

Thư viện xử lý âm thanh

Trung-Trực Trần

1. **Giới thiệu:** Trong Python, có một số thư viện phổ biến được sử dụng để xử lý âm thanh, các thư viện này giúp người dùng dễ sử dụng để thao tác với file âm thanh chẳng hạn như đọc, ghi, cắt, nối và xử lý các file, chúng ta cùng tìm hiểu một số thư viện phổ biến hiện nay của Python trong dữ liệu này:



Librosa: Thư viện mạnh mẽ cho phân tích và xử lý âm thanh và âm nhạc.

- **librosa.load:** Hàm để đọc file âm thanh.
- **librosa.example:** Hàm để tải một ví dụ âm thanh từ bộ dữ liệu mẫu của Librosa.

Scipy: là một thư viện Python mạnh mẽ cho tính toán khoa học và kỹ thuật.

- **scipy.fft:** Hàm để thực hiện biến đổi Fourier.
- **scipy.signal.spectrogram:** Hàm để tạo biểu đồ phổ của tín hiệu.

SpeechRecognition: Thư viện để trích xuất thông tin từ dữ liệu âm thanh.

- **recognizer.record:** ghi âm từ nguồn âm thanh đã chọn và trả về dữ liệu âm thanh dưới dạng một đối tượng.
- **recognizer.recognize_google:** Phương thức này được sử dụng để nhận dạng văn bản từ dữ liệu âm thanh bằng cách sử dụng dịch vụ nhận dạng giọng nói của Google (Cần kết nối internet).

gtts: là một thư viện Python cho phép bạn tạo và phát lại giọng nói từ văn bản sử dụng dịch vụ Text-to-Speech của Google.

Cần các thuộc tính sau để sử dụng thư viện này:

- `lang = 'vi'` (Ngôn ngữ trả về).
- `output_filename = 'record.mp3'` (Loại file âm thanh đầu ra).
- `content = "xin chào mọi người"` (đoạn văn bản muốn chuyển thành voice).

2. Ví dụ cách sử dụng thư viện xử lý âm thanh.

Librosa

-Cài đặt thư viện:

```
1 !pip install librosa
2
```

-Sử dụng thư viện librosa để visualize âm thanh:

```
1 import librosa
2
3 filename = librosa.example('nutcracker')
4 # Load the audio as a waveform 'y'
5 # Store the sampling rate as 'sr'
6 audio, sr = librosa.load(filename)
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10 time = np.linspace(0, audio.shape[0] / sr, num=audio.shape[0])
11 plt.plot(time, audio, color="blue")
12 plt.xlabel("Time (s)")
13 plt.ylabel("Amplitude (quantized)")
14 plt.title("Wav file visualization")
15 plt.axhline(y=0, color='r', linestyle='--')
16 plt.show()
17
```

Output: Biểu đồ cho thấy biên độ của sóng âm thanh theo thời gian.

speech recognition

-Cài đặt thư viện:

```
1 !pip install SpeechRecognition
2
```

-Sử dụng thư viện SpeechRecognition để chuyển âm thanh thành văn bản:

```
1 import speech_recognition as sr
2
3 r = sr.Recognizer()
4 audio_filename = 'data.wav'
5
6 my_audio = sr.AudioFile(audio_filename)
7 with my_audio as source:
8     audio = r.record(source)
9
10 print(type(audio))
11 your_speech = r.recognize_google(audio, language="vi-VN")
12 print("Audio transcription: ", your_speech)
13
```

Output: Hiện ra tận phía xa.

gtts

-Cài đặt thư viện:

```
1 !pip install gtts
2
```

-Sử dụng thư viện SpeechRecognition để chuyển âm thanh thành văn bản:

```
1 from gtts import gTTS
2 import librosa
3 import numpy as np
4
5 lang='vi'
6 output_filename = 'record.mp3'
7 content = "xin chào mọi người"
8 output = gTTS(content, lang=lang, slow=False) # text to speech
9
10 output.save(output_filename) # save google audio to a file
11 data, sr = librosa.load(output_filename) # load google audio
12 using librosa library
13
14 import IPython
15 IPython.display.display(IPython.display.Audio(np.transpose(data),
16 rate=sr)) # display audio
```

Output: Một đoạn âm thanh mp3 với nội dung là "Xin chào mọi người".

3. Bài tập:

- Hãy thực hiện các thư viện trên với các input âm thanh và văn bản khác nhau để có thể hiểu rõ hơn về cách sử dụng các thư viện này.

```
1 test_case_1 = Một file âm thanh tự do.
2 test_case_2 = "Tôi yêu AIO"
3 # Your code here
4
```

Output: Các đồ thị và đoạn văn bản tương ứng với file truyền vào.

Output: Một đoạn âm thanh mp3 với nội dung là "Tôi yêu AIO"