# 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

说话人识别最初是应用在司法领域，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数字密码验证，避免了数字密码遭冒充者窃取的风险。

除此之外还使得生物身份识别更简单易用，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，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语言实现，主要负责管理Android硬件的驱动。

第二层类库层开发依然由C语言，主要负责Android上一些常用组件的开发，例如SQLite数据库，WebKit浏览器内核，libc常用的C语言函数库，Dalvik虚拟机。

第三层是Android系统应用框架由Java语言实现，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，做到内核无关的特性，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。

第四层是Android应用程序的开发由Java语言实现，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。

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，Sharedpreference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。在第四层的Android应用程序调用第二层中自行编写的语音类库。

### 2.2 隐马尔科夫模型(HMM)

隐马尔可夫模型（Hidden Markov Model，HMM）是一个统计模型。它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。

在正常的马尔可夫模型中，状态参数对于观察者来说是完全直接可见的。而在隐马尔可夫模型中，状态参数并不是直接可见的，但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可夫模型。

在语音识别领域，最常用到的是隐马尔可夫模型的解码问题。

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)。

### 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，又能对相同用户同一。

人们能对不同人的语音做出区别是人们的耳蜗类似于一个滤波器组，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)。

## 3 技术路线

### 3.1 Java应用模块



### 3.2 Native应用模块



### 3.3 MVC模式

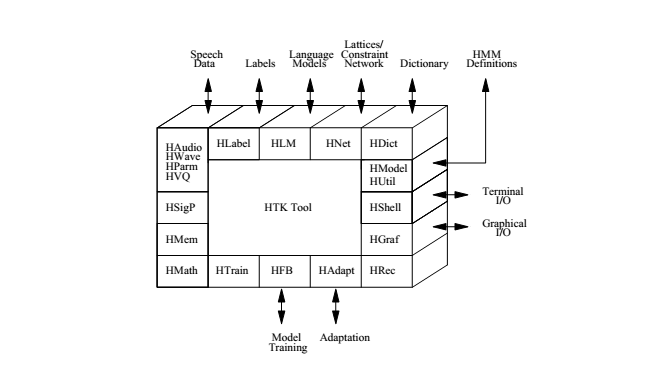


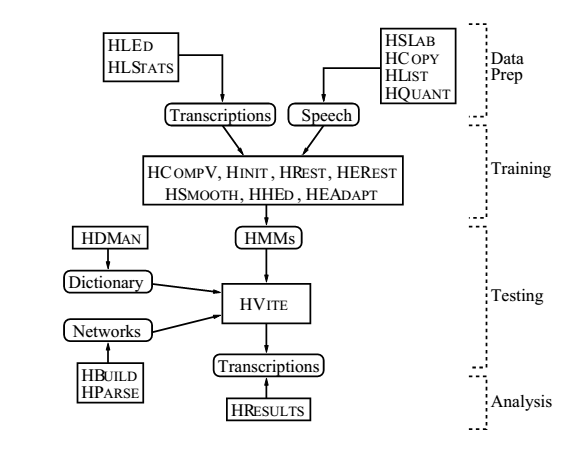
### 3.4 SQLite数据库以及ORM框架



### 3.5 Apache-commons-io类库

### 3.6 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] 李兴华，王月清．Java Web开发实战经典基础篇[M]．北京：清华大学出版社，2010．90~324．

[2] 赵俊峰，姜宁，焦学理，