Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.402 – 407

RESEARCH ARTICLE

SPEAKER RECOGNITION AND AUTHENTICATION

Rakesh D R[1], JAYASIMHAN N K[2]

[1]Dept. of instrumentation and technology, Rashtreeya Vidyalaya College of Engineering, India
[2]Dept. of instrumentation and technology, Rashtreeya Vidyalaya College of Engineering, India

[1] *rakeshdr02@gmail.com*

Abstract— In today's society, highly accurate personal identification systems are required. Passwords or pin numbers can be forgotten or forged and are no longer considered to offer a high level of security. The use of biological features, biometrics, is becoming widely accepted as the next level for security systems. Speaker Recognition and Authentication is a method of identifying persons from their voice. Speaker-specific characteristics exist in speech signals due to different speakers having different resonances of the vocal tract; these features are extracted from the voice of the speaker and used for the recognition of individual speaker. In this paper we present a real-time text dependent speaker recognition and authentication system which serve as intermediary step towards the implementation as an embedded system. Process is based on computing the Mel Frequency Cepstral Coefficients and the derived Dynamic Coefficients, while classifying features using a Dynamic Time Warping approach.

Key Terms: - Cepstral analysis; Delta Coefficients; real-time processing; Dynamic Time Warping

I. INTRODUCTION

This document Speaker recognition resides at the cross-border between two major areas of the computer science domain: biometry and namely natural language technologies. A quick web search on the topic will return numerous works, showing a high interest for developing applications using this technology as the main feature for solving problems related to identity theft, surveillance, authentication, etc. Speaker recognition also divides into two main application classes: speaker verification and identification. The first uses the voice individual features of a person, to give an answer over his claimed identity; the main objective is to secure an action or a location. This class is mainly related to biometry and its implementation benefits are described for example in [1] and its cited references.

Speaker identification also has a great potential for developing useful applications, like a timekeeping system for a working unit, identifying a person based on his voice characteristics when he "salutes" the application's capture device, recording relevant data. Another example application employs methods related to speaker identification to select and observe a certain communication channel between two persons, when one speaker is a tracked person.

The current work proposes a complete text dependent speaker identification prototype implemented using Mat lab. First, the main use cases of the application will be presented in Section II, followed in Section III by the description of the most important speech processing concepts employed in the system's development. Section IV describes the software implementation, while the conclusions are given in Section V.

II. USAGE SCENARIO

All First of all, the developed system must divide the users into two categories: administrators and users. The first type must be aware of the technologies involving speaker identification, being able to change the system's parameters when it notices that the performances are not satisfying. An administrator can change parameters related to the speech sampling rate, the bit rate, and can even manage parameters regarding the voice characteristics extraction. When handling software that stores voice reference models for numerous users, the administrator can even set speech processing options for each user.

The second category, represented by regular users will be involved into two main scenarios. First, a user will need to store a speech reference model, for an established word or phrase. In order to do that he will access a registration panel and he will start recording. When finished, he will be able to test the inputted speech in order to check if it has been recorded properly. If successful, he will then enter his name, thus ending the registration process which can be repeated any time the user wishes. The system administrator may request him to input two reference models, in order to use the second recording in periodic evaluations.

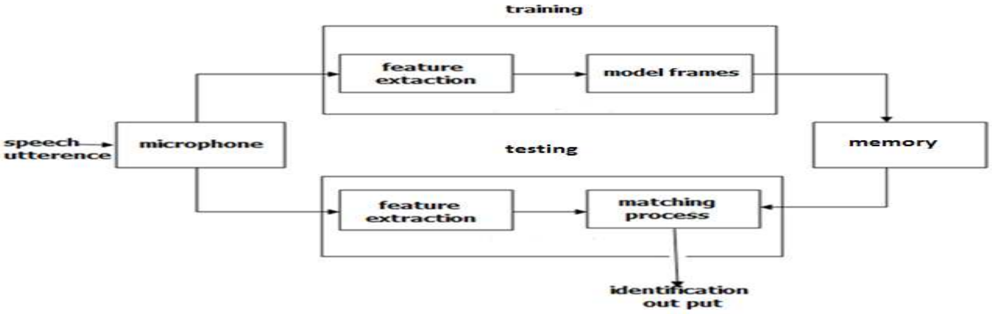

Figure 1- speaker recognition and authentication system

III. Speaker Identification Methods

This section describes the relevant speech processing instruments used to accomplish the main functionality of the application presented in the paper.

A. Speaker Identification Processing Chain

A system having as main objective speaker identification has a well-defined set of steps that have to be completed. Because the identification process resumes to selecting a certain person from a given set, the voice characteristics extracted from one input speech signal have to be compared with all the voice features associated to users registered in the application's database. Figure 1 depicted the main components of such a system.

The signal acquisition block captures the input speech signal and prepares it for the feature extraction algorithms. This preparation can be a time domain normalization as presented in [6] (which reduces the effect of signal's length variation when pronouncing the same word) or amplitude normalization, but its most important task is to reduce the noise.

A voice activity detector (VAD) discards the moments of silence from the input speech signal. One method to select the effective speech moments, presented in [7], is to construct a noise model during a calibration period, and to adapt the coefficients of a filter to eliminate the noise with the computed profile. Then, a signal window will be marked as speech if its energy is higher than an established threshold.

The feature extraction block selects the individual voice features from a captured input speech. When the identification is requested, the feature comparison block computes similarity measures for each pair of features passed as arguments. For an identification request, the input features of an unknown voice signal are compared to all the recordings inside the database and the user whose features generate the best match with the input data set will be claimed as the source of the analysed voice signal.

B. Noise Reduction

In order to select the relevant signal, we used an adaptive filter configured for noise cancellation which can also be adapted for echo cancellation.

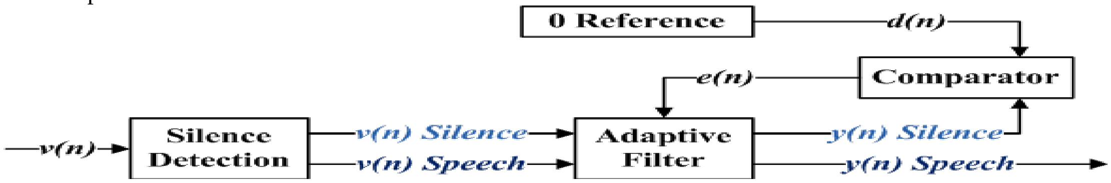

Figure 2. Adaptive filtering for noise cancellation

The echo cancellation is used only when the capture device is a microphone attached to a pair of headphones. When not in this situation, it can be deactivated by the application's administrator.

The noise cancellation was implemented as illustrated in Fig. 2. The silence moments from the input signal, denoted v(n), were considered as the noise signal and used to construct a noise profile, adapting a filter that can eliminate it. This process can run permanently, ensuring that the noise is removed properly even if its profile changes (e.g., when powering a server in the room, starting an engine sound, etc.). In a ready to use software system, having as main focus speech analysis and recognition, the filtering will be done in successive layers.

C. Feature Extraction

The feature extraction is the key task of any speech or speaker recognition system. It is responsible for selecting the interest part of the speech signal, and, in our case, for extracting the speaker's voice feature. We used a classical approach to implement this block, by computing the Mel Frequency Cepstral Coefficients (MFCC), $C_{i,j}$ and also the derived Dynamic Coefficients (i.e., Delta Coefficients, $\Delta_{i,j}$, and Delta Delta Coefficients, $\Delta^2_{i,j}$) for each input. A detailed presentation on how to tune the algorithm's parameters, and how they can influence the recognition performance, can be found in [8]. The algorithm's main steps are described in the following paragraphs. Let us denote by N the number of MFCC generated for each window.

First, the input speech signal is split into M overlapping windows with a fixed number of samples L: $w_1(n)$, $w_2(n)$, …, $w_M(n)$. Usually the overlapping fraction varies between 0.25 and 0.75, and the window's time length from 10 milliseconds to 50 milliseconds. A window can be multiplied by a weighting function like Hamming, Triangular, etc. Then, we proceed to computing the set of N coefficients for all the signal's windows, as described by the following steps.

For a k indexed window, the Fast Fourier Transform (FFT) The window's spectrum is then passed through each FFT filter from a bank of filters designed using the Mel perceptual scale, and described in detail in [8], resulting a set of filtered spectrums: $S_1W_K(f)$, $S_2W_K(f)$, …, $S_NW_K(f)$.

The decimal logarithm of the spectral power generated by each filtering operation is computed according to (1), and inserted into a vector P with N elements, i.e.,

$$P(i) = \log_{10}\left[\sum_{f=0}^{N_{FFT}} Si_{\mathrm{Wk}}(f)\, Si^{*}_{\mathrm{Wk}}(f)\right] \qquad (1)$$

A Discrete Cosine Transform (DCT) is applied to the constructed vector, according to (2), generating the set of MFCC for the selected window k:

$$C(i,k) = \begin{cases} \dfrac{2}{\sqrt{N}} \displaystyle\sum_{q=0}^{N-1} P_q \cos\left[\frac{\pi}{q}\left(q+\frac{1}{2}\right)i\right] & \text{if } i \neq 0 \\ \dfrac{1}{\sqrt{2}} \displaystyle\sum_{q=0}^{N-1} P_q & \text{if } i = 0 \end{cases} \qquad (2)$$

In the end, the resulting MFCC will be stored in a matrix, as follows:

$$C = \begin{vmatrix} c_{1,1}c_{1,2}\cdots\cdots\cdots c_{1,M} \\ c_{2,1}c_{2,2}\cdots\cdots\cdots c_{2,M} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ c_{N,1}c_{N,2}\cdots\cdots c_{N,M} \end{vmatrix} \qquad (3)$$

Also, we can add another coefficient, ci,N+1, to the obtained matrix by computing the signal window's energy as

$$c_{N+1,\mathrm{k}} = \sum_{n=0}^{L} \left|w_{\mathrm{k}}(n)\right|^2 \qquad (4)$$

The above coefficients matrix will be used in the following steps to compute the derivate Dynamic Coefficients ($\Delta_{i,j}$and $\Delta^2{}_{i,j}$*), according to.*

$$\Delta(p,q) = \frac{\sum\limits_{i=1} i(C^{'}[p+i,q] - C^{'}[p-i,q])}{2\sum\limits_{i=1}^{k} i^2}$$

$$\Delta^2(p,q) = \frac{\sum\limits_{i=1}^{k} i(\Delta^{'}[p+i,q] - \Delta^{'}[p-i,q])}{2\sum\limits_{i=1}^{k} i^2} \quad (5)$$

The equations are very similar to the ones describing a finite impulse response filter, except that the output is delayed with a number of k cycles. We can use this approach to determine the Delta Coefficients by following the steps below.

To compute the Delta Delta coefficients, the process is identical, except that the k parameter can be different.

- We copy the first line of C matrix on top of it, by a number of k times.
- We copy at the bottom of C matrix, its last line by a number of k times.
- In order to obtain the Delta Coefficients as a column in a Δ matrix, we apply h(n) defined as

$$h[n] = \frac{-(n-N)}{2\sum\limits_{i=1}^{N} i^2} \quad (6)$$

to the corresponding column in C' (the newly modified C matrix).

In order to obtain the first order Delta Coefficients, we need to apply the algorithm described above, having the C matrix as input. Then, we will consider the resulting Δ matrix as input to the same algorithm (eventually after modifying the k parameter) with the purpose of computing the Δ2 matrix. After completing all the steps described in the current subsection, the input speech signal will be transformed into a set of values representing the description of the speaker's voice characteristic. The values will be stored into an F matrix, defined as

$$F = \begin{vmatrix} C \\ \Delta \\ \Delta^2 \end{vmatrix} \quad (7)$$

The matrix F will be considered the feature matrix

D. Feature Comparison

In the proposed system, we have chosen the Dynamic Time Warping (DTW) algorithm to generate a similarity measure between the two given features. The algorithm has the main disadvantage that its complexity is O(N^2) but also has the advantage of reducing the issue generated by the misalignment between two compared sequences of values. The DTW generates a distance between two arrays. If the arrays are T and S, of N, respectively M values, the distance is represented by the D[N,M] value in a matrix having N rows and M columns, generated according to

$$D[i,j] = \left|S[i-1] - T[j-1]\right| + \min\begin{cases} D[i-1,j] \\ D[i,j-1] \\ D[i-1,j-1] \end{cases} \quad (8)$$

For two feature matrixes F1 and F2, we propose to compute a single similarity degree. A DTW distance will be calculated between the lines having the same index in F1 and F2; after that, the searched value will be the average distance, considering NMFCC as the number of MFCC, i.e.,

$$D(F1,F2)=\frac{\sum_{i=0}^{3*N_{\mathrm{MFCC}}} DTW(F1_i,F2_i)}{3*N_{\mathrm{MFCC}}} \qquad (9)$$

Finally, when a record inside the speaker features database generates the minimum average distance when compared with the input feature, the associated user will be identified as the one who produced the input speech signal.

IV. SOFTWARE IMPLEMENTATION

Mat lab is used for software implementation. A graphical user interface is created, which is divided into two parts namely training phase and testing phase.

In training phase the users are enrolled in the process. Voice samples of each individual user are taken and features are extracted using MFCC approach as described in the above section.

In testing phase voice from unknown speaker is taken and features are extracted as above and are compared with the features of users enrolled in the training process, dynamic time wrapping method is used for the comparison of the features and the distance measure is calculated, the user with least distance is declared as the authenticated user.

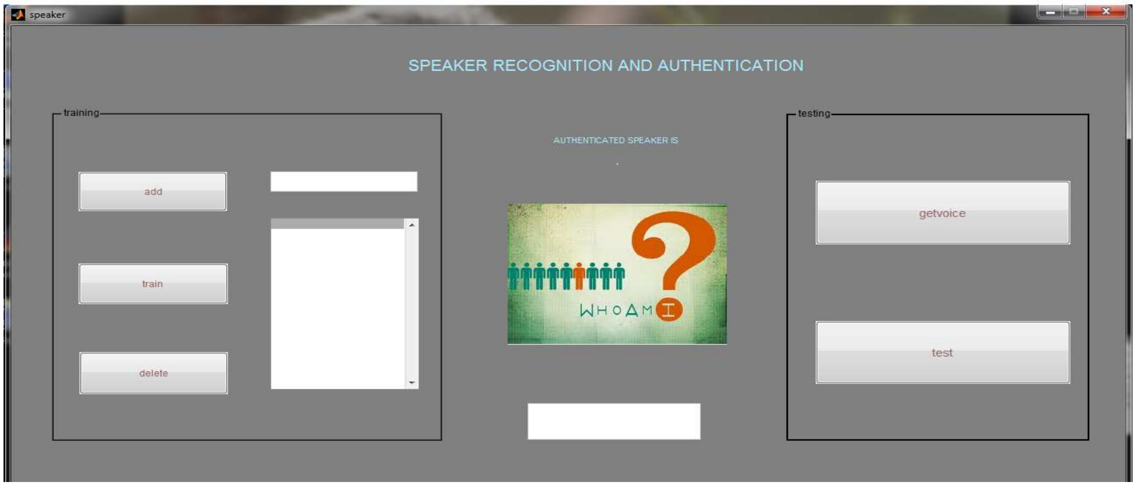

Figure 3. Graphical user interface for the system

- Add button in the graphical user interface adds the users to the system and the names of the registered user is shown in the pop menu beside the add button in the training panel of the graphical user interface.
- Train button in the graphical user interface extracts the features from the voice samples taken from the registered user and stores in the database.
- Delete button is used to remove the user from the process.
- Get voice button in the testing panel is used to record the voice sample from the unknown speaker.
- Test button performs feature extraction of the unknown voice sample and compares it with the registered user's features.
- The name and the photo of the authenticated speaker are shown in the respective fields of graphical user interface.

The recordings were captured using a low cost microphone, we used 44.1 kHz as sampling frequency, with 16 bits per sample but afterwards we down-sampled the wave files to 8 bits per sample at 8 kHz. We have chosen this approach because the main goal is to build an application capable to work with low cost equipment, which can be integrated into an embedded system.

V. **RESULTS AND CONCLUSION**

For each speaker, we used one record to build his reference feature and the other ones for recognition. We determined the Correct Identification Ratio (CIR) as a percentage, by splitting the number of successful identifications to the total number of attempts.16 MFCC coefficients is an optimal value for constructing the speaker's features.

The paper described a prototype designed for speaker recognition, working in real time, based on extracting the MFCC along with the derived Delta and Delta Delta Coefficients, and classifying the speakers using the DTW algorithm, where DTW algorithm is the most computational intensive block of the system.

REFERENCES

[1] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, et. al., "A Tutorial on Text-Independent Speaker Verification," EURASIP Journal on Applied Signal Processing, Vol. 4, pp. 430 – 451, 2004.

[2] R. Islam and F. Rahman, "Improvement of Text Dependent Speaker Identification System Using Neuro-Genetic Hybrid Algorithm in Office Environmental Conditions," IJCSI International Journal of Computer Science Issues, Vol. 1, pp. 42 – 48, 2009.

[3] P. Staroniewicz and W. Majevski, "SVM Based Text-Dependent Speaker Identification For Large Set of Voices," in Proc. 12th European Signal Processing Conference (EUSIPCO), 2004, Vol. 1, pp. 333 – 336.

[4] Qin Jin, "Robust Speaker Recognition," Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

[5] D. J. Hejna, "Real-Time Time-Scale modification of speech via the Synchronized Overlapp Add method," Msc. Project, MIT, 1990.

[6] D.K. Freeman, G. Cosier, C. B. Southcott, and I. Boyd, "The Voice Activity Detector for the Pan-European Digital Cellular Mobile Telephone Service," in Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 1989, Vol. 1, pp. 369 – 372.

[7] Dan Jurafsky, "Speech Recognition, Synthesis and Dialogue" in Stanford Course of Speech Recognition and Synthesis, L4, winter 2009.

[8] Mike Brookes – MATLAB Toolbox for Speech Processing – Department of Electronincs and Engineering Imperial - http://www.ee.ic.ac.uk /hp/staff/dmb/voicebox/voicebox.html.

[9] MIT – Massachusets Institute of Technology. FFTW Project.

[10] PCWorld Magazine "nVidia Eyes CUDA in Mobile Devices", 1st of December 2010