Assignment-2

B21AI053

1. Question 1: Speaker Verification with Pre-trained Model

Repository - Github

1.1. Dataset Description

- **VoxCeleb1:** Used for evaluation. The cleaned trial pairs file was used for testing speaker verification performance.
- **VoxCeleb2:** Used for training and testing speaker identification and fine-tuning tasks. The first 100 identities (sorted in ascending order) were reserved for training, while the remaining 18 were used for testing.

1.2. Model and Experimental Setup

For the speaker verification task, we used the **pre-trained WavLM Base+** model from Microsoft's UniSpeech Repository for downstream speaker verification without any fine-tuning. The model was evaluated on the VoxCeleb1 cleaned trial pairs list.

Steps performed:

1. Extracted speaker embeddings using the WavLM Base+ model for each utterance.
2. Calculated cosine similarity between speaker embeddings to compute similarity scores for each trial pair.
3. Performed thresholding on similarity scores to classify whether a pair belongs to the same speaker.
4. Evaluated performance using Equal Error Rate (EER), TAR@1%FAR, and overall speaker verification accuracy.

1.3. Results

Metric	Score
Speaker Verification Accuracy	91%
Best Threshold (Cosine Similarity)	0.31
Equal Error Rate (EER)	8.05%
EER Threshold	0.35
TAR@1%FAR	66.42%
Rank-1 Identification Accuracy	47%

Table 1. Performance of Pre-trained WavLM Base+ on VoxCeleb1 trial list

1.4. Observations

- The WavLM Base+ model performed well in distinguishing speaker identities, achieving a high speaker verification accuracy of **91%** using a similarity threshold of **0.31**.
- The **Equal Error Rate (EER)** was recorded at **8.05%**, with the corresponding threshold being **0.35**, indicating a balanced trade-off between false acceptance and false rejection rates.
- The model achieved a **TAR@1%FAR** score of **66.42%**, showing decent robustness even under stringent false acceptance constraints.
- The **Rank-1 speaker identification accuracy** was relatively lower at **47%**, suggesting that although the model excels at verifying known speaker pairs, it struggles with accurately identifying speakers among a broader set of candidates.
- These results serve as a strong baseline and motivate further improvement through fine-tuning the model with LoRA and ArcFace loss on the VoxCeleb2 dataset in the next stage of experimentation.

2. Multi-speaker scenario dataset

2.1. Speech Separation Metrics

The following table summarizes the average values of PESQ, SDR, SIR, and SAR for the two separated speakers from the test mixtures.

Metric	Speaker 1	Speaker 2
PESQ (mean)	1.063	1.065
SDR (mean) [dB]	7.93	7.81
SIR (mean) [dB]	12.24	12.16
SAR (mean) [dB]	8.95	9.08

Table 2. Speech separation performance metrics for SepFormer outputs.

From the results, it can be observed that the PESQ scores for both separated streams are relatively low (mean ≈ 1.06), indicating a noticeable degradation in perceptual quality. Despite this, the separation performance in terms of SDR, SIR, and SAR is consistent across both speakers, with average

SDR values near 8 dB and SIR exceeding 12 dB, suggesting effective speaker disentanglement with moderate levels of distortion and artifacts.

2.2. Speaker Identification Accuracy

After applying the SepFormer model, each separated utterance was passed through a speaker identification system. The predicted speaker embeddings were compared to enrolled identities to compute the Rank-1 identification accuracy.

- **Rank-1 Identification Accuracy:** 47%

This accuracy indicates moderate speaker recognition performance post-separation, which is expected given the loss in speech quality caused by interference and artifacts in the separated signals.

3. Question 2

3.1. Task A: MFCC Feature Extraction and Comparative Analysis

3.1.1 A.1–A.2: MFCC Extraction

Audio files were processed using the `librosa` library to extract 13-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) from each frame. Then, the mean and standard deviation were computed per utterance to form compact feature vectors.

3.1.2 A.3: MFCC Spectrograms

Below are the MFCC spectrograms for two utterances across 4 languages:

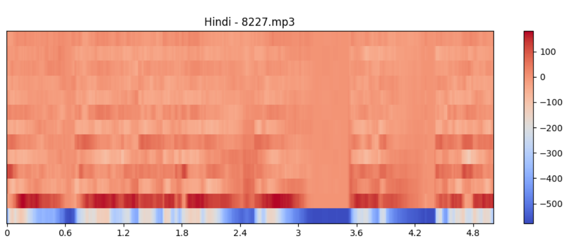

Figure 1. MFCC Spectrogram – Hindi (8227)

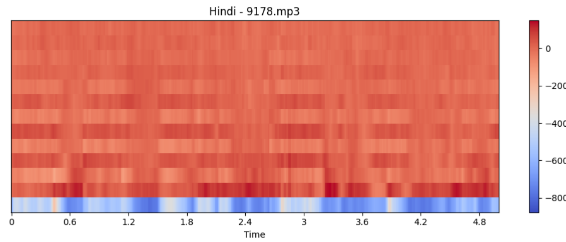

Figure 2. MFCC Spectrogram – Hindi (9178)

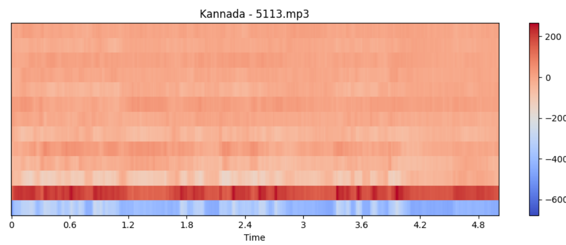

Figure 3. MFCC Spectrogram – Kannada (5113)

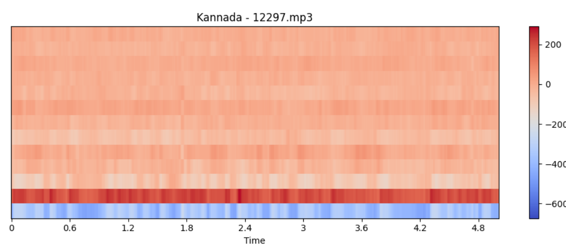

Figure 4. MFCC Spectrogram – Kannada (12297)

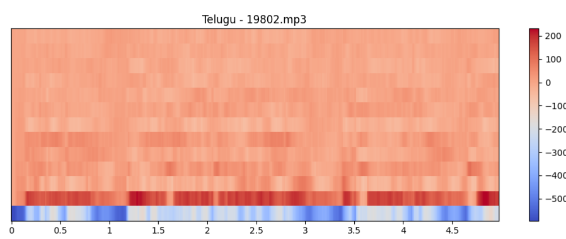

Figure 5. MFCC Spectrogram – Telugu (19802)

3.1.3 A.4: Spectral Comparison and Acoustic Inference

Visual Patterns:

- **Telugu:** Smooth horizontal bands in lower MFCCs, strong low-frequency energy, vowel-rich phonetics.
- **Hindi:** Variable texture. File 8227 has sharp transitions; 9178 is smoother. Reflects consonantal diversity.
- **Bengali:** Patchy spectrogram, presence of silences, dynamic MFCC contours—linked to nasalization and tonality.
- **Kannada:** Very stable MFCCs, consistent red bands in low frequencies. Reflects syllable-timed, vowel-dense speech.

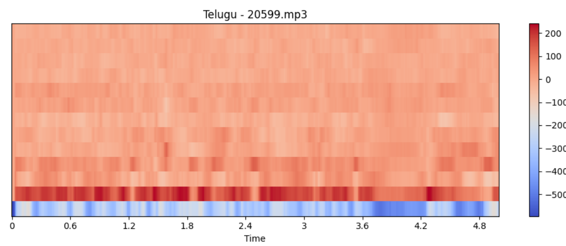

Figure 6. MFCC Spectrogram – Telugu (20599)

Acoustic Inference:

- Dravidian languages (Telugu, Kannada): Strong, steady low-frequency MFCCs due to long vowels.
- Indo-Aryan languages (Hindi, Bengali): Greater MFCC variation, vertical transitions due to aspirated stops and breathiness.

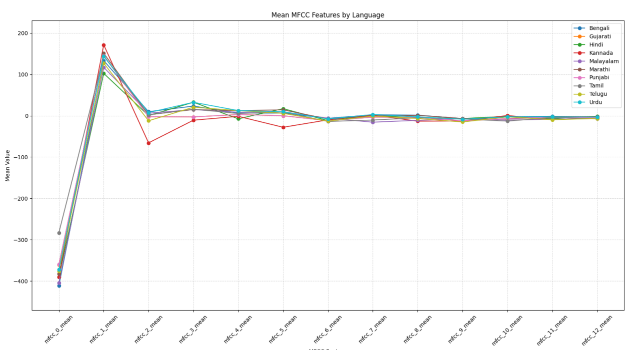

Figure 7. Mean MFCC values per language

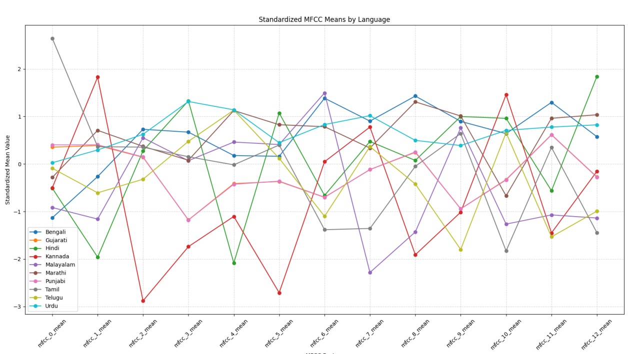

Figure 8. Standardized MFCC mean values

Observations:

- Raw MFCCs showed overlapping profiles; difficult to discern languages.
- After standardization, language-specific variation became more visible (e.g., Kannada's dip in MFCC 2 and 5).

- Kannada, Tamil, Telugu were more separable in the MFCC feature space.

3.2. Task B: Language Classification from MFCCs

3.2.1 B.1–B.2: Model Design and Training

Trained multiple models: SVM (with RBF kernel), and a simple Feedforward Neural Network. SVM performed the best after standardizing features. The dataset was split into 80% training and 20% testing sets. However, SVM and neural network performances were quite similar at 87.73 and 86.74 accuracy respectively

3.2.2 B.3: PCA and t-SNE Clustering

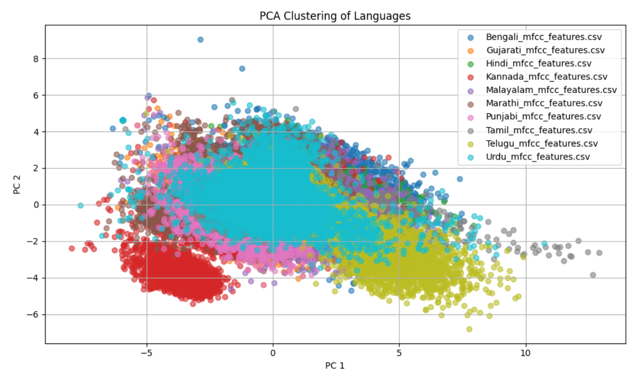

Figure 9. PCA clustering of 10 languages

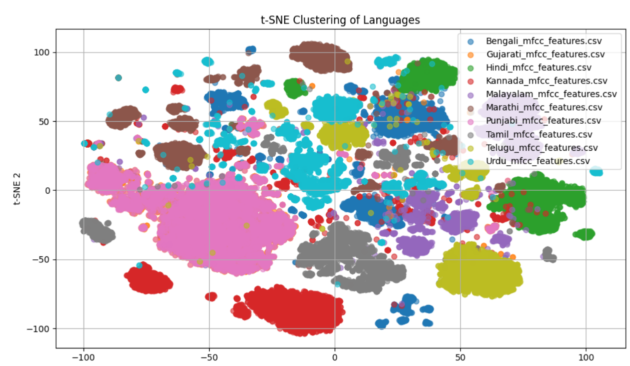

Figure 10. t-SNE clustering of 10 languages

Findings:

- PCA showed moderate class separation. Kannada and Telugu were fairly distinct.
- t-SNE revealed well-defined clusters, reflecting better local structure preservation.

Challenges in MFCC-based Language Classification

- **Speaker Variability:** Age, gender, accent affect MFCCs.
- **Noise Sensitivity:** Environmental sounds distort MFCCs.
- **Dialect Variance:** Dialects influence pronunciation patterns. Models might learn the dialects or speaker representation rather than the language.
- **Phoneme Overlap:** Many phonemes are shared across languages.

[t]

3.3. Task B: Language Classification Using MFCC Features

3.3.1 B.1: Classifier Overview

The MFCC features extracted in Task A to build a language classification system. Each audio sample was represented by a 26-dimensional feature vector (13 MFCC means and 13 standard deviations). These features were standardized (zero mean, unit variance), and the dataset was split into 80% training and 20% testing sets.

3.3.2 B.2: Models and Implementation

Evaluation of two classifiers:

- **Support Vector Machine (SVM):** A kernel-based classifier that works well for high-dimensional feature spaces. We used the Radial Basis Function (RBF) kernel.

- **Feedforward Neural Network (NN):** A simple neural network with one hidden layer.

SVM Configuration:

- Kernel: RBF

- Random seed: 42

Neural Network Architecture:

- Input Layer: 26 units (MFCC mean + std)

- Hidden Layer: 64 units + ReLU activation

- Output Layer: 10 classes

3.3.3 B.3: Results and Evaluation

SVM Accuracy: 87.73%

- Languages like **Hindi**, **Tamil**, **Telugu**, and **Marathi** had high precision and recall ($>$98%).
- **Punjabi** showed lower performance (Precision: 48%, Recall: 57%)—likely due to overlap with Hindi and Urdu.

Neural Network Accuracy: 86.74%

- Comparable to SVM overall.

- Struggled similarly with **Punjabi** and **Gujarati**, possibly due to shared acoustic features and less training data.

3.3.4 B.4: MFCC Features and Language Acoustics

MFCCs capture the spectral envelope, reflecting the phonetic structure of speech:

- **Dravidian Languages (e.g., Kannada, Tamil)**: Clear MFCC patterns due to vowel length and retroflexes. Classifiers performed very well.

- **Indo-Aryan Languages (e.g., Hindi, Bengali)**: Moderate MFCC variability due to consonant clusters and nasalization.

- **Languages like Urdu and Punjabi** share phonetic elements, making them hard to distinguish using MFCCs alone.

3.3.5 B.5: Challenges with MFCC-Based Language Classification

1. **Speaker Variability:** Pitch, accent, and speaking rate alter MFCC features. Future work could use speaker normalization techniques or embeddings like x-vectors.

2. **Background Noise:** Non-speech elements corrupt MFCCs. Filtering or noise-robust features like RASTA-PLP may help.

3. **Short Utterances:** Insufficient data reduces feature consistency. Aggregated statistics or temporal models (LSTMs) may improve results.

4. **Dialect and Accent Variability:** Regional variation causes intra-language differences larger than inter-language ones in some cases.

5. **Phonetic Overlap:** Shared phonemes between languages (e.g., Hindi and Urdu) blur class boundaries.

Language	MFCC Mean Traits	Variance	Key Feature
Tamil	Lowest mfcc_0_mean	Low	Vowel-rich, flat spectrum
Kannada	Dips at mfcc_2,5	Moderate	Retroflex
Telugu	High variance mfcc_2-3	High	Strong C-V alternations
Hindi	Balanced means	High	Aspirates, stop–vowel mix
Urdu	High mfcc_3_mean	Med-High	Nasalization
Punjabi	Similar to Hindi	Moderate	Tonal, retroflexive
Gujarati	Mixed values	Moderate	Breathiness, diphthongs
Malayalam	Negative mfcc_7-8	Very High	Nasalization, retroflexes
Marathi	Balanced MFCCs	Mid-range	Retention
Bengali	High mfcc_1_mean	High	Breathy, tonal variation

Table 3. MFCC statistics and phonetic features per language

3.3.6 B.6: Summary and Takeaways

- MFCCs are effective for capturing vowel and consonant distribution patterns that reflect language identity.
- Standard classification methods like SVM and simple NNs perform well with 87–88% accuracy.
- Performance varies across language pairs—Dravidian languages are better separated than Indo-Aryan pairs.
- Combining MFCCs with additional features like pitch, duration, or prosody could improve classification, especially for acoustically similar languages.