

学校代码： 10286

分类号： TN432

密 级： 公开

基于Android的声纹识别和语音识别系统的设计 司向军 东南大学

UDC： 621.3

学 号： 143581

**工程硕士学位论文**

基于Android的声纹识别和语音识别系统的设计

研究生姓名**： 司向军**

导师姓名**： 万隧人 教授**

**吕耀安 高工**

|  |  |
| --- | --- |
| 申请学位类别 工程硕士 | 学位授予单位  **东 南 大 学** |
| 工程领域名称 集成电路工程 | 论文答辩日期 **2017 年 月 日** |
| 研究方向 **集成电路器件与工艺** | 学位授予日期 **2017 年 月 日** |
| 答辩委员会主席 | 评 阅 人 |
|  |  |

20 年 月 日

Design of VOICEPRINT RECOGNITION

AND SPEECH RECOGNITION

BASED ON ANDROID

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

SI Xiangjun

Supervised by

Professor LI Bing

and

Senior Engineer LV Yaoan

School of Microelectronics

Southeast University

April 2017

**独创性声明**

**东 南 大 学 学 位 论 文 使 用 授 权 声 明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名： 日 期：

**东 南 大 学 学 位 论 文 使 用 授 权 声 明**

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名： 导师签名： 日 期：

摘要

近年来，随着计算机应用的广泛普及，尤其是随着Android平台的迅速发展，各种Android应用呈现爆发式的增长。与此同时破解用户移动设备进而盗窃用户应用中的数据的行为也越来越多，应用中涉及到用户私密性数据的安全性受到很大的威胁。同时随着应用数量的增加，对移动设备中应用的操作也越来越难，因此寻求一种简洁的、安全的身份识别和简化应用操作的方法显得越来越重要。同时语言是人与人之间交流的最方便、有效的沟通方式，也是人与机器沟通最直接的方式，因此设计一款基于Android平台的声纹识别和语音识别系统来保护用户隐私和方便用户操作具有很大的现实意义。

在本系统中，本文创新性的提出一种新的Shazam算法的应用：声纹识别，并改进Shazam的音频指纹特征，使之具有较高的抗噪能力。实现了该算法在声纹识别上的应用，并编写了测试软件。Shazam算法不限制被检测音频的长短，只要被检测音频是属于样本音频的一部分，即可获取被检测音频指纹，音频指纹是本文中进行声纹识别的关键点。相对传统的声纹识别方法：特征提取-模式匹配，该声纹识别 的过程更加简洁和清晰。本文中依据Shazam算法编写了测试代码，测试结果显示该算法的识别率可达到95%，符合本文预期。本文中的声纹识别部分主要是进行Android设备的加密部分，应用操作部分由语音识别来完成。语音识别部分采用的是科大讯飞语音SDK（Software Development Kit）来辨别语音语义。在本文提出的新的声纹识别部分和语音识别的共同作用下形成一套基于Android平台的声纹识别和语音识别系统，进而保护用户设备安全和方便用户操作。

本文创新性的提出基于Android的声纹识别和语音识别系统，将会对移动设备的安全性问题得到一定程度的解决，对设备的可操作性得到很大的提高。本文的工作和成果具有重大的参考意义。

**关键词：**Android平台，Shazam算法，音频指纹，声纹识别，语音识别

Abstract

In recent years, with the widespread application of computer, especially with the rapid development of Android platform, a variety of Android applications show explosive growth, followed by hacking the user's mobile phone and theiving of user data. Related to the security of user data privacy is under great threat in the application. At the same time the operation of the applition in the mobile devices is becoming more and more difficult. Therefore, searching a simple、secure identity and simplified application operation method is becoming more and more important. Simultaneoulsy language is the most convenient and effective way of communication between people. It is also the most direct way to communicate with machine. Therefore, design a voiceprint recognition and speech recognition system based on android platform to protect user privacy and make user’s operation easier has a great practical significance.

In this system, this paper proposes a new application of Shazam algorithm : Voiceprint recognition. The application of the algorithm in voiceprint recognition is realized. Shazam algorithm does not limit the length of the detected audio, as long as the detected audio is part of the sample audio, audio fingerprints which is the key to voiceprint recognition can be detected in this process. Compared with the traditional voiceprint recognition methods: feature extraction and pattern matching, the voiceprint recognition process is more concise and more clearly. In this paper, according to Shazam algorithm writes a test code. The test results show that the recognition rate of the algorithm can reach 95%, in line with the expectations of this article. In this paper, the voiceprint identification part of the Android device is mainly part of the encryption, the application part of the operation is completed by the voice recognition. The HKUST XunFei SDK (Software Development Kit) is used to identify the semantics in the part of speech recognition.

This paper proposes a new voiceprint recognition and speech recognition system based on Android, which will solve the security problem of mobile devices and improve the operability of the equipment greatly. The work and achievements of this paper are of great reference significance.

**Keywords:** Android platform, Shamaz algorithm, Audio fingerprint, Voiceprint recognition, Speech recognition

目录

[摘要 I](#_Toc484964285)

[Abstract III](#_Toc484964286)

[目录 V](#_Toc484964287)

[第一章 绪论 1](#_Toc484964288)

[1.1 研究背景及意义 1](#_Toc484964289)

[1.2 国内外研究现状 2](#_Toc484964290)

[1.3 研究内容及设计指标 3](#_Toc484964291)

[1.3.1 研究内容 3](#_Toc484964292)

[1.3.2 设计指标 3](#_Toc484964293)

[1.4 论文组织结构 4](#_Toc484964294)

[第二章 声纹识别和语音识别系统概述 5](#_Toc484964295)

[2.1 Android平台简介 5](#_Toc484964296)

[2.1.1 Android基本概念 5](#_Toc484964297)

[2.1.2 Android系统结构 5](#_Toc484964298)

[2.2 声纹识别技术 7](#_Toc484964299)

[2.2.1 声纹识别技术的概念和分类 7](#_Toc484964300)

[2.2.2 声纹识别的基本原理和系统结构 8](#_Toc484964301)

[2.2.3 声纹识别系统的特征提取 10](#_Toc484964302)

[2.2.4 声纹识别系统性能分析 11](#_Toc484964303)

[2.3 语音识别技术 11](#_Toc484964304)

[2.3.1 语音识别技术的基本概念 11](#_Toc484964305)

[2.3.2 音识别技术的基本原理 12](#_Toc484964306)

[第三章 基于Android的声纹识别和语音识别的系统原理 13](#_Toc484964307)

[3.1 现有声纹识别基本方法 13](#_Toc484964308)

[3.1.1 一般声纹识别的特征提取方法 14](#_Toc484964309)

[3.1.2 声纹模式识别 15](#_Toc484964310)

[3.2改进Shazam算法的基本原理 17](#_Toc484964311)

[3.2.1 Shazam算法基本原理 17](#_Toc484964312)

[3.2.2 改进Shazam算法声纹特征的提取 19](#_Toc484964313)

[3.2.3 检索匹配算法 21](#_Toc484964314)

[3.3 语音识别的实现 23](#_Toc484964315)

[3.3.1 特征参数的提取 23](#_Toc484964316)

[3.3.2 模板匹配方法 29](#_Toc484964317)

[3.4 本章小结 30](#_Toc484964318)

[第四章 基于Android的声纹识别和语音识别系统的实现 31](#_Toc484964319)

[4.1 系统概述 31](#_Toc484964320)

[4.1.1 整体系统框架及概述 31](#_Toc484964321)

[4.1.2 Android应用开发流程 32](#_Toc484964322)

[4.2 声纹识别模块实现 32](#_Toc484964323)

[4.2.1 数据库设计 32](#_Toc484964324)

[4.2.2 Android语音采集模块实现 34](#_Toc484964325)

[4.2.3 声纹识别核心算法模块实现 36](#_Toc484964326)

[4.3 语音识别模块实现 39](#_Toc484964327)

[4.3.1 环境搭建 39](#_Toc484964328)

[4.3.2 消息处理 40](#_Toc484964329)

[4.4 本章小结 40](#_Toc484964330)

[第五章 基于Android的声纹识别和语音识别系统测试分析 43](#_Toc484964331)

[5.1 实验条件 43](#_Toc484964332)

[5.1.1 硬件环境 43](#_Toc484964333)

[5.1.2 软件环境 43](#_Toc484964334)

[5.1.3 实验素材 43](#_Toc484964335)

[5.2 声纹识别系统测试 43](#_Toc484964336)

[5.2.1 待测语音和模板语音起始时间一致时的性能测试 44](#_Toc484964337)

[5.2.2 待测语音为部分模板语音时的性能测试 47](#_Toc484964338)

[5.3 系统的整体运行测试 48](#_Toc484964339)

[5.4 实验结果分析 50](#_Toc484964340)

[5.5 本章小结 50](#_Toc484964341)

[第六章 总结与展望 51](#_Toc484964342)

[6.1总结与展望 51](#_Toc484964343)

[致谢 53](#_Toc484964344)

[参考文献 55](#_Toc484964345)

第一章 绪论

1.1 研究背景及意义

使用语音进行沟通是我们日常生活中，最便捷、最常用、最简洁的交流沟通方式。而且语言是人类所特有的行为方式，语音则是人们交流沟通的基础。因此，语音就成为了传达信息、交流感情的主要途径。如今人类已经进入了网络时代和信息时代，基于现代化的手段，对语音信号进行处理，可以使得人们更加便捷，更加高效的使用语音进行交流。这方面的研究会对促进经济的发展和社会的进步，有重大的社会意义。

自计算机问世以来，人们就有让计算机理解人类语言，实现人机交互的梦想。随着计算机慢慢的向便携和移动终端的方向发展，人们更加需要摆脱鼠标和键盘的束缚，而是使用语音，来直接和计算机进行交流，进而控制计算机。在这样的情况下，一门新的计算机技术— —语音识别技术便应运而生了。

语音识别（Speech Recognition, SR），指的是计算机能够听懂人说的话，既是能够精确的识别出人说话的内容，并理解人说话的含义，做出相应的操作。伴随着社会的迅速信息化和经济的快速发展，网络技术、计算机技术、嵌入式技术、信号处理技术等的应用越来越重要。在身份认证，金融安全、刑事调查以及人机交互、公共安全和其他领域，都急需可靠，方便的识别技术。传统识别身份的技术主要是通过检测包含个人信息的事物入手，这种识别技术存在很多的不足，例如会对个人的隐私造成很大的威胁，或者个人携带的标识自身身份的物品丢失，如果被其他人冒名顶替，就很可能造成很大的损失。在高度信息化的今天，需要识别身份的场所越来越多，身份识别的重要性也越来越大。然而依据传统的身份识别和判定方法已经比较落后了，远远不能满足现代人们生活的需求。基于生物特性的身份判定和识别技术，拥有很独特的优势，为我们提供了一种高效、安全和便捷的方法[1]。

声纹识别（Voiceprint Recognition, VR），又被称为说话人识别，是根据说话人语音的生物特性来判断身份的技术[2]。声纹识别是语音识别的一种特殊形式。声纹识别是从说话人的语音信号中提取能识别出身份的特定生物因素，关键点是识别出不同人之间的特性差异；语音识别则是从语音信号中识别出说话人说出的内容，关键点是找到共性。

声纹识别和常见的红膜识别和指纹识别等一样是属于生物识别领域的识别技术，具有便捷和不会丢失和遗忘等优点。采用语音识别一个人的身份与其他形式的生物识别技术相比，具有很多的优势：

1）认证方法简洁，接受程度高，用户只需要说几句话就行了，不必像虹膜和指纹识别技术那样就行，需要把眼镜和手靠近扫描设备，而且红膜识别和指纹识别受环境的影响较大，比如当手指不干净的时候，会给指纹识别照成很大的影响；

2）语音数据的获取简单便捷，不需要很昂贵的设备，像红膜识别和指纹识别等仪器都比较昂贵，而语音输入只需要简单的语音输入设备；

3）相较于红膜识别和指纹识别，语音识别支持远程识别，在基于网络的识别应用中，如电子银行等，相比于红膜识别和指纹识别，更加便与应用。

随着计算机和微电子技术的快速发展和稳步提升，嵌入式系统已经进入到智能时代，并迅速发展并形成一个主流，最典型的设备就是手持移动端的普及，在目前移动端市场上，以Android为操作系统的移动终端已经非常普及了，占据市场80%的份额，使用的人数越来越多，市场也是非常的广阔。因此，在嵌入式系统中，尤其是在以Android为代表的手持移动端，实现一个能够判断用户身份，并能够识别出说话人的含义，进行相应的智能操作的声纹识别和语音识别系统，具有很大的理论研究和实际应用价值。

1.2 国内外研究现状

在1980年代，国外学者就已经开始对音频信息分类和检索技术进行研究。在对这项技术研究发展过程中，音频信号的处理主要集中在语音识别、说话人识别、音频分割等音频处理方面的研究，知道上世纪90年代末，音频信息分类和检索技术才慢慢成为国内外学者研究的重点，并逐渐形成研究热潮。仅仅在几年内，国内外的许多学者对检索技术和音频信息分类技术进行了细致的研究，使得这个领域，逐渐形成一个完善的体系，在很大的程度上加快了音频检索技术和音频信息处理的发展。

在1990年代中后期，声纹识别技术和语音识别技术已经成为了国内外专家的研究对象。直到现在，声纹识别技术已经成为了国内外学者研究及开发的前沿科学技术之一，国内外血多高校也纷纷投入人力和物力到这个领域，并取得了优秀的成果。如国外的麻省理工学院、南加州大学、西班牙马德里大学等。相比之下，国内学者及高校对基于声纹识别和语音识别技术的研究起步较晚，和国外先进的音频检索技术还有很大的差距，但也引起了广泛的关注。Muscle Fish公司（美国）是较早推出了较为完整的原型系统。该公司所做工作包括，首先对数据进行分帧，对每帧数据进行提取音调，亮度，响度，带宽等属性，而后再对属性序列计算其方差、均值、自相关值等，加上能量共13个特征，作为音频数据的矢量特征，通过这些数据的矢量特征，检索时采用马氏距离，通过比较样本特征矢量和数据库中存储的特征矢量，进而检索出结果。Jonathan Foote开发了一种基于量化树的方法，该团队通过提取音频数据的倒频谱特征MFCC（Mel Frequency Cepstral Coefficients），并通过借鉴语音分析中的方法，把音频数据的频谱表示并构造成一个量化树，最后得到的是一种量化柄直方图。新西兰团队New Zealand研究了音乐曲调和旋律的检索。Philips(荷兰)团队提出了一种基于全局信息的方法，该方法通过将整个频谱分割成许多的小块，每个小块由0或1来表示，通过这种表示方法，整个频谱可以用二进制数的序列进行表示，这种方法的特点是可以表示整个频谱的全局信息，然而缺点是信息量代表很差，受到噪声的影响很大。Shazam算法是英国Shazam公司基于特征点提出的方法，这种方法是通过在频谱图中找到能表示该语音特征的特征点，组成特征点对（Peak-Pairs），把特征点对的序列作为该片段的声纹，这个方法的特点是，不需要整个频谱的全局信息，只需要频谱的部分信息即可作为语音特征，优点是信息集中，抗噪性能好[3]。

目前来说，基于内容的声纹识别技术还是不成熟的，还有很大的研究和应用空间。虽然已经有很多的关于基于内容的声纹识别技术的研究，但是还有很多的方面需要改进，智能化和自动化的程度还远远不够。针对海量的音频数据，找到一种快速、准确的音频检索方法以便更好的满足用户的需求还需要进一步的研究。因此基于理论和实践工作还有很多工作要做，以便完善实用的基于内容的声纹识别。

1.3 研究内容及设计指标

### 1.3.1 研究内容

本课题研究的内容是基于Android的声纹识别和语音识别系统的设计，该系统主要应用于智能设备终端。具体来说，主要包括一下几个方面：

1）查找相关文献，分析并研究语音识别和声纹识别技术中的关键点；

2）分析并研究几种常见的声纹识别算法，主要包括支持向量机模型、隐马尔科夫模型、高斯 混合模型、动态时间规整（DTW）等，并在此基础上提出基于改进Shazam算法的声纹识别；

3）实现基于Android的声纹识别和语音识别系统的设计。

### 1.3.2 设计指标

本课题要实现基于Android的声纹识别和语音识别系统的设计，要达到的指标如下：

设计功能指标

1. 分析现有的声纹识别算法，实现基于Shazam算法的声纹识别功能；
2. 集成科大讯飞SDK，在此基础上实现基于Android的语音识别功能；
3. 以Android为开发平台，集成声纹识别和语音识别功能，并进行功能验证；

设计性能指标

1. 在无噪声环境下，声纹识别系统识别率不小于95%；
2. 在无噪声环境下，语音指定打开软件的成功率不小于90%。

1.4 论文组织结构

本文共分六个章节，其内容如下：

第一章，绪论。主要介绍课题的研究背景和国内外研究情况，以及研究难点和热点。

第二章，声纹识别和语音识别系统概述。介绍了基于内容的声纹识别系统的识别过程，声纹识别系统的评价指标，以及声纹识别和语音识别系统的主要应用。

第三章，基于Android的声纹识别和语音识别系统原理。介绍基于内容的声纹识别的特征分析，主要包括Chroma旋律特征，线性预测系数（LPC），线性预测倒谱系数（LPCC），多种声纹特征（Audio FingerPrint），Mel倒谱系数等，并在此基础上阐明基于Shazam算法的声纹识别，以及语音识别的实现原理。

第四章，基于Android的声纹识别和语音识别系统的实现。介绍本文系统实现的详细步骤，包括开发环境搭建、核心模块实现等。

第五章，基于Android的声纹识别和语音识别系统测试分析。介绍本系统的测试方法和内容，并编写测试软件对系统进行测试和分析。实验表明，基于Android的声纹识别和语音识别系统具有较高的鲁棒性，可以用于实际。

第六章，总结和展望。对本文所做工作进行了总结，并提出本论文工作上的不足，并对下一步工作进行了展望。

第二章 声纹识别和语音识别系统概述

2.1 Android平台简介

### 2.1.1 Android基本概念

Android是一种基于Linux的自由及开放源代码的操作系统，主要应用于移动设备，如平板电脑和智能手机，由Google及开放手机联盟领导及开发。Android操作系统最初由Andy Rubin开发，主要应用于手机，Google公司于2005年注资收购，继续开发Android操作系统。2007年，Google公司和84家软件开发商、硬件制造商以及电信运营商一起合作成立开放手持设备联盟，共同开发改善Android操作系统，同时Google公司以Apache开源许可证的授权方式，发布Android系统源代码，以便让合作商生产基于Android系统的智能手机，随着Android操作系统的发展，该系统逐渐扩展到其他移动终端设备及其他应用领域，并起到极大的促进作用。

2007年11月初，以Google公司为主的最初一批公司包括Broadcom、Intel、LG、HTC、马维尔等公司，一起成立了开放手持设备联盟（Open Handset Alliance, OHA）。该联盟成立的最初目的是为了使得移动市场环境更加开放和自由。同时他们的第一个产品：以Linux 2.6为内核的Android系统智能手机。随着该联盟的发展，ARM、华为、索尼等公司也相继加入OHA。

同时Android开源代码工程AOSP（Android Open Source Project）也成立了，该项目成立的主要目的是维护和更新Android操作系统的开源代码。除此之外，来自全球各地的开发者也相继成立开源社区，专门负责开发Android应用程序和第三方Android操作系统，用以保证Android操作系统的功能和性能。Android操作系统是真正意义上的开放性移动应用平台，由于这个特点，开发者可有效降低开发成本，从而降低移动设备的整体价格，扩大用户群体。

### 2.1.2 Android系统结构

Android系统结构如图2-1所示。Android系统架构采用了分层架构的思想，从上层到底层依次可以分为应用层、应用框架层、系统运行库层、Android运行时和Linux核心层，下面简要介绍每个层次所包含的的内容[4]。

1. 应用程序层

Android操作系统可以包含很多的应用程序，如即时通信程序、图片浏览器、电话拨号器和网页浏览器等应用程序。此外所用的Android应用程序都是用Java语言编写，利用Java语言跨平台性质，基于Android框架开发的应用程序可以不用编译运行于任何一台安装有Android系统的平台。并且Android应用程序都可以被第三方开发人员开发的应用程序替换，这点也和其他手持操作系统有很大的不同，使得Android系统更加的灵活、个性化 。

1. 应用程序框架层

应用程序框架层是一个应用程序的核心，是所有参与开发的程序员必须遵守的约定，可以在该框架上进行必要的扩展，但是程序要始终保持主体结构的一致性。应用程序的架构设计简化了组件的重用，任何一个应用程序都可以发布它的功能模块并且任何其它的应用程序都可以使用其所发布的功能块，帮助程序员快速开发应用程序，并且该应用程序重用机制也可以使用户可以方便的替换程序组件，基于这样的重用机制，用户就可以方便的进行替换平台本身的各种应用组件，使得开发更加容易进行。

1. 系统运行库层

系统运行库层主要包括两部分，分别是Android系统库（Android Libraries）和Android运行时（Android Runtime）。Android系统库是Android应用程序的基础，从图2-1可以看出，Android系统库是Android应用程序层和Linux内核层之间的桥梁；Android运行时，Android应用程序是由Java语言编写的，程序在Android运行时中运行，以提供给Android应用程序必要的组件。Android运行时又可分为核心库（Core Libraries）和Dalvik虚拟机两部分。其中核心库为Android应用程序提供了Java语言编程中的大多数API（Application Programming Interface）功能，例如android.os、android.net、android.media等；Dalvik虚拟机部分，Android应用程序与其他J2ME（Java 2 Micro Edition）程序不同，Android应用程序运行于一个独立的进程中，每一个应用程序都有一个属于它自己的Dalivik虚拟机实例。核心库和Dalvik虚拟机两部分分工不同，但共同完成系统运行时所需要的系统运行库依赖。

1. Linux内核层

Android系统的内核是Linux2.6，Android系统的核心系统服务，比如安全、内存管理、进程管理、网络协议、驱动模型等都依赖于Linux内核[4]，其中的驱动模型包括显示驱动、摄像头驱动、键盘驱动、WiFi驱动、Audio驱动、Flash内存驱动、Binder(IPC)驱动和电源管理等。Linux内核层也是作为软件和硬件的抽象层，是Android系统框架中的核心部分。

总之，Android的系统架构采用分层的架构思想，架构清晰，层次分明，协同工作，同时Android的系统架构不仅从宏观上认识了Android系统，同时也为学习和实践指明了方向。如果从事的是Android应用开发，则应该研究Android应用层序框架层和应用程序层；如果从事Android系统开发，则应研究Android系统库和Android运行时；如果从事Android驱动开发，则应该研究Android的Linux内核。

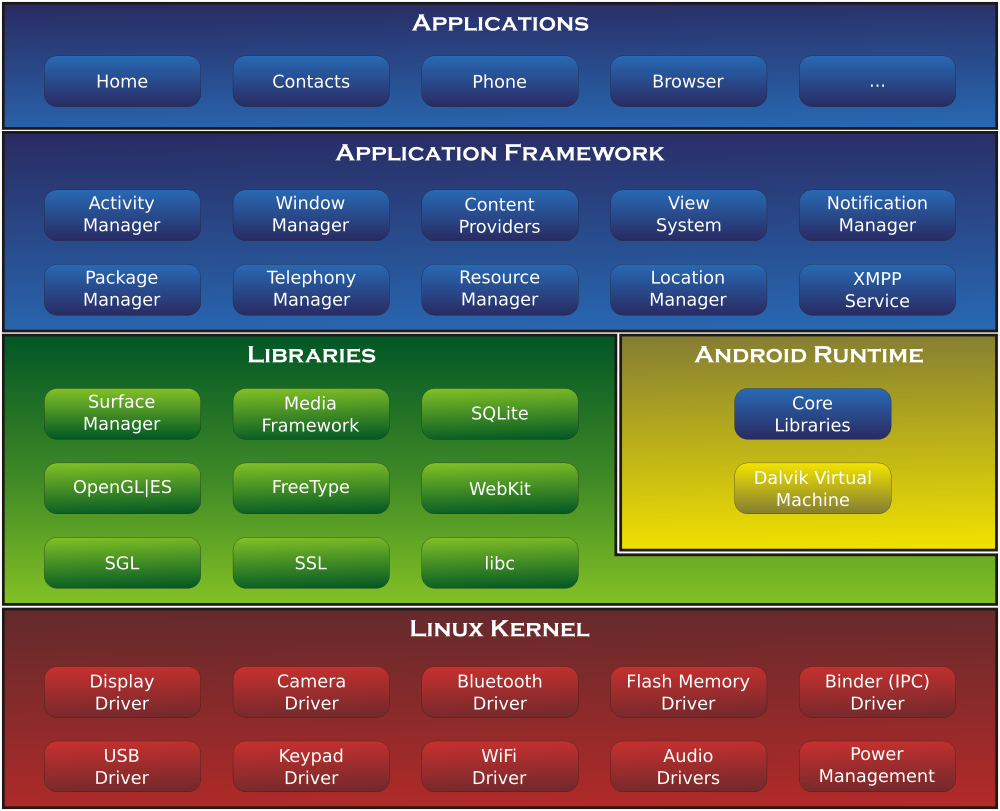


图2-1 Android系统结构

2.2 声纹识别技术

### 2.2.1 声纹识别技术的概念和分类

声纹识别又被称为说话人识别，是一种根据说话人的声音就可以判定说话人身份的技术，广义上来说声纹识别是语音识别的一种，它们都属于语音信号处理的范畴，在处理方法、处理过程及语音模型等方面有很多的相似之处，除此之外它们都依赖于语音信号自身的特点。不同之处在于声纹识别是辨认说话人的身份，关键点是不同说话人之间语音特点的差别；语音识别是识别出语音信号中所包含的语言内容和意义，而不用考虑说话人是谁，关键点是不同说话人之间共同的地方。说话人之间的语音差别有很多，比如说话人发声器官、声道之间、语速、发声习惯之间的差别，因此，声纹识别是一门涉及到生理学、语音信号处理、模式识别和人工智能技术的跨多学科的综合性研究课题。

声纹识别具有比指纹识别、红膜识别、人脸识别等其他生物识别技术的优势，主要包括一下几点：

1）所需采集设备成本低廉，且便于采集，易于推广；

2）不涉及敏感的个人隐私，用户的接受性强；

3）在某些场景下，比如电话网络中，识别人的语音特征是较少甚至是唯一可以获取到的生物特征，适合远程身份确认；

4）易于在识别人不知情的情况下完成识别，隐蔽性好；

5）声纹辨认和确认的算法的复杂度低。

由于具有上面所说的简单快捷及易于接受等特点，声纹识别作为人机交互的一种高效手段，目前在司法取证、音频数据处理和检索、侦探和安全及个性化服务等领域都有广泛的应用[5][6]。

声纹识别有不同的分类方法，第一种，按照最终完成任务的不同，声纹识别可以分为说话人确认（Speaker Verification, SV）和说话人辨认（Speaker Identification, SI）。说话人确认是识别一段语音是不是某个人所说，这个过程涉及到的是特定的参考模型和待识别模型之间的判别，系统只需要识别出“是”或“不是”即可。说话人辨认是识别出一段语音是多个待识别语音片段中的哪一个。根据判断待识别的语音片段是否在系统库中语音集合内，又可以把声纹识别分为开集和闭集辨认，开集辨认是待识别语音片段可能没有在待识别的语音集合内，由于需要比较，一般识别的错误率大于识别的正确率，并随着系统语音库的增加，整体系统的性能在不断下降；闭集辨认是假设待识别的语音片段已经在系统集合中。第二种，根据待识别人的训练和测试语音内容是否相同，声纹识别又可分成文本无关型、文本指定型、文本相关型。文本无关型是指，声纹识别结果的正确与否和语音内容无关，即是每次所说的内容可以不一样；文本相关型是指，声纹识别结果与语音内容息息相关，即说话人训练和识别时说话的内容要保持一致；文本指定型是指，在识别时，先由识别系统向待识别人指定要说的文本，只有当系统确认说话人所说的内容是指定的文本时，才进行声纹的识别，这点恰好弥补文本无关型和文本相关型的缺点，防止个人语音被盗用，具有很高的安全性[7]。

### 2.2.2 声纹识别的基本原理和系统结构

声纹识别系统的结构一般包括待识别语音预处理、特征提取、模式训练和识别等几个部分，除此之外，一个完整的声纹识别系统还需要自适应模块和阀值判定模块，如图2-2所示。

图2-2声纹识别系统一般原理图(虚线为可选部分)

该声纹识别系统模型的建立分为两大阶段，第一阶段是训练阶段，主要内容是对每一个说话人收集一定数量的语音材料，该系统会根据这这语音样本，通过多次训练建立符合说话人的训练模型或者模板。第二阶段是识别阶段，提取待识别说话人测试语音的特征参数，对比已经建立好的系统参考模型库，选择距离最相近的某个特定模板作为待识别说话人。如果还要进行说话人确认，只需要进行对提取的输入语音的特征参数进行判定即可，如果特征参数小于模型规定的阀值，系统就会接受，否者拒绝。最后系统会根据说话人语音动态以及实时性来自动调节参考模型。

声音预处理主要包括声音信号的采集和转化(模拟转数字)、预滤波、预加重、加窗处理、端点检测等过程[8][9][10]：

1）采集和转化。首先要采集说话人的语音，需要将音频信号转化为模拟信号，然后依据奈奎斯特采样定律把模拟信号转化为数字信号，采样频率一般为8kHz，10kHz，12kHz，16kHz等，量化精度一般采用12bit，16bit线化。人耳能听到的频率范围是20Hz-20KHz，实验证明，对语音的清晰度和可辨度有十分明显影响的最高频率是5.7kHz，所以依据奈奎斯特采样定律，即是采样频率大于语音信号的两倍带宽时，采集的样本不会丢失信息，通过理想滤波器可以从采样语音中不失真的还原语音。同时在采集和转化之前，还要进行预滤波，滤波的目的是在一定程度上抑制其他噪声的干扰，一般采用带通滤波器。目前信号的采集和转化还是比较方便的，利用PC机的声卡即可。具体过程如图2-3所示，语音信号经过Mac或者话筒的声模转换后传输到声卡，可以直接获得经过防混叠滤波、A/D、脉冲编码调制技术（PCM）量化处理的离散的数字信号，例如利用声卡获取的wav文件，可以直接作为存储材料。



图2-3采集和转化流程

2）预加重。因语音信号是声道和语音源激励相卷积的结果，在大约800Hz以上的频谱会按照6dB/倍频程跌落，为了使得整体信号的频谱变得更加平坦，使得整个低频和高频部分能用一样的信噪比求频谱，更便于声道特征的参数的分析，需要提高高频频谱。在采样/量化之后，需要用6dB/倍频程预加重滤波器来提高高频特性，其z传递函数形式如式(2.1)所示。

 (2.1)

上式中，*a*值接近于1，本文取a=0.95。

3）加窗处理。在对语音信号进行分帧之后，每一帧都当成平稳信号来处理，再用傅里叶展开每一项，这时就会出现吉布斯效应，这种效应是应该避免的，因此要对语音信号进行加窗处理。目的就是为了减少帧起始和结束的地方信号的不连续性问题和避免出现吉布斯效应。在语音信号数字处理中常用的窗函数是矩形窗和汉明窗，N为帧长，其表达式如(2.2)和(2.3)所示。

矩形窗

 (2.2)

汉明窗

 (2.3)  
 4）端点检测。所谓端点检测就是从包含语音的一段信号中寻找到语音的起点和终点，进而处理有效的语音信号，其依据是语音信号和噪声信号的特性不同。常用的端点检测的方法有基于鲁邦特征的方法，基于模型的方法和基于特征滤波的方法等。基于鲁棒特征的方法是找到表示语音和噪声分布分布在不同频域的差异来进行划分，这种划分方法有两方面的因素，其一，提取能够表征不同类型语音信号的特征，浊音、清音记忆噪音；其二，依据不同信噪比的信号能够检测出合适的门限值，一般采用的判断特征主要是短时能量、基频、周期度量、能量方差、熵等。基于模型的方法是通过建模的方式来区分语音和噪音。这种方法的缺点是不能应对不同的环境建立不同的模型，另外当模型和噪声环境稍有差距，性能严重下降。基于特征滤波的方法，是对特征进行滤波后在进行端点检测，常用的算法包括子空间滤波、能量差分自适应滤波等，但是此类方法需要的运算量极大，同时被测语音结构被改变，不能完全保存原始语音信息。

### 2.2.3 声纹识别系统的特征提取

声纹特征提取的过程实际上是减小数据量的过程，就是把原始语音信息中的冗余信息除掉，找到系统所需要的语音信息。它是声纹识别中一个相当重要的过程，语音信号包含说话者共性特征和个性特征，是个性特征和语音特征的卷积体，而非简单的叠加。

从语音信号中提取声纹特征要满足几个条件。第一能够长期保持稳定，第二对外部条件具有鲁棒性，第三易于从语音信号中提取声纹特征。根据参数稳定性，从声学方面来说，可以把说话人的语音特征分为两类。与说话人生理结构相关的固有特性和反应声道运动的动态特性。固有特性主要体现在语音频谱结构上，包含代表声道共振频谱包络特征信息和表征声带振动的频谱细节构造的特征信息，主要是指基音和共振峰特征参数，这两种参数不易被模仿，但是容易受到说话人健康状况的影响。动态特性主要是指反应声道运动的一些动态特征的参数，比如发音习惯和发音方式等，主要是指语音频谱结构随着时间的变化，含有特征参数的动态特性，这种特性是相对稳定且容易被模仿的。

声纹特征参数的提取依据是能够有效区分说话人，且对同一说话人语音发生的变化保持相对的稳定性。综合声纹特征可量化、系统总体性能、训练样本数量的大小等问题，目前声纹识别中经常用到的声学特征有如下几类：

1）谱包络参数。语音信号经过滤波器后，再用适当的抽样频率对其进行抽样，把抽样结果作 为待识别的特征参数；

2）听觉特征参数。模仿人耳对声音频率的感知特性，比如MFCC（梅尔倒谱系数）、Bark频率倒谱系数、线性预测等；

3）基音轮廓和共振峰的频率带宽和轨迹。这种特征参数是从人体发声器官中提取的。

4）线性预测系数和派生系数；

5）混合参数。为提高声纹识别精度，可以采用混合参量来构成矢量。比如将带通滤波器谱和逆滤波器谱相结合、动态参量和统计参量相结合、线性预测参数和基因轮廓相结合等。为了提高声纹识别系统的识别效果，应尽量选取相关性差异大的参量作为矢量[11][12]。

### 2.2.4 声纹识别系统性能分析

影响系统性能的因素有很多，对声纹识别系统，识别的正确率是衡量其好坏最直接的因素，但是影响识别率的因素有很多如：特征参数提取算法的精度、训练模型的混合阶数大小、端点检测的精度、性别不同引起的识别差异等。

声纹识别系统现阶段面临的挑战还有很多，主要有以下几点：

1）特征参数的稳定性：由于语音的变化受到说话人身体健康，环境等因素的影响，特征参数会出现非平稳特征，但在理想状态下，特征参数具有一定的稳定性和不易模仿等特性；

2）识别的可信度：现阶段的识别结果是根据匹配情况确定识别人，但可信程度仍然任重而道远；

3）系统自适应模块：系统的特征参数能够根据不断变化的环境来自动调节的性能；

4）声纹识别系统的稳定性。

2.3 语音识别技术

### 2.3.1 语音识别技术的基本概念

语音识别，根本目的是研究一种具有听觉功能的机器，能够听得懂人类的语音命令，进而理解人类的意图而做出响应的操作。语音识别技术的研究对象是语音信号，涉及语言学、信号处理、生理学、心理学、计算机科学等诸多领域，是模式识别的重要组成部分。语音识别技术已经有非常广阔的应用，比如：改变了人类计算机的操作方式，通过语音直接进行操作；听写机带来办公自动化的重大革命；通信方面实现两种语言直接的直接交流，即通过“语音识别—机器翻译—语音合成”将一种语言直接转化为另一种语言；通过语音来检索数据库；还有在汽车、飞行员在高速行驶中直接通过语音进行电话拨号或发布命令。

随着计算机和数字信号处理器运算能力及识别系统精度的提高，识别系统根据词汇量大小进行分类也不断进行变化。目前中等词汇量的识别系统到将来可能就是小词汇量的语音识别系统。这些不同的限制也确定了识别系统的困难程度[13][14]。

### 2.3.2 音识别技术的基本原理

语音识别技术的产生主要是为了满足人类和机器交流的需要，进一步通过语音指令来控制机器完成人类期望的工作。语音识别技术的应用过程中，一般分为两个步骤，第一步为系统的“学习”和“训练”过程，此过程最主要的是通过已知的语音模板，采用语音模型对语音模板进行训练和学习，在此基础上建立已训练完成语音信号的语言模型。第二步是语音信号的“测试”和“识别”过程。将待识别的语音信号经过预处理，特征提取后，和已经训练好的语音模型进行对比，通过建立好的语音模型对未知语音信号数据进行分析和识别，最后得出该语音信号的识别结果。语音识别的基本过程如图2-4所示。

图2-4语音识别基本过程

语音识别本质上是属于模式识别的范畴，从上图可以得知，语音识别的主要内容是模式匹配。首先要根据说话人语音特点简历语音模型，对输入的语音信息进行分析，提取特征参数，在此基础上通过训练得出语音识别所需要的模板。然后计算机通过已经建立好的语音模板和待识别的语音信号特征进行比较，根据一定的搜索和匹配方法，找到最优的与输入语音匹配的模板。进而根据模板的定义，通过查表就可得出计算机的识别结果。这种方法和语音特征的选取、语音模型的好坏、模板是否准确都有直接的关系[15][16]。

第三章 基于Android的声纹识别和语音识别的系统原理

本章先介绍现有声纹识别的基本方法及其识别原理，并介绍语音预处理的一般过程，在此基础上介绍Shazam算法的基本原理，并提出改进Shazam算法的声纹识别，并给出其原理。

3.1 现有声纹识别基本方法

根据本文调研，现有的声纹识别的一般流程如下图所示。



图3-1 一般声纹识别流程图

总体分为以下几个步骤：

1) 原始语音经过预处理得到处理之后的语音片段。此过程是为了对高频部分进行加重，增加语音高频部分的分辨率；

2) 对预处理之后的语音片段进行特征提取。特征提取是为了提取语音片段中的关键信息，这种关键信息要能够区分不同人的声纹特征，也就是要能够进行声纹识别；

3）如果进行声纹注册，则对待注册语音提取其声纹特征并存储在数据库中；

4）如果进行声纹确认，则同样对待识别的语音片段提取其声纹特征，然后和模型库中存储的声纹特征进行对比，得到相似度，最后识别出结果。

在语音系统中，语音模型一般可以用3-1图表示。

在数学上，语音信号的产生可以看作是激励信号经过线性系统后产生的，其中为声道模型，是离散时域的声道传输函数，通常用全极点函数来近似。由于每个人说话的声音是不同的，因此不同的人具有不同的声道模型，语音模型可用如下图表示。



图3-2 语音模型

进行声纹识别的第一步首先进行预处理，预处理的作用是对高频部分进行加重，增加语音高频部分的分辨率，预处理部分可以分为以下步骤：

1）采样和量化。由于人耳听觉的范围是20Hz-20000Hz之间，根据奈奎斯特-香弄采样定律可知，采样率要为人耳听觉上限的两倍，本文采样频率取=80000Hz，每个采样用8bit表示；

2）预加重。语音信号经过或者更高的采样频率采样后，模拟信号就转换为数字信号，再通过一阶高通滤波器以突出高频部分，传递函数为；

3）取音框。一般情况下一个音框为32ms，包含256个采样点，音框之间有128个重叠点，即是每位移128个采样点后再取下256个采样点作为音框，这样采样的作用是避免音框之间特性的变化过于剧烈；

4）加窗。为了消除两个音框两端的不连续性，避免分析时受到前后音框的影响，每个音窗要乘以汉明窗；

5）端点检测。利用语音信号的短时过零率和短时能量对语音信号进行端点检测，主要目的是删除不含语音信息的背景噪声和静音段，从而获取含有语音信息的语音段，减少语音信号处理中的计算量，提高识别率 [17][18]。

3.1.1 一般声纹识别的特征提取方法

经过预处理之后，就可提取语音特征。提取语音特征的过程，实际就是去除原语音信号中的冗余信息，减少数据量的过程。语音特征参数的提取应满足以下原则：能长期保持稳定，易于观测，对外部变量，例如说话人身体健康状况和情绪等不敏感；独立性，即与其他特性不相关。声纹识别研究中主要采用的特征参数有：Mel频率倒谱系数MFCC、线性预测系数LPC、线性预测倒谱系数LPCC等[19]。

（一）线性预测系数LPC

基本思想是依靠语音信号采样点之间的相关性，用过去采样点的值来预测现在或者未来采样点的值，在数学上表示即是语音的抽样能够用临近若干抽样点的线性组合来预测，使得线性预测抽样值和实际语音信号抽样值之间的误差在在均方准则下达到最小值来求解系数，此系数反映了语音信号的特征。

（二）线性预测倒谱系数LPCC

LPC倒谱系数一般通过语音信号的傅里叶变换，取模的对数，再反傅里叶变换得到，线性预测的系统函数的频率响应反应的是被分析信号的谱包络和声道的频率响应，可以通过求的对数log||做反傅里叶变换求出倒谱系数，此倒谱系数可以看作是描述语音信号的良好参数。比较彻底的去除了语音产生过程中的激励信息，主要反应声道响应是此倒谱系数主要的优点，更重要的只需通过十几个倒谱系数就可很好的描述语音信号的共振峰特征，因此，在声纹识别中有良好的应用[20]。

（三）Mel频率倒谱系数MFCC

MFCC梅尔倒谱系数与普通实际频谱倒谱分析不同，MFCC是基于人耳听觉特性，由于人耳听到声音的高低和声音的频率不成正比的，所以MFCC更符合人耳的听觉特性。Mel频率尺度大致对应实际声音频率的对数分布，可用以下公式表示

 (3.1)

实际频率f的单位是Hz，已知临界频率带宽会随着频率的变化而变化，且和Mel频率的增长一致。1000Hz以下时，成线性分布，带宽为100Hz左右；在1000Hz以上时呈现对数增长。可将语音频率划分为一系列的三角形滤波器序列既是Mel滤波器组[21]。

3.1.2 声纹模式识别

对于声纹识别系统，当语音特征被提取出来之后，要用识别模型为说话人建立模型，并根据特征进行分类，以便确定是属于哪个人。所谓声纹识别模型是指用什么模型来描述声纹特征在特征空间的分布。目前常用的模型可以分为四类分别是（1）带参数模型；（2）非参数模型；（3）人工神经模型（Artificial Neural Network, ANN）；（4）支持向量机模型（SVM, Support Vector Machine）。

参数模型用某种特定的概率密度函数来描绘语音特征在特征空间的分布，同时把该概率密度函数的参数作为说话人模型。比较成熟的参数模型有隐马尔科夫模型（Hidden Markov Model, HMM）、高斯混合模型（Gaussian Mixture, GMM）；非参数的模型是指说话人模型直接由语音特征运算得到，典型的非参数模型有矢量化模型（Vector Quantization, VQ）和模板匹配模型。

下面简单介绍这几种模型的基本原理

（一）隐马尔科夫模型

隐马尔科夫模型属于信号统计理论模型，对处理随机时序数据识别和预测有比较好的效果，HMM本质上是一种双重随机过程有限状态自动机，其双重随机过程是指满足Markov分布的状态转换Markov链，此过程是基本的随机过程，主要是描述状态转移；另一个随机过程描述的是每一个观察值和状态之间的统计对应关系，既是状态的观察输出概率密度函数。HMM应用于声纹识别中时，HMM类首先给定随机模型，再通过计算产生一个观察序列，其似然概率为，随机模型为从说话人训练语音中得到的特征矢量，再依据特征矢量得到的概率密度函数。给出每个人的随机模型后，观察序列的概率也被确定[22][23]。

HMM应用到声纹识别系统中经常会遇到一下三种基本问题：

1）给定模型，求某个观察序列O的概率；

2）给定模型和观察序列O，求可能性最大的隐藏状态序列；

3）对于给定的观察序列O，调整HMM参数，使得观察序列出现的概率最大；

上面三个问题的解决方法即为著名的HMM三大基本算法：前向-后向算法、Viterbi算法、和Baum-Welch算法[24]。隐形马尔科夫模型通常使用单状态的HMM，是比较流行的方法，效果较好。

（二）GMM（高斯混合模型）

高斯混合模型可以表示为M个高斯概率密度函数的加权和，其数学表达式为：

 (3.2)

其中*x*表示D维随机向量；(i=1,2,…M)是高斯概率密度函数；(i=1,2,…M)是各个成员函数的加权值[25][26]。所谓高斯混合模型就是对样本的概率密度分布进行估计，而估计采用的训练模型是几个高斯模型的加权和，具体由几个高斯模型加权而成要在模型训练之前确定。每一个高斯模型代表一个类，对样本的数据分别在几个高斯模型上投影，就会分别得到在各个类上的概率。然后可以选取概率最大的类作为判断结果。

（三）矢量化模型

现今主流的矢量化模型有两种：k-means分群发、学习向量量化法。k-means分群发是一种分隔式分群法，具体原理是在给定的码本向量数后反复修正，使得每个丛聚中的每一个点与丛聚中心的距离平方差最小。假设用k个码本向量代表每个说话人的特征参数向量，k-means算法可以用以下步骤概述：①任意产生k个值当做码本向量作为初始值；②搜寻临近值，对每一个特征向量点x进行计算，寻找与x最接近的群中心，并将x加入其中；③更新码本向量；④重复②③步骤，使得每一个特征向量点x和码本向量的平均距离小于一个门限值，最后所得的一组码本向量即代表每一位说话人的参考码本。学习向量量化法的基本原理主要是微调码本中的码本向量以减少一些分错的向量。可以分为以下几个步骤：①先使用分群法找出每位说话人的码本向量，当做初始值；②在初始值中任意挑选一个训练向量x，如果其属于说话者p，则将其标记为，接着找出与向量x距离最近的码本向量c，如果其为说话者q所有，则将c标记为，③如果p和q属于同一位说话人，则将码本向量向的方向拉近，否则把码本向量向的反方向推远，④如果训练结果达到要求则训练结束，否则回到步骤②继续进行微调码本向量。

矢量化模型训练时保留所有的特征矢量，识别时对每个矢量都找到训练矢量中最近的K个，据此进行识别，缺点是通常模型存储和相似计算的量都很大，这对于存储量有限的移动端是一个很大的局限。

（四）支持向量机模型

支持向量机模型是一种比较新的机器学习方法，近几年才刚刚起步，进行深入研究。在很多领域，比如人脸检测、文本识别和手写体识别等模式识别问题中，支持向量机模型已经有了很好的应用了，并取得了很好的效果。SVM是在统计学习理论基础上发展出来的理论模型方法，统计学理论是一种研究有限样本情况下的机器学习规律的理论。VapnikV等人是在20世纪中叶开始此方面的研究，到了20世纪90年代中期，伴随着理论的不断发展和成熟，同时由于神经网络的发展处于停滞状态，统计学习理论开始受到越来越广泛的重视。SVM在解决有限样本、非线性极高维模式识别问题中已经表现出许多优于已有方法的性能[27]。

（五）人工神经网络

人工神经网络（Artifical Neural Network, ANN）是对在人类对大脑神经网络理解的基础上采用物理可实现的系统来模拟人脑神经网络的结构和关系的系统。它从信息处理角度对人脑神经网络进行抽象，建立某种简单模型，按不同的连接方式组成不同的网络。神经网络是一种运算模型，由大量的节点(或称为神经元)之间相互连接构成 [28]。其缺点是训练量大，且模型的可推广性不好。

3.2改进Shazam算法的基本原理

Shazam算法相对上一章节的声纹识别算法来说，模型存储所占用的空间较小，这点可以使得该算法可以在有限存储空间的移动设备中的到广泛的应用；同时其训练量较小，模型的可推广性较好。

3.2.1 Shazam算法基本原理

Shazam算法的原理结构如下所示。



图3-3 Shazam算法原理图

Shazam算法提取的语音特征的方法比较简单。首先把语音信号从时域变换到频域，图3-4所示即是原始语音信号，然后经过预处理，增加语音的分辨率；其次在待识别的语音信号频谱中提取能量的峰值点，并记录这些能量峰值点所在的时间位置得到一个二维变量；最后根据一帧语音信号中单个峰值点的频率值和所在时间位置构成该帧信号的声纹特征，并存储到数据库中等待识别。下图显示了一段语音信号的语谱图以及从语谱图中提取到的能量峰值点所形成的星状图。

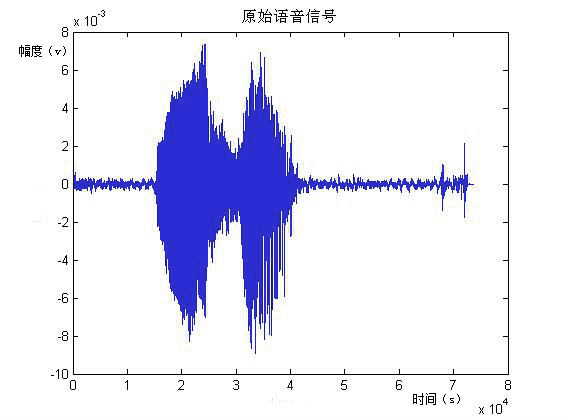


图3-4 原始语音信号

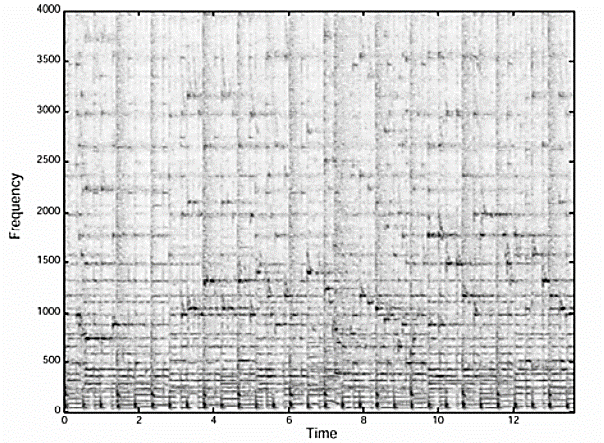


图3-5 语音信号的语谱图

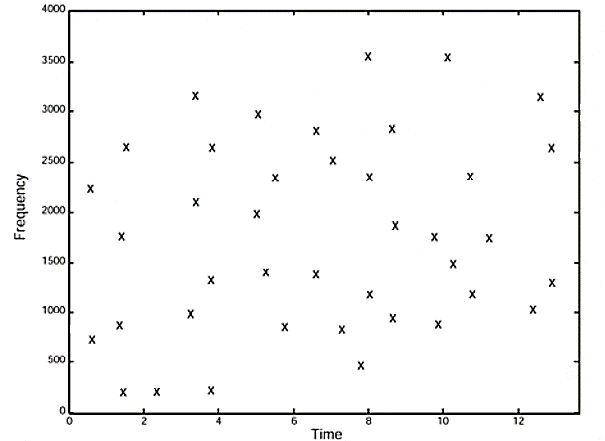


图3-6 从语谱图中提取的峰值特征点

峰值点是语音信号中的典型特征，所代表的是语音信号频谱中能量最高的一部分，能够显著的代表一段语音的特性。例如256个采样点经过傅里叶变换之后有128个频率位置，根据这128个频率值构成128个特征点。然后根据这些特征点构成该语音片段的声纹特征点，存储到数据库中等待识别。

3.2.2 改进Shazam算法声纹特征的提取

上节中介绍的Shazam算法的原理中，提取得到的128个频率值只能构成128个声纹特征点，特征点的选取范围比较小，区分度比较低，很可能会存在大量相同的峰值点，在此情况下使用该峰值点作为特征点构建索引，峰值点的碰撞的几率会比较高，会大大影响检索的效率。为此本文采用了特征点对的方法对该Shazam算法进行赶紧，算法的具体流程图如下所示。



图3-7 改进Shazam算法的系统流程图

本文中改进Shazam算法是将多个特征点构成特征点对，把特征点作为构建索引，所构建的特征对成为语音指纹，如图3-7所示。

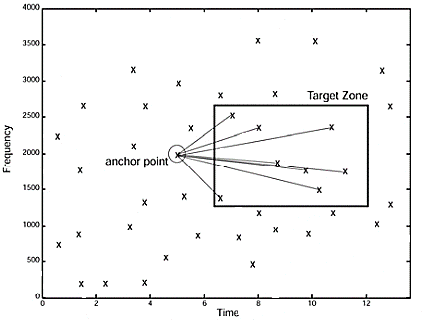


图3-8 多个特征点形成特征点对

使用多个特征点构成特征对的方法可以有效增加语音峰值点的区分度。例如，如果峰值点有128种可能的取值，则*n*个特征点构成的语音指纹的可选范围是(1<*n*<128)，显著提高了语音特征的区分程度。本文中Shazam算法采用特征点对之间的距离构建三元组，进而作为语音指纹。如图3-8所示，三元组语音指纹,其中*f*1和*f*2分别是第一和第二个峰值点在频率坐标上的位置，为两频率之间所对应的时间差，即。

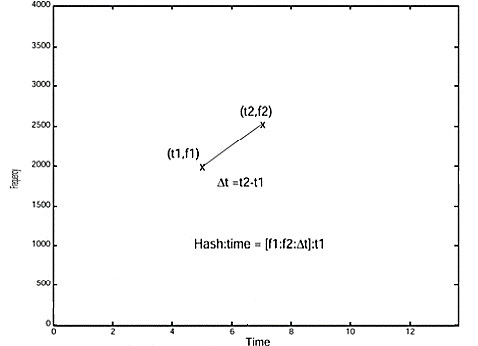


图3-9 特征点的频率和时间差

利用上述方法虽然能提高语音特征区分度，但是却在一定程度上降低了语音指纹的稳定性，原因是语音指纹能够正确匹配的前提条件是两个峰值点都要正确匹配。假如*p*是峰值点在语音片段经过各种干扰后仍是准确的概率，则两个峰值点都准确的概率是，所以语音特征的稳定性是低于峰值点的稳定性的。为了增加准确语音指纹的匹配数量，采用一个峰值点构建多个特征点对的方法，从而形成多个语音指纹，图3-8所示为其构建方法：每个特征点和其临近的目标区域中的所有峰值点构建特征点对，当前的特征点称为“锚点”，目标区域中的峰值点叫做“配对点”。引入目标区域的目的是增加特征点的匹配数目。例如,每个特征点都是一个锚点，目标区域的外联系数F是10，则生成的配对点的数目约等于10倍的特征点个数。通过限制外联系数，可以限制最终生成指纹的个数。实际上，外联系数可以用来衡量存储空间的耗费程度。通过配对点来代替单个峰值点的匹配会带来巨大的性能提升。假设频率轴为10bit，时间差也为10bit，则配对点会产生一个30bit信息量的指纹，与原始单用峰值点匹配多了20bit的数据，匹配数据也相应加快，这是因为匹配时产生的碰撞更少了。

采用配对点进行匹配也有缺点，它会使得某个峰值点存活的概率由*p*变为（即是概率变小），假定*p*是峰值点在原始语音信号和录制语音信号中都存在的概率，配对点也包含该峰值点的概率是。速度的提升是以概率的降低为代价的。但是可以通过减小外联系数来减小概率的降低。假定F=10，则某个锚点至少存活一个配对点的概率等于该锚点与其目标区域中至少有一个匹配点存活的概率。假设目标区域中的点是独立的，则某个锚点至少有一个匹配点存活的概率为：

 (3.3)

对于很大的F(F>10)和合理的*p*(*p*>0.2)会存在：

 (3.4)

所以匹配点加外联系数不会导致检索准确率的大幅下降。

以上就是本文中对Shazam算法改进的基本原理，用多个特征点构成的声纹识别特征点不会导致检索准确率的大幅下降。

3.2.3 检索匹配算法

在执行匹配之前，需要先从录取的语音样本中提取所有的语音指纹和时间偏移信息，这两点是检索匹配算法的关键之处。语音样本中提取的语音指纹都会和数据库中的语音指纹进行匹配，每一个匹配的语音指纹都会产生一个时间对：样本中的时间和数据库中的时间。根据对应的语音ID将时间进行分类。本文中对语音特征的检索先构建倒排索引，来快速确定语音指纹的位置。倒排索引的关键字是语音指纹，即是从语音片段中提取的峰值点对，对应存储的值是语音指纹所在语音的ID以及语音指纹的第一个锚点(峰值点)所在原始语音片段中的位置，即对应所在语音中的时间偏移。采用特征点对的方法可以增加关键字的数目，从而减少碰撞率。如图3-10所示，为该检索的倒排索引结构，从该检索结构中可以看出，一个语音指纹中所包含的信息主要有语音ID和所提取的指纹的偏移时间点。



图3-10倒排索引结构

当从语音样本提取的指纹都完成上述匹配后，通过扫描时间对获得正确的语音。每个语音片段对应的所有时间对构成了一个散点图，如果语音和样本匹配，则匹配上的特征会有相近的时间偏移，也即是样本中提取的指纹和数据库中的指纹具有相同的相对时间（这个相对时间就是样本在原始语音中的起始位置）。这样就可以将搜索问题简化为散点图中寻找明显对角线的问题，对该问题的解决，现在有很多现有的技术，例如Hough变换或者其他回归技术。但是这些方法复杂度太高，同时对异常值比较敏感，所以不适用。考虑到问题的特殊性，可以设计一个复杂度为的算法，其中N表示散点图中点的个数。

不失一般性，假设对角线的斜率为1.0（录音速度和原始语音速度一致），则样本和正确语音中匹配指纹的时间满足下面的关系式：

 (3.5)

其中表示原始语音中匹配指纹的时间，表示录音样本中对应指纹的时间。对于散点图中的每一个点，我们计算

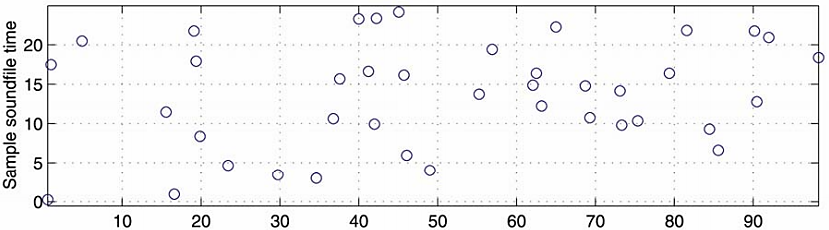
 (3.6)

当录音速度和原始语音速度不一致时，一般情况下录音速度为原始语音速度的0.5—2倍之间，设此因数为，此时样本语音和语音库中匹配指纹的时间应满足下面的关系式：

 (3.7)

然后计算的直方图并扫描获得直方图的最大值即是将时间对的两个时间相减，然后统计相同时间差个数的最大值，求最大值可以通过扫描完成，由于匹配对的原因，散点图通常会非常稀疏。同时每段语音对应的时间差个数通常很少，所以扫描过程会在几微秒内完成，每段语音对应的的分即是直方图的最大值，如果直方图最大值很大，那就意味着该语音片段就是要识别的正确语音。

图3-11展示了一个错误匹配的例子，图3-12展示了一个正确匹配的例子，若录音样本语音和数据库中原始语音语速不一致，此时图中的斜线的斜率应为0.5—2之间。



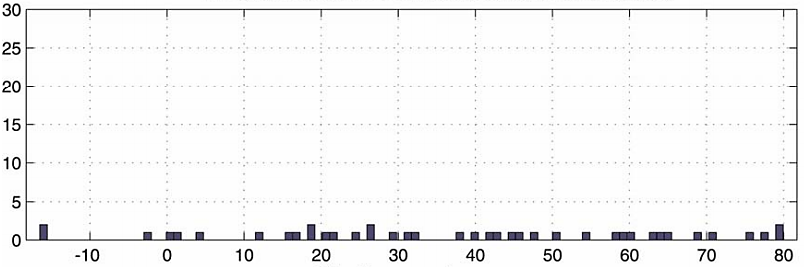
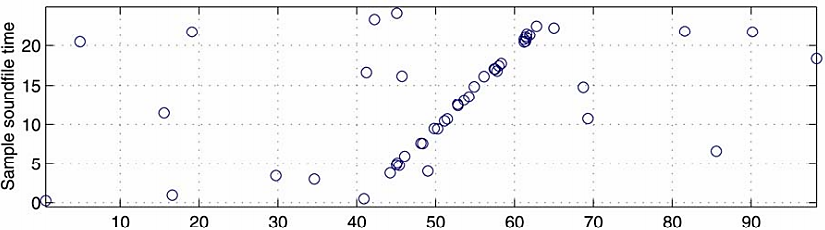


图3-11 错误匹配样本图



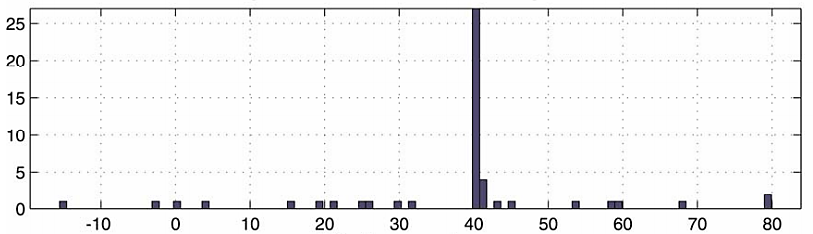


图3-12 正确匹配样本图

通过以上的描述，可以看出该匹配算法的核心思想是：录制的语音片段是数据库中某段音频的一部分，因此待查询语音片段经过适当的平移后和数据库中的某语音片段的子片段的内容是完全一致的，匹配程度最高，基于这个特点的考虑，Shazam首先找到所有匹配的语音指纹，然后再求出相同时间偏移差最多的匹配片段即为目标语音片段。

3.3 语音识别的实现

在对语音信号进行预处理之后，就可对处理后的语音信号提取语音识别所需要的信号特征，既是对原始语音信号剔除无关紧要的冗余信息后，就可对语音信号进行分析处理，以获得影响语音识别的重要信息。

在语音信号中，语音特征参数的是分帧提取的，每一帧提取的特征参数构成一个特征矢量，因此从语音信号中提取的语音特征矢量是一个矢量序列。这些语音矢量序列经过数据压缩后就可成为语音识别的模板。显而易见，语音特征参数的提取对最后识别的效果至关重要，所以选择的标准应满足：（1）各阶参数之间互不干扰，既是要有良好的独立性；（2）可以有效的代表语音特性，具有良好的区分性，包括其听觉特性和声道特性；（3）语音特征参数的计算要方便，其计算方法要高校，以便保证语音识别的实时性[30][31]。

3.3.1 特征参数的提取

语音特征参数的提取对语音识别系统性能的好坏起到关键性的作用，理想的特征参数的选择不仅对语音识别系统的识别率有很大的提高，同时对系统的性能也起着至关重要的作用，语音特征参数的选择的基本原理是，对经过预处理的语音信号经过一次变化，剔除掉语音信号中的冗余部分，把表示语音特征本质的参数提取出来。在此之后的处理都是在语音特征参数的建立之上，如果提取的语音特征参数不能很好的反应语音信号的本质，则语音识别就不会成功。目前为止，在已经实践了的前提下，已证明有效的特征参数有LPC线性预测编码系数、LPCC线性预测编码倒谱系数、MFCC梅尔频率倒谱系数[32][33]。

1. LPC系数特征

LPC系数特征是维纳在1947年首次提出的，在此之后，线性预测应用于许多领域，例如计算机、生物医学、机械工程等领域，在20世纪60年代末板仓等人最早将线性预测编码技术应用于语音识别中。线性预测编码技术作为一种语音信号处理工具，已经普遍的应用于语音信号处理的方方面面。

语音信号是一种时变信号，但如果观察时间为毫秒级别时，则可以得到近似稳定的信号。人的发音可以用声管模型来近似表示，既是可以用若干段前后连接的声管进行模拟。在声管模型中，主要包含两部分，一部分是浊音部分，另一部分是清音部分。在此情况下，如果可以通过计算出声管模型的参数或者声管模型的形状，就可以用此模型参数或者形状作为语音信号的特征完成语音识别的任务。

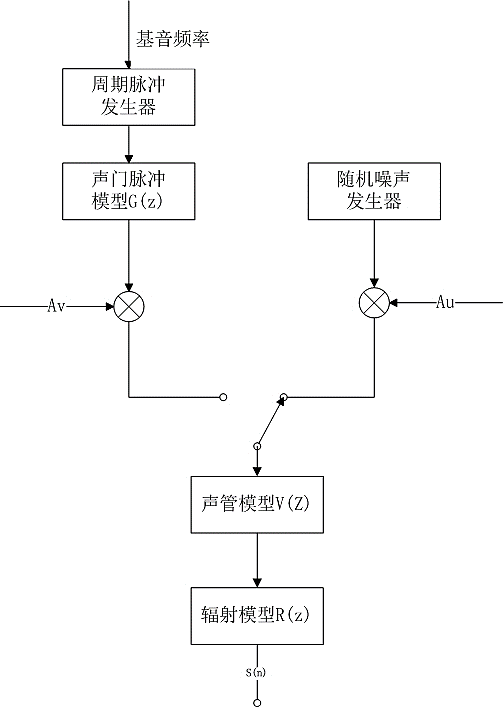


图3-13语音信号产生的声管模型

在数字信号处理中，可以用LPC线性预测编码算法来估计出全极点模型的参数。声管模型也有自身的优缺点，在对分析平稳的元音信号时效果较好，在分析辅音部分的信号时分析效果不好，原因是简单的声管模型不能精确的描述语音信号，例如，声管模型没有考虑鼻腔支路的耦合影响，因为两个鼻腔支路的耦合会使得两个声管并联而产生零点，使得语音的产生模型变得更为复杂，参数的估计也变得更复杂，但是实践证明全极点的声管模型，无论在语音识别还是在语音编码中都取得了较好的效果。

线性预测编码模型的基本思想是：对n时刻的采样信号s(n)，可以用前面p个样本进行线性逼近来表示：

 (3.8)

系数分别为前p个样本的权重值，为常数，上式可改为：

 (3.9)

其中Gu(n)为s(n)和实际n时刻采样点之间的误差，表现为一个归一化冲击响应及其增益系数的乘积。其在Z域中表达式为：

 (3.10)

从上式中可以得到系统的传递函数为：

 (3.11)

从图3-12可以看出，H(*z*)表示的就是声道模型和辐射模型之间的级联，可以用一个短时稳定的时变滤波器来表示。其中的参数是由发声器官进行控制并随着时间变化的，但在极短的时间内是认为稳定不变的。H(*z*)的激励是由清音开关控制，当激励为浊音时，激励源可以用归一化冲击序列和增益系数G的乘积来表示，其中冲击序列的周期可以用语音的基音周期来表示，声音的音量可以用增益系数G来表示。当激励为清音时，激励信号可以用随机噪声和增益系数G的乘积来表示。在一个分析帧内其周期和G通常是被认为为稳定的常数，但是实际上是随着时间变化的。

假设预测值为，则从3.48式可以看出

 (3.12)

其中为预测系数，为常数，可可得到预测误差为，则

 (3.13)

响应的误差传递函数为

 (3.14)

如果s(*n*)是由图3-12所示的线性系统产生，则实际值和预测值之间的误差可以用激励和增益的乘积来表示，即：。

线性预测分析的基本问题是要通过计算得到预测系数，但是语音信号的频谱特征是随着时间的变化而变化的，要计算给定时刻n的预测系数必须要在n时刻的前一段短时间语音帧中进行计算，否则将会不准确。对于某一特定的短时语音序列，在均方误差最小的基础上，如果能找到一个特定的阶数p和一组线性预测系数，则这组线性预测系数就是最佳的预测系数，可以用来表示这组最佳的线性预测系数。

在计算线性预测系数之前，要定义七点为n的短时语音信号和误差信号：

, (3.15)

误差的平方和为：

 (3.16)

对上式各阶LPC系数求导，令其分别为零：

 (3.17)

可以得到：

 (3.18)

根据上式可以看出，式中的为信号的短时协方差，即可以用下式表示：

 (3.19)

则式(3.58)可以写成：

 (3.20)

可知该式由p个方程组成，有p个未知数。求解该方程组，就可得到LPC系统的线性预测系数。其次，系统的最小均方误差为：

 (3.21)

求解该方程的方法有很多种，主要有协方差法、自相关法等，下面给出自相关法的求解步骤。

对短时语音信号进行定义的简单和有效的方法是假设当时的信号为零。既是相当于语音信号乘以有限窗函数，可用下式表示：

 (3.22)

其中窗函数就是第二章介绍的任意窗函数，有上式可知，时为0，所以当的时候也为零，表示没有预测误差，这是在理想的状态下。同样当时也没有预测误差，基于上式的加窗误差可以表示为：

 (3.23)

且可以表示为：

 (3.24)

根据上式，的值只取决于和的差值，和这两个变量之间的绝对值无关，定义

的自相关函数为则：

 (3.25)

则也可用自相关函数来表示，且具有自相关函数的性质，即是：

 (3.26)

式(3.62)可以表示为：

 (3.27)

用矩阵来表示为：

 (3.28)

从上式可以看出矩阵是对称矩阵，一个线性方程组具有这种性质时，可以用高校的递推算法解决，下面介绍一种递推方法——Durbin德宾递推法。

Durbin递推法的递推过程如下：

如果假定LPC预测器的预测阶数为*p*：

1. 计算
2. 计算
3. 计算
4. 
5. 计算
6. 令
7. 计算
8. 
9. 计算
10. 计算
11. 

根据比较结果，如果是，则令，转入(7)继续执行该算法，如果不是则终止该递推，输出，该系数便是LPC线性预测系数，在上式中，选取阶数的一般模型是，首先保证要有足够的极点来模仿声管模型的结构，其次，阶数增加时，线性预测的误差是趋于下降的，但是当阶数的值达到一定时(通常为14)，该线性预测误差趋于平缓。一般情况下声管模型的阶数的取值一般为14。

1. 线性预测编码倒谱系数LPCC

LPC倒谱系数一般通过语音信号的傅里叶变换，取模的对数，再反傅里叶变换得到，线性预测的系统函数的频率响应反应的是被分析信号的谱包络和声道的频率响应，可以通过求的对数log||做反傅里叶变换求出倒谱系数，此倒谱系数可以看作是描述语音信号的良好参数。比较彻底的去除了语音产生过程中的激励信息，主要反应声道响应是此倒谱系数主要的优点，更重要的只需通过十几个倒谱系数就可很好的描述语音信号的共振峰特征，因此，在声纹识别中有良好的应用[34][35]。

1. MFCC梅尔倒谱系数

由于LPC线性预测模型的精度依靠声管模型的精度，但在实际中语音信号的平稳特性不能随时满足。因此基于声管模型的线性预测系数的语音特征参数的鲁棒性不是很好，现在常用的语音参数为基于人的听觉的特征参数。人耳对不同频率的声波有不同的灵敏度，200Hz到5kHz频率段之间的声波对语音的清晰度有很大的影响，低音很容易掩盖高音，反之不容易，在低频处的声音掩蔽的临界带宽较高频端要小，因此，在低频到高频频带内按临界带宽的大小由密到稀安排一组带通滤波器。通过对语音信号进行滤波，把每个带通滤波器输出的信号的值作为语音信号的基本特征，对此特征再进行进一步的处理就可以作为语音识别系统的输入特征。基于这种方法，这种特征不依赖于语音信号的性质，对输入信号没有限制，除此之外又是基于听觉模型的研究成果，当信噪比较低时，有很好的识别性能。

MFCC是以“bark”为频率基准，和线性频率之间的转换关系为：

 (3.29)

MFCC的参数也是按照帧率计算的，首先通过FFT变换获得该语音信号帧的功率谱，然后再转换为Mel频率下的功率谱，其转换方法是在计算之前现在原语音频谱范围设置个带通滤波器：

 (3.30)

其中M和N分别为滤波器的个数和一帧语音信号中的样本点数。其中M一般取值为24，N取值256.滤波器在频率上表现为简单三角形，其中心频率用表示，它们在Mel频率尺度上是均匀分布的。在较低的频率区间内，和之间成线性关系。

一般MFCC参数的计算流程如下：

1）首先对原始语音信号进行采样，之后对每帧序列分别进行预处理、FFT变换，在对其结果取模的平方得到离散的功率谱。

2）计算离散功率谱通过M个滤波器后得到的功率值，得到M个参数。

3）计算的自然对数，得到

4）对进行离散正弦变换，得到

5）舍去直流分量，取

上述过程中的MFCC参数只代表了语音参数的静态特征，但通常情况下人耳对语音的动态特征较为敏感，语音动态参数的获取可以通过差分倒谱参数来描述这种动态特性，计算公式如下：

 (3.31)

其中和代表一个语音帧中的参数，为常数。

### 3.3.2 模板匹配方法

模式识别就是通过计算机，用数学模型求解的方法研究模式的自动处理和判读。在模式识别的各种方法中，模板匹配是最容易的一种，其数学模型容易建立。目前常用的模板匹配有三种技术：动态时间规整、隐马尔科夫、矢量量化技术。

(一)[动态时间规整](https://www.baidu.com/s?wd=%E5%8A%A8%E6%80%81%E6%97%B6%E9%97%B4%E8%A7%84%E6%95%B4&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)

动态时间规整是一种衡量两个时间序列之间相似度的方法，主要应用在语音识别领域来识别两段语音是否相同，在时间序列中，需要比较相似性的两段时间序列的长度可能并不相等，在语音识别领域表现为不同人的语速不同。而且同一个单词内的不同音素的发音速度也不同，比如说话人会把“A”这个音拖得很长，或把“i”发的很短。另外，不同时间序列可能仅仅存在时间轴上的位移，亦即在还原位移的情况下，两个时间序列是一致的。在这些复杂情况下，使用传统的欧几里得距离无法有效地求的两个时间序列之间的距离（或者相似性），DTW通过序列进行延伸和伸缩，来计算两个时间序列之间的。  
 (二)隐[马尔可夫](https://www.baidu.com/s?wd=%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)法  
 隐[马尔可夫](https://www.baidu.com/s?wd=%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)法是对语音信号的时间序列结构建立统计模型，将之看作一个数学上的双重[随机过程](https://www.baidu.com/s?wd=%E9%9A%8F%E6%9C%BA%E8%BF%87%E7%A8%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)：一个是用具有有限状态数的[Markov链](https://www.baidu.com/s?wd=Markov%E9%93%BE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)来模拟语音信号统计特性变化的隐含的[随机过程](https://www.baidu.com/s?wd=%E9%9A%8F%E6%9C%BA%E8%BF%87%E7%A8%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)，另一个是与[Markov链](https://www.baidu.com/s?wd=Markov%E9%93%BE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)的每一个状态相关联的观测序列的[随机过程](https://www.baidu.com/s?wd=%E9%9A%8F%E6%9C%BA%E8%BF%87%E7%A8%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)。前者通过后者表现出来，但前者的具体参数是不可测的。人的言语过程实际上就是一个双重随机过程，语音信号本身是一个可观测的时变序列，是由大脑根据语法知识和言语需要发出的音素的参数流。由此可见隐马尔可夫算法合理地模仿了这一过程，在描述语音信号的整体非平稳性和局部平稳性方面表现出了很好的性能，是一种很好的语音模型。

(三)[矢量量化](https://www.baidu.com/s?wd=%E7%9F%A2%E9%87%8F%E9%87%8F%E5%8C%96&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uH7-uH61PHIWryw9mhRd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnj0dPWTsPH61PH0krjRLrf)（VQ）  
 矢量量化是70年代后期发展起来的一种压缩技术，其基本思想是：将若干标量数据组构成一个矢量，然后在矢量空间给以整体量化，从而压缩数据而不损失多少信息。在传统的预测和变换编码中，首先将信号经某种映射变换成一个数的序列，然后对其一个一个进行标量量化编码。而在矢量化编码中，则是把输入数据几个一组的分成许多组，成组的量化编码，将这些数看成一个k维矢量，然后以矢量为单位逐个矢量进行量化。矢量量化是一种限失真编码，其原理仍可用信息论中的率失真函数理论来分析。而率失真理论指出，即使对无记忆信源，矢量量化编码也总是优于标量亮化。

3.4 本章小结

本章首先给出语音预处理的一般方法，在此基础上详细阐述基于改进Shazam算法的声纹识别方法，主要包括声纹特征的提取和检索匹配算法。其次给出了语音识别的实现，主要包括两个方面，特征参数的提取和模板匹配，这两个方面是语音识别中两个重要的过程。到此，基于Android的声纹识别和语音识别系统的实现的原理在本章中已经阐述完了，下章主要讲述系统的实现部分。

第四章 基于Android的声纹识别和语音识别系统的实现

本章首先给出整个系统实现框架图，在该框架图中首先通过声纹识别系统验证身份，然后再通过语音识别系统识别语音内容，根据具体识别的语音内容，系统进行相关操作。

4.1 系统概述

图4-1 系统框架

4.1.1 整体系统框架及概述

上图所示，系统以Android系统为开发平台，在此基础上实现语音识别和声纹识别，进而提取语音中的关键词信息实现操作相关应用的功能。首先是声纹识别，主要包括特征参数的提取和匹配识别，识别出待识别人的身份；其次进行语音识别，识别出语音的具体内容和语义，再通过语音解析，本文采用的是JSON解析，得到语音的关键字，最后根据语音关键词实现相关应用。例“提醒我明天下午三点开会”，这个待识别的语句，首先通过声纹识别系统判断说话人是否为系统最初设置的说话人，如果是，再通过语音识别系统识别出该段语音的文字内容，JSON解析得到该语句中的关键字，“提醒”、“明天下午三点开会”、“开会”。最后系统会根据具体关键词信息实现相关应用，例如如果本系统听到这段语音，就会自动添加一个明天下午三点的闹钟，其中内容是“开会”。如果声纹识别系统判断出该说话人不是系统最初设置的说话人，就会拒绝执行以上动作。

4.1.2 Android应用开发流程

Android应用开发实际原理框架图如图4-2所示。



图4-2 Android应用开发流程图

4.2 声纹识别模块实现

4.2.1 数据库设计

数据库从问世到如今，经历了许多结构上的变化。从早期的层次数据库和网状数据库到现在使用最多的关系型数据库和面向对象数据库。理论上，面向对象数据库管理系统在理论上是可行的的，但是还存在一些问题待解决，比如有效性方面的不足，直接导致该系统使用的不是很广泛。在实际应用中，关系型数据库是使用最多的数据库组织方式。关系型数据库实质上是一张二维表格，其中表格中的行表示现实世界中的概念或者事物，列表示这些概念或事物的属性。二维表中概念和事物之间的关联则用附加列和附加表来表示。E-R（Entity-Relationship）图既是实体-关系图可以很好的表示各个实体或者概念之间的关系，在数据库设计中是最基本、最常用的一种设计图。下面使用E-R图设计出本系统的数据库。



图4-3系统E-R图

（一）E-R图设计

本系统设计的实体包括：

1）普通用户：用户ID、用户姓名、用户性别、语音模型ID；

2）语音片段：语音片段ID、语音采样率、语音量化位数、语音长度；

3）语音模型：语音模型ID、语音模型参数、语音模型位宽；

4）数据库模板：语音模型ID、语音模型参数、语音模型位宽。

（二）数据表的设计

根据系统的E-R图可以设计出响应的数据表，下表中给出了各个字段的含义和数据类型。

表4.3普通用户数据表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 是否为主键 | 类型 | 长度 | 缺省值 | 允许空 |
| UserID | 用户ID | 是 | char | 6 | 无 | 否 |
| UserName | 用户姓名 | 否 | varchar | 10 | 无 | 否 |
| UserGender | 用户性别 | 否 | char | 2 | 无 | 是 |
| SpeechModID | 语音模型ID | 否 | varchar | 30 | 无 | 是 |

表4.4语音片段数据表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 是否为主键 | 类型 | 长度 | 缺省值 | 允许空 |
| SpeechID | 语音片段ID | 是 | char | 6 | 无 | 否 |
| SpeechSamRate | 语音采样率 | 否 | float | 5 | 无 | 否 |
| SpeechQuanBit | 语音量化位数 | 否 | int | 2 | 无 | 否 |
| SpeechModID | 语音长度 | 否 | int | 6 | 无 | 否 |

表4.5语音模型数据表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 是否为主键 | 类型 | 长度 | 缺省值 | 允许空 |
| SpeechModID | 语音模型ID | 是 | char | 6 | 无 | 否 |
| SpeechModPara | 语音模型参数 | 否 | float | 6 | 无 | 否 |
| SpeechModWidth | 语音模型位宽 | 否 | int | 2 | 无 | 否 |

表4.6数据库模板数据表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 含义 | 是否为主键 | 类型 | 长度 | 缺省值 | 允许空 |
| SpeechModID | 语音模型ID | 是 | char | 6 | 无 | 否 |
| SpeechModPara | 语音模型参数 | 否 | float | 6 | 无 | 否 |
| SpeechModWidth | 语音模型位宽 | 否 | int | 2 | 无 | 否 |

4.2.2 Android语音采集模块实现

Android语音采集模块可以用Android系统本身提供的MediaRecord类来实现，默认保存的格式是“.amr”，但系统处理的语音文件要求是“.wav”格式，其他格式的语音文件不便于处理，且“.amr”不是专业的语音文件，不便于对音频进行处理。如果想要实现特定格式的语音文件，要进行音频的处理或者音频的转换，且要自定义文件格式，并在写文件时加入特征语音格式的文件头或者文件尾。本文中系统的采集模块使用Android的AudioRecord来进行录音，然后按照“.wav”的格式进行封装。AudioRecord类和MediaRecord类相比，更加底层，其本身的方法很少，对实现一些指定格式的语音的录取很方便。

指定录制音频的格式可以用AudioFormat类来表示：



其中参数的含义为：

AudioFormat.Encoding.PCM\_SIGNED：采用脉冲编码调制技术，指定无符号的线性PCM数据；

SampleRate：采样频率，既是每秒钟采样的样本数，上文中已经说明，系统需要的采样频率为44100Hz；

SampleSizeInBits：每个样本的位数，既是用几位二进制表示，本文中采用16来表示；

Channel：声道数，既是单通道还是立体通道，本文中取值为2，采用立体通道；

FrameSize：每秒帧数。本文中采用系统默认值；

FrameRate：每帧包含的字节数，本文中采用系统默认值；

false：表示，在存储数据时用小端格式进行存储。

由于Android有对权限的保护机制，在进行语音采集时要记得申明权限：



其大致过程为，首先配置录音参数，主要有采样率、通道数、数据格式、大小端等。其次调用Android SDK自带的AudioRecord API接口获得录音对象，然后调用该对象的startRecording方法开启录音功能，同时设置录音状态为true，表示处于录音状态，开启音频文件写入线程；录制完成后调用录音对象的stop方法停止录音并将音频文件保存在SD卡中，以系统时间来命名区分不同时刻录制的音频文件。再调用release方法释放资源，将AudioRecord对象置空以方便下次继续使用，本文中通过线程来控制和实现开始录音和停止录音。

Java实现该功能的核心代码如下：

 在上述代码中，通过TargetDataLine来读取音频数据，读取的代码如下(running 变量是一个全局变量，由其他线程控制)，下图是根据上述代码实现的录制音频图示。



图4-4 录音图示

 上述中读取音频文件的各项参数的含义在本节中都已说明。

4.2.3 声纹识别核心算法模块实现

本文中声纹识别的核心算法模块主要包括两部分，其一是语音特征的提取；其二是检索匹配算法。语音的录取在上节中已将叙述过，只要按照个参数设置好即可

上述语音录取所得文件中保存的是时域信号，但是对我们分析语音数据用处不大，频域可以表示为时域信号的指纹或者签名，它提供了一种静态数据表示动态信号的方法，时域到频域的变换，可以用FFT快速傅里叶变换，目前最常用的算法是Cooley-Tukey algorithm，用Java语言实现的核心代码为：



在进行FFT变换之后，得到语音信号的频域信息，但是经过FFT变换有一个很大的缺陷就是丢失了原始语音信号的时间信息，例如，对一段语音来说，经过FFT变换之后看到的只是一系列的频率和频率的幅值，但是无法知道这些频率具体在该语音片段中的位置，但是这些位置信息很重要，因为正是这些位置信息决定了这段语音，故在本文中引入另一种技术：滑动窗口。滑动窗口只是对其中的一块原始语音信号进行FFT变换，数据块的大小可以通过多种方式确定，本文中为了满足系统要求，采用双声道、16-bit精度、44100Hz采样。此时1s的数据大小为44100\*2byte\*2声道，约为176kB,选择4kB当做数据块大小，每秒种需要对44个数据块进行FFT变换，这样的切分密度已经足够适应本文系统的要求。

下面是用Java实现的对分块的数据进行FFT变换的核心代码：



代码的内层循环将采样数据放入一个复数数组中(虚部为0)，外层循环遍历每一块数据块，并进行FTT变换。在对每一帧语音信号进行FFT变换之后，就可以构造音频指纹，这是声纹识别算法中最核心的部分，构造指纹最大的挑战是在于怎样从众多的频率中选出区分度最大的来，本文中采用最大幅值的频率作为构造音频指纹的基础。但幅值较大的频率跨度可能很广，从低音(32.70Hz)到高音 (4186.01Hz)都可能出现。为避免分析整个频谱，本文中将频谱分成多个子代，从每个子带中选择一个频率峰值。子带的范围为：低音子带为30-40Hz、40-80Hz和80-120Hz，中音和高音子带分别为120Hz-180Hz和180Hz-300Hz。每个子带的最大频率就构成了这一帧信号的签名，而这个签名又是整段语音信号指纹的一部分。

算法的核心代码如下：

 以上既是声纹识别算法中的核心代码模块。当提取完语音特征，存储在数据库中，作为模板使用，当进行声纹识别时，对原始语音同样进行上述相同的操作，提取出该段语音指纹，然后和模板库中语音指纹作比较(主要是对HashTag值作比较)，进而识别出结果。

4.3 语音识别模块实现

本文语音模块的实现是基于科大讯飞的语音SDK，由于该语音SDK在语音识别上有非常好的性能，能准确的识别出语音的内容和语义，符合本文对语音识别功能的实现要求，故本文中采用科大讯飞的语音SDK作为本文语音识别的功能实现部分。

4.3.1 环境搭建

本文中语音识别是基于科大讯飞语音SDK，要使用该语音SDK中的接口，只需要按照如下配置进行环境搭建即可。

1. 导入SDK

官网下载科大讯飞语音SDK，并将开发工具包中的libs目录下的Msc.jar和armeabi复制到Android工程的libs目录中。

1. 添加用户权限

在工程AndroidManifest.xml文件中添加如下权限：



（三）初始化

初始化既是创建语音配置对象，只有初始化之后才能使用语音识别的各项服务，通常将初始化放在程序的入口处。使用权限ID要换成申请的ID。



4.3.2 消息处理

本文中的消息处理指的是对语音识别后的返回数据进行相关处理，其中的重点是解析Json，因为在科大讯飞语音SDK中，对消息的封装和解析都是通过Json格式进行的。在进行声纹识别后，如果和数据库中存储的说话人的语音模型相匹配，则进行语音语义的解析，语音识别消息的处理主要包括语音的解析和关键词的提取，参考语音SDK的说明文档可知，语音识别后其结果以Json的形式返回，比如在本文中语音识别的解析结果，例如：“提醒我明天下午三点开会”，其形式为：

 对返回的Json数据进行解析，并提取其中的关键词，在此例子中为，data、content、name等，然后根据关键字的具体内容实现相关的功能，本例中“提醒我明天下午三点开会”这段语音，在说话者说完之后，语音识别系统提取关键词，系统会自动可打开闹钟APP，设置下午三点的闹钟。

4.4 本章小结

本章主要说明声纹识别和语音识别所涉及到的算法的实现，系统首先通过声纹识别判定是否是说话人本人，如果是则进行语音识别，再提取语音信息中的关键词，系统根据关键词会自动完成相关功能；如果声纹识别的结果不是说话者本人，则拒绝进行以后操作，要求说话者重新进行声纹识别。其中声纹识别的关键是在声纹特征的提取，本文中采用的是对不同频率范围内的频率分别求出最大幅值的频率，再对这些子带最大的频率求出它们的Hash值并存储，其后匹配的过程就是检索数据库匹配Hash值的过程。

第五章 基于Android的声纹识别和语音识别系统测试分析

上一章节对系统进行了实现，本章节首先搭建测试平台，并对基于Android的声纹识别和语音识别系统的进行了测试分析，表明本文设计的可实现性。

5.1 实验条件

5.1.1 硬件环境

系统的硬件开发环境中PC机的配置如表5.1所示：

表5.1 PC机环境配置

|  |  |
| --- | --- |
| 电脑型号 | 联想B475商务笔记本 |
| CPU | @1.9GHz |
| 处理器 | AMDA4-3300M |
| 内存 | 4GB |
| 操作系统 | Window7 64位操作系统 |

Android手机环境配置如下表所示：

表5.2 Android手机环境配置

|  |  |
| --- | --- |
| 手机型号 | Nexus6P |
| 屏幕 | 5.7英寸（2560\*1400像素） |
| 摄像头 | 前置800万像素，后置1230万像素 |
| 处理器 | 高通骁龙810 |
| 内存 | 3GB |
| 架构 | Cortex-A57 |
| 操作系统 | Android6.0 |

5.1.2 软件环境

本系统中用到的软件环境如表5.3所示，其中JDK1.8作为JAVA语言环境，Android Studio 2.0作为大型开发平台，运行在JDK1.7版本的JAVA语言环境中，SDK版本采用较为常用的的Android SDK R21版本。

表5.3 软件平台的环境配置

|  |  |
| --- | --- |
| JDK版本 | JDK1.8 |
| Android Studio版本 | Android Studio 2.0 |
| Android SDK | Android SDK21 |

5.1.3 实验素材

实验中采集了10个人的语音样本，语音样本的内容相同，按照每个人正常的语速即可，同时每人录10个不同的语句，每句约为10秒左右，分成10组，每一组中包括10人不同的语音样本。其中，语音每组的第一个样本作为待识别的语句，实验数据如图5-1所示，录音是在噪声分贝为35-40dBA的环境下进行的。

5.2 声纹识别系统测试

本文中对声纹识别系统做两种情况下的测试，一种是待识别语音和模板库中语音开始时间点一致；另一种是待识别语音为模板库中语音的任意一部分，时常为10s。

5.2.1 待测语音和模板语音起始时间一致时的性能测试

首先验证待识别的语音和数据库中存储的语音模板的开始端时间点一致时的识别效果。下图是实验所用到的数据。

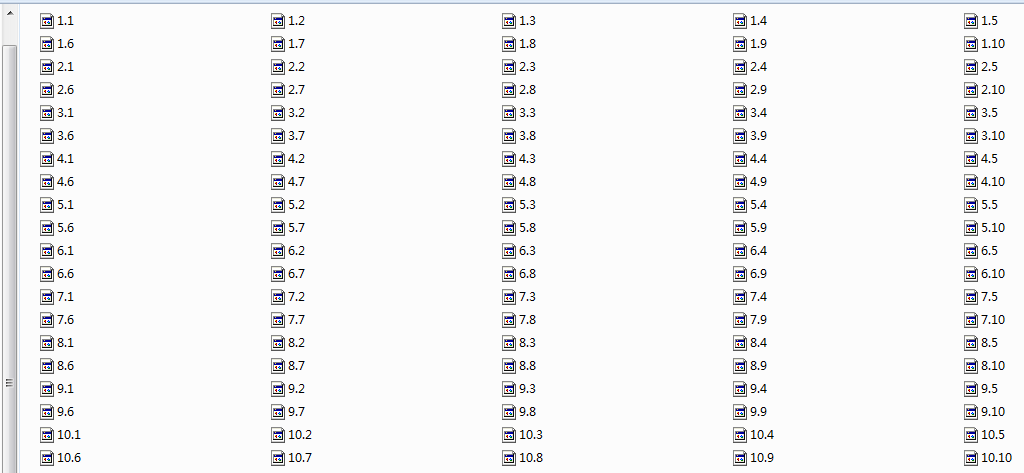


图5-1实验数据

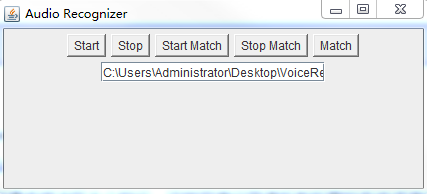


图5-2 声纹识别界面

首先对10段语音进行特征提取并保存，形成声纹识别语音模板库；其次对待识别的语音进行同样的操作，并和模板库中保存的数据进行对比，进而得到结果。本文中是以匹配点数的多少来对结果进行排序，匹配点越多，说明越相似；反之说明不同。各个按钮的含义如下：

Start：表示开始对文本框中该路径所表示的语音进行特征提取；

Stop：表示停止对文本框中该路径所表示的语音进行特征提取；

Start Match：表示开始对待识别的语音进行特征提取，如果系统麦克风打开，则使用系统录音；如果系统麦克风没有打开，则使用文本框中该路径中所指向的语音；

Stop Match：表示停止对待识别的语音进行特征提取；

Match：表示把待识别的语音和模板库中的语音模板进行对比。

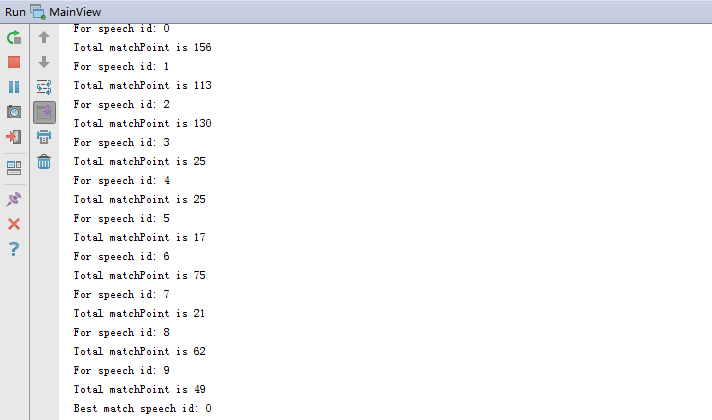
依据上述过程检测，对第一组实验数据进行验证，待识别语音ID为1.1，得到如图5-3所示结果： 

图5-3相同起始时间点声纹识别结果

从上图可以看出，识别结果符合预，正确的识别出了待识别语音，和上述理论分析中是一样的。同样以相同的步骤进行余下数据的验证。

待识别语音ID为1.1，对第一组数据得到结果如下：

表5.4 第一组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 1.10 | 匹配ID |
| 匹配数 | 156 | 113 | 130 | 25 | 25 | 17 | 75 | 21 | 62 | 49 | 1.1 |

待识别语音ID为2.7，得到第二组数据结果：

表5.5 第二组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 | 2.8 | 2.9 | 2.10 | 匹配ID |
| 匹配数 | 287 | 19 | 47 | 3 | 117 | 130 | 100 | 204 | 26 | 106 | 2.1 |

待识别语音ID为3.1，得到第三组数据结果：

表5.6 第三组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 3.1 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9 | 3.10 | 匹配ID |
| 匹配数 | 55 | 13 | 18 | 25 | 43 | 2 | 39 | 11 | 11 | 10 | 3.1 |

待识别语音ID为4.1，得到第四组数据结果：

表5.7 第四组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 4.1 | 4.2 | 4.3 | 4.4 | 4.5 | 4.6 | 4.7 | 4.8 | 4.9 | 4.10 | 匹配ID |
| 匹配数 | 81 | 62 | 9 | 5 | 1 | 12 | 18 | 63 | 1 | 14 | 4.1 |

待识别语音ID为5.1，得到第五组数据结果：

表5.8第五组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 5.1 | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 | 5.8 | 5.9 | 5.10 | 匹配ID |
| 匹配数 | 121 | 35 | 71 | 37 | 46 | 82 | 49 | 35 | 27 | 45 | 5.1 |

待识别语音ID为6.1，得到第六组数据结果：

表5.9 第六组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 6.1 | 6.2 | 6.3 | 6.4 | 6.5 | 6.6 | 6.7 | 6.8 | 6.9 | 6.10 | 匹配ID |
| 匹配数 | 113 | 13 | 25 | 2 | 38 | 17 | 22 | 9 | 54 | 16 | 6.1 |

待识别语音ID为7.1，得到第七组数据结果：

表5.10 第七组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 7.1 | 7.2 | 7.3 | 7.4 | 7.5 | 7.6 | 7.7 | 7.8 | 7.9 | 7.10 | 匹配ID |
| 匹配数 | 128 | 17 | 101 | 26 | 96 | 15 | 20 | 96 | 25 | 118 | 7.1 |

待识别语音ID为8.1，得到第八组数据结果：

表5.11 第八组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 8.1 | 8.2 | 8.3 | 8.4 | 8.5 | 8.6 | 8.7 | 8.8 | 8.9 | 8.10 | 匹配ID |
| 匹配数 | 94 | 33 | 56 | 10 | 32 | 10 | 5 | 18 | 28 | 17 | 8.1 |

待识别语音ID为9.1，得到第九组数据结果：

表5.12 第九组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 9.1 | 9.2 | 9.3 | 9.4 | 9.5 | 9.6 | 9.7 | 9.8 | 9.9 | 9.10 | 匹配ID |
| 匹配数 | 61 | 26 | 14 | 53 | 29 | 28 | 10 | 31 | 43 | 8 | 9.1 |

待识别语音ID为10.1，得到第十组数据识别结果：

表5.13 第十组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 10.1 | 10.2 | 10.3 | 10.4 | 10.5 | 10.6 | 10.7 | 10.8 | 10.9 | 10.10 | 匹配ID |
| 匹配数 | 45 | 9 | 26 | 4 | 5 | 15 | 2 | 2 | 13 | 3 | 10.1 |

在此情况下得到声纹识别的识别率为100%。

5.2.2 待测语音为部分模板语音时的性能测试

待测语音为部分模板语音时，截取任意模板语音中10s，进行声纹识别测试，测试数据如下，待测语音ID为1.11-10.11。

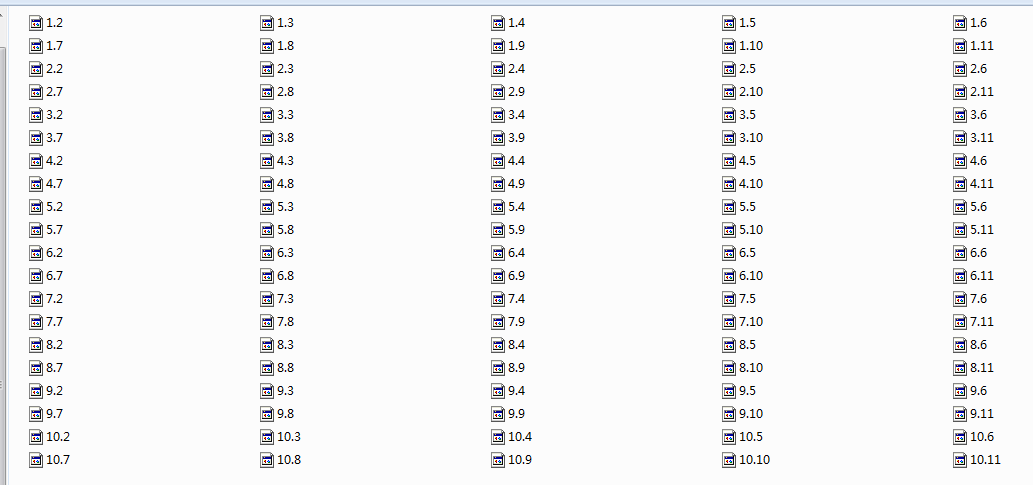


图5-4 实验数据

测试过程和上节测试过程一致，得到如下结果。

待识别语音ID为1.11，对第一组数据得到结果如下：

表5.14 第一组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 1.11 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 1.10 | 匹配ID |
| 匹配数 | 178 | 93 | 88 | 8 | 78 | 77 | 75 | 9 | 67 | 32 | 1.11 |

待识别语音ID为2.11，得到第二组数据结果：

表5.15 第二组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 2.11 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 | 2.8 | 2.9 | 2.10 | 匹配ID |
| 匹配数 | 267 | 23 | 13 | 34 | 67 | 122 | 112 | 89 | 78 | 68 | 2.11 |

待识别语音ID为3.11，得到第三组数据结果：

表5.16 第三组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 3.11 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9 | 3.10 | 匹配ID |
| 匹配数 | 98 | 15 | 36 | 3 | 56 | 55 | 33 | 26 | 32 | 15 | 3.11 |

待识别语音ID为4.11，得到第四组数据结果：

表5.17 第四组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 4.11 | 4.2 | 4.3 | 4.4 | 4.5 | 4.6 | 4.7 | 4.8 | 4.9 | 4.10 | 匹配ID |
| 匹配数 | 178 | 32 | 21 | 8 | 13 | 9 | 78 | 21 | 33 | 65 | 4.11 |

待识别语音ID为5.11，得到第五组数据结果：

表5.18 第五组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 5.11 | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 | 5.8 | 5.9 | 5.10 | 匹配ID |
| 匹配数 | 120 | 43 | 21 | 43 | 28 | 98 | 28 | 19 | 87 | 37 | 5.11 |

待识别语音ID为6.11，得到第六组数据结果：

表5.19 第六组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 6.11 | 6.2 | 6.3 | 6.4 | 6.5 | 6.6 | 6.7 | 6.8 | 6.9 | 6.10 | 匹配ID |
| 匹配数 | 178 | 82 | 67 | 6 | 65 | 27 | 18 | 6 | 63 | 19 | 6.11 |

待识别语音ID为7.11，得到第七组数据结果：

表5.20 第七组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 7.11 | 7.2 | 7.3 | 7.4 | 7.5 | 7.6 | 7.7 | 7.8 | 7.9 | 7.10 | 匹配ID |
| 匹配数 | 89 | 67 | 26 | 7 | 8 | 16 | 12 | 52 | 25 | 19 | 7.11 |

待识别语音ID为8.11，得到第八组数据结果：

表5.21 第八组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 8.11 | 8.2 | 8.3 | 8.4 | 8.5 | 8.6 | 8.7 | 8.8 | 8.9 | 8.10 | 匹配ID |
| 匹配数 | 189 | 23 | 78 | 66 | 46 | 19 | 28 | 118 | 19 | 21 | 8.11 |

待识别语音ID为9.11，得到第九组数据结果：

表5.22 第九组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 9.11 | 9.2 | 9.3 | 9.4 | 9.5 | 9.6 | 9.7 | 9.8 | 9.9 | 9.10 | 匹配ID |
| 匹配数 | 90 | 18 | 19 | 5 | 37 | 10 | 1 | 15 | 23 | 7 | 9.11 |

待识别语音ID为10.11，得到第十组数据识别结果：

表5.23 第十组数据识别结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 10.11 | 10.2 | 10.3 | 10.4 | 10.5 | 10.6 | 10.7 | 10.8 | 10.9 | 10.10 | 匹配ID |
| 匹配数 | 289 | 29 | 18 | 3 | 15 | 23 | 27 | 98 | 112 | 42 | 10.11 |

在此情况下得到声纹识别的识别率为100%.

5.3 系统的整体运行测试

对系统进行整体测试时要集成语音识别和声纹识别功能，并编写相关软件。对系统做功能测试，以软件测试常用的测试用例“白盒测试”为标准，对系统的功能和性能进行完整的测试，测试的范围是由内到外、由小到大、由程序开发人员到一般用户。本文中，分别对系统进行了单元测试、功能测试、集成测试、场景测试、系统测试等多种测试。

单元测试：在本系统的开发阶段，每当实现一个函数，随机对该函数的功能进行测试分析，并分析运行结果，在此基础上进行下一步的代码的编写，例如上一章节中对数据库和核心算法等重要函数的设计。

功能测试：本系统是按模块功能和整体设计划分的，每实现一个功能模块，就对其进行模块测试，并分析结果，例如上一章中对声纹识别核心算法功能模块的测试，分别进行十组，100个语音范例的识别测试，用来统计该声纹识别算法的识别性能。

集成测试：当实现系统的各个模块之后，就要按照事先设计好的逻辑关系对各个模块进行集成，集成之后对系统的整体性能进行测试，其中的重点是验证相互依赖的模块。

场景测试：完成系统的集成测试后，还要验证系统中关键模块能否完成用户场景，本文随机选取10人的声纹进行识别，以验证系统的声纹识别性能。

系统测试：系统测试既是对整个系统的性能进行测试，首先验证是否为说话人，在此基础上进行语音识别打开相关应用。本文中对10人的语音进行声纹识别，每人说10段语音，分成10组进行声纹识别，测试结果如下表所示。

表5.24 系统测试声纹识别结果

|  |  |  |  |
| --- | --- | --- | --- |
| 待识别ID | 实验次数 | 正确次数 | 识别率（%） |
| 1.1 | 100 | 97 | 97 |
| 2.1 | 100 | 93 | 93 |
| 3.1 | 100 | 98 | 98 |
| 4.1 | 100 | 97 | 97 |
| 5.1 | 100 | 95 | 95 |
| 6.1 | 100 | 100 | 100 |
| 7.1 | 100 | 97 | 97 |
| 8.1 | 100 | 93 | 93 |
| 9.1 | 100 | 90 | 90 |
| 10.1 | 100 | 99 | 99 |
| 平均 | 100 | 95 | 95.9 |

从该表中可以得到系统的整体平均识别率为95.9%。

在此基础上进行语音识别，进而依据说话人指令执行具体动作，例如“帮我查询一下明天的天气”等语音指令，下表为测试语音识别得到的效果：

表5.25 语音识别的识别结果

|  |  |  |  |
| --- | --- | --- | --- |
| 打开软件 | 实验次数 | 正确打开软件次数 | 识别率（%） |
| 天气 | 100 | 100 | 100 |
| 闹钟 | 100 | 99 | 99 |
| 厨房 | 100 | 98 | 98 |
| 音乐 | 100 | 100 | 100 |
| 视频 | 100 | 100 | 100 |
| 旅行 | 100 | 98 | 98 |
| 幼儿教育 | 100 | 99 | 99 |
| 安防 | 100 | 100 | 100 |
| 平均 | 100 | 99 | 99．25 |

因为语音识别是在声纹识别正确的基础上进行的，因此最后系统的识别率为两者正确识别率之积为95.18%。

5.4 实验结果分析

从实验数据可以得到，系统的整体识别率达到了最初的设计要求。错误识别或拒识别的原因，一是实验素材中部分语音的噪声部分较大，二是实验素材中识别人说话的语调、语速等和系统模板库中的原始语音相差较大，当达到一定程度时，就会出现错误识别的情况。从得到的识别结果可知，本文中采用改进的Shazam算法在文本相关的情况下的声纹识别效果和系统的整体识别效果较好，达到系统要求。

5.5 本章小结

本章对基于Android的声纹识别和语音识别系统进行测试，分别进行了声纹识别测试和整个软件系统测试，并对测试结果进行了分析，结果符合预期，达到了系统最初的设计要求。

第六章 总结与展望

6.1总结与展望

声纹识别是近年来身份识别中发展较快的一种识别方法，是生物识别的一种，特点是信号采集简单、使用方便、快速便捷。随着手持移动端的普及和发展，声纹识别已经成为一个很好的应用点。和移动平台相结合的声纹识别技术是现代生物识别的一种发展趋势，加强移动应用方面的声纹识别技术对于推动生物识别技术的发展起着非常重要的作用，因此本文从Android平台着手，进行了软件的系统性设计，基于Android平台开发了声纹识别和语音识别系统，本文主要完成了一下工作：

1）首先分析了语音信号的处理过程，并在此基础上建立了声纹识别的模型；

2）研究了比较流行的声纹识别算法，通过比较，结合移动应用的特点，本文选择基于Shazam算法的声纹识别；

3）搭建基于Android的声纹识别和语音识别系统，并设计友好的操作界面，并完成整个系统的测试，获得了较好的识别率，达到了最初的设计目的。

目前，国内外的许多专家在声纹识别和语音识别方面做了很多的研究工作，也取得了很大的

就。本文针对声纹识别进行了基于Android的移动应用开发，并进行了尝试性的研究工作，完成了与文本相关的声纹识别和语音识别系统的开发，并取得了较为满意的识别结果。其优点是算法简单，识别率高，需要存储的数据较少，比较适合规模较大场景。但是设计中还是存在一些不足：本文中的声纹识别是与文本相关的，与文本无关的声纹识别识别率并不是很好且本文中的识别率受到语音快慢的影响。本文后续工作如下：

1）改进Shazam算法，使得提取的声纹特征与文本无关，使得系统能够不局限于语音的内容，通过辨别说话者的音色进行判断；

2）寻找高效的自适应算法和降噪算法来提高系统性能的稳定性。

致谢

时间转瞬即逝，研究生三年的生活也会快就要过去了，我的研究生毕业设计论文也即将完成，在这里我要感谢在研究生期间所有给与我帮助和关怀的人。

首先，我要非常的感谢我的导师万隧人教授，感谢他在我读研的三年期间给与我学术上悉心指导和生活上无微不至的关怀，在学校，他不仅为我们提供了良好的学习环境和学习条件，平时还不断的指导我们学习和研究，让我们扎扎实实，刻苦努力的完成自己的课程和科研任务。他在学术上严禁求学的学术态度和扎实的学术知识以及认真的科研精神都深深的影响着我，这在我以后的工作和生活中将会产生极大的影响，同时在为人做事上给我提升的好榜样，在这里我要对万老师致以崇高的敬意和诚挚的谢意。

我还要感谢对我的学习和工作予以不断指导的刘勇和王刚老师，是他们耐心的督促和指导我完成学业和科研上的各项工作，使我较快的找到自己的研究方向，并尽快的融入学习和研究的生活，同时生活上给予我无微不至的照顾，关心我们的学习、生活、科研状况。在他们的帮助下我才能顺利的完成研究生三年的学习任务和科研任务。

同时还要感谢实验室的同学，他们在我研究生三年的学习和生活中给予了我许多的帮助和快乐，正是和他们在学习和生活中互帮互助、相互照应，我们才能找到自己心仪的工作并完成学业论文。同时感谢我的室友丁磊、郭安、金涛，感谢他们在研究生三年中给予我在生活和学习上的帮助，感谢他们给予我那么多美好的记忆。

最后我要由衷的感谢我的家人，感谢他们这么多年来对我默默无闻的奉献，在这里我要由衷的对他们说一声谢谢！

参考文献

1. 王敏妲.语音识别技术的研究与发展[J].微型机与应用，2009,28(23):1-2
2. 裴鑫.声纹识别系统关键技术研究[D].哈尔滨理工大学，2014
3. 钟明林.基于Android智能手机平台方案[D].山东大学，2010
4. 彭诗雅.基于声纹识别的身份认证技术研究[D].南京航空航天大学，2010
5. Proakis J G, Manolakis D G. Digital signal processing (3rd ed.): principles, algorithms, and applications[M]. Prentice-Hall, 1996
6. 郭慧娟.声纹识别的特征参数提取[J].太原师范学院学报(自然科学版)，2008,7(1):70-73
7. 李韵.声纹识别系统中特征参数提取方法的对比分析研究[D].成都理工大学，2016
8. Muda L, Begam M, Elamvazuthi I. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques[J]. Ttps, 2010, 2
9. 彭辉，魏玮，陆建华．特定人孤立词的语音识别系统研究[J].控制工程，2011,18(3):397-400
10. Wang A. An Industrial Strength Audio Search Algorithm[C]// Ismir 2003, International Conference on Music Information Retrieval, Baltimore, Maryland, Usa, October 27-30, 2003, Proceedings. DBLP．2003
11. 陈立伟，赵春晖，孙玲，等.一种语音信号线性预测系数的求解新方法[J]．应用科技，2005,32(8):12-13
12. 唐晓进.基于LPC倒谱的语音特征参数提取[J].山西电子技术，2012(6):15-16
13. 刘亚丽，杨鸿武，黄德智.基于加权Mel倒谱系数的说话人识别[J].计算机应用与软件，2009,26(9):24-27
14. Wang A. An Industrial Strength Audio Search Algorithm[C]// Ismir 2003, International Conference on Music Information Retrieval, Baltimore, Maryland, Usa, October 27-30, 2003, Proceedings. DBLP, 2003
15. Rabiner L, Juang B H. Fundamentals of speech recognition[J]. Englewood Cliffs N J, 2009, 1(1):353-356
16. 王坤卿.HMM模型在语音识别研究中的应用[J].电脑知识与技术，2008,4(34):444-446
17. 于娴，贺松，彭亚雄，等.基于GMM模型的声纹识别模式匹配研究[J].通信技术，2015,48(1):97-101
18. 林子明.基于GMM和VQ的说话人识别系统的研究[D].华南理工大学，2012
19. 李昆仑，黄厚宽，田盛丰.模糊多类SVM模型[J].电子学报，2004，32(5)：830-832
20. 郭华.基于神经网络的声纹识别系统设计与实现[D].上海交通大学，2008
21. Wang A. An Industrial Strength Audio Search Algorithm[C]// Ismir 2003, International Conference on Music Information Retrieval，Baltimore, Maryland, Usa, October 27-30, 2003, Proceedings. DBLP, 2003
22. Dahl G E, Yu D, Deng L, et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition[J]. IEEE Transactions on Audio Speech & Language Processing, 2012, 20(1):30-42
23. Graves A, Mohamed A, Hinton G. Speech Recognition with Deep Recurrent Neural Networks[J]. 2013, 38(2003):6645-6649
24. 刘雅琴，智爱娟.几种语音识别特征参数的研究[J].计算机技术与发展，2009,19(12):67-70
25. White D J, King A P, Shan D D. Voice recognition technology as a tool for behavioral research[J]. Behavior Research Methods, 2002, 34(1):1
26. 魏星，周萍.语音识别系统及其特征参数的提取研究[J].计算机与现代化，2009(9):167-168
27. 李萱.语音特征参数提取方法研究[D].西安电子科技大学，2006
28. 肖利君.基于DTW模型的孤立词语音识别算法实现研究[D].中南大学，2010
29. 宋小倩，周东升.基于Android平台的应用开发研究[J].软件导刊，2011,10(2):104-106
30. Smyth N. Android Studio Development Essentials[M]. CreateSpace Independent Publishing Platform, 2014
31. 郭霖.第一行代码[M].人民邮电出版社，2014
32. 杨睿娜.数据库设计过程中E-R图向关系模型的转换[J].硅谷，2009(11):32-48
33. RaghuRamakrishnan, JohannesGehrke,罗摩克里希纳,等.数据库管理系统原理与设计[M].清华大学出版社, 2004
34. 姚世明.JSON在Android和Web通信中的应用研究[J].通讯世界，2015(23):46-47
35. 封侣.基于JSON格式的自定义组合条件查询设计与实现[J].电子世界，2014(12):79-80