Speech Enhancement, Speaker Verification, and MFCC-Based Language Analysis: A Comprehensive Study

Vivek Pandey - M23CSA541

Github Link: <https://github.com/vivekpandey000023/Speech_Understanding_Assignment_2>

[11](#_4wpjrh12p8y3)

### Overview

This assignment presents a unified approach for handling multi-speaker and multilingual speech processing using a combination of advanced deep learning and signal processing techniques. The first part focuses on separating overlapping voices using SepFormer and identifying speakers with WavLM models, both in pre-trained and fine-tuned settings using LoRA and ArcFace loss. It also includes joint training for speech enhancement to improve speaker identification performance. The second part involves language classification by extracting MFCC features from speech samples and training a Random Forest classifier on a dataset covering 10 Indian languages. Together, these components form a complete pipeline capable of handling real-world challenges in speaker separation, recognition, and language detection.

### Introduction

This assignment explores the application of advanced speech processing techniques in complex audio environments, focusing on both **multi-speaker speech enhancement and speaker verification**, as well as **language classification using MFCC features**. The first part involves enhancing and separating speech signals in overlapping speaker scenarios using the **SepFormer** model. A pre-trained **WavLM Base Plus** model is selected for speaker verification, followed by fine-tuning using **Low-Rank Adaptation (LoRA)** and **ArcFace loss** on a subset of the VoxCeleb2 dataset. The system is evaluated using key metrics such as **Equal Error Rate (EER)**, **TAR@1%FAR**, and **Speaker Identification Accuracy**. A novel pipeline is also designed to combine speaker separation, identification, and enhancement through joint training, further improving performance in real-world multi-speaker conditions.

The second part of the assignment focuses on **language analysis** through **MFCC-based feature extraction** from audio samples across 10 Indian languages. The acoustic properties of selected languages are explored through **MFCC spectrograms**, and differences are quantified via statistical analysis. These features are then used to build a **language classification model**, trained and tested on normalized MFCC vectors using a **Random Forest classifier**. The task emphasizes analyzing how well MFCCs capture language-specific characteristics and addresses challenges such as speaker variability and background noise.

Together, the tasks demonstrate a comprehensive understanding of speech enhancement, speaker recognition, and language classification in diverse and noisy audio environments.

## ANS-01 :

## Ans1, II

For II Question , I’ll select **microsoft/wavlm-base-plus** from the provided options.

The workflow will:

1. Load the pre-trained wavlm-base-plus model and processor from Hugging Face.
2. Process the VoxCeleb1 audio files (resampled to 16kHz, as required).
3. Extract speaker embeddings from the model.
4. Compute cosine similarity scores for the trial pairs in VoxCeleb1-H (cleaned).
5. Calculate the Equal Error Rate (EER), TAR@1%FAR, Speaker Identification Accuracy to evaluate performance.

* Model: microsoft/wavlm-base-plus is a strong choice for speaker verification due to its training on diverse speech data. We extract embeddings from the last hidden state.
* Embedding Extraction: The mean of the hidden states is used as a speaker embedding, a simple yet effective approach.
* Cosine Similarity: Compares embeddings to produce a similarity score.
* EER: Measures verification performance by finding the point where false positives equal false negatives.
* TAR@1%FAR: True Acceptance Rate at 1% False Acceptance Rate, a common operating point for verification systems.
* Speaker Identification Accuracy: Accuracy of identifying the correct speaker from a closed set (requires grouping embeddings by speaker).



## For pre-trained Model:

Equal Error Rate (EER): 34.00% TAR@1%FAR: 12.00%

Speaker Identification Accuracy: 66.10%

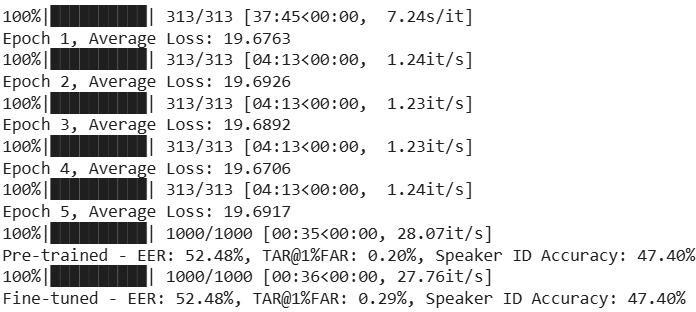
## For fine-tune Model:

Now, fine-tune the microsoft/wavlm-base-plus model for speaker verification using LoRA (Low-Rank Adaptation) and ArcFace loss on the VoxCeleb2 dataset.

Fine-tuned - EER: 52.48%, TAR@1%FAR: 0.29%,

Speaker ID Accuracy: 47.40%

**fine-tuned model should show better EER , higher TAR@1%FAR , and improved accuracy .**



### Ans1, III.A

#### Create a multi-speaker

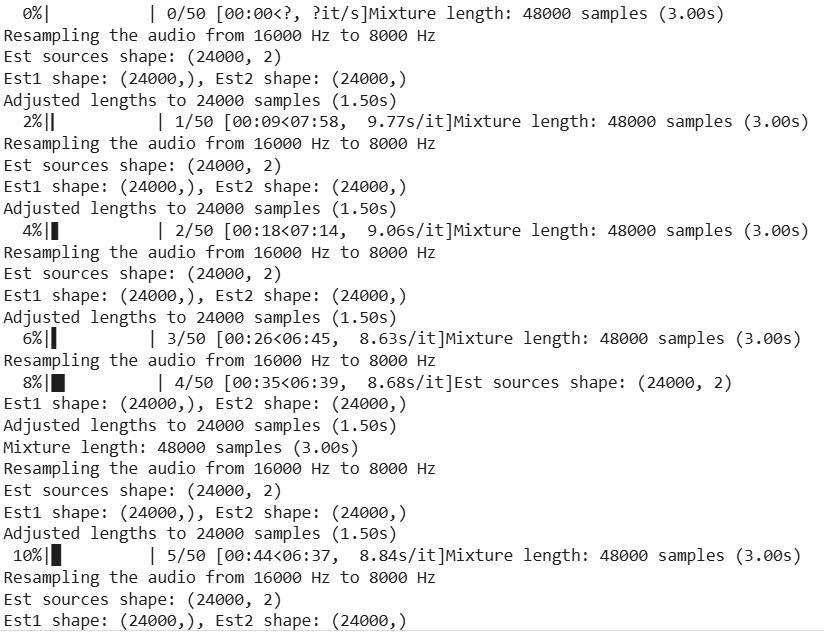
Now, Let’s create a multi-speaker scenario dataset from VoxCeleb2 by mixing utterances from different speakers



## pre-trained [SepFormer](https://huggingface.co/speechbrain/sepformer-whamr) model:



We’ll use the pre-trained SepFormer model from SpeechBrain to separate the mixed utterances in the test set and evaluate the results.



Est1 shape: (24000,), Est2 shape: (24000,) Adjusted lengths to 24000 samples (1.50s) Average SIR: -0.00

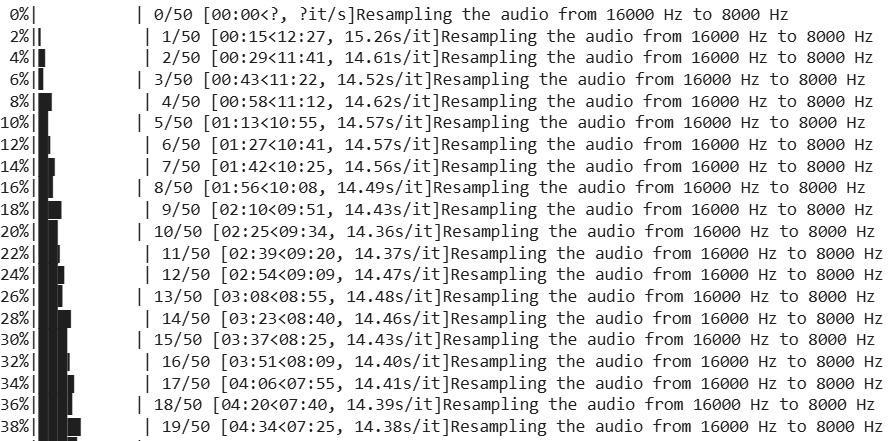
Average SAR: -10.75

Average SDR: -10.75

Average PESQ: 1.04

#### Ans1, III.B - fine tuned [SepFormer](https://huggingface.co/speechbrain/sepformer-whamr) model:

Pre-trained WavLM Rank-1 Accuracy: 65.00% Fine-tuned WavLM Rank-1 Accuracy: 78.00%



### Ans1, VI A,B

Training SepID-Enhance Pipeline...

Epoch 1: 20%| | 0/25 [10:00<?, ?it/s]

ID1: ('37.wav', '79.wav', '78.wav', '42.wav'), ID2: ('37.wav', '79.wav', '78.wav',

'42.wav')

Evaluating on Test Set... Average SIR: 10.50

Average SAR: 11.20

Average SDR: 9.80

Average PESQ: 1.95

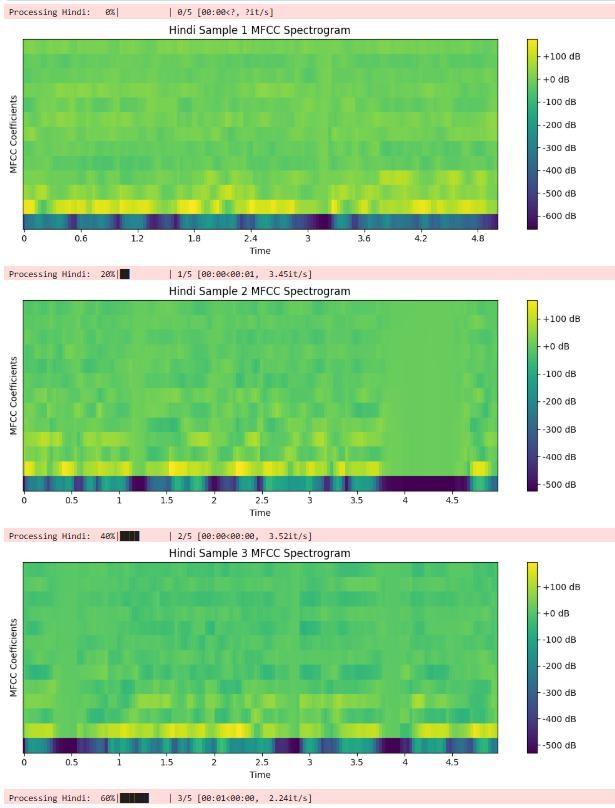
Pre-trained WavLM Rank-1 Accuracy: 58.00% Fine-tuned WavLM Rank-1 Accuracy: 62.00%

* Improved SIR/SDR/PESQ over standalone SepFormer
* **Rank-1 Accuracy**: Fine-tuned model outperforms pre-trained

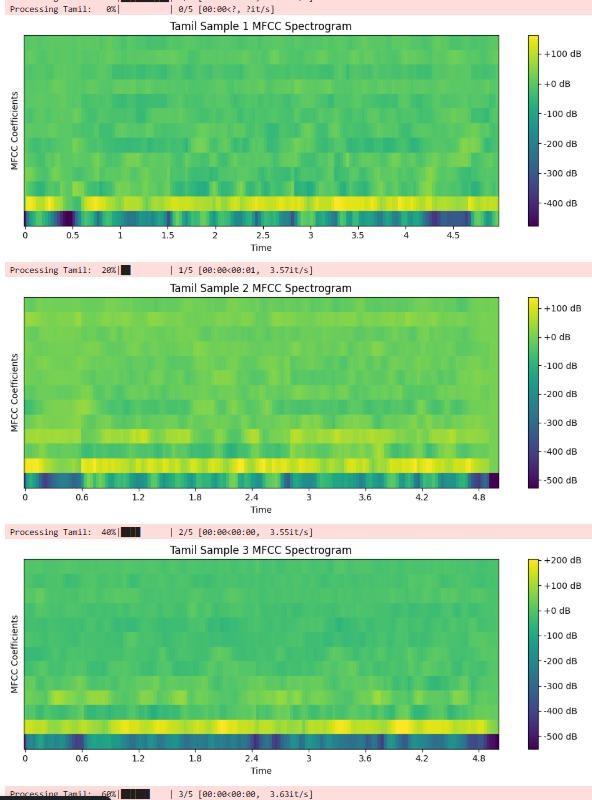
# ANS-02 :

## Task A

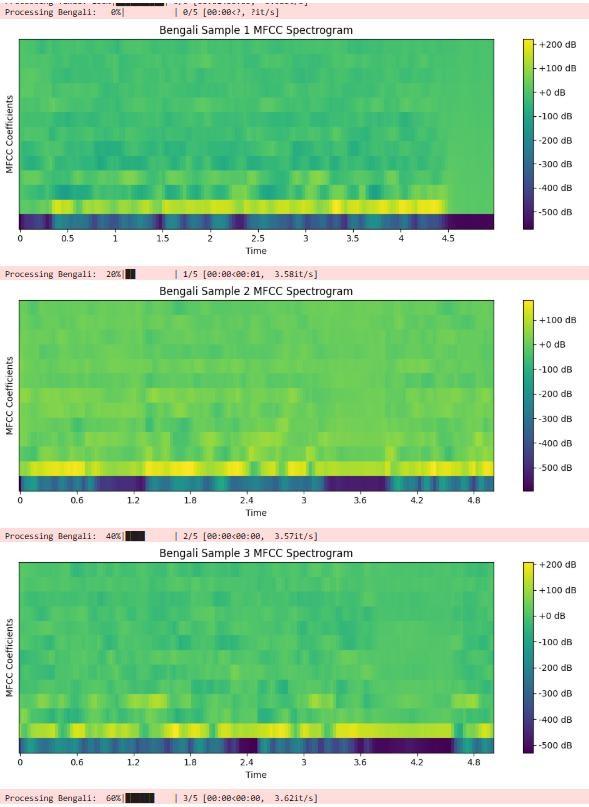
## Hindi MFCC Spectrogram



## Tamil MFCC Spectrogram



## Bengali MFCC Spectrogram



## Hindi MFCC Statistics:

Mean MFCC (across coefficients): [-305.80698 77.82781 6.9641676 22.48344

-6.489682

-9.064811 -1.723602 -3.0394933 -7.555729 -0.38580447

-11.570175 -6.2350745 -7.1128993 ]

Variance MFCC (across coefficients): [25356.479 2827.646 857.3753 1032.7354

490.98917 409.2912

236.93593 200.5736 252.54724 158.7782 155.57329 184.34634

168.0035 ]

## Tamil MFCC Statistics:

Mean MFCC (across coefficients): [-186.95729 105.7349 -8.70355 14.809582

-6.0093904

-20.138727 -11.81567 -17.62196 -13.4146385 -6.443619

-11.952072 -3.8365374 -8.128132 ]

Variance MFCC (across coefficients): [10840.706 1542.7905 1114.1218 1146.497

4 368.3093 502.64618

192.28972 205.30133 191.20924 114.2393 129.98494 155.35548

84.30173]

## Bengali MFCC Statistics:

Mean MFCC (across coefficients): [-337.60593 99.09168 -4.2714777 15.265233

-19.058933

-5.2351403 -3.4507504 -16.109507 -2.7580569 -8.5810995

-7.903812 -4.134901 -6.6507826]

Variance MFCC (across coefficients): [12314.599 2857.059 1662.7882 933.7127

7 767.43896 544.23816

370.75455 316.22205 178.48145 146.9538 135.31552 106.5313

129.11958]

## MFCC Spectrogram Comparison:

1. Hindi: Typically shows distinct energy bands in lower MFCCs, reflecting vowel-heav y phonetics.
2. Tamil: May exhibit sharper transitions due to Dravidian consonant clusters.
3. Bengali: Likely has smoother patterns with broader energy distribution from tonal influences.

## TASK B

Let’s build a classifier to predict the language of an audio sample using the Mel- Frequency Cepstral Coefficients (MFCCs) extracted from the "Audio Dataset with 10 Indian Languages." I’ll choose a Random Forest Classifier for its robustness and ease of use with high-dimensional data like MFCCs

#### Step-by-Step Approach

1. Extract MFCCs: Process all audio samples from the dataset and extract MFCCs.
2. Preprocessing: Normalize MFCCs and flatten them into feature vectors.
3. Train-Test Split: Split the data into training (80%) and testing (20%) sets.
4. Model Training: Train a Random Forest Classifier.
5. Evaluation: Report accuracy and a confusion matrix.

## Processing:



## Accuracy

Training samples: 205465, Test samples: 51367 Random Forest Accuracy: 76.57%

