IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 36, No. 6, December, 2022, pp. 815-823
Journal homepage: http://iieta.org/journals/ria

Accurate Reader Identification for the Arabic Holy Quran Recitations Based on an Enhanced VQ Algorithm

Mohammad A. Al-Jarrah[1*], Ahmad Al-Jarrah[2], Amin Jarrah[1], Mohammad AlShurbaji[1], Sharaf K. Magableh[1], Abdel-Karim Al-Tamimi[1,3], Nisreen Bzoor[1], Mamoun O. Al-Shamali[4]

[1] Computer Engineering Department, Yarmouk University, Irbid 21163, Jordan
[2] Applied Science Department, Ajloun University College, Al-Balqa Applied University, Ajloun 26816, Jordan
[3] Department of Computing, Sheffield Hallam University, Sheffield S11WB, UK
[4] Department of Fundamentals of Religion, Faculty of Shari'a and Islamic Studies, Yarmouk University, Irbid 21163, Jordan

Corresponding Author Email: jarrah@yu.edu.jo

https://doi.org/10.18280/ria.360601

Received: 8 August 2022
Accepted: 20 December 2022

Keywords:
vector quantization, MFCC, speaker identification, LBG, K-means clustering, holy quran, Arabic language

ABSTRACT

The Speaker identification process is not a new trend; however, for the Arabic Holy Quran recitation, there are still quite improvements that can make this process more accurate and reliable. This paper collected the input data from 14 native Arabic reciters, consisting of “Surah Al-Kawthar” speech signals from the Holy Quran. Moreover, this paper discusses the accuracy rates for 8 and 16 features. Indeed, a modified Vector Quantization (VQ) technique will be presented, in addition to realistically matching the centroids of the various codebooks and measuring systems’ effectiveness. Note that the VQ technique will be utilized to generate the codebooks by clustering these features into a finite number of centroids. The proposed system’s software was built and executed using MATLAB®. The proposed system’s total accuracy rate was 97.92% and 98.51% for 8 and 16 centroids codebooks, respectively. However, this study discussed two validation tactics to ensure that the outcomes are reliable and can be reproduced. Hence, the K-mean clustering algorithm has been used to validate the obtained results and discuss the outcomes of this study. Finally, it has been found that the improved VQ method gives a better result than the K-means method.

1. INTRODUCTION

Speech is the most vital interaction tool for human beings of all ages. So, transcribing human speech into words utilizing advances in information technology is challenging because human speech signals vary with different attributes, styles, or environmental noises [1, 2]. The feature of identifying the speaker by utilizing his/her speech for different words with different accents is undoubtedly different from the rules of speech recognition [3]. The difference is that one aims to identify the speaker regardless of the spoken words—the other aims to identify the words regardless of the speaker.

This paper develops an algorithm to identify the reciter of the Holy Quran in Arabic. The Holy Quran was written in Arabic more than 1400 years ago. After that, the Arabic language was developed, and many new accents appeared [4].

Regarding the number of words, Arabic could be considered one of the richest languages. According to the SEBIL center, the Arabic language contains more than 12.3 million words, 20 times more than English [5, 6]. The Arabic language contains 28 characters, where 25 are constant and three are long vowels. These numerous vocabulary counts have made the characteristics of the Arabic language different from any other language. In addition to the richness of Arabic, the Holy Quran recitation has specially formulated rules known as Tajweed [7].

Indeed, Tajweed rules ensure that the recitation is accurate and that each word is pronounced at a moderate speed [8]. Hence, The Holy Quran is considered one of the most important references for the Arabic language. It is also worth mentioning that the study of the Holy Quran is considered a hotspot topic because there are numerous dialects.

In previous studies, researchers focused on extracting the sound features of different recitations of the Holy Quran. They used several methods, i.e., the Mel-Frequency Cepstral Coefficient technique (MFCC). However, they didn’t achieve high accuracy in the extraction process of such systems [9].

Reciting Holy Quran is a duty for every Muslim. Still, this recitation must be free of errors that would conclude readers to wrong interpretations of the verses of the Holy Quran. Thus, Muslims are interested in learning the Holy Quran recitation. This type of learning demands numerous obstacles, for the readers at least. These difficulties include dedicating enough time to learning. First of all, the ultimate goal is to develop an automated Holy Quran recitation system utilizing the latest algorithm in speech recognition. Thus, the target system must analyze the reciter’s voice to decide the reading correctness. Indeed, the reciter’s voice includes features that belong to the reciter’s identity and the read text [10]. Therefore, one of the critical problems is establishing unique speech and language characteristics for each speaker separately.

Moreover, the majority of voice signal processing was limited to the English language. There was scarce work specifically in processing voice signals for the Arabic language and the Holy Quran [11]. Consequently, with all the technological development, these problems must be solved to amplify the number of learners.

This paper proposed a new algorithm for identifying the Holy Quran reciter's identity. This step utilizes extracted features from the reciter's voice signal and then detects the identity of this reciter using vector quantization and k-means clustering. Indeed, MFCC was used to extract the feature vectors (FVs) of the speech. Then, a modified version of Linde–Buzo–Gray (LBG) of VQ clustering was utilized to cluster these features into a set of centroids. Moreover, the K-mean clustering algorithm was also used to cluster feature vectors into groups represented by their centroid. Finally, a similarity measure is used to decide the reciter's identity.

It is worth mentioning that the proposed algorithm of this work should work correctly with all languages, not just Arabic, and all spoken words, not just the words of the Holy Quran. But here, the authors are handling the work with the Holy Quran words, as no significant work is done considering the Arabic language or the Holy Quran's unique tone.

The remaining of this paper is organized as follows: Section II contains the literature review for the previous related research studies, Section III includes the data preparation process, the MFCC features extraction process, the improved LBG methodology that has been used for generating the codebooks, and the suggested process for matching the codebooks, Section IV presents the measured data, the key findings, the accuracy, and the comparative results, Section V discusses the obtained results and sum up the conclusions.

2. RELATED WORK

Previous research studies investigated speakers' recognition systems using various strategies. However, most of these research studies either did not emphasize their results on the Arabic dialect or did not formally compare to other techniques' results. Indeed, methods such as VQ and k-means clustering algorithms for English speakers were presented previously in such systems [10, 11].

Qayyum et al. [12] suggested the use of Bidirectional Long Short-Term Memory (BLSTM), which is considered a type of Recurrent Neural Network (RNN). This method was suitable for speech modeling and processing the job of Quranic speaker identification. Their model was divided into three phases: audio preprocessing, extracting features using the MFCC technique and pushing them into the BLSTM model to identify the reciter correctly. Shi et al. [13] focused on recognizing drones by noticing the noise that fans were producing. They utilized the MFCC approach to extract the sounds' features and the Hidden Markov Model (HMM) in the classification process. Two schemes were applied in the features vectors (FVs) of the MFCC process. Accordingly, the classifier was trained based on the HMM, aiming to validate the impact of different noise types in each cluster on the performance of the recognition rate. The obtained experimental results confirm the effectiveness of the suggested system, even in noisy circumstances. AlKhatib and Eddin [14] investigated the accuracy of speaker voice identification in security systems. They used a digitalized system to fastly recognize and identify the extracted features to process the speech signals. Their system was based on processing the recorded speech signals, extracting the MFCC features, and then matching them with the saved codebook. Thus, they found that the more use of the system, the faster and more accurate the recognition process.

Debnath and Roy [10] presented a clustering topology in automatic speech recognition (ASR) to recognize the spoken digits by speakers. They intended to allow computers to identify the English words that any humans had spoken. They used MFCC to extract the features, clustering these features using the K-mean and Gaussian expectation maximization algorithms. Finally, the hard threshold strategy was performed to measure the effectiveness of the proposed system.

Singh and Joshi [15] presented an identification speaker system that assists in recognizing whisper and natural speech signals. They measured the accuracy of the identification process using two algorithms, i.e., MFCC and Exponential-FCC. They found that MFCC is much better for feature extraction, and GMM gives higher classification accuracy than K-mean clustering.

Devi et al. [16] proposed a hybrid technique for ASR based on an artificial neural network (ANN). They focused on improving the accuracy of predicting speakers. Firstly, they extracted the features using MFCC, then reduced their dimensions using an organizing feature map and enrolled them in multilayer perception. Finally, they trained a dataset for ten speakers and verified them using ANN. The accuracy of the obtained results was 93.33%, with a higher recognition rate.

Deng et al. [17] focused on improving the heart sound classification methodology. They modified the conventional methods used before and were ineffective in heart sound recognition. The improved MFCC was applied to extend the dynamic identifications for the sequential heart sound signals. They found that an accuracy of 98% might be achieved in two-class classification problems, i.e., pathological and non-pathological. Hourri and Kharroubi [18] developed a new orientation to use Deep Neural Network (DNN) learning approach in speech recognition. The extracted features using MFCC were transferred into promoting FVs. They concluded that pushing DNN to train feature noises for the speaker's feature dataset makes it more robust. Wibowo and Darmawan [19] focused on developing the learning methods for the Iqra intonation system. They used MFCC methods for voice feature merits and dynamic time warping to match the obtained features. Their best accuracy in the Iqra reading system was around 82%.

Section III presents the proposed method to determine the Holy Quran reciter's identity. The detailed approach of the proposed improved LBG algorithm of generating VQ codebooks and the detailed codebook matching process are discussed in detail.

3. HOLY QURAN RECITER RECOGNITION

Previous research has used several strategies to recognize speakers' voices, such as those in the studies [20-22]. For instance, the familiarity and voice recognition for acoustic-based representation to find the voice averages [20]. Moreover, the voice principle with non-native language speakers has been investigated [21]. Moreover, an enhanced approach to children's voices is utilized by Arunachalam [22]. This is to recognize the speech with hearing impairment by multi-sets of features and models.

However, as discussed before, the Holy Quran reciters follow stringent rules on top of the complexity of the Arabic language. Therefore, in this section, the process of the proposed method to recognize the Holy Quran reciter is presented. Figure 1 shows a flowchart of the proposed Holy Quran reciter identification system, including four stages, which are: the data preparation stage, MFCC feature extraction stage, codebook matching stage, and codebook matching stage. Each one of these four stages will be discussed in the following sections:

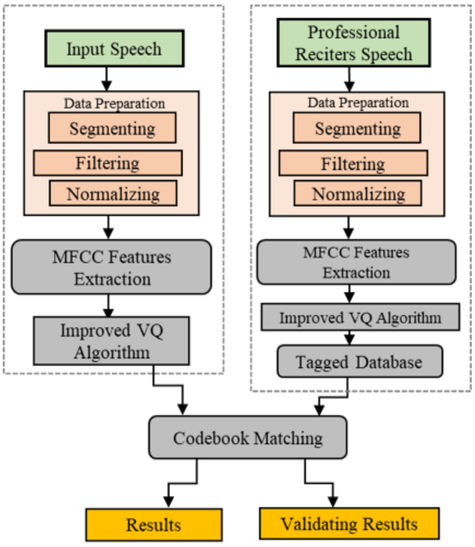

Figure 1. Flowchart of the proposed Holy Quran reciter recognition system

3.1 Data preparation and speech signals segmentation

Input speech signals usually have noise and zones of unwanted silence. Hence, in this paper, the method introduced by Giannakopoulos [23] is implemented using MATLAB®. In this paper, the proposed system's software was built and executed using MATLAB®. Hourri and Kharroubi algorithm removes the defect areas and produces convenient voice segments. Moreover, it comprises the signal energy, the spectral centroid methods, and the simple threshold criterion. Thus, by applying it, the Holy Quran verses, words, or even the letters' sounds could be easily separated and filtered probably. The filtration process can be done by changing the length of the non-overlapping short-term windows (frames) and the step's length. Furthermore, a min-max normalization method defined in Eq. (1) has been applied to the speech signal to ensure that the signal amplitudes of the different reciters are in the same range.

$$Signal_{norm} = \frac{S - S_{min}}{S_{max} - S_{min}} \qquad (1)$$

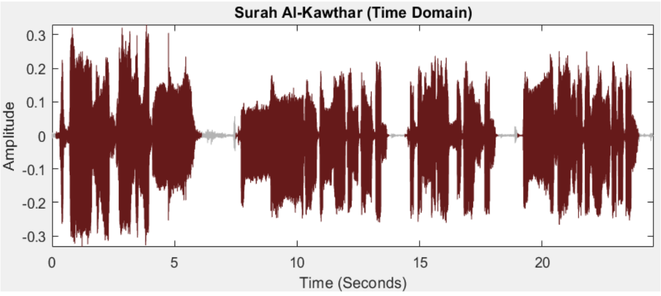

Figure 2. Surah Al-Kawthar's signal

where, S is the input signal, S_{min} is the minimum amplitude of all the signals in the database, and S_{max} is the maximum amplitude. Figure 2 shows the recorded signal for "سورة الكوثر / *Surah Al-Kawthar*" from the Holy Quran by a professional reciter after normalizing its amplitude.

It can be seen that the verses have been separated correctly, and all noises have been filtered. In this figure, the brown signals are the speech signal, and the gray ones are the areas of silence between verses.

3.2 MFCC features extraction

Feature extraction is a process in which speech signals are analyzed. The main features of the voice get extracted to be used in an automatic speech recognition system [8]. The MFCC is widely used for extracting voice features, and it is comprised of coefficient vectors related to each frame of voice [24]. Ahmad et al. [25] mentioned that the MFCC approach is the best method for analyzing Holy Quran verses, and it gives the highest accuracy. The process of implementing MFCC features can be briefly described in the block diagram shown in Figure 3 [8].

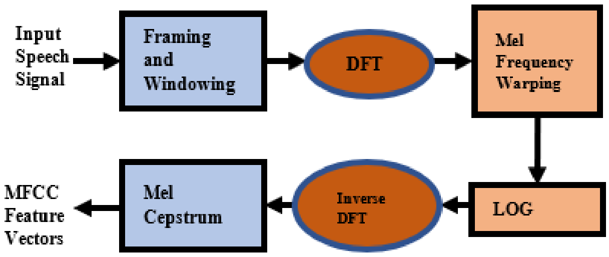

Figure 3. MFCC voice signal features extraction process

In addition, the formula that has been used to compute the MFCC features is written in Eq. (2) [24].

$$Mel(f) = 2595 \times log_{10}(1 + \frac{f}{700}) \qquad (2)$$

Several Feature Vectors represent the MFCC feature coefficients (FVs); each has the same number of coefficients but with different ranges. In addition, a log-energy FV (the zeroth coefficient) is often excluded because it carries only a tiny amount of information about the speaker [26, 27].

This paper utilizes a set of 12 MFCC FV coefficients, excluding the log-energy coefficients. The characteristics of the implemented MFCC function include the windowing length, which describes the length of each word segment or frame, which in this case is equal to 23 mS for each frame. This window length value was found suitable, as suggested in [2]. Furthermore, these frames are overlapped to ensure no information is lost at the end of each frame. Therefore, the overlap length equals 11.5 mS, and every frame's data exists in its previous and next frames' data. These two characteristics have an inverse relationship with the number of MFCC coefficients extracted for each speech signal. Increasing the time of the window length will decrease the number of MFCC coefficients and vice versa.

3.3 MFCC features clustering

Clustering is realized as finding homogeneous groups of data points in a dataset [28]. Each set of these groups is called a cluster, and each cluster has a center point named a centroid. Thereby, feature clustering algorithms can be used effectively for feature-matching techniques. Many clustering algorithms have been used recently for speaker recognition, including Vector Quantization (VQ), K-means Clustering, Support Vector Machine (SVM) [29], Neural Network (NN) [30, 31], Hidden Markov Modeling (HMM), and Dynamic Time Warping (DTW) [32]. In this paper, the VQ and K-means tactics have been used to identify the clusters. For feature clustering, this work proposes an improved VQ clustering algorithm based on the LBG-VQ technique in addition to utilizing k-means clustering for accuracy validation and comparison purposes.

3.4 Vector Quantization (VQ) algorithm

VQ is a data compression technique in which the data points are squeezed into a smaller dataset. Data compression is achieved by remapping these points to a finite even number of clusters represented by their centroids [29]. A code vector is defined as the centroid value of the cluster obtained by VQ. Each code vector contains the key points that represent all data points. The collection of all code vectors in a dataset is called the codebook [33]. A simplified two-dimensional VQ diagram shown in Figure 4 illustrates the conceptual Vector quantization. This figure shows the datapoints features of a Holy Quran reciter (a data point refers to one MFCC coefficient in two-dimension FVs). The clusters' data points are represented as colored dots, and their centroids are represented as crosses with the same color. The VQ technique refers to the centroids of the clusters as a code vector. For this reason, the VQ is considered a data compression technique [32].

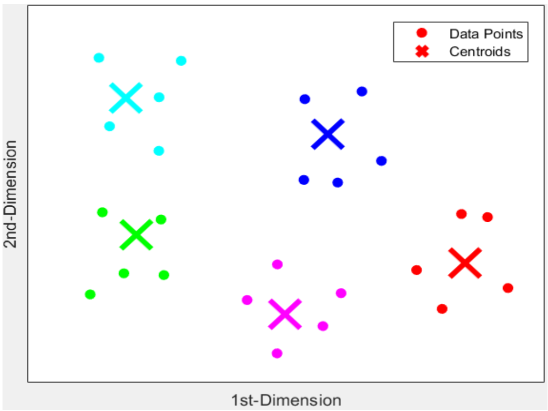

Figure 4. Simple VQ diagram

Clustering technique performance could be measured using clustering error criteria, the distance of every data point in a cluster from its centroid, which should be as lowest as possible [31]. Several algorithms were used to compute the clustering error criterion. However, this paper utilizes the squared Euclidean distance (SED), as described in equation (3). To decide on a point to join a cluster, the SED should always be the lowest to join this point to any other cluster. The SED for two vectors, V1 and V2, is described in formula (3) where N is the dimension of V1 and V2.

$$SED(V_1, V_2) = \sqrt[2]{\sum_{i=0}^{N-1} (V_{1i} - V_{2i})^2} \tag{3}$$

3.5 Proposed clustering algorithm based on LBG-VQ

After the FVs have been computed from a reciter's segmented input speech using MFCC, the codebook is created based on clustering the FVs, called a reciter-specific VQ codebook. The proposed clustering algorithm developed based on the BG-VQ algorithm clusters a set of L FVs to M (8 or 16) codebook vectors, where M is less than L [32, 33]. The proposed algorithm and the implementation are formally described in the following recursive procedures:

(1) The input is a set of L FVs. Each FV has 12 elements, as described in the MFCC feature extraction section. Initially, all FVs are grouped in one cluster.

(2) The centroid of the initial cluster, the mean, is calculated as described in Eq. (4). Then this centroid is stored in the M1 codebook. The dimension of the obtained centroid vector is 12.

$$M_1(L) = \frac{\sum_{i=1}^{k} FV_i\,(L)}{k} \tag{4}$$

(3) Each centroid, which represents one cluster of data, is utilized to generate two new centroids by adding and subtracting a minimal value, namely Epsilon (ε). The value of ε depends on the elements' values in each FV and is computed as proposed in Eq. (5).

$$\mathrm{\varepsilon(L)} = \big(\mathrm{max(L)} - \mathrm{min(L)}\big) * \mathrm{W} \tag{5}$$

where, W is a weight factor. This study found that the best value for W equals 15%. Accordingly, M_1 and M_2 can be computed in Eq. (6).

$$M_{1,2}(L) = \; M_{1,2}(L) \pm \varepsilon \tag{6}$$

(4) Afterward, the original cluster is divided into two clusters. Each FV in the original cluster is added to a cluster such that this FV is closer to the centroid of the joined cluster. To achieve this, D_1 and D_2 are computed according to (7).

$$\mathrm{D}_{1,2} = \sqrt[2]{\sum_{\mathrm{i=1}}^{\mathrm{k}} (\mathrm{FV_i\,(L)} - \mathrm{M}_{1,2}(\mathrm{L}))^2} \tag{7}$$

(5) After clustering all the FVs, the contoids M_1 and M_2 for the two clusters are recomputed. If the new centroids are not equal to the old ones, then steps 3 and 4 are repeated.

(6) Hence, two clusters with two centroids will be formed. Likewise, the same steps are repeated to have four centroids, eight centroids, and 16 centroids.

(7) Finally, a codebook with size N centroids can be created. Figure 5 illustrates the detailed flow chart steps of this improved LBG algorithm [32].

3.6 Codebooks matching

Codebook matching is defined as the process of searching in the reciters' database to identify the current reciter. First, one retrieves a codebook from the reciters' database and matches it with generated codebook. The generated codebook for the Holy Quran reciter is compared with all codebooks stored in the reciters' database. Each centroid in the generated codebook is coupled with its nearest centroid in the retrieved codebook. These two codebooks would be from different verses of similar or different reciters. The purpose of this step is to measure the matching error between the two codebooks based on the total SED between the coupled centroids. So, if the total SED is less than the threshold criteria, the reciter of the retrieved codebooks would be considered a candidate reciter for the generated one and added to a reciters' candidate list. After comparing the generated codebook with all codebooks in the database, the reciter with the lowest matching error in the reciters' candidate list is considered the reciter for the input signal. If the reciters' candidate is empty, then the reciter of the input signal is considered unknown.

For matching reciter-generated codebook (RGCB) with database codebooks, the following procedure has been performed:

(1) Read the codebook from the codebook database (CDB).

(2) Compute the SED between the centroids of RGCB and CDB and build a comparison matrix as shown in Table 1.

(3) Search the CED to find the smallest value, couple the row centroid with the column centroid, and add them to the centroid coupling array as shown in Table 2. Then, delete the selected row and the selected column from the comparison matrix.

(4) Repeat step (3) until the comparison matrix is empty.

(5) Repeat step (2) until the N centroids of the 1st codebook match with the N centroids of the 2nd codebook.

(6) Further steps are added to ensure that no two or more centroids of the 1st codebook match with the same centroid of the 2nd codebook.

(7) Calculate the total SEDs, the sum of all the SED in the coupling array.

(8) If the total SEDs are less than a threshold, add this CDB to the candidate reciters array.

(9) Repeat all these steps for each codebook in the codebook database.

(10) If the candidates' reciter array is not empty, the recognized reciter is the reciter with the least total SEDs; otherwise, the reciter is not identified as one in the codebook database.

(11) Calculate the total SED, the sum of all the SED in the coupling array.

Table 1. Calculated SED between RGCB and CDB

CDB RGCB	Centroid 1	Centroid 2	…	Centroid N
centroid 1	2.449507	1.703520	…	9.354522
centroid 2	1.63247	3.033757	…	11.49068
centroid 3	5.105006	4.8790418	…	6.78788
⋮	⋮	⋮	⋮	⋮
centroid n	12.4642	11.3808	…	3.015966

Table 2. Coupled RGCB and CDB centroids and their SED

RGCB	CDB	SED
Centroid 4	Centroid 6	1.06214
Centroid 6	Centroid 5	1.328485
Centroid 7	Centroid 7	1.630454
⋮	⋮	⋮
Total minimum SEDs		14.1194

For validating purposes and ensuring that the matching procedure is accurate and reliable, The total SED results from matching the 1st codebook and the 2nd codebook of the same verse of different reciters must have the same value when doing this interchangeably. Additionally, to evaluate the proposed algorithm, the recognition rate is defined as the percent of the correctly recognized reciter with respect to the total conducted recognitions. The recognition accuracy rate (RAR) is formulated as in formula (8), which is simply as the known error rate formula [34].

$$RAR = \frac{No.\ of\ correct\ recogntion}{Total\ No.\ of\ recognition\ experments} * 100\% \qquad (8)$$

Furthermore, the proposed algorithm performance evaluation metrics (precision and recall) were computed in Eqns. (9) to (10) [35].

$$Precision = \frac{TP}{TP + FP} \qquad (9)$$

$$Recall = \frac{TP}{TP + FN} \qquad (10)$$

where, "TP, TN, FP, and FN are the True Positive, True Negative, False Positive, and False Negative values, respectively.

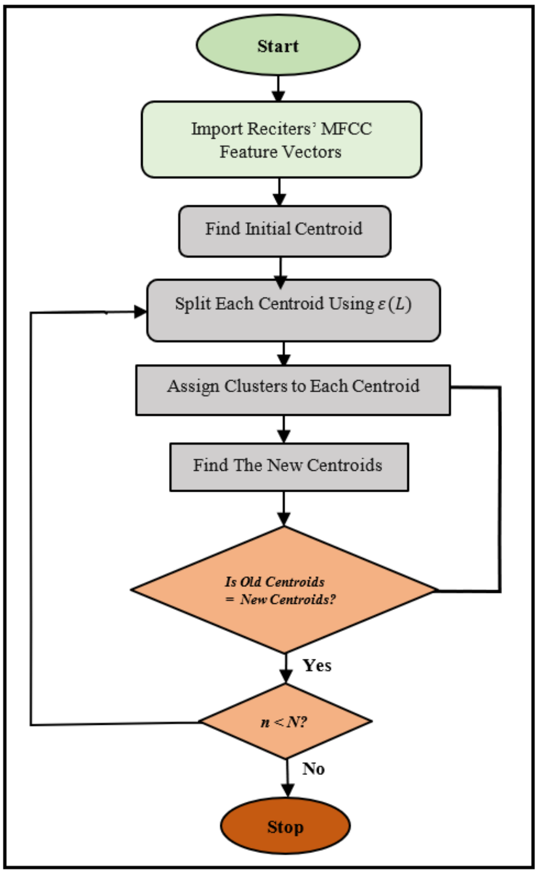
Start
Import Reciters' MFCC Feature Vectors
Find Initial Centroid
Split Each Centroid Using $\varepsilon(L)$
Assign Clusters to Each Centroid
Find The New Centroids
Is Old Centroids = New Centroids?
Yes
$n < N$?
No
Stop

Figure 5. Flowchart for the proposed clustering algorithm

4. RESULTS AND DISCUSSION

This section will discuss the proposed algorithm results step by step. Indeed, this paper aims to investigate the improved VQ technique in the Quranic reciter's voice recognition. Therefore, the proposed system's results are presented in the following subsection.

4.1 MFCC features extraction and clustering

Feature vectors (FVs) have been extracted for each reciter for the Holy Quran's verses using the MFCC method to build a database. The built database contains 130095 FVs with 255 coefficients for each reciter extracted from 130095 speech signal frames. A sample of MFCC FVs is shown in Table 3. Table 3 shows the high dimensionality of the generated space with a high coefficient range.

Table 3. Sample for the first six coefficients MFCC FVs

FVs Coeff.	1	2	3	4	5	…
1	-17.49	4.251	0.216	1.413	1.263	…
2	-15.77	4.496	0.029	2.064	1.359	…
3	-16.84	4.721	-0.182	2.466	1.547	…
4	-17.73	4.896	0.172	2.622	0.965	…
5	-17.51	4.096	1.268	1.463	0.459	…
6	-17.27	2.815	0.849	1.044	-0.107	…
⋮	⋮	⋮	⋮	⋮	⋮	

Table 4. Results of clustering 255 MFCC coefficients

2^n divided Clusters	1	2	2^2	2^3
Datapoints [double]	255 x 12	213 x 12 42 x 12	85 x 12 124 x 12 24 x 12 22 x 12	53 x 12 42 x 12 48 x 12 21 x 12 28 x 12 42 x 12 18 x 12 21 x 12

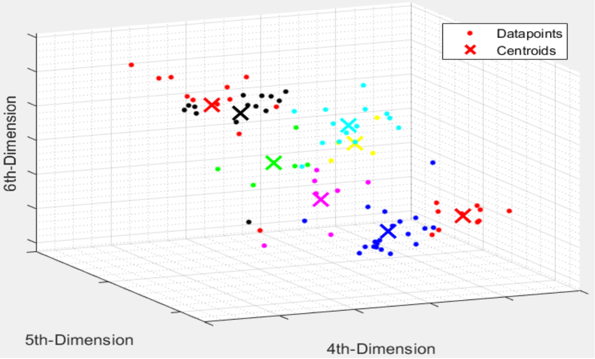

Figure 6. A 3D visualization for a sample FVs clusters and their centroids

The next step is to cluster the obtained MFCC coefficients into an N cluster. The proposed improved VQ algorithm divides the MFCC coefficient firstly into two clusters. Then, it divides it into 4, 8, and 16. Table 4 shows the clustering results of clustering 255 FVs, which are the MFCC for the verse " بسم الله الرحمن الرحيم / Bism Allah Ar-Raḥman Ar-Raḥim" for the professional reciter "Ahmed Amer." Column two of Table 4 shows that the proposed algorithm divides FVs into two clusters where the first one contains 231 vectors and the second one contains 42 vectors.

The centroids for the 8 and 16 clusters have been computed and considered a codebook for the reciter. Figure 6 visualize a codebook of the first verse of "بسم الله الرحمن الرحيم / Bism Allah Ar-Raḥman Ar-Raḥim." This codebook includes eight centroids (i.e., "cross" sign) with their associated clusters (i.e., "dot" sign.) using the proposed improved VQ method. The same color represents each cluster and its centroid. Figure 6 is a 3D graph for the first three coefficients of the FVs. As aforementioned, the number of dimensions depends on the number of FVs, which is 12.

4.2 Codebooks matching process

This section elaborates on how two codebooks match while comparing Holy Quran verses. For instance, suppose there are two codebooks for the same reciter, with the same number of centroids (i.e., eight centroids), but for two different verses. Figure 7 depicts a 3D graph for codebooks of the same reciter with different verses. The 1st codebook is for a part of the verse "بسم الله الرحمن الرحيم / Bism Allah Ar-Raḥman Ar-Raḥim," and the 2nd codebook is for a part of the verse "إنّا اعطيناك الكوثر / Inna Atainaka Al-Kausar." In the first verse, centroids are blue, while in the second verse, centroids are red.

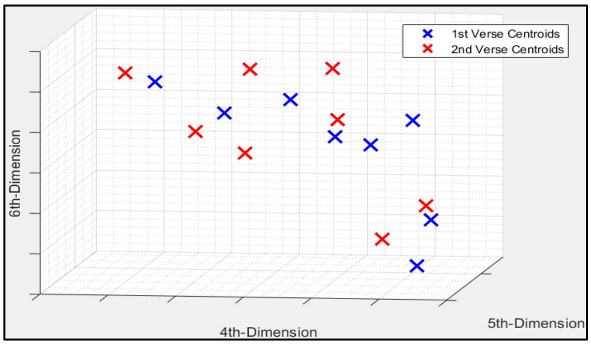

Figure 7. Two codebooks' centroids comparison

The proposed codebook matching process matches the 1st codebook's centroids with their nearest centroids in the 2nd codebook, depending on the SED. For instance, Table 5 shows the total SED between the codebooks of 5 professional reciters for the same verse. The total diagonal SEDs are zeroes because obviously, the centroids are in the exact match (11i.e., the same reciter with the same verse), such that, Xij = zero when i is equal to j.

Table 5. Total SED values of the matching process

Reciter	#1	#2	#3	#4	#5
#1	0	22.446	28.610	27.496	40.453
#2	22.446	0	17.530	24.746	35.952
#3	28.610	17.530	0	20.959	27.124
#4	27.496	24.746	20.959	0	28.289
#5	40.453	35.952	27.124	28.289	0

Additionally, from Table 5, it can be noticed that the total SED between two codebooks of different reciters is precisely the same value when altering the order of these two codebooks. In other words, the values are symmetrical around the diagonal (i.e., Xij is equal to Xji). Thereby, this provides concrete evidence that the codebook matching process is accurate.

5. RECITER IDENTIFICATION PROCESS AND EXPERIMENTAL RESULTS

The reciter that needs to be identified should have the minimum total SED value among all the other values. In other words, the matching error criterion for the same reciter based on different verses is expected to obtain the lowest value. Therefore, the above threshold value has been selected to achieve the best reciter identification accuracy. Table 6 illustrates the total SED values for five reciters. Each row shows the comparison between reciters reciting the second verse for "سورة الكوثر / Surah Al-Kawthar" with reciters reciting the first verse.

Additionally, this table shows sample results of whether a reciter has been identified. The sample result gives a recognition rate of 100% by setting a threshold value of less than 20, as illustrated in Table 6. Obviously, a zero SED value will be obtained if the same verse is used for comparison.

To evaluate the proposed reciter identification system, a codebook database has been developed for fifteen professional Holy Quran reciters where five of them are well-known Holy Quran reciters. The codebook (feature vectors) is computed for each verse and stored in the codebook's database. In the first phase, the recitations for the fifteen reciters were recorded, where each one of them recited the four verses of "Surah Al-Kawthar." In the second phase, some reciters are selected to be unknown. Then, they were asked to recite the first verse.

Table 6. Total SEDs comparison for reciter utilizing first and second verse of Surah Al-Kawthar

1st verse 2nd verse	**#1**	**#2**	**#3**	**#4**	**#5**
#1	**11.346 (Match)**	25.520 (No)	29.745 (No)	22.152 (No)	38.593 (No)
#2	29.121 (No)	**18.507 (Match)**	31.932 (No)	29.055 (No)	30.465 (No)
#3	37.518 (No)	36.879 (No)	**14.966 (Match)**	34.490 (No)	52.711 (No)
#4	27.470 (No)	33.443 (No)	32.659 (No)	**17.255 (Match)**	31.486 (No)
#5	33.496 (No)	32.619 (No)	38.739 (No)	25.113 (No)	**19.644 (Match)**

The proposed reciter identification system is utilized to identify the unknown reciters, and the recognition accuracy rate (RAR) is computed. Table 7 shows RAR for identifying the reciter where each row shows the result when utilizing one verse as input. The overall reciter identification accuracy equals 98.21%.

Table 7. The overall accuracy rate of the system

Unknown reciter verse	**Recognition Accuracy Rate %**	
	Utilizing 16 centroids	**Utilizing 8 centroids**
# 1	98.81%	85.71%
# 2	100%	100%
# 3	97.62%	83.33%
# 4	96.73%	71.43%
Average	98.21%	85.11%

For further performance study, precision and recall are also computed in the proposed system. This study presented the accuracy of the proposed algorithm utilizing 8 to 16 centroids for the codebook. Table 8 depicts the precision and recall for the prosed system utilizing 8 and 16 centroids codebooks. They have been computed based on each verse in Surah Al-Kawthar as input. Verse #2 shows the best precision and recall for identifying the reciter. The overall precision equals 95% and recall equals 96%.

Table 8. Precision and recall for the proposed algorithm utilizing 8 and 16 centroids

Utilized verse	**8 centroids**		**16 centroids**	
	Precesion	**recall**	**Precesion**	**Recall**
#1	85.71%	100%	100%	83.33%
#2	100%	100%	100%	100%
#3	83.33%	83.33%	100%	66.67%
#4	71.43%	83.33%	75%	100%
Average	91.13%	97.47%	95.04%	96.63%

Furthermore, a comparison study has been conducted with recent related research. Table 9 presents the comparison with the previous studies' total accuracy results. It can be noticed that the proposed system achieved the highest accuracy percent with 98.21% using the proposed system based on an improved VQ algorithm utilizing 16 centroids for the codebook (feature vector).

Table 9. Total systems' accuracy results compared with the literature

Reference papers	Accuracy %
This paper	98.21%
Debnath et al. [12]	97.25%
Singh et al. [16]	98%
Devi et al. [17]	93.33%
Deng et al. [18]	95.95%
Wibowo et al. [20]	82%

6. CONCLUSION AND FUTURE SCOPE

Arabic speech, especially the Holy Quran recitation, is considered a challenging speaker identification process. This concern was amplified due to the fact that scarce studies discussed the process of voice signals for the Arabic language and, more specifically, reading the Holy Quran. Actually, most systems mainly centralized the English speech signals. The proposed LBG-VQ algorithm should work perfectly with all languages, including Arabic, and all spoken words, not just the words of the Holy Quran. This work aims to increase the research intensity that considers the Holy Quran and to prove that the speaker identification systems recognize the Arabic speaker as well as they do with the English speaker. This research focuses principally on speech signals of the Holy Quran by using the speech signals of 14 professional reciters. The four verses of Surah "Al-Kawthar" were the prime Surah recited by the reciters to be utilized in this system. This study proposed the Holy Quran reciter identifier based on an improved LBG-VQ algorithm, a method for matching the centroids of the codebooks. The proposed improved LBG-VQ algorithm will keep iterating until the centroids of the codebooks reach their best optimal values and compare the findings of recent related published research. The ARs results show that the AR varies between 92.86% and 100%, with an average of 96.43%. Finally, a comparison proved the superiority of the proposed algorithm when utilizing 16 centroids.

ACKNOWLEDGMENT

The authors would like to acknowledge that this research is funded by Scientific Research Support Fund (SRSF), Ministry of Higher Education in Jordan. Furthermore, the authors acknowledge the support they received from the Faculty of Shari'a and Islamic Studies at Yarmouk University.

REFERENCES

[1] Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, J., Penn, G., Yu, D. (2014). Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(10): 1533-1545. https://doi.org/10.1109/TASLP.2014.2339736

[2] Abdel-Hamid, O. (2014). Automatic speech recognition using deep neural networks: New possibilities. Corpus ID: 65172660.

[3] Juan, S.S., Besacier, L., Tan, T.P. (2012). Analysis of Malay speech recognition for different speaker origins. 2012 International Conference on Asian Language Processing, pp. 229-232. https://doi.org/10.1109/IALP.2012.23

[4] Alrabiah, M., Alhelewh, N., Al-Salman, A., Atwell, E. (2014). An empirical study on the Holy Quran based on a large classical Arabic corpus. International Journal of Computational Linguistics (IJCL), 5(1): 1-13.

[5] https://lonet.academy/blog/learn-the-million-word-arabic-language/

[6] El-Khair, I.A. (2016). 1.5 billion words Arabic corpus. arXiv preprint arXiv:1611.04033. https://doi.org/10.48550/arXiv.1611.04033

[7] Nahar, K.M.O., Al-Khatib, R.M., Al-Shannaq, M.A., Barhoush, M.M. (2020). An efficient holy Quran recitation recognizer based on SVM learning model. Jordanian Journal of Computers and Information Technology (JJCIT), 6(4): 392-414. https://doi.org/10.5455/jjcit.71-1593380662

[8] Ahsiah, I., Noor, N.M., Idris, M.Y.I. (2013). Tajweed checking system to support recitation. 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 189-193. https://doi.org/10.1109/ICACSIS.2013.6761574

[9] Bezoui, M., Elmoutaouakkil, A., Beni-hssane, A. (2016). Feature extraction of some Quranic recitation using Mel-frequency cepstral coeficients (MFCC). 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), pp. 127-131. https://doi.org/10.1109/ICMCS.2016.7905619

[10] Debnath, S., Roy, P. (2020). Automatic speech recognition based on clustering technique. Advances in Intelligent Systems and Computing, pp. 679-688. https://doi.org/10.1007/978-981-13-7403-6_59

[11] Devika, A.K., Sumithra, M.G., Deepika, A.K. (2014). A fuzzy-GMM classifier for multilingual speaker identification. 2014 International Conference on Communication and Signal Processing, pp. 1514-1518. https://doi.org/10.1109/ICCSP.2014.6950102

[12] Qayyum, A., Latif, S., Qadir, J. (2018). Quran reciter identification: A deep learning approach. 2018 7th International Conference on Computer and Communication Engineering (ICCCE), pp. 492-497. https://doi.org/10.1109/ICCCE.2018.8539336

[13] Shi, L., Ahmad, I., He, Y., Chang, K. (2018). Hidden Markov model-based drone sound recognition using MFCC technique in practical noisy environments. Journal of Communications and Networks, 20(5): 509-518. https://doi.org/10.1109/JCN.2018.000075

[14] Alkhatib, B., Eddin, M.M.W.K. (2020). Voice identification using MFCC and vector quantization. Baghdad Science Journal, 17(Suppl.): 1019. https://doi.org/10.21123/bsj.2020.17.3(Suppl.).1019

[15] Singh, A., Joshi, A.M. (2020). Speaker identification through natural and whisper speech signal. Lecture Notes in Electrical Engineering, pp. 223-231. https://doi.org/10.1007/978-981-13-6159-3_24

[16] Devi, K.J., Singh, N., Thongam, K. (2020). Automatic speaker recognition from speech signals using self-organizing feature map and hybrid neural network. Microprocessors and Microsystems, 79: 103264. https://doi.org/10.1016/j.micpro.2020.103264

[17] Deng, M., Meng, T., Cao, J., Wang, S., Zhang, J., Fan, H. (2020). Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. Neural Networks, 130: 22-32. https://doi.org/10.1016/j.neunet.2020.06.015

[18] Hourri, S., Kharroubi, J. (2020). A deep learning approach for speaker recognition. International Journal of Speech Technology, 23(1): 123-131. https://doi.org/10.1007/s10772-019-09665-y

[19] Wibowo, A.S., Darmawan, I.D.M.B.A. (2021). Iqra reading verification with Mel frequency cepstrum coefficient and dynamic time warping. Journal of Physics: Conference Series, 1722(1): 012015. https://doi.org/10.1088/1742-6596/1722/1/012015

[20] Fontaine, M., Love, S.A., Latinus, M. (2017). Familiarity and voice representation: From acoustic-based representation to voice averages. Frontiers in Psychology, 8: 1180. https://doi.org/10.3389/fpsyg.2017.011

[21] Davis, R.O., Vincent, J., Park, T. (2019). Reconsidering the voice principle with non-native language speakers. Computers & Education, 140: 103605. https://doi.org/10.1016/j.compedu.2019.103605

[22] Arunachalam, R. (2019). A strategic approach to recognize the speech of the children with hearing impairment: different sets of features and models. Multimedia Tools and Applications, 78(15): 20787-20808. https://doi.org/10.1007/s11042-019-7329-6

[23] Giannakopoulos, T. (2009). A method for silence removal and segmentation of speech signals, implemented in Matlab. University of Athens, Athens.

[24] Radha, V., Vimala, C. (2012). A review on speech recognition challenges and approaches. World of Computer Science and Information Technology Journal, 2(1): 1-7.

[25] Ahmad, F., Yahya, S.Z., Saad, Z., Ahmad, A.R. (2018). Tajweed classification using artificial neural network. 2018 International Conference on Smart Communications and Networking (SmartNets), pp. 1-4. https://doi.org/10.1109/SMARTNETS.2018.8707394

[26] Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2): 257-286. https://doi.org/10.1109/5.18626

[27] Rojas, R. (2013). Neural Networks: A Systematic Introduction. Springer Science & Business Media. https://doi.org/10.1007/978-3-642-61068-4

[28] Likas, A., Vlassis, N., Verbeek, J.J. (2003). The global *k*-means clustering algorithm. Pattern Recognition, 36(2): 451-461. https://doi.org/10.1016/S0031-3203(02)00060-

2

[29] Vankayalapati, R., Ghutugade, K.B., Vannapuram, R., Prasanna, B.P.S. (2021). K-Means algorithm for clustering of learners performance levels using machine learning techniques. Rev. d'Intelligence Artif., 35(1): 99-104. https://doi.org/10.18280/ria.350112

[30] Wawage, P., Deshpande, Y. (2022). Real-time prediction of car driver’s emotions using facial expression with a convolutional neural network-based intelligent system. Acadlore Trans. Mach. Learn., 1(1): 22-29. https://doi.org/10.56578/ataiml010104

[31] Rehman, A., Butt, M.A., Zaman, M. (2022). Liver lesion segmentation using deep learning models. Acadlore Trans. Mach. Learn., 1(1): 61-67. https://doi.org/10.56578/ataiml010108

[32] Kumar, C.S., Rao, P.M. (2011). Design of an automatic speaker recognition system using MFCC, Vector Quantization and LBG algorithm. International Journal on Computer Science and Engineering, 3(8): 2942.

[33] Linde, Y., Buzo, A., Gray, R. (1980). An algorithm for Vector quantizer design. IEEE Trans. Comm, 28(1): 84-95. https://doi.org/10.1109/TCOM.1980.1094577

[34] AlShurbaji, M., Kader, L.A., Hannan, H., Mortula, M., Husseini, G.A. (2023). Comprehensive study of a diabetes mellitus mathematical model using numerical methods with stability and parametric analysis. International Journal of Environmental Research and Public Health, 20(2): 939. https://doi.org/10.3390/ijerph20020939.

[35] Torgo, L., Ribeiro, R. (2009). Precision and recall for regression. In International Conference on Discovery Science, pp. 332-346. Springer, Berlin, Heidelberg.