

# 《警察数数\_φ(□\_□◇)》

—公共地点人流量计算云监管平台

详细设计说明书

队伍名称: Invictus Dale

学校: 北京工业大学

## 目录

1 引言.....	3
1.1 背景.....	3
1.2 文档阅读者.....	3
2 综合描述.....	3
2.1 产品概况.....	3
2.2 运行环境.....	3
3 总体设计.....	3
3.1 总体框架.....	3
3.2 系统整体结构和及子系统划分.....	3
3.3 各子系统主要技术栈.....	4
3.4 整体数据交互.....	4
4 详细设计.....	5
4.1 数据库设计.....	5
4.2 安卓边缘 App 子系统.....	5
4.3 云端 H5 平台子系统.....	6
4.4 后台子系统.....	7
4.5 目标检测模块.....	7
5 主要功能说明.....	7
5.1 保持登录功能.....	7
5.2 边缘端异常检测功能.....	7
5.3 高危地点统计功能.....	8
6 待改进的地方.....	8

# 1 引言

## 1.1 背景

随着以深度学习为代表的人工智能技术的蓬勃发展，越来越多的智能化应用涌现并高效准确地服务着人类，解决了许许多多的传统人类难以感知和解决的问题。与此同时，5G 时代的到来，催生着各种边缘计算与云计算结合的数据处理方案，《警察数数》公共地点人流量计算云监管平台由此而来，本平台以计算机视觉为核心，边云协同架构为创新点，实现对公共地点人流量的实时智能监控。

## 1.2 文档阅读者

- (1) 开发人员。为系统设计、开发、测试等做参考。
- (2) 其他相关人员。了解本系统架构、主要系统功能等。

# 2 综合描述

## 2.1 产品概况

《警察数数》公共地点人流量监管平台分为三个部分：一是以 Django 为基础后台服务框架，负责整合后台各种静态资源与计算资源，响应云端与边缘端的服务请求；二是以 Vue 为基础的云端 H5 平台，负责综合数据管理与展示，视频上传等功能；三是以 Android 为基础的边缘端 APP 应用程序，主要负责进行实时目标检测任务和与云端的数据交互。

## 2.2 运行环境

边缘端 App: Android5.0 级以上的智能手机。

云端 H5 平台: 有互联网连接的 Win XP 以上系统的 Windows、IOS 计算机。

# 3 总体设计

## 3.1 总体框架

以边缘计算架构为基础，设计统一的数据存取标准，实现前后台以及边缘端程序的交互，各模块独立开发，并预留调用接口，最终整合成一个完整的多端平台。

## 3.2 系统整体结构和及子系统划分

系统总体结构划分如下图 3.1 所示。

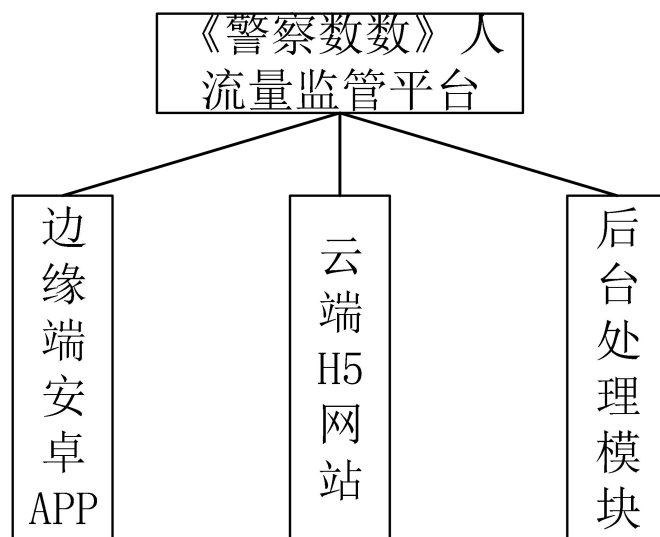


图 3.1 系统整体结构划分

### 3.3 各子系统主要技术栈

- 边缘端安卓 APP: QMUI+TensorFlowLite+SSD
- 云端 H5 网站: NodeJs+Vue+Echart+elementUI
- 后台处理模块: Django+TensorFlow+Faster-RCNN+FFmpeg+MySQL

### 3.4 整体数据交互

《警察数数》人流量云监管平台整体系统数据流图如下图 3.2 所示。

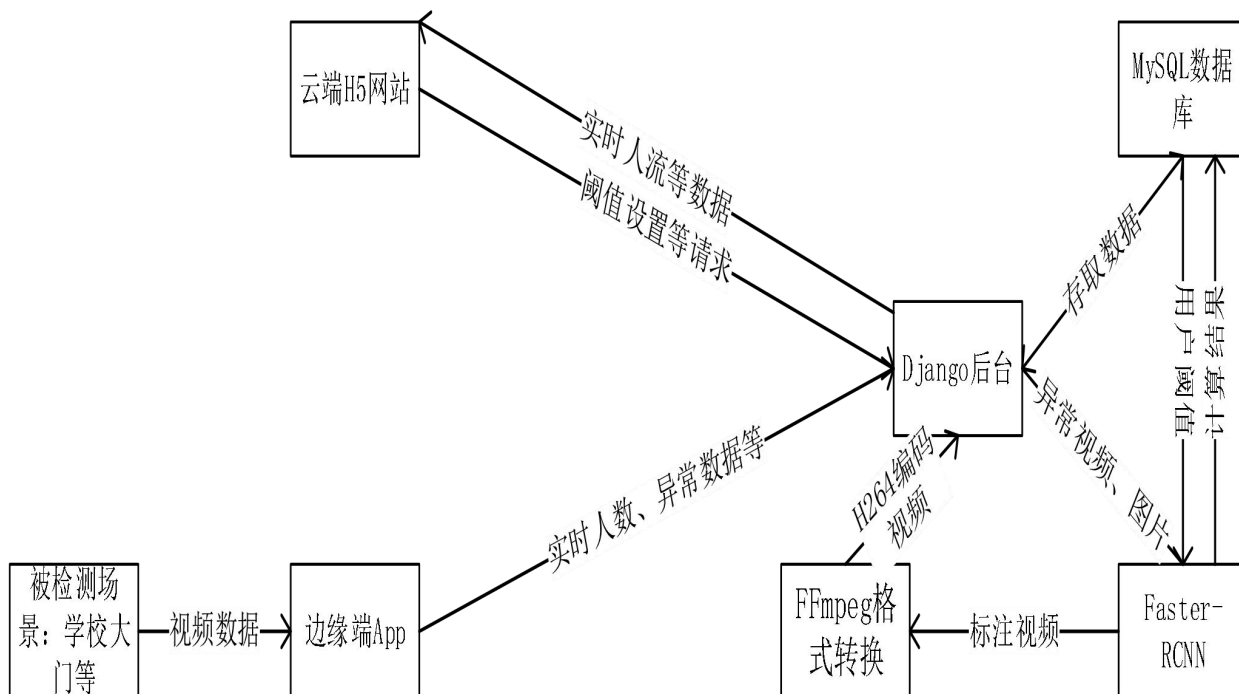


图 3.2 平台整体系统数据流图

## 4 详细设计

### 4.1 数据库设计

本平台数据库采取 MySQL，包含 Auth\_User 等 19 个表用于对用户账号、人流监测、用户阈值等数据的存储和读取。数据库表目录如下图 4.1 所示。



图 4.1 数据库表目录

以用户阈值表为例。该表用于存储用户在平台上所设置的阈值信息，作为云端、后台、边缘 App 端阈值的统一更新表。它一共有三个字段，分别是：编号，阈值，用户名如表 4.1 所示，是用户阈值表的定义信息。

表 4.1 用户阈值表

字段名	数据类型	说明
id	Int	自增列 主键
threshold	varchar	阈值
username	varchar	用户名

### 4.2 安卓边缘 App 子系统

除了基本的数据请求与展示功能,边缘端 App 子系统利用谷歌 TensorFlowLite 框架，部署经过了 op 转换的深度学习模型，实现了深度学习任务在手机上的高效执行。本地执行目标检测任务，并对每次的目标检测任务结果进行分析，对于

超过阈值的异常数据，上传云端，利用更高精度的 **Faster-RCNN** 模型进行人数的识别。

此部分主要功能模块包括：

1. 登录注册。
2. 功能登陆验证。
3. 阈值设置。
4. 检测场景设置。
5. 快速目标检测与异常上传。
6. 查看上传异常的云端计算结果。
7. 接收来自云端的监测建议。

此部分的功能亮点：实现了在边缘端的高效智能目标检测任务，同时能够接收来自服务器的参数更改，符合边云协同的架构思想。此部分使用的深度学习模型，是用 **coco** 数据集训练的，以轻量 **mobilenet** 为网络结构的 **SSD** 模型。该模型本身就具有轻量快速的特点，再经过 **TensorFlowLite** 的格式转化，得到一个体积小，但是几乎保持不变的移动轻量模型。

### 4.3 云端 H5 平台子系统

云端 H5 平台主要承担数据展示以及数据分析的作用。此部分的设计思路为，组件化开发，将每一个功能构建成为单独的组件，实现组件的复用。此部分的主要功能包括：

1. 注册与登陆。
2. 保持登陆状态。
3. 视频计算结果查看。以在线视频的方式查看后台对视频的计算结果。
4. 实时展示人流数据。
5. 查看边缘端上传异常。
6. 查看实时人流排行榜。
7. 查看在线设备数。
8. 人流历史统计。
9. 高危地点统计。
10. 监测建议。

11. 阈值设置。

12. 上传视频文件。可以自行上传视频文件进行目标检测任务。

此部分的功能亮点：实现了多种数据展示功能。实时人流数据采取轮询访问服务器的方式，实现了对人流数据的实时展示效果。

#### 4.4 后台子系统

后台子系统主要实现了对前台和边缘端请求的接收、计算与相应功能。此部分主要设计思路为，函数和模块式相应，不同的相应请求对应相应的函数计算，相互独立，互不干扰，可扩展性强。

此部分主要实现的功能：

1. 路由。解析来自 H5 平台和边缘端请求，映射到相应的响应函数实现响应。

2. 数据库连接。实现对数据库的增删改查。

3. 排队计算。由于目标检测任务耗费大量计算资源的，因此利用多线程实现了任务的排队计算，保证了在同一时刻只有一个目标检测任务进行，防止后台程序崩溃。

#### 4.5 目标检测模块

此部分利用 brainwash 和 SCUT 人头数据集训练了一个以 inceptionV2 为基础的 Faster-RCNN 模型，模型具有目标检测任务精度高，但是计算速度相对较慢。因此部署在服务器端，作为边缘 SSD 计算模型的重要补充。

### 5 主要功能说明

#### 5.1 保持登录功能

采用 session 与 cookie 结合的方式，在用户第一次登陆的时候，服务器端自动生成 sessionid 并分发到 H5 网站，并存储在 cookie 中，之后每发起请求时携带 cookie 进行请求，以此达到了保持登陆状态的功能。

#### 5.2 边缘端异常检测功能

边缘端的计算模型读取手机相机，获取实时的图片数据并转为 bitmap 格式，进行图像缩放等图像预处理后送入模型进行目标检测计算，对计算结果中置信度超过 50%的结果进行统计，若总数超过用户所设置的阈值，则存储预处理之前的图片，并上传到云端服务器，进行后续的高精度异常检测任务。

### 5.3 高危地点统计功能

此功能对不同地点的一个时间窗口内的人流记录进行求均值运算，对结果进行排序，实现对高危地点的统计功能。

## 6 待改进的地方

由于手机端在进行目标检测任务的过程中摄像头占用，无法对异常视频数据进行存储，除此以外上传视频占用大量的资源，会造成程序的卡顿，并且消耗大量的手机流量，因此边缘 App 端改为了存储异常图片并上传。除此以外，由于构建的场景识别模型无法准确识别地点而导致无法进行地点异常统计，因此边缘端 app 改为手动设置识别场景，并以不同场景存储人流数据，为后续数据分析工作提供准确无误的数据基础。