



自习室考勤系统

——基于pyqt+dlib+MySQL

目录

CONTENTS



绪论

选题背景
国内外相关研究综述
理论基础与文献综述
主要贡献与创新



环境部署

在canda中部署环境并第三方库
换源操作
与pycharm的关联



算法介绍

Pyqt功能介绍
Hog算法
模型加载算法
Threading
人脸比对算法



视频演示

主界面
子界面 交互



遇到的问题及解决方案

收获
问题及解决方案



绪论

- ✓ 选题背景
- ✓ 目标及意义
- ✓ 应用场景

选题背景



在现代社会，教育和学习对于个人和社会的发展至关重要。大多数学校和教育机构都设有自习室，供学生在课外时间进行学习和准备。然而，自习室的有效管理和考勤记录一直是学校管理者和教师们面临的挑战之一。传统的考勤方式，如手动签到或使用学生证扫描，存在许多问题，如易于伪造、耗时、容易出错等。

选题背景



近年来，随着人工智能和计算机视觉技术的快速发展，人脸识别技术逐渐成为一种可行的解决方案。人脸识别技术可以通过识别和比对学生的面部特征，实现自动化的考勤系统。这种技术能够准确地记录学生的到勤情况，提高自习室的管理效率，并降低人工操作的工作量。该选题旨在探讨人脸识别技术在自习室考勤中的应用。通过研究和分析人脸识别技术的原理、算法和应用案例，可以实现自动化考勤，便于数据统计分析以提高自习室管理效率，提升学生的自习效率。

目标及意义



- 1. 自动化考勤：**人脸识别技术可以自动记录学生的到勤情况，减少了学生和教师的手动操作，提高了考勤的准确性和效率。
- 2. 防止作弊：**传统的考勤方式容易受到学生的伪造和代签，而人脸识别技术可以通过对学生面部特征的识别和比对，有效地防止作弊行为的发生。
- 3. 数据统计和分析：**人脸识别系统可以生成详细的考勤数据，包括学生的到勤时间、离勤时间和考勤统计等信息。这些数据可以用于学生学习习惯的分析和评估，为学校 and 教师提供参考依据。
- 4. 提升自习室管理效率：**人脸识别技术可以帮助学校和教育机构更好地管理自习室资源，了解自习室的使用情况和繁忙时段，以便做出合理的调配和安排。

应用场景



1. **访问控制与安全**：人脸识别可用于替代传统的身份验证方式，如密码或门禁卡，实现安全门禁系统。它可以应用于企事业单位的办公楼、仓库、实验室等场所，以及公共场所如机场、地铁站、银行等。
2. **考勤与时间管理**：人脸识别可以用于学校、企业等组织的考勤系统，准确记录员工或学生的出勤情况，提高考勤效率，并避免考勤记录的作弊问题。
3. **人员监控与安防**：人脸识别技术可应用于视频监控系统中，对人员进行实时识别与监测。它可以帮助监控中心或安保人员识别可疑人员、追踪犯罪嫌疑人，提高公共场所的安全性。

应用场景



4. **金融服务**：人脸识别可以用于银行、ATM机等金融服务场所，用于身份认证、客户验证和交易授权，提高金融交易的安全性和便捷性。
5. **零售业与支付**：人脸识别可用于零售业的顾客识别，实现个性化推荐、会员管理和优惠服务。同时，它也可以用于人脸支付系统，使顾客在购物时无需现金或银行卡，仅通过人脸识别完成支付。
6. **个人设备解锁**：人脸识别可以用作智能手机、平板电脑、笔记本电脑等个人设备的解锁方式，提供方便的用户体验和设备安全性。
7. **医疗与健康**：人脸识别可应用于医院或医疗机构的身份验证和访客管理，确保医疗信息的安全性和准确性。



环境部署

- ✓ 在conda中部署环境并第三方库
- ✓ 换源操作
- ✓ 与pycharm的关联

Conda的环境创建及第三方库



```
(base) PS C:\Users\Administrator> conda env list
# conda environments:
#
base                    *  D:\Anaconda
FaceAi                  D:\Anaconda\envs\FaceAi
tensorflow_env          D:\Anaconda\envs\tensorflow_env
```

```
(base) PS C:\Users\Administrator> conda activate FaceAi
```

```
(FaceAi) PS C:\Users\Administrator> pip list
```

Package	Version
contourpy	1.1.0
cycler	0.11.0
cdlib	19.24.0
fonttools	4.40.0
importlib-resources	5.12.0
kiwisolver	1.4.4
matplotlib	3.7.1
numpy	1.23.2
opencv-contrib-python	3.4.11.45
opencv-python	3.4.11.45
packaging	23.1
Pillow	9.4.0
pip	23.1.2
pyarsing	3.1.0
python-dateutil	2.8.2
setuptools	67.8.0
six	1.16.0
wheel	0.38.4
zipp	3.15.0

Conda的换源操作



- 1、打开终端
- 2、编辑源列表文件
- 3、备份源列表文件
- 4、更改软件源
- 5、保存更改
- 6、更新软件包列表
- 7、完成

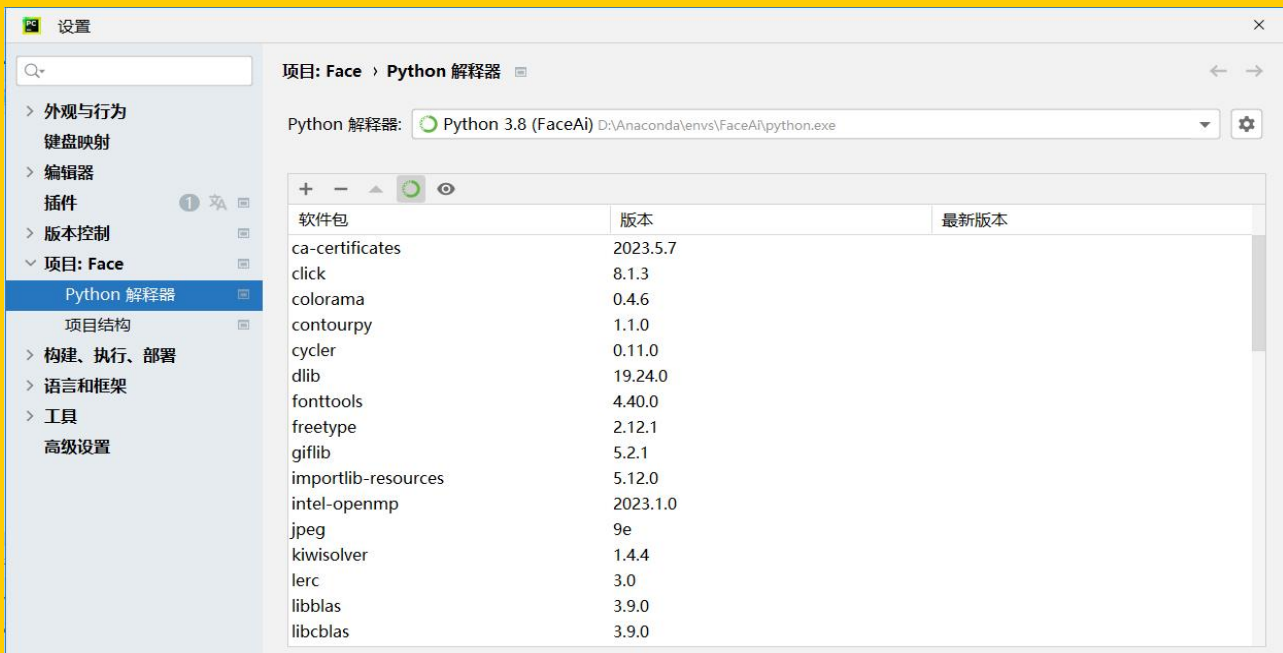
```
.condarc - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
channels:
- defaults
show_channel_urls: true
default_channels:
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
custom_channels:
conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch-lts: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

```
(base) PS C:\Users\Cotanlin> conda config --set show_channel_urls yes
```

与PyCharm的关联



在PyCharm中添加新的配置好的解释器





业务介绍

- ✓ 人脸注册与识别
- ✓ 信息管理（签到签退）
- ✓ 流量控制



算法介绍

- ✓ Pyqt功能介绍
- ✓ Hog算法
- ✓ 模型加载算法
- ✓ Threading
- ✓ 人脸比对算法

Pyqt功能介绍

Hog算法

模型加载算法

thread

人脸比对算法

Pyqt介绍



PyQt是使用Python语言编写的QT库的界面工具包。QT是一个跨平台的C++应用程序开发框架，可以用于创建高性能、高可靠性、图形用户界面友好的应用程序。

PyQt是 Riverbank Computing 开发的GUI小部件工具包。它是一个Python 接口，用于Qt，最强大、最流行的跨平台 GUI 库之一。PyQt 是Python 编程语言和 Qt 库的混合体。

Pyqt常用控件

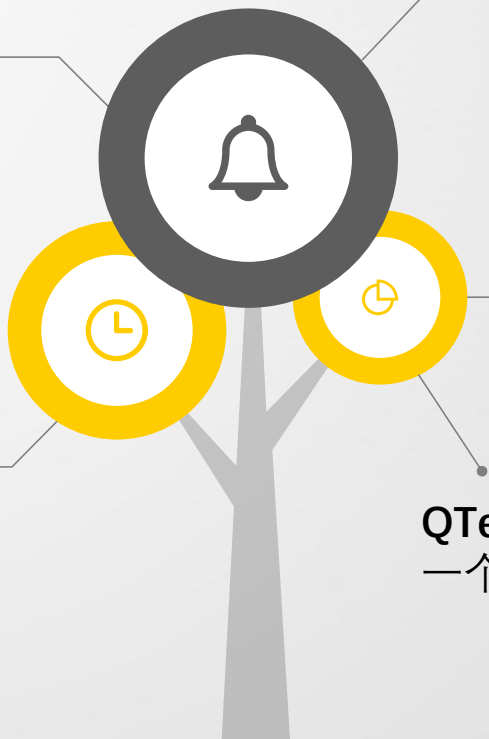
QLabel控件：用来显示文本或图像

QPixmap控件：可以在绘图设备上显示图像，通常放在QLabel或QPushButton类中

QLineEdit窗口控件：提供了一个单页面的单行文本编辑器

QTextEdit窗口控件：提供了一个单页面的多行文本编辑器

QPushButton窗口控件：提供了一个命令按钮



Pyqt常用的类

用于管理图形用户界面应用程序的控制流和主要设置

02

是所有用户界面对象的基类，所有的窗口和控件都直接或者间接继承自QWidget类。

04

📁 QApplication

📁 QMainWindow

📁 QWidget

📁 QDialog

01

为用户提供了一个应用程序框架，它拥有自己的布局，可以在布局中添加不同的控件

03

是对话框窗口的基类

Pyqt功能介绍

Hog算法

模型加载算法

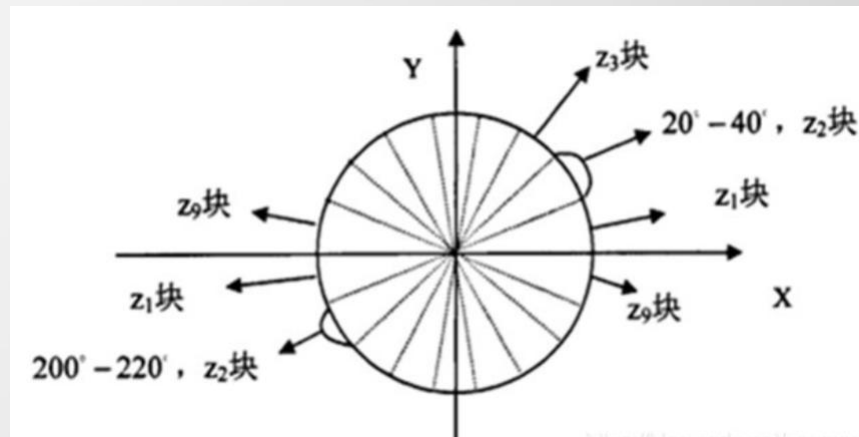
thread

人脸比对算法

hog介绍



HOG算法（方向梯度直方图）主要是为了基于各像素点的梯度提取出图像中目标的轮廓，用较少的特征更好地表达图像中目标信息。



Hog算法的过程



首先读取彩色图像并将其转化为灰度图像；并对灰度图像矩阵归一化，目的是为了减少光照和背景等因素的影响，本文主要用伽马校正法；选择合适的梯度算子来计算梯度图，主要分为x和y方向上的梯度；然后计算出合梯度的幅值和方向；划分检测窗口成大小相同的cell单元；组合相邻的cell单元成更大的相互重叠的块（block），便于充分利用重叠的边缘信息，然后统计整个块的直方图；对每个块内的梯度直方图归一化，综合所有块的信息，对HOG特征描述符可视化

Pyqt功能介绍

Hog算法

模型加载算法

thread

人脸比对算法

模型加载算法的介绍



模型加载算法通常是指在机器学习或深度学习中，将预训练的模型加载到内存中以供后续使用的过程。具体的加载算法可能会根据使用的框架或库而有所不同，下面是一个通用的加载

模型加载算法的过程



1. 定义模型结构：首先，需要定义模型的结构，包括网络层、激活函数、损失函数等。这通常是通过编写模型的类或函数来完成。
2. 初始化模型对象：根据定义的模型结构，创建一个模型对象。这个对象将用于加载权重和进行推断或训练。
3. 加载权重：将预训练的模型权重加载到模型对象中。权重可以是事先训练好的参数，也可以是保存在磁盘上的模型文件。加载权重的方法可能因框架而异，但通常都有相应的API或函数可用。
4. 设置推断参数（可选）：如果要进行模型的推断操作，可以设置相关的推断参数，例如批处理大小、输入尺寸等。
5. 使用模型：现在，加载的模型已经准备好使用了。根据具体的应用场景，可以将输入数据传递给模型进行推断或训练。

Dlib模型

shape_predictor_68_face_landmarks.dat是已经训练好的人脸关键点检测器。

形状：形状就是人脸上的有特征的位置

特征点：形状由特征点组成，图中的每一个黄点就是一个特征点

dlib_face_recognition_resnet_model_v1.dat是训练好的[ResNet](#)人脸识别模型。

```
# 1.加载人脸检测模型
```

```
self.face_hog_detector = dlib.get_frontal_face_detector()
```

```
#2.加载68个人脸检测模型
```

```
self.shape_detector = dlib.shape_predictor("../resource/shape_predictor_68_face_landmarks.dat")
```

```
# 3. 加载128个特征人脸向量对象
```

```
self.face_descriptor_extractor = dlib.face_recognition_model_v1("../resource/dlib_face_recognition_resnet_model_v1.dat")
```

```
# 打开摄像头，创建视频流
```

Dlib人脸识别的基本思路



Dlib中，人脸识别的基本思路为：

- 1、计算已知图片中所有人脸对应的特征向量；
- 2、计算要识别的未知图片中所有人脸对应的特征向量；
- 3、计算人脸之间的欧式距离；
- 4、如果两张人脸之间的欧式距离小于设定的阈值，则认为同一个人，否则认为不是同一个人

Pyqt功能介绍

Hog算法

模型加载算法

thread

人脸比对算法

Threading模块的介绍



threading模块是Python支持的[多线程](#)编程的重要模块，该模块是在底层模块thread的基础上开发的更高层次的多线程编程接口，提供了大量的方法和类来支持多线程编程。

threading.Timer的介绍



`threading.Timer` 是 `threading.Thread` 的一个派生类, 是在指定的时间 `n` 秒后执行一个函数功能。它会集成 `threading.Thread` 的很多属性和方法。

`Timer`的源码实现很简单, 收到一个任务后, 则创建一个线程, 线程逻辑里面最前面插入`sleep`。

threading.Timer



➤ threading.Timer 构造函数

```
timer=threading.Timer(interval, function, args=None,  
kwargs=None)
```

➤ 参数列表

interval: 以秒为单位, 指示该线程过多久启动

function: 指示该线程要调用什么函数, 执行什么功能

args: 适用于 function 的参数列表

kwargs: 适用于 function 的参数字典

threading.Timer 属性和方法



threading.Timer 属性和方法

序号	属性和方法	描述
1	属性 args	适用于 function 的参数列表
2	属性 interval	以秒为单位, 指示该线程过多久启动
3	属性 kwargs	适用于 function 的参数字典
4	方法 stop()	如果这个 Timer 还没有结束, 则结束之。

Pyqt功能介绍

Hog算法

模型加载算法

thread

人脸比对算法

人脸比对算法

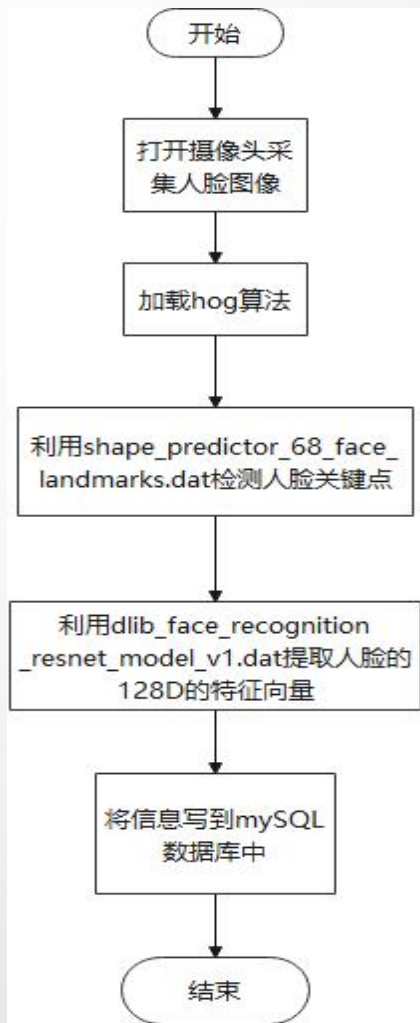


两个向量的欧氏距离越小，说明两个向量越接近。将这个结论放在人脸识别中，可以得出以下结论：

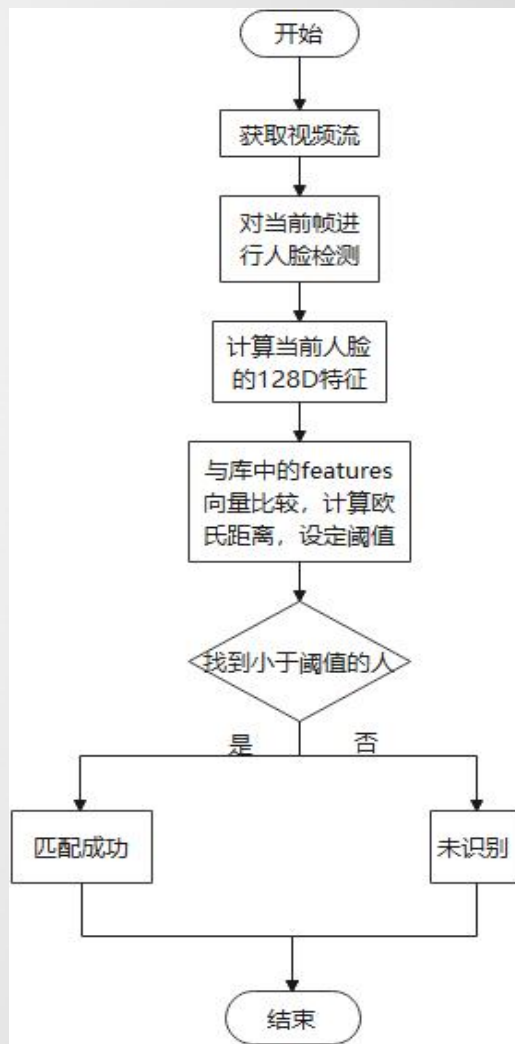
两张人脸特征向量的欧氏距离越小，说明两个人越相似。当欧氏距离小于某一个值时，则可以认为他们是同一个人。

基于这个结论，我们可以实现人脸识别。

人脸注册流程图



人脸识别流程图



人脸注册相关模块及函数功能介绍



人脸识别相关模块及函数功能介绍





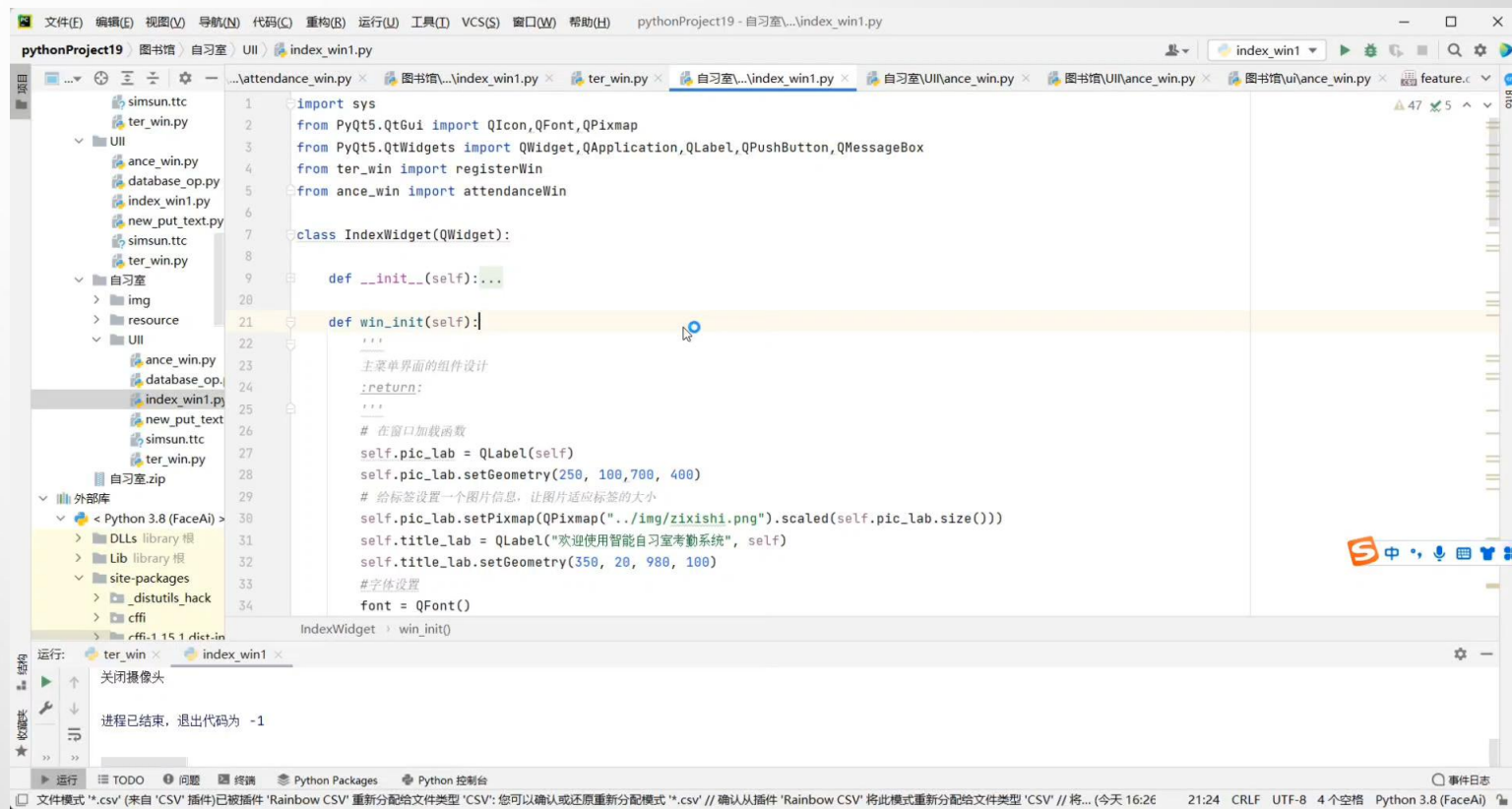
视频演示

- ✓ 主界面
- ✓ 子界面 交互



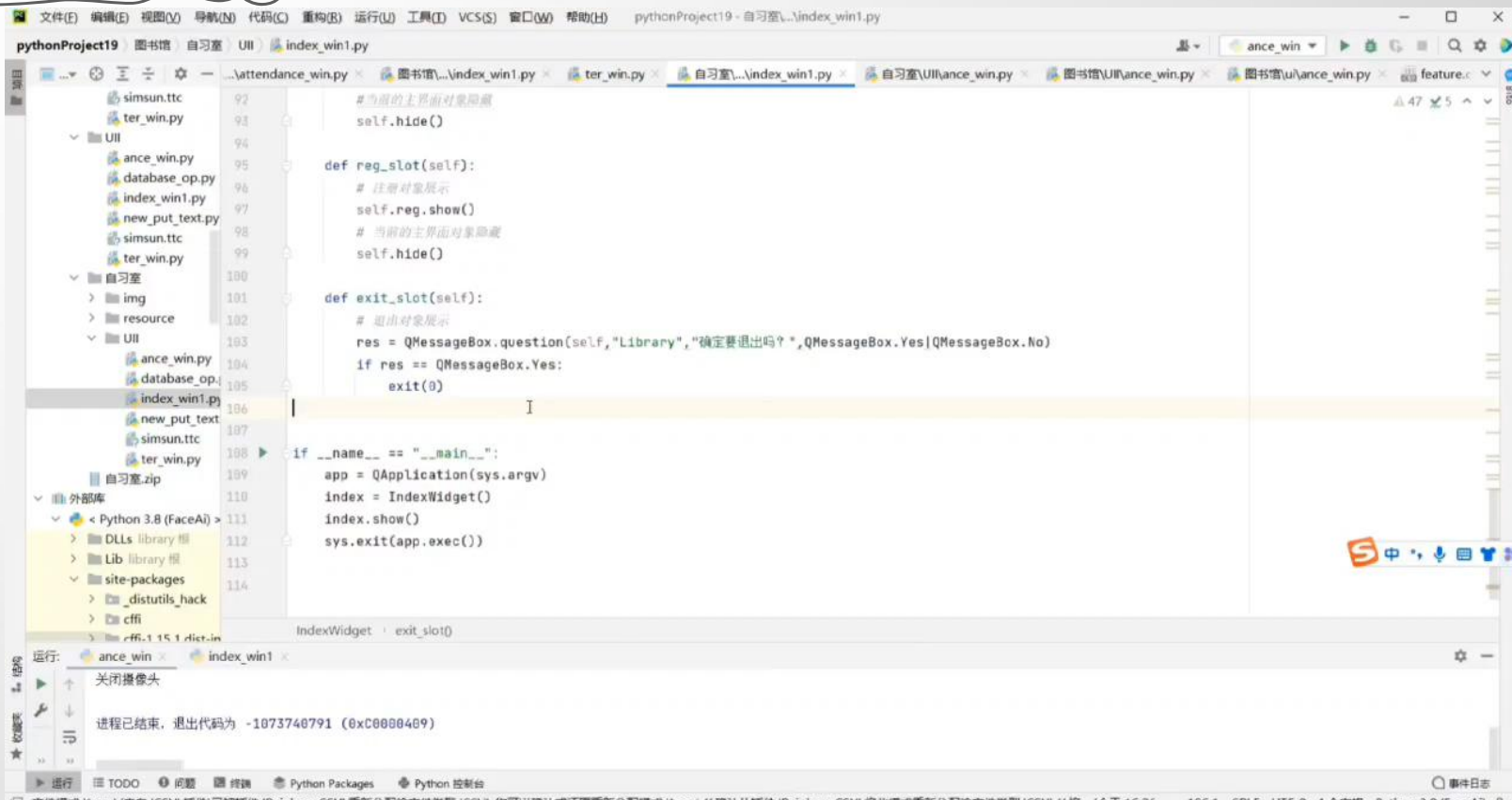


自习室门禁系统





自习室流量限定





数据库

数据库的功能



- 1、**数据定义功能**：DBMS（数据库管理系统 DataBase Management System）提供了DDL（数据定义语言 Data Definition Language），用户可以通过它方便的对数据库中的数据对象进行定义。
- 2、**数据组织、存储和管理**：DBMS要分类组织、存储和管理各种数据，包括数据字典、用户数据、数据的存取路径。
- 3、**数据操纵功能**：DBMS还提供了DML（数据操纵语言），用户可以使用DML操纵数据增、删、改、查。

数据库的功能



4、**数据库的事物管理和运行管理**：数据库在建立、运用和维护时由数据库管理系统统一管理、统一控制，以保证数据的安全性、完整性、多用户对数据的并发使用以及发生故障后的系统恢复

5、**数据库的建立和维护功能**：数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组功能和性能监视、分析功能等。这些功能通常是由一些实用程序或管理工具完成的

6、**其他功能**：DBMS与网络中其他软件系统的通信功能；一个DBMS与另一个DBMS或文件系统的数据转换功能；异构数据库之间的互访和互操作功能

MySQL数据库



MySQL是一个开源的关系型数据库管理系统，常用于Web应用程序的后端数据存储。它支持多种操作系统和编程语言，具有高性能、可靠性和可扩展性等特点。

ID	NAME	FACEINFO	
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100225	郑佳文	2023-07-11,...	19B
202005100225	郑佳文	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100221	祁麟	2023-07-11,...	19B
202005100225	郑佳文	2023-07-11,...	19B
(NULL)	(NULL)	(NULL)	0K

ID	NAME	FACEINFO	
202005100223	张雨	[-0.11094587296247482, 0.13244925439357758, 0.0753150805...	2K
202005100225	郑佳文	[-0.14397165179252625, 0.1267971694469452, 0.05802478641...	2K
202005100230	马晓璇	[-0.09662844985723495, 0.1234751045703888, 0.06919878721...	2K
202005100221	祁麟	[-0.11960184574127197, 0.12329735606908798, 0.0385044775...	2K
(NULL)	(NULL)	(NULL)	0K

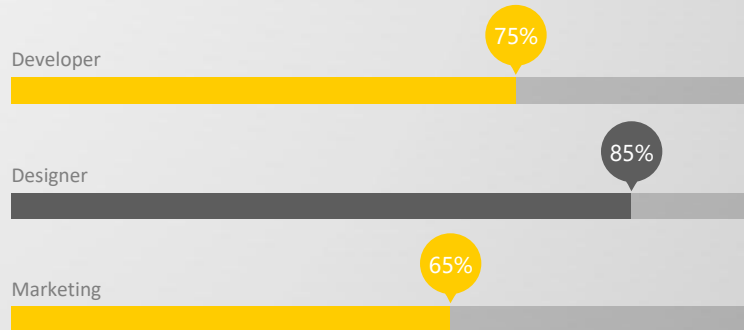


项目收获及问题解决



自习室考勤系统项目收获

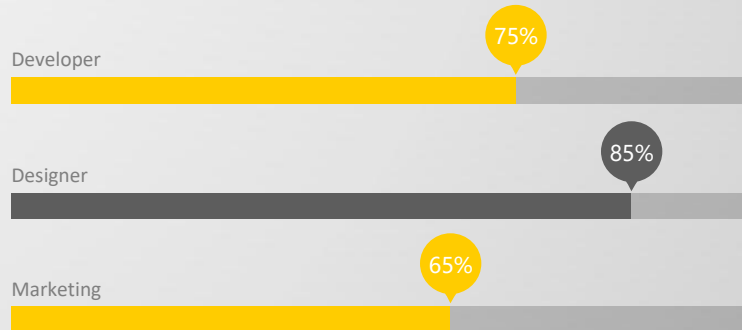
实践经验：项目实践是学习的最好方式之一，通过实际开发该系统，将获得宝贵的实践经验。面临真实的开发挑战，包括需求分析、系统设计、编码实现、错误调试等，这些经验将对日后的开发工作有很大帮助。





自习室考勤系统项目收获

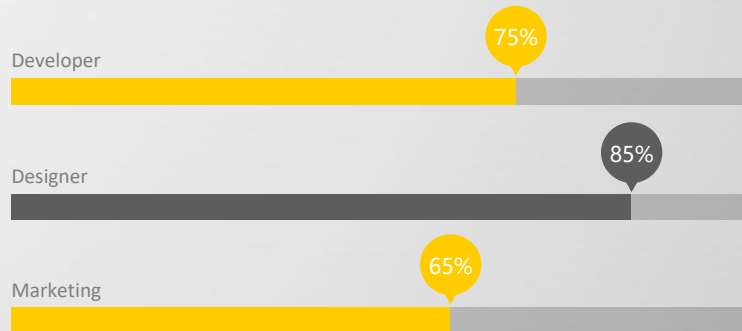
团队协作与沟通：如果该项目是在团队中进行开发，有机会锻炼团队协作和沟通能力。与团队成员合作，共同解决技术问题，分工合作完成任务，增强了团队合作和沟通技巧。





| 自习室考勤系统项目收获

技术掌握：通过该项目，学习和掌握PyQt和MySQL等技术的应用，了解集成人脸识别算法。对人脸识别技术有更深入的理解。学习人脸检测、特征提取、模型训练等关键概念和算法，了解人脸识别的原理和应用场景，为以后在人工智能领域的工作打下基础。



项目问题及解决方法



1. 点击认证按钮闪退

原因及方法：信息输入格式不对，将格式调整为utf-8

2. csv数据处理在向mysql数据库过程中输入与读取失败

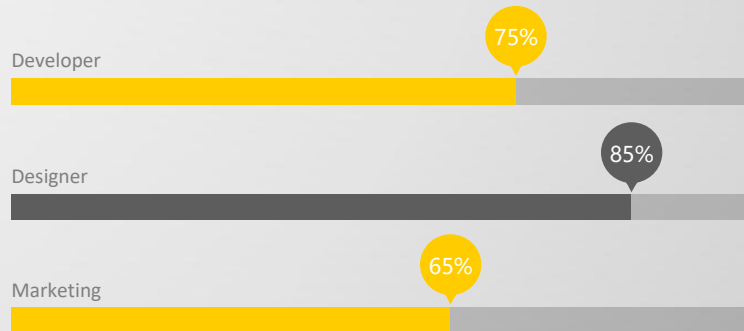
原因及方法：对mysql的语句格式编写有误，查询书籍，更正格式

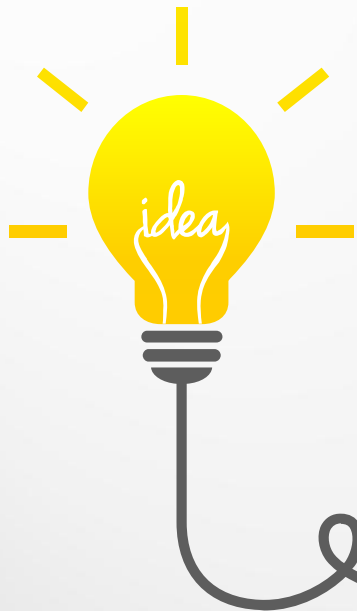
项目未解决问题：识别过程中有明显卡顿



| 自习室考勤系统

实际应用中还需再完善，完善打卡时间，
提高识别准确率，提高识别速度，提高
界面切换的流畅性





敬 请 各 位 老 师 批 评 指 正

T H A N K Y O U F O R W A T C H I N G