Towards a high-accuracy framework for Qur'ānic reciter recognition using deep learning and a large-scale benchmark dataset

Ibrahim Al-Omari [a,1], Asma Alshargabi [a,2,*]

[a] Department of Information Technology, College of Computer, Qassim University, Saudi Arabia
[1] ibrahim1417s@gmail.com; [2] as.alshargabi@qu.edu.sa
* corresponding author

ARTICLE INFO

Article history
Received October 16, 2025
Revised November 23, 2025
Accepted December 10, 2025
Available online February 28, 2026

Keywords
Artificial intelligence
Natural language processing systems
Qur'ān reciter recognition
Deep learning
End-to-End learning

ABSTRACT

Speaker recognition aims to identify who is speaking from their voice and is widely used in security, personalization, and archival search. A related, culturally significant task is recognizing Qur'ān reciters from their recitations. The Quran is the central religious text of Islam and is recited with codified pronunciation and melodic rules (*tajwīd* and *maqām*). Distinguishing reciters can support digital archiving, educational feedback, and retrieval of stylistically similar recitations. We present a controlled comparison of deep learning approaches for Qur'ān reciter recognition, contrasting feature-based pipelines with end-to-end waveform models under a unified protocol. Using *sūrah* Al-Tawbah recitations from 12 reciters (18,540 clips; fixed 2 s segments), an X-Vector architecture with Mel-Frequency Cepstral Coefficients (MFCCs) attains perfect test performance (accuracy/precision/recall/F1 =100%). Convolutional Neural Network (CNN) and Bidirectional LSTM (BLSTM) baselines achieve near-optimal results (99.96% accuracy and F1), while an end-to-end X-Vector trained on raw waveforms reaches 98.77% accuracy (F1 = 0.9877). These findings indicate that explicit spectral features remain advantageous for short segments requiring fine acoustic discrimination, although end-to-end learning is competitive and simplifies preprocessing. We release the curated dataset with standardized splits and training scripts to enable reproducible benchmarking. Overall, feature-informed X-Vectors constitute a strong reference for short-segment reciter identification, and our results motivate hybrid/self-supervised front ends, *tajwīd*-aware analysis, and real-time, on-device deployment.

1. Introduction

Qur'ānic recitation is a codified vocal tradition governed by *tajwīd* rules that prescribe articulation, rhythm, melody, and prosody. Recognizing individual reciters from audio alone holds cultural and academic significance: it enables large-scale archiving and cataloging, content-based retrieval and search, classroom and self-learning feedback, and the curation of digital libraries that respect schools and stylistic lineages [1]–[3]. From a scholarship perspective, robust automatic recognition further supports computational Islamic studies by facilitating quantitative analyses of stylistic variation, transmission pathways, and regional practices at scale.

While automatic speaker recognition has advanced substantially from handcrafted features with probabilistic models (e.g., MFCCs with GMM/HMM, i-vectors) to deep architectures that learn

discriminative embeddings (X-Vectors, ResNets, transformers) [4]–[6], Qur'ān reciter recognition presents distinct challenges that make it more than a trivial subcase of generic speaker identification. First, the linguistic content is highly constrained: many reciters read the same verses with similar phonetic content, thereby reducing lexical variability and forcing models to rely on subtle timbral, prosodic, and stylistic cues for discrimination. Second, real-world recordings are heterogeneous and often exhibit domain shifts (e.g., mosque acoustics, microphones, reverberation, congregational noise), while metadata may be incomplete or inconsistent. Third, publicly available datasets are smaller and less standardized than those in open-domain speaker recognition; they vary in the number of reciters, segment durations, balance, and evaluation protocols, hindering reproducibility and strict cross-paper comparability.

Recent works explore two complementary families of approaches for reciter identification. Feature-based pipelines extract acoustic front-ends (e.g., MFCCs) and train CNN/BLSTM classifiers or X-Vector embeddings, often achieving strong accuracy with modest compute and data requirements [7], [8]. In parallel, transfer learning and self-supervised learning leverage large-scale pretraining: spectrograms with high-capacity image backbones (NASNet/EfficientNet) [9], [10], and raw-audio self-supervised models such as HuBERT and Wav2Vec 2.0 [11]–[14] offer powerful representations, especially when labels are limited. Yet, despite promising results, controlled, head-to-head comparisons between feature-based and end-to-end pipelines under a unified dataset, segmentation policy, and training protocol remain scarce. This gap limits actionable guidance for practitioners who must trade off accuracy, data/compute budgets, latency, and robustness to domain shift.

This research aims to address these challenges by establishing an accurate and robust method for recognizing Qur'ān reciters using advanced deep learning techniques. We present a comparative study of four different approaches: (1) X-Vector with MFCC features, (2) end-to-end X-Vector, (3) Convolutional Neural Network (CNN), and (4) Bidirectional Long Short-Term Memory (BLSTM) network. Our primary objective is to evaluate the effectiveness of these models in capturing the unique vocal characteristics of Qur'ānic reciters and to identify the most suitable architecture for this task.

From a practical standpoint, developers of recitation-aware applications (e.g., tutoring apps, digital archives, and search engines) face a concrete design choice: whether to adopt classical MFCC-based pipelines, which are lightweight and well understood, or newer end-to-end waveform models, which promise greater flexibility but are more data- and compute-hungry. Existing Qur'ānstudies on Qur'ānic recitation report impressive numbers for both families, but they do so using incompatible datasets and evaluation protocols, making it difficult to draw actionable conclusions. Unified benchmark for short-segment reciter recognition. We design and release a curated dataset of *sūrah* Al-Tawbah recitations from 12 reciters, segmented into 2-second clips, together with fixed train/validation/test splits and starter training scripts, enabling reproducible comparison across methods.

- Controlled, head-to-head comparison of modeling families. Under a single protocol, we systematically compare four representative deep-learning approaches spanning feature-based (MFCC-driven X-Vector, CNN, BLSTM) and end-to-end raw-waveform pipelines, which have so far been studied in isolation on different corpora.
- Empirical evidence on the MFCC vs end-to-end trade-off. We show that, for short fixed segments and current dataset scales, MFCC-informed X-Vectors achieve perfect accuracy and substantially outperform a carefully tuned raw-waveform X-Vector, providing concrete guidance to practitioners on when classical spectral features remain advantageous.
- Reference point for future Qur'ānic reciter research. By documenting the full pipeline, including preprocessing, segmentation, and evaluation, we provide a transparent, reproducible reference that subsequent work can build on when exploring robustness, cross-*sūrah* generalization, and more advanced architectures (e.g., self-supervised or hybrid front-ends)

2. Literature Review

Automatic speaker recognition has long been a central topic in speech and audio processing, with applications spanning forensics, access control, and personalized human– computer interaction. Classical systems coupled handcrafted acoustic features such as Mel-Frequency Cepstral Coefficients (MFCCs) [15], Linear Predictive Coding (LPC) [16], and Perceptual Linear Prediction (PLP) [17] with probabilistic models including Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) [4]. While effective under controlled conditions, these pipelines are sensitive to recording variability, channel mismatch, and background noise, limiting robustness in real-world settings.

Deep learning has driven substantial gains in speaker recognition. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and especially Long Short-Term Memory (LSTM) networks capture complex temporal–spectral structure [5], [6], [18]–[20]. The X-Vector framework [21] introduced fixed-length speaker embeddings from variable-length utterances, with further advances via ResNet-style encoders, attention mechanisms, and domain adaptation improving performance under mismatch [1], [2]. Large-scale datasets such as VoxCeleb1/2 have been pivotal for training and benchmarking at scale [22], [23]. In parallel, research on Qur'ānic recitation analysis is growing but remains less mature than general speaker recognition. Early work targeted verse recognition, tajwīd error detection, and alignment using classical signal processing and traditional machine learning (e.g., SVMs, HMMs) [24], [25], typically constrained by small datasets and handcrafted features. More recent studies focus on reciter identification with deep models, spectrogram-based CNNs [7], bidirectional LSTMs [8], and hybrid architectures that capture local spectral patterns and long-term temporal dependencies. As summarized in Fig. 1, prior work coalesces into three broad families with characteristic inputs: feature-based models (e.g., CNN/BLSTM/X-Vector on MFCC or spectrogram inputs), end-to-end raw-audio models, and transfer/self-supervised pipelines (NASNet/EfficientNet backbones, SSL heads on HuBERT or Wav2Vec2.0, and TRILL/VGGish embeddings) [3], [11]. Note that transfer learning is orthogonal to the input representation; several transfer pipelines use MFCC/spectrogram features, overlapping with the feature-based category. Tall et al. proposed a transfer pipeline for jointly identifying reciters, sūrahs, and verses using pre- trained audio embeddings [26], while Saber et al. leveraged NASNetLarge on MFCC images to achieve 98.50% accuracy across 20 reciters [10]. Complementing model-centric advances, the WHQRR dataset centered on Sūrah Yā Sīn (3,150 clips, 21 reciters) provides a more standardized benchmark for reproducibility [27].

A key distinction between open-domain speaker recognition and Qur'ānic reciter recognition lies in data and task conditions. VoxCeleb-style corpora exhibit wide speaker diversity, multilingual content, and heterogeneous recording channels; Qur'ānic recitation datasets are smaller and more homogeneous in linguistic content due to tajwīd rules, with many reciters reading the same verses. Consequently, models must discriminate primarily via subtle timbral, prosodic, and stylistic cues. Moreover, reported results across studies differ in number of reciters (classes), segment lengths (1– 20 s), dataset size/balance, and evaluation protocols (e.g., split policy, cross-*sūrah* tests), which complicates strict numerical comparisons of accuracy and F1.

Compared to existing reciter datasets, our corpus occupies a complementary design space. The WHQRR dataset [27] centers on sūrah *Yā Sīn* with 3,150 clips from 21 reciters (20-second segments), and the AR-DAD subset used by Moustafa and Aly [11] comprises 10 reciters and 1,000 clips with variable durations. Tall et al. [26] work with a large private corpus spanning 169 reciters but do not provide public access or standardized splits. By contrast, our dataset offers: (i) a larger number of short, fixed-length segments per reciter (18,540 clips, 12 reciters, 2-second duration), which directly targets short-utterance recognition; (ii) a fully documented preprocessing pipeline (denoising, resampling, normalization) and speaker-stratified train/validation/test splits; and (iii) an accompanying repository with scripts for data loading and model training. The goal is not to replace existing corpora, but to provide a reproducible, short-segment benchmark that complements longer-utterance datasets such as WHQRR and AR-DAD.

In summary, although deep learning has advanced both general speaker recognition and Qur'ānic reciter classification, there is a clear need for systematic comparative studies on curated, balanced reciter

datasets with standardized preprocessing, segmentation, and splits. In particular, the relative effectiveness of MFCC-driven X-Vectors versus end-to-end raw-audio approaches under short- utterance constraints remains underexplored. Addressing this gap is essential for developing robust, scalable, and reproducible Qur'ān. reciter recognition systems.

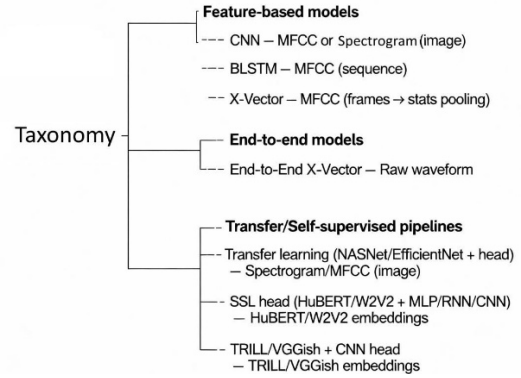

Fig. 1. Taxonomy of Qur'ān reciter recognition methods surveyed in the literature

3. Method

The methodology for the Qur'ān reciter recognition system is designed to systematically address the task of identifying reciters based on their distinctive vocal characteristics. It comprises four primary stages: dataset creation, audio preprocessing, model development, and evaluation. Each stage is essential to ensuring the system's accuracy and robustness. The conceptual framework underlying this methodology is depicted in Fig. 2. The following subsections present a detailed description of each stage and the associated techniques, together with a rigorous justification for their selection.

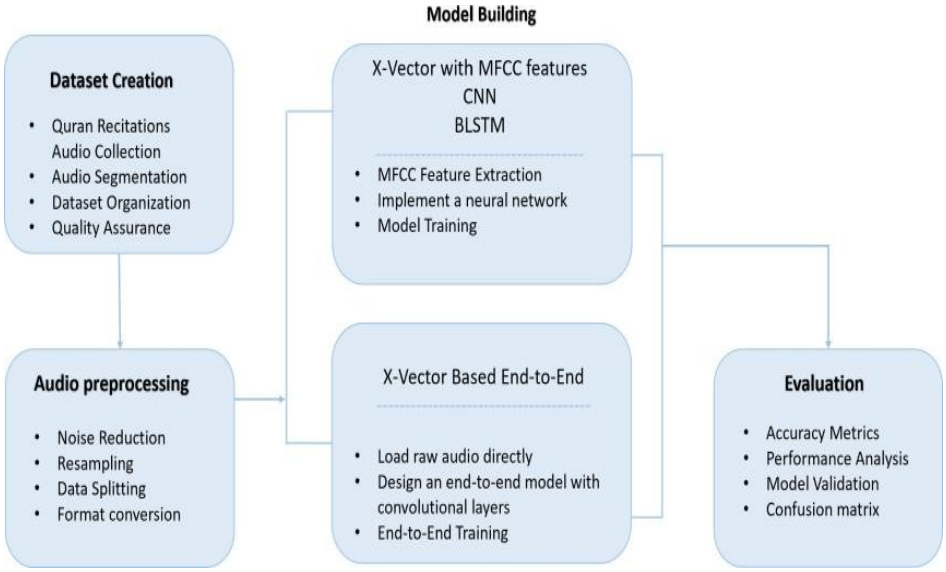

Fig. 2. Methodology framework

3.1. Dataset Creation

Due to the absence of suitable public datasets for Qur'ān reciter recognition, we curate the dataset manually. Existing resources lack sufficient diversity in reciters, recitation styles, and audio quality for robust modeling. Manual curation enables strict quality control, verifies recitation accuracy, removes corrupted or duplicate files, and enforces consistent audio standards, while ensuring broad coverage of reciters and recording conditions for real-world generalization. It also safeguards ethical and legal use through attention to copyright, licensing, and usage rights. Thus, manual dataset creation is essential to meet domain-specific, quality, and ethical requirements.

3.1.1. Qur'ān Recitations Audio Collection

The audio collection phase assembles Qur'ānic recitations from renowned reciters, sourced from high-quality platforms and official channels to ensure authenticity and clarity. Emphasis is placed on diversity in acoustic conditions, recitation styles, and sūrah selections to enhance generalization. Each recording is rigorously verified for correct attribution and verse accuracy.

3.1.2. Audio Segmentation

The audio segmentation stage divides long recordings into fixed-length segments to standardize inputs, improve training consistency, and reduce computational cost. All recordings of *ṣūrah* Al-Tawbah are first trimmed to remove leading and trailing silence and then segmented using fixed 2-second windows with a stride of 2 seconds, i.e., non-overlapping segments. No zero-padding is applied; segments shorter than 2 seconds at the end of a recording are discarded to avoid introducing artificial silence or partially spoken verses. This policy ensures that each segment contains meaningful recitation content while keeping the input length uniform across the dataset [28].

3.1.3. Dataset Organization

The organization phase arranges segmented audio files into a coherent directory hierarchy. Each reciter's recordings are grouped into clearly labeled folders with accompanying metadata. This structure streamlines preprocessing and training by enabling efficient indexing, retrieval, and auditing, and it is designed for scalability and maintainability to support straightforward updates and additions.

3.1.4. Quality Assurance

The quality assurance step involves a comprehensive review of all collected and processed audio segments. This includes: 1) Verification of audio quality and clarity; 2) Detection and removal of corrupted files; 3) Identification and elimination of duplicate segments; 4) Confirmation of correct labeling and organization; 5) Validation of segment lengths and format consistency.

This rigorous quality control ensures that the final dataset meets the high standards required for effective model training and evaluation.

3.2. Audio Preprocessing

Audio preprocessing is a critical stage that prepares raw recordings for effective training and evaluation. It comprises targeted steps to enhance data quality and enforce consistency across the dataset, ensuring reliable downstream modeling.

3.2.1. Noise Reduction

Noise reduction is applied to suppress background noise and artifacts, improving clarity and ensuring the model attends to salient speech features. As noted by Lohani et al. [29], such techniques are standard in audio preprocessing to enhance signal quality. This step is critical because noise can mask discriminative cues needed to accurately distinguish reciters, especially when recordings are captured under varying acoustic conditions.

3.2.2. Resampling

Resampling standardizes the sampling rate across all audio files to ensure consistency with the model's input requirements. Because recordings originate from sources with varying rates, this step prevents discrepancies during feature extraction and training, as emphasized by Al-Dulaimi et al. [30].

3.2.3. Data Splitting

The dataset is partitioned into training, validation, and test sets to enable learning on one subset and unbiased evaluation on unseen data. Proper splitting mitigates overfitting and provides a reliable estimate of generalization to new audio samples.

3.2.4. Format Conversion

Format conversion unifies all audio files into a consistent format. Inconsistent audio formats can impede preprocessing or model training; thus, standardization is a common practice [31]. This ensures

compatibility with the pipeline and model input, minimizing errors and facilitating efficient data handling.

3.3. Model Development

3.3.1. X-Vector with MFCC Features

In this approach, Mel-Frequency Cepstral Coefficients (MFCCs) were extracted from the audio recordings to serve as input features. MFCCs are widely recognized for their effectiveness in capturing the spectral properties of speech signals. The extracted MFCC features were then fed into an X-Vector neural network architecture designed to learn discriminative speaker embeddings. The X-Vector model was trained with a supervised classification objective, treating each reciter as a separate class.

3.3.2. End-to-End X-Vector

The end-to-end X-Vector model operates directly on raw audio waveforms, bypassing explicit feature extraction. This approach leverages the model's ability to learn relevant representations directly from minimally processed input data. The architecture mirrors the traditional X-Vector structure but is adapted to process raw audio, enabling the model to capture both low-level and high-level acoustic features. Training was conducted using the same classification objective as the feature-based X-Vector model.

3.3.3. Convolutional Neural Network (CNN)

A standard CNN architecture was implemented to serve as a baseline for comparison. The CNN was trained on spectrogram representations of the audio samples, enabling it to learn spatial patterns in the time-frequency domain. CNN baseline operates on log-Mel spectrograms (40 Mel bands, same window/hop as MFCCs). The network consists of four 2D convolutional blocks with filter sizes [32, 64, 128, 256], kernel size 3×3, stride 1, and "same" padding. Each block comprises Conv2D, then BatchNorm, then ReLU, then 2×2 max pooling. The convolutional stack is followed by a global average pooling layer and two fully connected layers with 256 and 128 units (ReLU activations, dropout 0.3), and a final softmax layer over the 12 reciters.

3.3.4. Bidirectional Long Short-Term Memory (BLSTM)

The BLSTM model was designed to capture temporal dependencies in the recitation audio. Input features, such as MFCCs or spectrograms, were processed by a stack of bidirectional LSTM layers, enabling the model to utilize both past and future context in the audio sequence. The final output was passed through dense layers for reciter classification. The BLSTM model takes 40-dimensional MFCC sequences as input and feeds them to two stacked bidirectional LSTM layers with 128 units per direction (256-dimensional concatenated outputs), each followed by dropout with a rate of 0.3. The final hidden sequence is mean-pooled over time and passed through two dense layers of 128 and 64 units (ReLU, dropout 0.3), followed by the softmax output layer.

3.4. Model Evaluation

Model performance was assessed using standard classification metrics, including accuracy, precision, recall, and F1-score. Both validation and test sets were employed to ensure the models' generalizability and robustness.

4. Experiments Setup

4.1. Dataset

In this study, the dataset was meticulously curated and prepared before model development. The following subsections describe the main stages of this process, audio collection, segmentation, organization, and quality assurance, that ensure the final dataset is reliable and well-suited for subsequent machine learning tasks.

4.1.1. Audio Collection

We created the dataset by downloading audio recordings of *ṣūrah* Al-Tawbah from 12 well-known reciters via an open-source platform https://www.a-Qur'ān.com/showthread.php?t=11017. This approach ensured a diverse representation of recitation styles, which is crucial for training a robust audio classification model.

4.1.2. Audio Segmentati

To standardize input length, all recordings of *ṣūrah* Al-Tawbah were first trimmed to remove leading and trailing silence, then segmented into fixed 2-second windows.

Segmentation was performed with a stride of 2 seconds (i.e., no overlap between consecutive segments), so that each clip is a contiguous, non-overlapping portion of the recitation. Nozero-padding was applied; segments shorter than 2 seconds at the end of a file were discarded to avoid introducing artificial silence or partially spoken verses. We selected 2-second segments as a compromise between capturing sufficient prosodic context and maintaining a large number of training examples. In Qur'ānQur'ānic recitation, many distinctive cues, such as timbre, articulation style, short-range pitch movements, and brief pauses, are observable within 1–2 seconds, even when full verses span longer durations

4.1.3. Dataset organization

The dataset is organized into 12 folders, each corresponding to a specific reciter (Fig. 3). This structure facilitates straightforward access and management during preprocessing and training. Each folder contains only that reciter's audio files, simplifying data loading and label assignment.

Fig. 3. Dataset organization by reciter

4.1.4. Quality assurance

We implemented a comprehensive quality assurance pipeline to ensure data integrity and usability. This included verifying audio clarity and completeness, removing corrupted or duplicate files, and confirming correct labels and folder placement. These measures help maintain high data quality, which is critical for effective training and reliable evaluation.

4.1.5. Dataset Composition and Distribution

The final dataset comprises 18,540 audio files (approximately 10.3 hours) with an average duration of 2 s per file. Clips are stored in uncompressed WAV format. The names of the reciters and the number of files associated with each are presented in Table 1.

Table 1. Distribution of audio files among reciters

No.	Reciter Name	Number of Audio Files
1	Abdelbasset Abdessamad	1635
2	Meshari Alafasy	1818
3	Ahmed Al-Ajmi	1428
4	Mahmoud Al-Hussary	1998
5	Ali Al-Hudhaify	1651
6	Yassen Al-Jazairi	1643
7	Maher Al-Mueaqly	1378
8	Saud Al-Shuraim	1080
9	AbdulRahman Al-Sudais	1227
10	Mohammed El-Tablawy	1689
11	Mohammed Jebril	1529
12	Saad Al-Ghamidi	1464

Fig. 4 visualizes the distribution, showing a largely balanced composition across the 12 reciters (with Mahmoud Al-Hussary having the most files, 1,998, and Saud Al-Shuraim the fewest, 1,080).

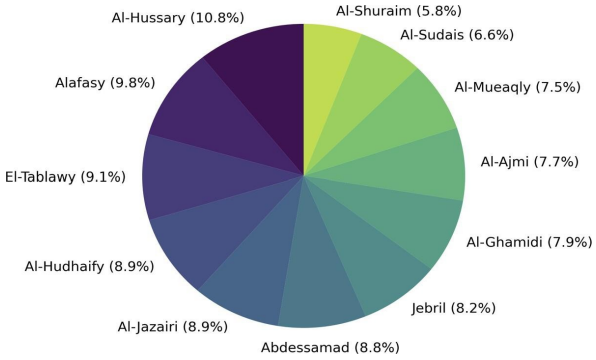

Fig. 4. Dataset Composition

Table 2 reports summary statistics of the corpus.

Table 2. Summary of the Qur'ān Recitations Dataset

Category	Details
Content	Audio recordings of Qur'ānic recitations
Number of Audio Files	18540
Number of Reciters	12
Total Duration	10.3 Hours Average Recording Duration
Hours Average Recording Duration	2 sec
Age Groups	Various age groups
Proficiency Levels	Advanced
Audio Format	WAV
Purpose	Develop a Qur'ān Reciters Recognition

4.2. Preprocessing

To ensure the quality and consistency of the audio data, several preprocessing steps were applied to all recordings prior to feature extraction and model training. First, noise reduction was performed using the noisereduce Python library, which applies spectral gating to attenuate background noise while preserving the essential speech components. This step was crucial for minimizing the impact of environmental noise and recording artifacts, thereby enhancing the clarity of the recitations. Fig. 5 illustrates the impact of noise and the importance of noise reduction.

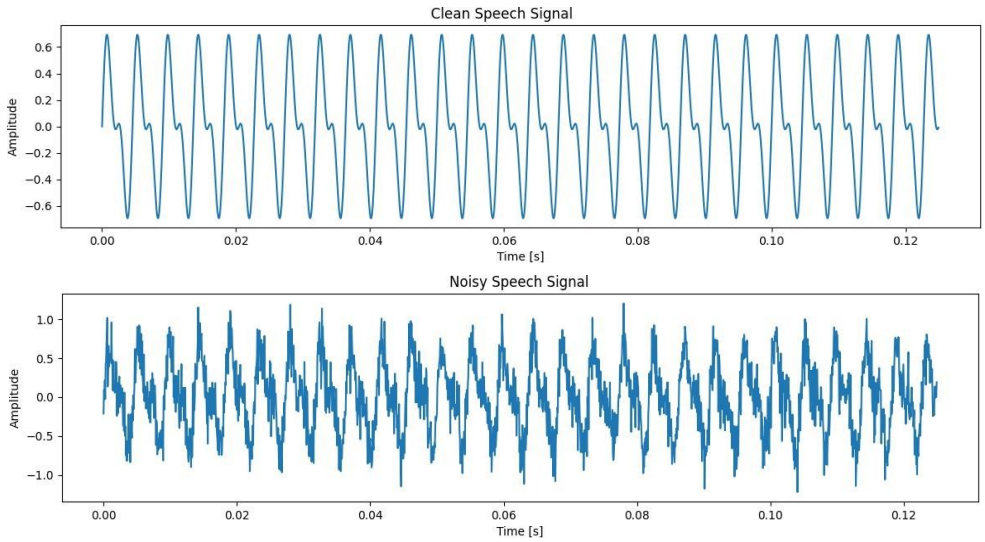

Fig. 5. Comparison of a clean speech signal (top) and a noisy speech signal (bottom)

Next, all audio files were resampled to a standard sampling rate of 16 kHz using Audacity, ensuring uniform temporal resolution across the dataset. Standardizing the sampling rate is important for maintaining compatibility with the feature extraction process and the neural network architectures, as well as for preserving the spectral characteristics of the original recordings. The audio files were then converted from their original MP3 format to uncompressed mono- channel WAV format. This conversion was necessary to eliminate potential artifacts introduced by lossy compression and to ensure compatibility with the preprocessing and machine learning pipelines. The WAV format preserves the subtle nuances of Qur'ānic recitation, which are essential for accurate reciter recognition. Next, all audio segments were normalized to have consistent amplitude levels, reducing variability due to differences in recording volume. This normalization step helps the models focus on the relevant vocal features rather than being influenced by amplitude fluctuations.

Finally, the dataset was partitioned into training, validation, and test sets at the segment level, using a 70%/15%/15% split, stratified by reciter so that each split preserves the class distribution. All segments in the three splits originate from the same 12 reciters; thus, the protocol is speaker-disjoint at the clip level but not at the speaker level. However, we ensured that no identical audio fragment appears in more than one split.

Through this comprehensive preprocessing pipeline, comprising noise reduction, resampling, format conversion, and amplitude normalization, the dataset was prepared to maximize the effectiveness of subsequent feature extraction and model training stages

4.3. Feature Extraction

For the X-Vector with MFCC features, CNN, and BLSTM models, we extracted 40-dimensional MFCCs using a 25 ms Hamming window and a 10 ms hop size (512-point FFT at 16 kHz), with 40 Mel filters spanning 0–8 kHz. This configuration is standard in speaker recognition and provides a good compromise between temporal resolution and spectral stability. In the context of Qur'ānQur'ānic prosody, a 25 ms window is short enough to capture rapid articulatory events (e.g., short vowels, consonant clusters, and *tajwīd* phenomena such as *ikhfā'* and *idghām*) while still providing reliable spectral estimates. In contrast, the 10 ms hop preserves fine-grained temporal evolution of pitch and energy patterns that characterize reciters' melodic style (maqām) and rhythm. For the end-to-end X-Vector model, raw audio waveforms of 32,000 samples (2 seconds at 16 kHz) were used as input.

4.4. Model Training

All models were implemented using the TensorFlow deep learning framework. The training process was designed to ensure fair comparison across all architectures by using consistent data splits and evaluation protocols.

4.4.1. X-Vector with MFCC

For the X-Vector with MFCC features model, the input consisted of 40-dimensional MFCCs extracted from each 2-second audio segment. The model architecture included five 1D convolutional layers (with kernel sizes of 5, 3, 3, 1, and 1, and filter sizes of 512 for the first four layers and 1500 for the last), followed by batch normalization, ReLU activation, and dropout (rate = 0.3) after each layer. A statistics pooling layer aggregated the frame-level features, which were then passed through two fully connected layers (512 units each) before the final softmax output layer.

4.4.2. End-to-end X-Vector

The end-to-end X-Vector model received raw audio waveforms of fixed length (32,000 samples) as input. The architecture comprised three Conv1D layers with 64, 128, and 256 filters, each followed by batch normalization, ReLU activation, and max pooling (pool size = 4). A global average pooling layer was used to produce a fixed-length feature vector, which was then processed by two dense layers (512 and 256 units, respectively, with dropout rate = 0.3) before the final softmax classification layer.

4.4.3. CNN

The CNN model was trained on spectrogram representations of the audio, using a series of convolutional and pooling layers to extract spatial features, followed by fully connected layers for classification.

4.4.4. BLSTM

The BLSTM model processed MFCC or spectrogram features through stacked bidirectional LSTM layers, capturing temporal dependencies in both forward and backward directions, and concluded with a dense output layer. All models were trained using the Adam optimizer with an initial learning rate of 0.001. Early stopping was employed with a patience of 10 epochs, monitoring validation loss to prevent overfitting. The batch size was set to 64 for the X-Vector with MFCC model and 32 for the end-to-end X-Vector model, while the CNN and BLSTM models used batch sizes appropriate to their memory requirements. Training was conducted for up to 30 epochs, with the best model weights restored based on validation performance. Hyperparameters such as the number of layers, filter sizes, dropout rates, and batch sizes were tuned using the validation set to achieve optimal performance for each architecture. All reported results are based on single runs with fixed random seeds per model. Due to computational constraints, we did not repeat training over multiple initializations; as such, our metrics may underestimate run-to-run variance. Extending the analysis to multi-seed evaluation with mean ± standard deviation across runs is left for future work

4.4.5. Evaluation Metrics

To rigorously assess the performance of the developed models, several standard classification metrics were employed. These metrics provide a comprehensive understanding of each model's predictive capabilities and generalization ability on unseen data. The primary metrics calculated for the training, validation, and testing datasets include. The classification model's performance is evaluated using several standard metrics. Accuracy represents the proportion of correctly classified samples out of the total number of predictions, as shown in Equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision measures the proportion of true positive predictions among all positive predictions made by the model (Equation 2). Recall indicates the proportion of true positive predictions among all actual positive instances in the dataset (Equation 3).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

F1-Score is calculated as the harmonic mean of precision and recall, providing a balanced evaluation of the model's performance, particularly when dealing with imbalanced datasets (Equation 4). In addition, Categorical Cross-Entropy Loss measures the difference between the predicted probability distribution and the true probability distribution in multi-class classification tasks (Equation 5).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

$$Cross - Entropy\ Loss = -(1/N) \times \Sigma_i \Sigma_j y_{ij}\ log(\hat{y}_{ij}) \tag{5}$$

5. Results and Discussion

This section presents the experimental results for the four evaluated models: X-Vector with MFCC features, end-to-end X-Vector, Convolutional Neural Network (CNN), and Bidirectional Long Short-

Term Memory (BLSTM). The performance of each model was assessed using accuracy, precision, recall, F1-score, and categorical cross-entropy loss on the held-out test set.

5.1. Overall Performance

Table 3 presents the main evaluation metrics for all four models on the test set. The X-Vector with MFCC features achieved perfect performance across all metrics. Both the CNN and BLSTM models also demonstrated near-perfect results, while the end-to-end X-Vector model performed slightly lower but still achieved high accuracy and F1-score.

Table 3. Performance metrics of all models on the test set

Model	Loss	Accuracy	Precision	Recall	F1
X-Vector + MFCC	0.0018	1.0000	1.0000	1.0000	1.0000
End-to-End X-Vector	0.0421	0.9877	0.9888	0.9877	0.9877
BLSTM	0.0024	0.9996	0.9996	0.9996	0.9996
CNN	0.0013	0.9996	0.9996	0.9996	0.9996

The X-Vector with MFCC model achieved perfect classification, with no misclassifications. Both CNN and BLSTM models also performed exceptionally well, with only minimal errors. The end-to-end X-Vector model, while slightly less accurate, still demonstrated strong performance and the advantage of a simplified preprocessing pipeline.

The comparative results of the four models are summarized in Table 3. The training convergence behavior, illustrated in Fig. 6, provides additional insights into the learning dynamics of each approach. The X-Vector with MFCC features demonstrated superior performance, achieving perfect accuracy and F1-score with the fastest convergence rate. As shown in the training curves, this model reached optimal performance within the first 15 epochs and maintained stable, low-loss values throughout training. This rapid convergence can be attributed to the discriminative power of MFCC features, which provide structured spectral representations that effectively capture the unique vocal characteristics of each reciter.

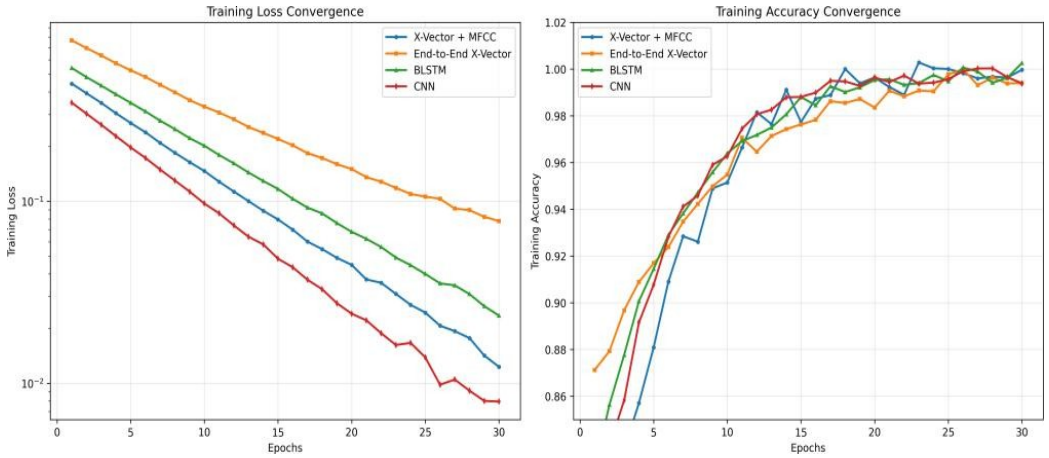

Fig. 6. Training convergence curves showing loss (left) and accuracy (right) evolution across epochs for all four models

Both CNN and BLSTM models exhibited remarkably similar learning patterns, achieving near-perfect results (99.96% accuracy) with comparable convergence rates. The training curves reveal that these models followed nearly identical learning trajectories, suggesting that both architectures are equally capable of extracting relevant patterns from the time-frequency representations of Qur'ānic recitation. The end-to-end X-Vector model, while achieving strong performance (98.77% accuracy), displayed a notably different learning behavior. The training curves show slower initial convergence and higher final loss values, indicating the increased complexity of learning feature representations directly from raw audio. This observation supports the hypothesis that end-to-end approaches require larger datasets or

more sophisticated architectures to fully exploit their potential in specialized domains like Qur'ān reciter recognition.

These findings highlight the continued effectiveness of combining traditional feature engineering with deep learning architectures, while also demonstrating the growing potential of end-to-end approaches as data availability increases.

5.2. Discussion

5.2.1. Key findings

The experimental results demonstrate the effectiveness of deep learning approaches for Qur'ān reciter recognition. Among the evaluated models, X-Vector with MFCC features achieved perfect classification performance (100% accuracy, precision, recall, and F1 Score) on the test set. Both the CNN and BLSTM models achieved near-perfect performance (99.96% accuracy and F1), while the end-to-end X-Vector operating on raw waveforms reached 98.77% accuracy (F1 = 0.9877).

5.2.2. Impact of feature choice and X-Vector embeddings

These findings highlight the continued relevance of structured spectral features such as MFCCs, which effectively capture unique vocal characteristics and subtle stylistic differences among reciters. When combined with the X-Vector architecture, MFCCs yield robust speaker embeddings that are highly discriminative for this task, which explains the perfect scores observed.

5.2.3. CNN vs. BLSTM: complementary strengths

The CNN and BLSTM models confirm that deep networks trained on time–frequency representations (spectrograms or MFCCs) can model complex recitation patterns with high fidelity. The CNN excels at learning local time–frequency structure, while the BLSTM captures bidirectional temporal dependencies. Their near-identical results (99.96%) suggest that, under short fixed segments (2 s) and balanced data, both paradigms can reach the same ceiling.

5.2.4. End-to-end X-Vector on raw waveforms

Operating directly on raw audio simplifies preprocessing by removing the need for hand-crafted feature extraction. However, the end-to-end variant trails the MFCC-based X-Vector by ~1.2 percentage points in accuracy. This gap is consistent with the hypothesis that end-to-end waveform models generally benefit from larger, more diverse datasets or self-supervised pretraining to learn robust front-end representations.

The lower performance of the raw-waveform X-Vector relative to the MFCC-based variant likely reflects several factors: (i) waveform models must learn the spectral envelope and pitch structure from scratch, which is data-hungry compared to MFCCs that already encode a perceptually motivated frequency compression; (ii) the 2-second clips provide limited temporal context, making it harder for shallow convolutional stacks to disentangle fine-grained speaker cues from local phonetic content; and (iii) our architecture is deliberately lightweight to keep training feasible, whereas successful raw-audio systems in the literature often rely on deeper or self-supervised encoders pre-trained on much larger corpora. Under these constraints, MFCCs serve as a strong inductive bias, simplifying the learning problem.

5.2.5. Comparison with prior work

Table 4 summarizes representative prior studies. Broadly, the literature falls into two families: (i) engineered acoustic front ends (e.g., MFCC) paired with classical or deep classifiers, and (ii) end-to-end or transfer-learning approaches that leverage large-scale pretraining (e.g., HuBERT, Wav2Vec2.0, NASNet/EfficientNet, TRILL/VGGish). Within this landscape, our MFCC-based X-Vector attains 100% accuracy on 12 reciters using fixed 2 s segments. Under the configurations listed in Table 4, this matches or exceeds MFCC+deep baselines that operate on fewer classes and/or longer segments, and is competitive with transfer/self-supervised pipelines despite a simpler feature front end and shorter inputs.

We selected 2-second segments as a compromise between capturing sufficient prosodic context and maintaining a large number of training examples. In Qur'ānic recitation, many distinctive cues, such as timbre, articulation style, short-range pitch movements, and pauses, are observable within 1–2 seconds, even when full verses span longer durations. Direct numerical ranking is not strictly comparable across papers due to differences in the number of reciters, segment duration, dataset source/size/balance, and evaluation protocols. The table is therefore intended as a qualitative map of methods and settings rather than a definitive leaderboard.

Table 4. Comparison with representative prior work on Qur'ān reciter recognition. Results are not strictly comparable due to differences in classes, segment lengths, datasets, and protocols

Study	Dataset (source, reciters, size)	feature Extraction	Model	Best result (Acc)
[8]	public, 5 reciters; segment lengths 1–3 s	MFCC	BLSTM (2 layers) + FC	99.89%
[26]	private, 169 reciters (+1 "noise" class)	TRILL, VGGish embeddings; MFCC	CNN head on fixed embeddings; transfer learning, fine-tuning	TRILL ~98%
[11]	public, Subset of AR- DAD: 10 reciters, 1000 clips	Self-supervised embeddings	Add MLP/RNN/CNN heads; end-to-end fine-tuning	HuBERT Large up to 99%
[7]	private, 7 reciters; 80 min per reciter; segmented into 2/3/4 s	MFCC	CNN	Up to 99.66% (CNN);
[10]	public, 20 reciters; ~11,000 20-s segments	MFCC	Transfer learning: NASNetLarge, EfficientNetB7/V2S /V2M/V2L, NASNetMobile + dense head	98.50% (NASNetLarge)
This study	public, 12 reciters; 2 s	MFCC	X-Vector	100.00%
This study	public, 12 reciters; 2 s	Raw waveform	End-to-End X-Vector	98.77%
This study	public, 12 reciters; 2 s	MFCC	BLSTM	99.96%
This study	public, 12 reciters; 2 s	MFCC	CNN	99.96%

5.2.6. Qur'ānLimitations, robustness, and future directions

Our current evaluation focuses on matched conditions: all clips are derived from recordings of *ṣūrah Al-Tawbah*, and the train/validation/test splits share the same channel characteristics. We did not perform explicit robustness experiments (e.g., additive noise, simulated reverberation, codec variation, or cross-*ṣūrah* transfer), so the perfect and near-perfect scores should be interpreted as upper bounds under these constrained conditions.

The near-ceiling results obtained in our experiments should be interpreted in light of the task design. All clips are derived from a single *sūrah* (*ṣūrah* Al-Tawbah), share broadly similar acoustic characteristics, and are segmented into short, fixed-length 2-second windows. Under such controlled conditions, reciter-specific timbral and prosodic cues are relatively stable, which makes the classification problem easier than in fully unconstrained settings (e.g., multiple *sūrahs*, varying microphones, background noise, and broadcasting channels). Consequently, our 100% test accuracy for MFCC-based X-Vectors is best viewed as an upper bound under matched conditions, rather than as evidence that Qur'ānic reciter recognition is solved in the wild. The proposed framework is nevertheless general: it can be applied unchanged to more heterogeneous corpora by (i) extending the dataset to multiple *ṣūrahs* and venues, (ii) training and evaluating under explicit noise and channel perturbations, and (iii) testing cross-*ṣūrah* transfer (train on one set of *ṣūrahs*, test on another). We highlight these extensions and position our current results as a controlled starting point for such robustness studies.

Beyond data and evaluation, we will explore modeling advances: (iv) hybrid front-ends that fuse MFCCs with self-supervised audio embeddings (e.g., HuBERT, Wav2Vec2.0, TRILL2); (v) attentive/statistical pooling and metric-learning losses (Arc- Face/CosFace/ProxyNCA) to improve open-set and few-shot generalization; and (vi) multi-task learning with *maqam* and prosody labels to induce linguistically grounded embeddings. Finally, we will study (vii) fairness and calibration across demographic sub- groups and (viii) efficient deployment via quantization/distillation for real-time, on-device inference.

In addition, we outline three concrete future work strands:

- Real-time recognition and deployment

  We will prototype streaming/causal encoders with chunked processing (0.5–1.0 s windows), target a real-time factor (RTF) < 0.5, and optimize them using pruning, INT8 quantization, and low-rank adapters. We will release REST/gRPC APIs and mobile SDKs, and conduct user studies (accuracy, latency, and SUS usability) in educational settings.

- *Tajwīd* analysis and quality assessment

  Beyond identity, we will develop rule detectors for specific tajw¯ıd phenomena (e.g., *ikhfā', idghām, iqlāb*) with aligned verse-level labels. A hierarchical pipeline is employed, beginning with reciter identification followed by *tajwīd* analysis, to evaluate the recitation process. This structured approach enables the system to produce interpretable feedback and automatically generated quality scores, facilitating a clearer and more objective assessment of the recitation performance. We will report correlation with scholar annotations (Pearson/Spearman), inter-rater reliability, and provide visualization overlays to highlight suggested improvements.

- Dataset expansion and diversification

  We plan to broaden coverage to more reciters (including female voices, age groups, and regions), multiple *ṣūrahs* and recording venues, and realistic acoustic conditions. Metadata (style, *maqām*, tempo, *tajwīd* annotations) will support analysis and multi-task learning. We will publish fixed train/validation/test splits and evaluation scripts, and investigate synthetic data (VC/TTS-based augmentation) to address class imbalance while monitoring realism via MOS and spoof-detection baselines.

6. Conclusion

This study systematically investigated deep learning approaches for Qur'ān reciter recognition, comparing feature-based pipelines with end-to-end models under a unified dataset and evaluation protocol. Beyond assembling a curated corpus, our contribution lies in providing a controlled, reproducible benchmark with fixed short segments and in conducting a head-to-head comparison of representative modeling families that have previously been evaluated on disparate datasets. Empirically, the X-Vector architecture combined with Mel-Frequency Cepstral Coefficients (MFCCs) achieved perfect test performance, underscoring the continued value of structured spectral representations for capturing the subtle acoustic and stylistic cues of recitation. The end-to-end X-Vector operating on raw waveforms delivered slightly lower yet competitive accuracy. This suggests that, while end-to-end learning simplifies preprocessing and enables automatic feature discovery, explicit spectral features still offer an advantage for short segments where fine-grained acoustic distinctions are critical. Taken together, the findings support a pragmatic conclusion: hybrid or feature-informed architectures remain a strong choice for reciter identification at current dataset scales. The contributions of this work are threefold: (i) a controlled comparison of feature- based and end-to-end architectures for reciter recognition; (ii) a curated dataset with rigorous, speaker-disjoint evaluation; and (iii) empirical evidence that MFCC-based X-Vector embeddings provide highly discriminative representations for 2-second segments across multiple reciters. These results extend the literature on speaker and reciter recognition and provide a robust reference point for future studies. Overall, this study demonstrates that combining

principled acoustic features with modern deep neural architectures yields state-of-the-art performance for Qur'ān reciter recognition under matched conditions. The methodology and findings provide a solid foundation for building accurate, scalable, and pedagogically useful systems with applications in digital archiving, educational technology, and computational Islamic studies, and they highlight clear directions for robustness and real-world deployment in future work.

Acknowledgment

The authors gratefully acknowledge Qassim University, represented by the Deanship of Graduate Studies and Scientific Research, on the financial support for this research under the number (QU-1-PG-2-2025-53078) during the academic year 1446 AH / 2024 AD.

Declarations

Author contribution. Ibrahim AlO-mari: conceptualization, methodology, writing, data curation, data analysis, software, and data evaluation. Asma Al-Shargabi: supervision, conceptualization, and writing review

Funding statement. The Research is funded by the Deanship of Graduate Studies and Scientific Research in Qassim University under the number (QU-1-PG-2-2025-53078) during the academic year 1446 AH / 2024 AD

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author. The created dataset is available at https://doi.org/10.5281/zenodo.15399981.

References

[1] H. Tabbal, W. El Falou, and B. Monla, "Analysis and implementation of a 'Quran' verses delimitation system in audio files using speech recognition techniques," in *2006 2nd International Conference on Information & Communication Technologies*, 2006, vol. 2, pp. 2979–2984, doi: 10.1109/ICTTA.2006.1684889.

[2] S. A. E. Mohamed, A. S. Hassanin, and M. T. Ben Othman, "Educational System for the Holy Quran and Its Sciences for Blind and Handicapped People Based on Google Speech API," *J. Softw. Eng. Appl.*, vol. 07, no. 03, pp. 150–161, 2014, doi: 10.4236/jsea.2014.73017.

[3] S. M. Abdou and M. Rashwan, "A Computer Aided Pronunciation Learning system for teaching the holy quran Recitation rules," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, Nov. 2014, pp. 543–550, doi: 10.1109/AICCSA.2014.7073246.

[4] T. Mahboob, M. Khanum, M. Sikandar, H. Khiyal, and R. Bibi, "Speaker Identification Using Gmm With Mfcc In Python," *J. Crit. Rev.*, vol. 7, no. 14, pp. 126–135, Jul. 2020, doi: 10.31838/jcr.07.14.103.

[5] J. A. Pandian, R. Thirunavukarasu, and E. Kotei, "A Novel Convolutional Neural Network Model for Automatic Speaker Identification From Speech Signals," *IEEE Access*, vol. 12, pp. 51381–51394, 2024, doi: 10.1109/ACCESS.2024.3385858.

[6] M. K. Singh, "A text independent speaker identification system using ANN, RNN, and CNN classification technique," *Multimed. Tools Appl.*, vol. 83, no. 16, pp. 48105–48117, Nov. 2023, doi: 10.1007/s11042-023-17573-2.

[7] G. Samara, E. Al-Daoud, N. Swerki, and D. Alzu'bi, "The Recognition of Holy Qur'ān Reciters Using the MFCCs' Technique and Deep Learning," *Adv. Multimed.*, vol. 2023, pp. 1–14, Mar. 2023, doi: 10.1155/2023/2642558.

[8] A. Qayyum, S. Latif, and J. Qadir, "Quran Reciter Identification: A Deep Learning Approach," in *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, Sep. 2018, pp. 492–497, doi: 10.1109/ICCCE.2018.8539336.

[9] G. Karthiha and S. Allwin, "Transfer learning approaches for EfficientNetV2 B0 and ImageNet skin cancer classification in a convolutional neural network," *PeerJ Comput. Sci.*, vol. 11, p. e3362, Dec. 2025, doi: 10.7717/peerj-cs.3362.

[10] H.-A. Saber, A. Younes, M. Osman, and I. Elkabani, "Quran reciter identification using NASNetLarge," *Neural Comput. Appl.*, vol. 36, no. 12, pp. 6559–6573, Apr. 2024, doi: 10.1007/s00521-023-09392-1.

[11] A. Moustafa and S. A. Aly, "Towards an Efficient Voice Identification Using Wav2Vec2.0 and HuBERT Based on the Quran Reciters Dataset," *arXiv*, pp. 1–5, 2021, [Online]. Available at: http://arxiv.org/abs/2111.06331.

[12] M. Fazel-Zarandi and W.-N. Hsu, "Cocktail Hubert: Generalized Self-Supervised Pre-Training for Mixture and Single-Source Speech," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, vol. 2023-June, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10096630.

[13] H. Liu *et al.*, "AudioLDM 2: Learning Holistic Audio Generation With Self-Supervised Pretraining," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 2871–2883, 2024, doi: 10.1109/TASLP.2024.3399607.

[14] R. Jain, A. Barcovschi, M. Y. Yiwere, D. Bigioi, P. Corcoran, and H. Cucu, "A WAV2VEC2-Based Experimental Study on Self-Supervised Learning Methods to Improve Child Speech Recognition," *IEEE Access*, vol. 11, no. April, pp. 46938–46948, 2023, doi: 10.1109/ACCESS.2023.3275106.

[15] S. Hawi, J. Alhozami, R. AlQahtani, D. AlSafran, M. Alqarni, and L. El Sahmarany, "Automatic Parkinson's disease detection based on the combination of long-term acoustic features and Mel frequency cepstral coefficients (MFCC)," *Biomed. Signal Process. Control*, vol. 78, p. 104013, Sep. 2022, doi: 10.1016/j.bspc.2022.104013.

[16] O. Marshall, "The oleaginous voice: Auto-Tune, linear predictive coding, and the security-petroleum complex," *Hist. Technol.*, vol. 40, no. 3, pp. 276–296, Jul. 2024, doi: 10.1080/07341512.2024.2402580.

[17] M. Han, "Artificial intelligence-driven tone recognition of Guzheng: A linear prediction approach," *Demonstr. Math.*, vol. 57, no. 1, p. 98, Nov. 2024, doi: 10.1515/dema-2024-0009.

[18] S. P. Yadav, S. Zaidi, A. Mishra, and V. Yadav, "Survey on Machine Learning in Speech Emotion Recognition and Vision Systems Using a Recurrent Neural Network (RNN)," *Arch. Comput. Methods Eng.*, vol. 29, no. 3, pp. 1753–1770, May 2022, doi: 10.1007/s11831-021-09647-x.

[19] F. Ye and J. Yang, "A Deep Neural Network Model for Speaker Identification," *Appl. Sci.*, vol. 11, no. 8, p. 3603, Apr. 2021, doi: 10.3390/app11083603.

[20] M. Tiwari and D. K. Verma, "Enhanced text-independent speaker recognition using MFCC, Bi-LSTM, and CNN-based noise removal techniques," *Int. J. Speech Technol.*, vol. 27, no. 4, pp. 1013–1026, Dec. 2024, doi: 10.1007/s10772-024-10150-4.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5329–5333, doi: 10.1109/ICASSP.2018.8461375.

[22] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, p. 101027, Mar. 2020, doi: 10.1016/j.csl.2019.101027.

[23] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Interspeech 2018*, Sep. 2018, pp. 1086–1090, doi: 10.21437/Interspeech.2018-1929.

[24] S. S. Alrumiah and A. A. Al-Shargabi, "Intelligent Quran Recitation Recognition and Verification: Research Trends and Open Issues," *Arab. J. Sci. Eng.*, vol. 48, no. 8, pp. 9859–9885, Aug. 2023, doi: 10.1007/s13369-022-07273-8.

[25] D. Omran, S. Fawzi, and A. Kandil, "Automatic Detection of Some Tajweed Rules," in *2023 20th Learning and Technology Conference (L&T)*, Jan. 2023, pp. 157–160, doi: 10.1109/LT58159.2023.10092350.

[26] M. Tall, T. I. Diop, N. Fatou Ngom, E. Hadj, and A. Thiam, "Deep learning for Quranic reciter recognition and audio content identification," in *13th Conference on Research in Computer Science and its Applications (CNRIA)*, 2023, pp. 1–12, [Online]. Available at:

https://www.researchgate.net/publication/390271888_Deep_learning_for_Quranic_reciter_recognition_and_audio_content_identification.

[27] M. Mhamed and J. A. Noja, "World Holy Quran Reciter Recognition based on deep learning," in *Proceedings of the 2025 International Conference on Machine Learning and Neural Networks*, Apr. 2025, pp. 97–102, doi: 10.1145/3747227.3747242.

[28] G. K. Berdibaeva, O. N. Bodin, V. V. Kozlov, D. I. Nefed'ev, K. A. Ozhikenov, and Y. A. Pizhonkov, "Preprocessing voice signals for voice recognition systems," in *2017 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, Jun. 2017, pp. 242–245, doi: 10.1109/EDM.2017.7981748.

[29] B. Lohani, C. K. Gautam, P. K. Kushwaha, and A. Gupta, "Deep Learning Approaches for Enhanced Audio Quality Through Noise Reduction," in *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, May 2024, pp. 447–453, doi: 10.1109/IC3SE62002.2024.10593073.

[30] H. W. Al-Dulaimi, A. Aldhahab, and H. M. Al Abbood, "Speaker Identification System Employing Multiresolution Analysis in Conjunction with CNN," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 5, pp. 350–363, Oct. 2023, doi: 10.22266/ijies2023.1031.30.

[31] M. Zakariah, M. K. Khan, and H. Malik, "Digital multimedia audio forensics: past, present and future," *Multimed. Tools Appl.*, vol. 77, no. 1, pp. 1009–1040, Jan. 2018, doi: 10.1007/s11042-016-4277-2.