图片面部识别考勤系统的实验报告

一、需求分析

随着信息技术和人工智能的发展，传统的考勤方式逐渐暴露出许多问题，如效率低，人工成本高，易出错等。因此，本项目的目的即是开发一个基于面部识别技术的考勤系统。

1. 用户需求：

教师能够方便地注册学生，通用照片来检查考勤，统计考勤情况等。

2. 系统功能：

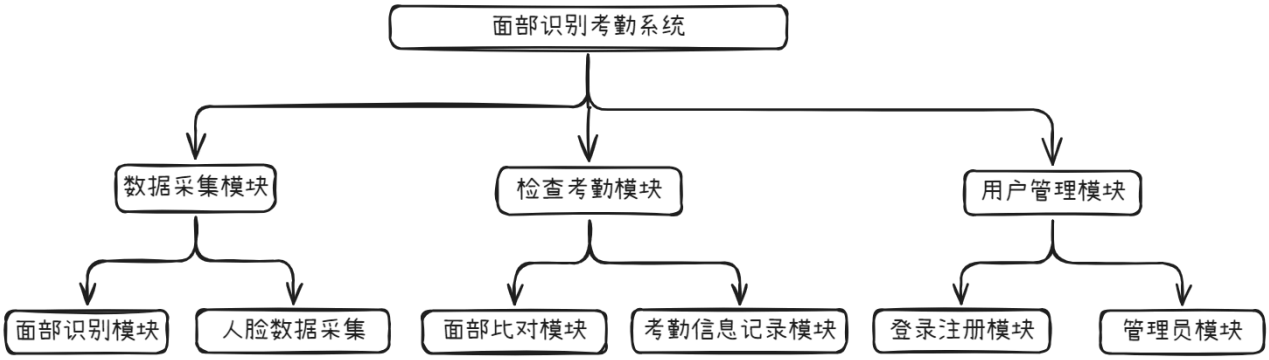
面部识别：系统可以通用照片来检测人脸，从而实现人脸的识别和编码，从而能够实现人脸的注册和匹配
数据管理：保存学生的人脸编码信息和姓名，同时记录学生每一次的考勤记录，方便最后的统计
UI界面：在前端给教师一个界面供教师完成照片上传，输入班级，考勤时间信息等，另外提供表格展示考勤的统计数据

3. 技术需求

安全性：保障学生的个人信息安全，防止未授权访问和数据泄露，在相应的视图都编写了认证功能，只有授权用户才能实现访问
高效性：具备较好的相应速度，能够在较短的时间内完成面部的识别和信息的传送，尽可能避免不必要的信息传送带来的额外开销
可拓展性：该系统具有良好的拓展性，可以加入考勤分析、通知推送、导出考勤结果统计等功能

二、系统设计

1. 系统的功能模块设计

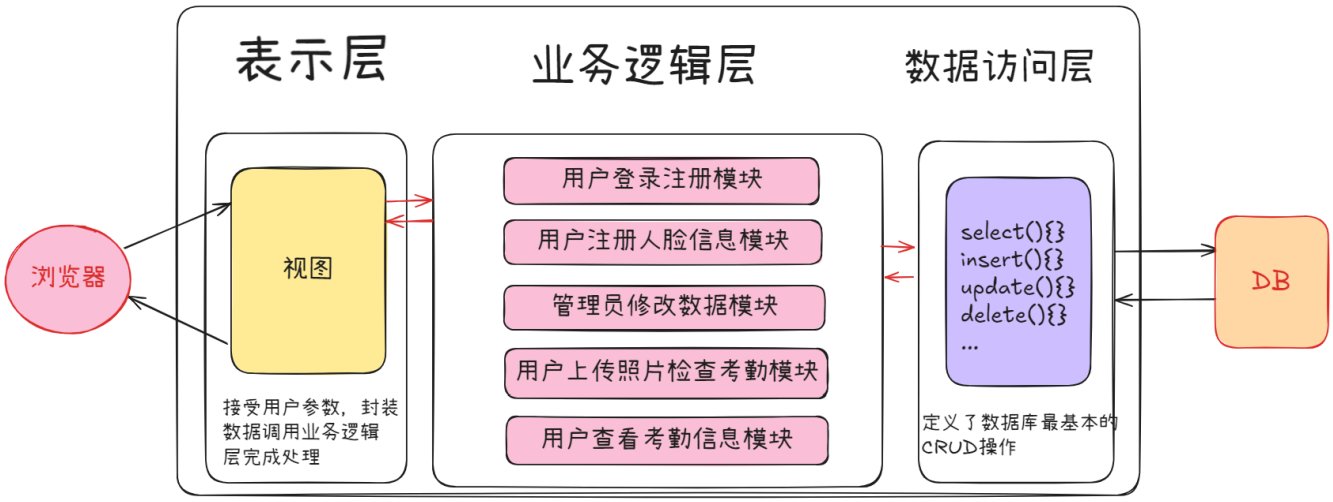


整体分为三个模块，分别是数据采集模块，检查考勤模块，以及用户管理模块。

- 在数据采集模块，包含通过图片实现面部识别的模块，以及对于人脸数据的编码和对应姓名的记录模块
- 在检查考勤模块中，包含考勤照片中人脸与注册人脸的面部信息比对模块和对于缺勤人员姓名，时间等信息的记录模块
- 用户管理模块中包含普通用户的登录注册模块以及管理员的管理模块

2. 系统整体的架构设计

2.1系统的架构风格选择



系统整体上采用了MVT的架构风格

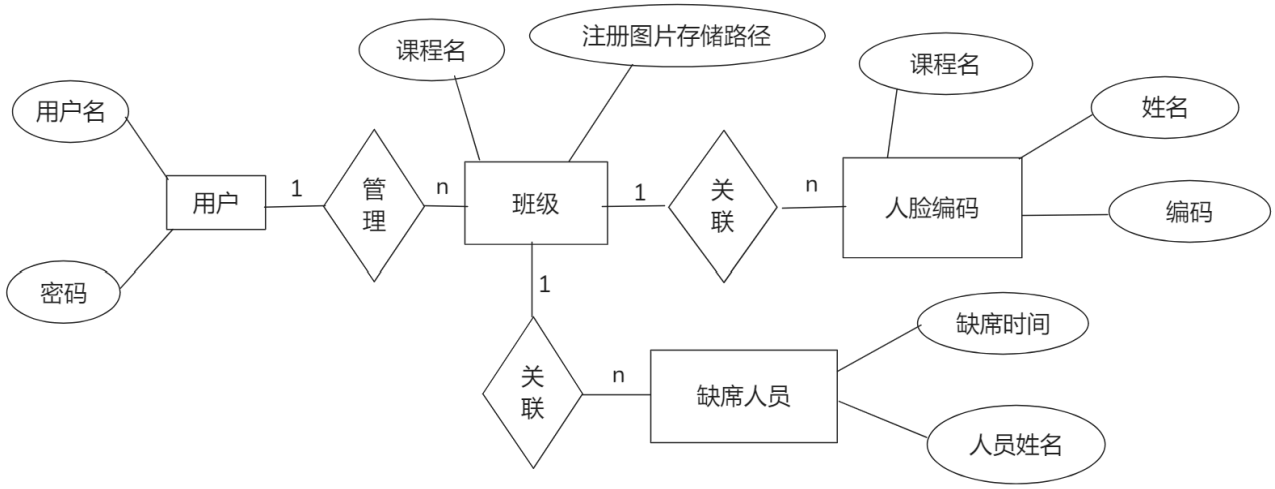
2.2系统架构风格的具体实现

- 前端的框架采用Vue.js
- 后端的框架采用Djiango

- 数据库采用MySQL

3. 数据库设计

3.1 数据库的概念结构设计（E-R图）



3.2 数据库的逻辑结构设计

3.2.1 表结构设计

（1） 用户表（User）

字段名	数据类型	约束	描述
user_id	INT	PRIMARY KEY, AUTO_INCREMENT	用户唯一标识
username	VARCHAR(50)	NOT NULL, UNIQUE	用户名
password	VARCHAR(255)	NOT NULL	密码

（2）班级表（Class）

字段名	数据类型	约束	描述
class_id	INT	PRIMARY KEY, AUTO_INCREMENT	班级唯一标识
class_name	VARCHAR(100)	NOT NULL	班级名称
user_id	INT	FOREIGN KEY (user_id) REFERENCES User(user_id)	管理员用户ID

（3）人脸编码表（FaceEncoding）

字段名	数据类型	约束	描述
face_encoding_id	INT	PRIMARY KEY, AUTO_INCREMENT	人脸编码唯一标识
class_id	INT	FOREIGN KEY (class_id) REFERENCES Class(class_id)	对应课程ID
name	VARCHAR(100)	NOT NULL	姓名
encoding	BLOB	NOT NULL	人脸编码数据

（4） 缺席人员表（AbsentStudents）

字段名	数据类型	约束	描述
absent_id	INT	PRIMARY KEY, AUTO_INCREMENT	缺席记录唯一标识
enrollment_id	INT	FOREIGN KEY (enrollment_id) REFERENCES Enrollment(enrollment_id)	学生与班级关系ID
absent_time	DATETIME	NOT NULL	缺席时间
student_name	VARCHAR(100)	NOT NULL	学生姓名

2. 关系设计

- **用户表和班级表之间**：一对多关系（一个用户可以管理多个班级）。
- **班级表和人脸编码表之间**：一对多关系（一个班级可以包含多个学生）。
- **班级表和缺席人员表之间**：一对多关系（一个班级可以有多个缺席记录）。

三、实验过程

1. 具体的搭建步骤

1.1 环境准备

1. 在本地创建一个Python的虚拟环境，避免包的版本依赖问题
2. 安装Djiang
3. 在终端激活虚拟环境

1.2 后端Djiango连接MySQL数据库

1. 在Djiango中安装Python的数据库适配器
2. 在settings文件中添加数据库配置部分
3. 运行数据库迁移，此举会在MySQL中创建Djiango需要的所有表

1.3 实现用户的登录注册应用

1. 在Djiango中安装REST framwork并添加到配置文件当中
2. 在项目中新建一个Application来负责用户的登录和注册模块
3. 在该Application对应的文件下是实现相关的逻辑
4. 设置Djiango使其可以跨域访问

1.4 业务逻辑的实现

1. 图片注册人脸信息
   大致流程：前端向后端发送照片，后端识别人脸后将位置信息发送给前端，前端通过此face_locations在前端依次展示图片中的人脸，获取人脸姓名等数据后发送给后端，后端通用face_locations来进行人脸的编码，最后将编码和姓名信息存入数据库
2. 图片实现考勤
   大致流程：前端发送照片和时间数据，后端识别人脸并进行编码，将此编码于数据库中的人脸信息进行比对，获得缺席的所有姓名信息，将此信息发给前端进行展示
3. 查看考勤信息
   大致流程：前端发送要查看的课程名和时间信息给后端，后端通过这两个信息来筛选符合条件的数据返回给前端展示，这里注意的是如果前端发送的某个条件为空，就返回忽略此条件的所有数据

2. 遇到的挑战和解决方案

1. 在人脸编码之后的比对部分，开始尝试直接计算通过编码向量的欧氏距离或者余弦距离来获得人脸的匹配关系，但是发现效果都不理想，尤其是当人脸有部分阴影，倾斜，或者有所遮挡时，结果的波动性很大。最后发现在face_recognition的包中一个专门用来比对人脸的compare_faces函数，尝试之后发现效果很好。
2. 在Djiango中，有自带的ORM模型，所以不需要自己写sql语句，只需要定义好相应的模型，然后在视图文件中相应的操作此模型就可以实现对于数据库的修改。在开发过程中，有一次我直接通过MySQL Workbench将一个table删除，然后程序就报错无法运行，即使删除代码中所有的migrations对应的文件，然后重新生成也无济于事。最后发现问题在于，需要删除数据库中migration table中对应删除表的内容，才可以正常运行。
3. 在开始使用face_recogniton包的时候，程序一直报错无法打开该包之下的一个.dat文件,尝试了使用管理员权限打开终端运行程序还是不行，以为是虚拟环境下对于.dat文件格式不支

持，更换Python的解释器还是不行，最后发现是因为face_recogniton包中对于中文路径没有办法解析，尝试更换文件目录之后正确运行

四、实验总结

首先一个人最终在规定的时间之内完成了这个项目，让我学习到了很多东西，这是我第一次使用Djiango框架，也是我第一次一个人连通前后端包括数据库，所以对于软件项目的架构和里面的一些细节有了更加深入的了解；另外也增长了问题的解决能力，因为不论是环境搭建还是代码调试，都会遇到各种各样的报错，从一开始的手足无措，等后来慢慢学会去沉下心来，遇到问题一点点排查，也包括更好地借助AI来辅助问题的分析与解决。

本次实验的由于是我一个人完成所有工作，所以时间稍微有些紧张，因此在前端页面的展示上，还有一些细节的部分没有来得及修饰，所以部分地方显得不是特别简洁和美观；在业务逻辑部分，也还有许多可以拓展补充的地方，例如现在只是可以根据时间和课程来筛选查看缺勤记录，但是后续可以增加更多的统计数据，例如一个班整个学期的缺勤记录，或者在某一个时间段内的缺勤记录统计，在比如还可以添加记录的导出功能，比如导出为Excel表格等形式。