

目 录

第一章 需求分析	3
第二章 概要设计	3
1. 功能模块	3
1.1 功能划分	3
1.2 功能描述	4
1.3 模块间接口	4
1.4 数据流图设计	5
2. 子系统划分	6
3. 运行环境	7
3.1 硬件运行环境	7
4. 人机界面:	7
4.1 设计原则	7
4.2 教师注册界面	7
4.3 教师登录界面	8
4.4 主界面	8
4.5 学生信息录入界面	8
4.6 学生管理模块界面	8
4.7 数据分析模块界面	8
第三章 详细设计	9
1. 总体业务流程	9
2. 子业务流程	9
2.1 用户注册流程	9
2.2 人脸录入流程	10
2.3 考勤签到流程	11
2.4 考勤管理流程	12
3. 界面设计	13
3.1. 设计原则	13
3.2 教师注册界面	13
3.3 教师登录界面	14
3.4 主界面	14
3.5 点名界面	15
3.6 学生信息录入界面	16
3.7 学生管理模块界面	17
3.8 数据分析模块界面	18
3.9 异常申报界面	20
4. 数据库设计	20
4.1 数据库逻辑设计	20
4.2 数据库表名设计	21
4.3 数据库表设计	21
4.4 数据描述	23

4.5 索引设计	26
4.6 安全设计	27
5. 关键算法	27
第四章 测试报告	28
1. 系统测试	28
1.1 测试环境	28
1.2 主要性能测试内容	28
1.3 数据测试	28
2. 检查结果分析	30
3. 测试结果:	30
第五章 安装及使用	31
1. 安装环境要求:	31
2. 安装主要流程:	31
第六章 项目总结	35
参考文献	35

第一章 需求分析

本项目旨在为学校提供一种智能化的考勤管理方案。考勤是教育过程中非常重要的一环，能够有效地提升学校的管理水平和教育质量。然而，传统的考勤方式存在着很多问题，如考勤效率低下、误差率高、容易被舞弊等。为了解决这些问题，我们决定开发一款基于视觉神经网络的智能考勤签到管家。

在市场上，虽然存在着一些竞品，如传统的人工考勤和二维码考勤等，但这些方式都存在着各种问题，无法满足学校对高效、准确、安全的考勤需求。因此，我们开发的这款智能考勤签到管家，能够通过视觉神经网络技术实现高效、准确、安全的考勤。

本作品的主要面向对象是学校和教师。学校可以使用此系统来管理学生的考勤情况，并通过统计和分析数据来提高教学质量；该系统主要功能包括学生信息查询、考勤管理、数据统计和分析等。其中考勤管理能够通过摄像头识别学生的面部特征，对学生进行自动考勤，同时还能记录学生的考勤状态、考勤时间等信息。

通过这些功能，教师和学校可以更好地管理学生的学习情况，提高教学质量。

总的来说，本作品的主要目标是提高学校管理水平和教学质量，为学生和教师提供更加高效、准确、安全的考勤和管理方案。

第二章 概要设计

1. 功能模块

1.1 功能划分

根据需求分析，我们可以将该项目分解为几个主要的功能模块，本系统的功能模块划分图如图所示。

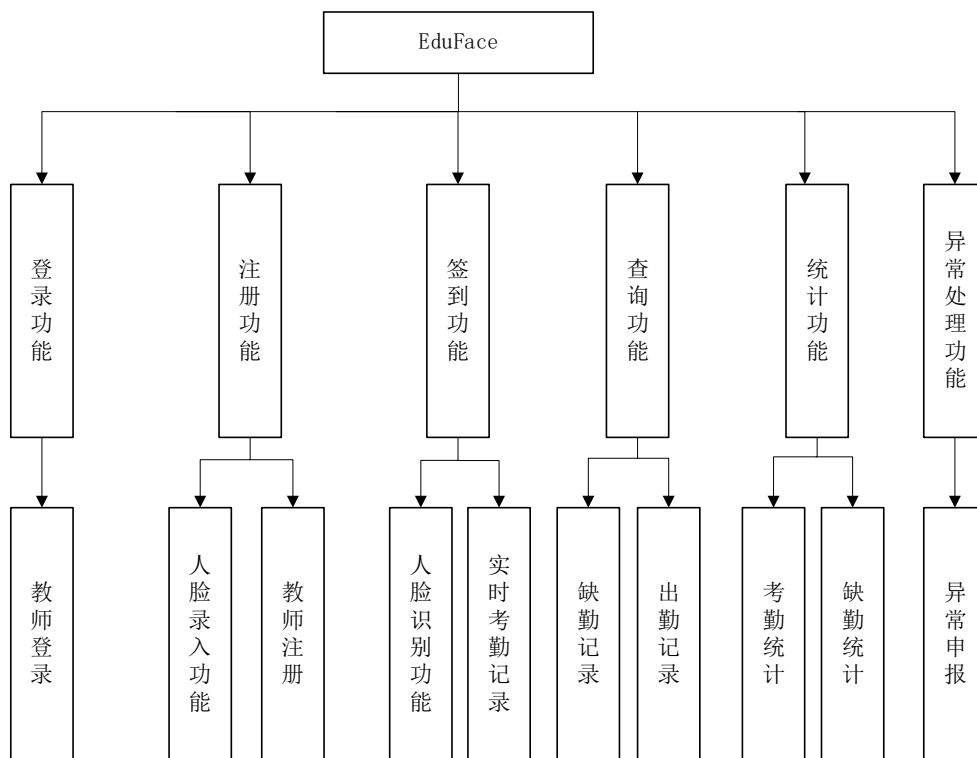


图 1.1 系统功能模块划分图

1.2. 功能描述

- (1) 签到功能：系统通过人脸识别算法对图像进行处理，提取人脸特征，并与数据库中存储的学生信息进行匹配，完成人脸识别。
- (2) 人脸录入功能：教师通过此功能对所授课班级学生进行人脸录入。
- (3) 查询功能：教师可以筛选、排序等功能对数据进行进一步处理。
- (4) 统计功能：系统将会对考勤数据进行分析，生成报表以及各种图表（如班级签到图表，学院签到图表），进行数据分析。
- (5) 异常处理功能：如果发现个人信息签到信息有误等，可以进行异常申报。
- (6) 访问校园官网：更具后台设定，用户可以快速访问学校官网、

1.3 模块间接口

本系统的模块间接口如下表所示。

表 1 模块间接口表

模块名称	被调用模块名称	接口名称	描述
教师登录模块	-	MainInterface()	验证教师登录凭据

教师注册模块	-	RegisterInterface_teacher()	注册教师账号
人脸检测模块	特征提取模块, 人脸识别模块	FaceDetection()	检测图像中的人脸
特征提取模块	人脸识别模块	FaceRecognize()	提取人脸特征
人脸识别模块	-	FaceRecognize()	识别人脸
考勤记录模块	-	record_attendance()	记录学生考勤情况
学生考勤查询模块	考勤记录模块	get_attendance_record()	获取学生考勤记录
学生信息录入模块	-	add_student()	添加学生信息
考勤数据统计模块	考勤记录模块	AttendanceChart()	计算考勤统计数据
学生出勤率分析模块	考勤记录模块	calculate_student_attendance_rate()	计算学生出勤率
学生管理模块	-	Student_Management()	删除学生信息

1.4 数据流图设计

1.4.1 顶层数据流图

本系统的顶层数据流图如下图所示。

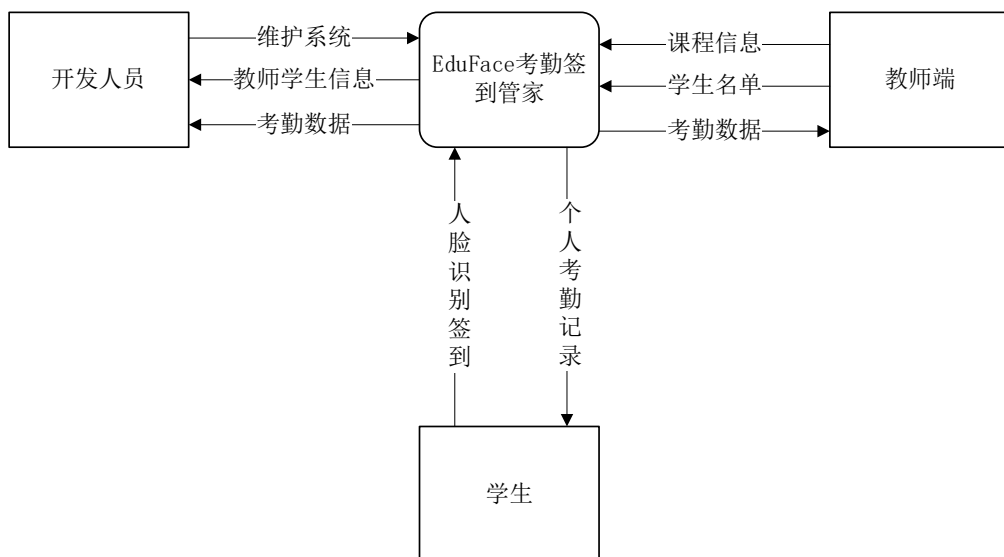


图 1.4.1 顶层数据流图

1.4.2 功能级数据流图

本系统功能级数据流图如图所示。

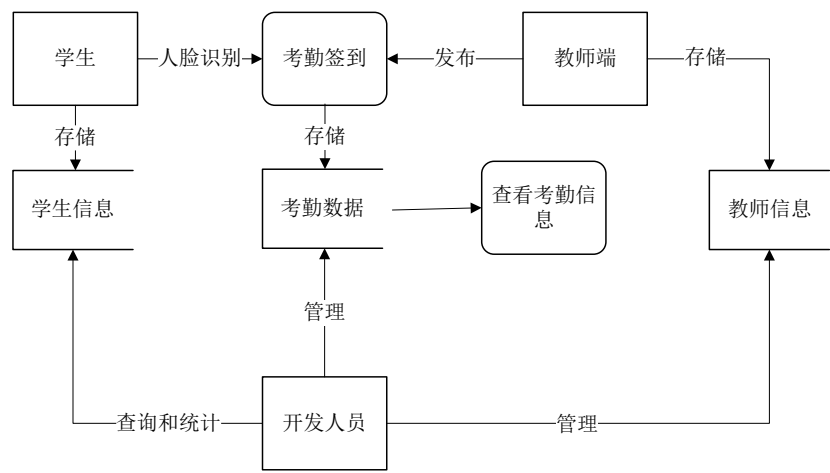


图 1. 4. 2 功能级数据流图

2. 子系统划分

子系统划分如下表所示。

表 2 子系统划分表

子系统编号	子系名称	子系统功能描述
1	个人信息管理系统	1) 注册、登陆功能的实现 2) 个人信息的维护和更新
2	人脸采集管理系统	1) 对采集到的人脸图像进行预处理，人脸检测、姿态校正、光照调整。 2) 将处理后的人脸图像存储到数据库中，以备后续的人脸识别和考勤记录。
3	人脸识别管理系统	1) 初始化人脸识别算法库和人脸特征提取器，准备开始人脸识别。 2) 对采集到的人脸图像进行人脸特征提取，生成人脸特征向量。 3) 对数据库中存储的人脸特征向量进行匹配，计算相似度，并确定匹配结果。
4	考勤记录管理系统	1) 初始化考勤记录器和考勤统计器，准备开始记录考勤信息。 2) 接收人脸识别子系统返回的匹配结果，并根据匹配结果生成考勤记录。 3) 统计学生的考勤数据，并计算出勤率、迟到、早退和旷课等数据。
5	报表管理系统	1) 对数据库中存储的考勤数据进行分析 and 统计，计算出各种考勤指标。 2) 根据用户的需求，生成各种报表和分析结果，并进行可视化展示。 3) 将生成的报表和分析结果导出，以供用户进行进一步处理和分析。
6	系统管理子系统	1) 初始化系统管理器和权限控制器，准备开始管理系统。

		2) 对系统进行配置，包括数据库连接、人脸识别算法库选择、系统参数设置等。 3) 对系统进行权限管理，包括用户管理、角色管理、权限分配等。 4) 记录系统的操作日志，并对系统进行监控和维护。
--	--	---

3. 运行环境

3.1. 硬件运行环境

- 1) 中央处理器：作为整个系统的核心，承担数据存储、算法计算、人脸匹配等主要功能。
- 2) 摄像头：用于采集人脸图像。采用高分辨率的摄像头，并根据实际情况选择合适的拍摄角度和距离，以获得清晰的人脸图像。
- 3) 网络设备：包括交换机、路由器等，摄像头及其他设备，构建整个系统的局域网。
- 4) 显示器：用于显示考勤结果、系统状态等信息。
- 5) 电源设备：包括服务器、摄像头、显示器等设备的电源供应。

3.2. 软件运行环境

1. 操作系统：Windows10 以上系统。
2. 数据库管理系统：用于存储考勤数据和人脸特征等信息，选择 MySQL。
3. 图像处理库：用于对采集的人脸图像进行处理和优化，以提高人脸识别的准确率，使用 OpenCV。

4. 人机界面：

4.1 设计原则

本系统的人机界面设计遵循以下原则：

- 简洁：界面尽可能简单直观，减少用户思考成本，提高使用效率。
- 一致性：不同模块的界面设计应该保持一致，避免用户需重新适应不同的界面风格。
- 易学易用：用户可以轻松地学会和使用系统。

4.2 教师注册界面

- 姓名输入框：教师在此输入自己的姓名。

- 密码输入框：教师在此输入自己的密码。
- 工号输入框：教师在此输入自己的工号。
- 性别选择：教师点击单选按钮选择自己的性别。
- 注册并登录按钮：教师单击此按钮以完成注册并登录。

4.3 教师登录界面

- 姓名输入框：教师在此输入自己的姓名。
- 密码输入框：教师在此输入自己的密码。
- 登录按钮：教师单击此按钮登录。

4.4 主界面

- 一键点名按钮：教师点击此按钮发布签到，点击后可选择课程、班级进行签到。
- 学生录入按钮：教师点击此按钮进入学生录入界面。
- 考勤记录按钮：教师点击此按钮进入查看考勤记录表界面。
- 学生管理按钮：教师点击此按钮查看所有学生信息。
- 校园官网按钮：教师点击此按钮进入校园官网。
- 异常申报按钮：教师点击此按钮处理申诉。
- 数据分析按钮：教师点击此按钮查看各班级出勤率等信息。

4.5 学生信息录入界面

- 学生信息录入包含学生学号、姓名、班级、性别的输入框，以及录入人脸按钮；

4.6 学生管理模块界面

- 学生信息列表，用于显示所有学生的基本信息，例如学生学号、姓名、性别等。
- 搜索框，用于根据关键字搜索学生信息。

4.7 数据分析模块界面

- 一个选项菜单，用于选择不同的范围进行出勤率分析。

- 一个出勤率统计图表，显示学生出勤率的数据和趋势。

界面采用图表和文字相结合的形式，直观地显示出勤率的情况，并且方便用户对某些课程出勤率的分析 and 比较。

第三章 详细设计

1. 总体业务流程

本系统总体业务流程图如下：

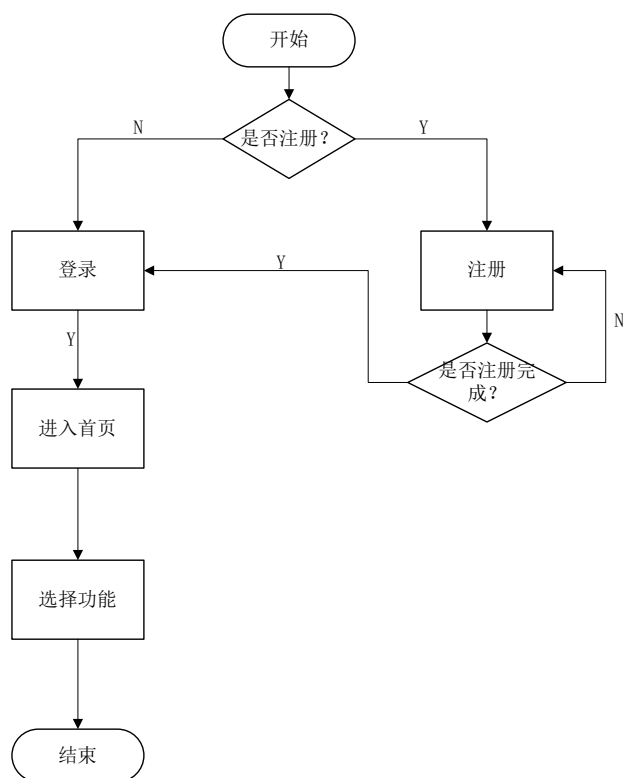


图 1 总体业务流程图

2. 子业务流程

2.1 用户注册流程

用户注册流程描述了用户如何注册系统账号并登录系统，如下图所示。

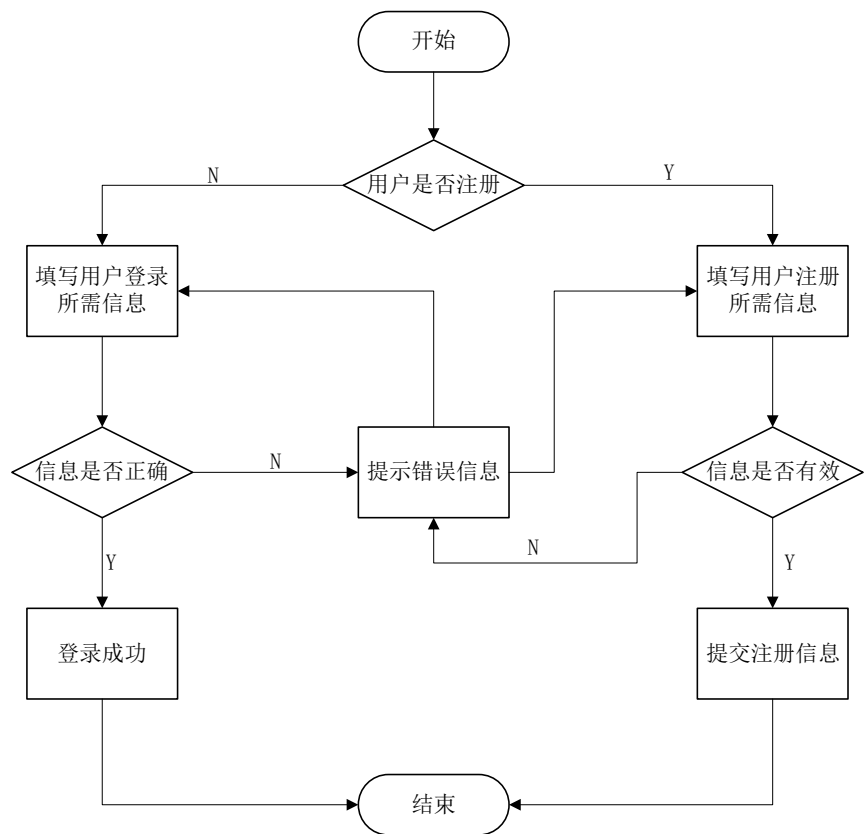


图 2.1 用户注册流程图

2.2 人脸录入流程

人脸录入流程描述了用户如何在系统中录入自己的人脸信息，如下图所示。

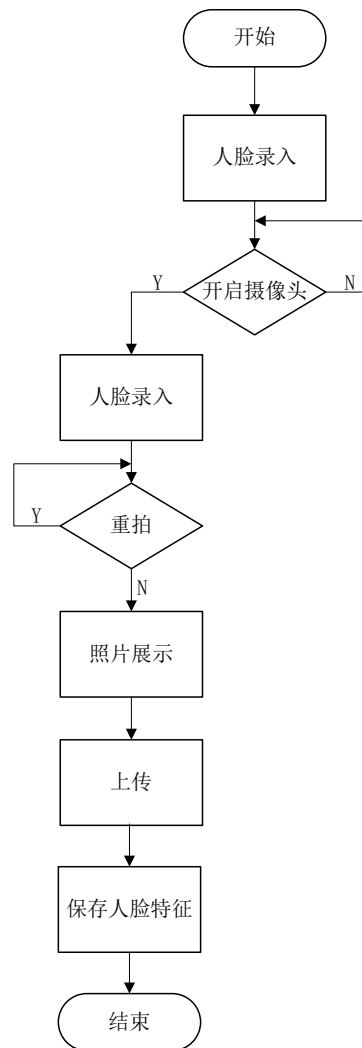


图 2.2 人脸录入流程图

2.3 考勤签到流程

考勤签到流程描述了用户如何进行考勤打卡操作，并如何通过人脸识别技术进行打卡验证。如下图所示。

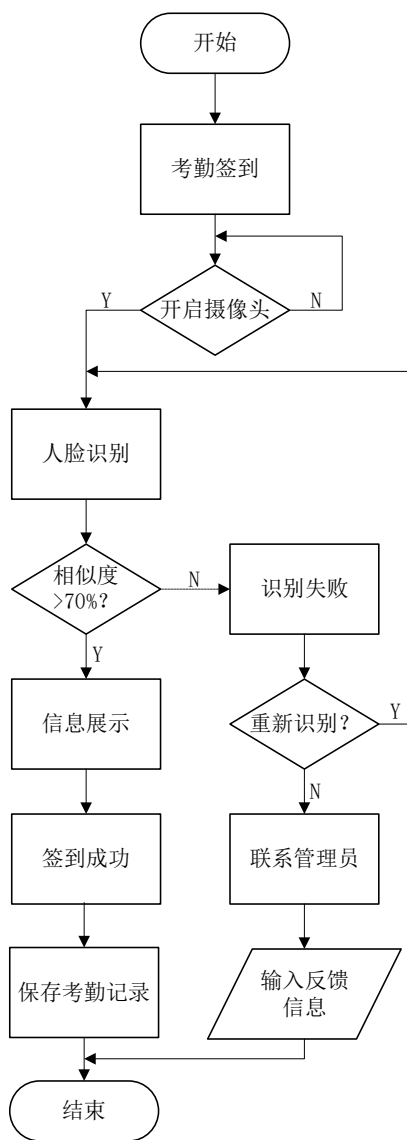


图 2.3 考勤签到流程图

2.4 考勤管理流程

考勤管理流程描述了教师如何对考勤数据进行管理，包括数据查询、审核、修改等操作。

下图为考勤数据查询功能的流程图。

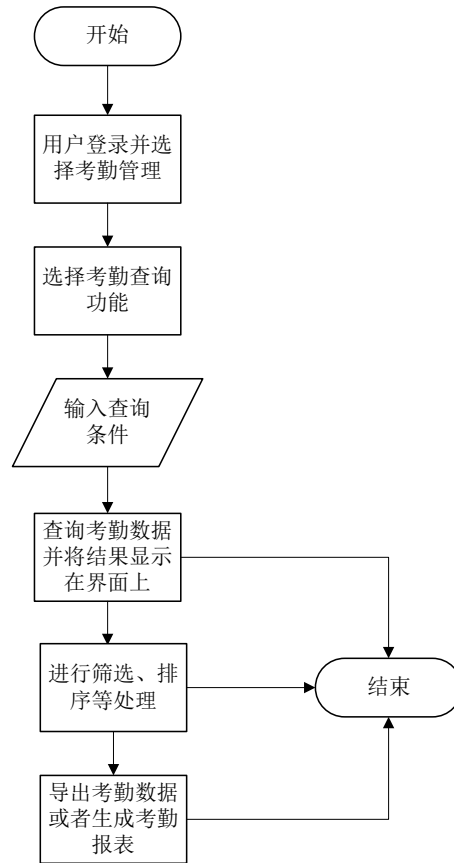


图 2.4 考勤数据查询流程图

3. 界面设计

3.1. 设计原则

本系统的界面设计遵循以下原则：

- **简洁明了：**界面尽可能简单直观，减少用户思考成本，提高使用效率。
- **一致性：**不同模块的界面设计应该保持一致，避免用户需要重新适应不同的界面风格。
- **易学易用：**用户可以轻松地学会和使用系统。

3.2 教师注册界面

该界面用于教师进行注册，如下图所示：



图 3.2 教师注册界面

3.3 教师登录界面

该界面用于教师进行登录，如下图所示：



图 3.3 教师登录界面

3.4 主界面

该界面为系统的主界面，如下图所示：



图 3.4 系统主界面

3.5 点名界面

在教师点击一键点名按钮后，将弹出以下界面供教师进行考勤：

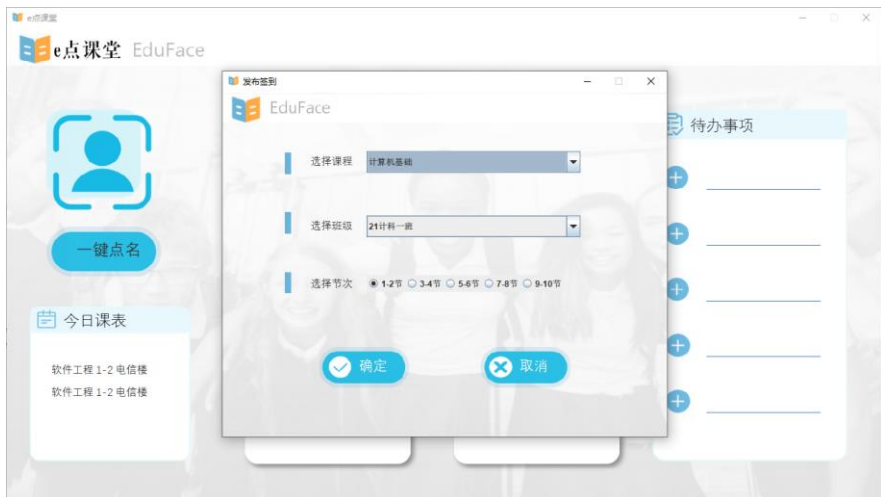


图 3.5.1 一键点名界面

在选择课程、班级和节次后，点击确定，将弹出考勤界面，如下图所示：

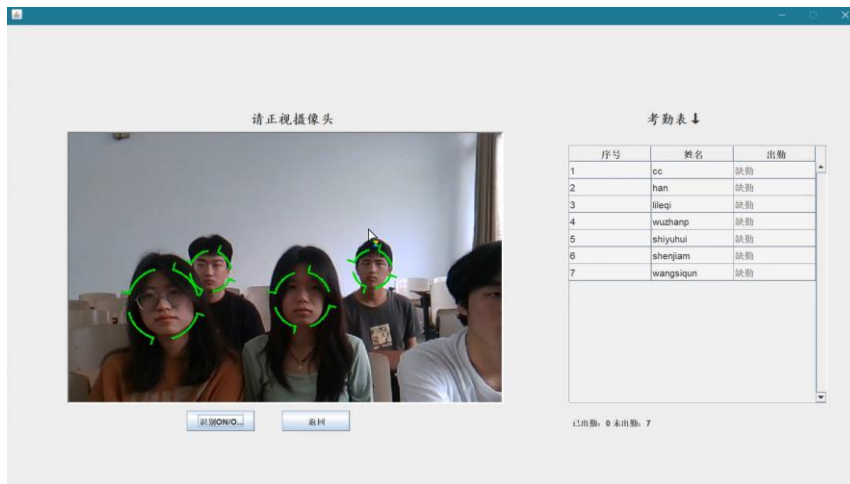


图 3.5.2 考勤界面

点击识别后:

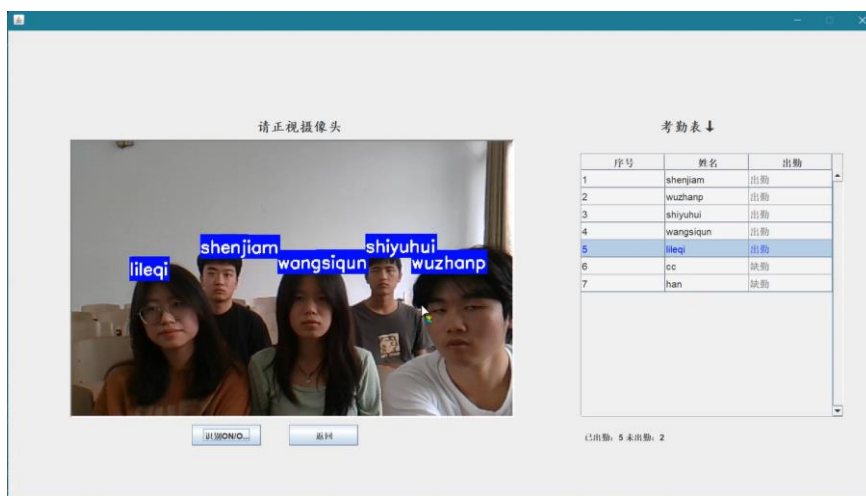


图 3.5.3 识别界面

3.6 学生信息录入界面

该界面用于老师录入学生相关信息，如下图所示：



图 3. 6. 1 学生信息录入界面

3. 7 学生管理模块界面

该界面用于老师进行学生管理，可导出班级花名册或导入花名册，还可以查看班级学生列表：



图 3. 7. 1 学生管理进入界面

学生列表界面：



图 3. 7. 2 学生列表界面

导入导出界面：

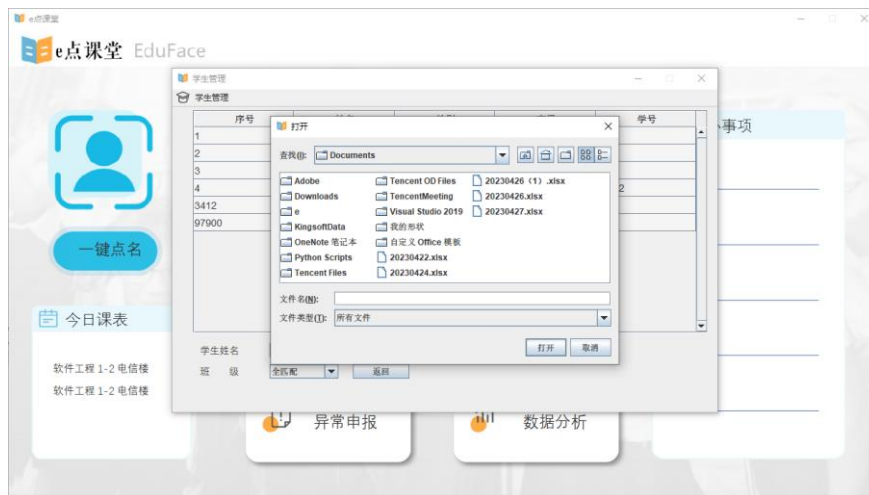


图 3.7.3 导入导出界面

3.8 数据分析模块界面

该界面是教师进行数据分析的主要界面，总体出勤是根据课程生成的考勤柱状图表，如下图所示。

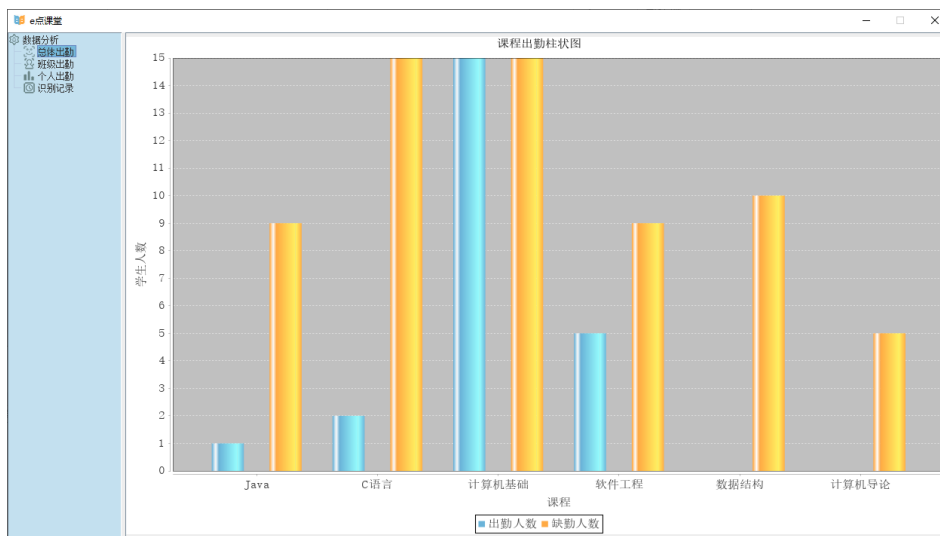


图 3.8.1 数据分析界面-总体出勤

班级出勤是通过考勤记录表统计每个班级的出勤率与缺勤率，生成的饼状图，如下图所示。

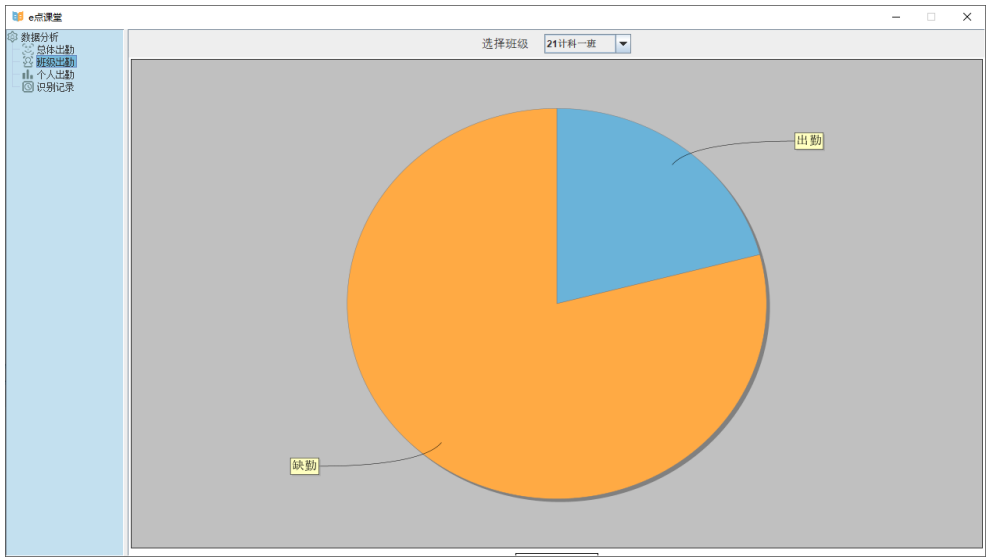


图 3.8.2 数据分析界面-班级出勤

个人出勤则是对每个学生的出勤次数与缺勤次数做统计，如下图所示。

姓名	班级	出勤次数	缺勤次数
hw	21计科一班	0	37
chen	21计科一班	0	37
zhang	21计科一班	1	21
Du	21广电二班	1	0
cc11	21计科一班	7	9
cc	21计科二班	7	12
han	21计科一班	21	16

图 3.8.3 数据分析界面-个人出勤

识别记录则是每次考勤成功的学生，在数据库中都会留下一张识别图片，如下图所示。

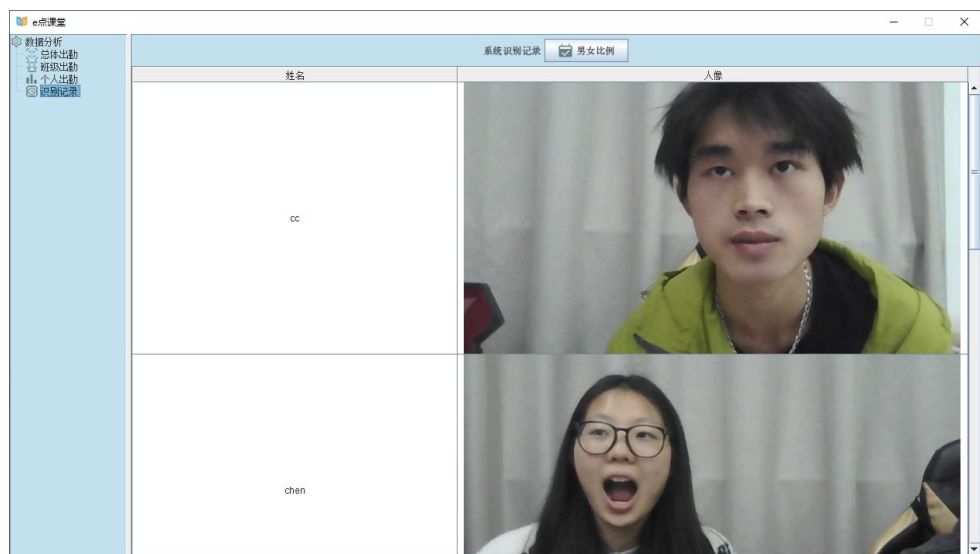


图 3.8.4 数据分析界面-识别记录

3.9 异常申报界面

此界面可供教师处理申诉，如下图所示：



图 3.9.1 异常申报界面

4. 数据库设计

4.1 数据库逻辑设计

1. 学生（学号，学生姓名，性别，班号）
2. 教师（工号，姓名，性别，教授班级号，所授课程号）

3. 课程（课程 ID, 课程名称, 上课周数, 星期几, 节次, 班级 ID, 教室, 教师 ID, 课程所属系名）
4. 班级（班级号, 学生学号, 学生姓名, 班主任）
5. 考勤信息（考勤编号, 学号, 工号, 考勤时间, 课程 ID, 考勤状态）
6. 教师-班级（班级 ID, 工号）
7. 班级-课程（课程 ID, 班级 ID）

4.2 数据库表名设计

数据库表名设计如下表所示：

表 4.2.1 数据库表名设计

编号	表名	表功能说明
1	Student	学生表
2	Teacher	教师表
3	Course	课程表
4	Class	班级表
5	AttendanceInfo	考勤信息表
6	classroom_class	教室-班级表
7	class_course	班级-课程表

4.3 数据库表设计

4.3.1 Student 表设计见下表。

表 4.3.1 Student 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	class_no	班号	INT	否	外键
2	student_id	学生 ID	INT	否	主键
3	sname	学生姓名	varchar(50)	否	
4	class_name	班级名称	varchar(100)	否	
5	sex	学生性别	varchar(2)	否	

4.3.2 Teacher 表设计见下表。

表 4.3.2 Teacher 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
----	-------	-------	------	--------	-------

1	teacher_id	教师 ID	INT	否	主键
2	tno	工号	INT	否	
3	tname	姓名	varchar(50)	否	
4	sex	性别	varchar(2)	否	
5	class_id	课号	INT	否	外键
6	password	密码	varchar(100)	否	

4.3.3 AttendanceInfo 表设计见下表。

表 4.3.3 AttendanceInfo 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	attendance_id	考勤 ID	INT	否	主键
2	course_id	课程 ID	INT	否	外键
3	student_id	学生 ID	INT	否	外键
4	teacher_id	教师 ID	INT	否	
5	attendance_time	考勤时间	DATETIME	否	
6	attendance_state	考勤状态	varchar(10)	否	

4.3.4 Course 表设计见下表。

表 4.3.4 Course 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	course_id	课程 ID	INT	否	主键
2	course_name	课程名称	varchar(100)	否	
3	week	上课周数	INT	否	
4	day_of_week	星期几	INT	否	
5	class_time	节次	INT	否	
6	class_id	班级 ID	varchar(100)	否	外键
7	Classroom	教室	varchar(100)	否	
8	teacher_id	教师 ID	INT	否	外键
9	department	课程所属系名	varchar(100)	否	

4.3.5 Class 表设计见下表。

表 4.3.5 Class 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	class_id	班级 ID	INT	否	主键
2	class_name	班级名称	varchar(100)	否	
3	department	所属系别	varchar(100)	否	
4	major	所属专业	varchar(100)	否	
5	class_count	班级人数	INT	否	
6	teacher_id	老师 ID	INT	否	外键

4.3.6 teacher_class 表设计见下表。

表 4.3.6 teacher_class 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	teacher_id	工号	varchar(100)	否	主键/外键
2	class_id	班级 ID	INT	否	主键/外键

4.3.7 class_course 表设计见下表。

表 4.3.7 class_course 表

序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	course_id	课程 ID	INT	否	主键/外键
2	class_id	班级 ID	INT	否	主键/外键

4.4 数据描述

4.4.1 实体关系

考虑本系统的基本功能和对象，我们总结了以下实体：

1. 学生（Student）：包括学生 id、姓名、性别、年龄、所属班级属性。
2. 教师（Teacher）：包括教师 id、姓名、性别、密码、课号属性。
3. 考勤记录（AttendanceRecord）：包括考勤编号、考勤日期、上课时间、课程 id、教师 id、学生 id、考勤状态等属性。
4. 课程（Course）：包括课程 id、课程名称、上课周数、星期几、节次、班级 id、教师 id、所属系别等属性。
5. 班级（Class）：包括班级 id、班级名称、所属系别、教师 id、教室 id 等属性。
6. 关联（class_course）：包括课程 id，班级 id。
7. 拥有（teacher_class）：包括教师工号，班级 id。

4.4.2 实体图

1. 学生实体如下图所示：

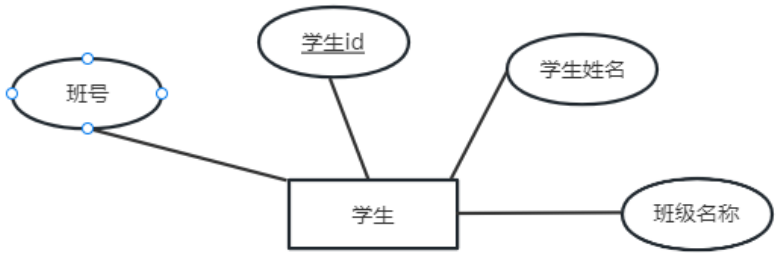


图 4.4.2.1 学生实体图

2. 教师实体如下图所示：

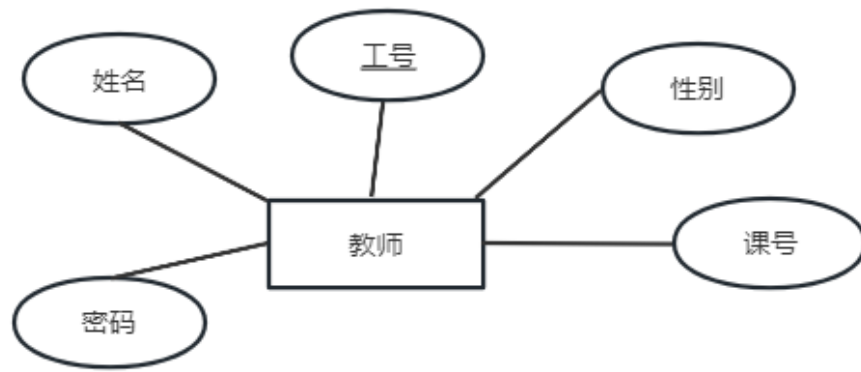


图 4. 4. 2. 2 教师实体图

3. 课程实体如下图所示：



图 4. 4. 2. 3 课程实体图

4. 班级实体如下图所示：

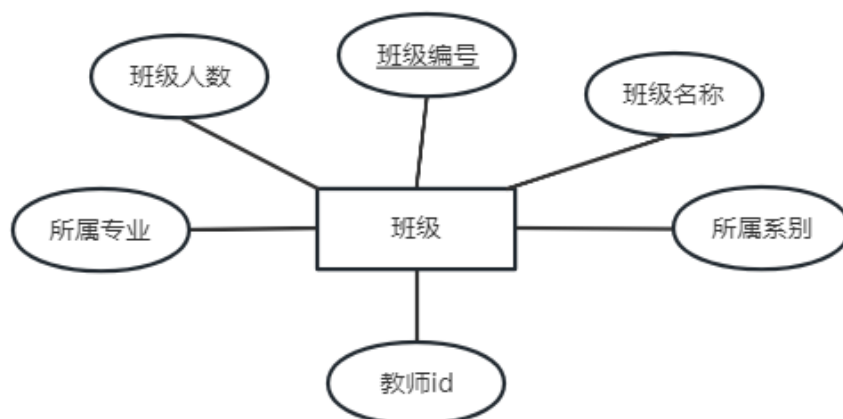


图 4.4.2.4 班级实体图

5. 考勤实体如下图所示：

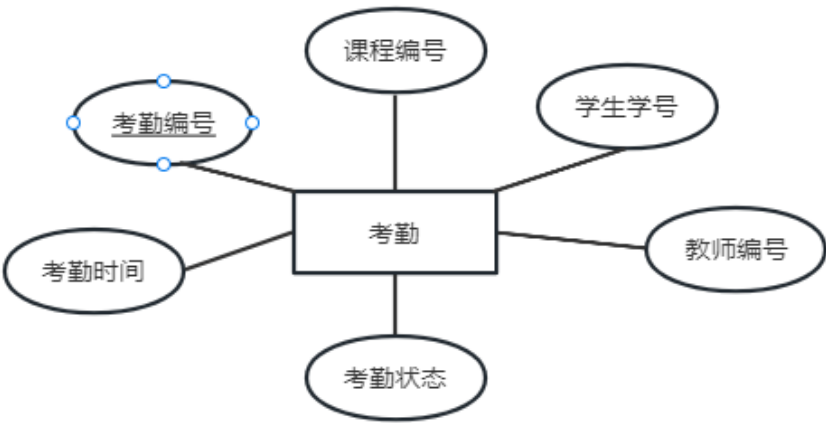


图 4.4.2.5 考勤实体图

6. 关联实体如下图所示：

在数据库表设计中，关联实体对应的表为 class_course 表：

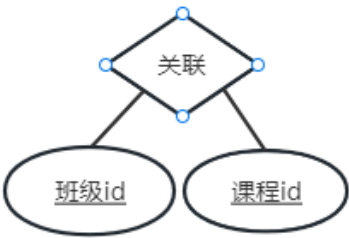


图 4.4.2.6 关联实体图

7. 拥有实体如下图所示：

在数据库表设计中，关联实体对应的表为 teacher_class 表：

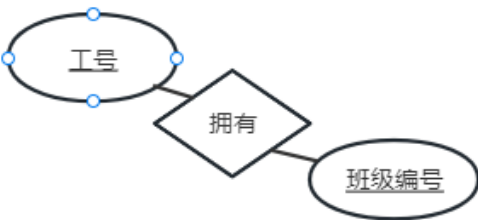


图 4.4.2.6 拥有实体图

8. 实体关系如下图所示：

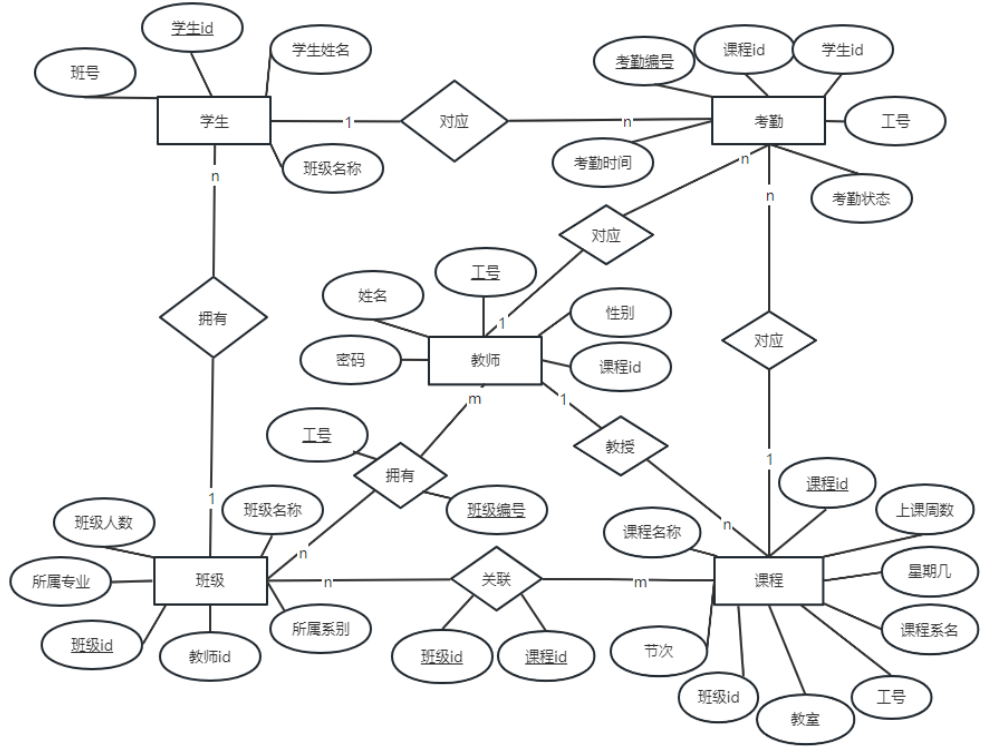


图 4.4.2.7 关联实体图

4.5 索引设计

索引的设计需要考虑到实际查询的情况，以及对性能的影响。以下是对数据库的索引设计：

学生表（student）：

student_id：主键，自动创建了唯一索引，不需要再创建额外索引

class_id：学生所属班级的外键，创建索引以优化班级相关查询

教师表（teacher）：

teacher_id：主键，自动创建了唯一索引，不需要再创建额外索引

课程表（course）：

course_id：主键，自动创建了唯一索引，不需要再创建额外索引

teacher_id：课程教师的外键，创建索引以优化教师相关查询

考勤记录表（attendance）：

attendance_id：主键，自动创建了唯一索引，不需要再创建额外索引

student_id：考勤记录对应学生的外键，创建索引以优化学生相关查询

course_id: 考勤记录对应课程的外键, 创建索引以优化课程相关查询

班级表 (class):

class_id: 主键, 自动创建了唯一索引, 不需要再创建额外索引

教师班级表 (teacher_class):

teacher_id: 班级所使用的教室的外键, 创建索引以优化教室相关查询

class_id: 教室所使用的班级的外键, 创建索引以优化班级相关查询

课程班级表 (class_course):

class_id: 课程所属班级的外键, 创建索引以优化班级相关查询

course_id: 班级所上的课程的外键, 创建索引以优化课程相关查询

4.6 安全设计

1. **访问控制:** 限制对数据库的访问权限, 只允许授权用户和应用程序访问数据库, 并且使用最小权限原则, 即只授予用户和应用程序必要的权限, 避免赋予过高的权限。
2. **身份验证和授权:** 使用强大的身份验证和授权机制, 例如用户名和密码、双因素认证等, 确保只有授权用户能够访问数据库。
3. **数据加密:** 对数据库中的敏感数据进行加密, 包括数据传输和存储过程中的加密, 以保护数据的机密性和完整性。
4. **审计和监控:** 设置数据库审计和监控机制, 记录数据库的访问和操作日志, 定期审查数据库日志, 检测和应对潜在的安全威胁。
5. **更新和维护:** 定期更新数据库软件和补丁, 确保数据库系统处于最新的安全状态, 并且采取措施防止数据库服务器受到已知的安全漏洞和攻击的影响。
6. **数据备份和恢复:** 定期进行数据库备份, 并将备份数据存储在安全的位置, 以便在数据丢失或数据库损坏时能够快速恢复数据。

5. 关键算法

在考勤过程中, 首先需要进行人脸识别, 即对学生进行人脸验证和识别。人脸识别是一种计算机视觉技术, 主要涉及以下几个关键技术:

1. **人脸检测:** 通过图像处理技术, 检测出图像中的人脸位置和大小。人脸检测是人脸识别的第一步, 精准的人脸检测能够提高后续的识别准确率。
2. **特征提取:** 通过人脸检测得到的人脸图像, 提取出对识别有用的特征, 例如面部轮廓、

眼睛、嘴巴等特征。

- 3. **特征匹配:** 将提取出的人脸特征与已有的人脸特征库中的特征进行比对, 寻找最佳匹配。
特征匹配算法需要考虑到误差和容错率, 以保证识别准确性。
- 4. **人脸识别模型训练:** 为了提高人脸识别的准确率, 需要使用大量的人脸图像数据对人脸识别模型进行训练。
- 5. **数据库管理:** 存储和管理人脸图像、特征以及识别结果等信息。数据库需要具备高效的存储和查询能力, 以应对大规模的人脸数据量。

在实现考勤过程中, 以上关键技术需要结合使用, 同时还需要考虑一些实际问题, 例如光照、姿态、遮挡等对识别准确率的影响。

第四章 测试报告

1. 系统测试

1.1 测试环境

操作系统	IE情况	杀毒软件	是否测试	备注
Windows-2008-Scrvcn-CN-32	IE6.0	无	×	无
Windows-7SP4-CN-32	IE6.0	无	√	正常
Windows-10-FT-32	IE6.0	无	√	正常
Windows 11-CN-64	IE7.0	无	√	正常
macOS 11	IE7.0	无	√	正常
Mac OS X 10.11	IE6.0	无	×	无
Ubuntu Linux	IE7.0	无	√	正常
Gentoo Linux	IE6.0	无	×	无

1.2 主要性能测试内容

测试项	测试方法	测试工具	备注
编译运行调试	性能测试	IntelliJ IDEA 2021.2.1	正常
基本功能	性能测试	IntelliJ IDEA 2021.2.1	正常
响应时间	性能测试	IntelliJ IDEA 2021.2.1	正常

1.3 数据测试

1.3.1 用户登录模块

测试账号密码输入是否正确，测试输入错误的账号密码，是否提示错误信息



图 1.3.1 登录失败

1.3.2 人脸检测模块

测试人脸检测模块、特征提取模块、识别模块的准确性和响应速度。



图 1.3.2 测试人脸检测模块

1.3.3 考勤管理模块

考勤管理模块测试导入导出功能安全性、准确性、响应速度

改了逻辑代码，以及 UI 设计。

技术指标：

运行速度：系统的响应速度在合理的范围内，可以满足用户的基本需求。

安全性：系统采用了密码登录和人脸识别的双重认证方式，保障了用户的安全性。

扩展性：系统的模块化设计和松耦合架构，使得系统可以方便地进行扩展和升级。

部署方便性：系统的安装和部署过程简单易行，不需要复杂的配置和环境。

可用性：系统的用户界面友好易用，可以满足用户的基本需求，并且在测试中表现良好。

第五章 安装及使用

1. 安装环境要求：

1.1 系统要求

- 操作系统：Windows 10 或更高版本
- 内存：至少 2GB RAM
- 存储空间：至少 10GB 可用空间

1.2 安装 Java 17 运行时环境（JRE）

- 下载并安装 Java 17 运行时环境（JRE）
- 设置系统环境变量 JAVA_HOME，指向 Java 的安装目录。

1.3 配置 OpenCV 运行时环境

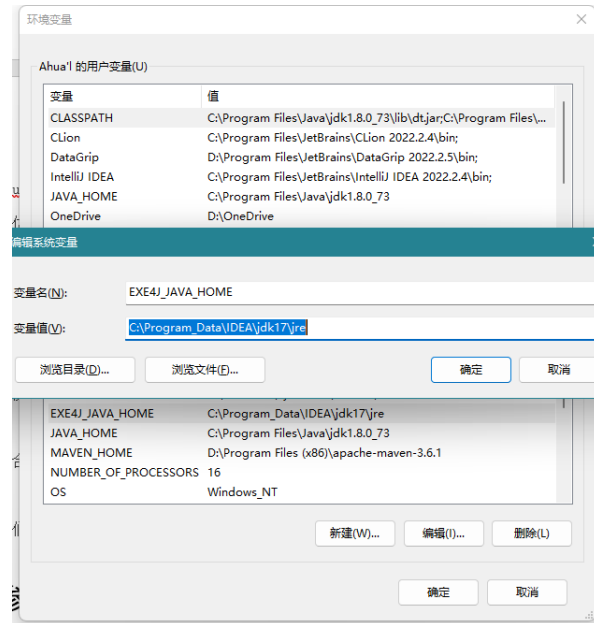
- 下载并安装 OpenCV_3_14v 的公开库。
- 设置系统环境变量 OpenCV_HOME，指向 OpenCV_x6w4 文件的安装目录。

1.4 安装 MySQL 数据库

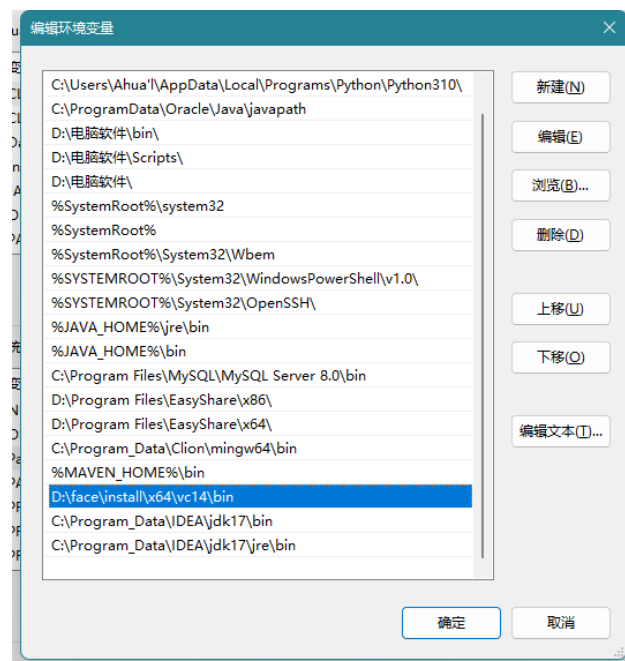
- 下载并安装 MySQL 数据库，使用 MySQL 8.0 版本。
- 启动 MySQL 服务。

2. 安装主要流程：

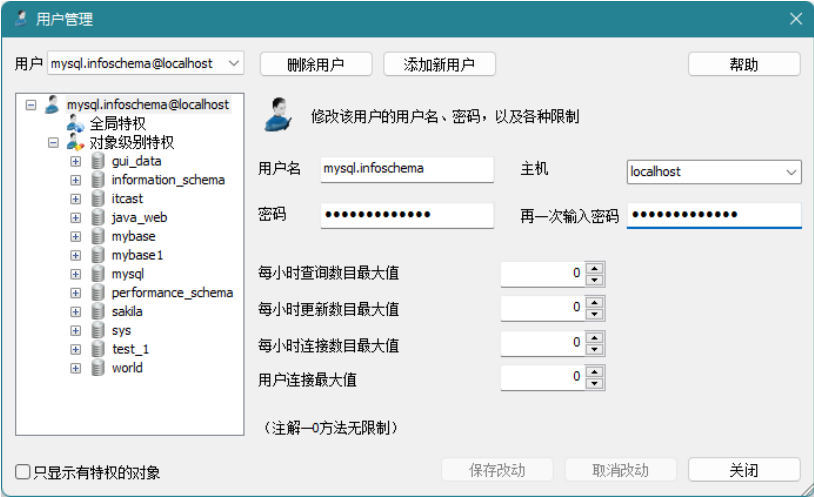
2.1 在 oracle 官网下载 jdk17，配置 Java 17 运行时环境



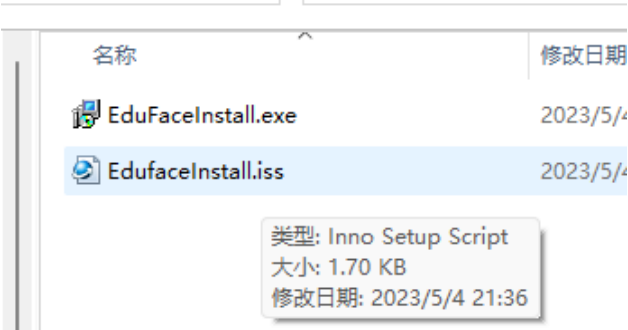
2.2 在 opencv 官网下载 OpenCV 公共库, 配置 OpenCV3_x64_v14 系统环境变量



2.3 配置 MySQL 设置 root 的用户, 123456 的密码和默认登录权限。



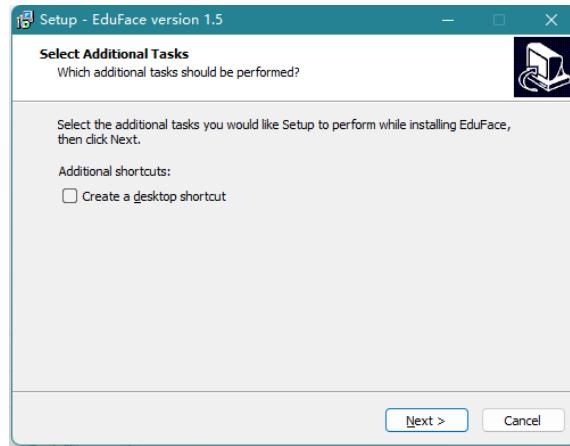
2.4 从分享链接中下载安装包，解压下载的压缩包



2.5 从分享链接中下载 SQL 建表语句压缩包，解压压缩包，通过压缩包创建数据库

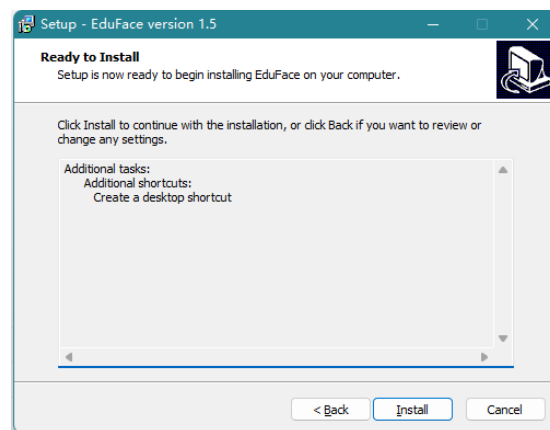
名称	修改日期	类型
course.sql	2023/4/30 18:50	SQL Text File
kaoqing.sql	2023/4/30 18:49	SQL Text File
student_1.sql	2023/4/30 18:46	SQL Text File
t_class.sql	2023/4/30 18:53	SQL Text File

2.6 运行安装包程序



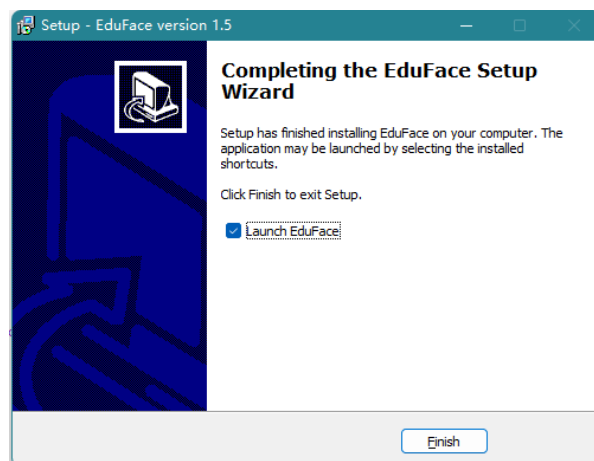
点击下一步

2.7 选择安装目录



点击安装

2.8 安装完成运行软件



点击完成

第六章 项目总结

在开发过程中，我们遇到了许多挑战和困难，但通过团队协作和不断努力，最终完成了这个基于视觉神经网络的智能考勤签到管家——EduFace。

首先，在项目协调方面，我们采用了敏捷开发的方法，将整个项目分为多个迭代周期，每个周期完成一部分功能的开发和测试。在团队成员之间建立了良好的沟通机制，及时反馈问题和解决方案，以确保项目进展顺利。

在任务分解方面，我们采用了分层次的任务分解方法，并分配给不同的团队成员。通过这种方法，我们可以更好地跟踪任务完成情况，并在开发过程中快速解决问题。

在克服困难方面，我们发现项目中难免会出现各种问题，如技术难点、代码优化、安全性等问题。针对这些问题，我需要不断学习和探索，从网上搜索解决方案、参考文档和论坛等进行学习，同时也需要自己不断尝试和实践。这样才能不断提高自己的能力，更好地应对项目中遇到的各种问题。

在水平提升方面，我们发现项目开发是一个不断提高自己技术水平的过程。在开发 EduFace 的过程中，我们需要学习相关的技术和算法，如人脸识别算法、深度学习等，并不断优化自己的代码和技术水平。

在升级演进方面，我们认为 EduFace 还有很大的升级空间。我们计划进一步优化算法和界面设计，提高产品的可用性和用户体验。例如，可以将人脸识别算法进一步优化，提高识别的准确率和速度。同时，也可以增加更多的功能和模块，如远程签到、考勤统计分析等。这些升级和演进将使 EduFace 更加完善和实用。在未来，我们还将进行手机端 APP 的开发，以使得我们的产品能够接触更广的用户面。

在商业推广方面，我们计划通过社交媒体宣传、线上线下活动等方式推广产品，并与相关合作伙伴合作，将产品推向市场。

总之，这个项目不仅是技术上的一个重要突破，也是团队协作和成长的一个重要经验。我们将继续努力，在之后不断迭代更新应用，以推出更好更快的产品。

参考文献

- [1] 学习 OpenCV3 (中文版). [美] Kaehler / Gary Bradski 著. 阿丘科技, 刘昌祥, 吴雨培, 王成龙, 崔玉芳译. 清华大学出版社. 2018. 7
- [2] 软件工程 (第 2 版). 曾强聪 赵歆 阳王东 著. 中国水利水电出版社. 2018. 12
- [3] 数据库系统概论 (第 5 版). 王珊, 萨师煊 著. 高等教育出版社. 2014. 9