监控程序代码

实现功能

监控显示 -- 活体检测 -- 发送 15 秒的活动录像

解决点

使用子程序检查视频帧活体 - 使主程序流畅
通过对是否获取新录像、是否继续录像进行判断解决重复录像

1 路径设置

In [ ]:
```
paths = '/home/pi/luo_video/'
```

2 邮件发送

In [ ]:
```
def sendemail(text,subject,path):
            print('read for      send {0}'.format(path))
            import os
            import smtplib
            from email.mime.text import MIMEText
            from email.mime.image import MIMEImage
            from email.mime.multipart import MIMEMultipart
            _user='@qq.com'
            _pwd=''
            _to='7@qq.com'
            msg = MIMEMultipart('related')
            content=MIMEText('<html><body><table><tr><td style="font-size:1.5em;color:wh
            #img = MIMEImage(img_data)
            #img.add_header('Content-ID', 'imageid')
            #msg.attach(img)
            msg.attach(content)
            filename = path.split('/')[-1]
            basename = os.path.basename(filename)
            #txt
            h = open(path,'rb')
            content2 = h.read()
            part2 = MIMEText(content2,'plain','utf-8')
            #
```

```
    part2['Content-Type'] = 'application/octet-stream'
    #
    part2['Content-Disposition'] = 'attachment;filename=%s' % basename
    #
    # part2.add_header('Content-Disposition', 'attachment', filename=('utf-8', '', bas
    msg['From']=_user
    msg['to']=_to
    msg['Subject'] = subject
    msg.attach(part2)
    print('begin')
    try:
        s=smtplib.SMTP_SSL('smtp.qq.com',465)
        s.login(_user,_pwd)
        print('login success')
        s.sendmail(_user,_to,msg.as_string())
        s.quit()
        print('success')
    except smtplib.SMTPException as e:
        print(e)

def sendemails(pathn,now1,listm):
    print('#'*30)
    print('restart minute')
    print(listm)
    restart = 0
    mm='100'
    now1,now,m = formattime()
    offnow = now
    offnow1= now1
    offpathn = paths+'{0}.avi'.format(offnow)
    print('{0} '.format(now))
    while True:
        now1,now,m = formattime()
        if int(m) in listm and int(m)!=int(mm):
            print('')
            now1,now,mm = formattime()
            restart = 1

        if restart == 1:
            time.sleep(10)
            print('')
            try:
                print(offpathn)
                sendemail('from email',offnow1,offpathn)
                offnow1 = now1
                offpathn = paths+'{0}.avi'.format(offnow)
                restart = 0
            except:
                print('')
```

3 格式化时间

```
In [ ]: def formattime():
            import time
            localtime = time.localtime(time.time())
            now1 = time.strftime("%Y-%m-%d %H:%M:%S", localtime )
            now = time.strftime("%Y%m%d%H%M", localtime )
            m = time.strftime("%M", localtime )
            return now1,now,m
```

4 活体检测 - 发送 - 子程序

```
In [ ]: def check(mydict):

            #gray_pic = mydict['gray_pic']
            pre_frame = mydict['pre_frame']
            #pre_frame = gray_pic
            new=time.time()
            new1=time.time()
            num = 0
            while True:

                #gray_pic = mydict['gray_pic']
                start = mydict['start']
                frame = mydict['frame']

                numsend = mydict['numsend']
                numsave = mydict['numsave']
                t = mydict['t']
                gray_pic = mydict['gray_pic']
                #gray_pic = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                if mydict['close']==1:
                    break
                if start is None or frame is None  or gray_pic is None:
                    print('')
                    time.sleep(1)
                    continue
                else:

                    end = time.time()
            # ,5FPS
                    seconds = end - start
            #if seconds < 1.0 / fps:
            #    time.sleep(1.0 / fps - seconds)
```

```
        gray_pic = cv2.resize(gray_pic, (480, 480))
#
        gray_pic = cv2.GaussianBlur(gray_pic, (21, 21), 0)
#



        if pre_frame is None:

            pre_frame = gray_pic
        else:
            pre_frame = cv2.resize(pre_frame, (480, 480))
        #
            #pre_frame = cv2.GaussianBlur(pre_frame, (21, 21), 0)
        # absdiff


            img_delta = cv2.absdiff(pre_frame, gray_pic)


        # threshold(,,,)
            thresh = cv2.threshold(img_delta, 30, 255, cv2.THRESH_BINARY)[1]
        #
            thresh = cv2.dilate(thresh, None, iterations=2)
            pre_frame = gray_pic
        # findContours(,,)
            try:
                img ,contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR
            except:
                contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_EXT
            for c in contours:
            #
            # contourArea

                if cv2.contourArea(c) <    mydict['flex']:
                    print(cv2.contourArea(c))
                    continue
                else:
                    sys.stdout.write('------>{0}\r'.format(cv2.contourArea(c)))
                    sys.stdout.flush()

                #print("")

                    TI = time.strftime('%Y-%m-%d-%H-%M-%S', time.localtime(time.t

                    text = 'pepole move in'+TI
                    mydict['getvideo']='t'
```

```
                        #mydict['getvideo'] = '1'
                        cv2.putText(frame, text, (150, 15), cv2.FONT_HERSHEY_SIMPLEX,
                    #

                        pathsimg=paths + "JC"+TI+ '.jpg'

                if mydict['send'] == '1':
                    print('')

                    try:
                        cv2.imwrite(pathsimg, frame)
                    except:
                        TI = time.strftime('b-%Y-%m-%d-%H-%M-%S', time.localtime(tim
                    new1 = time.time()
                    if True:

                        try:

                            sendemail(platform.node(),TI,mydict['sendpath'])
                            mydict['send'] = '0'
                        except:
                            mydict['send'] = '0'
                    break
```

5 -录像 - 显示 - 主程序

```
In [ ]: from multiprocessing import Process
        import multiprocessing
        import time
        #import dlib
        import cv2
        import sys
        import platform

        if __name__=='__main__':
            caps = 0#'http://admin:admin@192.168.1.106:8081/'
            t=1 #
            savevideo = 0  #
            mydict=multiprocessing.Manager().dict()
            mydict['pre_frame'] = None
            mydict['gray_pic'] = None
            mydict['start'] = None
            mydict['frame'] =None
            mydict['t'] = t
            mydict['close'] = 0
            mydict['text'] = ''
```

```
    mydict['flex'] = 1  #
    mydict['numsend'] = 20#
    mydict['numsave'] = 20#
    mydict['send'] = '0'
    mydict['getnew']= '1'
    mydict['getvideo'] = 't'
    listm=[0,10,20,30,40,50]
    fps = 5
    pre_frame = None
    getbegin = 0
    h = open(paths+'open','w')
    h.close()
    print('')
    pathn = '1'
    now1 = '1'
    p = Process(target=check,args=(mydict,))
    p.start()
    if type(caps) ==str:
        time.sleep(3)
    cap = cv2.VideoCapture(caps)

    #'http://admin:admin@10.93.146.109:8081')  #
#

    fourcc = cv2.VideoWriter_fourcc(*"DIVX")
    cap.set(4,640)
    cap.set(3,640)
    a=int(cap.get(3))
    b=int(cap.get(4))
    print(a,b)
    num=0
    mm='1'
    a=int(cap.get(3))
    b=int(cap.get(4))
    start = None
    frame = None
    t = None
    gray_pic = None


    while(cap.isOpened()):
        s = int(time.strftime("%S", time.localtime(time.time()) ))
        ret, frame = cap.read()
        start = time.time()


        if ret:
            gray_pic = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            now1,now,m = formattime()
```

```
if mydict['getvideo'] == 't' :
    sys.stdout.write('{0}{1}\r'.format('                                        ',now1))
    sys.stdout.flush()



    if mydict['getnew'] == '1' :
        now1,now,mm = formattime()
        mydict['getpath'] = getpath = paths+'{0}.avi'.format(now)
        out = cv2.VideoWriter(getpath, fourcc, 15, (a,b))
        mydict['getnew'] = '0'
        getbegin = time.time()

    else:
        out.write(frame)

    if time.time()-getbegin>19:
        print(time.time()-getbegin)

        out.release()

        mydict['sendpath'] = getpath
        mydict['send'] = '1'
        mydict['getnew']= '1'
        mydict['getvideo'] = '0'

if savevideo == 1:
    if int(m) in listm and int(m)!=int(mm):
        try:
            print('')
            now1,now,mm = formattime()
            out.release()
            print('')
            out = cv2.VideoWriter(paths+'{0}.avi'.format(now), fourcc, 20, (
            #restart = '1'
            pathn = paths+'{0}.avi'.format(now)
            #h = open('video.log','w')
            #h.write(pathn +'@'+str(restart)+'@'+now1)
            #h.close()
            print(pathn)

        except:
            pass
    if num == 0:
        out = cv2.VideoWriter(paths+'{0}.avi'.format(now), fourcc, 20, (a,b)
        pathn = paths+'{0}.avi'.format(now)
        #h = open('video.log','w')
```

```
                #h.write(pathn +'@'+'0'+'@'+now1)
                #h.close()
                print(pathn)
                num+=1



        mydict['start']=start
        mydict['gray_pic']=gray_pic
        mydict['frame'] = frame


        # absdiff

        #check(pre_frame,gray_pic,start)
        mydict['pre_frame'] = gray_pic


        #frame = face_find(frame)
        frame = cv2.putText(frame, now1, (0, 15), cv2.FONT_HERSHEY_SIMPLEX, 0.4,
        frame = cv2.putText(frame, mydict['text'], (150, 15), cv2.FONT_HERSHEY_SIM
        if    savevideo ==1:
            out.write(frame)
        cv2.imshow('frame',frame)
    # q
        if cv2.waitKey(1) & 0xFF == ord('q'):
            restart=2
            #h = open('video.log','w')
            #h.write(pathn +'@'+str(restart)+'@'+now1)
            #h.close()
            cap.release()
            #out.release()
            cv2.destroyAllWindows()
            mydict['close']=1
            break
    else:
         break
print('')
```