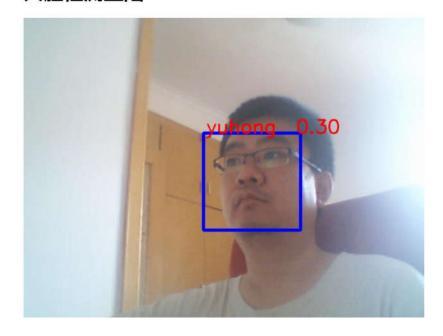


智能系统与控制



人脸检测登陆



树莓派网络控制:

人脸识别登陆

于泓 鲁东大学 信息与电气工程学院 2022.6.18

人工智能学院



任务: 设计一个基于人脸检测的登陆页面, 当出现允许登陆的人脸时, 可以进入系统, 实现小灯控制, 环境检测等功能。





人脸识别部分

(1) 在 main.py 中构造一个全局变量 存储人脸识别所需的 识别器 存储的人脸特征 等

```
print("加载人脸检测器")
det face = dlib.get frontal face detector()
# 加载标志点检测器
det landmark = dlib.shape predictor("shape predictor 68 face landmarks GTX.dat") # 68点5
sp = dlib.shape predictor("shape predictor 5 face landmarks.dat") # 5点
# 加载人脸特征提取器
facerec = dlib.face recognition model v1("dlib face recognition resnet model v1.dat")
# 加载训练好的人脸模型
model = np.load('trainer.npz')
face vectors = model['face vectors']
face ids = model['ids']
opt face = dict()
opt face['detector'] = det face # 人脸检测器
opt face 'recongnizer'] = facerec # 人脸特征提取器
                              # 关键点提取器
opt face['sp']=sp
opt face ['face ids'] = face ids # 已存储的人脸id
opt face 'face vectors'] = face vectors # 已存储的人脸矢量
opt face ['face allowed'] = ['yuhong'] # 允许通过认证的人脸ID
                                                                           人脸检测标志位
                                    # 认证成功标志_____
opt face['flag'] = False
```



(2) 构造生成器 实现视频显示和 人脸检测

```
视频
def gen_face_recognition(camera,opts)
    detector = opts['detector']
    sp = opts['sp']
    facerec = opts['recongnizer']
    face vectors = opts['face vectors']
    face ids = opts['face ids']
    face allowed = opts['face allowed']
     while True:
         img = camera.get frame()
        if imq is None:
            time.sleep(0.1)
             continue
         # BGR 转 gray
         img gray = cv2.cvtColor(img, cv2.COLOR BGR2GRAY)
         # BGR 转 RGB
         img rgb = cv2.cvtColor(img, cv2. COLOR BGR2RGB)
         # 进行人脸检测
        dets = detector(img gray, 1)
```

人工智能学院



```
# 遍历检测的人脸
for k, d in enumerate(dets):
    # 画框
   cv2.rectangle(img, (d.left(), d.top()), (d.right(), d.bottom()), (255, 0, 0), 3)
    # 标志点检测
   shape = sp(img rgb, d)
    # 获取人脸特征
    face vector = facerec.compute face descriptor(img rgb, shape)
    # # 进行识别返回ID与距离
    face id, dis = face recognize(np.array(face vector), face vectors, face ids)
   if (dis < 0.45):
       str out = "%s %.2f"%(face id,dis)
       if face id in face allowed:
           opts['flag'] = True
    else:
       str out = "unknown %.2f"%(dis)
       # 检测结果文字输出
       cv2.putText(img, strout, (d.left()+5, d.top()), cv2.FONT HERSHEY SIMPLEX, 1, (0,0,255), 2)
    ret, jpeg = cv2.imencode('.jpg',img)
    # 对图像进行编码输出
   yield(b'--frame\r\n'+b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n\r\n')
```



main.py 构建路由

```
@app.route('/',methods=['GET', 'POST'])
                                                    主页面
□def face():
    if request.method == 'GET':
        opt face['flag'] =False
        return render template('video face.html', videourl=url for('video face rec'))
                                                  主页面中视频显示部分
@app.route('/video face rec')
□def video face rec():
    return Response (gen face recognition (m camera, opt face), mimetype='multipart/x-mixed-replace; boundary=frame')
@app.route('/checkFace', methods=['GET', 'POST']) —
                                                                 变量检测
□def checkFace():
                                                                 查看是否检测到人脸
    print(opt face['flag'])
    if opt face['flag'] == True:
        print("跳转")
        return jsonify({'flag':'True'})
    else:
        return jsonify({'flag':'Flase'})
```

人工智能学院



```
<!DOCTYPE html>
=<head>
    <!-- <meta http-equiv="refresh" content="1" charset="UTF-8"> -->
    <title>人脸检测登陆</title>
</head>
 <script>
    function getFlag() {
                var xmlHttp = new XMLHttpRequest();
                var baseUrl = "/checkFace";
                xmlHttp.open( "GET", baseUrl, false);
                xmlHttp.send( null );
                return xmlHttp.responseText;
            }
     function changeUrl() {
                var Ino Flag = getFlag();
                var json = JSON.parse(Ino Flag);
                if (json.flag == "True")
                       window.location.assign("/light")
            }
     function addTimer() {
                setInterval(changeUrl, 5000);
    </script>
```