British Journal of Mathematics & Computer Science
14(1): 1-15, 2016, Article no.BJMCS.23034
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Arabic Isolated Word Speaker Dependent Recognition System

Amer El Kourd[1*] and Kaouther El Kourd[2]

[1]Department of Computer, Islamic University, Gaza, Palestine.
[2]Department of Physics, EPST School of Algiers, Algiers, Algeria.

Authors' contributions

This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJMCS/2016/23034
Editor(s):
(1) Tian-Xiao He, Department of Mathematics and Computer Science, Illinois Wesleyan University, USA.
Reviewers:
(1) Shuai Tao, Dalian University, China.
(2) Esam Al Qaralleh, University for Technology, Amman, Jordan.
(3) S. Selva Nidhyananthan, Mepco Schlenk Engineering College, India.
(4) Surbhi Mathur, Gujarat Forensic Sciences University, India.
Complete Peer review History: http://sciencedomain.org/review-history/13021

Original Research Article

Received: 11th November 2015
Accepted: 24th December 2015
Published: 19th January 2016

Abstract

This paper describes the development of a new Arabic isolated word speaker dependent recognition system based on a combination of several features extraction and classifications techniques. Where, the system combines the methods outputs using a voting rule. The dataset used in this system include 40 Arabic words recorded in a calm environment with 5 different speakers. We compared 5 different methods which are pairwise Euclidean distance with Mel-Frequency cepstral coefficients (MFCC), Dynamic Time Warping (DTW) with Formants features, Gaussian Mixture Model (GMM) with MFCC, Dynamic Time Warping (DTW) with MFCC features and Itakura distance with Linear Predictive Coding features (LPC) and we got a recognition rate of 85.23%, 57% , 87%, 90%, 83% respectively. In order to improve the accuracy of the system, we tested several combinations of these 5 methods. We find that the best combination is MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura with an accuracy of 94.39% but with large computation time of 2.9 seconds. In order to reduce the computation time of this hybrid, we compare several subcombination of it and find that the best performance in trade off computation time is by first combining MFCC | Euclidean + LPC | Itakura and only when the two methods do not match the system will add Formant | DTW + MFCC | DTW methods to the combination, where the average computation time is reduced to the half to 1.56 seconds and the system accuracy is improved to 94.56%.

**Corresponding author: E-mail: el_kourd@yahoo.com;*

Keywords: *Arabic speech recognition; isolated word; MFCC; FORMANTS; LPC; GMM; DTW; DWT; Euclidean; Itakura; hybrid system.*

1 Introduction

Automatic Speech Recognition system (ASR) is used to convert spoken words into text. It has very important applications such as command recognition, dictation, foreign language translation, security control (verify the identity of the person to allow access to services such as banking by telephone). ASR makes writing on computers applications much easier and faster than using keyboards and could help handicapped people to interact with society. Also, it could be used to remotely turn on/off the home lights and electrical appliances.

ASR has two main types Discrete Word Recognition Systems and Continuous Speech Recognition Systems; and each type can be further subdivided into two categories as Speaker Dependent and Speaker Independent. Speaker dependent speech recognition systems operate only on the speech of a particular speaker for which the system is trained while the Speaker Independent Systems can be operated on the speech of any speaker.

Speech production is a complicated process. Even though people may sound alike to the human ear, everybody, to some degree, have a different and unique annunciation in their speech. Even the same speaker cannot produce the same utterance twice. Moreover, speech can be distorted by noise due to background noise, noise generated by microphones or different background environment during training and testing as well as emotional and the physical conditions of an individual. Speech variation are due to speaking style, speaking rate, gender, age, accent, environment, health condition, prosody, emotional state, spontaneity, speaking effort, dialect ,articulation effort, …etc.

ASR is still a challenging task; its performance is still far below the human one and the accuracy of current recognition systems is not sufficient especially the Arabic ones. Although Arabic is currently one of the most widely spoken language in the world, there has been relatively little speech recognition research on Arabic compared to the other languages [1,2,3].

The critical problem in developing highly accurate Arabic speech recognition systems is the choice of feature extraction and classification techniques [4,12,13]. Currently, most of the speech recognition system use Mel Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Models (HMM) in classification. System combination is one of the emerging techniques that can combine pattern techniques advantages and improve speech recognition accuracy. Very rare research in Arabic recognition has tried combination of features and classification approach. Bourouba et al. [5] presented a new arabic digit recognition system based on classifier combination of HMM and a supervised classifier (SVM or KNN) with MFCC and the log energy and pitch frequency feature extraction combination method .They found that using HMM classifier alone the accuracy is 88.26% and improved with the combined system to 92.72%. The limitation of their system is in using weak features and combined two slow classification methods. In this paper, we propose a new Arabic speech recognition system based on a combination of several features extraction and classifications techniques. In the proposed method, we use a word boundary detector in the preprocessing to automatically identify the words in the input signal by using the energy and the zero crossing rate. Then, we apply discrete wavelet transform to the speech signal before extracting the features to improve the accuracy of the recognition and to make the system more robust to noise.

After that, we try to find the best features combination between the most famous features extraction techniques: MFCC, Formants and Linear Predictive Coding (LPC). LPC has always been a popular feature due to its accurate estimate of the speech parameters and efficient computational model of speech [6]. The Formants represent the acoustic resonances produced by the dynamics of the vocal tract and depend on the shape of the mouth when producing sounds. Also, formants are important in determining the phonetic content of speech and require small storage and can be computed quickly. MFCC is one of the most popular feature extraction techniques used in speech recognition. It is based on the frequency domain of Mel scale

for human ear scale. Speech signal is expressed in the Mel frequency scale, in order to capture the important characteristics of speech.

Finally, we need to test several combinations between simple, fast and accurate classification approaches in order to find the best hybrid that improves the recognition accuracy and with the least computation times. We choose Gaussian Mixture Model (GMM), Template Matching with dynamic time wrapping and Pairwise Euclidean distances classification methods. GMM is very competitive when compared to other pattern recognition techniques. It is more simple and faster than HMM with very small or no performance degradation and do not require large training data and time consumption as neural network method. Template based approach is one of the simplest and earliest approaches which is very simple and fast, as compared with the HMM and ANN. It determines the similarity between unknown spoken word with each reference object in the training data and selecting the word with smallest distance. It has low error rates for distinctive words in speaker dependent isolated word recognition, and has simple programming requirements. In the similarity measure, we will use two distance methods: Euclidean and Dynamic Time Warping (DTW). Where Euclidean distance is a simple and fast algorithm and it is one of the most commonly used distance measures. Also, Dynamic Time Warping is widely used in the small-scale speech recognition systems. It is used to measure the similarity between two words which may vary in time to cope with different speaking speeds.

2 Problem Formulation

2.1 Data collection

In the data collection stage we recorded 40 Arabic words with 5 different speakers (3 male and 2 female) using HP G62 Core I3 laptop microphone with sampling frequency of 8 kHz, 16-bit PCM WAV format. Each speaker read every word 8 times (5 of them are used in training and the remaining are used in the test phase). The list of the words is shown in Table 1:

Table 1. List of words used in the system

1	امام	9	اسرع	17	تكلم	25	احمل	33	اثنين
2	خلف	10	تمهل	18	اسكت	26	انظر	34	ثلاثة
3	يمين	11	افتح	19	اجب	27	انطلق	35	اربعة
4	يسار	12	اغلق	20	فوق	28	اهد	36	خمسة
5	اعلى	13	انزل	21	ابدأ	29	نعم	37	ستة
6	اسفل	14	اصعد	22	توقف	30	لا	38	سبعة
7	تحرك	15	اقرأ	23	اكمل	31	صفر	39	ثمانية
8	قف	16	اكتب	24	امسح	32	واحد	40	تسعة

2.2 Software

Two software programs are used during the development of the recognition system

- MATLAB R2010a: is used in writing the code of the system. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
- Praat software: is used in voice editing and spectrum analysis of the collected data.

2.3 System block diagram

The speech recognition system consists of two stages, a training stage and a recognition stage both stages have common blocks which are wave recording, speech pre-processing, word boundary detection and features extraction. The output of the training stage is a reference model .In the recognition stage the extracted features are compared with the reference model and the word that has the best match will be the output. Fig. 1 shows the block diagram of the System.

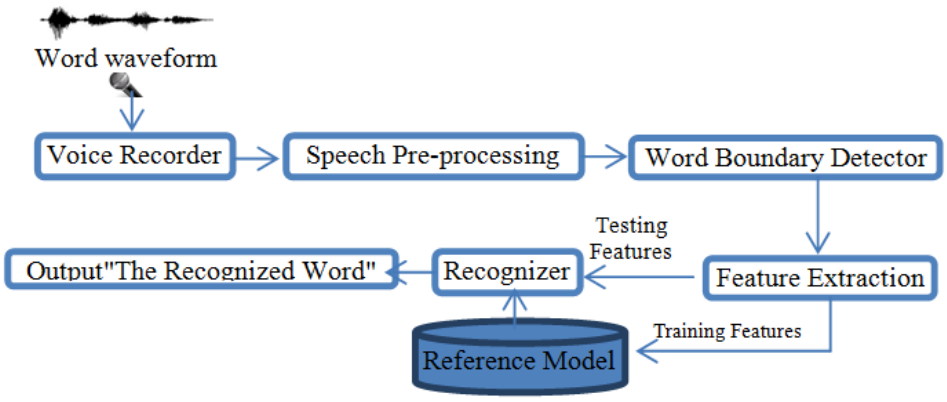

Fig. 1. System block diagram

2.4 Pre-processing

Preprocessing is used before features extraction in order to reduce noise in speech signal and to enhance recognition accuracy.

In the first step of pre-processing we remove the DC offset of the signal, since the microphone with A/D converter may add a DC offset voltage to the output signal. Removing the DC offset is important in order to determine the boundary of words.

In the second step, we make normalization on speech signals by dividing the signal by its maximum absolute value to make the signals comparable regardless of differences in magnitude.

Finally, we applied discrete wavelet transform to the speech signal before extracting the features to improve the accuracy of the recognition and to make the system more robust to noise. We tested several wavelets families and levels: Haar (Daubechies 1), Daubechies 2, Daubechies 3, Daubechies 5, Daubechies 15, Coiflets, Symlets, Discrete Meyer; we find best result by using second level Daubechies wavelets. The discrete wavelet transform divide the signal into approximation and detail coefficients, we take only the approximation coefficients vector as input for feature extraction stage.

2.5 End point detection

We use end point detection to extract the word speech and remove the background noise and silence at the beginning and end of the word speech. End point detection improves performance of an ASR system in terms of accuracy and speed.

The block diagram of the End Point Detection is shown in Fig. 2:

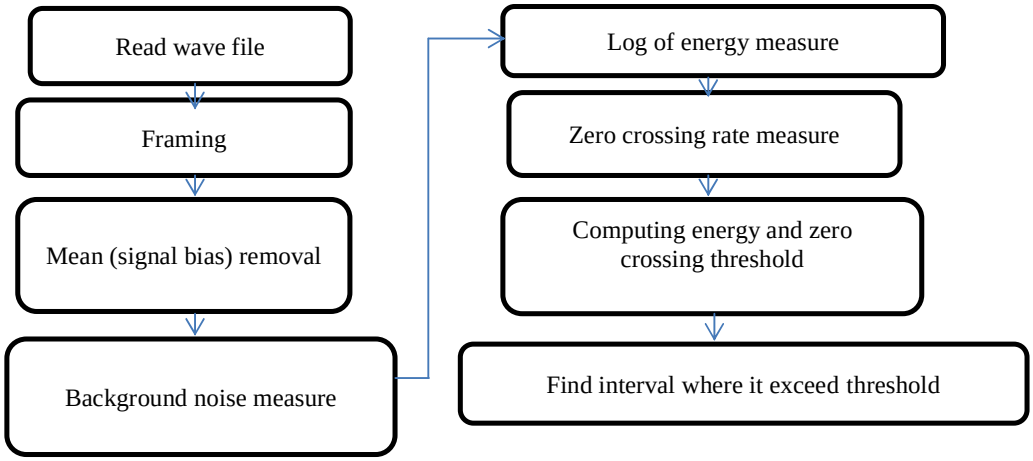

Fig. 2. End point detection block diagram

In the first step, we divide the sound into small frames of size 20 ms with 50% overlap, in order to have a stationary sound.

In the second step, we remove the mean for each frame to reduce the effect of noise.

In step 3, we estimate the noise in speech by computing the Log of energy and zero crossing rates of the silence signal frames.

In step 4, we measure the Log of energy Es using equation1:

$$Es = Log\left(e + \sum S(n)^2\right) \quad (1)$$

Where S (n) is signal values in the frame and e is a small positive constant added to prevent the computing of log of zero.

In step 5, we measure the zero crossing rate which refers to the number of times speech samples change sign in a given frame. Equation2 is used to compute the zero crossing rate Zcr(m).

$$Zcr(m) = \sum_{n=1}^{N} \frac{\left|sgn\left(S_m(n)\right) - sgn\left(S_m(n-1)\right)\right|}{2} \quad (2)$$

Where:

Zcr(m): is the zero crossing rate in the frame m
Sgn: is the sign function
S_m (n): is the speech signal in the sample number n in the frame m
N: is the frame size

In step 6, we measure the energy threshold using equation 3:

$$T_E = \mu_E + \propto \times \sigma_E \quad (3)$$

Where: μ_E is the mean and σ_E is the standard deviation of the energy of the noise frames. The α term is constant that have to be fine tuned according to the characteristics of signal. We tested several values of α in

the range from zero to one and we find that the best word boundary detection and system accuracy are with: α=0.5.

In step 7, we measure Zero-crossing rate threshold T_Z using equation 4:

$$T_Z = \max(\mu_Z + (\beta \times \sigma_Z), \quad 25) \tag{4}$$

Where: μ_Z is the mean and σ_Z is the standard deviation of the zero crossing rates of the noise frames and β are parameters obtained by experiments. We find a best value of β is 0.5. Also, according to many research the zero crossing rate of speech should be greater than 25 zero crossing per frame. Therefore the term 25 is included in the equation.

In step 8, we test each frame by comparing its energy and zero crossing rates with the energy and zero crossing thresholds. In order to find the start point and the end point of the word.

The pseudo code to find the start and end points of the word speech is shown below:

Algorithm 1: Endpoints detection

For each frame i in the speech signal
If frame_energy (i)≥T_E OR frame_zerocrossing (i)≥ T_Z
Then mark this frame as the Start point of the possible word
Elseif Start is found AND 9 successive frames do not satisfy threshold criteria
Then End point is the first frame before the 9 successive frames
End
Calculate number of frames between Start and End points
If it is greater than 25 frames (0.5 second).
Then a word is detected
Else we disregard it and we repeat the procedure to find other possible words.
End
End
The detected word is saved to be used for the feature extraction phase.

2.6 Feature extraction

In this paper, a combination of several famous features (MFCC, LPC, Formants) has been used to improve the accuracy of the system.

2.6.1 Mel-frequency cepstral coefficients (MFCC)

MFCC is one of the best known and most commonly used features for speech recognition. The Block diagram of MFCC is shown in the Fig. 3.

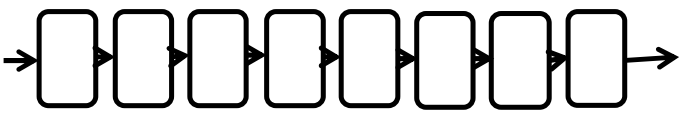

Fig. 3. MFCC block diagram

The step-by-step computations of MFCC are shown below:

Step 1: We divide the signal into small frames of length 32ms.
Step 2: We multiply the framed signal with an overlapped hamming window (Overlap =10 ms), to eliminate unwanted signal like noise and interference joined with the signal.
Step 3: We compute the Fast Fourier Transform (FFT) of the windowed frames to convert the signal from time domain to frequency domain to get the frequency content of speech signal in current frame.
Step 4: We compute the coefficients of a 22 triangular Mel filter banks, which are linearly spaced below 1000 Hz and logarithmic thereafter, since the information carried in low frequency components of the speech signal is more important than the high frequency components.
Step 5: We multiply these filters with power spectrum obtained in step 3 and we normalize it and we calculate the logarithm of each Mel power spectrum coefficient.
Step 6: We apply Discrete Cosine Transform (DCT) to the results of step5, and we get the Mel Frequency Cepstrum Coefficients (MFCCs).

2.6.2 Linear predictive coding (LPC)

LPC has been considered one of the most powerful techniques for speech analysis. LPC relies on the lossless tube model of the vocal tract. For accurate vocal tract model: The order of the LPC should be greater than sample rate/1000 + 2. In this paper we compute 12 LPC coefficient using Levinson-Durbin Algorithm.

2.6.3 Formants

The formant frequencies are obtained by finding the angle of the roots of the LPC coefficients. We sort these Formants frequencies then we take only the first 3 Formants, since they are the most important in determining the uttered word.

2.7 Training stage

In the training stage we create the reference model for the training speech signals. This reference contains the LPC, MFCC and Formants features and their gaussian mixture models.

2.7.1 Training with Gaussian mixture model

To create the reference model, we use Gaussian mixture model to fit the extracted features of the training data. Gaussian Mixture Models form clusters by representing the probability density function of observed variables as a mixture of multivariate normal densities. Mixture models of the gmdistribution class use expectation maximization (EM) algorithm to fit data, which assigns posterior probabilities to each component density with respect to each observation. Clusters are assigned by selecting the component that maximizes the posterior probability. The posterior probabilities for each point indicate that each data point has some probability of belonging to each cluster. Gaussian mixture modeling uses an iterative algorithm that converges to a local optimum.

To find the gaussian mixture model for each word that fit the training data and estimate its parameters, we use the Matlab command *gmdistribution.fit* with 5 Gaussian mixture components, 3 Replicates, diagonal covariance matrices and Maximum iterations of 500.

2.8 Recognition stage (test phase)

In the recognition stage a combination of recognition methods are used.

2.8.1 Euclidean distances

We use a Pairwise Euclidean distances between columns of MFCC test features matrix with each MFCC training matrices in the reference models.

First we calculate the Euclidean distance D between each column in x with each column in y.

$$D = \sqrt{\sum(x-y)^2} \quad (5)$$

Where x is the MFCC test features and y is the MFCC training features.

Then we find the minimum m value of each row in D. The distance d between x and y will be the average of m.

$$d=\text{Average}(m) \quad (6)$$

We repeat the above procedure to find the distance d between x and each training vector. The training vector that has the smallest distance d to the test vector x is the recognized word.

2.8.2 Dynamic time warping (DTW)

Dynamic Time Warping (DTW) is a technique that finds the optimal alignment between two time series if one time series may be "warped" non-linearly by stretching or shrinking it along its time axis. This warping between two time series can then be used to find corresponding regions between the two time series or to determine the similarity between the two time series.

Speech is a time-dependent process. Hence the utterances of the same word will have different durations, and utterances of the same word with the same duration will differ in the middle, due to different parts of the words being spoken at different rates. To obtain a global distance between two speech patterns (represented as a sequence of vectors) a time alignment must be performed. DTW resolves this problem by aligning the words properly and calculating the minimum distance between them. The local distance measure is the distance between features at a pair of frames while the global distance from beginning of utterance until last pair of frames reflects the similarity between two vectors. We used dynamic time warping to classify the MFCC and formants features. Since these features have not the same dimension. The algorithm of DTW is as follow:

Algorithm 2: Dynamic time warping

Purpose: Global distance between testing and training features

Input:

X: test features Formants

Y: training features Formants

Size(X) =[r, n]

Size(Y) =[r, m]

X and Y have same number of rows but different number of column $m \neq n$

D: Global distance, an n × m matrix.

Output: dist=D (n, m) the global distance.

Initialization:

Set all elements values in D to infinity.

Set the start element in D to zero, D (1, 1) =0.

Procedure:

for i=1:n

for j=1:m

$d = \sqrt{\sum_{k=1}^{r}(X(k,i)-Y(k,j))^2}$ where d: is the local distance (Euclidean distance between the two feature points and r is number of rows)

D(i,j)=d+ minimum of (D(i-1 ,j), // insertion

D(i , j-1), // deletion

D(i-1, j-1)) // match

end

end

Comparing the test features with each of the training features the one that have the smallest value of "dist" is considered the recognized word

2.8.3 Gaussian mixture model GMM recognizer

During the testing stage, we extract the MFCC vectors from the test speech and compare it with estimating GMM model of each word and use a probabilistic measure to determine the source word with maximum a posteriori probability (maximizing a log-likelihood value). The log-likelihood value is computed using the posterior function in Matlab.

2.8.4 Itakura distance (comparing two sets of LPC coefficients)

Given two vectors of LPC coefficients, it is often necessary to compute the "distance" between two LPC vectors in pattern recognition application such as speech recognition. The Euclidean and manhattan distance measures are not appropriate for comparing two vectors of LPC coefficients since the coefficients are not independent. The most useful distance measures for LPC coefficients are Itakura distance, which is defined as:

$$D_I(a, \hat{a}) = \log \frac{a^T R_x a}{\hat{a}^T R_{\hat{x}} \hat{a}}$$

Where

a and $\hat{a}$ are the pth-order LPC coefficients computed from two (windowed) speech frames x(n) and x($\hat{n}$) respectively.
Rx is the Toeplitz matrix calculated from the autocorrelation of the signal x(n).

3 Results

3.1 System graphic user interface (GUI)

We designed the system in a graphic user interface GUI in Matlab to make it simple to use. We have in the GUI 4 buttons:

- Start button: when pressed the system will start recording the sound for 2 minutes then recognizes the word
- Stop button: used to stop everything and remove any occurring errors
- Clear button :used to clear the workspace and command windows in Matlab and the textbox of the canvas
- Exit button: used to close the program and exit

The recognized word will appear in the textbox and each time you press the start button and read new word, it will be displayed next to it. Fig. 4 shows an example of reading 3 words.

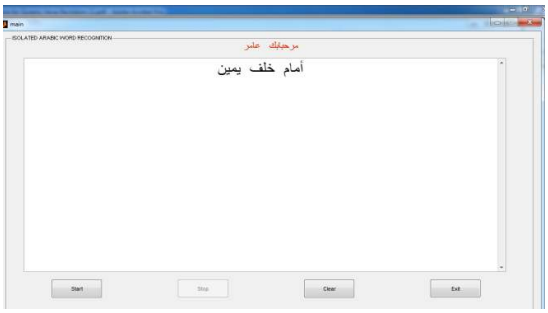

Fig. 4. Example of reading 3 words

3.2 Recognition methods experiments and results

We have performed different experiments using the following 5 methods and with their combination using a voting rule:

M1: Pairwise Euclidean Classification with MFCC features (MFCC | Euclidean)
M2: DTW Classification with Formants features (Formant | DTW)
M3: GMM Classification with MFCC features (MFCC | GMM)
M4: DTW Classification with MFCC features (MFCC | DTW)
M5: Itakura Classification with LPC features (LPC | Itakura)

In order to evaluate the recognition rate for each method, we calculate the method accuracy for each speaker using its 120 test data (40 words repeated 3 times). Then the overall accuracy of the method is the average accuracy of the 5 speakers.

Table 2. Recognition rates of the 5 methods

Method	Average accuracy
M1	85.23%
M2	57%
M3	87%
M4	90%
M5	83%

We find the recognition rates of the methods is 85.23%, 57%, 87%, 90%, 83% when using MFCC+Euclidean, Formants+DWT, MFCC+GMM, MFCC+DWT and LPC+ Itakura respectively. The worst case is with Formants and the best one is with MFCC and using Dynamic Time Warping classification.

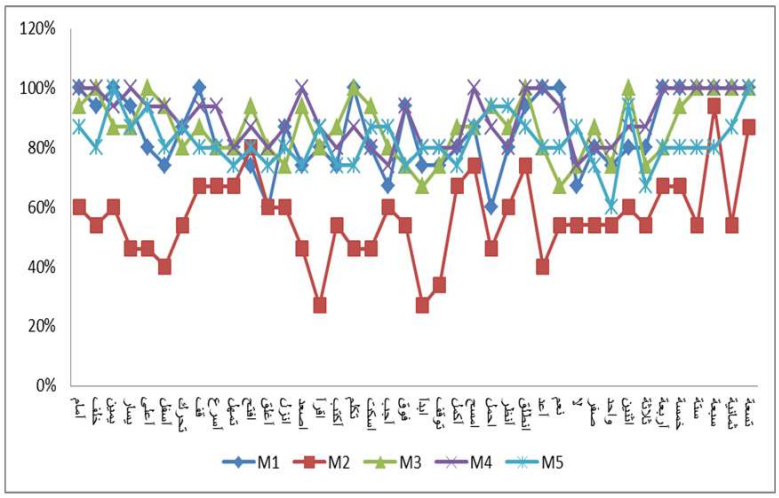

Fig. 5. Words accuracy by the diffferent methods

Also we see from the Fig. 6, that MFCC+DWT (M4) outperforms the other methods for most of the words and only for few words MFCC+GMM (M3) outperform M4 and it is clear from the graph that the worst method is Formants+DWT (M2) for almost all the words.

3.2.1 Combination of the methods

In the following table we will compare the performance (accuracy and execution time) of the different combination of the 5 methods using a voting rule. Where, the recognized item is the one that is recognized by the maximum number of methods. When no match between the methods, then we take the output of the best single method in the combination.

Let:

$M_i + M_j$: plus sign means combining the two methods
$(M_i + M_j) \rightarrow M_k$: means that the system will combine only M_i and M_j and when they did not give same classification then it will add M_k to the combined classifier.

Also, we need to combine at least 3 methods. Since combining 2 methods will have same output of the best single method. For example M1+M2:

If M1 output= M2 output then M1+M2 output= M1 output.
If M1 output ≠ M2 output then M1+M2 output= M1 output (will take the output of best single method).

Table 3. Performance of the different combinations of the 5 methods

Method combinations	Average accuracy	Average computation time (second)
M1	85.23%	0.7
M2	57%	0.3
M3	87%	0.2
M4	90 %	2.3
M5	83%	0.6
M1+M2+M3	85.73%	0.8
M1+M2+M4	92.33%	2.6
M1+M2+M5	86.94%	1.4
M1+M3+M4	92.5%	2.6
M1+M3+M5	90.27%	1.5
M1+M4+M5	93.83%	2.9
M2+M3+M4	92.39%	2.4
M2+M3+M5	89.94%	0.9
M2+M4+M5	92.39%	2.7
M3+M4+M5	93.72%	2.7
M1+M2+M3+M4	93.22%	2.6
M1+M2+M3+M5	90.72%	1.5
M1+M2+M4+M5	**94.39%**	2.9
M1+M3+M4+M5	93.60%	2.9
M2+M3+M4+M5	93.94%	2.7
M1+M2+M3+M4+M5	93.39%	3

From Table 3, we see that the best combination is M1+M2+M4+M5 (MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura) with an accuracy of 94.39% but its time computation is the largest 2.9 seconds. Also, MFCC with Gaussian mixture method is the fastest method with only 0.2 second but when it is combined with other methods does not give best result. This is due that our training data is not big enough and in our experiment, if we increase the training data this will increase the execution time too much, which is not suitable in our combination system case.

Also, we notice that such a combination can degrade the performance as in M1+M2+M3 the combined recognition rate is 85.73% is lower than the accuracy of M3 alone with 87%. For example if M3 has correct output whereas, M1and M2 have the same wrong outputs. Then the output of the voting system will be wrong even that M3 is correct.

Also we notice that the best single method is MFCC feature with Dynamic time warping but it is the most time consuming of all the single methods.

Since M1+M2+M4+M5 is the best method. We will select this combination and we will try to reduce the time computation by combining only two methods and when they do not match we will add another method to the combination. Also, we need to make the method M4 in the last decision of the combination since it is the most time consuming.

Table 4 shows the sub combination of M1+M2+M4+M5 to find the best accuracy and time computation.

Table 4. Subcombination of M1+M2+M4+M5 performances

	Average accuracy	Average computation time (second)
M1+M2→M5+M4	93.56%	1.55
M1+M5→M2+M4	**94.56%**	**1.56**
M2+M5→M1+M4	92.9%	1.75
M1+M2+M5→M4	92.7%	1.53

From the above table we find that the best one is M1+M5→M2+M4. Where first the system will combine the two fast methods M1 and M5 (MFCC | Euclidean + LPC | Itakura) and only when the two methods do not match the system will add other combination M2+M4 (Formant | DTW + MFCC | DTW). We notice that the average time computation of the datasets is reduced to the half and is less than the time of the single method M4 alone. Since M1+M5 have a match in 26 words and consumes 0.8 second whereas only 14 words will use M1+ M2+ M4 + M5 which consumes 2.9 second.

The positive effect of combination method on the recognition rate is clearly observed in Fig. 6, where best single method M4 has 90% and the combination of the methods improve the accuracy significantly to 94.56%. This is due that features combination adds an important speech parameters. Where MFCC gives some of the features of the words and the Formants and LPC give other features and combining them together will add more information of the words. Also, when using different classification method it improves the accuracy since the two methods will give the same classification to the word only when it has a high probability to be correct classification.

3.3 Comparison with other researches

In this section we will try to compare our proposed system with similar systems in previous researches that use features or classifications combinations. Table 5; summarizes the recognition rates obtained from the previous approaches. By comparing our system with the previous researches we conclude that our proposed system is very good and competitive to the other approaches. In our system we used 40 Arabic words whereas the others have used only 10 digits and only one with 19 words. Also, In order to have ideal comparison we need to have common database and same computer and software properties and with clear environment.

Table 5. Comparisons with previous researches

Paper title	Features	Data type	Classification methods	Dataset	Recognition accuracy
New Hybrid System (Supervised Classifier/Hmm) For Isolated Arabic Speech Recognition [5]	MFCC + log energy + pitch	Arabic digits	HMM + SVM /KNN	920 samples (10 digits x 92 speakers)	92.72%
The second-order derivatives of MFCC for improving spoken Arabic digits recognition using Tree Distributions approximation Model and HMMs [7]	MFCC+ Log (energy) + (Δ and ΔΔ)	Arabic digits	HMMs+VQ	8800 samples (10 digits x 10 repetitions x 88 speakers)	98.41%
Efficient DTW-Based Speech Recognition System for Isolated Words of Arabic Language [8]	MFCC+ Log (energy) + (Δ and ΔΔ)	Arabic words and digits	DTW	1710 samples (30 speaker x 19 words x 3 repetitions)	98.5%
Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition [9]	LPC + LPCC + Delta LPC	Arabic digits	VQ+HMM	1500 samples (50 speakers x 3 repetition x 10 digits).	91%
Multi-band based recognition of spoken Arabic numerals using wavelet transform [10]	Wavelet + MFCC	Arabic digits	HMM	data set consists of 500 utterances by 50 speakers	88.46%
A Comparison of DHMM and DTW for Isolated Digits Recognition System of Arabic Language [11]	MFCC+Energy+ (Δ and ΔΔ)	Arabic digits	DTW+DHMM	500 samples (5 speakers x 10 digits x 10 repetitions)	92%
Our Proposed System	MFCC+ LPC+ Formants	Arabic words and digits	Euclidean+ DTW+ Itakura	600 Samples (5 speakers x 40 digits x 3 repetitions)	94.56%

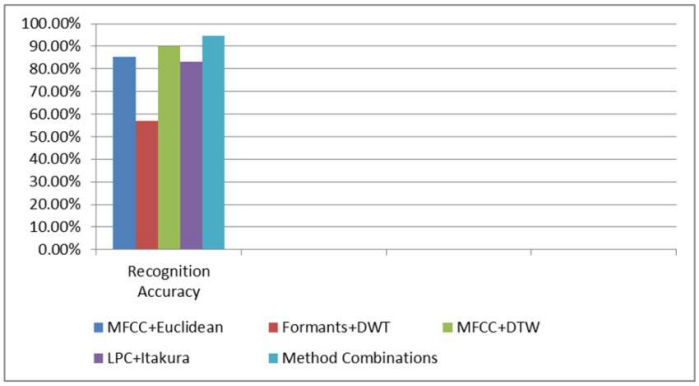

Fig. 6. Recognition accuracy for different methods

4 Conclusion

In this paper, we designed a new speaker dependent isolated Arabic word speech recognition system based on a combination of several methods outputs using a voting rule. We compare 5 different methods which are MFCC+Euclidean, Formants+DTW, MFCC+GMM, MFCC+DTW and LPC+Itakura and we get a recognition rate of 85.23%, 57%, 87%, 90%, 83% respectively. In order to improve the accuracy of the system, we tested several combinations of these 5 methods. We find that the best combination is MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura with an accuracy of 94.39% but its time computation is the largest 2.9 seconds. Also, we find that some combination can degrade the performance of the system. In order to reduce the computation time of this hybrid, we compare several subcombination of this hybrid and we find that the best performance in trade off computation time is with the system combining MFCC | Euclidean + LPC | Itakura and only when the two methods do not match the system will add the other combination Formant | DTW + MFCC | DTW. Where the average computation time is reduced to the half is 1.56 seconds and the system accuracy is improved become 94.56%.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Kirchho K, Bilmes J, Henderson J, Schwartz R, Noamany M, Schone P, Ji G, Das S, Egan M, He F, Vergyri D, Liu D, Duta N. Novel approaches to Arabic speech recognition. Technical Report, Ohns-Hopkins University; 2002.

[2] Vergyri D, Kirchhoff K. Automatic diacritization of Arabic for acoustic modelling in speech recognition. Editors, Coling, Geneva; 2004.

[3] Vergyri D, Kirchhoff K, Duh K, Stolcke A. Morphology based language modeling for Arabic speech recognition. In Proceedings of Interspeech, Germany. 2004;2245-2248.

[4] Choiy H, Gutierrez R, Choiz S, Choe Y. Kernel oriented discriminant analysis for speaker-independent phoneme spaces. icpr; 2008.

[5] Bourouba H, Djemili R, et al. New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition. 2nd Information and Communication Technologies, ICTTA-06; 2006.

[6] Atal BS, Schroeder MR. Predictive coding of speech signals. In Report of the 5th Int. Congress on Acoustics; 1968.

[7] Hammami N, Bedda M, Farah N. The second-order derivatives of MFCC for improving spoken Arabic digits recognition using tree distributions approximation model and HMMs. In Proc. IEEE Intl. Conf. on Communications and Information Technology (ICCIT). 2012;1-5.

[8] Darabkh KA, Khalifeh AF, Jafar I, Bathech BA, Sabah SW. Efficient DTW-based speech recognition system for isolated words of Arabic language. In the Proceedings of the International Conference on Electrical and Computer Systems Engineering (ICECSE2013), Switzerland, May; 2013.

[9] Bahi H, Sellami M. Combination of vector quantization and hidden Markov models for Arabic speech recognition. Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Beirut, Lebanon. 2001;96-100.

[10] Alkhaldi W, Fakhr W, Hamdy N. Multi-band based recognition of spoken Arabic numerals using wavelet transform. Proceedings of the 19th National Radio Science Conference (NRSC'01), Alexandria University, Alexandria, Egypt, March 19-21; 2002.

[11] Hachkar Z, Farchi A, Mounir B, El Abbadi J. A comparison of DHMM and DTW for isolated digits recognition system of Arabic language. International Journal on Computer Science and Engineering. 2011;3(3):1002-1008.

[12] Jain AK, Duin RPW, Mao J. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000;22(1):4–38.

[13] Koerich A. Improving classification performance using metaclasses. In IEEE International Conference on Systems, Man and Cybernetics. 2003;717–722.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/13021