Human Interaction and Emerging Technologies (IHIET-FS 2026), Vol. 235, 2026, 59–69
https://doi.org/10.54941/ahfe1007261

Bridging Traditional Islamic Scholarship and Modern AI: A Human-Centered Voice Recognition System for Quran Reciter Identification

Omar I. Alsaleh

Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 8707 Riyadh, Saudi Arabia

ABSTRACT

Identifying Quran reciters from short audio excerpts can support reciter discovery, religious learning, and user engagement, yet practical deployment is challenging because recordings are often captured in noisy everyday environments. This paper presents a human-centered voice recognition system for Quran reciter identification that combines audio preprocessing, feature engineering, neural classification, and mobile deployment into a single user-facing service. The workflow begins with Quran-recitation metadata linked to online audio sources, followed by download, cleaning, duplicate filtering, normalization, fixed-length segmentation, and extraction of Mel-Frequency Cepstral Coefficients (MFCCs) and embedded utterance features. The learning task was formulated as reciter classification from short clips, with experiments comparing 10-, 15-, and 20-second durations as well as artificial neural network (ANN) and recurrent neural network (RNN) models. The results showed that a 15-second clip provided the best balance between usability and recognition capability. The ANN outperformed the RNN and was selected as the final deployed model, achieving 97.8% accuracy on the test set and 99.0% on the training set under controlled conditions. A mobile application and server-side inference service were then implemented to deliver the system to users. Although deployment remained constrained by the mismatch between clean training audio and noisy real-world recordings, the study demonstrates the feasibility of a practical, human-centered AI system for Quran reciter identification.

Keywords: Quran reciter identification, Speaker identification, Acoustic feature extraction, Artificial neural networks

INTRODUCTION

In contemporary digital environments, Quran recitations are widely accessible through broadcast media, streaming platforms, and social channels, yet listeners often encounter recitations by unfamiliar voices and are unable to determine the identity of the reciter. This creates a practical barrier for users who wish to continue listening to a specific reciter, follow a preferred recitation style, or use that discovery as part of religious learning and spiritual engagement. This need is best understood as a human-centered problem rather than a purely technical one. The present work addresses it through a mobile application that captures

Received March 14, 2026; Revised March 29, 2026; Accepted April 17, 2026; Available online June 22, 2026

a short recording, extracts representative acoustic features, and returns the predicted reciter name.

From a technical perspective, the problem belongs to voice-based classification, but it has domain-specific constraints that make direct deployment non-trivial. The system is designed specifically for Quran recitation and does not attempt general speaker identification across arbitrary speech contexts. The operational workflow is further constrained by short recordings, ultimately settling on 15-second clips as a practical compromise between model informativeness and user burden. This design choice is important from a human-centered standpoint: a recording that is too short may not contain enough stable vocal evidence, whereas a longer recording may be inconvenient when the user is listening from radio, television, or another uncontrolled source. The proposed workflow therefore combines short-duration audio capture, acoustic feature extraction based primarily on Mel-Frequency Cepstral Coefficients (MFCCs) and embedded utterance representations, and supervised classification using neural models, followed by integration into a mobile interface that ordinary users can operate with minimal technical knowledge.

The work also addresses an important gap between laboratory performance and deployment reality. While controlled training data can produce strong classification results, real-world usage introduces background noise, device variability, and interruptions in recitation, all of which affect inference quality. These practical conditions influenced both model selection and deployment scope, including the decision to deploy the system for 20 reciters rather than the larger experimental ambition explored during development. Within this context, the paper makes three main contributions. First, it presents a human-centered formulation of Quran reciter identification as an assistive mobile service for reciter discovery and religious learning. Second, it documents an end-to-end machine-learning pipeline covering audio preparation, feature engineering, neural-model comparison, and mobile deployment. Third, it reports a deployment-oriented design in which an artificial neural network (ANN) outperformed a recurrent neural network (RNN) and was selected as the final model, while also discussing the real-world challenges that limited scalability under noisy operating conditions. The remainder of the paper reviews recent related work, describes the system architecture, data preparation, and modeling protocol, presents the mobile application and deployment design, reports the main results, and concludes with a discussion of the study's implications and future directions.

RELATED WORK

Recent work confirms that Quran reciter identification is an emerging but still relatively specialized research area. A recent review of the field shows that prior studies span traditional signal-processing methods, machine-learning classifiers, and deep-learning approaches, while also highlighting persistent challenges related to dataset diversity, recording quality, and evaluation consistency (Alomari et al., 2025). This suggests that the problem is technically feasible, but that performance comparisons across studies

should be interpreted carefully because many systems are developed under different corpus sizes, feature choices, and recording conditions.

Among the most directly relevant recent studies, Saber et al. (2024) addressed Quran reciter identification using MFCC-based visual representations and transfer learning across a dataset of 20 reciters, showing that deep image-based models can capture reciter-specific acoustic patterns effectively. Mhamed and Noja (2025) also explored deep learning for Quran reciter recognition using a chapter-focused dataset and short audio segments, further supporting the viability of learned discriminative representations for this task. Taken together, these studies demonstrate strong benchmark potential for reciter identification, but they are generally oriented toward controlled experimental settings rather than user-facing mobile interaction under everyday listening conditions.

Adjacent research has focused on broader Quran-audio processing tasks rather than reciter identity alone. Alfadhli et al. (2024) presented an end-to-end Quran recitation recognition framework using a hybrid Connectionist Temporal Classification (CTC)/attention architecture and a dataset covering the full Quran recited by multiple reciters. Salameh et al. (2024) introduced a crowdsourced Quranic audio dataset for non-Arabic speakers, highlighting both the importance and the difficulty of building annotated Quran-audio resources at scale. These studies are valuable because they show growing momentum in Quran-related audio AI, yet their primary targets differ from the present work. Recitation recognition, correctness assessment, and dataset construction are closely related problems, but they do not directly solve the user problem of identifying who is reciting from a short audio excerpt.

The broader speaker-identification literature also provides useful insight for this application. Keser and Gezer (2025) showed that speaker-identification performance remains highly dependent on feature design, classifier choice, and noise conditions, even when strong modern models are used. Similarly, Cesarini and Costantini (2024) demonstrated that noise and reverberation remain important real-world factors affecting speech recognition performance and feature robustness. These observations are especially relevant for short user-recorded clips captured in uncontrolled environments and are consistent with recent short-speech speaker-recognition research (Deng et al., 2025). In this context, the present work is positioned at the intersection of Quran-specific reciter identification and practical human-centered deployment. Unlike prior studies that mainly emphasize benchmark performance, this paper focuses on short-duration audio, operational usability, and the transition from laboratory modeling to a deployable mobile service.

SYSTEM REQUIREMENTS AND OVERALL ARCHITECTURE

The proposed system was designed as a user-facing mobile service for identifying Quran reciters from short audio excerpts. Its requirements were defined around a simple operational scenario: a user encounters a recitation, records a short sample, and expects the system to return the most likely reciter with minimal effort and delay. Beyond this primary function, the system also supports session continuity through a history view and future

model improvement through optional user contribution of labeled recordings or source links. From a human-centered perspective, these requirements are important because the target interaction occurs in everyday listening situations rather than in a controlled laboratory setting. The consolidated functional and non-functional requirements are summarized in Table 1.

Table 1: Core functional and non-functional requirements.

Type	Requirement
Functional	The system shall allow the user to record an audio clip for reciter identification.
Functional	The recorded clip shall be processed and submitted to the AI model for inference.
Functional	If the reciter is identified successfully, the system shall display the reciter's name to the user.
Functional	The system shall keep a history of previous identification results.
Functional	The system shall allow the user to contribute an audio clip or a source URL together with the reciter's name.
Functional	The system shall provide an information/about interface for communication and basic application details.
Non-functional	The system should return the result within approximately 30 seconds.
Non-functional	The recording duration should be fixed at 15 seconds.
Non-functional	The application requires an internet connection for backend communication.
Non-functional	The application requires microphone permission for audio capture.

At the functional level, the system must allow the user to record an audio clip, submit it for inference, and view the returned reciter identity. It must also preserve previously identified items so users can revisit them later. In addition, the application provides a contribution pathway through which users may upload either an audio clip or a source URL together with the corresponding reciter label, creating a mechanism for future data growth and model enhancement. A lightweight informational page is also included to support communication and general application information. These requirements show that the system is not only a classifier but also an interactive service that combines inference, traceability of past use, and a limited data contribution workflow.

At the non-functional level, the system is constrained by response time, clip duration, connectivity, and device permissions. The response should be returned within a practical interaction window, the recording length is fixed at 15 seconds, internet access is required because inference is handled through a deployed backend, and microphone access is necessary for real-time capture. These constraints directly shape both usability and technical design. Fixing the recording duration simplifies the user experience and aligns the front-end interaction with the model's training setup, while backend inference

reduces mobile-side computational burden and supports maintainability of the deployed model.

The resulting architecture follows a client-server pattern, as illustrated in Figure 1. The mobile application acts as the front-end interface through which the user records or uploads audio. The audio is then transmitted to a server-side inference service, where preprocessing and model execution are performed before the predicted reciter identity is returned to the application. The application presents the result to the user and stores the interaction in a local history view. In parallel, the contribution function sends user-submitted recordings or URLs with reciter labels to cloud storage for later curation and potential retraining. This architecture separates interaction logic from model-serving logic, which is advantageous for iterative model updates, centralized maintenance, and future scaling to a broader set of reciters.

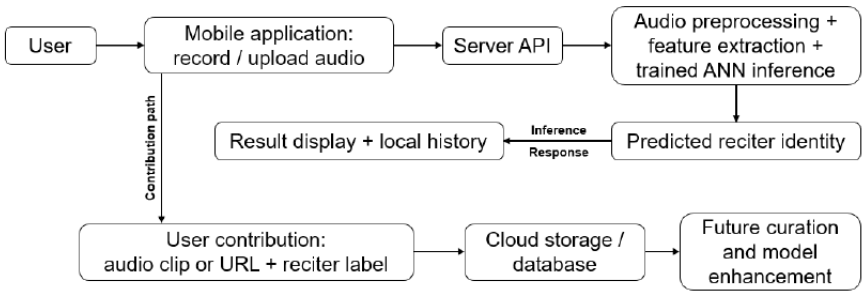

Figure 1: End-to-end system architecture.

DATA PREPARATION AND FEATURE ENGINEERING

The machine-learning pipeline began with a Quran-recitation dataset distributed as metadata rather than ready-to-train audio. The source data comprised three comma-separated value files describing reciters, suras, and recitation records, while the recitation entries themselves contained URLs rather than stored audio files. As a result, the first stage of the workflow was data acquisition: the recitation URLs were programmatically downloaded and organized into an audio corpus suitable for preprocessing and experimentation. The original metadata covered 1117 reciters, but after cleaning and retaining only entries with usable downloadable audio, the working corpus represented 954 reciters. Because the metadata had been aggregated from multiple websites, the raw records also contained duplicate entries referring to the same recitation. To reduce this problem, duplicates were identified at the level of reciter, sura, and recitation style, and only one representative instance was retained when multiple records matched the same combination.

After download, the audio files were standardized before feature extraction. The original recordings were available in compressed formats, so they were converted to WAV to simplify downstream signal processing. During conversion, all files were resampled to 22050 Hz and converted to single-channel audio

in order to maintain a uniform representation across recordings and reduce unnecessary variation in the input pipeline. Additional utilities were used to detect and remove corrupted outputs generated during conversion. Since the recordings varied substantially in duration, they were then segmented into fixed-length clips so that every training instance would produce a consistent feature representation. This step also linked the training configuration to the intended user interaction, since the deployed system was designed around short recordings and ultimately operated with 15-second clips.

Feature engineering focused on representing reciter-specific vocal characteristics without feeding raw waveform data directly into the classifier. Two feature families were used: Mel-Frequency Cepstral Coefficients (MFCCs) and embedded utterance representations. MFCCs were extracted as compact spectral descriptors of the recitation signal, where experiments explored settings up to 30 coefficients, while embedded utterance features were used as higher-level speaker representations produced by a pretrained voice-embedding pipeline. During experimentation, the two feature types were examined separately and then combined. The final feature preparation strategy used data augmentation, extracted both feature sets offline, and concatenated them into a single representation before model training. This approach was preferred because on-the-fly extraction created a severe computational bottleneck and slowed iterative experimentation.

A practical challenge in this stage was computational cost rather than modeling difficulty alone. Reading and processing large volumes of audio for many reciters required substantial preprocessing time, especially when repeated across multiple experiments. To make experimentation feasible, the extraction pipeline was parallelized with multithreading and optimized around faster audio I/O. This substantially reduced end-to-end preprocessing time and allowed repeated model comparisons without recomputing the full pipeline from scratch during every training run. Figure 2 summarizes the data preparation and feature-engineering workflow used in the study.

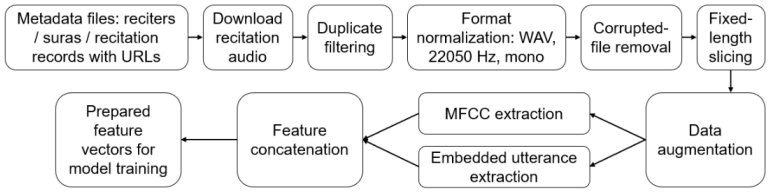

Figure 2: Data preparation and feature-engineering pipeline.

MODEL DEVELOPMENT AND EXPERIMENTAL PROTOCOL

The learning objective was to predict the identity of the reciter from a fixed-length audio representation derived from each recorded clip. Because the final system was intended for real user interaction rather than offline benchmarking alone, model development was organized as an iterative process that balanced recognition performance with deployment realism. The experiments began with a small subset of five reciters to validate the preprocessing and modeling

pipeline, after which the scope was gradually expanded to 10, 20, 30, 50, and 100 reciters. Although the broader development goal was to scale further, the deployed configuration was ultimately restricted to 20 reciters because the gap between clean training audio and user-recorded audio made larger-scale deployment less reliable. This staged protocol helped separate proof-of-concept model behavior from practical operating constraints.

The experimental design compared several key input settings before final model selection. First, three clip durations were examined: 10, 15, and 20 seconds. These settings produced broadly similar recognition capability under clean conditions, but 15 seconds was adopted as the operational choice because it provided a better balance between acoustic sufficiency and user convenience. Second, the feature pipeline evaluated both Mel-Frequency Cepstral Coefficients (MFCCs) and embedded utterance representations. For MFCCs, experiments examined multiple coefficient settings up to 30 coefficients, where a small but consistent gain was observed relative to lower-dimensional alternatives. Embedded utterance features were extracted as fixed-length speaker representations, and separate trials showed that both feature families were effective. The final protocol therefore combined them rather than treating them as competing alternatives. To reduce mismatch between training and real usage, the selected pipeline also incorporated audio augmentation with background noise and a pass-filtering step intended to reduce the quality gap between studio-like source recordings and mobile recordings captured in noisier conditions.

Two neural model families were then compared using the prepared features. The recurrent baseline was an RNN built around two Long Short-Term Memory (LSTM) layers followed by a linear classification layer. The feedforward alternative was a multilayer ANN composed of five layers: input, three hidden layers, and output, with batch normalization applied throughout the hidden stack. In the comparative experiments, the RNN was trained for 10 epochs with a learning rate of 0.001, while the ANN was trained for 6 epochs with the same learning rate. These trials showed that the ANN was more effective for the task and was therefore selected as the final model family for the deployed system.

After model-family selection, the final ANN configuration was refined as the main experimental protocol. The selected network used hidden-layer sizes of 200, 180, and 150 neurons, with batch normalization applied at each layer except the input. Training used the Adam optimizer with 0.9 momentum and a learning rate of 0.002. Because the dataset was imbalanced across reciters, weighted cross-entropy was used during learning, and weighted accuracy was treated as the primary evaluation metric through Matthew's correlation coefficient (MCC) as implemented in Scikit-learn. A confusion matrix was also used during the early pilot stage when the experiments focused on five reciters. This protocol defined the final configuration whose quantitative outcomes are reported in the next section.

MOBILE APPLICATION AND DEPLOYMENT

The trained classifier was delivered through an Android mobile application connected to a server-side inference service. On the user side, the application was designed around a simple interaction model: record a short recitation sample, submit it for analysis, and receive the predicted reciter identity through a clear interface. The main screen provides direct access to recording, while the navigation structure also includes history, contribution, and about pages. The recording duration is fixed at 15 seconds so that user interaction matches the input conditions used during model development. This design keeps the task simple for non-technical users while preserving consistency between training and deployment.

At the implementation level, the mobile client was built for Android and uses standard audio-capture and countdown mechanisms to manage timed recording. Recorded clips are stored locally with timestamp-based naming, allowing users to revisit prior interactions through a history page and replay saved recordings if needed. In addition to inference, the application includes a contribution workflow through which users can submit either an audio clip or a recitation URL together with the associated reciter label. This contribution path was included to support future data collection and iterative improvement of the recognition model.

Model serving was implemented through a lightweight backend based on Flask and hosted on PythonAnywhere. The mobile client communicates with the backend through HTTP requests, sending the recorded audio and receiving the prediction result as the reciter name. User-contributed materials are stored separately in cloud databases for later review and potential retraining. This deployment strategy separates the mobile interface from model execution, reducing computation on the device while enabling centralized maintenance of the classifier. At the same time, the deployed application preserves the practical constraints identified throughout the study: it requires internet access, microphone permission, and operation in environments where recording quality may vary substantially.

RESULTS

The experiments showed that all three tested clip lengths, namely 10, 15, and 20 seconds, were capable of supporting reciter classification under clean laboratory conditions. However, their practical value was not identical. Shorter clips could lose useful vocal evidence when the reciter paused, slowed down, or interrupted the recitation, while longer clips imposed a greater burden on users who might be recording from radio, television, or other uncontrolled sources. For this reason, 15 seconds was adopted as the final operating duration, as it provided a better balance between recognition capability and usability.

The feature-level experiments also produced two important findings. First, increasing the number of MFCC coefficients up to 30 yielded a small but consistent improvement over lower-dimensional alternatives. Second, when MFCC-based inputs were compared with embedded utterance representations under the same neural-network setting, the performance difference was approximately 0.6%, indicating that both feature families were similarly

effective in this task. These comparisons are summarized in Figure 3(a) and Figure 3(b), respectively. Together, these findings supported the later decision to use both feature families rather than rely on only one representation.

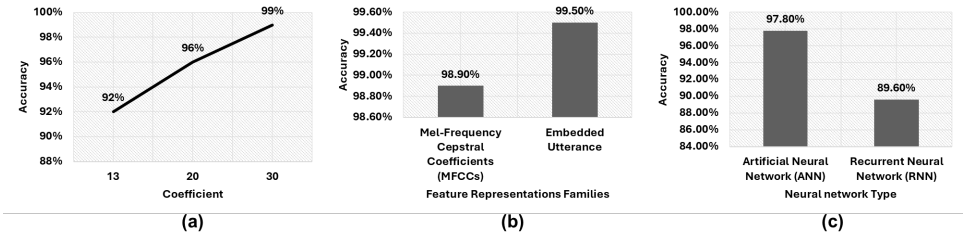

Figure 3: Experimental comparison results: (a) accuracy across different MFCC coefficient settings; (b) comparison between MFCC and embedded utterance representations; and (c) comparison between ANN and RNN models.

At the model level, the comparative experiments showed that the ANN performed better than the RNN and was therefore selected as the final classifier for deployment. This comparison is summarized in Figure 3(c). Using the selected hyperparameters, the final ANN achieved 97.8% accuracy on the test set and 99.0% on the training set. These values indicate that the classification task was highly learnable under controlled data conditions and that the chosen feature pipeline provided strong discriminative information for reciter identity.

The most consequential result, however, was not the laboratory accuracy alone, but the observed gap between laboratory and deployment conditions. The main limitation of the system was that the training audio was largely clean and professionally recorded, whereas real users were expected to record through mobile devices in noisy environments. This mismatch reduced practical performance and forced the deployed system to narrow its scope from larger experimental settings to 20 reciters. Several mitigation attempts were explored, including background-noise augmentation, user-side noise filtering, and pass filtering of the training data to make it closer to real recordings. Although these steps improved the situation to some extent, they did not remove the problem completely. As a result, the final outcome should be interpreted as a strong proof of technical feasibility under controlled conditions, combined with a realistic demonstration of the deployment challenges that arise in user-facing audio AI systems.

DISCUSSION

The results show that Quran reciter identification can be implemented effectively as a user-facing AI service when the problem is framed around short, practical interactions rather than idealized laboratory assumptions. The choice of a 15-second recording window is especially important in this regard. It reflects not only a technical compromise between input sufficiency and model performance, but also a usability compromise that respects the realities of how users encounter recitation audio in everyday settings. In this

sense, the contribution of the work is not limited to classification accuracy alone; it also demonstrates how a specialized audio-recognition task can be shaped into a human-centered mobile interaction for reciter discovery and religious learning.

At the same time, the study highlights a recurring challenge in applied audio machine learning: high laboratory accuracy does not guarantee equally strong field performance. The largest gap emerged from the difference between clean training recordings and noisy user-side recordings, which affected reliability and limited the deployed scope to 20 reciters. This observation is consistent with broader speaker-identification research showing that performance remains sensitive to feature design, classifier choice, and acoustic conditions, particularly under noise and reverberation (Keser and Gezer, 2025; Cesarini and Costantini, 2024). It also supports the view that future progress in Quran reciter identification will depend not only on stronger classifiers, but also on better alignment between training data, deployment conditions, and user behavior. From this perspective, the present work should be understood as both a functional deployed system and a realistic case study in translating specialized AI methods into culturally meaningful human-centered technology.

CONCLUSION

This paper presented a human-centered voice recognition system for Quran reciter identification that combines audio preprocessing, feature engineering, neural classification, mobile application development, and server-side deployment into a single user-facing service. The study showed that short-duration reciter identification is technically feasible, and that a 15-second interaction window provides a practical balance between model informativeness and user convenience. Under the selected experimental setting, the artificial neural network outperformed the recurrent neural network and was adopted as the final deployed classifier.

At the same time, the work demonstrated that real-world deployment remains more challenging than laboratory evaluation. The main limitation arose from the mismatch between clean training recordings and noisy user-recorded audio, which reduced practical reliability and limited the deployed system to 20 reciters despite broader experimental ambitions. Even so, the developed system provides a meaningful proof of concept for applying AI to a specialized cultural and educational domain in a way that supports user needs rather than abstract benchmark performance alone. Future work should focus on expanding the number of supported reciters, improving robustness to noisy and varied recording environments, and incorporating user feedback to verify predictions and support continuous model improvement.

ACKNOWLEDGMENT

The author gratefully acknowledges Abdulaziz Al Huzaymi, Faisal Al Qahtani, Ibrahim Al Mubark, Salem Al Mutairi, and Saud Al Dalbuh for their valuable contributions to the development and implementation of the system described in this paper.

REFERENCES

Alfadhli, S., Alharbi, H. and Cherif, A. (2024) 'qArI: A Hybrid CTC/Attention-Based Model for Quran Recitation Recognition Using Bidirectional LSTMP in an End-to-End Architecture', IEEE Access, 12, pp. 95762–95777.

Alomari, I., Alshargabi, A. and Hadwan, M. (2025) 'Techniques of Quran reciters recognition: a review', IAES International Journal of Artificial Intelligence, 14(3), pp. 1683–1695.

Cesarini, V. and Costantini, G. (2024) 'Reverb and Noise as Real-World Effects in Speech Recognition Models: A Study and a Proposal of a Feature Set', Applied Sciences, 14(23), 11446.

Deng, F., Huang, R., Jiang, P., Yu, L. and Deng, L. (2025) 'Dense-Fusion2Net a more efficient and lightweight short speech speaker recognition system with time-frequency channel attention', Scientific Reports, 15, 9601.

Keser, S. and Gezer, E. (2025) 'Comparative analysis of speaker identification performance using deep learning, machine learning, and novel subspace classifiers with multiple feature extraction techniques', Digital Signal Processing, 156, 104811.

Mhamed, M. and Noja, J.A. (2025) 'World Holy Quran Reciter Recognition based on deep learning', in Proceedings of the 2025 International Conference on Machine Learning and Neural Networks, pp. 97–102.

Saber, H.-A., Younes, A., Osman, M. and Elkabani, I. (2024) 'Quran reciter identification using NASNetLarge', Neural Computing and Applications, 36, pp. 6559–6573.

Salameh, R., Al Mdfaa, M., Askarbekuly, N. and Mazzara, M. (2024) 'Quranic Audio Dataset: Crowdsourced and Labeled Recitation from Non-Arabic Speakers', Procedia Computer Science, 246, pp. 2684–2693.