TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
Tập 20, Số 7 (2023): 1155-1165
Website: https://journal.hcmue.edu.vn
ISSN: 2734-9918

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
Vol. 20, No. 7 (2023): 1155-1165
https://doi.org/10.54607/hcmue.js.20.7.3646(2023)

Bài báo nghiên cứu

CƠ SỞ TOÁN VÀ MFCCS – TRÍCH XUẤT ĐẶC TRƯNG ÂM THANH

Nguyễn Thế Cường**, Nguyễn Thanh Vi, Trương Ngọc Hải
Trường Sĩ quan Thông tin, Việt Nam
**Tác giả liên hệ: Nguyễn Thế Cường – Email: nckcbnckcb@gmail.com*
Ngày nhận bài: 18-10-2022; ngày nhận bài sửa: 31-10-2022; ngày duyệt đăng: 27-4-2023

TÓM TẮT

Hình ảnh và ngôn ngữ (âm thanh, văn bản) là các dạng thông tin quan trọng nhất mà con người đều sử dụng hàng ngày. Đối với lĩnh vực Trí tuệ nhân tạo (AI), hình ảnh và ngôn ngữ cũng là những dữ liệu quan trọng nhất để làm nguyên liệu khi xây dựng các ứng dụng thực tiễn. Các thuật toán học máy (ML) sẽ được huấn luyện dựa trên những dạng dữ liệu như vậy. Tuy nhiên, làm thế nào để đưa một hình ảnh, một đoạn văn bản hay một đoạn âm thanh thành một ma trận hoặc véc-tơ (trích xuất đặc trưng) để đưa vào các thuật toán ML. Có nhiều phương pháp để xử lí đối với từng dạng dữ liệu là hình ảnh hay ngôn ngữ. Dữ liệu dạng âm thanh dường như còn mơ hồ đối với số đông các nhà nghiên cứu, bởi vì chúng không được hiển thì như hình ảnh hay văn bản. Cơ sở Toán học của xử lí dữ liệu âm thanh còn ít được để ý tới. Trong bài viết này, chúng tôi đề cập cơ sở Toán học và phương pháp MFCCs (Mel-Frequency Cepstral Coefficients) nhằm trích xuất các đặc trưng của dữ liệu dạng âm thanh.

Từ khóa: Audio features; Fourier Transform; Mel-Frequency Cepstral Coefficients

1. Giới thiệu

Những năm gần đây, Trí tuệ nhân tạo (AI) đang len lỏi vào từng ngõ ngách của đời sống. Các ứng dụng, các bài báo khoa học về chủ đề AI xuất hiện hàng ngày trên các tạp chí trong và ngoài nước. Với sức mạnh của máy tính được cải thiện nhiều lần so với trước đây, dữ liệu lớn (Big data) cũng đã và đang phát triển không ngừng. Song hành với AI và Big data, không thể không nhắc tới xử lí dữ liệu và các thuật toán tự học (ML). Có thể nói AI là sự kết hợp phức tạp giữa các phương pháp xử lí dữ liệu và các thuật toán ML. Đối với mỗi bài toán thực tiễn, công việc đầu tiên là xử lí dữ liệu hay trích xuất các đặc trưng của dữ liệu để tạo thành các ma trận, các véc-tơ và sau đó đưa bộ dữ liệu đã được xử lí vào huấn luyện. Hai dạng dữ liệu quan trọng nhất là hình ảnh và ngôn ngữ, cùng với đó là các kĩ thuật xử lí dữ liệu như: giảm chiều dữ liệu, phân tích các thành phần chính, số hóa dữ liệu văn bản, đưa miền thời gian về miền tần số bằng cách sử dụng biến đổi Fourier (Lyons, 2022) đối với dữ liệu dạng âm thanh… So với dữ liệu dạng hình ảnh và văn bản, việc trích xuất các đặc trưng

Cite this article as: Nguyen The Cuong, Nguyen Thanh Vi, & Trương Ngoc Hai (2023). Mathematics foundation and MFCCS – Audio feature extraction. *Ho Chi Minh City University of Education Journal of Science, 20*(7), 1155-1165.

của dữ liệu âm thanh dường như còn mơ hồ hơn cả. Bởi vì chúng không hiển thị như văn bản hay hình ảnh, chúng được nghe bằng tai và sự cảm nhận. Có thể kể đến một số phương pháp trích xuất đặc trưng âm thanh tiêu biểu như Mel-Frequency Cepstral Coefficients (MFCCs – tạm dịch là các hệ số phổ quang tần số Mel) (Md. Sahidullah, & Goutam Saha (2012; Ahmed Sajjad et al., 2017; Archek Praveen Kumar et al., 2017) Linear Predictive Coefficients (LPC) (Gulbakshee J. Dharmale, & Dipti D Patil, 2019; Bodke & Satone (2018); Perceptral Linear Predictive (PLP) Coefficients Mohammed, Hussein (2018); Discrete Wavelet Transform (DWT) Mohammad Hasan Rahmani, 2018).

Vậy thực chất tín hiệu âm thanh là gì, chúng được số hóa như thế nào, chúng có các đặc trưng gì và bằng cách nào để trích xuất các đặc trưng của âm thanh? Trong khuôn khổ bài viết này, chúng tôi làm rõ một số vấn đề về: Tín hiệu âm thanh, Cơ sở Toán học và phương pháp Mel-Frequency Cepstral Coefficients (MFCCs – tạm dịch là các hệ số phổ quang tần số Mel) (Md. Sahidullah, & Goutam Saha (2012; Ahmed Sajjad et al., 2017; Archek Praveen Kumar et al., 2017) nhằm trích xuất các đặc trưng của tín hiệu âm thanh.

2. Nội dung

2.1. Cơ sở Toán học của xử lí tín hiệu âm thanh

Để giải quyết các bài toán trong lĩnh vực tương tác người-máy (Cowie, 2001), chẳng hạn như dịch vụ chăm sóc khách hàng, chatbot, trợ lí ảo, chúng ta cần làm việc với dữ liệu dạng văn bản hay âm thanh. Trong mục này chúng tôi làm rõ về cơ sở Toán học của xử lí tín hiệu âm thanh.

2.1.1. Tín hiệu âm thanh đối với học máy

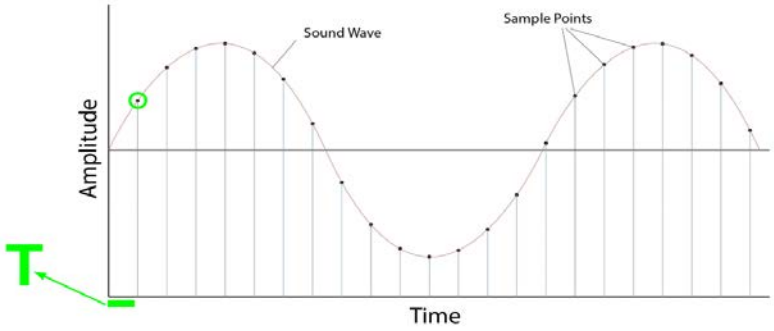
Hiểu một cách đơn giản, âm thanh là các sóng lan truyền giao động cơ học của các phần tử môi trường vật chất. Dạng sóng mang các yếu tố thông tin về tần số, cường độ và âm sắc, có thể tuần hoàn hoặc không tuần hoàn. Dạng sóng có biên độ lớn, ta nghe thấy âm thanh lớn, dạng sóng có tần số cao, ta nghe thấy âm thanh cao. Đối với lĩnh vực học máy (ML) [10], một dạng sóng thường được biểu diễn bởi một hàm theo thời gian:

$$y(t) = A\sin(2\pi ft + \varphi) \quad (1)$$

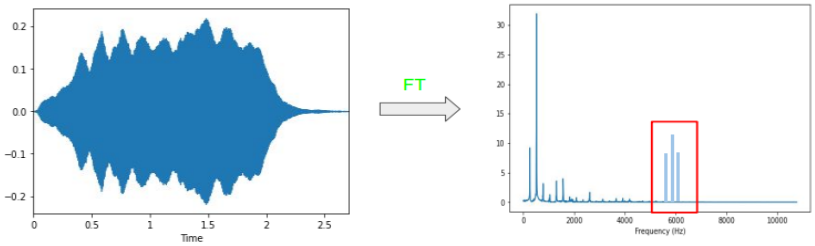
Trong đó A, f, t, φ tương ứng lần lượt là biên độ, tần số, thời gian và pha ban đầu của một dạng sóng âm thanh liên tục theo thời gian $y(t)$. Ta có thể sử dụng phương pháp lấy mẫu để chuyển đổi từ dạng sóng liên tục sang dạng tần số (dãy các giá trị rời rạc), tỉ lệ mẫu $S_r = 1\backslash T$ thường được chọn là 44100, với T là khoảng thời gian giữa 2 mẫu liên tiếp (Hình 1).

2.1.2. Biến đổi Fourier đối với tín hiệu âm thanh

Trong lĩnh vực học máy, âm thanh có các dạng đặc trưng như: các đặc trưng miền thời gian (bao biên độ, căn bậc hai của trung bình của bình phương năng lượng, tỉ lệ băng qua trục hoành), các đặc trưng miền tần số (tỉ lệ dải năng lượng, tâm quang phổ, băng thông), quang phổ. Trong đó việc sử dụng biến đổi Fourier (FT) (Lyons, 2022) để chuyển từ miền thời gian về miền tần số nhằm trích xuất quang phổ. Biến đổi Fourier nhằm phân tích một âm thanh phức thành các thành phần tần số của nó (Hình 2).



Hình 1. *Lấy mẫu*



Hình 2. *Biến đổi Fourier đưa miền thời gian (t) về miền tần số (f)*

Biến đổi Fourier thực: nhằm so sánh tín hiệu theo thời gian $g(t)$ với rất nhiều các tần số của các sóng sin. Đối với mỗi tần số $f \in \mathbb{R}$, biến đổi Fourier cho ta một độ lớn $d_f \in \mathbb{R}$ và một pha $\varphi_f \in \mathbb{R}$, độ lớn d_f cao tức là độ tương tự giữa sóng $\sin(2\pi \cdot (ft - \varphi))$ và tín hiệu theo thời gian $g(t)$ cũng cao,

$$d_f = \max_{\varphi \in [0,1)} \left(\int g(t) \cdot sin\left(2\pi(ft - \varphi)\right) dt \right), \tag{2}$$

$$\varphi_f = arg \max_{\varphi \in [0,1)} \left(\int g(t) \cdot \sin\left(2\pi(ft - \varphi)\right) dt \right). \tag{3}$$

Biến đổi Fourier phức: ta có thể mã hoá cả độ lớn và pha trong một số phức $c_f = \frac{d_f}{\sqrt{2}} e^{-i2\pi\varphi_f}$. Giả sử ta có một tín hiệu âm thanh liên tục $g(t)$ $g: \mathbb{R} \to \mathbb{R}$, biến đổi Fourier phức có thể được mô tả một cách ngắn gọn như sau:

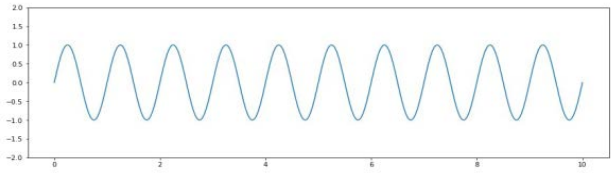
$$\hat{g}(f) = c_f \qquad \hat{g}: \mathbb{R} \to \mathbb{C},$$

$$\hat{g}(f) = \int g(t)\, e^{-i2\pi ft} dt = \int g(t) \cos(-2\pi ft) dt + i \int g(t) \sin(-2\pi ft)\, dt \tag{4}$$

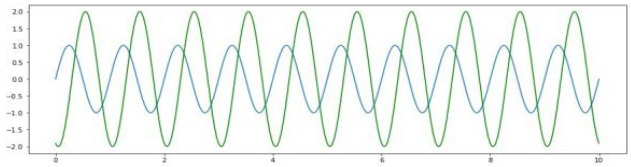
Biến đổi Fourier ngược (IFT): Để đưa tín hiệu từ miền tần số về miền thời gian, ta sử dụng biến đổi Fourier ngược (xem Hình 3, 4, 5)

$$g(t) = \int c_f \cdot e^{i2\pi ft}\, df. \tag{5}$$

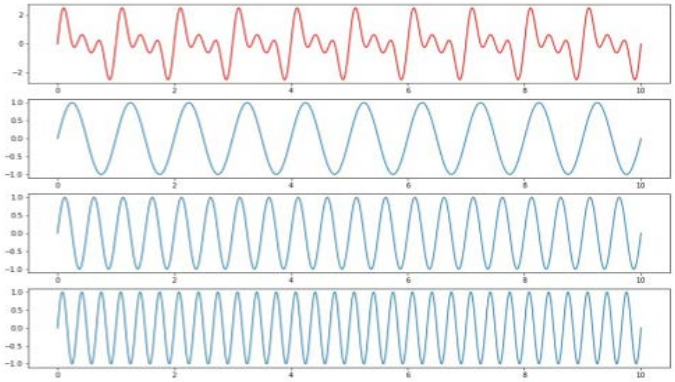
Trong đó, $e^{i2\pi ft}$ là tone nguyên thuỷ của tần số f, $c_f \cdot e^{i2\pi ft}$ là tone nguyên thuỷ có trọng là độ lớn và pha biểu thị trong số phức c_f, $\int c_f \cdot e^{i2\pi ft}\, df$ là tổng tất cả các sóng sin có trọng.



Hình 3. $e^{i2\pi ft}$



Hình 4. $c_f e^{i2\pi ft}$



Hình 5. $\int c_f \cdot e^{i2\pi ft}\, df$

Biến đổi Fourier rời rạc (DFT): Để số hoá một âm thanh liên tục ta thực hiện lấy mẫu, một tín hiệu kĩ thuật số $g(t)$ có thể được xấp xỉ bởi một tín hiệu rời rạc $x(n)$:

$g(t) \approx x(n)$

$t = nT$

Với T là quãng thời gian giữa 2 mẫu. Biến đổi Fourier $\hat{g}(f)$ sẽ được xấp xỉ bởi biến đổi Fourier rời rạc $\hat{x}(f)$ của tín hiệu rời rạc $x(n)$ tương ứng:

$$\hat{g}(f) = \int g(t)\, e^{-i2\pi ft} dt$$

$$\hat{x}(f) = \sum_n x(n)\, e^{-i2\pi fnT}. \quad (6)$$

Trong 1 vòng (chu kì 2π) (tức thời gian của 1 chu kì là NT, tần số là $\frac{1}{NT} Hz$ hay $\frac{2\pi}{NT}$rad/s), chọn số lượng mẫu N là một số hữu hạn, để thuận tiện cho biến đổi Fourier ngược ta cũng chọn số lượng tần số cơ bản bằng với số lượng mẫu N,

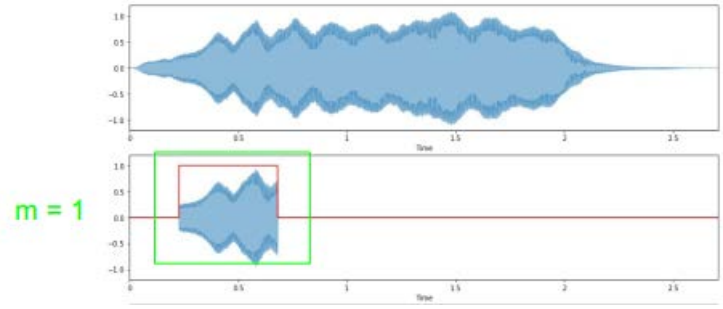
$$F(k) = \frac{k}{NT} = \frac{kS_r}{N}, k = [0, \dots, N-1], \quad (7)$$

như vậy ta có:

$$\hat{x}\left(\frac{k}{NT}\right) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi \frac{k}{NT} nT} . \quad (8)$$

Khi $k = \frac{N}{2}$ ta có $F\left(\frac{N}{2}\right) = \frac{S_r}{2}$ (tần số Nyquist), lúc này ta có biến đổi Fourier nhanh (FFT), FFT hoạt động khi số lượng ngăn tần N là một hàm mũ cơ số 2.

Biến đổi Fourier thời gian ngắn (STFT): Nhằm tránh mất mát thông tin, một file âm thanh dài sẽ được chia nhỏ thành các frame có chồng lấp, sau đó ta áp dụng FFT và hàm windowing $w(k)$ cho từng frame (Hình 6).



Hình 6. *Từ DFT-sang-STFT*

$$S(k,m) = \sum_{n=0}^{N-1} x(n + mH) \cdot w(n) \cdot e^{-i2\pi \frac{k}{NT} nT},$$

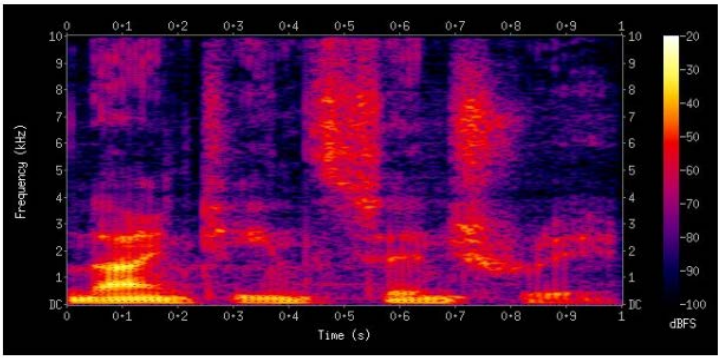
với m là frame thứ m, H là hopsize (phần chồng lấp giữa 2 frame liên tiếp). Như vậy, nếu áp dụng DFT cho một tín hiệu âm thanh ta được một vector quang phổ N chiều (N hệ số Fourier phức), áp dụng STFT ta sẽ được một ma trận quang phổ Y có cỡ (số lượng ngăn tần, số lượng frames), với

$$\#ngăn\ tần\ =\ N = \frac{framesize}{2} + 1 \quad (9)$$

$$\#frames = \frac{\#samples - framesize}{hopsize} + 1. \quad (10)$$

Độ lớn quang phổ tại ngăn tần thứ k và frame thứ m:

$Y(k,m) = |S(k,m)|^2$



Hình 7. *Quang phổ*

Như vậy để trích xuất đặc trưng miền tần số của một tín hiệu âm thanh, chúng ta sử dụng phương pháp lấy mẫu và trên cơ sở của biến đổi Fourier phức để thu được các thông tin về tần số và độ lớn quang phổ. Trong nhiều bài toán (đặc biệt là nhận dạng giọng nói) thì quang phổ chưa phải là một sự lựa chọn tốt. Do đó ta cần thêm vài bước nữa để có được MFCCs (Md. Sahidullah, & Goutam Saha, 2012), phương pháp trích xuất các đặc trưng một tín hiệu âm thanh phổ biến hơn và hiệu quả hơn quang phổ.

2.2. Mel-Frequency Cepstral Coefficients (MFCCs)

Thang Mel: Là thang cao độ cảm nhận, được người nghe đánh giá và có khoảng cách các quãng bằng nhau. Điểm tham chiếu giữa thang Mel và phép đo tần số thông thường xác định bằng cách gán cao độ cảm nhận 1000 mels cho âm có tần số 1000 Hz. Công thức phổ biến để biến đổi f Hz thành m mels là

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right); f = 700\left(10^{m/2595} - 1\right). \quad (11)$$

Để trích xuất Mel-spectrograms ta làm theo các bước như sau:

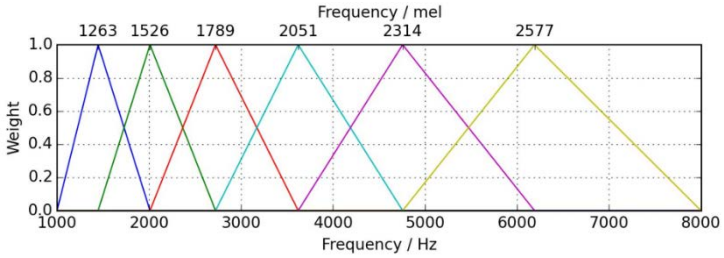
- Sử dụng STFT đối với tín hiệu âm thanh.
- Biến đổi các biên độ thành dBs.
- Biến đổi các tần số thành thang Mel.

a) Chọn số lượng dải mel.

b) Xây dựng các ngân hàng lọc mel (Mel-filter-banks).

1. Biến đổi tần số thấp nhất và cao nhất thành Mel.
2. Tạo các dải mel cách đều.
3. Biến đổi các điểm đó trở lại Herzt.
4. Làm tròn đến ngăn tần gần nhất.
5. Tạo các bộ lọc tam giác.

c) Áp dụng các ngân hàng lọc mel đối với quang phổ.

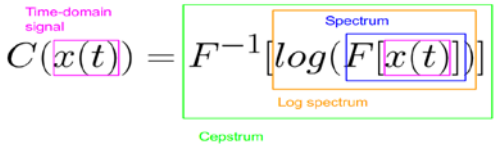


Hình 8. *Ngân hàng lọc Mels*

Sau bước trên ta có ma trận (kích thước ngân hàng lọc Mel) $M = (\#dải\ Mels, \#ngăn\ tần)$, áp dụng ngân hàng lọc Mels đối với quang phổ $Y = (\#ngăn\ tần, \#frames)$ ta có ma trận đặc trưng như sau:

$$MY = (\#dải\ Mels, \#frames) \quad (12)$$

Cuối cùng, để trích xuất **cepstrum** (tạm dịch là phổ quang) ta thực hiện biến đổi Fourier ngược (dùng biến đổi Cosin rời rạc) để chuyển từ miền tần số về miền thời gian, ta thu được MFCCs.

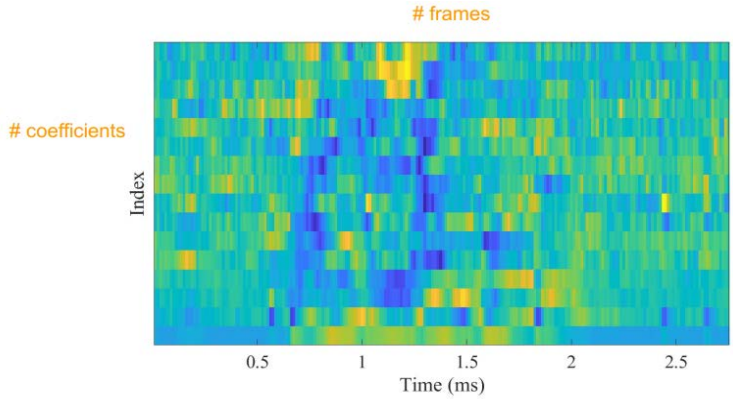


Hình 9. *Tính toán Cepstrum*

Tóm lại, các bước của phương pháp trích xuất đặc trưng tín hiệu âm thanh MFCCs như sau:

Dạng sóng → STFT → Log(độ lớn quang phổ) → thang Mel → biến đổi Cosin rời rạc → MFCCs.

Sử dụng biến đổi cosin rời rạc (một dạng đơn giản của biến đổi Fourier) cho ta các hệ số là các giá trị thực, do đó sẽ giảm số chiều của phổ quang biểu diễn. Thông thường ta sẽ chọn 13 hệ số đầu tiên trong các hệ số của MFCCs (chứa hầu hết các thông tin như: các đỉnh biên độ, bao quang phổ…), các hệ số tiếp theo là đạo hàm và đạo hàm cấp 2 của MFCCs, như vậy thông thường có 39 hệ số MFCCs trong một frame.



Hình 10. *Minh hoạ MFCCs*

2.3. Cài đặt

Trong phần này, chúng tôi cung cấp phần cài đặt bằng ngôn ngữ lập trình Python3 để trích xuất các đặc trưng MFCCs của các tập dữ liệu âm thanh.

```
#Khai báo các thư viện
import librosa
import os
import json
#Trích xuất MFCCs từ tập dữ liệu có mã nguồn mở Surrey Audio-Visual Expressed
#Emotion (SAVEE) (Philip Jackson & Sanaul Haq, 2015) và lưu dưới dạng file .json
DATASET_PATH = "SAVEE"
JSON_PATH = "SAVEE/SAVEE.json"
```

```
SAMPLES_TO_CONSIDER = 22050 # số lượng mẫu
def preprocess_dataset(dataset_path, json_path, num_mfcc = 39, n_fft = 2048, hop_length
= 512):
    """:tham số dataset_path (dạng string): đường dẫn tới tập dữ liệu
    :tham số json_path (string): đường dẫn lưu file .json của MFCCs
    :tham số num_mfcc (số nguyên): số lượng các hệ số đặc trưng muốn trích xuất
    :tham số n_fft (số nguyên): độ dài một frame (số lượng mẫu) áp dụng FFT.
    :return: trả kết quả là file .json lưu các ma trận đặc trưng MFCCs của các file âm thanh
"""
    data = {'mapping': [], 'labels': [], 'MFCCs': [], 'files': []}
    for i, (dirpath, dirnames, filenames) in enumerate(os.walk(dataset_path)):
        if dirpath is not dataset_path:
         # lưu nhãn trong mapping
           label = dirpath.split('/')[-1]
           data['mapping'].append(label)
           print("\nProcessing: '{}'".format(label))
           #xử lí tất cả các files và lưu MFCCs
           for f in filenames:
              file_path = os.path.join(dirpath, f)
              #Tải file audio và tách để đảm bảo độ dài nhất quán giữa các file khác nhau
              signal, sample_rate = librosa.load(file_path)
              if len(signal) >= SAMPLES_TO_CONSIDER:
                 signal = signal[:SAMPLES_TO_CONSIDER]
                 #trích xuất MFCCs
                 MFCCs = librosa.feature.mfcc(signal, sample_rate, n_mfcc = num_mfcc, n_fft
=       n_fft, hop_length = hop_length)
                    #lưu trữ dữ liệu để phân tích các bản ghi
                    data["MFCCs"].append(MFCCs.T.tolist())
                    data["labels"].append(i - 1)
                    data["files"].append(file_path)
                    print("{}: {}".format(file_path, i - 1))
    #lưu dữ liệu trong file .json
    with open(json_path, 'w') as fp:
        json.dump(data, fp, indent = 4)

if __name__ == "__main__":
    preprocess_dataset(DATASET_PATH, JSON_PATH)
```

2.4. Áp dụng

Trong bài viết này, chúng tôi sử dụng các tập dữ liệu có mã nguồn mở như sau:

Surrey Audio-Visual Expressed Emotion (SAVEE).

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS).

Toronto emotional speech set (TESS).

Crowd-sourced Emotional Mutimodal Actors Dataset (CREMA-D).

Sau khi áp dụng phần cài đặt 2.2 để có được MFCCs của các tập dữ liệu âm thanh, chúng tôi áp dụng thuật toán phân loại nhị phân Support Vector Machine (SVM) (Manas Jain, 2020) để xây dựng một ứng dụng nhận dạng sắc thái giọng nói.

Các tập dữ liệu được chia theo tỉ lệ 90/10, tức 90% dùng cho huấn luyện và 10% dùng cho kiểm thử. Sử dụng đánh giá chéo 10 lần (10-fold-CV) để đánh giá độ chính xác. Chúng tôi sử dụng ngôn ngữ lập trình Python và các thư viện cần thiết để cài đặt và đánh giá độ chính xác của ứng dụng. Các tập dữ liệu đều được tiêu chuẩn hoá (standardization) sau khi dùng MFCCs để trích xuất các đặc trưng (39 đặc trưng đầu tiên). Các tập dữ liệu trên đều được chia thành 2 lớp, mỗi lớp bao gồm nhiều cụm như sau: Lớp {+}(positive) bao gồm các cụm 'happy', 'neutral', 'pleasant and surprise' có thể có cả cụm 'calm'; lớp {-}(negative) bao gồm các cụm 'anggry', 'disgust', 'fear' và 'sad'. Với một file âm thanh mới, ứng dụng sẽ phân loại một file âm thanh vào lớp {+}(positive) hoặc lớp {-}(negative).

Kết quả huấn luyện và kiểm thử được mô tả trong bảng dưới đây:

Độ chính xác kiểm thử, độ chính xác 10-fold-CV, thời gian thực hiện của SVM và SVM trên các tập dữ liệu

Các tập dữ liệu (số lượng file âm thanh)	**MFCCs và SVM** Độ chính xác điểm thử (%) 10-fold-CV (%) Thời gian thực hiện (s
SAVEE (480)	70.8 66.9 +/- 8.0 9.99
RAVDESS (1,440)	60.4 59.3 +/- 3.5 123.46
TESS (2,800)	98.9 98.9 +/- 0.7 710.69
CREMA-D (7,442)	67.9 68.4 +/- 1.4 5,779.38

Từ bảng kết quả chúng ta có thể thấy rằng: đối với tập có ít dữ liệu SAVEE (480 files âm thanh), (MFCCs và SVM) cho độ chênh lệch khá lớn giữa độ chính xác kiểm thử (70.8)

và 10-fold-CV (66.9 +/- 8.0), điều này là bởi bộ dữ liệu SAVEE có số lượng nhỏ (480) khiến cho mô hình không ổn định. Về cơ bản, đối với các bộ dữ liệu có số lượng lớn, độ chính xác phân loại của phương pháp (MFCCs và SVM (Manas Jain, 2020; Sinith et al., 2015) đạt được tương đối cao và ổn định. Tập dữ liệu TESS đạt được độ chính xác cao nhất trong phân loại đối với phương pháp này, điều này là bởi chỉ có 2 diễn viên thu âm tập dữ liệu (26 tuổi và 64 tuổi, cách biệt khá lớn), dữ liệu không bị chồng lấp nhiều.

4. Kết luận

Bài viết đã làm sáng tỏ rằng, tư tưởng Toán học của trích xuất đặc trưng miền tần số của tín hiệu âm thanh chính là dùng phương pháp lấy mẫu và biến đổi Fourier rời rạc phức. Để chuyển từ miền tần số về miền thời gian, phương pháp MFCCs sử dụng thang cao độ cảm nhận Mel và biến đổi Cosin rời rạc độ lớn quang phổ ở bước cuối cùng. Các đặc trưng MFCCs của một file âm thanh là một ma trận có cỡ (số lượng các hệ số MFCCs, số lượng frames). Ngôn ngữ lập trình Python3 được sử dụng để cài đặt MFCCs, thực hiện ví dụ với các tập dữ liệu có mã nguồn mở (SAVEE, RAVDESS, TESS, CREMA-D). Từ kết quả thực nghiệm cho thấy phương pháp MFCCs nhằm trích xuất các đặc trưng âm thanh, kết hợp với SVM cho ta kết quả phân loại sắc thái giọng nói tương đối tốt và ổn định. Trong thực tế, dữ liệu thu thập được chưa đủ tốt, do sự phức tạp của giọng nói mỗi người là khác nhau về tần số, cao độ, năng lượng, tốc độ, cảm xúc… Qua đó cho thấy việc nghiên cứu và cải tiến các kĩ thuật trích xuất đặc trưng của âm thanh vẫn cần được thúc đẩy và quan tâm hơn nữa.

❖ ***Tuyên bố về quyền lợi:*** *Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.*

TÀI LIỆU THAM KHẢO

Ahmed Sajjad, Ayesha Shirazi, Nagma Tabassum, Mohd Saquib, & Naushad Sheikh (2017). Speaker Identification and Verification Using MFCCs and SVM. *International Research Jounal of Engineering and Technology (IRJET), 4*(2).

Archek Praveen Kumar, Ratnadeep Roy, Sanyog Rawat, & Prathibha Sudhakaran (2017). Continuous Telugu Speech Recognition throught Combined Feature Extraction by MFCCs and DWPD Using HMM based DNN Techniques. *International Journal of Pure and Applied Mathematics, 114*(11).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bodke, R. D., & Satone, M. P. (2018). A review on Speech Feature Techniques and Classification Techniques. *International Journal of Trend in Scientific research and Development, 2*(4).

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Kollias, S., Fellenz, W., Taylor, J. (2001). Emotion recognition in humancomputer interaction. in *IEEE Signal Process*.

Gulbakshee J. Dharmale, & Dipti D. Patil (2019). Evaluation of Phonetic System for Speech Revognition on Smartphone. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(10).

Lyons, R. G. (2022). Understanding digital signal processing's frequency domain. *RF Design magazine*.

Manas Jain, Shruthi Narayan, Pratibha Balaji, Bharath K. P., Abhijit Bhowmick, Karthik, R, & Rajesh Kumar Muthu (2020). Speech Emotion Recognition using Support Vector Machine. *Electrical Engineering and Systems Science, Audio and Speech Processing*.

Md. Sahidullah, & Goutam Saha (2012). Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication, 54*(4), 543-565.

Mohammed Hussein, Alkassab, M., Mohammed, H., Abdulaziz Hind, & Jagmagji Ahmed (2018). Speech Recognition System with Different Methods of Feature Extraction. *International Journal of Innovative Research in Computer and Communication Engineering, 6*(3).

Mohammad Hasan Rahmani, Farshad Almasganj, & Seyyed Ali Seyyedsalehi (2018). Audio-visual feature fusion via deep neural networks for automatic speech recognition. *Digital Signal Processing*.

Philip Jackson, & Sanaul Haq (12 April 2015). Justdreamweaver.com. Retrieved from https://www.kaggle.com/datasets/ejlok1/surrey-audiovisual-expressed-emotion-savee?resource=download

Sinith, M. S., Aswathi, E., Deepa, T. M., Shameema, C. P., & Shiny, R. (2015). Emotion Recognition from Audio Signals using Support Vector Machine. in *IEEE Recent Advances in Intelligent Computational Systems*, Trivandrum.

MATHEMATICS FOUNDATION AND MFCCS – AUDIO FEATURE EXTRACTION

Nguyen The Cuong*, Nguyen Thanh Vi, Trương Ngọc Hai

Telecommunications University, Vietnam

**Corresponding author: Nguyen The Cuong – Email: nckcbnckcb@gmail.com*

Received: October 18, 2022; Revised: October 31, 2022; Accepted: April 27, 2023

ABSTRACT

The most significant types of information that people use daily are image and language (sound and text). Images and language are the most crucial data to employ as raw materials when developing real-world applications in the field of artificial intelligence (AI). On these kinds of data, machine learning (ML) algorithms will be trained. However, feature extraction is the process of converting an image, text, or audio file into a matrix or vector for use in machine learning algorithms. Processing visual or linguistic input can be done in a variety of ways. Because audio data are not presented as visuals or text, most researchers find them to be unclear. Little attention has been paid to the mathematical foundations of audio data processing. The mathematical foundation and the MFCCs (Mel-Frequency Cepstral Coefficients) approach to extract the features of the audio data are discussed in this article.

Keywords: Audio features; Fourier Transform; Mel-Frequency Cepstral Coefficients