**Instruction**

**Team 8 Sparks**

Download the whole file from GitHub and extract it.

Make sure your computer has python version 3 (we use python 3.5 in our pc).

Install PyQt5, PyAudio, PyGame, espeak, python\_speech\_features, speech\_recognition, google\_api\_client, numpy, webbrowser and all the APIs that you don’t have.

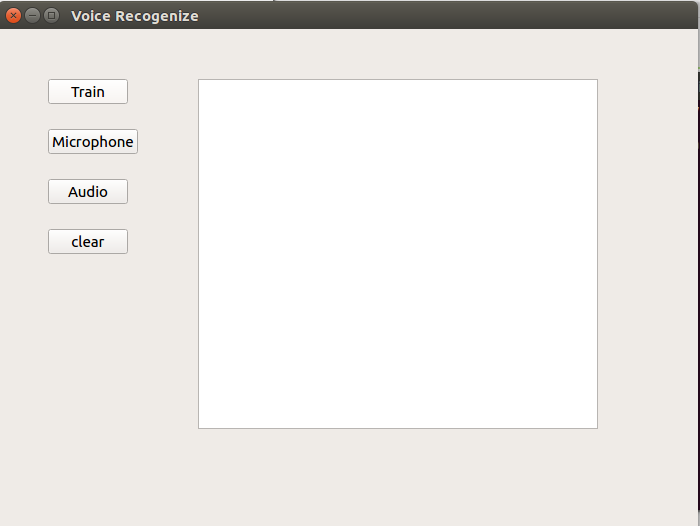
Put your training person’s wav files in the person folder and let each person’s file in the folder with their name. Those wav files are better in

**Usage:**

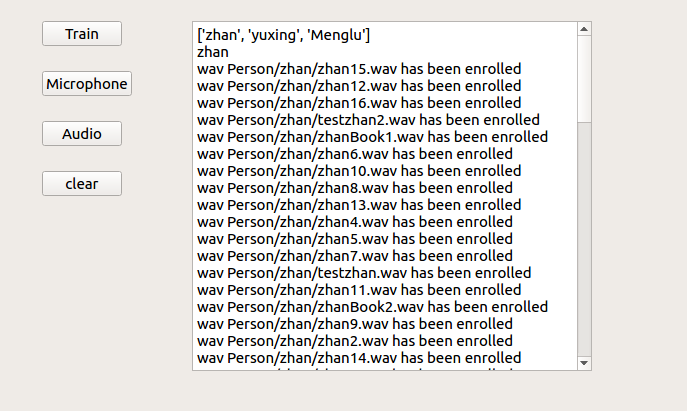
**Train model:**

Use command:

*python3 TestInterface\_2.py* in the command line to run the gui. The window below would appear.



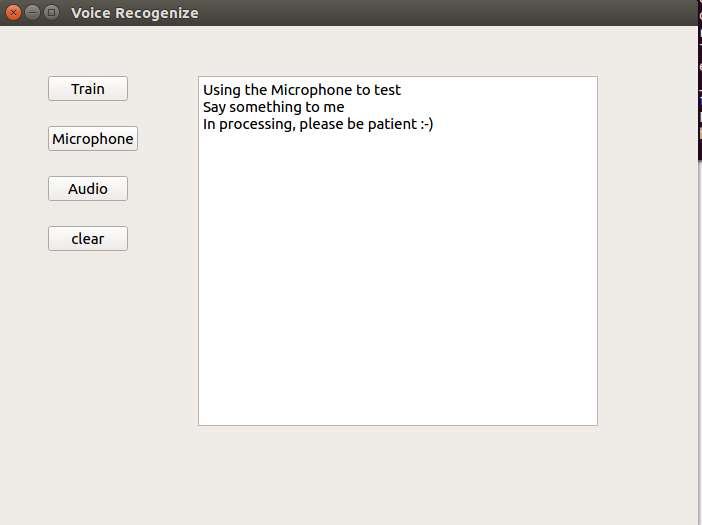
Click the ‘train’ button if you haven’t trained your voice before. It may take a long time according to your voice file’s size. The training process would show like the picture below. When the train process is completed, it will show ‘finish’ in the below.

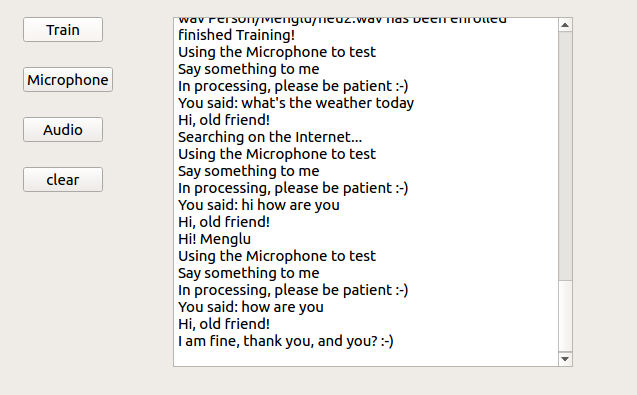


**Starting Test:**

**Microphone Test:**

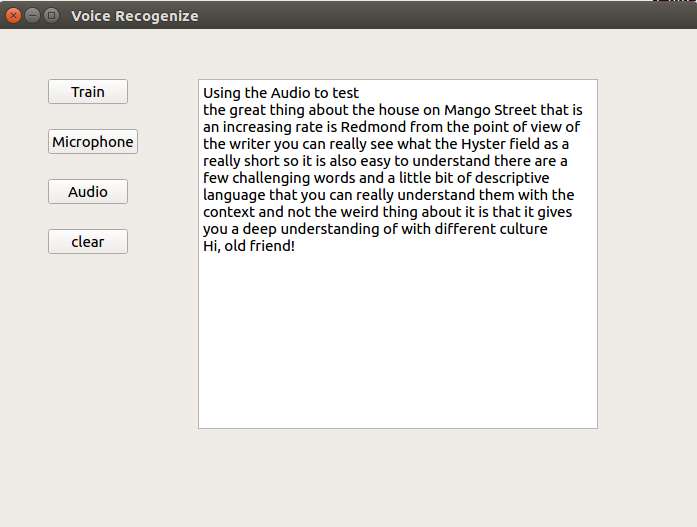
Click ‘microphone’ button to test your voice. It will display ‘Say something to me’ and it records your voice for about 5 seconds. When it finishes recording, a ‘In processing’ tip will display. Then your voice will be recognized and printed in the widget with answer from our system.





**Audio Test:**

Click ‘audio’ button to test the local audio file in the ‘audio’ folder. It will transcribe the whole text and display in the widget. If you want to change the audio file to test, please change it in the audio\_final.py to put your new audio file path.



And click ‘clear’ button will reset the text area.

What’s more, if you don’t want to use gui to run the system, you can use command line to do the same process.

Train:

*python3 train-class.py or python3 train-speaker-model.py*

Test:

*python3 microphone\_recognition.py*

*python3 audio\_transcribe.py*

**Others:**

Temp.txt stores the former speaker’s name. tempfile.txt stores the words you say when the system couldn’t understand you.