面向 5G 智能驾驶无人车语音交互系统的设计与实现

梅正阳　李　敏　王静雯　许崇彩
（宿迁学院，江苏 宿迁 223800）

摘要：本文选取科大讯飞的语音技术作为研究样板，采用软件和硬件结合的方式对语音交互系统进行开发和研究，利用树莓派开发板作为硬件支撑，使用 Python 和 MySQL 作为软件构造，设计基于隐马尔可夫模型的语音识别模块，基于深度学习的语音合成模块和基于自然语言处理的语音互动模块。完成面向 5G 智能驾驶无人车语音交互系统的设计与实现。

关键词：树莓派；Linux；语音交互；智能驾驶

现在用户对于智能汽车的需求在逐步增大，同时对智能汽车的要求也越来越高，越来越严格，但是现在在语音交互系统中语音识别出错率的研究与提升还非常的稀少，基于用户需求与市场现状，对语音交互系统的研究和思考具有非常重要的意义与前景。

在本项目中，智能语音交互系统包含如下模块的设计与实现：语音合成模块，即在事先制定好的数据库中搜索相对应的文本信息，通过运行代码，调用科大讯飞的 SDK 包，在树莓派中将文本文字转换成语音输出；语音识别模块，即将树莓派连接麦克风，通过 Python 代码调用科大讯飞的 SDK 包，将从麦克风输入的音频信息转换成文字输出在显示屏上；语音互动模块，即通过 Python 代码调用科大讯飞的 SDK 包，完成语音识别与语音合成功能，并在用户输入语音之后，将用户输入的语音信息与事先设定好的数据库中的文本信息进行匹配，获取响应的回答并合成为语音输出。

一、总体设计

本系统是基于树莓派平台建立的，调用了科大讯飞的语音合成，语音识别和离线命令词识别，并配合 Python 语言，通过 SSH 将树莓派与计算机连接，实现了语音合成模块，语音识别模块，语音交互模块以及远程喊话模块，系统整体设计框架图如图 1.1 所示。

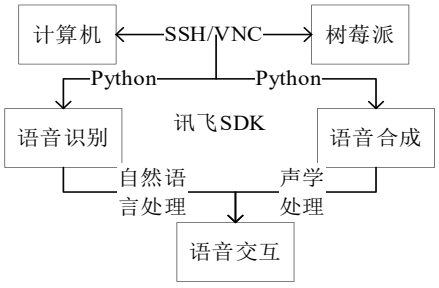


图 1.1 系统设计框架

二、硬件介绍及设计

（一）树莓派简介

树莓派 4B 主板只有名片大小，但功能却十分强劲。主板采用的是 ARM Cortex-A72 1.5GHz（四核）处理器，板载蓝牙、WIFI、USB3.0 两个、USB2.0 两个、千兆以太网接口、3.5MM 音频接口、支持双屏 HDMI 显示。树莓派是基于 Linux 系统的微型计算机，英文名写为 Raspberry Pi，相比于日常使用的笔记本或者台式电脑，拥有小巧，便宜，功耗低等优点，但同时具有速度慢，性能低等缺点。因为树莓派可以控制各种传感器，电动机等，因此树莓派的用途十分的广泛，例如利用树莓派作为控制中心搭建智能车，将树莓派连接硬件用来做数据采集等。

（二）系统硬件平台设计

1. 安装树莓派系统

烧录系统需要通过树莓派官方提供的树莓派镜像烧录器 Raspberry Pi Imager 烧录，为了匹配科大讯飞的 SDK 包中提供的文件，需要烧录 Raspberry Pi OS（32 bit）系统，在烧录完成系统之后，将 TF 插入树莓派开发板的 TF 卡槽中。可以通过 finalshell 与树莓派进行远程连接，接成功之后可以在 finalshell 界面中看到连接成功的时间，计算机的 ip 地址等信息，可以在 finalshell 中直接输入 Linux 系统的指令操控树莓派终端。

2. 连接外设

树莓派需要连接外部设备进行录音，这里使用的是 USB 接口的麦克风进行连接，树莓派操作系统会在插入麦克风时自动检测，在完成麦克风连接之后可在终端输入 lsusb 查看麦克风是否接入成功，如果成功则会显示麦克风与接口。在之后输入录音指令的时候，需要选择对应的外设端口，否则录音失败。

三、系统软件设计

（一）科大讯飞 SDK 包介绍

SDK（软件开发工具包）是为创建应用软件而集成的软件工具包，一般包含有专属的软件包、框架、操作系统以及硬件平台等。科大讯飞 SDK 是科大讯飞基于多种功能集合而来的工具包，当开发人员要实现相应语音识别的功能时，就需要调用相应的 SDK。科大讯飞 SDK 包包含了 API 文档，运行程序生成的可执行文件，音频文件，msc 调试生成的日志，语法样例，调用 SDK 所需头文件，Windows64 位动态库，Windows32 位动态度，Raspberry32 位动态库以及语音识别示例，语音听写示例与语音合成示例，发音人 jet 文件。因为科大讯飞 SDK 包是复制到树莓派系统目录下的，调用所需要的 so 库和 h 头文件并不在树莓派系统默认搜索库中，因此需要执行指令 sudo cp/home/pi/RaspberryPiSDK/Linux_voice_1.109/libs/RaspberryPi/libmsc.so/usr/local/lib 和 cp/home/pi/RaspberryPiSDK/Linux_voice_1.109/include/*/usr/local/include 将文件复制到默认搜索文件夹下。在此之后，需要建立新的文件夹用于存放与编写代码，完成单轮语音识别功能与单次语音合成功能。

（二）原理介绍

1. 语音识别原理

语音识别是指机器将用户输入的语音或者音频转化成文字的技术，这项技术需要机器通过不断的学习，将获取到的语音按照一定的方式进行分类，以此找到对应语音最佳的结果，大致分为以下步骤，首先需要对输入的语言进行预处理，其次提取特征，接着根据已有的声学模型和语音模型进行语音编码，最后进行语音的最后处理输出。现阶段的语音识别技术已经趋向成熟，利用

企业提供的开源 SDK 包即可获取目前系统学习到的过程，不再需要用户对机器进行训练。常见的语音识别方法有三种，一是动态时间规整法，二是隐马尔可夫模型，三是人工神经网络模型。

2. 语音合成原理

语音有三大关键成分：信息，音色和韵律。信息是指用户输入的语音说了什么内容，语音是信息的载体；音色是指用户的身份，不同的人有不同的音色；韵律是指用户的说话方式，包含声音的高低，语速的快慢等。语音合成的本质是将系统中的文本文字转换成语音输出出来，首先需要对待合成的文本进行韵律处理，其次采用拼接法将预先录制的语音库进行对比拼接，以合成自然、流畅的语音，最后进行语音输出。

3. 语音交互原理

智能语音交互式基于语音识别，语音合成，自然语言处理等技术的人机交互体验，适用于多种场景，在智能车的应用中一般包含导航，音频，通讯，车辆基本控制，车辆信息查询，生活信息查询等功能。智能语音交互采用的主要技术是自然语言处理技术。自然语言处理技术主要包含四个部分的处理，分别是语法分析，词法分析，语义分析和语境分析。其中最主要的就是语法分析，因为语法分析可以将输入的语音信息划分为单元进行分析，通过分析单元之间的依存关系获得其语法结构，例如句子的主谓宾，句子之间的并列，从属关系等，将识别到的语句与语音库中进行一些特征词的对比，然后进行语音合成，以达到语音交互的目的。

（三）软件程序设计

1. 语音识别软件设计

在语音识别的代码中，含有一个主函数和两个子函数，首先编写主函数登录讯飞服务器，并对请求的业务类型，领域，语言，方言，音频采样率，识别结果格式和结果编码格式进行固定值指定。

随后编写主函数中需要用到的子函数，第一个子函数的功能是上传用户词表，即利用 fopen（）函数用只读模式打开事先写好的用户词表文件，并调用 fseek（）函数对流上的文件指针进行重定位，在获取完音频文件大小之后再次进行重定位，随后利用 malloc（）函数动态分配内存一块大小为音频文件大小的连续空间，再利用 fread（）函数读取用户词表的内容，读取结束调用 MSPUploadData（）函数上传用户词表。最后编写获取文本结果的子函数，在子函数中需要定义音频状态，端点检测和识别状态，利用 ftll（）函数获取音频文件大小，利用 fread（）函数读取音频文件，调用 QISRSessionBegin（）创建语音识别会话，指定是否需要语法，利用 while（）函数按照 200ms 每次的频率获取音频长度，在 while（）函数中调用函数 QISRAudioWrite（）上传语音数据，调用函数 QISRGetResult（）获取识别的文本结果，调用 usleep（）函数进行代码延迟，用于模拟人说话的时间间隙。

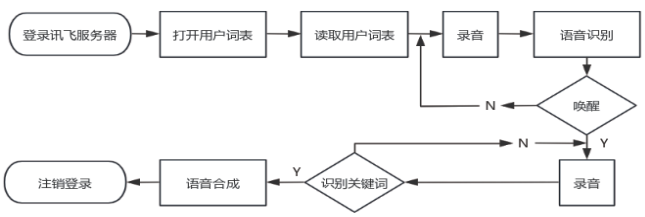
在编写完代码之后运行相应的 sh 文件，并在指定路径录入音频文件，并执行生成的二进制文件，按照提示选择是否需要上传用户词表，完成单词语音识别，在屏幕上看见语音识别结果。

2. 语音合成软件设计

首先编写结构体用于存放音频的基本信息，其次编写主函数，在主函数中建立变量存放登录参数，合成音频数字发音方式，音量，音调，语速，发音人，采样率和文本格式编码，合成的语音文件名称与需要合成的文本。因为在语音合成时，可能会因为音频没有合成结束而返回再次获取服务器合成的语音音频，所以在给这些变量赋值的时候需要使用 const（）函数，使得这些变量不会因为一些外部原因改变，可以提高程序的安全性和可靠性。随后编写文本合成的子函数，在子函数中调用函数 QTTSSessionBegin（）和函数 QTTSTextPut（）创建语音合成会话并上传需要合成的语音文本。调用函数 QTTSAudioGet（）获取服务器合成的语音数据，之后修改音频文件头数据的大小并将修正过的数据写回文件头部，最后调用函数 QTTSSessionEnd（）结束语音合成会话。在代码编写结束之后就可以执行 sh 文件，使其在相应文件夹下合成二进制文件，进入该文件所在目录并运行，获得通过调用 API 生成的语音文件，可利用 aplay 指令播放音频。

3. 语音交互软件设计

语音互动主要流程如图 3.1 所示。语音交互主要是将语音识别和语音合成组合起来，因此需要编写语音识别和语音合成的代码，然后再编写主函数，通过在主函数中调用语音识别和语音合成的代码，读取用户词表等方式完成语音互动的功能。首先按照前面写的语音识别流程与语音合成流程编写子代码，在子代码中，进行一些子函数的编写，首先获取 so 镜像文件所在位置，然后登录科大讯飞服务器，最后编写子函数提供主函数调用。在语音合成子代码中，子函数 login（）用于登录科大讯飞服务器，子函数 QTTS_Session_Begin（）用于分配语音合成资源，子函数 QTTS_Text_PUT（）用于写入要合成的文本，子函数 QTTS_Audio_Get（）用于获取合成的音频文件，子函数 QTTS_Session_End（）用于结束本次语音合成，子函数 XF_voice（）用于填写参数，打开音频文件并配置声道数，量化位数和取样频率。在语音识别子代码中，在类 Msp 包含用于登录，设置频率，设置语音间隙，获取音频文件并获取识别结果等子函数，在类 Msp 之后的子函数 XF_text（）用于调用类 Msp，主要功能是登录科大讯飞服务器，编写参数，获取文本并注销登录。在主函数中，首先通过 open（）函数用只读模式打开用户词表，利用 os.system（）让树莓派在终端执行录音的指令，然后调用语音识别子代码中的 XF_text（）函数识别录音内容，并判断录音内容是否为唤醒词，如果是，则回复“我在”，并继续下面的代码，如果不是，则返回重新录音。在系统回复“我在”之后，判断录音内容是否有用户词表中的关键词，如果检测到关键词，则根据关键词进行回复，例如检测到关键词“校”，则回复“我是宿迁学院的”，如果系统识别到语音为“提问结束”，则退出系统，不再进行互动。



（四）开机自启动设计

使用 Python 调用 Linux 系统指令，编写 Python 代码调用科大讯飞 SDK 包中语音合成部分的文件，将文件命名为 start.py 并放入 Linux 系统的程序文件 bashrc 中，这样就可以设置树莓派开机启动，在树莓派开机时进行语音播放。

参考文献：

[1] 刘正晨 . 结合发音特征与深度学习的语音生成方法研究 [D]. 中国科学技术大学，2018.

项目基金：江苏省高等教育基础科学研究项目 20KJB510042