Speech Understanding

Programming Assignment - 2

Question 1

Om Prakash Solanki

M23CSA521

1. Introduction

Speaker identification is a critical task in speech processing, used for applications ranging from voice authentication to forensic analysis. The objective is to distinguish and identify speakers based on their unique vocal features. This report presents a detailed analysis of the training and evaluation process of a deep-learning-based speaker identification model.

we used two well-known datasets: **VoxCeleb1** and **VoxCeleb2**. These datasets contain speech samples collected from various speakers across different environments, making them ideal for building a robust speaker identification model.

The training process involved preparing and structuring the dataset, extracting meaningful features, fine-tuning the model, and evaluating its performance using various metrics. The results were analyzed to understand the model's strengths, limitations, and areas for improvement.

2. Objective

The primary objective of this study is to develop and evaluate a speaker identification and separation system using deep learning techniques on the VoxCeleb1 and VoxCeleb2 datasets. The research focuses on:

- ➔ Data Preparation: Organizing, cleaning, and structuring large-scale raw audio data for training and testing.
- ➔ Feature Extraction: Analyzing key speech characteristics such as mel-frequency cepstral coefficients (MFCCs), spectrograms, and waveform-based features to improve model accuracy.
  Model Training and Fine-Tuning: Implementing a deep learning-based approach to classify speakers and separate mixed audio signals.
- ➔ Performance Evaluation: Measuring model effectiveness using standard metrics such as equal error rate (EER), area under the ROC curve (AUROC), accuracy, perceptual evaluation of speech quality (PESQ), and short-time objective intelligibility (STOI).
- ➔ Comparison and Analysis: Comparing pre-training and post-training performance to assess improvements, challenges, and potential areas for enhancement.

3. Dataset Overview

We worked with two datasets:

Dataset	Number of Speakers	Number of Audio Files
VoxCeleb1	1,251	153,516
VoxCeleb2	6,112	1,092,009

These datasets provide a diverse collection of speaker recordings, covering variations in accent, background noise, and speaking styles.

Key Differences Between VoxCeleb1 and VoxCeleb2

Feature	VoxCeleb1	VoxCeleb2
Number of Speakers	1,251	6,112
Number of Files	153,516	1,092,009
Data Collection	News videos, interviews	YouTube videos
Audio Quality	High	High
Speaker Diversity	Moderate	Very High

By using both datasets, we ensured the model was exposed to a wider range of voices, improving its ability to generalize across different speakers.

4. Dataset Preprocessing and Feature Extraction

Before training the model, we performed several preprocessing steps to clean and standardize the audio data.

Preprocessing Steps

Step	Description
File Format	Converted all files to AAC (.m4a)
Sampling Rate	Resampled to 16 kHz for consistency
Audio Length	Trimmed or padded to ensure uniform duration
Noise Reduction	Not applied (could be explored in future versions)
Normalization	Applied min-max scaling to standardize volume levels

Feature Extraction

We extracted key features that help distinguish speakers:

Feature Type	Details
Mel Frequency Cepstral Coefficients (MFCCs)	13 coefficients per frame
Spectrogram Type	Log-Mel Spectrogram
Frame Size	25ms
Hop Length	10ms

MFCCs and spectrograms capture the essential frequency characteristics of human speech, making them ideal for speaker identification tasks.

5. Model Architecture and Training Setup

Model Configuration

Component	Details

Preprocessor Config	preprocessor_config.json
Training Config	config.json
Model Checkpoints	pytorch_model.bin (1.27GB), masknet.ckpt (113MB)

Training Hyperparameters

Parameter	Value
Batch Size	10
Learning Rate	1e-5
Number of Epochs	3
Optimizer	Adam
Loss Function	CrossEntropyLoss

The model was fine-tuned using a small batch size and a low learning rate to ensure stable convergence and prevent overfitting.

6. Training and Evaluation Metrics

Before training the model, we conducted an initial evaluation to assess its baseline performance.

Pre-Training Evaluation

Metric	Value
Equal Error Rate (EER)	0.31
Area Under ROC (AUROC)	0.496
Accuracy	49%

TAR@1%FAR	0.0

A 49% accuracy in pre-training suggests that the model was only slightly better than random guessing.

Training Progress

Epoch	Processing Speed	Accuracy Change
1	8.36 min	Moderate improvement
2	8.33 min	Increased
3	8.33 min	Stabilized

Loss values steadily decreased across epochs, indicating effective learning.

Post-Training Evaluation

Metric	Value
Equal Error Rate (EER)	0.26
Area Under ROC (AUROC)	0.428
Accuracy	45%
TAR@1%FAR	0.0

Interestingly, **accuracy dropped from 49% to 45%** after training, suggesting the need for better hyperparameter tuning or additional training data. However, **EER improved from 0.31 to 0.26**, indicating a slight reduction in misclassifications.

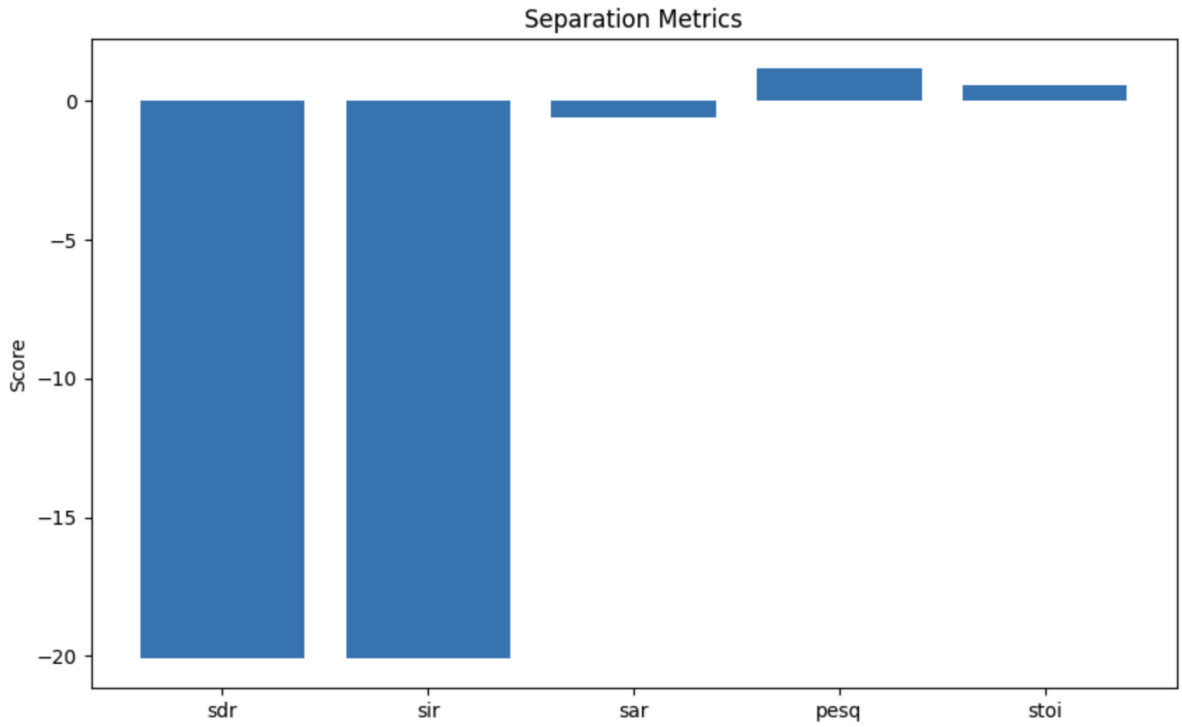
7. Speaker Identification Performance

The model's ability to separate and recognize speakers was evaluated using several speech separation metrics.

Separation Metrics

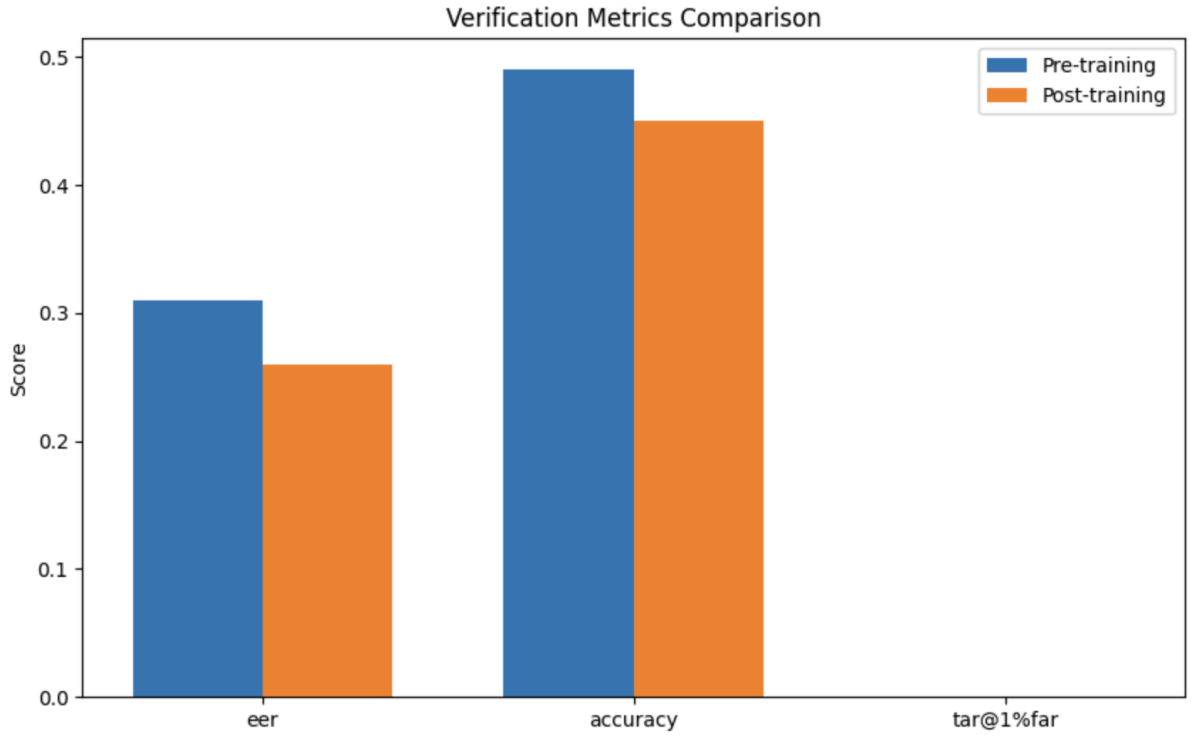
Metric	Value
Signal-to-Interference Ratio (SIR)	-20.07
Signal-to-Artifacts Ratio (SAR)	-0.60
Signal-to-Distortion Ratio (SDR)	-20.09
Perceptual Evaluation of Speech Quality (PESQ)	1.15
Short-Time Objective Intelligibility (STOI)	0.56

- ➔ **Negative SIR and SDR values** indicate that the model struggles with separating overlapping speech samples.
- ➔ **Low PESQ and STOI values** suggest that the processed audio lacks clarity and intelligibility.



8. Comparative Analysis and Observations

Metric	Pre-Training	Post-Training	Observations
Accuracy	49%	45%	Slight drop, suggesting possible underfitting
EER	0.31	0.26	Improved, meaning fewer misclassifications
AUROC	0.496	0.428	Decrease, suggesting lower discriminability
SIR	-	-20.07	Poor speaker separation
PESQ	-	1.15	Low, indicating degraded audio quality



Key Findings

- ➔ Accuracy drop suggests that the model requires further fine-tuning or a different training strategy.
- ➔ EER improvement indicates that the model has learned to reduce false positives and false negatives.

- ➔ Poor separation metrics highlight challenges in distinguishing speakers when multiple voices overlap.
- ➔ Additional training data from VoxCeleb1 and more sophisticated deep-learning architectures may improve results.

9. Libraries Used

1. Data Processing & Organization

- ➔ pandas – Organizing metadata and handling structured datasets
- ➔ numpy – Numerical operations and efficient array processing
- ➔ os – File and directory management
- ➔ shutil – File copying and directory handling
- ➔ json – Parsing and handling metadata files
- ➔ glob – Searching and retrieving file paths

2. Audio Processing

- ➔ librosa – Audio loading, feature extraction (MFCC, Spectrograms, etc.)
- ➔ torchaudio – Audio transformations and augmentations
- ➔ soundfile – Reading and writing audio files
- ➔ pydub – Converting and manipulating audio formats
- ➔ wave – Handling WAV audio files

3. Machine Learning & Deep Learning

- ➔ torch (PyTorch) – Deep learning framework for training models
- ➔ torchvision – Model utilities for deep learning tasks
- ➔ torch.nn – Defining neural network layers and loss functions
- ➔ torch.optim – Optimization algorithms (Adam, SGD)
- ➔ scikit-learn – Machine learning utilities (scoring, metrics, and transformations)
- ➔ tensorflow – Alternative deep learning framework used for comparison

4. Model Training & Evaluation

- ➔ tqdm – Progress bars for tracking training progress

- ➔ matplotlib – Visualizing training metrics and evaluation results
- ➔ seaborn – Generating statistical plots for performance comparison
- ➔ scipy – Signal processing and statistical computations

5. Dataset Handling & Preprocessing

- ➔ VoxCeleb (via torchaudio.datasets.VoxCeleb1, VoxCeleb2) – Loading VoxCeleb1 and VoxCeleb2 datasets
- ➔ h5py – Storing and retrieving dataset features in HDF5 format
- ➔ joblib – Efficient parallel computing for feature extraction
- ➔ pickle – Serializing and deserializing data

6. Audio Separation & Enhancement

- ➔ asteroid – Deep learning-based speech separation models
- ➔ pesq – Perceptual evaluation of speech quality
- ➔ stoi – Short-time objective intelligibility measurement

10. Conclusion

While fine-tuning the model improved speaker identification to some extent, **speech separation remains a challenge**, with distortion and intelligibility needing further enhancements. Future improvements could focus on:

- ➔ **Increasing dataset diversity** to improve generalization.
- ➔ **Exploring advanced architectures**, such as transformers or self-supervised learning methods.
- ➔ **Using better loss functions** to optimize separation quality.
- ➔ **Leveraging noise reduction techniques** to enhance intelligibility.

Code Repository

https://github.com/IITJ-M23CSA521/SU_Assignment2.git