

strdsty 使用手册

目录

文 件 修 订 记 录	1
1. 总体功能描述	2
2. 运行环境	3
硬件要求	3
软件要求:	3
3. 软件编译环境	4
1) 数据采集端	4
训练	4
测试	4
2) 服务器端	6
1.1. 批量上传摄像头数据	8
1.2. 查询所有摄像头数据	9
1.3. 根据经纬度查询摄像头数据	9
3) 小程序客户端	10
4. 软件安装说明	11
5. 小程序启动	12
6. 授权确认与拒绝	13
7. 地图功能	15
8. 数据库软件	16
9. 数据库脚本文件	17

文件修订记录

版本号	生成日期	作者	修订内容
V1.0	2022-04-11	合作开发	原始版本

1. 总体功能描述

系统能够通过多线程监控实现动态检测一个封密环境内的人流密度，能动态分析与之相连的监考系统的监控画面并分析得出各个区域的人数，客户端可以访问服务器获取统计结果。

系统分三层结构，数据采集端、服务端以及小程序客户端。

数据采集端主要通过深度学习从 malldataset 数据集中随机选取 10 张图片进行人数预测，然后用复现模型训练进行数据计算并将相应数据打包成 json 数据包，采用 http 协议，发送至相关的 MySQL 数据库中。

服务端采用 SpringBoot+Mybatis 框架，数据库采用的是 MySQL 数据库。服务端采用三层架构进行解耦，Mapper 层向下连接 MySQL 数据库，Service 层连接 Mapper 层与 Controller 层，Controller 层则是对外提供接口。

小程序客户端使用 HTTP 协议按照格式访问服务端提供的接口，从而达到数据上云和数据采集端与客户端通信的效果。

2. 运行环境

硬件要求

类 别	基本要求
服务器端	CPU 4 核 GPU GTX 1060(6G) 内存 8G 以上；硬盘剩余空间不低于 50G；
客户端	手机 8G 内存及以上；硬盘空间 128G 及以上

软件要求：

类别	名 称	基本环境
服务器端	操作系统	支持 Windows 10
	数据库软件	支持 MySQL5.6
	深度学习框架	Tensorflow 2.2
	第三方库	Opencv-py 等
客户端	操作系统	Android
	其它软件	微信

3. 软件编译环境

1) 数据采集端

本系统数据采集端使用 python 3.6 语言、深度学习框架 Tensorflow 2.2 以及相关第三方库 opencv 开发，需要用相同语言、框架等编译。

使用说明

深度学习:进入项目的根目录的 scripts 目录下，输入命令行指令进行相关训练与测试。本程序训练的模型为根据 Multi-scale Convolution Neural Networks for Crowd Counting 论文复现的 MSCNN 模型。

训练: python train.py

运行参数:

```
-h, --help                show this help message and exit
-e EPOCHS, --epochs EPOCHS
                           how many epochs to fit
                           默认值: 5
-v SHOW, --show SHOW     if show training log
                           默认值: 'yes'
-b BATCH, --batch BATCH
                           batch size of train
                           默认值: 8
-d DATASET, --dataset DATASET
                           which dataset to train
                           默认值: 'malldataset'
-p PRETRAINED, --pretrained PRETRAINED
                           load your pretrained model in folder
                           root/models
                           默认值: 'no'
```

例如: python train.py -e 50 -b 8 -d malldataset

表示对神经网络进行 50 次迭代训练，每次训练样本为 8 个，投喂的数据集为'malldataset'

目前支持训练的数据集有三个: malldataset、shanghaitechdataset、Crowdataset，其中 Crowdataset 为自制数据集

训练建议: 对于基本要求的软硬件配置条件下，batch 不应大于 8。针对 malldataset 数据库进行 50 次迭代训练约需要 4~6h。

若需要支持训练新的数据集则需要在 data 模块里编写相关的功能类。

测试: python train.py

运行参数:

```
-h, --help                show this help message and exit
-s SHOW, --show SHOW     if show test result map
                           默认值: 'yes'
-d DATASET, --dataset DATASET
```

root/models

默认值: 'malldataset'

例如: `python test.py -d malldataset`

表示从 `malldataset` 数据集中随机选取 10 张图片进行人数预测,生成一张包含原图,原图真实人数,预测的密度图像,预测人数的图片,并保存至根目录的 `results` 下。

目前支持测试的数据集有三个: `malldataset`、`shanghaitechdataset`、`Crowdataset`,其中 `Crowdataset` 为自制数据集

若需要支持测试新的数据集则需要在 `data` 模块里编写相关的功能类。

数据计算:编译运行根目录下的 `scripts` 中的 `fdd.py`。

目前仅支持模拟情况下的实时数据计算。

主要功能:复现模型(该模型通过在多个数据集上进行训练测试获得,模型泛化能力强,实际主要 `malldataset` 数据集上训练获得模型。),若时间更新(以当前分钟更新为准)则读取根目录下的 `data` 路径下的十个文件夹中的相关图片(模拟的十个监控平台)。图像预处理之后,进行人数预测。将预测结果打包成 `json` 数据包,采用 `http` 协议,发送至相关的 `MySQL` 数据库中。若读取图片失败,即认为监控设备可能处在维修或网络连接故障,返回指定的值用于表示异常。在基础要求的软硬件配置下,此过程大于需要 `3s+5s`,其中 `5s` 用于延时解决因短暂的网络问题导致的图片上传延迟的问题。若发送的数据被对应的服务器成功接受并保存至 `MySQL` 数据库中则返回 `200`,反之可能为 `400` 等值。相关信息会打印在命令行窗口。注释指定的 `print` 语句可能实现信息打印的屏蔽。

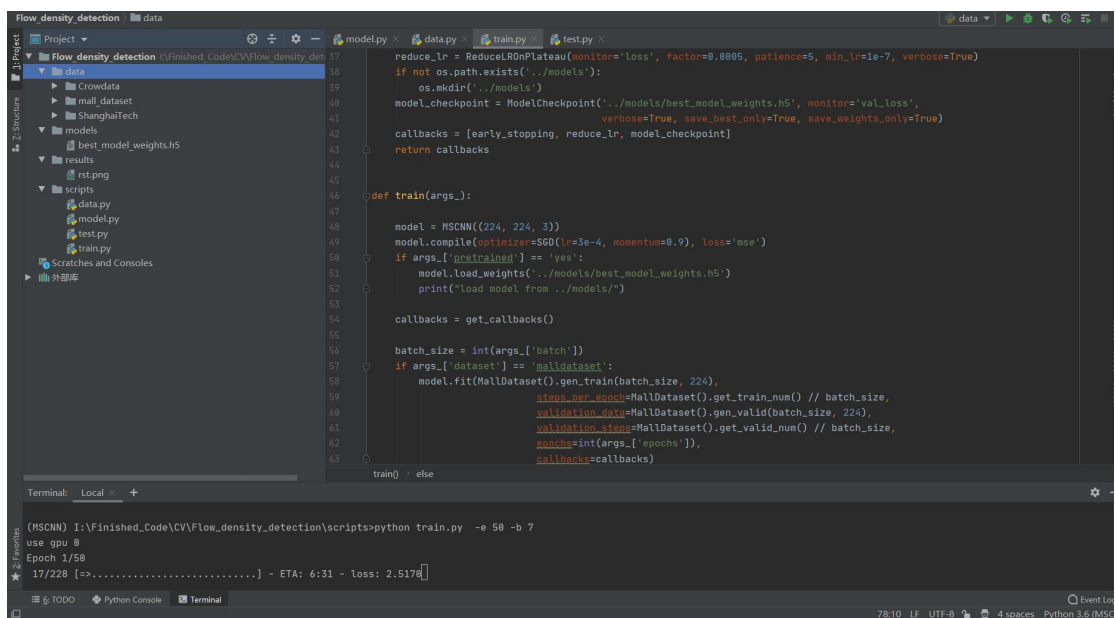


图 1: 训练界面

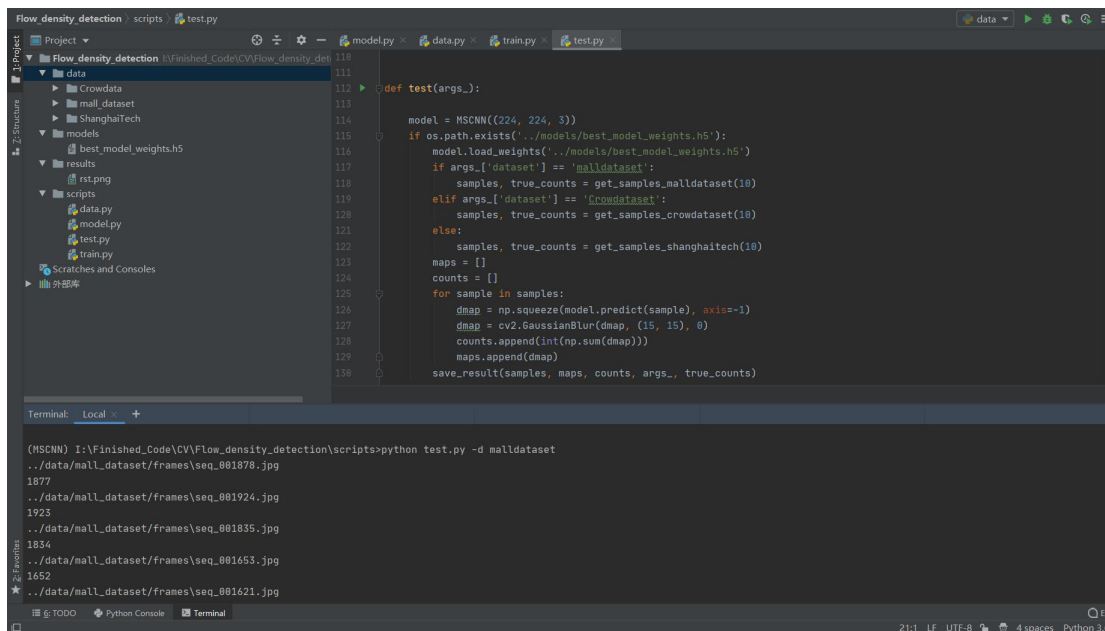


图 2：测试界面

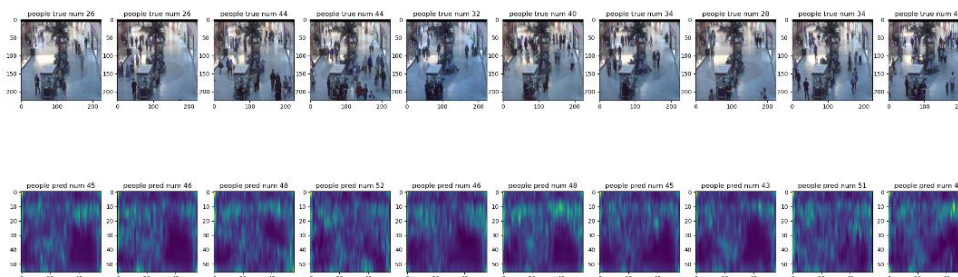


图 3：测试结果

2) 服务器端

- 本系统服务器端使用 Idea 进行开发，需要使用相同软件进行开发编译。

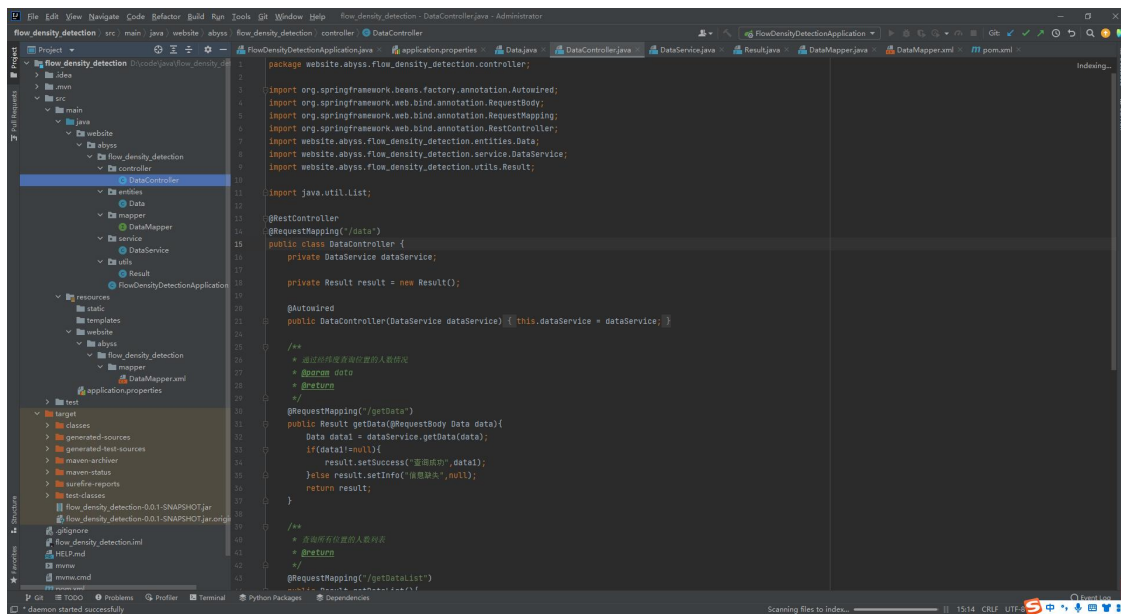


图 4：服务器端运行界面

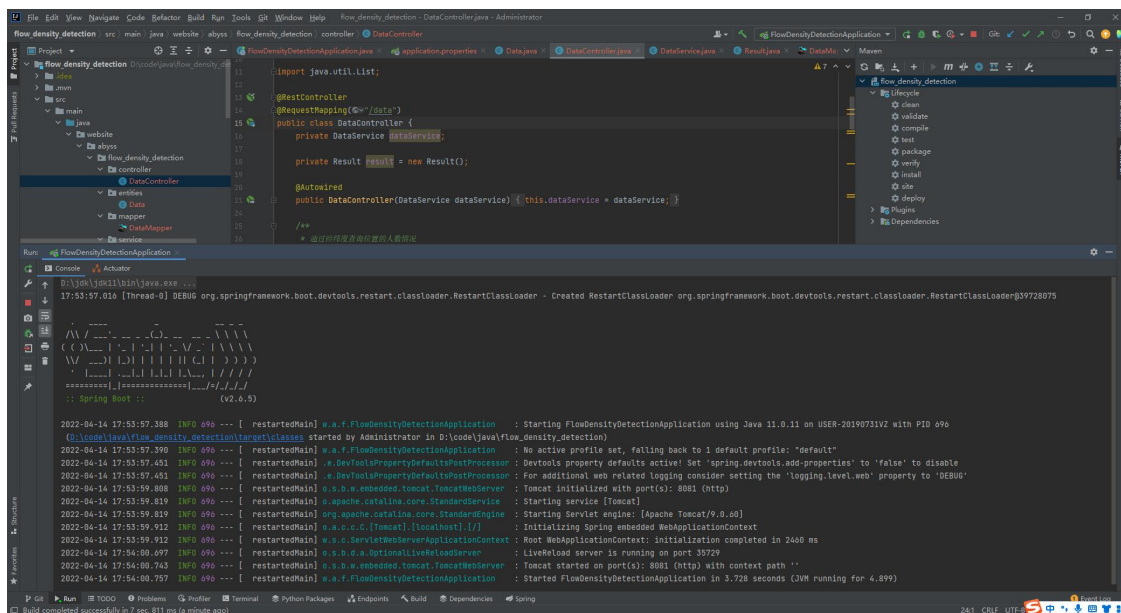


图 5：运行成功画面

● 服务器端部署说明

服务端通过 IDE 的 package 命令打包打成 jar 包，然后进入到 jar 包对应目录下，再输入该命令即可运行成功：

```
nohup java -jar flow_density_detection-0.0.1-SNAPSHOT.jar &
exit
```

● 数据包格式

当访问接口成功时，状态码为 200，失败则为 400。

```

public class Result {
    //状态码
    private int code;
    //提示信息
    private String msg;
    //数据
    private Object result;
    //操作成功
    public void setSuccess(String msg, Object result){
        this.code=200;
        this.msg=msg;
        this.result=result;
    }
    //操作失败
    public void setInfo(String msg, Object result){
        this.code=400;
        this.msg=msg;
        this.result=result;
    }
}

```

图 6：数据包格式

● 接口说明

1.1. 批量上传摄像头数据

URL 服务器 IP: 端口号/data/setDataList

接收参数 List(装 Data 类的 List 列表的 json 格式)

返回值 Result 类 (json 格式)

若是新摄像头节点访问该接口则执行下列 sql 语句:

```
insert into `data`(camera_id,latitude,longitude,num) values ({cameraId},{latitude},{longitude},{num})
```

若是老摄像头节点访问该接口则执行下列 sql 语句:

```
update `data` set num = {num} where latitude = {latitude} and longitude = {longitude}
```

1.2. 查询所有摄像头数据

URL 服务器 IP: 端口号/data/getDataList

接收参数 无

返回值 List(装 Data 类的 List 列表的 json 格式)

sql 语句:

```
select * from `data`
```

1.3. 根据经纬度查询摄像头数据

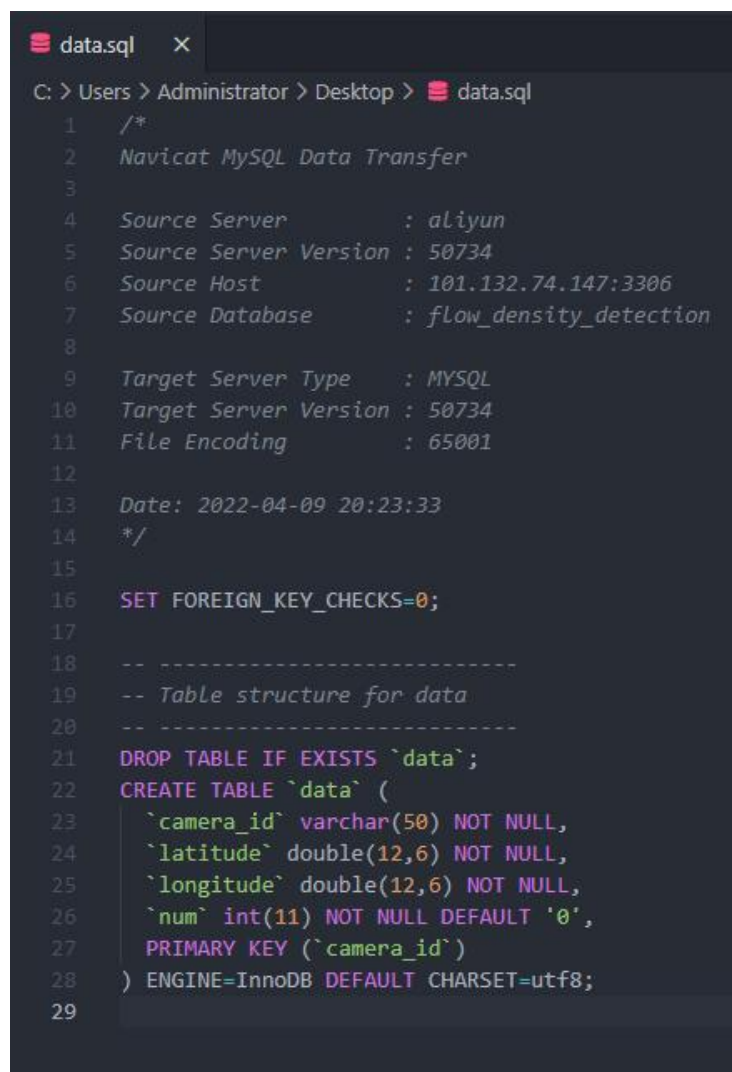
URL 服务器 IP: 端口号/data/getDataList

接收参数 Data 类 (json 格式)

返回值 Data 类 (json 格式)

sql 语句:

```
select * from `data` where latitude = #{latitude} and longitude = #{longitude}
```



```
data.sql x
C: > Users > Administrator > Desktop > data.sql
1  /*
2  Navicat MySQL Data Transfer
3
4  Source Server      : aliyun
5  Source Server Version: 50734
6  Source Host       : 101.132.74.147:3306
7  Source Database  : flow_density_detection
8
9  Target Server Type : MYSQL
10 Target Server Version: 50734
11 File Encoding    : 65001
12
13 Date: 2022-04-09 20:23:33
14 */
15
16 SET FOREIGN_KEY_CHECKS=0;
17
18 -- -----
19 -- Table structure for data
20 -- -----
21 DROP TABLE IF EXISTS `data`;
22 CREATE TABLE `data` (
23   `camera_id` varchar(50) NOT NULL,
24   `latitude` double(12,6) NOT NULL,
25   `longitude` double(12,6) NOT NULL,
26   `num` int(11) NOT NULL DEFAULT '0',
27   PRIMARY KEY (`camera_id`)
28 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
29
```

图 7: 数据库脚本文件

3) 小程序客户端

本软件小程序客户端采用微信开发者工具进行开发，需要用相同软件进行编译。

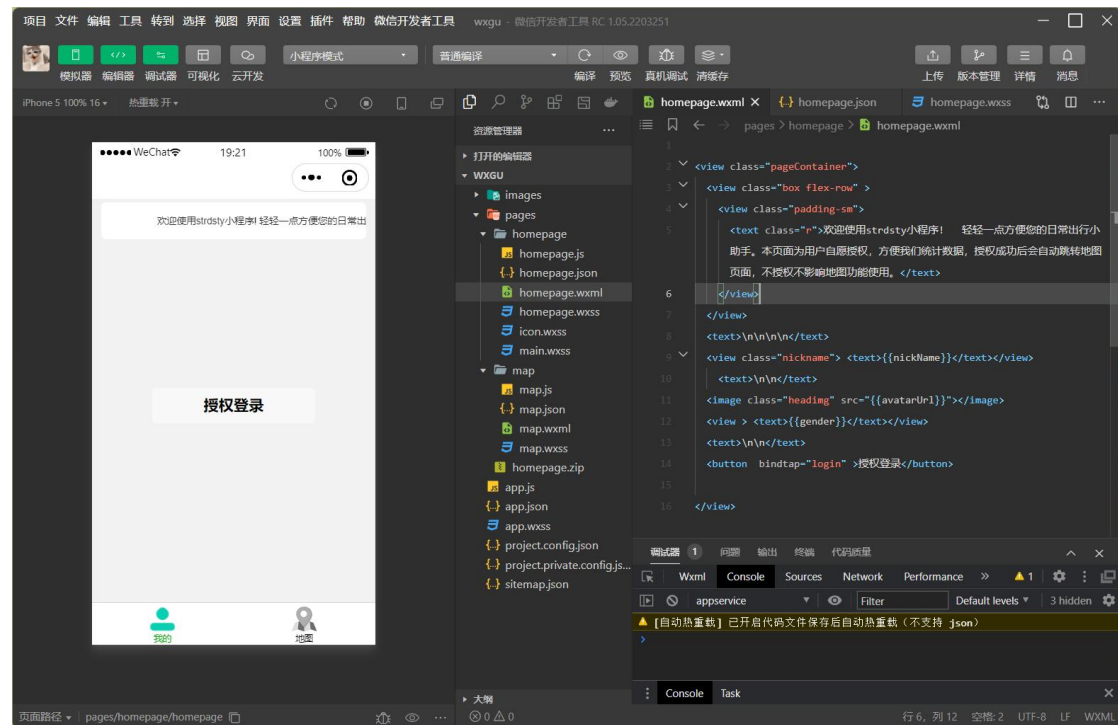


图 8：小程序客户端运行成功界面

4. 软件安装说明

本软件将发布至微信小游戏，可以在微信中搜索相关名称，然后点击使用。小程序默认页面：页面顶部设置了滚动字幕作为公告栏，方便显示系统通知和用户须知等内容。



图 9 程序默认界面

5. 小程序启动

程序默认状态为用户身份未授权的状态，用户自愿授权，方便后台统计某些数据，授权成功后会自动跳转地图页面，不授权不影响地图功能使用。最底部是程序的导航栏，分为“我的”和“地图”两部分，分别用于显示授权的用户信息和各监测点的实时人数。



图 10：授权确认界面

6. 授权确认与拒绝

当用户点击“授权登录”按钮后，页面底部会出现弹窗，此时用户可以点击“拒绝”或“允许”选择是否对程序进行授权。

若用户同意授权，程序会跳转至“地图”页面，直接显示各观测点的实时人数，用户可直接查看，效果如图 11。

若用户拒绝对程序进行授权，程序会显示对应的消息提示框对用户提示，此时程序不会获取用户信息，效果如图 12。



图 11: 授权跳转页面



图 12: 拒绝授权页面

完成授权后，用户点击底部“我的”，会跳转用户信息页面，显示用户名和头像基本信息。



图 13: 授权后用户信息页面

7. 地图功能

程序通过访问服务器的数据库获取每个监测点的实时人数并显示在当前页面。如果获取的数据类型或数据范围出现错误，会在站点信息中显示“设备故障”。本次测试在江苏大学内设置了 10 个监测点，因此设置江苏大学为地图中心点，用户可以通过缩放查看各个监测点的实时信息。



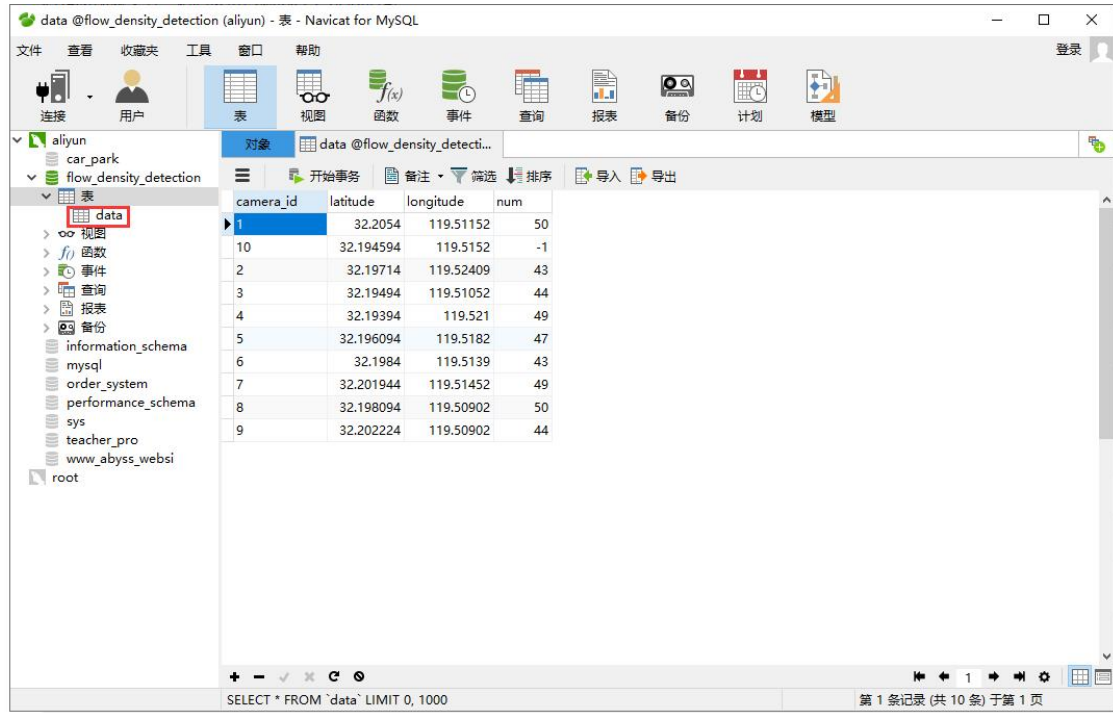
图 14 人数显示页面



图 15 监测点信息页面

8. 数据库软件

数据库使用的是 mysql5.6，表如下



data @flow_density_detection (aliyun) - 表 - Navicat for MySQL

文件 查看 收藏夹 工具 窗口 帮助

连接 用户 表 视图 函数 事件 查询 报表 备份 计划 模型

对象 data @flow_density_detect...

开始事务 备注 筛选 排序 导入 导出

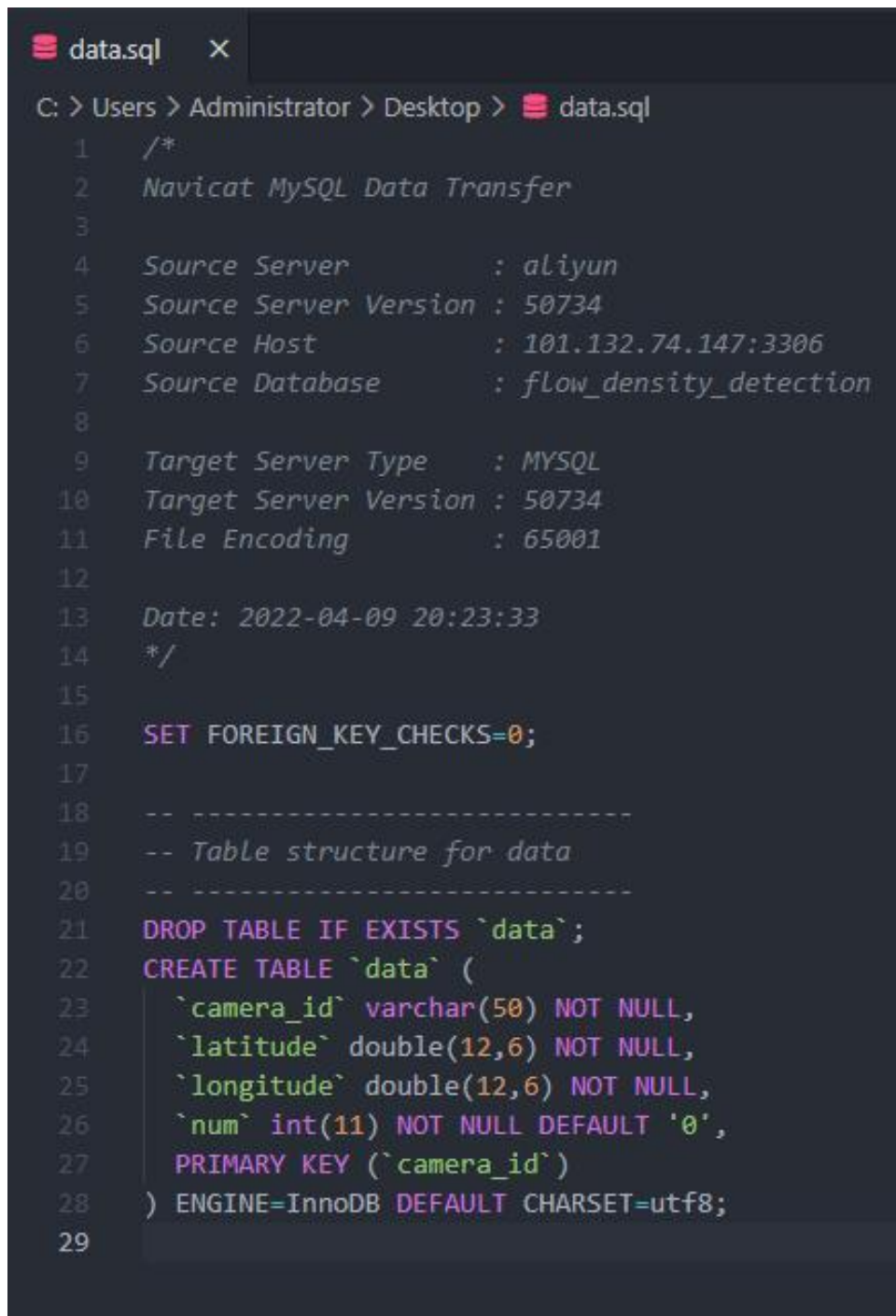
camera_id	latitude	longitude	num
1	32.2054	119.51152	50
10	32.194594	119.5152	-1
2	32.19714	119.52409	43
3	32.19494	119.51052	44
4	32.19394	119.521	49
5	32.196094	119.5182	47
6	32.1984	119.5139	43
7	32.201944	119.51452	49
8	32.198094	119.50902	50
9	32.202224	119.50902	44

SELECT * FROM `data` LIMIT 0, 1000

第 1 条记录 (共 10 条) 于第 1 页

图 16：数据库表

9. 数据库脚本文件



```
data.sql X
C: > Users > Administrator > Desktop > data.sql
1  /*
2   Navicat MySQL Data Transfer
3
4   Source Server      : aliyun
5   Source Server Version : 50734
6   Source Host        : 101.132.74.147:3306
7   Source Database     : flow_density_detection
8
9   Target Server Type   : MYSQL
10  Target Server Version : 50734
11  File Encoding        : 65001
12
13  Date: 2022-04-09 20:23:33
14  */
15
16  SET FOREIGN_KEY_CHECKS=0;
17
18  -- -----
19  -- Table structure for data
20  -- -----
21  DROP TABLE IF EXISTS `data`;
22  CREATE TABLE `data` (
23    `camera_id` varchar(50) NOT NULL,
24    `latitude` double(12,6) NOT NULL,
25    `longitude` double(12,6) NOT NULL,
26    `num` int(11) NOT NULL DEFAULT '0',
27    PRIMARY KEY (`camera_id`)
28  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
29
```

图 17: 数据库脚本文件