import pyaudio

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks

import wave

import time

import speech\_recognition as sr

# Parameters

FORMAT = pyaudio.paInt16

CHANNELS = 1

RATE = 44100

CHUNK = 1024

THRESHOLD\_ENERGY = 80000 # Adjust this value according to your environment

MIN\_VOICE\_DURATION = 0.5 # Minimum duration of voice segment in seconds

MAX\_VOICE\_DURATION = 5.0 # Maximum duration of voice segment in seconds

SAVE\_PATH = "detected\_voice.wav"

def plot\_spectrum(fft\_magnitude):

plt.clf()

plt.plot(np.arange(len(fft\_magnitude)), fft\_magnitude)

plt.xlabel('Frequency (Hz)')

plt.ylabel('Magnitude')

plt.title('FFT Spectrum')

plt.pause(0.01)

def detect\_voice(fft\_magnitude):

peaks, \_ = find\_peaks(fft\_magnitude, height=THRESHOLD\_ENERGY, distance=100)

if len(peaks) > 0:

print("Voice detected!")

return True

else:

return False

def record\_voice(stream):

frames = []

start\_time = None

while True:

data = stream.read(CHUNK)

np\_data = np.frombuffer(data, dtype=np.int16)

fft\_data = np.fft.fft(np\_data)

fft\_magnitude = np.abs(fft\_data)

plot\_spectrum(fft\_magnitude)

if detect\_voice(fft\_magnitude):

frames.append(data)

if start\_time is None:

start\_time = time.time()

elif start\_time is not None:

duration = time.time() - start\_time

if duration >= MIN\_VOICE\_DURATION:

break

else:

frames = []

start\_time = None

return frames

def save\_voice(frames):

# Write the recorded voice segment to a WAV file

p = pyaudio.PyAudio()

wf = wave.open(SAVE\_PATH, 'wb')

wf.setnchannels(CHANNELS)

wf.setsampwidth(p.get\_sample\_size(FORMAT))

wf.setframerate(RATE)

wf.writeframes(b''.join(frames))

wf.close()

# Use SpeechRecognition to transcribe the recorded voice segment

recognizer = sr.Recognizer()

with sr.AudioFile(SAVE\_PATH) as source:

audio\_data = recognizer.record(source)

try:

text = recognizer.recognize\_google(audio\_data)

print("Transcribed text:", text)

except sr.UnknownValueError:

print("Google Speech Recognition could not understand the audio")

except sr.RequestError as e:

print("Could not request results from Google Speech Recognition service; {0}".format(e))

def main():

p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,

channels=CHANNELS,

rate=RATE,

input=True,

frames\_per\_buffer=CHUNK)

print("\* Recording")

try:

while True:

frames = record\_voice(stream)

if frames:

save\_voice(frames)

print("Voice segment saved to:", SAVE\_PATH)

except KeyboardInterrupt:

print("\* Stopped")

stream.stop\_stream()

stream.close()

p.terminate()

if \_\_name\_\_ == "\_\_main\_\_":

main()