Optimization of Convolutional Neural Networks Using the Fuzzy Gravitational Search Algorithm

Submitted: 20th December 2019; accepted: 30th March 2020

Yutzil Poma, Patricia Melin, Claudia I. González, Gabriela E. Martínez

DOI: 10.14313/JAMRIS/1-2020/12

Abstract: *This paper presents an approach to optimize a Convolutional Neural Network using the Fuzzy Gravitational Search Algorithm. The optimized parameters are the number of images per block that are used in the training phase, the number of filters and the filter size of the convolutional layer. The reason for optimizing these parameters is because they have a great impact on performance of the Convolutional Neural Networks. The neural network model presented in this work can be applied for any image recognition or classification applications; nevertheless, in this paper, the experiments are performed in the ORL and Cropped Yale databases. The results are compared with other neural networks, such as modular and monolithic neural networks. In addition, the experiments were performed manually, and the results were obtained (when the neural network is not optimized), and comparison was made with the optimized results to validate the advantage of using the Fuzzy Gravitational Search Algorithm.*

Keywords: *Neural Networks, Convolutional Neural Network, Fuzzy Gravitational Search Algorithm, Deep Learning*

1. Introduction

Convolutional neural networks (CNN) are a deep learning architecture that is inspired by the visual structure of living system [1].

In 1962 Hubel and Wiesel performed work on the primary visual cortex of a cat and found that the cells in the visual cortex are sensitive to small sub-regions of the visual field, called the receptive field. These cells are responsible for the detection of light in the responsive field [1]. The first simulated model in the computer which was inspired by the works of Hubel and Wiese is the Neocognitron that was proposed by Fukushima. This network is considered as the predecessor of CNN and was based on the hierarchical organization of neurons for the transformation of an image [2].

The CNN helps to identify and classify images, which are adapted to process data in multidimensional arrays. One of the main advantages of using these neural networks is that they reduce the number of connections and the number of parameters to be trained compared to the fully connected neural network. The first time that a convolutional neural network was used was for the recognition of handwritten digits using a neural network with back-propagation [3]. In recent years, fully connected networks have been employed in several applications, such as in the optimization of a modular neural network (MNN) that applies a swarm of particles with a fuzzy parameter [4]. In [5], modular neural networks are utilized for pattern recognition using the ant colony paradigm for network optimization, in addition, traditional neural networks (NN) are adopted for facial recognition using as a pre-processing a fuzzy edge detector [6]. In another work using the integration of an MNN based on the integral of Choquet with Type-1 and Type-2 applied to face recognition [7]. Another application is the design of a hybrid model using modular neural networks and fuzzy logic to provide the diagnosis of a person's risk of hypertension [8]. Optimization of neural networks using a genetic algorithm (GA) and the method of Particle swarm optimization (PSO) we presented in [9]. Other works are the use of genetic optimization of MNN with fuzzy response integration [10]. The optimization of the modular neuronal network based on a hierarchical multi-objective genetic algorithm [11]. Also the optimization of the modular granular neuronal network using the fireflies algorithm [12]. The optimization of the weights of a neural network using GA and PSO, using supervised backpropagation learning and a Type-2 fuzzy system [13], or in the implementation of a new model of neural networks, which is based on the Learning Vector Quantization (LVQ) algorithm for the classification of multiple arrhythmias [14].

Recently, the CNNs have been used in various applications, such as in the reading of system checks, where character recognizers are utilized, combined with global training techniques [15]. They have also been applied in the automatic detection and blurring of plates and faces in order to protect privacy in Google Street View [16]. There are some experimental applications in which these networks have been used in obstacles detection at a great distance, employing a deep hierarchical network trained to extract significant characteristics of an image, where the classifier can predict the transfer capacity in real time, this views obstacles and paths between 5 to more than 100 meters and is adaptive [17].

The aim of a CNN is the extraction of characteristics of the images and, to improve the obtained results, optimization methods that generate better solutions are applied. One of these many methods that exist to optimize is the Gravitational Search Algorithm (GSA), which is based on Newton's law of gravity, and another of these optimization methods is the Fuzzy Gravitational Search Algorithm (FGSA) [18], which is a variation of the Gravitational Search Algorithm (GSA) [19], but unlike its predecessor, it changes the Alpha parameter through a fuzzy system which tends to increase or decrease, in comparison with other methods where Alpha has a static value [20-23].

The main contribution of this paper is the proposed optimization of a Convolutional Neural Network, with the FGSA method, which obtains the number of images per block (Bsize) for the training phase in the CNN, the number of filters in the convolutional layer and, finally, the filter size in the same layer.

The paper is structured as follows, Section 2 presents the background about the basic concepts of the CNNs. Section 3 describes the proposed method to optimize the convolutional neuronal network using the FGSA method. Section 4 explains the results obtained when the Bsize value, the filter size and the number of filters are optimized for the FGSA method, the same values are changed manually in both cases (ORL and CROPPED YALE). Finally, Section 5 presents some conclusions of the general experimentation achieved by the case studies presented.

2. Literature Review

This Section presents the basic concepts necessary to understand the proposed method.

2.1. Deep Learning

A deep learning architecture is a CNN that is inspired by the visual structure. It is an automatic learning technique, which allows computers to be taught to do what is natural for humans; they learn based on examples. A computer model can learn to perform classification tasks from sounds, text or images [24]. Knowing the hierarchy of concepts allows the computer to learn simple concepts to more complicated concepts [25]. Deep learning achieves impressive results thanks to recognition accuracy.

It requires data labeled in large quantities, in addition to a significant power of calculation, for this reason, they help GPUs because their high performance and parallel architecture are more efficient for their processes [26].

Deep learning models are trained using extensive sets of neural network architectures and tagged data, they learn directly with the data, without the need for manual extraction of features such as the use of data pre-processing methods.

2.2. Deepness

In a discrete mathematics architecture, depth refers to the depth of the corresponding graph or drawing, that is, the longest path from an input node to an output node. In the neural network, the depth corresponds to the number of layers of the neural network [27]. Traditional neural networks have 2 to 3 hidden layers, while deep networks have up to 150 layers.

Learning methods use neural network architectures, so why deep learning models are called "deep neural networks".

2.3. Convolutional Neural Networks

Convolutional Neural Networks or also called ConvNet, are a very popular type of deep neural networks, which perform feature extraction of characteristics of the input data. It is constituted by different types of layers, each of which obtains important characteristics. In the end, it classifies the characteristics of the image, resulting in the corresponding recognition [28].

CNN has gone through a phase of evolution in which some publications have established more efficient ways to train these networks using GPUs [29-30].

2.4. Convolution Layer

This layer generates new images called "Character Map", which accentuates the unique characteristics of the input data. This layer contains filters (kernels) that convert the images into new images, they are called "Convolution Filters" and consist of two-dimensional arrays of 5 * 5, and in recent applications up to 1 * 1 have been used. The convolution is represented in (1).

$$\mathbf{R}(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{i=1}}^{\mathbf{k}} \sum_{\mathbf{j=1}}^{\mathbf{p}} \mathbf{M}(\mathbf{i},\mathbf{j}) * \mathbf{A}(\mathbf{x}+\mathbf{1},\mathbf{y}+\mathbf{j}) \tag{1}$$

where:
M: represents the mask,
i: is the mask line,
j: is the column of the mask,
A: is the image,
x: is the row of the characteristics matrix,
y: is the column of the characteristics matrix,
A: is the characteristics matrix,
k: is the row of the filter size,
p: is the column of the filter size.

2.5. Non-Linearity Layer

Several activation functions are applied after the convolution layer. The most commonly used activation functions are normally hyperbolic tangent, sigmoid and rectified linear units (ReLU). Compared to other functions ReLU, is preferable for CNNs because these networks train faster [31].

2.6. Pooling Layer

Also called "grouping" layer, it is responsible for reducing the size of the image and combines the neighboring pixels in a certain area, taking small blocks of the convolution layer and sub-samples to obtain an output, or a single representative value [32-33], which consists of a set of pixels, of whose average or maximum is calculated [34] as the case may be.

2.7. Classifier Layer

After convolution and layer accumulation, a fully connected layer is used, in which each pixel is a separate neuron as a multilayer perceptron. This layer has as many neurons as the number of classes to predict, in this layer the neural network recognizes or classifies the images that it will obtain as output [35-37].

In the CNN, there is an internal process that defines the number of times that will be trained (Batch), as well as the number of images (Block / Bsize) that will be included in the CNN training.

2.8. Fuzzy Gravitational Search Algorithm

The Fuzzy gravitational search algorithm is a method, based on agents, that has been used in several applications, such as the optimization of modular neural networks in pattern recognition [38] and the optimization of modular neural networks in the recognition of echocardiograms [39].

In this method, agents are objects that are determined by their masses. All objects are attracted to each other, thanks to the force of gravity, in turn, causes a global movement of all objects and maintains a direct communication with the masses.

As the Alfa parameter changes, different gravitation and acceleration can be obtained for each agent, which improves FGSA performance.

The Alfa parameter was optimized by means of a fuzzy system, where the ranges were determined to give a wider value to look for the Alpha [18]. It was decided to use the fuzzy variables: Low, Medium and High with triangular membership functions, which are the following: Low: [-50 0 50], Medium [0 50 100], High [50 100 150].

The fuzzy system with which the new Alfa is obtained has 3 fuzzy rules which are:

1. If the Iteration is Low then the Alpha is low.
2. If the Iteration is Medium then the Alpha is medium.
3. If the Iteration is High then the Alpha is High.

Figure 1 shows the flow diagram of the FGSA method, which generates the initial population and evaluates the fitness for each of the agents, updates the value of G, which is the gravitation and provides the best and worst agent of the population then subsequently calculates M that is the mass and with the help of the fuzzy system obtains the value of alpha, which is the acceleration, updates the speed and position, and finally returns the best solution found [18]. In Figure 2 we can find the fuzzy system to obtain the new alpha value.

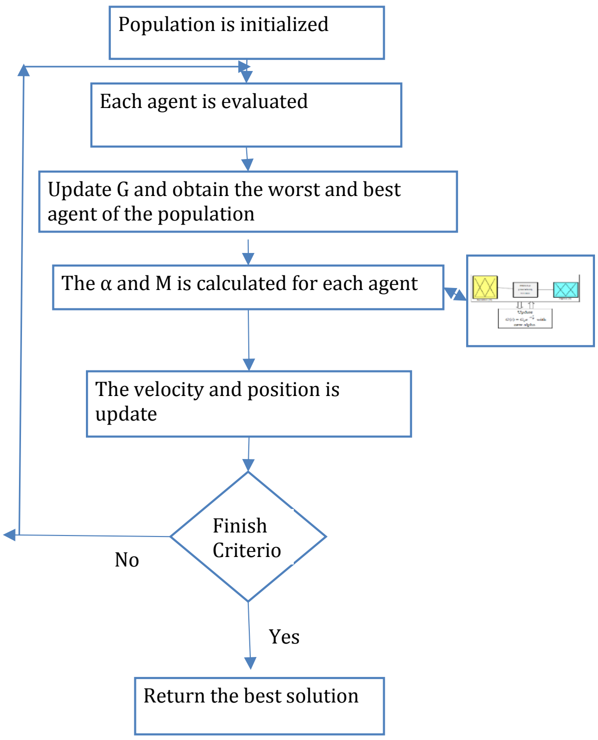

Fig. 1. The Flow chart of FGSA [18]

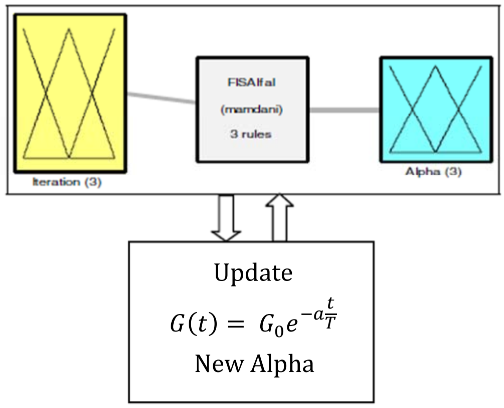

Fig. 2. Fuzzy System for the new alpha parameter [18]

3. Proposed Method

The proposed model begins with the input images (database), then the FGSA method will be responsible for optimizing the neural network, thus obtaining the best architecture for CNN. Figure 3 shows the detailed general proposal where the input data are entered (images from the ORL or CROPPED YALE database), continuing the interaction between the FGSA and the convolutional neuronal network. Figure 4 details the FGSA and CNN method, where together they work to obtain a higher percentage of recognition, since the FGSA generates a random matrix that is passed to CNN and in it each agent (vector of the initial matrix) is evaluated, in each agent the values that will be optimized are given by "Bsize", the number of filters and the filter size, which will then be evaluated in the neural network, ending with the highest recognition rate of the database used.

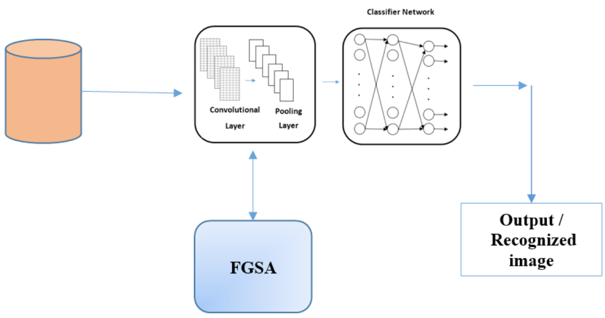

Fig. 3. General proposal

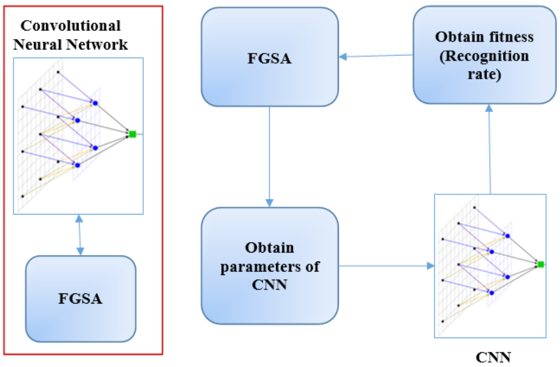

Fig. 4. Details when the CNN works together with FGSA

The FGSA generates a matrix of possible solutions in a designated range, for the Bsize it was searched between 10 to 100 random values to be generated, for the filter size between 1 to 10 and for the number of filters between 10 to 50. These values were considered because tests were performed, where each of the values to be optimized were modified manually and, therefore an estimated range of the best values to be used was obtained, in order to have a satisfactory result in the recognition of the images. In Table 1 we can notice a red rectangle, which designates the "Initial Matrix" that the FGSA generates in a random way, each row of the initial matrix is an agent (shown in the green rectangle) that has 3 dimensions: the first one is the value of the Bsize, the second value corresponds to the number of filters and finally the third value designates the value of the filter size of the convolution layer.

Tab. 1. Initial Matrix

Number of agent	Bsize (10-100)	Number of filters (10-50)	Filter size (1-10)
1	37	10	1
2	10	20	3
3	24	38	9
.			
.			
.			
15	60	42	10

3.1. Architecture of the CNN

In Table 2 shows the architecture of CNN, which will be used for each case study.

Tab. 2. Architecture of CNN

Layer	Observation	Activation function
Input	M * N	-
Convolution	FGSA Obtain number and Size of filter of convolution (x*x)	ReLU
Pooling	1 layer, medium (2*2)	-
Hidden layers	100 nodes	ReLU
Output	40/38 nodes	Softmax

3.2. Variables to Be Optimized

For the CNN optimization, three main variables were selected, which will help the network obtain a better recognition percentage.

Based on previous experiments [40], it was shown that varying the value of Bsize has a great influence on the training of the network, since the Bsize selects the training data and calculates the adjustment or updating of the weights, this contributes to the network training faster because it repeats the process fewer times and this decreases the training time.

The variables that were considered to be optimized are the following:

1. The number of blocks: is the variable that blocks all the images that will enter the training stage in the convolutional neural network.
2. The number of filters: this parameter is used in the convolution layer, it is the number of filters to be used in this layer, which obtain the characteristics map.
3. Filter size: is the variable that is used to define the filter size of the convolution layer, which extracts the data to form the characteristics map.

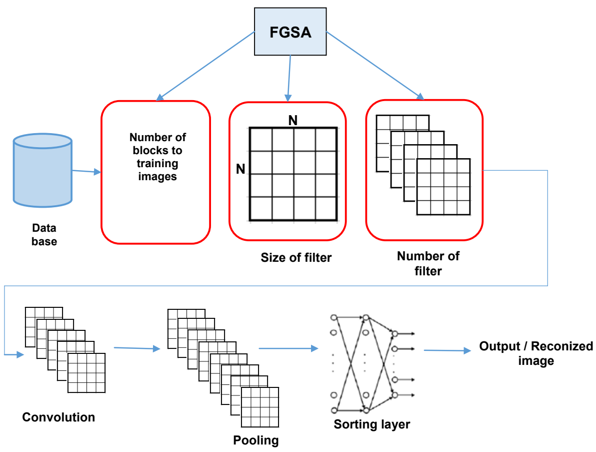

Fig. 5. Detail of process when FGSA optimized variables

Figure 5 shows the detailed development of how the FGSA method contemplates the optimization of the considered variables for a better result of the image pattern recognition of the CNN, using the variables of Bsize which, as mentioned above, are responsible for dividing the input images to pass to network training, the number of filters that determines the number of feature maps to be extracted from the image and the size of the filter in the convolution layer, which takes the sample size to form the feature map.

4. Results and Discussion

In this experiment, tests were performed with a CNN using the FGSA to optimize it, using for this case study the ORL database, which contains 400 images of human faces, this consists of 40 humans and 10 images taken at different angles of the face of each of them, with a size of 112 * 92 pixels in .pgm format for each image. In Figure 6 we can find some examples of the ORL database, the parameters used for the CNN are shown in Table 3, as well as the parameters used in the FGSA method are presented in Table 4, once this experiment was completed, it was concluded with the results presented in Table 5, where the highest recognition value is 91.25% when the value of Bsize is 37.

Table 6 presents the results obtained from the manual modification of the "Bsize" value in the CNN using 20 filters of 9*9, it is verified that the optimal value (obtained by the FGSA) is when the value of Bsize is 37. The test was performed where the Bsize value is modified from 10 in 10 to reach the value of 100. With this test, the highest recognition rate was 91.25 %.

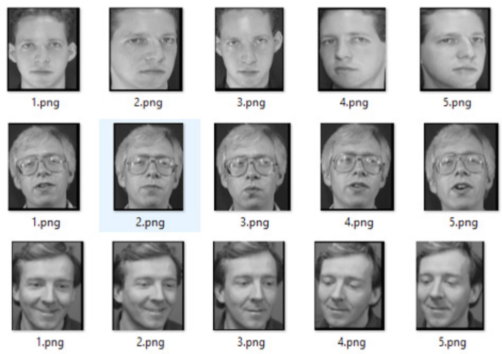

Fig. 6. Images from the ORL database

Tab. 3. Parameters of CNN for the ORL database

CNN parameters	
Epochs	50
Total Images	400
Training Images	320
Testing Images	80
Number of blocks to training /Bsize	FGSA
Size of filter	9*9
Number of filter	20

Tab. 4. Parameters of the FGSA

FGSA parameters	
Iterations	15
Agents	15
Dimension	3

Tab. 5. Results of Bsize optimized with FGSA

No. Experiment	Recognition rate (%)	Value Bsize
1	90	19
2	91.25	37
3	91.25	37
4	91.25	37
5	91.25	37
6	91.25	37
7	91.25	37
8	91.25	37
9	91.25	37
10	91.25	37
11	91.25	37
12	91.25	37
13	91.25	37
14	91.25	37
15	91.25	37

Tab. 6. Bsize modified manually from CNN using ORL without FGSA

Value of Bsize	Recognition rate (%)	Time	
		Seconds	Minutes
10	81.25	855.39	14.25
20	87.5	931.39	15.52
30	72.5	840.59	14.00
37	91.25	1022.20	17.03
40	1.25	778.97	12.98
45	87.5	860.46	14.34
50	52.5	825.60	13.76
60	52.5	953.76	15.89
70	56.25	891.44	14.85
80	33.75	1116.92	18.61
90	42.5	928.33	15.47
100	7.5	956.11	15.93

Once the FGSA has found the best Bsize parameter with a value of 37, it starts with the optimization of the next value. For this test, a simulation was performed, where the value of the number of filters used in the convolution layer is modified manually. As a result of this test, the best value for the number of filters in the network is 20, because it consumes less resources and processing time, although when the number of filters is 50, it also reaches the same percentage of recognition that is 91.25 %, but with the difference that it consumes more resources and time. The results can be found in Table 7.

Tab. 7. Number of filter is manually modified

Number of filters	Recognition rate (%)	Time	
		Seconds	Minutes
10	36.25	586.00	9.76
15	3.75	926.46	15.44
20	**91.25**	**1170.00**	**19.51**
30	73.75	1720.00	28.64
40	81.25	2350.00	39.10
50	91.25	2530.00	42.14
60	87.5	4600.00	76.74
70	67.5	5830.00	97.15
80	6.25	4630.00	77.22
90	90	5130.00	85.55
100	46.25	8500.00	141.59

In Table 8 the results of 15 experiments can be observed, where the Bsize value is 37; the value of the number of filters of the convolution layer belonging to the CNN was optimized, obtaining that the best result is when the number of filters in this layer is 20, which results in a recognition of 91.25% of the ORL image database.

Tab. 8. Number of filters optimized with FGSA

No. Experiment	Number of filters	Recognition rate (%)
1	48	73.75
2	**20**	**91.25**
3	20	91.25
4	20	91.25
5	20	91.25
6	20	91.25
7	20	91.25
8	20	91.25
9	20	91.25
10	20	91.25
11	20	91.25
12	20	91.25
13	20	91.25
14	20	91.25
15	20	91.25

In Table 9 it can be noted that the FGSA method optimized the filter size parameter of the convolution layer, using the optimal fixed parameters of Bsize in 37 and the number of filters in 20. The best value obtained for the optimized parameter is 3*3, resulting in a recognition percentage of 93.75 %.

Tab. 9. Size of filter optimized with FGSA

No. Experiment	Filter Size	Recognition rate (%)
1	35*35	88.75
2	**9*9**	**91.25**
3	**3*3**	**93.75**
4	3*3	93.75
5	3*3	93.75
6	3*3	93.75
7	3*3	93.75
8	3*3	93.75
9	3*3	93.75
10	3*3	93.75
11	3*3	93.75
12	3*3	93.75
13	3*3	93.75
14	3*3	93.75
15	3*3	93.75

Tab. 10. Size of filter manually modified

Size of filter	Recognition rate (%)	Seconds
1*1	77.5	1208.93
3*3	**93.75**	**1241.41**
5*5	78.75	1246.49
7*7	2.5	1208.90
8*8	0	2.89
9*9	91.25	1147.19
10*10	0	2.48
11*11	68.75	1279.61
12*12	0	2.42
13*13	87.5	1399.91
14*14	0	2.44
15*15	83.75	1575.83
17*17	88.75	1797.61
19*19	86.25	4094.52
21*21	83.75	2543.02
23*23	87.5	2848.60
25*25	87.5	2118.82
27*27	81.25	2764.26
29*29	85	2295.67
31*31	76.25	2326.02
33*33	81.25	2079.25
35*35	88.75	2230.88
37*37	77.5	2202.14
39*39	78.75	2271.43
41*41	83.75	2707.54
43*43	80	2796.67
45*45	3.75	2749.67
47*47	82.5	3190.60
49*49	80	3133.40
50*50	0	5.07

Tab. 11. Comparative with others methods using the ORL database

Preprocessing Method	Type of network	Optimization Method	Integrator of response	Recognition rate (%) Max
None [41]	Monolithic Neural Network	No	Not apply	5
Sobel operator [41]	Monolithic Neural Network	No	Not apply	5
Sobel + T1FLS [41]	Monolithic Neural Network	No	Not apply	93.75
Sobel + IT2FLS [41]	Monolithic Neural Network	No	Not apply	95
Sobel + GT2 FLS [41]	Monolithic Neural Network	No	Not apply	96.5
Prewitt operator [41]	Monolithic Neural Network	No	Not apply	5
Prewitt + T1FLS [41]	Monolithic Neural Network	No	Not apply	93.75
Prewitt + IT2FLS [41]	Monolithic Neural Network	No	Not apply	95
Prewitt + GT2 FLS [41]	Monolithic Neural Network	No	Not apply	96.85
Gradient Morphologic [42]	Modular Neural Network (3 Modules)	No	Sugeno Integral	87.22
IT1MGFLS [42]	Modular Neural Network (3 Modules)	No	Sugeno Integral	88.6
IT2MGFLS [42]	Modular Neural Network (3 Modules)	No	Sugeno Integral	85.98
GM [42]	Modular Neural Network (3 Modules)	No	Choquet Integral	88
IT1MGFLS [42]	Modular Neural Network (3 Modules)	No	Choquet Integral	92.59
IT2MGFLS [42]	Modular Neural Network (3 Modules)	No	Choquet Integral	91.9
Not apply*	**Convolutional Neural Network**	**FGSA**	**Not apply**	**93.75**

In order to confirm the results presented in Table 9, a manual experiment was carried out. In this test, the Bsize and the number of filters are fixed, with a value of 37 and 20 respectively; but, on the other hand, the filter size parameter was modified manually. It was observed that when the size of the filter is an even number, at the time of pooling the data does not coincide with the operations performed, whereas when the filter size is an odd number, the pooling operations conclude without problems. Table 10 shows the results achieved in this manual experiment; where the best obtained result is when the filter size has a value of 3 * 3 with a recognition rate of 93.75%.

Table 11 presents a comparison of other methods and their maximum recognition percentages obtained using different neural networks, such as monolithic and modular neural networks. The results of the CNN are compared with the results published in other works where other types of neural networks are implemented, some of the results of these networks are better than those of CNN because they use methods of preprocessing, response integrators, modularity as well as cross-validation methods for data selection. Table 12 shows the data used in the performed experiments.

In order to verify that the result of the FGSA method is the best, the manual test was performed where the Bsize parameter is modified with values of 10 in 10 in which it can be verified that the best result is when the Bsize value is 38, we can visualize these results in Table 16, since; although with other parameters (20 and 30) we also obtain the same percentage of recognition in the solution that the FGSA produces, it takes less time; therefore, it uses less processing time.

Tab. 12. Data used for the comparison with others methods

Information	Values
Data	400
Training	80%
Testing	20%

In other case of study the CROPPED YALE database was used, which contains 380 images of human faces of 192 * 168 pixels in .pgm format, they are 38 people with 10 images for each of them, some examples of this data base are present in Figure 7. Table 13 details the parameters used for the CNN, as well as in Table 14, describes the parameters used of the FGSA method.

In Table 15 we can find the results obtained by running the CNN using the FGSA to optimize the Bsize variable, where the best value for Bsize is 38 with a 100 % recognition rate.

Fig. 7. Example of CROPPED YALE database

Tab. 13. CNN parameters used for the CROPPED YALE database

CNN parameters	
Epochs	50
Total Images	380
Training Images	304
Testing Images	76
Number of block for images to training /Bsize	FGSA
Size of filter	9*9
Number of filter	20

Tab. 14. FGSA parameters used with the case it CROPPED YALE

FGSA parameters	
Iterations	15
Agents	15
Dimensions	3

As can be seen in Table 17 the best value (38) of the Bsize parameter was maintained, this value was obtained based on the performed experiments. Manual tests were performed in which, the number of filters was modified, having as best recognition rate (100%) that the number of filters of the convolution layer is 20; although the same result was reached with the values of 50,60,70,80,90 and 100 in the number of filters, these require more computational resources and time to reach the same result. For this reason, it is concluded that the best value is 20, since it reaches the best recognition percentage in less time.

In Table 18, the results of 15 experiments are presented, the value of the Bsize is 38 and the number of filters of the convolution layer was optimized with the help of the FGSA method, thus achieving the best result in the number of filters, obtaining 100% image recognition from the CROPPED YALE database.

Tab. 15. Bsize optimized using FGSA

No. Experiment	Value of Bsize	Recognition rate (%)
1	10	98.65
2	**38**	**100**
3	38	100
4	38	100
5	38	100
6	38	100
7	38	100
8	38	100
9	38	100
10	38	100
11	38	100
12	38	100
13	38	100
14	38	100
15	38	100

Tab. 16. Results when the "Bsize" is manually modified

Bsize	Rate recognition	Time	
		Seconds	Minutes
10	98.68	3007.37	50.12
20	**100**	**2991.79**	**49.86**
30	**100**	**3294.58**	**54.90**
38	**100**	**2984.36**	**49.73**
40	13.15	3024.60	50.41
50	2.63	3240.29	54.00
60	2.63	2833.34	47.22
70	2.63	9213.57	153.55
80	26.31	3053.14	50.88
90	3.94	4570.23	76.17
100	5.26	3549.22	59.15

Tab. 17. Results when the number of filters is modified manually

Number of filters	Recognition rate (%)	Time	
		Seconds	Minutes
10	6.58	1706.20	28.436
15	5.26	2424.70	40.41
16	3.95	3214.20	53.57
17	2.63	3726.50	62.10
18	97.37	3891.60	64.86
19	6.58	4043.50	67.39
20	**100**	**3436.30**	**57.27**
30	9.21	155690.00	2594.83
40	5.26	6507.10	108.45
50	100	8264.20	137.73
60	100	9710.80	161.84
70	100	10833.00	180.55
80	100	14942.00	249.03
90	100	12748.00	212.46
100	100	35962.00	599.36

Tab. 18. Results when the number of filters is optimized with FGSA

No. Experiment	Number of filter	Recognition rate (%)
1	18	97.37
2	**20**	**100**
3	20	100
4	20	100
5	20	100
6	20	100
7	20	100
8	20	100
9	20	100
10	20	100
11	20	100
12	20	100
13	20	100
14	20	100
15	20	100

Next, the data used for several experiments can be found in Table 19, thus having the comparison of other methods using different types of neural network such as Modular and Monolithic and their percentages of recognition in Table 20, where they used response integrators, as well as the pre-processing of images, modularity and cross-validation methods for data selection.

Tab. 19. Data used for comparative methods using the CROPPED YALE database

Information	Values
Total data	380
Training	70% – 80%
Testing	20% – 30%

4.1. The Best Architectures

Table 21 shows the collection of the best architectures found from the optimized values (Bsize, Number of filters and Filter Size), performing 15 experiments per each database. For the ORL database we obtained the 93.75% recognition rate when the nu m ber of blocks or Bsize are 37 and the filter size are 3*3, while for the CROPPED YALE database obtained 100% recognition rate when the parameter Bsize is 38, in Both cases the number of filters is 20.

5. Conclusion

Based on the experiments performed with the Convolutional Neuronal Network, it is concluded that these neural networks help the recognition of images, since, they are designed for these uses; however, if an optimization method is applied to these CNN, such as in this case it was the FGSA method, they have better results and help to obtain the architecture to arrive at a more optimal solution in pattern recognition applications.

Tab. 20. Comparative CNN with others methods using CROPPED YALE database

Preprocessing method	Type of network	Optimization method	Integrator of response	Recognition rate Max
Sobel T1 [7]	Modular Neural Network (5 Modules)	No	Choquet Integral	100
Sobel T2 [7]	Modular Neural Network (5 Modules)	No	Choquet Integral	100
None [41]	Monolithic Neural Network	No	Not apply	6.57
Sobel Operator [41]	Monolithic Neural Network	No	Not apply	2.63
Sobel + T1FLS [41]	Monolithic Neural Network	No	Not apply	100
Sobel + IT2FLS [41]	Monolithic Neural Network	No	Not apply	100
Sobel + GT2 FLS [41]	Monolithic Neural Network	No	Not apply	100
Prewitt Operator [41]	Monolithic Neural Network	No	Not apply	5.26
Prewitt + T1FLS [41]	Monolithic Neural Network	No	Not apply	100
Prewitt + IT2FLS [41]	Monolithic Neural Network	No	Not apply	100
Prewitt + GT2 FLS [41]	Monolithic Neural Network	No	Not apply	100
IT1MGFLS [42]	Modular neural network (3 Modules)	No	Sugeno Integral	99.91
IT2MGFLS [42]	Modular neural network (3 Modules)	No	Sugeno Integral	99.41
IT1MGFLS [42]	Modular neural network (3 Modules)	No	Choquet Integral	99.86
IT2MGFLS [42]	Modular neural network (3 Modules)	No	Choquet Integral	98.5
Not apply*	**Convolutional Neural Network**	**FGSA**	**Not apply**	**100**

Tab. 21. The best architecture for each case

Image Database	Number of blocks/ Bsize	Number of filters	Filter size	Recognition rate (%)
ORL	37	20	3*3	93.75
CROPPED YALE	38	20	9*9	100

It was demonstrated that the optimization methods are reliable and the obtained results with these are the same as the tests performed manually, where the values of "Bsize", number of filters and filter size were varied in the CNN, which, verifies that the optimization method (FGSA) represents a good way to find and build the best architecture of the network. Resulting in a high recognition percentage in the case studies presented.

It is also planned to optimize other parameters of the CNN, as well as to search for another architecture of the network, modifying the number of layers and neurons per layer in the classification or adding convolution, to obtain better models and these can be applied to any pattern recognition problem.

It is considered that with more time we can perform deeper explorations, such as increasing the number of agents in the FGSA and the number of iterations/experiments, in this way, we will probably obtain better percentages of recognition and reduce the processing time. As future work, type-2 fuzzy logic could be incorporated to improve results [43-44].

ACKNOWLEDGEMENTS

We thank our sponsor CONACYT and Tijuana Institute of Technology for the financial support provided with the scholarship number 816488.

AUTHORS

Yutzil Poma – Tijuana Institute of Technology, B.C., Tijuana, México, e-mail: yutpoma@hotmail.com.

Patricia Melin* – Tijuana Institute of Technology, B.C., Tijuana, México, e-mail: pmelin@tectijuana.mx.

Claudia I. González – Tijuana Institute of Technology, B.C., Tijuana, México, e-mail: cgonzalez@tectijuana.mx.

Gabriela E. Martinez – Tijuana Institute of Technology, B.C., Tijuana, México, e-mail: gmartinez@tectijuana.mx.

*Corresponding author

REFERENCES

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", *The Journal of Physiology*, vol. 160, no. 1, 1962, 106–154.

[2] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological Cybernetics*, vol. 36, no. 4, 1980, 193–202, DOI: 10.1007/BF00344251.

[3] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network". In: D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 2*, 1990, 396–404.

[4] D. Sánchez, P. Melin and O. Castillo, "Fuzzy Adaptation for Particle Swarm Optimization for Modular Neural Networks Applied to Iris Recognition". In: P. Melin, O. Castillo, J. Kacprzyk, M. Reformat and W. Melek (eds.), *Fuzzy Logic in Intelligent System Design*, 2018, 104–114, DOI: 10.1007/978-3-319-67137-6_11.

[5] F. Valdez, O. Castillo and P. Melin, "Ant colony optimization for the design of Modular Neural Networks in pattern recognition". In: *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, 163–168, DOI: 10.1109/IJCNN.2016.7727194.

[6] C. I. Gonzalez, J. R. Castro, O. Mendoza and P. Melin, "General type-2 fuzzy edge detector applied on face recognition system using neural networks". In: *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, 2325–2330, DOI: 10.1109/FUZZ-IEEE.2016.7737983.

[7] G. E. Martínez, P. Melin, O. D. Mendoza and O. Castillo, "Face Recognition with a Sobel Edge Detector and the Choquet Integral as Integration Method in a Modular Neural Networks". In: P. Melin, O. Castillo and J. Kacprzyk (eds.), *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*, 2015, 59–70.

[8] P. Melin, I. Miramontes and G. Prado-Arechiga, "A hybrid model based on modular neural networks and fuzzy systems for classification of blood pressure and hypertension risk diagnosis", *Expert Systems with Applications*, vol. 107, 2018, 146–164, DOI: 10.1016/j.eswa.2018.04.023.

[9] F. Valdez, P. Melin and O. Castillo, "Modular Neural Networks architecture optimization with a new nature inspired method using a fuzzy combination of Particle Swarm Optimization and Genetic Algorithms", *Information Sciences*, vol. 270, 2014, 143–153, DOI: 10.1016/j.ins.2014.02.091.

[10] P. Melin, D. Sánchez and O. Castillo, "Genetic optimization of modular neural networks with fuzzy response integration for human recognition", *Information Sciences*, vol. 197, 2012, 1–19, DOI: 10.1016/j.ins.2012.02.027.

[11] P. Melin and D. Sánchez, "Multi-objective optimization for modular granular neural net-

works applied to pattern recognition", *Information Sciences*, vol. 460-461, 2018, 594–610, DOI: 10.1016/j.ins.2017.09.031.

[12] D. Sánchez, P. Melin and O. Castillo, "Optimization of modular granular neural networks using a firefly algorithm for human recognition", *Engineering Applications of Artificial Intelligence*, vol. 64, 2017, 172–186, DOI: 10.1016/j.engappai.2017.06.007.

[13] F. Gaxiola, P. Melin, F. Valdez, J. R. Castro and O. Castillo, "Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO", *Applied Soft Computing*, vol. 38, 2016, 860–871, DOI: 10.1016/j.asoc.2015.10.027.

[14] P. Melin, J. Amezcua, F. Valdez and O. Castillo, "A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias", *Information Sciences*, vol. 279, 2014, 483–497, DOI: 10.1016/j.ins.2014.04.003.

[15] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, DOI: 10.1109/5.726791.

[16] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven and L. Vincent, "Large-scale privacy protection in Google Street View". In: *2009 IEEE 12th International Conference on Computer Vision*, 2009, 2373–2380, DOI: 10.1109/ICCV.2009.5459413.

[17] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller and Y. LeCun, "Learning long-range vision for autonomous off-road driving", *Journal of Field Robotics*, vol. 26, no. 2, 2009, 120–144, DOI: 10.1002/rob.20276.

[18] A. Sombra, F. Valdez, P. Melin and O. Castillo, "A new gravitational search algorithm using fuzzy logic to parameter adaptation". In: *2013 IEEE Congress on Evolutionary Computation*, 2013, 1068–1074, DOI: 10.1109/CEC.2013.6557685.

[19] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences*, vol. 179, no. 13, 2009, 2232–2248, DOI: 10.1016/j.ins.2009.03.004.

[20] O. P. Verma and R. Sharma, "Newtonian Gravitational Edge Detection Using Gravitational Search Algorithm". In: *2012 International Conference on Communication Systems and Network Technologies*, 2012, 184–188, DOI: 10.1109/CSNT.2012.48.

[21] A. Hatamlou, S. Abdullah and Z. Othman, "Gravitational search algorithm with heuristic search for clustering problems". In: *2011 3rd Conference on Data Mining and Optimization (DMO)*, 2011, 190–193, DOI: 10.1109/DMO.2011.5976526.

[22] S. Mirjalili, S. Z. Mohd Hashim and H. Moradian Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm", *Applied Mathematics and Computation*, vol. 218, no. 22, 2012, 11125–11137, DOI: 10.1016/j.amc.2012.04.069.

[23] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization". In: *2010 International Conference on Computer and Information Application*, 2010, 374–377, DOI: 10.1109/ICCIA.2010.6141614.

[24] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT Press, 2016.

[25] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, Springer International Publishing, 2018.

[26] "Greedy Layer-Wise Training of Deep Networks". Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf. Accessed on: 2020-06-23.

[27] E. de la Rosa Montero, "El aprendizaje profundo para la identificación de sistemas no lineales," Thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Mexico, 2014 (In Spanish), https://www.ctrl.cinvestav.mx/~yuw/pdf/MaTesER.pdf. Accessed on: 2020-06-23.

[28] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks*, 1998, 255–258.

[29] Y. Bengio and P. Lamblin, "Greedy layer-wise training of deep networks," Adv. neural, no. 1, pp. 153–160, 2007.

[30] K. Chellapilla, S. Puri and P. Simard, "High Performance Convolutional Neural Networks for Document Processing," In: Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule (France), Suvisoft, 2006.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on Machine Learning*, 2010, 807–814.

[32] M. Ranzato, F. J. Huang, Y.-L. Boureau and Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, 1-8, DOI: 10.1109/CVPR.2007.383157.

[33] J. Yang, K. Yu, Y. Gong and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 1794–1801, DOI: 10.1109/CVPR.2009.5206757.

[34] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, "End-to-end text recognition with convolutional

neural networks". In: *Proceedings of the 21st International Conference on Pattern Recognition*, 2012, 3304–3308.

[35] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, Berkeley, California: Apress, 2017, DOI: 10.1007/978-1-4842-2845-6.

[36] R. Venkatesan and B. Li, *Convolutional Neural Networks in Visual Computing: A Concise Guide*, CRC Press, 2017.

[37] L. Lu, Y. Zheng, G. Carneiro, and L. Yang, (eds.), *Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets*, Springer International Publishing, 2017.

[38] B. González, F. Valdez, P. Melin and G. Prado-Arechiga, "Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition", *Expert Systems with Applications*, vol. 42, no. 14, 2015, 5839–5847, DOI: 10.1016/j.eswa.2015.03.034.

[39] B. González, F. Valdez, P. Melin and G. Prado-Arechiga, "Fuzzy logic in the gravitational search algorithm enhanced using fuzzy logic with dynamic alpha parameter value adaptation for the optimization of modular neural networks in echocardiogram recognition", *Applied Soft Computing*, vol. 37, 2015, 245–254, DOI: 10.1016/j.asoc.2015.08.034.

[40] Y. Poma, P. Melin, C. I. González and G. E. Martínez, "Optimal Recognition Model Based on Convolutional Neural Networks and Fuzzy Gravitational Search Algorithm Method". In: O. Castillo and P. Melin (eds.), *Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine*, 2020, 71–81, DOI: 10.1007/978-3-030-34135-0_6.

[41] C. I. González, P. Melin, J. R. Castro, O. Mendoza and O. Castillo, "General Type-2 Fuzzy Edge Detection in the Preprocessing of a Face Recognition System". In: P. Melin, O. Castillo and J. Kacprzyk (eds.), *Nature-Inspired Design of Hybrid Intelligent Systems*, 2017, 3–18, DOI: 10.1007/978-3-319-47054-2_1.

[42] G. E. Martínez, O. Mendoza, J. R. Castro, A. Rodríguez-Díaz, P. Melin and O. Castillo, "Comparison between Choquet and Sugeno integrals as aggregation operators for pattern recognition". In: *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, 2016, 1–6, DOI: 10.1109/NAFIPS.2016.7851628.

[43] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez and O. Castillo, "Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms", *Engineering Letters*, vol. 13, no. 2, 2006.

[44] F. Gaxiola, P. Melin, F. Valdez, J. R. Castro and O. Castillo, "Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO", *Applied Soft Computing*, vol. 38, 2016, 860–871, DOI: 10.1016/j.asoc.2015.10.027.