AutoDriving：无人机项目第一阶段报告

Author：邓琛龙　2018202077

Group ID：7

注：第一阶段进行时间：2020.10.9—2020.10.30

一、第一阶段目标简述

1.1. 购买DJI Tello无人机，学习掌握Tello SDK以及相关接口

购买Tello无人机后，学习相关SDK以及接口使用，尽量做到熟悉相应操控接口。

1.2. 实现简单语音识别，并利用其操控无人机执行简单操作

利用相关语音识别引擎或自训练分类技术，使无人机能够听懂简单指令并作出相关动作。

1.3. 实现视频流处理，能检测追踪简单目标

利用Tello上自带的摄像头向电脑终端传输图像，并在电脑上实现图像处理，返回相应指令操控无人机。

二、第一阶段技术路线

2.1. 语音路线

我们首先利用电脑的麦克风或耳机进行收音，并将一段时间内的音频进行采样处理。

```
def record():
    CHUNK = 1024
    FORMAT = pyaudio.paInt16          #量化位数
```

```
    CHANNELS = 1                        #采样管道数
    RATE = 16000                        #采样率
    RECORD_SECONDS = 2
    WAVE_OUTPUT_FILENAME = "output.wav" #文件保存的名称
    p = pyaudio.PyAudio()               #创建PyAudio的实例对象
    stream = p.open(format=FORMAT,      #调用PyAudio实例对象的open方法创建流Stream
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    frames_per_buffer=CHUNK)
    frames = []                         #存储所有读取到的数据
    print('* 开始录音 >>>')             #打印开始录音
    for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
        data = stream.read(CHUNK)       #根据需求，调用Stream的write或者read方法
        frames.append(data)
    print('* 结束录音 >>>')             #打印结束录音
    stream.close()                      #调用Stream的close方法，关闭流
    p.terminate()                       #调用pyaudio.PyAudio.terminate() 关闭会话
    wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')   #写入wav文件里面
    wf.setnchannels(CHANNELS)
    wf.setsampwidth(p.get_sample_size(FORMAT))
    wf.setframerate(RATE)
    wf.writeframes(b''.join(frames))
    wf.close()
```

然后将录制结束的音频文件output.wav批量转送到百度开放平台的语音处理系统中，进行识别处理，进一步得到我们的语音指令内容

```
def cognitive():                            #读取文件
    def get_file_content(filePath):
        with open(filePath, 'rb') as fp:
            return fp.read()

    result = client.asr(get_file_content('output.wav'), 'wav', 16000, {
        'dev_pid': 1537,                    #识别本地文件
    } )
    result_text = result["result"][0]

    print("you said: " + result_text)

    return result_text
```

下一步便可以利用解析出来的指令，向Tello发送相关指令信号，指引其执行相关操作：

```
def action():
    if result == "开始。":
        '''
        command = "command"
        send ( command )
        '''
        print("开始")
    if result == "起飞。":
        #send("takeoff")
        drone.takeoff()
        print("tello无人机起飞")

    ......
```

2.2. 视频路线

视频路线我们主要想法是使用opencv中相关方法，对Tello返回的图像进行处理，并检测目标（如人脸）做出相应的动作。

首先我们需要从Tello中取回视频流的关键帧：

```
 # 读入Tello图像
def telloGetFrame(myDrone, w= 360,h=240):
    myFrame = myDrone.get_frame_read()
    myFrame = myFrame.frame
    img = cv2.resize(myFrame,(w,h))
    return img
```

然后我们利用已经训练好的人脸识别模型，对关键帧中的图像数据进行人脸识别操作，并返回人脸的坐标位置：

```
def findFace(img):
    faceCascade =
cv2.CascadeClassifier('./face_recognition/haarcascade_frontalface_default.xml'
)
    imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(imgGray,1.1,6  )

    myFaceListC = []
    myFaceListArea = []

    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
        cx = x + w//2
        cy = y + h//2
```

```
        area = w*h
        myFaceListArea.append(area)
        myFaceListC.append([cx,cy])

    if len(myFaceListArea) !=0:
        i = myFaceListArea.index(max(myFaceListArea))
        return img, [myFaceListC[i],myFaceListArea[i]]
    else:
        return img,[[0,0],0]
```

下一步可以通过查找和上一次分析中人脸图像位置的差，来控制Tello飞行的转向、运行速度，以达到追踪人脸的目的。

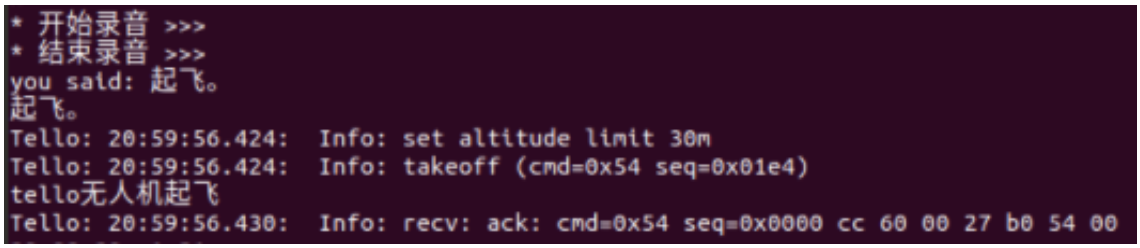
```
 # 追踪人脸
def trackFace(myDrone,info,w,pid,pError):

    ## PID
    error = info[0][0] - w//2
    speed = pid[0]*error + pid[1]*(error-pError)
    speed = int(np.clip(speed,-100,100))


    print(speed)
    if info[0][0] !=0:
        myDrone.yaw_velocity = speed
    else:
        myDrone.for_back_velocity = 0
        myDrone.left_right_velocity = 0
        myDrone.up_down_velocity = 0
        myDrone.yaw_velocity = 0
        error = 0
    if myDrone.send_rc_control:
        myDrone.send_rc_control(myDrone.left_right_velocity,
                                myDrone.for_back_velocity,
                                myDrone.up_down_velocity,
                                myDrone.yaw_velocity)
    return error
```

三、阶段性成果展示

语音方面，已经能够实现对所有常用命令的识别和操控，样例语音识别结果如下：



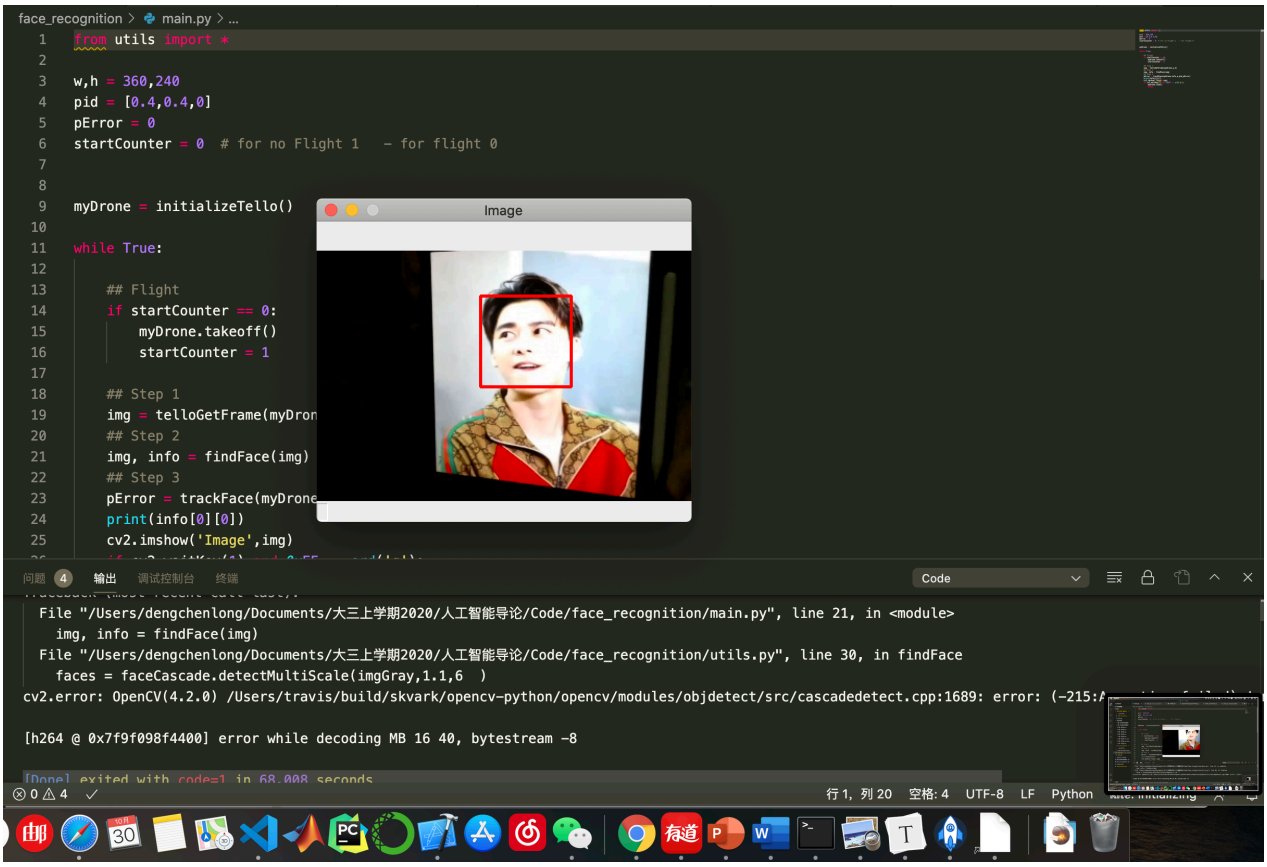
img1:起飞相应命令的识别、执行

并且，我们录制了相应视频展示。视频存放位置为百度网盘

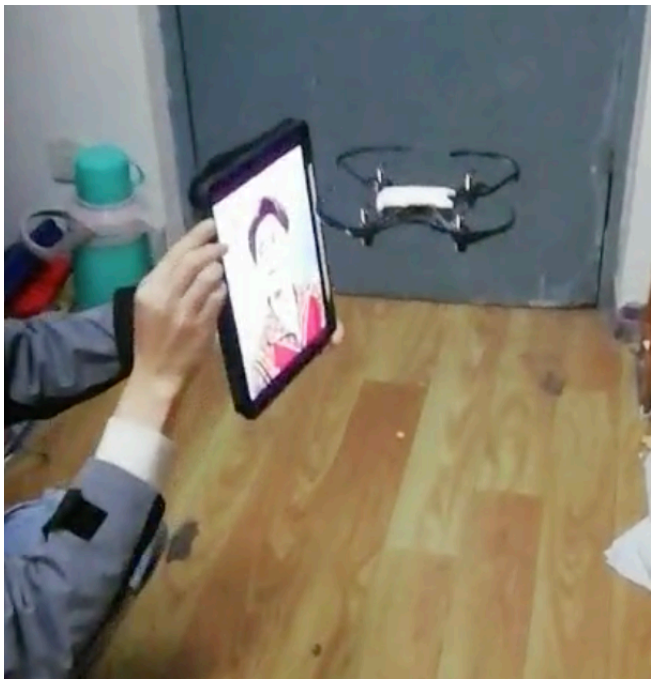
链接：https://pan.baidu.com/s/1n9AiKVfvzHp84XZP-VlgWQ

提取码：i7j0

视频方面，我们已经实现了对人脸的检测，同时无人机会自动对检测到的人脸进行追踪运动。



人脸追踪识别：电脑端视角



人脸追踪识别：无人机端视角

同样地，我们录制了相应视频作为展示。链接与提取码与上面语音识别相同，为同一个链接下的两个视频。

四、下一阶段展望

下一阶段，我们将在已有的基础上继续实现更高难度的目标寻找、自动飞行功能，并将根据难度赋予Tello不同程度的自我创作飞行动作功能。同时，针对第一阶段中存在的部分问题（如识别依赖网络等问题），我们也将寻找方法作出改进！