# Speaker Verification using the Gaussian mixture model

**Abhay Diwakar**
Student
IIIT Delhi
abhay20164@iiitd.ac.in

**Chehak Malhotra**
Student
IIIT Delhi
chehak21141@iiitd.ac.in

**Krishna Somani**
Student
IIIT Delhi
krishna21058@iiitd.ac.in

**Rahul Kishore Gorai**
Student
IIIT Delhi
rahul23070@iiitd.ac.in

**Rishabh Dhawan**
Student
IIIT Delhi
rishabh23002@iiitd.ac.in

## Abstract

Speaker recognition, a pivotal area in audio and speech processing, has gained significant attention due to its wide range of applications, such as in security, authentication, and personalised services. This report presents an EDA over the dataset given (Speaker Recognition Dataset). After reading various research papers, we concluded that GMM, when combined with MFCC features, produces promising results. We plan to extend this work further and increase the robustness of the model by introducing various combinations of features like Pitch, MFCC delta, MFCC double delta etc.

## 1 Introduction

In an increasingly digital world where voice interfaces, voice assistants, and telecommunication systems are an integral part of our daily lives, the need for robust and accurate speaker identification methods has become more pronounced than ever. Speaker identification, the process of determining and verifying the identity of a speaker based on their voice characteristics, holds significant relevance in diverse applications ranging from security and forensics to personalized user experiences and accessibility.

This report explores the domain of speaker identification using Gaussian Mixture Models (GMMs). GMMs have emerged as a powerful and versatile tool for modelling the acoustic features of speech, making them a useful tool of modern speaker identification systems. Utilizing GMMs, coupled with innovative feature extraction techniques, statistical modelling, and machine learning, gives accurate and efficient solutions for identifying and verifying speakers across a wide spectrum of applications.

In the speaker recognition task, feature extraction is mainly used. Extracting features is a process of holding useful statistics of data from a speech signal while eliminating unwanted signals such as noise. Here, the conversion of the original acoustic wave into a tightly packed representation of the signal feature selection technique.

In section 2 of this report, we have briefly described the inferences we have gathered from the dataset with the help of various plots and graphs done in EDA. Section 2.1 and 2.2 describes the contents of the two folders, i.e., the Audio folder and the Noise folder. In Section 2.3, we have explained all the preprocessing steps that can included for our dataset to help improve the feature extraction part and further help in the speaker verification process. Section 3 briefly describes the various features that can be included in input to improve the robustness of our model. Section 4 includes a Literature

Review of all the related research papers that we have read and found relevant to our project topic. Section 5 briefly explains the steps/stages involved in the process of successful speaker identification.

## 2 Data Preprocessing

**Input:** Raw audio signal / raw audio data

Processed pre-emphasised audio signal with silence and noise removed and endpoints identified.

Preprocessing the audio data is especially important for improving the quality of the data, making it more suitable for various analysis and recognition tasks, and also ensuring that it meets the specific requirements of the task that is needed to be completed. It plays a fundamental role in extracting meaningful information from audio recordings and enhancing the performance of already existing audio data. Pre processing will mainly involve improving the already existing data, transforming it to more manageable form, i.e. removing any bits of unwanted or unrelated data and making the data overall better for analysis (making our model). It will mainly involve:

- **Framing:** In framing we typically divide the continuous audio signal into short frames. This step helps in analyzing the signal in smaller and more manageable segments. But here since the large audio files are already given to us fragmented into 1 second chunks, we might not need to use framing to, divide the audio chunks further.
- **Sampling and adding noise from noise folder:**
  Since noise files are not sampled at 16000 Hz we resampled them to 16000 Hz sample rate. Thus 6 noise files were split into 354 noise samples, where each is 1 sec long. Then, before extracting features we added a random 1 sec sample to the training audio sample. In general, audio recordings often contain background noise. Training models on data that include such noise can make them more robust and better handle real-world, noisy environments.
- **Pre-Emphasis:** pre-emphasis is used to improve signal quality at the output of a data transmission. Preemphasis in audio data is a signal processing technique used to boost the higher frequencies of an audio signal relative to the lower frequencies before it is recorded or transmitted. The preemphasis process involves applying a filter to the audio signal, which boosts the amplitudes of the higher-frequency components. The specific filter used is typically a first-order high-pass filter, which increases the magnitude of the higher frequencies while leaving the lower frequencies relatively unchanged.

**Output:** Processed pre-emphasised audio signal with silence and noise removed and endpoints identified.

## 3 Feature Extraction

While working with audio data, features are often extracted in various domains, including frequency, time, and cepstral domains. Frequency Domain features include Centroid, contrast, flatness, rolloff, flux, spectral entropy. They capture information related to the frequency content of the signal. Time Domain include mean, standard deviation, minimum, maximum, median, skewness, kurtosis, zero-crossing rate etc. and their purpose is to describe characteristics of signal and changes in the signal over time. Then Cepstral Domain Features, include Mel-frequency cepstrum coefficients (MFCC), delta coefficients, double-delta coefficients, LPCC etc. Their purpose is to emphasize human perceptual characteristics, reduce dimensionality, and capture temporal dynamics. Particularly useful in speech and audio processing We are using cepstral features including MFCC, MFCC delta and double delta. MFCC involves framing the signal, applying the Fourier transform, mel-frequency warping, log transformation, and discrete cosine transform (DCT). Its advantages include dimensionality reduction, noise robustness, and effective capture of speech characteristics, making it a key tool in tasks such as speech recognition. MFCC (Mel-Frequency Cepstral Coefficients) are often paired with its delta and double-delta coefficients to capture further how the audio signal changes over time. This addition provides crucial information on temporal dynamics, aiding tasks like speech and speaker recognition. Delta features highlight transitions between speech units, enhancing the overall discriminative power and performance of the feature set. On thourough literature review and testing we concluded using MFCC (mel-frequency cepstrum coefficients) and its delta and double delta coefficients are optimal for our task.

### 3.1 Why MFCC and MFCC delta?

MFCC is designed to mimic how humans hear and perceive sound. It replicates the human hearing system, intending to implement its working principle artificially, assuming the human ear is a reliable speaker recognizer. It emphasizes features of the audio signal important for human speech perception while discarding less relevant information. In paper [2], Jadhav et al. summarized the accuracy scores for various modelling and methods and their features in a table(figure 5). From that, we can see that using MFCC and its Deltas has proven to be very accurate. Further, in the paper [9] by N. Dehak et

| Reference | Modeling method | Features | Datasets | No. of Speakers | Accuracy in % |
|---|---|---|---|---|---|
| Ling Feng [3] | HMM | MFCC | ELSDSR | 22 | 95.48 |
| A. Mansour *et al.* [6] | SVM | SDC+OAA | IEMOCAP | 10 | 91.34 |
| | | SDC+OAO | IEMOCAP | 10 | 90.90 |
| L. Zhu *et al* [2] | VQ | Weighted LPCC | Local | 20 | 94.67 |
| S. Paulose *et al.* [10] | GMM | MFCC | TIMIT | 630 | 96 |
| | | IHC | TIMIT | 630 | 81 |
| H. Veisi *et al.* [4] | HMM | MFCC | Local | 20 | 85 |
| Alsulaiman *et al.* [9] | GMM | MFCC | Local | 50 | 91.41 |
| El-Yazeed *et al.* [24] | Modified Grouped VQ | MFCC (16) | Local | 100 | 90 |
| Sakka *et al.* [25] | DTW | MFCC spaced sub-bands | Local | 20 | 93 |
| N. Dehak *et al.* [11] | GMM | MFCC (13) + Delta (26) | NIST 2006 SRE | 700 | Analysed the EER on both genders |
| M. Alsulaiman *et al.* [14] | GMM | MDLF and MDLF-MA | LDC KSU | 267 | The result of phoneme characteristics and RR's |
| Revathi A *et al.* [17] | GMM | MF-PLP | TIMIT | 50 | 91 |
| Proposed method | GMM | MFCC | MARF | 28 | 96.42 |

RR-Recognition Rate, DTW-Dynamic Time Warping, PLDA-Probabilistic Linear Discriminant Analysis, EER-Equal Error Rate, UBM-Universal Background Model.

Figure 1: Comparison Of The Proposed Method With Literature

al. the MFCC and Delta coefficients are merged for Gaussian modelling and estimation of EER on both genders is done. This encouraged us to use MFCC, delta and double delta coefficients for our task. We also tried using other features like Linear Predictive Cepstral Coefficients (LPCC), which is a feature representation commonly used in speech and audio signal processing. LPCC combines aspects of both linear prediction (LP) and cepstral analysis to capture the spectral characteristics of the signal.

## 4 Literature Review

### 4.1 Broad

A review of speech authentication and verification methods suggests various methods such as using linear prediction cepstrum coding (LPCC), the linear prediction coding (LPC) coefficients, Mel frequency cepstral coefficient (MFCC) and likelihood ratio (LR) interpretation. [4][5]

Another method is using GMM made using MFCCs as the feature vectors. GMM-UBM method where a Universal Background Model (UBM) is obtained using the speech samples from various speakers is also used as an improvement. [6]

### 4.2 Specific

The individual gaussian components of a GMM are shown to represent some general speaker-dependent spectral shapes that are effective for modelling speaker identity. There is the notion that the individual component densities of a multi-modal density like GMM may model some underlying set of acoustic classes which reflect some general speaker dependent vocal tract configurations to classify speaker identity. It is also observed that a linear combination of gaussian basis functions can represent a large class of sample distributions.[1]

GMM modeling technique with two different features Inner Hair Cell Coefficients (IHC) and MFCC has been used where MFCC has a higher accuracy.[2]

The UBM technique is incorporated into the GMM speaker identification system to reduce the time requirement for recognition significantly.31.2 percent relative error reduction is obtained from the combination of both techniques.[3]

# 5 Classification Technique

## 5.1 Gaussian Mixture Models (GMMs)

Gaussian Mixture Models (GMMs) are probabilistic models commonly used for representing complex distributions in a variety of applications, including pattern recognition and signal processing. In the context of speaker verification, GMMs are employed to model the distribution of acoustic features extracted from the voice data of different speakers.

A GMM is characterized by a weighted sum of multiple Gaussian components, each representing a distinct mode in the data distribution. Mathematically, the probability density function (PDF) of a GMM is defined as:

$$P(x) = \sum_{i=1}^{K} w_i \cdot N(x; \mu_i, \Sigma_i) \tag{1}$$

where:

$x$ is the feature vector,
$K$ is the number of Gaussian components,
$w_i$ is the weight of the $i$-th component,
$\mu_i$ and $\Sigma_i$ are the mean vector and covariance matrix of the $i$-th component,
$N(x; \mu_i, \Sigma_i)$ is the multivariate Gaussian distribution.

The parameters of the GMM are estimated through the Expectation-Maximization (EM) algorithm, maximizing the likelihood of the observed data.

We have implemented a Gaussian Mixture Model (GMM) based speaker verification system in which its objective is to verify whether a given speech utterance belongs to a specific enrolled speaker.

## 5.2 Why we use GMM for speaker verification ?

We choose Gaussian Mixture Models (GMMs) for speaker verification models due to the following reasons:

### 5.2.1 Flexibility in Modeling Speaker Specific Patterns

One of the major advantages of GMM in speaker verification lies in its ability to model complex and variable patterns in speech data. The flexibility of GMMs allows them to capture the diverse acoustic characteristics associated with different speakers. By representing each speaker with a mixture of Gaussian, GMMs can model the inherent variability in speech signals, making them well-suited for speaker verification tasks.

### 5.2.2 Probabilistic Framework

GMMs provide a probabilistic framework for speaker verification, enabling the estimation of the likelihood that a given voice sample belongs to a particular speaker. This probabilistic nature allows for a more statistical approach to speaker modeling than deterministic methods.

## 5.3 Steps of GMM

### a) Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation involves determining parameter values that maximize the probability of observing the given data. In the context of GMM for speaker verification, the parameters (denoted as $\lambda$) are initialized, and mean and variance values are iteratively adjusted using known data to maximize the likelihood function. The likelihood function is often expressed using the natural logarithm for convenience. GMM, through MLE, constructs self-clustering boundaries by modeling data as a probabilistic mixture. Each component in the mixture represents a Gaussian distribution with its unique parameters and associated variance variables.

**b) Expectation Maximization (EM)**

The Expectation-Maximization (EM) algorithm is employed for estimating underlying variables, especially those from a mixture distribution. This iterative algorithm is useful when dealing with incomplete or unexpected data. The EM algorithm aims to find the maximum probability by iteratively updating parameters. The steps include:

- **Initialization:** Begin by setting initial values for weights ($w_i$), means ($\mu_i$), and covariances ($\Sigma_i$) for each Gaussian component.
- **Probability Computation:** For each data point and Gaussian component, compute the probability $P(Z = i|x, \mu)$ using the current parameters and the Gaussian distribution.
- **Parameter Update:** Use the calculated probabilities to iteratively update the parameters ($\mu$) through the Expectation step, adjusting weights, means, and covariances.
- **Convergence Check:** Repeat steps 2 and 3 until the parameters converge, ensuring little change between consecutive iterations. Convergence indicates the optimal parameter values have been found.

The formulae for update of different values are (M-Step):

**1. Effective Number of Data Points Assigned to the $k$-th Component:**

$$N_k = \sum_{i=1}^{N} r_{ik}$$

This equation calculates the effective number of data points assigned to the $k$-th component.

**2. Updated Weight of the $k$-th Component:**

$$\pi_k = \frac{N_k}{N}$$

This equation determines the updated weight of the $k$-th component.

**3. Updated Mean Vector of the $k$-th Component:**

$$\mu^k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} x_i$$

It calculates the updated mean vector of the $k$-th component.

**4. Updated Covariance Matrix of the $k$-th Component:**

$$\Sigma^k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (x_i - \mu^k)(x_i - \mu^k)^T$$

This equation computes the updated covariance matrix of the $k$-th component.
These equations are essential components of the Expectation-Maximization (EM) algorithm for training Gaussian Mixture Models (GMMs). They iteratively update the model parameters ($\pi_k, \mu^k, \Sigma^k$) to maximize the likelihood of the observed data given the GMM.

**c) Log Likelihood**

In Gaussian Mixture Models (GMMs), the log-likelihood is a measure of how well the model explains the observed data. The expression

$$\ln\left(p(X|\pi, \mu, \Sigma)\right) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j) \right\} \tag{2}$$

represents the logarithm of the likelihood function in the context of Gaussian Mixture Models (GMMs). Here's a breakdown:

$N$ is the number of data points in the dataset.

$K$ is the number of components (clusters) in the GMM.

$\pi_j$ represents the weight of the $j$-th component in the mixture.

$N(x_i|\mu_j, \Sigma_j)$ is the probability density function (PDF) of the $j$-th Gaussian component evaluated at the data point $x_i$ with mean $\mu_j$ and covariance matrix $\Sigma_j$.

The expression captures the log-likelihood of the data given the parameters $(\pi, \mu, \Sigma)$ of the GMM. It involves summing over all data points ($\sum_{i=1}^{N}$) and taking the logarithm of the sum of the weighted probabilities of each data point belonging to any of the Gaussian components ($\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j)$). This log-likelihood is a key component in the training and optimization of GMMs, often utilized in the Expectation-Maximization (EM) algorithm.

## 6 Experiment Details

The proposed system performs on 5250 training speech samples and 2251 testing speech samples from the kaggle dataset. The speeches were of 1 sec duration each.

1. **Introduction:** The outcomes of this comprehensive experiment are centered around the development of a speaker recognition system employing Gaussian Mixture Models (GMM). The primary objective of the project was to establish a resilient system capable of accurately identifying speakers from a diverse dataset consisting of 1-second WAV files.
2. **Dataset and Preprocessing:** The dataset comprised audio recordings, each lasting one second, organized into folders representing individual speakers. To ensure consistency, a separate folder contained noise samples, with their sampling rates standardized to 16,000 Hz. A meticulous preprocessing phase was undertaken to organize the dataset systematically, allowing for streamlined training and testing processes.
3. **Feature Extraction:** Feature extraction was a crucial step, employing Mel-frequency cepstral coefficients (MFCCs) to capture the distinctive acoustic signatures of each speaker. The extraction process included both clean audio and versions with added noise, introducing a layer of complexity to the feature set. The decision to segment audio into 1-second slices aimed to preserve critical acoustic features, enhancing the system's adaptability to real-world scenarios.
4. **GMM Model Training:** Gaussian Mixture Models (GMMs) played a pivotal role in speaker modeling, with a distinctive model trained for each speaker. The GMMs, characterized by four components (n_components = 4), underwent training utilizing the Expectation-Maximization (EM) algorithm, iterating up to a maximum of 160 times. This approach aimed to effectively capture intricate acoustic patterns unique to individual speakers, establishing a robust foundation for the recognition system.
5. **Testing and Evaluation:** The trained GMM models were subjected to rigorous testing on a dedicated test set. The evaluation process involved comparing the predicted speaker labels against ground truth labels from the test set. This meticulous testing phase sought to gauge the system's generalization capabilities and its accuracy in correctly identifying speakers not encountered during training.
6. **Speaker Comparison:** A noteworthy feature of the system was its ability to compare two speakers based on their extracted features and trained GMM models. This functionality provided a granular understanding of the system's discriminatory capabilities, shedding light on how effectively it could distinguish between different speakers.
7. **Test Pairs and Evaluation:** The system's performance was further scrutinized through the evaluation of specific test pairs outlined in `test_pairs.txt`. This focused evaluation provided insights into the system's efficacy in controlled scenarios, offering nuanced observations on its ability to discern between pairs of speakers.
8. **Example Speaker Comparisons:** Two illustrative examples of speaker comparisons using specific audio files were included in the report. These practical demonstrations highlighted the system's real-world applicability, showcasing its potential in diverse scenarios.

## 7 Results

The speaker recognition system exhibited promising results, accurately identifying speakers based on acoustic features. The testing and evaluation phases provided valuable insights into the system's generalization capabilities and accuracy metrics.

| Preprocessing | Features | EER |
|---|---|---|
| - | MFCC, Delta, Double Delta | 0.246 |
| - | LPCC, MFCC | 0.190 |
| - | MFCC, Delta, Double Delta, Spectral Bandwidth, Spectral Contrast, Spectral Flatness, Spectral RollOff | 0.177 |
| Normalization | MFCC, Delta, Double Delta, Spectral Bandwidth, Spectral Contrast, Spectral Flatness, Spectral RollOff | 0.236 |
| Normalization | MFCC, Delta, Double Delta | 0.274 |
| Noise | LPCC, MFCC | 0.257 |
| Normalization + Noise | MFCC, Delta, Double Delta | 0.367 |
| Noise | MFCC, Delta, Double Delta | 0.237 |

Figure 2: Comparison Of EER using different Parameters

The ability to compare speakers and evaluate specific pairs underscored the system's practical utility. The outcomes of these experiments provide a solid foundation for future refinements and optimizations, aiming to elevate the system's overall performance and real-world applicability. The comprehensive nature of these experiments positions the speaker recognition system as a robust tool for diverse applications.

We used equal error rate as performance metric for our different versions of models. Equal Error Rate (EER) is a crucial metric in evaluating the performance of a speaker recognition system. It represents the point at which the rates of false acceptance and false rejection are equal, indicating a balanced trade-off between the two.

On trying different things we listed down values of EER for our code. But we found 0.177 EER best when we used MFCC delta, double delta and some other spectral features

In the context of our speaker recognition system, achieving an EER of 0.177 signifies an equilibrium where the system is equally adept at accepting true speaker identities and rejecting impostors. This low EER value underscores the system's effectiveness in striking a balance between sensitivity and specificity, crucial for real-world applications where accurate speaker identification is paramount.

## 8 Conclusion

The implemented speaker recognition system, leveraging Gaussian Mixture Models (GMMs) and MFCC feature extraction demonstrates promising outcomes in identifying speakers from a diverse set of 1-second WAV files. The intricate nature of sound, meticulously addressed through thoughtful audio signal feature extraction, contributes to the system's delicate yet robust performance.

In this study for the MFCC technique, coupled with the GMM system, enhances speaker recognition accuracy. The likelihood correlation, assessed using Gaussian mixture models, emerges as a simple yet effective method for detection across various speech samples. The system, tailored for 5 speakers, achieves an Equal Error Rate (EER) of 0.177, emphasizing its reliability.

Future advancements will entail expanding the system to accommodate a larger pool of speakers and optimizing computational efficiency through diverse preprocessing and feature extraction methods. The commitment is to maintain precise speaker classification accuracy while streamlining computational processes.

## References


[1] Reynolds, D.A. Rose, Richard. (1995). Robust text-independent speaker identification using Gaussian Mixture speaker models. Speech and Audio Processing, IEEE Transactions on. 3. 72 - 83. 10.1109/89.365379.

[2] Jadhav, Ajinkya Dharwadkar, Nagaraj. (2018). A Speaker Recognition System Using Gaussian Mixture Model, EM Algorithm and K-Means Clustering. International Journal of Modern Education and Computer Science. 10. 19-28. 10.5815/ijmecs.2018.11.03.

[3] Rong Zheng, Shuwu Zhang and Bo Xu, "Text-independent speaker identification using GMM-UBM and frame level likelihood normalization," 2004 International Symposium on Chinese Spoken Language Processing, Hong Kong, China, 2004, pp. 289-292, doi: 10.1109/CHINSL.2004.1409643.

[4] J. H. Gambhir and V. V. Patil, "A Review On Speech Authentication And Speaker Verification Methods," 2021 Fourth International Conference on Microelectronics, Signals Systems (ICMSS), Kollam, India, 2021, pp. 1-6, doi: 10.1109/ICMSS53060.2021.9673603.

[5] Huapeng Wang and Jun Yang, "The fusion of forensic speaker verification systems," 2011 4th International Congress on Image and Signal Processing, Shanghai, China, 2011, pp. 2450-2453, doi: 10.1109/CISP.2011.6100731.

[6] Poddar, Arnab Sahidullah, Md Saha, Goutam. (2018). Improved i-vector extraction technique for speaker verification with short utterances. International Journal of Speech Technology. 21. 10.1007/s10772-017-9477-2.

[7] W. M. Campbell, D. E. Sturim and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," in IEEE Signal Processing Letters, vol. 13, no. 5, pp. 308-311, May 2006, doi: 10.1109/LSP.2006.870086.

[8] Garcia-Romero, Daniel Espy-Wilson, Carol. (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems.. Proc. Interspeech. 249-252. 10.21437/Interspeech.2011-53.

[9] Dehak, Najim, Pierre Dumouchel, and Patrick Kenny. 2007. “Modeling Prosodic Features with Joint Factor Analysis for Speaker Verification.” IEEE Transactions on Audio, Speech, and Language Processing 15 (7): 2095–2103. https://doi.org/10.1109/tasl.2007.902758.