Fusion of Spectrograph and LPC Analysis for Word Recognition: A New Fuzzy Approach

Reza HoseinNezhad
Postdoctoral Research Fellow
School of Eng. and Science
Swinburne Univ. of Technology
John Street, Hawthorn
Victoria 3122
Australia
rhoseinnezhad@swin.edu.au

Behzad Moshiri[*]
Associate Professor
ECE Department
Faculty of Engineering
University of Tehran
North Kargar Ave., Tehran
Iran
moshiri@ut.ac.ir

Parisa Eslambolchi
M.Sc. Student
ECE Department
Faculty of Engineering
University of Tehran
North Kargar Ave., Tehran
Iran
eslambol_par@engineer.com

*Also Control & Intelligent Processing, Center of Excellence ECE Department, University of Tehran, Iran

Abstract - *Word Recognition is generally difficult and imprecise if we use just one method. In this article, data fusion is applied to word recognition by integration of two features extracted form human speech: speech spectrograph and time domain features (spectral coefficients). Four different methods are applied to fusion of these features, including weighted averaging, k-means clustering, fuzzy k-means and fuzzy vector quantization. Simulation results show that fusion of time domain and spectrograph features yields more precise and satisfactory results compared to other methods of word recognition that use just one speech feature for word recognition, like FVQ/MLP (fuzzy vector quantization combined with multi-layered perceptron neural network). The importance of this result is prominent if the signal to noise ratio is low.*

Keywords: word recognition, fuzzy clustering, LPC analysis

1 Introduction

Automatic speech recognition has been an interesting major topic during last 4 decades and it has been referred as a modern man-machine interface in science fiction stories. The main reason of this assertion can be high-speed data entry, unsupervised learning and simple and cheap sensor [1]. Although there are many methods for word, phrase and phoneme recognition, there is no machine that exactly percepts personage, independent of person, accent, environment and talking subject.

There are various methods for word recognition such as Hidden Markov Models [2], Neural Networks [3] and Hybrid Gaussian Method [4]. It is important to note that in voice processing, a single voice model scarcely models the voice perfectly. Even if there were such a perfect voice model, it would not be useful due to complexity. In order to overcome the baffle, it is recommended to exploit different models and estimate the result with data fusion methods to use the benefits of speech recognition based on each model.

In this article, we try to perform speech recognition via voice frequency or spectrogram and voice cepstral coefficients. After feature extraction, association of extracted features to database words is evaluated by using both methods. The structure of data fusion based speech recognition is depicted in Fig. 1.

Data sets and preprocessing are explained in Sec. 2. Spectrograph and LPC analysis methods are reviewed in Sec. 3. Classical and fuzzy k-means clustering and fuzzy vector quantization methods are reformulated and explained in Sec. 4, as three alternative methods for decision level fusion in word recognition. Simulation results and conclusions are presented in Secs. 5 and 6.

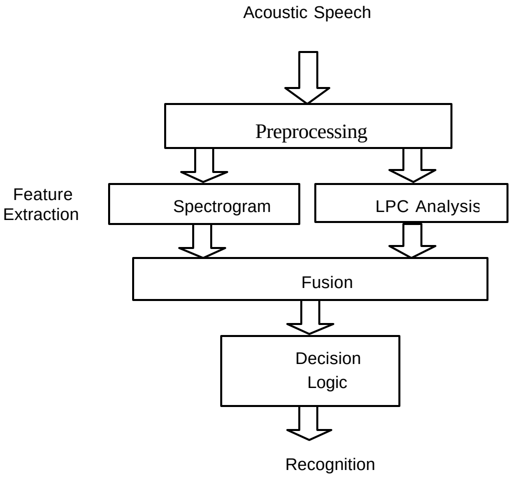

Fig. 1. Structure of the proposed word recognition method

2 Data Sets and preprocessing

The audio data, used for simulation, are ten isolated words each pronounced 50 times by a person in different conditions of SNR, speed of vocal expression and accents. The words are Persian numbers from one to ten. Five hundred voice samples were generated, which 300 of them were used for training and the rest for validation. The voice samples were recorded in a noisy environment but for the sake of acoustic noise reduction, headset was used. The sound was sampled at 8000 Hz with 16 bits per sample. Prior to feature extraction, some preprocessing was performed on the recorded sound. At first the silence detection [5] and spectral subtraction [6] were applied to reduce the additive noise. Then pre-emphasis and hamming window filters were applied. Finally the speech spectrum was extracted.

According to importance of frequency selection for comparison purposes, the produced spectrogram was filtered to choose some important frequencies. They must be selected carefully. In [1] it is shown that the dominant part of human voice spectrum is concentrated in frequency intervals of [200-800] and [1600-2000] and the rest has less importance. Hence we have used more frequency points or filters in these regions and a lower number of filters in the less important regions. In this research work, 50 filters were applied for evaluation of the voice spectrum.

3 Spectral Analysis and Specification

After preprocessing and signal preparation, the speech features (spectrogram and LPC coefficients) must be extracted so that the decision maker can recognize the pronounced word based on them. In order to modify the recorded sound so that it is comparable with the trained data, every recorded sound must be standardized. After start and end point detection, we divided the recorded sound into 50 sections and added zeros or deleted the samples till reached 250. For this purpose we added zeroes to the sound or deleted samples monotonically [1]. Feature extraction methods are explained in the following subsections.

3.1 Spectrograph method

We used 50 filters to produce the spectrograph of the sound spectrum. FFT amplitudes at 50 predefined frequencies for the 250 sample windows were measured and entered as columns of *Sp* matrix. We continued this procedure and filled the columns of *Sp* matrix for all 50 windows. The resulting *Sp* matrix can be used for comparing with the saved or trained data or can be used in the training phase.

In training phase *Sp* matrices for N = 50 sound samples were produced then *mean* and *variance* matrices were calculated and saved for comparison in recognition phase. A Gaussian similarity function performs this comparison as a classifier. The similarity function is defined in such a way that it can evaluate the similarity of a point (*i,j*) in *Sp* matrix with corresponding point in trained data. One of the appropriate possible definitions for such a function is expressed as follows:

$$f_{i,j,k} = \exp\left[-\tfrac{1}{2}\left(\frac{(M_R(i)-M_T(j,k))^2}{K_M^2} + \frac{(V_R(i)-V_T(j,k))^2}{K_V^2} \right) \right] \quad (1)$$

where $M_R(i)$ and $V_R(i)$ are the mean and the variance of the i^{th} frequency component, from the 50 frequencies of the recorded samples, respectively. Similarly $M_T(j,k)$ and $V_T(j,k)$ are the mean and variance of the j^{th} frequency components of the k^{th} trained voice sample, respectively. K_M and K_V are parameters whose appropriate values are chosen by trial and error. A typical plot of this function is depicted in Fig. 2.

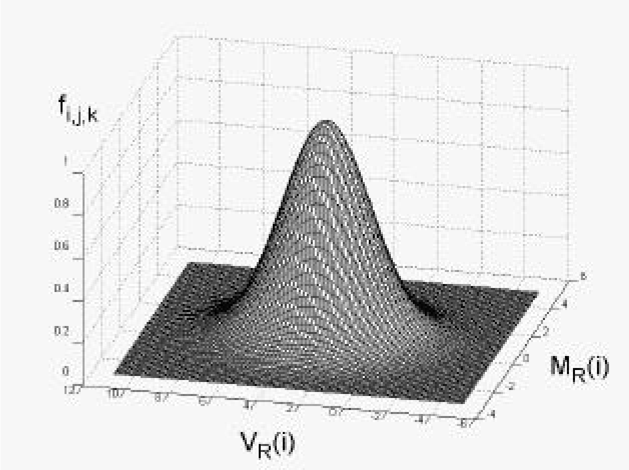

Fig. 2. A typical plot of the Gaussian similarity function with parameter values: K_M = 1, K_V = 2, $M_T(j,k)$ = 1, $V_T(j,k)$ =3

This similarity is evaluated for each of the 50 sections in each of the 50 frequencies. Thus, 2500 values are resulted. The total similarity is measured by the following equation:

$$S_k = \sum_{i=1}^{50}\sum_{j=1}^{50} f_{i,j,k} \quad (2)$$

where S_k is the similarity of the recorded voice with the k^{th} trained voice by the spectrogram method. Fig. 3 shows the spectrograph of ten Persian numbers from one to ten. It was produced by GRAM software that is downloadable from the internet.

3.2 Linear prediction coding (LPC)

LPC (Linear Predictive Coding) time domain analysis is applied to the voice after pre-emphasis, hamming windowing and autocorrelation. LPC coefficients are determined by Durbin-Levinson method and then autocorrelation coefficients will be converted to cepstral coefficients by LPC analysis. According to the higher accuracy and more robustness of the cepstral coefficients with respect to the LPC coefficients, we adopted cepstral coefficients. In cepstral analysis 12 coefficients have been used. After determining the cepstral coefficients, a reduction filter weighs the coefficients and their derivatives [1, 7]. *Cp* is defined as the matrix that consists of the cepstral and their derivative coefficients in different time durations. Similar to the previous section, there are training and recognition phases. In training phase *Cp* matrix is produced for N=50 voice sections then the *mean* and *variance* matrices are determined and saved to be compared in the next phase.

In recognition phase the saved *Cp* matrix is compared to the previously saved matrix, which is associated with the trained data. As a result of this comparison, T_k values are obtained as follows:

$$T_k = \sum_{i=1}^{n} \sum_{j=1}^{50} g_{i,j,k} \qquad (3)$$

where n is the number of coefficients in Cp and $g_{i,j,k}$ is a similarity measure that is defined with the same formulation of $f_{i,j,k}$, for evaluation of the similarity of the i^{th} cepstral coefficient of the saved matrix with respect to the j^{th} cepstral coefficient of the k^{th} trained voice sample. Indeed, here each T_k value is the similarity of the recorded voice sample with the k^{th} trained voice sample by spectrogram method. Cepstral Coefficients of Persian numbers from 1 to 10 are depicted in Fig. 4.

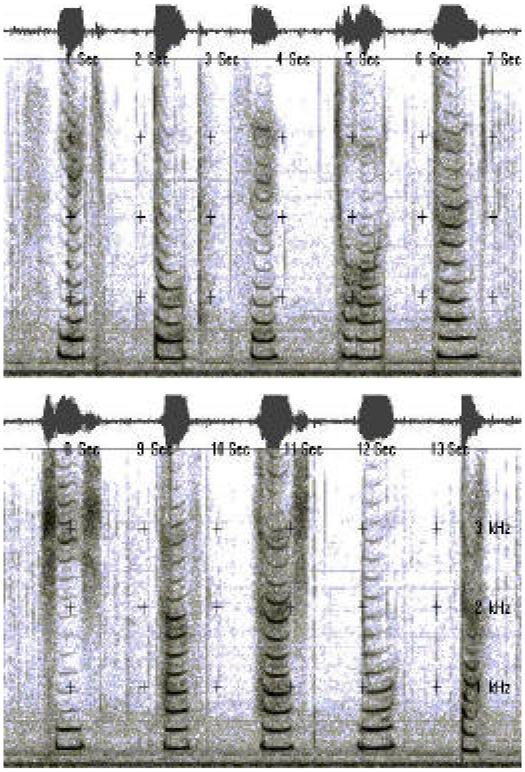

Fig. 3. Spectrograph of 10 Persian numbers from 1 to 10

4 Data Fusion Techniques for Classification Purposes

During the recent two decades, data fusion methods have been widely utilized for object identification by multiple sensors or multiple information sources. In these applications, a decision is made about an object by gathering information about it from several sources or sensors and fusing them together by using a data fusion technique. While the information/data volume increased, the importance of fusion of information/data obtained from multiple sensors increased. Fusion methods effectively fastened data processing and extracting information from existing data as much as possible [8, 9]. Several methods have been proposed for data fusion using neural networks, clustering algorithms, pattern recognition techniques, syntactic models, fuzzy logic, etc. They are generally categorized to centralized, autonomous, or hybrid fusion methods.

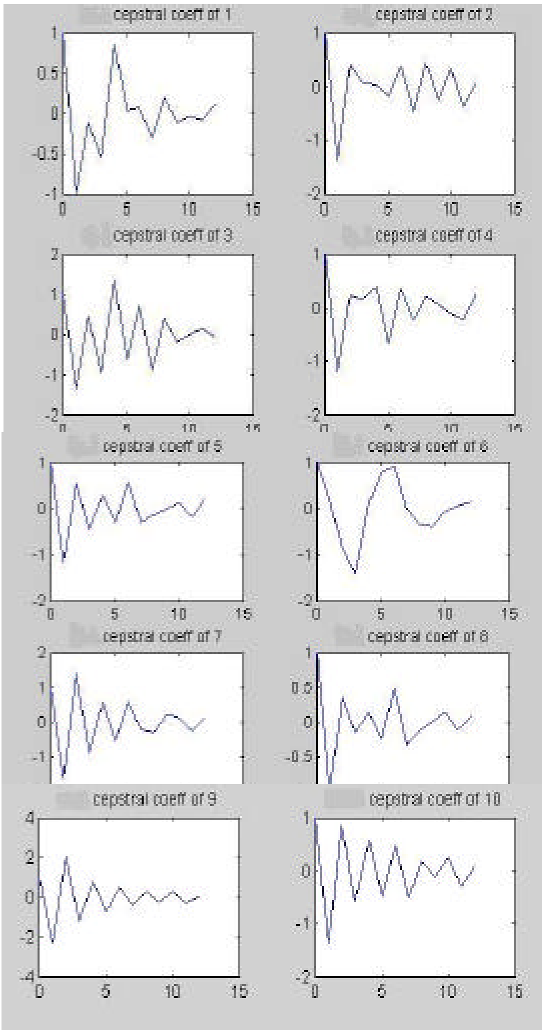

Fig. 4. Cepstral coefficients of 10 Persian numbers from 1 to 10

Decision level fusion expresses integration of information provided by multiple logical sensors (sensor plus a preliminary decision making). This information is all related with a decision to be made as the result of the fusion process. Examples of decision-level fusion methods include weighted decision methods, classical inference, Bayesian inference, and Dempster-Shafer method [10, 11].

Clustering methods refer to a wide variety of methods that attempt to subdivide a data set into subsets (clusters). Fuzzy clustering algorithms consider each cluster as a fuzzy set, while a membership function measures the possibility that each training vector belongs to this set. Fuzzy k-means (FKM) and fuzzy vector quantization (FVQ) have applied to decision level fusion in this research work. In weighted averaging method, we calculate the S_k and T_k values which are the measure of similarity of the k^{th} word in the trained database with the input word. Then, they must be fused together so that the most similar word is recognized.

One important aspect in this method is how to tune the averaging weights. Since the spectrograph and cepstral coefficients are speaker dependent, these weights must be obtained by trail and error. In order to obtain these coefficients, we can estimate them according to their true recognition percentage, e.g. if the cepstral recognizes the word by the precision of 86% and the spectrograph method performs it by the precision of 90%, then we assign P2 = 0.86 and P1= 0.9 and the total similarity is obtained as below:

$$SIM_k = \frac{S_k \times \text{P1} + T_k \times \text{P2}}{\text{P1} + \text{P2}} \qquad (4)$$

SIM_k is the degree of similarity between the input word and the k^{th} trained word. By using SIM_k values, following decision is made:

$$\begin{aligned} &A = MAX(SIM_k) \\ &\text{if } (A > A_{th}) \text{ then} \\ &\quad INPUT\_WORD = k^{th} \end{aligned} \qquad (5)$$

In this decision making criterion, if i^{th} word has the maximum similarity then it will be chosen as the pronounced word. If no word satisfies the above criterion then *not matched* response will be announced.

4.1 K-Means Clustering

In classical K-Means clustering *mean* and *variance* matrices are not required. Instead, *Sp* and *Cp* matrices are applied as training information sources. In this algorithm, each training vector is assigned to a cluster, which has the minimum distance from its center. This means that training vector must minimize the following function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{M} \left\| x_i - y_j \right\|^2 \qquad (6)$$

where x_i is a training vector and y_j is a codebook vector located at the cluster center.

Training vectors are very important in this method because the similarity of input word is evaluated according to its distance form each saved cluster center [12].

4.2 Fuzzy K-Means clustering (FKM)

In FKM algorithm each vector will be assigned to all clusters by a membership value in [0,1]. This membership value shows the likelihood of the input word to each of the saved words [12, 13]. In training phase the following cost function must be minimized:

$$J_m = \sum_{j=1}^{k} \sum_{i=1}^{M} u_j(x_i)^m \left\| x_i - y_j \right\|^2 \qquad 1 < m < \infty \qquad (7)$$

where x_i is a training vector, y_j is a codebook vector, located at the cluster center and $u_j(x_i)$ is the membership value for x_i belonging to j^{th} cluster and ||.|| is a geometric distance norm. The membership function is defined as follows:

$$\begin{aligned} &u_j(x_i) \in [0,1], \forall i, j \\ &\sum_{j=1}^{k} u_j(x_i) = 1 \\ &u_j(x_i) = \frac{1}{\sum_{l=1}^{k} \left(\frac{\left\| x_i - y_j \right\|^2}{\left\| x_i - y_l \right\|^2} \right)^{\frac{1}{m-1}}} \end{aligned} \qquad (8)$$

where, the parameter m controls the fuzzification of the clustering process. Again, similarity values are evaluated based on the J_m function values [14].

4.3 Fuzzy vector quantization (FVQ)

FVQ clustering is a soft decision making method. In the initialization step of this algorithm, each training vector is assigned to a codebook vector which is concentrated at a cluster center. Here the membership function $u_j(x_i)$ is equal to one when $||x_i - y_j||^2$ is zero and zero when $||x_i - y_j||^2$ is more than or equal to $d_{\max}(x_i)$. Otherwise membership value is calculated by the following formula:

$$u_j(x_i) = \left(1 - \frac{\left\| x_i - y_j \right\|^2}{d_{\max}(x_i)} \right)^m \qquad (9)$$

where μ is a positive integer that controls the fuzzification of the clustering process.. Each training vector is assigned to one cluster. Notice that similar to FKM, FVQ algorithm does not classify fuzzy data [11, 12].

5 Simulation Results

The data used in simulation, are the voice data samples which were described in Sec. 2. The recording format was 16 bits with 8kHz sampling rate. The effectiveness of the proposed fusion approach in the output of word recognition process was indicated by using isolated words for a specific speaker with different SNR levels. The algorithms described in Sec. 4 have been implemented to fuse results coming from the two different methods of speech recognition. The simulation results are abstracted in Tables 1 to 3.

In these tables FR and FA mean *False Rejection* and *False Acceptance* rates, respectively and the effectiveness of the proposed word recognition method is evaluated and compared with other methods according to these two rates. When the results of two different methods were combined, using the results coming from FKM succeeded the best fusion. A 1% false rejection rate and 0% false acceptance rate was obtained in SNR=30db. We can see this appropriate performance in other SNRs too.

The performance of fuzzy clustering algorithms especially FKM is shown to be apparently better than classical k-means or weighted average method. For more performance comparison, the result of the proposed approach is compared with the result of a combined fuzzy quantization method (FVQ) and multi-layered perceptron (MLP) neural network [15]. It must be mentioned that the input for all the systems is the same. The performance of FVQ/MLP can be evaluated by noisy input words with different SNRs. The system is trained with noiseless words (just cepstral coefficients have been used for training) and validated by noisy inputs. The simulation results for different number of neurons in the hidden layer of the MLP are presented in Table 4.

Table 1: Simulation Results by data fusion (SNR = 30db) KM = K-Means, FKM = Fuzzy K-Means, FVQ = Fuzzy Vector Quantization

Weighted Average		KM		FKM		FVQ	
FR	FA	FR	FA	FR	FA	FR	FA
4%	0%	1%	0%	1%	0%	5%	0%

Table 2: Simulation Results by data fusion (SNR = 20 db)

Weighted Average		KM		FKM		FVQ	
FR	FA	FR	FA	FR	FA	FR	FA
5%	2%	4%	2%	2%	1%	10%	1%

Table 3: Simulation Results by data fusion (SNR = 10 db)

Weighted Average		KM		FKM		FVQ	
FR	FA	FR	FA	FR	FA	FR	FA
12%	3%	8%	2%	3%	1%	11%	3%

Table 4: Simulation Results by FVQ/MLP for different numbers of neurons in hidden layer (P)

SNR(dB)	P=10		P=15		P=20	
	FR	FA	FR	FA	FR	FA
35	5%	5%	6%	4%	4%	3%
30	10%	12%	8%	8%	5%	8%
25	9%	7%	6%	6%	6%	5%
20	15%	13%	10%	12%	8%	10%

According to simulation results, the false rejection rate in SNR=35db and P = number of hidden neurons = 20 is about 4% and false acceptance rate is about 3%. For SNR=20db and the same number of neurons in the hidden layer of the MLP, false rejection rate is about 8% and false acceptance rate is about 10%. Hence, the rates of invalid recognitions by FVQ/MLP approach are obviously more compared to our proposed approach.

The disadvantage of our fusion-based recognition method is its high computational load which is noticeably larger than the computational load of FVQ/MLP approach. Indeed computational time is almost 2 minutes for the proposed method and 1 second for FVQ/MLP method, for the same processing and memory platform.

6 Conclusions

In this article the appropriate performance of data fusion in word recognition application has been investigated and compared with other approaches that use just one speech feature for word recognition, like FVQ/MLP. The use of fuzzy clustering algorithms for decision making and decision-level fusion in automatic isolated word recognition systems is proposed in this paper. Results coming from two speech recognition methods (spectrograph and time domain methods) were combined by using weighted average, K-Means clustering, Fuzzy K-Means and Fuzzy Vector Quantization. Simulation results show that fusion of fuzzy clustering output results a more appropriate performance compared to classical k-means or other known clustering algorithms. The results of the fusion-based technique were compared with the results of a combined fuzzy vector quantization and multi-layer perceptron (FVQ/MLP) method. The rates of false rejections and false acceptances of our recognition method is much more desirable than the result of other methods. The proposed approach drastically increases the accuracy of word recognition and especially in low SNRs it has a very high accuracy comparing with FVQ/MLP. High computational load is the price which is paid for this highly accurate word recognition outcome.

References

[1] Rabiner L., Juang B. H., *Fundamentals of speech recognition*, Prentice Hall, 1996

[2] Sato T., Ghulam M., Fukuda T., Nitta T., *Confidence scoring for accurate HMM-based word recognition by using SM-based monophone score normalization,* ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Volume 1, pp. I/217-I/220, 2002

[3] Yang S., Er M. J., Gao, Y., *A high performance neural-networks-based speech recognition system,* Proceedings of the International Joint Conference on Neural Networks, Volume 2, pp. 1527-1531, 2001

[4] Rigazio L., Tsakam, B., Junqua J. C., *Optimal Bhattacharyya centroid algorithm for Gaussian clustering with applications in automatic speech recognition,* ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Volume 3, pp. 1599-1602, 2000

[5] Penack J., Nelson D., The NP Speech Activity Detection Algorithm, *ICASSP-95*, Volume 1, pp. 381-384, Detroit, USA, May 1995

[6] Boll S. F., Suppression of Acoustic Noise in Speech Using Spectral Subtraction, *IEEE Trans. on ASSP*, Volume ASSP-27, No. 2, pp. 113-120, April 1979

[7] Jankowski C. R., A Comparison of Signal Processing Front Ends for Automatic Word

Recognition, *IEEE Trans. On Speech and Audio Processing*, Volume 3, 1995

[8] Tahani H., Keller J. M., Information Fusion in Computer Vision Using Fuzzy Integral Operator, *IEEE Trans on Systems, Man and Cybernetics*, Volume 20 , May 1994

[9] Dasarathy B. V., *Decision Fusion*, IEEE Computer Society Press, 1994

[10] Farrell K. R., Ramachandran R. P., Mammone R.J., An Analysis of Data Fusion Methods for Speaker Verification, *IEEE Trans. On Systems, Man and Cybernetics*, 1998

[11] Chibelushi C. C., Mason J. S. D., Deravi F., Feature Level Data Fusion for Bimodal Person Recognition, *IEEE Trans. On Systems, Man and Cybernetics*, 1997

[12] Chatzis V., Bors A. G., Pitas I., Multimodal Decision-Level Fusion for Person Authentication, *IEEE Trans. On Systems, Man and Cybernetics*, Volume 29, No. 6, Nov. 1999

[13] Karayiannis, Nicolaos B. An Axiomatic Approach to Soft Learning Vector Quantization and Clustering, *IEEE Trans. On N.N.* ,Volume 10, No. 5, Sept. 1999

[14] Gajdos S., Lorincz A., Fuzzy_Based Clustering of Speech Recognition Database , *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 3, pp. 2744-2749, Oct. 1998

[15] Cong L., Combining Fuzzy Vector Quantization and NN Classification For Robust Isolated Word Recognition, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2 pp. 1723-1727, Oct. 1998