面向智能手机的说话人识别研究

计 111（10111939）陈楚楠

摘要：目前说话人识别研究主要是基于专业录音设备和处理机，来提取用户的语音中的特征并训练模型和识别确认处理。主要适用于大型安全领域。整套解决方案的可移动性不强，且不利于大规模普及和部署。因此本课题旨在研究开发出一个面向智能手机的说话人识别系统，在智能手机上获取用户语音，并在手机上提取特征并训练模型和识别确认处理。从而实现说话人识别方案的大规模普及和快速部署。本课题开发的说话人识别系统主要面向 Android 平台手机用户。

关键词：智能手机，说话人识别，特征提取，训练模型，Android

1 研究背景

1.1 课题背景

说话人识别最初被广泛应用是在司法领域，用于帮助对嫌疑人的查证或判定罪犯，经过不断发展后进入安保和军事领域，用于一些机密场所出入控制，机要设备的使用控制和战场监听等等。

出此之外，说话人识别还有着广阔的市场应用前景。例如在通信和互联网尤其是新兴的移动互联网领域，说话人识别技术可以应用于语音拨号，电话银行，信息服务，安全控制，账户登录，E-mail，即时通信等。

1.2 开发背景

1.2.1 说话人识别

说话人识别是一种生物认证技术。通过语音信号中的波形变化反映说话人生理和应为上的特征，并根据特征识别说话人。这些特征涉及到说话人的年龄，性别，感情，种族等等。

说话人识别的大致过程是首先录入说话人的语音样本，提取其中的语音特征并保存以待应用。在识别时将待测试的语音的特征与保存的语音特征做对比，从而确定说话人身份。

根据识别系统对待识别语音内容的不同，又可以分为文本相关和文本无关两种方式。

文本相关的说话人识别方式要求说话人发音的关键词或关键句子与训练文本相同，且在识别时也要按照相同内容发音。

文本无关的说话人识别方式在训练和识别时都不对说话人的的发音内容做要求，其识别对象是任意的语音信号。

由于文本无关的说话人识别方式的无法控制的特性，而文本相关的识别系统的鲁棒性较强，本课题所研究开发的系统采用文本相关的说话人识别方式。

1.2.2 Android 平台

本课题开发的说话人识别系统是面向 Android 平台手机用户。Android 是一种基于 Linux 的开源的自由的操作系统，主要使用于智能手机，平板电脑，相机，游戏机，电视机等移动设备。最初由由 Andy Rubin 开发，后在 2005 年 8 月被 Google 收购，随后 Google 以 Apache 开源许可证和 GPL 的授权方式，发布了 Android 的源代码。

2008 年，在 GoogleI/O 大会上，谷歌提出了 Android 架构图，在同年 9 月，谷歌正式发布了 Android 1.0 系统。2011 年第一季度，Android 在全球的手机市场份额跃居第一。在 2014 年第二季度，分析机构 Strategy Analytics 发布的智能手机操作系统全球情况报告显示，目前 Android 操作系统的全球市场份额已达 84.6%，是有史以来最高比重。

从 Android 刚刚发布时的 1.0 版本到目前的 5.0 版本，Android 逐渐从一个崭新的系统发展为最成熟的移动端操作系统。

Android 作为市场份额最大的手机端操作系统，得益于其的平台开放性和开发开放性。Google 允许任何手机终端厂商加入 Android 平台联盟，也正因为众多的手机厂商加盟，使得 Android 平台拥有了众多的开发者，开发的 Android 应用更加丰富，用户数越来越多。

Google 提供给 Android 一个相当自由的开发环境，有着不停更新的 Development Toolkit，和较少约束的系统 API，使得开发者可以自由地进行开发。

1.3 研究的目的与意义

通过本课题的面向 Android 平台手机用户的说话人识别系统，可以对用户进行生物身份识别确认以替代原本的手机上的数字密码验证，避免了数字密码遭非法用户窃取的风险。

除此之外本课题还使得生物身份识别系统更简单易用，在安保、司法等以外领域的地方部署，可以极大地促进说话人识别技术的发展。

最后，目前移动互联网行业发展迅速，但是安全性还是原地踏步，本系统还可以与 Android 平台上的其他应用进行绑定，以极大地增强一些移动互联网应用的安全性，例如手机支付钱包等功能。

2 文献综述

2.1 Android 开发

Android 开发有 4 个层次(如图 2.1 所示)，自下而上分别是 Linux 内核层开发，类库层开发，应用框架层开发，应用层开发。

第一层 Linux 内核层开发由 C 实现，主要负责管理 Android 硬件的驱动。

第二层类库层开发依然由 C 实现，主要负责 Android 上一些常用类库和组件的开发，例如数据库 SQLite，浏览器内核 WebKit，常用的 C 函数库 libc，运行应用的虚拟机 Dalvik。

第三层是 Android 系统应用框架，由 Java 实现，包含了 Activity Manager 界面管理，Notification Manager 通知管理，Resource Manager 资源管理等等，实现了对 Linux 内核层和类库层的封装，让 Android 不过度依赖与 Linux 内核，达到内核无关的特性，让 Android 框架开发和 Android 应用开发能在不考虑 Linux 内核和驱动下顺利完成，以吸

引更多的开发者。

第四层是 Android 应用程序的开发，由 Java 实现，可以利用 Android 提供的 API 接口实现各种功能的应用，例如信息服务，拍照，录音，定位等等，以及各种界面控件完成用户交互，例如文本框 EditText，按钮 Button 等等。

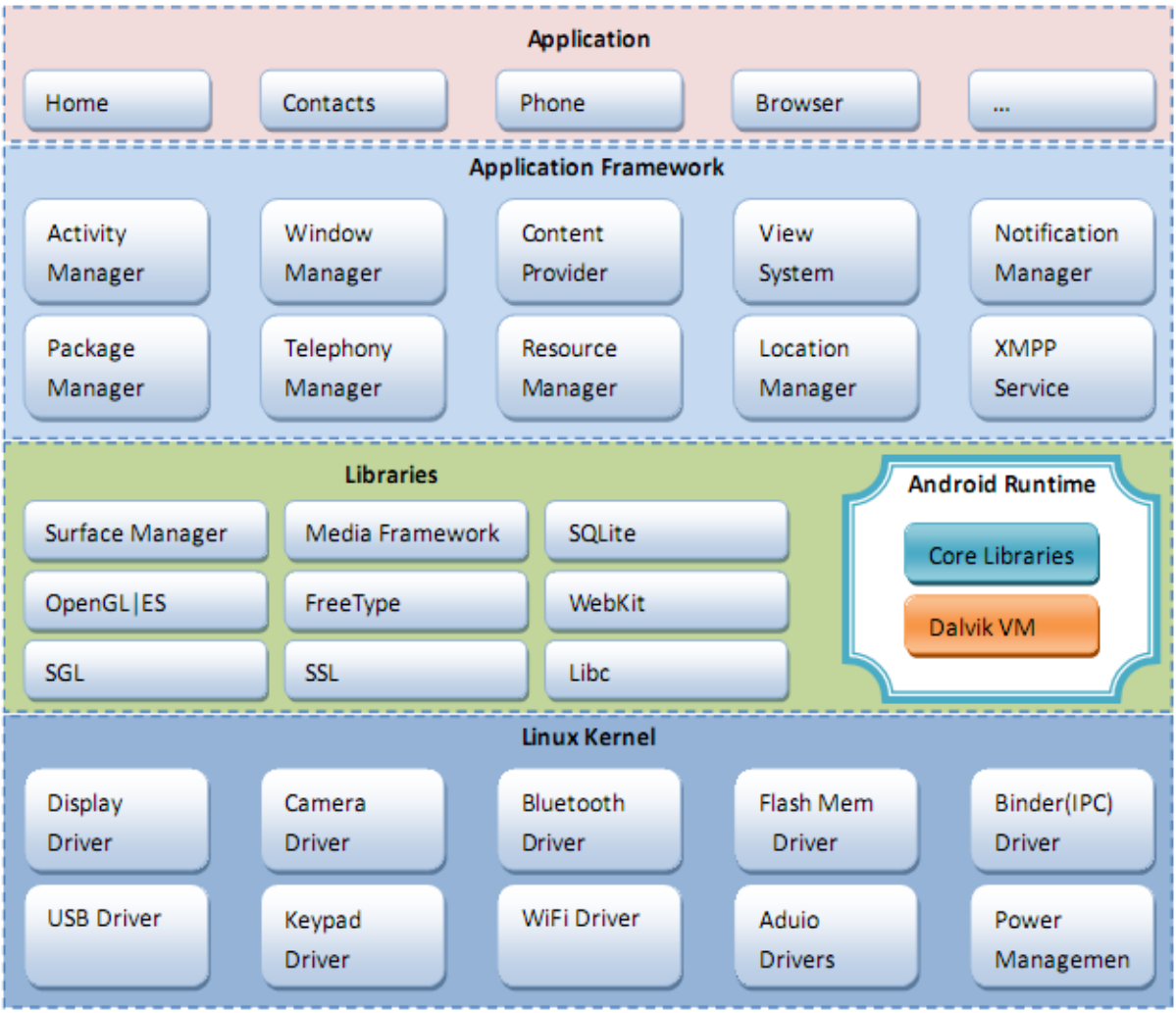

图 2.1 Android 开发层次图

Android 应用程序由 4 大组件构成，分别是 Activity、Service、Broadcast Receiver、Content Provider。Activity 是 Android 应用与用户交互的界面，可以显示一些控件也可以监听并处理用户的事件做出响应。Broadcast Receiver 用来对外部事件进行过滤，从而只获得自己感兴趣的内容。Service 是没有用户界面的程序，主要用来开发如监控类等后台程序。Content Provider 用于给应用提供各种各样的数据，例如 SQLite 数据库，SharedPreferences 配置文件。

本课题的开发的说话人识别系统主要在第二层和第四层完成。先在第二层中移植好 HTK 语音工具箱类库，再在第四层中开发一个应用调用这个类库，完成语音特征提取，训练建模，测试识别等任务。

2.2 隐马尔科夫模型(HMM)

隐马尔科夫模型（Hidden Markov Model，HMM）是一个统计模型，用来描述一个含有隐含未知参数的状态跳转过程。其要点在于从可观察的参数中找出这一系列状态跳转过程中的隐含参数。然后利用这些参数来作进一步的分析，例如语音识别。

在正常的马尔科夫模型中，状态参数对于观察者来说是完全直接可见的。而在隐马尔科夫模型中，状态参数并不是直接可见的，但模型中受状态影响的某些参数变量则是可见的。

例如在某地的某人在雨天喜欢宅在家中，在晴天时喜欢出去踢球，那么当你得知这个人在某一天出去踢球(或宅在家中)，就可以推断出这个人所在的地方的天气是晴天(或雨天)。这个人告诉了你的他的活动，也就是一个受天气影响的参数变量，而我们对天气并不是可知的，这样一个系统就是隐马尔科夫型模型。

在语音识别领域，最常用到的两个模型分别是高斯混合模型和隐马尔科夫模型。高斯混合模型常用于文本无关的语音识别，而隐马尔科夫模型常用于文本相关的语音识别。

通过给定测试序列(测试用户的语音特征序列) 和模型参数(事先训练好的语音特征序列集合)，再利用 Viterbi 算法寻找某种意义上最优的状态序列(测试用户的身份)。这就是在文本相关语音识别领域使用隐马尔科夫模型的过程。

2.3 语音特征提取

语音特征提取是指采用数字技术与模拟技术相结合，选择和提取语音信号中的特征，得到说话人的模型，而不是直接从语音信号中得到。无论是训练时的语音还是测试时的语音，都要通过语音特征提取才能进行后续建模处理。

对于说话人识别，语音信号中提取到的特征需要满足能较好地区别不同用户的能力，又能对相同用户具有同一性。

人们能对不同人的语音做出区别是因为人们的耳蜗类似于一个滤波器组，可以对语音信号进行过滤后找出不同说话人的特征。因此想要在计算机上对语音进行特征提取就要模拟出一个类似人耳的滤波器组。

当频率在 1000HZ 以下时，人们的耳朵的感知能力与频率成线性关系；但当频率在 1000HZ 以上时，人们的耳朵的感知能力与频率不再构成线性关系，而更偏向于对数关系，这就使得人们的耳朵相比于对比高频信号来说对低频信号更敏感，故采用 Mel 频率倒谱系数(即 MFCC)。

Mel 频率倒谱系数提取过程如图 2.2 所示：

1). 对语音信号 s(n)进行预加重，分帧，加窗等处理，得到每个帧的时域信号 x(n)。

2). 将上述对时域信号 x(n)后补 0，直至形成长度为 N 的序列(N 一般为 256)，然后进行快速傅立叶变换(FFT)后得到线性频谱 x(k)：

$$x(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi k}{N}} (0 \le n, k \le N-1) \qquad \text{——公式 2-1}$$

3). 将上一步的线性频谱 x(k)通过 Mel 频谱滤波器，并取每个三角形的滤波器频率带宽内的所有信号幅度加权和作为某个带通滤波器的输出，从而得到 Mel 频谱，并通过对数能量的处理，得到对数频谱 s(m)：

$$s(m) = \ln[\sum_{m=1}^{M=1} |x(k)|^2 H_m(k)](0 \le m < M) \quad \text{——公式 2-2}$$

$H_m(k)$为各个带通滤波器的传递函数。

4). 将上一步的对数频谱 s(m)经过离散余弦变换(DCT)得到倒谱频率，即可得到 Mel 频率倒谱系数 c(n)：

$$c(n) = \sum_{m=1}^{M=1} S(m) \cos(\frac{\pi n\left(m+\frac{1}{2}\right)}{M})(0 \le m < M) \quad \text{——公式 2-3}$$

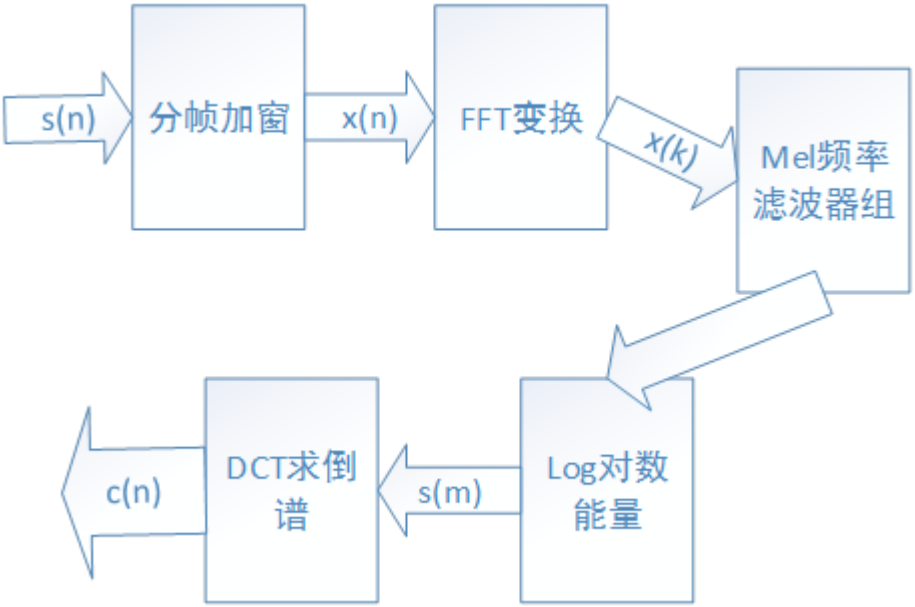

图 2.2 Mel 频率倒谱系数提取过程

3 技术路线

3.1 功能模块

经过简单的需求分析，本课题开发的说话人识别系统主要有一下几个模块(图 3.1)：

- 锁屏界面模块
  通过 Android 系统的 API 对 Android 手机进行锁定，通过 AudioRecord 类实现对用户的语音的记录，以进行特征提取和解码，当通过系统确认时锁定解除。
- 登陆模块
  实现系统的登录界面，要求用户输入密码，防止非法用户登录本系统进行修改。
- 主界面模块
  实现系统的主界面，是其他功能模块的入口所在。
- 训练模块
  实现系统的训练界面，通过 AudioRecord 类实现对用户的语音的记录，并利用

HTKTools 中的 HCopy 进行特征提取处理得到 MFCC 模型，再利用 HTKTools 中的 HInit 和 HRest 依次进行建模训练得到迭代两次过后的 HMM 模型。

- 测试模块
  实现系统的测试界面，通过 AudioRecord 类实现对用户的语音的记录，并利用 HTKTools 中的 HCopy 进行特征提取处理得到 MFCC 模型，再利用 HTKTools 中的 HVite 对 MFCC 模型和原本的 HMM 模型结合打分，得到测试结果。
- 用户管理模块
  实现系统的用户管理界面，通过与数据库的关系映射实现用户的添加删除功能。
- 设置模块
  实现系统的设置界面，通过与数据库的关系映射实现阈值等信息的修改。

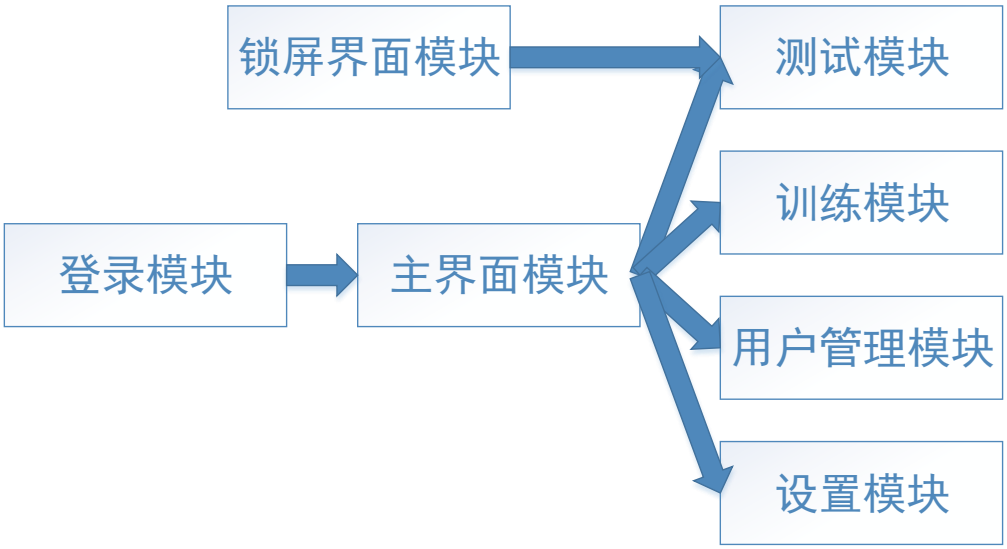

图 3.1 功能模块

3.2 Native 功能模块

Native 功能模块是指由 C 编写但是运行在 Java 虚拟机上的功能模块，拥有较好的性能，并且兼容 C 中众多的类库工具包。在 Android 应用开发中需要先利用 NDK 对 C 代码进行编译构建成 so 动态库，再添加到应用中。经过简单的测试，Native 功能模块包含两个过程，分别是训练过程和测试过程，主要用到三个模块(图 3.2)：

- 特征提取模块
  对 AudioRecord 类得到的实例 WAV 文件进行特征提取，得到 MFCC 特征文件。
- 训练模块
  对若干个 MFCC 特征文件进行建模训练，得到 HMM 模型。
- 测试模块
  依据已有的 HMM 模型群，对某个 MFCC 特征文件进行解码。

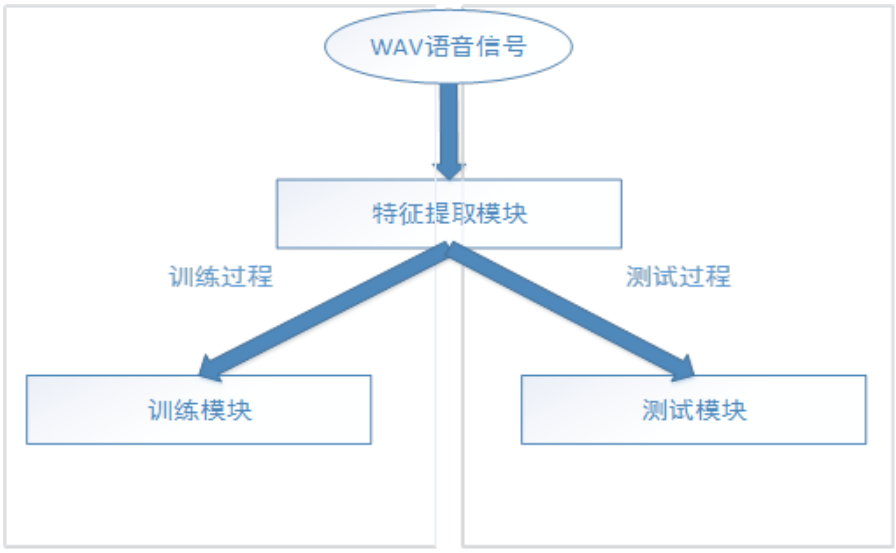

图 3.2Native 功能模块

3.3 MVC 模式

Android 上应用开发普遍遵循 MVC 设计模式，MVC 的全称是 Model 模型，View 视图和 Controller 控制器。在 Android 应用上的体现就是 Entity 实体例如文件、用户，XML 界面布局和 Activity 界面模块。通过 MVC 设计模式，可以将数据，业务逻辑和界面解耦，将这三部分分别都集中部署，在定制数据结构、改进业务逻辑或修改界面时可以不需要重新编写全部代码。使用 MVC 设计模式可以提高代码质量，提升开发效率。

本系统的 MVC 框架设计如图 3.3 所示。

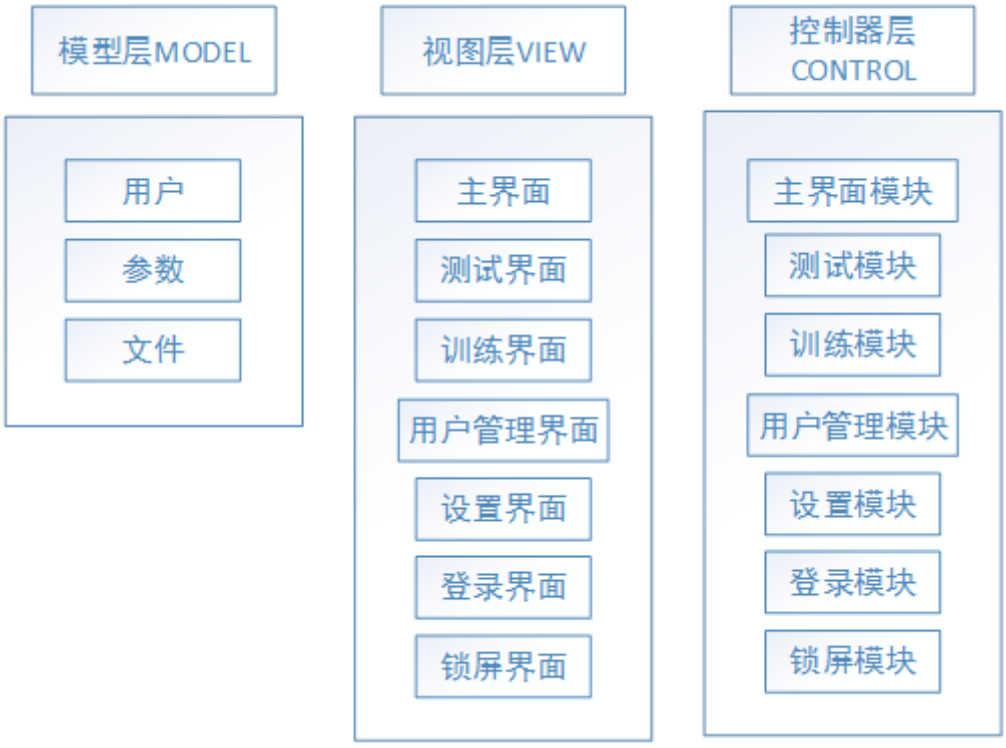

图 3.3 MVC 分层设计

3.4 SQLite 数据库以及 ORM 框架

SQLite 是 Android 应用中最常用的数据库，不同于 MySql、MS SQL Server 等客户端服务器数据库，SQLite 是一个单文件数据库。使用 C 构建，并集成在 Android 系统的第二层——类库层中，每个 Android 应用都可以获得一个专属的 SQLite 数据库。Android 应用在开发时可以使用 SQLiteOpenHelper 类中的方法，将应用的 Context 作为参数传入，得到这个应用的 SQLiteDatabase，之后就可以通过 execSQL 方法将 SQL 语句传入并执行。

然而使用 SQLiteOpenHelper 提供的方法操作数据库并不符合面向对象的软件开发思想。因此本课题在操作数据库的过程中使用了 ORM 对象关系映射(Object Relational Mapping)框架，目前 ORM 框架可谓汗牛充栋，但为了能扎实掌握这一门技术以及更好地适应于本课题的在该方面的需求，在借鉴了一些 ORM 框架后，改编了其中之一 xUtils，使其成为更小更适合本课题的 xutil_orm 框架。通过使用 ORM 框架，不仅可以省去撰写 SQL 语句的困扰，还可以更好地对一些异常和事务进行处理。

本课题中预计主要将用户，文件路径，以及相关时间等数据存放于 SQLite 数据库中。其中用户表与用户实体的映射如图 3.4 所示。

NAME	ISTRAINED	PATH	TRAIN_TIME	TEST_TIME
张三	TRUE	…/USER1/HMM/…	2015/3/7 10:18	2015/3/7 10:18
李四	FALSE	…/USER2/HMM/…	2015/3/8 10:18	2015/3/8 10:18
王五	TRUE	…/USER3/HMM/…	2015/3/9 10:18	2015/3/9 10:18
赵六	FALSE	…/USER4/HMM/…	2015/3/10 10:18	2015/3/10 10:18
陈七	TRUE	…/USER5/HMM/…	2015/3/11 10:18	2015/3/11 10:18

图 3.4 用户数据结构及用户表样例

3.5 Apache-commons-io 包

本课题中涉及到不少文件的操作，例如 WAV 语音文件的新建、复制与删除，MFCC 特征文件的新建、复制与删除，HMM 模型文件的新建、复制与删除，以及文件路径的获取和传递。为了使得本系统有更好的健壮性，以及遵守软件工程中“不重复造轮子”的思想，本系统中使用了 Apache-commons-io 包。通过其中的 FileUtils 类中的不同的方法可以清晰简单地对文件、文件夹进行新建、修改、删除、复制、获取路径等操作。使用 Apache-commons-io 类库可以提升开发效率，并且更好地处理一些文件异常，增强应用的健壮性。

3.6 HTK 工具箱

3.6.1 HTK 工具箱结构

HTK 工具箱是一个用于构建和操作隐马尔科夫模型的便携式工具箱。HTK 主要用于语音识别的研究，虽然已被用于包括语音合成研究字符识别和 DNA 测序的许多其他应用程序。

HTK 工具箱的多数功能已被建立成库模块。这些模块确保了每个工具对外部的接口使用完全相同的方法，并提供了通用功能的中心资源。图 3.5 说明了 HTK 工具箱的架构和输入输出接口。

- HShell 控制用户的输入输出以及与操作系统的交互。
- HMem 控制所有的内存管理。
- HMath 提供数学支持。
- HSigP 完成语音分析所需的信号量处理操作。
- HLabel 提供标签文件的接口。
- HLM 用于语言模型文件。
- HNet 用于网络和网格。
- HDict 用于字典。
- HVQ 用于矢量量化码本。
- HModel 用于隐马尔科夫模型（HMM）的定义。
- HWave 完成所有在波形阶段的语音的输入输出。
- HParm 完成语音的参数化阶段。
- HWave 和 HLabel 支持多文件格式并允许数据从其他系统输入。
- HAudio 支持音频的直接输入。
- HGraf 提供简单的交互式图形。
- HUtil 提供大量用于计算 HMM 的通用程序。
- HTrain 和 HFB 支持多种 HTK 训练工具。
- HAdapt 提供多种适应工具的支持。
- HRec 包含了识别过程中使用的主要函数。

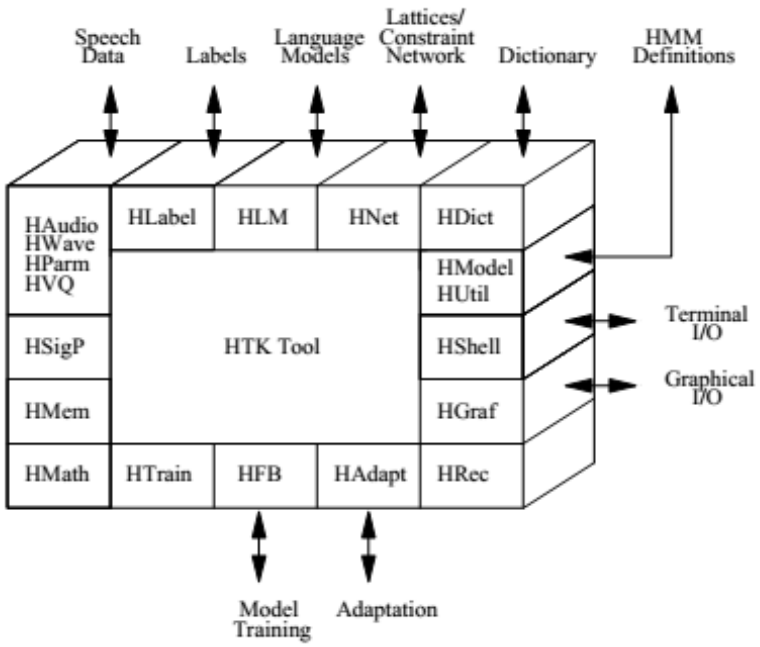

图 3.5 HTK 工具箱的架构和输入输出接口

3.6.2 HTK 工具箱使用过程

HTK 工具箱的使用有四个主要的阶段：数据准备，训练，测试，分析。(图 3.6)

- 数据准备
  对于语音信号，可以使用 HCopy 参数化波形，得到 MFCC 特征文件。
- 训练
  由 HInit 读入所有的引导训练数据，并迭代计算一组初始参数值。通过 HInit 计算的初始参数值由 HRest 进一步重估，直至收敛。
- 测试
  HVite 使用令牌传递算法来执行基于维特比的语音识别。HVite 把输入看作是一个网络，该网络描述了所容许的单词序列，并定义了每个单词如何发声的字典和一组 HMM。它将单词网络转换成音素网络，之后在每个音素实例附上合适的 HMM 定义，以实现运转。接着就可在一系列存储的语音文件上或者直接的音频输入上进行识别。
- 分析
  基于 HMM 的识别系统的性能评估由 HResults 完成。

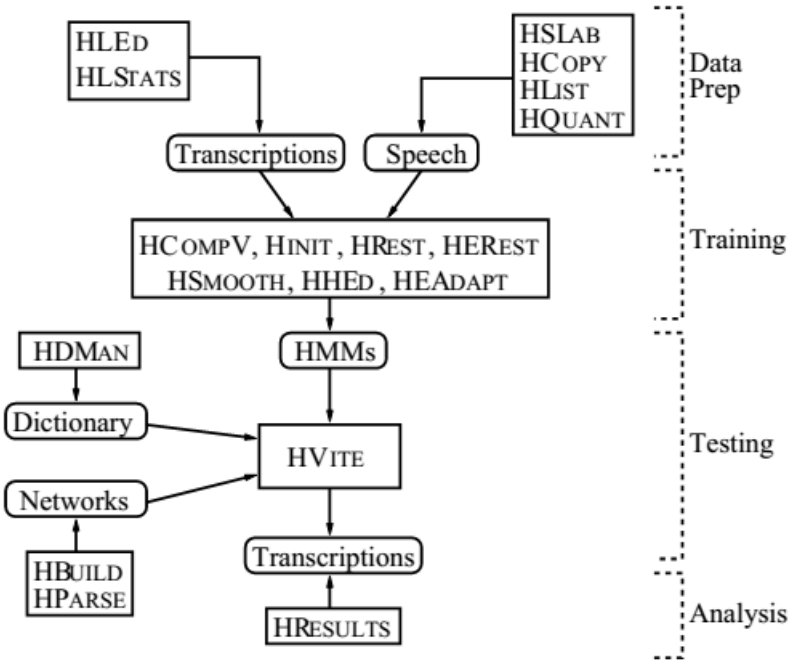

图 3.6 HTK 工具箱使用的四个阶段

经过过需求分析后，本课题中对 HTK 工具箱的使用主要如图 3.7 所示，对一些手机端上不常用的功能进行了裁剪，使得本系统更为小巧稳定。

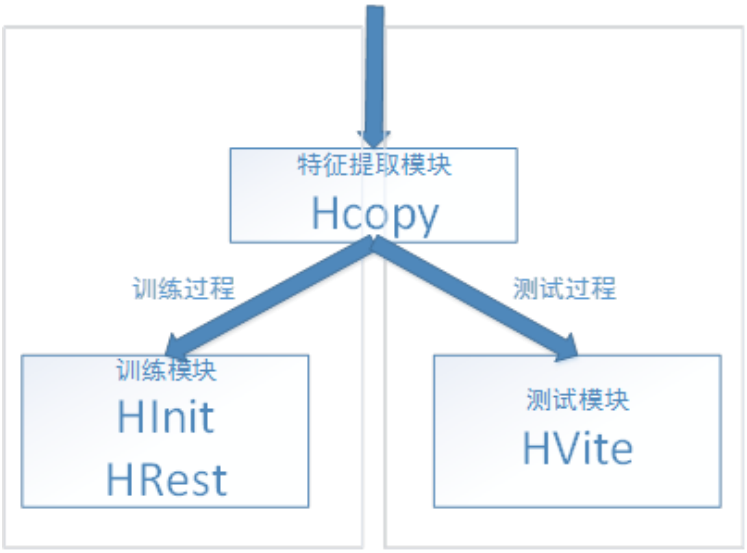

图 3.7 裁剪使用 HTK 工具箱

4 进度安排

2014.11-2015.1 查阅文献，收集资料，熟悉课题，完成文献翻译和开题报告
2015.2-2015.3 系统需求分析，系统概要设计
2015.3-2015.4 系统详细设计
2015.4-2015.5 系统实现和测试
2015.6 撰写论文，参加论文答辩

5 参考文献

[1] Cambridge University Engineering Department (CUED). HTKBook[EB/OL]. http://htk.eng.cam.ac.uk/docs/docs.shtml. 2006

[2] Anthony Larcher, Kong Aik Lee, Bin Ma 等. Text-dependent speaker verification: Classifiers, databases and RSR2015[J]. Speech Communication 2014，60：56–77

[3] Zhizheng Wu,Nicholas Evans, Tomi Kinnunen 等. Spoofing and countermeasures for speaker verification: A survey [J]. Speech Communication 2014，60：130–153

[4] wikipedia.org. Android[EB/OL]. http://en.wikipedia.org/wiki/Android_(operating_system). 2015

[5] apache.org. Commons IO Javadoc[EB/OL]. http://commons.apache.org/proper/commons-io/download_io.cgi. 2014

[6] The SQLite Development Team. Appropriate Uses For SQLite[EB/OL]. http://www.sqlite.org/whentouse.html. 2015

[7] 李刚.疯狂 Android 讲义.第二版[M].北京：电子工业出版社，2013.1-343.

[8] (美)Bruce Eckel 著 陈昊鹏 译.Java 编程思想[M].北京：机械工业出版社，2007.1-649

[9] 李建平 林劼.生物特征的安全计算理论与技术[M].成都：电子科技大学出版社，2011，1-270

[10] android.com. Android NDK [EB/OL]. http://developer.android.com/tools/sdk/ndk/index.html. 2015

[11] android.com. Android Developer Tools[EB/OL]. http://developer.android.com/tools/help/adt.html. 2015

[12] 陈泉金.基于 HTK 的连续语音识别技术研究[D].南京：南京邮电大学，2010

[13] 安徽大学计算机智能与信号处理教育部重点实验室.基于 HTK 的汉语语音售票系统的设计与实现[D].合肥：安徽大学，2010

[14] M.E. Forsyth, A.M. Sutherland, J.A. Elliott, M.A. Jack. HMM speaker verification with sparse training data on telephone quality speech[J]. Speech Communication 1993，13：411－416

[15] Sunita Chauhan , Ping Wang . A computer-aided MFCC-based HMM system for automatic auscultation[J]. Expert Systems with Applications. 39, 2, 1 2012, 2157–2165

[16] 赵力.HMM 在说话人识别中的应用[J].电路与系统学报，2001-9，6（3）：51-57

[17] jamesju. 基于 HTK 语音工具包进行孤立词识别的使用教程[EB/OL]. http://my.oschina.net/jamesju/blog/116151. 2013