Speaking Tuner

date:2021-2022

name: Speak_Tuner

technologies used: speech recognition (SR), Text-to-Speech (TTS), Signal processing(Numpy),

startup folder(shell:startup)

In the beginning of 2021 I met a visually impaired person , after some discussion he tells how hard it is for him to tune his instrument since every tuner in the is visual signaling based .

So i got the idea why not to try to make one of my own which uses audio to communicate with the user ?

Here my third project began .

the first was to enable him to interact with it. Having an alexa at home , i began to search for Speech recognition library in python and began to create a script allowing him to communicate with the app .

Once done i got to find a way for the app to tells back the output and i found out about pyttsx3 a text-to-speech library allowing me to transform text into audio .

the next step was to find a way to record an manipulate audio , at first i thought about recording .wav file and the analyze it but it was possible to do it live , so I implemented it live .

Then i transform my sine into frequency and amplitude arrays using what's called Fast Fourier Transform (FFT) from the Numpy library , but with come also its own problem (like i have to half of frequencies etc.).

I now needed to get the fundamental frequency But before i got to pass all the a the frequencies to a low pass filter to remove ambiance noises based on their amplitude . After extracting the fundamental freq i got to transform it to the closest note and evaluate it correctness (too high / low).

Now that i have the note and its offset i just had to TTS to communicate it to the user.

I got three version of this app :

-the basic python script

-the compiled exe app : since the user cant navigate easily through his computer i had to setup a startup shortcut script in the shell:startup windows folder which allow the app to start at computer startup

-the Raspberry Pi version: a more optimized version which doesn't used speech recognition but a external button connected to the RPI's GPIO

This project reflect my ability to think about solutions to specific problem , to use technologies never encountered before ,to understand the science behind and so one.