



Team 14

GVHD: TS. NINH KHÁNH DUY

TEAM 14

**BÁO CÁO THỰC HÀNH
XỬ LÍ TÍN HIỆU SỐ**





Team 14

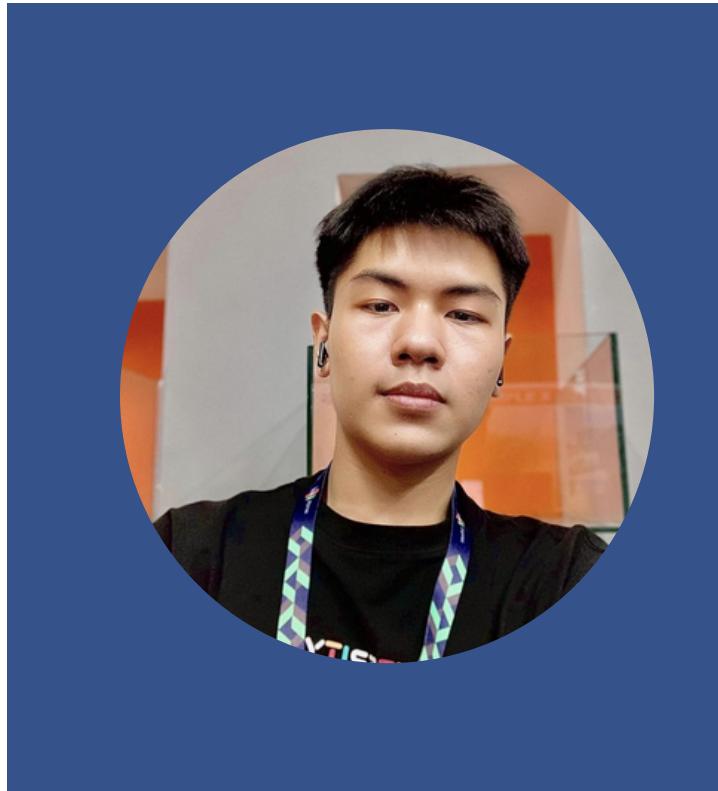
OUR BEST TEAM



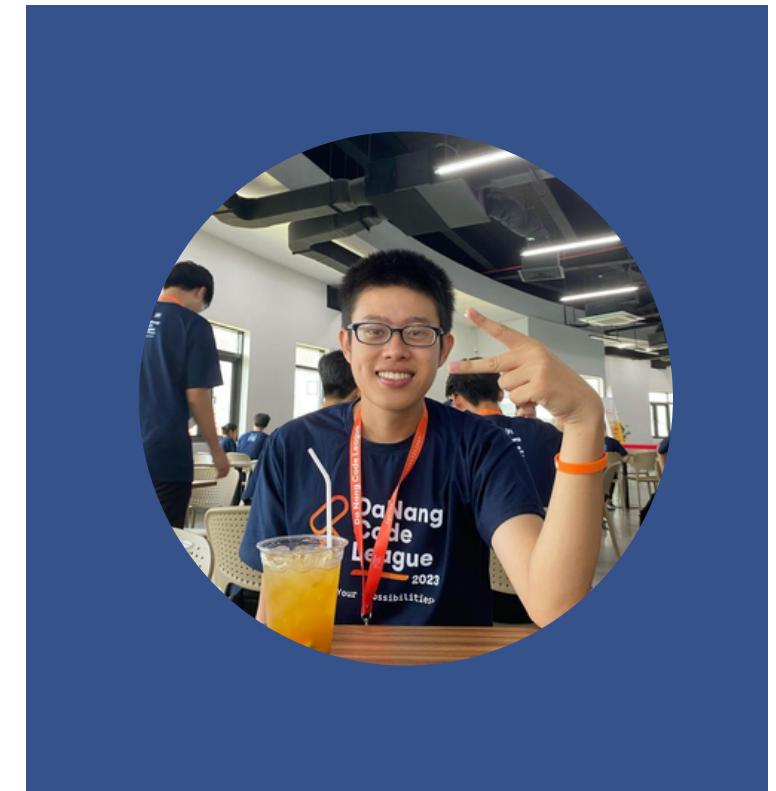
LÊ ANH TUẤN
BÀI 1 - SPECTROGRAM



PHẠM NGUYỄN ANH PHÁT
BÀI 2 - THRESHOLD



PHẠM GIA HÙNG
BÀI 2 - FFT



TRẦN ĐỨC TRÍ
BÀI 3 - MFCC, KMEAN



NỘI DUNG

Bài 1

Phân tích đặc trưng phổ các nguyên âm của nhiều người nói

Bài 2

**Nhận dạng nguyên âm không phụ thuộc người nói dùng
đặc trưng phổ FFT**

Bài 3

**Nhận dạng nguyên âm không phụ thuộc người nói dùng
đặc trưng phổ MFCC và phương pháp K-Mean**

Kết luận

Nhận xét chung



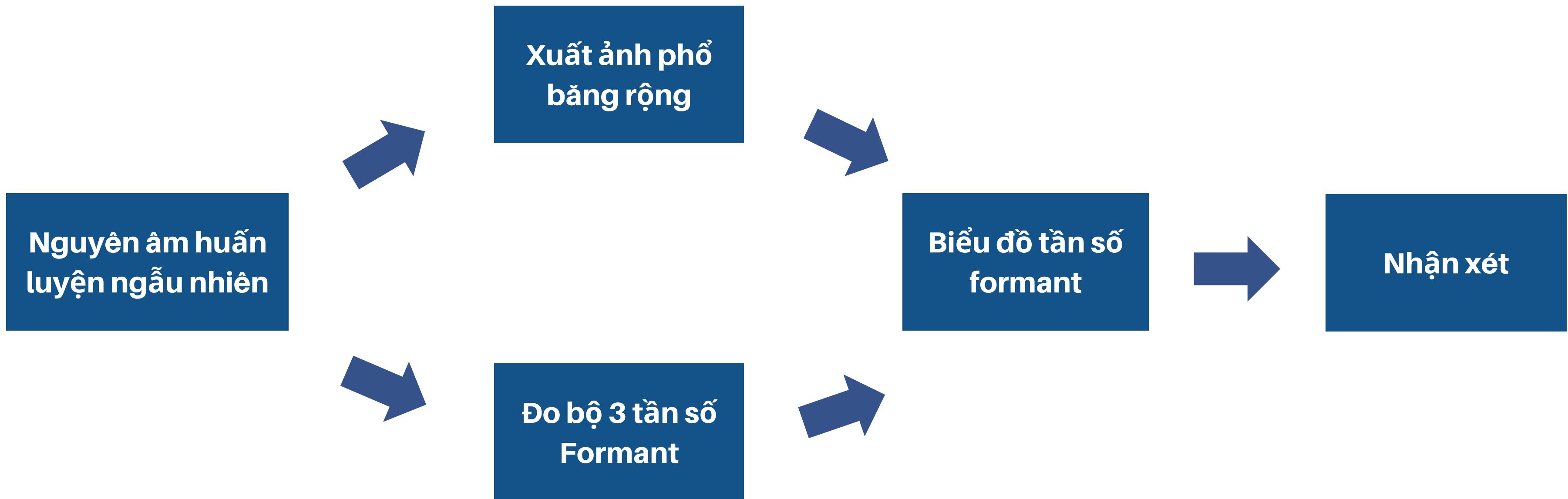
BÀI 1

PHÂN TÍCH ĐẶC TRƯNG PHỔ CÁC NGUYÊN ÂM CỦA NHIỀU NGƯỜI NÓI



Team 14

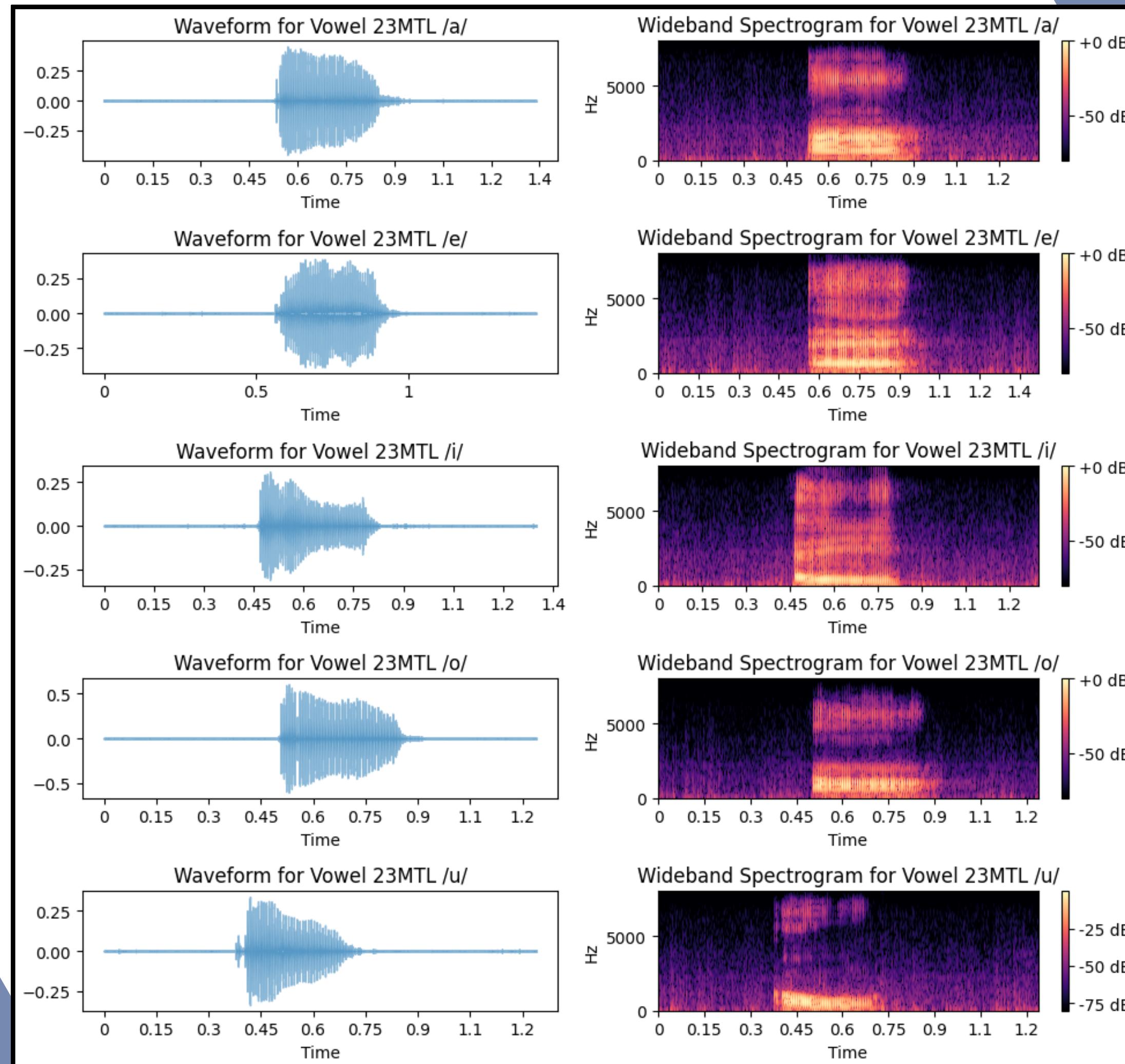
Quá trình xử lý





Team 14

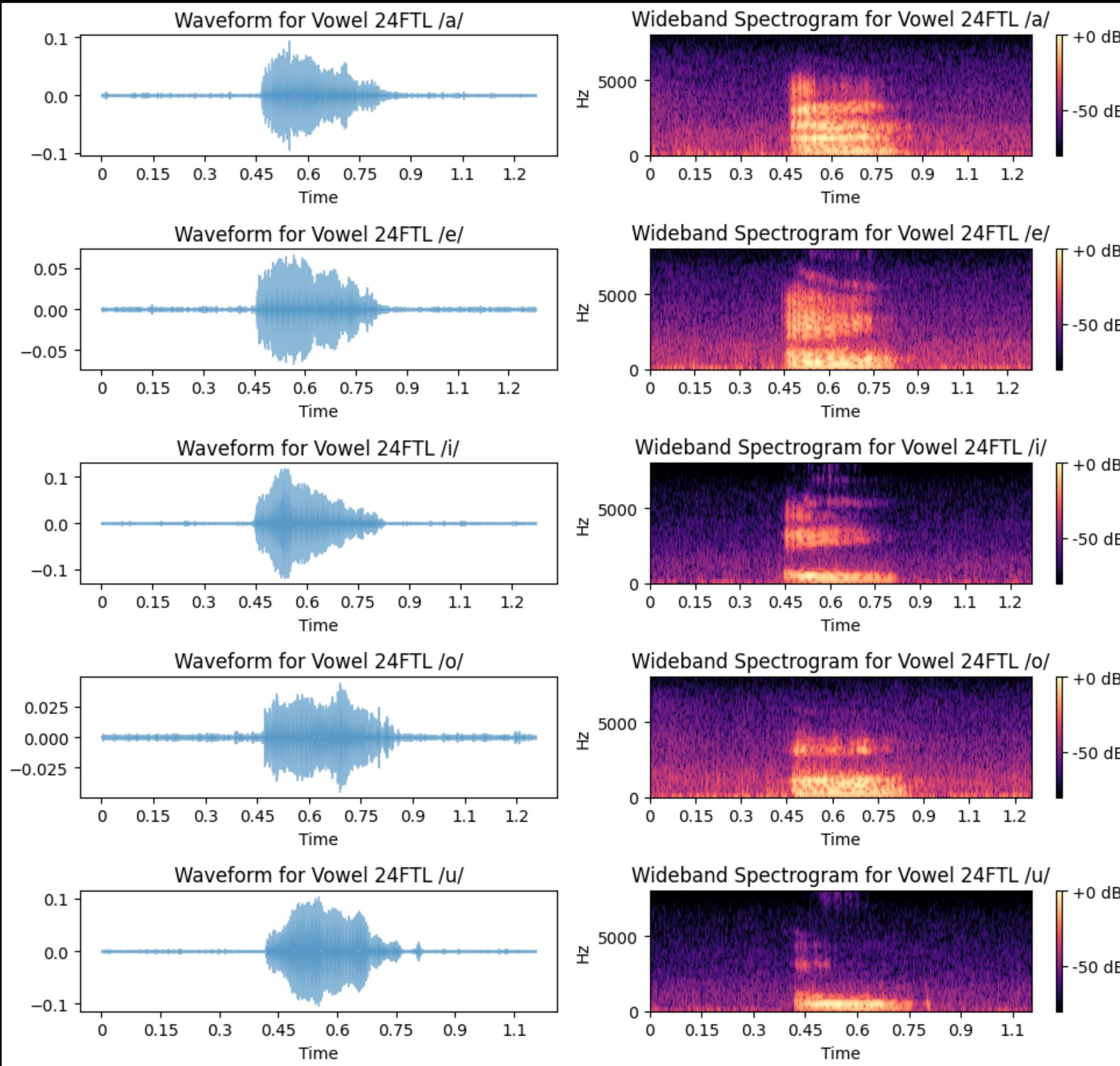
Ảnh phô 23MTL





Team 14

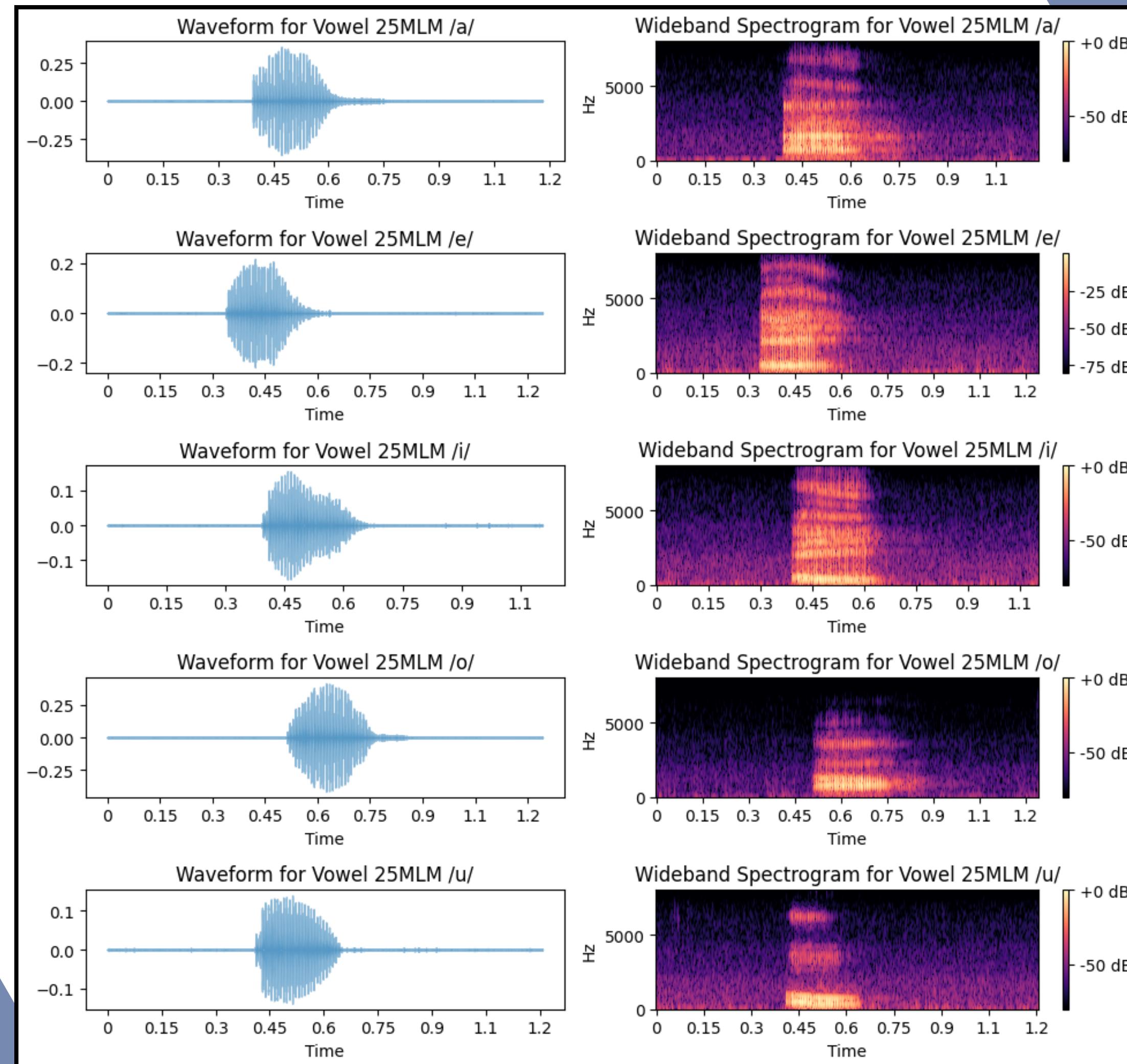
Ảnh phô 24FTL





Team 14

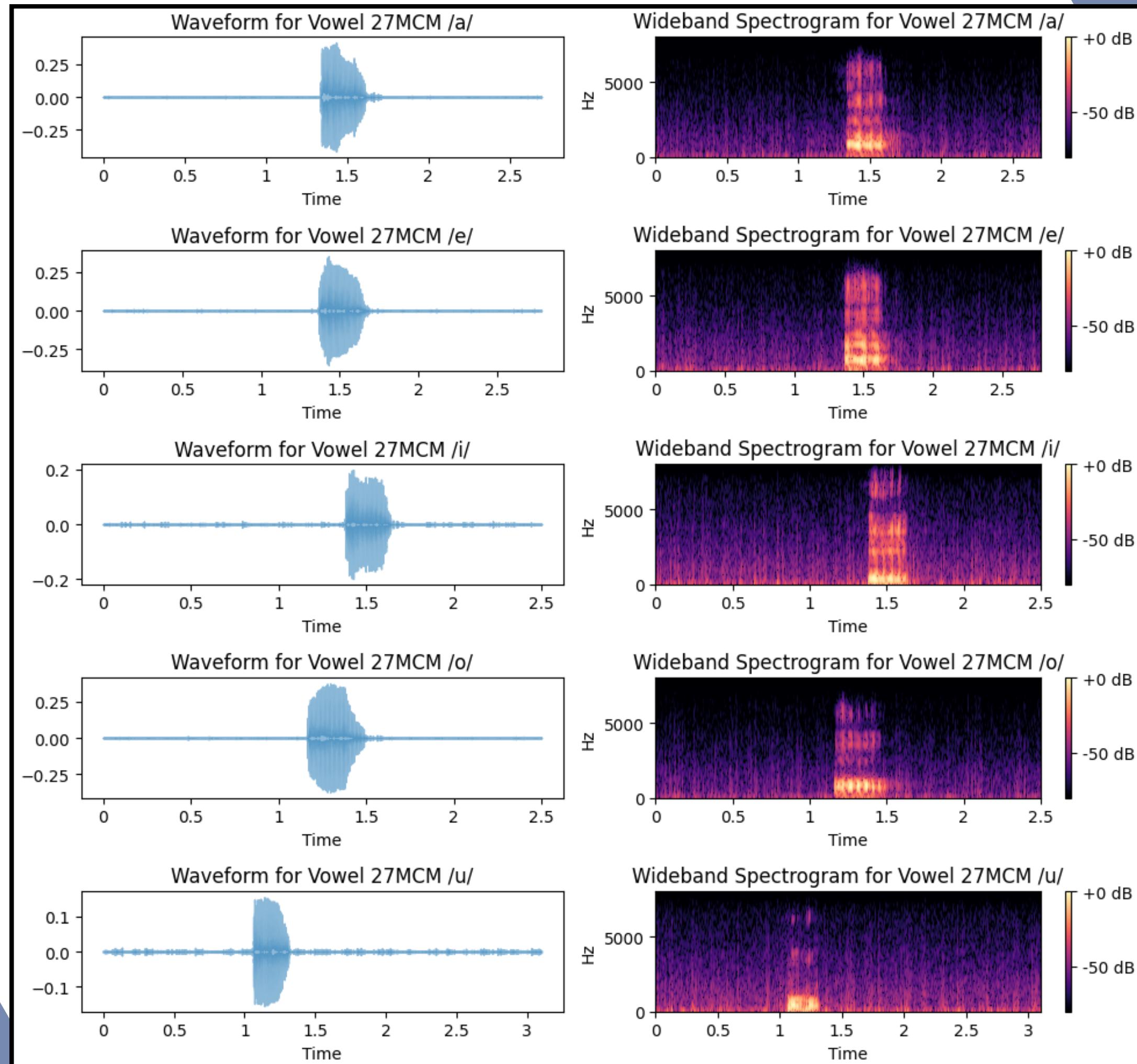
Ảnh phô 25MLM





Team 14

Ảnh phô 27MCM





CODE TRÍCH XUẤT ẢNH PHỔ RỘNG CHO TỪNG NGƯỜI

```
// paste your code here
def plot_vowel_waves_and_wideband_spectrograms(folder_name, speaker):
    """Plot the waveform and wideband spectrogram of the audio signal."""
    vowels = ['a', 'e', 'i', 'o', 'u']
    file_paths = {vowel: os.path.join(folder_name, f"{speaker}/{vowel}.wav") for vowel in vowels}
    n_rows = len(vowels)

    plt.figure(figsize=(10, 2 * n_rows))
    title = os.path.basename(folder_name)
    plt.suptitle(title, fontsize=16)

    for idx, (vowel, path) in enumerate(file_paths.items(), 1):
        audio, sr = load_audio(path)
        S_dB, hop_length = calculate_wideband_spectrogram(audio, sr)

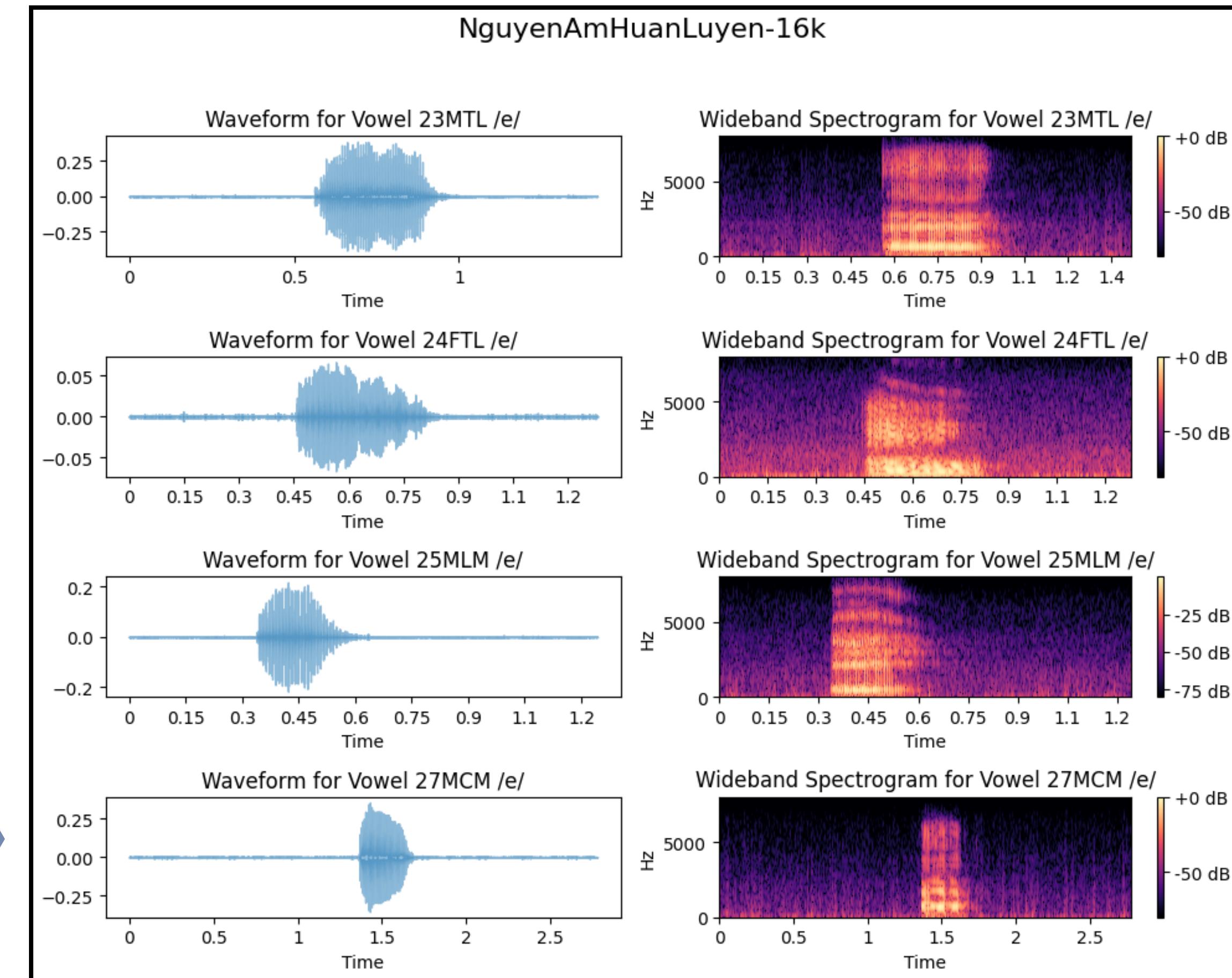
        plot_waveform(audio, sr, speaker, vowel, idx, n_rows)
        plot_wideband_spectrogram(S_dB, sr, hop_length, speaker, vowel, idx, n_rows)

    plt.tight_layout(rect=[0, 0, 1, 0.95]) # Adjust the layout to make room for the suptitle
    plt.show()
```



Team 14

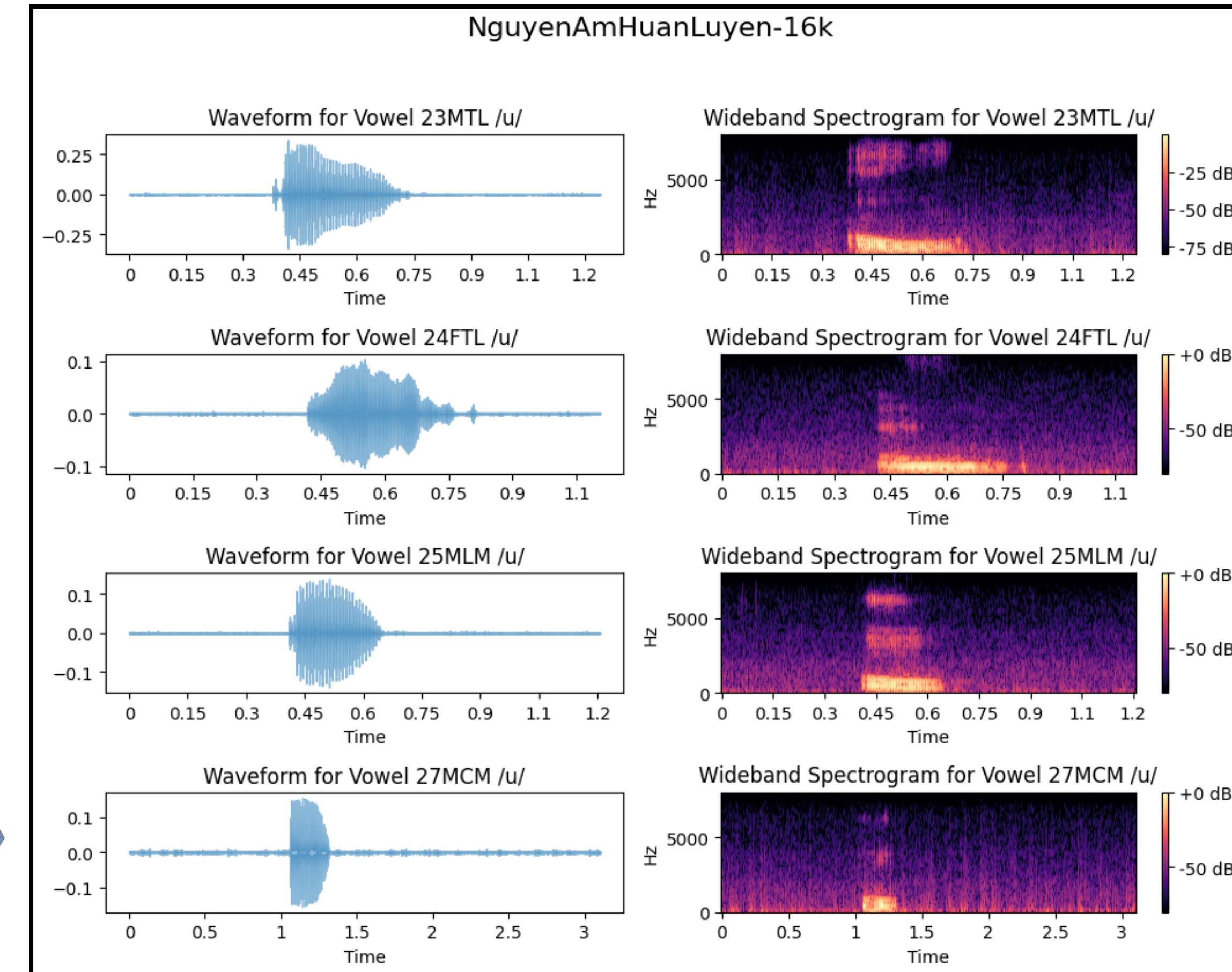
Ảnh phô nguyên âm */e/*





Team 14

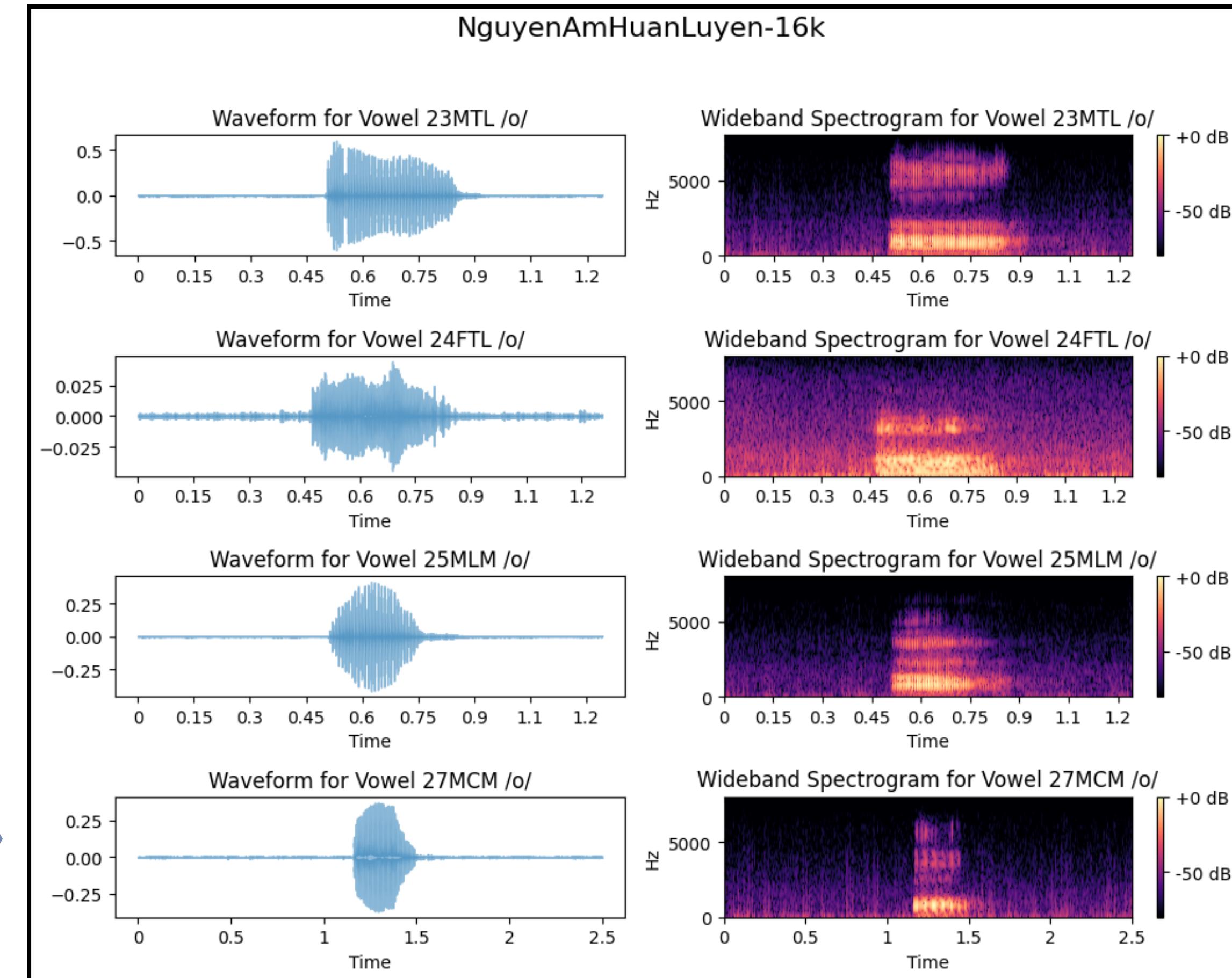
Ảnh phổ nguyên âm **/u/**





Team 14

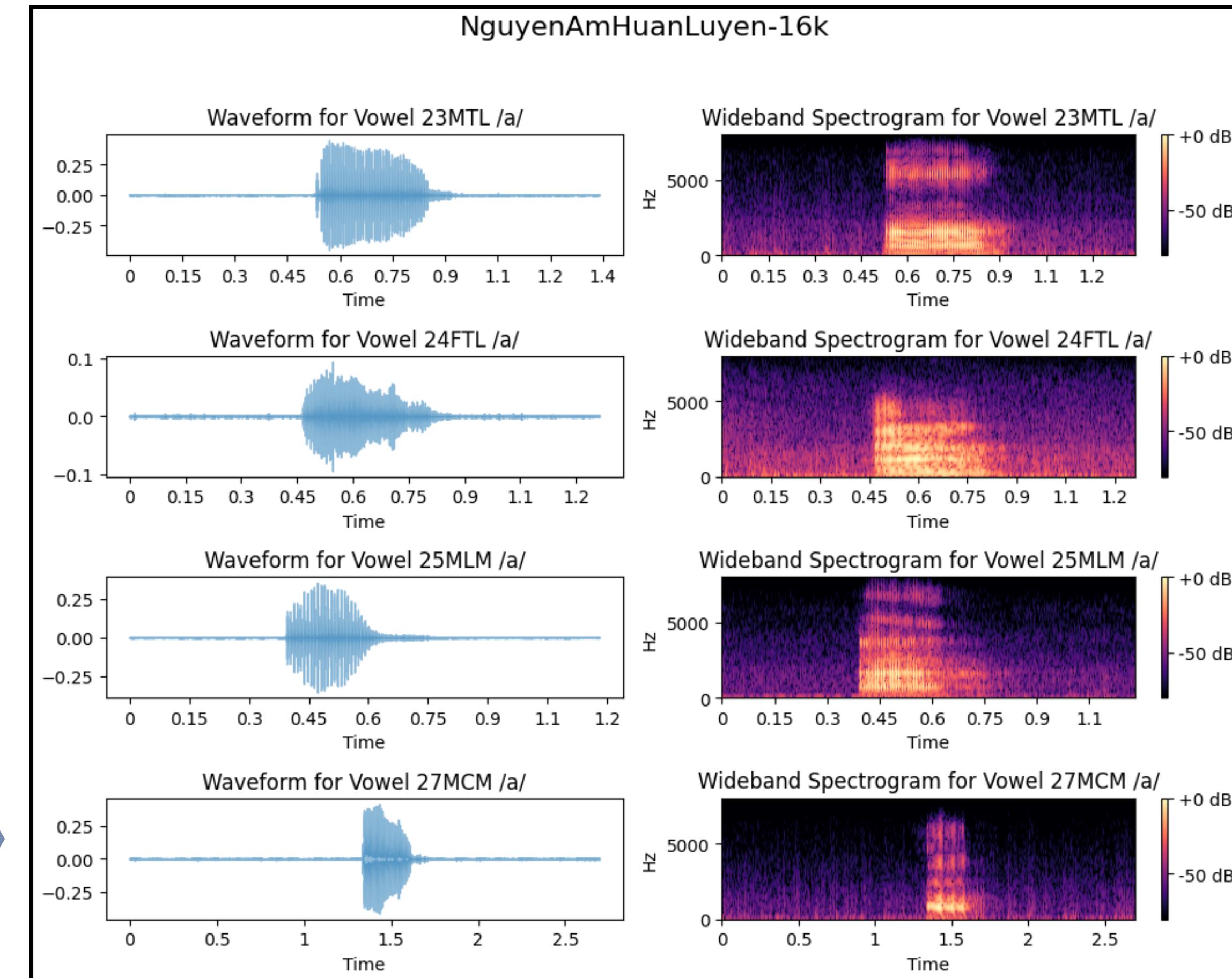
Ảnh phổ nguyên âm */o/*





Team 14

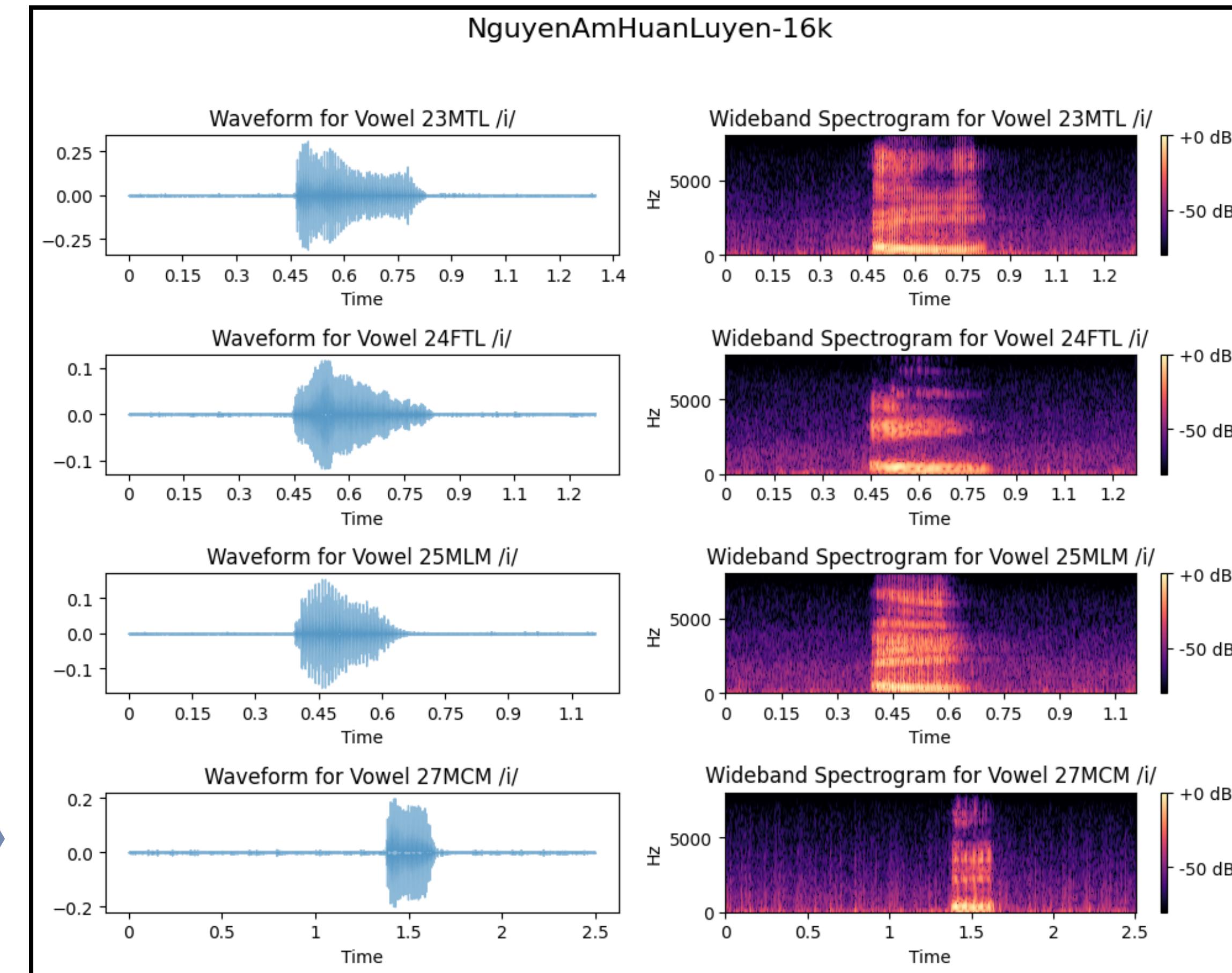
Ảnh phổ nguyên âm **/a/**





Team 14

Ảnh phổ nguyên âm */i/*





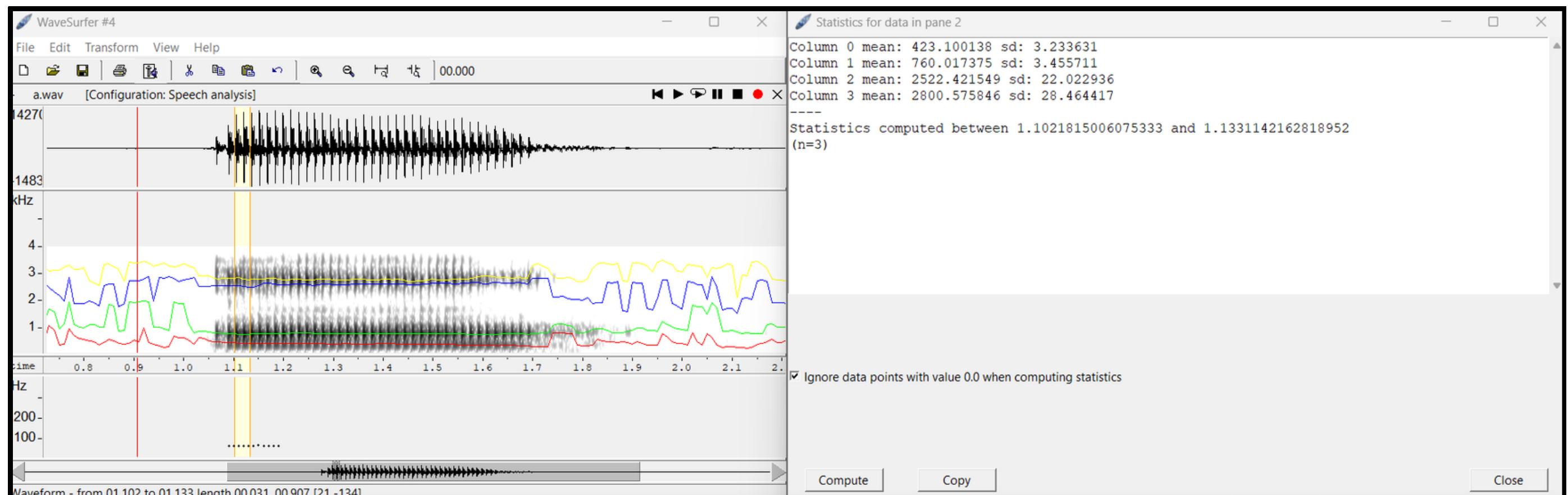
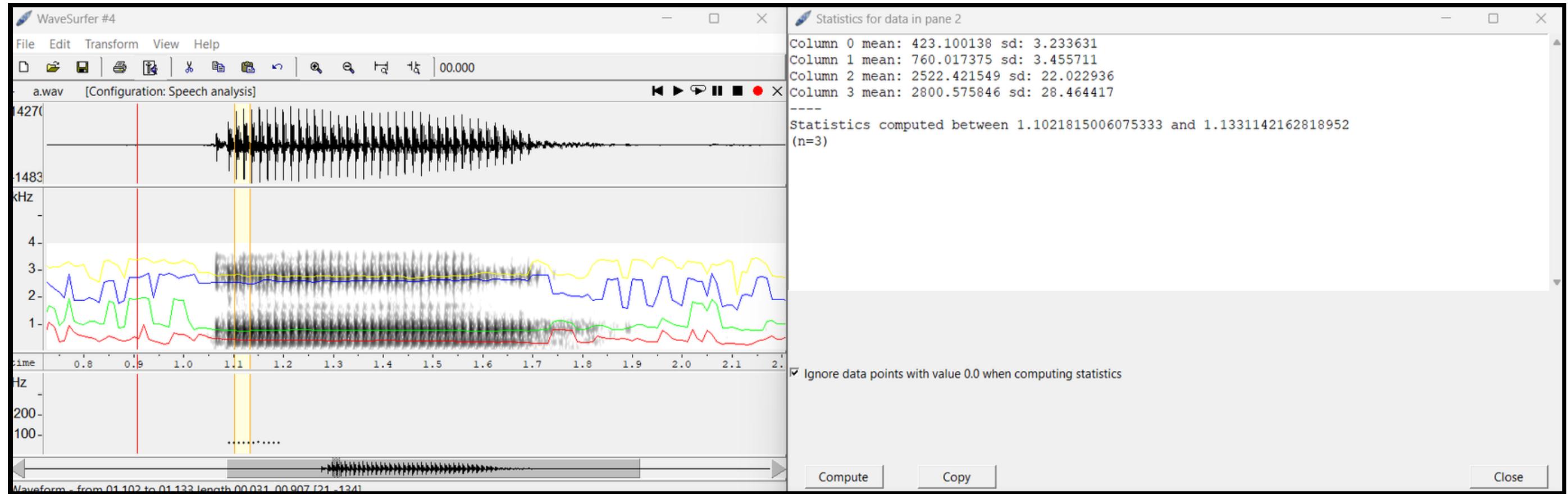
CODE TRÍCH XUẤT ẢNH PHỔ RỘNG CHO TỪNG NGUYÊN ÂM

```
def plot_vowel_waves_and_wideband_spectrograms_vowel(folder_name, speakers, vowel):
    """Plot the waveform and wideband spectrogram of the audio signal for a specific vowel."""
    file_paths = {f"{speaker}/{vowel}": os.path.join(folder_name, f"{speaker}/{vowel}.wav") for speaker in speakers}

    n_rows = len(speakers)
    plt.figure(figsize=(10, 2 * n_rows))
    title = os.path.basename(folder_name)
    plt.suptitle(title, fontsize=16)
    for idx, (speaker, path) in enumerate(file_paths.items(), 1):
        try:
            audio, sr = load_audio(path)
            S_dB, hop_length = calculate_wideband_spectrogram(audio, sr)
            speaker = speaker[:-2]
            plot_waveform(audio, sr, speaker, vowel, idx, n_rows)
            plot_wideband_spectrogram(S_dB, sr, hop_length, speaker, vowel, idx, n_rows)
        except Exception as e:
            print(f"Error processing {vowel} for speaker {speaker}: {e}")

    plt.tight_layout(rect=[0, 0, 1, 0.95]) # Adjust the layout to make room for the suptitle
    plt.show()
```

Đo bộ 3 tần số formant bằng Wavesurfer



Kết quả 3 lần đo cho tất cả các mẫu

Lần 1:																
STT	ID	/a/			/e/			/i/			/o/			/u/		
		F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	23MTL	423	760	2522	340	971	2898	459	2068	2776	741	1060	2021	537	863	2469
2	24FTL	1109	2032	2680	470	959	2435	452	2659	3440	390	1091	3053	461	978	3136
3	25MLM	782	1527	2562	526	2114	2913	475	2057	2825	770	1209	2163	473	856	3361
4	27MCM	788	1351	2314	707	1670	2338	374	2129	3552	700	964	3533	381	739	1883
Lần 2:																
STT	ID	/a/			/e/			/i/			/o/			/u/		
		F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	23MTL	417	772	2600	350	998	2921	388	2281	2854	723	1044	1999	455	748	2429
2	24FTL	1307	2027	2686	448	982	2448	441	2781	3040	904	1468	3047	469	901	3124
3	25MLM	809	1569	2613	498	2143	2933	376	2155	2918	766	1162	2180	414	745	3087
4	27MCM	782	1316	2343	694	1721	2297	366	2105	3535	724	967	3561	379	734	1654
Lần 3:																
STT	ID	/a/			/e/			/i/			/o/			/u/		
		F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	23MTL	426	795	2579	345	992	2957	382	2397	3087	713	1028	2017	405	682	2488
2	24FTL	1180	1927	3048	398	916	2649	404	2793	3381	484	1086	3114	471	2006	3258
3	25MLM	744	1572	2579	505	2191	2888	365	2184	2958	761	1098	2247	376	722	3031
4	27MCM	787	1285	2323	771	1299	2363	356	2221	3544	725	938	3534	356	702	1744

Kết quả trung bình tất cả các mẫu sau 3 lần đo

STT	ID	/a/			/e/			/i/			/o/			/u/		
		F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	23MTL	422	776	2567	345	987	2925	410	2249	2906	726	1044	2012	466	764	2462
2	24FTL	1199	1995	2805	438	952	2511	432	2745	3287	592	1215	3071	467	1295	3173
3	25MLM	778	1556	2584	510	2149	2912	405	2132	2900	765	1156	2197	421	774	3160
4	27MCM	786	1318	2327	724	1563	2333	365	2152	3543	716	956	3543	372	725	1761

Kết quả trung bình tất cả các mẫu sau 3 lần đo

<i>/a/</i>	F1	F2	F3
Mean (Hz)	796	1411	2571
STD (Hz)	318	508	195
CV (%)	40	36	8

<i>/e/</i>	F1	F2	F3
Mean (Hz)	504	1413	2670
STD (Hz)	161	565	296
CV (%)	32	40	11

<i>/i/</i>	F1	F2	F3
Mean (Hz)	403	2320	3159
STD (Hz)	28	288	314
CV (%)	7	12	10

Kết quả trung bình tất cả các mẫu sau 3 lần đo

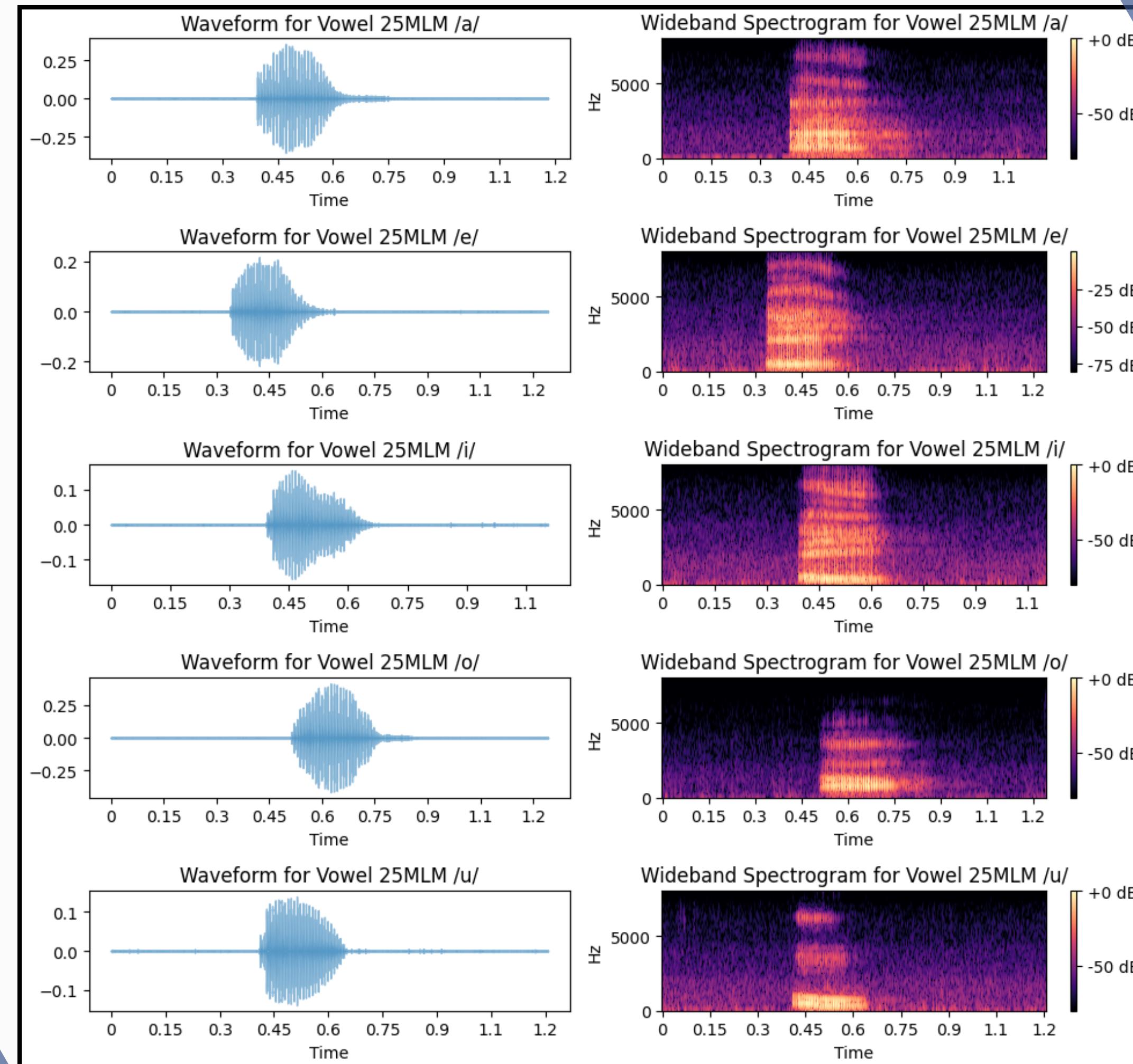
<i>/o/</i>	F1	F2	F3
Mean (Hz)	700	1093	2706
STD (Hz)	75	116	724
CV (%)	11	11	27

<i>/u/</i>	F1	F2	F3
Mean (Hz)	432	890	2639
STD (Hz)	45	271	673
CV (%)	10	30	26



Team 14

Ảnh phô 25MLM





Kết quả trung bình sau 3 lần đo cho mẫu 25MLM

Lần	<i>/a/</i>			<i>/e/</i>			<i>/i/</i>			<i>/o/</i>			<i>/u/</i>		
	F1	F2	F3	F1	F2	F3									
1	423	760	2522	340	971	2898	459	2068	2776	741	1060	2021	537	863	2469
2	809	1569	2613	498	2143	2933	376	2155	2918	766	1162	2180	414	745	3087
3	744	1572	2579	505	2191	2888	365	2184	2958	761	1098	2247	376	722	3031
Mean	778	1556	2584	510	2149	2912	405	2132	2900	765	1156	2197	421	774	3160

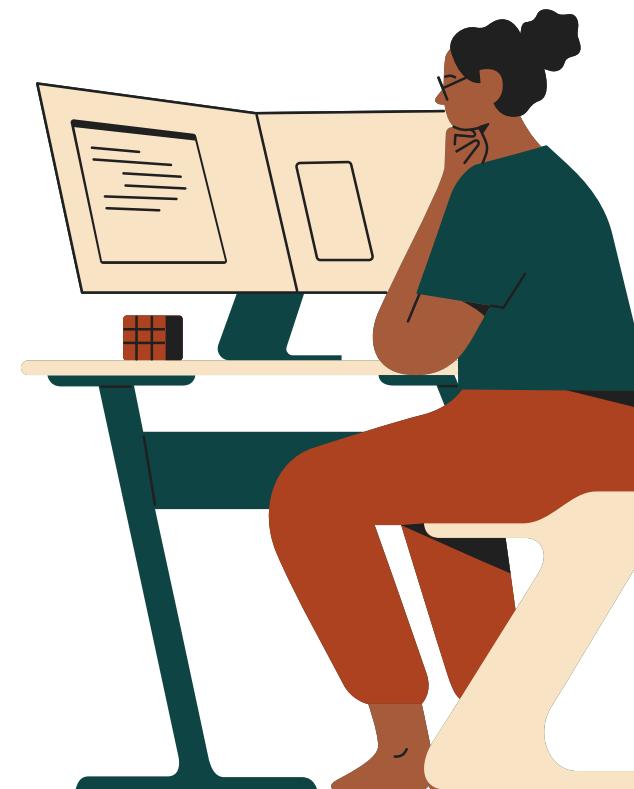


Kết quả trung bình sau 3 lần đo của 25MLM

<i>/a/</i>	F1	F2	F3
Mean (Hz)	778	1556	2584
STD (Hz)	33	25	26
CV (%)	4	2	1

<i>/e/</i>	F1	F2	F3
Mean (Hz)	510	2149	2912
STD (Hz)	14	39	22
CV (%)	3	2	1

<i>/i/</i>	F1	F2	F3
Mean (Hz)	405	2132	2900
STD (Hz)	61	66	68
CV (%)	15	3	2

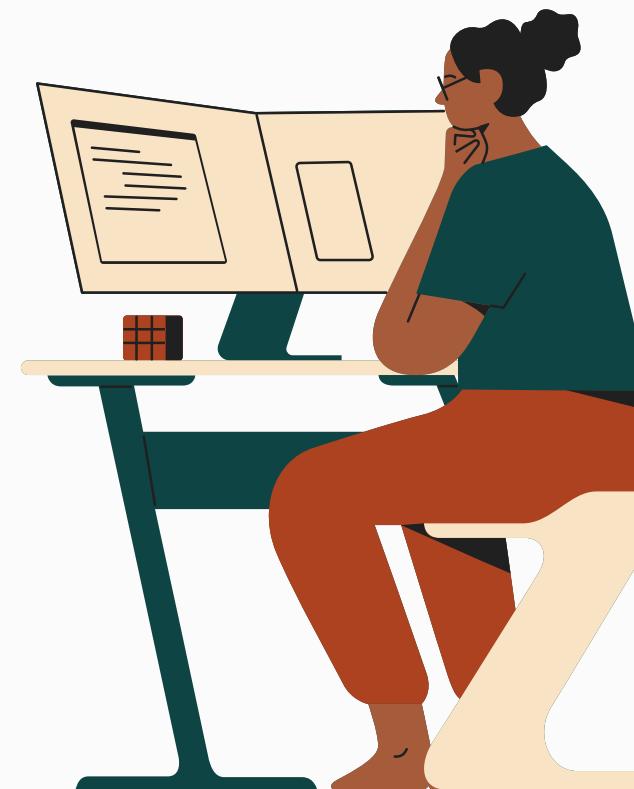




Kết quả trung bình sau 3 lần đo cho mẫu 25MLM

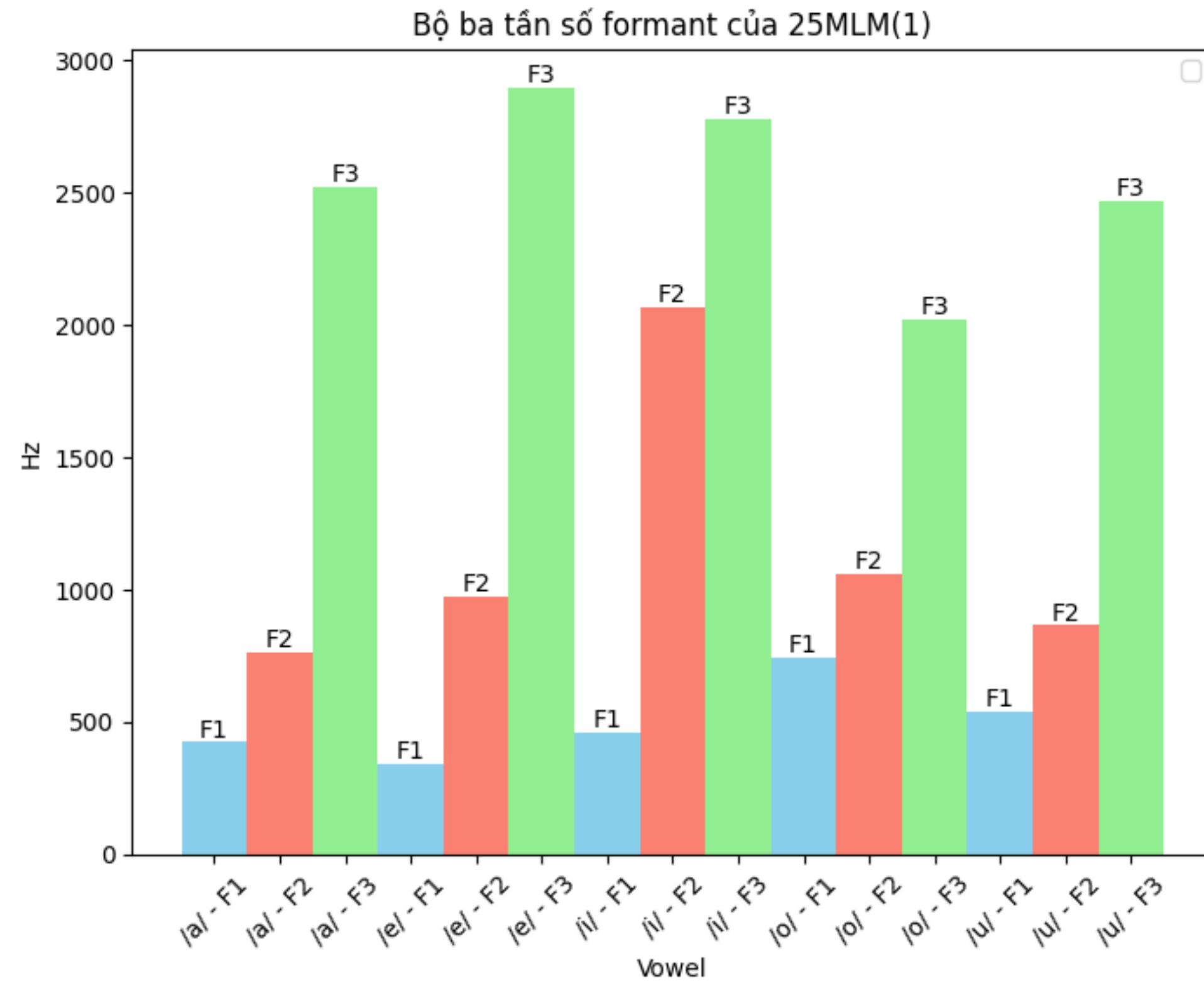
/ɔ:/	F1	F2	F3
Mean (Hz)	765	1156	2197
STD (Hz)	5	56	45
CV (%)	1	5	2

/u/	F1	F2	F3
Mean (Hz)	421	774	3160
STD (Hz)	49	72	176
CV (%)	12	9	6



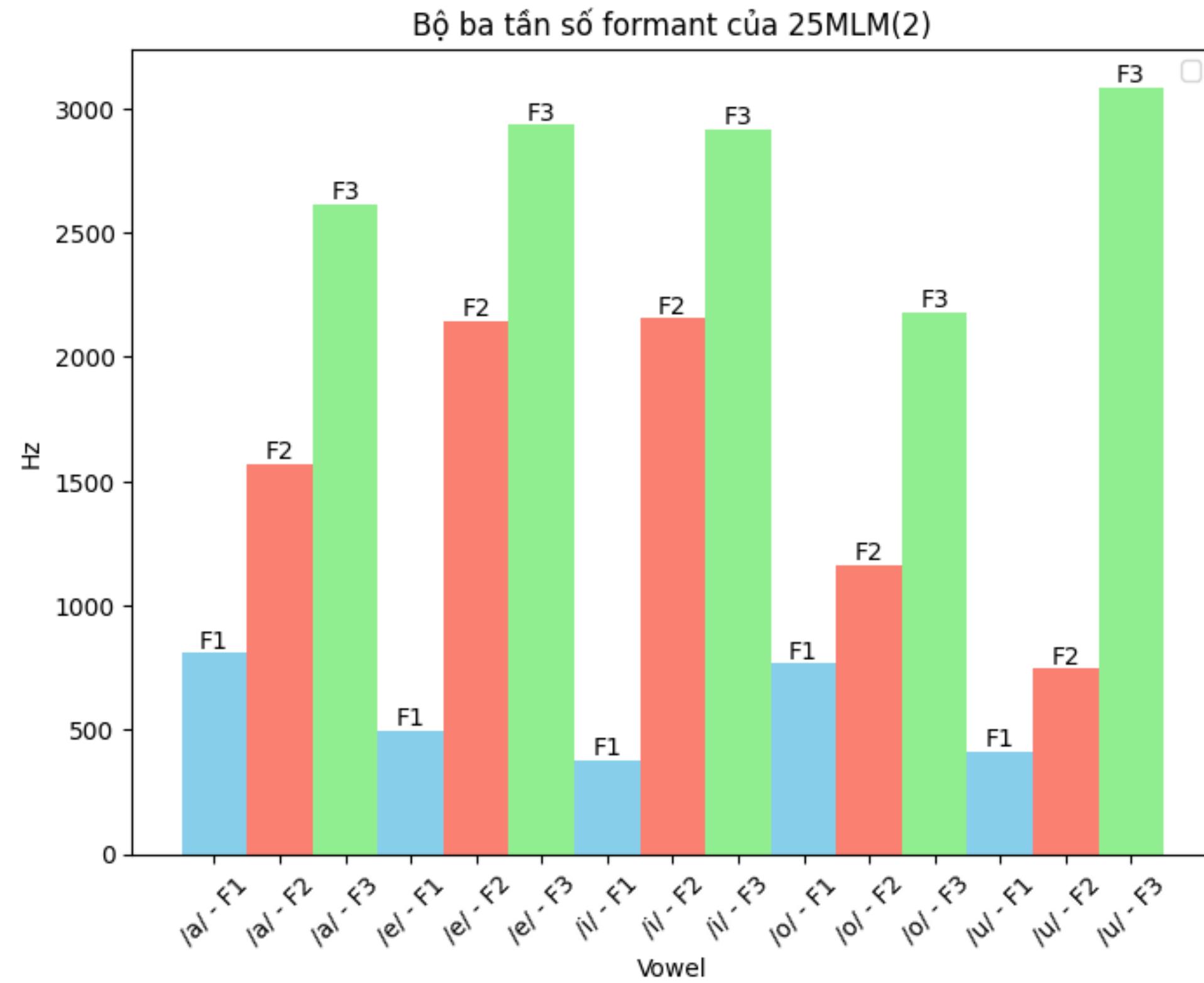


Biểu đồ cột bộ ba tần số formant của 25MLM - Lần đo thứ nhất



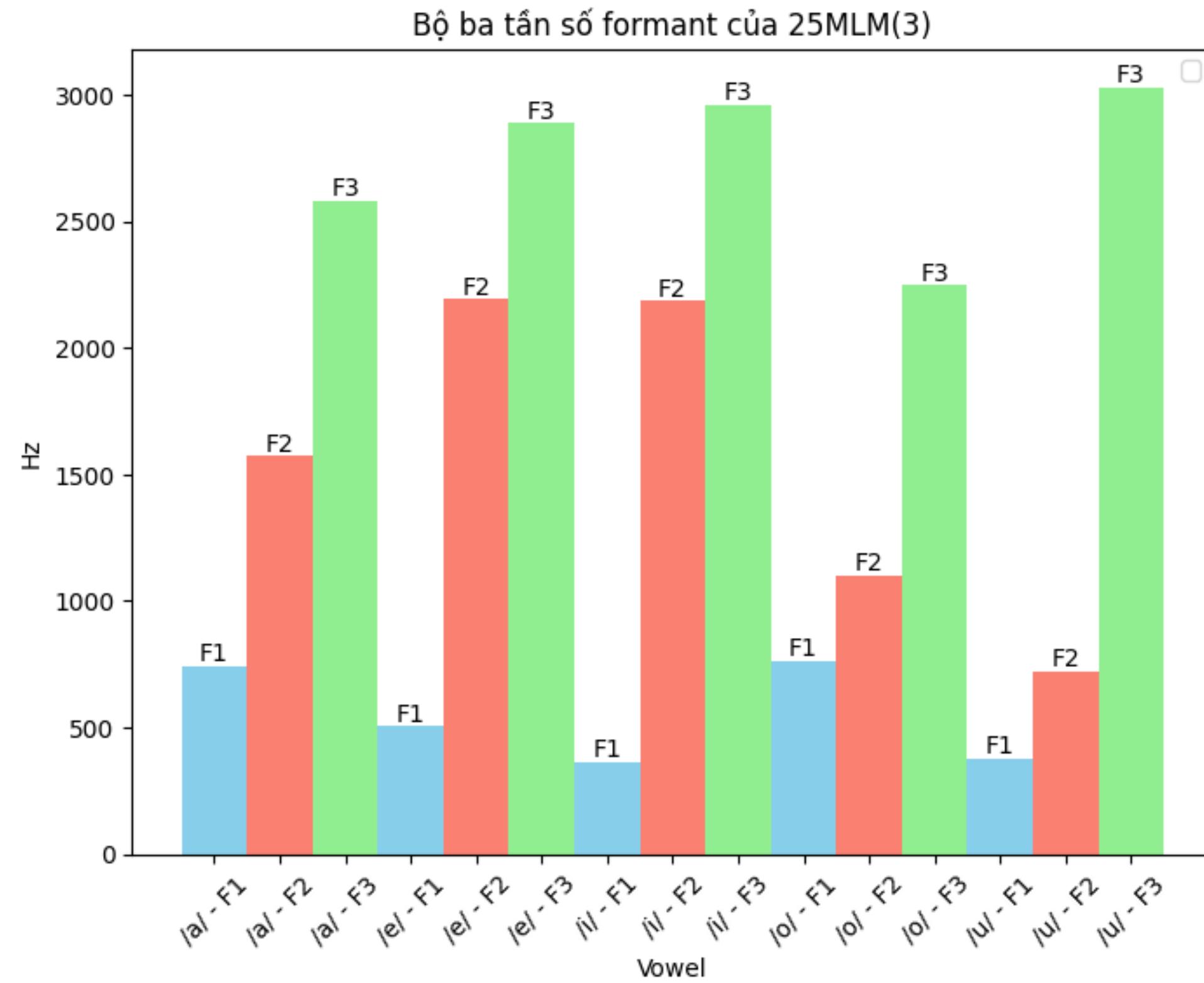


Biểu đồ cột bộ ba tần số formant của 25MLM - Lần đo thứ hai



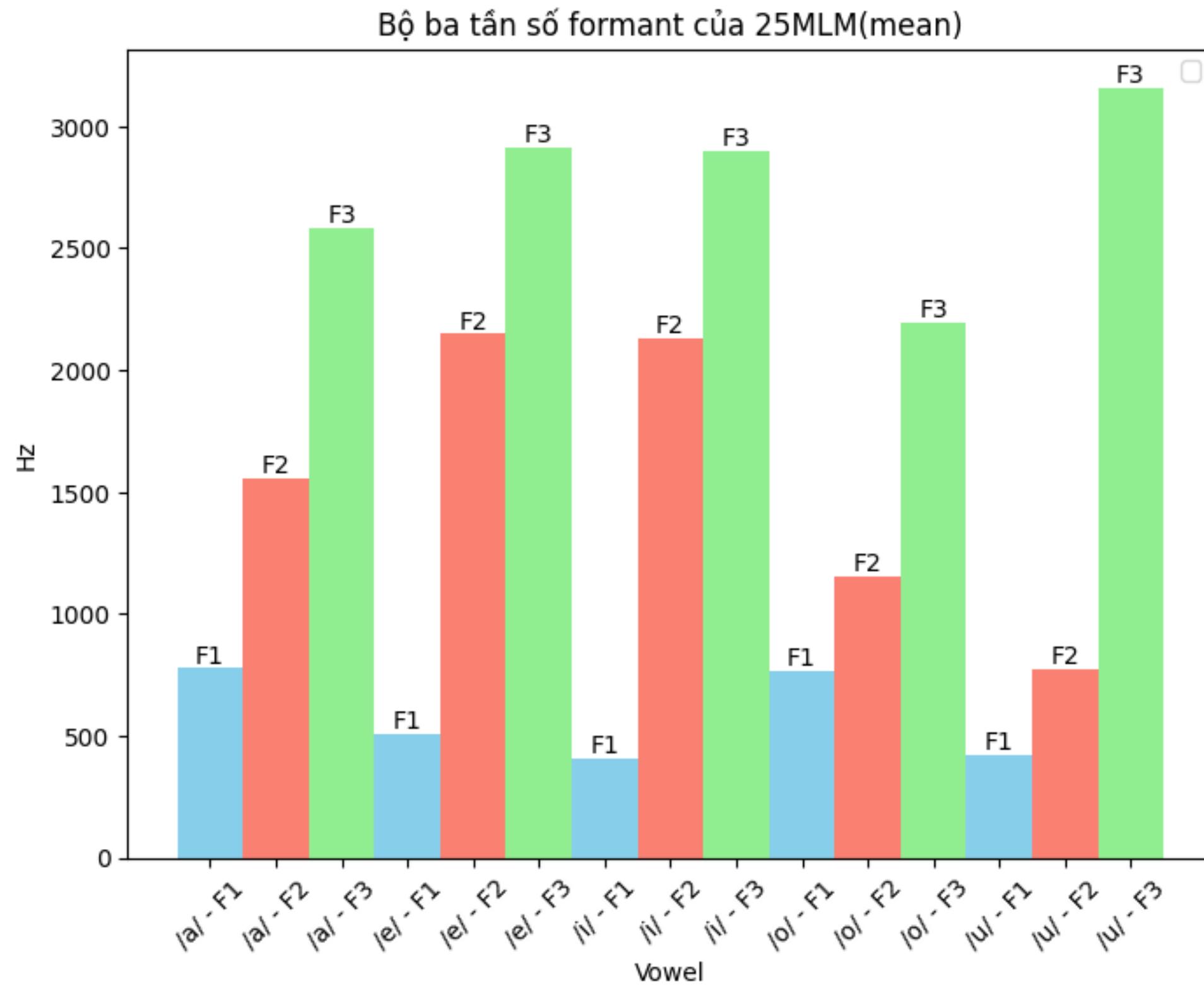


Biểu đồ cột bộ ba tần số formant của 25MLM - Lần đo thứ ba





Biểu đồ cột bộ ba tần số formant của 25MLM - Trung bình cộng của 3 lần đo





CODE TRÍCH XUẤT BIỂU ĐỒ HÌNH CỘT CHO BỘ BA TÂN SÔ FORMANT

```
def plot_chart(filepath):
    df = pd.read_csv(filepath)
    print(df)
    vowels = list(df.columns)
    print(vowels)
    title = f"Bộ ba tần số formant của {os.path.basename(filepath).split('.')[0]}"
    colors = ['skyblue', 'salmon', 'lightgreen']
    bar_colors = [colors[i % len(colors)] for i in range(len(vowels))]
    bar_width = 1
    bar_name = ['F1', 'F2', 'F3']
    for index, row in df.iterrows():
        plt.figure(figsize=(8, 6))
        plt.title(title + f"({index + 1 if index < 3 else 'mean'})")
        bars = plt.bar(vowels, row, color=bar_colors, width=bar_width)
        for i in range(len(bars)):
            yval = bars[i].get_height()
            plt.text(bars[i].get_x() + bars[i].get_width()/2.0, yval, bar_name[i % 3], va='bottom', ha='center')
        plt.xlabel('Vowel')
        plt.ylabel('Hz')
        plt.xticks(range(len(vowels)), vowels, rotation=45)

        plt.legend()
    plt.show()
```

Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

1. Nhận xét chung

- Cả 5 nguyên âm đều có tần số formant ít biến thiên và khá ổn định
- Nguyên âm có F1 mean cao nhất là nguyên âm /a/, thấp nhất là nguyên âm /i/
- Nguyên âm có F2 mean cao nhất là nguyên âm /e/, thấp nhất là nguyên âm /u/
- Nguyên âm có F3 mean cao nhất là nguyên âm /u/, thấp nhất là nguyên âm /i/
- Theo thứ tự bộ ba tần số formant(F1, F2, F3), độ lớn của tần số của mỗi nguyên âm đều tăng dần từ formant bậc thấp đến cao.

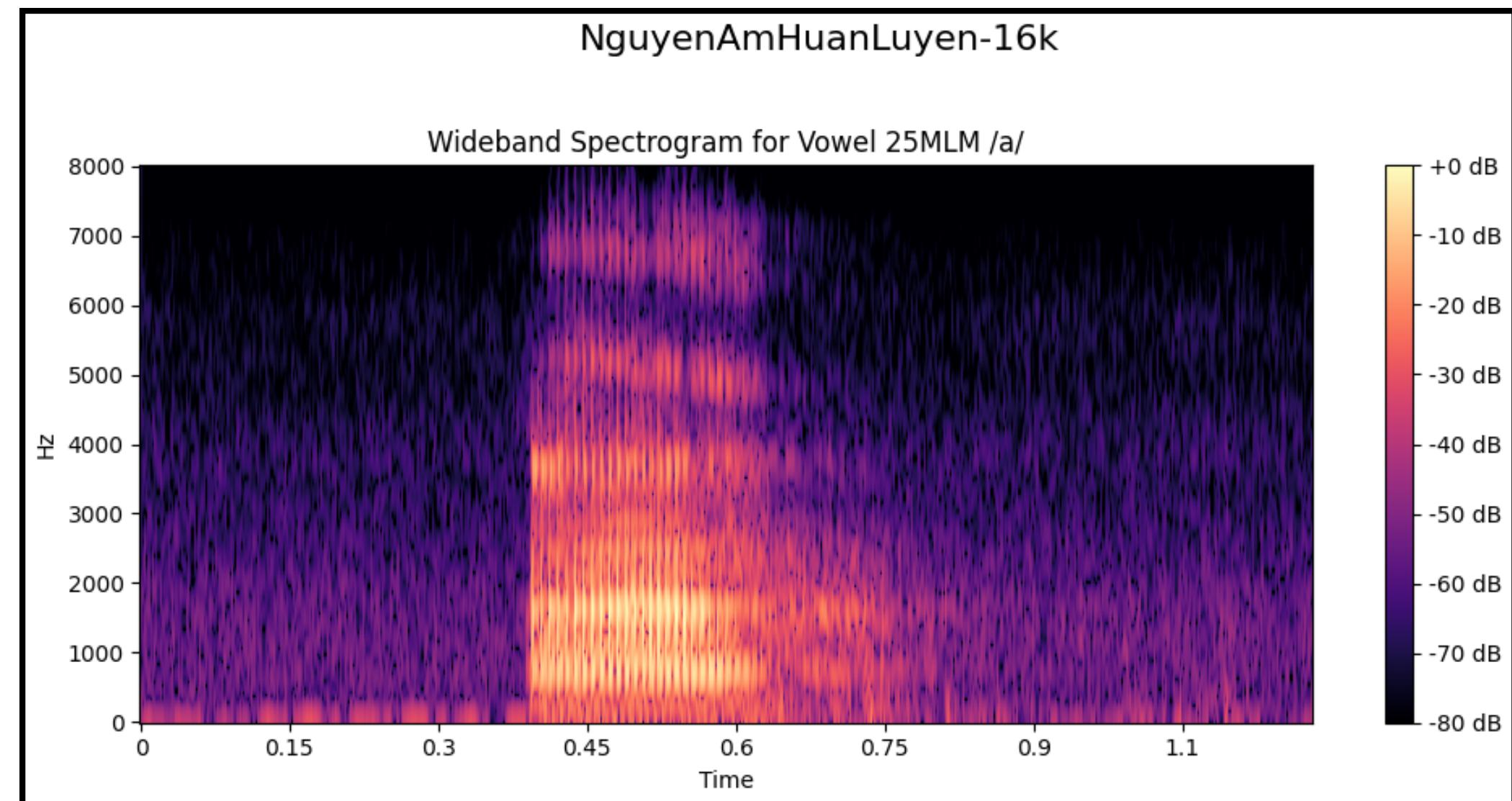


Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

2. Nhận xét từng nguyên âm

* Nguyên âm /a/

- Năng lượng của nguyên âm /a/ chủ yếu rơi vào dải tần từ 0 - 5500 Hz còn từ 5500Hz trở đi năng lượng tập trung rất ít không đáng kể. Năng lượng tín hiệu phân bố trên miền thời gian từ 0.38s đến 0.76s

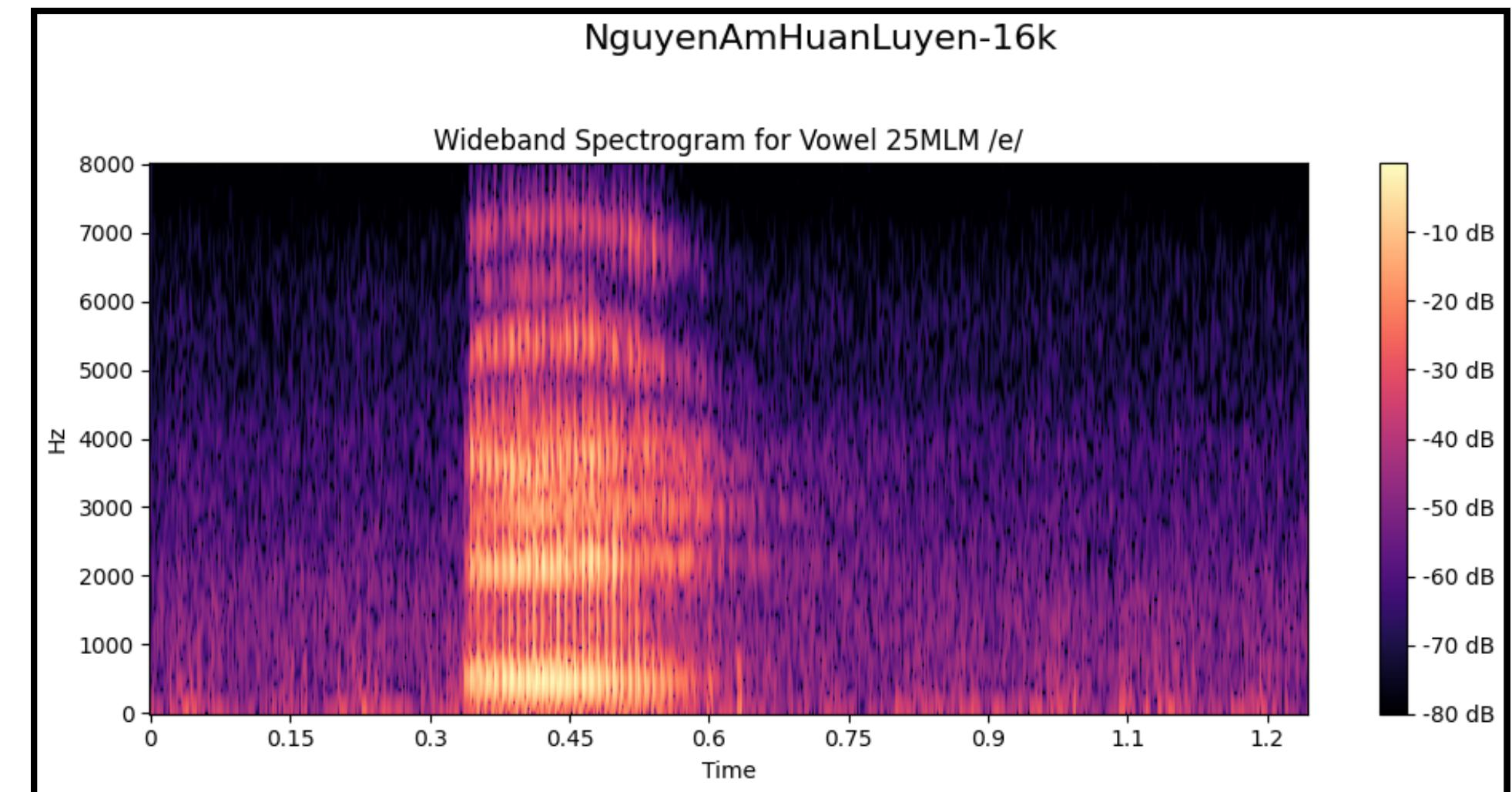


Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

2. Nhận xét từng nguyên âm

* Nguyên âm /e/

- Năng lượng của nguyên âm /e/ chủ yếu rơi vào dải tần từ 0 - 5900 Hz còn từ 5900Hz trở đi năng lượng tập trung rất ít không đáng kể. Năng lượng tín hiệu phân bố trên miền thời gian từ 0.33s đến 0.6s

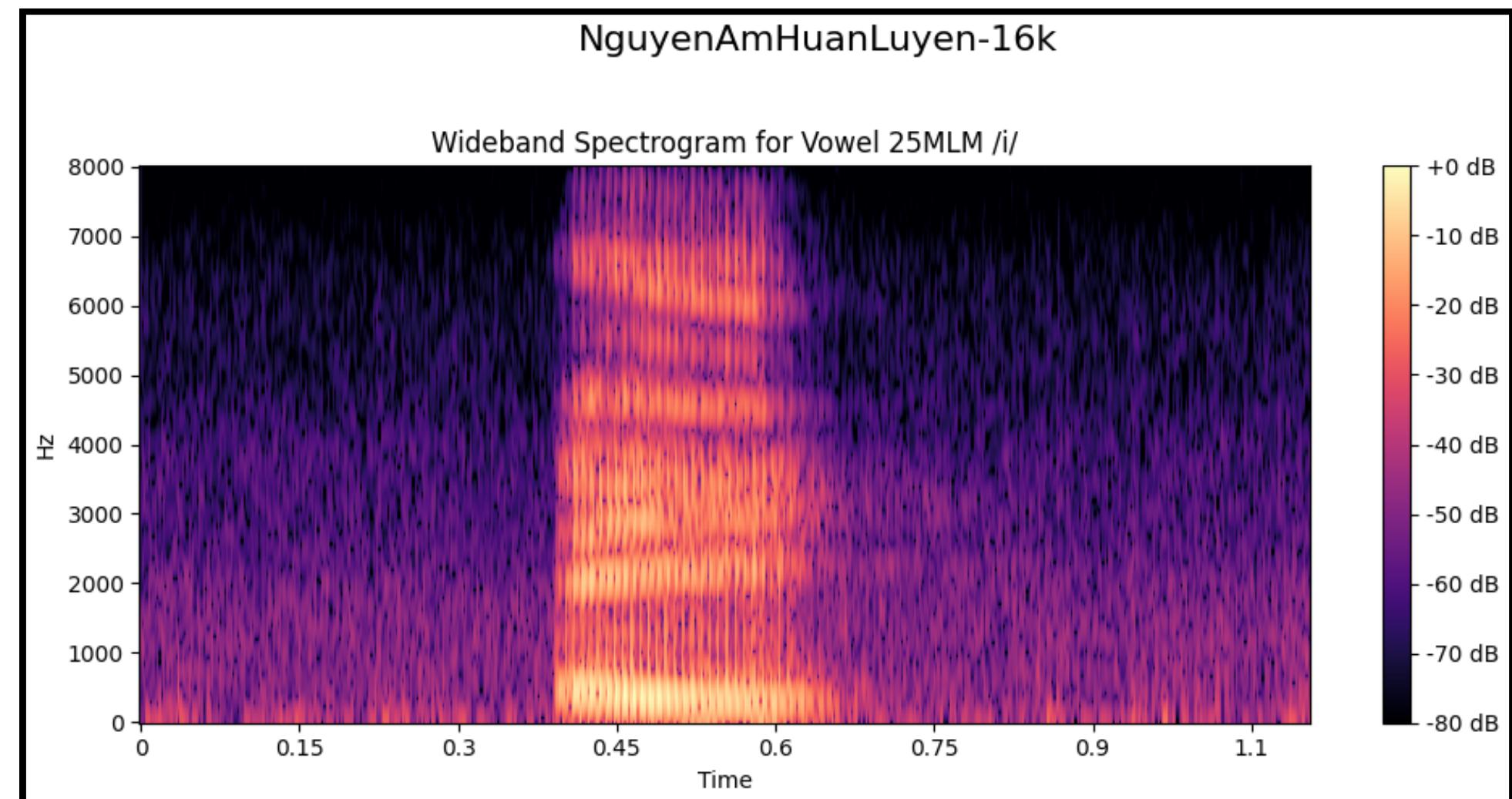


Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

2. Nhận xét từng nguyên âm

* Nguyên âm /i/

- Năng lượng của nguyên âm /i/ chủ yếu rơi vào dải tần từ 0 - 5200 Hz còn từ 5200Hz trở đi năng lượng tập trung rất ít không đáng kể. Năng lượng tín hiệu phân bố trên miền thời gian từ 0.38s đến 0.67s

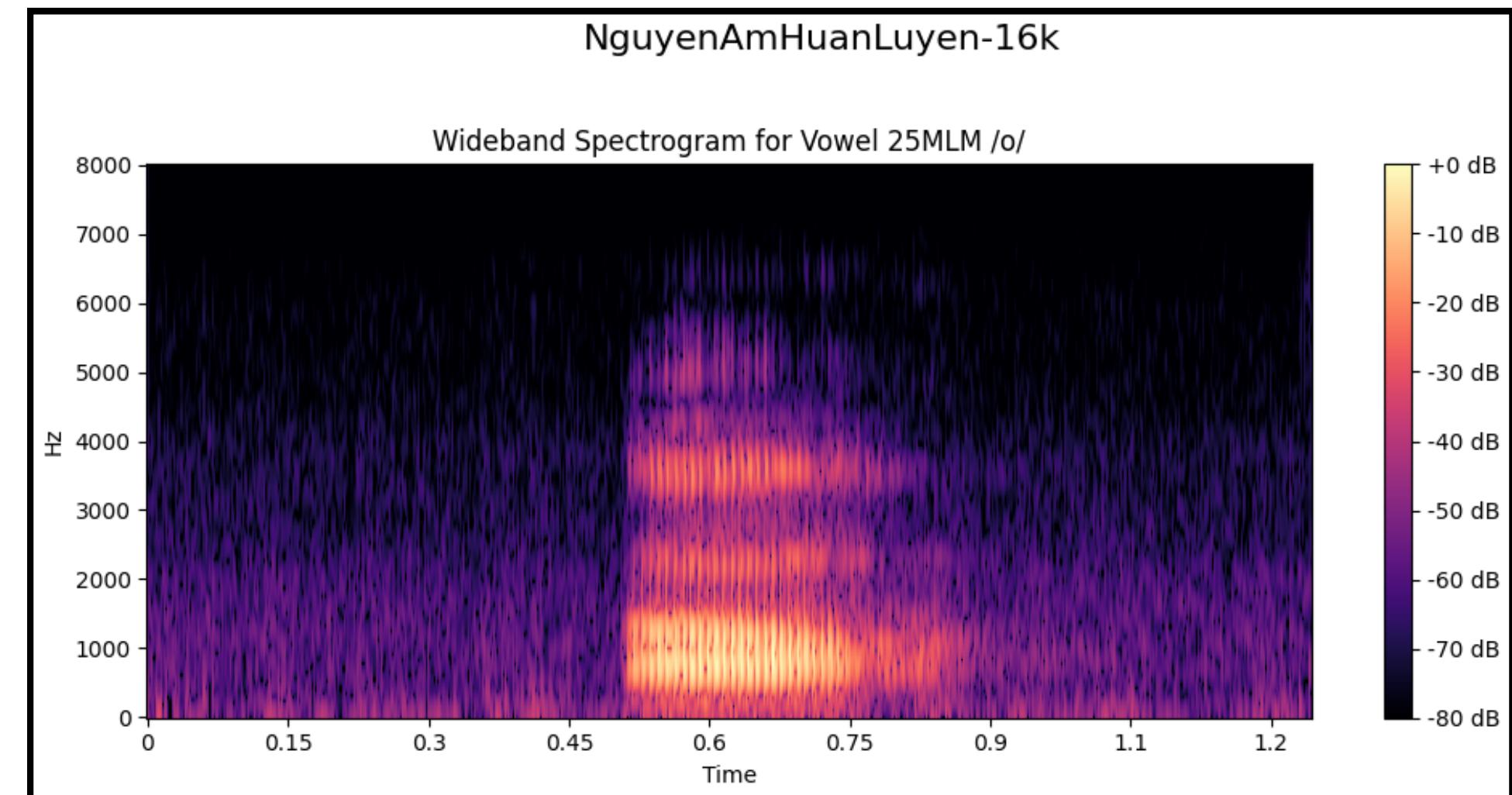


Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

2. Nhận xét từng nguyên âm

* Nguyên âm /o/

- Năng lượng của nguyên âm /u/ chủ yếu rơi vào dải tần từ 0 - 4100 Hz, đặc biệt cao ở dải tần từ 0 - 1800Hz còn từ 4200Hz trở đi năng lượng tập trung rất ít không đáng kể. Năng lượng tín hiệu phân bố trên miền thời gian từ 0.5s đến 0.87s

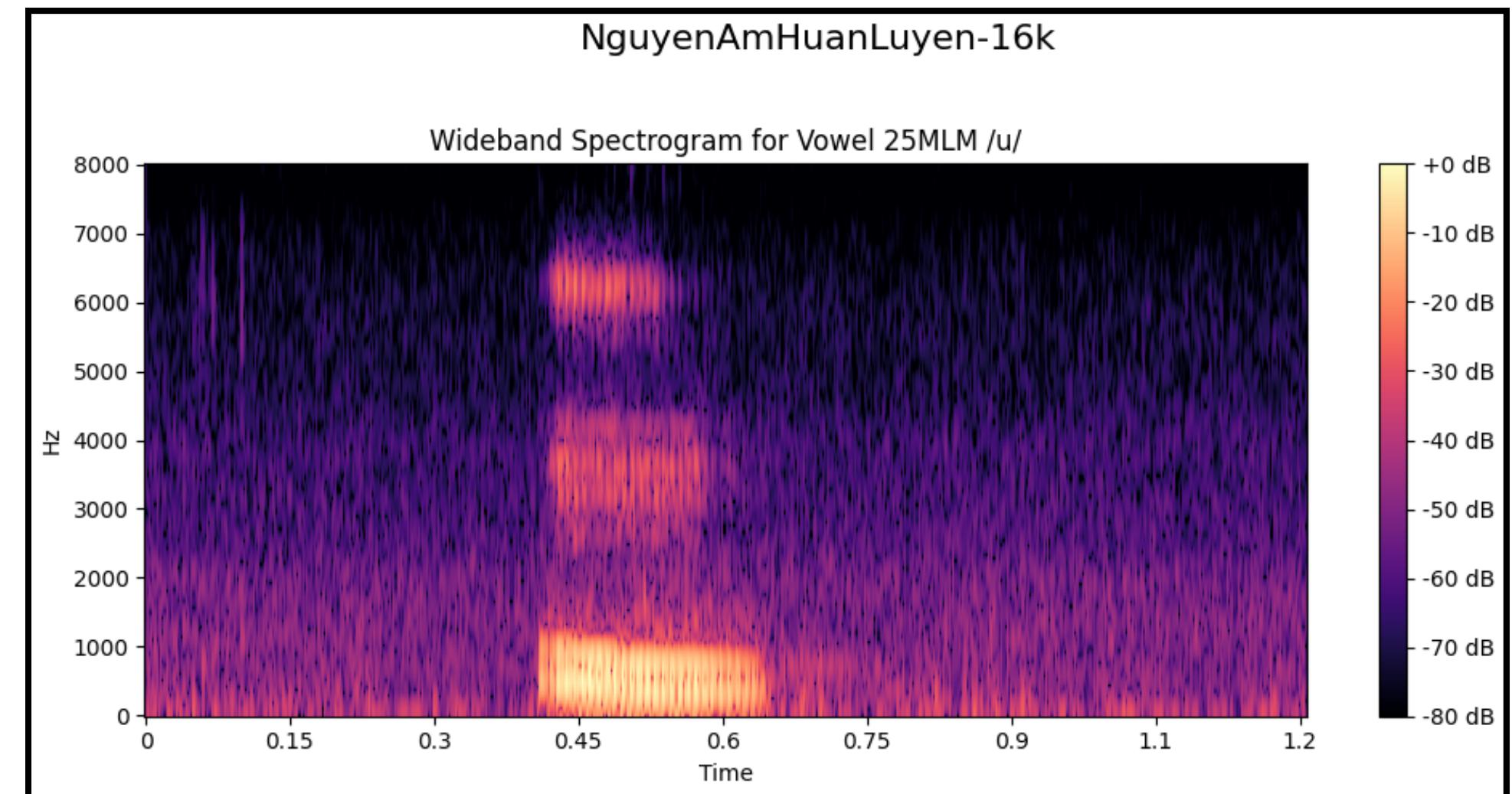


Nhận xét về sự khác biệt đặc trưng phổ giữa các nguyên âm của mẫu 25MLM

2. Nhận xét từng nguyên âm

* Nguyên âm /u/

- Năng lượng của nguyên âm /u/ chủ yếu rơi vào dải tần từ 0 - 4700 Hz, đặc biệt cao ở dải tần 0 - 1600Hz còn từ 4700Hz trở đi năng lượng tập trung rất ít không đáng kể. Năng lượng tín hiệu phân bố trên miền thời gian từ 0.41s đến 0.65s





CODE TRÍCH XUẤT ẢNH PHÔ RỘNG TỪNG NGUYÊN ÂM

```
def plot_wideband_spectrograms_vowel(folder_name, speakers, vowel):
    """Plot the wideband spectrogram of the audio signal for a specific vowel."""
    file_paths = {f"{speaker}/{vowel}": os.path.join(folder_name, f"{speaker}/{vowel}.wav") for speaker in speakers}

    n_rows = len(speakers)
    plt.figure(figsize=(10, 5 * n_rows))
    title = os.path.basename(folder_name)
    plt.suptitle(title, fontsize=16)
    for idx, (speaker, path) in enumerate(file_paths.items(), 1):
        try:
            audio, sr = load_audio(path)
            S_dB, hop_length = calculate_wideband_spectrogram(audio, sr)
            speaker = speaker[:-2]
            plt.subplot(n_rows, 1, idx)
            librosa.display.specshow(S_dB, sr=sr, hop_length=hop_length, x_axis='time', y_axis='linear')
            plt.colorbar(format='%.2f dB')
            plt.title(f"Wideband Spectrogram for Vowel {speaker} /{vowel}/")
        except Exception as e:
            print(f"Error processing {vowel} for speaker {speaker}: {e}")

    plt.tight_layout(rect=[0, 0, 1, 0.95]) # Adjust the layout to make room for the suptitle
    plt.show()
```

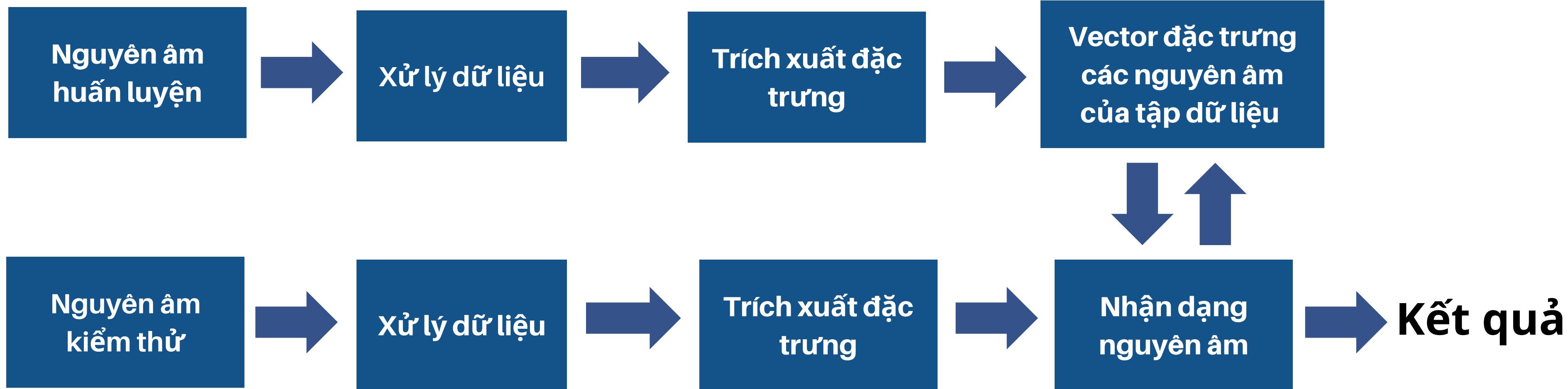


BÀI 2

PHÂN TÍCH ĐẶC TRƯNG PHỔ CÁC NGUYÊN ÂM CỦA NHIỀU NGƯỜI NÓI

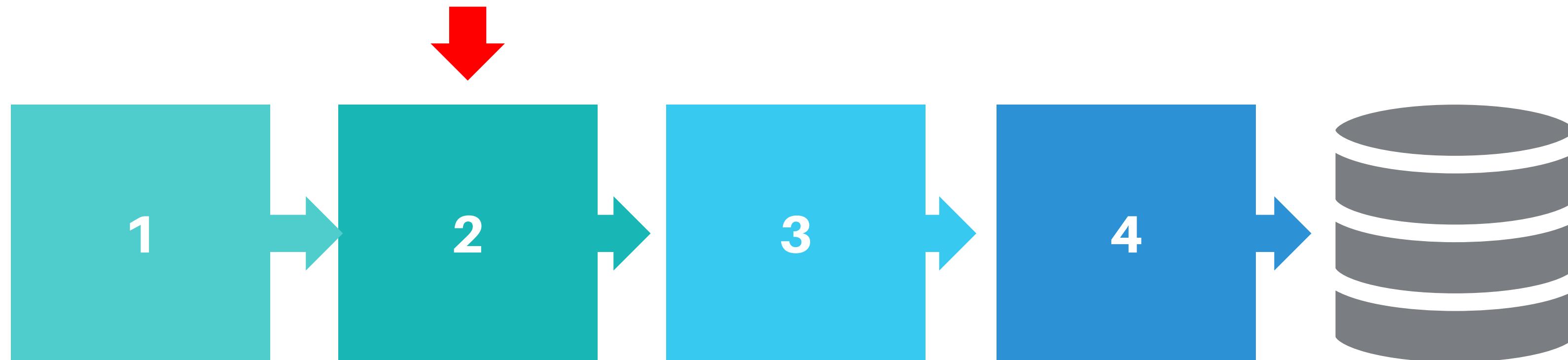


QUÁ TRÌNH XỬ LÝ





XỬ LÍ DỮ LIỆU HUẤN LUYỆN



Tập tín hiệu
huấn luyện

Xử lý tín hiệu
âm thanh

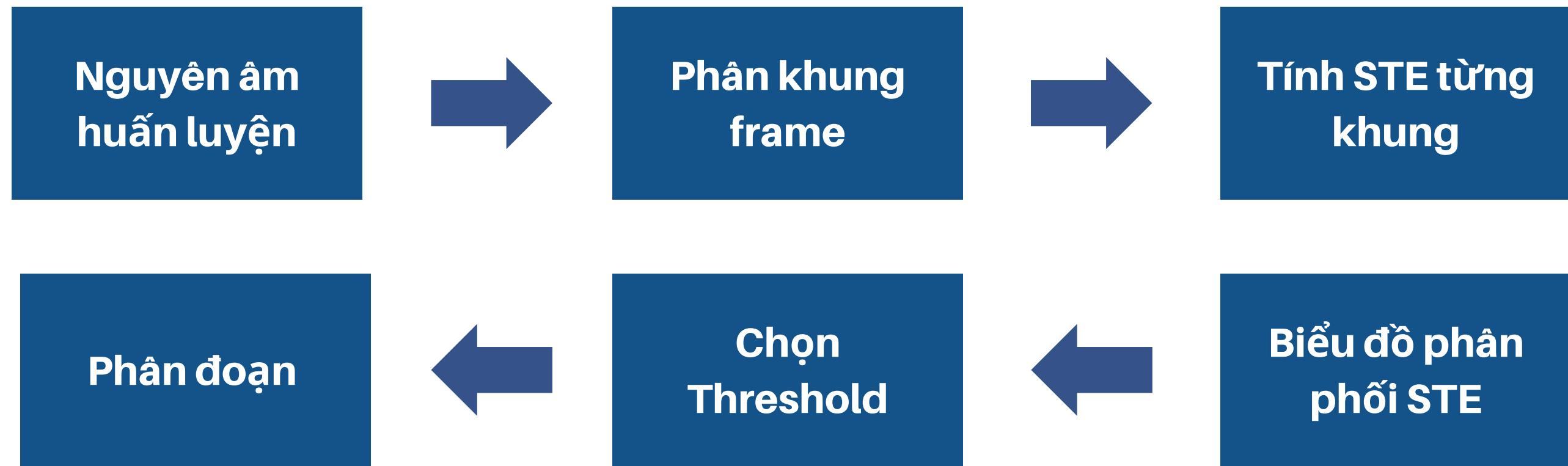
Trích xuất
vector đặc
trưng từng
tín hiệu

Tính toán
vector đặc
trưng từng
nguyên âm

Lưu trữ
vector đặc
trưng từng
nguyên âm

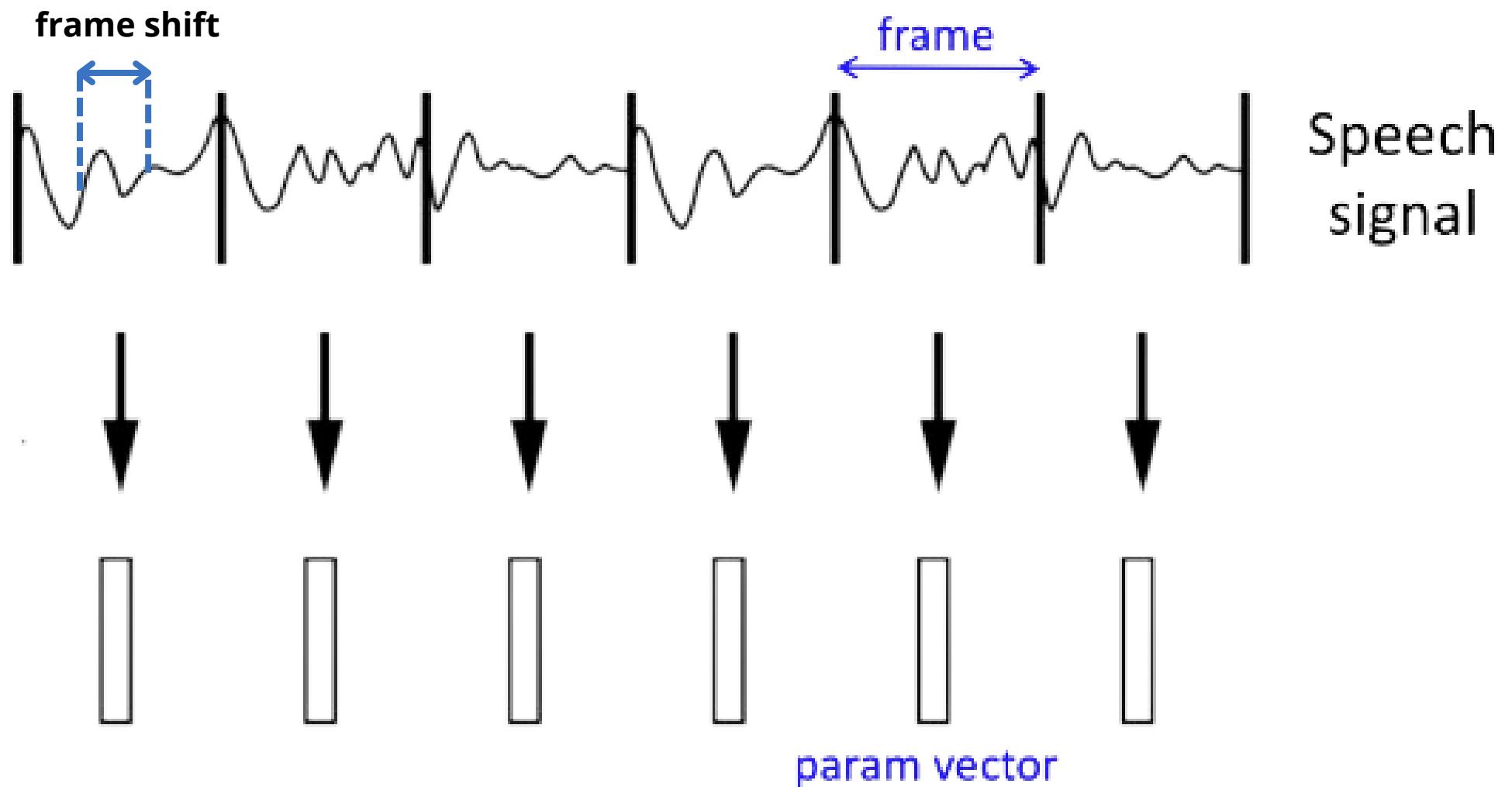


CÁC BƯỚC XỬ LÝ DỮ LIỆU





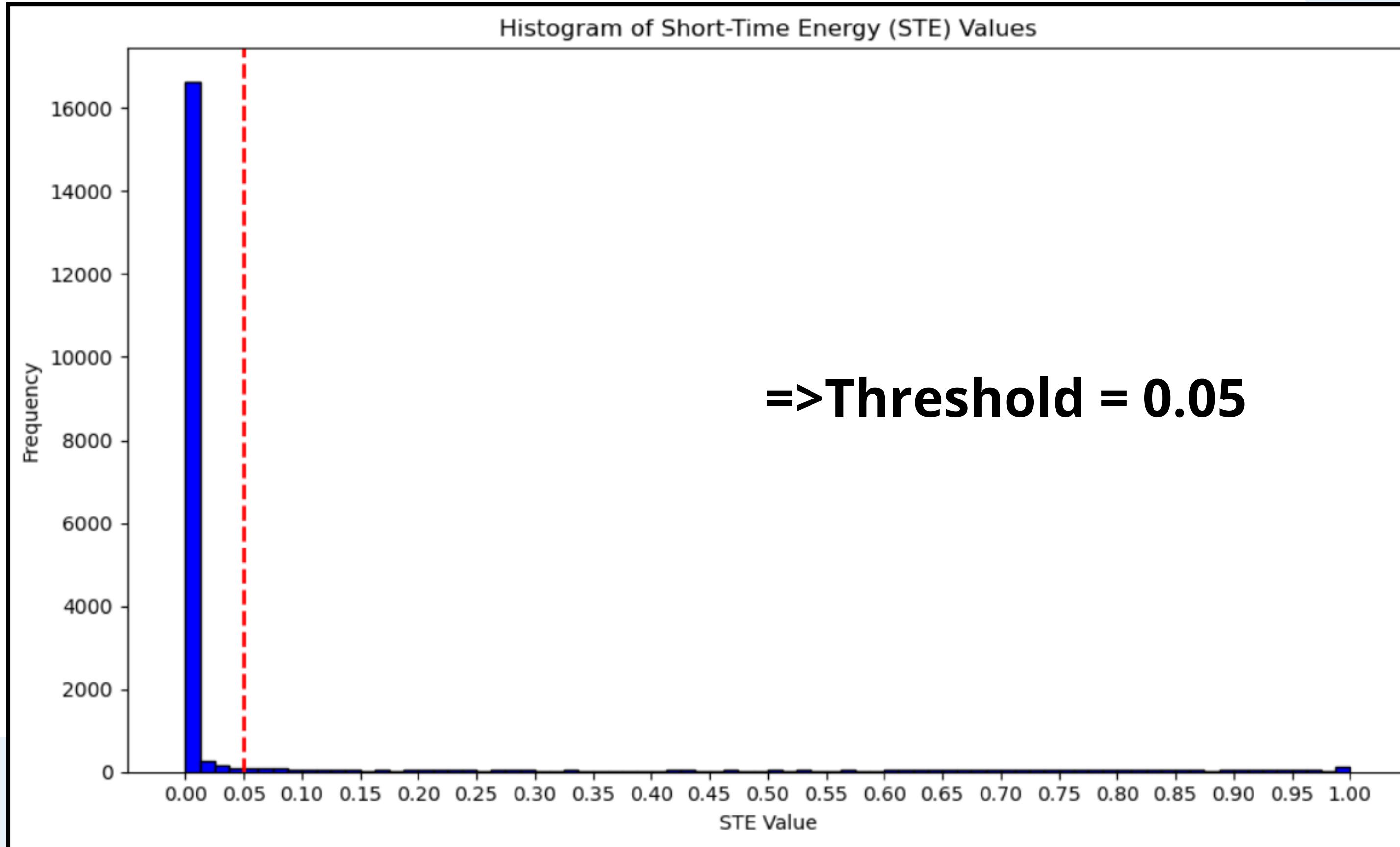
Phân chia các khung frame từ từ tín hiệu



Frame size = 20ms
Frame shift = 10 ms

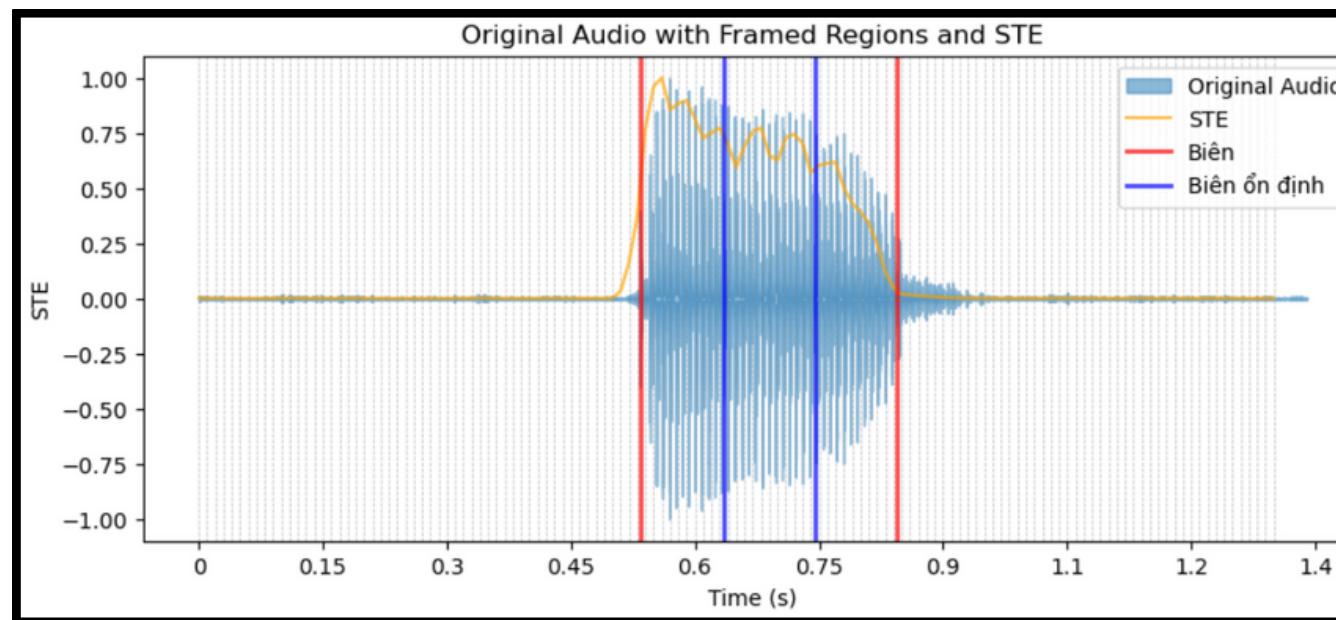


Tìm ngưỡng phân loại nguyên âm bằng biểu đồ phân bố STE

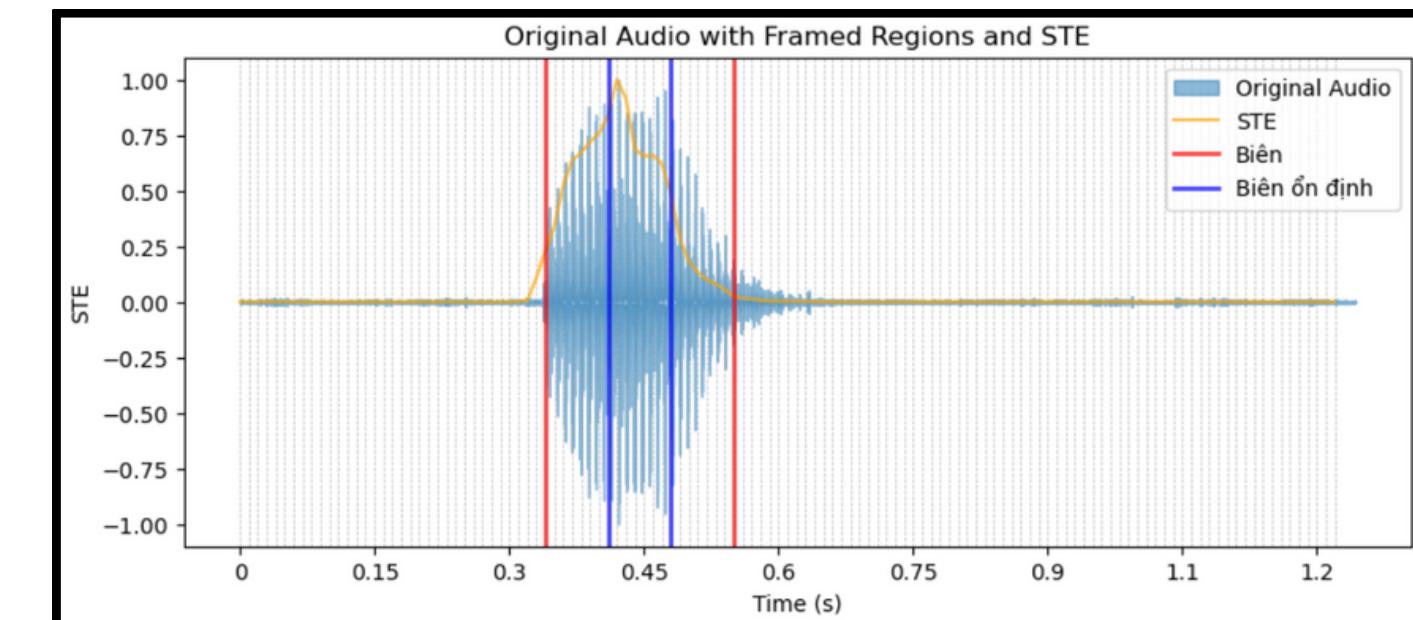




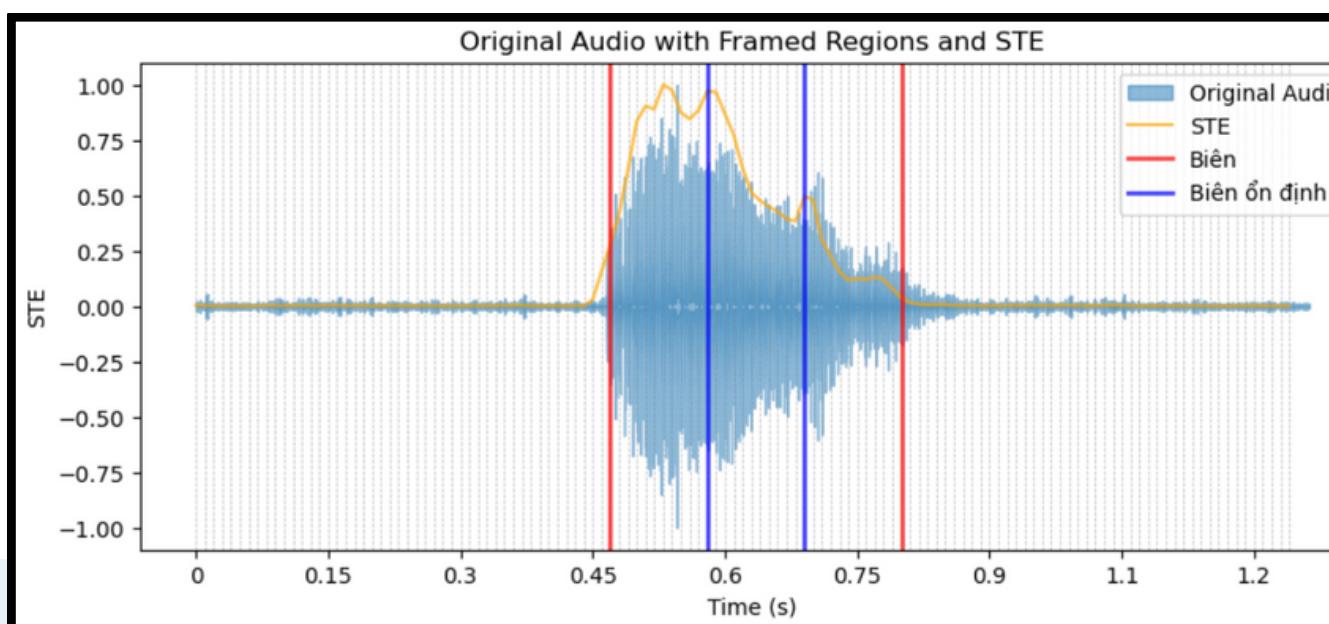
Phân đoạn và lấy phần ổn định của nguyên âm



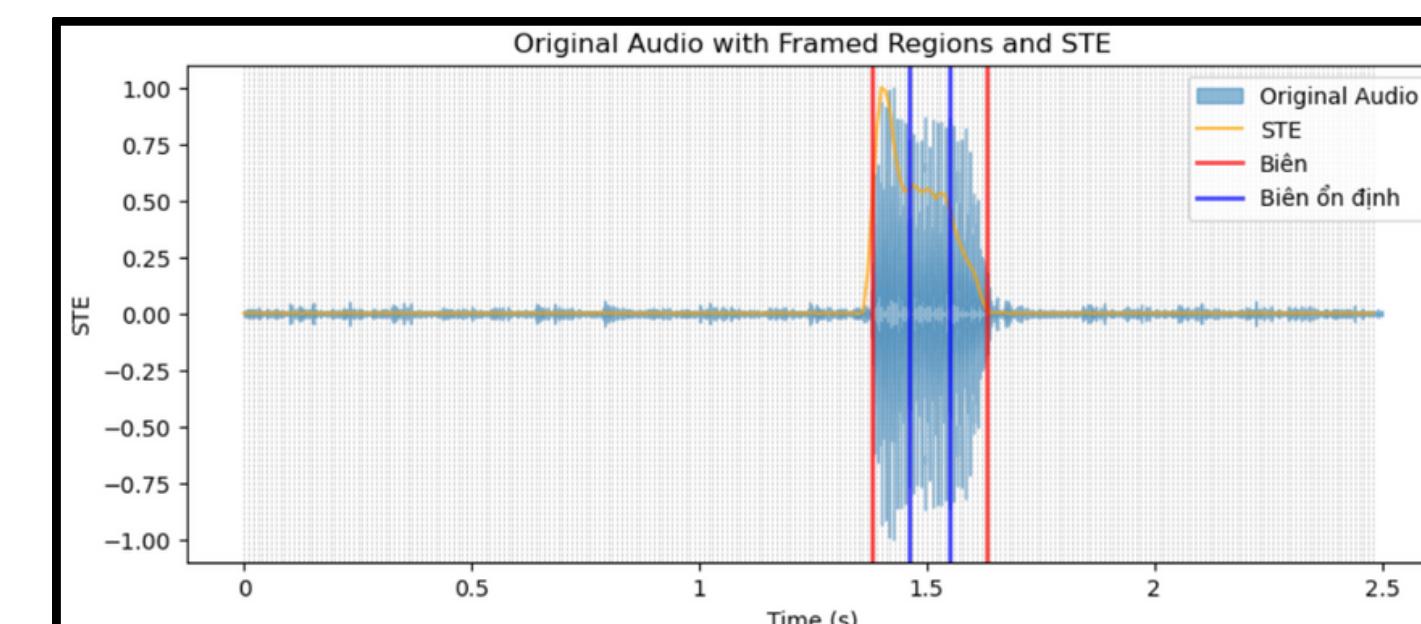
23MTL/a.wav



24FTL/a.wav



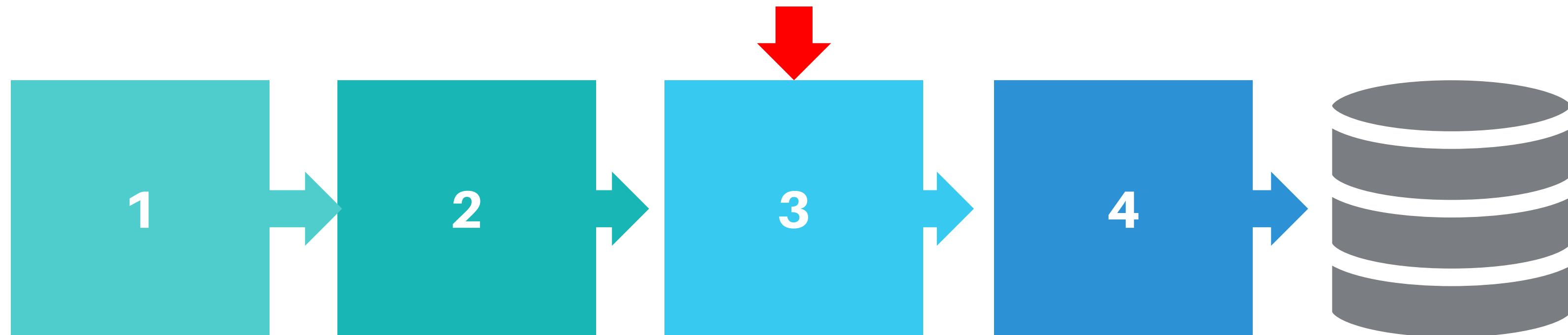
25MLM/e.wav



27MCM/l.wav



CÁC BƯỚC TÍNH VECTOR ĐẶC TRƯNG



Tập tín hiệu
huấn luyện

Xử lý tín hiệu
âm thanh

Trích xuất
vector đặc
trưng từng
tín hiệu

Tính toán
vector đặc
trưng từng
nguyên âm

Lưu trữ
vector đặc
trưng từng
nguyên âm



Team 14

CODE TRÍCH XUẤT VECTOR ĐẶC TRƯNG FFT

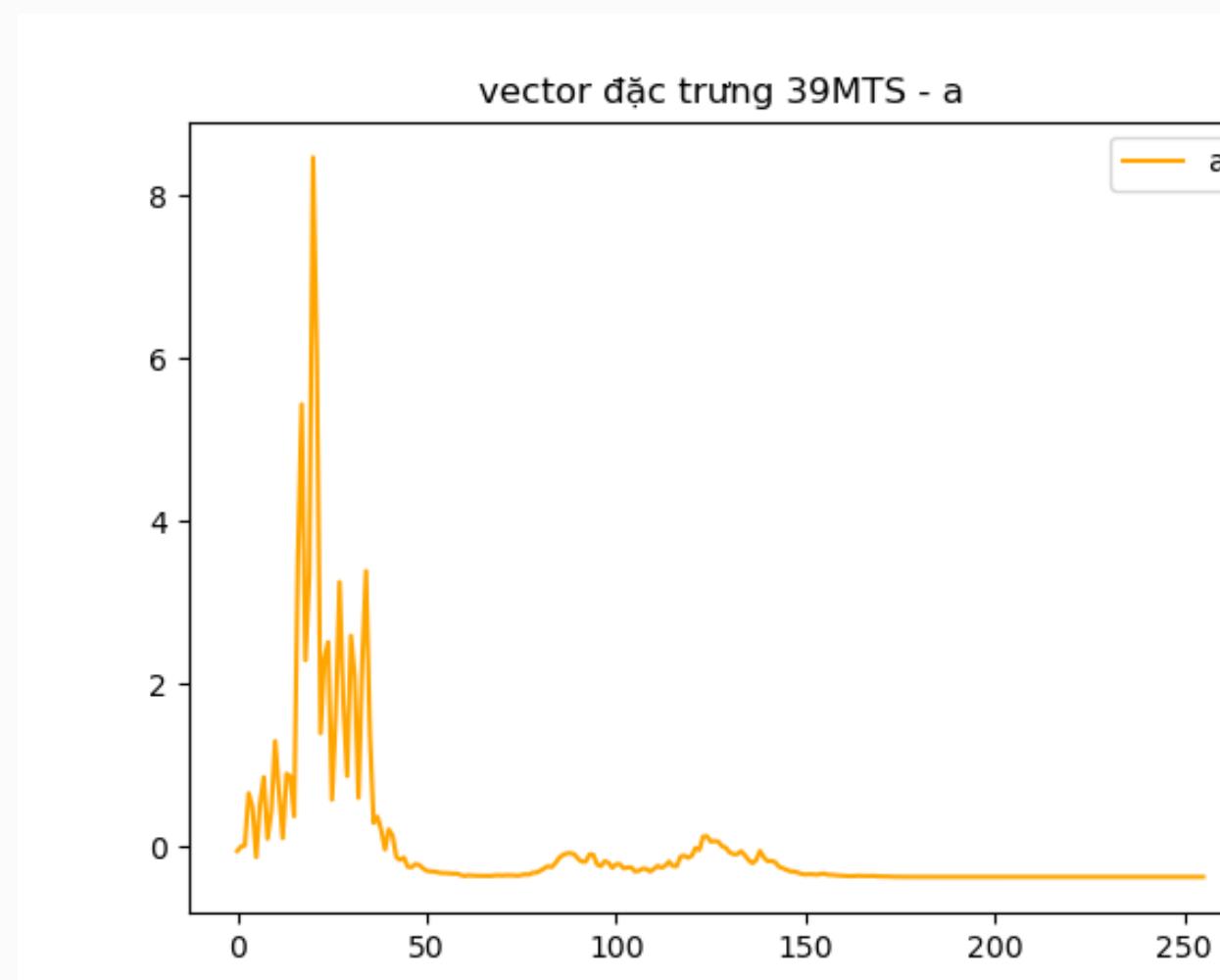
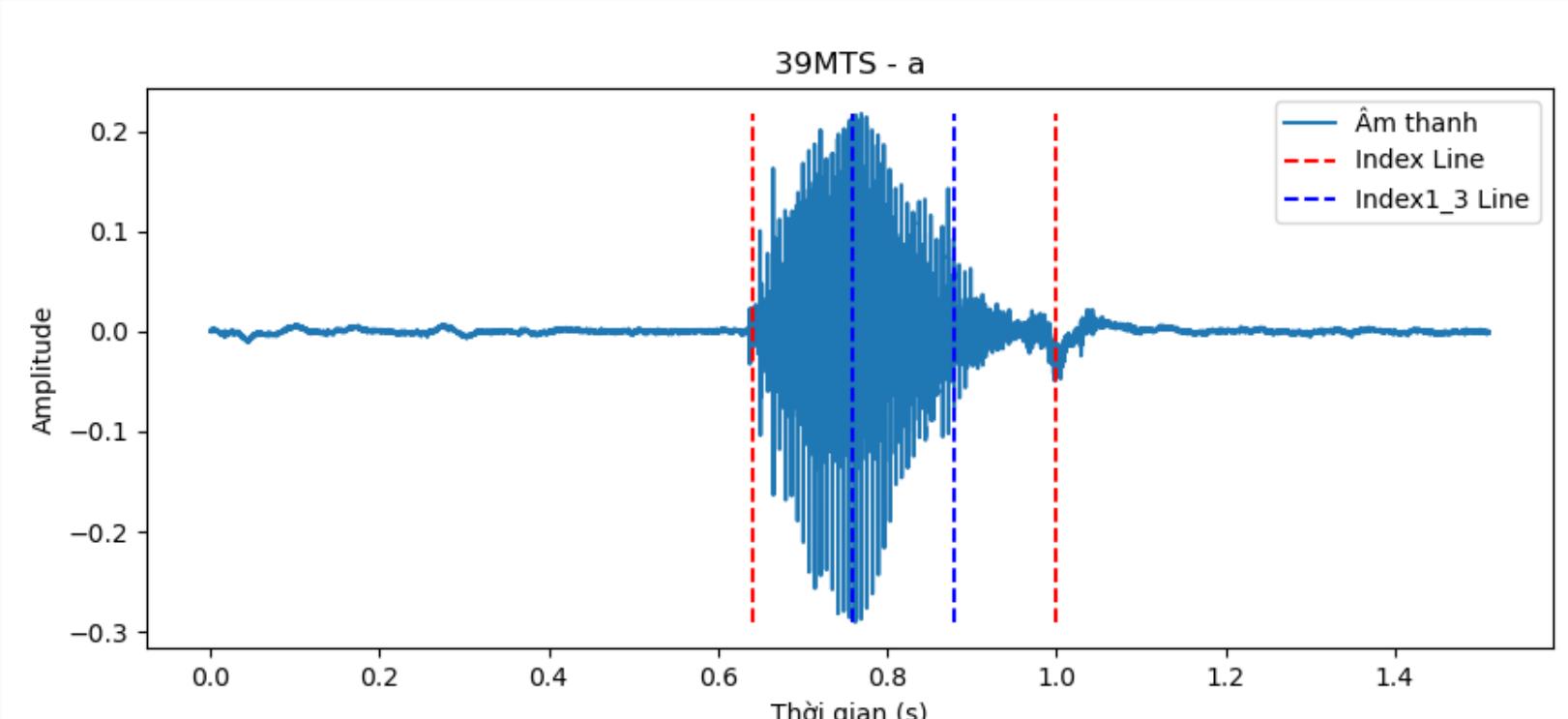
```
def extract_vectorFFT(y, sr, FFT_POINTS, frame_mode, window_function):
    frame_shift = int(frame_mode[1] * sr)
    frame_len = int(frame_mode[0] * sr)
    frame_num = (len(y) - frame_len) // frame_shift + 1

    fft_vector = [] # Use a Python list to store individual FFT vectors
    #chọn chế độ cửa sổ
    if window_function=='hamming':
        hm = hamming(frame_len)
    elif window_function=='bartlett':
        hm = bartlett(frame_len)
    elif window_function=='blackman':
        hm = blackman(frame_len)
    elif window_function=='kaiser':
        hm = kaiser(frame_len, 14)
    elif window_function=='hann':
        hm = hann(frame_len)

    for i in range(frame_num):
        start = i * frame_shift
        finish = start + frame_len
        frame = y[start:finish]
        yy = frame * hm
        fft_vector.append(np.abs(fft(yy, n=FFT_POINTS)))

    fft_array = np.array(fft_vector)
    normalized_array = fft_array / np.max(np.abs(fft_array))
    vector_mean = np.mean(normalized_array, axis=0).reshape(-1, 1)
    return vector_mean
```

TRÍCH XUẤT VECTOR ĐẶC TRƯNG FFT



Lấy tín hiệu vùng ổn định

Trích xuất N vector FFT của
N khung tín hiệu

Tính vector đặc trưng cho
1 nguyên âm của 1 người nói

Tính vector đặc trưng 1 nguyên
âm của nhiều người nói

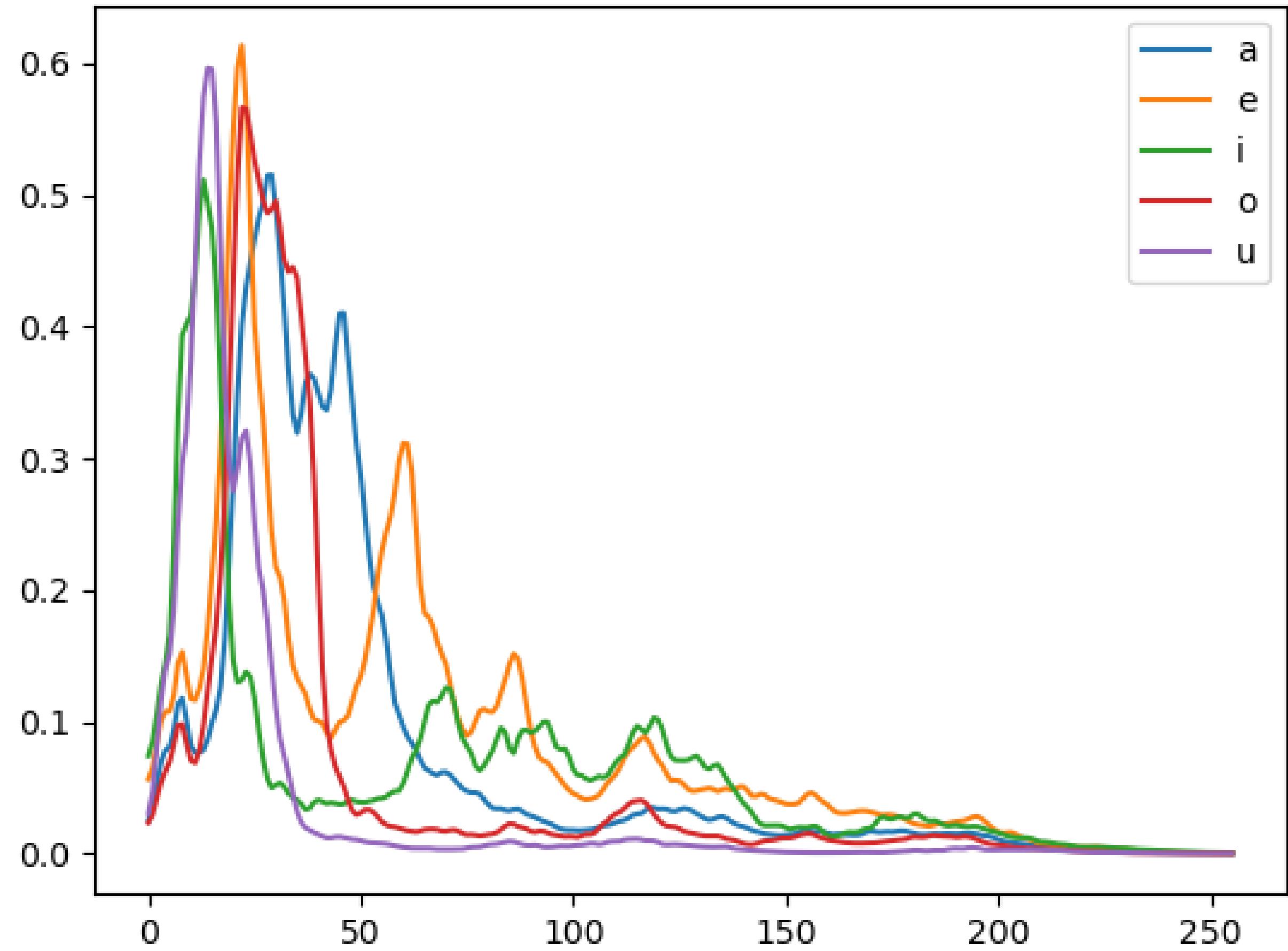
Tổng hợp vector đặc trưng
trên tập huấn luyện



Team 14

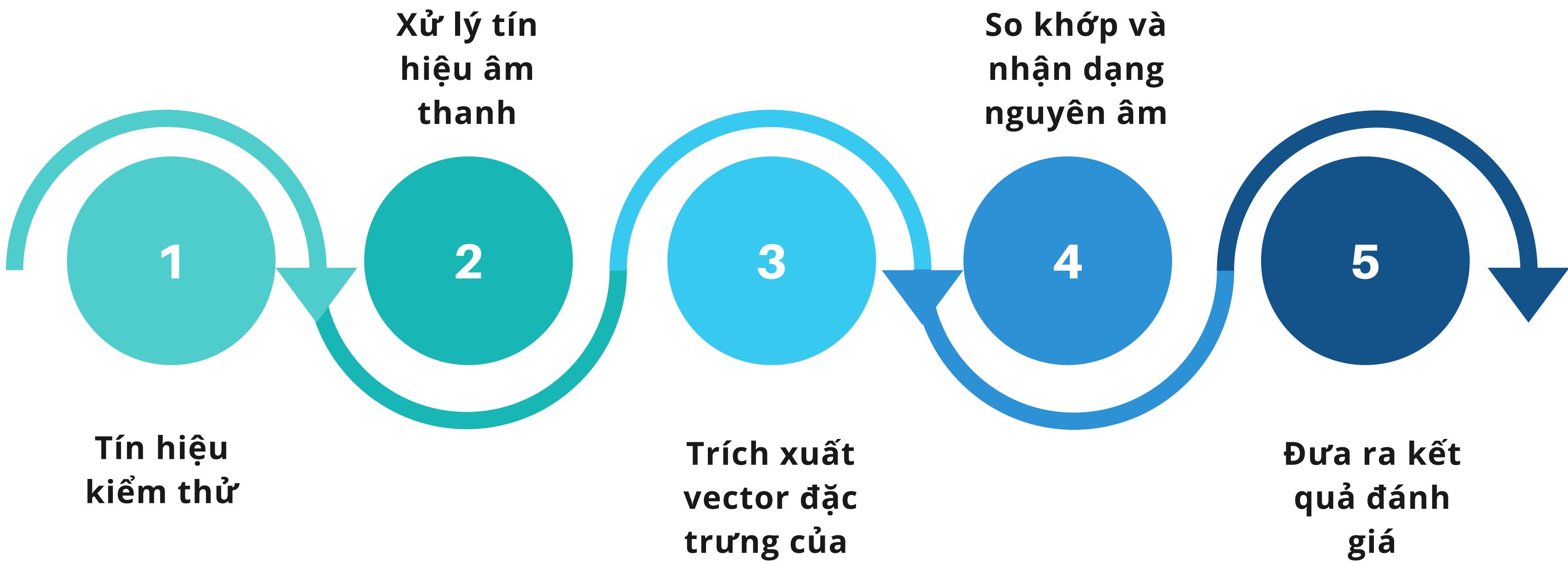
VECTOR ĐẶC TRƯNG 5 NGUYÊN ÂM BẰNG FFT

Vector đặc trưng 5 nguyên âm





CÁC BƯỚC XÁC ĐỊNH NHÃN





Team 14

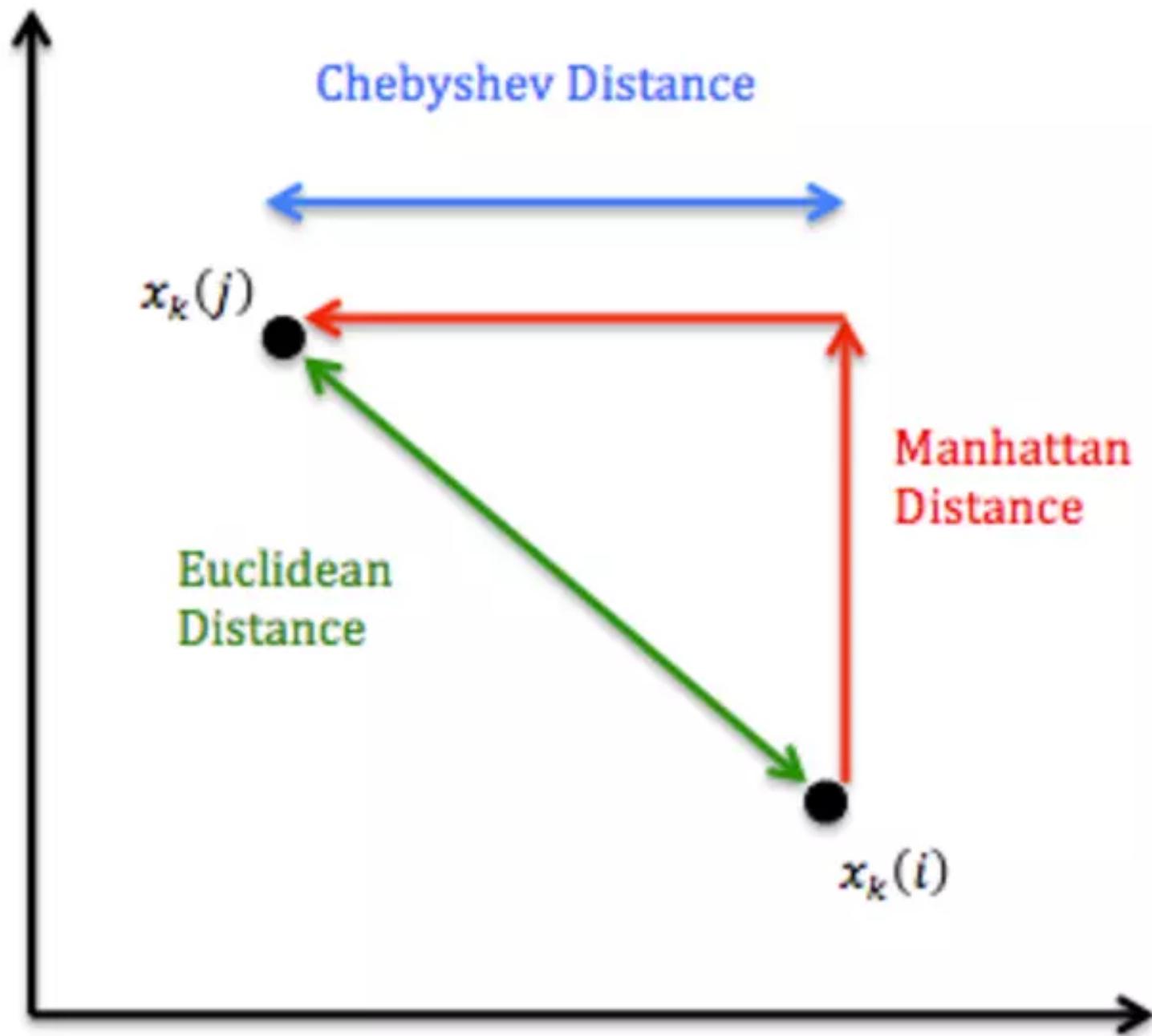


IDENTIFY

SO KHỚP VÀ NHẬN DẠNG NGUYÊN ÂM

MISSION

So khớp vector FFT
của tập kiểm thử
với các vector đặc
trưng của từng
nguyên âm đã trích
xuất

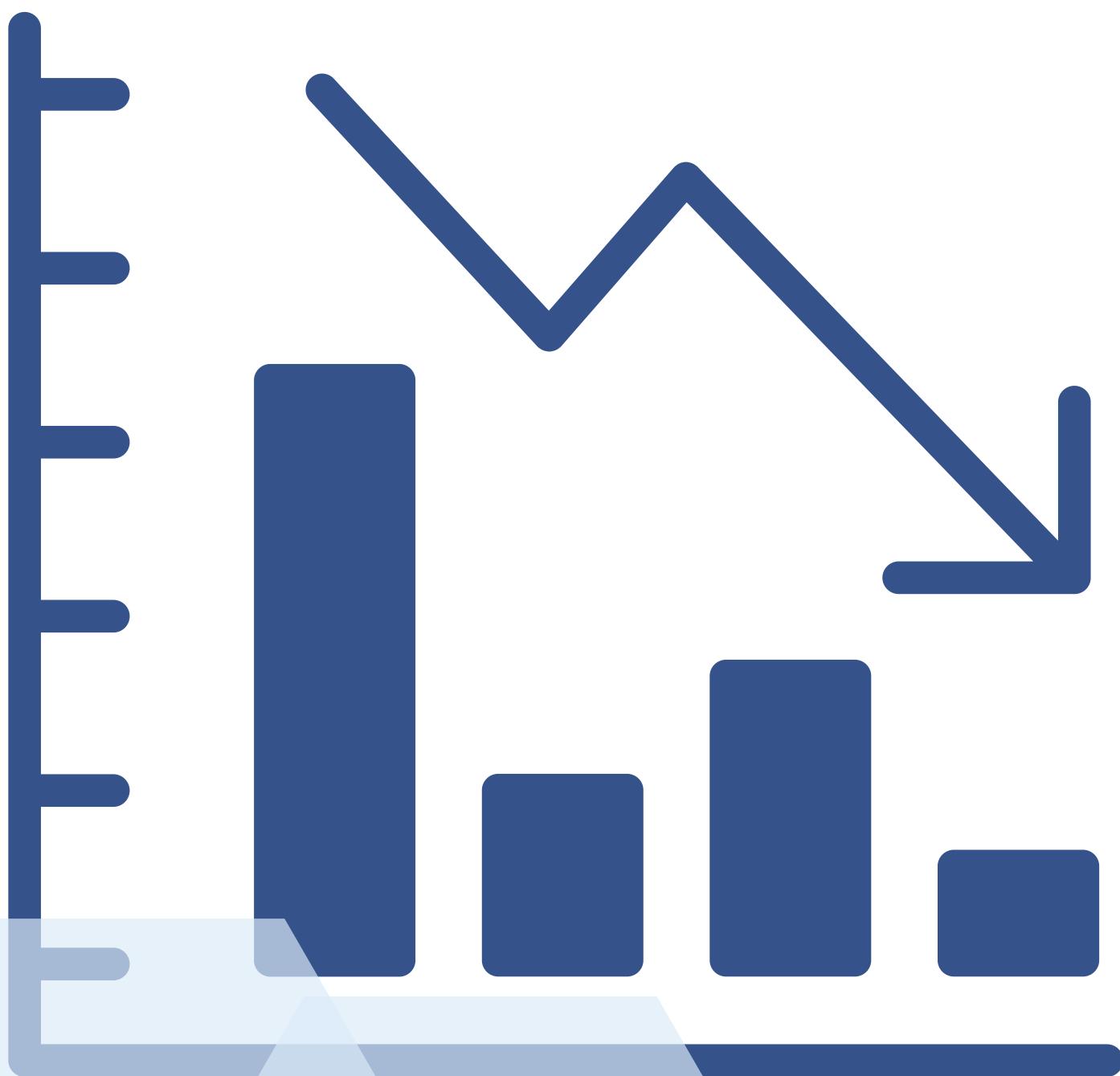


Tính khoảng cách Euclidean

So sánh khoảng cách của vector FFT
của tín hiệu kiểm thử so với các
vector đặc trưng

Tìm khoảng cách nhỏ nhất

So khớp tín hiệu và đưa ra kết quả
nhận dạng nguyên âm



KẾT QUẢ & ĐÁNH GIÁ THUẬT TOÁN FFT

MISSION

Khảo sát mức độ ảnh hưởng của các tham số này lên kết quả nhận dạng. Đưa ra kết quả



Kiểm tra độ chính xác trên các tham số

FFT_POINTS= 512

FFT_POINTS	FRAMES_MODES	WINDOW_FUNCTIONS	Độ chính xác
512	(0.04, 0.02)	hamming	83.81
512	(0.04, 0.02)	bartlett	83.81
512	(0.04, 0.02)	blackman	84.76
512	(0.04, 0.02)	kaiser	86.67
512	(0.04, 0.02)	hann	83.81
512	(0.03, 0.015)	hamming	84.76
512	(0.03, 0.015)	bartlett	84.76
512	(0.03, 0.015)	blackman	88.57
512	(0.03, 0.015)	kaiser	88.57
512	(0.03, 0.015)	hann	85.71
512	(0.02, 0.01)	hamming	89.52
512	(0.02, 0.01)	bartlett	88.57
512	(0.02, 0.01)	blackman	88.57
512	(0.02, 0.01)	kaiser	90.48
512	(0.02, 0.01)	hann	89.52

FFT_POINTS= 1024

FFT_POINTS	FRAMES_MODES	WINDOW_FUNCTIONS	Độ chính xác
1024	(0.04, 0.02)	hamming	80.0
1024	(0.04, 0.02)	bartlett	80.0
1024	(0.04, 0.02)	blackman	84.76
1024	(0.04, 0.02)	kaiser	86.67
1024	(0.04, 0.02)	hann	81.9
1024	(0.03, 0.015)	hamming	84.76
1024	(0.03, 0.015)	bartlett	84.76
1024	(0.03, 0.015)	blackman	87.62
1024	(0.03, 0.015)	kaiser	87.62
1024	(0.03, 0.015)	hann	85.71
1024	(0.02, 0.01)	hamming	89.52
1024	(0.02, 0.01)	bartlett	89.52
1024	(0.02, 0.01)	blackman	89.52
1024	(0.02, 0.01)	kaiser	90.48
1024	(0.02, 0.01)	hann	89.52

FFT_POINTS= 2048

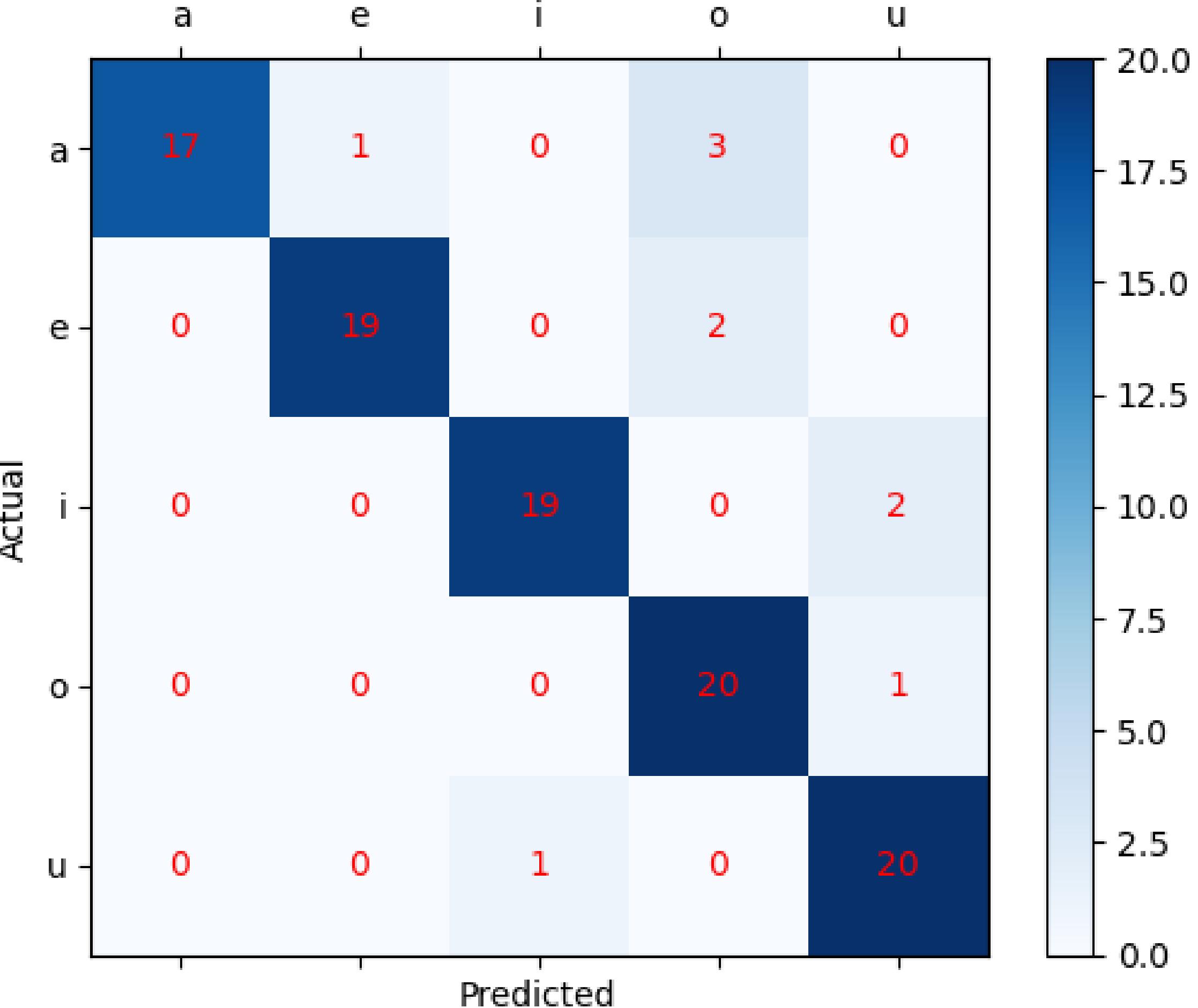
FFT_POINTS	FRAMES_MODES	WINDOW_FUNCTIONS	Độ chính xác
2048	(0.04, 0.02)	hamming	80.95
2048	(0.04, 0.02)	bartlett	80.95
2048	(0.04, 0.02)	blackman	84.76
2048	(0.04, 0.02)	kaiser	86.67
2048	(0.04, 0.02)	hann	81.9
2048	(0.03, 0.015)	hamming	83.81
2048	(0.03, 0.015)	bartlett	84.76
2048	(0.03, 0.015)	blackman	87.62
2048	(0.03, 0.015)	kaiser	88.57
2048	(0.03, 0.015)	hann	84.76
2048	(0.02, 0.01)	hamming	88.57
2048	(0.02, 0.01)	bartlett	89.52
2048	(0.02, 0.01)	blackman	89.52
2048	(0.02, 0.01)	kaiser	90.48
2048	(0.02, 0.01)	hann	89.52



Team 14

KẾT QUẢ MA TRẬN NHÂM LẦN VỚI ĐỘ CHÍNH XÁC CAO NHẤT

Confusion Matrix





Team 14

KẾT QUẢ ĐÚNG SAI TRÊN TẬP DỮ LIỆU KIỂM THỬ

Speaker	Validation Matrix				
	a	e	i	o	u
01MDA -	1	1	1	1	1
02FVA -	1	0	1	1	1
03MAB -	1	1	1	1	0
04MHB -	1	1	1	1	1
05MVB -	0	1	1	1	1
06FTB -	1	1	1	1	1
07FTC -	1	1	1	1	1
08MLD -	1	1	1	1	1
09MPD -	0	1	1	1	1
10MSD -	1	1	1	1	1
11MVD -	1	1	1	1	1
12FTD -	1	0	0	1	1
14FHH -	1	1	1	1	1
15MMH -	1	1	1	1	1
16FTH -	0	1	1	1	1
17MTH -	1	1	1	1	1
18MNK -	1	1	1	1	1
19MXK -	1	1	1	1	1
20MVK -	0	1	1	0	1
21MTL -	1	1	1	1	1
22MHL -	1	1	0	1	1



BÀI 3

NHẬN DẠNG NGUYÊN ÂM KHÔNG PHỤ THUỘC NGƯỜI NÓI DÙNG ĐẶC TRƯNG PHỔ MFCC VÀ PHƯƠNG PHÁP K-MEAN



SƠ ĐỒ THUẬT TOÁN MFCC ÁP DỤNG PHÂN CỤM KMEANS

5-Step Ordering Process

1

Phân đoạn tín hiệu dựa trên đặc trưng STE

2

Trích xuất MFCC của 1 khung tín hiệu

3

Tính tất cả các vector MFCC của các khung ổn định cho 1 nguyên âm của 21 người

4

Tính K vector đặc trưng cho 1 nguyên âm của nhiều người nói

5

Tổng hợp vector đặc trưng và tiến hành so khớp, nhận dạng



Team 14

CODE TRÍCH XUẤT VECTOR MFCC

```
extract-MFCC-vector.py

def extract_vectorMFCC(audio_signal, sample_rate):
    """
    Extract MFCC feature vectors from an audio signal.

    Parameters:
    - audio_signal (np.ndarray): Audio signal.
    - sample_rate (int): Sample rate.

    Returns:
    - np.ndarray: MFCC feature vectors.
    """
    frame_shift = int(FRAME_SHIFT * sample_rate)
    frame_len = int(FRAME_SIZE * sample_rate)
    frame_num = (len(audio_signal) - frame_len) // frame_shift + 1

    hamming_window = hamming(frame_len)
    mfcc_vectors = []

    for i in range(frame_num):
        start_index = i * frame_shift
        end_index = start_index + frame_len
        frame = audio_signal[start_index:end_index]
        windowed_frame = frame * hamming_window

        # Compute MFCC for the frame
        mfcc_vectors.append(librosa.feature.mfcc(y=windowed_frame, n_fft=512, sr=sample_rate, n_mfcc=MFCC_POINTS).T)

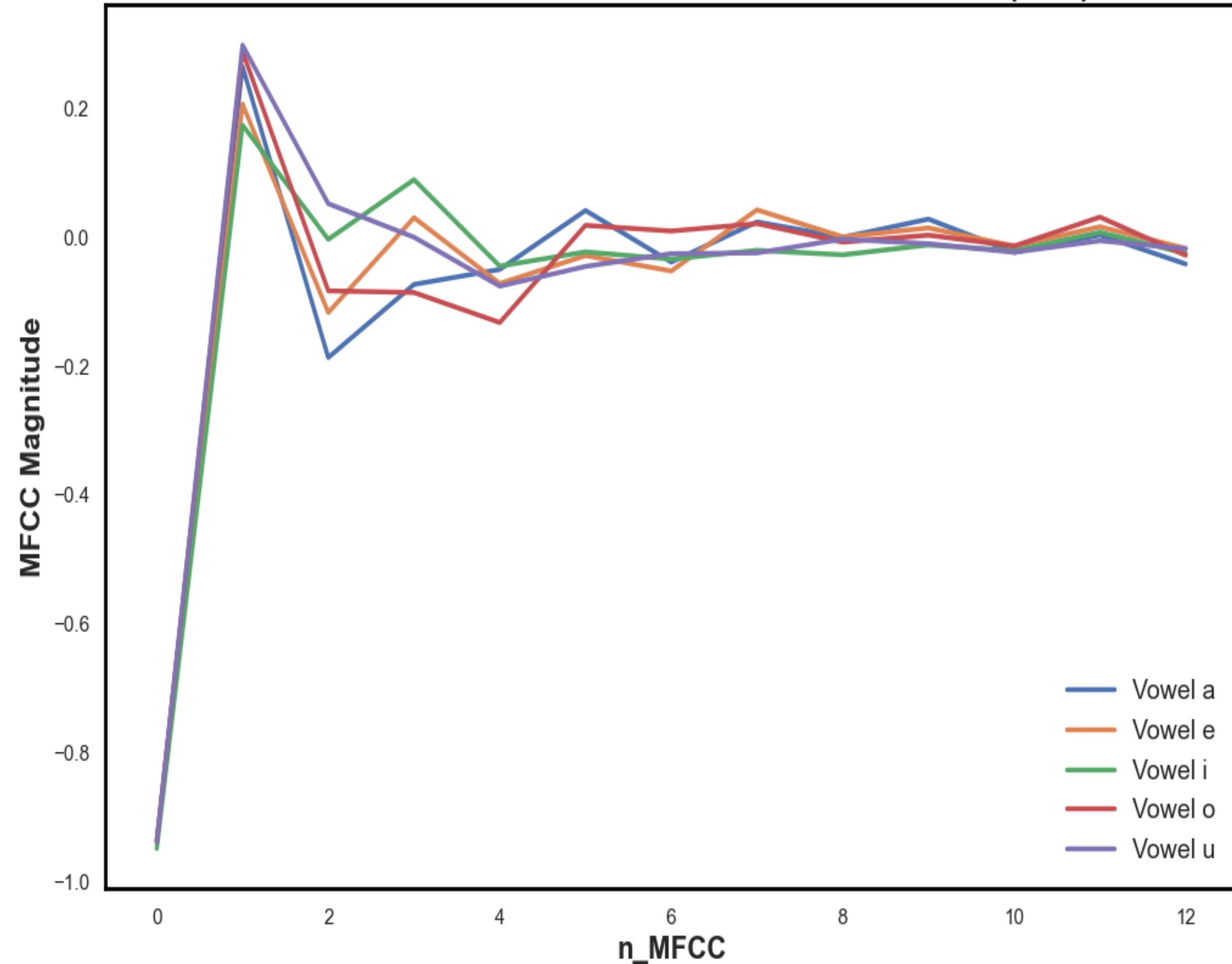
    mfcc_array = np.array(mfcc_vectors)
    normalized_array = mfcc_array / np.max(np.abs(mfcc_array))
    return normalized_array
```



Team 14

TRÍCH XUẤT VECTOR MFCC ĐẶC TRƯNG 5 NGUYÊN ÂM (K=1)

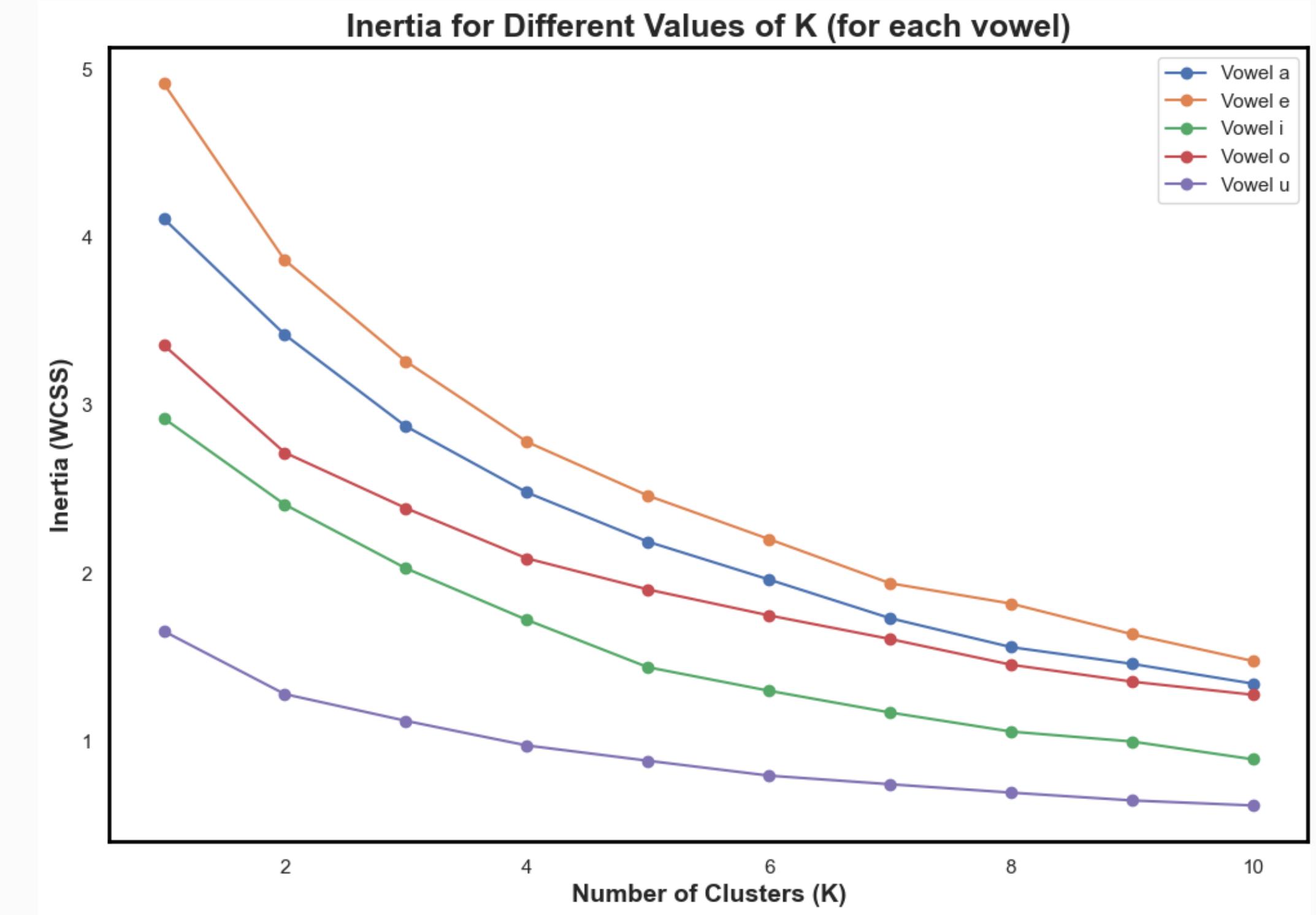
Mean of MFCC Feature Vectors for Each Vowel (K=1)





Team 14

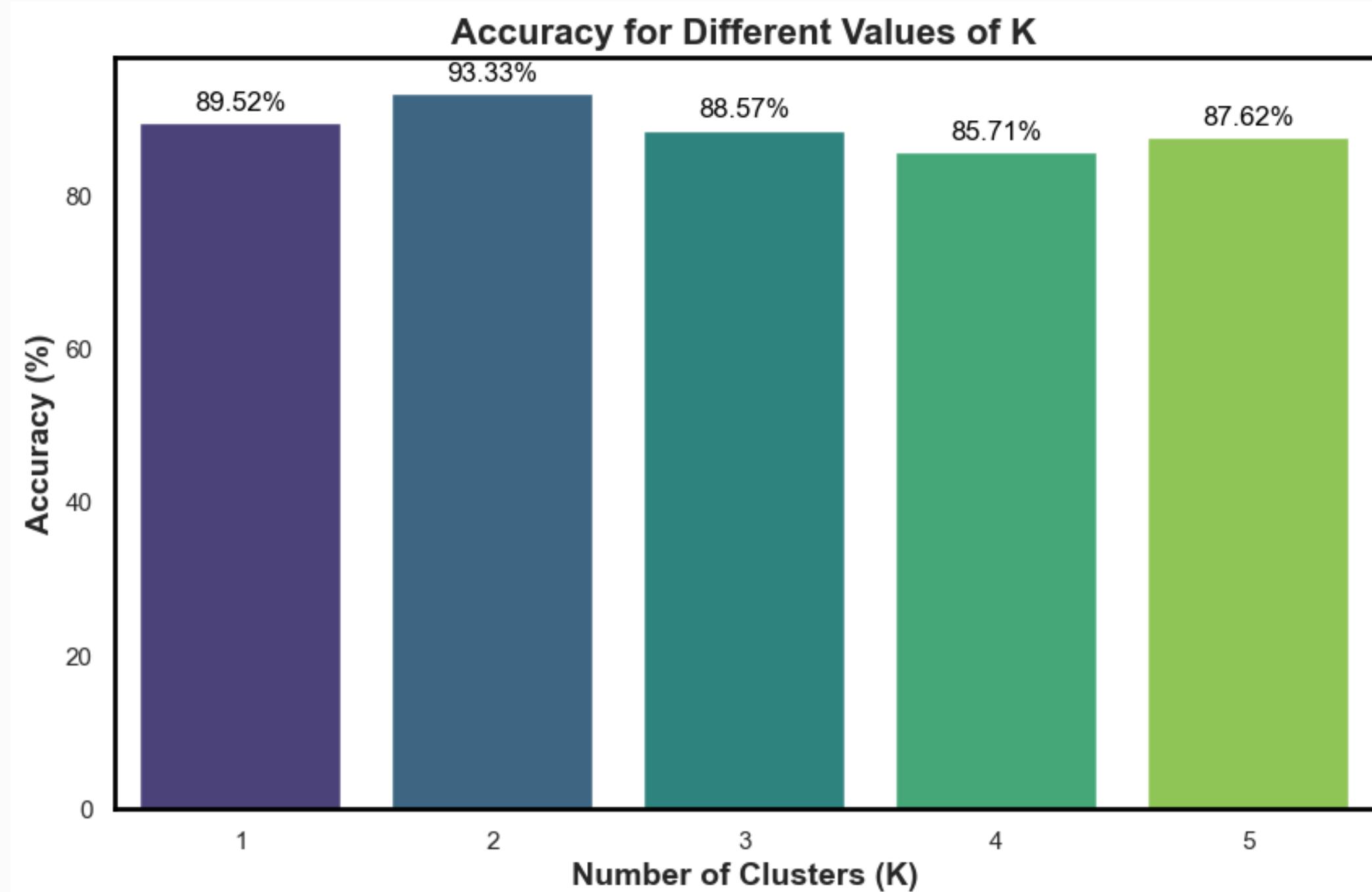
PHƯƠNG PHÁP ELBOW ĐỂ XÁC ĐỊNH GIÁ TRỊ TỐI ƯU CỦA K





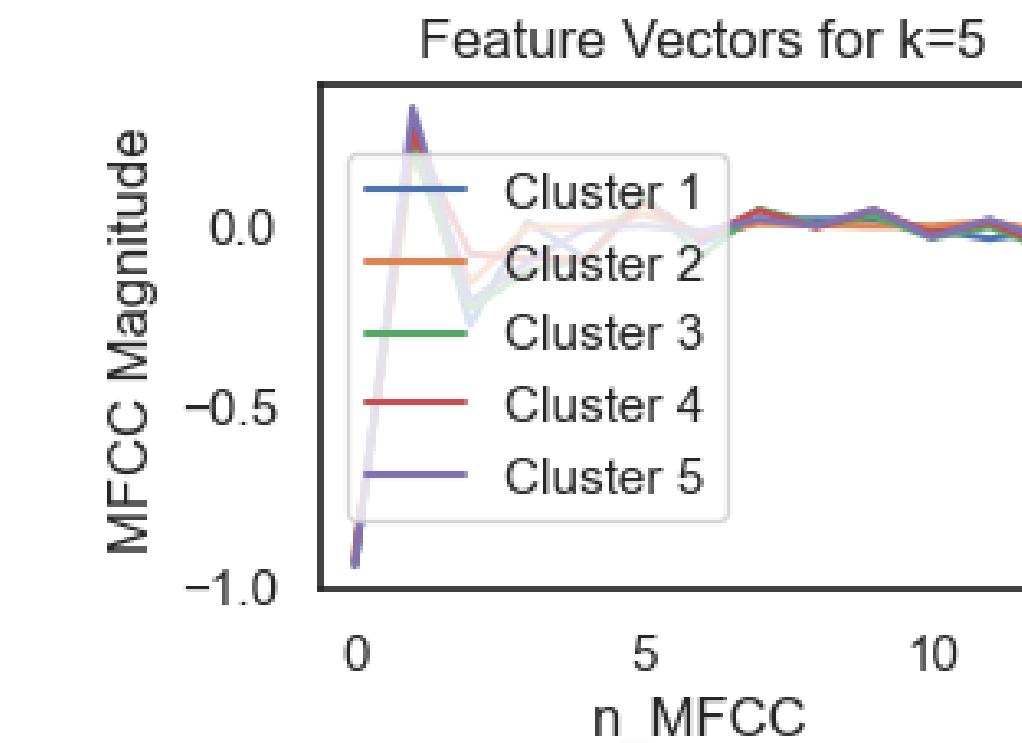
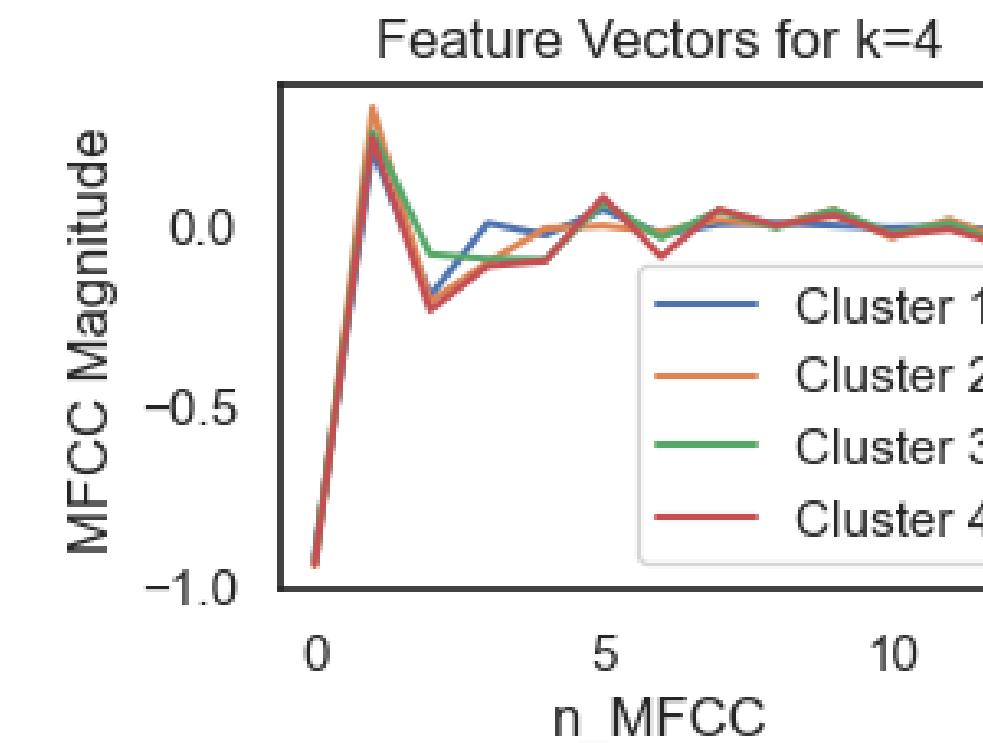
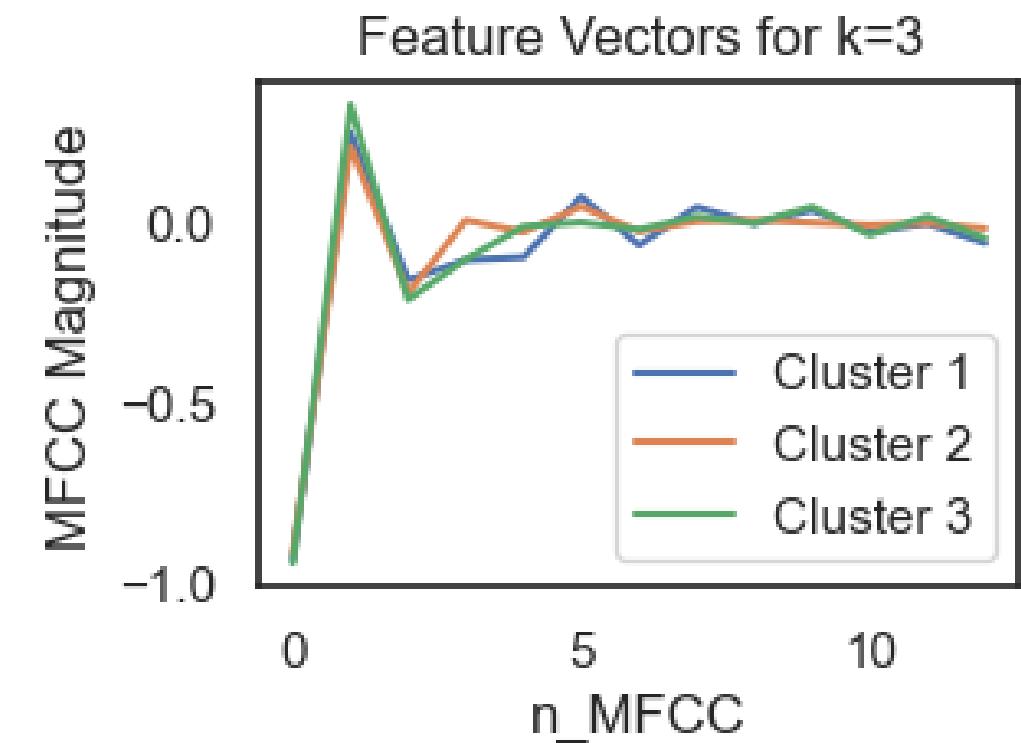
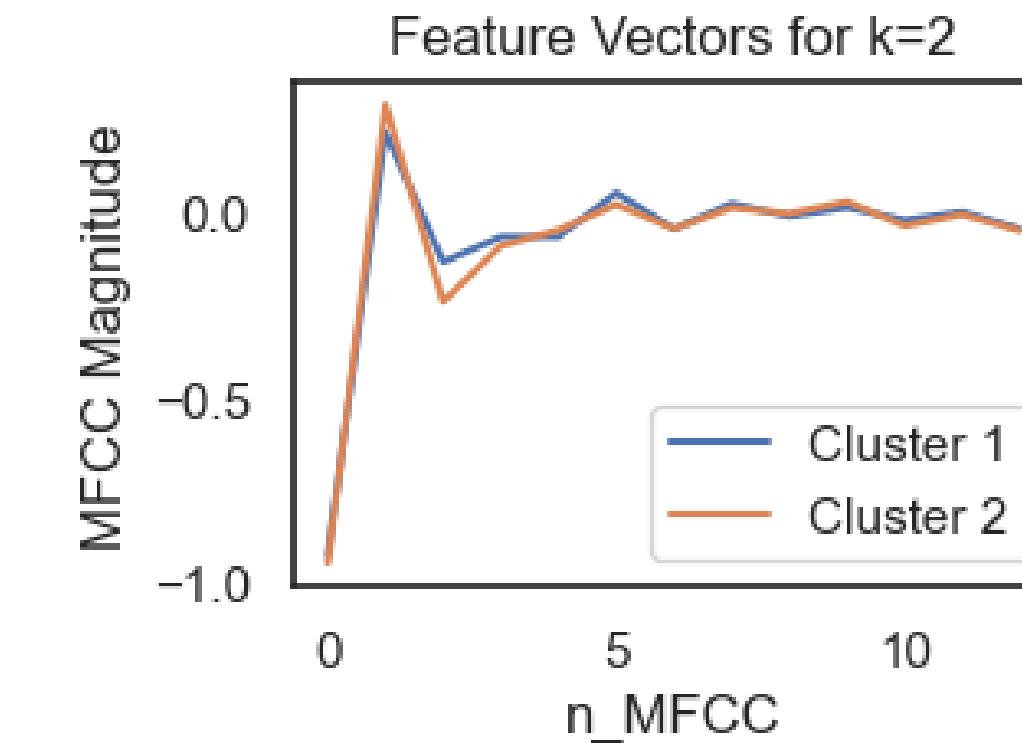
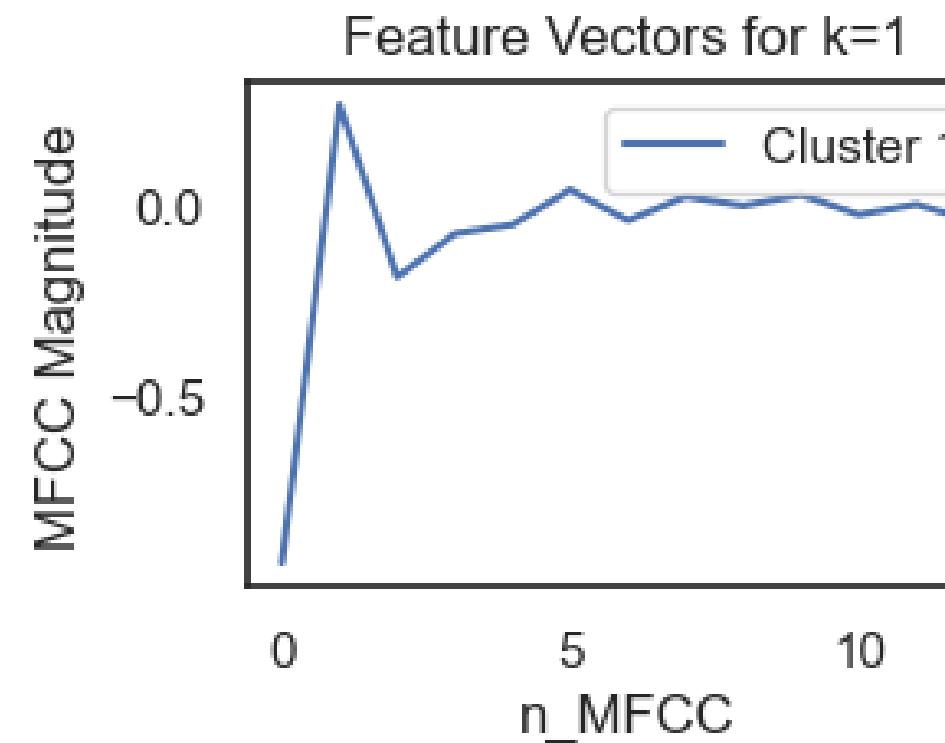
Team 14

KHẢO SÁT ĐỘ CHÍNH XÁC VỚI CÁC GIÁ TRỊ K=1,2,3,4,5





CÁC PHÂN CỤM VECTOR ĐẶC TRƯNG CỦA NGUYÊN ÂM ‘a’





Team 14

MA TRẬN ĐÚNG SAI VỚI GIÁ TRỊ TỐI ƯU K=2

Validation Matrix

Speaker	a	e	i	o	u
01MDA	1	1	1	1	1
02FVA	1	1	0 e	1	1
03MAB	1	1	1	1	0 e
04MHB	1	1	1	1	1
05MVB	1	1	1	1	1
06FTB	1	1	1	1	1
07FTC	1	1	1	1	1
08MLD	1	1	1	1	1
09MPD	0 o	1	1	1	1
10MSD	1	1	1	1	0 i
11MVD	1	1	1	1	1
12FTD	1	1	1	1	1
14FHH	1	1	1	1	1
15MMH	1	1	1	1	1
16FTH	0 e	1	1	1	1
17MTH	1	1	1	1	1
18MNK	1	1	1	1	1
19MXK	1	1	1	1	1
20MVK	1	1	1	0 u	1
21MTL	1	1	1	1	1
22MHL	1	1	0 u	1	1



Team 14

MA TRẬN NHÂM LÃN VỚI GIÁ TRỊ TỐI ƯU K=2

'A'  90,5%

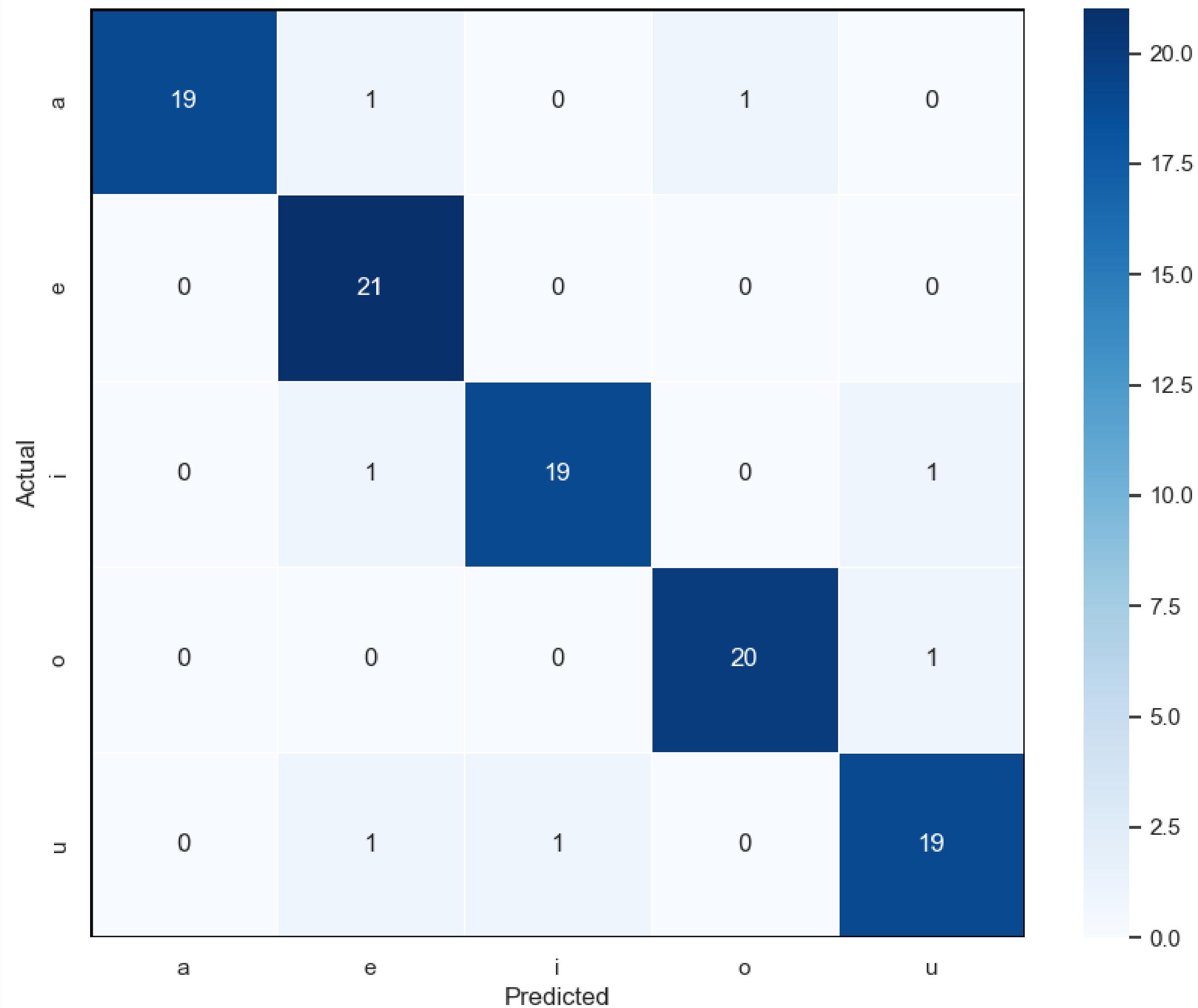
'E'  100%

'I'  90,5%

'O'  95,2%

'U'  90,5%

Confusion Matrix





KHẢO SÁT ĐỘ CHÍNH XÁC DỰA TRÊN PHỤ THUỘC THAM SỐ ĐẦU VÀO

WINDOW_FUNCTIONS	(frame_size, frame_shift))	K-means clustering	Accuracy
hamming	(0.03, 0.015)	2	0.933
hamming	(0.02, 0.01)	2	0.848
hamming	(0.03, 0.015)	3	0.886
hamming	(0.02, 0.01)	3	0.876
hamming	(0.03, 0.015)	4	0.857
hamming	(0.02, 0.01)	4	0.886
hamming	(0.03, 0.015)	5	0.876
hamming	(0.02, 0.01)	5	0.848
bartlett	(0.03, 0.015)	2	0.895
bartlett	(0.02, 0.01)	2	0.876
bartlett	(0.03, 0.015)	3	0.838
bartlett	(0.02, 0.01)	3	0.838
bartlett	(0.03, 0.015)	4	0.886
bartlett	(0.02, 0.01)	4	0.838
bartlett	(0.03, 0.015)	5	0.857
bartlett	(0.02, 0.01)	5	0.876
kaiser	(0.03, 0.015)	2	0.886
kaiser	(0.02, 0.01)	2	0.829
kaiser	(0.03, 0.015)	3	0.848
kaiser	(0.02, 0.01)	3	0.838
kaiser	(0.03, 0.015)	4	0.857
kaiser	(0.02, 0.01)	4	0.867
kaiser	(0.03, 0.015)	5	0.829
kaiser	(0.02, 0.01)	5	0.914



Team 14

NHẬN XÉT CHUNG





BẢNG THỐNG KÊ ĐỘ CHÍNH XÁC DÙNG ĐẶC TRƯNG PHỔ FFT

Đặc trưng phổ	N_FFT (Số chiều)	Độ chính xác (%)
FFT	512	90.48%
	1024	90.48%
	2048	90.48%



BẢNG THỐNG KÊ ĐỘ CHÍNH XÁC DÙNG ĐẶC TRƯNG PHỔ MFCC VÀ KMEAN

Đặc trưng phổ	Số cụm K	Độ chính xác (%)				
		1	2	3	4	5
MFCC $N_{MFCC} = 13$ $K= 2$	89.52%	93.33%	88.57%	85.71%	87.62%	



NHẬN XÉT CHUNG

- Độ chính xác khi dùng đặc trưng phổ MFCC nhìn chung tốt hơn FFT
- Tăng số chiều N_FFT không thay đổi độ chính xác thuật toán
- Tương tự, khi thay đổi số phân cụm K-mean thì độ chính xác thuật toán không tăng tuyến tính
- Không phải lúc nào cửa sổ hamming cũng cho ra kết quả tốt nhất.





Team 14

CẢM ƠN

Cảm ơn thầy Duy và các bạn đã
chú ý lắng nghe <3 <3 <3



nkduy@dut.udn.vn



<http://scv.udn.vn/nkduy>



0935-043-201

