A Modular Neural Network with Fuzzy Response Integration for Person Identification Using Biometric Measures

Magdalena Serrano and Patricia Melin

Tijuana Institute of Technology, Tijuana México
epmelin@hafsamx.org

Abstract. This paper describes an intelligent system for person identification with biometric measures such as signature, fingerprint and face. We describe the neural network architectures used to achieve person identification based on the biometrics measures. Simulation results show that the proposed method provides good recognition. Fuzzy integration of the three modules is tested on a single computer and also in a distributed environment.

1 Introduction

At the moment, systems based on biometric recognition have gained importance in applications that require the identification of users or restricted access. Compared with conventional methods based on using keys, we have the advantage that the biometric features may not be provided, copied or stolen.

These kinds of systems are usually easy to maintain. A biometric system is essentially a pattern recognition system that operates in the following manner: capturing a biometric measure, a set of features are extracted and compared with another group the features.

Biometric identification techniques are very diverse, since any element of a person is potentially usable as a biometric measure. Even with the diversity of existing techniques, to develop a biometric identification system, it is an entirely separate scheme from the technique used. The most used biometric measures are: fingerprint, iris, voice, signature, face, ear, hand geometry, vein structure, retina, etc.

The need for a way to identify the human being in a unique way, has led researchers to implement a wide range of methods.

Until today biometric methods have been implemented using different devices for the creation of patterns and generate the code that identifies the individual biometric measures. This is why, in this work we consider one of the most used biometric measures throughout history, which is the fingerprint, and other ones such as the face and signature.

O. Castillo et al. (Eds.): Evolutionary Design of Intelligent Systems, SCI 257, pp. 159–183.
springerlink.com

2 Neural Networks

A neural network is a model to perform a computational simulation of parts of the human brain using replication behavior in small-scale patterns that it performs for producing results from the perceived events.

An artificial neural network (ANN), often just called a "neural network" (NN), is a mathematical or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning [2].

1. Biological neural networks are made up of real biological neurons that are connected or functionally related in the peripheral nervous system or the central nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.
2. Artificial neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real, biological nervous system is highly complex and includes some features that may seem superfluous based on an understanding of artificial networks.

2.1 Structure of an Artificial Neural System

The artificial neural system is composed by several components that are necessary to structure the system. In figure 1 we show these components: the neuron, layers, and networks.

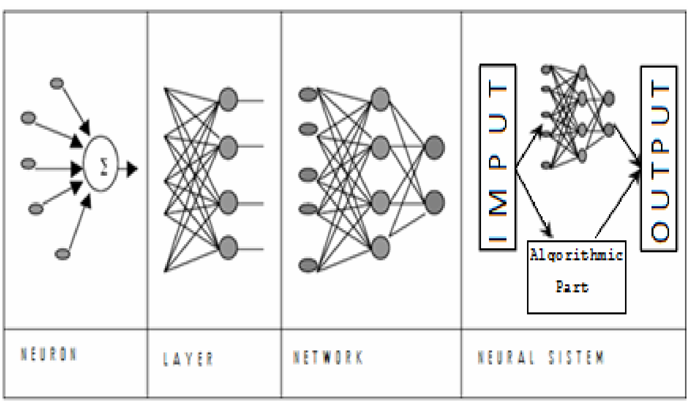

Fig. 1. Hierarchical structure of a system based on artificial neural networks

2.2 Modular Neural Networks

A modular neural network is a neural network characterized by a series of independent neural networks moderated by an intermediary. Each independent neural network serves as a module and operates on separate inputs to accomplish some subtask of the task the network is intended to perform.

The intermediary takes the outputs of each module and processes them to produce the output of the network as a whole. The intermediary only accepts the modules' outputs—it does not respond to, nor otherwise signal, the modules. Also the modules do not interact with each other [4].

The advantage is that if the model supports naturally a breakdown into more simple functions, the application of a modular network translates into faster learning. Each module can be built differently, in a way that meets the requirements of each subtask. A modular neural network can be represented by the scheme shown in figure 2.

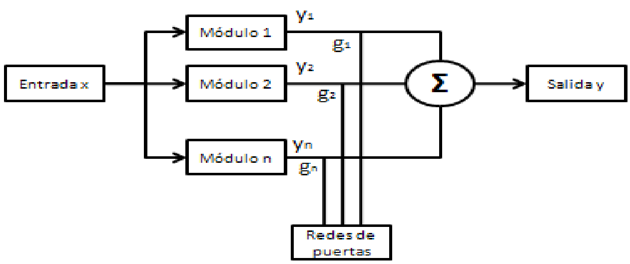

Fig. 2. A modular neural networks architecture.

3 Characteristics of a Biometric Measure

A biometric measure is a feature that can be used to make an identification. Whatever the measure, it must meet the following requirements:

1. Universality: means that anyone should have that characteristic.
2. Uniqueness: the existence of two people with identical characteristics has a very small probability.
3. Permanent Characteristics: the characteristic does not change over time
4. Quantification: the characteristic can be measured in a quantitative form.

3.1 Architecture of a Biometric System for Personal Identification

A biometric system has three basic components: The first is responsible for the acquisition of any analog or digital biometric feature of a person, such as the acquisition of a fingerprint image using a scanner. The second handles the compression, processing, storage and comparison of data acquired with the stored data. The third component provides an interface to applications on the same or another system. In figure 3 we show the phases of a biometrical identification system.

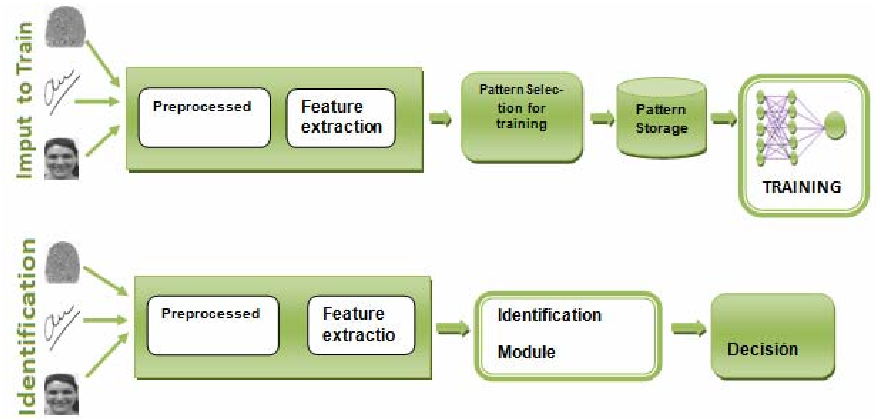

Fig. 3. Phases of a biometrical identification system.

4 Problem Statement and Proposed Method

The problem is to develop a hybrid system for person identification using parallel processing implementation determined by the biometric measures. We must develop a module for each of the biometric measures in the neural network architecture to implement:

- Fingerprint.
- Signature.
- Face.

Starting from the 3 above-mentioned modules it is important to make the unification of the 3 biometric measures to make the system obtain a higher percentage of identification, which aims at making the integration of the 3 systems into one, as shown in Figure 4.

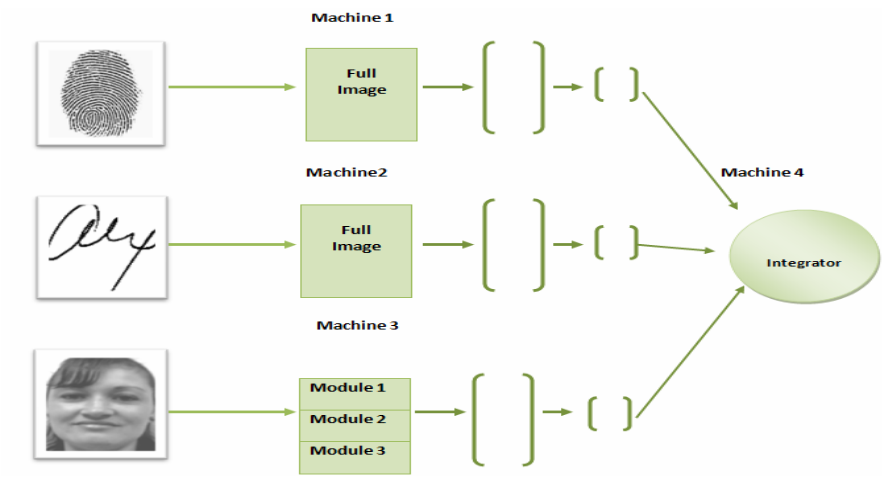

Fig. 4. Proposed scheme for the development of the system.

The implementation was done in 4 core computers, i.e, with 4 computers with 4 processors to activate 16 processors in parallel.

4.1 Biometric Databases

The databases were obtained from students and professors of the Master in science in computer science from Tijuana Institute the Technology.

The database consists of 10 Samples of signature, fingerprint and face, taken from 30 people, giving a total of 300 samples. We used for training the first 7 images and leaving the last 3 for identification. In total we trained the network with 210 images and 90 are left for the identification for face, fingerprint and signature.

The databases are shown in figures 6, 9, 10 and 11.

4.1.1 Signature Database

For the normalization of this database the images were cut to limit the signature, this produced a resizing of the images. We also applied Wavelets and the Sobel operator to preprocess the image, as shown in Figure 5.

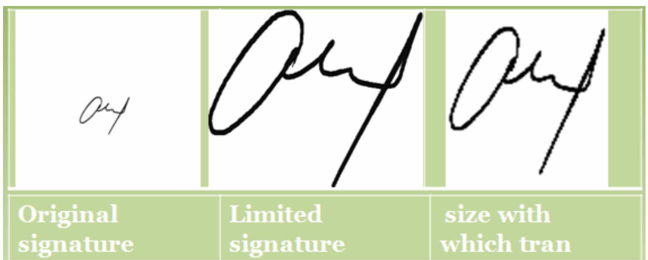

Fig. 5. Preprocessed database

Fig. 6. Database with examples of signatures.

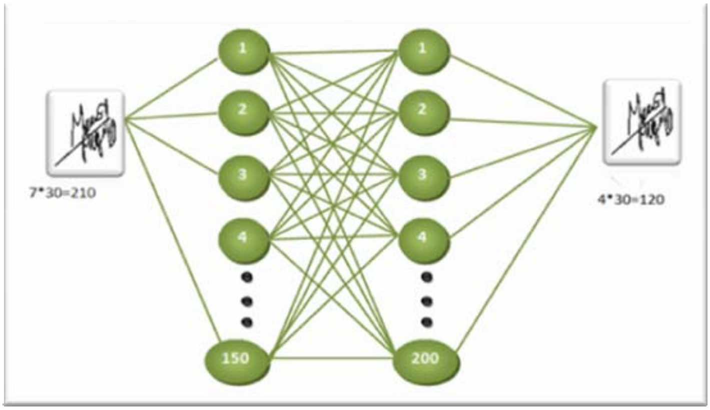

Fig. 7. Architecture of the net for signature.

After the preprocessing phase, the database can be visualized in Figure 6 with several examples of signatures .

The architecture for training the net is shown in Figure 7, were we have 2 hidden layers.

4.1.2 Fingerprint Database

For the normalization of the database, a cut of the standard image of the limitation of the fingerprint was performed, preprocessing was also performed, and the methods used for this are the following:

- Gradient magnitude
- Sobel edge masks
- Skeleton.
- Wavelets.

In figure 8 we show an example of fingerprint preprocessing. We show the gradient magnitude and skeleton methods applied to the fingerprint.

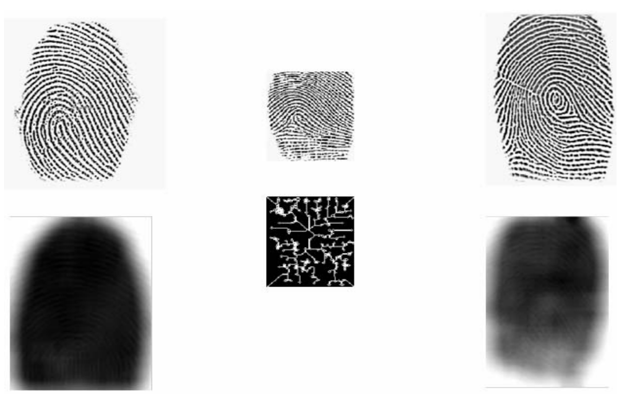

Fig. 8. Preprocessing of the fingerprint.

After preprocessing the database of images, they go in to the network with a dimension of [75 * 50] as shown in Figure 9.

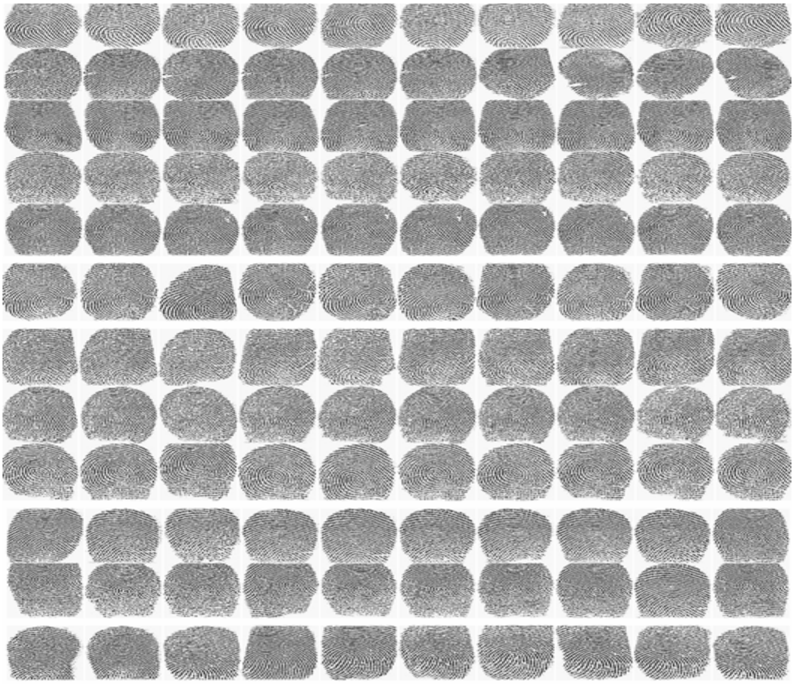

Fig. 9. Database of fingerprints

The architecture for training the net is shown in figure 10 and it has 2 hidden layers.

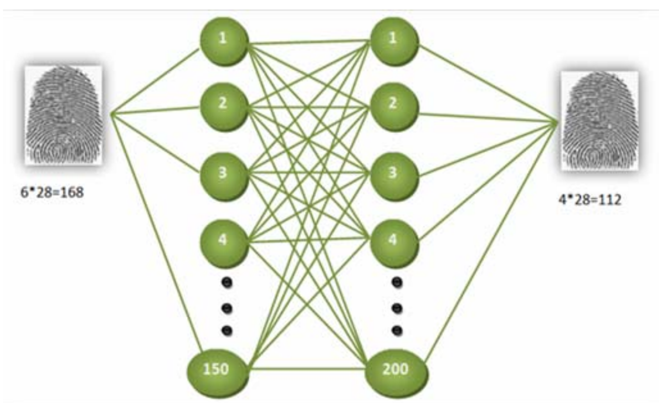

Fig. 10. Architecture of the neural network for fingerprint recognition.

4.1.3 Face Database

The face data base consist of a mixture of the ORL database and data acquired with students and professors of the institution, and has a different treatment for this reason, we first converted the images to gray tones and made a resizing to [92*112] in BMP format. The face database of the institution is shown en figure 11.

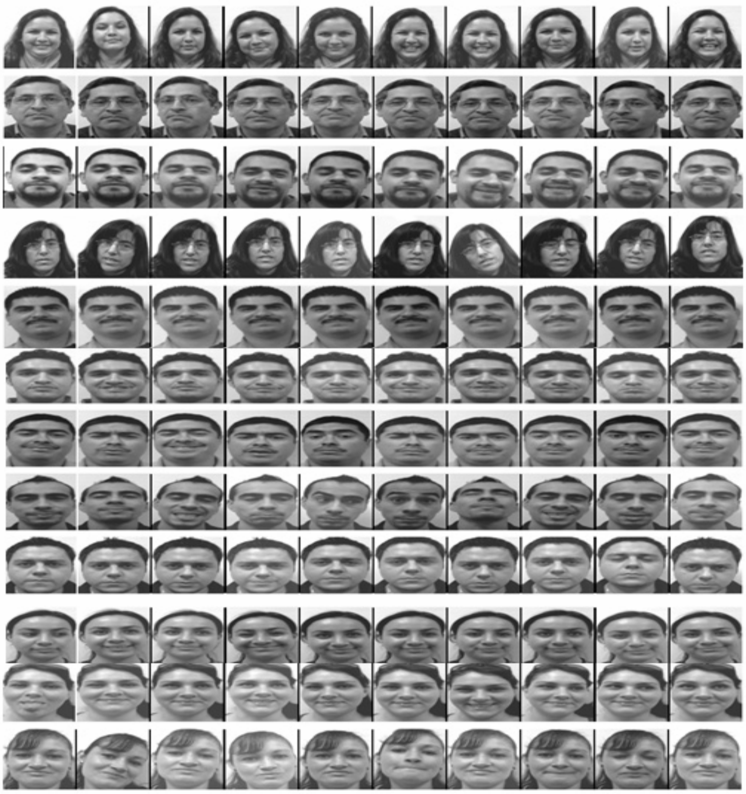

Fig. 11. Database of faces.

4.1.4 Design of the Neural Network Structure for Face Recognition

The number of neurons is allocated using an empirical expression created by Renato Salinas[8] and can be explained as follows:

- Input neurons: 2* (k+m), the activation function is a Tangent sigmoid.
- Hidden neurons: (k+m), the activation function is a Tangent sigmoid.
- Output neurons: the activation function is a Tangent logarithm sigmoid.

Where k is the number of individuals to train and m is the number of samples to train for each individual. In this case, these were 40 individuals for the ORL database , and 30 for the database of the institute and 7 samples therefore, k = 40, m = 7 and k = 30, m = 7.

Taking into account the previous data the architecture to train the modules is shown in Figure 12.

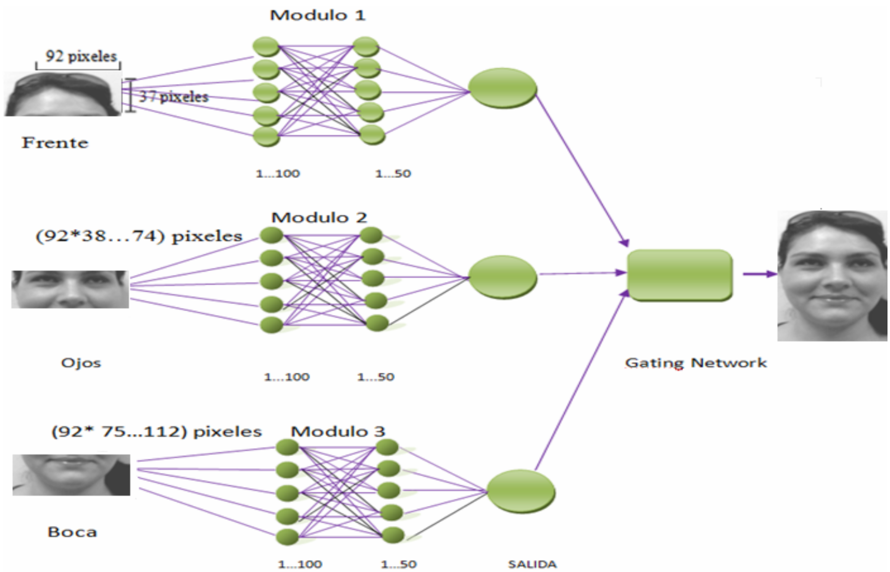

Fig. 12. Architecture of the neural network for face recognition.

5 Fuzzy Integration of Face, Signature and Fingerprint

We used a fuzzy logic module for the integration of the neural networks outpus. This method provides better performance in the subjective assignation of imputs from each of the individual networks;we show a model where one can see the integration of neural networks with fuzzy logic, so that the outputs of the neural network are processed by a fuzzy inference [11] mechanism, which can be seen illustrated in figure 13.

This phase consisted of a fuzzy system that can integrate the results of three different biometric measures (face, signature and fingerprint) and using this response as a result.

This system is designed with three input variables because each input belongs to biometric measure,this will increase if both the input variables, and the granularity of the number of rules would increase in a quantitative way.

Now we describe the design of the fuzzy integrator, since the granularity of the input variables, which are given as low, medium and high and the result is given by module1, module2 and module3 the we have the architecture shown in Figure 14. This type of fuzzy inference is of Mamdani form, which uses both the inputs

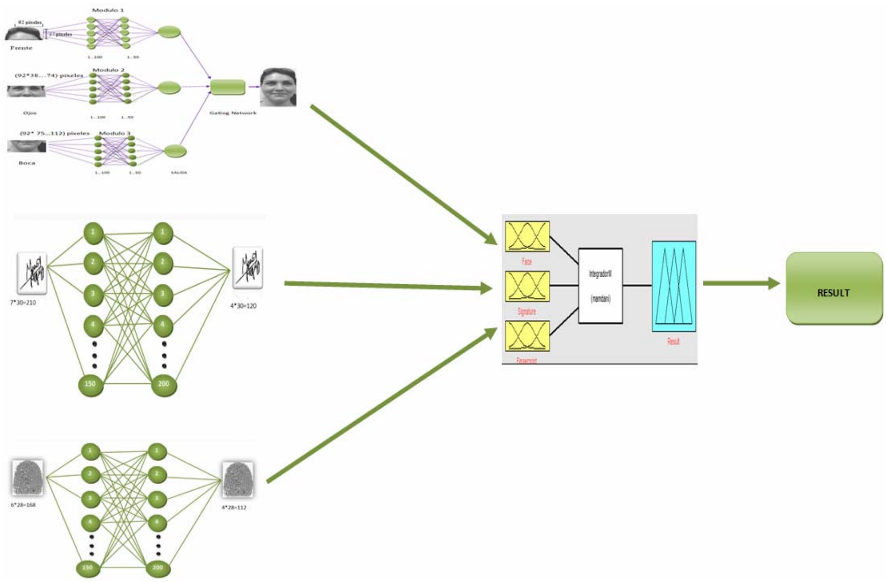

Fig. 13. Architecture of integration using fuzzy logic for integration of three biometrics measures.

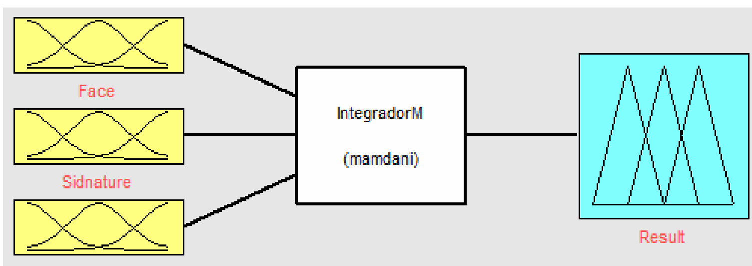

Fig. 14. Architecture of fuzzy Inference system.

as outputs of linguistic form. The gaussian membership functionare defined by the following equation.

$$\text{Gaussian}\ (x;\, c,\, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{1}$$

For the granularity of the variables we used the “medium”,” low” and “high” membership functions, where these three linguistic values, distributed between 0 and 1.

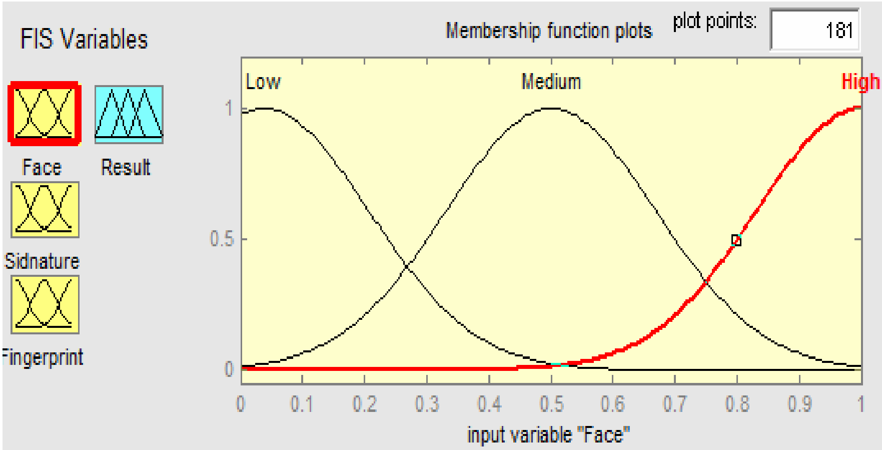

Fig. 15. granulation of the variable membership Face.

As shown in figure15, the face input variable has 3 membership functions of Gaussian type, which have the following parameters:

Low: parameter that ranges from [0,0.5]
Media: parameter that ranges from [0,1]
High: parameter ranges from [0.5, 1]

The Gaussian membership function for each of the linguistic values is defined as follows:

$$\mu_{Low}=e^{\left(\frac{x-0.05704}{0.17}\right)} \tag{2}$$

$$\mu_{Medium}=e^{\left(\frac{x-0.050}{0.1699}\right)} \tag{3}$$

$$\mu_{High}=e^{\left(\frac{x-1}{0.1699}\right)} \tag{4}$$

The membership functions for the variables of the signature and fingerprint have the same granularity as the one given to the face Gaussians and is defined in the same way as shown in the following equations:

Signature:

$$\mu_{Low}=e^{\left(\frac{x-0.05704}{0.17}\right)} \tag{5}$$

$$\mu_{Medium}=e^{\left(\frac{x-0.050}{0.1699}\right)} \tag{6}$$

$$\mu_{High}=e^{\left(\frac{x-1}{0.1699}\right)} \tag{7}$$

Fingerprint:

$$\mu_{Low}=e^{\left(\frac{x-0.03704}{0.17}\right)} \tag{8}$$

$$\mu_{Medium}=e^{\left(\frac{x-0.050}{0.1699}\right)} \tag{9}$$

$$\mu_{High}=e^{\left(\frac{x-1}{0.1699}\right)} \tag{10}$$

The output is granulated in membership functions, which have a range from 0 to 1, the granularity of the output variable shown in Figure 16.

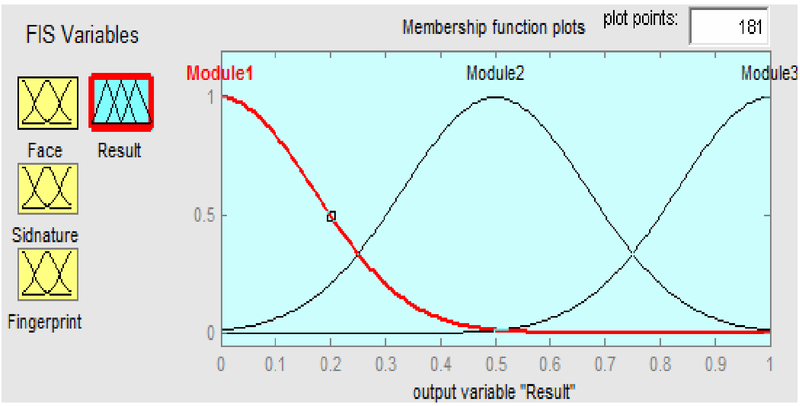

Fig. 16. Granulation of the output in the variable Result.

For the Gaussian outputs it is important to mention the ranges which are given as follows:

Module1: [0, 0.5]
Module2: [0, 1]
Module3: [0.5,1]

If we replace both the standard deviation and the mean in the equation of the Gaussian then we have:

$$\mu_{Module1}=e^{\left(\frac{x-6.939e-018}{0.1699}\right)} \tag{11}$$

$$\mu_{Module2}=e^{\left(\frac{x-0.050}{0.1699}\right)} \tag{12}$$

$$\mu_{Module3}=e^{\left(\frac{x-1}{0.1699}\right)} \tag{13}$$

Granulation of the output Variable is given as Modulo1, Modulo2, and Modulo3, which belong to the biometric face, fingerprint and signature respectively. This is important because when choosing a result response it is necessary to check which of the three modules is the one that has generated the correct answer. For this reason it is necessary that the rules cover all possible combinations for the solution of the problem.

Now we consider the optimization of the fuzzy inference system, the optimization can be in the type of system which can be Mamdani or Sugeno, and the membership function, that can be Gaussian, trapezoidal, triangles, etc.. In our case for the optimization, we use the 27 possible rules and leave the system with linguistic variables.

The rules were made taking into account all possible cases that may be present as shown in Table1.

Table 1. Rules for the fuzzy inference system.

Rules	**Face**	**Signature**	**Fingerprint**	**Result**
1	Low	Low	Low	Module1
2	Low	Low	Medium	Module2
3	Low	Low	High	Module3
4	Low	Medium	Low	Module2
5	Low	Medium	Medium	Module2
6	Low	Medium	High	Module3
7	Low	High	Low	Module2
8	Low	High	Medium	Module2
9	Low	High	High	Module2
10	Medium	Low	Low	Module1
11	Medium	Low	Medium	Module1
12	Medium	Low	High	Module3
13	Medium	Medium	Low	Module1
14	Medium	Medium	Medium	Module1
15	Medium	Medium	High	Module3
16	Medium	High	Low	Module2
17	Medium	High	Medium	Module2
18	Medium	High	High	Module2
19	High	Low	Low	Module1
20	High	Low	Medium	Module1
21	High	Low	High	Module1
22	High	Medium	Low	Module1
23	High	Medium	Medium	Module1
24	High	Medium	High	Module1
25	High	High	Low	Module1
26	High	High	Medium	Module1
27	High	High	High	Module1

Based on the fuzzy rules of table 1 we can simulate the performance of the fuzzy system, which is illustrated in figure 17.

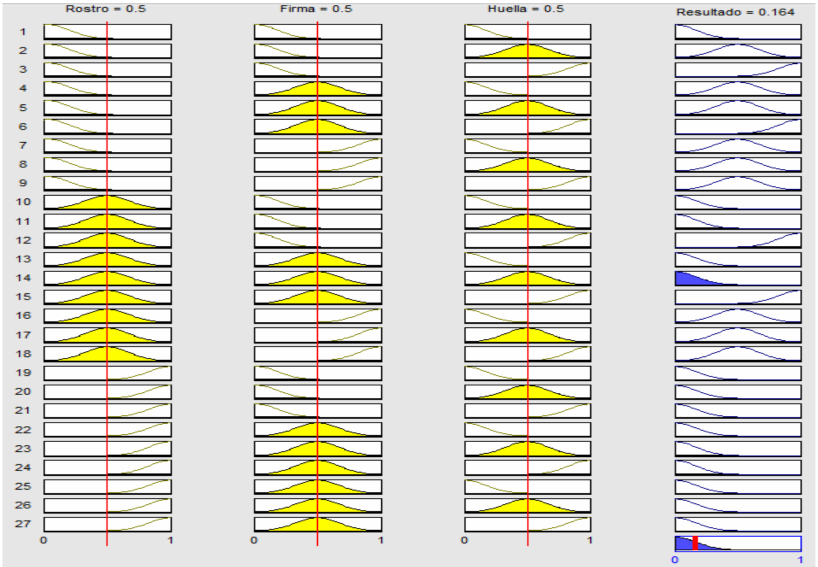

Fig. 17. Rule viewer

The general behavior of the fuzzy can be appreciated with non_linear surface generated by the fuzzy module. This is illustrated in three dimensions, with two biometric measures in each case.

Other fuzzy systems were made with triangular and trapezoidal membership functions, with a structure similar to the above mentioned inference system, which

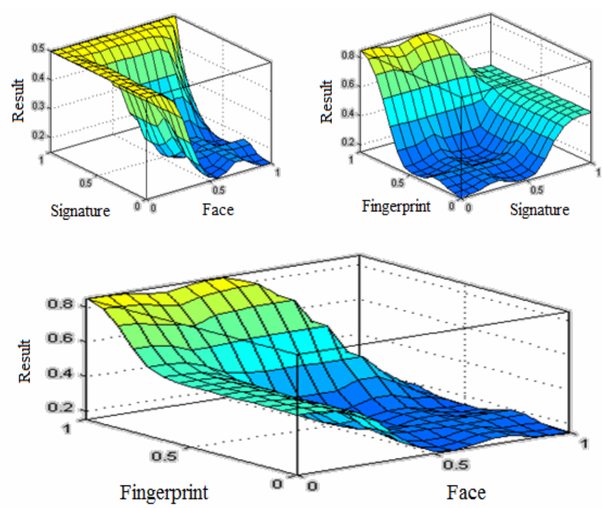

Fig. 18. Surface visor

is only referred to its realization, for the granulation with triangular membership functions we how use the following equation:

$$Triangle\,(x_i, a, b, c, d) max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{14}$$

In implementing the fuzzy inference system with a Mamdani-type model with trapezoidal membership function, we used the formula of trapezoids which establishes the granularity of the values of variables.

$$\text{Trapezoid}\,(x_i, a, b, c, d) max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}\right), 0\right) \tag{15}$$

The results of these fuzzy inference systems were similar to the case of Gaussians and we have decided to use that experience and those who have achieved better results in identification of persons

6 Parallel Processing

Distributed or parallel computing speeds up the MNN execution of a program through its division into fragments that can be run simultaneously, each on one processor. Thus a program running on "n" processors might execute n times faster than using a single processor.

We chose to assign a fixed IP to each computer, so, to control the order of the machines in the network according to how they are located, and in this way the only work that the switch has will be to transfer the information between the processors, so that the computers are configured with the following assignments:

192.168.1.	Module1
192.168.1.	Module2
192.168.1.	Module3
192.168.1.	Module4

Figure 19 shows how the groups were placed on their shelves containers. If more computers are added in the cluster it is only necessary to assign the following IP address.

6.1 Distributed Computing Toolbox and MATLAB Distributed Computing Engine (MDCE)

The mdce service ensures that all other processes are running and that it is possible to communicate with them. Once the mdce service is running, you can use the nodestatus command to obtain information about the mdce service and all the processes it maintains.

Fig. 19. Order of the machines

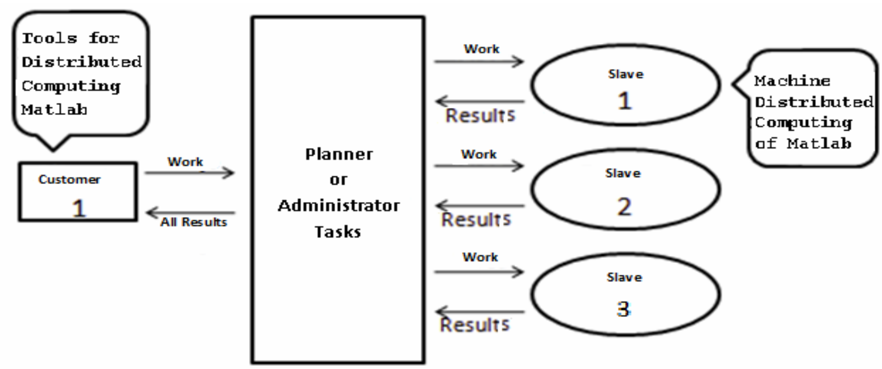

Fig. 20. Overview of Matlab as a computing distributed system

The distributed computing toolbox and MDCE allow us to coordinate and execute operations simultaneously on a cluster of computers to speed up execution of work (jobs) in MATLAB.

Figure 20 shows the interaction between machine, which uses the distributed computing toolbox to define jobs and tasks, and the MDCE. The planner can be

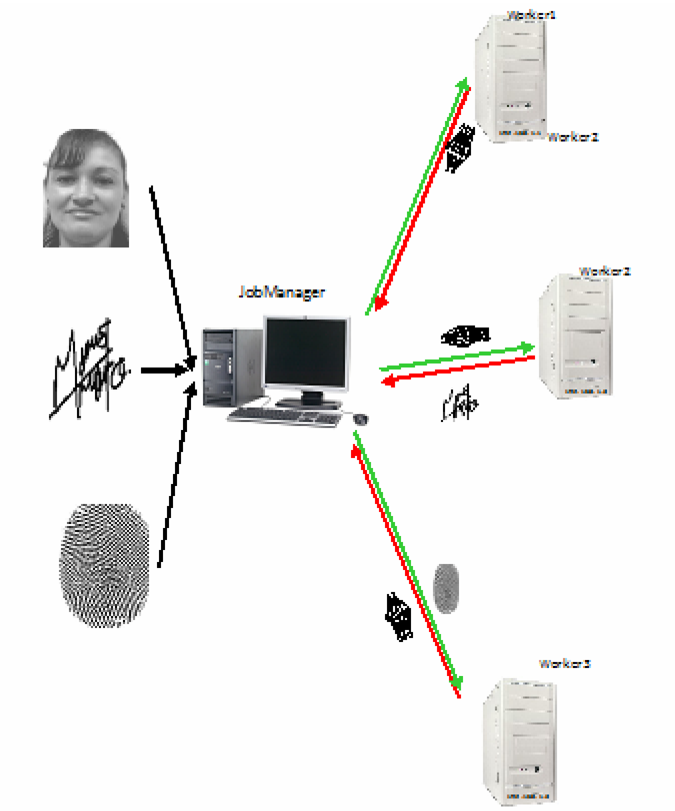

Fig. 21. Structure of the distributed algorithm in the Cluster

the manager of The Math Works jobs, included as part of MATLAB Distributed Computing Engine, a planner or a third party.

Management meetings conducted by the MDCE are shown in Figure 20. Processing requests of different clients are handled separately, a request at a time, for a single instant of time is carried out the task by the client 1,then 2 and so on for the n clients that can connect to the planner of tasks assigned to each period of time for processing.

6.2 Configuration Process

On each cluster node open a screen command line (MS-DOS) and type the following:

Cd Program files\ MATLAB\R2008a\Toolboox\distcomp\bin

1. Stop the execution of earlier versions of MDCE

   mdce stop

2. Install the new version of MATLAB and MDCE, after you continue with the next steps.

mdce install

3. MDCE starts the service on all cluster nodes

mdce start

4. Start the JobManager typing the following command.

startjobmanager -name <MyJobManager>
-remotehost <job manager hostname> -v

Verify that the Manager is running on the host specified.

5. Start the slaves (workers)

startworker -jobmanagerhost <job manager hostname>
-jobmanager <MyJobManager> -remotehost <worker host-name> -v

6. Verify that the slaves are running on each computer:

nodestatus -remotehost <worker hostname>

It should be mentioned that the program receives the JobManager, which is the computer that controls all processes and this is where all tasks are carried out for pre-processing for biometric measures. As shown in Figure 21. This is how to distribute the work in the processor to do the biometric processing, after that the results are based on the fuzzy integrator and the Gating Network.

7 Simulation Results

In this section, we only show the results obtained for each of the biometric measures taking the tests with 5 methods of training and choosing the one with the better results. In table 1 we show the different methods of training used with the data bases and the method for integration, which was the gating network.

We perform several trainings in order to arrive to these results, although these are on average the best results that we have obtained for each of the biometric measures.

The results are given in three importants stages, the first is where the results of each of the biometric measures are presented separately, the second is where the three biometric measures are integrated through a fuzzy inference and tested on a single computer The third stage is the integration of the three biometric measures using response integration on distributed computing in parallel. We used different methods of training to check which one was the most appropriate biometric measurement. In Table 2 we shown the training methods used.

Table 2. Training methods for the neural networks.

Abbreviation	Training method	Integrator
TRAINGDX	Gradient descendent with momentum and adaptive learning rate backpropagation	Gating Network
TRAINGDA	Gradient descendent with adaptive learning rate bagpropagation	Gating Network
TRAINSCG	Scaled conjugate gradient backpropagation	Gating Network
TRAINRP	Resilient backpropagation	Gating Network
TRAINCGB	Conjugate gradient backpropagation with Powell-Beales restarts	Gating Network

7.1 Results of the Three Biometric Measures Considered Separately

In this section we show separate results for the biometric measures.

7.1.1 Signature Results

The results for the measurement of biometric signatures were very good, we achieved an average recognition of 99%, which is considerably good for the development of the system, and these results are shown below in Tables 3 and 4.

Table 3. Best simulation results without learning rate.

Training	Error Goal	Training Methods	Identification	Error	% Identification
1	0.0000001	Traingdx	87	3	87/90(97.50%)
2	0.0000001	Traingda	87	3	87/90(97.50%)
3	0.0000001	Trainscg	86	4	86/90(96.66%)
4	0.0000001	Trainrp	50	40	50/90(66.66%)
5	0.0000001	Traincgb	68	22	68/90(81.66%)

Table 3 shows the best training without the use of learning rate, as can be seen, we have a percentage of recognition of 97.50 with the training methods “traingdx” and “traingda”.

Table 4. Best simulation results with learning rate of 0.001.

Training	Error Goal	Training Methods	Identification	Error	% Identification
1	0.00000001	Traingdx	89	1	89/90(99.16%)
2	0.00000001	Traingda	87	3	87/90(97.50%)
3	0.00000001	Trainscg	86	4	86/90(96.66%)
4	0.00000001	Trainrp	50	40	50/90(66.66%)
5	0.00000001	Traincgb	68	22	68/90(81.66%)

Table 4 shows the best results that were obtained for signature with a learning rate of 0.001, increasing our percentage of identification here to a 99.16% and the best Method of training was “traingdx”.

7.1.2 Fingerprint Results

Several tests were made to reach the results shown in tables 5 and 6. In Table 5 we show the results of tests made without learning rate and a target error of 0.00000001 of 86.60%.The best results were obtained with the traingdx and traingda methods of training.

Table 5. Best results without learning rate

Training	Error Goal	Training Methods	Identification	Error	% Identification
1	0.00000001	Traingdx	75	15	75/90(86.60%)
2	0.00000001	Traingda	97	15	75/90(86.60%)
3	0.00000001	Trainscg	95	17	73/90(84.82%)
4	0.00000001	Trainrp	34	56	56/90(50%)
5	0.00000001	Traincgb	61	29	83/90(74.10%)

Better results were obtained by adding a learning rate of 0.001 with the same goal error is achieved with this increase above the percentage of identification in a 89.28% as shown in table 6, the best training method was trainscg.

Table 6. Best results with learning rate of 0.001

Training	Error Goal	Training Methods	Identification	Error	% Identification
1	0.0000001	Traingdx	99	13	92/112(88.39%)
2	0.0000001	Traingda	97	15	97/112(86.60%)
3	0.0000001	Trainscg	100	12	100/112(89.28%)
4	0.0000001	Trainrp	50	62	50/112(44.64%)
5	0.0000001	Traincgb	86	26	86/112(76.78%)

7.1.3 Face Results

For the face identification we have results for the data base of the institution and for the data base of the ORL, in both situations we performed different tests. In table 6 we show results that include a case of 100% identification for the data base from the institution by means of the training method traingdx.

Table 7. Results of original database take in the Tijuana Institute of Technology

Training	Error Goal	Training Methods	Epoch	Time	% Identification
1	0.0000001	Traingdx	500,500,500	2:11	90/90(100%)
2	0.0000001	Traingda	500,500,500	2:03	89/90(98.33%)
3	0.0000001	Trainscg	500,500,500	8:21	88/90(96.6%)
4	0.0000001	Trainrp	500,125,240	2:15	82/90(86.66%)
5	0.0000001	Traincgb	209,134,126	3:9	86/90(93.33%)

In order to verify the effectiveness of the result of the 100% of identification, tests of cross validation were realized where the positions of the images of the data base were alternated to verify that the percentage of identification persists, verifying if there is a result that fails as it is possible to be appreciated in table 7, another difference that can be appreciated with the cross validation is that the training method varies since with the normal data base we have the method of training traingdx and for these tests with which better results were achieved with trainscg, with a 95% of identification.

Table 8. Simulation results of cross validation

Train	Error Goal	Training Methods	Epoch	Time	% Identification
1	0.0000001	Traingdx	500,500,500	2:11	85/90(91.6%)
2	0.0000001	Traingda	500,500,500	2:03	84/90(90%)
3	0.0000001	Trainscg	500,500,500	8:21	87/90(95%)
4	0.0000001	Trainrp	500,125,240	2:15	82/90(86.66%)
5	0.0000001	Traincgb	209,134,126	3:9	86/90(93.33%)

Table 9. Best results of Three Biometrics measures with better training methods.

Biometric Measure	Training Method	Error Goal	% Identification
Face	Traingdx	0.0000001	90/90=100%
	TRainscg		88/90=97%
	Trraingda		89/90=99%
Face with Cross-Validation	Traingdx	0.0000001	85/90=94%
	TRainscg		87/90=96 %
	Trraingda		84/90=93%
Signature	Traingdx	0.00000001	87/90=96%
	TRainscg		87/90=96%
	Trraingda		86/90=96.66%
Signature with Cross-Validation	Traingdx	0.00000001	89/90=99%
	TRainscg		86/90=95%
	Trraingda		87/90=96%
Fingerprint	Traingdx	0.00000001	77/90=88.39%
	TRainscg		73/90=84.82%
	Trraingda		70/90=86.60%
Fingerprint with Cross-Validation	Traingdx	0.00000001	82/90=92.85%
	TRainscg		70/90=86.60%
	Trraingda		78/90=89.28%

For each of the biometric measures we have had very good results in an independent manner. It is important to consider the fact that in the future, when making the integration of the 3 biometric measures the percentage of identification will increase, because of the modularity and thus having a greater system reliability and performance.

Table 10. Results of Fuzzy integrator for the biometrics in set.

Type of Fuzzy Integrator	% of Identification	Identification Error	Biometric Measure	Range of Measure	Activations in each module
Mandami, Gausianas	100%	0	Face	0.3333	16
			Signature	0.6666	12
			Fingerprint	1	2

Table 11. Results of Integration in Distribute Compute with Fuzzy System, fuzzy integrals and measures Sugeno and gating network.

Image	Gating Network		Fuzzy Integral and Sugeno Measures		Fuzzy Logic Type- 1		Workers	Training Time
	% Rec	% Ident	% Rec	% Ident	% Rec	% Ident		
60	60/60 100%	60/60 100%	60/60 100%	60/60 100%	60/60 100%	60/60 100%	3 Rem	1 min 37 seg
80	80/80 100%	78/80 97%	80/80 100%	72/80 9%	80/80 100%	79/80 98.7%	3 Rem	2 min 27seg
90	90/90 100%	90/90 100%	90/90 100%	89/90 96%	90/90 100%	90/90 100%	3 Rem	3 min 58seg
120	120/120 100%	118/120 98%	120/120 100%	110/120 91%	120/10 100%	120/120 100%	3 Rem	5 min 49seg

The best results of the three biometric measures with the best training methods of are show in the table 8, where you can view face and face with cross-validation, signature and signature with cross validation as that fingerprint.

7.1.4 Best Results

In table 9 we show the best results of the three biometric measures with the three best training methods for each biometric measure.

7.2 Results of Fuzzy Response Integrator

For the three biometric measures a Mamdani type fuzzy integrator was used,which was developed using Gaussian type membership functions.
Table 10 shows that fuzzy integrator for the combination of the measures in a single value has achieved a 100% identification.

With 0 errors and is reached to verify that the three modules are activated to obtain this result, so the efficiency of a module compensates for other deficiencies. For the fuzzy inference system takes values between 0 and 1, as follows: from 0.3333 to the module face 0.6666 for Release 1 of the signature and fingerprint.

7.3 Results Using a Cluster

A consequence of the positive results in the previous stage it was decided to launch processes to remote nodes that are in the cluster of computers, just to verify that the results persist even when the database grows, the components can be seen in table 11, we use 7 training images per person and leaving 3 is to identify, 90 pictures to 30 persons and 60 to 20 persons, although the first tests were conducted for a number of 20 persons, a same operation to train with 6 images and allow for the identification is 4 a total of 120 for 30 persons and 80 to 20 persons.

In the table we show both the percentage of appreciation that is what is done with the images that were training as the percentage of identification that is what interests us is that the testing is done with the images that were not previously trained, It can be seen that the results are good.

8 Conclusions

We can conclude that working with biometric measures is much more reliable than with other forms of authentication and the fact that combining several biometric measures helps to have a greater percentage of reliability because measures cover the deficiencies of the others within a system.

At this moment, we have good results with the independent biometric measures, so it is possible to say that good results may be obtained after having of the integration of 3 measures. We will try this in the near future.

It is important to note that the goal of using parallel computing is to ensure that if the databases increase in thousands, the execution time for the identification of a person and the percentage of identification are reliable. Therefore we note that our research work with regard to biometric measures was much more effective with the “trainscg” training method and the best integrator were thegatting network and type-1 fuzzy systems.

Acknowledgment. We would like to express our gratitude to CONACYT, and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] Castillo, O., Melin, P.: Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. IEEE Transactions on Neural Networks 13(6), 1395–1408 (2002)

[2] Roger Jang, J.-S., Sun, C.-T., Mizutani, E.: Neuro-fuzzy and Soft Computing a Computational Approach to Leatning and Machine Inteligence

[3] Faros, A.: Biologically Inspired Modular Neural Networks. Blacksburg, Virginia (May 2000); Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)

[4] Melin, P., Gonzalez, F., Martinez, G.: Pattern Recognition Using Modular Neural Networks and Genetic Algorithms. In: Proc. IC-AI 2004, Las Vegas, Nevada, USA, pp. 77–83 (2004)

[5] Romero, M.: Sistemas Modulares, Mezcla de expertos y sistemas Hibridos (March 2004)

[6] Nuñez, R.: ITT, Redes Neuronales Modulares en Paralelo para el Reconocimiento de Personas (February 2008)

[7] Pérez, A.: Reconocimiento y verificación de firmas manuscritas off-line basado en el seguimiento de sus trazos componentes, eslova.com (2002/2003)

[8] Fu, H.-C., Lee, Y.-P., Chiang, C.-C., Hsiao-Tien: Divide and conquer learning and modular perceptron networks in IEEE transaction on neural networks. In: Goldberg, D. (ed.) Genetic Algorithms, vol. 12. Addison Wesley, Reading (1988)

[9] Salinas, R.: Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Red Neuronal de Arquitectura Paramétrica en Reconocimiento de Rostros, Universidad Diego Portales (August 2008), `http://cabierta.uchile.cl/revista/17/articulos/pdf/paper4.pdf`

[10] Romero, L.: Dpto. de Informática y Automática. Universidad de Salamanca- España. Calonge Cano Teodoro, Dpto de Informática. Universidad de Valladolid. España, Redes Neuronales y Reconocimiento de Patrones, `http://lisisu02.fis.usal.es/airene/capit1.pdf`

[11] Sugeno, M.: Theory of Fuzzy integrals and its application. Tokyo Institute of Technology (1974)

[12] The matworks Inc., MATLAB Distributed Computing Engine Systems Administrator's Guide (2007)