

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I



BÁO CÁO BÀI TẬP LỚN

MÔN HỌC : HỆ CƠ SỞ DỮ LIỆU ĐA PHƯƠNG TIỆN

ĐỀ TÀI: XÂY DỰNG HỆ CSDL NHẬN DẠNG TIẾNG CÔNG CỤ/DỤNG CỤ

Giảng viên hướng dẫn: Nguyễn Đình Hóa

Lớp: D19-027

Nhóm BTL: 07

Danh sách thành viên:

Thân Tuấn Bảo B19DCCN061

Nguyễn Văn Dũng B19DCCN128

Ninh Thị Phụng B19DCCN516

Hà Nội – 2023

Mục lục

I. Bộ dữ liệu âm thanh công cụ/dụng cụ	1
1. Xây dựng, sưu tầm bộ dữ liệu âm thanh công cụ/dụng cụ có động cơ	1
2. Đặc điểm dữ liệu âm thanh công cụ/dụng cụ có động cơ	1
II. Các kỹ thuật xử lý và nhận dạng âm thanh công cụ/dụng cụ hiện hành	2
1. Kỹ thuật sử dụng học máy/học sâu	2
2. Kỹ thuật trích xuất đặc trưng.....	2
3. Kỹ thuật phân tích tín hiệu âm thanh	3
4. Kỹ thuật đồng bộ hóa tín hiệu	4
5. Kỹ thuật sử dụng xác suất	4
III. Xây dựng hệ thống nhận dạng âm thanh công cụ/dụng cụ	5
1. Sơ đồ khối hệ thống.....	5
2. Quy trình thực hiện.....	5
3. Các thuộc tính âm thanh dùng để nhận dạng	5
4. Kỹ thuật trích rút các thuộc tính âm thanh	5
5. Lưu trữ các thuộc tính âm thanh.....	6
6. Nhận dạng âm thanh bằng các thuộc tính âm thanh.....	8
IV. Kết quả xây dựng hệ thống	9
1. Demo hệ thống	9
2. Đánh giá kết quả.....	12

I. Bộ dữ liệu âm thanh công cụ/dụng cụ

1. Xây dựng, sưu tầm bộ dữ liệu âm thanh công cụ/dụng cụ có động cơ
 - Bộ dữ liệu âm thanh công cụ/dụng cụ có động cơ gồm các loại sau đây:
 - + Máy bơm nước
 - + Máy cưa
 - + Máy cắt cỏ
 - + Máy hút bụi
 - + Máy khoan
 - + Máy nén khí
 - + Máy phát điện
 - + Máy thổi lá
 - + Máy ép trái cây
 - + Máy sấy tóc
 - Có tổng cộng 137 file của 10 loại âm thanh công cụ/dụng cụ, mỗi file có độ dài 10s, được ghi âm thủ công và sưu tầm từ nhiều nguồn trên internet và ghi lại âm thanh lúc hoạt động của từng loại dụng cụ được nói ở trên.
2. Đặc điểm dữ liệu âm thanh công cụ/dụng cụ có động cơ
 - Tần số: Âm thanh công cụ/dụng cụ có động cơ thường có tần số cao hơn so với âm thanh tự nhiên.
 - Độ lớn: Âm thanh công cụ/dụng cụ có động cơ có độ lớn cao hơn các loại âm thanh khác, dao động từ khoảng 70-110 dB nên thường rất ồn.
 - Độ phức tạp: Âm thanh của các công cụ/dụng cụ có động cơ thường có độ phức tạp cao hơn so với các âm thanh khác, bởi vì âm thanh công cụ/dụng cụ có động cơ thường có nhiều thành phần âm thanh khác nhau phối hợp với nhau để tạo ra âm thanh cuối cùng.
 - Độ nhiễu: Âm thanh của các công cụ/dụng cụ có động cơ chứa nhiều tạp âm và nhiễu từ nhiều nguồn khác nhau (môi trường xung quanh, các bộ phận khác trong công cụ/dụng cụ)
 - Tính bất ổn định: Dữ liệu âm thanh các công cụ/dụng cụ có động cơ thường không ổn định và thay đổi theo thời gian, vì các động cơ không hoạt động ở tốc độ và công suất ổn định.
 - Tính lặp lại: Dữ liệu âm thanh các công cụ/dụng cụ có khả năng lặp đi lặp lại nhiều lần mà vẫn giữ nguyên tính chất âm thanh.

II. Các kỹ thuật xử lý và nhận dạng âm thanh công cụ/dụng cụ hiện hành

1. Kỹ thuật sử dụng học máy/học sâu

Thuật toán máy học là một trong những kỹ thuật được sử dụng rộng rãi trong nhận dạng tiếng ồn và tiếng nền, bao gồm cả nhận dạng tiếng công cụ/dụng cụ. Đây là một phương pháp dựa trên việc xử lý dữ liệu thô thông qua các bước chuyển đổi và trích xuất đặc trưng, sau đó sử dụng các mô hình máy học để phân loại và nhận dạng. Các bước xử lý dữ liệu thô và trích xuất đặc trưng bao gồm:

Tiền xử lý dữ liệu: Loại bỏ tiếng ồn và tiếng nền, tách riêng tiếng của công cụ/dụng cụ để đưa vào phân tích.

Trích xuất đặc trưng: Từ tín hiệu âm thanh, trích xuất ra các đặc trưng như tần số, biên độ, và thời gian để mô tả tính chất của tín hiệu.

Sau khi có các đặc trưng, các thuật toán máy học được sử dụng để phân loại và nhận dạng tín hiệu âm thanh.

Các thuật toán phổ biến bao gồm:

- Support Vector Machine (SVM): SVM là một thuật toán học có giám sát dựa trên việc tìm ra đường biên phân cách giữa các lớp dữ liệu. Nó có thể được sử dụng để phân loại các mẫu âm thanh dựa trên các đặc trưng được trích xuất.
- Random Forest (RF): RF là một thuật toán học có giám sát, dựa trên việc xây dựng một tập hợp các cây quyết định ngẫu nhiên. Nó có thể được sử dụng để phân loại các mẫu âm thanh dựa trên các đặc trưng được trích xuất.
- Convolutional Neural Networks (CNN) là một kiểu mạng neural được sử dụng rộng rãi trong việc xử lý ảnh và âm thanh. Cấu trúc CNN bao gồm các lớp convolution, pooling và fully-connected layer. Lớp convolution được sử dụng để trích xuất đặc trưng của tín hiệu âm thanh, trong đó các bộ lọc được sử dụng để tìm kiếm các đặc trưng của tín hiệu âm thanh. Sau đó, lớp pooling được sử dụng để giảm kích thước của đặc trưng, từ đó giảm số lượng tham số và tính toán của mô hình. Cuối cùng, fully-connected layer được sử dụng để phân loại tín hiệu âm thanh thành các lớp.
- Recurrent Neural Networks (RNN) là một kiểu mạng neural được sử dụng để xử lý dữ liệu chuỗi như tín hiệu âm thanh. Cấu trúc RNN bao gồm một chuỗi các đơn vị RNN, trong đó mỗi đơn vị RNN nhận đầu vào từ đơn vị trước đó và đầu ra từ đơn vị hiện tại. Điều này cho phép mô hình học được sự phụ thuộc giữa các khung thời gian liên tiếp của tín hiệu âm thanh, từ đó giúp cải thiện độ chính xác của mô hình.
- Long Short-Term Memory (LSTM) network là một kiểu RNN được thiết kế để giải quyết vấn đề gradient vanishing và gradient exploding trong quá trình huấn luyện mạng neural. Cấu trúc LSTM bao gồm các đơn vị LSTM, trong đó mỗi đơn vị LSTM bao gồm một bộ nhớ dài hạn, một bộ nhớ ngắn hạn và các cổng gate để điều chỉnh việc lưu trữ và truyền thông tin giữa các đơn vị.

2. Kỹ thuật trích xuất đặc trưng

Kỹ thuật trích xuất đặc trưng là quá trình chuyển đổi tín hiệu âm thanh từ miền thời gian sang miền tần số để tạo ra các đặc trưng mô tả tính chất của tín hiệu. Các đặc trưng này được sử dụng để huấn luyện mô hình nhận dạng tiếng công cụ/dụng cụ.

Có nhiều phương pháp trích xuất đặc trưng khác nhau, tuy nhiên, trong các kỹ thuật trích xuất đặc trưng tiếng công cụ/dụng cụ thường sử dụng các phương pháp sau:

- Mel-Frequency Cepstral Coefficients (MFCC): Là một trong những kỹ thuật phổ biến nhất trong trích xuất đặc trưng âm thanh. MFCC sử dụng bộ lọc Mel để tách các thành phần tần số trong tín hiệu âm thanh, sau đó sử dụng phân tích Cepstral để trích xuất các đặc trưng.
- Linear Predictive Coding (LPC): Sử dụng mô hình tuyến tính để mô tả tín hiệu âm thanh. LPC phân tích tín hiệu thành các khối có độ dài nhất định và trích xuất các hệ số tuyến tính để mô hình hóa tín hiệu.
- Gammatone Frequency Cepstral Coefficients (GFCC): Tương tự như MFCC, GFCC sử dụng bộ lọc Gammatone để tách các thành phần tần số trong tín hiệu âm thanh. Sau đó sử dụng phân tích Cepstral để trích xuất các đặc trưng.
- Spectral Subband Centroid Features (SSC): Sử dụng các bộ lọc tần số để phân tách tín hiệu âm thanh thành các dải tần số và tính toán trung bình của tần số trung tâm của mỗi dải để trích xuất các đặc trưng.

Các kỹ thuật trích xuất đặc trưng này thường được sử dụng đồng thời với các kỹ thuật khác như FFT (Fast Fourier Transform), để tạo ra các đặc trưng chi tiết hơn và có tính phân loại tốt hơn. Sau khi trích xuất đặc trưng, các đặc trưng này được sử dụng để huấn luyện mô hình nhận dạng tiếng công cụ/dụng cụ.

3. Kỹ thuật phân tích tín hiệu âm thanh

Các kỹ thuật phân tích tín hiệu âm thanh là quá trình chuyển đổi tín hiệu âm thanh từ miền thời gian sang miền tần số và tạo ra các biểu diễn số học mô tả tính chất của tín hiệu âm thanh. Dưới đây là một số kỹ thuật phân tích tín hiệu âm thanh:

- Short-Time Fourier Transform (STFT): là kỹ thuật phân tích tín hiệu âm thanh bằng cách chia tín hiệu thành các phân đoạn và áp dụng Fourier Transform trên từng phân đoạn để tạo ra một biểu diễn tần số của tín hiệu. Kỹ thuật STFT được sử dụng để phân tích tín hiệu âm thanh có tính chất thay đổi nhanh chóng theo thời gian.
- Wavelet Transform (WT): là kỹ thuật phân tích tín hiệu âm thanh dựa trên việc biến đổi Wavelet, một loại hàm số với tính chất giúp phân tích tín hiệu ở các tần số khác nhau và độ phân giải thời gian khác nhau. Kỹ thuật WT được sử dụng để phân tích tín hiệu âm thanh có tính chất đa tần số và đa thời gian.
- Matching Pursuit (MP): là kỹ thuật phân tích tín hiệu âm thanh dựa trên việc tìm kiếm các hàm sóng gần đúng cho tín hiệu âm thanh, bằng cách sử dụng một tập hữu hạn các hàm sóng cơ bản và tối ưu hoá việc biểu diễn tín hiệu âm thanh bằng cách kết hợp các hàm sóng cơ bản đó.
- Spectral Flatness Measure (SFM): là kỹ thuật phân tích tín hiệu âm thanh bằng cách tính toán tỷ lệ giữa năng lượng trung bình của tín hiệu ở các tần số cao và thấp. Kỹ thuật SFM được sử dụng để phân tích tính chất của tín hiệu âm thanh theo tần số.
- Zero Crossing Rate (ZCR): là kỹ thuật phân tích tín hiệu âm thanh bằng cách đếm số lần tín hiệu âm thanh băng qua trục thời gian và tính toán tần số của các băng tần có số lần băng qua trục thời gian nhỏ nhất. Kỹ thuật ZCR được sử dụng để phân tích tính chất của tín hiệu âm thanh theo thời gian.

- Energy Entropy (EE): đo lường mức độ ngẫu nhiên của các mẫu tín hiệu âm thanh trong một khung thời gian nhất định. EE càng gần 1 thì tín hiệu càng ngẫu nhiên và ngược lại.

4. Kỹ thuật đồng bộ hóa tín hiệu

Kỹ thuật đồng bộ hóa tín hiệu âm thanh là một phương pháp phổ biến trong xử lý tín hiệu, được sử dụng để đồng bộ hóa tín hiệu giữa hai hay nhiều tín hiệu khác nhau. Điều này có thể được sử dụng để tìm ra mối liên hệ giữa các tín hiệu, hoặc để đồng bộ hóa các tín hiệu với nhau để phân tích dữ liệu.

Các kỹ thuật đồng bộ hóa tín hiệu này có thể được sử dụng độc lập hoặc kết hợp với các kỹ thuật khác để tối ưu hóa quá trình xử lý tín hiệu âm thanh.

- Cross Correlation Function (CCF) là một kỹ thuật đồng bộ hóa tín hiệu thường được sử dụng trong xử lý tín hiệu âm thanh. CCF được sử dụng để tìm sự tương đồng giữa hai tín hiệu âm thanh bằng cách tính toán độ tương quan giữa chúng. Kết quả của CCF là một hàm độ tương quan, thường được biểu diễn bằng đồ thị.
- Coherence Function (CF) là một kỹ thuật đồng bộ hóa tín hiệu khác nhưng tập trung vào sự tương quan giữa hai tín hiệu trên nhiều tần số khác nhau. CF đo lường độ tương quan giữa các tín hiệu ở mỗi tần số, từ đó ta có thể tìm ra các tần số nào giữa hai tín hiệu có độ tương quan cao nhất.
- Mutual Information Function (MIF) là một kỹ thuật đồng bộ hóa tín hiệu dựa trên thông tin. MIF tính toán lượng thông tin chung giữa hai tín hiệu, dựa trên sự phân bố của chúng. Nó đo lường độ tương quan không chỉ giữa các tín hiệu trên cùng một tần số, mà còn giữa các tín hiệu trên nhiều tần số khác nhau. Kết quả của MIF thường được biểu diễn bằng đồ thị, thể hiện độ tương quan giữa các tín hiệu theo tần số.

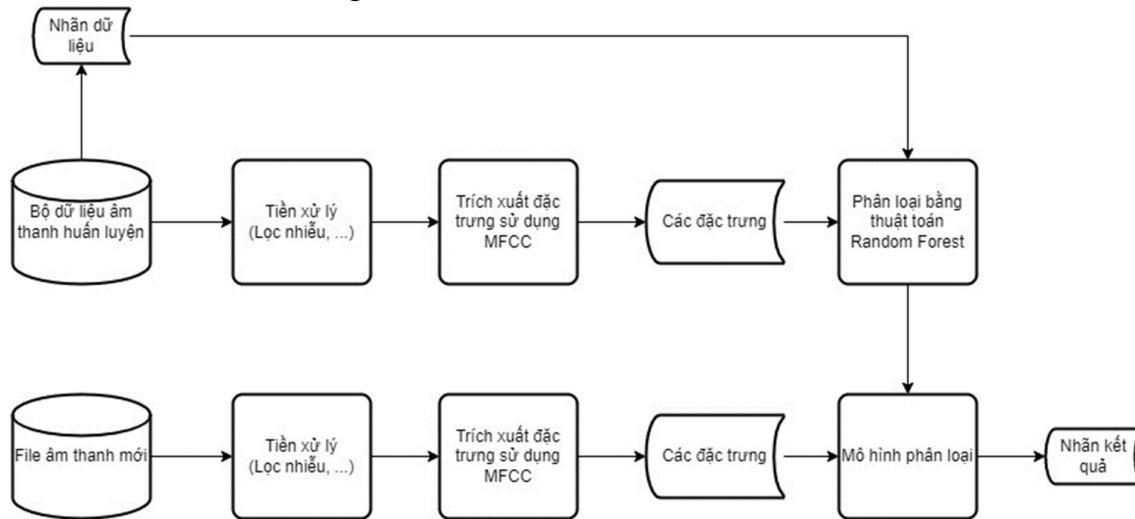
5. Kỹ thuật sử dụng xác suất

Kỹ thuật sử dụng phương pháp xác suất trong nhận dạng âm thanh là một phương pháp phân loại phổ biến được sử dụng trong machine learning. Đây là một phương pháp thống kê dựa trên xác suất để xác định xác suất thuộc về từng lớp phân loại.

- Maximum Likelihood Estimation (MLE) là một phương pháp xác suất để ước tính giá trị của các tham số trong phân phối xác suất của các lớp phân loại. Với MLE, chúng ta cần tính toán xác suất của mỗi lớp phân loại dựa trên dữ liệu huấn luyện và sau đó sử dụng các xác suất này để phân loại các tín hiệu âm thanh mới.
- Bayes: Đây là một phương pháp xác suất dựa trên định lý Bayes, trong đó ta tính toán xác suất của một lớp phân loại dựa trên các xác suất a priori của lớp và các đặc trưng của tín hiệu âm thanh đó.
- Gaussian Mixture Models (GMM): GMM là một phương pháp xác suất để mô hình hóa phân phối xác suất của từng lớp phân loại bằng cách sử dụng một hỗn hợp các phân phối Gaussian. Trong GMM, chúng ta tính toán xác suất của một tín hiệu âm thanh mới thuộc về từng lớp phân loại bằng cách sử dụng phân phối xác suất của các lớp phân loại và các đặc trưng của tín hiệu âm thanh.

III. Xây dựng hệ thống nhận dạng âm thanh công cụ/dụng cụ

1. Sơ đồ khối hệ thống



2. Quy trình thực hiện

- Bước 1: Thu thập dữ liệu: 10 loại công cụ, mỗi loại 10-17 file
- Bước 2: Tiền xử lý dữ liệu: loại bỏ nhiễu/tiếng ồn
- Bước 3: Trích xuất đặc trưng: sử dụng MFCC tách lấy các đặc trưng âm thanh
- Bước 4: Xây dựng mô hình phân loại: sử dụng thuật toán Random Forest
- Bước 5: Đưa vào 1 file âm thanh mới, thực hiện tiền xử lý dữ liệu và trích xuất đặc trưng file âm thanh đó.
- Bước 6: Xác định nhận kết quả: sử dụng các đặc trưng của file âm thanh mới và mô hình phân loại đã xây dựng.

3. Các thuộc tính âm thanh dùng để nhận dạng

- Phổ âm thanh: MFCC được tính toán dựa trên phổ âm thanh, được tính bằng cách biến đổi Fourier của tín hiệu âm thanh.
- Bảng thông: MFCC thường áp dụng một bộ lọc thông qua để giảm bớt các tín hiệu nhiễu, giữ lại các tín hiệu tần số cần thiết, bảng thông của bộ lọc sẽ ảnh hưởng đến đặc trưng được trích xuất từ MFCC.
- Độ cao thấp của âm thanh: MFCC có thể sử dụng các bộ lọc thông qua để giữ lại các tín hiệu tần số thấp và cao, vì vậy độ cao thấp của âm thanh cũng ảnh hưởng đến kết quả trích xuất của MFCC.

4. Kỹ thuật trích rút các thuộc tính âm thanh

Sử dụng thuật toán MFCC với các bước thực hiện:

- Bước 1: Tiền xử lý tín hiệu âm thanh: Tín hiệu âm thanh được cắt thành các khung (frame) có độ dài bằng nhau, thường là từ 20 đến 40 ms. Các khung này thường có chồng lên nhau để đảm bảo rằng không có thông tin âm thanh nào bị mất đi.
- Bước 2: Chuyển đổi Fourier: Các khung tín hiệu âm thanh được chuyển đổi từ miền thời gian sang miền tần số bằng cách sử dụng biến đổi Fourier.

- Bước 3: Tính toán công suất phổ: Công suất phổ của các băng thông được tính toán bằng cách bình phương giá trị của biến đổi Fourier.
- Bước 4: Chuyển đổi Mel: Các băng thông được chuyển đổi sang thang đo Mel, thay vì thang đo tần số thông thường. Thang đo Mel được sử dụng để tương thích với cách thức người ta nghe và tạo ra các băng thông có độ rộng khác nhau.
- Bước 5: Tính toán hệ số cepstral: Hệ số cepstral của các băng thông được tính toán bằng cách sử dụng biến đổi cepstral ngược. Các hệ số cepstral này được sử dụng để biểu diễn các đặc trưng của tín hiệu âm thanh.
- Bước 6: Lấy các hệ số đặc trưng: Các hệ số cepstral được lấy từ các băng thông Mel đầu tiên, thường là từ 2 đến 13 hệ số, để tạo thành các đặc trưng MFCC.

5. Lưu trữ các thuộc tính âm thanh

- Lưu trữ: gồm 2 file csv:
+ data.csv: lưu tên file và nhãn tương ứng

	File	label
0	mbn1.wav	máy bơm nước
1	mbn2.wav	máy bơm nước
2	mbn3.wav	máy bơm nước
3	mbn4.wav	máy bơm nước
4	mbn5.wav	máy bơm nước
...
132	fruit_juicer (5).mp3	máy ép trái cây
133	fruit_juicer (6).mp3	máy ép trái cây
134	fruit_juicer (7).mp3	máy ép trái cây
135	fruit_juicer (8).mp3	máy ép trái cây
136	fruit_juicer (9).mp3	máy ép trái cây

+ result.csv: lưu tên file và các đặc trưng trích xuất được từ file đó

1	File	mfcc_0	mfcc_1	mfcc_2	mfcc_3	mfcc_4	mfcc_5	mfcc_6	mfcc_7	mfcc_8	mfcc_9	mfcc_10	mfcc_11	mfcc_12
2	mbn1.wav	-268.443	41.45589	-40.6123	-17.3089	-31.9069	-17.8202	-13.4606	12.78661	10.94906	14.1955	-0.08314	3.914842	-9.04675
3	mbn2.wav	-157.457	101.9742	-42.5885	31.30961	-26.7378	-9.93208	-14.7117	8.302465	-20.1059	-13.3742	-5.51798	7.278675	0.465725
4	mbn3.wav	22.72515	55.8529	-27.9483	20.35818	-14.2075	2.928701	-25.8633	-3.05038	-15.1525	-3.81055	-9.21434	3.244201	-23.7741
5	mbn4.wav	-233.082	192.382	-19.4263	31.76979	36.67768	-7.73439	-16.7531	23.78727	-5.70057	16.71775	-0.26444	1.998877	-16.34
6	mbn5.wav	-36.1867	117.9082	-16.1412	15.00468	-5.85617	3.264681	-15.8164	0.979555	-17.3629	11.2409	-7.88404	9.728157	-15.4155
7	mbn6.wav	-113.577	136.3532	-19.3504	17.73816	-5.46864	12.44634	-16.7498	5.402029	-14.356	4.853527	-5.03356	3.447941	-4.71077
8	mbn7.wav	-107.309	48.0686	-60.5302	-21.2652	-44.5527	-21.4401	-16.0515	20.7137	12.72545	18.91189	-2.13689	6.336563	-13.7543
9	mbn8.wav	-100.822	139.0729	-18.9824	16.41153	-7.36613	10.52822	-13.8579	6.707379	-14.6547	5.929122	-4.88111	5.334737	-7.07179
10	mc1.wav	-22.5014	47.49359	-18.6206	26.63965	-22.3627	-12.966	-23.3976	15.15613	-5.45535	14.47306	-5.46593	13.9801	-7.75958
11	mc10.wav	-1.34692	46.5529	-50.729	37.08826	-29.7348	6.616201	-23.591	13.6999	-11.0696	6.218587	-2.68239	-3.90128	-14.6548
12	mc11.wav	18.43271	20.90355	-22.364	36.68855	-39.3016	17.48967	-27.1357	-6.98288	-5.4407	-9.96986	-15.9381	6.645403	-22.0081
13	mc12.wav	-16.4344	62.52892	-17.3406	71.27704	-44.4789	20.5988	-25.6707	-2.41419	-0.71459	-17.1102	-32.2471	1.945415	-26.2852
14	mc2.wav	9.435771	37.36424	-21.2674	24.46916	-25.2717	-7.77905	-21.3554	13.6602	-8.61692	4.750053	-12.6185	8.132424	-10.2545
15	mc3.wav	-36.5245	25.6064	-39.5841	23.32405	-16.9387	-1.99793	-21.4089	13.06431	-10.6964	2.668909	-15.4478	7.918291	-13.6103
16	mc4.wav	-73.4815	44.13964	-32.287	30.6364	-11.2614	0.217207	-18.7669	15.67368	-8.07376	4.917903	-14.8488	3.959556	-14.1877
17	mc5.wav	-43.5633	32.63647	-65.3399	39.44424	-20.2449	6.42043	-33.8944	21.91284	-5.97919	4.056254	-0.85664	-9.78797	-14.5705
18	mc6.wav	-43.5635	42.52202	-46.1605	29.95967	-34.6769	1.366336	-24.6509	9.122868	-16.7011	3.382825	-0.55025	-1.30091	-10.4854
19	mc7.wav	-86.1972	54.18678	-46.4455	32.57837	-30.6827	7.129736	-29.4477	4.935384	-12.8018	-2.0972	-7.10565	-1.58845	-9.33796
20	mc8.wav	-38.8419	107.1829	-42.8794	52.07447	-8.49111	23.78289	-37.2583	0.838307	0.815511	-3.63224	-13.0592	-7.69228	-20.129
21	mc9.wav	-16.2785	75.5047	-69.0396	11.36662	-20.9779	7.59485	-29.653	9.429579	-4.60369	-3.01209	-4.02395	-1.56081	-9.2108
22	lawn_tract	-69.0681	80.70787	0.206327	36.59134	3.029694	13.82882	-29.2545	12.66514	-10.0104	7.975102	-19.139	2.92819	-15.7094
23	lawn_tract	-82.5312	91.37546	6.175099	33.57434	-5.20876	19.34633	-26.9835	14.37229	-8.62243	6.825354	-16.5759	2.628215	-13.6273
24	lawn_tract	-70.0766	63.25048	-13.9111	43.70974	11.77545	24.91438	-24.9292	12.59903	-13.2333	7.862697	-20.0259	4.45444	-12.2589
25	lawn_tract	-82.8008	81.09065	-7.47005	41.39213	8.426119	22.62912	-25.3489	11.05293	-14.8277	9.087428	-19.7794	3.353025	-14.2082
26	lawn_tract	-87.504	62.12037	-10.1439	43.63283	12.62176	32.80283	-20.5811	17.03018	-13.0647	9.060861	-20.2353	4.195116	-12.9814
27	lawn_tract	-75.1267	80.29114	-6.22806	40.08039	8.276268	20.18104	-26.5569	10.19692	-14.0121	9.272512	-19.561	3.090148	-14.9748

112	leaf_blow	-6.24037	85.15421	-48.2892	18.69583	-23.6264	7.052747	-18.2021	-3.19964	-18.1424	6.582866	-1.70351	3.53153	-29.4511
113	leaf_blow	-3.01804	82.94132	-48.6789	18.08607	-24.479	6.846465	-19.9232	-4.04539	-21.0457	6.95527	-4.15772	3.064242	-28.0973
114	leaf_blow	-4.48077	84.47226	-48.0347	16.9971	-23.1204	6.360878	-19.7412	-3.32722	-19.1434	6.345545	-2.11739	2.99159	-28.9876
115	leaf_blow	-6.20717	86.16838	-47.6483	17.71091	-23.9966	7.611083	-17.9772	-3.53502	-17.5386	6.708882	-2.42085	5.236147	-30.0135
116	leaf_blow	-5.60765	85.35505	-47.6807	18.35918	-23.2839	7.10338	-18.123	-3.36614	-17.7882	6.972552	-2.56456	4.640732	-29.9715
117	leaf_blow	-5.12939	84.57125	-49.1496	18.28687	-24.0931	6.219942	-18.3538	-5.2001	-20.1639	6.305224	-3.55414	3.329895	-28.7431
118	leaf_blow	-6.56393	85.1396	-48.3587	18.75922	-23.9897	6.847038	-18.1066	-3.56122	-18.1472	6.288479	-1.82122	3.533663	-29.3479
119	leaf_blow	-5.58195	85.34838	-47.6826	18.35944	-23.1683	7.084496	-18.2222	-3.47911	-17.806	6.954162	-2.20509	4.35324	-30.0533
120	leaf_blow	-5.8243	85.27281	-47.7611	18.12676	-23.1254	6.880081	-18.1625	-2.88917	-18.1399	6.566138	-1.51193	3.525257	-29.6314
121	leaf_blow	-6.30082	85.09093	-48.5754	18.72963	-24.1331	6.616259	-18.0195	-3.86612	-18.2892	6.117168	-2.00262	3.256601	-29.3478
122	fruit_juice	-70.8389	24.05307	-31.0741	35.67492	-12.0146	9.576018	-11.6094	3.031029	-8.26789	-0.31818	-25.9994	1.262801	-25.1773
123	fruit_juice	-71.4532	23.97479	-30.8645	35.79229	-12.1079	9.888994	-11.7023	2.921487	-8.75879	-0.81683	-26.6054	1.512902	-24.8437
124	fruit_juice	-71.547	23.76745	-31.1717	35.75627	-12.2688	9.76539	-11.6714	2.850952	-8.84051	-0.81457	-26.393	1.509083	-24.7844
125	fruit_juice	-71.19	23.83441	-30.9796	35.54881	-12.1983	9.794655	-11.3312	3.063049	-8.82707	-0.69606	-26.1843	1.615177	-24.625
126	fruit_juice	-70.8873	24.05084	-31.0843	35.8021	-11.9566	9.477221	-11.745	3.000285	-8.34439	-0.34206	-25.9767	1.184942	-25.1221
127	fruit_juice	-70.9415	24.0778	-31.176	35.72357	-12.051	9.700883	-11.7064	2.879585	-8.27629	-0.36261	-26.0141	1.297956	-25.1495
128	fruit_juice	-70.8944	24.68893	-30.4682	36.13701	-11.5827	9.758594	-11.93	2.530912	-7.98078	-0.38467	-26.3968	0.895222	-25.0359
129	fruit_juice	-70.8579	24.62701	-30.4553	36.04573	-11.6471	9.706329	-11.8868	2.530456	-8.10434	-0.44572	-26.5241	0.866799	-25.1111
130	fruit_juice	-71.1646	23.58231	-31.0957	35.56454	-12.4616	9.746478	-11.6049	3.059925	-8.77291	-0.64059	-26.0543	1.368511	-24.6858
131	fruit_juice	-70.9826	24.71735	-30.5647	36.1072	-11.8428	9.633986	-12.0111	2.54268	-8.03999	-0.45984	-26.7706	1.1554	-25.1623
132	fruit_juice	-71.312	24.22572	-30.5726	35.72696	-12.1595	9.816149	-11.9888	2.536802	-8.76228	-0.64744	-26.7963	1.27066	-24.8436
133	fruit_juice	-71.32	23.65343	-31.0467	35.51161	-12.5881	9.622101	-11.7287	3.121811	-8.81252	-0.62864	-26.2678	1.371097	-24.5824
134	fruit_juice	-71.2173	23.67365	-31.1152	35.60107	-12.3998	9.615681	-11.9684	2.800702	-8.86991	-0.24382	-26.2149	1.196714	-24.5665
135	fruit_juice	-71.0338	24.0222	-31.0868	35.63085	-12.1916	9.513719	-11.6325	3.042179	-8.58662	-0.4176	-26.2707	1.14932	-24.8016
136	fruit_juice	-71.1457	23.8263	-31.1272	35.63924	-12.2791	9.310399	-11.8895	2.749009	-8.85632	-0.33988	-26.1956	1.154913	-24.7164
137	fruit_juice	-71.1964	23.48615	-31.0011	35.43718	-12.2963	9.51227	-11.7537	2.864253	-8.68676	-0.64581	-26.0196	1.035065	-24.5447
138	fruit_juice	-70.8239	24.73985	-30.4342	36.08114	-11.6885	9.736908	-11.8181	2.640625	-7.9434	-0.43551	-26.4172	0.941006	-25.0137

6. Nhận dạng âm thanh bằng các thuộc tính âm thanh

Sử dụng thuật toán Random Forest: Random Forest là một thuật toán học máy có tính toán mạnh mẽ và được sử dụng rộng rãi cho các bài toán phân loại, hồi quy và nhận dạng. Nó là một phương pháp thuộc họ các thuật toán Ensemble Learning, trong đó nhiều cây quyết định (Decision Trees) được kết hợp lại để tạo thành một mô hình mạnh hơn.

Các bước thực hiện thuật toán:

- Bước 1: Tạo ra các cây quyết định (Decision Trees): Random Forest sử dụng nhiều cây quyết định trong quá trình huấn luyện. Mỗi cây quyết định được huấn luyện trên một tập con của dữ liệu, được lấy mẫu ngẫu nhiên từ tập huấn luyện. Sự ngẫu nhiên này giúp tạo ra độ đa dạng giữa các cây quyết định.
- Bước 2: Chọn các thuộc tính ngẫu nhiên: Khi xây dựng mỗi cây quyết định, chỉ một số thuộc tính ngẫu nhiên được chọn để đưa vào cây. Quá trình này giúp giảm đa dạng hóa các cây và tránh overfitting.
- Bước 3: Huấn luyện cây quyết định: Mỗi cây quyết định được huấn luyện trên tập con dữ liệu được lấy mẫu. Quá trình huấn luyện cây quyết định bao gồm việc chia dữ liệu dựa trên các thuộc tính để tối ưu hóa việc phân loại các mẫu dữ liệu.
- Bước 4: Tính toán đa số phiếu bầu: Khi có một mẫu dữ liệu mới cần được phân loại, mỗi cây quyết định sẽ đưa ra một dự đoán. Dựa trên các dự đoán này, mô hình sẽ tính toán đa số phiếu bầu để quyết định kết quả phân loại cuối cùng.
- Bước 5: Đánh giá và điều chỉnh: Mô hình Random Forest có thể được đánh giá bằng cách sử dụng các phương pháp như cross-validation hoặc phân chia tập kiểm tra. Nếu cần thiết, bạn có thể điều chỉnh các siêu tham số của Random Forest như số lượng cây, độ sâu cây, số lượng thuộc tính được chọn ngẫu nhiên để tối ưu hiệu suất của mô hình.

Sau khi đã sử dụng MFCC để trích xuất được đặc trưng của bộ dữ liệu âm thanh, ta tiến hành sử dụng thuật toán Random Forest để xây dựng Model rồi nhận dạng file âm thanh.

IV. Kết quả xây dựng hệ thống

1. Demo hệ thống

- Khai báo thư viện

```
import librosa
import numpy as np
from scipy.signal import butter, freqz, findfreqs, filtfilt
from scipy import signal
import os
import pandas as pd
```

- Tiền xử lý dữ liệu âm thanh

```
# Tiền xử lý dữ liệu âm thanh
fs = 44100
lowcut = 20
highcut = 20000
Wn = (2 * lowcut / fs, 2 * highcut / fs)

def preprocess_audio(audio_file):
    # Chuyển đổi tần số lấy mẫu
    signal, sr = librosa.load(audio_file, sr=22050)
    # Cắt 1 giây đầu của tệp âm thanh
    signal = signal[sr:]
    # Lọc tạp âm bằng bộ lọc số học
    b, a = butter(2, Wn, btype='band')
    signal_filtered = filtfilt(b, a, signal)
    return signal_filtered, sr
```

- Hàm trích xuất đặc trưng MFCC

```
# Trích xuất đặc trưng MFCC
def extract_features(signal, sr):
    # Sử dụng khung cửa sổ trượt độ dài 512 mẫu
    hop_length = 512
    # Trích xuất đặc trưng MFCC với số cấp MFCC là 20
    mfccs = librosa.feature.mfcc(y=signal, sr=sr, n_mfcc=13, hop_length=hop_length)
    # Trung bình các giá trị MFCC trên toàn bộ tệp âm thanh
    features = np.mean(mfccs.T, axis=0)
    return features
```

- Hàm xây dựng mô hình Random Forest

```
from sklearn.ensemble import RandomForestClassifier
def train_model(X_train, y_train):
    rf_model = RandomForestClassifier(n_estimators=100)
    rf_model.fit(X_train, y_train)
    return rf_model
```


- Hàm dự đoán nhãn âm thanh mới

```
def predict_audio(audio_file, model):
    signal, sr = preprocess_audio(audio_file)
    features = extract_features(signal, sr)
    prediction = model.predict([features])
    return prediction[0]
```

- Đọc dữ liệu từ file CSV

```
# Đọc dữ liệu từ file CSV
data = pd.read_csv('data.csv')
print(data)
```

	File	label
0	mbn1.wav	máy bơm nước
1	mbn2.wav	máy bơm nước
2	mbn3.wav	máy bơm nước
3	mbn4.wav	máy bơm nước
4	mbn5.wav	máy bơm nước
..
132	fruit_juicer (5).mp3	máy ép trái cây (fruit juicer)
133	fruit_juicer (6).mp3	máy ép trái cây (fruit juicer)
134	fruit_juicer (7).mp3	máy ép trái cây (fruit juicer)
135	fruit_juicer (8).mp3	máy ép trái cây (fruit juicer)
136	fruit_juicer (9).mp3	máy ép trái cây (fruit juicer)

[137 rows x 2 columns]

```
# Khởi tạo dictionary để lưu trữ đường dẫn đến các tệp âm thanh
audio_files = {}
```

```
# Lặp qua từng hàng trong dữ liệu
for index, row in data.iterrows():
    # Lấy đường dẫn và nhãn từ các cột tương ứng
    file_path = row['File']
    label = row['label']

    # Thêm đường dẫn và nhãn vào dictionary
    audio_files[file_path] = label
print(audio_files)
```

```
{'mbn1.wav': 'máy bơm nước', 'mbn2.wav': 'máy bơm nước', 'mbn3.wav': 'máy bơm nước', 'mbn4.
máy bơm nước', 'mbn6.wav': 'máy bơm nước', 'mbn7.wav': 'máy bơm nước', 'mbn8.wav': 'máy bơm
av': 'máy cưa', 'mc11.wav': 'máy cưa', 'mc12.wav': 'máy cưa', 'mc2.wav': 'máy cưa', 'mc3.wa
'mc5.wav': 'máy cưa', 'mc6.wav': 'máy cưa', 'mc7.wav': 'máy cưa', 'mc8.wav': 'máy cưa', 'mc
(1).mp3': 'máy cắt cỏ (lawn tractor)', 'lawn_tractor (10).mp3': 'máy cắt cỏ (lawn tractor)'}
```

- Tách dữ liệu thành 2 tập: features_list là các thuộc tính huấn luyện, labels_list là các nhãn huấn luyện tương ứng.

```
features_list = []
labels_list = []
for file, label in audio_files.items():
    file_path = label+'/'+file
    signal, sr = preprocess_audio(file_path)
    features = extract_features(signal, sr)
    features_list.append(features)
    labels_list.append(label)
```

- Lưu các đặc trưng trích xuất được vào 1 file CSV

```
df = pd.DataFrame({'File': audio_files.keys(), 'mfcc': features_list})
mfcc_df = pd.DataFrame(df['mfcc'].tolist(), columns=[f'mfcc_{i}'.format(i) for i in range(13)])
result_df = pd.concat([df['File'], mfcc_df], axis=1)
print(result_df)

if os.path.exists('result.csv'):
    os.remove('result.csv')

result_df.to_csv('result.csv', index=False)
```

	File	mfcc_0	mfcc_1	mfcc_2	mfcc_3 \
0	mbn1.wav	-268.443020	41.455886	-40.612264	-17.308881
1	mbn2.wav	-157.457301	101.974231	-42.588496	31.309613
2	mbn3.wav	22.725151	55.852903	-27.948332	20.358179
3	mbn4.wav	-233.081943	192.381976	-19.426254	31.769795
4	mbn5.wav	-36.186673	117.908198	-16.141211	15.004685
..
132	fruit_juicer (5).mp3	-71.217267	23.673646	-31.115233	35.601067
133	fruit_juicer (6).mp3	-71.033822	24.022198	-31.086751	35.630850
134	fruit_juicer (7).mp3	-71.145664	23.826301	-31.127248	35.639239
135	fruit_juicer (8).mp3	-71.196352	23.486149	-31.001125	35.437178
136	fruit_juicer (9).mp3	-70.823922	24.739848	-30.434238	36.081140

	mfcc_4	mfcc_5	mfcc_6	mfcc_7	mfcc_8	mfcc_9 \
0	-31.906897	-17.820238	-13.460588	12.786608	10.949058	14.195497
1	-26.737790	-9.932082	-14.711695	8.302465	-20.105950	-13.374212
2	-14.207521	2.928701	-25.863263	-3.050384	-15.152455	-3.810549
3	36.677684	-7.734391	-16.753056	23.787275	-5.700570	16.717753
4	-5.856167	3.264681	-15.816407	0.979555	-17.362912	11.240901
..
132	-12.399774	9.615681	-11.968393	2.800702	-8.869912	-0.243822
133	-12.191555	9.513719	-11.632528	3.042179	-8.586624	-0.417597
134	-12.279091	9.310399	-11.889496	2.749009	-8.856320	-0.339884
135	-12.296336	9.512270	-11.753697	2.864253	-8.686758	-0.645805
136	-11.688465	9.736908	-11.818146	2.640625	-7.943396	-0.435513

- Huấn luyện mô hình Random Forest

```
# Huấn luyện mô hình
rf_model = train_model(features_list, labels_list)
```

- Dự đoán nhãn cho file âm thanh mới

```
import ipywidgets as widgets
from IPython.display import display
```

```
uploader = widgets.FileUpload(
    accept='*.csv',
    multiple=False
)
```

```
def handle_upload(change):
    uploaded_file = uploader.value
    if uploaded_file:
        file_name = next(iter(uploaded_file))
        file_content = uploaded_file[file_name]['content']

        current_directory = os.getcwd()
        file_path = os.path.join(current_directory, file_name)
        with open(file_path, 'wb') as file:
            file.write(file_content)

        prediction = predict_audio(file_path, rf_model)

        print(f'Phân loại của tệp âm thanh {file_path} là: {prediction}')

uploader.observe(handle_upload, names='value')
display(uploader)
```

 Upload (6)

Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\leaf_blower (7).mp3 là: máy thổi lá (leaf blower)
 Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\vacuum_cleaner (9).mp3 là: máy hút bụi (vacuum cleaner)
 Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\mbn5.wav là: máy bơm nước
 Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\lawn_tractor (3).mp3 là: máy cắt cỏ (lawn tractor)
 Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\generator (6).mp3 là: máy phát điện (generator)
 Phân loại của tệp âm thanh C:\Users\84903\Documents\HCSLDPT-nhom10\fruit_juicer (5).mp3 là: máy ép trái cây (fruit juicer)

2. Đánh giá kết quả

- Tổng quát: Kết quả dự đoán chưa thật sự quá tốt, rơi vào khoảng 70%.
- Cụ thể :
 - Vẫn còn nhận dạng sai giữa các dụng cụ có âm thanh như là giữa máy máy thổi lá, máy phát điện, máy cắt cỏ.
 - Máy cưa, máy sấy tóc và máy bơm nước có tỷ lệ nhận dạng đúng cao hơn.
- Nguyên nhân:
 - Nhận dạng còn sai sót do số lượng dữ liệu dùng để đào tạo mô hình chưa phải là quá lớn.
 - Trích xuất đặc trưng cho mỗi file vẫn còn hạn chế.
 - Thuật toán sử dụng (Random forest) vẫn chưa phải tối ưu cho bài toán nhận dạng âm thanh.