

**物联网应用课程设计**

课程名称： 物联网应用课程设计

设计名称： 智能家居系统

专业班级：物联网1502学号：2015003588

学生姓名： 刘港

指导教师： 郝晓丽

2019年 1月 8日

目录

[一、应用需求 2](#_Toc534966049)

[1.1视频监控现状 3](#_Toc534966050)

[1.2本系统所做优化 3](#_Toc534966051)

[1.3系统功能需求分析 3](#_Toc534966052)

[二、基本要求 3](#_Toc534966053)

[2.1人脸识别实现 3](#_Toc534966054)

[2.2目标检测 3](#_Toc534966055)

[2.3智能报警与录像 3](#_Toc534966056)

[三、 实验步骤 4](#_Toc534966057)

[3.1人脸数据生成 4](#_Toc534966058)

[3.2人脸识别 6](#_Toc534966059)

[3.3目标检测 10](#_Toc534966060)

[四、心得体会 15](#_Toc534966061)

**一、**应用需求

1.1视频监控现状

视频监控系统是利用视频技术探测并显示监控场景，并在此基础上做一定的功能

扩展形成完善的系统或网络。随着时代的发展，视频监控的方案也在不断地变化以适

应新的要求。传统的视频监控主要经历了模拟视频监控、半模拟半数字化视频监控以

及如今的数字网络化视频监控。由于传统视频监控需要安保人员时刻盯着监控画面，

费时费力，并且会出现分神的情况。除此之外，传统的视频监控需将数据全部存储下

来，造成很大的内存浪费，而且回放视频的时候也需要在海量的画面中寻找线索使得

率低下。为解决这些问题，需要利用计算机视觉技术将图像处理的相关算法结合到监

控系统中，发展更为智能的新一代视频监控系统。

1.2本系统所做优化

我们所做的智能家居加入了智能监控系统，不同于普通的监控，结合opencv实现

了很大的优化。

目前，安防视频监控都需要人为地去主动监控，一方面浪费了人力资源，另一方

面人会受到自身和周围环境的影响，而计算机的分析判断能力会避免这些问题的发生，

且之前介绍的几种传统的监控模式也存在价格和功耗等问题，本次实验针对这些问题

提出新的方案，主要利用了嵌入式价格低，功耗小等优点，将其作为服务器，与之对

应的 PC 客户端实现远程的视频监控，接着结合算法使得整个监控系统更加智能化。

1.3系统功能需求分析

（1）能够实时显示监控区域情况。

（2）能够通过远程网络访问。

（3）自动检测移动物体的存在。

（4）自动定位标记画面中的人脸并进行人脸识别。

（5）自动报警和智能录像。

（6）数据库的增删改查。

（7）人脸识别解锁房门

二、基本要求

2.1人脸识别实现

采用opencv haar特征检测+LBPH算法进行人脸识别，实现人脸解锁房门

2.2目标检测

启用监控功能后，可以对房间进行监控，采用帧差法实现目标检测

2.3智能报警与录像

结合人脸识别算法识别人脸，进而实现自动报警功能，识别陌生人后会录制近期画面，可自行查看。

三、 实验步骤

3.1人脸数据生成

**generate.py:**

def generate():

face\_cascade = cv2.Cascad import cv2

def generate():

face\_cascade = cv2.CascadeClassifier('./cascades/haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier('./cascades/haarcascade\_eye.xml')

camera = cv2.VideoCapture(0)

count = 0

while (True):

ret, frame = camera.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces:

img = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

f = cv2.resize(gray[y:y+h, x:x+w], (200, 200))

cv2.imwrite('./jm/%s.pgm' % str(count), f)

print (count)

count += 1

cv2.imshow("camera", frame)

if count==100:

break

if cv2.waitKey(1) & 0xff == ord("q"):

break

camera.release()

cv2.destroyAllWindows()

if \_\_name\_\_ == "\_\_main\_\_":

generate()eClassifier('./cascades/haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier('./cascades/haarcascade\_eye.xml')

camera = cv2.VideoCapture(0)

count = 0

while (True):

ret, frame = camera.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces:

img = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

f = cv2.resize(gray[y:y+h, x:x+w], (200, 200))

cv2.imwrite('./jm/%s.pgm' % str(count), f)

print (count)

count += 1

cv2.imshow("camera", frame)

if count==100:

break

if cv2.waitKey(1) & 0xff == ord("q"):

break

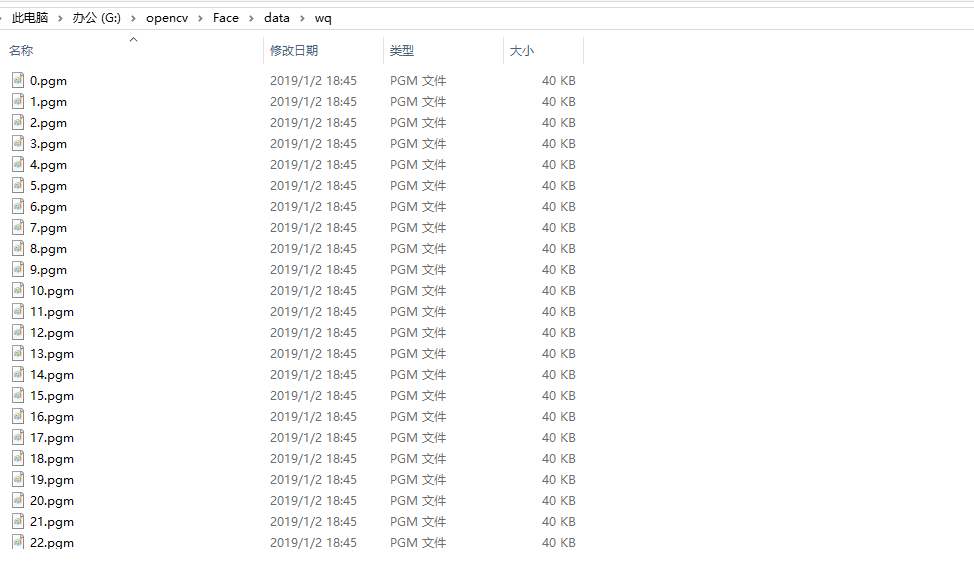
camera.release()

cv2.destroyAllWindows()

if \_\_name\_\_ == "\_\_main\_\_":

generate()

生成训练图片，放在对应人物的文件夹中：



查看指定目录下的训练图片，看有无效果较差的图片，效果较差的图片应该删掉，以免影响训练效果，用check.py 实现：

for i in range(100):

str='./data/gzk/%d.pgm' %(i)

img=cv2.imread(str)

str\_check='./data/check/%d.jpg' %(i)

cv2.imwrite(str\_check,img)

部分图片如下：



3.2人脸识别（卫乾负责）

**face\_rec.py:**

def read\_images(path,sz=None):

c=0

X,y=[],[]

for dirname,dirnames,filenames in os.walk(path):

for subdirname in dirnames:

subject\_path=os.path.join(dirname,subdirname)

for filename in os.listdir(subject\_path):

try:

if(filename==".directory"):

continue

filepath=os.path.join(subject\_path,filename)

im=cv2.imread(os.path.join(subject\_path,filename),cv2.IMREAD\_GRAYSCALE)

if(sz is not None):

im=cv2.resize(im,(200,200))

X.append(np.asarray(im,dtype=np.uint8))

y.append(c)

except IOError (errno,strerror):

print ("I/O error({0}): {1}".format(errno, strerror))

except:

print ("Unexcepted error:",sys.exc\_info()[0])

raise

c=c+1

return [X,y]

def conf\_web():

#连接服务器

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())#允许连接不在know\_hosts文件中的主机

ssh.connect(hostname='178.128.62.47', port=22, username='root', password='liugang666')

transport = paramiko.Transport(('hostname',22))

transport.connect(username='wupeiqi',password='123')

sftp = paramiko.SFTPClient.from\_transport(transport)

def face\_rec():

names=['gzk','wq']

ages=['8','20']

count=[0,0]

'''if len(sys.argv)<2:

print("USAGE: facerec\_demo.py </path/to/images>")

sys.exit()'''

#[X,y]=read\_images(sys.argv[1])

[X,y]=read\_images("G:/opencv/Face/data")

y=np.asarray(y,dtype=np.int32)

model=cv2.face.LBPHFaceRecognizer\_create()

model.train(np.asarray(X),np.asarray(y))

camera=cv2.VideoCapture(0)

face\_cascade=cv2.CascadeClassifier("G:/opencv/Face/cascades/haarcascade\_frontalface\_default.xml")

#face\_cascade=cv2.CascadeClassifier("G:/opencv/Face/cascades/haarcascade\_frontalface\_alt.xml")

#face\_cascade=cv2.CascadeClassifier("./train.xml")

WarningCount=0

BestMatch=0

while(True):

read,img=camera.read()

if cv2.waitKey(1)&0xff==ord("q"):

break

faces=face\_cascade.detectMultiScale(img,1.3,5)

for(x,y,w,h)in faces:

img=cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

roi=gray[x:x+w,y:y+h]

try:

roi=cv2.resize(roi,(200,200),interpolation=cv2.INTER\_LINEAR)

params=model.predict(roi)

print("Lable: %s,Confidence:%.2f" % (params[0],params[1]))

Str\_Waiting="Waiting..."

if params[1]<=50:

cv2.putText(img,"UnLock !",(x+w,y+80),cv2.FONT\_HERSHEY\_SIMPLEX,1,255,2)

cv2.putText(img,"name:"+names[params[0]],(x+w,y),cv2.FONT\_HERSHEY\_SIMPLEX,1,255,2)

cv2.putText(img,"age:"+ages[params[0]],(x+w,y+40),cv2.FONT\_HERSHEY\_SIMPLEX,1,255,2)

BestMatch=1

break

else :

WarningCount+=1

if WarningCount>=20 :

cv2.putText(img,"Warning!!!",(x,y-20),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),2)

else:

cv2.putText(img,Str\_Waiting,(x,y-20),cv2.FONT\_HERSHEY\_SIMPLEX,1,255,2)

except:

continue

if BestMatch==1:

break

cv2.imshow("Final",img)

cv2.waitKey(0)

cv2.destroyAllWindows()

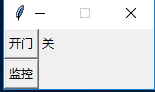
#连接数据库并发送解锁信息

return BestMatch

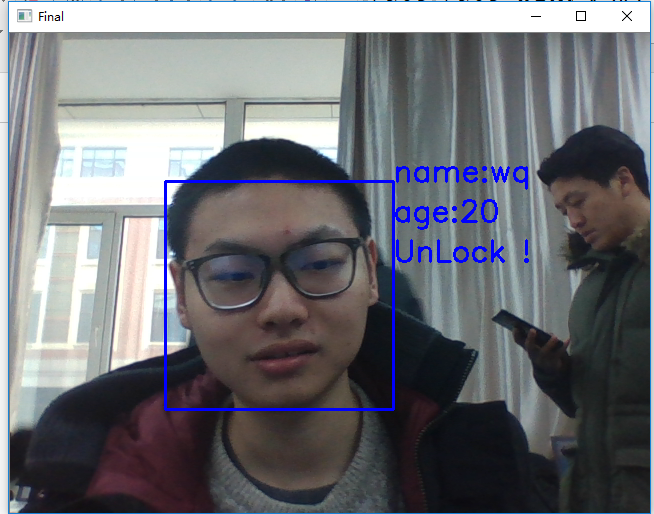
if \_\_name\_\_=="\_\_main\_\_":

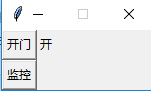
face\_rec()

用人脸识别来开门，初始门的状态为关：



点击开门按钮可以进行人脸识别，当置信度达到指定阈值后可以认为是可靠的识别，即开门：

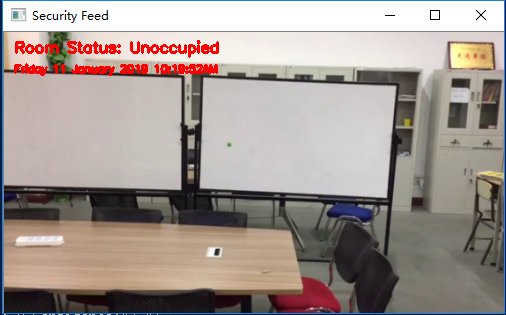




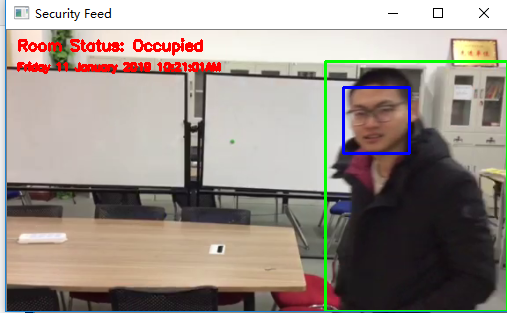
3.3目标检测(刘港负责)

初始时房间是空的，一个人进入房间后开始识别，框出移动的人，然后尝试识别此人的面部，将面部信息与录入的主人信息进行匹配，若匹配出为主人，则不报警，若匹配出为陌生人，则进行报警。

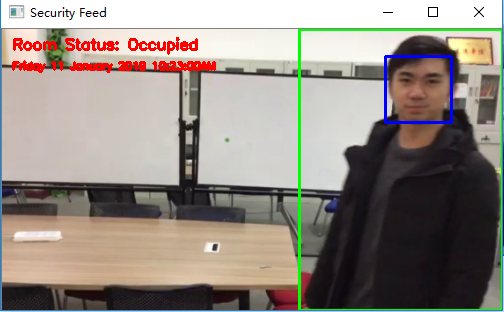
打开监视界面，此时为空房间



进入一个人，进行目标检测，并识别面部，经匹配为主人



之后进入另一个人，进行目标检测识别面部，判断为陌生人，并报警。





源代码：home.py

# construct the argument parser and parse the arguments

def read\_images(path,sz=None):

c=0

X,y=[],[]

for dirname,dirnames,filenames in os.walk(path):

for subdirname in dirnames:

subject\_path=os.path.join(dirname,subdirname)

for filename in os.listdir(subject\_path):

try:

if(filename==".directory"):

continue

filepath=os.path.join(subject\_path,filename)

im=cv2.imread(os.path.join(subject\_path,filename),cv2.IMREAD\_GRAYSCALE)

if(sz is not None):

im=cv2.resize(im,(200,200))

X.append(np.asarray(im,dtype=np.uint8))

y.append(c)

except IOError (errno,strerror):

print ("I/O error({0}): {1}".format(errno, strerror))

except:

print ("Unexcepted error:",sys.exc\_info()[0])

raise

c=c+1

return [X,y]

def face\_rec(roi):

face\_cascade=cv2.CascadeClassifier("G:/opencv/Face/cascades/haarcascade\_frontalface\_default.xml")

faces=face\_cascade.detectMultiScale(roi,1.3,5)

global alarm\_status

for(x,y,w,h)in faces:

img=cv2.rectangle(roi,(x,y),(x+w,y+h),(255,0,0),2)

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

roi=gray[y:y+h,x:x+w]

try:

roi=cv2.resize(roi,(200,200),interpolation=cv2.INTER\_LINEAR)

params=model.predict(roi)

print("Lable: %s,Confidence:%.2f" % (params[0],params[1]))

if params[1]<=100:

#给数据库数据，告知安全，家中的人是主人，alarm\_status=0

alarm\_status=0

print("master，safe")

else:

global waringcount

global out

if waringcount<10:

waringcount+=1

out.write(roi)

if waringcount==10:

# 将C:/Users/ice/Desktop/1/video/home.avi 上传至服务器/home.avi

sftp.put('C:/Users/ice/Desktop/1/video/home.avi', '/home.avi')

#给数据库数据，告知不安全，家中的人是陌生人，alarm\_status=1

alarm\_status=1

print("Uploaded video to server!!")

except:

continue

def conf\_web():

#连接服务器

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())#允许连接不在know\_hosts文件中的主机

ssh.connect(hostname='178.128.62.47', port=22, username='root', password='liugang666')

transport = paramiko.Transport(('hostname',22))

transport.connect(username='wupeiqi',password='123')

sftp = paramiko.SFTPClient.from\_transport(transport)

def main():

waringcount=0

alarm\_status=0

out=None

avg = None

motionCounter = 0

#训练文件，进行识别

[X,y]=read\_images("G:/opencv/Face/data")

model=cv2.face.LBPHFaceRecognizer\_create()

model.train(np.asarray(X),np.asarray(y))

camera = cv2.VideoCapture(0)

warnings.filterwarnings("ignore")

#视频输出设置

fourcc = cv2.VideoWriter\_fourcc(\*'XVID') #编码器

out = cv2.VideoWriter('C:/Users/ice/Desktop/1/video/home.avi',fourcc,20.0, (640,480)) #保存结果

print("[INFO] warming up...")

lastUploaded = datetime.datetime.now()

camera = cv2.VideoCapture("./final.mp4")

while(True):

grabbed,frame = camera.read()

if (grabbed is False):

print ("failed to grab frame.")

break

timestamp = datetime.datetime.now()

text = "Unoccupied"

# resize the frame, convert it to grayscale, and blur it

frame = imutils.resize(frame, width=500)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

gray = cv2.GaussianBlur(gray, (21, 21), 0)

# if the average frame is None, initialize it

if avg is None:

print("[INFO] starting background model...")

avg = gray.copy().astype("float")

continue

# accumulate the weighted average between the current frame and

# previous frames, then compute the difference between the current

# frame and running average

cv2.accumulateWeighted(gray, avg, 0.5)

frameDelta = cv2.absdiff(gray, cv2.convertScaleAbs(avg))

# threshold the delta image, dilate the thresholded image to fill

# in holes, then find contours on thresholded image

thresh = cv2.threshold(frameDelta, 5, 255,

cv2.THRESH\_BINARY)[1]

thresh = cv2.dilate(thresh, None, iterations=2)

image,cnts, \_= cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE)

# loop over the contours

for c in cnts:

# if the contour is too small, ignore it

if cv2.contourArea(c) < 7500:

continue

# compute the bounding box for the contour, draw it on the frame,

# and update the text

(x, y, w, h) = cv2.boundingRect(c)

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

roi=frame[y:y+h,x:x+w]

face\_rec(roi)

text = "Occupied"

# draw the text and timestamp on the frame

ts = timestamp.strftime("%A %d %B %Y %I:%M:%S%p")

cv2.putText(frame, "Room Status: {}".format(text), (10, 20),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 2)

cv2.putText(frame, ts, (10, 40), cv2.FONT\_HERSHEY\_SIMPLEX,

0.35, (0, 0, 255), 2)

cv2.imshow("Security Feed", frame)

key = cv2.waitKey(1) & 0xFF

if key == ord("q"):

break

#cv2.waitKey(0)

camera.release()

out.release()

cv2.destroyAllWindows()

return text

四、心得体会

通过此次课程设计，我学会了图像处理的相关技术，使我更加扎实的掌握了有关方面的知识，在设计过程中虽然遇到了一些问题，但经过一次又一次的思考，最终解决了问题，受益匪浅。