



人工智能语音实训平台指导书

www.whwzzc.com

武汉唯众智创科技有限公司

二〇二〇年

目录

(一) 语音采集类.....	2
1.1 语音采集实验.....	2
1.2 语音波形显示实验.....	7
1.3 语音编码实验.....	13
1.4 语音采样频率转换实验.....	19
(二) 语音信号类.....	24
2.1 语音信号强度实验.....	24
2.2 白噪音信号实验.....	28
2.3 语音短时傅里叶变换实验.....	32
2.4 音频自动增益频控制实验.....	37
(三) 语音检测类.....	42
3.1 语音端点检测实验.....	42
(四) 语音噪音类.....	50
4.1 语音增强实验.....	50
4.2 语音添加噪音实验.....	55
(五) 语音模型类.....	60
5.1 LSTM 声学模型训练实验.....	60
5.2 情感分析实验.....	66
5.3 知识图谱关系抽取实验.....	70
(六) 声源定位类.....	76
6.1 实时声源定位实验.....	76
(七) 语音识别类.....	82
7.1 语音识别实验.....	82
7.2 分词识别实验.....	89
7.3 词性标注实验.....	94
7.4 命名识别实验.....	98
(八) 语音合成类.....	102
8.1 语音合成实验.....	102

（一）语音采集类

1.1 语音采集实验

一、问题

在人工智能语音识别领域经常会遇到需要处理语音信号的场景，那么这些语音信号从哪里来呢？首先我们肯定需要通过麦克风、拾音器、语音阵列等外设获取语音信号，在 Linux 系统中如何控制外设采集语音信号，并且将采集到的信号进行保存呢？

二、方案

首先我们需要准备能够采集到语音信号的外设，其中包括麦克风、语音阵列、拾音器均可对外界的语音信号进行采集。在 Linux 操作系统中可以使用 arecord 命令进行语音采集，在采集的过程中我们要根据不同的外设设置不同的参数，以确保语音信号采集准确无误。同时我们需要通过 Python 编程实现检测“开始采集语音信号”的命令以及“退出采集语音信号”的命令。

三、环境

硬件环境：有稳定网络的机房、唯众人工智能语音识别实训平台硬件、语音识别套件包。

软件环境：唯众人工智能语音识别实训平台软件环境。

四、实验步骤

1、连接电源, 打开唯众人工智能语音识别实训平台, 打开方式为按下主板电源键, 默认开机自动登录(若需要开机密码则选择用户 AI-Platform, 密码默认为 123456) , 开机界面如下图 4-1。



图 4-1 唯众人工智能 (AI) 语音识别实训平台开机界面

LX 终端

唯众人工智能语音识别实训平台默认系统为 Linux 系统 (Fedora) , 为了满足在人工智能语音识别开发过程中的各种需求, 本镜像系统中已经预装了 Python3、OpenCV3、Numpy、TensorFlow、Keras 等多个依赖, 并且已配置好相关环境, 学生和老师无需重复去进行开发环境搭建。

2、创建自己的项目文件夹

首先点击界面左下角的“微型黑屏标签”, 进入到“LX 终端”。本实验为语音采集实验, 根据要求, 在 AI-Platform 的文件夹下创建一个文件

夹“test”进入到“test”文件夹创建一个“test1-1”文件夹,在“test1-1”文件夹中创建一个“yuyincaiji.py”的文件,在该文件中进行代码编程。

3、按照方案描述编写相关代码,使用python3运行编写的代码,并且进行debug,排查错误,得到预期效果。

知识点:

创建文件夹的方法为: `mkdir xxx`; 创建一个“test”文件夹的方法为: `mkdir test`。

进入到某个目录的方法为: `cd xxx`; 如进入到test文件夹的方法为: `cd test`。

创建.py文件可以使用vi编辑器,方法为: `vi xxx.py`,编写完毕后按“ESC”键退出编辑,按“Shift+:”输入wq,保存修改的文件。

通过命令 `pwd` 可以查看当前的路径,通过命令 `ls` 可以查看当前路径下的目录。

`arecord` 方法: 采集音频信号。

`sys.stdin.read` 方法: 读取数据,不需要回车。

五、实验代码

```
import warnings
import subprocess
import sys
import tty
import termios

def readchar():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch

def readkey(getchar_fn=None):
    getchar = getchar_fn or readchar
    c1 = getchar()
    if ord(c1) != 0x1b:
        return c1
    c2 = getchar()
    if ord(c2) != 0x5b:
        return c1
    c3 = getchar()
    return chr(0x10 + ord(c3) - 65)

while True:

    print("按's'开始录音, 按'q'退出录音")

    key = readkey()
    print(key)
    if key == 'q':
        break
    elif key == 's':

        print("==== 录音开始 ... ====")

        # On-board microphone
        # subprocess.call(["arecord", "-D", "MainMicCapture", "-r",
```