Understanding and Experimenting with Speaker-Identification

Gurarmaan S. Panjeta
Department of Computer Science
IIT Delhi
cs5200426@iitd.ac.in

Abstract

Speaker identification is a fundamental problem in speech processing with diverse applications in security, personalized systems, and human-computer interaction. The task involves mapping an audio sample to a specific individual in a pre-defined database, leveraging unique vocal characteristics. Over the years, significant advancements have been made in the field through a combination of signal processing techniques and machine learning models, including recent strides with deep learning. This report aims to provide a comprehensive introduction to the problem of speaker identification, detailing key components in the pipeline such as feature extraction and model architectures. Special attention is given to modern approaches leveraging Automatic Speech Recognition (ASR) and embedding-based systems.A thorough case study is done for NVIDIA's NeMo toolkit, which offers state-of-the-art pre-trained models and customizable pipelines for ASR and Speaker Identification tasks. By evaluating the performance and practical utility of NeMo models, the case study highlights both the strengths and limitations of cutting-edge solutions, paving the way for further research and application in the field.

1 Introduction

In today's technology-driven world, advanced products have become an integral part of everyday life, reshaping habits and routines in profound ways. As technology progresses, there is a clear shift toward more user-friendly and human-oriented innovations. Biometric identification stands out as a key advancement, offering seamless and intuitive ways to identify individuals. These systems are gradually replacing traditional authentication methods, which often require prior learning or effort to use effectively. Notable examples of biometric applications include facial recognition systems in airport terminals [1] and voice assistants like Apple's Siri [2], both of which illustrate the potential of biometrics to simplify interactions and improve convenience in daily life.

Sound has always been one of the most natural and direct ways for humans to express themselves, communicate, and engage in interaction. Over time, technological innovations like the telephone [3] have revolutionized how we use sound to connect, evolving from the home phone to functional mobile phones, and eventually to modern smartphones. Despite these advancements in form and function, the core principle of using voice to convey information and foster communication has remained unchanged. Voice is not only intuitive but also one of the most convenient means of transmitting messages. Building on this, identifying individuals through their voice and understanding their dialogue content opens the door to personalized services. Such innovations have the potential to enhance daily life by making interactions more practical and seamless.

Speaker Identification is the process of recognizing a person based on their voice characteristics, such as pitch, tone, and speaking patterns. The input is typically an audio sample, while the output is the identification of the speaker, often matched to a database of known individuals. This technology has a wide range of applications, including security (e.g., voice-based authentication for access control),

personalized services (e.g., customizing voice assistants like Siri or Alexa), forensics (e.g., matching voices in criminal investigations), and healthcare (e.g., patient voice recognition). It can also be used in multimedia for content indexing and speaker diarization [4]. By leveraging unique voice traits, speaker identification enhances both convenience and security in various contexts.

The evolution of systems in tasks like speaker identification reflects a broader trend in artificial intelligence and machine learning. Initially, systems relied on manual, rule-based processing, where engineers designed handcrafted rules and heuristics to analyze data. These approaches were limited by their rigidity and inability to generalize beyond predefined conditions. The next stage of advancement introduced feature extraction [5] combined with machine learning, where human experts identified key features and patterns via complex statistical analysis —that were fed into models like Support Vector Machines (SVMs) [6] or Gaussian Mixture Models (GMMs) [7]. These systems were more adaptable but heavily dependent on the quality of feature engineering. The advent of deep learning [8] [9] marked a paradigm shift, allowing models to learn hierarchical feature representations directly from raw data, eliminating the need for manual feature design.

Traditional speaker identification techniques [10] [11] relied on carefully designed pipelines that extracted and processed acoustic features from audio signals. The audio was first divided into short frames (e.g., 20-40 milliseconds), assuming the signal's properties remain stationary within each frame. Fourier Transforms [12] were used to convert these frames from the time domain to the frequency domain, enabling analysis of frequency components. From this, techniques like the Discrete Cosine Transform (DCT) [13] and Mel-Frequency Cepstral Coefficients (MFCC) [11] were applied to extract compact and meaningful features representing the vocal tract's characteristics. These frame-level features were then accumulated and averaged across the audio input to form a global representation of the speaker. Models such as Gaussian Mixture Models (GMMs) or Support Vector Machines (SVMs) were typically used to classify these features into speaker identities.

Modern deep learning models for speaker-identification [14] [15] utilize advancements like convolutional layers [16], transformers [8], and batch normalization [17] to achieve high accuracy and robustness. Convolutional layers extract local patterns from spectrograms, while transformers model long-range dependencies in sequential audio data. These models generalize well to novel data by learning hierarchical representations directly from raw audio, reducing reliance on handcrafted features. Batch normalization stabilizes training, while techniques like data augmentation and dropout improve robustness. Additionally, deep models can be fine-tuned [18] on new datasets, enabling adaptation to domain-specific scenarios without requiring retraining from scratch.

2 Traditional Methods - Case Study - MFCC

While this discussion goes into the specific case of [11], it is important to note that other popular features/architectures were based on similar principles and differed only in their implementations.

2.1 Feature extraction

Mel-Frequency Cepstral Coefficients (MFCCs) represent the vocal tract features efficiently for tasks like speaker identification. They are extracted in a 5-step pipeline as shown in Figure 1.

1. First, the audio signal is divided into short overlapping frames (e.g., 20-40 ms) to capture stationary properties. The paper chooses a half-overlapped 26 ms frame, as in Figure 3
2. A window function, such as a Hamming window, is then applied to each frame to reduce discontinuities at the frame edges.
3. Next, the Fast Fourier Transform (FFT) converts the frames into the frequency domain, allowing analysis of the signal's spectral content. This also has the advantage of being smoother than the time-domain, especially within the short time durations considered per-frame.
4. Mel-based (Figure 2) bandpass filters (Figure 4) are applied to the FFT output to mimic the human auditory system by focusing on perceptually relevant frequency bands.
5. Finally, the Discrete Cosine Transform (DCT) is used to extract compact coefficients.

Coefficients obtained for each frame are collected and passed to the GMM pipeline for identification.

Figure 1: MFCC Pipeline

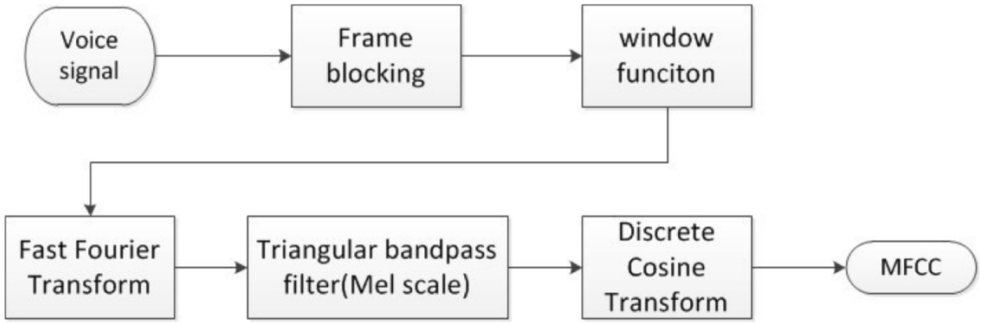


Figure 2: Human-ear sensitivity based MeI Scale

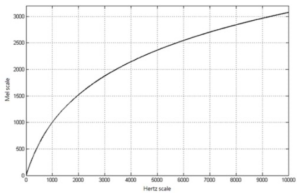


Figure 3: Splitting signal into frames

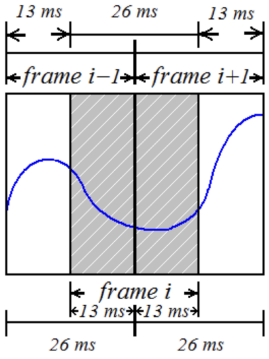
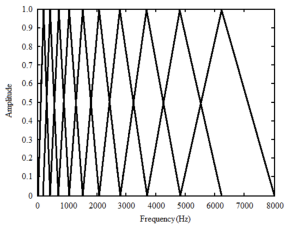


Figure 4: Band Pass Filters for feature extraction



2.2 K-means Clustering

K-means clustering is performed to group the MFCC coefficients into 128 clusters to reduce computational load on the GMM component. This approximation greatly saves computational effort, but results in an inadvertent loss of information. Modern techniques like attention have larger context-windows that alleviate this problem without loss-of-information.

2.3 GMM-based Identification

The features obtained are put in an Expectation-Maximization (EM) Cycle that tunes the parameters of the GMM values such that we compute the features of the source that is most likely to have emitted this voice sample. Then, the Bhattacharyya distance (Figure : **??**) is computed between the computed (probability distribution of the) source and the users present in the database. The best-matching (highest similarity) database entry is returned as the predicted source of this signal.

2.4 Drawbacks

- Scaling : The GMM step becomes prohibitively large as the number of users scale.
- No Pre-trained : Cannot leverage information from other datasets to improve performance.
- Hand-crafted : Requires careful modulation of each of the many moving parts in the pipeline.
- Serial : Cannot parallelize operations across batches, serious limitation for batch processing.
- Over-fitting : Manual intervention is required to make sure model does not fit to the training data. This greatly inhibits performance in unseen examples, especially for the low-data setting.

Figure 5: MFCC + GMM Pipeline

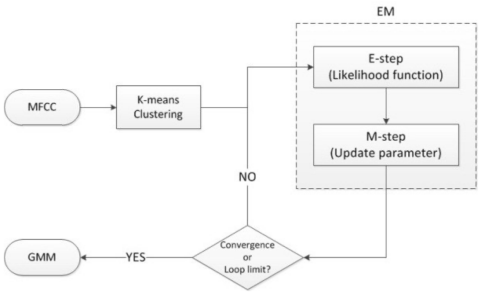


Figure 6: Bhattacharyya Distance

$$d_{BA}(\lambda_u, \lambda_i) = \frac{1}{8}(\mu_u - \mu_i)^T \left(\frac{\Sigma_u + \Sigma_i}{2}\right)^{-1} (\mu_u - \mu_i) + \frac{1}{2} ln \frac{\left|\frac{1}{2}(\Sigma_u + \Sigma_i)\right|}{\sqrt{|\Sigma_u| + |\Sigma_i|}}$$

3 Modern Method - Case Study - TitaNet

3.1 Building Blocks

Before we discuss the TitaNet architecture, we need to familiarize ourselves with some Deep Learning Modules that are relevant to the task of Speaker-identification.

Attention : Attention mechanisms are a powerful tool in modern deep learning, enabling models to focus on the most relevant parts of an input sequence while processing data. In speaker identification tasks, attention is used to weigh different segments of an audio signal based on their importance for identifying a speaker, ensuring the model prioritizes distinctive voice characteristics. This is particularly useful for handling variability in audio, such as background noise or redundant information. Self-attention, as used in transformers, captures global dependencies across the sequence, making it highly effective for analyzing long and complex audio inputs. By allowing dynamic weighting of features, attention mechanisms enhance the robustness and accuracy of speaker identification systems, even in challenging conditions.

Convolution : Convolutions are a foundational operation in deep learning, designed to capture local patterns in data by applying filters over small regions of the input. In modern speaker identification tasks, convolutions are used to process audio spectrograms or feature maps, identifying critical patterns such as pitch, harmonics, and temporal variations that are unique to a speaker. By leveraging shared weights and spatial locality, convolutional layers efficiently extract meaningful features while reducing computational complexity. These layers are particularly effective at capturing fine-grained details in audio, making them well-suited for distinguishing between speakers. Convolutional architectures also generalize well to diverse and unseen data, enhancing the robustness of speaker identification systems.

ReLU : ReLU (Rectified Linear Unit) is an activation function widely used in deep learning, defined as ReLU(x)=max(0,x). It introduces non-linearity into the model while maintaining computational efficiency, allowing networks to learn complex patterns. In speaker identification tasks, ReLU is used to enable deep models to capture intricate relationships in audio features, such as unique speaker traits from spectrograms or embeddings. Its simplicity avoids the vanishing gradient problem common in earlier activation functions like sigmoid, ensuring effective training of deep networks. By facilitating the learning of rich and discriminative representations, ReLU plays a crucial role in modern speaker identification pipelines.

Dropouts : Dropout is a regularization technique used in deep learning to prevent overfitting by randomly deactivating a fraction of neurons during training. This forces the network to learn more robust and distributed representations by preventing reliance on specific neurons. In speaker identification tasks, dropout is particularly useful for handling overfitting when training on limited or noisy audio datasets, ensuring the model generalizes better to unseen speakers and conditions. By introducing randomness, dropout reduces the model's sensitivity to variations in the training data, making it more resilient to speaker variability and background noise. This enhances the overall reliability and accuracy of speaker identification systems.

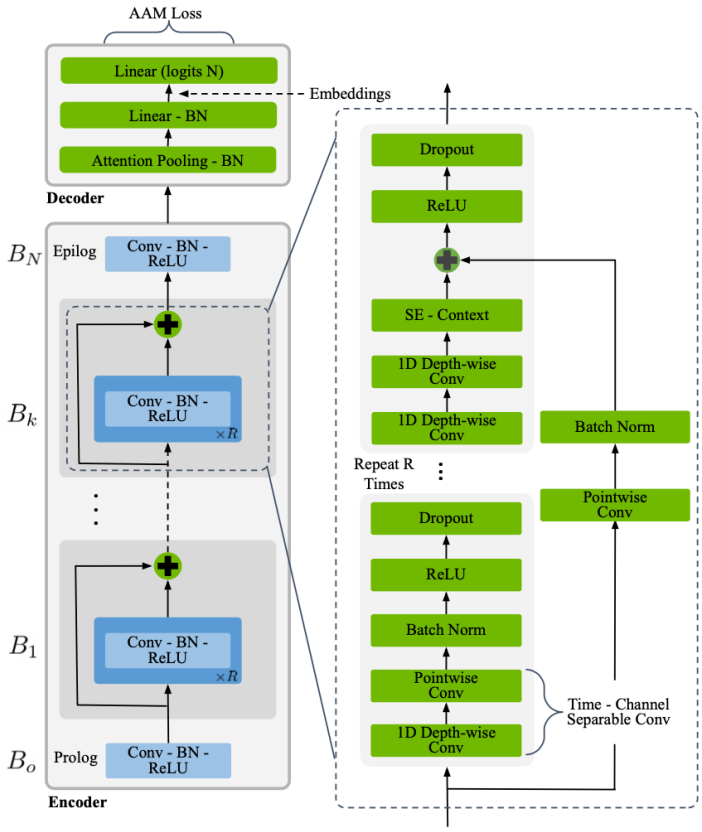
Batch Normalization: Batch Normalization is a technique used in deep learning to stabilize and accelerate training by normalizing the inputs of each layer. It reduces internal covariate shift by ensuring that the mean and variance of activations remain consistent during training, leading to faster

convergence and improved generalization. In speaker identification tasks, batch normalization helps handle the variability in audio data by normalizing features across batches, making the model more robust to differences in audio quality, background noise, and speaker variability. It also allows the use of higher learning rates and reduces the sensitivity to parameter initialization, which is critical for training deep models efficiently. By improving stability and performance, batch normalization plays a key role in modern speaker identification pipelines.

3.2 Architecture and Philosophy

In the TitaNet model, modules like convolution, ReLU, batch normalization, dropout, and attention work together to process audio data and identify speakers effectively. Convolutional layers extract local patterns from input features like spectrograms, capturing critical speaker-specific traits such as pitch and timbre. ReLU introduces non-linearity, enabling the model to learn complex relationships in these features. Batch normalization stabilizes and accelerates training by normalizing intermediate outputs, while dropout prevents overfitting by randomly deactivating neurons. Attention mechanisms enhance the model's focus by dynamically weighting important audio segments, improving robustness to noise or irrelevant data. Together, these components create a cohesive pipeline (Figure 7) that learns hierarchical, discriminative representations of speakers, ensuring accurate and generalizable identification.

Figure 7: TitaNet Architecture



3.3 Strengths

1. Scaling : Attention Architectures are able to scale to arbitrarily large sequences.
2. Pre-trained : Can leverage information from other datasets to improve performance. Greatly helps performance for scenarious with data-scarce applications, where the pre-training provides a solid foundation.
3. Self-learnt features : All features and transformations are data-based, no manual intervention required.

4. Parallel : Parallelized operations across batches to perform simultaneous predictions, boosting applications that require a central server.
5. No Over-fitting : Interventions like Dropouts and Batch Normalization stabilise learning across examples and prevent overfitting. This greatly aids performance in novel, unseen settings, off-the-shelf.

4 Experiments

4.1 Experimental Setup

NVIDIA's NeMo is a scalable, open-source framework for developing AI models in areas like large language models, speech applications, and multimodal tasks. It offers pre-trained models for Automatic Speech Recognition (ASR), Text-to-Speech (TTS), and speaker identification and verification, enabling accurate speech-to-text, text-to-speech, and voice-based authentication.

I implement the inference pipelines for two of Nvidia's NeMo models - *ecapa_tdnn* and *speakerverification_en_titanet_large*.

The experiment enables a user to do the following :

Exp 1 : Inference Pipeline

Input: <path_to_audio_file_1>, <path_to_audio_file_2>
Output: True if same speaker, False otherwise
Model: *ecapa_tdnn* or *speakerverification_en_titanet_large*

Furthermore, to enable real-life interactivity, I write a script with the *ffmpeg* library that enables the following :

Exp 2 : Data Conversion

Input: Any ffmpeg compatible audio file format (mp3, m4a, aac etc.)
Output: wav file, compatible with Exp 1
Library: *ffmpeg*

Now, one can record an audio real-time on a mobile application, transfer it and run Exp 1 with it.

Finally, I create a library of audio samples to enable user identification with a library.

Exp 3 : Library Querying

Input: <Voice Sample A>, <Library L>
Output: User Key corresponding to A
Via: Exp 1 + Library Querying + Library Generation

4.2 Observations and Results

I made some interesting observations while experimenting with models and audio samples:

- Model Size : TitaNet Large performs better than the small version. It is pre-trained on a larger corpus and is able to make better inferences.
- Language Invariance : TitaNet Large is able to classify the correct user irrespective of the language they speak.
- Tone : The models are not able to decipher the same person speaking in different "tones".
- Context length : The larger the sample, the more confident and correct the inference. It is easier to fault on a smaller observation.

The pipeline and experiments are open-sourced on my github. This repository makes it easy to access Nvidia's NeMo models without going through a complicated code-base. As a future extension, I plan to experiment with fine-tuning these models and porting this code to deployable applications.

References

[1] C. Zhan, W. Li, and P. Ogunbona, "Face recognition from single sample based on human face perception," in *2009 24th International Conference Image and Vision Computing New Zealand*, 2009, pp. 56–61. DOI: 10.1109/IVCNZ.2009.5378360.

[2] *Apple and siri*, http://www.apple.com/tw/ios/siri/, Last Accessed: 20-Nov-2024.

[3] K. Kushlev and M. R. Leitao, *The effects of smartphones on well-being: Theoretical integration and research agenda*, 2020. arXiv: 2005.09100 [cs.HC]. [Online]. Available: https://arxiv.org/abs/2005.09100.

[4] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, *A review of speaker diarization: Recent advances with deep learning*, 2021. arXiv: 2101.09624 [eess.AS]. [Online]. Available: https://arxiv.org/abs/2101.09624.

[5] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon 2016*, IEEE, Mar. 2016, 1–6. DOI: 10.1109/secon.2016.7506650. [Online]. Available: http://dx.doi.org/10.1109/SECON.2016.7506650.

[6] Y. Tang, *Deep learning using linear support vector machines*, 2015. arXiv: 1306.0239 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1306.0239.

[7] C. Viroli and G. J. McLachlan, *Deep gaussian mixture models*, 2017. arXiv: 1711.06929 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1711.06929.

[8] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[9] H. Wang and B. Raj, *On the origin of deep learning*, 2017. arXiv: 1702.07800 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1702.07800.

[10] J.-D. Wu and Y.-J. Tsai, "Speaker identification system using empirical mode decomposition and an artificial neural network," *Expert Systems with Applications*, vol. 38, no. 5, pp. 6112–6117, 2011, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.11.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410012509.

[11] F.-Y. Leu and G.-L. Lin, "An mfcc-based speaker identification system," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 1055–1062. DOI: 10.1109/AINA.2017.130.

[12] P. Zeman, *Discrete and fast fourier transform made clear*, 2019. arXiv: 1908.07154 [cs.DS]. [Online]. Available: https://arxiv.org/abs/1908.07154.

[13] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974. DOI: 10.1109/T-C.1974.223784.

[14] N. R. Koluguri, T. Park, and B. Ginsburg, *Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context*, 2021. arXiv: 2110.04410 [eess.AS]. [Online]. Available: https://arxiv.org/abs/2110.04410.

[15] N. R. Koluguri, J. Li, V. Lavrukhin, and B. Ginsburg, *Speakernet: 1d depth-wise separable convolutional network for text-independent speaker recognition and verification*, 2020. arXiv: 2010.12653 [eess.AS]. [Online]. Available: https://arxiv.org/abs/2010.12653.

[16] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: 1511.08458 [cs.NE]. [Online]. Available: https://arxiv.org/abs/1511.08458.

[17] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, 2015. arXiv: 1502.03167 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1502.03167.

[18] S. Chen, Y. Hou, Y. Cui, W. Che, T. Liu, and X. Yu, *Recall and learn: Fine-tuning deep pretrained language models with less forgetting*, 2020. arXiv: 2004.12651 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2004.12651.