## 项目介绍

### 简要背景

校园安全一直是各大高校关注的重要问题之一。为了提高校园监控的效率和准确性，开发了一款基于PyQt和OpenCV的校园监控系统。该系统通过调用API实现车辆检测和人员检测，旨在帮助学校更好地管理校园安全，及时发现和处理潜在的安全隐患。

### 功能介绍

**车辆检测**：

* 1. 利用摄像头实时捕捉校园内的视频流。
  2. 通过API进行车辆检测，识别并统计不同类型的车辆（如小汽车、卡车、巴士、摩托车和三轮车）。
  3. 在视频帧中标注检测到的车辆类型和数量。

**人员检测**：

* 1. 利用摄像头实时捕捉校园内的视频流。
  2. 通过API进行行人检测，识别并统计行人数量。
  3. 在视频帧中标注检测到的行人数量。

**主界面**：

* 1. 友好的用户界面，便于操作和监控。
  2. 可切换到车辆检测和人员检测界面。
  3. 显示系统Logo和背景图片，提升用户体验。

### 使用场景

**校园安保**：

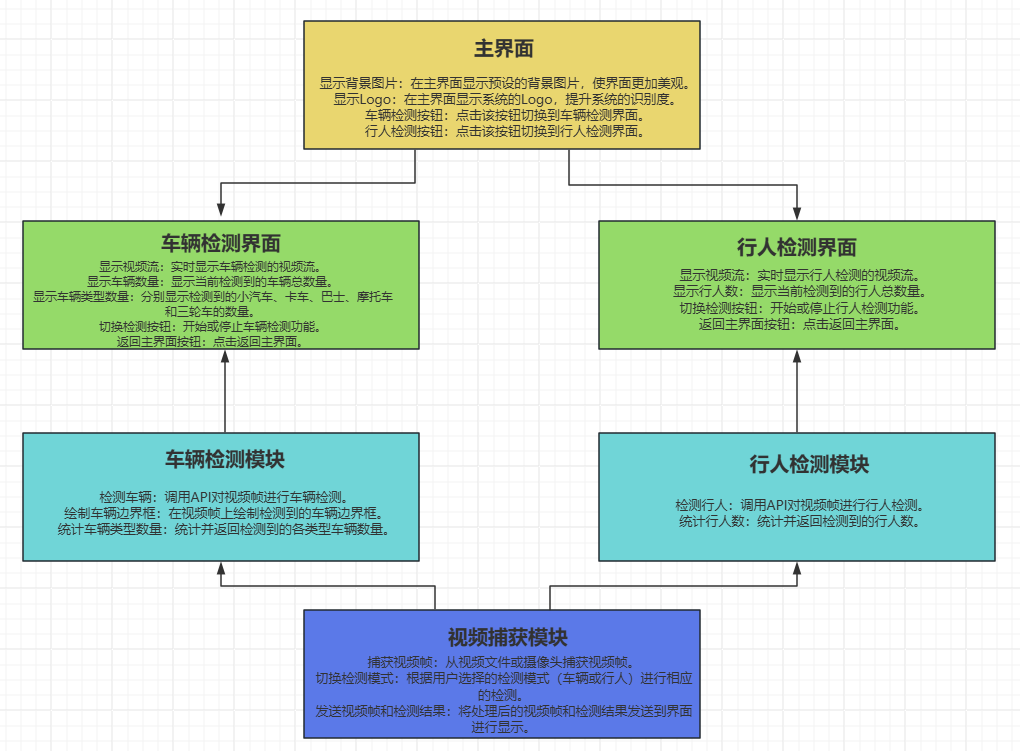
* 1. 安保人员可以通过该系统实时监控校园内的车辆和行人流量，及时发现异常情况。
  2. 统计分析不同时间段的车辆和人员数量，优化安保资源的配置。

**交通管理**：

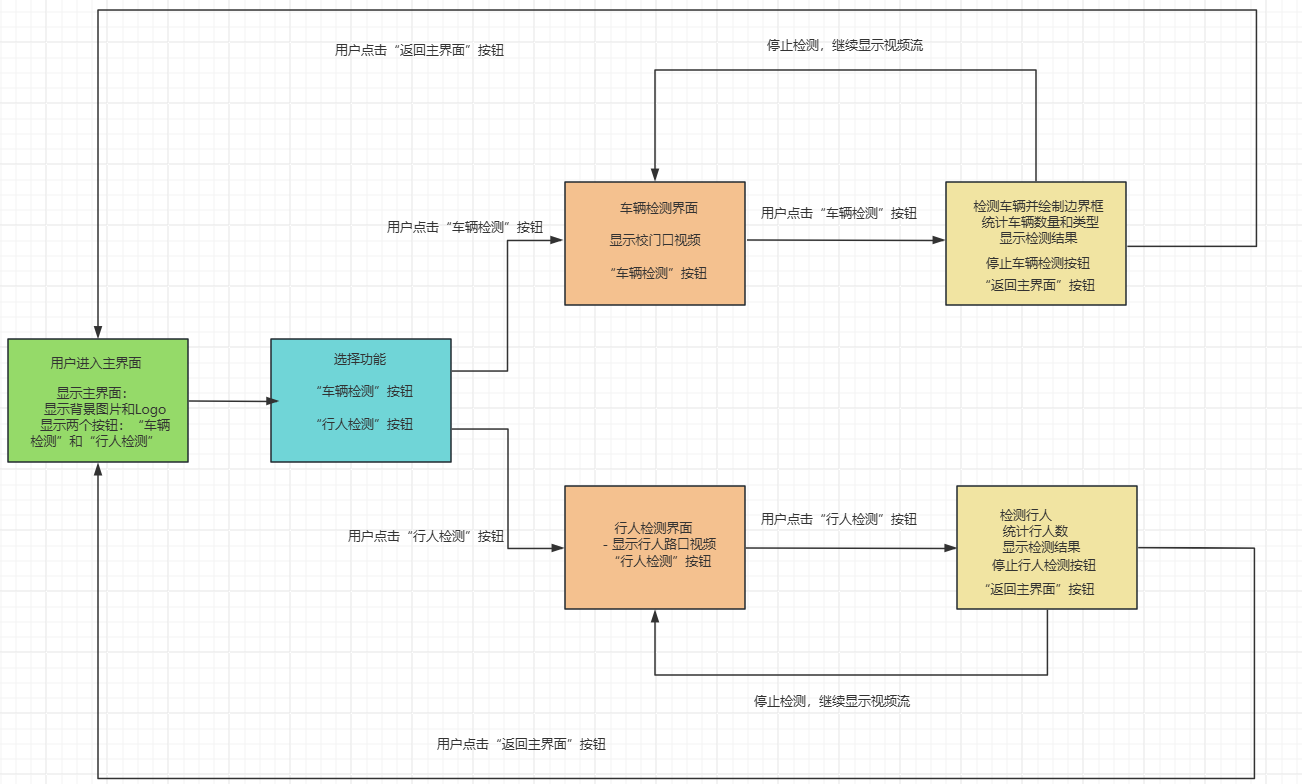
* 1. 学校管理部门可以通过该系统监控校园内的交通情况，及时疏导交通，避免拥堵。
  2. 统计分析校园内的车辆类型和数量，为制定校园交通规划提供数据支持。

通过该系统的部署和使用，能够显著提高校园安全管理的效率和准确性，为创建平安校园提供有力的技术支持。

功能结构图：



流程图



### 技术结构

#### 1. 应用程序结构

* **main.py**：主程序入口，启动应用程序并显示主界面。
* **data/**：存放资源文件（背景图片、Logo、视频文件）。
* **detection/**：检测模块，包括车辆检测和行人检测。
* **Preprocessing/**：预处理模块，负责视频捕获。
* **ui/**：用户界面模块，包括主界面、车辆检测界面和行人检测界面。

#### 2. 模块结构和功能

**主程序入口 (main.py)**

* + 负责启动应用程序，创建并显示主界面。
  + 调用 QApplication 类来管理应用程序的主事件循环。
  + 创建 MainWindow 实例并显示。

**数据文件夹 (data/)**

* + 存储系统所需的资源文件，包括背景图片 (bg.jpg)、Logo图片 (logo.png) 和视频文件 (vd1.mp4 和 vd4.mp4)。

**检测模块 (detection/)**

* + **vehicle\_detection.py**：
    - 调用百度API进行车辆检测。
    - 使用 OpenCV 处理视频帧，绘制检测到的车辆边界框，统计车辆数量和类型。
  + **person\_detection.py**：
    - 调用百度API进行行人检测。
    - 返回检测到的行人数。

**预处理模块 (Preprocessing/)**

* + **video\_capture.py**：
    - 负责视频捕获，从视频文件中读取视频帧。
    - 根据用户选择的检测模式（车辆或行人）调用相应的检测模块。
    - 使用 PyQt 的 QThread 类实现多线程，发送处理后的视频帧和检测结果到界面进行显示。

**用户界面模块 (ui/)**

* + **main\_window.py**：
    - 主界面，显示背景图片和Logo。
    - 包含两个按钮：“车辆检测”和“行人检测”。
    - 点击按钮切换到相应的检测界面。
  + **vehicle\_window.py**：
    - 车辆检测界面，显示视频流和车辆检测结果（各类型车辆数量）。
    - 包含开始/停止检测和返回主界面的按钮。
  + **person\_window.py**：
    - 行人检测界面，显示视频流和行人检测结果（行人数）。
    - 包含开始/停止检测和返回主界面的按钮。

#### 3. 技术栈

* **PyQt**：用于创建桌面应用程序的用户界面。
* **OpenCV**：用于视频处理和检测结果的绘制。
* **百度API**：用于车辆检测和行人检测。
* **Python**：作为主要的编程语言，整合各个模块。

#### 4. 工作流程

**启动应用程序**

* + 通过 main.py 启动，显示主界面。

**用户选择检测模式**

* + 点击“车辆检测”按钮或“行人检测”按钮，切换到相应的检测界面。

**视频捕获**

* + 调用 video\_capture.py 捕获视频帧。

**调用检测模块**

* + 根据选择的检测模式，调用 vehicle\_detection.py 或 person\_detection.py 进行检测。

**显示检测结果**

* + 在检测界面显示视频流和检测结果（车辆数量及类型或行人数）。

**切换界面**

* + 用户可以点击“返回主界面”按钮返回主界面，或在检测界面间切换。

通过上述技术结构，可以清晰了解校园监控系统的各个模块及其功能，实现车辆检测和行人检测的完整流程，从视频捕获、检测到结果显示，提供了全面的技术解决方案。

界面设计：低保真原型图：

（直接用的项目结果图）

主界面



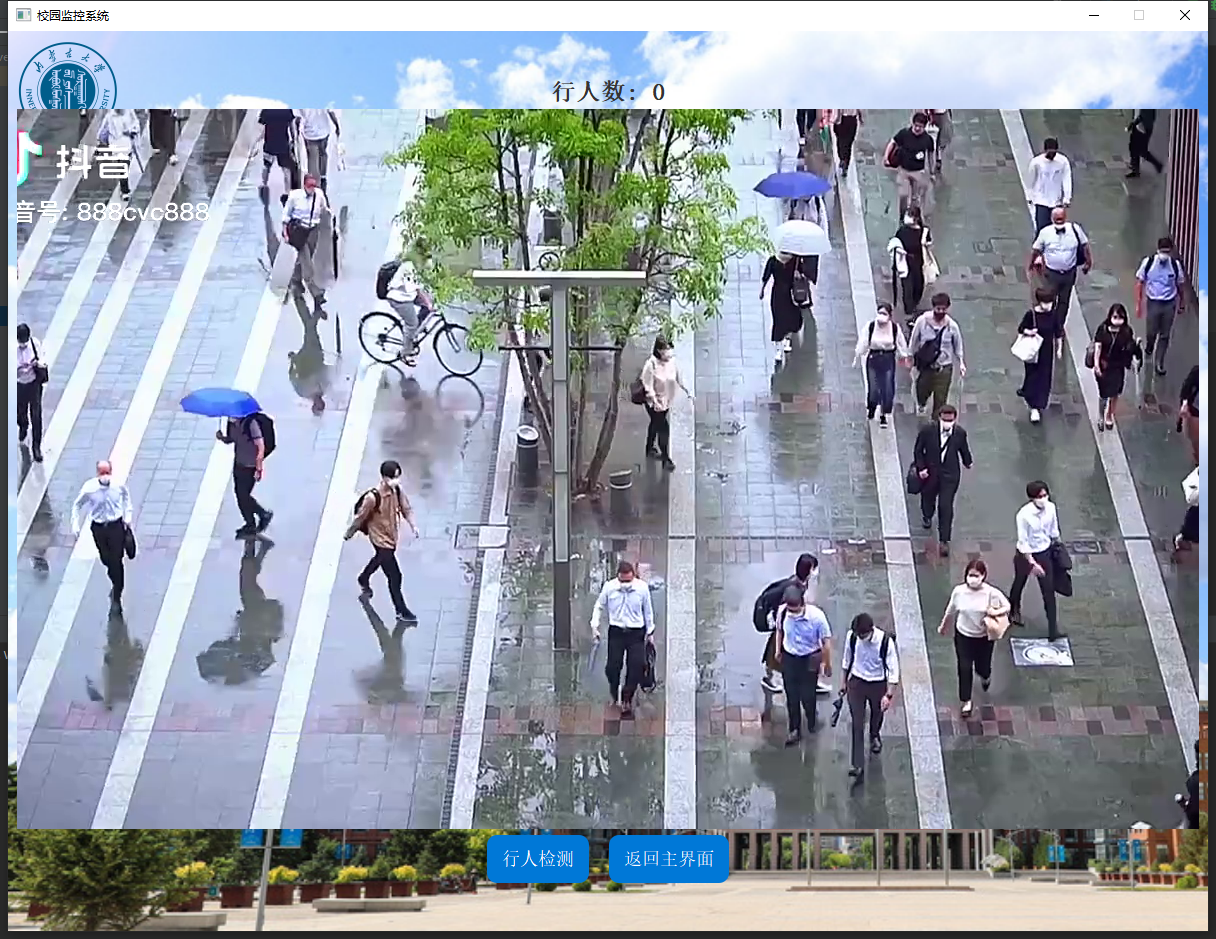
车辆检测界面：



进行车辆检测：



行人检测界面：



进行行人检测：



### 总结

本项目开发了一款基于PyQt和OpenCV的校园监控系统，旨在提升校园安全管理的效率和准确性。通过调用百度API，该系统实现了实时的车辆检测和行人检测功能，能够有效地监控校园内的交通和人员流动情况。

#### 项目特点

1. **多功能检测**：系统包括车辆检测和行人检测两大功能模块，能够同时满足不同的监控需求。
2. **实时监控**：利用视频捕获模块，实现了对视频流的实时处理和检测，提供了及时的监控数据。
3. **友好界面**：通过PyQt创建的用户界面直观、易用，使用户能够轻松切换和操作各功能模块。
4. **高效检测**：调用百度API进行车辆和行人的检测，结合OpenCV的图像处理能力，确保检测的准确性和高效性。

#### 技术结构

项目采用模块化设计，各个功能模块独立但相互协作，包括主程序入口、数据存储、检测模块、预处理模块和用户界面模块。技术栈主要包括PyQt用于界面开发，OpenCV用于视频处理，百度API用于检测服务，整体采用Python编写。

#### 功能流程

1. 用户启动应用程序，显示主界面。
2. 用户选择检测模式（车辆检测或行人检测），系统切换到相应的检测界面。
3. 视频捕获模块捕获视频帧并发送到检测模块。
4. 检测模块调用百度API进行检测，并将结果显示在界面上。
5. 用户可以切换检测模式或返回主界面进行其他操作。

#### 使用场景

* **校园安保**：提高安保人员的监控效率，及时发现和处理异常情况。
* **交通管理**：帮助学校管理部门监控和疏导校园内的交通。
* **应急响应**：在紧急事件发生时，快速定位和跟踪车辆和人员的位置，提高应急响应效率。

#### 项目展望

未来，系统可以进一步扩展和优化，如增加更多检测类型、提高检测精度和速度、结合机器学习算法进行智能分析等，以满足更多场景的需求，进一步提升校园安全管理的能力。

本项目通过技术与实际应用的结合，为校园安全管理提供了有效的解决方案，不仅提高了监控的效率和准确性，也为未来的技术扩展和创新提供了基础。通过持续的优化和改进，相信该系统将在校园安全管理中发挥越来越重要的作用。