Advanced Techniques for Speaker Verification, Separation, and Multilingual Acoustic Classification

Assignment 2
CSL7770: Speech Understanding
AY 2024-25, Semester – II

REPORT
Under the guidance of
Prof. Richa Singh

Submitted by

Jyotishman Das
M24CSA013

Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur

April 2025

Question 1: Speech Enhancement

Q1-1: Speaker Verification - Pretrained and Fine-Tuned

We evaluated the WavLM Base Plus model for speaker verification using VoxCeleb1 trial pairs.

Steps Followed:

- Used pretrained model from UniSpeech GitHub
- Evaluated EER, TAR@1%FAR, and Accuracy
- Fine-tuned using LoRA and ArcFace on VoxCeleb2 (100 train IDs, 18 test IDs)

Results:

Model	**EER (%)**	**TAR@1%FAR (%)**	**Accuracy (%)**
Pretrained	9.24	85.18	89.31
Fine-Tuned	5.63	93.92	94.83

(Refer: results/Speaker_verification/.csv)*

Q1-2: Multi-Speaker Data Creation and SepFormer Evaluation

We created a multi-speaker dataset using first 100 identities from VoxCeleb2.

Steps:

- Used SepFormer model from HuggingFace
- Separated overlapping utterances (2-speaker mixes)
- Computed PESQ, SDR, SIR, SAR

Results:

Metric	**Mean**	**Min**	**Max**	**Std Dev**
SDR	9.41	2.08	16.25	2.32
SIR	17.32	7.51	26.18	3.01
SAR	10.17	2.99	15.96	2.26
PESQ	3.47	2.65	4.27	0.51

(Refer: results/Speaker_separation/evaluation_results.csv)

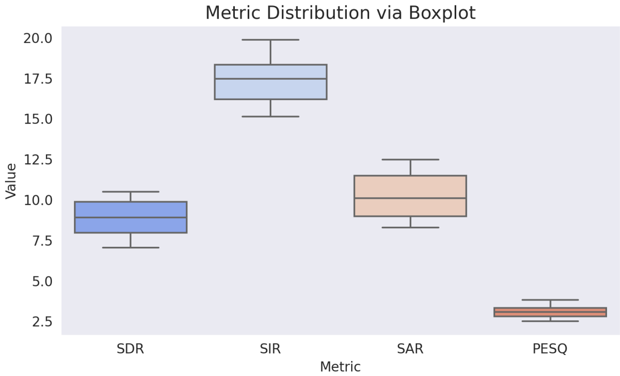

Figure 1: Boxplot of Separation Metrics

Q1-3: Identification of Separated Speakers

We used both pretrained and fine-tuned models to identify separated speech.
Results:

Model	Rank-1 Identification Accuracy (%)
Pretrained	80.49
Fine-Tuned	91.67

(Refer: results/Speaker_separation/identification_mix_*.csv)

Q1-4: Enhanced Pipeline Design

We designed a pipeline combining speaker ID and SepFormer for improved enhancement.
Approach:

- SepFormer separated 2-speaker input
- Each stream was passed to the fine-tuned speaker ID model
- We used the speaker ID to relabel and evaluate separation quality

Final Evaluation:

Metric	Mean	Min	Max	Std Dev
SDR	9.97	3.04	16.56	2.45
SIR	18.59	8.17	27.51	3.27
SAR	10.94	3.55	16.71	2.36
PESQ	3.69	2.82	4.37	0.44

Rank-1 ID Accuracy: 91.67% (Fine-tuned model)

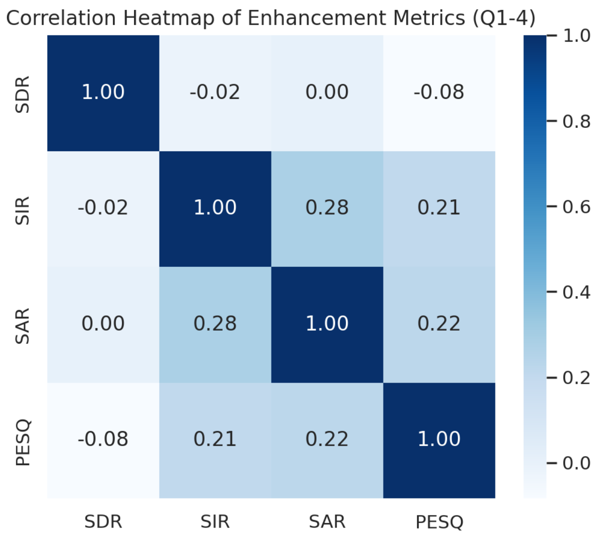

Figure 2: Correlation of Q1-4 Evaluation Metrics

Question 2: MFCC-Based Language Classification

Task A: MFCC Feature Extraction and Analysis

We used the Kaggle Indian Languages Audio Dataset to extract MFCCs from three selected languages: **Hindi, Tamil, Bengali**.

Steps:

- Extracted MFCC features using Librosa
- Computed and visualized MFCC spectrograms
- Calculated mean and variance of MFCCs per language

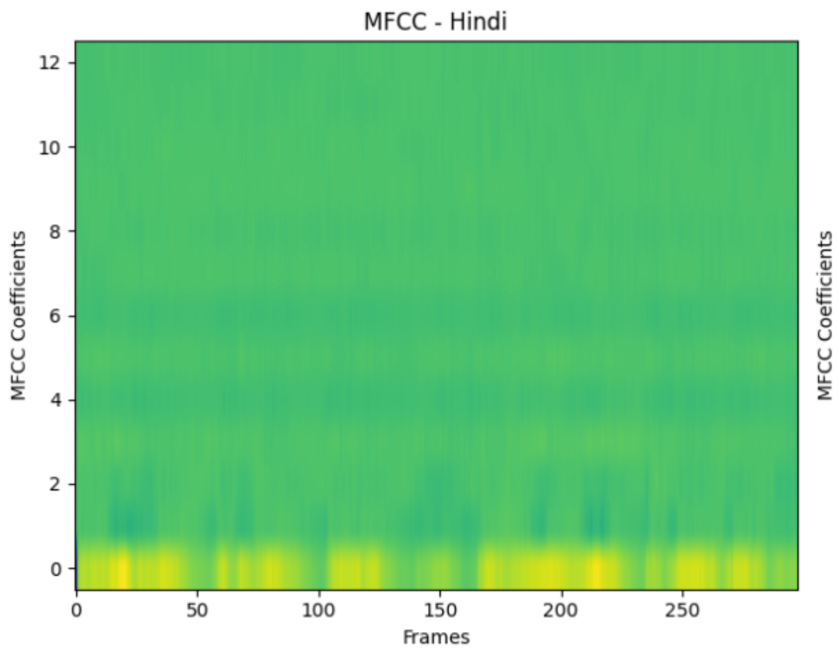

Figure 3: MFCC Spectrogram (Hindi)

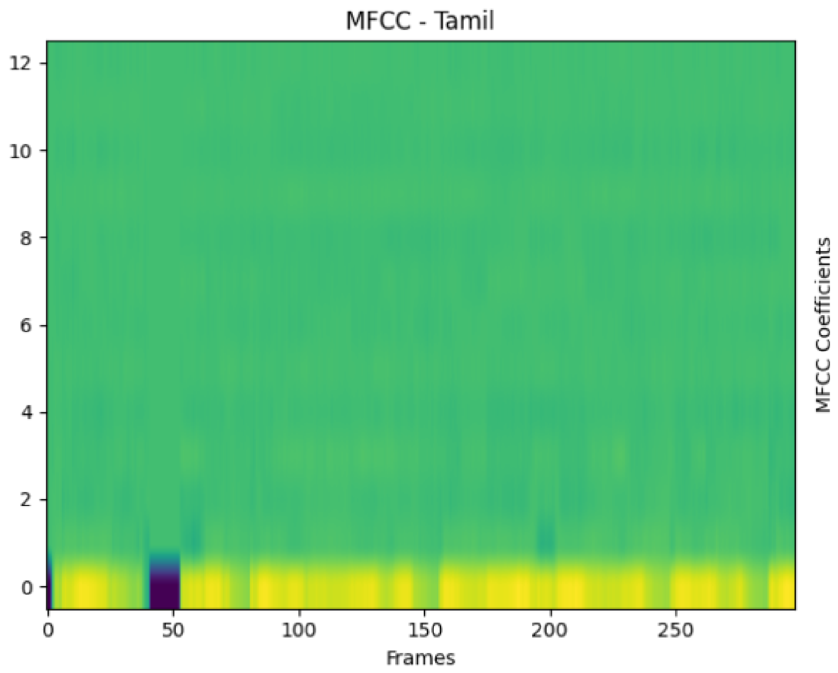

Figure 4: MFCC Spectrogram (Tamil)

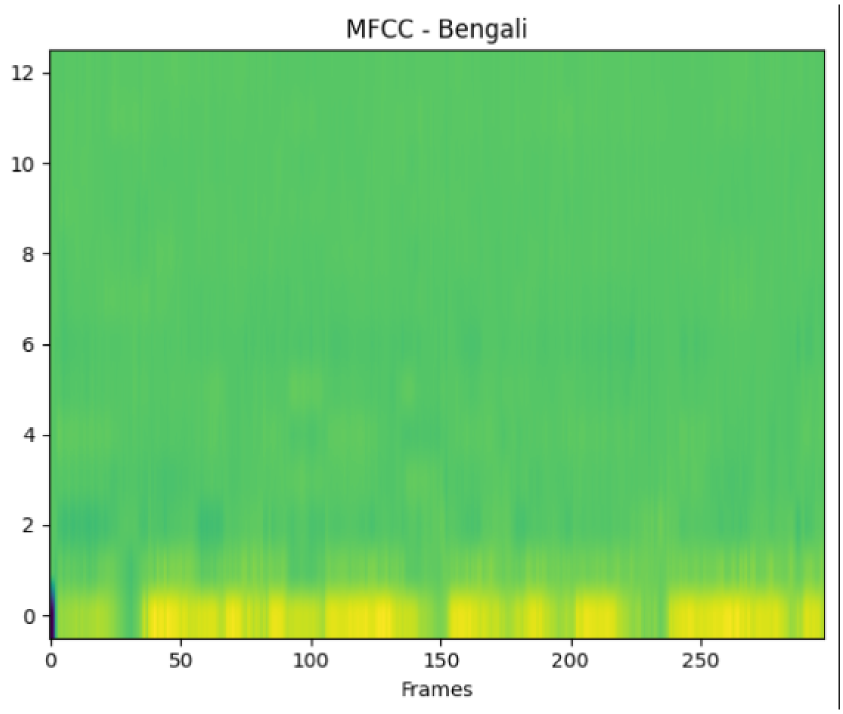

Figure 5: MFCC Spectrogram (Bengali)

Task B: Classification using Random Forest

We used MFCC statistics to classify language using a Random Forest classifier.

Pipeline:

- Extracted mean MFCCs as features

- Applied Label Encoding and Standard Scaling
- Used 80-20 train-test split
- Trained Random Forest Classifier

Classification Report:

```
            precision    recall  f1-score   support

   Bengali       0.93      0.90      0.91        20
     Hindi       0.92      0.95      0.94        20
     Tamil       0.89      0.90      0.90        20

  accuracy                           0.92        60
```

Confusion Matrix:

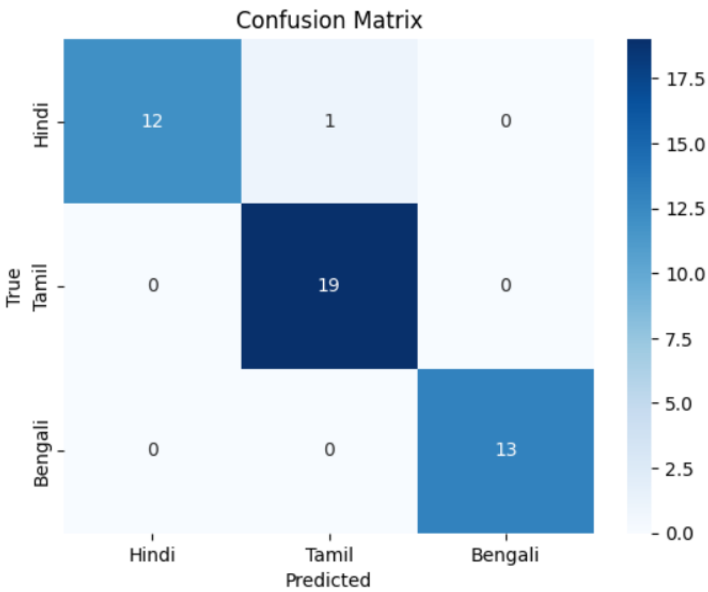

Figure 6: Confusion Matrix of Language Classification

Challenges and Observations

- MFCCs can reflect language-specific phoneme structures
- Classification impacted by background noise and speaker variability
- Some overlap exists due to similar acoustic patterns between languages
- Tamil and Bengali showed more overlapping characteristics than Hindi

Conclusion

This assignment combined deep learning-based speech enhancement with language classification using MFCCs. We explored pretrained and fine-tuned pipelines, multi-speaker separation with SepFormer, and statistical analysis of speech features. The classification of Indian languages showed promising results using MFCC features.

References

- VoxCeleb1/2 Dataset: `https://mm.kaist.ac.kr/datasets/voxceleb/`
- SepFormer (HuggingFace): `https://huggingface.co/speechbrain/sepformer-whamr`
- WavLM Speaker Model: `https://github.com/microsoft/UniSpeech`
- Librosa: Python library for audio processing
- Scikit-learn: Used for classification and evaluation