

# 需求抽象

## 描述：

如题，将需求抽象为适合计算机系学生体质的语言。

## 背景

随着高校校园规模的扩大以及交通工具的多样化，校园交通安全管理成为校园管理的重要组成部分。本项目旨在基于 Flutter 框架开发一个跨平台的智能交通管理系统，利用视频监控数据与行为识别算法，针对人车混行的状态下进行异常行为检测与识别，从而保障校园内的交通安全。通过对视频数据的采集与分析，构建校园关键路口的交通流动规律模型，并结合优化决策方法，进行智能化交通管理。

系统将围绕校园交通环境中的关键路口，通过对平面交叉路口混合交通行为的观察，识别异常交通行为，构建多信息融合的优化决策机制，最终通过 Flutter 实现面向移动设备的智能交通管理应用。

## 核心需求

按 MoSCoW 优先级进行划分：

### Must Have

- 实时视频监控数据的采集与处理。
- 异常交通行为的自动识别与检测。
- 基于行为识别的优化交通管理决策。
- 校园关键路口的交通流动规律建模与分析。
- 基于 Flutter 框架的跨平台应用开发，支持 Android 与 iOS 设备。
- 通过 Dart 语言构建智能交通管理系统的业务逻辑。
- 对人车混合交通的静态与动态特征（度、介数、特征向量等）的分析与展示。

- 提供交通异常事件的可视化界面，支持实时监控与警报功能。

### Should Have

- 支持历史交通数据的回放与分析。
- 应用接口（API）或数据导出功能，以便与其他校园系统集成。
- 支持多个摄像头的数据处理与综合分析。

### Could Have

- 基于移动设备的用户报告功能，允许用户上报交通异常。
- 交通流量预测模型，帮助预判未来的交通情况。

### Won't Have

- 复杂的 3D 地图展示功能。
- 高级的 AR/VR 增强现实功能。

## 系统架构概述

本系统采用典型的前后端分离架构，主要分为以下几大模块：

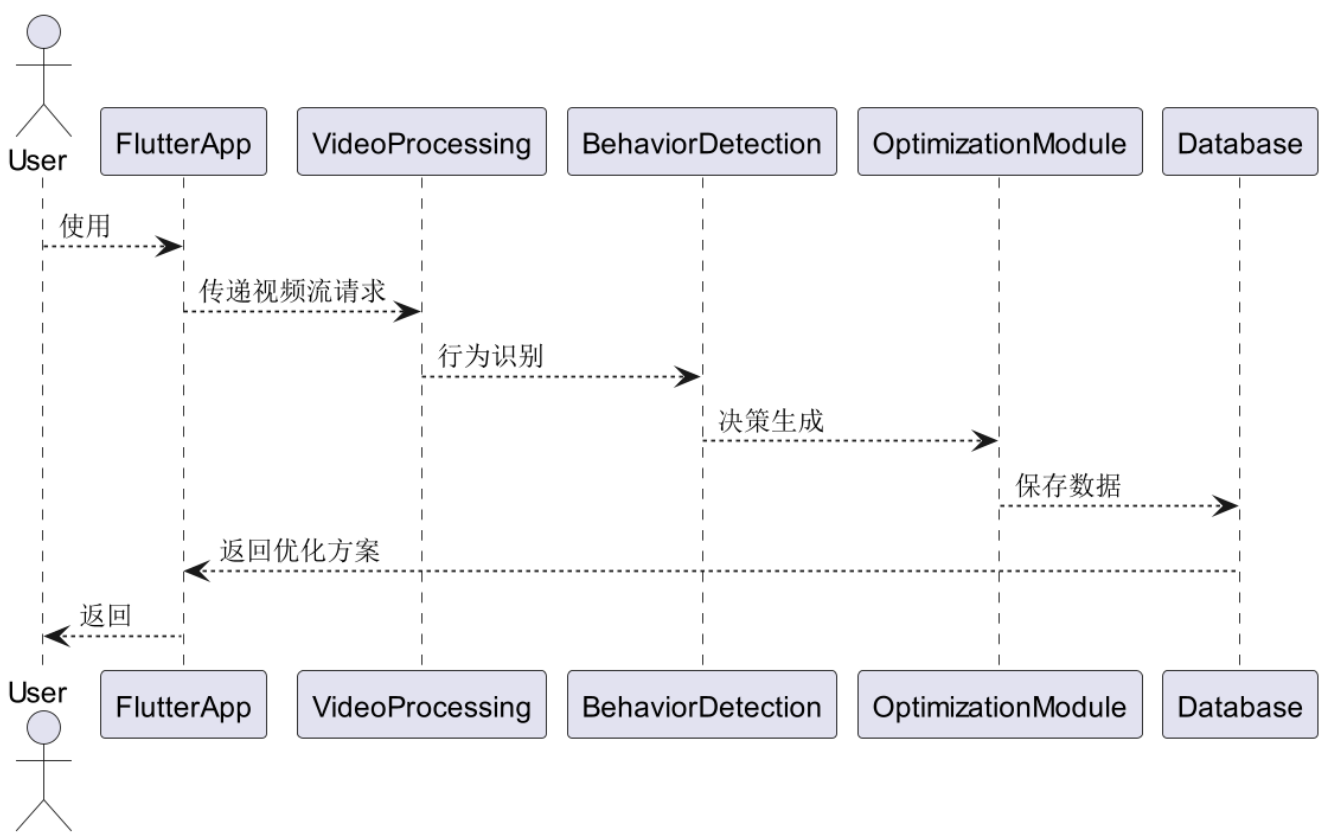
1. 前端 Flutter 应用：跨平台支持 iOS 和 Android，提供用户界面，展示实时监控视频、交通流动状态和异常行为预警。
2. 视频处理与行为识别模块：处理来自摄像头的视频流数据，执行基于深度学习的异常行为检测算法，识别出诸如逆行、闯红灯等异常行为。
3. 优化决策模块：基于识别到的行为，使用多信息融合算法生成优化决策，提供交通管理建议。
4. 数据存储与管理模块：基于关系型数据库存储实时与历史的交通数据、事件日志和用户反馈。

## 技术栈

- 前端：Flutter（跨平台），使用 Dart 语言。
- 后端：使用 Python（如 TensorFlow 或 OpenCV）进行视频处理和算法实现。

- 数据库：MySQL（用于存储交通流数据、异常行为日志等），可根据具体需求扩展至分布式数据库。

架构图



数据库设计

系统将使用 MySQL 数据库，以下是几个关键数据表的结构设计：

```
Table video_records [note: "视频记录表"]{
  id int [pk, increment, note:"视频记录 ID"]
  camera_id int [note: "摄像头 ID"]
  timestamp datetime [note: "记录时间"]
  file_path enum("处理中","已处理") [note:"视频状态"]
}

Table abnormal_behaviors [note:"异常行为记录表"]{
```

```

id int [pk, increment, note:"异常行为记录 ID"]
video_id int [ref: > video_records.id,note: "视频记录 ID"]
behavior_type enum("逆行","超速") [note: "异常行为类型"]
location varchar [note: "行为发生地点"]
timestamp datetime [note: "行为发生时间"]
severity int [note: "行为的严重程度(1~5)"]
}

```

```

Table traffic_flow [note: "交通流量分析表"] {
id int [pk, increment, note: "流量记录 ID"]
location varchar [note: "监控的路口位置"]
timestamp datetime [note: "流量数据采集时间"]
vehicle_count int [note: "车辆数量"]
pedestrian_count int [note:"行人数量"]
}

```

```

Table decision_records [note: "决策记录表"] {
id int [pk,increment, note: "决策记录 ID"]
timestamp datetime [note: "决策生成时间"]
decision_details text [note:"决策具体内容"]
executed bool [note:"决策是否执行"]
}

```

各表通过外键进行关联，例如，异常行为记录表和视频记录表通过 `video\_id` 进行关联。



## 里程碑

### 环境搭建与前后端功能实现 (第 1-3 天)

- Flutter 前端开发环境搭建完毕。
- MySQL 数据库部署完成并创建基础数据表。
- 后端视频处理模块的开发环境配置好，能读取视频流数据。
- 完成 Flutter 前端 UI 设计，支持显示实时监控视频和异常行为警报。
- 实现前端与后端 API 的交互，能够从后端获取交通数据。

### 视频处理与行为识别算法、优化决策模块与数据库集成 (第 4-7 天)

- 视频处理模块能够对实时视频进行解析，准确检测异常行为。
- 行为识别算法训练完成，并能处理至少 70% 的行为识别准确率。
- 完成多信息融合的优化决策算法开发。
- 系统生成的决策能够准确提供交通管理建议，并存入数据库。

### 系统测试与性能优化 (第 8-9 天)

- 完成整个系统的集成测试，确保前后端交互流畅。
- 针对性能进行优化，包括前端响应速度和后端处理效率。

### 部署与上线 (第 10-14 天)

- 将系统部署到服务器，并在实际环境中进行测试。
- 完成正式版本发布，并生成项目总结报告。

## 最终成果

### 功能验收

- 校验系统是否满足需求部分的所有 Must Have 功能，例如实时监控、异常行为识别、优化决策生成等。
- 前端 Flutter 应用在 Android 和 iOS 平台上的一致性与稳定性。

### 性能评估

- 测试视频处理模块的帧率以及异常行为检测的准确率，目标是检测准确率在 80% 以上。
- 评估系统的响应时间，确保从异常行为检测到决策推送的时间在合理范围内（例如不超过 5 秒）。

## 用户反馈

- 收集用户体验反馈，特别是对 UI 设计、使用流畅度和报警机制的反馈。
- 根据用户反馈调整界面和算法，提升系统易用性和准确性。

## 系统稳定性和扩展性测试(待定)

- 监控系统的稳定性，评估其在高并发情况下的表现。
- 检查数据库性能，确保支持大规模的交通数据存储和处理。

## 类似案例

### 1. 华南师范大学校园智能交通管理系统

1. [j.jscnun.2014.06.029.pdf](#)

### 2. 基于车联网技术的校园智能交通管理系统

### 3. 高校校园交通分析与智能化管理系统