Available online at www.sciencedirect.com
SCIENCE DIRECT
Applied Soft Computing xxx (2006) xxx–xxx
Applied Soft Computing
www.elsevier.com/locate/asoc

Recognition of human speech phonemes using a novel fuzzy approach

Ramin Halavati *, Saeed Bagheri Shouraki, Saman Harati Zadeh

Artificial Intelligence Lab 308, Computer Engineering Department, Sharif University of Technology, Tehran, Iran

Received 24 January 2005; received in revised form 14 February 2006; accepted 24 February 2006

Abstract

Recognition of human speech has long been a hot topic among artificial intelligence and signal processing researches. Most of current policies for this subject are based on extraction of precise features of voice signal and trying to make most out of them by heavy computations. But this focus on signal details has resulted in too much sensitivity to noise and as a result, the necessity of complex noise detection and removal algorithms, which composes a trade-off between fast or noise robust recognition.

This paper presents a novel approach to speech recognition using fuzzy modeling and decision making that ignores noise instead of its detection and removal. To do so, the speech spectrogram is converted into a fuzzy linguistic description and this description is used instead of precise acoustic features. During the training period, a genetic algorithm finds appropriate definitions for phonemes, and when these definitions are ready, a simple novel operator consisting of low cost functions such as Max, Min, and Average makes the recognition. The approach is tested on a standard speech database and is compared with Hidden Markov model recognition system with MFCC features as a widely used speech recognition approach.

Keywords: Fuzzy modeling; Speech recognition; Genetic algorithms

1. Introduction

Recognition of human speech is a problem with many solutions, but still open because none of the current methods are fast and precise enough to be comparable with recognition capability of human beings. Several methods exist for recognition of human phonemes such as Hidden Markov models [1], time delay neural networks [2], support vector classifiers with HMM [9], independent component analysis [16], HMM and neural-network hybrid [23], and more. A common problem of many speech-processing approaches is the very high sensitivity to noise or environmental effects, which results in the necessity of noise detector and eliminator algorithms and in turn, slow and complex speech recognizer systems.

There are two motivations to start a fuzzy-logic approach towards the problem of creating a fast and accurate speech recognizer: First, as we humans do [8], instead of detection and removal of the noise, a speech recognition system can just ignore it and fuzzy inference and pattern recognition is a very suitable tool for robust results in existence of high noise level. Second, as Zadeh states [26], we have experienced that more precise computations in cognitive tasks does not necessarily result in more precise results and it may even result in poorer answers and this might be due to the very fuzzy nature of cognitive problems.

Based on these ideas, a fuzzy-logic approach to cognitive tasks such as speech recognition seems to be very suitable and there are many contributions to fuzzy speech recognition. But in most of these approaches, fuzzy logic is just a high-level decision maker, which makes the final step of recognition or is just used as a means of integrating data from different sources while the system still works with conventional precise acoustic data of speech signals. For example, in [19,20], the inputs of a conventional neural network are replaced with fuzzy numbers and it is used on a standard speech database to recognize speech samples. As indicated before, the method does not make any specific fuzzy data conversion and has only used fuzzy techniques to make the neural network more robust versus small perturbations. Very much the same [18], use fuzzy numbers as the inputs of a dynamic programming approach to compare signal patterns with previously learned word patterns. Again, the only contribution of fuzzy is in making DP inputs robust to small changes. In [4,5], a general fuzzy integrator is used to combine the different evidences in an automatic speech

* Corresponding author. Tel.: +98 21 66164642; fax: +98 21 66019246.
E-mail addresses: halavati@ce.sharif.edu (R. Halavati), bagheri-s@sharif.edu (S.B. Shouraki), harati@ce.sharif.edu (S.H. Zadeh).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2006.02.007

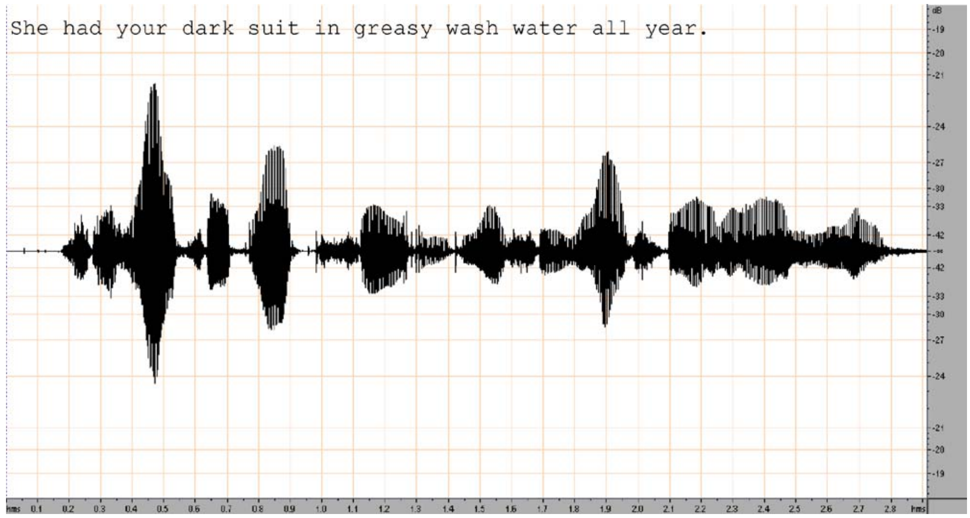

Fig. 1. Speech signal: normal view.

recognition process. As it is specified in those papers, they have used conventional speech features for decision-making and the only role of fuzzy is in combining the final decisions.

Also in [12], a general fuzzy pattern recognizer system tries to separate specific utterances of speech signal using standard acoustic parameters, again fuzzy is only used as the final pattern recognizer and the speech data is the same as other approaches [13,14,25,6,15,17] are other samples of neuro-fuzzy approaches to speech recognition, again using standard speech signal features and a neural network as the final decision maker. And at last [25,3], add fuzzy numbers to Hidden Markov model methods to make the input data immune to noise.

Hence, it can be concluded that despite the existence of numerous cited speech recognizers that use fuzzy logic in some part of their decision making, there is no approach that is designed based on fuzzy thinking and they can all be called fuzzifications of an existing crisp approach. To have a speech recognizer with fuzzy soul, this paper presents an approach to phoneme recognition which converts the speech spectrogram into a fuzzy linguistic description at the first step and uses this representation instead of conventional precise acoustic data during the training and recognition stages. To do so, different phonemes are represented with linguistic terms and the recognition is done by a simple operator using low cost functions such as maximum, minimum, and average.

The rest of this paper is organized as follows: In the next two sections, fuzzy representation of speech data and recognition approach are discussed, then, following the training algorithm details come the experimental results and comparisons and at last come the conclusions and future works.

2. Linguistic representation of speech spectrogram

Most speech processing systems use the spectrogram of speech signal as their initial source of data. The spectrogram of speech signal is a two-dimensional image which presents the amplitude of each available frequency at each time sample of the given voice signal. Fig. 1 presents a voice signal and Fig. 2 presents its spectrogram. As it is seen, in a normal representation of voice signal (Fig. 1), we just have the signal amplitude at each time frame (each sample). Similarly, the time axis of the spectrogram (Fig. 2), represents time, but the vertical axis represents the frequencies and the brightness of each point specifies the amplitude of that certain frequency at that time.

As the human auditory system recognizes the lower frequency signals with higher resolution in compare with higher frequencies, the vertical axis of the spectrogram is usually stretched so that it presents low frequencies with more precision and high frequencies with less. A standard method for this stretching is the Mel scale [24], which stretches the frequency based on an experimental diagram (presented in Fig. 3). After this stretching, the spectrogram of Fig. 2 is represented as in Fig. 4.

The source data for our model is the Mel-stretched spectrogram of the voice signal. The first step of both recognition and training processes is the conversion of this spectrogram into a fuzzy linguistic description. The fuzzification approach is based on three major ideas:

(1) A human recognizer does not read the spectrogram with full precision and pays attention only at local features such as "A short length of high amplitude around 6 kHz frequency" or "A long trend of high frequency starting from 6 kHz moving towards 4 kHz".
(2) A human does not decide based on precise speech amplitudes and only a rough measure is sufficient such as high or low amplitude.
(3) We do not count speech frames and use relative lengths such as a long or short.

Based on these ideas and to reduce the size of input data, we divided the frequency axis into 25 bands and made a local data reduction in each band[1]: Each column of the spectrogram in

[1] We chose 25 ranges just because Mel scaling has 25 ranges, but any other range count can also be used and it can be tested in further experiments.

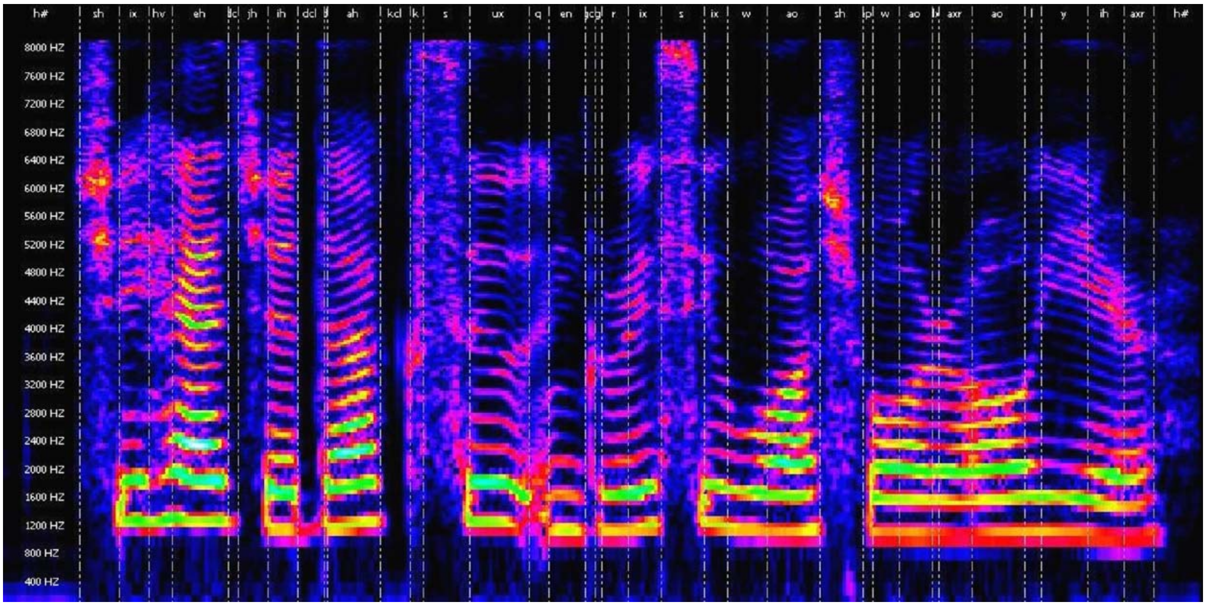

Fig. 2. Spectrogram of a speech signal, separated vertically with phoneme positions.

each band is considered as one block and will be represented with a rough measure of that block's high amplitude values. To choose this measure, we introduce a simple operator called *Max10*, which gives a rough maximum of a set of numbers.

In general, Max N is defined as an operator, which can have a range of effects from *maximum* to *average* of a set of numbers. To do so, Max N sorts the input numbers and takes the average of the highest N percent of values. Using Max N instead of normal maximum helps in more robustness versus unwanted high values due to noise. Max N is formally represented in Eq. (1), the data reduction process is depicted in Fig. 5 and the result over the entire spectrogram is presented in Fig. 6. Note that after this conversion, the image has only 25 values in each vertical line (frequency axis) but the time axis is not altered.

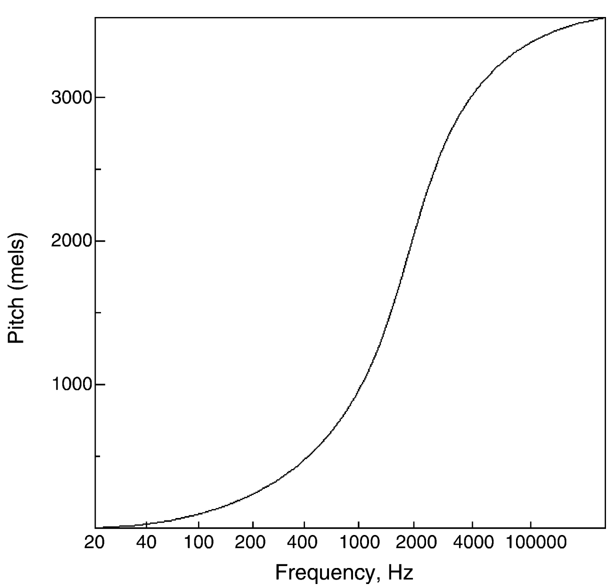

Fig. 3. The Mel scale diagram.

The entire input conversion process is represented in Fig. 7 diagram.

$$\begin{cases} V: \quad \text{a vector of } M \text{ numbers} \\ \mathrm{Sort}(V): \quad \text{a function that sorts } V \text{ so that } \mathrm{Sort}(V)_i \\ \qquad\qquad \text{will be the highest value of } V \text{ and } \mathrm{Sort}(V)_M \\ \qquad\qquad \text{will be the lowest} \\ \mathrm{Max}\, N(V) = \dfrac{\left(\sum_{(N/100)\times M}^{1} \mathrm{Sort}(V)_i\right)}{\left(\dfrac{N}{100}\times M\right)} \end{cases} \tag{1}$$

To represent a phoneme with linguistic terms using this data, a gradient of color names is used to represent different ranges of amplitudes which starts from black for lowest amplitude, then blue, magenta, red, yellow, green, cyan and finally fading to white showing the highest value. Each of these colors spans above a range of amplitudes and is described with a trapezoidal fuzzy membership function as in Fig. 8. Also, as different phonemes have different lengths (it is shown in Figs. 2, 4 and 6), five fuzzy lengths (very short, short, average, long and very long) are also defined with trapezoidal shapes to express phonemes' lengths. Our primary assumptions for the exact shape of color and length sets are presented in Table 1, but the values are later optimized using the training approach, which is presented in Section 4.

Using the above definitions, each phoneme can be described by an expression stating its length, and its probable colors for each band. Fig. 9 presents an example for phoneme definition. Note that a disjunction of different colors can be used in expressing each band's color.

Based on this type of phoneme modeling, a phoneme recognizer system must have the definitions for fuzzy colors (definition of trapezoids for each of eight colors), definition of

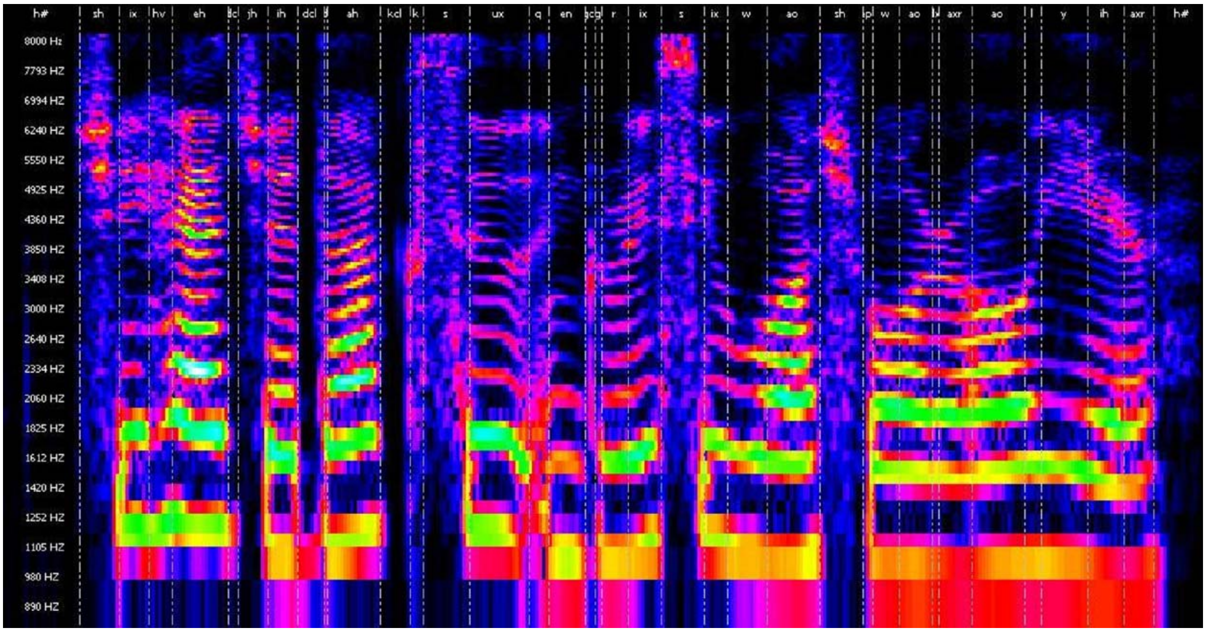

Fig. 4. Spectrogram of speech signal after Mel stretching.

fuzzy lengths, and description of all phonemes using the previously stated colors and lengths which includes a length value for each phoneme and color values for each frequency band of every phoneme.

To use it formally in the next section, we will express phoneme P with a set of 25×2 fuzzy color sets (two possible colors for each band) and a fuzzy set for its length as:

P.Color$_{i,j}$: for band colors with $i \in \{1, \ldots, 25\}$ and $j \in \{1, 2\}$ for 25 bands and two possible colors for each band.
P.Length: for phoneme length.

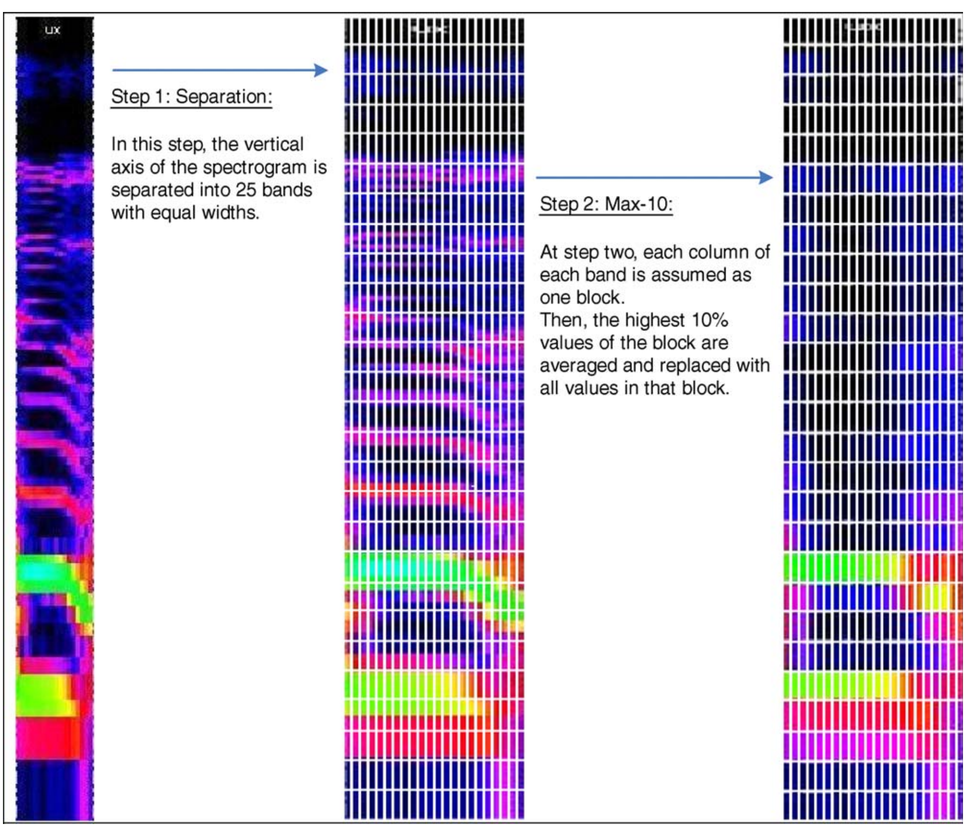

Fig. 5. A sample for data reduction process.

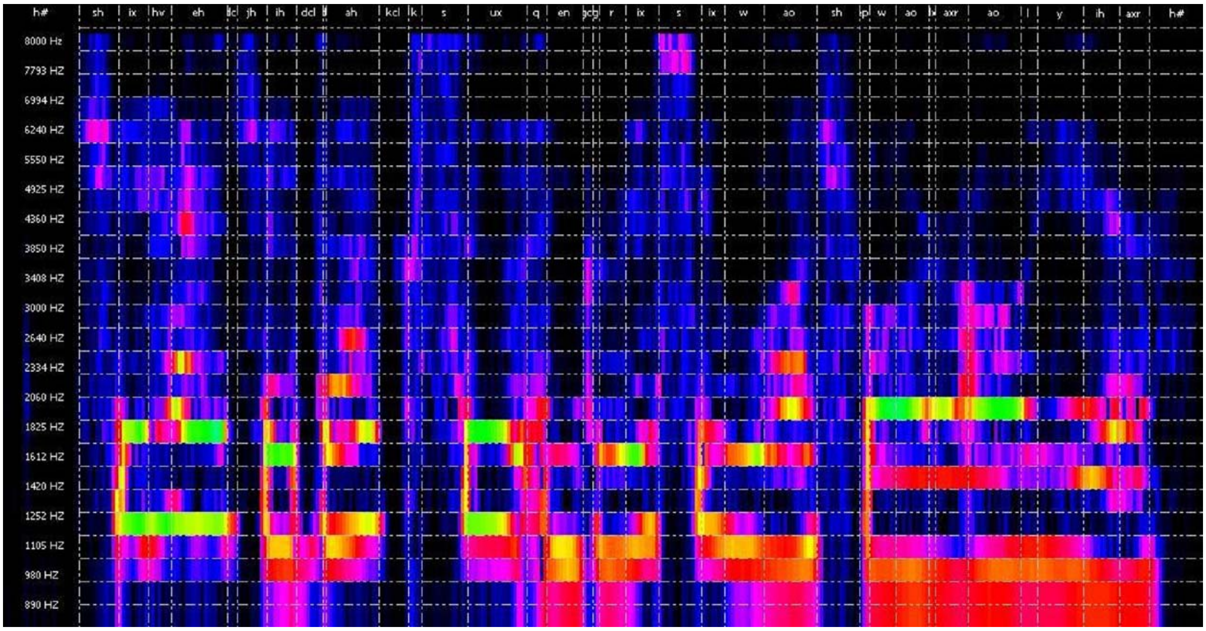

Fig. 6. Spectrogram after data reduction.

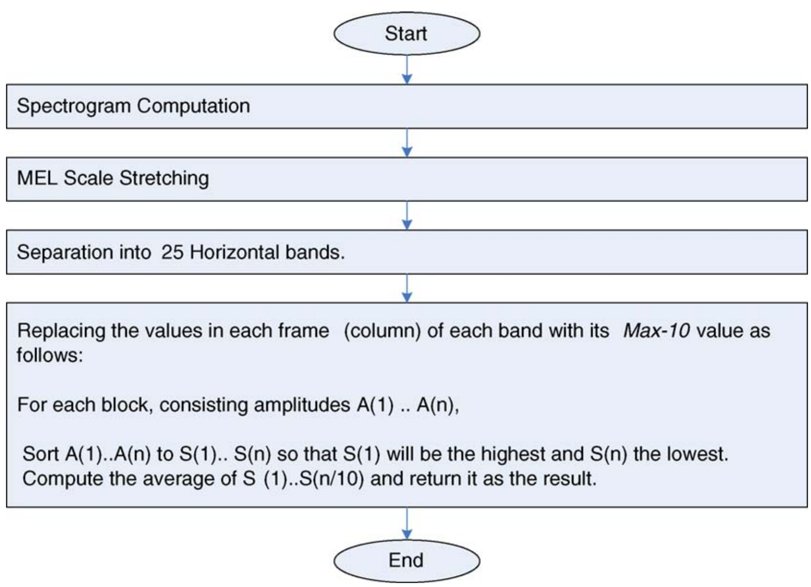

Fig. 7. Input conversion process.

3. The recognition process

Assuming that a suitable set of fuzzy colors, fuzzy lengths and phoneme descriptors exist, this part presents the recognition approach which classifies given voice inputs into phoneme classes. To categorize the given input, the degree of compatibility of the input to each of the phoneme descriptions is computed and the input is classified into the phoneme class, which has the highest compatibility value.

Thus, at the beginning of the comparison stage, we have an input and several color patterns, each pattern for one phoneme. To compare the input with each of the color patterns, we have $25 \times N$ values for the input (25 rows, each for one band and N

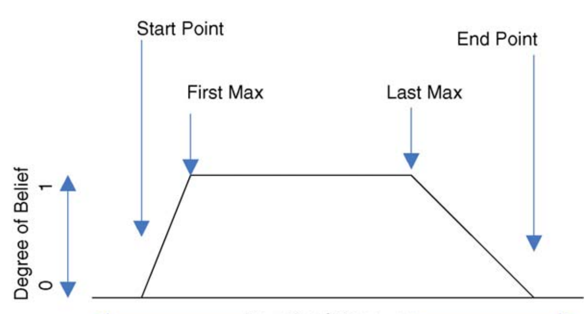

Fig. 8. Trapezoidal fuzzy set definitions for colors and lengths.

Table 1
Initiate definition of colors and lengths

	Start point	First max	Last max	End point
Black color	0	0	0.05	0.35
Blue color	0	0.05	0.22	0.50
Purple color	0	0.23	0.35	0.63
Red color	0.05	0.36	0.50	0.80
Yellow color	0.23	0.51	0.63	0.93
Green color	0.36	0.63	0.80	0.99
Cyan color	0.51	0.81	0.93	0.99
White color	0.63	0.94	0.99	0.99
Very short length	2	2	3	5
Short length	2	2	6	10
Medium length	2	6	12	17
Long length	8	12	20	30
Very long length	12	18	69	99

columns, as the input patterns my have arbitrary number of frames) and a color pattern that has 25×2 colors (25 rows, each for one band and two colors for each band because a disjunction of two colors is allowed). The comparison of these two items is done in three steps:

3.1. Step 1: single frame pattern matching

The first step is to compute how much each frame of the given input belongs to the specified color pattern of the phoneme. Thus, if the input signal has N frames, the compatibility of each frame to current phoneme's describing colors is computed independent of the other $N-1$ frames. To compute this compatibility, a *Min of Max* approach is used: First, each input value in each band is compared with the disjunctive colors describing that band and the maximum value is selected, then the minimum of these values for all bands is chosen and regarded as the frame's compatibility with the pattern. Assuming vector V as input and P as the pattern, Eq. (2) formally represents the similarity computation formula and a sample of this task is presented in Fig. 10.

$$\text{Similarity} = \underset{i=1,\ldots,25}{\text{Min}}\left(\underset{j=1,2}{\text{Max}}(\text{P.Color}_{i,j}(V_i))\right) \quad (2)$$

3.2. Step 2: color compatibility aggregation

Once the compatibility of each frame with the color pattern is computed, the results on all frames of the given sample must be aggregated. To make the aggregation, we must get a general measure of all frames similarity to the required pattern. If we just use the minimum or maximum operators, we highly increase the risk of noise sensitivity as a single good or bad sample can affect the decision. Thus, we have defined another operator similar to Max N called Average N, which computes the average of $N\%$ of middle values in a set of numbers.

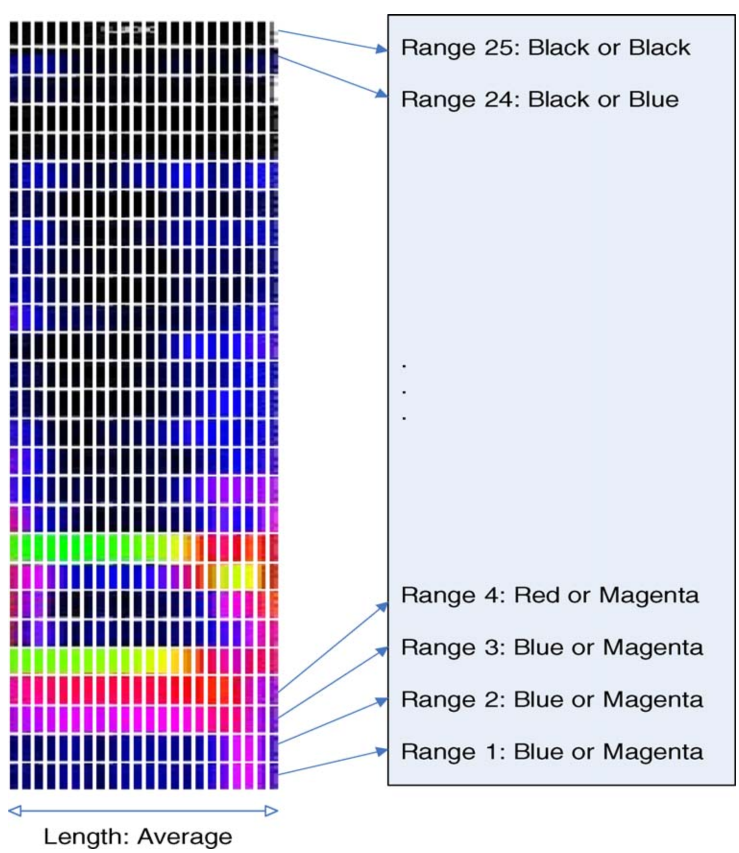

Fig. 9. A sample description for a phoneme. For each band, two most promising colors represent the band as described in training algorithm.

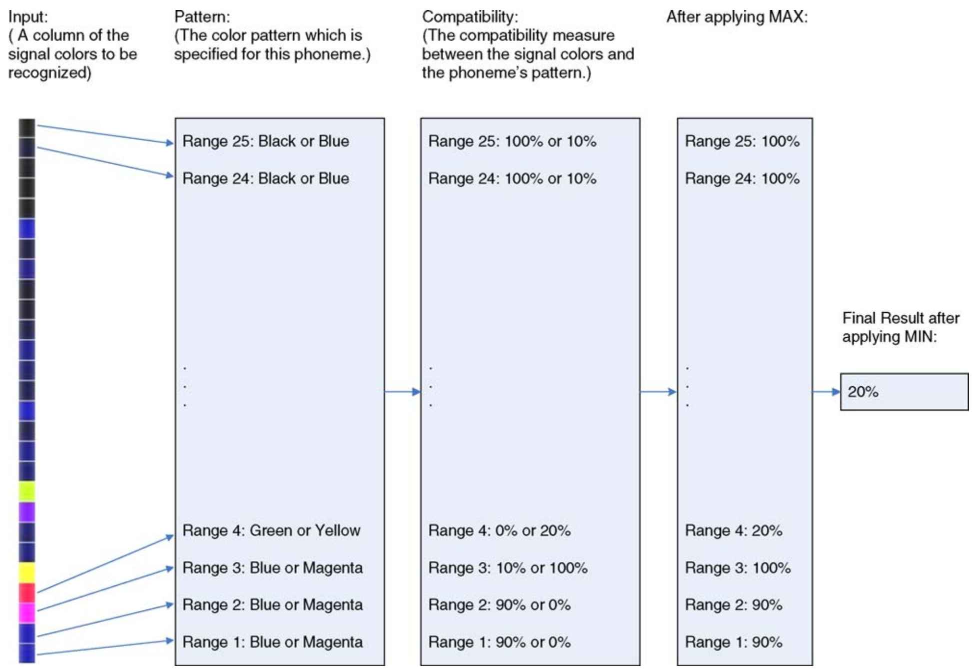

Fig. 10. Sample of *Min of Max* computation of step 1.

As expressed in Eq. (3), the operator sorts the given numbers, excludes the lower and higher $(100 - N/2)\%$ of numbers and averages the rest.

$$\begin{cases} V: & \text{a vector of } M \text{ numbers} \\ \mathrm{Sort}(V): & \text{a function that sorts } V \text{ so that } \mathrm{Sort}(V)_i \\ & \quad \text{will be the highest value of } V \text{ and } \mathrm{Sort}(V)_M \\ & \quad \text{will be the lowest} \\ X = \dfrac{100 - N}{2} \times M & \\ \text{Average } N(V) = \dfrac{\left(\sum_{M-X}^{i=X} \mathrm{Sort}(V)_i\right) \leftrightarrows}{M - X \times 2} & \end{cases} \tag{3}$$

Using this operator, we apply *Average80* on results of previous step to prune 10% of the lower and higher results and average the reset as the final decision. Fig. 11 shows a sample of this step.

3.3. Step 3: length matching

The input of the second step is the compatibility measure of the sample with the specified pattern of colors. In this part, the sample's length (number of frames) is compared with the fuzzy linguistic term specifying the phoneme length and its compatibility is multiplied by the color compatibility of the input to make the final compatibility value. Fig. 12 shows the overview of the three steps together.

4. Training algorithm

In the previous two sections, our fuzzy modeling of the speech signal and our way of comparing the input data and the linguistic description of phonemes were presented. But we need an appropriate set of definitions to perform the recognition task. This section presents the training algorithm we have used to define phoneme patterns. To do so, we have used a conventional genetic algorithm [10], which will be briefly described in the next subsections.

4.1. The genome and its fitness

The first necessity for a genetic algorithm approach is to present a complete recognizer in the form of a genome. Fig. 13 shows the structure of our genomes which includes the definition of all eight colors, definition of the five length sets and description of all phonemes.

Genome's fitness is defined as a general measure on how correct it can recognize all training samples. To compute it, all training samples are categorized based on the definitions in that genome and the genome receives its fitness based on how many of these categorizations have been correct. The fitness can be computed using Eq. (4) formulas.

$$\begin{cases} \mathrm{G}: & \text{A genome} \\ \mathrm{S}: & \text{The set of training samples} \\ \mathrm{CorrectPhoneme}(s): & \text{The correct phoneme class of sample } s \\ \mathrm{Rec}(G, s): & \text{The phoneme class that a recognizer} \\ & \quad \text{which is built based on genome } G, \\ & \quad \text{classifies the input } s \text{ into} \\ \mathrm{CorrectSet}(G, S) = & \{s \mid s \in S \text{ and } \mathrm{Rec}(G, s) \\ & \quad = \mathrm{CorrectPhoneme}(s)\} \\ \mathrm{Fitness}(G) = & \dfrac{|\mathrm{CorrectSet}(G, S)|}{|S|} \end{cases} \tag{4}$$

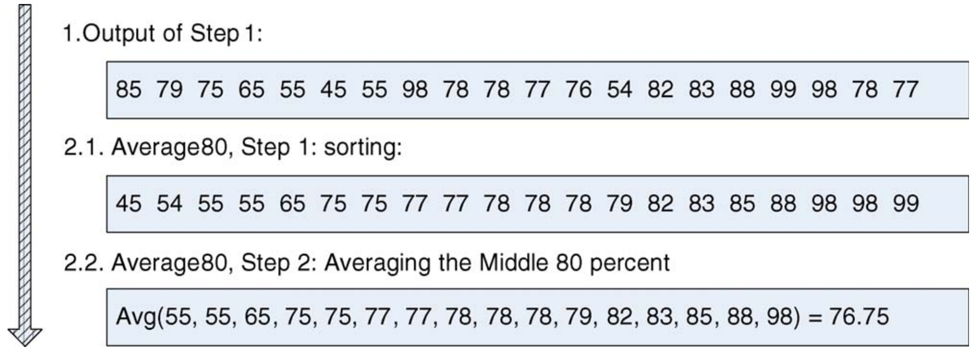

Fig. 11. Total frames' color matching at step 2 of recognition.

4.2. Mutation operator

The mutation operator receives a genome as its input and creates a new genome by duplicating the previous one with minor changes. To do so, we copy a genome and randomly change one of its parts as follows:

If the selected part is the definition of a color or a length, one of its four points is changed, randomly by +0.01 or −0.01 and then, if the four points would not be in ascending order (due to mutation), they are resorted. Fig. 14 presents the possible mutations for a sample color definition.

And if the selected part is a phoneme definition, one of its colors or its length can alter to a close linguistic value, for example, if it is defined that phoneme X has color red at its first band, the definition may change to colors purple or yellow at that band.

4.3. Cross over operator

The cross over operator receives two genomes and creates two new genomes by randomly choosing different parts from each genome. In other words, assuming the two original genomes as parents and the new genomes as children, the definition of each of the colors for each of the children is randomly taken from one of the parents and the same thing happens for length and phoneme definitions.

4.4. Control process of training

The training process starts by randomly creating 100 genomes which all have color and length definitions as specified in Table 1, but their phoneme definitions are set

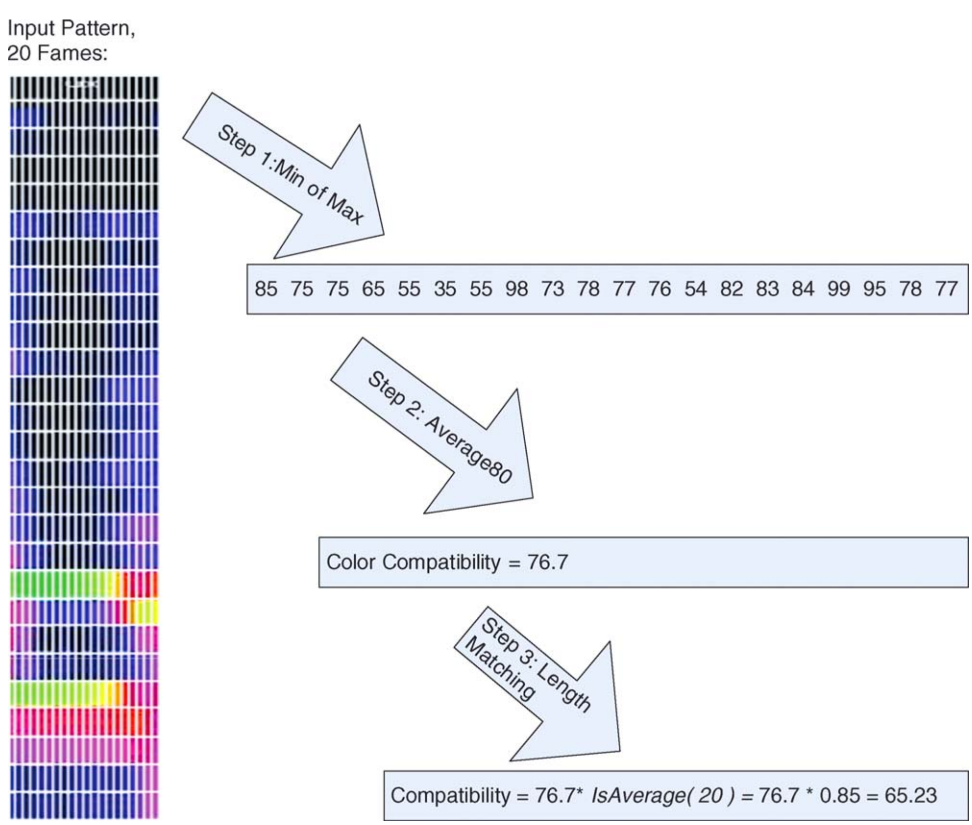

Fig. 12. Overview of recognition step.

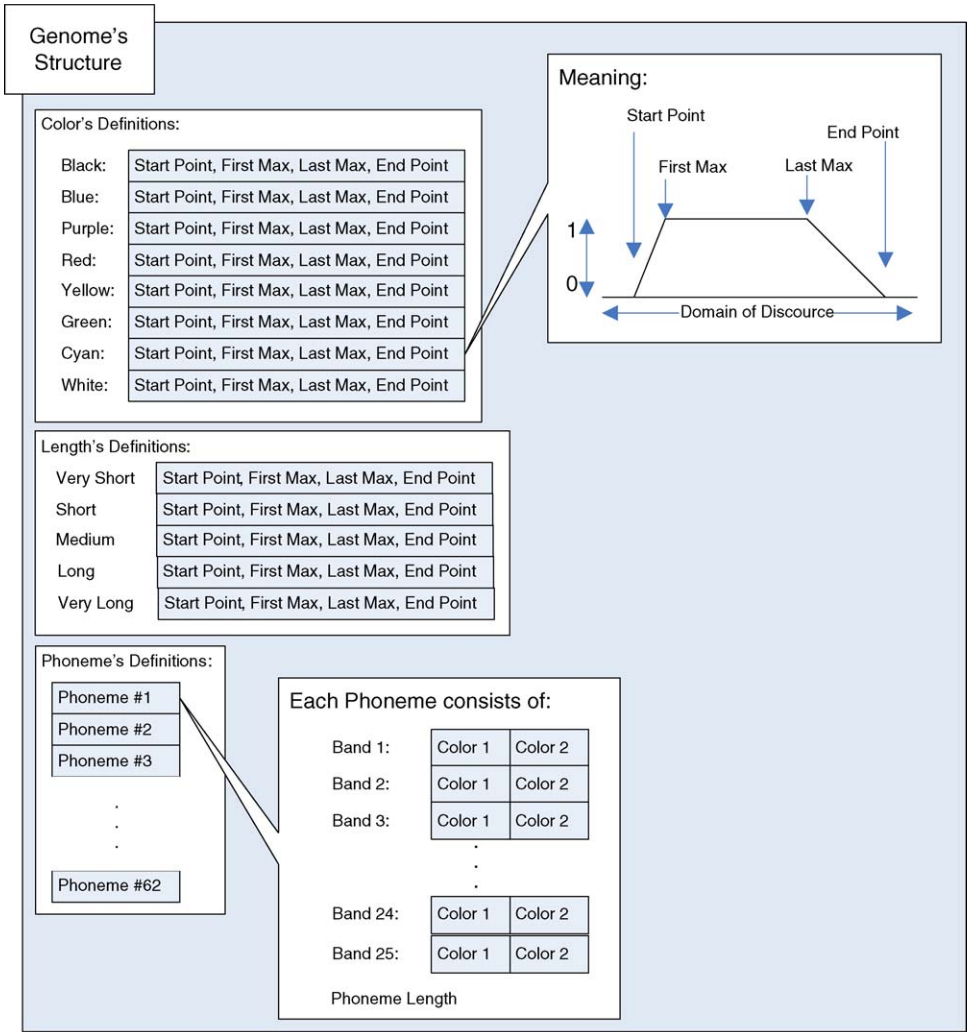

Fig. 13. Genome's structure.

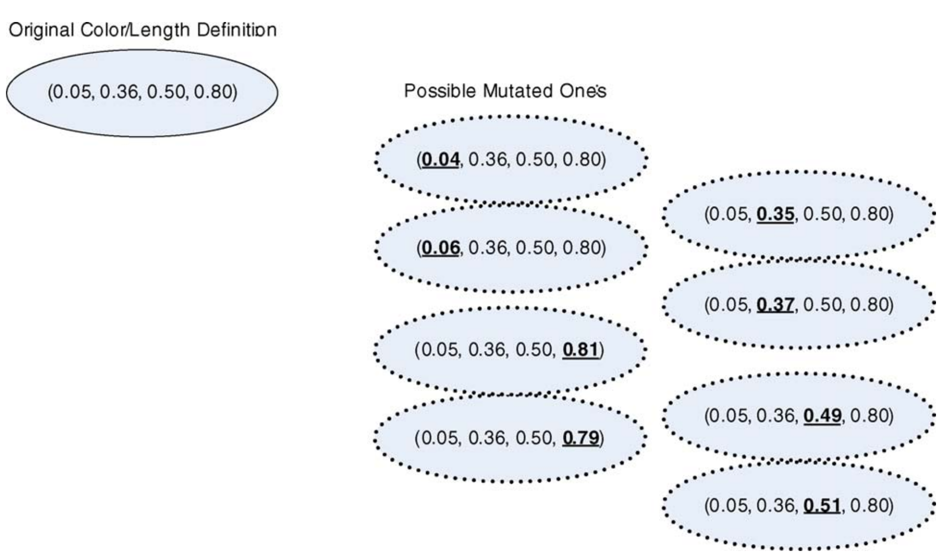

Fig. 14. Possible mutations for a sample color/length definition.

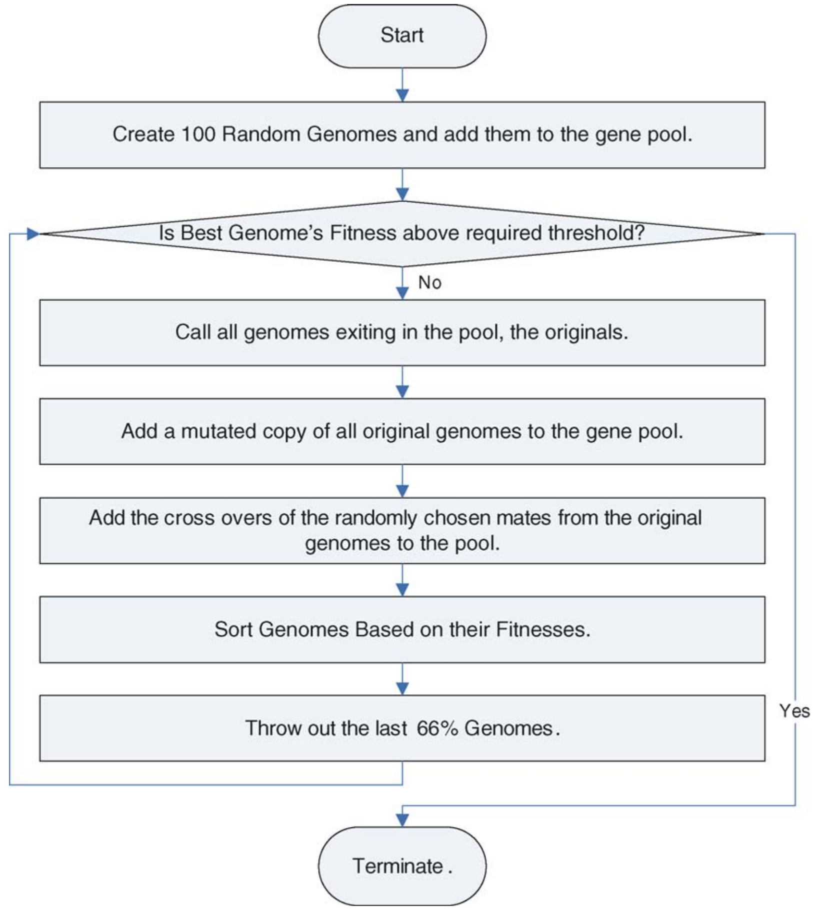

Chart 1. Training diagram.

randomly. Assuming a minimum required fitness to stop the optimization, the rest of the process is run for as many iterations as necessary.

In every iteration each existing genome creates a new genome using the mutation operator. Also each genome randomly pairs with another genome and using the cross over operator, they together create two new genomes and add them to the gene pool. By this point, we have 300 genomes, where 200 are new and created from the old 100 one. Then, the genomes are sorted based on their fitness values and 200 of the genomes which have least fitness values are dropped off, and then, another iteration starts.

Therefore, using crossover and mutation operators, we explore the space of possible genomes and pruning based on fitness guides the search towards areas with better genome. The training process is entirely presented in Chart 1 and the results are represented in the next section.

It must be noted that, at the beginning of every iteration, each genome will have one and exactly one mutation with 100% probability, and also pairs with another genome and performs a cross over, again with 100% probability. Therefore, we can assume a mutation rate equal to (1/size of genome) which becomes $((1/(8+5)\times 4+62\times(1+2\times 25))=0.03\%)$ and a cross over rate equal to 100%

5. Experimental results

The proposed algorithm is tested on Timit[2] speech database with 62 phoneme classes. The benchmark system is an HMM based isolated phoneme recognition system with MFCC features [1,7,21,22]. Training and testing was performed separately for single speaker and multiple speaker cases. For single speaker tests, we chose 20 randomly selected speakers from Timit and for multiple speaker tests, we extracted 20 sets, each including samples of five randomly chosen female speakers and five male speakers. In all tests, training was performed with 90% of samples for each set and testing with the remaining 10% of the same set. Each test is repeated five times and with six different noise levels, namely clean, 30, 20, 10, 0 and −10 dB. Fig. 15 presents the average recognition rates in tests on clean data and Fig. 16 presents the recognition performance in existence of different noise levels.

As it is common in speech recognition, besides reporting the recognition rate for cases that the proposed system has correctly

[2] The DARPA TIMIT speech database is produced to provide acoustic phonetic speech data for the development and evaluation of automatic speech recognition systems. It consists of utterances of 630 speakers that represent the major dialects of American English.

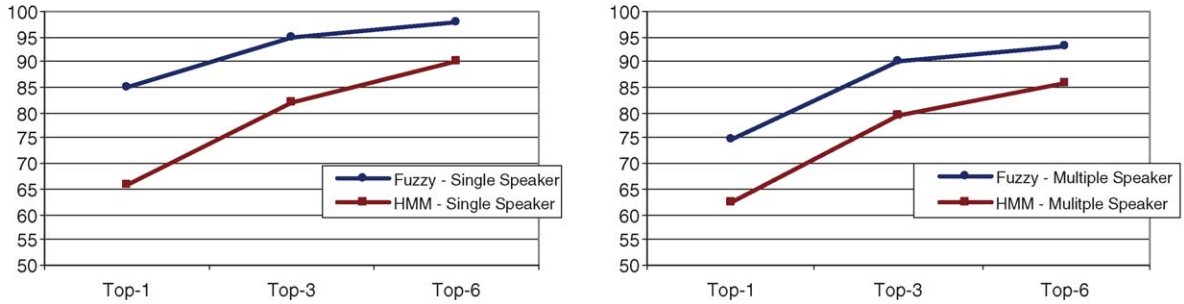

Fig. 15. Recognition rate comparisons of the proposed algorithm on single speaker and multiple speaker test sets and its comparison with MFCC–HMM benchmark[4].

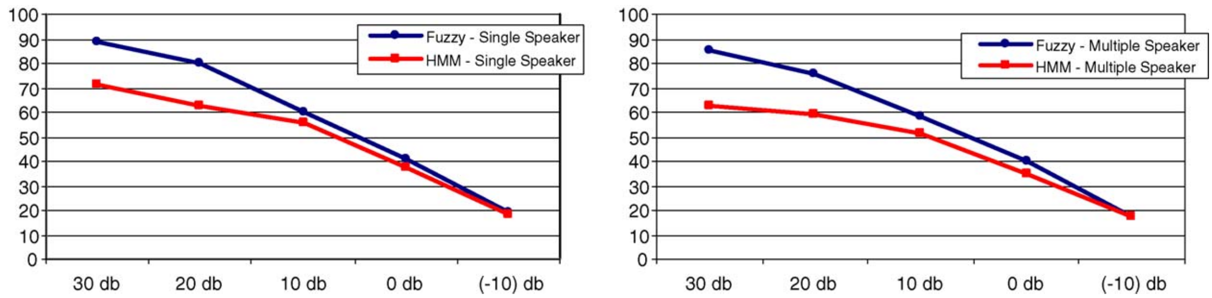

Fig. 16. Recognition performance in noisy environments. Vertical axis: recognition rate; horizontal axis: signal to noise ratio (SNR) of additive white noise.

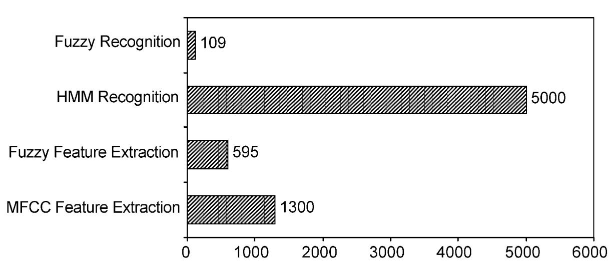

Fig. 17. Recognition speed for 10 min of segmented samples. Horizontal axis: processing time (ms).

classified the given phoneme, we have reported the cases where the correct answer has been among the top three scores of the algorithm and also the top six.[3] To do so, instead of just choosing the class with highest similarity to the input, we have chosen the classes, which had the top three or six similarities. Training in all cases was continued until the progress rate of best fitness dropped below 0.1% in 1000 iterations.

As depicted in Fig. 15, for single speaker tests, our method has been able to identify the phonemes correctly in 85% of cases while the *Top-3* recognition rate was 95% and *Top-6* has been 98% and these values are 20–10% lower for the benchmark HMM–MFCC system. Multiple speaker tests have also shown a similar advantage in compare with HMM–MFCC results.

Moreover, as expected from the fuzzy nature of the system and depicted in Fig. 16, this approach has shown 20–28% more immunity against noise in normal noisy environment (SNR: 30–10 dB) and this immunity decreases while the noise level approaches the amounts that makes the input not identifiable for human beings but it is always above that of HMM–MFCC.

Also, to have a measure of computational complexity of our approach in compare with MFCC–HMM feature extraction and recognition, we ran a test to recognize 10 min of segmented samples by both systems. As depicted in Fig. 17, the proposed approach has been more than two times faster in feature extraction and 50 times faster in recognition.

From the above results, it can be concluded that the proposed fuzzy model has been quite successful in correctly identifying human speech phonemes and this capability has been achieved with much less computational effort in comparison with traditional methods such as Hidden Markov models. Also, the system has been able to quite successfully ignore noise in noise levels, which are ignorable to human recognizers as its results are much better than a current popular method.

6. Conclusion and future work

Despite the existence of several methods for speech recognition, the problem is still open as no algorithm is both fast and accurate enough to be an ultimate answer for recognition of human speech in industrial and commercial applications. To have a fast and noise robust speech recognizer system, this paper presents a fuzzy modeling of speech signal

[3] This is done because in most speech recognition applications, we have a predefined set of words to recognize (called the dictionary) and we can use the combined probabilities of recognized phonemes to make the final decision. Thus, making a hard decision on the phoneme level is neither necessary, nor advisable.

[4] Top-N Correct Answer: The recognition rate where the correct answer is among the top N scores of the algorithm.

by representation of speech spectrogram with linguistic terms. The major difference with previous not fuzzy approaches is in ignorance of details that make the recognition slow and sensitive to small perturbations or noise. And the major difference with other existing fuzzy models is in the fact that this model does not use conventional acoustic features of voice signal and is totally based on fuzzy-thinking, using new features of speech signal which are defined by rough linguistic terms, much a like how a human can read and translate a speech spectrogram into phoneme classes.

The approach is tested over a standard database for single speaker and multiple speaker phoneme recognition in both noise-free and noisy environments. The algorithm has proved to be quite successful in mapping samples to phonemes and also has successfully *ignored* noise instead of its detection and removal. The comparison results with a widely used approach (Hidden Markov model with MFCC feature set) are also presented and while the proposed method is much simpler and uses much less computational power, it has gained significant better results in dealing with noise and recognition in noisy environment.

As a next step to improve the algorithm, we are working on fuzzy models for word identification and also other approaches to translate the spectrogram into a linguistic description, so that to decrease the amount of data and increase noise immunity.

References

[1] B. Babaali, H. Sameti, The sharif speaker-independent large vocabulary speech recognition system, in: Proceedings of the 2nd Workshop on Information Technology and Its Disciplines, Kish Island, Iran, 2004.
[2] M.R. Berthold, A time delay radial basis function network for phoneme recognition, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 7, Orlando, 1994. pp. 4470–4473.
[3] A.D. Cheok, S. Chevalier, M. Kaynak, K. Sengupta, K. Chung, Use of a novel generalized fuzzy hidden Markov model for speech recognition, in: Proceedings of the IEEE International Conference on Fuzzy Systems, vol. 3, 2002, pp. 1207–1210.
[4] S. Chang, S. Greenberg, Application of fuzzy-integration-based multiple-information aggregation in automatic speech recognition, in: Proceedings of the IEEE Conference on Fuzzy Information Processing, Beijing, 2003.
[5] S. Chang, S. Greenberg, Syllable-proximity evaluation in automatic speech recognition using fuzzy measures and a fuzzy integral, in: Proceedings of the 12th IEEE International Conference on Fuzzy Systems, vol. 2, 2003, pp. 828–833.
[6] D.D. Doye, U.V. Kulkarni, T.R. Sontakke, Speech recognition using modified fuzzy hypersphere neural network, in: Proceedings of the International Joint Conference on Neural Networks, vol. 1, 2002, pp. 65–68.
[7] J. Duchateau, K. Demuynck, D.V. Compernolle, Fast and accurate acoustic modeling with semi-continuous HMMs, Speech Commun. 24 (1) (1998) 5–17.
[8] H.L. Dreyfus, What computers still can't do, in: A Critique of Artificial Reason, MIT Press, 1972.
[9] S.E. Golowich, D.X. Sun, A support vector/Hidden Markov model approach to phoneme recognition, in: Proceedings of the ASA Statistical Computing Section, 1998, pp. 125–130.
[10] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[12] A. Juneja, C. Espy-Wilson, An event-based acoustic-phonetic approach for speech segmentation and E-set recognition, in: Proceedings of the International Conference on Phonetic Sciences, Barcelona, 2003.
[13] N. Kasabov, R. Kozma, M. Watts, Phoneme-based speech recognition via fuzzy neural networks modeling and learning, Inform. Sci. 110 (1998) 61–79.
[14] N. Kasabov, G. Iliev, Hybrid system for robust recognition of noisy speech based on evolving fuzzy neural networks and adaptive filtering, in: Proceedings of the International Joint Conference on Neural Networks, vol. 5, 2000, pp. 91–96.
[15] S.S. Kim, D.J. Lee, K.C. Kwak, J.H. Park, J.W. Ryu, Speech recognition using integra-normalizer and neuro-fuzzy method, conference record of the asilomar conference on signals, Syst. Comput. 2 (2000) 1498–1501.
[16] O.W. Kwona, T.W. Lee, Phoneme recognition using ICA-based feature extraction and transformation, Signal Process. 84 (6) (2004) 1005–1019.
[17] K.F. Leung, F.H.F. Leung, H.K. Lam, P.K.S. Tam, Recognition of speech commands using a modified neural fuzzy network and an improved GA, in: Proceedings of the IEEE International Conference on Fuzzy Systems, vol. 1, 2003, pp. 190–195.
[18] P. Mills, J. Bowls, Fuzzy logic enhanced symmetric dynamic programming for speech recognition, in: Proceedings of the 5th IEEE International Conference on Fuzzy Systems, 1996.
[19] P.A. Nava, Neuro-fuzzy system for speech recognition, in: Proceedings of the International Congress on Electronics and Electrical Engineering, vol. 20, Chihuahua, 1998. pp. 503–507.
[20] P.A. Nava, J.M. Taylor, Speaker independent voice recognition with a fuzzy neural network, in: Proceedings of the 5th IEEE International Conference on Fuzzy Systems, vol. 3, 1996, pp. 2049–2052.
[21] Yuichi Ohkawa, Akihiro Yoshida, Motoyuki Suzuki, Akinori Ito, Shozo Makino, An optimized multi-duration HMM for spontaneous speech recognition, in: Proceedings of the EUROSPEECH-2003, 2003, pp. 485–488.
[22] L.R. Rabiner, A tutorial on Hidden Markov models and selected applications in speech recognition, IEEE Proc. 77 (2) (1989) 257–286.
[23] P. Schwarz, M. Cernocky, J. Cernocky, Phoneme recognition based on TRAPs, in: Proceedings of the Workshop on Multimodal Interaction and Related Machine Learning Algorithms, 2004.
[24] S.S. Stevens, J. Volksman, The relation of pitch to frequency, Am. J. Psychol. 53 (1940) 329.
[25] D. Tran, M. Wagner, Fuzzy hidden Markov models for speech and speaker recognition, in: Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS, 1999, pp. 426–430.
[26] L.A. Zadeh, From computing with numbers, to computing with words, a new paradigm, Int. J. Appl. Math. 12 (3) (2002) 307–324.