基于大数据的

教室资源调控系统

详细设计说明书

Ver 1.6

|  |  |  |  |
| --- | --- | --- | --- |
| **项目名称** | 基于大数据的教室资源调控系统 | | |
| **项目编号** |  | **项目经理** | 严根 |
| **文档编号** |  | **文档版本** | 1.6 |
| **编制** | 全体成员 | **审批** | 严根 |

**修订记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修订日期** | **修订描述** |
| 1.0 | 严根 | 2021.3.19 | 整理全部格式，初步审稿 |
| 1.1 | 严根 | 2021.3.19 | 修改目录页码 |
| 1.2 | 陶义帆 | 2021.3.23 | 完善模块部分，添加图表题注 |
| 1.3 | 谢梓聪 | 2021.3.25 | 修改并完善缩略语部分 |
| 1.4 | 严根 | 2021.3.25 | 进一步修改标题级别 |
| 1.5 | 陈云 | 2021.3.26 | 修改表格注释、内容格式，模块关系图 |
| 1.6 | 王美婷、陶义帆 | 2021.3.26 | 修改目录、标题 |

目录

[1. 引言 1](#_Toc67599065)

[1.1 编写目的 1](#_Toc67599066)

[1.2 适用范围 1](#_Toc67599067)

[1.3 缩略语 1](#_Toc67599068)

[2. 开发环境 2](#_Toc67599069)

[3. 子系统逻辑结构 2](#_Toc67599070)

[3.1 模块功能 2](#_Toc67599071)

[3.1.1 图像识别模块 2](#_Toc67599072)

[3.1.2 数据传输模块 3](#_Toc67599073)

[3.1.3 智能推荐模块 3](#_Toc67599074)

[3.2 模块组成 3](#_Toc67599075)

[3.2.1 模块总关系 3](#_Toc67599076)

[3.2.2 图像识别模块 3](#_Toc67599077)

[3.2.3 数据传输模块 3](#_Toc67599078)

[3.2.4 智能推荐模块 4](#_Toc67599079)

[4. 模块实现设计 4](#_Toc67599081)

[4.1 图像识别模块程序设计 4](#_Toc67599082)

[4.1.1 数据集制作 4](#_Toc67599083)

[4.1.1.1 数据集获取 4](#_Toc67599084)

[4.1.1.2 数据集划分 4](#_Toc67599085)

[4.1.1.3 数据集打标 5](#_Toc67599086)

[4.1.2 Yolov5模型建立 6](#_Toc67599087)

[4.1.2.1 具体模型选择 6](#_Toc67599088)

[4.1.2.2 修改数据配置文件 6](#_Toc67599089)

[4.1.2.3 修改模型配置文件 6](#_Toc67599090)

[4.1.3 Yolov5模型训练 7](#_Toc67599091)

[4.1.3.1 选择预训练权重 7](#_Toc67599092)

[4.1.3.2 确定训练模型参数 7](#_Toc67599093)

[4.1.3.3 执行训练命令 7](#_Toc67599094)

[4.1.3.4 可视化分析 7](#_Toc67599095)

[4.1.4 Yolov5模型验证 8](#_Toc67599096)

[4.1.4.1 修改检测框参数 8](#_Toc67599097)

[4.1.4.2 模型预测 8](#_Toc67599098)

[4.2 数据传输模块程序设计 8](#_Toc67599099)

[4.3 智能推荐模块程序设计 9](#_Toc67599100)

[4.3.1 空教室查询后端设计 9](#_Toc67599101)

[4.3.2 智能推荐方案算法设计 9](#_Toc67599102)

[4.3.3 小程序前端界面设计 10](#_Toc67599103)

[5. 数据库设计 10](#_Toc67599104)

[5.1 数据访问频度和流量 10](#_Toc67599105)

[5.2 数据库选型 10](#_Toc67599106)

[5.3 数据安全性及保密设计 10](#_Toc67599107)

[5.4 数据字典设计 12](#_Toc67599108)

## 1. 引言

### 1.1 编写目的

在概要设计阶段中，已解决了实现该系统需求的程序模块设计问题，包括系统模块间的关系设计，以及数据结构和模块结构的设计等。在本报告中将对在本阶段中对系统所做的所有详细设计进行说明。

本阶段依据概要设计的要求，确定应该如何具体地实现具有目标功能的系统，从而为产品开发阶段将描述直接翻译成用具体的程序语言书写的程序打下基础。

### 1.2 适用范围

团队成员

产品审核人员

相似系统的设计人员

### 1.3 缩略语

表 1 缩略语

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | YOLOv5 | 一种目标检测算法 |
| 2 | GPU | 图形处理单元 |
| 3 | ssh | 安全外壳协议 |

## 2. 开发环境

表 2 开发环境

|  |  |  |
| --- | --- | --- |
| **工具名称** | **作用** | **备注** |
| Linux下Pycharm+Python3.8+Pytorch | 图像识别模块深度学习训练开发环境 | 由于深度学习对GPU的要求，且大数PC不符合要求，因此采用Colab（GPU环境）挂载谷歌云文件的方式进行模型训练,后续会部署树莓派下 |
| 树莓派下 Python3.8 | 用于将图像识别得到的人数数据更新到服务器的数据库上以便后续智能推荐使用 | 利用python ssh连接远程服务器并更新数据库数据 |
| Windows下  微信开发者工具小程序前端+Java Spring服务器后端 | 智能推荐系统开发环境 | 采用小程序前端便捷开发环境和Java Spring boot强大后端处理 |

## 3. 子系统逻辑结构

### 3.1 模块功能

#### 3.1.1 图像识别模块

该模块用于从教室各内置摄像头获取实时图像并传输到树莓派上，并通过使用深度学习训练后的模型进行人头识别从而获取人数。

#### 3.1.2 数据传输模块

衔接图像识别模块，用于把树莓派本地获取到的各教室人数数据传输到服务器端，实时更新部署好的数据库教室表人数这一属性值。

#### 3.1.3 智能推荐模块

前端小程序界面用于用户使用，打开后系统会在基础空教室查询模块的基础上基于时间地点和历史用户查询进行智能推荐。

### 3.2 模块组成

#### 3.2.1 模块总关系

实时更新数据库

智能推荐模块

教室人数

数据传输模块

图像识别模块

---

图 1 模块总关系

#### 3.2.2 图像识别模块

Yolov5模型验证

Yolov5模型训练

Yolov5模型建立

数据集制作

图 2 图像识别模块

#### 3.2.3 数据传输模块

数据库更新

pymysql连接到云端数据库

ssh连接到服务器

图 3 数据传输模块

#### 3.2.4 智能推荐模块

#### 

空教室查询模块

小程序前端界面

智能推荐方案设计

图 4 智能推荐模块

## 4. 模块实现设计

### 4.1 图像识别模块程序设计

#### 4.1.1 数据集制作

#### 4.1.1.1 数据集获取

由于我们研发的是一个基于大数据的教室资源调控系统，因此我们的数据集就来自我们本校的教室。我们通过教务处的摄像头共获取了2000张有关教室的图片，并将其格式改为.jpg存在images目录中。

#### 4.1.1.2 数据集划分

在获得这些照片后，将其分为三部分训练集，验证集，测试集。训练集：验证集：测试集=8:1:1，所以训练集共1600张，验证集200张，测试集200张，除此需要创建一个python脚本用于按可改变的比例划分三类数据集并分别生成相应的图片名称集到train.txt、test.txt、val.txt，且该过程为随机划分。

#### 4.1.1.3 数据集打标

用labelImg软件对获取的训练集，验证集进行标注，对准图片上有人头的地方进行画框，画完框之后会显示对该框的标注类别，选择person这个选项。labelImg默认的class分类有很多，这里我们只需检测人即可，所以可将除了person之外的其他类别删除。

标注后一般会生成一种.xml文件用于保存标注后的人头框位置和类别信息，接着将.xml文件全部保存在annotations目录下。

生成包含标注信息的xml文件后，由于xml文件标签内容并不是yolov5模型训练数据集的标准数据集格式，因此我们还要对数据进行进一步加工处理。即使用python编写一个vocLabel脚本来从xml文件中提取标注后的具体信息， 提取到的内容一行代表一个目标，具体格式为class x\_center y\_center width height。第一列是class的索引，计数从0开始，由于我们只检测人且分类中只有人，所以第一列全部是0，后4列是x\_center/image\_width、y\_center/image\_height，width/image\_width、height/image\_height，取值范围由于经过归一化处理过因此为0~1。

#### 4.1.2 Yolov5模型建立

##### 4.1.2.1 具体模型选择

Yolov5模型针对不同训练数据集和处理精度，在核心网络结构的基础上主要提供了四个预训练方案，分别对应4个可用于训练的预训练权重，包含yolov5l， yolov5s，yolov5m ，yolov5x，因此我们用这4个预训练权重分别进行了数据集的训练，为并通过可视化的训练结果和参数指标对比来挑选出最优的权重。

##### 4.1.2.2 修改数据配置文件

原本的yolov5模型中数据配置文件名为coco128.yaml，其中定义了训练图片的路径，与验证集相同的图片，目标类别数，类名列表。其中共有nc（number of class）80种，类别名字在列表names=[ ]中，针对我们只检测人的情况，设置一个新的数据配置文件名为fchd.yaml，将nc设置为1，names=[ person ]。

##### 4.1.2.3 修改模型配置文件

从 ./models目录下选择要训练的模型，这里随机选一个yolov5s.yaml。将该模型配置文件下的nc（number of class）由原来的80设置为1，因为我们只检测人这一个类别，且数据集中只有person。

#### 4.1.3 Yolov5模型训练

##### 4.1.3.1 选择预训练权重

在实际开发中我们对4个预训练权重分别进行了训练，在这里我们仅以yolov5s.pt为例进行后续说明。

##### 4.1.3.2 确定训练模型参数

img-size（训练和测试数据集的图片尺寸(即分辨率)，默认640），设置为640。

batch-size （每次传输的数据个数），设置为16。

epochs (代表样本集合要被训练遍历几次),设置为300。

--cfg选择模型文件（models/yolo5s.yaml）。

--data选择数据集文件(data/fchd.yaml)。

--weights指定预训练权重文件（这里以yolov5s为例yolov5s.pt）。

##### 4.1.3.3 执行训练命令

Python train.py --img 640 --batch 16 --epochs 300 --data ../data/fchd.yaml --cfg models/yolov5s.yaml --weights weights/yolov5s.pt 。

##### 4.1.3.4 可视化分析

在训练过程中我们会通过模型训练前设定好的可视化接口，通过网页形式实时查看模型训练过程中的迭代改进，并实时查看各项训练指标是否达标，训练完成后，我们还会对各个与训练权重的训练指标进行对比，挑选出最适合我们这个学习过程的训练权重。

#### 4.1.4 Yolov5模型验证

##### 4.1.4.1 修改检测框参数

针对原有模型中标定框的参数不符合当前项目中目标密度高的特点，因此我们对其进行了适当改动。对于标定框，取消label标识，取消置信度，改颜色框为红色，除此之外将模型识别出的人数标定在整个图片右下角处，以便直观查看检测效果和大致精确度。

##### 4.1.4.2 模型预测

在模型训练完毕后，就会在weights文件夹下生成两个预训练的模型，分别是best.pt和last.pt。best.pt：保存的是中间一共比较好模型，last.pt：训练结束后保存的最后模型。

后续我们会对测试集整个做预测来查看检测效果，执行下列命令：python detect.py --weight weights/best.pt --source ../data/imgsets/test，即可得到识别后的图像，图片中所有预测到的人头都会被标上框，且图片右下角可显示图片中的人数。

### 4.2 数据传输模块程序设计

首先通过Python的包sshtunnel连接远程服务器并通过root用户和密码登录服务器，后续使用Python的包pymysql的connect方法，连接运行在服务器上的MySQL数据库并通过指定数据库用户密码登录，然后可通过pymysql里的其他增删改查方法去操作数据库，这里将图像识别模块得到的人数数据更新到原为空值的教室人数一列。

### 4.3 智能推荐模块程序设计

#### 4.3.1 空教室查询后端设计

在Linux服务器端搭载MySQL服务器，导入教务系统教室排课数据库，在服务器上部署JDK1.8环境，使用SSM(Spring Boot+Spring MVC+Mybatis)来搭建Restful API接口以便前端调用，包含查询某时段（校历上第几周周几第几节课）某教室是否有排课记录的接口。

#### 4.3.2 智能推荐方案算法设计

初步方案为：分为推荐和订阅，订阅即为用户订阅指定的教室后，在打开小程序后我们会优先展示订阅教室的基本信息包含教室、总座位数、实际占用的座位数等，并且会在当前非排课教室中综合拥挤度和当前位置、以及短期内是否有排课记录等进行排序，用户可最快速的找到适合自己的自习室；推荐即为根据用户历史使用系统的记录来为用户推荐潜在的用户未曾发现的适合自己的教室，在用户首次使用系统时我们会对用户做一个小的问卷调查，来通过一些问题来了解用户更偏爱教室的类型，后续我们每次会基于时间、地点、偏爱的教室类型以及是否选择最小程度地换教室来来为用户推荐出一个完整的方案而不仅仅是推荐一个教室。以上方案均通过小程序前端根据后端获取得到的基本数据基于一定程度的算法处理实现。

#### 4.3.3 小程序前端界面设计

根据以上模块的设计，可见，小程序前端除了登录和我的信息页面之外，主要需做两个页面，三个模块，一个页面做简单的自助空教室查询，一个页面做智能推荐包含订阅情况和方案推荐，方案推荐需要用户的一些手动参数指定。除此之外，用户初次使用系统时还会有一个问卷页面。

## 5. 数据库设计

### 5.1 数据访问频度和流量

该系统数据库部分主要为学生自助查询和系统推送方案使用，因此在大课下课期间和考试前后会有大量用户访问系统，频度和流量随着用户使用人数和当前课程安排有密切关系，预估后期投入使用后访问量每小时平均1000次。

### 5.2 数据库选型

关系型数据库MySQL。

### 5.3 数据安全性及保密设计

由于该系统主要使用数据库的部分仅用于查询，且更新操作仅用于数据传输模块的自动更新，因此数据安全性和保密在该系统中几乎无需设计。

### 5.4 数据字典设计

## 

图 5 数据字典