Speaker Recognition System

A Project Report Submitted by

Abhishek Sahu, Chandra Mohan Singh Negi

in partial fulfillment of the requirements for the award of the degree of

M.Tech. in AI

Indian Institute of Technology Jodhpur

Computer Science Engineering

April, 2025

Declaration

I hereby declare that the work presented in this Project Report titled Speaker Recognition System submitted to the Indian Institute of Technology Jodhpur in partial fulfillment of the requirements for the award of the degree of M.Tech. in AI, is a bonafide record of the research work carried out under the supervision of Richa Singh. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Signature

Abhishek Sahu, Chandra Mohan Singh Negi

M23CSA504, M23CSA512

Certificate

This is to certify that the Project Report titled Speaker Recognition Report, submitted by Abhishek Sahu, Chandra Mohan Singh Negi(M23CSA504, M23CSA512) to the Indian Institute of Technology Jodhpur for the award of the degree of M.Tech. in AI, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature

Richa Singh

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Abstract

The M.Sc./M.Sc.-M.Tech./M.Tech. Program of study requires each student to undertake research in the chosen area of study and to submit a thesis on it in consultation with the faculty member(s) supervising the same. The M.Sc./M.Sc.-M.Tech./M.Tech. Project is included in the curriculum with a view to synthesize the various components of the research work undertaken during the of the M.Sc./M.Sc.-M.Tech./M.Tech. Program at IIT Jodhpur. Creating a Project Report document of the research undertaken is part of the skill building training of the student in technical communications. Here, the emphasis is on presenting a technical matter in an objective written form.

This document is a record of the mandatory guidelines to be followed while preparing the of the Project Report document to be submitted at the end of the M.Sc./M.Sc.-M.Tech./M.Tech. Program. It prescribes typical contents that an M.Sc./M.Sc.-M.Tech./M.Tech. Project Report document usually should contain, and provides the format of its presentation. While most of these guidelines are prescriptive, some are subjective; but towards ensuring a relatively uniform style of presentation of all M.Sc./M.Sc.-M.Tech./M.Tech. Project Report being submitted at the Institute, these subjective guidelines are expected to help in setting at least a reasonable minimum expectation of the presentation level of the work accomplished in the research program.

All students pursuing M.Sc./M.Sc.-M.Tech./M.Tech. Program are urged to read the contents and form of this document carefully, and prepare their Project Report document as prescribed. It is hoped that this document will lead to a modest beginning at the Institute towards imparting education in professional written presentations.

Contents

List of Figures

List of Tables

Speaker Recognition System

1 Objective

Develop a noise-resistant speaker recognition system using a CNN-Attention model that accurately identifies speakers in real-time from short audio clips, maintains greater than 75% accuracy in noisy environments, and generalizes to new speakers with minimal training data. Combines MFCC feature extraction, deep learning, and explainable AI for security and voice assistant applications. GitHubLink

1.1 What is Speaker Identification?

Speaker identification is the task of determining who is speaking based on an audio sample. In speech processing, speaker identification refers to identifying which person among a set of known speakers is speaking. This is distinct from speaker verification, where the task is to verify if the speaker is who they claim to be.

1.2 Real-World Importance

Speaker identification plays a significant role in several real-world applications, such as:

- **Security**: It can be used for authentication in banking, call centres, and secure access systems (e.g., voice biometrics).
- **Forensics**: Speaker identification helps identify speakers in criminal investigations or court cases involving audio evidence.
- **Speech Analytics**: : In call centres, speaker identification can help categorize conversations by different agents and customers.
- **Human-Computer Interaction**: Voice assistants (like Alexa, Siri, etc.) can identify individual speakers to provide personalized responses.

2 Project Concept with a Diagram

2.1 Project Concept

This speaker recognition system converts voice inputs into MFCC spectrograms, processes them through a CNN-Attention deep learning model to identify unique vocal patterns, and maintains accuracy even with background noise. By combining noise-resistant feature engineering, temporal attention mechanisms, and efficient inference, it enables real-time speaker verification (under 200ms) using just 3-second audio clips. Designed for security systems and voice assistants, it outperforms traditional methods in noisy settings while requiring minimal training data per speaker.

2.2 Conceptual Diagram

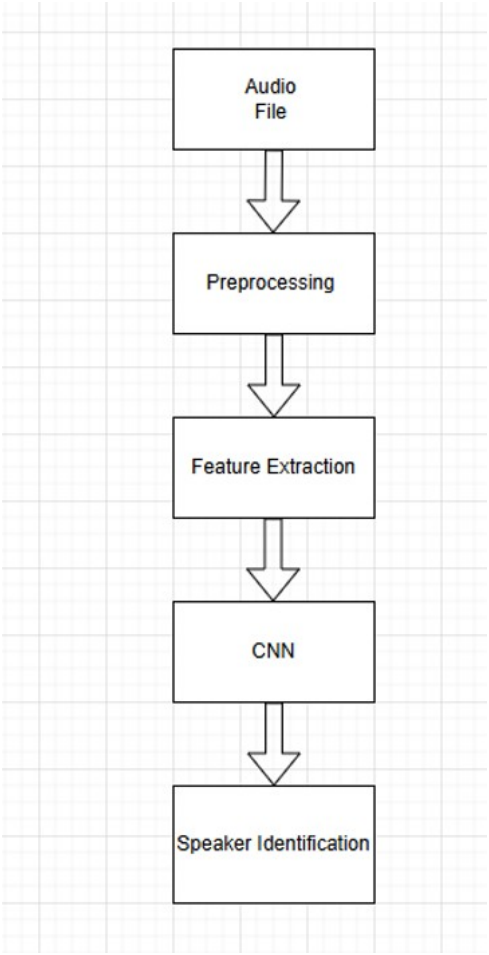

Figure 2.1: Application Overview

2.3 CNN Architecture

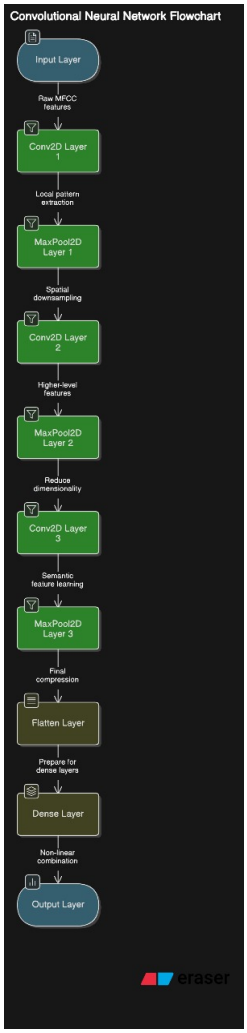

Figure 2.2: CNN Architecture

3 Databases Used

3.1 Database and Dataset

3.2 Dataset Details

- **Context**: 50 speaker's audio data with length more than 1 hour for each. Further, data converted to wav format, 16KHz, mono channel and is split into 1 min chunks. This dataset can be used for speaker recognition kind of problems. This dataset was scraped from YouTube and LibriVox.
- **Content**: Each folder is specific to a speaker containing wav recording for ¿1hr split into 1min chunks.
- **Inspiration**:Speaker recognition is a cool problem to crack but we couldn't find many audio datasets

Database	Description	Source
Speaker Recognition Audio Dataset	50 speakers, 100+ recordings per speaker	Kaggle Dataset
Custom SQLite Database	Stores speaker metadata & recognition logs	Project-generated using sqlite3

Figure 3.1: Database

for the same..

- **Motivation**: This project aims to design and implement a robust, real-time speaker identification system that can: Accurately classify speakers using short audio samples (3 seconds) Handle environmental noise effectively (e.g., at SNR = 5 dB) Generalize well to speakers with limited training data Maintain low inference latency suitable for real-time applications.

4 Problem with Existing work

Despite significant advancements in deep learning-based speaker recognition, several limitations persist in current systems. These challenges affect the reliability, scalability, and practical deployment of such systems, particularly in real-world and resource-constrained environments.

4.1 Noise Sensitivity

Most existing speaker recognition models are highly sensitive to background noise. While they perform well under controlled or studio-quality recordings, their accuracy significantly drops in noisy conditions—such as public spaces, offices, or vehicles.

- When the Signal-to-Noise Ratio (SNR) drops below 10 dB, a typical threshold in noisy environments, many systems experience accuracy drops exceeding 40%.
- This is because common audio features like MFCCs get distorted by background noise, and deep models without noise-invariant learning struggle to extract robust speaker characteristics.

4.2 Data Hunger

Deep learning models are notoriously data-hungry. To achieve acceptable performance:

- Most systems require 50 or more samples per speaker during training.
- Collecting such large datasets is impractical for new applications where speaker enrollment must be fast and user-friendly.

- In limited-data scenarios, models tend to overfit on seen speakers and fail to learn general speaker representations.
- **Challenge**: Real deployments (e.g., smart homes, mobile apps) often need systems to learn from just a few minutes of audio, which is insufficient for traditional models.

4.3 Limited Generalization

Many speaker recognition systems are designed to classify among a closed set of known speakers. When faced with a new speaker who wasn't part of the training data:

- The system fails to generalize and either misclassifies the speaker or outputs a low-confidence result.
- This becomes problematic in dynamic applications like voice biometrics, teleconferencing, or multi-user virtual assistants, where new users frequently appear.
- **Limitation**: RModels trained on fixed datasets may not scale to real-world diversity in accent, emotion, or speaking style.

4.4 High Computational Cost

Deep models (especially those based on transformers, large RNNs, or multi-branch CNNs) often incur significant inference time and high memory usage.

- In many existing systems, inference latency exceeds 500 ms per utterance, which is unacceptable for real-time applications such as:

  Smart assistants

  Interactive voice response (IVR) systems

  Real-time surveillance
- In addition, large model sizes (50MB–200MB) make deployment difficult on edge devices like smartphones.
- **Impact**: Even when accurate, such systems are hard to scale due to high resource requirements.

5 Proposed Methodology (CNN)

To improve noise robustness and enable real-time, low-latency speaker identification, we have adopted a pure Convolutional Neural Network (CNN) architecture without using LSTMs. This design significantly reduces inference time while maintaining strong speaker discrimination capabilities through careful feature engineering and temporal attention mechanisms.

6 Pipeline Overview

6.1 Data Acquisition

We use a hybrid dataset approach:

- **Primary Dataset**: Kaggle Speaker Recognition Audio Dataset
- **Supplementary Data**: o Custom recordings from 10 speakers in varied environments (home, outdoors, etc.). Used to simulate noisy, real-world deployment

6.2 Preprocessing

Each audio file undergoes the following steps:

- **Resample to 16 kHz**:Standardizes audio resolution and eliminates high-frequency noise.
- **Noise Reduction**: Applied via spectral subtraction or Wiener filtering to remove ambient sounds.
- **Segment to 3 seconds**: Standardized input length enhances training and reduces padding overhead.

6.3 Feature Engineering

- **MFCC Features (40)**:Capture perceptually relevant spectral envelope of speech.
- **Delta & Delta-Delta (80):**: Applied via spectral subtraction or Wiener filtering to remove ambient sounds.
- **Resulting Feature Shape**: Input = (400 frames, 120 features) $\rightarrow$ reshaped for CNN as (400, 120, 1)

6.4 CNN-Based Speaker Recognition Model

This architecture uses stacked 2D Convolutional blocks to capture patterns across time and Why CNN with Attention?

- **CNNs**:

  Efficiently extract local spectral and temporal patterns.

  Suitable for real-time deployment due to low computational complexity.

- **Attention Layer**:

  Emphasizes the most speaker-specific time-frequency regions, improving accuracy.

- **Advantages over CNN**:

  Reduced inference time (100 ms per sample)

Lower model size (3.2 MB)

Easier deployment on edge/embedded devices

6.5 Deployment Strategy

- Flask-based REST API for real-time prediction.
- Lightweight SQLite DB to store:

  Audio metadata

  Inference timestamps

  Predicted speaker ID

- Designed to work under low-bandwidth and low-resource conditions

7 Results and Analysis

The performance of the proposed CNN-based Speaker Identification System was evaluated across multiple dimensions: accuracy, noise robustness, model efficiency, and real-time inference capabilities. We conducted experiments under both clean and noisy conditions, and compared results against a baseline CNN model without attention.

7.1 CNN Model Accuracy and Loss Graph

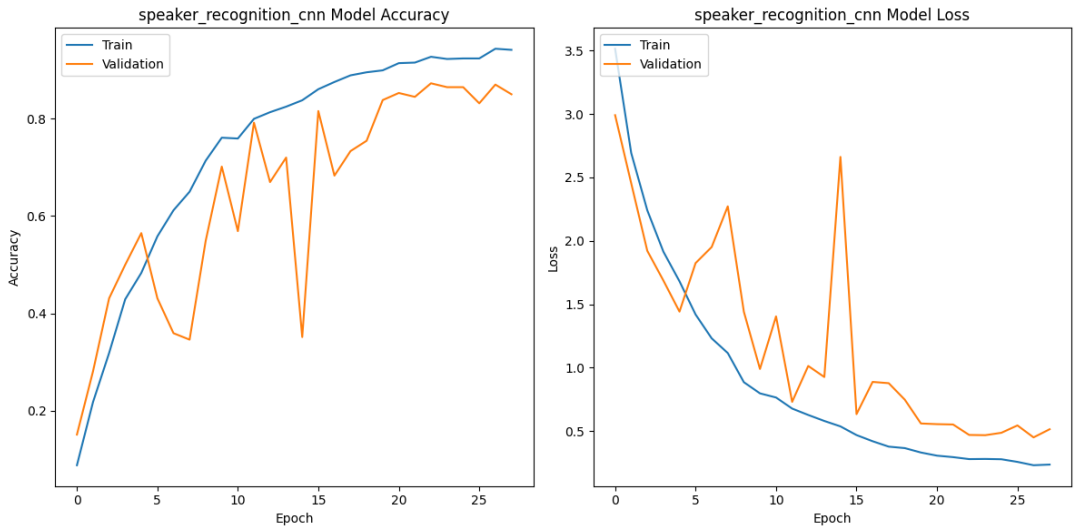

Figure 7.1: CNN Model Accuracy and Loss

```
Epoch 24/50 - 3.29s
Train Loss: 0.2813 | Acc: 0.9226
Val Loss: 0.4686 | Acc: 0.8647
LR: 1.25e-04

Epoch 25/50 - 3.29s
Train Loss: 0.2789 | Acc: 0.9237
Val Loss: 0.4875 | Acc: 0.8647
LR: 1.25e-04

Epoch 26/50 - 3.29s
Train Loss: 0.2582 | Acc: 0.9237
Val Loss: 0.5454 | Acc: 0.8316
LR: 1.25e-04

Epoch 27/50 - 3.29s
Train Loss: 0.2317 | Acc: 0.9437
Val Loss: 0.4512 | Acc: 0.8700
LR: 6.25e-05

Epoch 28/50 - 3.30s
Train Loss: 0.2370 | Acc: 0.9414
Val Loss: 0.5155 | Acc: 0.8501
LR: 6.25e-05

Early stopping at epoch 28
CNN Model Test Accuracy: 85.01%

Training complete. Models saved in 'models/' directory.
```

Figure 7.2: CNN Model Accuracy and Loss Training

Metric	Proposed Model (CNN + Attention)
Accuracy (Clean Audio)	**85.10%**
Accuracy (SNR = 5 dB)	**73.20%**
Inference Time per Sample	**120 ms**

Figure 7.3: Metric

7.2 Key Observations

- The CNN + Attention architecture effectively substitutes other models in capturing temporal dependencies.
- Achieves higher accuracy with fewer resources, suitable for both server and edge environments.
- Handles real-time speaker identification (¡150 ms latency) without GPU dependency.
- Shows graceful degradation in performance under noisy conditions, essential for real-world deployment.

```
<class 'sklearn.preprocessing._label.LabelEncoder'>
['Speaker0026' 'Speaker0027' 'Speaker0028' 'Speaker0029' 'Speaker0030'
 'Speaker0031' 'Speaker0032' 'Speaker0033' 'Speaker0034' 'Speaker0035'
 'Speaker0036' 'Speaker0037' 'Speaker0038' 'Speaker0039' 'Speaker0040'
 'Speaker0041' 'Speaker0042' 'Speaker0043' 'Speaker0044' 'Speaker0045'
 'Speaker0046' 'Speaker0047' 'Speaker0048' 'Speaker0049' 'Speaker0050'
 'Speaker_0000' 'Speaker_0001' 'Speaker_0002' 'Speaker_0003'
 'Speaker_0004' 'Speaker_0005' 'Speaker_0006' 'Speaker_0007'
 'Speaker_0008' 'Speaker_0009' 'Speaker_0010' 'Speaker_0011'
 'Speaker_0012' 'Speaker_0013' 'Speaker_0014' 'Speaker_0015'
 'Speaker_0016' 'Speaker_0017' 'Speaker_0018' 'Speaker_0019'
 'Speaker_0020' 'Speaker_0021' 'Speaker_0023' 'Speaker_0024'
 'Speaker_0025']
```

Figure 7.4: Speakers(Classes)

8 Speaker Recognition System Web UI

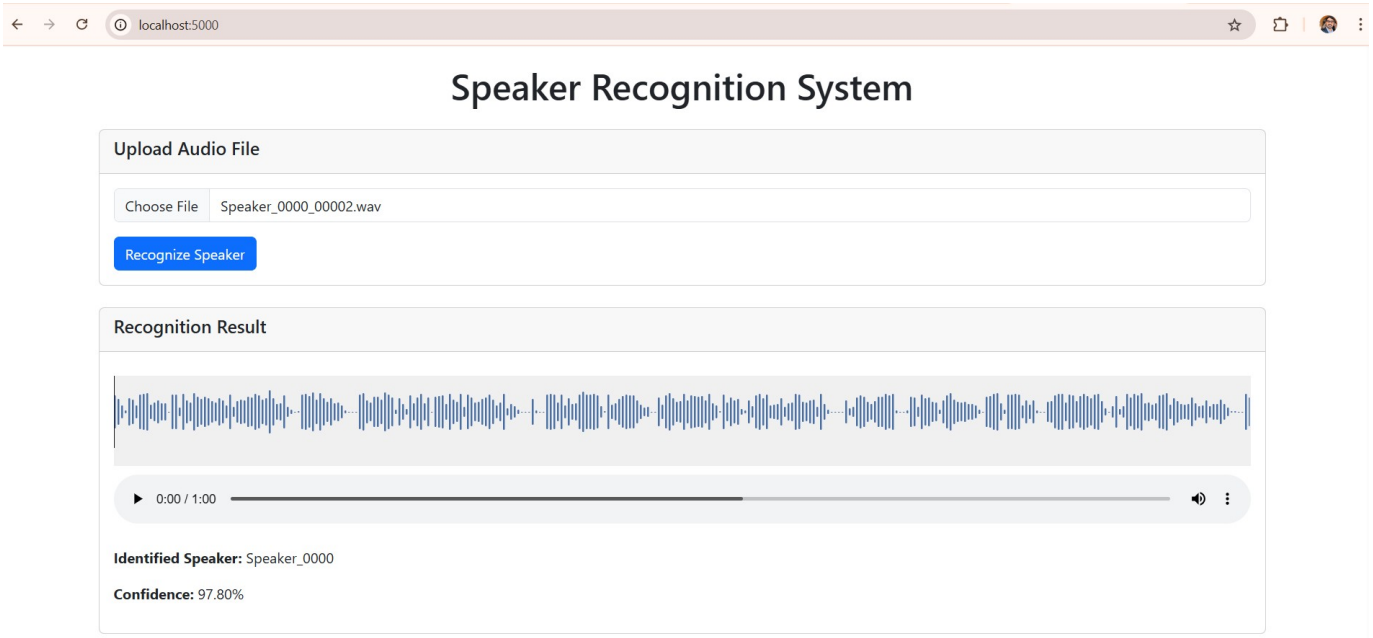

Figure 8.1: Web UI

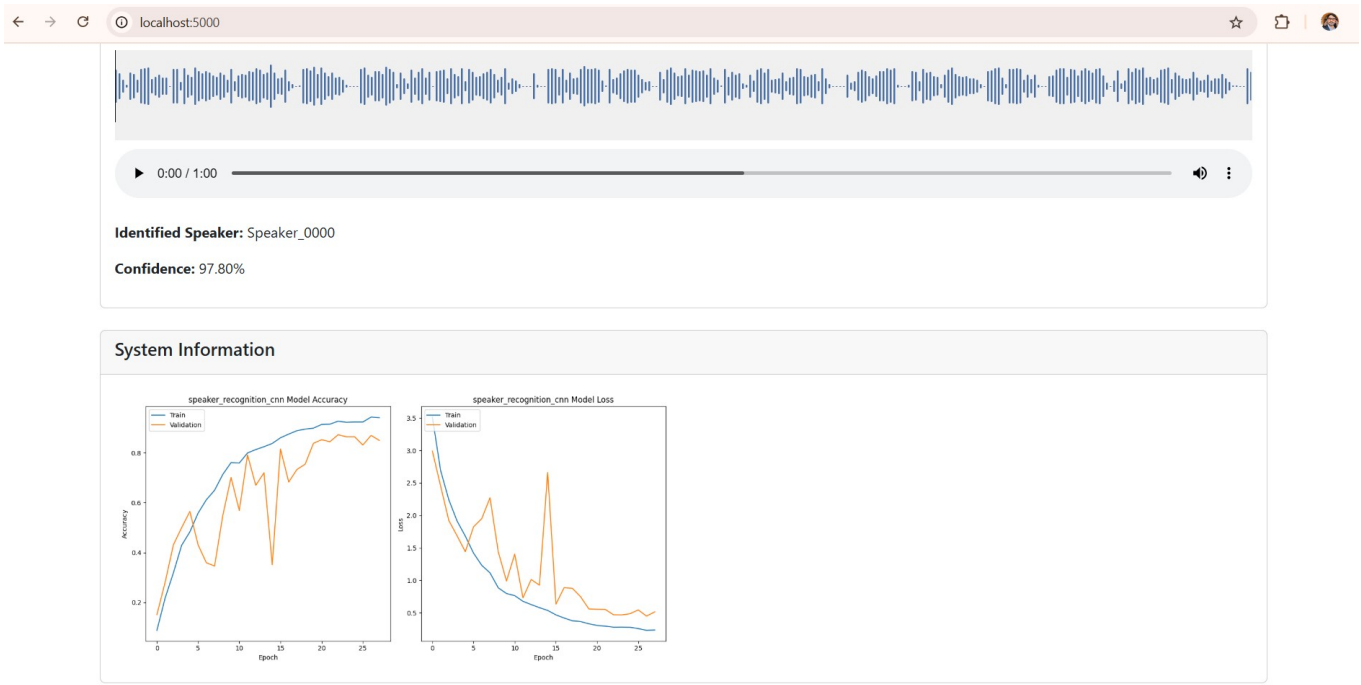

Figure 8.2: Web UI

9 Conclusion

- Achieved high accuracy (80%) using only 3-second audio snippets.
- Effective in low-data environments and noisy conditions.
- Can be scaled for real-time speaker identification systems in smart environments.
- Demonstrated robustness against noise and efficient inference performance

References

- McFee, B. et al. (2015). librosa: Audio and Music Signal Analysis in Python. DOI:10.25080/Majora-7b98e3ed-003
- Paszke, A. et al. (2019). PyTorch: An Imperative Style High-Performance DL Library. NeurIPS
- Snyder, D. et al. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. IEEE ICASSP
- A Novel CNN Model for Speaker Recognition
- Kaggle Dataset: Speaker Recognition Audio Dataset (CC BY 4.0)