**1 项目主要研究**

1.1 了解研究背景与研究现状

声纹识别技术的原理，声纹识别技术的发展历程，声纹识别技术的现状

针对学习和调研的结果，完成综述的撰写

1.2 搭建系统的基本框架

设计带安全验证的声纹识别系统的逻辑流程，设计简单的GUI界面

基于pyqt5的gui界面，可以展示运行流程

1.3 口型识别模块--定位

捕捉视频文件中每帧的人脸区域，在此基础上通过颜色与移动特征定位嘴唇区域

完成了固定帧率下视频文件的逐帧捕捉，并加以方形框进行标识，进而导出为新的视频

**2 项目研究成果**

嘴唇区域定位 源文件

基于skvideo的音频信号特征提取 源文件

**3 项目季度报告**

3.1. 如何将声纹识别与口型识别相匹配？

基于常识的推论是，由于人类通常能够将听到的声音与给定的一组嘴唇运动进行匹配，因此嘴唇运动与由语音特征表示的听到的声音高度相关。

然而，目前还缺乏同时融合时空视听信息来解决音频流和视频流是否匹配的根本问题的研究。

AVR系统的方法是利用从一个模态中提取的信息，通过对缺失信息的补充来提高另一模态的识别能力。其根本问题是找到音频流和视频流之间的对应关系

有些学者的初步探索：

一个耦合的三维卷积神经网络（3D-CNN）架构，它可以将两种模式映射到一个表示空间中

三维CNNs同时从空间和时间两个维度提取特征，从而在相邻帧中捕获和连接运动信息

—>设计非线性映射

—>与输入空间相比，具有显著的降维，并且针对区分匹配和非匹配视听流进行了优化

3.2. 嘴唇特征提取的实现

OpenCV的接口可以实现人脸定位。

cvHaarDetectObjects是opencv1中的函数，opencv2中人脸检测使用的是 detectMultiScale函数。它可以检测出图片中所有的人脸，并将人脸用vector保存各个人脸的坐标、大小（用矩形表示），函数由分类器对象调用

3.3. ivector

因为从UBM模型自适应到每个说话人的GMM模型时，只改变均值，对于权重和协方差不做任何调整，所以说话人的信息大部分都蕴含在GMM的均值里面。GMM均值矢量中，除了绝大部分的说话人信息之外，也包含了信道信息。联合因子分析(Joint Factor Analysis, JFA)可以对说话人差异和信道差异分别建模，从而可以很好的对信道差异进行补偿，提高系统表现。但由于JFA需要大量不同通道的训练语料，获取困难，并且计算复杂，所以难以投入实际使用。由Dehak提出的，基于I-Vector因子分析技术，提出了全新的解决方法。JFA方法是对说话人差异空间以与信道差异空间分别建模，而基于I-Vector的方法是对全局差异进行建模，将其二者作为一个整体进行建模，这样处理放宽了对训练语料的限制，并且计算简单，性能也相当。

3.4. GMM-UBM

GMM(高斯混合模型)。它将空间分布的概率密度用多个高斯概率密度函数的加权来拟合，可以平滑的逼近任意形状的概率密度函数，并且是一个易于处理的参数模型，具备对实际数据极强的表征力。但反过来，GMM规模越庞大，表征力越强，其负面效应也会越明显：参数规模也会等比例的膨胀，需要更多的数据来驱动GMM的参数训练才能得到一个更加通用（或称泛化）的GMM模型。

GMM-UBM模型最重要的优势就是通过MAP算法对模型参数进行估计，避免了过拟合的发生，同时我们不必调整目标用户GMM的所有参数（权重，均值，方差）只需要对各个高斯成分的均值参数进行估计，就能实现最好的识别性能。

3.5 Raspberry

Raspberry（树莓派），是一种微型单板计算机，拥有丰富的硬件接口，能够安装任何ARM平台的操作系统。一方面，它有利于推动编程教育；另一方面，它与其他许多软件有较好的兼容性和拓展性。例如MATLAB上就有为树莓派定制的MATLAB支持包和Simulink支持包，以便将MATLAB上的代码转为可供树莓派运行的版本。此外亦有FFmpeg等软件支持音视频解码。此外，Raspberry支持许多硬件配件，通过给Raspberry装配相应摄像头和麦克风配件并安装相应配件，可以快速获取即时音视频信息以配合声纹/口型识别。因此，以Raspberry作为原型机进行相关测试，具有较好的通用性。

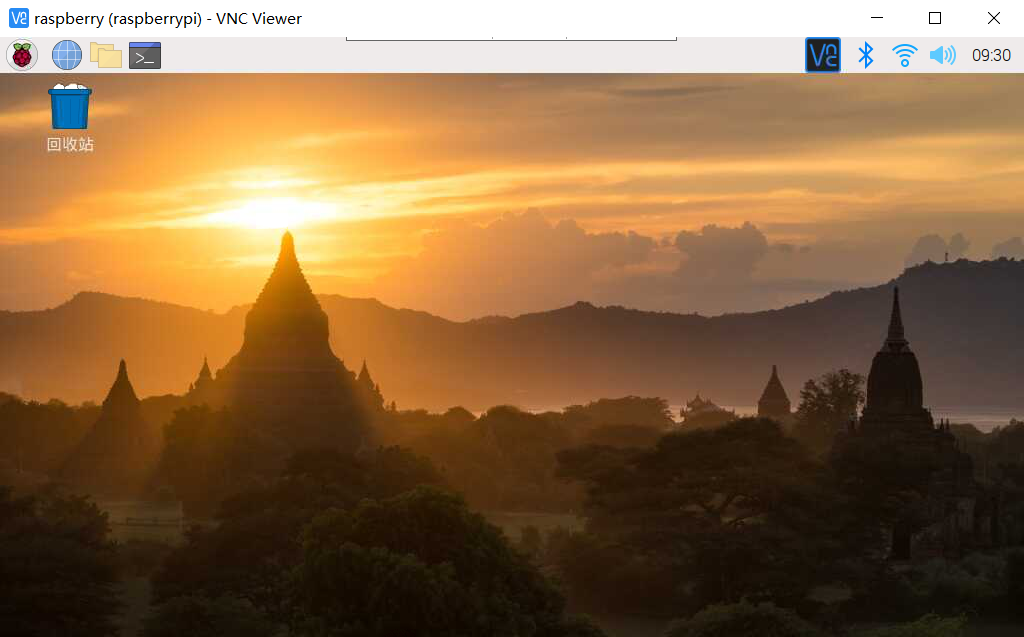


图 1 树莓派

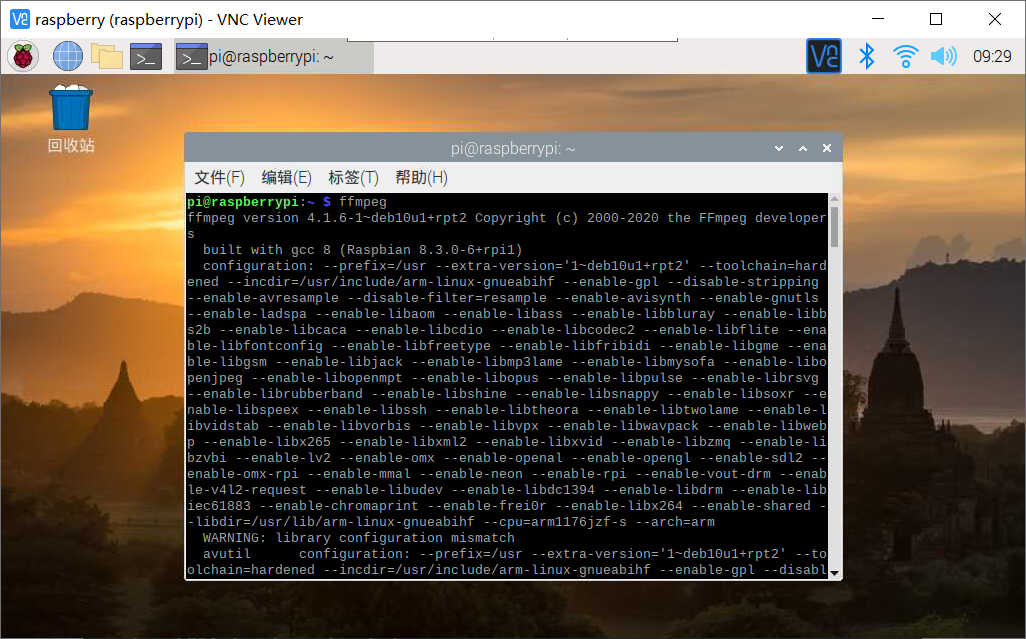


图 2 运行FFmpeg界面

3.6 声纹识别模块

初步完成了输入训练音频-预处理-训练识别模型，输入验证音频-预处理-验证的框架，其中预处理步骤包括了盲源分离与音频切割以适配周围环境嘈杂的情况，同时考虑到数据量较小时i-vector系统表现优于x-vector系统，训练与识别模型暂时使用MATLAB AUDIO工具包中ivectorSystem及相关函数实现（已修复原函数中的一处错误）。测试时使用了1422段完整音频训练i-vector系统，以5个说话人共10段完整音频作为登记（enroll）数据，验证结果总体正确率达70-80%，基本符合立项时的目标，但仍然存在提升空间，目前暂定计划为将i-vector系统替换为文本相关的d-vector系统。

1. n=[62,65,28,43];
2. s=zeros(1,198);
3. iv=ivectorSystem('SampleRate',fs,'InputType','audio');
4. adsTrain=audioDatastore('D:/source\_mono/\*/\*.wav','FileExtensions','.wav','LabelSource','foldernames')
5. trainLabels = adsTrain.Labels;
6. adsTrain = transform(adsTrain,@(x)extract(afe,x));
7. enrollLabels = adsEnroll.Labels;
8. adsEnroll = transform(adsEnroll,@(x)extract(afe,x));
9. trainExtractor(iv,adsTrain,'UBMNumComponents',64, 'UBMNumIterations',5, 'TVSRank',32, 'TVSNumIterations',3);
10. trainClassifier(iv,adsTrain,trainLabels, 'NumEigenvectors',16, "PLDANumDimensions",16,"PLDANumIterations",5);
11. **for** i=1:710
12. **for** j=1:4
13. dist(j)=sqrt(sum((y(i,:)-c(j,:)).^2));
14. **end**
15. loc=find(dist(:)==min(dist));
16. loc=loc(1);
17. name=cell2mat(ttmp.textdata(i));
18. name=name(21:30);
19. adsTest=audioDatastore(['D:/source\_mono/',name,'/\*.wav'],'FileExtensions','.wav','LabelSource','foldernames');
20. reset(adsTest);
21. sim=zeros(1,198);
22. **for** j=1:numel(adsTest.Files)
23. features=extract(afe,read(adsTest));
24. res=identify(iv,features,'plda');
25. **for** k=1:181
26. **for** p=1:199
27. **if** p<=198 && contains(cell2mat(tttmp.textdata(p)),**string**(res.Label(k)))
28. break;
29. **end**
30. **end**
31. **if** p<=198 && idx(p)==loc
32. sim(p)=sim(p)+**double**(res.Score(k));
33. **end**
34. **end**
35. **end**
36. s(i)=sum(sim)/n(loc)/numel(adsTest.Files);
37. **end**

**4 项目后期具体工作计划**

2021.7 -- 2021.9

通过调研，确定文本无关的声纹识别的技术路线，搜集相关资料，小组合作完成相关知识的储备

阅读相关文献，探索嘴唇区域的运动特征与发声的相关性

熟悉skvideo、pyaudio、ffmpeg的使用

2021.10 -- 2021.11

学习基于python和mfcc的音频特征提取过程，以及相关库的调用、特征参量的含义

嘴唇区域的运动特征与发声的相关性的算法实现

引入音频中所提取的特征参量，作为相关性的判据

完成口型识别模块的基本独立功能

2021.11 -- 2021.12

完善口型识别模块的基本独立功能，错误接受率与错误拒绝率的相关问题

学习将摄像头+麦克风获取的视频信息作为数据源，进而实现对采集信息的即时处理与返回

初步搭建声纹识别模块的程序框架，针对其中的技术难点制定下一步计划