**实验一**

**班级：数媒1801 姓名：王禹洁 学号：1191180102**

1. **实验目的**

通过建立自己的人脸检测识别系统，深入理解人机交互设计。

1. **实验要求**

能够调用摄像头实时抓取图像，如果图像中出现人脸，能够检测定位人脸，同时保存人脸，建立自己的人脸样本集；训练人脸识别模型；测试人脸识别模型。

1. **实验内容**

* 进入主界面：

计算机生成了可选文字:
1B实琥
人脸检测
训练模型
0
人脸识别
FACERECOGNITIONSYSTEM
注册
数据库管理
f000ree09nition实现
人脸识别

* 使用LBPH算法实现的人脸识别：
* 点击人脸检测按钮：

计算机生成了可选文字:
recognltlon.py
importCV2
0
DataManage•py
0
facedatamanage•py
0
facereco.py
Unknown•Jpg
0
DataRecord.py
D:\PythonV&iRÉIJ
?ts
D：
1
2
3
一1人脸采
《8《2
打开摄像头
wn•JPCJ
开启人脸检测
e系统已采隼幀数
taRecord.ui
锈，乁了
开始采集人脸数据
aconcla3\pythonXe
e1．9．6
就
fromthepyganeC
采集当前捕获幀
impor、tnumpyaSnp
fromPyQt5impor、tQtCore,QtGui
fromPyQt5.QtCor、e]．「叩田、tQTimer、，QRegExp,pyqtS1gnaI
f「〕PyQt5.QtGui1「叩田、
Qlnage,QPixnap,QIcOn，QRegExpVaIidato
f「〕PyQt5.Qtl!idgets1「叩田气
Q囗1钔一og，QAppIication,Ql!idget,QMe
f「〕PyQt5.uic1「叩田、t]一03d凵1
1「叩田、t
1「叩田、t
1「叩田、t
1「〕卩囗「
1「叩田、t
-Logging
-Logging“config
qUeUe
thr、eading
sqlite3
摄像头未开启
《f00门datet1me1「叩田、tdatetilne
系统志
数景入
，Qe数掮库状态
sag
数据库初始化
已存人脸样本数：
奉佶息
增加用户信息
确认值息
英文名
同步到数据库

1. 点击数据初始化按钮，进行数据初始化。

计算机生成了可选文字:
3
4
from
from
impor、tQtCore,QtGui
PyQt5
PyQt5.QtCoreimpor、tQTimer,QRegExp,pyqtSigna1
人胆采集
打开摄像头
开启人脸检灕「叩〔〕
系统已采隼幀数
口
开始采集人脸数据《
pyganecommunity
采集当前捕获帧
PY
Py
t
t
t
da
t5.Qtl!idgets
t5．U1C1「〕卩囗「
Ogg1ng
1「叩田、tQ囗1钔一og，QAppIicationQl!idget,
]一o裉用1
QMessageBox
ogging“config
hreading
qIite3
摄像头未开启
etime廿叩田、tdatetine
[2020／10／1蠶1一：0到
ht
二0艹地》《
数景入
数库状态
数据库初始化
已存人脸样本数：
基奉佶息
增加用户信息
确认值息
英文名
同步到数据库

1. 点击增加用户信息按钮，添加用户信息：

计算机生成了可选文字:
4
from
from
人脸采集
打开摄像头
开启人脸检灕「叩{〕
系统已采隼幀数
口
PYQ
Impor、t
PyQt5.QtCoreimpor、tQTimer,QRegExp,pyqtSigna1
t5.Qtl!idgets1「叩田、
Q囗1钔一og，QAppIication,Ql"idget,
PY
Py
t
t
da
QMessageBox
t5．U1c1「叩田、t]一O用1
Ogg1ng
hreading
qIite3
etime
英文名
1「〕卩囗
开始采集人脸数据《n/
ePYg3mecommunityh
采集当前捕获帧
Main
[2020／10／1822：31一：0到Success
t
个人偏息
谲输入10位学号
谲输入10位内汊字
谲输入10个字母以内
确定
二0艹地》《
数景入
数库状态
．数据库初始化
已存人脸样本数：
基奉佶息
增加用户信息
确认值息
英文名
同步到数据库

1. 点击打开摄像头按钮，打开摄像头。

计算机生成了可选文字:
3
4
from
from
PyQt5
PY
Py下
impor、tQtCore,QtGui
PyQt5.QtCoreimport
QTimer,QRegExp,pyqtSigna1
人脸采集
关闭摄像头'
开启人脸检测
系统已采幀数
口
开始采集人脸数据《
ePYg3mecommunity
采集当前捕获帧
idgets1
LJiC
ging.ct〕nfl
eading
t
t
da
lite3
e
Import
daetime
Application,QWidg
1
Malnwlndow.
[2020／10／1；31：0
ht
数捻景入
数库状态
．数据库初始化
已存人脸样本数：
基奉佶息
增加用户信息
确认值息
11911日0102
英文名
同步到数据库

1. 点击开启人脸检测按钮。

计算机生成了可选文字:
3
4
from
from
PyQt5impor、tQtCore,QtGui
PyQt5.QtCoreimport
QTimer,
QRegExp,
人胆采集
关闭摄像头；
关闭人脸检渺
系统已采幀数
开始采集人脸数据
ePYg3mecommunity
采集当前捕获帧
PY
Py
t
t
t10《
pyqtSigna1
lication,QWidgee,::Q
Widgets1
uiC
ing.config
eading
lite3
ssageBc
1
e
import
1
daetlme
罾aIn豐Indo豐．
[2020／10／122；31一：0到
ht
UCC"《s
二0艹地》《
景入
数库状态
．数据库初始化
已存人脸样本数：
增加用户信息
确认值息
11911日0102
英文名
同步到数据库

1. 点击开始采集人脸数据的按钮

计算机生成了可选文字:
3
4
from
from
PyQt5
PY
Py
t
t
import
PyQt5.QtCore
QtCore,QtGui
import
QTimer,
etime
QRegExp，
人脸采集
关闭摄像头'
关闭人脸检测一
系统已采隼幀数
@结束当前人脸采#
ePYg3mecommunity
采集当前捕获幀
pyqtSigna1
icationQWidget
U1C
Imp
Ing.conflg
eading
lite3
1
e
import
t10《
[2020／10／11一：“
ht
UCC"《s
：据初始化完成
数景入
数库状态
数据库初始化
已存人脸样本数：
增加用户信息
确认值息
11911日0102
英文名
同步到数据库

1. 点击采集当前捕获帧的按钮，获取当前摄像头内人脸照片。

* 第一次采集时是50帧，可以变换角度再进行其他角度的采集。每一次是50帧。到300帧以后就不再以50帧为基本单位采集，而是一帧一帧增加。

计算机生成了可选文字:
4
from
PyQt5.QtCore
PY
Py
t
da
h
import
4
QTimer,
人脸采纂
关闭摄像头'
关闭人脸检测
Y
系统已采隼幀数
L3L7!H
@结束当前人脸采#
．6
t
ePYg3ecommunity
．采集当前捕获幀
QRegExp,pyqtSigna1
plication,QW氤訁壶
0
二0艹地》《
MessageBc
eading
lite3
e
import
daetime
[2020／10／1，蠶1一：0到
景入
数库状态
．数据库初始化
已存人脸样本数：
基奉佶息
增加用户信息
确认值息
11911日0102
英文名
同步到数据库

1. 点击结束当前人脸采集的按钮，结束采集数据。
2. 关闭人脸检测。
3. 关闭摄像头。
4. 点击同步到数据库，将信息同步到数据库。图像数据集存到相应文件夹下。

计算机生成了可选文字:
3
4
from
from
impor、tQtCore,QtGui
PyQt5
PyQt5.QtCoreimpor、tQTimer,QRegExp,pyqtSigna1
t5.Qtl!idgets1「叩田、tQ囗1钔一og，QAppIication,Ql!idget,
QMessageBox
6
t
人脸采集
打开摄像头
开启人脸检灕即{〕
系统已采幀数
卩y开始采集人脸数据〕
ePYg3mecommunity
采集当前捕获帧
PY
Py
t
t
t
da
h
t5．U1c1「叩田、t]一O用1
Ogg1ng
ogging“config
hreading
qIite3
etine
0OpenCVFaceRecognitionSystem-DataR..
1191180102已添加/更新到数据库。
王禹洁的人論数据采集已完成！
OK
datetine
数捻景入
数库状态
．数据库初始化
已存人脸样本数：
增加用户信息
确认值息
号
11911日0102
英文名
[2020／10／1《；31一：0-Success，据初始化完成
t
pData\Loca1-\Telnp\1\pip-req-blJi1-d-sxpslnzt6\01)encv\lnodlJ1-es\vi
io\
同步到数据库

* 点击训练模型按钮：

计算机生成了可选文字:
Import
〕mPt5.QtCoreimortQTimer,
数答理
用「tQI
数掩状态
Iidgets廿叩围、
过toad
数据库初始化
。训数
囗直方图均衡
《tetj丿讠眶、td3tet
系鳢日志
0/人脸识别0额nnd“．卩
Y
QRegExp,pyqtSigna1
QPixrnap,QIc0n，QRegE
3ge，
1
《》的一讼，FaceID
。学号
。姓名
QTextClJl、SOI、
英名
注
删用户
学号询
开始查询
询结果
FaceID：
．〔：pp除用户n瞿s一na
y0\App囗3t3\Loca]一\T叩1pip-req-blJiId-sxpsnzt6\openc\/\lnodlJIes\videoio\sl、c\cap_ln
注：FaceID为一1说明该用户的人脸数据未被训练

1. 点击数据库初始化，初始化数据库。

计算机生成了可选文字:
、0mPyQt5impor、tQtCore,QtGui
、0mPt5.QtCor、eimortQTimer,
QRegExp，
QPixmap，
pyqtSigna1
人媵庳
QIcOn，QRegE
t田、
QI
数掩状态
'idgets廿叩围、
、t_toad
刷新数据库
《卜脸样本：
IP训数
囗直方图均衡
d3t_et_1
开始训练
过d3tet
系鳢日志
0．一0022：43：04]
Success：据初始化完
3ge，
FaceID
1
姓名
孙谢婷
王手洁
QTextCur、sor、
《英文名
sunyltlng
1191180101
21191180102
2020
2020
删用户
学号询
开始查询
询结果
FaceID：
o/人脸识别0额nnd“．卩
t
Y
y0\App囗3t3\Loca]一\T叩》P
p-req-blJiId-sxpsnzt6\openc\/\lnodlJIes\videoio\sl、c\cap_ln
“CPP
除用户n瞿“s一na
注：FaceID为一1说明该用户的人脸数据未被训练

1. 左边栏可以训练人脸模型（直方图均衡点击与否都可）

训练成功：

计算机生成了可选文字:
链接
数答理
数库状态
．刷新数据库
已存人脸样本数：
训数
囗直方图均衡
开始训练
系鳢日志
Success：据初始化完
音频视频
录制
学号
FaceID
时闫戥
人庳
孙谢婷
王手洁
英名
sunyltlng
1191180101
21191180102
删用户
学号询
2020
2020
开始查询
询结果
0OpenCVFaceRecognitionSystem-DataManage×
Success!系统已生ü./recognizer/trainingData.yml
人論数据训练完成！
OK
FaceID：
删除用户
注：FaceID为一1说明该用户的人脸数据未被训练

1. 右边栏进行数据库内的用户信息管理。输入学号可以进行用户信息查询，也可以删除信息。

* 点击人脸识别按钮：

计算机生成了可选文字:
cv2
ntJmPYaSnp
扭像头实时画面
t51「叩田、ttC田、t3UI
5.QtC01、e1「叩田过QTine(、QRegExp,
pyqtSigna1-
5．QtGu11「叩田、tQlnage,QPixnap,QIcO的QRegExpVaIidat01、，QTextC
5.Qtl!idgets1「叩田、tQ囗1钔一og，QAppIication,Ql!idget,QMessage30x
5．U1c1「叩田、t]一O用1
gg1ng
gging“config
打开摄像头
S0
语音识别
语音播报
reading
tite3
1「〕p00
datetine
摄像头未开启
Deb四设首
置信度國值
q-blJi1-[l-sxsnzt6\
功能开关
、/人脸跟腙
人脸识别
、/报警系统
、/设备报警
系缔日志
[2020／10／1日22：45：51]Success：
据状态正常，发现用F：2
^识别设置
月．乬刀刂／．0上目过上囗do．py
\口方图均《
e「叩\1\pip一
囗调试模
自动报警國值
encv\「'lotllJ1-es\vi[le
ap一「旧f“Cpp
35〕
anonyl',l
注：以上调整只有开启调试模式才会触发

* 1. 可以直接点击摄像头的按钮，打开摄像头。

计算机生成了可选文字:
Y．扭像头实时画面
tsimpor、tQDi
importloadUi
XP,PVq
Applicat。
迎调试模
lg
al
谛音识别
egExpVa1彐dator
QWidget,QMessageBox
语音播报
record
识别设置
g-lng
ging.confl
hreadirg
sq1ite3
SYS
fromdatetime-npor、t
Deb四设首
datet
thePYgmecommunity.fit吓，
]一0b钔一的\L尾田、s\appveyol、\「0。性忄是：LDC?]一\Te「叩\\哲《一
6\0
功能开关
、/人脸跟腙
人脸识别
、/报警系统
、/设备报誓
系统日志
据状态正常，发现用F：2
ncv\lnodul-es\videoio\sl、c\c
置信度國值
囗直方图均衡化
0
注．以上调整只有开启调试模式才会触发
」动报警國值
0

* 1. 打开摄像头之后勾选上人脸识别，即可进行人脸识别。

计算机生成了可选文字:
3
rackin
record
人识别设置
囗直方图均衡化
from
from
PyQt5impor、tQtCore,QtGui
Y．扭像头实时画面
loadUi
eg
adi
ite3
datetime》．rpor、t
date
XP,PyqICJ
Applicati
囗调试模
打开摄像头
al
谛音识别《
egExpVa7_idat01
QWidget,QMessageBox
语音播报
功能开关
、/人脸跟腙
、/人脸识别
、/报警系统
、/设备报警
系统日志
据状态正常，发现用F：2
p巳0《tonCanc
Deb四设首
置信度國值
自动报警國值
0
0
注．以上调整只有开启调试模式才会触发

* 1. 或者可以直接点击语音识别，对着电脑大声说出：“打开摄像头”，即可打开摄像头。勾选上人脸识别后，对着电脑大声说出：“开始检测”，即可进行人脸识别。对着电脑大声说出：“关闭摄像头”，即可关闭摄像头。
  2. 在进行人脸识别的过程中，点击语音播报。系统会对识别出的人脸进行播报。
  3. 勾选调试模，调节置信度阈值和自动报警阈值，可以增加或减小对人脸识别的精确度。置信度过高，好处是可以识别出人脸，但认错率也较高。置信度过低，可以降低认错率，但是识别率也相应降低。
  4. 若是系统未识别出是哪个人，积累到一定次数会进行响铃报警。
  5. 勾选上直方图均衡化可以提高精确度。
  6. 再次点击“打开摄像头”可以关闭摄像头
* 使用Face Recognition下实现的人脸识别：
* 点击注册：

计算机生成了可选文字:
人脸采集
打开摄像头
拍照
[2020／10／1日23：03：07]
Success：据初始化
摄像头未开启
数景入
数库状态
已存人脸样本数：
佶息
增加用户信息
确认值息
英文名
同步到数据库

与LBPH使用过程类似，唯一不同的是在点击打开摄像头之后，点击拍照。得到一张人像照片即可在第三步进行识别。无须进行训练。

* 点击数据库管理：

计算机生成了可选文字:
数库状态
数据库初始化
已存人脸样本数：
系鳢日志
学号
FaceID
人脸庳
姓名
英名
注
删用户
学号询
开始查询
询结果
FaceID
删除用户
注．FaceID为一1说明该用户的人脸数据未被训练

与LBPH过程类似，可以进行数据库内的信息管理。进行用户新的删减查询。

* 点击人脸识别：

计算机生成了可选文字:
语音识别
语音播报
打开摄像头
人脸识别
设置
信度國值
0
图像大小
0
系统日志
[2020／10／1日23：07：2引
Success：据状态正
常，发现用F：2
扭像头实时画面
摄像头未开启
0
验证偏息
学号
文名
学号
文名
学号
文名

1. 点击打开摄像头，再打开人脸识别。进行人脸识别。

2. 或者同LBPH过程，直接使用语音识别打开摄像头并开始检测。

3. 识别出的人脸会在左侧显示照片和用户信息。

计算机生成了可选文字:
捏像头实时画面
语音识别
语音粞报
关闭摄像头
关闭人脸识别
信度國
碸圆像大小
[2020/1@1823：07．2引
“数据状态正
success。
韋，发现用尸数．2
验证俪息
王禹
11911就102
文名
难y」
文名
文名

1. 未识别出也会显示提示。

计算机生成了可选文字:
字体
语音识别
语音播报
关闭摄像头
关闭人脸识别
设置
信度國值
图像大小
系统日
[2020／10／1日23：07：2引
Success：据状态正
常，发现用F：2
扭像头实时画面
验证偏息
Unknown
学号
Unknown
文名
Unknown
学号
文名
学号
文名

5. 在检测出人脸后，可以点击语音播报，系统会对识别出来的人脸进行播报。

6. 可以在设置中调节参数来调节识别精度。

7. 也可以对多个人脸进行识别检测。

1. **实验技术和关键步骤**

本系统采取了两种不同的人脸识别方法来实现两种不同的人脸识别流程。这两种方法分别是Opencv自带的LBPH人脸识别算法和Python中的开源库face recognition人脸识别库。

1. 使用LBPH算法的人脸识别系统实现流程

* 人脸检测并采集人脸数据
* 连接数据库：利用数据库来存储学生信息。

1. 使用SQLite数据库，导入数据库的包。

import sqlite3

1. 设置数据库的名称和数据保存的文件路径。

self.database = **'./Facebase.db'** *# 数据库名称*self.datasets = **'./datasets'** *# 数据保存的文件*

1. 初始化用户信息字典

self.userInfo = {**'stu\_id'**: **''**, **'cn\_name'**: **''**, **'en\_name'**: **''**} *#用户信息字典*

1. 连接数据库

conn = sqlite3.connect(self.database)  
cursor = conn.cursor()

1. 创建数据表

cursor.execute(**'''CREATE TABLE IF NOT EXISTS users (  
 stu\_id VARCHAR(10) PRIMARY KEY NOT NULL,  
 face\_id INTEGER DEFAULT -1,  
 cn\_name VARCHAR(10) NOT NULL,  
 en\_name VARCHAR(16) NOT NULL,  
 created\_time DATE DEFAULT (date('now','localtime'))  
 )  
 '''**)

1. 查询数据表并返回信息。

cursor.execute(**'SELECT Count(\*) FROM users'**)  
result = cursor.fetchone()

* Opencv打开摄像头：本摄像头编号为0。

self.cap = cv2.VideoCapture()

self.cap.open(camID)

self.cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 640)  
self.cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 480)

ret, frame = self.cap.read()

* 人脸检测技术：使用Opencv DNN的人脸检测技术（目前Opencv系列中的精度最高的人脸检测技术）。

优点：

1）准确度高；

2）在CPU上能够实时运行；

3）适用于不同的人脸方向：上，下，左，右，侧面等。

4）甚至在严重遮挡下仍能工作；

5）可以检测各种尺度的人脸。

关键代码：

blob = cv2.**dnn.blobFromImage**(cv2.resize(frame, (300, 300)), 1.0,  
 (300, 300), (104.0, 177.0, 123.0))  
self.faceCascade.setInput(blob)  
faces=self.faceCascade.forward()

* 日志管理：进行日志管理能够很好记录当前程序状况。清晰地显示当前状态。可以用在整个界面的下端，便于进行操作，且日志具有线程安全性。

self.logQueue = queue.Queue() *# 日志队列*

self.logOutputThread = threading.Thread(target=self.receiveLog, daemon=True)  
self.logOutputThread.start()

* 训练模型
* 连接数据库

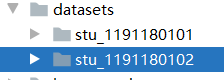
与人脸检测中连接数据库的方法相同。增加了删除数据库中用户信息的功能，可以删除用户信息及其人脸数据。

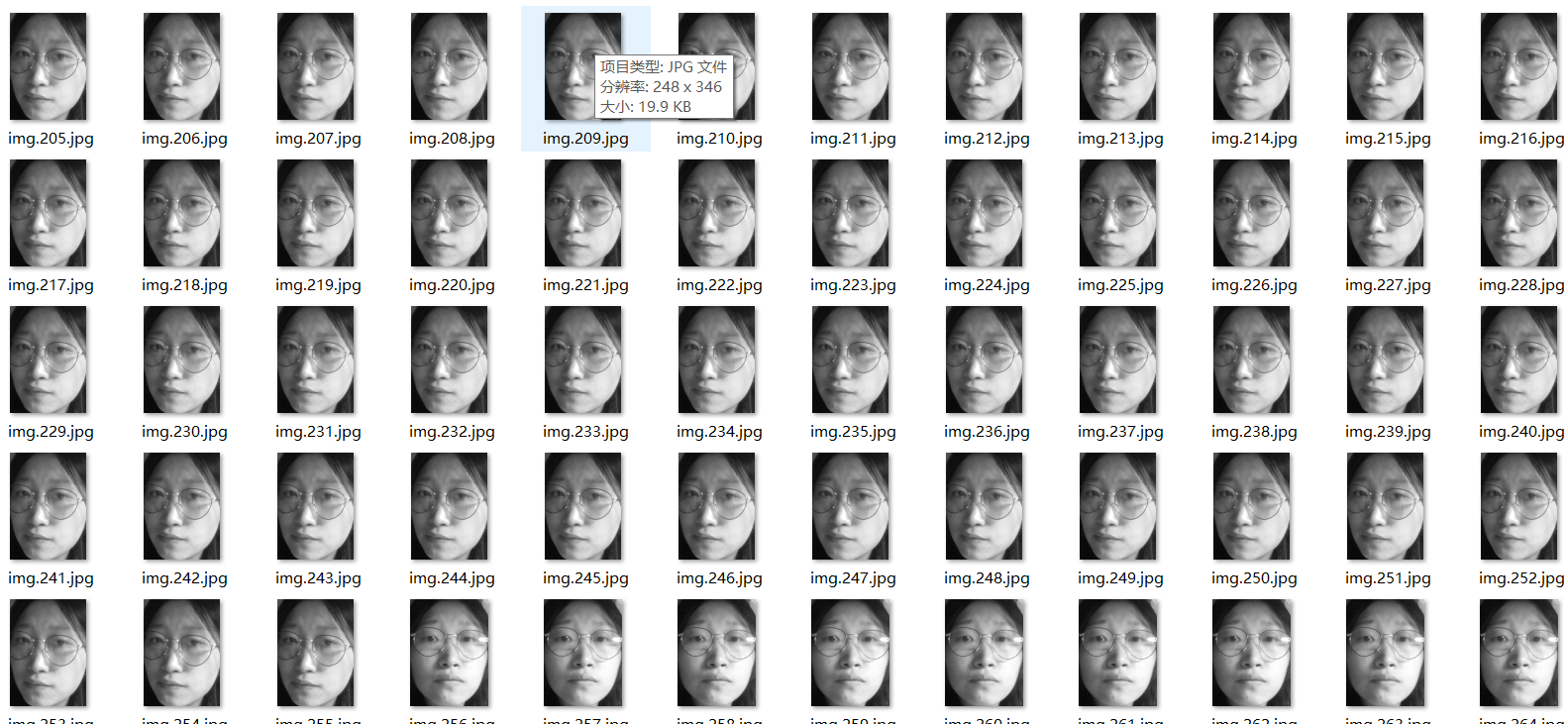
* 准备数据集

人脸数据并不存在数据库中，而是存在一个专门放人脸图片的文件夹下，每个学号对应一个文件夹（文件夹以学号命名），在准备数据集时，将数据库中的学号提取出来，找到相应的文件夹。



相应的人脸数据集：





* 使用数据集进行训练

1. 训练模型使用的是Opencv中的LBPH人脸识别算法中的训练函数。
2. 在使用此识别器训练之前要先创建识别器。

ace\_recognizer = cv2.face.LBPHFaceRecognizer\_create()

1. 利用识别器进行训练。

face\_recognizer.train(faces, np.array(labels))

1. 将训练过的数据保存在.yml文件中，以便后续识别使用。

face\_recognizer.save(**'./recognizer/trainingData.yml'**)

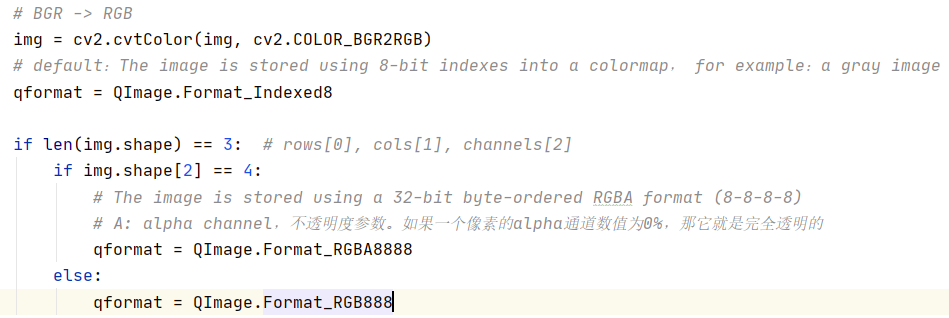
* 人脸识别
* 连接数据库
* 打开/关闭摄像头并显示照片

1. 打开摄像头与人脸检测相同
2. 关闭摄像头且将计时器停止：

self.timer.stop()

self.cap.release()

1. 由于在Opencv中，获取的Mat图像是BGR形式的，而我们平时所用的照片格式都为RGB，因此将图片转成可在Qlabel上显示的格式



1. 显示在Qlabel上并自适应大小

outImage = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)  
qlabel.setPixmap(QPixmap.fromImage(outImage))  
qlabel.setScaledContents(True) *# 图片自适应大小*

* 设备响铃报警

设备响铃主要用到Python中的playsound库，可以播放格式为.wav的音频。

winsound.PlaySound(**'./alarm.wav'**, winsound.SND\_FILENAME)

* 多线程

1. 为实现Opencv下进行的人脸识别、人脸跟踪等功能与设备响铃，语音识别、语音播报等功能不冲突，采取多线程措施。将Opencv线程单独执行。

class FaceProcessingThread(QThread):

1. 开始进程：

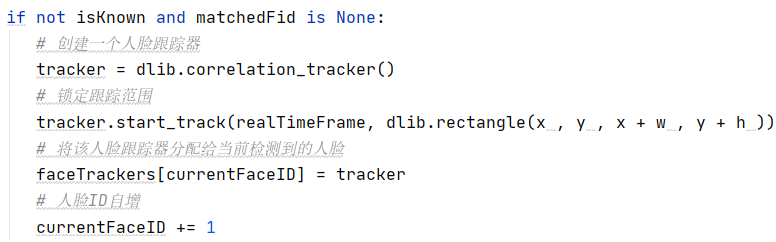
def run(self):

1. 结束进程：

def stop(self):

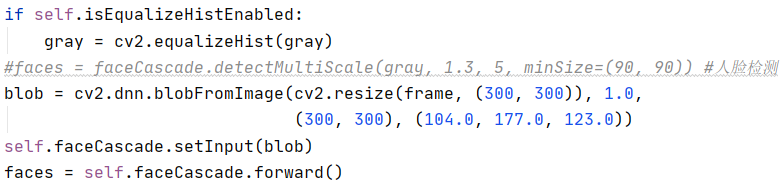
* 人脸跟踪：

1. 人脸跟踪在Opencv线程中(run()函数)实现
2. 人脸跟踪使用到Dlib开源库。
3. 关键代码：



* 人脸识别

1. 人脸识别使用的是LBPH识别器中人脸识别函数。
2. 首先仍要进行人脸检测：



1. 创建LBPH识别器并读取训练好的数据集：

if not isTrainingDataLoaded and os.path.isfile(CoreUI.trainingData):  
 recognizer = cv2.face.LBPHFaceRecognizer\_create() *#LBPH识别器* recognizer.read(CoreUI.trainingData)  
 isTrainingDataLoaded = True

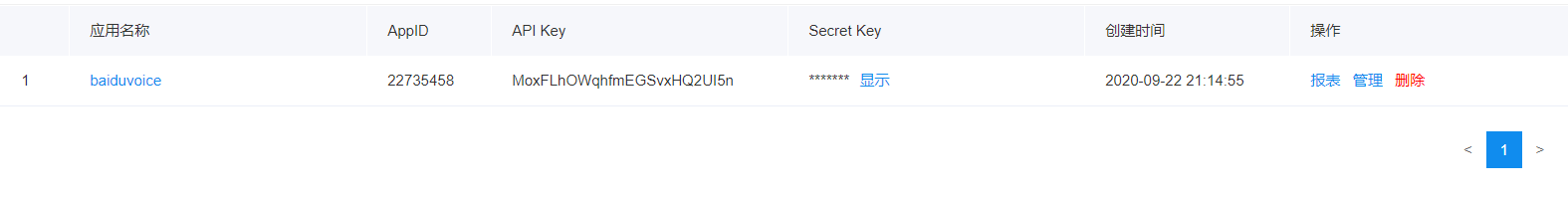
1. 进行人脸识别：

face\_id, confidence = recognizer.predict(gray[startY:endY, startX:endX])

* 语音识别与播报

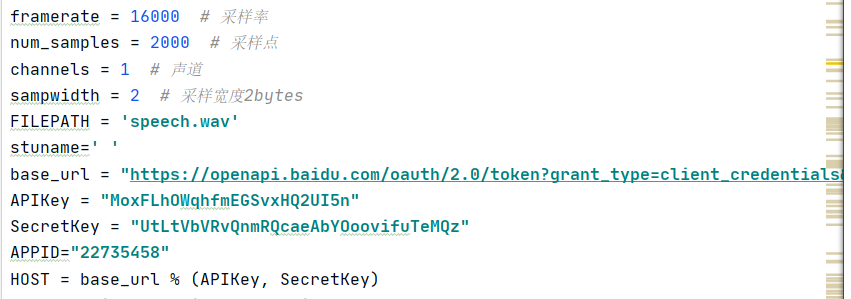
语音识别和播报都使用到了百度API，首先要在百度智能云中创建账号并创建应用。可以得到相应的APIKey,SecretKey,APPID。在Python中导入百度API的语音库。即可使用相应的功能。



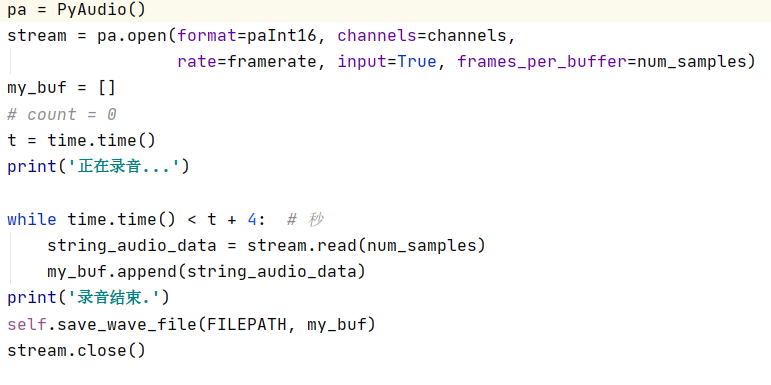


* 语音识别

1. 设置采样点，采样率，声道，采样宽度等参数。



1. 导入PyAudio音频处理库。创建一个音频流并将声音读入到音频文件中。



1. 将音频流数据解码存入data字典中，获取百度语音API的地址，将数据上传进行识别。

data = {  
 **'format'**: FORMAT,  
 **'rate'**: RATE,  
 **'channel'**: CHANNEL,  
 **'cuid'**: CUID,  
 **'len'**: len(speech\_data),  
 **'speech'**: SPEECH,  
 **'token'**: token,  
 **dev\_pid'**: dev\_pid  
}  
url = **'https://vop.baidu.com/server\_api'**headers = {**'Content-Type'**: **'application/json'**}  
*# r=requests.post(url,data=json.dumps(data),headers=headers)*print(**'正在识别...'**)  
r = requests.post(url, json=data, headers=headers)  
Result = r.json()  
if **'result'** in Result:  
 return Result[**'result'**][0]  
else:  
 return Result

* 语音播报

1. 将识别出的人脸对应的名字传入语音播报函数。
2. 语音播报函数关键代码：

client = AipSpeech(APPID, APIKey, SecretKey)

result = client.synthesis(voice, **'zh'**, 1, {**'vol'**: 8, **'per'**: 0})

1. 使用playsound播放mp3格式的音频。为使下次写入新的数据时能够再次写入并且播放，在每次播放完后移除mp3文件。

playsound(**'auido.mp3'**)  
os.remove(**'auido.mp3'**)

1. 使用face recognition库的人脸识别系统实现流程：

Face recognition库实现人脸识别与LBPH实现人脸识别不太相同的地方时前者不需要训练，它是一个封装好的库，存好照片和信息后直接进行识别即可。

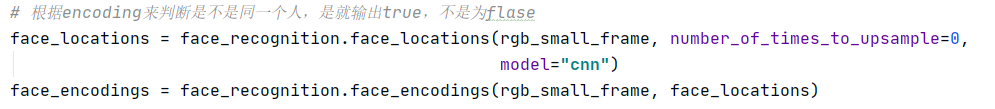
* 拍照（采集人脸数据）
* 与LBPH相同，连接数据库（新建的数据库）之后填写用户信息，再拍照，将照片按照用户的用户名命名存入指定路径的文件夹中。



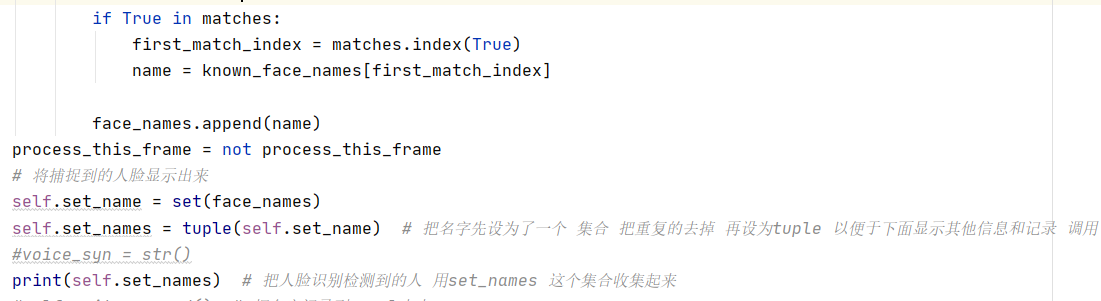
* 人脸识别
* 将文件夹中的人脸图片读取出来进行解码。



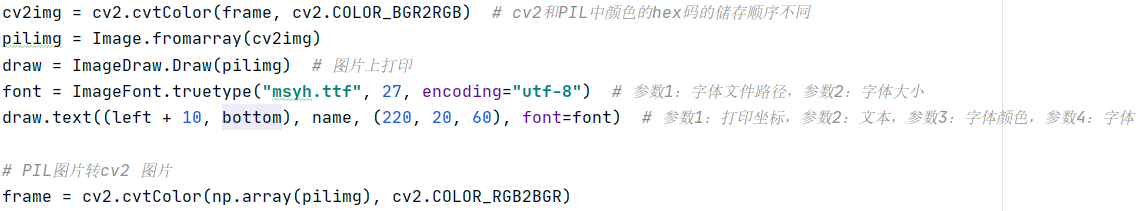
* 打开摄像头后捕捉人脸并将捕捉到的人脸照片与以解码的照片进行对比。



* 如果匹配，则将检测到的人收集到集合中。



* 由于Opencv本身的函数不能显示汉字，因此要进行转换（将汉字贴到图片上，再将照片进行显示）。



1. 界面设计（使用PyQt5工具）
2. 颜色：背景选定为灰色，文本字体选定为黑色。

原因：起初定白色底，但是当背景设置透明度时，白色底会与桌面串色，并不方便使用。因此选定偏灰色。

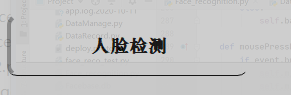
1. 字体：GroupBox上的标志性字体为华文琥珀，静态文本及按钮上的字体选择华文中宋。可以看起来不像宋体和黑体那样死板。主界面中的英文字体选择Algerian（英文字体）。







1. 圆边框的界面和按钮看起来比较柔和，简洁。
2. 将Mainwindow设置为透明，而在上的控件全部设置为半透明。可以增强科技感。
3. 设置按钮点下去与弹上来的动态效果。



1. 将界面传统的最大最小化按钮以及关闭按钮隐去，换为红黄绿替代的圆形按钮。（设置之后要添加相应的事件处理函数）



1. **实验总结**

通过这一次实验，我不但实践了Pyqt5的使用设置，也了解并比较了很多人脸识别与人脸检测的算法，并做了一定的总结：

1. 人脸检测：最初我的人脸检测精度很低，人脸检测框每次会框住脸部及其周边的很大一部分，导致训练效果减弱，识别精度也下降，实际上在Opencv中有很多人脸检测的算法，通过查阅资料，我将其分类总结：

* **Opencv Haar人脸检测：**

这是我最初使用的人脸检测方法，非常常规，但精度很低。

**优点：**

1）几乎可以在CPU上实时工作；

2）简单的架构；

3）可以检测不同比例的人脸。

**缺点：**

1）会出现大量的把非人脸预测为人脸的情况；

2）不适用于非正面人脸图像；

3）不抗遮挡。

* **OpenCV DNN 人脸检测：**

**优点：**

1）在这四种方法中是最准确的；

2）在CPU上能够实时运行；

3）适用于不同的人脸方向：上，下，左，右，侧面等。

4）甚至在严重遮挡下仍能工作；

5）可以检测各种尺度的人脸。

**缺点：**

基本上没有什么明显的缺点

* **Dlib HoG人脸检测：**

**优点：**

1）CPU上最快的方法；

2）适用于正面和略微非正面的人脸；

3）与其他三个相比模型很小；

4）在小的遮挡下仍可工作。

**缺点：**

1）不能检测小脸，因为它训练数据的最小人脸尺寸为80×80，但是用户可以用较小尺寸的人脸数据自己训练检测器；

2）边界框通常排除前额的一部分甚至下巴的一部分；

3）在严重遮挡下不能很好地工作；

4）不适用于侧面和极端非正面，如俯视或仰视。

* **Dlib CNN人脸检测：**

**优点：**

1）适用于不同的人脸方向；

2）对遮挡鲁棒；

3）在GPU上工作得非常快；

4）非常简单的训练过程。

**缺点：**

1）CPU速度很慢；

2）不能检测小脸，因为它训练数据的最小人脸尺寸为80×80，但是用户可以用较小尺寸的人脸数据自己训练检测器；

3）人脸包围框甚至小于DLib HoG人脸检测器。

1. 人脸识别：

人脸识别的算法更多了，通过查阅资料我分为以下几类：

* Opencv下经典算法：

1. LBPH人脸识别算法：LBPH是利用局部二值模式直方图的人脸识别算法。相比EigenFace和FisherFace，更多的是整数计算，优势是可以通过各种逻辑操作来进行优化，因此效率较高。且LBP特征对光照具有较好的鲁棒性。
2. FisherFace人脸识别算法：是一种基于LDA（线性判别分析）的人脸识别算法。LDA的中心思想是最大化类间距离，最小化类内距离。
3. EigenFace人脸识别算法：是一种基于统计特征的方法，将人脸图像视为随机变量，并用统计方法辨别不同人脸特征模式。EigenFace的基本思想是，从统计的观点，寻找人脸图像分布的基本元素，即人脸图像样本集协方差矩阵的特征向量，以此近似的表征人脸图像，这些特征向量成为特脸。

* Python中的face recognition开源库（利用深度学习但是封装好的库）：face\_recognitiond的原理模型是ResNet，ResNet网络具有29个转换层。
* 深度学习人脸识别算法

总结：这一次实验让我学到了很多非常有用的知识，这次没有用到深度学习。事实上，我认为使用深度学习也会非常有趣（奈何还是有很多看不懂）。