An Improved Fuzzy Neural Network for Solving Uncertainty in Pattern Classification and Identification

M. Hariri*, S. B. Shokouhi* and N. Mozayani**

Abstract: Dealing with uncertainty is one of the most critical problems in complicated pattern recognition subjects. In this paper, we modify the structure of a useful Unsupervised Fuzzy Neural Network (UFNN) of Kwan and Cai, and compose a new FNN with 6 types of fuzzy neurons and its associated self organizing supervised learning algorithm. This improved five-layer feed forward Supervised Fuzzy Neural Network (SFNN) is used for classification and identification of shifted and distorted training patterns. It is generally useful for those flexible patterns which are not certainly identifiable upon their features. To show the identification capability of our proposed network, we used fingerprint, as the most flexible and varied pattern. After feature extraction of different shapes of fingerprints, the pattern of these features, "feature-map", is applied to the network. The network first fuzzifies the pattern and then computes its similarities to all of the learned pattern classes. The network eventually selects the learned pattern of highest similarity and returns its specific class as a non fuzzy output. To test our FNN, we applied the standard (NIST database) and our databases (with 176×224 dimensions). The feature-maps of these fingerprints contain two types of minutiae and three types of singular points, each of them is represented by 22×28 pixels, which is less than real size and suitable for real time applications. The feature maps are applied to the FNN as training patterns. Upon its setting parameters, the network discriminates 3 to 7 subclasses for each main classes assigned to one of the subjects.

Keywords: Classification, Fingerprint, Fuzzy Neural Network, Fuzzy Neurons, Identification, Supervised Learning Algorithm.

1 Introduction

Pattern recognition system should work well in presence of pattern rotation, translation and scaling; besides it should make decision about displacement, elimination and addition of patterns. We are dealing with dynamic and uncertain images and patterns, so the fingerprint identification is a problem highly depends on an expert's experience, knowledge, and experimental skills.

Recently, Neural Network have been used in pattern recognition problems [1]-[3], especially where the input patterns are shifted in position and scaled. For instance Fukumi and Perantonis et al. introduced a neural pattern recognition system, which is invariant to the translation and rotation of input patterns [4] and [5]. An unattractive feature of such networks is that the number of weights and complexity increase greatly as the network grows.

Classification and identification in presence of uncertainty are an important problem in pattern recognition. It is believed that the effectiveness of human brain is not only due to precise cognition; but it also exploits fuzzy reasoning. Consequently, the fuzzy network theory has proved itself to be of significant importance in pattern recognition problems [6] and [7].

The features of fuzzy systems (ability of fuzzy information processing, using fuzzy algorithms) from one side and the features of neural networks (learning ability and high speed parallel structure) from another side make a fuzzy neural network system. One of the most important advantages of FNN is supervised learning, when we use it for pattern matching. While the learning capability is an advantage from viewpoint of Artificial Neural Network (ANN), the formation of the linguistic rule base will be advantage from the viewpoint of Fuzzy Inference System (FIS) [8].

Fused FN architecture contains ANN shared data structures and knowledge representations. A common way to apply a learning algorithm in fuzzy flexible

Iranian Journal of Electrical & Electronic Engineering, 2008.
Paper first received 3rd January 2007 and in revised form 6th February 2008.
* The Authors are with the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
E-mail: mahdi_hariri@iust.ac.ir.
** The Author is with the Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

systems is to represent it in a special ANN like architecture. However the conventional ANN learning algorithms (gradient descent) can not be applied directly to such a system. This problem can be tackled by using the new, non standard, Fuzzy neuron cells and learning algorithm [9].

Ghazanfari & Lucas proposed an expert system for pattern recognition realized by a fuzzy neural network, and applied it successfully to the diagnosis of separate Persian alphabets [10]. Exploiting fuzzy neurons in neural network has attracted attention recently. Yamakava et al. applied a simple fuzzy neuron model in a neural network for character recognition without any specific learning algorithm [11]. Pseudo Outer Product (POP) is another FNN method introduced by Zhou and Quek. It employs POP learning algorithm to define fuzzy rules affirmed by educational data [12]. This method is applied by Nikian to Persian signature identification [13].

Kwan and Cai, also proposed an FNN composed of fuzzy neurons to recognize distinct English alphabets [14].

Menhaj & Azizzadeh, by introducing a new type of fuzzy neuron, used this method for recognition noisy and shifted patterns of Farsi characters [15]. Rouhani and Menhaj applied this network to recognition of distinct Farsi alphabets by dividing input pattern to separated regions and recognition pre defined patterns like horizontal, vertical lines and dots in these regions [16]. In spite of its great flexibility, this network only does the clustering instead of precise classifying in all mentioned applications and using an unsupervised learning algorithm. Pal et. al. have done a great research on this network to optimize its operation; they changed the definition of some fuzzy neurons by employing “soft computing” and “class label vectors”. They also introduced the approximate relationship of network parameters [17].

In this paper, we use fuzzy neurons and a part of the network structure, introduced by Kwan and Cai. In addition, we have introduced new neurons and complement layers which in turn lead to considerable optimized performance of the Kwan and Cai’s network. We tested the network for fingerprint classification. This new designed fuzzy neural network can work as a complete classifier.

2 Fuzzy Neurons and Fuzzy Neural Network

A fuzzy neuron of N weighted inputs $(w_i, x_i, i = 1 \text{ to } N)$ and M outputs $(j = 1 \text{ to } M)$ is shown in Fig. 1, where all inputs and weights have real values and the outputs are positive real numbers in the range of [0, 1]. In fact, they refer to the values of membership functions of fuzzy sets. In other words, each output shows how much a specified input pattern $\{x_1, x_2, \ldots x_N\}$ belongs to the corresponded fuzzy set [14].

The useful notations on the neuron operation are as follows:

h[ ] is an aggregation function, while z is the net input of the fuzzy neuron:

$$z = h[w_1x_1, w_2x_2, \ldots, w_Nx_N] \tag{1}$$

f[ ] is an activation function, and T is its threshold level:

$$s = f[z - T] \tag{2}$$

$g_j[\ ]$ is output function of the FN network:

$$y_j = g_j[s] \qquad \text{for} \qquad j = 1 \text{ to } M \tag{3}$$

Membership functions will be considered for all input patterns in the forms of $\{x_1, x_2, \ldots, x_N\}$ according to M fuzzy sets. Consequently, fuzzy neurons are able to interpret and process the fuzzy information. In general form, weights, activity thresholds, output functions and their internal trade off can be set during the process of learning.

A fuzzy-neural network has an adaptive property due to the structure of its units (i.e. fuzzy neurons) .This property enables the network to recognize various patterns. Aggregation and activity functions are some of natural features of a fuzzy neuron. Different choices can be defined for the functions h[ ] and f[ ], where will change the attributes and features of neurons. Hence, various kinds of fuzzy neurons can be defined.

The basic network has four feed forward layers consisting of defined fuzzy neurons. The structure is shown in Fig. 2 [14]-[17].

Each neuron of the first layer corresponds to one pixel of an input pattern. We can define it either real value of building block pixels (e.g. raw alphabet patterns) or encoded values related to the desired features of input image (e.g. processed fingerprints). The imposed relations on $(i, j)^{th}$ fuzzy neuron of the first layer are:

$$s^1_{ij} = z^1_{ij} = x_{ij} \qquad \text{for } i = 1 \text{ to } N_1 \text{ , } j = 1 \text{ to } N_2 \tag{4}$$

$$y^1_{ij} = s^1_{ij} / X_{max} \qquad \text{for } i = 1 \text{ to } N_1 \text{ , } j = 1 \text{ to } N_2 \tag{5}$$

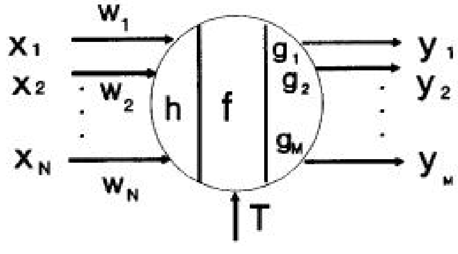


Fig. 1 A fuzzy neuron.

Here $x_{i,j}$ is $(i,j)^{th}$ value of the input array and ($\circ \le x_{ij} \le X_{max}$), hence output $y_{ij}^{[1]}$ will be normalized.
The purpose of second layer is fuzzification of the input patterns through a weight function w [m,n].

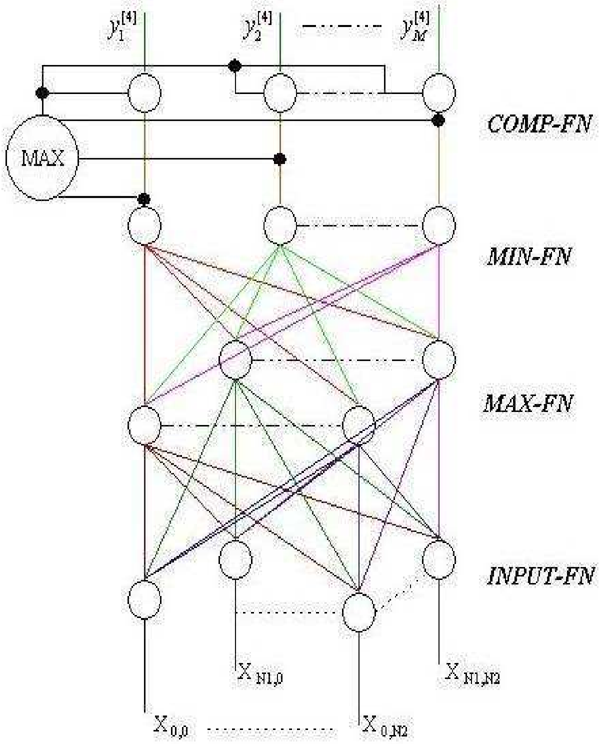


Fig. 2 Four layer feed forward FNN.

The state of the $(p,q)^{th}$ Max-FN is:

$$s_{sq}^{[2]} = \max_{i=1}^{N_1}\left[\max_{j=1}^{N_2}[W[p-i,q-j]\,y_{ij}^{[1]}]\right] \qquad (6)$$

Here, $w[p-i,q-j]$ is the weight for connecting of the $(i,j)^{th}$ input FN of the first layer to the $(p,q)^{th}$ Max-FN of the second layer,

$$\begin{aligned} &w[m.n] = \exp\left(-\beta^2\left(m^2+n^2\right)\right) \\ &\text{for} \quad m = -\left(N_1-1\right)\text{to}\left(N_1-1\right) \\ &\qquad\quad n = -\left(N_2-1\right)\text{to}\left(N_2-1\right) \end{aligned} \qquad (7)$$

In fact, the weight function $w[m,n]$ fuzzifies our network.
Each Max-FN in this layer has M different outputs (M is the number of fuzzy neurons of the third layer):

$$\begin{aligned} &y_{pqm}^{[2]} = g_{pqm}[s_{pq}^{[2]}] \\ &\text{for} \quad p=1 \text{ to } N_1, \quad q=1 \text{ to } N_2, \quad m=1 \text{ to } M \end{aligned} \qquad (8)$$

where $y_{pqm}^{[2]}$ is the m^{th} output of the $(p,q)^{th}$ Max-FN.
Depending on the designer's plan and the used matching procedure, the output function $g_{pqm}[s_{pq}^{[2]}]$ can be determined by the learning algorithm. As an alternative case, we may choose similarity criterion, i.e. isosceles triangles with heights equal to 1 and base lengths of α. Hence, the output functions will be;

$$\begin{aligned} y_{pqm}^{[2]} &= g_{pqm}[s_{pq}^{[2]}] \\ &= \begin{cases} 1-2\left|s_{pq}^{[2]}-\theta_{pqm}\right|/\alpha & \text{if} \quad 0 \le \left|s_{pq}^{[2]}-\theta_{pqm}\right| \le \alpha/2 \\ 0 & \text{if} \quad \text{o.w.} \end{cases} \\ &\text{for} \quad \alpha \ge 0, \quad p=1 \text{ to } N_1, \quad q=1 \text{ to } N_2, \quad m=1 \text{ to } M \end{aligned} \qquad (9)$$

Here, θ_{pqm} is the central point of the $g_{pqm}[s_{pq}^{[2]}]$ function, from which its distance shows similarity or dissimilarity to a certain pattern for a triangular output function.

$$y_{pqm} = 1 \text{, Complete matching (full similarity)} \qquad (10)$$

The output of the m^{th} Min-FN in the third layer is,

$$y_m^{[3]} = s_m^{[3]} = \min_{p=1}^{N_1}\left(\min_{q=1}^{N_2}(y_{pqm}^{[2]})\right) \quad \text{for} \quad m=1 \text{ to } M \qquad (11)$$

Each learned pattern corresponds to one competitive fuzzy neuron in the fourth layer. So we will have M separate neuron with non-fuzzy output in this layer. If an input pattern is the most similar to m^{th} learned pattern, then the output of m^{th} Comp-FN in the fourth layer will be 1 while other outputs will be 0. The equations for the fourth layer are shown in the Eq. (12 to 14).

$$s_m^{[4]} = z_m^{[4]} = y_m^{[3]} \qquad \text{for} \quad m=1 \text{ to } M \qquad (12)$$

$$\begin{aligned} y_m^{[4]} &= g[s_m^{[4]} - T] \\ &= \begin{cases} 0 & \text{if} \quad s_m^{[4]} < T \\ 1 & \text{if} \quad s_m^{[4]} = T \end{cases} \quad \text{for} \quad m=1 \text{ to } M \end{aligned} \qquad (13)$$

$$T = \max_{m=1}^{M}(y_m^{[3]}) \quad \text{for} \quad m=1 \text{ to } M \qquad (14)$$

This network suffers from a problem; if two images from a same pattern, with a partial translation, rotation, deformation, noise distortion or even removal of some parts are applied to the network, it will be not able to identify their similarities with the main learned pattern. Instead the applied patterns will be identified as a new class.

Creating a proper fingerprint pattern depends on many factors. So we will have several different classes (depending on the ability of the network to recognize similar patterns) instead of just one class as the index of one person's fingerprint. This will make the process of decision-making and recognition rather difficult. We propose a new designed fuzzy neural network with improved capabilities not only in character recognition, Kwan and Cai had introduced for their network, but also in flexible patterns, like fingerprint, recognition and identification as well.

3 The Improved, New Designed FNN

In our network we consider a distinct and unique class for each person containing his/her specified fingerprints. These images are acquired when a person puts his finger on the touch panel with different pressures. This work helps to consider all possible and common shapes of fingerprints. After different steps of fingerprint recovering and processing, we convert features of each fingerprint to an array with adequate dimensions and apply it to the network. The network goes to the learning stage. During the training, it is recognizing the similar arrays and saving all of them in a single collection, we called it 'Sub-class'. Those images similarities other members of a group are not accessible by the FNN will be saved in a separate class. Following a distinct fuzzy neuron will be specified for them in the third and fourth layers. According to Fig. 3, we assumed all classes created in the 4^{th} layer as sub-classes. We categorize all of the sub-classes depending on the number of main classes. Each separate class (person) has several sub-classes. Outputs of all sub-classes of a special class are steered with a single neuron which's in the 5^{th} layer. This neuron indicates a main unique class corresponding to one person.

If we use M distinct patterns for network training (from the view point of fingerprints we exploit M distinct people) we will have M specific classes and also M neurons in the 5^{th} layer. Now if we acquire K fingerprints from each person and apply them to the FNN, depends on setting strategy for the parameters, these fingerprints will be compared and the similar ones compose a distinctive sub-class. There are m_i subclasses $(i = 1 \text{ to } M)$ for each class where $m_i \le K$. In the fifth layer we can define two kinds of fuzzy neurons upon our aim; sometimes we only want to recognize the main class to which the input fingerprint belongs to. The defined equations for the fifth layer are:

$$s_m^{[5]} = z_m^{[5]} = \sum_{m=1}^{m_i} y_m^{[4]} \tag{15}$$

$$y_j^{[5]} = g_j[s_m^{[5]}] \qquad \text{for} \qquad j = 1 \text{ to } M \tag{16}$$

$$y_j^{[5]} = g[s_m^{[5]}] = \begin{cases} 1 & \text{if} \quad sm[5] > 0 \\ 0 & \text{if} \quad sm[5] \le 0 \end{cases} \tag{17}$$

An adder fuzzy neuron (Sum-FN) would be also necessary in the next layer to compute the total sum of its inputs;

$$z = \sum_{i=1}^{n} w_i x_i \tag{18}$$

For recognizing the membership degree of the input pattern to the main classes, we also want to find the subclasses to which the input pattern bears more similarity.

In order to achieve this aim, we change the output function of the 5th layer to the form of Eq. (19):

$$y_j^{[5]} = g[s_m^{[5]}] = \begin{cases} 1 & \text{if} \quad sm[5] > 0 \\ 0 & \text{if} \quad sm[5] \le 0 \end{cases} \tag{19}$$

Now the fifth layer not only specifies the main classes to which the input pattern belongs to, but also determines the number of similar sub-classes as well.

The parameters of the network related to decision making and processing are: the parameters of the output functions of the Max-FNN in the second layer, α and θ_{pqm} (for each set of p, q and m), the fuzzification function parameter β, the number of fuzzy neurons in the third and fourth layers, and the number of sub-classes recognized in each main class i (m_i).

Also we have to specify M, the total number of single main classes (i.e. the number of subjects under test), and Ki, the number of crude patterns of each main class.

K is set to maximum number of subclasses at the beginning of the training phase (it should be applied to the network in order to create the requisite subclasses in the i^{th} main class).

We define T_f as the fault tolerance threshold of our proposed FNN (i.e. similarity limitation) where $(\circ \le Tf \le 1)$.

The steps involved in the learning algorithm are as follows;

Step 1: Create N1×N2 input fuzzy neurons in the first layer and N1×N2 Max-FN in the second layer.

Choose the appropriate values for α (α>0) and β. We initialize the total number of individuals under test (M_{total}) and the number of acquired fingerprints for each individual (subject) $K_i (i = 1 \text{ to } M)$.

Step 2: Set i=0 and M=0.

Step 3: Set $k = 1$ (k is the number of the training patterns of each main class ($k = 1, \ldots, K_i$)

Step 4: Set $M = M + 1$, if $M > M_{total}$ then the algorithm will be finished, otherwise put $i = M$, the M^{th} fuzzy neuron is generated in the fifth layer and set $m_i = 0$ (begin to enter the fingerprints of a subject)

Step 5: Set $m_i = m_i + 1$, create $m_i{}^{th}$ Min-FN in the third layer and $m_i{}^{th}$ Comp-FN in the fourth layer. Compute θ_{pqmi} from the Eq. (26).

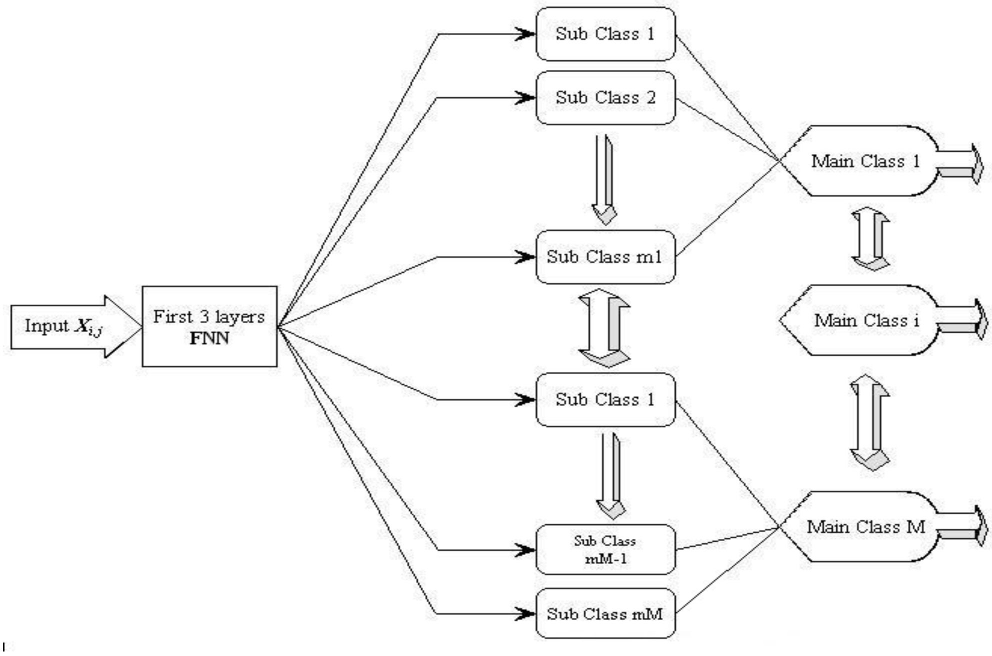


Fig. 3 The optimized new FNN with five layers.

$$\theta_{pqmi} = s^{[2]}_{pqmi} = \max_{i=1}^{N_1}\left(\max_{j=1}^{N_2}\,(w[p-i,h-j]\;x_{ijk})\right) \qquad (20)$$

$$\text{for} \quad p = 1 \text{ to } N_1, \quad q = 1 \text{ to } N_2$$

where θ_{pqmi} is the central point of the m_i th output function (it means the $m_{i-1} + m_i$ th branch) of the (p, q)th Max-FN of the second layer, and $x_k = \{x_{ijk}\}$ is the kth learning pattern of M's main class.

Step 6: Set $k = k + 1$, if $k \geq K_i$ then go to step 3, otherwise input the kth training pattern to the network and compute the output of the FNN in the fourth layer (fuzzy neurons in the third & fourth layers and M number of fuzzy neurons in the fifth layer),

$$NUM = \sum_{i=1}^{M} mi$$

$$\delta = 1 - \max_{j=1}^{mi}(y^{[3]}_{jk}) \qquad (21)$$

based on Eq.(21) δ shows the level of dissimilarity, and $y^{[3]}_{jk}$ is the output of the jth Min-FN of the third layer for the kth training pattern, X_k.

Step 7: Compare δ with T_f. If $\delta \leq T_f$ go to step 6, else $(\delta > T_f)$ go to step 5.

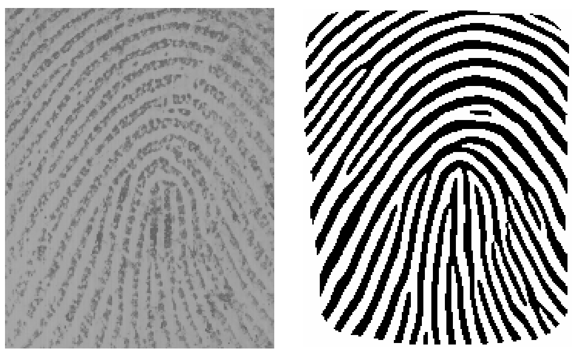
4 Fingerprints Features Extraction

The application of the proposed FNN in pattern recognition consists of two stages:

The first stage is creating the database for the fuzzy neural network; we train the network with different patterns in different groups when each group consists of one individual fingerprints.

In the second stage, we apply an unknown pattern to the network; then the network will decide about the most similar learned pattern and the related subset.

According to the processing flowchart in [18, 19, and 20], some methods for fingerprint segmentation and binarization are: segmentation with constant threshold [21, 22], regional average threshold [21], gray levels variance [23] and edge detection with Marr filter. We applied adaptive filters for fingerprint segmentation. Filtering is applied on image by convolution a K×K mask with fingerprint image in spatial domain. This filter must have odd dimensions and axis symmetric, their minimum dimensions must have ridge and it's beside furrow width. These filters make not only fingerprint binarization but also matching ridge [24]. By using this method, the fingerprint ridges and furrows are clearly indicated in Fig. 4.



(a) (b)

Fig. 4 (a) An input unprocessed fingerprint image, (b) recovered image.

Based on the recovered image, we construct the fingerprint directional image. Firstly, converting the fingerprint to a block directional image, we extract the singular points [20, 25], as shown in Fig. 5.
The thinning method has been applied to reduce ridge width to one pixel. Here we used the Sherman thinning method [26]. In this method we first find the edge of ridges then map a 3×3 window on it as its 1 point is black, at last we reserve or delete this pixel within Sherman method flowchart.
Then we extract minutiae features according their types (end ridge and branch point), their directions (16 directions, 22.5 degrees apart from each other, numbered from 0 to 15) and their positions (length x and width y) from the point directional and thinned image as shown in Fig. 6. In order to make the feature map (fingerprint features), we will encode the features of fingerprint according the encoding procedure in [19]. The extracted features for a sample fingerprint are shown in Fig. 7.

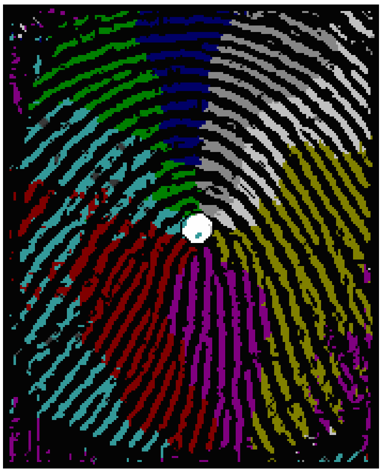


Fig. 5 A smoothed block directional image with a core, (using a block of size 8×8).

(a) (b) (c)

Fig. 6 (a) Point directional image of a recovered fingerprint, (b) Smoothed image, (c) Thinned image.

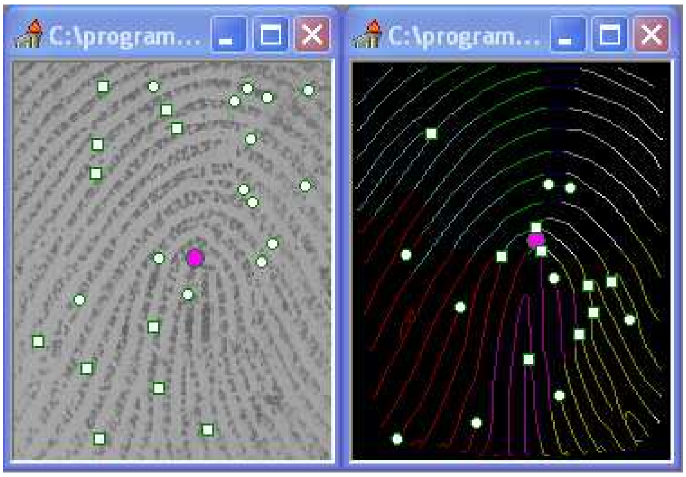
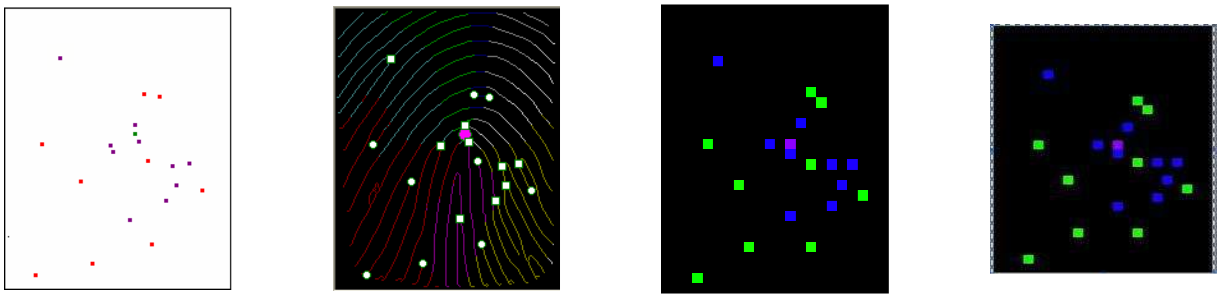


Fig. 7 Fingerprint feature extraction.



(a) (b) (c) (d)

Fig. 8 (a) Fingerprint minutiae, and singular point, (b) Feature extracted from thinned image (feature map), (c) The minimized image of the fingerprint feature map in original size (1/64), and (d) Fingerprint feature map (enlarged 8 times).

As this network process on all pixels of input image In order to speed up the process during the identification and verification phases, the dimension of the encoded image is minimized, as shown in Fig. 8. It is done by considering a few important features instead of the whole pixels of a fingerprint. This can impressively decrease the image size and feature map applied to the network and consequently the FNN processing time.

Preparing the features, we will classify the pattern images of each individual in one main class and apply them to the FNN upon main class files (from 1 to N).

5 FNN Results for Fingerprints

According to Table 1, we acquired an average number of 100 fingerprint images of different shapes from the data base, NIST 4. We applied the processed images to the FNN in the form of 10 subset patterns (i_m, i= 1 to 10) of main classes (main class j, j= 1 to N).

Table 1 The ten main groups used as training patterns of FNN.

Training groups	Applied patterns to train the FNN
Subject no.1	10 tended arch fingerprint with a core
Subject no.2	10 radial loop fingerprint with a core
Subject no.3	12 ulnar loop fingerprint with a core
Subject no.4	12 whorl fingerprint with a central core (whorl)
Subject no.5	10 tended arch fingerprint with a core
Subject no.6	8 ulnar loop fingerprint with a core
Subject no.7	10 whorl fingerprint with two axial core
Subject no.8	8 whorl fingerprint with two axial core
Subject no.9	10 radial loop fingerprint with a core
Subject no.10	12 ulnar loop fingerprint with a core

Also we implemented our database by using a high precision system. The fingerprint images of different individuals are acquired in 176×224 pixels with 307 dpi resolutions. After the primary process, they are converted to the feature maps of 22×28 pixels, ready to be applied to the adaptive FNN. To have a common mode and flexible system, the key parameters of the FNN should be determined. To reach an optimized answer for a typical pattern, a trial and error approach is used. The parameters have been changed for a specified input, and the results of the most acceptable outputs are reported.

5.1 Decision Threshold or Dissimilarity Factor (T_f)

T_f is called the similarity threshold which quantifies the difference between the input pattern and the learned patterns. If amount of dissimilarity between the input and the learned pattern is less than T_f, the network will assume both patterns similar, otherwise the input pattern will be considered as a new one. The effect of T_f is shown in Table 2.

As shown in Table 3, by increasing T_f , the network finds more patterns in the main group similar to the input pattern. Consequently the number of subclasses will decrease and vice versa. It should be considered that inappropriate increasing of T_f will decrease the precision of recognition and classification of the FNN.

5.2 Fuzzification Parameter (β)

This parameter determines the effect of pixel on the Fuzzy Neurons. We use a weight function called W, to fuzzify input patterns,

$$w[m,n] = \exp[-\beta^2(m^2+n^2)] \\ m = -(N_1-1) \text{ to } (N_1-1)\ ,\ n = -(N_2-1) \text{ to } (N_2-1) \tag{22}$$

A 3-D plot of this function is shown in Fig. 9 for different values of β. The amount of β will affect on the sharpness of the lens like w function.

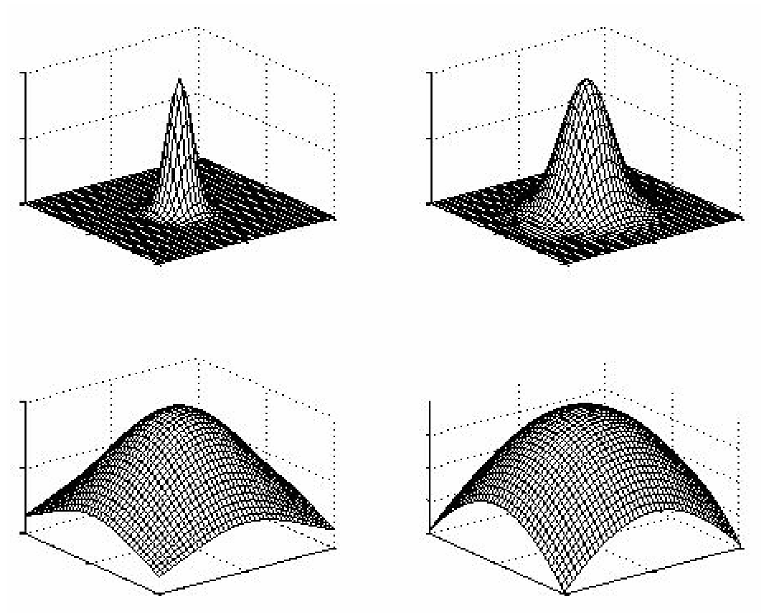


Fig. 9 w lens like function for different values of β respectively from right to left and up to down (0.6, 0.1, 0.3, 0.05).

The amount of β will affect on the sharpness of the lens like w function.

The Eq. (23) for computing β is proposed, from which the boundary values of β are determined;

$$0.5 = \exp[-\beta^2(\delta_x^{\ 2} + \delta_y^{\ 2})] \qquad (23)$$

where σ_x and σ_y are respectively the largest shifted pixels in x and y directions.

Table 4 shows the effect of β on the recognition rate of similar patterns appointed from different training sets of Table 1.

By comparing Table 4(b) and 4(c), we can find that despite of decreasingβ, the found sub-classes do not decrease in several rows like 5th, 9th and 9th unlike other ones. So decreasing β dose not always increases the number of recognized similarities. Actually it depends on the type and position of new points taking part in the matching procedure

Table 2 Similarity rate of FNN trained by Fingerprint sets of table 1 with different T_f.

(a) $(\alpha = 2.5 \quad \beta = 0.2 \quad Tf = 0.1)$.

Subject no.	Pattern Numbers		Found Similarities										
	full	Sub class	1	2	3	4	5	6	7	8	9	10	11
1	10	9	1	2	3	4	5	6,9	7	8	10	-	-
2	10	7	1,4	2,6,7	3	5	8	9	10	-	-	-	-
3	12	10	1	2	3	4,7	5	6,8	9	10	11	12	-
4	12	10	1,4	2	3	5	6,10	7	8	9	11	12	-
5	10	8	1	2	3	4	5,8,10	6	7	9	-	-	-
6	5	5	1	2	3	4	5	-	-	-	-	-	-
7	5	5	1	2	3	4	5	-	-	-	-	-	-
8	7	7	1	2	3	4	5	6	7	-	-	-	-
9	5	5	1	2	3	4	5	-	-	-	-	-	-
10	12	11	1,3	2	4	5	6	7	8	9	10	11	12

(b) ($\alpha = 2.5 \quad \beta = 0.2 \quad Tf = 0.2$) **.**

Subject no.	Pattern numbers		Found Similarities							
	full	Sub class	1	2	3	4	5	6	7	8
1	10	6	1,2	3	4,6,9	5,7	8	10	-	-
2	10	5	1,4	2,5,6,7	3	8,9	10	-	-	-
3	12	8	1,9	2	3	4,7	5	6,8,12	10	11
4	12	8	1,2,4,6,10	3	5	7	8	9	11	12
5	10	4	1	2	3,8	4,5,6,7,9,10	-	-	-	-
6	5	4	1	2	3,5	4	-	-	-	-
7	5	4	1,3	2	4	5	-	-	-	-
8	7	5	1	2	3,4	5,6	7	-	-	-
9	5	4	1	2,3	4	5	-	-	-	-
10	12	7	1,3,10	2	4,5,12	6	7,8	9	11	-

Table 3 Sub classification rate based on similarity of FNN trained by subject set of Table 1 with different Tf ($\alpha = 2.5$ and $\beta = 0.2$).

Subject no.	Total number of training pattern	Founded Subclasses		
		Tf=0.1	Tf=0.2	Tf=0.3
1	10	9	6	3
2	10	7	5	4
3	12	10	8	5
4	12	10	8	6
5	10	8	4	4
6	5	5	4	3
7	5	5	4	3
8	7	7	5	5
9	5	5	4	4
10	12	11	7	5

Table 4 Recognition rates of similar patterns trained by the fingerprints of table 1 for different values of β .

(a) ($\alpha = 2.5 \quad Tf = 0.2 \quad \beta = 0.3$) **.**

Subject no.	Pattern Numbers		Found Similarities									
	full	Sub class	1	2	3	4	5	6	7	8	9	10
1	10	8	1	2,4	3	5	6,9	7	8	10	-	-
2	10	7	1,4	2,6,7	3	5	8	9	10	-	-	-
3	12	9	1	2	3	4,7	5	6,8,12	9	10	11	-
4	12	10	1,4	2	3	5	6,10	7	8	9	-	-
5	10	8	1	2	3	4,10	6	7	5,8	9	-	-
6	5	4	1	2	3,5	4	-	-	-	-	-	-
7	5	4	1,3	2	4	5	-	-	-	-	-	-
8	7	7	1	2	3	4	5	6	7	-	-	-
9	5	5	1	2	3	4	5	-	-	-	-	
10	12	10	1,3	2	4,7	5	6	8	9	10	11	12

(b) ($\alpha = 2.5$ $Tf = 0.2$ $\beta = 0.15$) .

Subject no.	**Pattern Numbers**		**Found Similarities**						
	full	Sub class	1	2	3	4	5	6	7
1	10	3	1,2,3,5,10	6,7,9	4,8	-	-	-	-
2	10	4	1,2,4,6,7	3,5	8,9	10	-	-	-
3	12	5	1,9	2,3	4,7	5,6,8,11,12	10	-	-
4	12	7	1,2,4,6,10	3	5,9	7	8	11	12
5	10	4	1	2	3,8	4,5,6,7,9,10	-	-	-
6	5	3	1,4	2	3,5	-	-	-	-
7	5	3	1,2,3	4	5	-	-	-	-
8	7	5	1	2,4	3	5,6	7	-	-
9	5	4	1	2,3	4	5	-	-	-
10	12	6	1,3,10	2	4,5,12	6,9	7,8	11	-

(c) ($\alpha = 2.5$ $Tf = 0.2$ $\beta = 0.1$)

Subject no.	Pattern Numbers		Founded Similarities				
	full	Sub class	1	2	3	4	5
1	10	1	Whole patterns	-	-	-	-
2	10	2	1,2,3,4,5,6,7	8,9,10	-	-	-
3	12	4	1,4,7,9	2,3	5,6,8,11,12	10	-
4	12	5	1,2,4,6,10	3,5,9	7,8	11	12
5	10	4	1	2	3,5,8	4,6,7,9,10	-
6	5	2	1,4	2,3,5	-	-	-
7	5	2	1,2,3,4	5	-	-	-
8	7	4	1	2,3,4	5,6	7	-
9	5	4	1	2,3	4	5	-
10	12	3	1,2,3,4,5,10,12	6,7,8,9	11	-	-

From Table 5, it is observed that the recognition process shows more sensitivity toward the certain values of β. The domain of our filter will expand, if β becomes smaller.

Therefore, more adjacent points are considered in matching and recognition process.

Naturally, if these points become closer to each other, larger amounts of β (i.e. less expansion for the filter) can be considered. We need smaller values of β along with larger expansion of the filter to recognize similar adjacent points.

5.3 Similarity Evaluation Range (α)

The parameter α states the width of function used to determine the degree of similarity between the various points of input, and learned patterns in the data base. Also it states the range of similarity between the input, and the learned patterns of the FNN. Table 6 shows the results of applying various amount of α on the fingerprints. If α=1, then each pattern of a special main class is recognized as a separate sub-class. For this case, the width of triangular membership function is not sufficient to find similar features and to determine similarity of images.

Table 5 Sub classification rate based on similarity of FNN trained by subject set of table 1 with different β ($\alpha = 2.5$ and $Tf = 0.2$).

Subject no.	Total number of training pattern	Founded Subclasses			
		$\beta = 0.1$	$\beta = 0.15$	$\beta = 0.2$	$\beta = 0.3$
1	10	1	3	6	8
2	10	2	4	5	7
3	12	4	5	8	9
4	12	5	7	8	10
5	10	4	4	4	8
6	5	2	3	4	4
7	5	2	3	4	4
8	7	4	5	5	7
9	5	4	4	4	5
10	12	3	6	7	10

Table 6 Recognition rate of similar patterns trained by the fingerprints of table 1 for different values of α.

(a) ($\beta = 0.2$ $Tf = 0.2$ $\alpha = 1$) .

Subject no.	1	2	3	4	5	6	7	8	9	10
Original patterns	10	10	12	12	10	5	5	7	5	12
Founded sub classes	10	9	12	11	10	5	5	7	5	12

(b) ($\beta = 0.2$ $Tf = 0.2$ $\alpha = 1.5$) **.**

Subject no.	Pattern Numbers		Found Similarities										
	full	Sub class	1	2	3	4	5	6	7	8	9	10	11
1	10	10	1	2	3	4	5	6	7	8	9	10	-
2	10	7	1,4	2,6,7	3	5	8	9	10	-	-	-	-
3	12	10	1	2	3	4,7	5	6,8	9	10	11	12	-
4	12	10	1,4	2	3	5	6,10	7	8	9	-	-	-
5	10	8	1	2	3	4	5,8,10	6	7	9	-	-	-
6	5	5	1	2	3	4	5	-	-	-	-	-	-
7	5	5	1	2	3	4	5	-	-	-	-	-	-
8	7	7	1	2	3	4	5	6	7	-	-	-	-
9	5	5	1	2	3	4	5	-	-	-	-	-	-
10	12	11	1,3	2	4	5	6	7	8	9	10	11	-

Table 7 Sub classification rate based on similarity of FNN trained by subject set of table 1 with different α ($\beta = 0.2$ and $Tf = 0.2$).

Subject no.	Total number of training pattern	Founded Subclasses			
		$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 2.75$
1	10	10	10	8	5
2	10	9	7	7	5
3	12	12	10	9	6
4	12	11	10	10	8
5	10	10	8	8	4
6	5	5	5	4	4
7	5	5	5	4	4
8	7	7	7	7	5
9	5	5	5	5	4
10	12	12	11	10	7

Table 8 The Similarity patterns founded with the trained ones for the subject 1 (Network False Acceptance).

(a) $(\alpha = 2.5, \beta = 0.2, (Tf = 0.3))$.

Subject No. 1 / Sub Classes of subject	1	2	3	4	5	6	7	8	9	10
1	1	1	1	2	1	2	2	3	2	1
2	2	*	3	*	2	*	1	*	*	2
3	4	4	4	*	2	*	3	*	*	2
4	*	*	*	*	*	*	1	*	*	*
5	*	3	3	*	*	*	*	*	*	*
6	*	*	2	*	*	*	*	*	*	1
7	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*
9	1	1	1	*	*	*	*	*	*	2
10	2	2	1	2	2	2	*	2	2	*
False Acceptance	40%	40%	60%	10%	30%	10%	30%	10%	10%	50%
False Average	29%									

(b) $(\alpha = 2.5, \beta = 0.2, (Tf = 0.2))$.

Subject No. 1 / Sub Classes of subject	1	2	3	4	5	6	7	8	9	10
1	1	1	2	3	4	3	4	5	3	6
2	2	*	*	*	2	*	1	*	*	4
3	*	4	4	*	2	*	3	*	*	2
4	*	*	*	*	*	*	1	*	*	*
5	*	3	3	*	*	*	*	*	*	*
6	*	*	2	*	*	*	*	*	*	1
7	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*
9	1	*	*	*	*	*	*	*	*	2
10	2	2	1	*	2	2	*	2	*	*
False Acceptance	30%	30%	40%	0%	30%	10%	30%	10%	0%	40%
False Average	22%									

We see in Table 7, as we expected, by increasing α the number of found subclasses will decrease. However, the variation is not the same for all subjects and patterns, for example the variation of the 1st and 3rd subjects is more than the 8th and 9th.
In fact the width of similarity membership function varies for different fingerprint patterns and is affected by the position of features in relation to each other. A small value of α contains many features, if the patterns are so close to each other.
If we choose a great value of α , to enhance reparability as much as possible, T_f should be considered as well (i.e. during training, T_f has to be small). Table 8 shows a sample of processed algorithm for calculating of false acceptance for subject 1, two values of T_f have been used to show the effectiveness of this factor for false acceptance. α and T_f are to be sufficiently small, and β is great enough to enable the FNN to distinguish all separated training patterns.
For patterns with one singular point, depending on their type and parameter values, the average amount of false acceptance was 20% to 30% and the false rejection ranged from 15% to 20%.
In double singular point patterns, false acceptance varied from 10% to 15% and false rejection fluctuated 10%.

6 Conclusions

The network proposed in reference (Kwan and Cai 1994) was only a clustering network with unsupervised learning algorithm which user didn't have any direct control on found clusters. But, the implemented FNN is a fully classified system with the supervised learning algorithm and we can define the number of classes at the beginning of the training stage. The introductory network introduces a new method to classify of patterns. It classifies them by the shape of the input pattern that may be pattern's real shape or its feature illustration dealing to networks' parameters. In the case of increasing the number of patterns, the number of network's processing points should be increased (i.e. number of features which is extracted from patterns or image's dimension should be increased). To get the highest accuracy, all of the points enter to procession. So the classifying capability for many patterns is provided by considering the amount of image points, characters, point distance, characters type and their differences to set the network's parameters. Therefore, by increasing β and decreasing α and $T_{f,}$ we can improve the accuracy of recognition.
The network is suitable for any kind of pattern classification problems by fixing the defined parameters, related to type and dimensions of patterns.

We used fingerprints as the most complicated pattern for assessment of the network. The results for the fingerprints are reasonable, so it will be clear that for the other types of the patterns with less complexity, for example OCR, we will achieve much better results. By increasing the network accuracy and capability, we can apply gray scale images to the network instead of binary images. This is one of the major points that we can use the other types of the features for applying to the network such as direction, type, number and etc. as feature coding which we used for this research.
The accuracy of the proposed network is increased by adding the classification layer; this layer makes the user to quit from determining of the network parameters by trying and faulting for the appropriate number of the classes. Because, the number of the main classes is defined at first for us and the network, we don't have any unpredictable extra class in the output.
We modify and improve Kwan and Cai UFNN and design a precise and comprehensive supervised learning algorithm for it. To show our new SFNN capability we used fingerprint pattern and our testing result is more considerable than early UFNN, especially when it can't find any similarity for several pattern of one class, our network classifies them in one class at least.
After all of the above we haven't any claim that our SFNN has better result among fingerprint matching methods but we show that, it can have better performance than some inflexible Neural Networks and primary UFNN in classification and identification for flexible patterns.

References

[1] Nadler N., Psmith E., *Pattern Recognition Engineering,* John Wiley, 1998, chapter 1.

[2] Lau C., *Neural Networks: Theoretical, Foundations and Analysis,* Piscataway, NJ, IEEE press, 1992. chapter 1.

[3] Sanchez-Sinencio E. and Lau C., *Artificial Neural Networks: Paradigms, and Hardware Implementations,* Piscataway, NJ, IEEE Press, 1992, chapter 2.

[4] Fukumi M., Omatu S., and Takeda F., "Rotation-invariant neural pattern recognition system with application to coin recognition," *IEEE Trans. on Neural Networks*, Vol. 3, No. 2, pp. 272-279, 1992.

[5] Perantonis S. J. and Lisoba P. J. G., "Translation, rotation, and scale invariant pattern recognition by high order neural networks and moment classifiers," *IEEE Trans. on Neural Networks*, Vol. 3, No. 2, pp. 241-251, 1992.

[6] Bezdek J. C., and Pal S. K., *Fuzzy Models for Pattern Recognition,* Piscataway, NJ, IEEE Press, 1992.

[7] Kandel A., *Fuzzy Techniques in Pattern Recognition,* John Wiley, New York, 1982.

[8] Abraham A. and Nath B., "Evolutionary Design of Neuro-Fuzzy Systems-A Generic Framework," *4th Japan Australian joint Workshop on Intelligent and Evolutionary Systems*, Japan, Vol. 1, No. 4, pp. 560-568, Nov. 2000.

[9] Abraham A., "Neuro-Fuzzy Systems: State of the Art Modeling Techniques," *6th International Conference on Artificial and Natural Neural Networks*, IWANN 2001, Granada, Germany, June 2001.

[10] Gazanfary A., Lucas C., "Designing expert pattern recognition system by fuzzy neural network," *Proc. 1st Iranian Conference on Electrical Eng.*, Amirkabir University, Vol. 2, pp. 326-337, 1993.

[11] Yamakawa T., and Tomoda S., "A fuzzy neuron and its application to pattern recognition," *Proc. 3rd Int. Fuzzy System Associate Congress*, Japan, pp. 30-38, 1989.

[12] Zhou R. W., and Quek C., "A pseudo outer-product fuzzy neural network," *Proc. ICNN95*, pp. 957-962, 1995.

[13] Nikian A., *Identification of Persian Signatures by Using Fuzzy Neural Networks,* Ph.D. Thesis, Iran University of Science & Technology, 2000.

[14] Kwan H. K., and Cai Y., "A fuzzy neural network and it's application to pattern recognition," *IEEE Trans. on Fuzzy Systems*, Vol. 2, No. 3, pp. 185-193, Aug. 1994.

[15] Menhaj M. B., Azizzadeh H., "Pattern Classification and Recognition Using a Cascaded Fuzzy Feed-Forward Neural Net," *Amirkabir journal,* No. 31, pp. 196-205, 1999.

[16] Rouhani M., Menhaj M. B., "Fuzzy Neural Net with Application in Distinct Farsi Character Recognition," *Fifth ICEE*, May 7-9, 1997.

[17] Nikhil R. P., Gautam K. M., and Eluri V. K., "Comments on fuzzy neural network and its application to pattern recognition," *IEEE Trans. on Fuzzy Systems*, Vol. 7, No. 4, pp. 479-480, Aug. 1999.

[18] Rao C. K., and Black K., "Finding core point in fingerprint," *IEEE Trans. on Computer*, Vol. C-27, No. 1, 1978.

[19] Hariri M., and Shokouhi Sh. B., "Fingerprints classification and identification by a novel fuzzy neural network," *Proc. HART2004*, University of Fukui, Japan, Vol. 1, pp. 277-282, 2004.

[20] Hariri M., and Shokouhi Sh. B., "Feature Extraction and Classification of Fingerprints using a Novel Fuzzy Neural Network," *WSEAS Int. Conf. on Signal, Speech and Image Processing*, Greece, Vol. 1, pp. 112-117, 2005.

[21] Mohamed S. M., and, Nyongesa H. D., "Automatic Fingerprint Classification system Using Fuzzy Neural Techniques," *IEEE,* 2002.

[22] Palumba W., Swamination P., Srihani S. N., "Document Image Binarization, Evaluation of

Algorithms," *SPIE*, *Applications of Digital Image processing* IX, Vol. 697, pp. 278-285, 1995.

[23] Mehtre B. M., Chatterjee B., "Segmentation of fingerprint Images, A Composite Method," *Pattern Recognition*, Vol. 22, No. 4, pp. 381-385, 1989.

[24] Gorman L. O., and, Nickerson J. V. "On Approach to Fingerprint Filter Design," *Pattern Recognition*, Vol. 22, No. 4, pp. 29-38, 1989.

[25] Srinivasan V. S., and Murthy N. N., "Detection of singular points in fingerprint images," *Pattern Recognition*, Vol. 25, No. 2, pp. 139-153, 1992.

[26] Fairhurst M. C., *Computer vision for Robotic Systems,* Printice Hall Inc, U.K., 1988.

M. Hariri received his B.Sc. and M.Sc. degrees with honors in Electronic Engineering from Amirkabir University of Technology (AUT) and Iran University of Science and Technology (IUST) in 2001 and 2004 respectively. Currently he is a Ph.D. student at Iran University of Science and Technology. He is a member of IEEE. His research interests include image processing, pattern recognition, intelligence network and its applications in pattern recognition & biometrics.

Sh. B. Shokouhi received his B.Sc. and M.Sc. in Electronics in 1986 and 1989 respectively both from the Department of Electrical Engineering of Iran University of Science and Technology (IUST). He received his Ph.D. in Electrical Engineering in 1999 from School of Electrical Engineering, University of Bath, England. Since 2000, he has worked as an assistant professor in the Department of Electrical Engineering (IUST). Dr B. Shokouhi is a member of Iranian Scientists of Electrical Engineering (ICEE), Mechatronics Engineering (ISME) and Machine Vision and Image Processing (MVIP). His research interests include Machine Vision algorithms & hardware implementations, pattern recognition and hybrid human Identification Systems.

N. Mozayani received his B.Sc. degree in Electrical Engineering (Computer Hardware) from Sharif University of Technology (Tehran, IRAN); M.Sc. degree in Telematics and Information Systems from Supelec (France) and Ph.D. degree in Informatics from University of Rennes 1 (Rennes, France). He is currently Assistant Professor in the Department of Computer Engineering at Iran University of Science & Technology. His research interests include soft computing and computer networks.