工具：Jupyter，内核为Python3.12

题目：结合视频播放与图片人脸识别，对一段视频进行人脸识别

我们先导入opencv库、操作系统接口模块和NumPy函数库

1. **import** cv2
2. **import** os
3. **import** numpy as np

然后我们创建一个人脸识别器对象，并加载我们训练好的识别器（详情请见识别器训练说明）

1. recogizer=cv2.face.LBPHFaceRecognizer\_create()
2. recogizer.read("D:/trainer/trainer.yml")

接着我们定义一个名为FaceDetection的函数

1. def FaceDetection(img):

在函数内部，我们先用cv2.cvtColor函数将图片转化为灰度图像，以便减少计算量

1. gary = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

紧接着我们用cv2.CascadeClassifier函数导入Haar级联分类器，加载至FaceDetect这个变量中（注意修改地址）

1. FaceDetect = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")

然后我们使用detectMultiScale函数来检测人脸，导入灰度图像，其他参数我们则保持默认

1. face = face\_detect.detectMultiScale(gary)

接着我们用cv2.rectangle函数在图片上画矩形

1. **for** x,y,w,h in face:
2. cv2.rectangle(img,(x,y),(x+w,y+h),color=(0,0,255),thickness=1)
3. cv2.rectangle(img,(x-10,y-10),(x+w+10,y+h+10),color=(0,255,0),thickness=1)

然后我们用predict函数评估置信度

1. ids,confidence=recogizer.predict(gary[y:y+h,x:x+w])

然后我们根据置信评分来判断是否匹配，如果是就在矩形框上打出姓名

1. **if** confidence > 80:
2. cv2.putText(img, 'kiki', (x + 10, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 255, 0), 1)

最后我们返回图像

1. cv2.imshow('result', img)

完整的函数为：

1. def FaceDetection(img):
2. gary = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
3. face\_detect = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")
4. face = face\_detect.detectMultiScale(gary)
5. **for** x,y,w,h in face:
6. cv2.rectangle(img,(x,y),(x+w,y+h),color=(0,0,255),thickness=1)
7. ids,confidence=recogizer.predict(gary[y:y+h,x:x+w])#confidence为置信评分
8. **if** confidence > 80:
9. cv2.putText(img, 'kiki', (x + 10, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 255, 0), 1)
10. cv2.imshow('result', img)

然后我们用cv2.VideoCapture函数导入预先准备好的视频（注意修改地址）

1. vc=cv2.VideoCapture("D:/VID\_20231121\_192509.mp4")

随即我们用isOpened函数来判断视频初始化是否成功，成功返回True，不成功返回False

1. **if** vc.isOpened():
2. open, frame=vc.read()
3. **else**:
4. open=False

当open为True的时候，我们执行一个循环，并用vc.read函数读取当前帧

1. **while** open:
2. ret,frame=vc.read()

当视频结束时，break退出循环

1. **if** frame **is** None:
2. **break**

我们用img来接收被裁剪掉一半高度和宽度的图像（注：原视频高为480，宽为800）

1. **if** ret == True:
2. img=frame[120:360,200:600]

接着我们调用FaceDetection函数，将img作为参数

1. FaceDetection(img)

我们用cv2.waitKey来设置帧率，此处设为25（即为40帧），当用户按下q时退出循环，当按下空格时，使delay为0，视频暂停，并按任意键继续

1. **if** cv2.waitKey(25) == 113:
2. **break**
3. **if** cv2.waitKey(25) == 32:
4. cv2.waitKey(0)

由于我们在FaceDetection函数中已经imshow了，所以我们最后直接写上释放空间并删除窗口的代码

1. vc.release()
2. cv2.destroyAllWindows()

完整的代码为:

1. import cv2
2. import os
3. import numpy as np
4. recogizer=cv2.face.LBPHFaceRecognizer\_create()#创建一个人脸识别器对象
5. recogizer.read("D:/trainer/trainer.yml")#加载识别器
6. def FaceDetection(img):
7. gary = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
8. face\_detect = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")
9. face = face\_detect.detectMultiScale(gary)
10. **for** x,y,w,h in face:
11. cv2.rectangle(img,(x,y),(x+w,y+h),color=(0,0,255),thickness=1)
12. ids,confidence=recogizer.predict(gary[y:y+h,x:x+w])#confidence为置信评分
13. **if** confidence > 80:
14. cv2.putText(img, 'kiki', (x + 10, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 255, 0), 1)
15. cv2.imshow('result', img)
16. vc=cv2.VideoCapture("D:/VID\_20231121\_192509.mp4")
17. **if** vc.isOpened():
18. open, frame=vc.read()
19. **else**:
20. open=False
21. **while** open:
22. ret,frame=vc.read()
23. **if** frame is None:
24. **break**
25. **if** ret == True:
26. img=frame[60:300,200:600]
27. FaceDetection(img)
28. **if** cv2.waitKey(25) == 113:
29. **break**
30. **if** cv2.waitKey(25) == 32:
31. cv2.waitKey(0)
32. vc.release()
33. cv2.destroyAllWindows()