3-4管理黑名单用户用例说明

|  |  |
| --- | --- |
| **前置条件** | 1. 管理员已成功登录系统，且具备相关权限。 2. 系统已存在一个基础的黑名单用户列表。 3. 系统能够实时识别进出校园的用户身份，并与黑名单进行比对。 |
| **基本流程** | 1. 管理员通过系统界面进入“黑名单管理”模块。 2. 系统展示当前黑名单用户的列表。 3. 管理员选择“添加黑名单用户”，输入用户身份信息（如姓名、学工号、人脸识别ID等）。 4. 系统验证输入的信息格式是否正确，若信息有效，系统将该用户添加至黑名单，并提示操作成功。 5. 管理员选择“删除黑名单用户”，系统展示用户列表，管理员选择需要删除的用户。 6. 系统从黑名单中删除该用户，并提示操作成功。 7. 管理员可以查看黑名单用户的详细信息（如添加时间、身份信息等）。 8. 管理员完成黑名单管理操作，退出该模块。 |
| **扩展流程** | 3A. 如果管理员输入的用户信息已经存在于系统中，系统会提示用户信息重复，并要求管理员重新输入。  5A.如果删除的用户信息不存在或已被删除，系统会提示“用户不存在”或“该用户已被删除”。 |
| **异常流程** | 无。 |
| **后置条件** | 无。 |
| **补充说明** | 无。 |

表3-5 查看异常事件记录用例说明

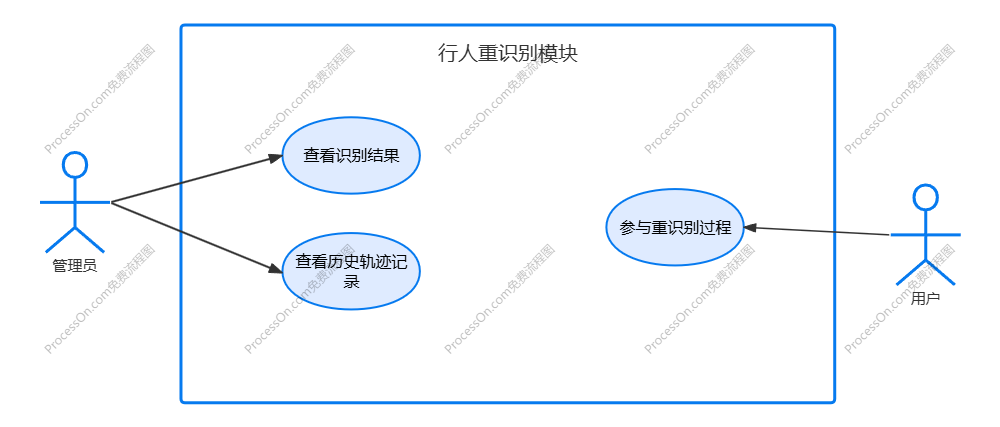
|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC5 | **用例名称** | 查看异常事件记录 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 该用例描述了管理员在异常事件识别模块中查看过去的异常事件记录的过程。管理员能够通过系统查看所有已记录的异常事件，包括黑名单用户进入、持刀事件、火灾报警以及危险区域停留等事件的详细信息，以便进一步分析和处理潜在的安全风险。 | | |
| **前置条件** | 1. 管理员已登录系统，并拥有查看异常事件记录的权限。 2. 系统已经接收到并记录了异常事件（如黑名单用户进入、持刀事件、火灾等）。 3. 系统已存储并更新了所有的异常事件数据。 | | |

续表3-5 查看异常事件记录用例说明

|  |  |
| --- | --- |
| **基本流程** | 1. 管理员登录系统，并进入异常事件识别模块。 2. 管理员选择“查看异常事件记录”选项。 3. 系统展示异常事件记录列表，按事件类型（如黑名单用户、持刀、火灾、危险区域停留）进行分类，并提供时间范围过滤选项。 4. 管理员选择具体的事件类型或设置时间范围过滤条件（如指定日期或特定事件类型）。 5. 系统根据选择条件展示符合条件的异常事件记录，记录中包含事件的详细信息（如时间、位置、事件类型、相关人员等）。 6. 管理员查看详细记录，分析每个事件的发生原因，判断是否需要进一步的处理或行动。 7. 管理员可以导出异常事件记录或打印事件日志（可选）。 |
| **扩展流程** | 4A. 管理员可以点击某个事件记录，系统展示该事件的详细信息，如事件发生时间、监控视频截图、涉及人员的身份信息、处理结果等。 |
| **异常流程** | 无。 |
| **后置条件** | 无。 |
| **补充说明** | 无。 |

* + 1. 行人重识别模块

行人重识别模块主要实现对校园内动态视频流中同一用户的身份识别和轨迹追踪。该模块与校园内的多个摄像头系统配合工作，利用计算机视觉和深度学习算法，确保在不同摄像头视角下对同一用户进行准确识别。系统能够通过实时视频分析，跟踪用户的行踪，并通过比对多次识别结果确保同一用户在不同时间、不同地点的身份一致性。此外，系统能够自动记录用户的活动轨迹。管理员可通过系统查看和分析用户的历史轨迹数据。该功能用例图如图3-3所示。

图3-3行人重识别用例图

下面对于该部分中的两个主要用例查看识别结果、查看历史轨迹记录进行详细描述。

表3-6 查看识别结果用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC6 | **用例名称** | 查看识别结果 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 管理员可以通过系统查看行人重识别模块对不同视频流中同一用户身份的识别结果，了解用户的活动轨迹，并分析是否存在异常行为。该功能帮助管理员快速获取校园内的实时监控数据和历史数据，为潜在的安全隐患提供早期预警。 | | |
| **前置条件** | 1. 行人重识别模块已经与校园内的摄像头系统正常连接，并且摄像头已经开始采集视频流。 2. 系统已经完成对用户的身份识别和轨迹追踪工作。 3. 管理员已登录系统并获得相应权限。 | | |
| **基本流程** | 1. 管理员登录系统，进入行人重识别模块。 2. 系统展示所有用户的活动轨迹选项，管理员选择需要查看的用户。 3. 管理员选择查看特定用户的历史轨迹数据，包括不同摄像头的识别结果。 4. 系统展示该用户在不同时间、地点的行踪数据，包括各摄像头捕获的图像、时间标记和位置坐标。 5. 管理员查看识别结果，可以通过数据分析判断用户行为是否异常。 6. 系统允许管理员对识别结果进行导出或保存，便于后续分析和报告。 | | |
| **扩展流程** | 无。 | | |
| **异常流程** | 6A.若系统无法生成识别结果（如系统崩溃或数据丢失），管理员将收到“数据未加载”或“识别结果不可用”的错误提示。 | | |

续表3-6 查看识别结果用例说明

|  |  |
| --- | --- |
| **后置条件** | 1. 系统展示的行人重识别结果被管理员成功查看。 2. 管理员根据识别结果分析行为并可能采取后续安防措施。 3. 如果有导出或保存需求，相关数据被成功导出并保存。 |
| **补充说明** | 系统在展示用户轨迹时会提供时间轴、摄像头位置、用户的实时状态和动态行为等信息，确保管理员能够全面了解该用户的活动。 |

表3-7 查看历史轨迹记录用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC7 | **用例名称** | 查看历史轨迹记录 |
| **活动者** | 管理员 | **优先级** | 高 |
| **描述** | 管理员通过系统查看和分析历史轨迹数据，及时发现异常行为和潜在的安全风险。该功能使管理员能够回溯用户在校园内的行动轨迹，确保校园安全监控的有效性和及时性。 | | |
| **前置条件** | 1. 系统正常运行，且已经接入校园内的摄像头及动态视频流数据。 2. 行人重识别模块已成功识别并记录校园内用户的轨迹数据。 3. 管理员已登录系统并获得查看历史轨迹记录的权限。 | | |
| **基本流程** | 1. 管理员登录校园安防监测系统。 2. 管理员选择“行人重识别”模块中的“查看历史轨迹记录”功能。 3. 系统展示历史轨迹查询界面，管理员可以输入查询条件（如时间范围、用户ID、特定区域等）。 4. 管理员提交查询请求，系统根据条件查询用户的历史轨迹数据。 5. 系统返回符合条件的历史轨迹记录，包括每个时间点的位置信息、摄像头识别数据等。 6. 管理员分析和查看历史轨迹记录，识别是否存在异常行为。 7. 管理员根据分析结果采取相应措施（如进一步调查或发出警报）。 8. 管理员退出查看历史轨迹记录功能，返回系统首页。 | | |
| **扩展流程** | 3A.管理员在查询历史轨迹时没有设置具体查询条件，系统展示所有历史轨迹数据，管理员可以从中筛选和查看。 | | |
| **异常流程** | 3A.管理员输入无效的查询条件（如错误的用户ID、时间格式不正确等），系统提示“查询条件无效”，并要求管理员重新输入正确条件。 | | |
| **后置条件** | 1. 管理员成功查看历史轨迹记录并根据需要采取相应的安全措施。 2. 如果存在异常轨迹，管理员已收到预警并进行相应处置。 3. 系统中记录了管理员的查询操作，供后续审计和分析使用。 | | |
| **补充说明** | 无。 | | |

* 1. 非功能性需求分析

校园安防监测系统除了满足基本的功能性需求外，还需满足以下非功能性需求，以确保系统的稳定性、可扩展性和用户体验。

（1）易用性：系统应具备用户友好的界面，设计简洁、布局合理，操作流程直观。管理员和普通用户在使用过程中能够快速掌握系统操作，减少学习成本。特别是对于监控信息和异常事件预警的展示，需确保信息清晰、反馈及时，帮助管理员高效进行决策和处理。

（2）正确性：由于校园安防监测系统涉及到用户身份验证和异常事件的实时监测与预警，系统的准确性至关重要。特别是在黑名单用户监控、火灾报警、持刀行为监测等方面，系统必须能够精确识别异常事件，避免误报或漏报，以确保对校园安全的有效保障，防止对用户或校园环境产生不必要的干扰。

（3）高性能：为了实现实时监控和预警，系统需要具备高性能的数据处理能力。摄像头视频流的实时处理、行人重识别的高效匹配、异常事件的快速预警都要求系统能够在低延时下进行数据处理，确保监控过程不中断、预警信息能够及时发送，避免因性能瓶颈影响系统响应速度和用户体验。

（4）安全性：作为涉及校园安全的系统，必须保证系统本身的安全性。系统应对所有用户操作进行严格的身份验证和权限控制，确保只有授权人员才能访问敏感数据或修改系统配置。同时，系统应具备数据加密功能，防止数据泄露或被篡改。

这些非功能性需求是保障校园安防监测系统长期稳定运行和提供高质量服务的基础，确保系统能够在复杂的校园环境中持续提供安全保障。

* 1. 本章小结

本章首先对校园安防监测系统的用户及其基本需求进行了分析，明确了系统的主要目标是保障校园内的安全，提供实时的身份验证、异常事件识别和行人重识别功能。随后，本章详细分析了系统的各项功能需求，确保系统能够满足管理员和用户的实际需求。最后，本章对系统的非功能性需求进行了说明，为系统的稳定运行和优质用户体验提供保障。

1. 校园安全监测系统中的概要设计【撰写人：赵雨滔】

本章以系统整体需求为基础，对校园安全监测系统的概要设计进行阐述。

* 1. 系统体系结构

校园安全监测系统的整体体系架构如图4-1所示。系统采用前后端分离的设计模式，前端部分采用Vue框架进行开发，主要负责构建用户界面和提供用户交互功能。

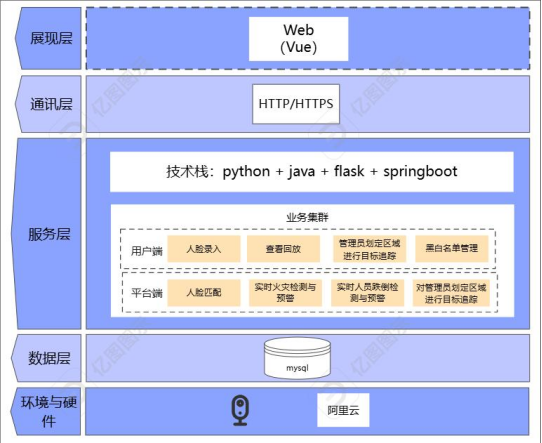


图4-1 系统架构图

后端部分主要负责业务逻辑的处理和数据交互。其中，Flask框架主要用于处理与前端交互的HTTP/HTTPS请求，实现用户信息的验证、数据的传输等功能。而Springboot框架则更多地用于处理业务逻辑。

系统存储的数据来源于三类：用户信息数据，监控数据，异常事件数据。

* 1. 系统功能结构

在第三章需求分析的过程中，本系统有3大模块。3大模块分别是身份验证模块、异常事件识别模块、行人重识别模块。系统总体功能模块结构图如图4-2所示。

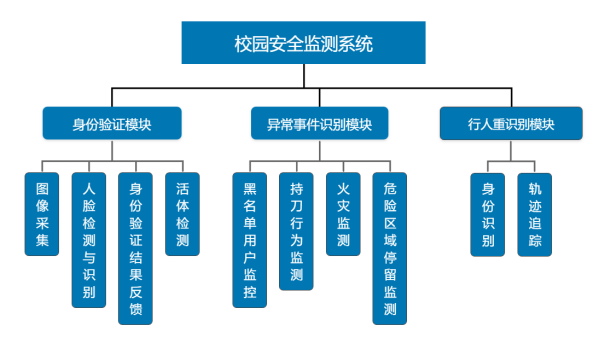


图4-2 校园安全监测系统总体结构图

身份验证模块负责通过人脸识别技术对进入校园的用户进行身份验证。该模块设计包括摄像头图像采集、人脸检测与识别算法实现、身份验证结果反馈功能和活体检测功能。

异常事件识别模块旨在实时监控校园内的潜在安全风险，并触发相应的预警。该模块设计包括黑名单用户监控、持刀行为监测、火灾监测以及危险区域停留监测功能

行人重识别模块实现对校园内动态视频流中同一用户的身份识别和轨迹追踪。该模块包括身份识别和轨迹追踪功能。

* 1. 系统数据库设计

本节主要介绍校园安全监测系统的数据库设计，给出系统ER图，包括9个实体和2个实体间关系，并给出详细设计，包括字段、类型、属性及字段描述。系统ER图如图4-3所示。

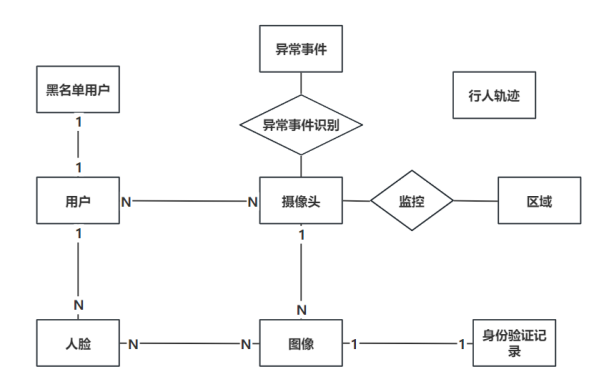


图4-3 系统ER图

表详细设计如表4-1至4-11所示。

表4-1 用户表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：User** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| user\_id | INT | 主键自增 | 用户唯一标识符 |
| name | VARCHAR(100) | 不为空 | 用户姓名 |
| student\_id | VARCHAR(50) | 可空 | 学生ID |
| employee\_id | VARCHAR(50) | 可空 | 教职工ID |
| role | VARCHAR(15) | 不为空 | 角色类型 |
| phone\_number | VARCHAR(15) | 可空 | 联系电话 |
| email | VARCHAR(100) | 可空 | 邮箱地址 |

表4-2 摄像头表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：Camera** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| camera\_id | INT | 主键自增 | 摄像头唯一标识符 |
| location | VARCHAR(255) | 不为空 | 安装位置 |
| status | ENUM('active', 'inactive') | 不为空 | 运行状态 |
| description | TEXT | 可空 | 设备描述 |

表4-3 图像表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：Image** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| image\_id | INT | 主键自增 | 图像唯一标识符 |
| camera\_id | INT | 外键 | 关联摄像头ID |
| capture\_time | DATETIME | 不为空 | 捕获时间 |
| file\_path | VARCHAR(255) | 不为空 | 文件存储路径 |

 表4-4人脸表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：Face** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| face\_id | INT | 主键自增 | 人脸唯一标识符 |
| user\_id | INT | 外键 | 关联用户ID |
| image\_id | INT | 外键 | 关联图像ID |
| feature\_data | BLOB | 不为空 | 人脸特征数据 |

表4-5 身份验证记录表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：AuthenticationLog** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| log\_id | INT | 主键自增 | 日志唯一标识符 |
| image\_id | INT | 外键 | 关联图像ID |
| result | BOOLEAN | 不为空 | 验证结果（true/false） |
| timestamp | DATETIME | 不为空 | 记录时间 |

表4-6 异常事件表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：Event** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| event\_id | INT | 主键自增 | 事件唯一标识符 |
| type | ENUM('blacklist\_entry', 'suspicious\_behavior') | 不为空 | 事件类型 |
| timestamp | DATETIME | 不为空 | 发生时间 |
| details | TEXT | 可空 | 事件详情 |
| blacklisted\_user\_id | INT | 外键， 可空 | 关联黑名单用户ID |

表4-7 行人轨迹表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：PedestrianTrack** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| track\_id | INT | 主键自增 | 轨迹唯一标识符 |
| user\_id | INT | 外键 | 关联用户ID |
| start\_time | DATETIME | 不为空 | 开始时间 |
| end\_time | DATETIME | 可空 | 结束时间 |
| path\_points | TEXT | 不为空 | 路径点列表 |

表4-8 异常事件识别表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：CameraEvent** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| camera\_event\_id | INT | 主键自增 | 中间表唯一标识符 |
| camera\_id | INT | 外键 | 关联摄像头ID |
| event\_id | INT | 外键 | 关联事件ID |
| detection\_time | DATETIME | 不为空 | 检测时间 |

表4-9 区域表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：Region** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| region\_id | INT | 主键自增 | 区域唯一标识符 |
| name | VARCHAR(100) | 不为空 | 区域名 |
| description | TEXT | 可空 | 区域描述 |

表4-10 区域-摄像头表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：CameraRegion** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| camera\_region\_id | INT | 主键自增 | 中间表唯一标识符 |
| camera\_id | INT | 外键 | 关联摄像头ID |
| region\_id | INT | 外键 | 关联区域ID |

表4-11 黑名单用户表设计

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：BlacklistedUser** | | | |
| **字段名** | **类型** | **属性** | **描述** |
| blacklisted\_user\_id | INT | 主键自增 | 黑名单用户唯一标识符 |
| user\_id | INT | 外键 | 关联用户ID |
| reason | VARCHAR(255) | 不为空 | 加入黑名单原因 |

* 1. 本章小结

本章在第三章需求分析的基础上，介绍了系统的整体架构设计，并对系统功能模块进行划分，最后介绍系统的数据库设计，提供了系统整体ER图以及各个表的设计细节。

1. 校园安全监测系统中的详细设计与实现【撰写人：陈曦红】

本章以校园安全监测系统的概要设计为基础，进行详细的设计与实现。

## 5.1基于人脸识别的身份验证模块

### 5.1.1身份验证模块流程设计

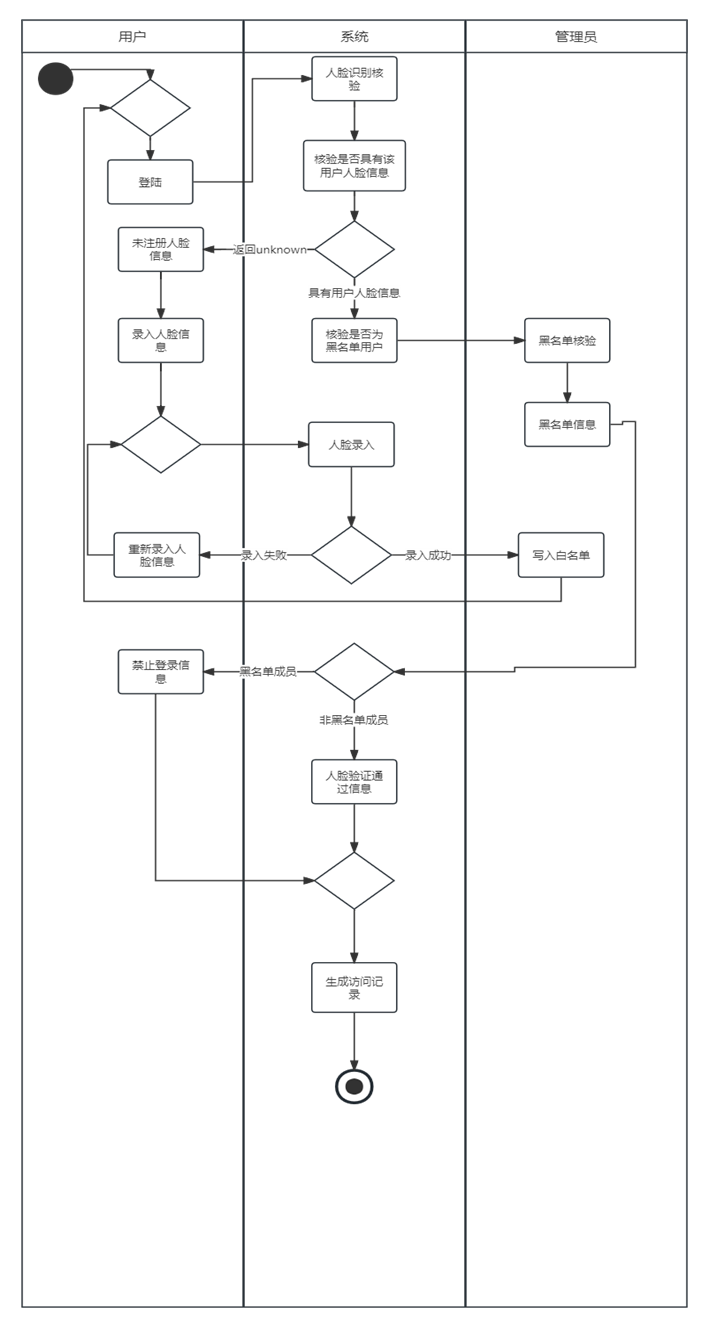


图5-1 人脸识别模块流程图/活动图

用户登录后先进行人脸匹配，系统检测是否储存了该用户的人脸信息，如果是未注册的人脸信息则先联系管理员进行人脸录入成功后写入白名单，再进行重新匹配；如果检测到储存了该用户的人脸信息，则检验该用户是否为黑名单用户，如若是黑名单用户，则产生禁止访问的信息，非黑名单成员，则验证通过。

### 5.1.2身份验证**模块类设计**

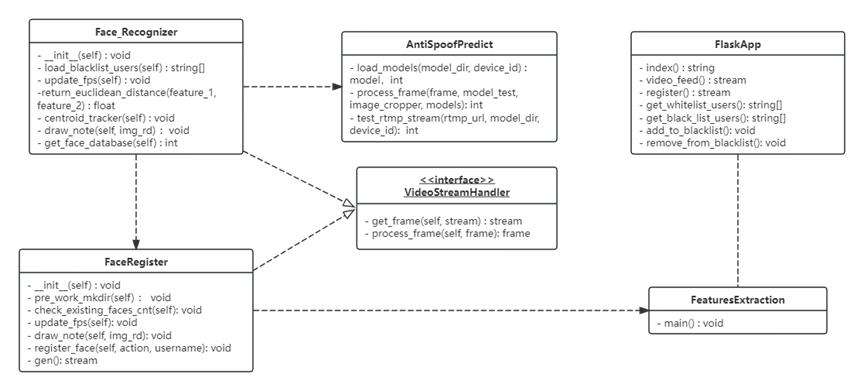


图5-2 人脸识别模块类图

当服务器端监听到客户端发来的请求时，使用FlaskApp控制类进行处理。Flaskapp类依赖于人脸注册业务类FaceRegister类，人脸识别类Face\_Recognition类，同时，FaceRegister类依赖于特征提取处理类FeaturesExtraction类，他们之间通过方法调用进行交互。VideoStreamHandler为接口类，用于处理从服务器获取的视频流，FaceRegister类和Face\_Recognition类实现该接口，用于逐帧处理图片，并将处理之后的结果以multipart/x-mixed-replace格式输出。它们之间通过flask进行连接。

### 5.1.3身份验证模块时序设计

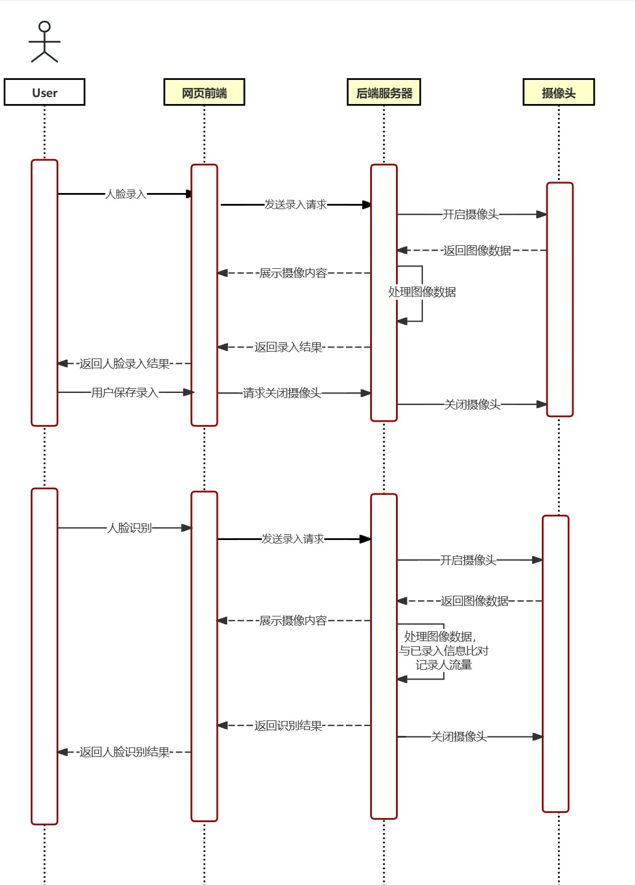


图5-3 人脸识别模块时序图

### 5.1.4**人脸识别模块实现**

图5-4为人脸录入界面。用户登陆之后，需要进行人脸识别。而在人脸识别之前，首先需要在人脸注册界面进行人脸信息录入。在用户名处输入该人脸信息对应的姓名，然后点击保存，即可完成该人脸信息的录入。



图5-4 人脸录入界面

当用户已经录入人脸信息之后，通过人脸识别界面，可以识别出该用户的身份信息。如果该用户尚未在系统中注册人脸，那么识别出的结果为unknown，如果该用户在系统中已经注册过人脸，并且未被拉入黑名单，那么识别出其注册人脸信息时的名字。如图5-5为人脸识别界面所示。如果该用户已经在系统中注册过人脸，并且被拉入了黑名单，那么其人脸识别的结果为Fobbiden，即人脸识别界面登录失败，禁止其入内。如图5-10黑名单用户人脸识别所示。



图5-5无活体检测接入的人脸识别界面

为了预防用户使用视频、照片等方式进行人脸匹配，我们对该恶意检测行为进行识别。当用户为真人，即通过正常的方式进行人脸识别时，识别框为蓝色，并且能够识别出该用户的名字。如图5-6所示。但是，当用户采用照片，视频等方式进行恶意人脸识别时，将使用红框进行识别，并且识别名称为Fake。如图5-7所示。

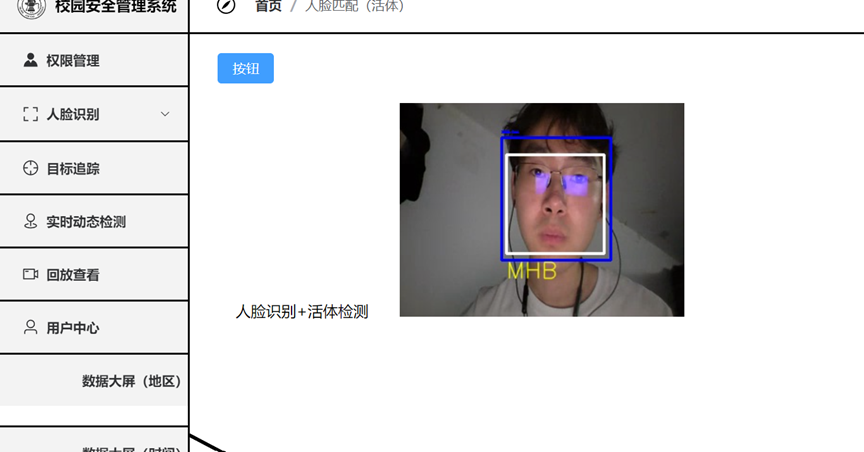


图5-6有活体检测接入的人脸识别界面



图5-7有活体检测接入的身份冒用界面

当用户在人脸录入界面录入人脸信息之后，首先默认都是白名单用户。在白名单用户栏显示所有的白名单用户。如图5-8所示。管理员有权限将白名单用户设置为黑名单，同时也有权限将黑名单用户恢复白名单。如图5-9所示，将白名单用户置为黑名单，并且在黑名单栏显示所有的黑名单用户。



图5-8白名单



图5-9黑名单



图5-10黑名单用户人脸识别

5.2异常事件识别模块

### 5.2.1异常事件识别模块流程设计

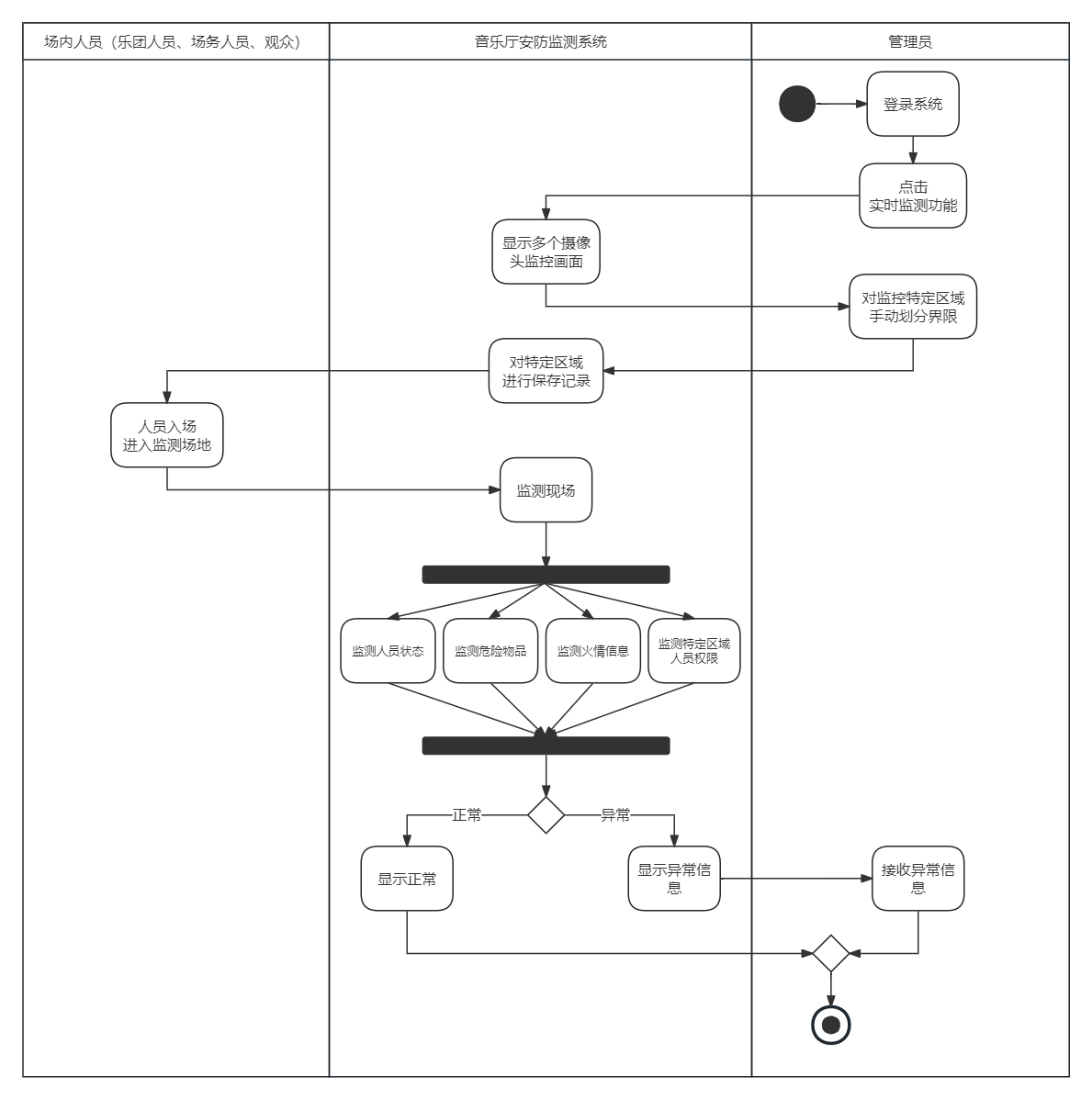


图5-11异常事件识别模块活动图/流程图

用户登录后选择实时监测功能，系统将会检测摄像头下的监控画面，用户可以对摄像头内的区域进行手动划分界限，然后系统会对特定区域进行记录的保存，如果有人员进入划定的区域内，并逗留了3s以上，系统则会认定其为异常事件并报警，而且系统还会监测人员是否持刀、是否有火灾隐情，以及是否有黑名单用户进入校园这三类情况的发生，如果有系统会接收到异常事件发生的信息，并在1s内做出响应。

### 5.2.2异常事件识别模块类设计descript

图5-12异常事件识别模块类图

ObjectDetection、PersonMonitoring、RegionMonitoring、FireMonitoring这四个类分别进行人员持刀、黑名单用户进入、危险区域逗留以及火情的监测，Warning类负责报警部分的功能。MonitoringEntity初始化所有的监控类。IBaseController定义了最基本的接口，BaseController负责上述五种实体类的具体方法的调用。

### 5.2.3异常事件识别模块时序设计

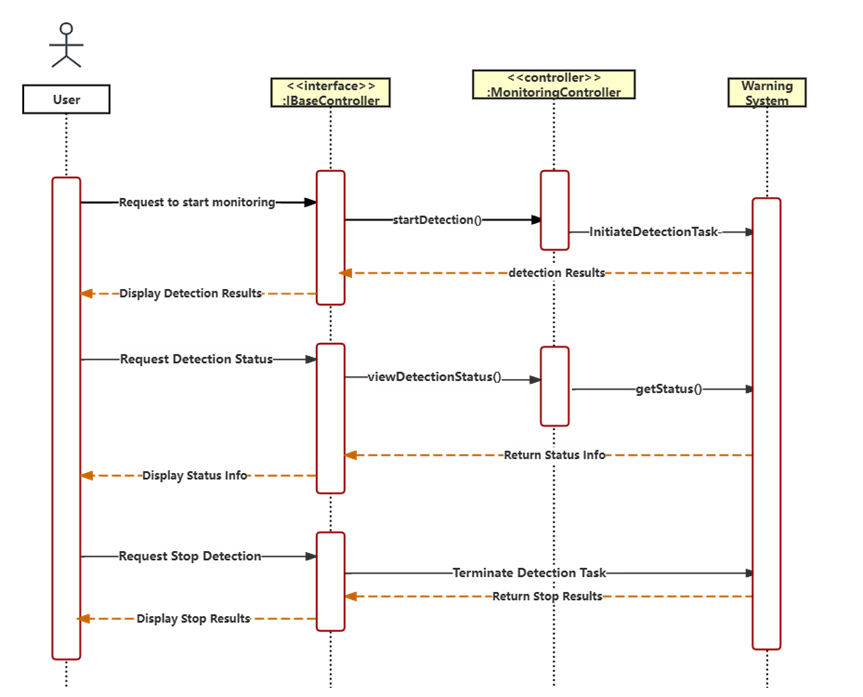


图5-13异常事件识别模块时序图

### 5.2.4异常事件识别模块实现

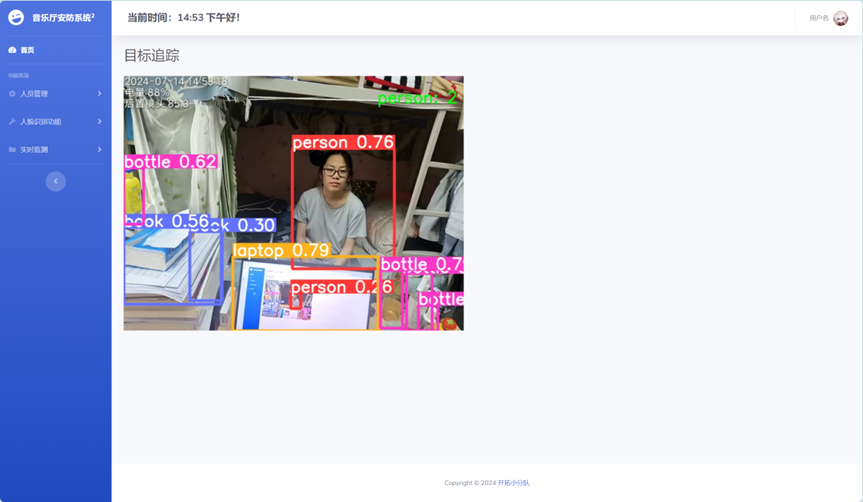


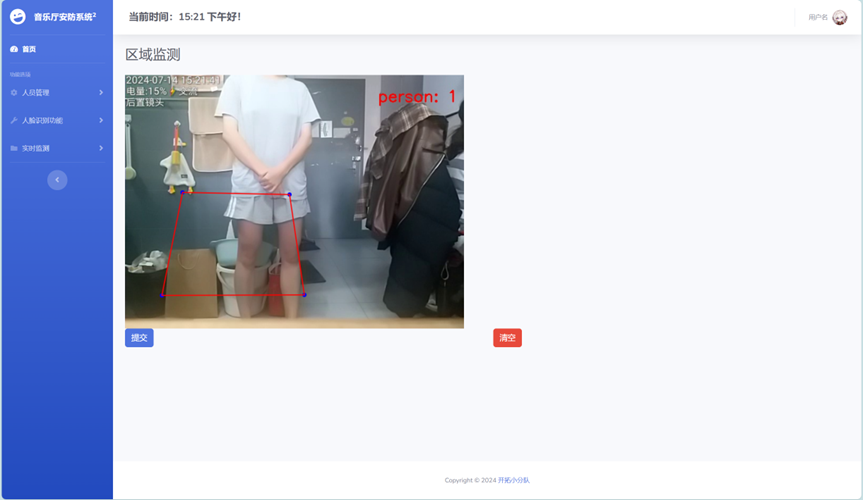
图5-14异常事件识别模块实现图1

图5-15 异常事件识别模块实现图2

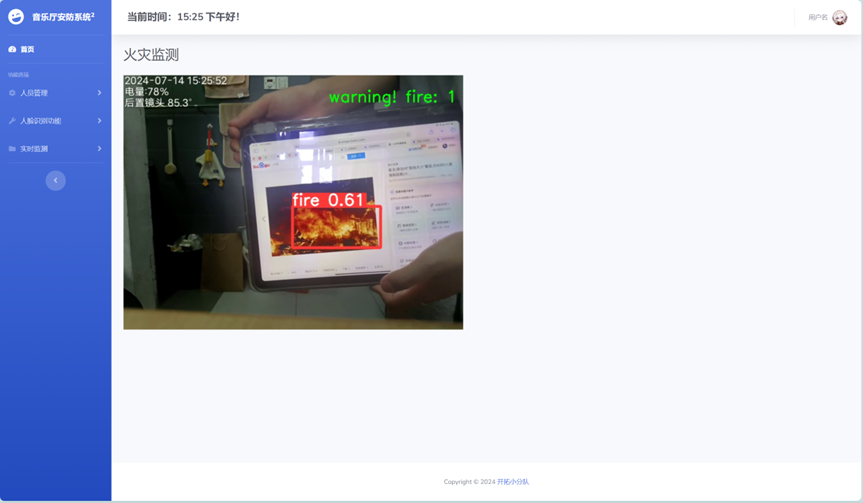


图5-16 异常事件识别模块实现图3

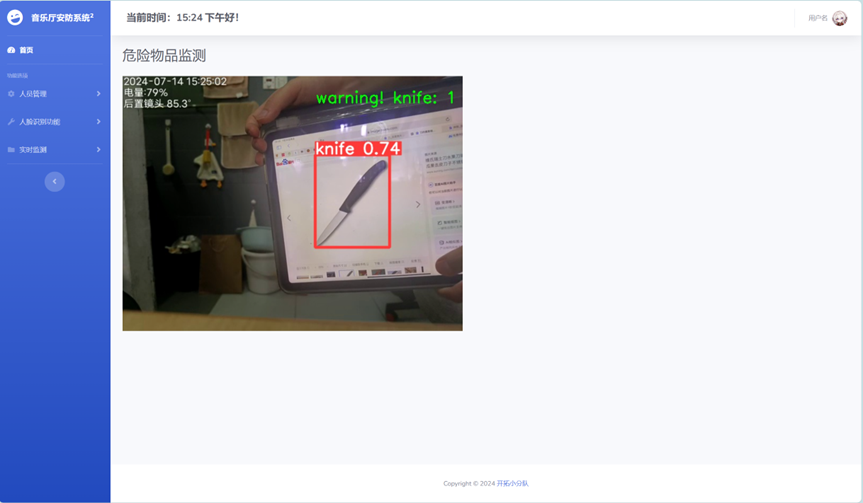


图5-17异常事件识别模块实现图4

## 5.3行人重识别模块

### 5.3.1行人重识别模块流程设计

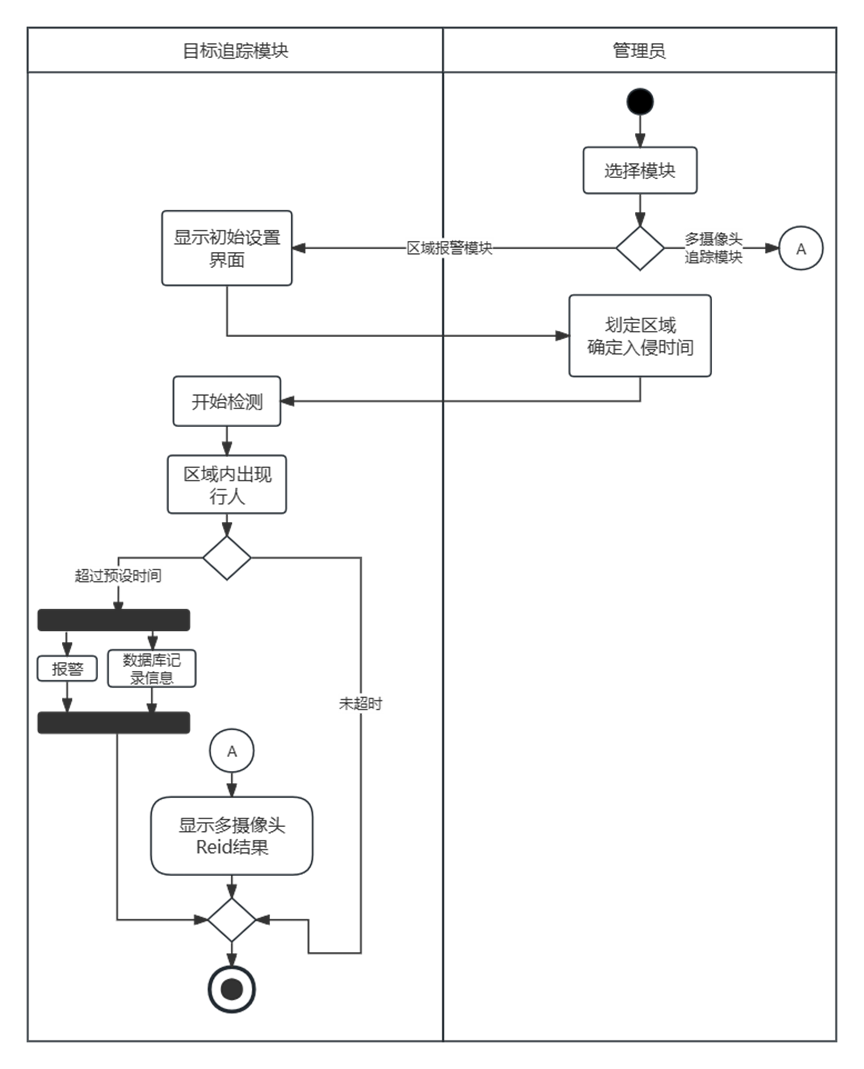


图5-18行人重识别模块活动图/流程图

管理员登录后，选择多摄像头识别模块，系统会不断对摄像头内的用户进行监测，在前端页面上显示多摄像头reid的结果。

### 5.3.2行人重识别模块类设计

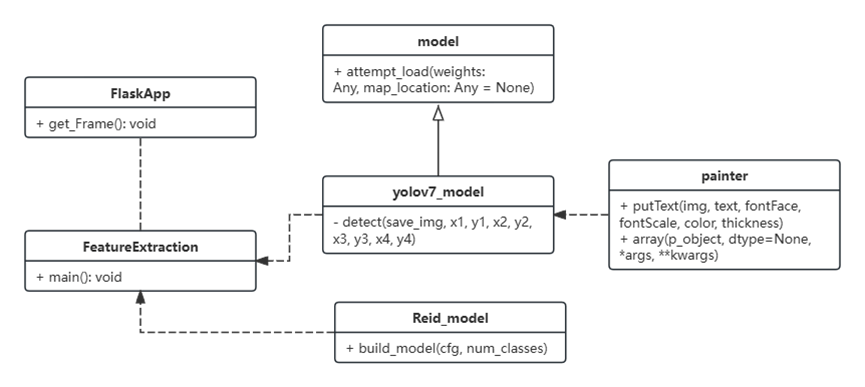


图5-19行人重识别模块类图

原理上，本项目使用的yolov7\_reid模型和上方用于区域报警的模型差不太多。主要的特点在于引入了Reid\_model模块进行行人重识别。在FeatureExtraction类的main()函数里调用Reid，实现对多个视频流的相同人的reid。

### 5.3.3行人重识别模块时序设计

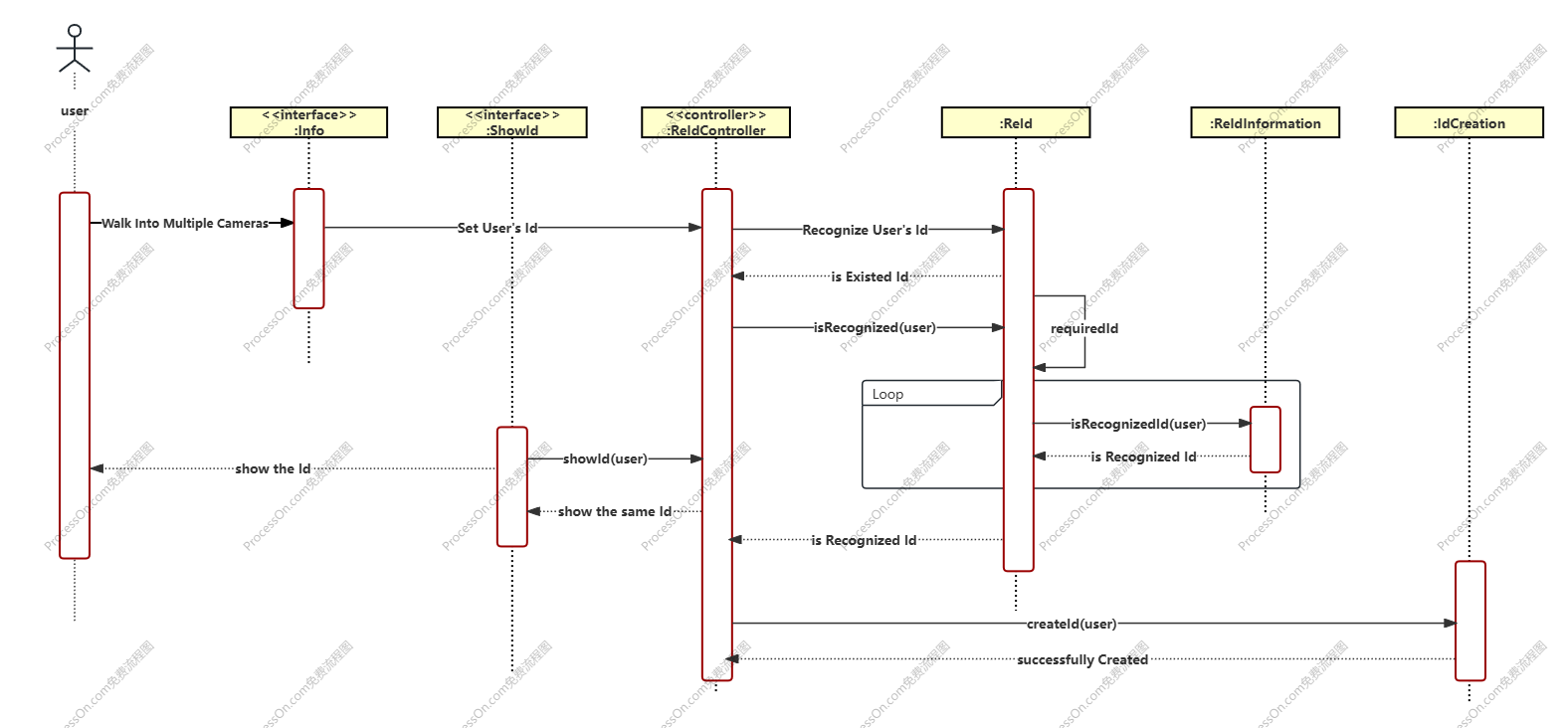


图5-20行人重识别模块时序图

### 5.3.4行人重识别模块实现

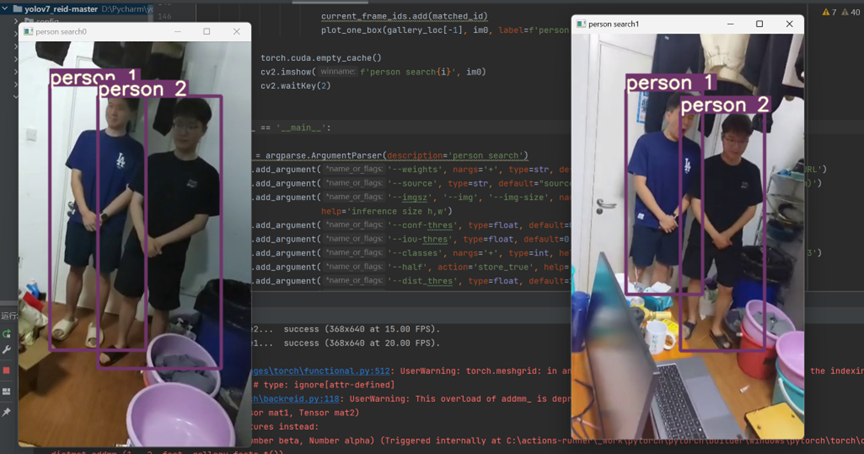


图5-21行人重识别模块实现图

## 5.4本章小结

本章在第四章系统概要设计的基础上，介绍了系统的每一个功能模块的具体实现方法，包括身份验证模块、异常事件识别模块以及行人重识别模块，同时附上对应的实现效果图，展示系统的实现效果。

1. 系统测试【撰写人：赵雨滔】

本章节中，以第三章需求分析为基础，对系统进行功能性和非功能性测试。

* 1. 测试目标及准备

功能测试主要验证系统的所有模块是否按照需求正确实现。非功能测试主要评估系统的并发处理能力和用户界面友好性，确保系统能够稳定高效地工作。

测试环境所需的硬件与软件版本信息如表6-1所示。

表6-1 系统测试环境

|  |  |
| --- | --- |
| 服务 | 测试环境 |
| 硬件详情 | CPU：i7 8550U，内存：8G |
| 数据库服务 | MySQL：8.0,Redis：5.0.6 |
| 软件详情 | Flink：1.9.2，Kafka：2.4.0 |
| 操作系统 | Windows 11 |

* 1. 功能测试

功能测试按照第三章中用例描述划分的功能模块，使用黑盒测试的方法。

w

表6-2 用户身份验证测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC1 | **用例名称** | 用户身份验证 |
| **测试日期** | 2024-11-18 | **优先级** | 高 |
| **测试目的** | 验证系统能否正确识别用户身份，并在验证成功后展示用户信息 | | |
| **测试准备** | 用户已注册并录入人脸信息，系统正常运行 | | |
| **测试流程** | 1.用户站在身份验证区域前开始识别人脸。  2.面部图像信息比对。  3.如果比对成功，系统显示用户的身份信息。 | | |
| **预期结果** | 1.系统成功识别用户身份，并显示用户信息。  2.用户信息包括姓名、学工号、权限等。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-3 黑名单用户监控测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC2 | **用例名称** | 黑名单用户监控 |
| **测试日期** | 2024-11-18 | **优先级** | 高 |
| **测试目的** | 验证系统能否正确识别黑名单用户并触发警报 | | |
| **测试准备** | 系统已设定黑名单用户，摄像头正常运行 | | |
| **测试流程** | 1.黑名单用户进入校园区域。  2.系统通过摄像头捕获用户面部图像。  3.系统将用户面部图像与黑名单进行比对。  4.如果比对成功，系统触发警报并通知管理员。 | | |
| **预期结果** | 1.系统成功识别黑名单用户。  2.系统触发警报并通知管理员。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-4 查看用户历史轨迹测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC3 | **用例名称** | 查看用户历史轨迹 |
| **测试日期** | 2024-11-18 | **优先级** | 高 |
| **测试目的** | 验证系统能否正确展示用户的活动轨迹 | | |
| **测试准备** | 系统已记录用户的活动轨迹，管理员已登录系统 | | |
| **测试流程** | 1.管理员登录系统，进入行人重识别模块。  2.管理员选择“查看历史轨迹记录”功能。  3.管理员输入查询条件。  4.系统根据条件查询用户的历史轨迹数据。  5.系统返回符合条件的历史轨迹记录。 | | |
| **预期结果** | 1.系统成功展示用户的历史轨迹记录。  2.记录包括每个时间点的位置信息、摄像头识别数据等。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

* 1. 非功能测试

本次非功能测试主要分为两部分，第一部分是站在系统管理员角度，测试系统并发处理能力；第二部分是站在用户角度，测试用户界面友好性。具体测试结果如下：

表6-5 系统并发处理能力测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC4 | **用例名称** | 系统并发处理能力 |
| **测试日期** | 2024-11-18 | **优先级** | 高 |
| **测试目的** | 验证系统在高并发情况下能否稳定运行，处理大量请求 | | |
| **测试准备** | 使用LoadRunner模拟100个并发用户，系统正常运行 | | |
| **测试流程** | 1.使用LoadRunner模拟100个用户同时进行身份验证。  2.记录系统响应时间和资源占用情况。  3.观察系统是否有异常或崩溃现象。 | | |
| **预期结果** | 1.系统响应时间不超过2秒。  2.系统资源占用率不超过80%。  3.系统无异常或崩溃现象。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-6 用户界面友好性测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC5 | **用例名称** | 用户界面友好性 |
| **测试日期** | 2024-11-18 | **优先级** | 中 |
| **测试目的** | 验证系统的用户界面是否友好，操作流程是否直观 | | |
| **测试准备** | 系统正常运行，邀请5名普通用户和5名管理员进行测试 | | |
| **测试流程** | 1.普通用户和管理员分别登录系统  2.尝试进行常用操作。  3.记录用户在操作过程中遇到的问题和建议。 | | |
| **预期结果** | 1.用户能够快速上手，操作流畅。  2.用户界面简洁、布局合理，操作流程直观。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

* 1. 本章小结

本章主要对系统进行了功能测试和非功能测试，测试均全部通过。

# 7 总结与展望【撰写人：马海博】

本文根据校园安防实际需求，设计并实现了一套功能完善的校园安防监控系统。本章总结了项目开发过程中的主要工作，同时结合系统的实现和运行情况，分析了目前系统的不足，提出了改进方案，并对系统未来发展方向进行了展望。

## 7.1 全文总结

本文的主要工作内容总结如下：

（1）本文介绍了校园安防监控系统开发的背景，分析了异常行为（如黑名单人员入侵、持刀事件、火灾、危险区域滞留等），明确了现有安防手段的不足，提炼核心需求，将系统分为人脸识别身份验证、异常事件识别、行人重识别三大模块，并针对这些模块进行了需求细化与用例描述。

（2）学习了系统设计与开发过程中涉及的技术与理论，包括 Dlib 人脸识别、EAR 活体检测、YOLOv5 目标检测、Spring Boot 与Flask后端框架、MySQL 数据库及 Vue.js 前端开发框架。根据需求分析，对系统进行功能模块拆分，同时完成了数据库设计，明确了各个实体及其关系。

（3）在概要设计的基础上，对系统的各功能模块进行详细设计，包括身份验证模块的人脸识别与活体检测流程设计，异常事件识别模块的多任务目标检测与报警机制，行人重识别模块的跨摄像头轨迹跟踪算法等。结合系统框架，对每个模块进行编码实现，并完成了系统功能测试和性能测试。

本文设计的校园安防监控系统，集成了多项技术，能够在实际应用中提供实时、高效的校园安防监控能力，不仅提升了校园管理的智能化水平，也为保障学生和教职工的安全提供了有力支持，具有重要的实际意义和社会价值。

## 7.2 系统展望

在系统设计与实现过程中，虽然已经具有高效的校园安防监控能力，但仍存在可以改进的空间。

首先，在人脸识别身份验证的活体检测方面，现阶段的活体检测仅基于 EAR 值判断眨眼行为。未来可结合头部动作、嘴部动作、光线变化等多维度特征，进一步提高活体检测的鲁棒性，防止更多形式的伪造攻击。

其次，在异常事件识别方面，系统目前主要基于 YOLOv5 模型进行检测，尽管效果良好，但在复杂场景（如光线变化、遮挡）中，仍需提升检测和识别精度。未来可以引入更先进的目标检测与跟踪算法（如 YOLOv8 和 DeepSORT），并结合时序分析和深度学习模型实现对异常行为的更准确识别。

最后，系统的用户体验与扩展性仍有改进空间。未来可引入数据可视化功能，提供更加直观的安全分析界面，并加入更多交互功能（如实时风险通知、智能分析报告生成），提升系统对管理人员的友好度。

综上所述，校园安防监控系统的构建是一个持续迭代的过程。随着技术的进步和校园安防需求的变化，该系统将不断升级完善，为智慧校园的建设提供更强大的技术支持，并最终实现更全面的安全防护目标。

参考文献【谢雨桐】

[1] 王智.智慧校园系统中人脸识别技术的应用与研究[D].中北大学,2019.

[2] 许德刚,王露,李凡.深度学习的典型目标检测算法研究综述[J].计算机工程与应用,2021,57(08):10-25.

[3] 董新颐.基于人工智能的人脸图像识别校园安防系统设计[J].软件,2022,43(01):151-153.

[4] 韩佳南,范永超.基于人工智能的校园监控及通信系统的设计与研究[J].中国新通信,2022,24(01):22-23.

[5] 刘朝斌,孙雪,刘剑,等.基于物联网的高校校园智能安防建设探索[J].深圳大学学报(理工版),2020,37(S1):128-133.

[6] 张峰.应用SpringBoot改变Web应用开发模式[J].科技创新与应用,2017,7(23):193-194

[7] 沈宏伟.基于Flask的企业内网安全系统的设计与实现[D].北京交通大学,2018：1-5.

[8] 刘兆丰.Dlib在人脸识别技术中的运用[J].电子制作,2020,28(21):39-48.

[9] GANG PAN, LIN SUN, ZHAOHUI WU, et al. Eyeblink-based Anti-Spoofing in Face Recognitio from a Generic Webcamera[C]//IEEE 11th International Conference on Computer Vision (ICCV 2007), vol.4. 2007:1879-1886.

[10] 邱天衡,王玲,王鹏,等.基于改进YOLOv5的目标检测算法研究[J].计算机工程与应用,2022,58(13):63-73.

[11] 封彦,鲁磊.基于Yolov5和Dlib的在线违规行为监控识别算法[J].光电子技术, 2023, 43(3):276-282.

[12] 林佳文.基于深度学习的行人重识别系统的设计与实现[D].大连:大连理工大学,2021:34-43.