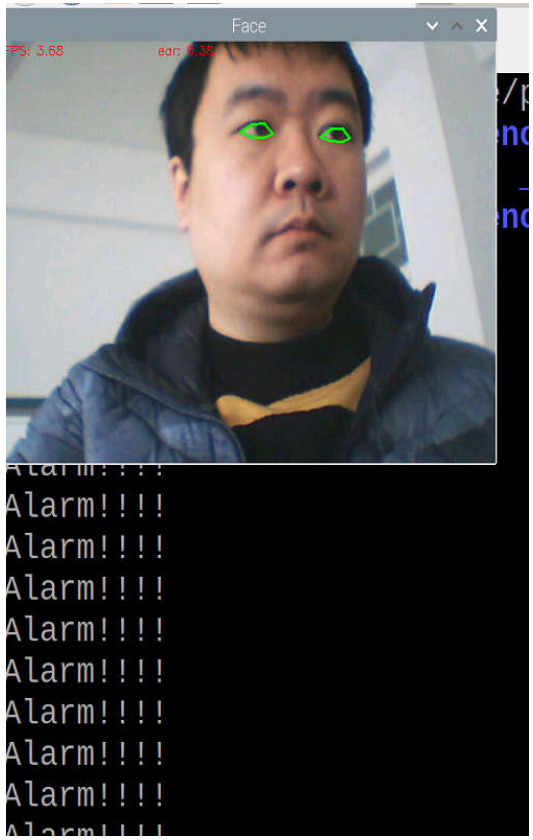
智能系统与控制

树莓派：OpenCV 闭眼检测



于泓
鲁东大学
信息与电气工程学院
2022.2.15

实验任务

- （1）检测是否闭眼
- （2）如果闭眼的时间过长，驱动蜂鸣器报警

闭眼检测的基本原理

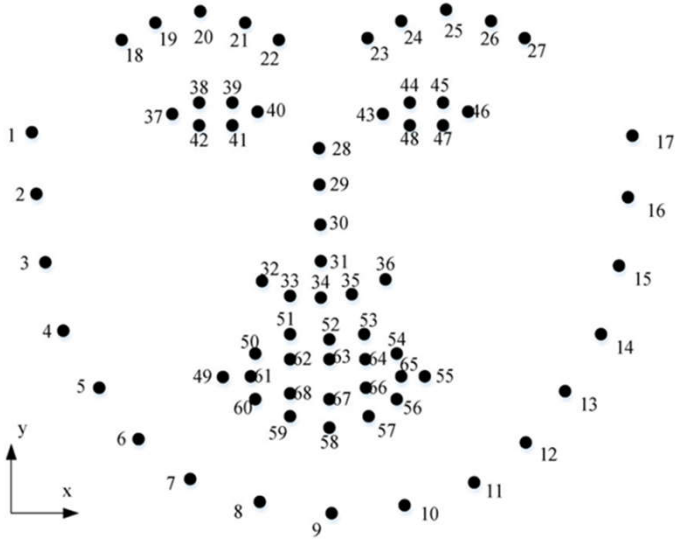
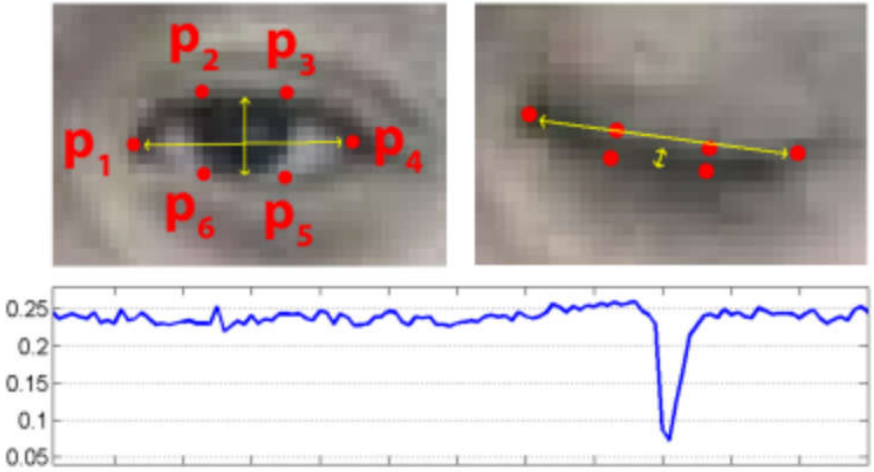


图 2.13: 人脸的 68 个标志点



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

EAR 小于某个阈值表示闭眼

```python
import cv2
import dlib
import numpy as np
from pin_dic import pin_dic
import RPi.GPIO as GPIO


def eye_aspect_ratio(pts):
    A = np.sqrt(np.dot(pts[1]-pts[5],pts[1]-pts[5]))
    B = np.sqrt(np.dot(pts[2]-pts[4],pts[2]-pts[4]))
    C = np.sqrt(np.dot(pts[0]-pts[3],pts[0]-pts[3]))

    ear = (A+B)/(2.0*C)

    return ear
```

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

```
if __name__ == "__main__":

    # 蜂鸣器初始化
    pin_sig = pin_dic['G16']

    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(pin_sig, GPIO.OUT)
    GPIO.output(pin_sig, GPIO.HIGH)

    # 创建人脸检测器
    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.dat")  # 68点

    # 打开摄像头
    cap = cv2.VideoCapture(0)
    freq = cv2.getTickFrequency() # 系统频率

    # 闭眼阈值
    th_ear = 0.21

    # 闭眼时间阈值
    th_count = 10
```

蜂鸣器，低电平驱动

判定是否闭眼

闭眼时间的阈值

```python
while True:
    # 读取一帧图像
    success, img = cap.read()

    # 转换为灰度
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    t1 = cv2.getTickCount()
    # 检测人脸区域
    face_rects = det_face(gray, 0)

    for ret in face_rects:

        # 标志点检测
        landmarks = det_landmark(gray, ret)

        # 遍历所有关键点
        pts = []
        for part in landmarks.parts():
            pts.append((part.x,part.y))

        index_eye1 = [36,37,38,39,40,41]
        index_eye2 = [42,43,44,45,46,47]

        ear1 = eye_aspect_ratio(np.array(pts)[index_eye1])
        ear2 = eye_aspect_ratio(np.array(pts)[index_eye2])

        cv2.polylines(img, [np.array(pts)[index_eye1]], True, (0, 255, 0), 2)
        cv2.polylines(img, [np.array(pts)[index_eye2]], True, (0, 255, 0), 2)

        ear = (ear1+ear2)/2

        cv2.putText(img, 'ear: %.2f'%(ear), (200, 15), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,255))
```

关键点读取

计算两眼的EAR

```
        if ear < th_ear:
            count = count +1

        else:
            count = 0


        if count > th_count:
            print('Alarm!!!!')
            GPIO.output(pin_sig, GPIO.LOW)
        else:
            GPIO.output(pin_sig, GPIO.HIGH)


    t2 = cv2.getTickCount()

    fps = freq/(t2-t1)
    # 显示速度
    cv2.putText(img, 'FPS: %.2f'%(fps), (0, 15), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,255))

    # 显示检测结果
    cv2.imshow("Face",img)

    # 按q退出
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
GPIO.cleanup()
```

如果闭眼，count+1

count过大就打印信息
并驱动蜂鸣器报警