 https://doi.org/10.3311/PPee.42209

Periodica Polytechnica Electrical Engineering and Computer Science, 69(4), pp. 444–451, 2025

Advanced Speaker Identification with CNNs and Maximum Likelihood Criterion

Ahcene Abed[1*], Aissa Amrouche[2], Redha Bendoumia[3], Abdelghafour Herizi[1], Ahmed Bouchekhlal[4]

[1] Laboratory of Electrical Engineering (LGE), University of M'sila, PO Box166 Ichebilia, 28000 M'sila, Algeria
[2] Department of Electronics, Faculty of Technology, University of Blida 1, Blida, Algeria
[3] Laboratory of Detection, Information and Communication, Department of Electronics, University of Blida 1, Algeria
[4] Higher School of Signals (HSS), Po Box 11 Kolea,42070, Tipaza, Algeria
[*] Corresponding author, e-mail: abed.ahcene@univ-msila.dz

Received: 11 September 2025, Accepted: 11 December 2025, Published online: 22 December 2025

Abstract

Speaker identification is a crucial topic in various fields, including linguistics, speech acoustic technology, and artificial intelligence. Despite the progress, speaker identification remains a challenge, particularly in acoustically noisy contexts or when the speakers are phonetically similar. Moreover, concerns regarding privacy and data protection frequently arise in speaker identification, particularly concerning the use of personal audio data. Signal processing and machine learning techniques have significantly advanced, improving the accuracy and resilience of voice recognition systems. New methods, including Convolutional Neural Networks (CNN), are advancing voice information extraction performance. This study aims to develop a Speaker Identification System based on deep learning techniques. These techniques have gained widespread recognition in the field of automatic acoustic signal processing. Many researchers have used convolutional neural networks, and the recognition phase is based on the cross-entropy criterion. This article proposes an advanced technique to combine convolutional neural networks with the maximum likelihood criterion. This proposed technique has yielded promising results when compared to traditional systems, such as Vector Quantization (VQ), and Gaussian Mixture Model (GMM). The suggested approach achieves an accuracy of 87.97% using all the data from the LibriSpeech corpus.

Keywords

speaker identification, MFCC, VQ, GMM, maximum likelihood

1 Introduction

Automatic Speaker Recognition (ASR) is a technique for identifying individuals based on their voice. It includes information regarding the speaker's identity. This category includes speaker verification (SV) and identification (SI) duties. An SV system determines whether the stated identity belongs to the client or an imposter. However, in identification, the system selects a person from a known group of persons. SI is divided into closed-set identification and open-set SI. It is further divided into text-independent and text-dependent systems. Several techniques have been developed for ASR. They include vectorial, statistical, connectionist, and predictive methods.

Because of their ability to discriminate, short-time features, such as MFCC, have been extensively employed in SISs. Long-term characteristics, like prosody, are essentially a speaker's ingrained qualities. In SI, pitch and energy work incredibly well, especially when the channels are mismatched, and the data is chaotic. The vocal tract's structural variations between speakers result in speaker specific information being included in prosodic elements. One unfeasible aspect of prosodic characteristics is the vast amount of information required for precise recognition. Much study has been done on SI in different backdrop contexts during the last few decades [1–3].

Finding the best feature set to represent the speaker is a significant difficulty in SI. As a result, decision-making relies heavily on the quality and quantity of information available. The message and identity are encoded on multiple levels of abstraction. Everyone has a distinct voice, pitch, and style of speaking. Humans frequently use spectrum, auditory information, and prosodic data in everyday discourse. However, none of this information is sufficient

Cite this article as: Abed, A., Amrouche, A., Bendoumia, R., Herizi, A., Bouchekhlal, A. "Advanced Speaker Identification with CNNs and Maximum Likelihood Criterion", Periodica Polytechnica Electrical Engineering and Computer Science, 69(4), pp. 444–451, 2025. https://doi.org/10.3311/PPee.42209

to distinguish between two individuals. So, all of this data must be used to determine a speaker's identity.

This study aims to illustrate the significant improvement of deep neural networks over previous methods. It focuses on closed-set speaker recognition in text-independent mode. Several techniques were employed, including VQ, GMM, and CNNs.

The remainder of this work is structured into three sections. The first section discusses ASR systems and introduces the relevant tasks. It contains three modules: audio analysis, system learning, and identification. The second section describes the procedures utilized to install the proposed systems, and finally, the outcomes are presented and reviewed.

2 Speaker recognition background

Fig. 1 depicts the basic structure of speaker recognition, a decision-making process that employs speech signal features to determine a speaker's identity among a collection of P speakers. The limitation on linguistic content eliminates one of the most variable sources. The SI architecture consists of three primary components: speech analysis, system training, and the identification phase.

The learning step consists of gathering multiple speech signals provided by the speaker. Signals must be recorded during several sessions and separated in time to approximate real conditions as closely as possible.

To make a comparative study, the proposed system is implemented using several approaches: VQ, GMM, and CNN.

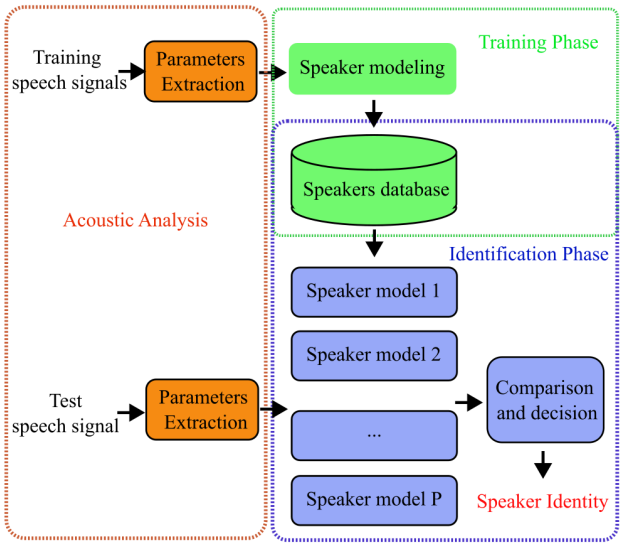

Fig. 1 Speaker Identification System

2.1 Mel Frequency Cepstral Coefficients (MFCC)

Fig. 2 depicts a modular cepstral representation built on a filter bank. High-pass filtering pre-emphasizes the spoken signal. It improves the upper frequencies of the spectrum. They are frequently minimized throughout the speech-production process.

$$x_p(t) = x(t) - a \cdot x(t-1) \tag{1}$$

α is generally taken as a value in the range [0.95, 0.98]. The use of this filter follows empirical experimentation. The speech signal is analyzed using a sliding Hamming window of 40 ms with a 50% overlap. In this case, the signal is assumed to be quasi-stationary.

$$H(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N}\right) & 0 \le n < N \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

Using the Fourier transform, the signal spectrum is obtained by transforming each frame into the frequency domain. A filter bank multiplies the latter. It is a series of band pass filters placed evenly across the Mel scale. The position of the central frequency of the filters is provided by:

$$f_{mel} = 1000 \cdot \frac{\log\left(1 + \dfrac{f}{1000}\right)}{\log 2} \tag{3}$$

The resulting frame is transformed into a logarithmic scale. Finally, a discrete cosine transform was used to get the cepstral coefficients. The expression of these coefficients is given by:

$$c_n = \sum_{k=1}^{K} S_k \cdot \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \quad n = 1, 2, \ldots, L \tag{4}$$

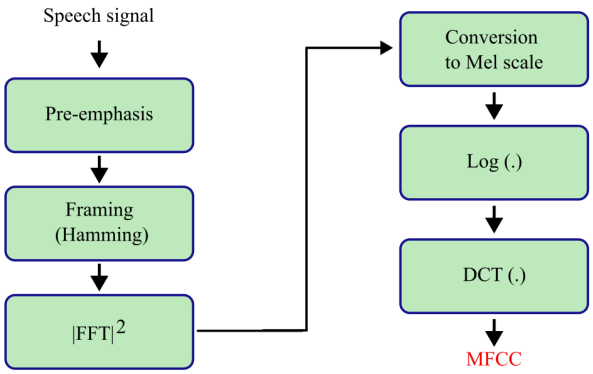

Fig. 2 MFCC Extraction

K is the number of spectral coefficients calculated before, S_k are the spectral coefficients, and L is the number of cepstral coefficients we want to calculate ($L \leq K$). In the speech signal, the dynamic information is different from one speaker to another. Cepstral derivatives often give this information. The first cepstral coefficient derivatives (Δ) give information about the vector's variation over time. However, the second cepstral coefficient derivative (ΔΔ) gives speech acceleration information. These coefficients are given by:

$$\Delta c_m = \frac{\sum_{k=-1}^{l} k^2 \cdot c_{m+k}}{\sum_{k=-1}^{l} |k|} \tag{5}$$

$$\Delta\Delta c_m = \frac{\sum_{k=-1}^{l} k^2 \cdot c_{m+k}}{\sum_{k=-1}^{l} k^2} \tag{6}$$

2.2 Vector quantization

Vector quantization comprises a finite set of D quantization levels, which are associated by a function to a vector of K samples produced by the source. The set D represents the vector quantization dictionary.

We can define vector quantization as applying the space Q of dimension K in R_K to a subset Y in R_K. Let $x = \{x_1, x_2, ..., x_K\}$ be a vector of the set Q, and the x quantification amounts to representingit by a close vector y_1 of a finite dictionary $Y = \{y_1, y_2, ..., y_M\}$. The dictionary Y is obtained by partitioning R_K into M classes C_i, each represented by its centroid y_i. Any vector $x \in C_i$ will be represented by y_i. This substitution introduces a quantization error. This error increases as the distance between X and Y is more excellent.

The iterative algorithm LBG allows us to build a dictionary from a set of learning vectors [4]. The size of the dictionary is an even number. The different steps of this algorithm are:

- Initialization: An initial dictionary comprises a single vector, the gravity center of the learning set. Let d_0 be this vector.
- Splitting: All the elements d in number $2k$ of the dictionary are "exploded" into two vectors. It is done, for example, by transforming each vector d into $d + \varepsilon$ and $d - \varepsilon$. Where ε is a random vector of variance adapted to the points of the cloud associated with d.
- Convergence: Apply the k-means algorithm to the dictionary of $2k + 1$ elements thus constituted. After convergence, we get an "optimal" dictionary of $2k + 1$ elements.
- Stop: Increment k. If $k > k_0$, the algorithm ends; otherwise, we return to the splitting step. Where k_0 is a value fixed in advance.

The acoustic vectors are encoded in N dictionaries corresponding to N different speakers. The quantization errors on each codebook are accumulated through the test signal. The average distortion on the codebook of speaker i is:

$$D^i = \frac{1}{L}\sum_{l=1}^{L} \min_{1\leq j\leq M} d\left(a_l, b_j^i\right) \tag{7}$$

The final decision for SI is given by:

$$\tilde{S}_V = \arg\min_{1\leq i\leq N} D^i \tag{8}$$

2.3 Gaussian mixture model

A weighted sum of many Gaussian distributions is called a GMM. Each one has a mean vector μ and a covariance matrix Σ. The following equation gives a Gaussian mixture:

$$P\left(x \mid \lambda\right) = \sum_{i=1}^{M} p_i b_i\left(x\right), \tag{9}$$

where x is a dimension D vector, M denotes the Gaussian components, w_i is the weight of the mixture, which verifies the condition $\sum_{i=1}^{M} w_i = 1$, and $b_i(x)$ is a multidimensional Gaussian distribution:

$$b_i\left(x\right) = \frac{1}{\left(2\pi\right)^{D/2}\left|\Sigma_i\right|^{1/2}} e^{-\frac{1}{2}\left(x-\mu_i\right)'\Sigma_i^{-1}\left(x-\mu_i\right)}. \tag{10}$$

A single model λ characterizes the Gaussian mixture. This model combines the mean vector μ, the covariance matrix Σ_i, and the mixture weight w_i. It is given by:

$$\lambda = \left\{p_i, \mu_i, \Sigma_i\right\} i = 1, ..., M \tag{11}$$

Several techniques are available to estimate the parameters of GMMs. Maximum likelihood (ML) estimation remains the most popular. Indeed, for a sequence of T vectors $X = \{x_1, x_2, ..., x_T\}$, assumed to be independent, the ML of the GMM is given by:

$$P\left(X \mid \lambda\right) = \prod_{t=1}^{T} P\left(x_t \mid \lambda\right) \tag{12}$$

Unfortunately, the analytical maximization of this function is not easy. The expectation-maximization algorithm solves this problem. The basis of this algorithm is to estimate, from an initial model λ_0, a new model λ in an iterative way. The estimated one λ must verify $p(X|\hat{\lambda}) \geq p(X|\lambda)$ condition. The two steps of this algorithm are :

- Expectation Step: Calculate the posterior probability

$$\Pr\left(i \mid x_t, \lambda\right) = \frac{p_i b_i\left(x_t\right)}{\sum_{k=1}^{M} p_k b_k\left(x_t\right)} \tag{13}$$

- Maximization Step : The estimated parameters may given by

$$\tilde{w}_i = \frac{1}{T}\sum_{t=1}^{T} p\left(i \mid x_t, \lambda\right) \tag{14}$$

$$\tilde{\mu}_i = \frac{\sum_{t=1}^{T} p\left(i \mid x_t, \lambda\right) x_t}{\sum_{t=1}^{T} p\left(i \mid x_t, \lambda\right)} \tag{15}$$

$$\tilde{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p\left(i \mid x_t, \lambda\right) x_t^2}{\sum_{t=1}^{T} p\left(i \mid x_t, \lambda\right)} - \tilde{\mu}_i^2 \tag{16}$$

In the identification phase, each speaker l (among speakers) is represented by its GMM λ_l. The goal is to identify the model that provides the highest probability for a particular observation sequence:

$$\tilde{S} = \arg\max_{1 \le k \le P} \Pr\left(\lambda_k \mid X\right) \tag{17}$$

Or

$$\tilde{S} = \frac{\arg\max_{1 \le k \le P} p\left(X \mid \lambda_k\right) p\left(\lambda_k\right)}{p\left(X\right)} \tag{18}$$

If all the apriori probabilities $p(\lambda_k)$ are equal and $p(X)$ are the same for all speakers. We use the logarithm and the independence between the observations to get:

$$\tilde{S}_G = \arg\max_{1 \le k \le P} \sum_{t=1}^{T} \log p\left(x_t \mid \lambda_k\right), \tag{19}$$

where $p(x_t|\lambda_k)$ is given by Eq. (9).

2.4 Convolutional Neural Network

CNNs are a subset of artificial Neural Networks (NN) that applies a mathematical process known as convolution. Their architecture consists of two major parts: convolutional and classification. A CNN is an NN with at least one layer that uses convolution rather than ordinary matrix multiplication [5]. They are commonly employed in image processing [6, 7]. In recent years, many researchers have integrated deep neural networks into speaker recognition systems [8, 9].

A CNN generally comprises a convolutional layer, a pooling layer, a fully connected layer, and the output or classification layer. Additional layers, such as the batch normalization [10] and the dropout layer [11], can increase the CNN's performance.

The convolution layer identifies edges, color patches, and other visual aspects. The convolution filter generates images known as output feature maps. Increasing the number of convolution filters enhances the number of identified features.

The pooling layer is a down-sampling procedure that comes after the convolution layer. The most common types of pooling are maximum and average pooling. In these circumstances, the pooling layer takes the maximum or average value.

The fully connected (FC) layer tries to connect a flattened input to all neurons. CNN designs finish with FC layers. Its primary goal is to improve objectives such as class scores.

After a convolution layer, the CNN architecture uses a nonlinear function called ReLU (Rectified Linear Unit). It replaces all negative values in the array with zero. ReLU aims to introduce nonlinearity in CNNs so that they perform better.

The posterior probabilities, obtained from the final classification layer for each frame, are utilized to compute the ultimate score using maximum likelihood criteria.

$$\tilde{S}_C = \arg\max_{1 \le k \le P} \sum_{t=1}^{T} \Pr\left(x_t \mid L_i\right) \tag{20}$$

Where P is the number of speakers, and $Pr(x_t|L_i)$ is a matrix containing estimates of the posterior probabilities that the t^{th} coefficients were the source of the i^{th} speaker observation.

3 Materials and methods

The experiments were conducted using an HP EliteBook 820 G1 laptop with an Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz processor and 8GHz RAM. This reflects the enormous amount of time required to implement the different phases of the proposed system.

3.1 Speakers corpora

To evaluate the proposed systems, we used the LibriSpeech corpus. This corpus, produced from audio books that are part of the LibriVox project, has 1000 hours of speech sampled at 16 kHz. This corpus is available for free download. It also provides speech model training data that has been created separately and pre-constructed speech models.

We used a set of 250 speakers, 125 of whom were male and 125 female. To ensure balance among all speakers, we took 10 minutes for each one, with 8 minutes used for learning and the remaining 2 minutes reserved for the testing phase.

3.2 Acoustic signal parameters

The speech signals are transformed into vectors of MFCC parameters.

Table 1 summarizes the characteristics of this operation:

4 Results and discussion

The evaluation of the SI performance is done as follows: the speech signal to be tested is first passed through the acoustic analysis module and then transformed into acoustic vectors $\{x_1, x_2, ..., x_t\}$. The series of acoustic vectors is partitioned into many segments, each including acousticvectors. The first two segments are:

$$\overbrace{x_1, x_2, \ldots, x_T}, x_{T+1}, x_{T+2}, \ldots \tag{21}$$

$$x_1, \overbrace{x_2, \ldots, x_T, x_{T+1}}, x_{T+2}, \ldots \tag{22}$$

Suppose the estimated identity matches the real identity. The frame is correctly identified. The accuracy gives the final performance evaluation:

$$\text{Acc} = \frac{\text{N}^{\text{br}}\text{of correct segments}}{\text{Total N}^{\text{br}}\text{of tested segments}} \tag{23}$$

4.1 Classical methods for SI

Fig. 3 illustrates the recognition step for the two methods. The input speech signal is transformed into MFCC Parameters. These parameters are used to find the speaker identity by ML criteria for GMM or minimum distortion with VQ.

Each speaker is modeled by a GMM, and VQ model. The Gaussian components number and the VQ dictionary are varied from 8, 16, 32, 64 and 128.

Fig. 4 shows the performance evaluation of the two SISs. It shows that the performance of both systems increases rapidly when the model order varies between 8 and 64. However, between 64 and 128, we observe that the identification performance stabilizes. This means that the minimum model order for these methods is 64. In addition,

Table 1 MFCC Extraction

Parameters	Value
Window	Hamming
Length	40 ms
Overlap	50%
MFCC Order	14 = 13 + 1(E)
First Derivatives	14Δ
Second derivatives	14ΔΔ

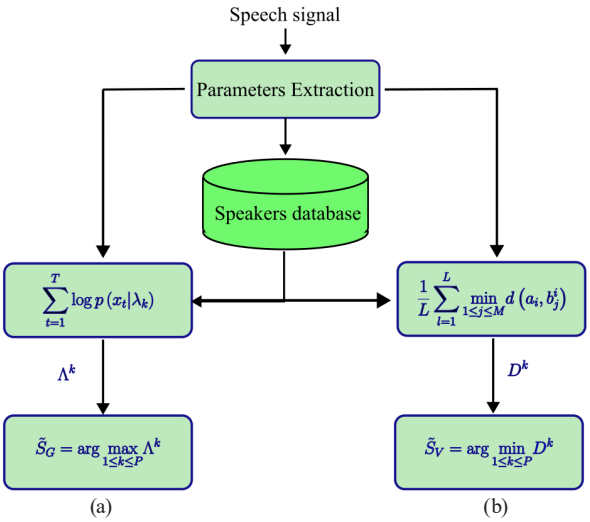

Fig. 3 Classical methods testing: (a) GMM, (b) VQ

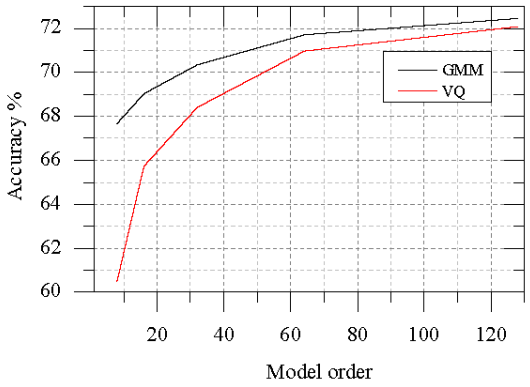

Fig. 4 Classical methods accuracy

GMM SI performs better than VQ. GMM SI shows its best performance of 72.47 % with 128 Gaussian components.

Table 2 summarizes the acquired results.

4.2 CNN speaker identification

Fig. 5 illustrates the CNN SI architecture. It is made up of 25 multiple interconnected layers of neuron units. The parameterization module transforms the speech signal into a 15 × 42 parameters matrix (MFCC). A concatenation step then groups all the columns into a 1 × 630 parameter vector. The system uses this vector as an input for training or recognition.

Table 2 Classical methods performances

Model Order	GMM	VQ
8	67.68%	60.49%
16	69.04%	65.74%
32	70.36%	68.41%
64	71.73%	70.97%
128	72.47%	72.09%

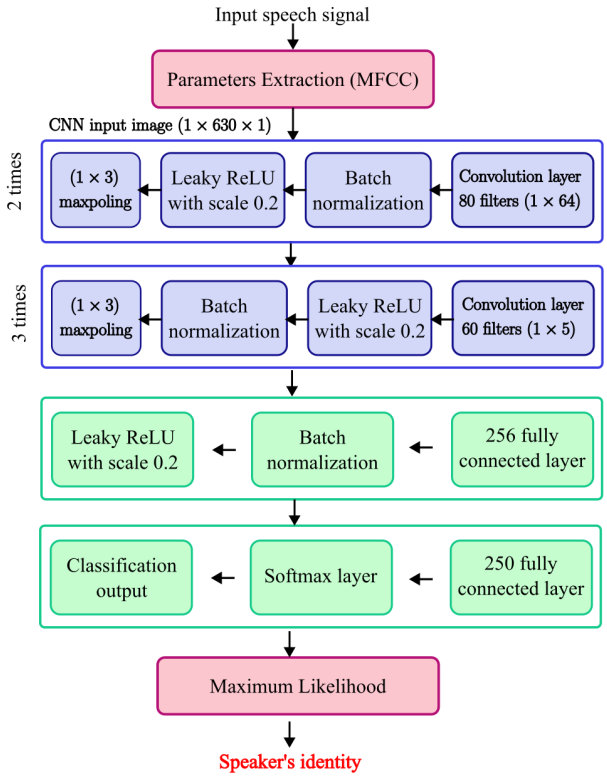

Fig. 5 CNN/ML-SI architecture

These data will transit through numerous hidden layers with varying levels of abstraction. Finally, the output layer assigns the final categorization to the input data. The CNN speaker characterization may be summarized as follows:

- The model includes five convolutional layers, followed by batch normalization, Leaky ReLU, and Maxpooling.
- There are 256 FC layers, which dropouts may precede.
- Nonlinear layers are most commonly utilized with the ReLU and the scaled exponential linear unit (SELU).
- Pooling layers use average or maximum pooling.
- A final softmax layer ascertains the output unit activation function for multi-class classification problems to calculate the conditional probabilities for classifying the input data.
- The training phase takes 1311 minutes and 9 seconds to complete.
- Identify the speaker with ML criteria.

Fig. 6 shows the learning phase of the CNN SIS. From the LibriSpeech corpus, a total of 250 speakers were used. The system undergoes training for 5 epochs, with each epoch including 4400 iterations, totaling 22000 iterations.

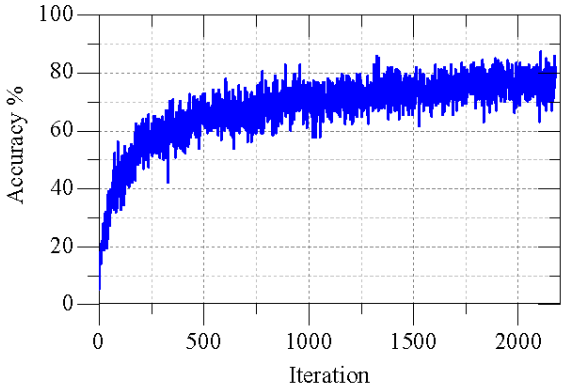

Fig. 6 CNN-SI training

Fig. 7 depicts the frame-level and ultimate-level accuracy vs. epoch number. The findings indicate that the system improves as epochs grow. The final accuracy score is 64.59% for CNN and 87.97% for CNN/ML. This technique performs well for SI.

To validate the choice of CNN model, we implemented the speaker identification system using various deep learning techniques: LSTM, BiLSTM, and RNN. The four systems were tested in the same conditions, like the number of speakers, the exact size of the training and test signals, and so on. Fig. 8 shows how CNNs perform well compared to other models.

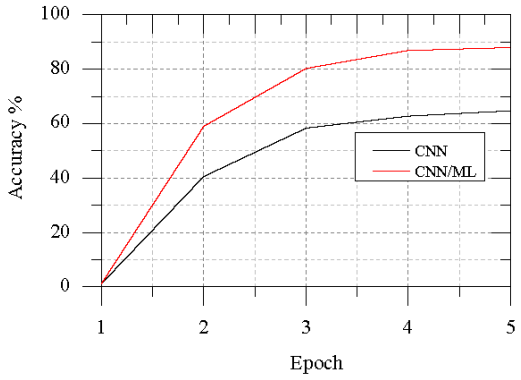

Fig. 7 CNN-SI performances

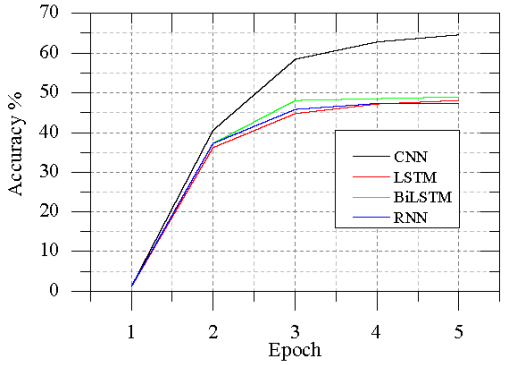

Fig. 8 Deep Learning-SI performances

We used the suggested model (CNN/ML) on TIMIT corpora to check the results we got. This corpus contains 6,300 phonetically balanced sentences from 630 speakers. 250 speakers were selected at random (125 male and 125 female).

The primary issue with TIMIT corpora pertains to the size of the speech signals. The majority of signals do not surpass 2 minutes. To address this issue, we implemented a data augmentation module utilizing Time Stretching and Noise Addition methods. This enhances the resilience and generality of speaker recognition models by mimicking various speaking circumstances.

Fig. 9 shows the accuracy obtained using the CNN/ML model with the TIMIT corpus. This accuracy is given for two types of data: MFCCs alone and MFCCs with their primary and secondary derivatives.

These findings demonstrate that the suggested approach works effectively with this corpus. For just six epochs, the highest rate is 84%. This shows that it is possible to use the proposed CNN/ML system in real applications.

4.3 Comparative study

We compared different approaches in terms of time spent on training and recognition. Table 3 summarizes this study. GMM is the finest method in terms of training time, taking only 96.68 seconds for each speaker. However, in the recognition phase, the CNN method performs faster than other methods.

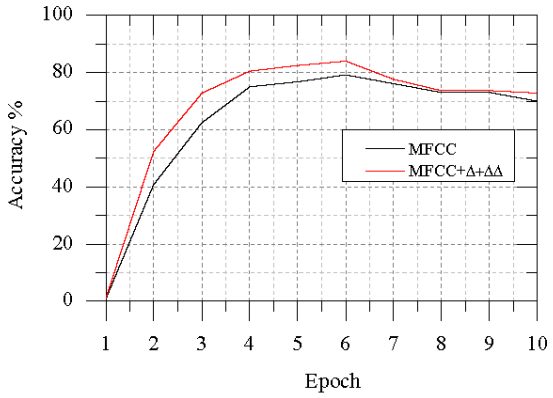

Fig. 9 CNN/ML performance with TIMIT corpora

Table 3 Comparative analysis of elapsed time

Method	Model Order	Training/speaker	Recognition
CNN	5 epochs	313 s	1.43 s
VQ	128	156 s	3.55 s
GMM	128	97 s	6.90 s

The results of this study were compared to those in the work of Saritha et al. [12]. Table 4 summarizes this comparison, demonstrating that the CNN/SGDM system, with an accuracy of 86.47%, significantly improved the SIS compared to other models.

5 Conclusions

Most speaker recognition research is based on traditional GMM, VQ systems. However, in this research, we developed a deep learning-based model, CNN, commonly used in image processing. This model requires a large dataset. Using the speech signal directly reduces the system's performance. First, we constructed the SIS using GMM and VQ. These systems achieve good accuracies, with 72.47%, 72.09%, respectively for GMM and VQ. The MFCC algorithm is used to parameterize speech signals. All models are assessed against the LibriSpeech speech corpus.

Finally, we used CNNs to implement the proposed system. An acoustic parameter extraction module (MFCC) increases their performance. The acoustic coefficient matrix is concatenated into a single vector to generate the input vector.

The final system had an accuracy rate of 64.69%. By applying the maximum likelihood criterion, the system shows a significant performance improvement with an accuracy of 87.97%. CNN/ML outperforms conventional models. We also demonstrated the effectiveness of the proposed model using the TIMIT speech corpus. This validates its implementation in real-world applications.

Future work will focus on increasing the suggested system's performance by combining it with another system or by improving the extraction of acoustic parameters. Furthermore, it will investigate its performance in real world noise contexts and telephone speech signals.

Table 4 Comparative analysis of different architecture

Ref	Model	Accuracy (%)
[12]	CNN	72.97
	SincNet	78.39
	RANet/Adam	79.51
	RANet/SGDM	82.57
This work	VQ	72.47
	GMM	72.09
	CNN	64.69
	CNN/ML	87.97

References

[1] Boubakeur, K. N., Debyeche, M., Amrouche, A., Bentrcia, Y. "Prosodic Modelling based Speaker Identification", In: 2022 2nd International Conference on New Technologies of Information and Communication (NTIC), Mila, Algeria, 2022, pp. 1–6. ISBN 978-1-6654-8974-4 https://doi.org/10.1109/NTIC55069.2022.10100506

[2] Singh, N., Khan, R., Shree, R. "MFCC and Prosodic Feature Extraction Techniques: a Comparative Study", International Journal of Computer Applications, 54(1), pp. 9–13, 2012. https://doi.org/10.5120/8529-2061

[3] Singh, N., Khan, R. "Extraction and representation of prosodic features for automatic speaker recognition technology", In: Fifth International Conference on AITMC (AIM-2015), Proceedings of Advanced in Engineering and Technology, Bangalore, India, 2015, pp. 1–7. https://doi.org/10.13140/RG.2.1.1673.5203

[4] Variani, E., Lei, X., McDermott, E., Moreno, I. L., Gonzalez-Dominguez, J. "Deep neural networks for small foot print text dependent speaker verification", In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 4052–4056. ISBN 978-1-4799-2893-4 https://doi.org/10.1109/ICASSP.2014.6854363

[5] Amrouche, A., Bentrcia, Y., Boubakeur, K. N., Abed, A. "DNN-Based Arabic Speech Synthesis", In: 9th International Conference on Electrical and Electronics Engineering (ICEEE), Alanya, Turkey, 2022, pp. 378–382. ISBN 978-1-6654-6754-4 https://doi.org/10.1109/ICEEE55327.2022.9772602

[6] Amrouche, A., Abed, A., Hezil, N., Zalagh, M., Boubakeur, K. N., Zitouni, A. "DNN-SVM Automatic ECG Signals Heartbeat Classification", In: 10th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 2023, pp. 95–99. ISBN 979-8-3503-0429-9 https://doi.org/10.1109/ICEEE59925.2023.00025

[7] Junliang, C. "CNN or RNN: Review and experimental comparison on image classification", In: 2022 IEEE 8th International Conference on Computer and Communications (ICCC), Chengdu, China, 2022, pp. 1939–1944. ISBN 978-1-6654-5051-5 https://doi.org/10.1109/ICCC56324.2022.10065984

[8] Ravanelli, M., Bengio, Y. "Speaker recognition from a raw waveform with sincnet", In: IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 2018, pp. 1021–1028. ISBN 978-1-5386-4334-1 https://doi.org/10.1109/SLT.2018.8639585

[9] Jalil, A. M., Hasan, F. S., Alabbasi, H. A. "Speaker identification using convolutional neural network for clean and noisy speech samples", In: First International Conference of Computer and Applied Sciences (CAS), Baghdad, Iraq, 2019, pp. 57–62. ISBN 978-1-7281-4048-3 https://doi.org/10.1109/CAS47993.2019.9075461

[10] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. "Dropout: A simple way to prevent neural networks from over fitting", Journal of Machine Learning Research, 15(56), pp. 1929–1958, 2014.

[11] Benaziz, B., Zitouni, A., Amrouche, A. "Children Autism Detection Using Residual CNN and EfficientNetB1", In: International Conference on Electrical Engineering and Advanced Technology (ICEEAT), Batna, Algeria, 2023, pp. 1–5. ISBN 979-8-3503-8348-5 https://doi.org/10.1109/ICEEAT60471.2023.10425848

[12] Saritha, B., Laskar, M. A., Laskar, R. H., Choudhury, M. "Raw waveform based speaker identification using deep neural networks", In: 2022 IEEE Silchar Subsection Conference (SILCON), Silchar, India, 2022, pp. 1–4. ISBN 978-1-6654-7100-8 https://doi.org/10.1109/SILCON55242.2022.10028890