Engineering Applications of Artificial Intelligence 64 (2017) 172–186

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Optimization of modular granular neural networks using a firefly algorithm for human recognition

Daniela Sánchez, Patricia Melin *, Oscar Castillo

Tijuana Institute of Technology, Tijuana, Mexico

A R T I C L E I N F O

Keywords:
Modular neural networks
Granular computing
Firefly algorithm
Human recognition
Ear recognition
Face recognition
Pattern recognition

A B S T R A C T

In this paper a new optimization method for modular neural network (MNN) design using granular computing and a firefly algorithm is proposed. This method is tested with human recognition based on benchmark ear and face databases to verify the effectiveness and the advantages of the proposed method. Nowadays, there are a great number of optimization techniques, but it is very important to find an appropriate one that allows for better results depending on the area of application. For this reason, a comparison of techniques is presented in this paper, where the results achieved for ear recognition and face recognition by the proposed method are compared against a hierarchical genetic algorithm in order to know which of these techniques provides better results when a modular granular neural network is optimized and applied to pattern recognition mainly for human recognition. The parameters of modular neural networks that are being optimized are: the number of modules (or sub granules), percentage of data for the training phase, learning algorithm, goal error, number of hidden layers and their number of neurons. Simulation results show that the proposed approach combining the firefly algorithm with granular computing provides very good results in optimal design of MNNs.

1. Introduction

Human recognition based on biometric measures has allowed achieving better control or access to information or restricted areas, or simply to know the identity of a person (Kaur, 2016; Solanki and Pittalia, 2016). There is a plethora of works where different methods have been developed to find robust systems that provide security using biometric measures, such as face (Ch'ng et al., 2012), ear (Sánchez and Melin, 2014), fingerprint (Chandana et al., 2015; Sankhe et al., 2016), iris (Homayon, 2015), voice (Zhang et al., 2015), among others. Some of the main techniques used in the development of these works are neural networks (Haykin, 1994; Hassoun, 2003), fuzzy logic (Zadeh and Kacprzyk, 1992), data mining (Witten et al., 2011), evolutionary computation (Eiben and Smith, 2015), granular computing (Yao, 2005; Zhong et al., 2016), to mention only a few. Within of area of evolutionary computation, there is a wide variety of approaches as for instance; the already well-known genetic algorithm (GA) (Man et al., 1999; Holland, 1975), ant colony system (ACO) (Dorigo, 1992), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), or more recently, the cuckoo optimization algorithm (COA) (Rajabioun, 2011) or the firefly algorithm (FA) (Yang, 2009; Yang and He, 2013), among others. The combination of two or more of the above described techniques forms a hybrid intelligent system. These kinds of systems have been proposed in some works (Sánchez and Melin, 2014; Hidalgo et al., 2009; Martínez-Soto et al., 2015; Farooq, 2015), where better results have been achieved than when an individual technique is used. For this reason, in this paper a hybrid intelligent system is proposed, where techniques, such as modular neural network, granular computing and a firefly algorithm are combined.

A modular neural network is an improvement on the conventional artificial neural network (ANN), where if a task can be divided into subtasks each of these sub-tasks is learned by an expert sub-module. This technique has been successfully used in pattern recognition, particularly to human recognition using different biometric measures (Hidalgo et al., 2009; Pastur-Romay et al., 2016; Balarini et al., 2012; Basu et al., 2010). In this work, a modular neural network using a granular approach is used. Granular computing defines a granule as one of the numerous small particles forming a larger unit (Yao, 2005; Zadeh, 1998). This approach has been successfully combined with other areas (Bargiela and Pedrycz, 2006), and in the proposed method, granular computing is applied to granulate the information that the modular neural network is going to learn. This kind of neural networks was already proposed in Sánchez and Melin (2014), where a comparison among MGNN, MNN

* Corresponding author.
E-mail addresses: danielasanchez.itt@hotmail.com (D. Sánchez), pmelin@tectijuana.mx (P. Melin), ocastillo@tectijuana.mx (O. Castillo).

http://dx.doi.org/10.1016/j.engappai.2017.06.007
Received 21 October 2016; Received in revised form 7 April 2017; Accepted 5 June 2017
Available online 4 July 2017

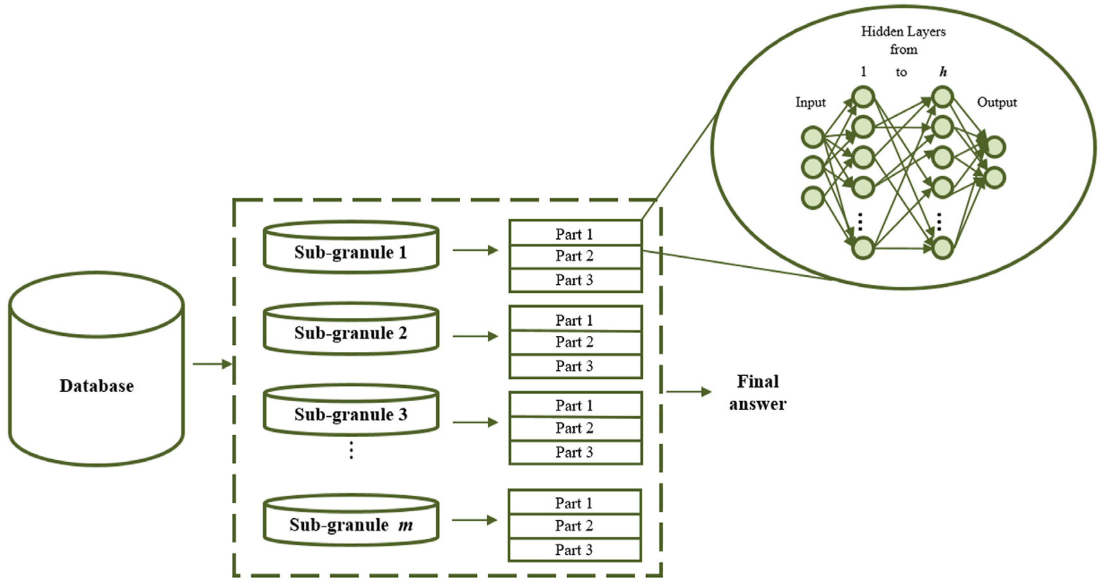

Fig. 1. The general architecture of the proposed method.

and ANN was performed and the advantages of MGNN were widely demonstrated. In that work the architecture of the MGNN was optimized by a hierarchical genetic algorithm (HGA). A HGA is an improvement to the conventional genetic algorithm, which has control genes to allow the activation and deactivation of genes and this allows solving complex problems (Raikova and Aladjov, 2002). In this paper, the contribution is the design of modular neural network architectures using a granular approach and these models are applied to human recognition based on the ear (Gutierrez et al., 2010) and face (Mendoza et al., 2010), but the proposed method can be applied to other biometric measures. The design of the MGNN is performed using a firefly algorithm, where the MGNN parameters must be found by the proposed algorithm, in this case the number of modules (sub granules), percentage of data for the training phase, learning algorithm, goal error, number of hidden layers and their number of neurons

This paper is organized as follows. The proposed method is described in detail in Section 2. The results obtained using the proposed method are presented and explained in Section 3. Statistical comparisons of results to measure the advantage of the proposed method are presented in Section 4. Finally, in Section 5, the conclusions and future work are offered.

2. Proposed method

The general architecture of the proposed method is described in this section; this method designs modular granular neural networks architectures using a firefly algorithm.

2.1. General architecture of the proposed method

The proposed method is based on modular neural networks with a granular approach. This kind of neural network was proposed in Sánchez and Melin (2014), where the advantages that these modular granular neural networks have over the conventional neural networks were presented, and basically this is because if a problem can be divided into smaller sub problems, each sub module (or sub granule) can be an expert on a part of the problem. In that work (Sánchez and Melin, 2014), the optimization of the MGNN was performed using a hierarchical genetic algorithm, and in this work a comparison between HGA and FA is performed to find out which of these techniques is better to perform the optimization. The main idea is to find the number of sub-modules (sub-granules) and each of these sub-modules is divided into 3 parts, because each image is also divided into 3 parts in a pre-processing process, and this part is going to be described later.

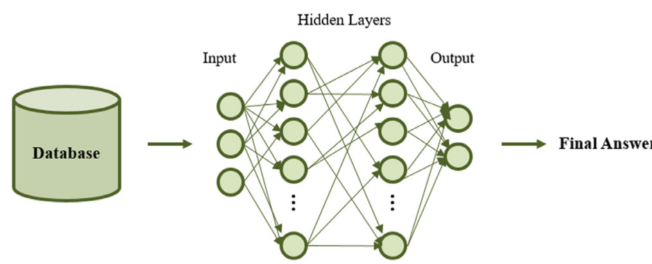

Fig. 2. An example of an artificial neural network.

In Fig. 1, the granulation process also used in Sánchez and Melin (2014) is illustrated. A database represents a whole granule, and this granule is divided into sub granules, but each sub granule can have different size, and for these applications, it would be different number of persons learned by each sub module (sub granule).

2.1.1. Description of the modular granular neural network

The modular granular neural networks were proposed in Sánchez and Melin (2014), and the main difference between this kind of neural network and a conventional neural network is the division of a task into subtasks and then obtaining a final decision. In Fig. 2, an example of an artificial neural network is presented where a whole database is considered for its inputs and depending of the type of transfer functions outputs are obtained, so depending on the application a final output or result is achieved.

In Fig. 3, an example of a modular granular neural network can be found. As it was mentioned before this kind of neural network divides a task into subtasks. In this case, the whole database is a main granule, this granule can be divided into sub granules, and each sub granule represents the inputs of an independent neural network. The internal processes are the same as for a conventional neural network including the learning process, but for obtaining a final decision, the responses of modules (neural networks) must be combined or integrated. There are methods to perform this integration task, such as fuzzy integration, the winner takes all or a gating network.

In this work, each sub granule is divided into 3 sub modules as Fig. 1 shows, because the images are divided into 3 parts. To combine

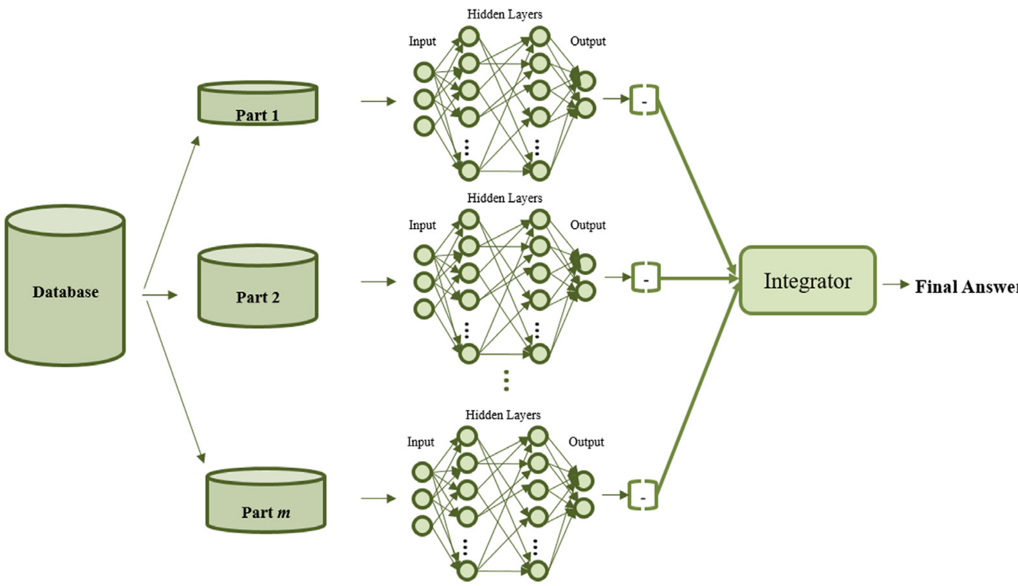

Fig. 3. An example of a modular granular neural network.

their responses the winner takes all method is used. The neurons of the hidden layers use a hyperbolic tangent sigmoid transfer function, and this transfer function limits the output of each neuron between a value of −1 and 1. On the other hand, in the output layer the neurons use a sigmoid transfer function, and this transfer function limits the output of each neuron between a value of 0 and 1. The learning process is performed using backpropagation algorithms. The backpropagation algorithms seek to minimize the error function using the gradient descent method. This error is internally calculated in the training phase (learning process) and it is independent of the error of the testing phase that the proposed method seeks to minimize.

2.1.2. Description of the firefly algorithm

This algorithm was originally proposed in Yang (2009) and Yang and He (2013). This algorithm is based on the flashing and behavior of the fireflies. This algorithm uses three basic rules (Yang, 2009; KumarSrivastava and Singh, 2016):

1. A firefly is unisex, and can be attracted to others no matter of their sex.
2. The attractiveness is proportional to the brightness, and they both decrease as their distance increases, and for a couple of fireflies, the firefly with less brightness will move toward the brighter one. If there is no brighter one than a particular firefly, then it will move randomly.
3. The brightness of a firefly is given by the objective function.

In Fig. 4, the pseudo code of the firefly algorithm based on these rules is shown.

In Yang and He (2013), the variation of attractiveness β with the distance r is defined by equation:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{1}$$

where β_0 is the attractiveness at $r = 0$. The movement of a firefly i to another brighter one j is defined by the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j^t - x_i^t \right) + \alpha_t \epsilon_i^t \tag{2}$$

where the first term, the position x_i represents a firefly in the iteration t, the second term is due to the attraction between a firefly j and a firefly i, and in the third term, ϵ_i^t is a vector of random numbers with a

```
Objective function f(x), x = (x1, ..., xd)^T
Generate initial population of fireflies xi (i = 1, 2, ..., n)
Light intensity Ii at xi is determined by f(xi)
Define light absorption coefficient γ
while (t <MaximumGeneration)
      for i = 1 : n all n fireflies
          for j = 1 : i all n fireflies
              if (Ij > Ii), Move firefly i towards j in d-dimension;
              end if
               Attractiveness varies with distance r via exp[− γr]
               Evaluate new solutions and update light intensity
         end for j
      end for i
Rank the fireflies and find the current best
end while
```

Fig. 4. Pseudo code of the firefly algorithm.

randomization parameter represented by α_t. This parameter is the initial randomness scaling factor and is defined as:

$$\alpha_t = \alpha_t \delta^t \tag{3}$$

where δ is a value between 0 and 1. The values for α, β and δ applied in this work are shown later. The firefly algorithm uses a random array, which allows moving the fireflies and avoiding a local minimal problem.

2.1.2.1. Description of the firefly algorithm for MGNN optimization. This algorithm seeks to minimize the recognition error and the objective function is given by the equation:

$$f = \sum_{i=1}^{m} \left(\left(\sum_{j=1}^{n_m} X_j \right) / n_m \right) \tag{4}$$

where $\boldsymbol{m}$ is the total number of modules, $\boldsymbol{X_j}$ is 0 if the module provides the correct result and 1 if not, and $\boldsymbol{n_m}$ is total number of data/images used for testing (information of the testing set, i.e. the modular granular neural network did not use it in the learning process) in the corresponding module.

Each dimension represents a parameter that will be optimized. The number total of dimensions can be calculated as:

$$\boldsymbol{Dimensions} = 2 + (3 * \boldsymbol{m}) + (\boldsymbol{m} * \boldsymbol{h}) \tag{5}$$

where $\boldsymbol{m}$ is the maximum number of modules that the firefly algorithm can use, and $\boldsymbol{h}$ is maximum of number of hidden layers per module that

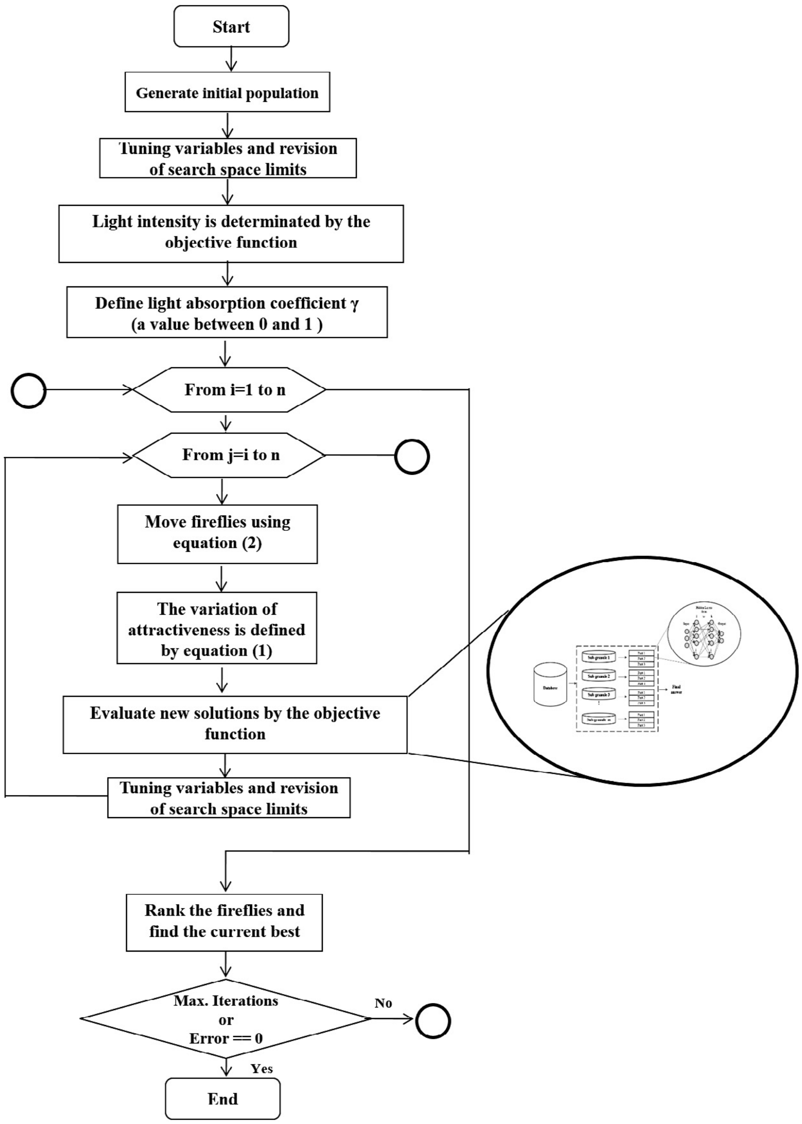

Fig. 5. Diagram of the proposed method.

the firefly algorithm can use to perform the optimization. These variables can be established depending on the application or the database, but to have a fair comparison versus (Sánchez and Melin, 2014), these variables have the same values in both optimizations. A continuous firefly algorithm is used, but to fine-tune it discrete variables and an additional process is added, where variables such as the number of modules, the number of hidden layers and the number of their neurons are rounded. In this process an update to the limits of the search space was also performed.

For the firefly algorithm parameters, in the case of the number of fireflies and maximum number of iterations, also the same values were established to have a fair comparison between the genetic algorithm, and in the case of number of individuals and maximum number of generations that are parameters of this last method. The rest of the firefly algorithm parameters are based on the parameters recommended in Yang (2009) and Yang and He (2013). In Table 1, the used parameters are presented. In Fig. 5, the diagram of the proposed method is illustrated.

2.2. Proposed method applied to human recognition

In this work to perform the simulations, three backpropagation algorithms are used: gradient descent with scaled conjugate gradient (SCG), gradient descent with adaptive learning and momentum (GDX) and gradient descent with adaptive learning (GDA). These algorithms were selected because according to the literature, they are the fastest algorithms and in other works better results are obtained when they are used (Sánchez and Melin, 2014; Gutierrez et al., 2010; Sánchez et al., 2015):

In this work as in Sánchez and Melin (2014), ***m*** and ***h*** are set to 10. To establish the search space, the minimum and maximum values used are shown in Table 2. This firefly algorithm has two stopping conditions: when the maximum number of iterations is achieved and when the best firefly has an error value equal to zero.

2.2.1. Data selection, databases and pre-processing

The selection of data, the description of the databases for testing the proposed method and the pre-processing applied to these databases is presented below.

2.2.1.1. Data selection. The neural networks have two important phases. The first, when the neural network learns the information or patterns (learning process) and the second, when the network simulates other information that the neural network did not learn. It is important to have a good selection of information that the neural network will learn, for this reason in Sánchez and Melin (2014), a new method

Table 1
Table of parameters.

Parameter	Value
Firefly (n)	10
Maximum number of iterations (t)	30
α	0.01
β	1
δ	0.97

Table 2
Table of values for search space.

Parameters of MNNs	Minimum	Maximum
Modules (m)	1	10
Percentage of data for training	20	80
Error goal	0.000001	0.001
Learning algorithm	1	3
Hidden layers (h)	1	10
Neurons for each hidden layers	20	400
Epoch	–	2000

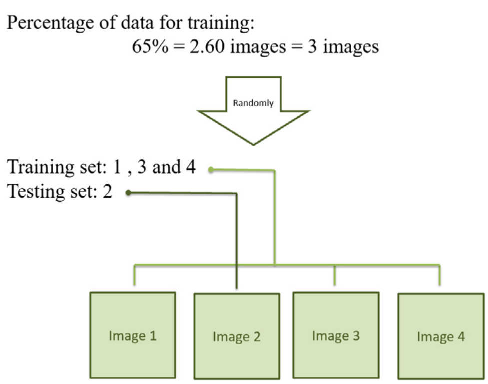

Fig. 6. Example of selection of data for training and testing phase.

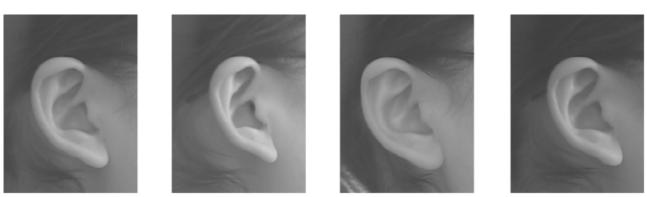

Fig. 7. Sample of the Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB).

to select information or images was proposed, where depending on a percentage of data provided by the optimization technique, sets of training and testing are established. The images per person are divided into a training set and testing set. The training set is learned by the modular granular neural network and, the testing set is used to verify the robustness and the ability to generalize of the MGNN. In this work, this division is determined by the firefly algorithm using the percentage of data for the training phase where depending of the number of images of the database the percentage is converted into number of images. The selection of images for each set is randomly performed. Owing to that the selection process is randomly, the quality of the input data is not controlled. The idea is to allow to the MGNN learns randomly information to develop its capacity for generalization. In Fig. 6, an example is illustrated, where a person has 4 images (as ear database) and 3 of them are for the training phase.

2.2.1.2. Database of ear. The proposed method is tested using the database from the Ear Recognition Laboratory of the University of Science & Technology Beijing (USTB). The database contains 77 persons (each person has 4 images of one ear). The image dimensions are 300 × 400, with BMP format (Database, 2015). A sample of the images is shown in Fig. 7.

For this database, the next pre-processing is performed: the ear image is manually cut, a resizing of the new image to 132 × 91 pixels is performed and automatically the image is divided to three regions of interest (helix, shell and lobe), this preprocessing was already performed in Sánchez and Melin (2014). The pre-processing process is shown in Fig. 8.

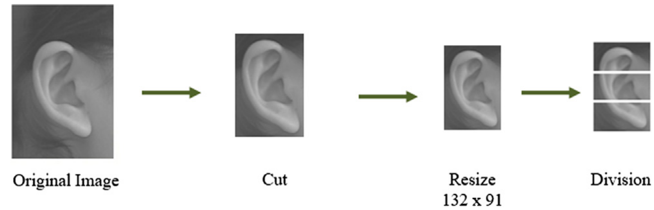

Fig. 8. Sample pre-processing for ear database.

Fig. 9. Sample of the ORL database from the AT&T laboratories cambridge.

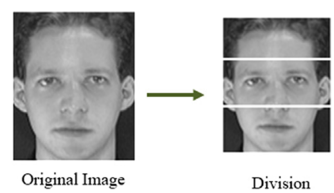

Fig. 10. Sample pre-processing for ORL database.

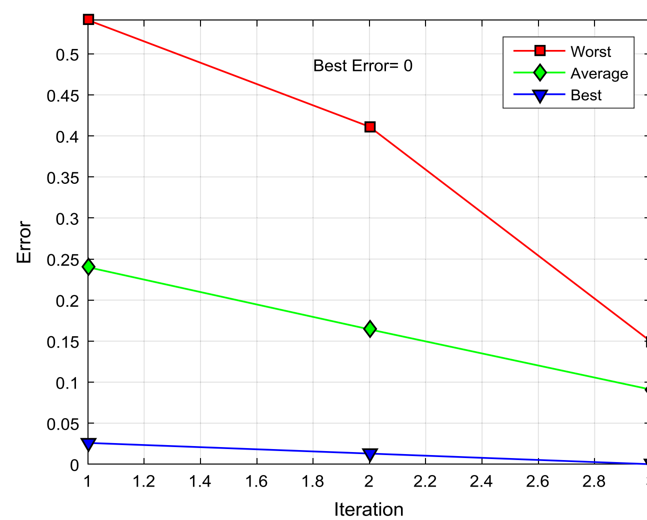

Fig. 11. Convergence of evolution #18.

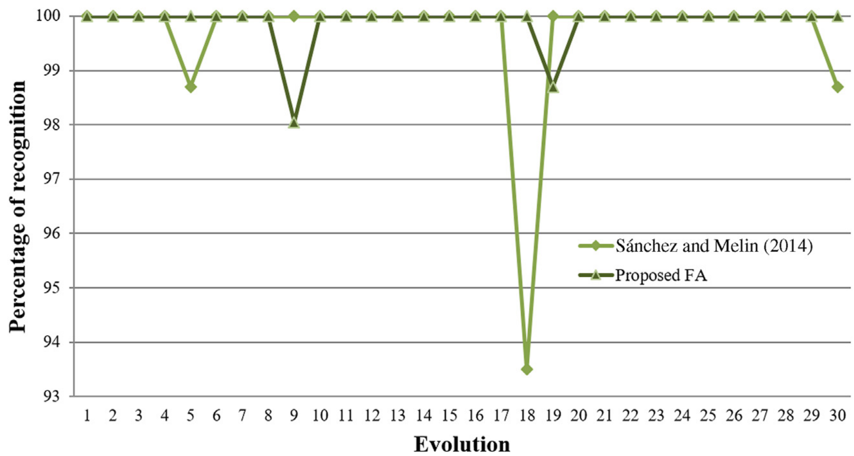

Fig. 12. Obtained recognition rates (up to 80%).

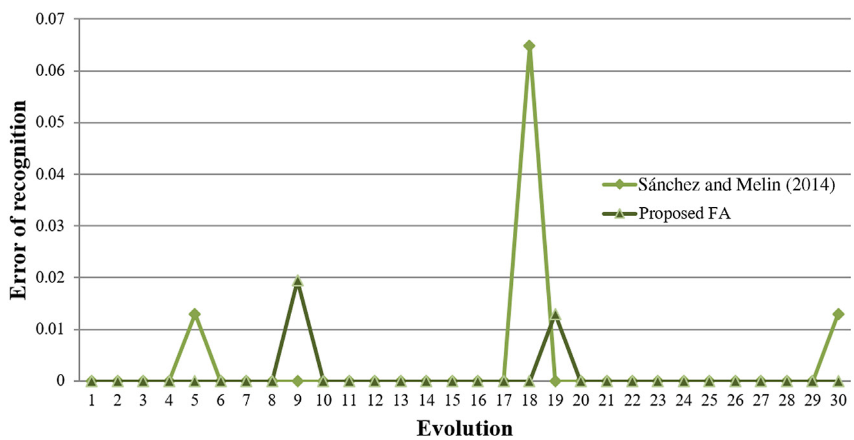

Fig. 13. Obtained recognition errors (up to 80%).

2.2.1.3. Database of faces. The second database contains 40 persons, each person has 10 images, is from the AT&T Laboratories in Cambridge. The image dimensions are 92 × 112 pixels, with PGM format. Fig. 9 shows a sample of the images of this database (Database of Human Iris, 2015).

The pre-processing process for this database is presented in Fig. 10. Each image is automatically divided into three regions of interest (front, eyes and mouth).

3. Experimental results

The results achieved by the proposed method when applied to human recognition are presented in this section. A summary of the results is presented in Section 3.3. For the ear database, in Sánchez and Melin (2014), 6 tests were performed (4 non-optimized tests and 2 optimized tests). Statistically it was found that optimized are better than non-optimized tests, for this reason, in this work, only the two optimized tests are replicated (30 evolutions for each test) to compare both optimization techniques. For the face database, 30 evolutions were also performed using up to 80% of the data, and 30 evolutions were also performed using up to 50% of the data. All the results shown in the tables and figures were obtained by experimenting with information of the testing set, i.e. information not provided to the modular granular neural networks in the learning process.

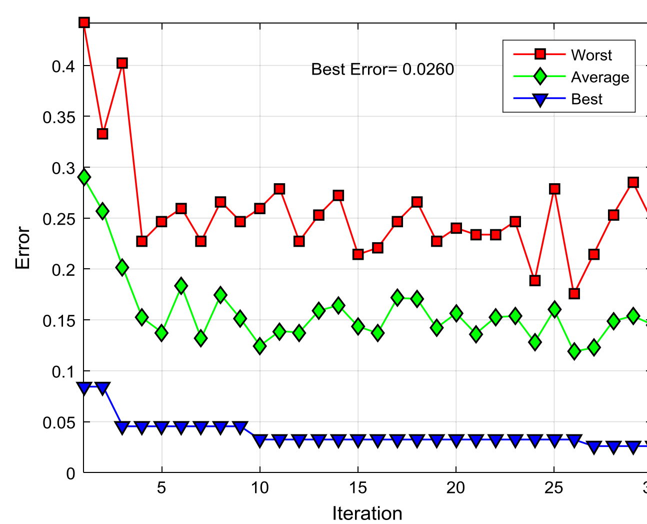

Fig. 14. Convergence of evolution #7.

3.1. Ear results

The results achieved using the ear database are presented below. Only the best results are presented.

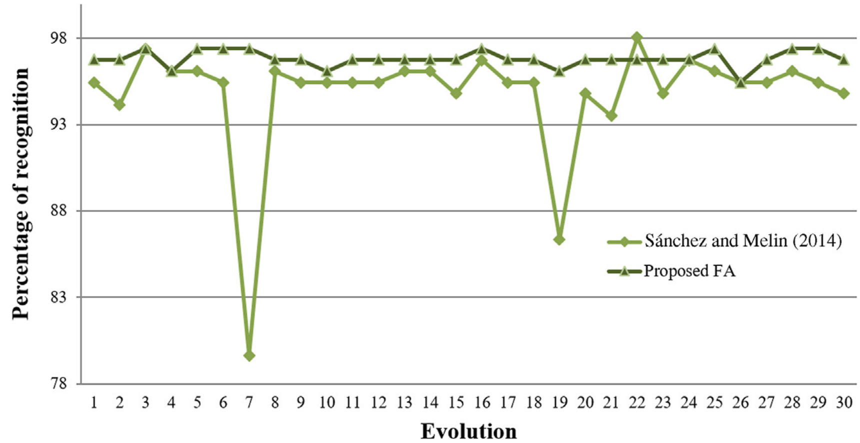

Fig. 15. Obtained recognition rates (up to 50%).

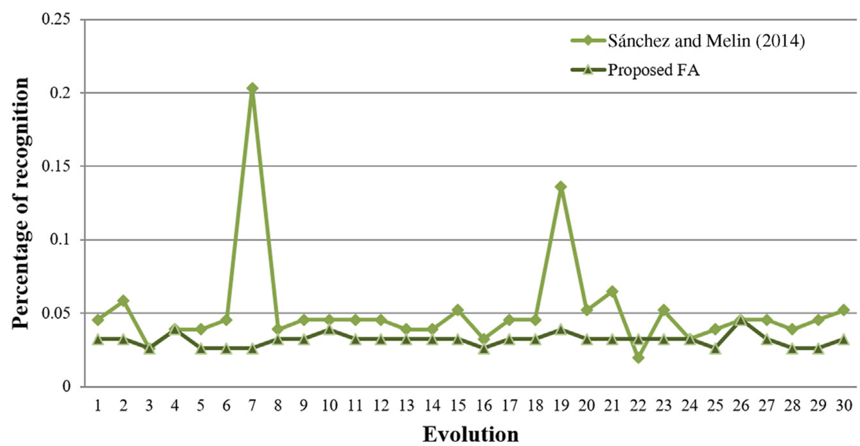

Fig. 16. Obtained recognition errors (up to 50%).

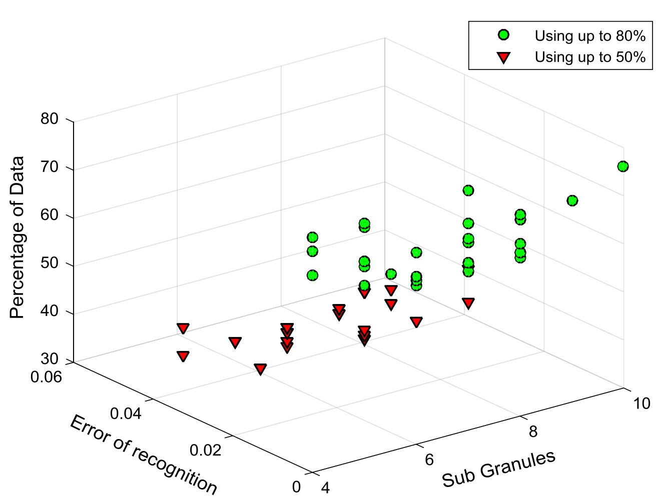

Fig. 17. Analysis of results (ear database).

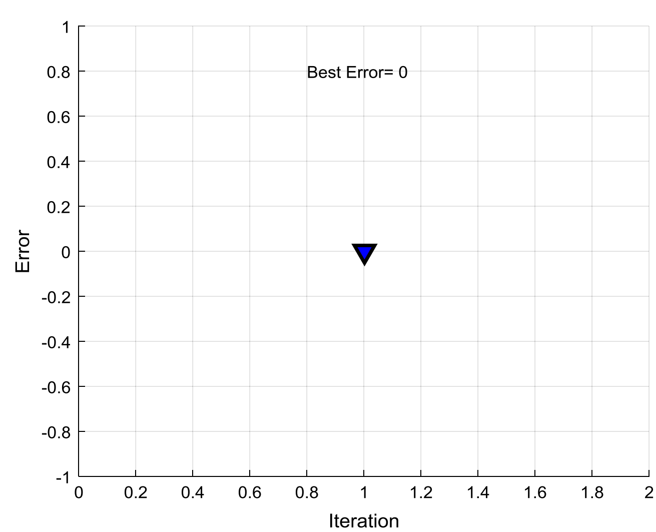

Fig. 18. Convergence of evolution #1.

3.1.1. Results using a percentage of data for training up to 80%

In this test, the proposed firefly algorithm can use up to 80% of data for the training phase. In Sánchez and Melin (2014), this test is called "test3_80%". In Table 3 the best 10 results are shown.

The behavior of evolution #18 is presented in Fig. 11. This evolution was one of the fastest evolutions to obtain a 100% of recognition rate (an error value equal to zero).

Table 3
The best 10 results (up to 80%).

Evolution	Images		Num. Hidden layers and Num. of neurons	Persons per module	Rec. rate	Error
	Training	Testing				
1	64% (1, 2 and 3)	36%(4)	5(150,137,100,158,112) 5(106,108,97,178,155) 5(153,201,104,100,71) 5(125,104,61,125,74) 5(185,60,174,89,199) 4(93,216,46,31) 5(121,209,133,168,106) 4(97,148,181,87)	Module #1(1–8) Module #1(9–12) Module #1(13–21) Module #1(22–26) Module #1(27–33) Module #1(34–39) Module #1(40–58) Module #1(59–77)	100% (77/77)	0
2	70% (2, 3 and 4)	30%(1)	3(77,129,151) 5(83,132,20,33,51) 5(47,26,142,219,163) 5(160,61,69,83,116) 4(185,77,167,63)	Module #1(1–5) Module #1(6–30) Module #1(31–51) Module #1(52–70) Module #1(71–77)	100% (77/77)	0
3	65% (2, 3 and 4)	35%(1)	5(103,135,128,173,49) 4(89,35,73,66) 5(136,86,145,92,76) 4(115,130,117,81) 5(73,116,136,101,20) 4(143,133,81,88)	Module #1(1–9) Module #1(10–25) Module #1(26–38) Module #1(39–45) Module #1(46–60) Module #1(61–77)	100% (77/77)	0
4	73% (2, 3 and 4)	27%(1)	5(110,120,85,166,181) 4(132,72,49,109) 4(57,102,162,96) 4(215,116,65,118) 3(37,108,53) 5(201,48,62,91,36) 4(36,26,121,70)	Module #1(1–4) Module #1(5–16) Module #1(17–27) Module #1(28–35) Module #1(36–51) Module #1(52–66) Module #1(67–77)	100% (77/77)	0
5	66% (1,2 and 3)	34%(4)	5(117,93,147,137,95) 5(50,92,123,138,113) 4(107,98,141,150) 5(105,132,82,75,136) 5(96,111,151,101,97) 5(74,73,67,71,134) 5(97,121,139,70,207) 4(117,137,85,137)	Module #1(1–8) Module #1(9–11) Module #1(12–22) Module #1(23–30) Module #1(31–41) Module #1(42–50) Module #1(51–64) Module #1(65–77)	100% (77/77)	0
6	79% (1, 2 and 3)	21%(4)	3(101,140,41) 3(111,156,151) 4(63,31,28,25) 5(103,146,163,128,195) 2(141,57)	Module #1(1–17) Module #1(18–30) Module #1(31–45) Module #1(46–55) Module #1(56–77)	100% (77/77)	0
7	76% (2, 3 and 4)	24%(1)	3(202,98,47) 2(30,160) 3(94,23,214) 1(124)	Module #1(1–29) Module #1(30–33) Module #1(34–54) Module #1(55–77)	100% (77/77)	0
8	70% (2, 3 and 4)	30%(1)	4(37,133,125,45) 4(167,101,50,214) 5(78,122,33,131,185) 4(198,160,166,59) 5(103,96,93,111,87) 5(52,32,89,128,55) 4(126,103,213,118)	Module #1(1–8) Module #1(9–17) Module #1(18–33) Module #1(34–37) Module #1(38–51) Module #1(52–65) Module #1(66–77)	100% (77/77)	0
10	63% (2, 3 and 4)	37%(1)	5(98,147,39,193,217) 5(135,158,155,55,126) 5(57,110,39,116,21) 4(20,59,164,194) 4(211,68,108,46) 5(128,148,167,218,151) 2(185,217) 1(205)	Module #1(1–4) Module #1(5–13) Module #1(14–27) Module #1(28–32) Module #1(33–42) Module #1(43–60) Module #1(61–63) Module #1(64–77)	100% (77/77)	0
11	64% (1, 2 and 3)	36%(4)	5(167,166,212,127,132) 4(116,107,133,36) 4(186,192,201,205) 4(198,201,61,57) 5(33,79,47,113,37)	Module #1(1–15) Module #1(16–34) Module #1(35–39) Module #1(40–59) Module #1(60–77)	100% (77/77)	0

In Fig. 12, the recognition rates obtained by the proposed firefly algorithm and the HGA proposed in Sánchez and Melin (2014) are illustrated, and in Fig. 13, the recognition errors obtained by both techniques are also presented.

Due to the great deal of information that is used for the testing phase the convergence to an error equal to zero is more frequent in most of the evolutions. The average recognition rate of the 30 evolutions for this test was 99.89% and the average of the error was of 0.001.

3.1.2. Results using a percentage of data for training up to 50%

In this test, the proposed firefly algorithm can use up to 50% of data for the training phase. In Sánchez and Melin (2014), this test is called "test3_50%". In Table 4 the best 10 results are presented.

Table 4
The best 10 results (up to 50%).

Evolution	Images		Num. Hidden layers and Num. of neurons	Persons per module	Rec. rate	Error
	Training	Testing				
1	38% (2 and 3)	62% (1 and 4)	4(111,174,67,153) 5(158,186,57,72,130) 5(154,62,128,48,117) 4(161,180,45,149) 4(101,146,87,189) 4(140,155,106,192)	Module #1(1–18) Module #2(19–36) Module #3(37–38) Module #4(39–43) Module #5(44–61) Module #6(62–77)	96.75% (149/154)	0.0325
2	42% (2 and 3)	58% (1 and 4)	4(127,171,102,106) 5(122,151,131,176,78) 4(71,97,125,163) 5(21,140,78,187,119)	Module #1(1–20) Module #2(21–36) Module #3(37–57) Module #4(58–77)	96.75% (149/154)	0.0325
3	43% (2 and 4)	57% (1 and 3)	4(124,132,126,83) 4(129,101,209,74) 5(105,161,96,129,170) 5(147,79,125,150,205) 5(171,143,108,164,139) 4(135,103,89,212) 4(118,139,110,162)	Module #1(1–15) Module #2(16–22) Module #3(23–38) Module #4(39–53) Module #5(54–64) Module #6(65–75) Module #7(76–77)	97.40% (150/154)	0.0260
5	41% (2 and 3)	59% (1 and 4)	5(122,122,106,147,196) 5(138,142,101,136,192) 3(167,114,32) 3(110,157,56) 5(96,103,139,137,155) 5(156,164,104,149,65) 5(140,159,74,142,28) 4(127,157,146,96) 3(142,39,118)	Module #1(1–10) Module #2(11–17) Module #3(18–30) Module #4(31–40) Module #5(41–42) Module #6(43–51) Module #7(52–66) Module #8(67–71) Module #9(72–77)	97.40% (150/154)	0.0260
6	41% (2 and 3)	59% (1 and 4)	5(133,172,123,126,83) 5 (129,108,110,246,79) 4 (145,193,186,143) 4 (141,166,124,27) 4 (107,157,111,163) 3(84,75,117) 4(192,109,175,24)	Module #1(1–10) Module #2(11–24) Module #3(25–28) Module #4(29–38) Module #5(39–59) Module #6(60–63) Module #7(64–77)	97.40% (150/154)	0.0260
7	40% (2 and 3)	60% (1 and 4)	4(98,233,187,199) 4(137,181,133,147) 5(144,204,99,142,121) 5(150,95,197,162,203) 5 (122,20,54,92,129) 5(141,65,68,188,106) 4(147,205,130,74)	Module #1(1–10) Module #2(11–12) Module #3(13–20) Module #4(21–30) Module #5(31–45) Module #6(46–58) Module #7(59–77)	97.40% (150/154)	0.0260
16	39% (2 and 3)	61% (1 and 4)	4(129,176,194,157) 5 (160,109,168,153,76) 1 (168) 4 (143,96,122,214) 3 (136,93,71)	Module #1(1–28) Module #2(29–37) Module #3(38–53) Module #4(54–64) Module #5(65–77)	97.40% (150/154)	0.0260
25	43% (2 and 4)	57% (1 and 3)	5(121,89,133,133,52) 4 (116,108,94,140) 5(126,99,175,238,194) 5 (121,155,183,78,54) 4(139,128,78,31)	Module #1(1–2) Module #2(3–22) Module #3(23–42) Module #4(43–61) Module #5(62–77)	97.40% (150/154)	0.0260
28	40% (2 and 3)	60% (1 and 4)	5(152,134,94,101,237) 5(119,104,95,106,43) 5(204,124,138,106,215) 2(146,93) 4(107,163,115,188) 5(101,187,144,87,209) 3(120,183,183) 3(139,111,122)	Module #1(1–7) Module #2(8–10) Module #3(11–13) Module #4(14–20) Module #5(21–31) Module #6(32–45) Module #7(46–62) Module #8(63–77)	97.40% (150/154)	0.0260
29	43% (2 and 4)	57% (1 and 3)	5(130,133,118,177,215) 5(135,130,102,105,128) 5(123,97,117,85,167) 5(142,117,116,123,200) 5(91,125,129,165,96) 4(126,131,124,55) 4(135,117,116,90)	Module #1(1–5) Module #2(6–10) Module #3(11–24) Module #4(25–33) Module #5(34–38) Module #6(39–58) Module #7(59–77)	97.40% (150/154)	0.0260

The behavior of evolution #7 is illustrated in Fig. 14. This evolution obtained a 97.40% recognition rate, and an error value equal to 0.0260.

In Fig. 15, the recognition rates obtained by the proposed firefly algorithm and the HGA proposed in Sánchez and Melin (2014) are reported, and in Fig. 16, the recognition errors obtained by both techniques are also presented. Visually, it can be noticed that the results obtained by the firefly algorithm are more stable.

In Fig. 17, an analysis of results is presented. It can be noted, that for evolutions using up to 80% of data for training, while more information is learned by the MGNN more sub granules are need, but the recognition

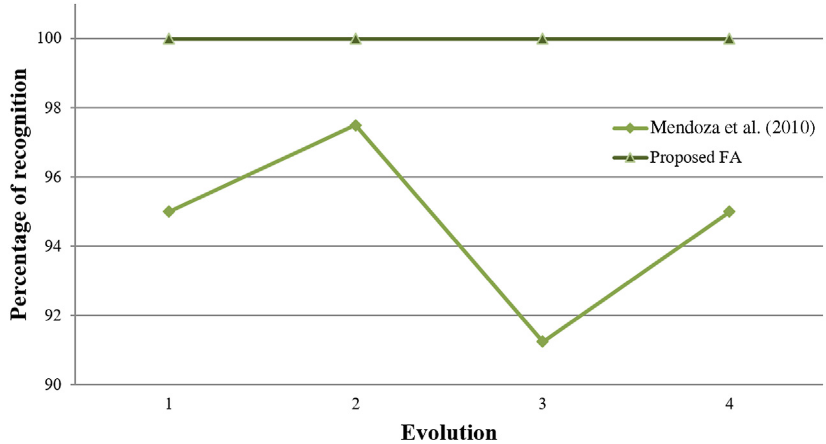

Fig. 19. Obtained recognition rates (up to 80%, face database, comparison 1).

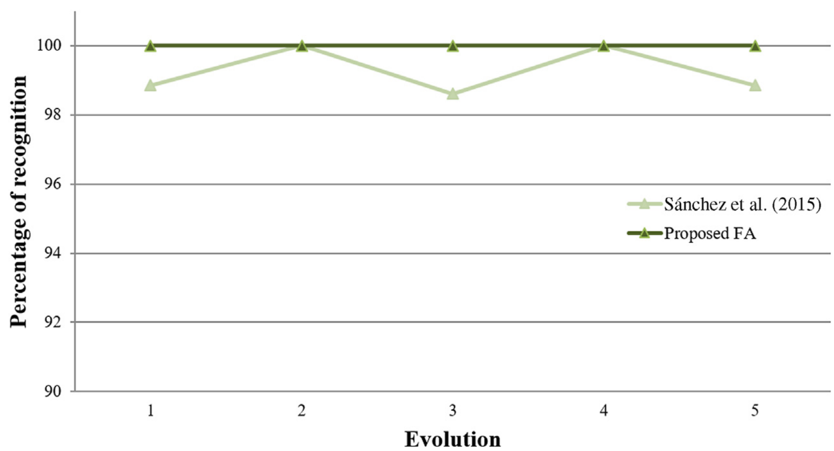

Fig. 20. Obtained recognition rates (up to 80%, face database, comparison 2).

error is reduced. In the case of the evolutions using up to 50% of data for training, most of the evolutions use between 6 and 8 sub granules, but the error of recognition is greater than when using up to 80%.

3.2. Face results

The results achieved, using the face database, are presented below. Graphical comparisons with other works are also presented.

3.2.1. Results using a percentage of data for training up to 80%

In this test, the proposed firefly algorithm can use up to 80% of data for the training phase. In Table 5, the results are shown using the proposed method.

The behavior of evolution #1 is illustrated in Fig. 18. This evolution was one of the fastest evolutions to obtain a 100% recognition rate.

In Fig. 19, the recognition rates obtained by the proposed firefly algorithm and Mendoza et al. (2010) are shown. The granulation of the information allowed having the maximum recognition rate in the first iteration; it means that using the granulation without an optimization technique was enough to have better results than a conventional neural network proposed in Mendoza et al. (2010). In Fig. 20, the recognition rates obtained by the proposed firefly algorithm and Sánchez et al. (2015) are illustrated.

3.2.2. Results using a percentage of data for training up to 50%

In this test, the proposed firefly algorithm can use up to 50% of data for the training phase. In Table 6, the results for the face database are presented.

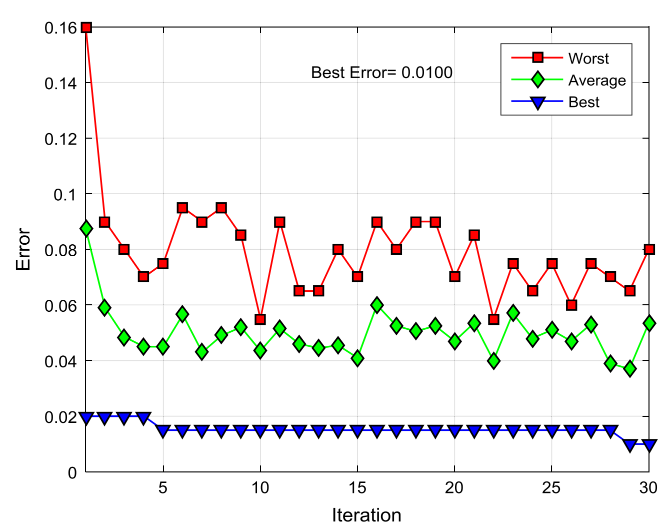

Fig. 21. Convergence of evolution #2.

The behavior of evolution #2 is illustrated in Fig. 21. This evolution obtained a 99.00% of recognition rate, and error value equal to 0.0100.

In Fig. 22, the recognition rates obtained by Sánchez et al. (2015), Azami et al. (2013) and the proposed method are illustrated. In Fig. 23, the recognition rates obtained by Ch'ng et al. (2012) and the proposed

Table 5
The results for the face database (up to 80%).

Evolution	Images		Num. Hidden layers and Num. of neurons	Persons per module	Rec. rate	Error
	Training	Testing				
1	80% (1, 2, 3, 5, 6, 7, 9 and 10)	20% (4 and 8)	5(231,86,56,179,110) 4(142,193,157,147) 5(95,185,191,129,174) 3(69,225,36) 5(120,236,124,38,114) 5(73,234) 3(227,61,237)	Module #1(1–3) Module #2(4–8) Module #3(9–5) Module #4(16–18) Module #5(19–25) Module #6(26–34) Module #7(35–40)	100% (80/80)	0
2	80% (1, 2, 3, 4, 7, 8, 9 and 10)	20% (5 and 6)	3(118,162,184) 4(96,200,248,173) 5(144,69,214,97,187) 5(23,143,248,99,117) 5(113,193,161,137,72)	Module #1(1–8) Module #2(9–14) Module #3(15–22) Module #4(23–33) Module #5(34–40)	100%(80/80)	0
3	80% (2, 3, 4, 5, 7, 8, 9 and 10)	20% (1 and 6)	3(245,240,122) 3(98,49,229) 3(247,177,81) 4(106,77,49,83) 2(124,51)	Module #1(1–6) Module #2(7–15) Module #3(16–26) Module #4(27–36) Module #5(37–40)	100% (80/80)	0
4	80% (1, 2, 3, 5, 6, 7, 9 and 10)	20% (4 and 8)	3(32,37,125) 2(234,107) 4(134,134,171,54) 3(188,185,200) 3(220,181,166) 3(100,133,52) 1(242)	Module #1(1–4) Module #2(5–10) Module #3(11–15) Module #4(16–22) Module #5(23–27) Module #6(28–31) Module #7(32–40)	100% (80/80)	0
5	80% (1, 2, 3, 4, 7, 8, 9 and 10)	20% (5 and 6)	3(35,150,141) 4(126,21,14,12) 4(55,46,111,125) 3(121,123,101) 3(101,95,48)	Module #1(1–11) Module #2(12–20) Module #3(21–30) Module #4(31–35) Module #5(36–40)	100% (80/80)	0

Table 6
The results for face database (up to 50%).

Evolution	Images		Num. Hidden layers and Num. of neurons	Persons per module	Rec. rate	Error
	Training	Testing				
1	50% (1, 5, 7, 8 and 10)	50% (2, 3, 4, 6 and 9)	4(179,139,152,34) 5(90,180,160,77,99) 4(147,129,144,207) 5(106,102,102,234,102) 4(136,99,131,107) 4(165,125,114,99) 4(146,163,159,216) 5(174,182,38,189,190)	Module #1(1–3) Module #2(4–10) Module #3(11–15) Module #4(16–20) Module #5(21–22) Module #6(23–29) Module #7(30–38) Module #8(39–40)	98.50% (197/200)	0.0150
2	50% (4, 7, 8, 9 and 10)	50% (1, 2, 3, 5 and 6)	5(156,111,139,216,228) 5(164,121,154,155,115) 5(144,107,157,118,244) 5(60,154,44,229,30) 5(139,160,151,156,163) 5(86,169,163,136,244) 4(111,141,135,139)	Module #1(1–9) Module #2(10–14) Module #3(15–21) Module #4(22–28) Module #5(29–35) Module #6(36–38) Module #7(39–40)	99% (198/200)	0.0100
3	50% (2, 5, 6, 7 and 10)	50% (1, 3, 4, 8 and 9)	4(138,150,153,180) 4(148,54,107,89) 5(162,155,117,226,86) 5(88,87,63,186,92) 4(127,155,45,164) 4(119,47,56,140)	Module #1(1–9) Module #2(10–14) Module #3(15–24) Module #4(25–29) Module #5(30–36) Module #6(37–40)	98% (196/200)	0.0200
4	50% (4, 5, 6, 8 and 9)	50% (1, 2, 3, 7 and 10)	5(152,189,42,83,129) 5(129,137,138,110,105) 5(131,81,147,139,103) 5(140,194,131,226,190) 5(134,127,71,219,227) 3(147,147,176) 5(108,184,80,169,89)	Module #1(1–8) Module #2(9–16) Module #3(17–23) Module #4(24–26) Module #5(27–31) Module #6(32–34) Module #7(35–40)	98% (196/200)	0.0200
5	50% (2, 5, 6, 9 and 10)	50% (1, 3, 4, 7 and 8)	4(176,133,113,238) 5(163,140,105,109,228) 4(155,120,131,37) 3(151,123,110) 4(163,97,136,120) 4(167,84,215,248) 3(136,72,76)	Module #1(1–6) Module #2(7–14) Module #3(15–17) Module #4(18–19) Module #5(20–28) Module #6(29–36) Module #7(37–40)	97.50% (195/200)	0.0250

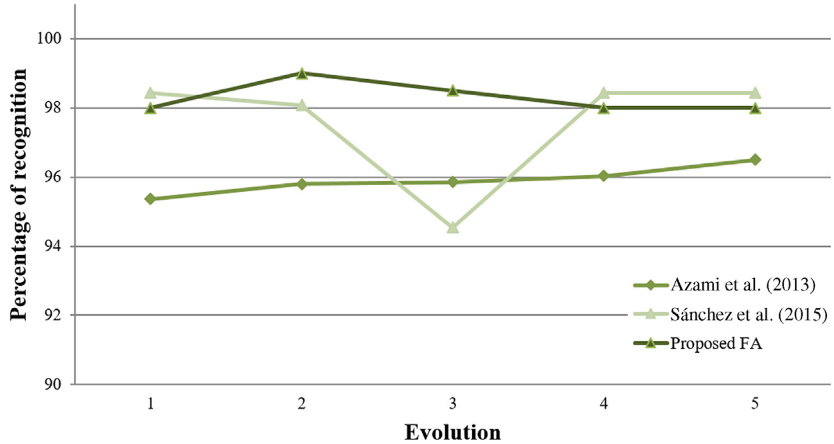

Fig. 22. Obtained recognition rates (up to 50%, face database, comparison 1).

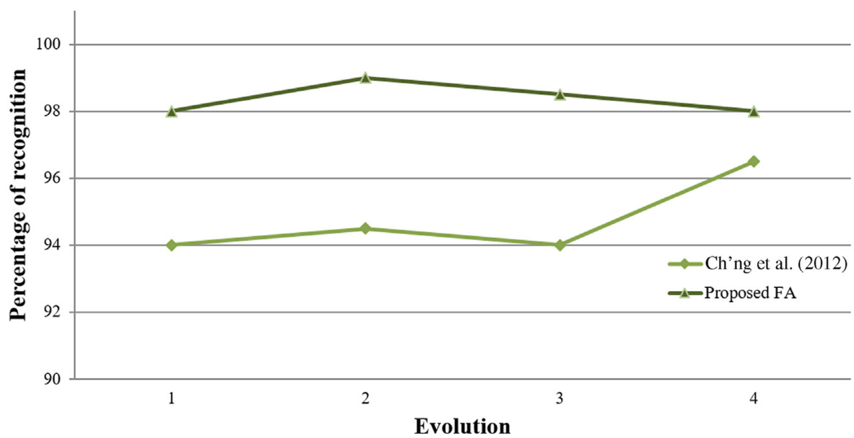

Fig. 23. Obtained recognition rates (up to 50%, face database, comparison 2).

method are shown. In these figures, it is possible to observe how the proposed method maintains stable results and most of the times they are better than the results achieved by the other works.

In Fig. 24, an analysis of results is presented. It can be noted that, as in the ear evolutions, for evolutions using up to 80% of data for training, while more information is learned by the MGNN, a higher number of sub granules is need, but the recognition error is smaller than in the other test. In the case of the evolutions using up to 50% of data for training, most of them use between 6 and 8 sub granules, but the recognition error increases.

3.3. Summary of results

In this section a summary results and comparisons with other works using the same databases and using neural networks are presented. The summary of results obtained with the proposed method applied to the ear database is shown in Table 7.

The comparison among the optimized results obtained using the proposed method and other optimized works for the ear database are presented in Table 8. In Sánchez and Melin (2014), the optimization of a conventional neural network was shown to compare results with the modular granular neural network. The MGNN in that work was also optimized using a hierarchical genetic algorithm. In Melin et al. (2012), modular neural network optimization is proposed using a hierarchical genetic algorithm, but without a granular approach, this means, the MNN always had the same number of sub-modules, only the parameters of the 3 sub-modules were optimized and the number of persons learned by each module was also left fixed.

The comparison among the optimized results obtained using the proposed method and other similar works for the face database are presented in Table 9. In Azami et al. (2013), a neural network is proposed based on a conjugate gradient algorithm (CGA) and a principal component analysis. In Ch'ng et al. (2012), the principal components analysis (PCA) and the linear discriminant analysis (LDA) methods are used. In Sánchez et al. (2015), modular neural network with a granular approach is used, but in that work, the granulation is performed using non-optimized trainings to assign a complexity level to each person and to form sub-granules with persons that have the same complexity level. That method was recommended for databases with a large number of people. In Mendoza et al. (2010), a comparison of fuzzy edge detectors based on the image recognition rate as performance index calculated for a neural network is proposed.

4. Statistical comparison of results

Statistical t tests were performed to verify if the results achieved by the proposed method have sufficient evidence to say that better results are obtained when a firefly algorithm is applied to perform an optimization of granular neural networks. This section is divided into 2 parts. First, the results using up to 80% of data, for the training phase, are compared and later, those using only 50%. In these t tests, the recognition rates previously presented were used.

4.1. Results using a percentage of data for training up to 80%

The values obtained in the t-test, between Sánchez and Melin (2014) and the proposed method, are presented in Table 10. The t-value was

Table 7
The summary of results.

Method	Number of images for training	Recognition rate (Best)	Recognition rate (Average)
Proposed method (Ear database)	3 (Up to 80%)	100%	99.69%
Proposed method (Ear database)	2 (Up to 50%)	98.05%	94.81%
Proposed method (ORL database)	8 (Up to 80%)	100%	99.27%
Proposed method (ORL database)	5 (Up to 50%)	99%	98.3%

Table 8
Table of comparison of optimized results (ear database).

Method	Number of images for training	Recognition rate (Best)	Recognition rate (Average)
Sánchez and Melin (2014) (ANN)	3	100%	96.75%
Melin et al. (2012) (MNN)	3	100%	93.82%
Sánchez and Melin (2014) (MGNN)	3	100%	99.69%
Proposed method (MGNN)	**3**	**100%**	**99.89%**
Sánchez and Melin (2014) (ANN)	2	96.10%	88.53%
Sánchez and Melin (2014) (MGNN)	2	98.05%	94.81%
Proposed method (MGNN)	2	97.40%	96.82%

Table 9
Table of comparison of optimized results (ORL Database).

Method	Images for training	Recognition rate Best (%)	Recognition rate Average (%)
Azami et al. (2013) (CGA + PCA)	5	96.5%	95.91%
Ch'ng et al. (2012) (PCA + LDA)	5	96.5%	94.75%
Sánchez et al. (2015) (MGNNs + Complexity)	5	98.43%	97.59%
Proposed method	5	99%	98.3%
Mendoza et al. (2010) (FIS)	8	97.50%	94.69%
Sánchez D. (MGNNs + Complexity)	8	100%	99.27%
Proposed method	**8**	**100%**	**99.9%**

Table 10
Values of the ear database (up to 80%).

	N	Mean	Standard deviation	Error standard deviation of the mean	Estimated difference	t value	P value	Degree of freedom
Sánchez and Melin (2014) (MGNN)	30	0.0030	0.0121	0.0022	0.00195	0.8303	0.4118	35
Proposed method	30	0.00108	0.00421	0.00077				

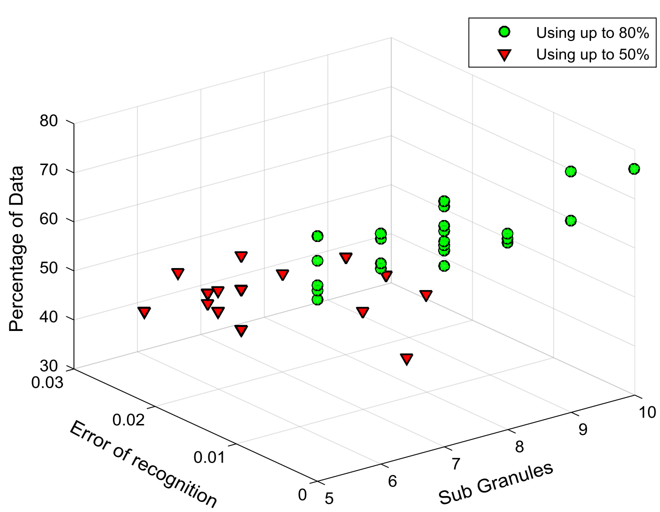

Fig. 24. Analysis of results (face database).

0.8303, which means that there is not sufficient evidence to say that the ear results were significantly improved with the proposed method.

The different values obtained in the t-test for the ORL database between the proposed method against (Mendoza et al., 2010; Sánchez et al., 2015) are presented in Table 11. The t-values were of 4.12 and 1.98, which means that there is sufficient evidence to say that the face results were significantly improved using the proposed method.

4.2. Results comparisons of tests using a percentage of data for training up to 50%

The values obtained in the t-test, between Sánchez and Melin (2014) and the proposed method, are presented in Table 12. The t-value was of 3.15, which means that there is sufficient evidence to say that face results were improved with the proposed method.

The different values obtained in the t-test for the ORL database, between the proposed method versus (Azami et al., 2013; Ch'ng et al., 2012; Sánchez et al., 2015) are presented in Table 13. The t-values were respectively 8.82, 5.65 and 0.91, which means that there is sufficient evidence to say that the face results were improved using the proposed method compared with Ch'ng et al. (2012) and Azami et al. (2013).

5. Conclusions

In this paper, the optimization of modular granular neural networks architectures using a firefly algorithm is proposed. The main goal is to minimize the recognition error. The main contribution of the method is the design of optimal architectures for human recognition applied to the ear and face as biometric measurements and focusing on the division of information into sub granules. The design of the MGNN architectures also consists in the optimization of the number of modules (sub granules), percentage of data for the training phase, goal error, learning algorithm, number of hidden layers and their respective

Table 11
Values of ORL database (up to 80%).

Test	N	Mean	Standard deviation	Error standard deviation of the mean	Estimated difference	t value	P value	Degree of freedom
Mendoza et al. (2010) (MG + FIS2)	4	94.69	2.58	1.3	−5.31	−4.12	0.026	3
Proposed method	4	100	0	0				
Sánchez et al. (2015) (MGNNs + Complexity)	5	99.27	0.676	0.30	−0.73	−2.42	0.072	4
Proposed method	5	100	0	0				

Table 12
Values of the ear database (up to 50%).

	N	Mean	Standard deviation	Error standard deviation of the mean	Estimated difference	t value	P value	Degree of freedom
Sánchez and Melin (2014) (MGNN)	30	0.0518	0.0345	0.0063	0.01999	3.15	0.0037	30
Proposed method	30	0.03182	0.00462	0.00084				

Table 13
Values of the ORL database (up to 50%).

Test	N	Mean	Standard deviation	Error standard deviation of the mean	Estimated difference	t value	P value	Degree of freedom
Azami et al. (2013) (CGA + PCA)	5	95.91	0.409	0.18	−2.390	−8.82	2.156E−05	8
Proposed method	5	98.30	0.447	0.20				
Ch'ng et al. (2012) (PCA + LDA)	4	94.75	1.19	0.60	−3.550	−5.65	0.004	4
Proposed method	5	98.30	0.447	0.20				
Sánchez et al. (2015) (MGNNs + Complexity)	5	97.59	1.71	0.76	−0.714	−0.91	0.406	5
Proposed method	5	98.30	0.447	0.20				

number of neurons. The achieved results are compared with other works to prove the effectiveness of the proposed method, mainly with a previous work, where MGNNs are optimized using a hierarchical genetic algorithm. Statistically the results achieved by the proposed method are better especially when the firefly algorithm can use up to 50% of the data for the training phase, but compared with other works, the proposed method was statistically better using up to 80 and 50 of percent of the information. As a conclusion, based on the results achieved by the proposed method better performance on human recognition is possible, mainly with respect to those works that do not use a granular approach. The main advantage of the firefly algorithm over the hierarchical genetic algorithm comes from its ability to compare each pair of fireflies, where a firefly is moved towards a brighter one, and this allows having several better solutions, unlike the genetic algorithm that only considers a better solution in each iteration. As future work, a multi-objective approach can be applied to minimize the complexity of MGNNs architectures, i.e. simultaneously reduce the number of sub granules (sub modules), percentage of information for the training phase, hidden layers and their neurons besides of error of recognition.

References

Azami, H., Malekzadeh, M., Sanei, S., 2013. A new neural network approach for face recognition based on conjugate gradient algorithms and principal component analysis. J. Math. Comput. Sci. 6, 166–175.

Balarini, J., Nesmachnow, S., Rodríguez, M., 2012. Facial recognition using neural networks over GPGPU. CLEI Electron. J. 15 (3), 1–12.

Bargiela, A., Pedrycz, W., 2006. The roots of granular computing. In: IEEE International Conference on Granular Computing, GrC, pp. 806–809.

Basu, J., Bhattacharyya, D., Kim, T., 2010. Use of artificial neural network in pattern recognition. Int. J. Softw. Eng. Appl. 4 (2), 23–34.

Chandana, C., Yadav, S., Mathuria, M., 2015. Fingerprint recognition based on minutiae information. Int. J. Comput. Appl. 120 (10), 39–42.

Ch'ng, S., Seng, K., Ang, L., 2012. Modular dynamic RBF neural network for face recognition. In: 2012 IEEE Conference on Open Systems, ICOS, pp. 1–6.

Database Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB). 2015. [Online]. http://www.ustb.edu.cn/resb/en/index.htm.

Database of Human Iris. 2015. Database of Human Iris Institute of Automation of Chinese Academy of Sciences (CASIA). [Online]. http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp.

Dorigo, M., 1992. Optimization, learning and natural algorithms. Ph.D. Thesis, Politecnico di Milano, Italy.

Eiben, A., Smith, J., 2015. Introduction To Evolutionary Computing. Springer.

Farooq, M, 2015. Genetic algorithm technique in hybrid intelligent systems for pattern recognition. Int. J. Innov. Res. Sci. Eng. Technol. 4 (4), 1891–1898.

Gutierrez, L., Melin, P., Lopez, M., 2010. Modular neural network integrator for human recognition from ear images. In: IJCNN, Barcelona, Spain, pp. 1–5.

Hassoun, M., 2003. Fundamentals of Artificial Neural Networks: A Bradford Book.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan Coll Div. .

Hidalgo, D., Castillo, O., Melin, P., 2009. Type-1 and type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms. Inform. Sci. 179 (13), 2123–2145.

Holland, J., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

Homayon, S., 2015. Iris recognition for personal identification using LAMSTAR neural network. Int. J. Comput. Sci. Inf. Technol. 7 (1), 1–8.

Kaur, G., 2016. A review on ear based biometric identification system. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 6 (3), 548–552.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of the IEEE international Joint Conference on Neuronal Networks, pp. 1942–1948.

KumarSrivastava, A., Singh, H., 2016. An enhance firefly algorithm for flexible job shop scheduling. Int. J. Comput. Appl. 6 (5), 1–17.

Man, K., Tang, K., Kwong, S., 1999. Genetic Algorithms: Concepts and Designs. Springer.

Martínez-Soto, R., Castillo, O., Aguilar, L., Rodríguez Díaz, A, 2015. A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers. Int. J. Mach. Learn. Cybern. 6 (2), 175–196.

Melin, P., Sánchez, D., Castillo, O., 2012. Genetic optimization of modular neural networks with fuzzy response integration for human recognition. Inform. Sci. 197, 1–19.

Mendoza, O., Melin, P., Castillo, O., Castro, J., 2010. Comparison of fuzzy edge detectors based on the image recognition rate as performance index calculated with neural networks. In: Soft Computing for Recognition Based on Biometrics. Springer, pp. 389–399.

Pastur-Romay, L., Cedrón, F., Pazos, A., Porto-Pazos, A., 2016. Deep artificial neural networks and neuromorphic chips for big data analysis: Pharmaceutical and bioinformatics applications. Int. J. Mol. Sci. 17, 1–26.

Raikova, R., Aladjov, H., 2002. Hierarchical genetic algorithm versus static optimization investigation of elbow flexion and extension movements, Vol. 35, pp. 1123–1135.

Rajabioun, R., 2011. Cuckoo optimization algorithm. Appl. Soft Comput. J. 11, 5508–5518.

Sánchez, D., Melin, P., 2014. Optimization of modular granular neural networks using hierarchical genetic algorithms for human recognition using the ear biometric measure. Eng. Appl. Artif. Intell. 27, 41–56.

Sánchez, D., Melin, P., Castillo, O., 2015. Optimization of modular granular neural networks using a hierarchical genetic algorithm based on the database complexity applied to human recognition. Inform. Sci. 309, 73–101.

Sankhe, A., Pawask, A., Mohite, R., Zagade, S., 2016. Biometric identification system: a finger geometry and palm print based approach. Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 5 (3), 1756–1763.

Solanki, K., Pittalia, P., 2016. Review of face recognition techniques. Int. J. Comput. Appl. 133 (12), 20–24.

Witten, I., Frank, E., Hall, E., 2011. Fuzzy Logic for the Management of Uncertainty. Morgan Kaufmann.

Yang, X., 2009. Firefly algorithms for multimodal optimization. In: Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications, Vol. 5792, pp. 169–178.

Yang, X., He, X., 2013. Firefly algorithm: Recent advances and applications. Int. J. Swarm Intell. 1 (1), 36–50.

Yao, Y., 2005. Perspectives of granular computing. In: IEEE International Conference on Granular Computing, GrC, pp. 85–90.

Zadeh, L., 1998. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Comput. 2, 23–25.

Zadeh, L., Kacprzyk, J., 1992. Fuzzy Logic for the Management of Uncertainty. Wiley-Interscience.

Zhang, Y., Yu, D., Seltzer, M., Droppo, J., 2015. Speech recognition with prediction-adaptation-correction recurrent neural networks. In: ICASSP, South Brisbane, QLD, pp. 5004–5008.

Zhong, C., Pedrycz, W., Wang, D., Li, L., Li, Z., 2016. Granular data imputation: A framework of Granular Computing. Appl. Soft Comput. 46, 307–316.