Truy vấn âm thanh

Thành viên:
- Nguyễn Anh Khoa 18520923
- Trần Thị Phương Thảo 18521422
- Võ Quốc An 18520440

Nội dung

1. Giới thiệu
2. Tổng quan về dữ liệu âm thanh
3. Hướng tiếp cận
4. Mô hình truy vấn
5. Demo

Giới thiệu

Nội dung đề tài: **tìm kiếm bài hát** dựa vào **âm thanh** truyền vào

- Input: Tệp âm thanh định dạng mp3 or wav

- Output: Tên bài hát của tệp âm thanh

Dataset

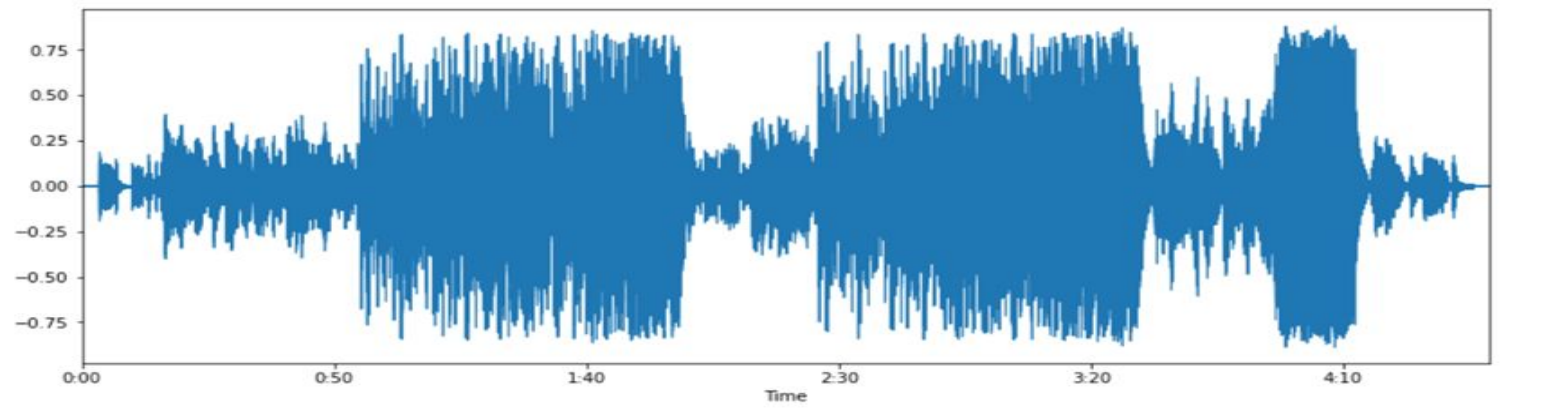
- Crawl bài hát theo các thể loại từ zingmp3

- Bộ dataset có 1070 bài hát.

nguồn: https://github.com/hatienl0i261299/Zingmp3

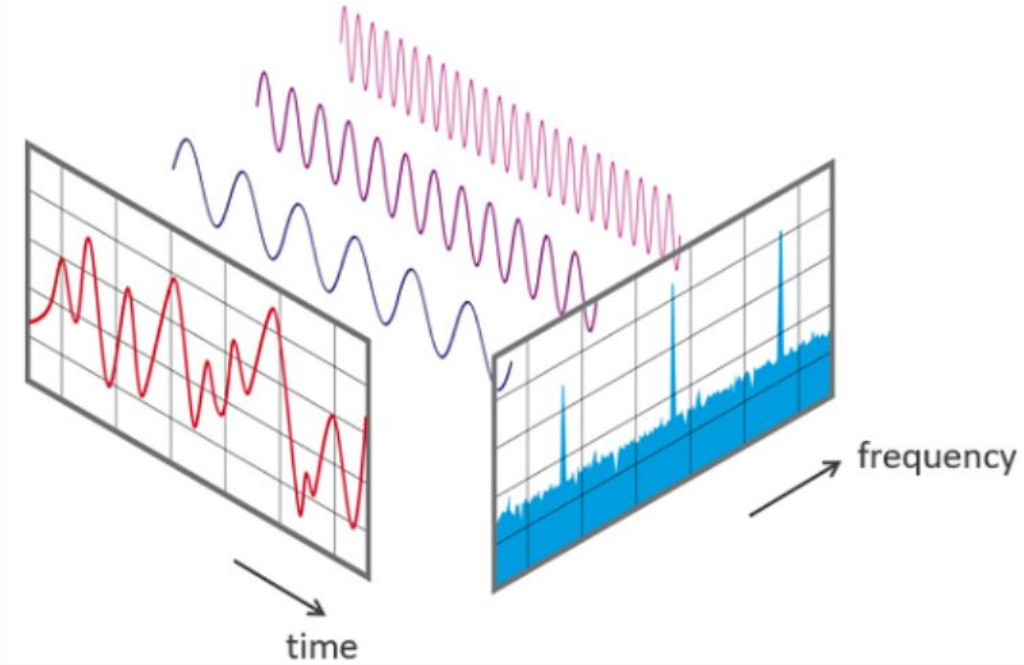
Tổng quan dữ liệu âm thanh

- Âm thanh là một dữ liệu **phi cấu trúc**
- Dữ liệu dạng âm thanh là một **chuỗi các biên độ kèm thời gian tương ứng** tức là dữ liệu dạng âm thanh có dạng sóng.
- Các sóng âm thanh được số hóa bằng cách lấy mẫu chúng ở những khoảng thời gian riêng biệt được gọi là tốc độ lấy mẫu
- Mỗi mẫu là biên độ của sóng tại một khoảng thời gian cụ thể

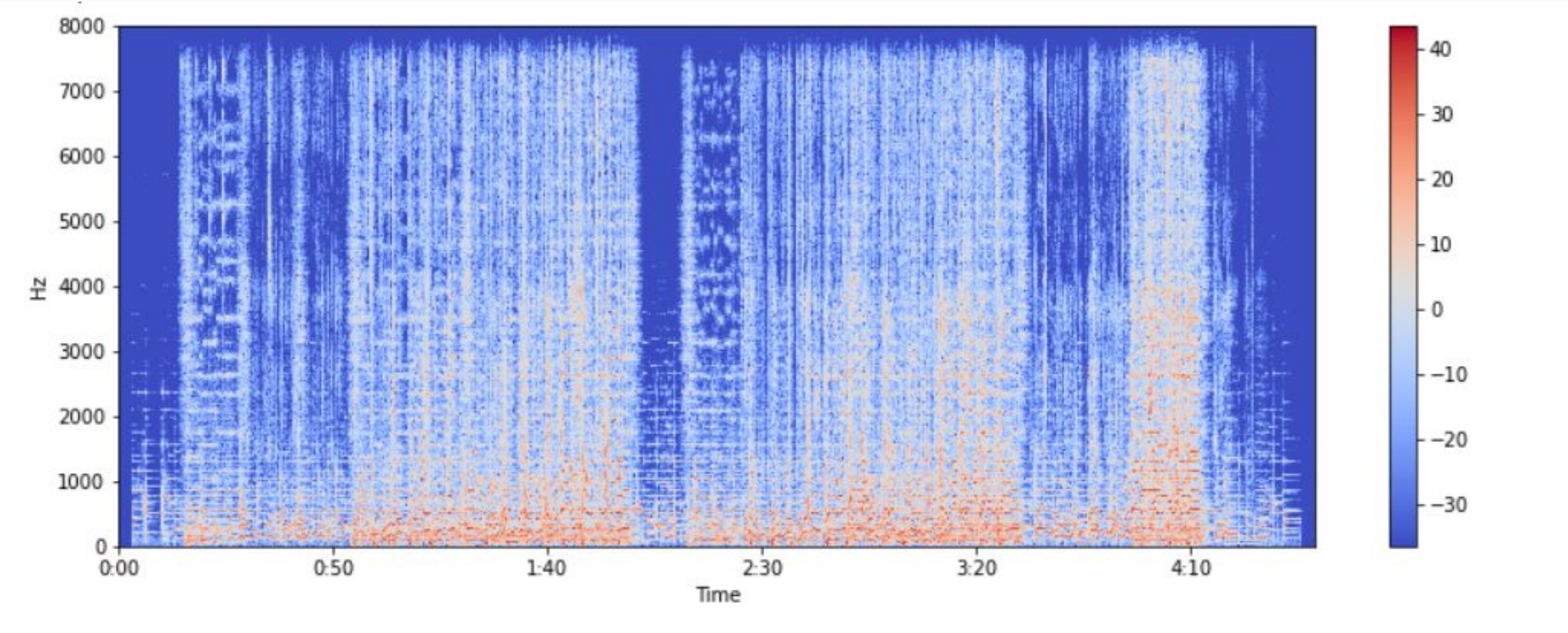


Tổng quan dữ liệu âm thanh

Âm thanh ở dạng tín hiệu liên tục, lấy mẫu với 1 tần số lấy mẫu xác định để chuyển từ dạng liên tục về dạng rời rạc ⇒ biến đổi **Fourier**

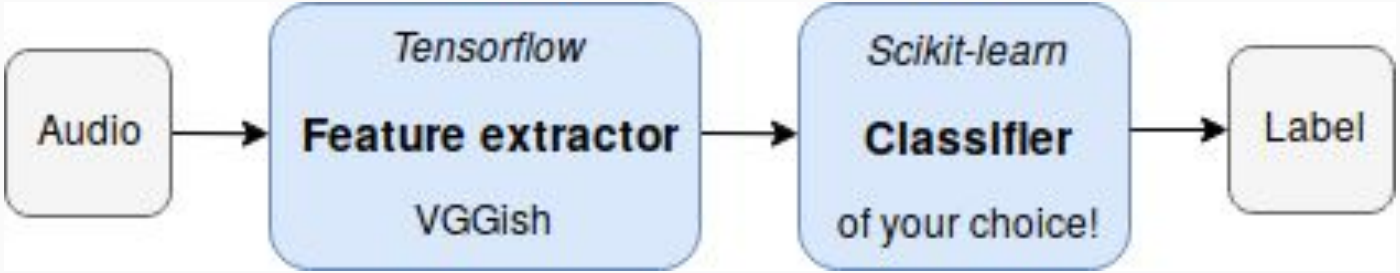


Tổng quan dữ liệu âm thanh

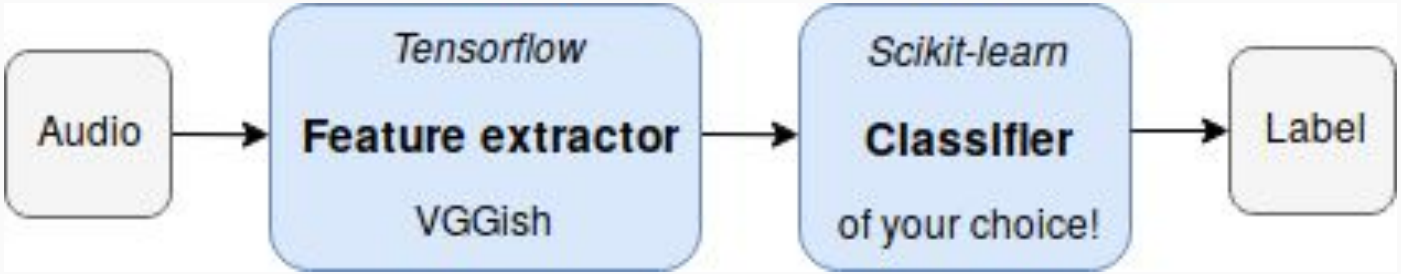


Hướng tiếp cận

- Local feature
- Deep feature



Deep feature VGGish



VGGish: 1 biến thể của VGG 16, có 1 số thay đổi

- input: vector 96*64 log mel spectrogram
- Bỏ đi lớp convolutional và maxpool layers cuối, nên sẽ có 4 nhóm convolution/pool layers thay vì 5

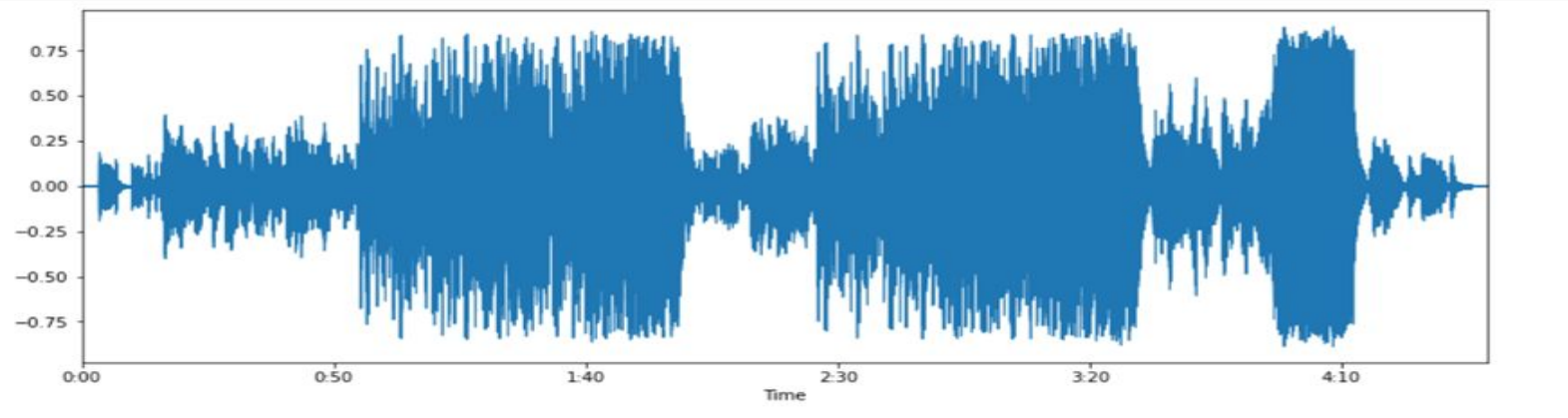
Output: các vector 128 chiều

nguồn: models/research/audioset/vggish at master · tensorflow/models · GitHub

Xử lý dữ liệu âm thanh

2. Trích chọn đặc trưng

- Sử dụng thư viên **librosa** và thư viện **python_speech_features** để trích chọn đặc trưng.
- Đặc trưng dựa trên biên độ âm thanh



Đặc trưng MFCC

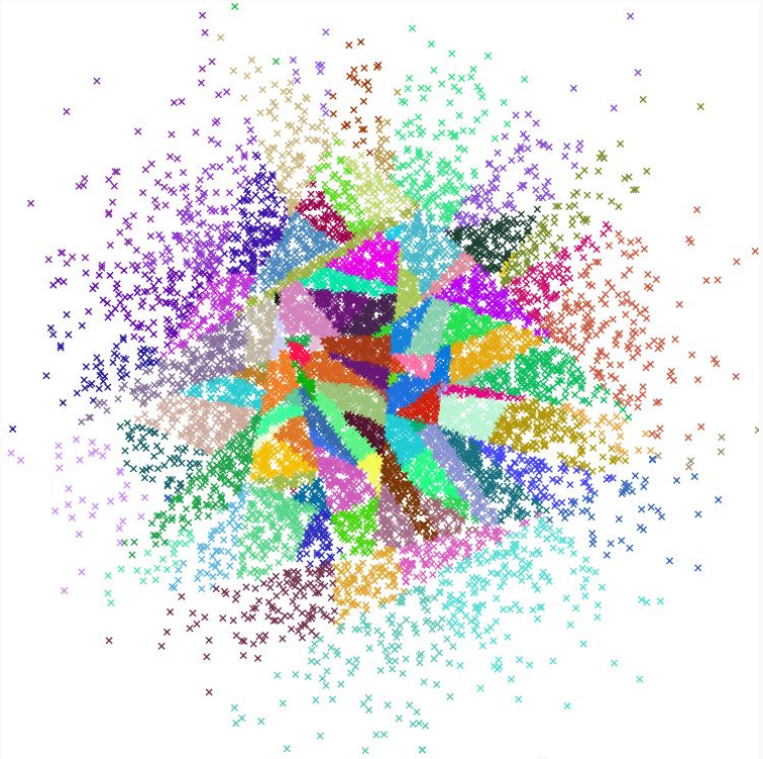
- **Đặc trưng dựa trên đường bao phổ tần số MFCC**

```
def extract_features(y, sr=16000, nfilt=10, winsteps=0.02):
        feat = mfcc(y, sr, nfilt=nfilt, winstep=winsteps)
        return feat
```

- Độ dài của mỗi thuộc tính MFCC là **nfilt** =10
- Việc lấy 10 MFCC và thời gian lấy mẫu là 0.02 giây là quá nhỏ và chưa đủ đặc trưng. Chính vì thế chúng ta sẽ gộp 10 MFCC này lại thành một vector 100 chiều. Đại diện cho mỗi đoạn âm thanh 200ms.

Truy vấn bằng Approximate Nearest Neighbors

Vì không gian lớn **nên** chúng ta sử dụng đến giải thuật **Approximate Nearest Neighbors Oh Yeah (Annoy)** để tìm kiếm với tốc độ nhanh.



- Và cuối cùng đó là đưa ra top 5 bài hát phù hợp nhất với đoạn âm thanh các bạn vừa nhập vào .

```
1 from collections import Counter
2
3 most_song = Counter(results)
4 most_song.most_common()[:5]
```

```
[('Đen Đá Không Đường.mp3', 842),
 ('Story Of My Life.mp3', 80),
 ('Mãi Mãi Là Một Lời Nói Dối (Piano Version).mp3', 60),
 ('Hallelujah.mp3', 53),
 ('Bên Em Là Biển Rộng.mp3', 50)]
```

Tài liệu tham khảo

1. https://github.com/spotify/annoy
2. models/research/audioset/vggish at master · tensorflow/models · GitHub
3. https://arxiv.org/pdf/1709.04396v2.pdf A Tutorial on Deep Learning for Music Information Retrieval
4. Audio Transfer Learning with Scikit-learn and Tensorflow | Jordi Pons

Cám ơn mọi người đã lắng nghe !!!