Eng. & Technology, Vol.24, No.8, 2005

Speaker Recognition System Using Cellular Automata

Aladdin J.Abdulwahid*   Ahmed A. Al-Attab,*   Moaid A. Fadhil,*

Received on: 24/12/2002
Accepted on:28/6/2005

Abstract

Speaker recognition and voice recognition are subfields of speech processing by computer. They work on the principle that there are features of speech that can be used to discriminate one speaker from another through three stages, preprocessing, feature extraction and classification.

In preprocessing stage average magnitude and zero crossing rate were used to detect start and endpoint of the speech. In feature extraction stage average pitch and 12-linear prediction coefficient were used to represent the important characteristics of the speech. In classification stage most methods used in patterns recognition perform some kind of comparison of a feature-vector with some reference-vector. No things like that is happening here, a new approach is presented based on a set of uniform cellular automata (CA).

Computation in CA has been studied from different perspectives and has been constructed for various specific computation tasks as far as it is shown capable of universal computation. So the main object of this research is to discover the capability of the cellular automata of performing one-dimensional and two-dimensional pattern classification when these patterns are feature vectors of speech.

Genetic algorithm was used also with cellular automata as an evolutionary supervised learning algorithms for implementing this task.

Keywords: Speech recognition cellular, genetic algorithm,

نظام تمييز المتكلم باستخدام الماكينة الخلوية

الخلاصة

معالجة الكلام باستخدام الحاسب الآلي من المجالات المهمة، والذي يتضمن عدد من الحقـــول الفرعيـــة منها تمييز المتكلم وتمييز صوت المتكلم وفقاً لمبدأ وجود خصائص هامة في صوت الإنسان تختلف من شخص لآخر يعتمد عليها في التمييز خلال المراحل التالية:

أولاً المعالجة الابتدائية حيث يتم تحديد بداية ونهاية الكلام، وفي هذا البحث تم استخدام معدل السعة مع درجة التقاطع الصفري لإنجاز هذه المرحلة ثانياً استخلاص السمات الهامة وقد وجدنا أن معدل النغمـــة الأساسية مع 12 من معاملات التنبؤ الخطي كافية للتمثيل السمات الهامة للمتكلم. أخيراً مرحلة التصنيف حيث في معظم الطرق العامة تعتمد على أجراء مقارنة بين المتجهات المرجعية مع المستخلصة. في هذا البحث تم تقديم طريقة جديدة بأستخدام الماكينة الخلوية المتماثلة.

الماكينة الخلوية درست مسبقاً من وجهات نظر مختلفة كما تم بناءها أيضاً لمختلف المهـــام الحـــسابية الخاصة لذا كان هدف هذا البحث هو الكشف عن كفاءة هذه التقنية في عملية تصنيف عدد من الأنماط أحادية وثنائية البعد إذا كانت هذه الأنماط هي متجهات الخواص للكلام.

* Dept. of Computer Sciences, UOT., Baghdad, IRAQ.

https://doi.org/10.30684/etj.24.8A.6
2412-0758/University of Technology-Iraq, Baghdad, Iraq

الخوارزميات الجينية أستخدمت أيضا مع الماكينة الخلوية للحصول على القواعد المناسبة لأجراء هــذه
العملية وكخوارزمية تعليمية مراقبة ومتطورة.

Introduction

Speech processing by computer is a deep and growing area of research. It is encompassing electrical engineering, computer science, linguistics, speech communi-cation, and telecommu-nications, among others. There are four distinct subfields of speech processing: Speech synthesis, speech recognition, speaker recognition and voice recognition[9].

Speaker recognition is concerned with extracting information about individuals from their speech in order to determine or validate their identity [16]. From speech alone good guesses can be made as to whether the speaker is male or female, adult or child. A person's mood, emotional state and attitude; anger, fear, belligerence, sadness, reluctance and elation may all be detectable in the speech signal.

The most heavily investigated sub-area of speaker recognition is complementary to speech recognition where both techniques use similar methods of speech signal processing [16]. Humans are adept at speaker recognition. A human can identify familiar speakers on the telephone after listening to a very short segment of speech.

The range of sounds that can be produced by a human being is related to the physical size and shape of the speaker's vocal tract. The elasticity of the tissue in the vocal tract also affects the sounds that are produced by an individual. With so many physical parameters contribu-ting to the range of sounds that each individual can make [16,19], there is reason to believe that a person can be uniquely identified by voice alone.

2. Speaker Verification Versus Speaker Identification

Speaker recognition divided into speaker verification and speaker identification, speaker verify-cation is determining whether a speaker is who claims to be, for example, to gain entry to a secure area. Verification systems must deal with two kinds of errors: false rejection and false approval. Speaker identification is the process of determining which speaker, if any, in a group of known speakers, closely matches an unknown speaker. The identification may be closed set, where it is assumed that the unknown is in the set of known speakers; or open set, where the unknown speaker may or may not be in the set of known [9,15].

3. Speaker Recognition Structure and Fundamental Task

The whole procedure for speaker recognition can be modeled by the following three modules as Figure (1) [2].

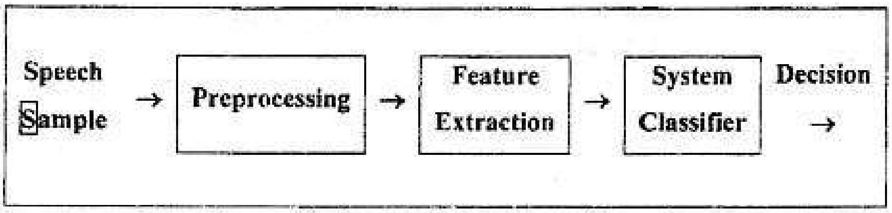


Figure (1): Speaker Enrolment Procedure

3.1 Preprocessing

The first module in speaker recognition will be the preprocessing of the input speech data. In this module, the input speech data used either for training the system or for the testing procedure are imposed to certain signal processing algorithms. This preprocessing module will include endpoint detection, segmentation and windowing.

3.1.1 Endpoint Detection

Detecting when a speech utterance begins and ends is a basic problem in speech processing. This is often referred to as endpoint detection [11]. In a speaker recognition system, it is a fundamental task to detect the endpoints of speech signals. In other words, the system must detect and remove the non-voiced parts in the user's recorded utterance. The main motivation behind endpoint detection is that the processing of these non-voiced parts is bound to undermine the performance of our system. Moreover, we reduce the total processing time of the system by removing these unwanted parts [14].

There are certain metrics that are normally used to identify a spoken utterance from the background noise present in a recorded utterance as following.

3.1.1.1 Short-Time Energy and Average Magnitude

The amplitude of a speech signal varies with time. In particular, the amplitude of unvoiced segments is generally much lower than the amplitude of voiced segments. Therefore, the short-time energy of the speech signal provides a convenient representation that reflects amplitude variations.

$$En = \sum_{m=n-N+1}^{n} \left[x(m)^{*} w(n-m)\right]^{2} \qquad \ldots(1)$$

where x(m) is the amplitude and w(n) is window function. From this equation we define an average magnitude function.

$$Mn = \sum_{m=n-N+1}^{n} \left|x(m)\right|^{*} w(n-m) \qquad \ldots(2)$$

where the weighted sum of absolute values of the signal is computed instead of the sum of squares [15].

3.1.1.2 Short-Time Average Zero-Crossing Rate

Zero-crossing rate is a measure of the number of times in a given time interval that the amplitude of the speech signal passes through a value of zero because of its random nature. The definition is:

$$Z_n = \sum \frac{1}{2}\left|sign[x(n)] - sign[x(n-1)]\right| \ldots(3)$$

where

$\mathrm{sign}[x(n)] = 1 \qquad x(n) \geq 0$

$\mathrm{sign}[x(n)] = -1 \qquad x(n) < 0$

The computation of Zn requires to check samples in pairs to determine where the zero-crossing occurs and then the average is computed over N consecutive samples. If the zero-crossing rate is high, the speech signal is unvoiced, while if the zero crossing rate is low, the speech signal is voiced [11,15].

We can see in Figure (2) that this algorithm performs accurate endpoint detection.

3.1.2 Segmentation and Windowing

The segmentation and windowing of speech signals is segmented into frames of constant length M that overlap each other. Each of these frames is then multiplied by window W(n), which yields a set of speech samples weighted by the shape of the window [14]. The most widely used windows in speech analysis is hamming window. The following equation defines the hamming window shown in Figure (3).

$$W(n)=0.54-0.46\cos(2\pi^{*}n/N-1) \quad ; 0 \leq n \leq N-1 \qquad \ldots(4)$$
$$= 0 \quad ; elsewhere$$

In this equation, n equals the sample index: 0, 1, 2, and so on up to N-1.[8,14,15].

3.2 Features Extraction

Before identifying any voices or training a person to be identified by system, the speech signal must be processed to extract important characteristics of speech. By using only the important speech characteristics, the amount of data used for comparisons is greatly reduced and thus, less copulation and less time is needed for comparisons or for training.

The acoustic speech wave has information about the speaker dependent parameters such as vocal tract length, vocal tract shape, vibration of vocal cords ..etc. There are several techniques which exist to extract parameters in time domain or frequency domain analysis [3,9]. According to previous extensive research, the following acoustic parameters have been found useful for speaker recognition [9].

3.2.1 Linear Predictive Coefficients (LPC)

One of the most powerful speech analysis techniques is the method of linear predictive analysis. This method has been one of the basic speech processing techniques over the past years, used to estimate the basic speech parameters [15]. The basic concept behind linear predictive analysis is that a speech sample can be approximated by a linear combination of past speech samples [8,14,15].

The main equation for LPC is:

$$\sum_{i=1}^{p} a_i \sum_{n} x(n-j)x(n-i) = \sum_{n} x(n)x(n-j), j=1,2,\ldots,p \qquad \ldots(5)$$

This is a set of plinear equations for p unknowns $a_1 \ldots a_p$, solving it is equivalent to inverting a p* p matrix [2, 11, 47, 19].

There are several methods evident in [15] which are used to solve the linear prediction equation such as autocorrelation method and covariance method.

It is possible to simplify the above equation in terms of the autocorrelation function as:

$$R(m)=\sum_{n} x(n)x(n+m) \qquad \ldots(6)$$

this give the matrix

$$\begin{bmatrix} R(O) & R(1) & \ldots & R(p-1) \\ R(1) & R(O) & \ldots & R(P-2) \\ \ldots & \ldots & \ldots & \ldots \\ R(P-1) & R(P-2) & \ldots & R(O) \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \ldots \\ R(p) \end{bmatrix} \qquad \ldots(7)$$

Several efficient recursive procedures have been devised for solving this system of equations. The most popular and well known of these methods are Durbin-Levinson method [2, 15, 19].

3.2.2 Pitch and Pitch Contour

Pitch or equivalently, fundamental frequency (F0) which takes form as a result of vocal cord vibration is one of the most important problems in speech processing. When the pitch is used as a feature its statistical or dynamical value could be used.

A good pitch estimator ought to fail when presented with a periodic input and so gives a reliable indication of whether the frame of speech is voiced or not [19]. It is extracted from speech signal either in time domain or frequency domain.

The algorithm which is used in this work operates by finding a peak position which exceeds a certain threshold in appropriate range in the ACF of prediction residual signal [15]. In figure (4), a curve is drawn through four points to predict the position of the fifth, and only the prediction error is actually transmitted. If the order of linear prediction is high enough, and if the coefficients are chosen correctly, the predication will closely model the resonances of the vocal tract. Thus the error will actually be zero, except at pitch pulses. So this is a good way to determine the pitch by examining the error signal [19]. In this method the accuracy based on the choice of the range and the threshold value. If the segment is voiced, the reciprocal of the location of the interpolated peak defines F0[11].

3.3 Pattern Classification

The result of a signal processing is to produce the values of a set of features. These features are the linear prediction coefficients and the average pitch in this work.

A pattern classifier is a system for estimating the class to which a pattern belongs. Some techniques available to do this function are Dynamic Time Warping (DTW). Vector Quantization (VQ), Hidden Markov Model (HMM), Artificial Neural Networks (ANN) and other.

In this work new classifier engine so-called Cellular Automata (CA) has been presented. So this work aims to investigate to what extent the cellular automata efficient range when it is used as classifier model for speaker recognition and voice recognition.

3.3.1 Cellular Automata as Binary Classifier

CAs were introduced in the late 1940's by John Von Neumann and Stanislaw Ulam. The late John Von Neumann (1903-1957) is considered to be the founder of cellular automatastudy [1].

CAs are computer model that try to emulate the way the laws of nature are supposed to work in nature. They are discrete dynamical systems that display complex behavior despite their simple construction. Figure (5) shows an example of such a CA.[10,17].

The CAs consist of a number of constituting elements, often called cells. All cells are identical and simple of construction. The values of the cells evolve in discrete time steps according to deterministic rules that specify the value of each site in terms of the values of neighboring cells. Like in figure above, the state of a cell at time t+1 depends on the state of number of cells (in some defined neighborhood) at time t[17,20]

4. Building the Cellular Automata

Building the cellular automata includes some concepts as the following:

4.1 The Cell and States Set

The basic element of a CA is the cell. A cell is a kind of a memory element and stores state. In the simplest case, each cell can have the binary states 1 or 0 (life/death). In more complex simulation the cells can have more different states [17].

4.2 The Lattice

The cells are arranged in a spatial web. The simplest one is the one dimensional "lattice" meaning that all cells are arranged in a line like a string of pearls. The most common CA's are built in one or two dimensions. On the other hand the one dimensional CA has the big advantage, that it is very easy to visualize [17].

4.3 Neighborhoods

The updating of a given cell requires one to know only the state of the cells in its vicinity. There are several possible lattices and neighborhood structure [20] and there is no restriction on the size of the neighborhood but if the neighborhood is too large the complexity of the rule may be unacceptable [7].

4.4 Two Dimensional Cellular Automata Neighborhood

The common neighborhood definitions and used in this work is the Von Neumann neighborhood: Four cells, the cell above and below, right and left from each cell are called the Von Neumann neighborhood of this cell. The radius of this definition is 1, as only the next layer is considered [7,17]. Figure (6).

4.5 One-Dimensional Cellular Automata Neighborhood

A one-dimensional (1D) cellular automation is a linear chain of automata. The neighborhood relationship in one-dimensional automata consists of predecessors and successors. One of the better-studied neighborhoods for one-dimensional automata consists of the automation itself and its two nearest neighbors.

4.6 Boundary Condition

In cellular automata a site belonging to the lattice boundary does not have the same neighborhood as other internal sites. It is possible to define type of boundaries, used in this work, [7]:-

4.6.1 Periodic Boundary Condition (PBC)

A CA is said to be periodic boundary CA if the extreme cells are adjacent to each other as in figure (7) of one-dimensional lattice. In the case of a two-dimensional lattice the left and right sides are connected, and so are the upper and lower side [7].

4.7 Transition Function of the Cellular Automata

The most important aspect of a cellular automaton is the transition function. Of course the transition rule depends on, the neighborhood, and the state set. The new state of each automaton is a function of its own state and that of its four immediate neighbor's west, east, north and south as Figure (8). This family of automata can be defined by means of five bits that represent the state of the neighbor with state of the automaton itself. Thus there are 2^5 = 32 different input values and 2^ 32 possible state change rules. The rule expressed as a lookup table that lists for each local neighborhood and the update states are referred to the output bits of the rule table.

One of the main problems arising in the application of CA to real-world problems is how we can find the suitable rule that can perform a desired behavior especially in two dimensional CA where the search space is large and we can 't apply all rules to CA configuration to estimating the best rule. Genetic search is used to give us the solution for this problem [1].

5. Genetic Algorithm

Genetic algorithm is search algorithm based on the mechanics of natural selection and natural genetics. The beginning work on genetic algorithms was by John Holland, from the University of Michigan at the beginning of the 60s[18].

5.1 The Natural Perspective for Genetic Algorithms

The population can be simply viewed as a collection of interacting creatures. As each generation of creatures comes and goes, the weaker ones tend to die away without producing children, while the stronger mate combines attributes of both parents to produce new, and perhaps unique children to continue the cycle. Occasionally, a mutation creeps into one of the creatures, diversifying the population even more [18].

5.2 Genetic Algorithm Operators

Moving from a randomly created population to a well adapted population is a good test of algorithm. The simplest form of genetic algorithm involves three types of operation: Selection crossover, and mutation applied to each of the successive generation.

5.2.1 Selection

Selection is the component that guides the algorithm to the solution by preferring individuals with high fitness over low-fitted ones. It can be a deterministic operation, but in most implementations it has random components [4]. We used two types of selection strategies as follows:

1) ***Roulette wheel selection***

In the case of genetic algorithm roulette wheel could be used to select individuals for further reproduction. The wheel corresponds to fitness array and the marble is a random unsigned integer less than the sum of all fitness in population. The marbles value is less than the current fitness element, the corresponding individual [4]. Figure (9).

2) ***Tournament selection***

Pairs of individuals are picked at random from the population, the higher fitness is copied into a mating pool. This is repeated until the mating pool is full. It is possible to adjust its selection pressure by changing tournament size. The winner of the tournament is the individual with the highest fitness of the tournament competitors, and the winner is inserted into mating pool. This fitness difference provides the selection pressure, which drives the genetic algorithm to improve the fitness of each succeeding generation [5].

5.2.2 Crossover

To perform crossover, two of the parent candidates are chosen to mate at an arbitrary point in the binary string, the remainders of their binary strings are swapped. For example, two binary strings are chosen to mate. The Genetic Algorithm decides to perform crossover after the M bit in each string. After the crossover is completed the two off spring then replace the two weakest solutions in the population. Two parents may or may not be replaced in the original population for the next generation. The common crossover types used in this work is single point crossover [13].

In a single point crossover, a point is picked in the chain randomly joined the two parents together. All bits before that point are taken from one parent. The remaining bits are then taken from the other. This method is often used for GA, which operate on binary strings. For other problems or coding, other crossover methods can be useful. Figure (10).

5.2.3 Mutation

The mutation operation introduces random changes in structures in the population. The mutation operation can be beneficial in reintroducing diversity in population that may be tending to converge prematurely. The individual is selected with probability proportional to the normalized fitness. The result of this operation is one offspring whereas without the mutation, all the individuals in population will eventually be the same (because of exchange of genetic material). Mutation in case of binary string is just inverting a bit in the selected chromosome [5,13].

6. Speaker Recognition Training

In order to evaluate a speaker recognition system, we need to create a database of speakers. A set of sample utterances is recorded for each user, so as to be used for training and testing in the system. In this work case, we have used sound forge application and suitable-quality microphone in anechoic room during the recording stage to transform these utterances to "wav" files that can be easily manipulated by almost any windows program. We have saved them as Microsoft wave mono files, using 11025 Hz sampling rate and 8 bits/sample. This is a common choice, used in may speeches processing applications.

The speech data consisted of 560 repetitions for 3 words "project, عبور, computer" spoken by 14 speakers (7 males, 7 females). For each speaker, we have used 40 utterances of one word, 30 utterances providing the essential input data for the system to train and the remaining where then used for the verification and identification of the user's identity. Collecting of this data was achieved on two separate sessions for each speaker during two month.

In the analysis stage each utterance was first manipulated by an ends point detector, the output utterance was divided into a number of frames of 23.2 ms with 50% overlapping. The frame was windowing by hamming window and passing to feature extraction stage where speech data is reduced to much smaller amounts of data which represent the important characteristics of the speech. The output of this stage is a vector that contains 13 parameter data. The accumulated of these vectors are so-called feature vectors that are used as pre-classified patterns. Figure (11)

6.1 Evolving Cellular Automata with Genetic Algorithms

We have used a genetic algorithm to search CA rule table to perform classification task. Each chromosome in the population represented a candidate rule table. It consisted of the output bits of the rule table. The chromosomes representing rules were thus length 2^5. The size of the space in which CA worked was too large for any kind of exhaustive evaluation. The GA began with initial population of 100 chromo-somes generated randomly. The fitness for each individual was particularized by the following: The CA is fed with a training set of pre-classification patterns and evolved with the indicated individual. The local fitness for each cell in the grid is computed and returned the individual, with best local fitness and cell location (classification cell).

The new population was formed by single-point crossover between randomly chosen pairs of elite individuals with crossover rate 0.6 from the old population. The parent individuals were chosen under tournament selection method. The offspring from each crossover were each mutated at exactly two randomly chosen positions with mutation rate 0.001. This process was repeated until stop criterion satisfied for a single run of the GA. The best CA rule for each person is used for classification stage using the final state of the classification cell as a bunary output.

6.1.1 Local Fitness Computation

Fitness function is used to evaluate the fitness of the individuals in population), better solutions will get higher score. Evaluation function directs population towards progress because good solutions will be selected during selection process and pour solutions will be rejected [5,18]. Local fitness is defined by the following equation:

$$Fitness = \sqrt{\frac{(Sensitivity)^2 + (Specificity)^2}{2}} \quad \ldots(9)$$

where sensitivity is computed on the whole training set as the frequency with which the cell has reached the "1" state in the presence of patterns belonging to the target class, specificity is estimated as the frequency with which the cell has reached the "0" state in the absence of the target class. The use of the mean square root ensures that a good balance between specificity and sensitivity is reached [1]. This fitness is local, as it is measured only on the final state of each cell.

6.1.2 Stop Criterion

Stop criterion for the system is 100% performance level for each speaker or a certain number of GA generations (40 generations in this work).

7. Voice Recognition Model

In voice recognition model we recognize male and female voices. The activity of the model takes place in a 1D with PBC and length of 10 cells as in figure (12). The site value will consist of 0 or 1. The neighborhood size will be fixed at r=1 with Von Neumann neighborhood. A 1D CA starts out with an initial configuration (IC) of cells states and this configuration change in discrete time steps in which all cells are updated simultaneously according to the CA "rule". Just as in 1D CA each cell of a lattice computes the same function so ID CA requires one rule be designed for all cells. The input to the ID CA is the average pitch feature which is encoded as the IC. The average pitch for a speaker varies considerably from one individual to another but by itself is not sufficient to distinguish between many speakers. The output is decoded from the configuration reached at some later time step.

7.1 Transition Function of the 1D Cellular Automata

A one-dimensional CA has cells in two possible states-on or off. Each cell only has two neighbors and the state of a cell is dependent on its own and its two neighbor's states in the previous generation. Thus, there are eight possible combinations of the states of those three cells, namely 000, 001, 010, 011, 100, 101, 110, 111, expressed as a lookup table which maps to 0 or 1 at the next time step and the rule defines the state of the cell in question for each situation. Thus we can have a total of 2^8=256 different rules. Any of these 256 different rules can be used to evolve the cellular automaton. Figure (12).

7.2 Voice Recognition Training

The speech database consisted of 800 repetitions for 4 words" project, عبور, computer, مـــشروع "spoken by 20 speakers (10 males, 10 females), 700 utterances were used for training while the remaining utterances were used for testing.

The 1D CA is fed with training set of male/female patterns and evolved with each individual in the population (CAP) of CA rule tables. The local fitness for each cell in the CA is computed by the equation (14) where Male Sensitivity is computed on the whole training set as the frequency with which the cell has reached the "1" state in the presence of patterns belonging to the male class, Female Specificity is estimated as the frequency with which the cell has reached the "0" state in the presence of patterns belonging to the female class like 2D CA. The best CA rule is used for classification stage using the final state of the cell with highest-fitness as a binary output. The problem arises in this model is not the space size of the rules but the steps required to achieve suitably-Fitness cell, we will chow that later in the experiments. Figure (13).

8. Speaker Recognition Model Experiment Configuration

In these series of experiments, we are going to evaluate the performance of the developed system during using different configuration First of all experiments were done to construct our feature vectors using different Linear prediction coefficients extracting from each input frame. We constructed our feature vectors with 4-LPC, 6-LPC, 8-LPC, 10-LPC, 12-LPC, 14-LPC and 16-LPC in addition to the average pitch in each experiment, where we needed to examine the performance of speaker verification configurations. We used a set of two speakers, 1-male and 1-female as a database for these experiments. After each one system's performance was evaluated to observe the effect of using different feature vectors. The next plots of correct acceptance rate and correct rejection rates in speaker verification system exhibit this effect. As we can see, the system performs better with LPC configuration 12, 14 and 16 than the others, and at the same time we can see that the 12-LPC is the suitable from where the system performs, space store and time processing. Figure (14).

The second series of experiments were that associated with training phase in order to obtain the best rules for all trainee speakers. During this experiments we concentrated on three parameters: selection methods, population size and number of GA generation, in the meantime the parameters controllers of signal analysis and CA configuration were invariable as table (1), table (2).

LPC	12 coefficient
Frame Size	23.2 ms
Overlapping	11.6 ms
Window Type	Hamming Window

Table (1): Parameters controllers for signal analyses.

CA Dimension	2D CA
Lattice Size	13X11
Neighborhood Type	Von Neumann
Boundary Condition	PBC
Radius	1
Number of CA steps	13

Table (2): CA configuration for speaker recognition system.

In the first experiment implemented using selection methods, we used two types of these methods and compare between them in performance, roulette wheel selection method and tournament selection method, the parameters values for GA that were used in first experiment are given in table (3).

Selection Method	Roulette Wheel
Population Size	100
No of Generation	30
Chromosome Length	32
Crossover Operator	Single point
Crossover Probability	0.6
Mutation Operator	Change
Mutation Probability	0.001
Feature Vector Size	13
Thaining Vector	150
Person Vectors	30
Non Person Vectors	120

Table (3): Parameters value for GA in first experiment of SRT

In the second trial the same parameters in table (3) were used with the exception of selection method operator replaced by tournament selection on the same initial population. The results of this experiment are exhibited in table (4) and table (5) for five speakers 3 males and 2 females were used as data base in the training of this experiment. If we have a look at these tables we can clearly see that by using tournament selection rules with best fitness attainment rapidly than roulette wheel selection. Figure (15).

Tables (4),(5). Performances of system on five speakers under roulette wheel

Speaker Name	Sensitivity	Specificity	Fitness		Sensitivity	Specificity	Fitness
OrC	0.65	0.888	0.778		0.875	0.900	0.888
HdC	0.975	0.800	0.891		1.00	0.806	0.908
AsC	1.00	0.787	0.900		1.00	0.787	0.900
KbC	.900	0.56	0.875		.900	0.856	0.875
SrC	0.875	.900	0.888		0.95	0.869	0.91

selection and tournament selection.

Others experiments were carried out to discover rules more suitable than the previous. In the beginning population size was increased from 100 individuals to 150 individuals finally to 200 individuals with 20 generations under tournament selection as in Figure (16) for male speaker.

After that we increased the number of generation from 20 generation to 30 generation finally to 40 generation with 200 individuals in population size also under tournament selection as in Figure (17), Figure (18) and Figure (19) for some speakers, the final experiment was implemented on all speakers database to obtain the results exhibited in table (6), some of these results can be obtained better if we continue to increase the number of generation but really we could not do this because the training time also will increase to exceed our faculty.

Speaker Name	Speaker Rule	Classification Cell		Sensitivity	Specificity	Fitness
		Row	Column			
AdP	FFA8EDEO	14	7	1.00	0.950	0.975
AiA	41755C27	4	4	0.950	0.950	0.950
AsC	15C10F9A	5	4	1.00	0.887	0.945
DaP	E0F5DD5D	6	7	1.00	0.900	0.951
HdC	A700A5	14	7	1.00	0.950	0.975
HmP	C085512	6	5	0.925	1.00	0.963
KbC	F8A0D8D	3	6	0.975	0.875	0.926
NaP	7F5F070F	6	8	0.975	0.950	0.963
NnA	7E824A	2	5	0.925	0.806	0.867
OrC	4500F0E3	7	7	0.975	0.925	0.949
RaA	FA40F822	3	4	1.00	0.743	0.881
SrC	7FA8BBA0	12	4	0.975	0.981	0.978
WdA	600B346	10	3	1.00	0.837	0.922
WnA	206E000A	5	5	0.975	0.918	0.947

Table (6): Results of speaker recognition training for 14 speakers under 40 generations of GA on population of 200 individuals.

9. Speaker Recognition Testing

In each system to measure its performance it must have test phase which runs after the training phase. As we know in training phase system requires a set of utterances to implement this phase. Likewise testing phase system requires a set of new utterances to implement this phase also. In this system we use a new set of 140 utterances spoken by 14 speakers from whom previous utterances had been recorded. As we have shown before speaker recognition comprises two types of applications, speaker verification and speaker identification therefore we had implemented two types of testing as follows:

9.1 Speaker Verification Implementation

In speaker verification system the person wishing to be verified first utters his verification word. The spoken utterance must be accurately pinpointed. This is the job of the endpoint detection procedure described previously. Once the beginning and end of the utterance have been found, it is blocked into 23.2 ms. After that a series of measurements and parameter estimates are used to provide pattern which represents the utterance. In particular, an LPC analysis is used to give 12 predictor parameters, a pitch detector is used to measure the average pitch. Finally CA-based classifier with claimed identity rule and its classification cell are called from database. The obtained pattern presented to this CA-based classifier which evolved for 13 steps under claimed identity rule. In final step the classification cell is checked to make the decision, if its final state equals "one" the person is accepted "authentication" otherwise the person is rejected Figure (20).

As we mentioned before in speaker verification there are two classes of errors false acceptance and false rejection. A false rejection occurs when the system incorrectly rejects a correct speaker. A false acceptance occurs when the system incorrectly accepts an imposter. Table (7) exhibits the results from speaker verification testing.

9.2 Speaker Identification Implementation

There are many similarities between the problems of speaker verification and speaker identification. In terms of the utterance recording and signal processing aspects where the stages applied to speaker verification in order to provide the pattern that represents the utterance are the same stages as those applied to speaker identification to provide the same pattern. The differences begin after the pattern was obtained. To identify the unknown speaker identification to provide the same pattern. The differences begin after the pattern was obtained. To identify the unknown speaker the pattern is presented into CA based-male-female classifier to make decision. If the speaker is male then pattern is presented into the set of male CA-based classifier separately in which each one must be evolved with this pattern through 13 steps. The final state for classification cell of each classifier is checking to make the decision. If the final state of any one of these cells equals one then it returns the speaker name whose CA-based classifier is his, else it returns unknown speaker. If the pattern is for female speaker then it is presented into a set of female CA-based classifier separately and applying the same stage implemented above for male speaker Figure (21).

In speaker identification there are two types of error which occurs when the system does not know the correct person or return incorrect person. In addition to this types of error, a problem can emerge associative with this speaker identification system. This problem is called classification collisions. It occurs when one or more of input pattern belong to more than one class into the same output pattern. In other words for the same pattern presented to number of CA-based classifiers there are more than one classification cell its final state is equal to "one". Two types of solutions to this problem were used. The first solution was implemented with extending the system architecture by adding a set of classifiers trained on pattern belonging to the pairs of classes that produced the highest number of collisions. The second solution was by one of the most easily implementable methods, as it does not require much additional training, it is the so-called stacked generalizer [1].

Table (7) exhibits the results from speaker identification testing.

Speaker Name	Sensitivity	Specificity	Speaker Verification Performance	Speaker Identification Performance
	Correct Acceptance	Correct Reject		
AdP	0.900	0.875	0.887	0.700
AiA	0.800	0.950	0.878	0.800
AsC	0.900	0.875	0.887	0.900
DaP	1.00	0.800	0.905	0.800
HdC	0.900	0.975	0.9382	1.00
HmP	0.800	1.00	0.905	0.800
KbC	0.800	0.900	0.851	0.800
NaP	0.900	0.950	0.925	0.900
NnA	0.700	0.875	0.792	0.700
OrC	0.900	0.950	0.925	0.900
RaA	0.900	0.825	0.863	0.900
SrC	1.00	0.950	0.975	1.00
WdA	1.00	0.825	0.916	1.00
WnA	0.900	0.925	0.9125	0.600

Table (7): Results of speaker verification and identification testing on 14-Speakers.

10. Voice Recognition Model Training Experiments

We have mentioned previously that in one-dimension cellular automata the number of rules is limited. This gives us the ability to apply these rules with presented patterns without need to use genetic algorithm. The problem arisen is when all these rules are applied and the rule with suitable fitness does not appear. So it was necessary to recourse to the number of cellular automata steps. This is done by change the steps number for number of trials and checking the result every once to make sure that we obtain a suitable rule from this trial. We examined 499 experiments beginning from 1D cellular automata with 2 steps ascending to 500 steps. Following is a planning figure for final experiment, which explain the rules fitness, maximum fitness and minimum fitness with the indicated steps Figure (22) and table (9). During all these experiments the CA configuration was stable with the exception of number of steps as table (8).

CA Dimension	1DCA
Lattice Size	1X10
Neighborhood Type	Von Neumann
Boundary Condition	PBC
Radius	1

Table (8): CA configuration Information for voice recognition

Number of CA steps:500			
		Max.Fitness	Min-Fitness
Fitness Value		1.00	0.5071
Classification Cell	Row	1	1
	Column	2	4
The Rule		11110000	01111110

Table (9): Result of training Voice recognition model with 1D CA of 500 steps.

11. Voice Recognition Model Test

Test phase of voice recognition model was carried out in the same steps which were carried out in the training phase. In the beginning speaker voice was recorded, analyzed to extract the average pitch which was represented by the feature vector and presented as pattern to the rule, obtained during the training phase and stored previously in the database with the number of steps of (500 steps) and classification cell. The recognition depended on the final state of this cell: if it is equals "1" the speaker is male otherwise female Figure (23). We have tested 100 vectors (50 males, 50 females) and the result is explained in table (10) and table (11).

Speaker	Sex	Pattern No.	Result	
			Male	Female
AdP	Male	5	5	0
AiA	Male	5	5	0
AsC	Male	5	5	0
HmP	Male	5	5	0
OrC	Male	5	5	0
SrC	Male	5	4	1
WdA	Male	5	5	0
MrA	Male	5	5	0
BrC	Male	5	4	1
ArP	Male	5	5	0

Table (10): Voice recognition test on 50 vectors of 10 males

Speaker	Sex	Pattern	Result	
		No.	Male	Female
DaP	Female	5	0	5
HdC	Female	5	0	5
KbC	Female	5	0	5
NaP	Female	5	1	4
NnA	Female	5	0	5
RaA	Female	5	0	5
WnA	Female	5	0	5
R1A	Female	5	0	5
AmP	**Female**	**5**	**0**	**5**
ArP	Male	5	5	0

Table (11): Voice recognition test on 50 vectors of 10 Females

12. Results Discussion and Conclusion

In order to study the performance of the system during the previous series of experiments, first of all, we are going to discuss some results of these experiments.

If we have a look at Figure (13) we can clearly see that using more coefficients of a feature set can enhance the performance of our system. This is mainly due to the fact that by increasing the order of the coefficients, the vocal tract is modeled more efficiently. The relationship between the system's performance and the order of the coefficients, the vocal tract is modeled more efficiently. The relationship between the system's performance and the order of the feature set can not be linear, implying that the performance of our system may not improve after a certain number of coefficients.

In GA to achieve best result we must change some operators whatever manner adapted in our work. The selection operator is the bottleneck for GA operation whereas if the selection method has sufficiency qualification this will give more possibility to improvement of the offspring for the next generation. If we have a look at table (4) and table (5) we can clearly see that using tournament selection enhances the performance of our system faster than roulette wheel selection because in tournament selection rate, and the individuals with the highest fitness have highest selection rate. On the other hand in roulette wheel each individual in the population has a roulette wheel slot size in proportion to its fitness.

If we have a look at Figure (14) we can clearly see increasing the population size can enhance the performance of our system. This is done mainly because increasing the number of individual in the population size will give us the ability to check performance for a large number of individual in each generation, on condition that this population size will not increase the time of processing to the level that effect on the number of generations, as we can see in figure (15) by increasing the number of generation we will obtain better offspring than its predecessor. This is not main there are linear relationships between them, implying after a certain number of generations, the system tends to have the same performance.

If we look at the table (7) we can see the performance of speaker verification system for 14 speakers, when the percentage for collection of the correct acceptance is 88% and the percentage for collection of the correct reject is 90%. So the performance of speaker verification system is 89% accuracy. The second part of the testing experiments is for speaker identification system where we can see the results also in table (7). The speaker identification performance was evaluated by the percentage of collection of the correct identification for 14 speakers which equals 84% accuracy. The third part of our system is the voice recognition or male and female classification. A one-dimension cellular automaton was used for this model because the feature vector contains one feature only. The execution of this model was better than that of the previous model from the time processing and store size. During the training phase we had carried out a series of experiments. We obtain rule with 100% performance. In testing phase we applied this rule to test pattern from database and from real time recording shown in table (10) and table (11) where we can see the performance of the model is 97% accuracy.

Using cellular automata as binary classifier for speaker recognition and voice recognition present high efficiency and demonstrates its capability to achieve this type of tasks. Also we found that 12-LPC with average pitch features were sufficient to discover the owner of this identity.

The important feature for male/female classification is that the average pitch feature. It is very useful in the discrimination process while previous research refers only to maximum pitch feature.

13. Proposals for Future Work

1) Using the already developed infrastructure one can conduct a series of experiments in order to explore the effect of the following on the system's performance:

1. The length of each frame.
2. The overlapping ratio in segmentation.

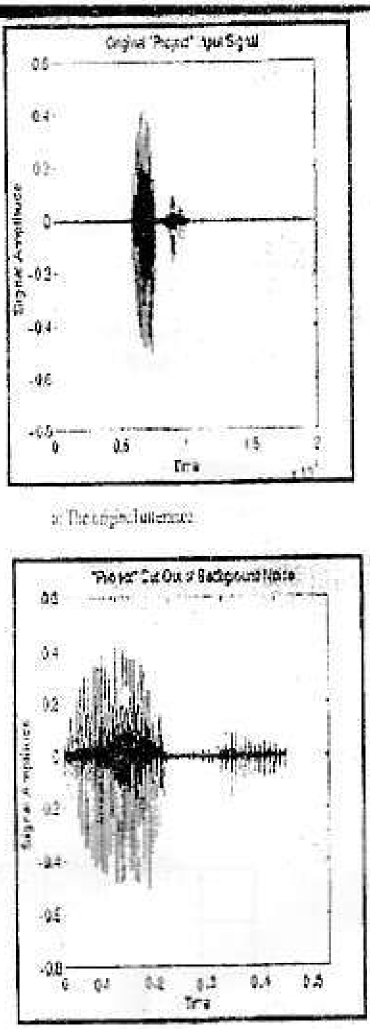
2) Combining Cellular Automata as speaker recognition model and other speaker modeling techniques, such as Neural Networks and Vector Quantisation to implement the task of the pattern classification.

3) Finally, if we are go to use a speaker recognition system in a real life environment, this implies that we are not going to use sample speech recordings from a speech database, then may be we will have to deal with the possible background noise, speech enhancement techniques have been proposed for use in speaker recognition systems.

References

1. Adorni, G., Bergenti, F., Cagnoni, S., "A Cellular-Programming Approach to Pattern Classification", Proceedings of the First European Workshop on Genetic Programming", Vol.1391, pp.142-150, Springer-Verlag, 1998.
2. Ainsworth, W.A., "Speech Recognition by Machine", Peter peregrines Ltd, London, U.K., 1988.

3. Bishnu, S.A., "Automatic Recognition of Speakers from Their Voice", IEEE, Vol.64, pp.460-475, 1975.
4. Bodenhofer, U., "Genetic Algorithms: Theory and Applications", Fuzzy Logic Laboratorium, Linz-Hagenberg, Austria, 2002. http://www.flll.unilinz.ac.at/lectures/GA/notes.pdf.
5. Busetti. F., "Gnetic Algorithms Overview", technical report, 1998. http://www.flll.unilinz.ac.at/lectures/GA/notes.pdf.
6. Chaudhuri, P.P., Chowdhury, D.R., Nandi. S., Chattopathyay, S., "Additive Cellular Automata: Theory and Application, Volume", IEEE Computer Society press, Los Alamos, California, 1997.
7. Droz, M., Chopared, B., "Cellular Automata Modeling of Physical System, "Cambridge University Press, UK., 1998.
8. Fallside, F., Woods, W.A., Computer Speech Processing, Prentice-Hall International (U.K.), 1985.
9. Ganesh, N., Sugumaran, K., "Speaker Verification in Forensic Application", Surprise 95 Journal, Vol.4, Imperial College of Science Technology and Medicine, London, 1995.
10. Kortekaas, R., "Cellular Automata and Speech Recognition", Institude of Phontic Sciences University of Amsterdam, Report No.120, 1992.
11. Macbeth, S., "Speaker Independent Connected Speech Recognition", Report, FGC, New York, 2000, http://www.fifthgen.com/speaker-independent-connected-s-r.htm.
12. Markel, J.D., Gray, A.H., "Linear Prediction of Speech", Springer-Verlag, Berlin, 1976.
13. Moore, T., Schumacher, D., "Genetic Algorithms and their Application to Continuum Generation", The Ohio State University, REU, 2001, http://www.physics.ohio-state.edu/~reu/02reu/REU2001reports/tmoorepaperout.final.pdf.
14. Rabinar, L.R., Juang, B.H., "Fundamentals of Speech Recognition", Pretice-Hall, New Jersy, 1993.
15. Rabinar, L.R., Schafer, R.W., "Digital Processing of Speech Signals", prentice-Hall, New Jersey, 1978.
16. Rodman, R.D., "Speker Recognition of Disguised Voices: A Program for Research", Department of Computer Science North. Carolina State University, U.S.A., 1998.
17. Schatten, A., "Cellular Automata Digital Worlds", Tutorial, Vienna University of Technology, http://www.ifs.tuwien.ac/~aschatt/into/ca/ca.html.1999.
18. Sullivan, M., "An Introduction to Genetic Algorithm to Genetic Algorithms", 1997, http://www.cs.qub.ac.uk/~M.Sullivan/ga/ga index.html.
19. Witten, I.H., "Principles of Computer Speech, "Acaddemic Press INC.(London) LTD, 1982.
20. Wolfram, S., "Cellular Automata", article, 1983, http://www.StephenWolfram.com/publications/articles/ca/83-cellular/index.html.



a: The original utterance

b : Utterance after end point detection

Figure (2): End point detection

Figure (3): Hamming Window

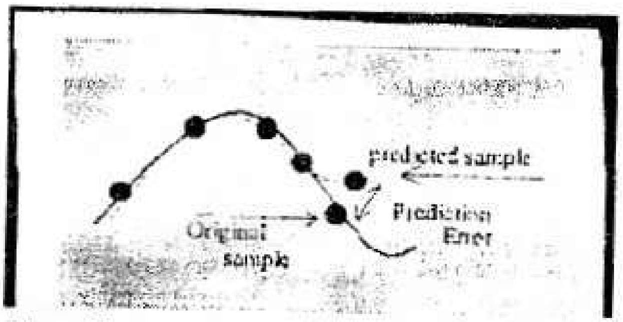


Figure (4): A curve is drawn through four points to predict the position of the fifth.

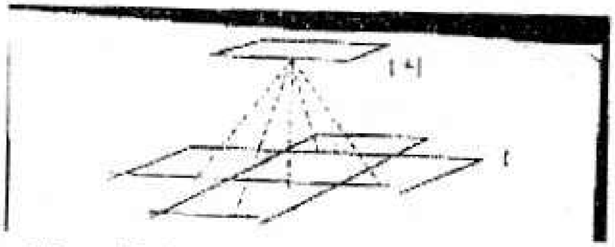


Figure (5): (part of) a cellular automaiz whose constructing elements are represented by squares.

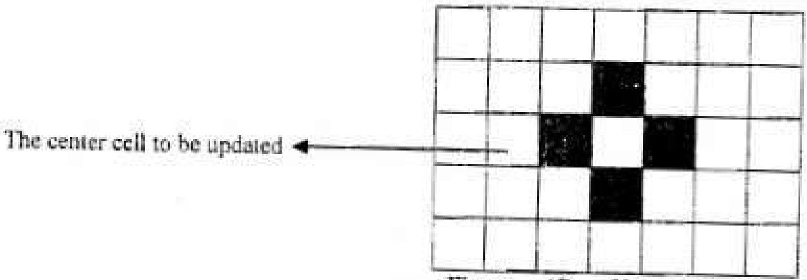


Figure (6). Von Neumann neighborhood

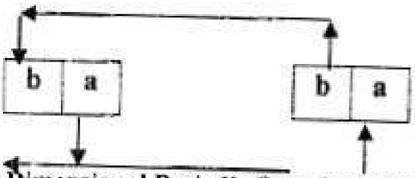


Figure (7): One-Dimensional Periodic Boundary Cellular Automata

	NO	
WE	CE	ES
	SO	

Figure (8): Immediate neighbor's west, east, north and south for automaton CE.

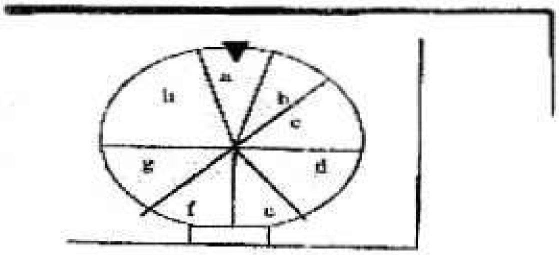
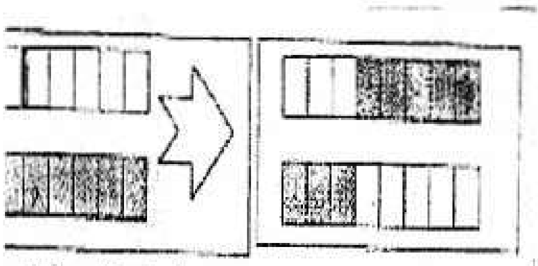


Figure (9): A Roulette-Wheel selection method

Figure (9): A roulette-wheel selection method.



e (9) Shows Single point crossover. White and Gray represent

genetic material (genes).

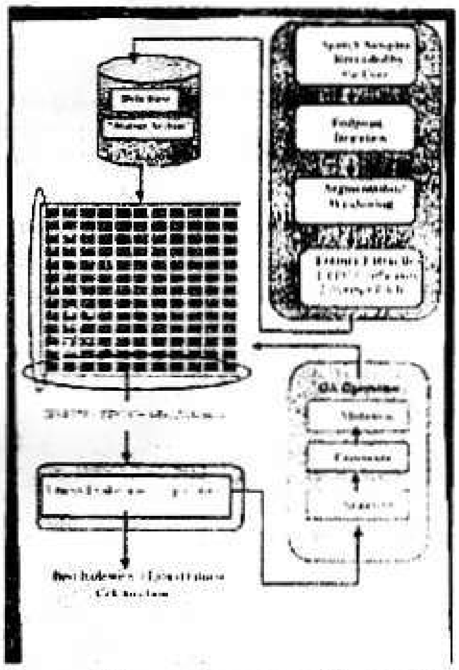


Figure (11): speaker recognition training

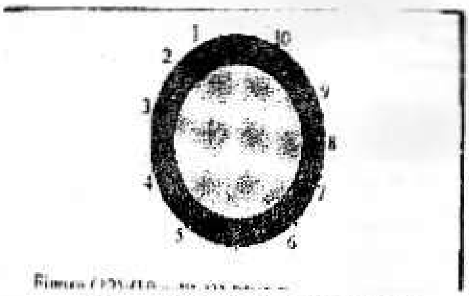


Figure (12): (10 cell) ID PBC Cellular automata

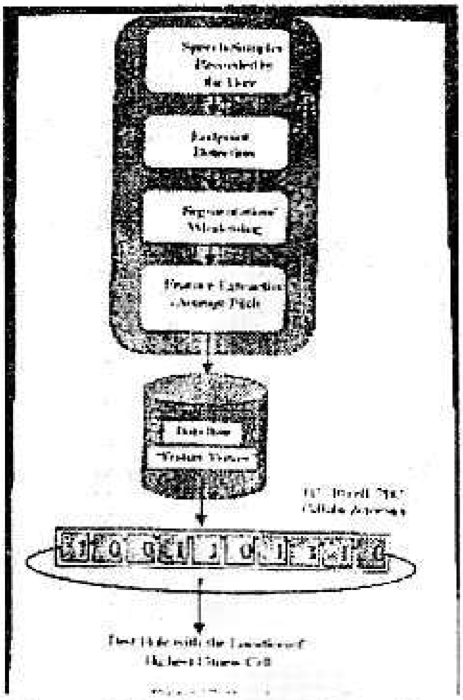

Speech Samples Recorded by the User
Endpoint Detection
Segmentation/ Windowing
Feature Extraction Average Pitch
Data Base
Feature Vectors
1D Based Rule Cellular Automata
Best Rule with the Location of Highest Fitness Cell

Figure (13): voice recognition training

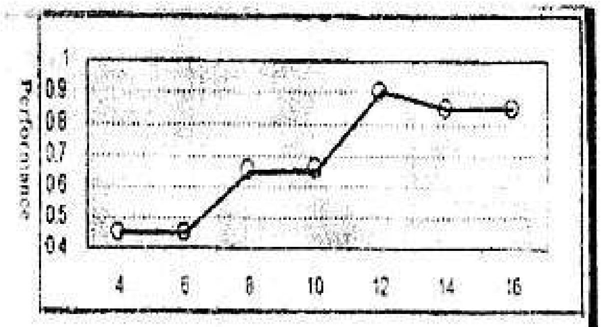


Figure (13): Correct Acceptance Rate with LPC of (4,6,..16)

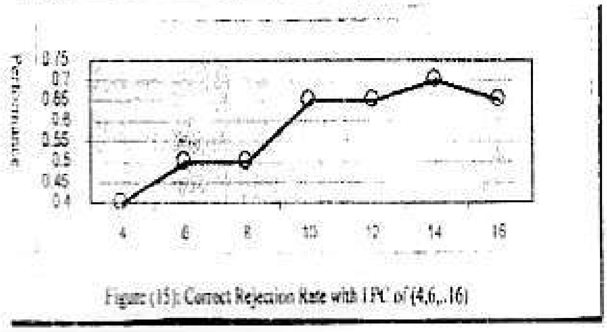


Figure (15): Correct Rejection Rate with LPC of (4,6,..16)

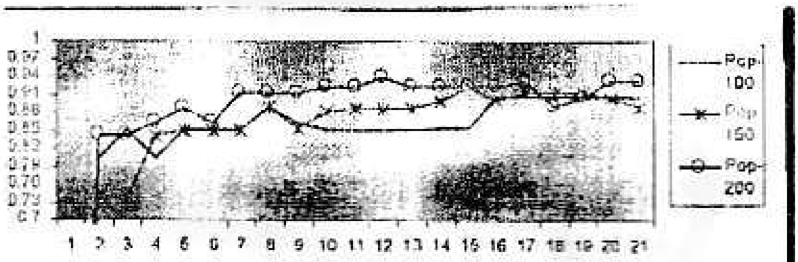


Figure (16): Rule Performance under GA of 20 generations on 100,150,and 200 individuals in the initial population for female speaker

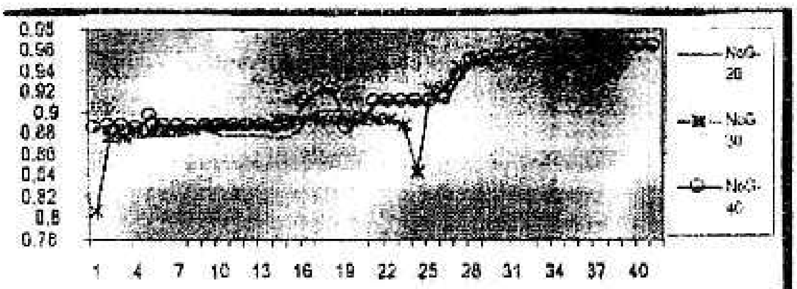


Figure (17): Rule Performance under GA of 20,30 and 40 generations on 200 individuals in the initial population for male speaker .

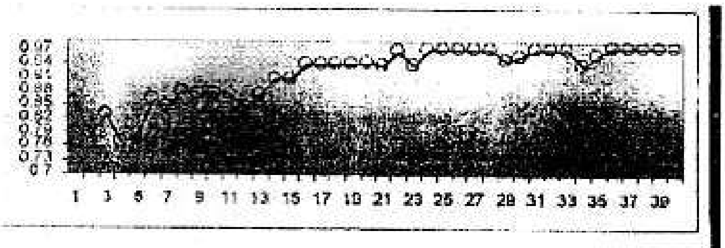


Figure (18) System Performance for Speaker SrC under 40 Generations of GA on population of 200 individuals

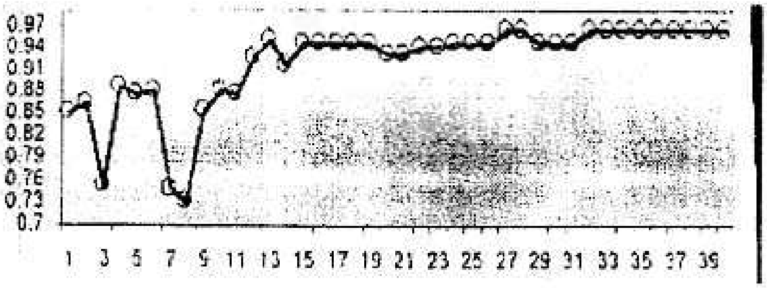


Figure (19): System performance for speaker SrC under 40 generations of GA on population of 200 individuals

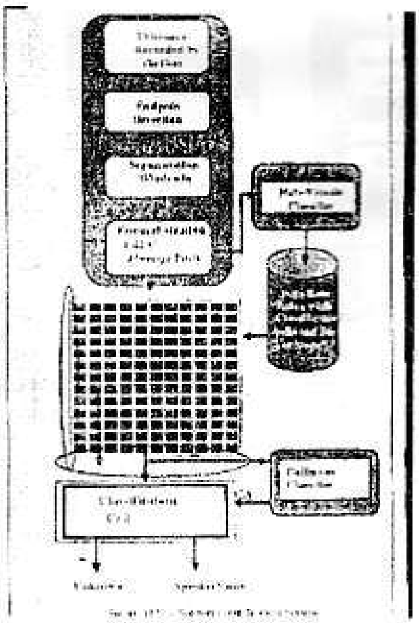


Figure (20) Speaker identification system

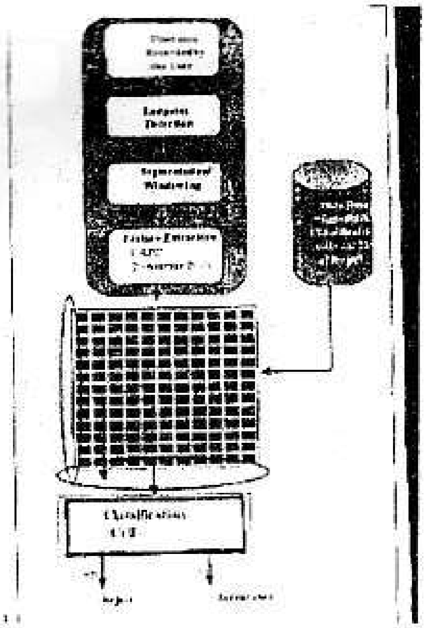


Figure (21) Speaker verification system

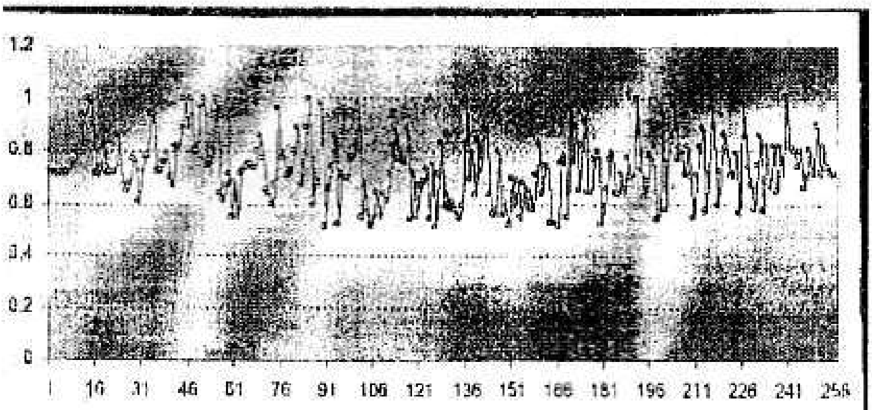


Figure (22): Result of 256 rules in column and their fitness value in row for voice recognition training with 1D CA of 500 steps

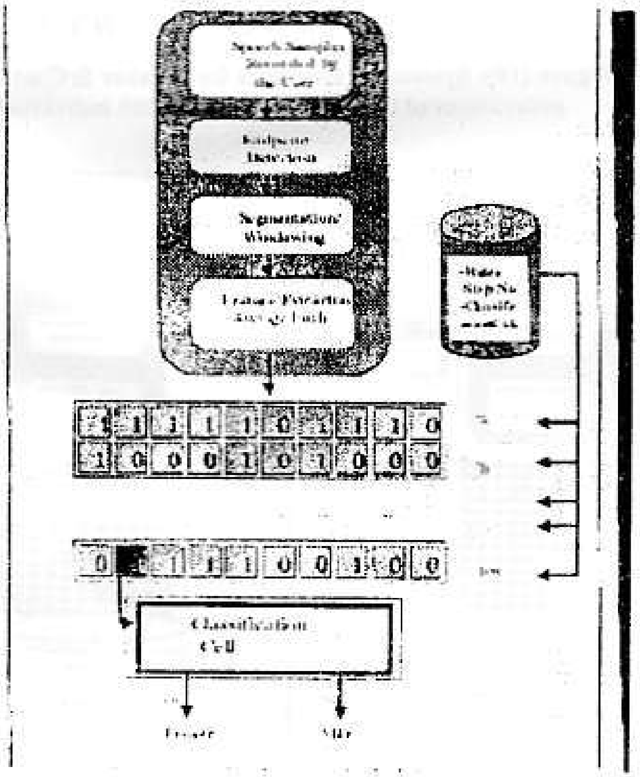


Figure (23): Voice recognition model test