Type-2 Fuzzy Logic in Pattern Recognition Applications

Patricia Melin

Abstract Type-2 fuzzy systems can be of great help in image analysis and pattern recognition applications. In particular, edge detection is a process usually applied to image sets before the training phase in recognition systems. This preprocessing step helps to extract the most important shapes in an image, ignoring the homogeneous regions and remarking the real objective to classify or recognize. Many traditional and fuzzy edge detectors can be used, but it is very difficult to demonstrate which one is better before the recognition results are obtained. In this work we show experimental results where several edge detectors were used to preprocess the same image sets. Each resulting image set was used as training data for a neural network recognition system, and the recognition rates were compared. In this paper we present the advantage of using a general type-2 fuzzy edge detector method in the preprocessing phase of a face recognition system. The Sobel and Prewitt edge detectors combined with GT2 FSs are considered in this work. In our approach, the main idea is to apply a general type-2 fuzzy edge detector on two image databases to reduce the size of the dataset to be processed in a face recognition system. The recognition rate is compared using different edge detectors including the fuzzy edge detectors (type-1, interval, and general type-2 FS) and the traditional Prewitt and Sobel operators.

Keywords Interval type-2 fuzzy systems • Image processing • Pattern recognition • Edge detection

P. Melin (✉)
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

R. John et al. (eds.), *Type-2 Fuzzy Logic and Systems*,
Studies in Fuzziness and Soft Computing 362,
https://doi.org/10.1007/978-3-319-72892-6_5

1 Introduction

Edge detection is one of the most common approaches to detect discontinuities in gray scale images. Edge detection can be considered an essential method used in the image processing area and can be applied in image segmentation, object recognition systems, feature extraction and target tracking [1].

There are several edge detection methods, which include the traditional ones, such as Sobel [2], Prewitt [3], Canny [1], Robert [4], Kirsch [5], and those based in type-1 [4, 6–8], interval type-2 [9–12] and general fuzzy systems [13, 14]. In Melin et al. [14] and Gonzalez et al. [13], some edge detectors based on GT2 FSs have been proposed. In these works the results achieved by the GT2 FS are compared with others based on a T1 FS and with an IT2 FS. According with the results obtained in these papers, the conclusion is that the edge detector based on GT2 FS is better than an IT2 FS and a T1 FS.

In other works, like in [15], an edge detector based on T1 FS and other IT2 FS are implemented in the preprocessing phase of a face recognition system. According with the recognition rates achieved in this paper the authors conclude that the recognition system has better performance when the IT2 fuzzy edge detector is applied.

In this paper we present a recognition approach illustrated with faces, which is performed with a monolithic neural network. In the methodology, two GT2 fuzzy edge detectors are applied over two face databases. In the first edge detector a GT2 FS is combined with the Prewitt operator and the second with the Sobel operator. The edge datasets achieved by these GT2 fuzzy edge detectors are using as the inputs of the neural network in a face recognition system.

The aim of this work is to show the advantage of using a GT2 fuzzy edge detector in pattern recognition applications. Additionally, make a comparative analysis with the recognition rates obtained by the GT2 against the results achieved in [15] by T1 and T2 fuzzy edge detectors.

The remainder of this paper is organized as follows. Section 2 gives a review of the background on GT2 FS. The basic concepts about Prewitt, Sobel operator, Low-pass filter and high-pass filter are described in Sect. 3. The methodology used to develop the GT2 fuzzy edge detector is explained in Sect. 4. The design of the recognition system based on monolithic neural network is presented in Sect. 5. The recognition rates achieved by the face recognition system and the comparative results are show in Sect. 6. Finally, Sect. 7 offers some conclusions about the results.

2 Overview of General Type-2 Fuzzy Sets

The GT2 FSs have attracted attention from the research community, and have been applied in different areas, like pattern recognition, control systems, image processing, robotics and decision making to name a few [16–21]. It has been demonstrated that a GT2 FS can have the ability to handle great uncertainties.

In the following we present a brief description about GT2 FS theory, which are used in the methodology proposed in this paper.

2.1 Definition of General Type-2 Fuzzy Sets

A General type-2 fuzzy set $\left(\tilde{\mathrm{A}}\right)$ consists of the primary variable x having domain X, the secondary variable u with domain in $\mathrm{J}_{\mathrm{x}}^{\mathrm{u}}$ at each $x \in X$. The secondary membership grade $\mu_{\tilde{\mathrm{A}}}(\mathrm{x},\mathrm{u})$ is a 3D membership function where $0 \leq \mu_{\tilde{\tilde{A}}}(\mathrm{x},\mathrm{u}) \leq 1$ [22–24]. It can be expressed by (1)

$$\tilde{A} = \left\{ \left((x,u), \mu_{\tilde{A}}(x,u) \right) | \forall x \in X, \forall u \in J_x^u \subseteq [0,1] \right\}. \tag{1}$$

The footprint of uncertainty (FOU) of $\left(\tilde{\mathrm{A}}\right)$ is the two-dimensional support of $\mu_{\tilde{\mathrm{A}}}(\mathrm{x},\mathrm{u})$ and can be expressed by (2)

$$FOU\left(\tilde{A}\right) = \left\{ (x,u) \in X \times [0,1] | \mu_{\tilde{A}}(x,u) > 0 \right\}. \tag{2}$$

2.2 General Type-2 Fuzzy Systems

The general structure of a GT2 FLS is shown in Fig. 1 and this consists of five main blocks, which are the input fuzzification, fuzzy inference engine, fuzzy rule base, type-reducer and defuzzifier [19].

In a GT2 FLS first the fuzzifier process maps a crisp input vector into other GT2 input FSs. In the inference engine the fuzzy rules are combined and provide a

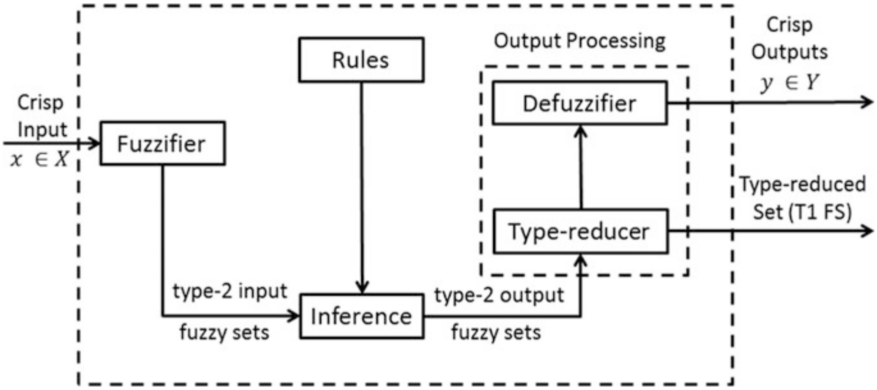

Fig. 1 General type-2 fuzzy logic system

mapping from GT2 FSs input to GT2 FSs output. This GT2 FSs output is reduced to a T1 FSs by the type-reduction process [25, 26].

There are different type-reduction methods, the most commonly used are the Centroid, Height and Center-of-sets type reduction. In this paper we applied Centroid type-reduction. The Centroid definition $C_{\tilde{A}}$ of a GT2 FLS [27–29] is expressed in (3)

$$C_{\tilde{A}} = \{(z_i, \mu(z_i)) | z_i \in \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i}, \\ \mu(z_i) \in f_{x_1}(\theta_1) \times \ldots \times f_{x_N}(\theta_N), \theta_i \in J_{x_1} \times \ldots \times J_{x_N}\} \tag{3}$$

where θ_i is a combination associated to the secondary degree $f_{x_1}(\theta_1)*\cdots*f_{x_N}(\theta_N)$.

2.3 General Type-2 Fuzzy Systems Approximations

Due to the fact that a GT2 FSs defuzzification process is computationally more complex than T1 and IT2 FSs; several approximation techniques have been proposed, some of them are the zSlices [21, 30] and the $\alpha -$plane representation [31, 32]. In these two approaches the 3D GT2 membership functions are decompose by using different cuts to achieve a collection of IT2 FSs.

In this paper the defuzzifier process is performed using $\alpha -$plane approximation, which is defined as follow.

An α-plane for a GT2 FS $\tilde{A}$, is denoted by $\tilde{A}_\alpha$, and it is the union of all primary membership functions of $\tilde{A}$, which secondary membership degrees are higher or equal than α ($0 \leq \alpha \geq 1$) [31, 32]. The $\alpha -$plane is expressed in (4)

$$\tilde{A}_\alpha = \{(x, u) | \mu_{\tilde{A}}(x, u) \geq \alpha, \forall x \in X, \forall u \in [0, 1]\} \tag{4}$$

3 Edge Detection and Filters

In this Section we introduce some concepts about filters and edge detectors (Prewitt and Sobel) used in image processing areas; since, these are critical for achieving good pattern recognition.

3.1 Prewitt Operator

The Prewitt operator is used for edge detection in digital images. This consists of a pair of 3 × 3 convolution kernels which are defined in (5) and (6) [33].

$$Prewittx = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{5}$$

$$Prewitty = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{6}$$

The kernels in (5) and (6) can be applied separately to the input image (I), to produce separate measurements of the gradient component (7), (8) in horizontally (gx) and vertically orientation (gy) respectively [33].

$$gx = Prewittx * I \tag{7}$$

$$gy = Prewitty * I \tag{8}$$

The gradient components (7) and (8) can be combined together to find the magnitude of the gradient at each point and the orientation of that gradient [2, 34]. The gradient magnitude (G) is given by. (9).

$$G = \sqrt{gx^2 + gy^2} \tag{9}$$

3.2 Sobel Operator

Sobel operator is similar to the Prewitt operator. The only difference is that the Sobel operator use the kernels expressed in (10) and (11) to detect the vertical and horizontal edges.

$$Sobelx = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{10}$$

$$Sobely = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{11}$$

3.3 Low-Pass Filter

Low-pass filters are used for image smoothing and noise reduction; this allows only passing the low frequencies of the image [15]. Also is employed to remove high spatial frequency noise from a digital image. This filter can be implemented by using (12) and the mask (highM) used to obtained the highPF is expressed in (13).

$$lowPF = lowM * I \tag{12}$$

$$lowM = \frac{1}{25} * \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

3.4 High-Pass Filter

High-pass filter only allow the high frequency of the image to pass through the filter and that all of the other frequency are blocked. This filter will highlight regions with intensity variations, such as an edge (will allow to pass the high frequencies) [15]. The high-pass (highPF) filter is implemented by using (14)

$$highPF = highM * I \tag{14}$$

where highM in (14) represents the mask used to obtained the highPF and this is defined by (15)

$$highM = \begin{bmatrix} -1/16 & -1/8 & -1/16 \\ -1/8 & 3/4 & -1/8 \\ -1/16 & -1/8 & -1/16 \end{bmatrix} \tag{15}$$

4 Edge Detection Improved with a General Type-2 Fuzzy System

In our approach two edge detectors are improved, in the first a GT2 FS is combined with Prewitt operator and the second with the Sobel operator. The general structure used to obtain the first GT2 fuzzy edge detector is shown in Fig. 2. The second fuzzy edge detector has a similar structure; we only change the kernel by using the Sobel operators in (10) and (11), which are described in Sect. 3.

The GT2 fuzzy edge detector is calculated as follows. To start, we select a input image (I) of the images database; after that, the horizontal *gx* (7) and vertical *gy* (8) image gradients are obtained; moreover, the low-pass (12) and high-pass (14) filters are also applied over (I).

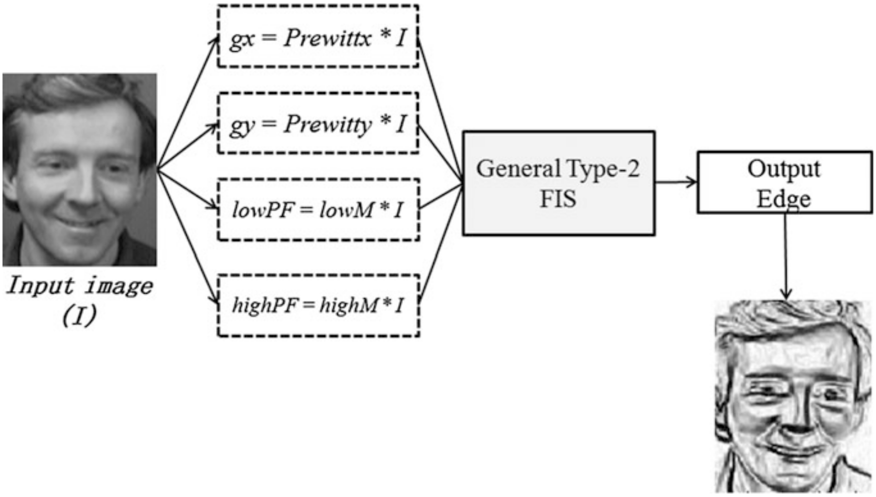

Fig. 2 Edge detector improved with Prewitt operator and GT2 FSs

The GT2 FIS was built using four inputs and one output. The inputs are the values obtained by *gx* (7), *gy* (8), *lowPF* (12) and *highPF* (14); otherwise, the output inferred represents the fuzzy gradient magnitude which is labeled as Output Edge.

An example of the input and output membership functions used in the GT2 FIS is shown in Figs. 3 and 4 respectively.

In order to objectively compare the performance of the proposed edge detectors against the results achieved in Mendoza [15], we use a similar knowledge base of fuzzy rules; these rules were designed as follows.

1. If (dx is LOW) and (dy is LOW) then (OutputEdge is HIGH)
2. If (dx is MIDDLE) and (dy is MIDDLE) then (OutputEdge is LOW)
3. If (dx is HIGH) and (dy is HIGH) then (OutputEdge is LOW)
4. If (dx is MIDDLE) and (highPF is LOW) then (OutputEdge is LOW)
5. If (dy is MIDDLE) and (highPF is LOW) then (OutputEdge is LOW)
6. If (lowPF is LOW) and (dy is MIDDLE) then (OutputEdge is HIGH)
7. If (lowPF is LOW) and (dx is MIDDLE) then (OutputEdge is HIGH)

5 Face Recognition System Using Monolithic Neural Network and a GT2 Fuzzy Edge Detector

The aim of this work is to apply a GT2 fuzzy edge detector in a preprocessing phase in a face recognition system. In our study case the recognition system is performed using a Monolithic Neural Networks. As already mentioned in Sect. 5, the edge

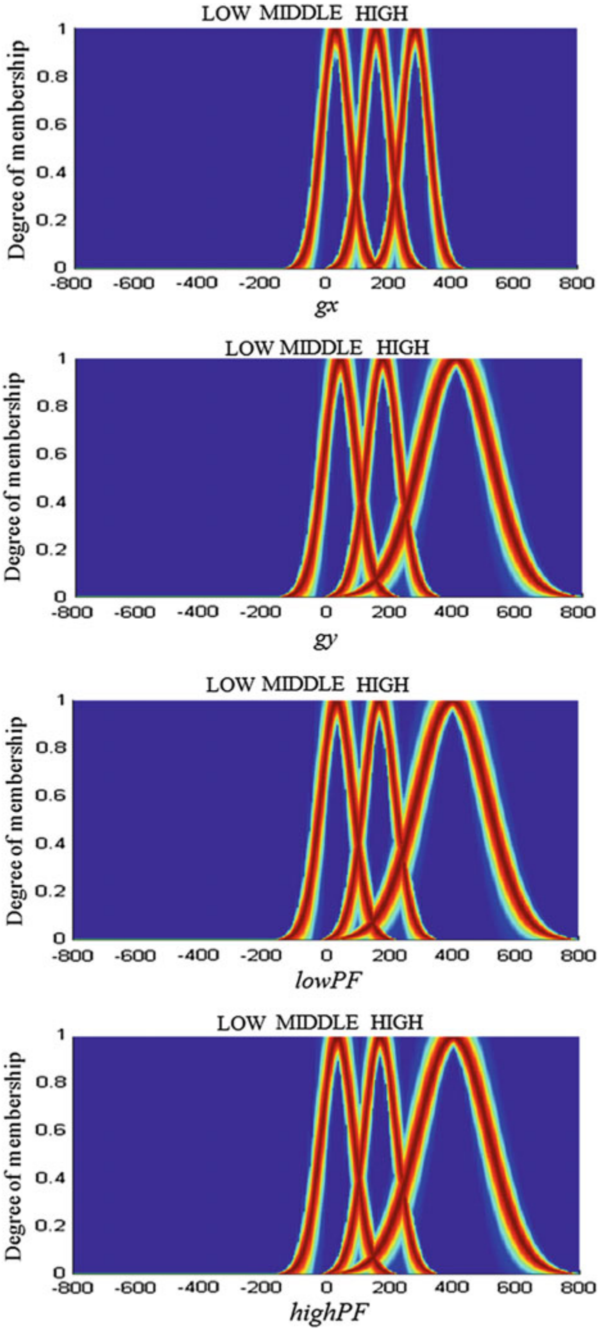

Fig. 3 Input membership functions using in the GT2 FIS

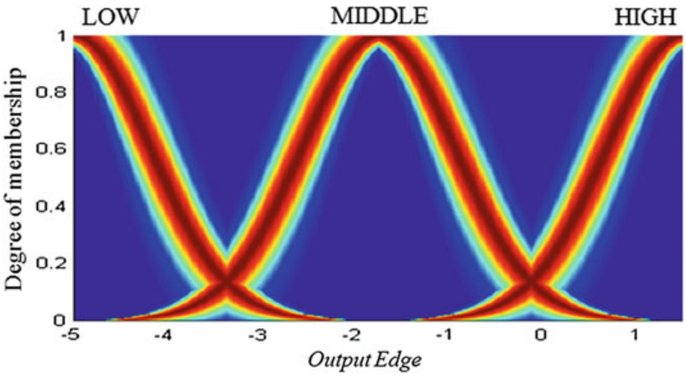

Fig. 4 Output membership function using in the GT2 FIS

detectors were designed using GT2 fuzzy combined with Prewitt and Sobel operator.

In Fig. 4 as an illustration, the general structure used in the proposed face recognition system is shown. The methodology used in the process is summarized in the following steps.

A. Select the input images database

In the simulation results two benchmark face databases were selected; in which are included the ORL [35] and the Cropped Yale [36–38].

B. Applied the edge detection in the input images

In this preprocessing phase, the two GT2 fuzzy edge detectors described in Sect. 5 were applied on the ORL and Cropped Yale database.

C. Training the monolithic neural network

The images obtained in the edge detection phase are used as the inputs of the neural network. In order to evaluate more objectively the recognition rate, the k-fold cross validation method was used. The training process is defined as follow.

1. Define the parameters for the monolithic neural network [15].
   - Layers hidden: two
   - Neuron number in each layer: 200
   - Learning algorithm: Gradient descent with momentum and adaptive learning.
   - Error goal: 1e-4.
2. The indices for training and test k folds were calculated as follow.
   - Define the people number (p).
   - Define the sample number for each person (s).

- Define the k-folds $(k=5)$.
- Calculate the number of samples (m) in each fold by using (16)

$$m=(s/k)\cdot p \tag{16}$$

- The train data set size (i) is calculated in (17)

$$i=m(k-1) \tag{17}$$

- Finally, the test data set size (18), are the samples number in only one fold.

$$t=m \tag{18}$$

- The train set and test set obtained for the three face database used in this work are show in Table 1.

3. The neural network was training k-1 times, one for each training fold calculated previously.
4. The neural network was testing k times, one for each fold test set calculated previously.

Table 1 Information for the tested database of faces

Database	People number (p)	Samples number (s)	Fold size (m)	Training set size (i)	Test size (t)
ORL	40	10	80	320	80
Cropped Yale	38	10	76	304	76

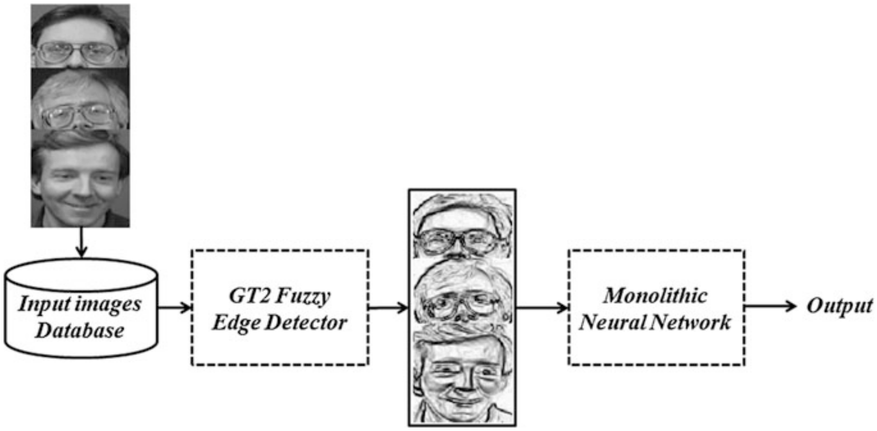

Fig. 5 General structure for the face recognition system

Finally, the mean of the rates of all the k-folds are calculated to obtain the recognition rate (Fig. 5).

6 Experimental Results

This section provides a comparison of the recognition rates achieved by the face recognition system when different fuzzy edge detectors were applied.

In the experimental results several edge detectors were analyzed in which are included the Sobel operator, Sobel combined with T1 FLS, IT2 FLS and GT2 FLS. Besides these, the Prewitt operator, Prewitt based on T1 FLS, IT2 FLS and GT2 FLS are also considered. Additional to this, the experiments were also validated without using any edge detector.

The tests were executed using the ORL and the Cropped Yale database; an example of these faces database is shown in Table 2. The parameters used in the monolithic neural network are described in Sect. 5. Otherwise, the training set and testing set that we considered in the tests are presented in Table 1, and these values depend on the database size used.

It is important to mention that all values presented below are the results of the average of 30 simulations achieved by the monolithic neural network. For this reason, the results presented in this section cannot be compared directly with the results achieved in [15]; because, in [15] only are presented the best solutions.

In the first test, the face recognition system was performed using the Prewitt and Sobel GT2 fuzzy edge detectors. This test was applied over the ORL data set. The

Table 2 Faces database

Database	Examples
ORL	
Cropped Yale	

Table 3 Recognition rate for ORL database using GT2 fuzzy edge detector

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate (%)
Sobel + GT2 FLS	87.97	0.0519	96.50
Prewitt + GT2 FLS	87.68	0.0470	96.25

mean rate, standard deviation and max rate values achieved by the system are shown in Table 3. In this Table we can note that better results were obtained when the Sobel GT2 fuzzy edge detector was applied; with a mean rate of 87.97, and standard deviation of 0.0519 and maximum rate of 96.50.

As a part of this test in Table 4, the results of 30 simulations are shown; these results were achieved by the system when the Prewitt GT2 fuzzy edge detector is applied.

Table 4 Recognition rate for ORL database using Prewitt GT2 fuzzy edge detector

Simulation number	Mean rate (%)	Standard deviation	Max rate (%)
1	83.50	0.0408	88.75
2	87.75	0.0357	92.50
3	89.75	0.0323	93.75
4	88.50	0.0427	95.00
5	70.25	0.3728	91.25
6	86.75	0.0420	91.25
7	89.00	0.0162	91.25
8	87.25	0.0205	88.75
9	86.75	0.0227	90.00
10	90.00	0.0441	**96.25**
11	89.50	0.0381	95.00
12	90.00	0.0265	92.50
13	89.75	0.0205	92.50
14	89.25	0.0360	92.50
15	92.00	0.0189	93.75
16	88.00	0.0447	93.75
17	86.25	0.0605	**96.25**
18	90.25	0.0323	95.00
19	90.50	0.0189	92.50
20	88.25	0.0227	91.25
21	89.00	0.0323	92.50
22	89.25	0.0167	91.25
23	89.00	0.0503	95.00
24	87.00	0.0447	91.25
25	89.25	0.0189	92.50
26	85.00	0.0776	93.75
27	87.00	0.0512	93.75
28	89.75	0.0399	95.00
29	86.00	0.0408	90.00
30	86.00	0.0503	93.75
Average	87.68	0.0470	92.75

In another test the system was considered using the Cropped Yale database. The numeric results for this experiment are presented in Table 5. In this Table we can notice that both edge detectors achieved the same max rate value; but, the mean rate was better with the Sobel + GT2 FLS.

As part of the goals of this work, the recognition rate values achieved by the system when the GT2 fuzzy edge detector is used, were compared with the results obtained when the neural network is training without edge detection, the Prewitt operator, the Prewitt combined with T1 and IT2 FSs; also, the Sobel operator, the Sobel edge detector combined with T1 and IT2 FSs. The results achieved after to apply these different edge detection methods are show in Tables 6 and 7.

Table 5 Recognition rate for Cropped Yale database using GT2 fuzzy edge detector

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
Sobel + GT2 FLS	93.16	0.0328	100
Prewitt + GT2 FLS	97.58	0.0328	100

Table 6 Recognition rate for ORL database

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
None	2.59	0.0022	5.00
Sobel operator	2.70	0.0037	5.00
Sobel + T1FLS	86.16	0.0486	93.75
Sobel + IT2FLS	87.35	0.0373	95.00
Sobel + GT2 FLS	**87.97**	**0.0519**	**96.50**
Prewitt operator	2.70	0.0036	5.00
Prewitt + T1FLS	87.03	0.0386	93.75
Prewitt + IT2FLS	87.54	0.0394	95.00
Prewitt + GT2 FLS	87.68	0.0470	96.25

Table 7 Recognition rate for Cropped Yale database

Fuzzy system edge detector	Mean rate (%)	Standard deviation	Max rate
None	2.83	0.0042	6.57
Sobel operator	2.63	0.0025	2.63
Sobel + T1FLS	97.52	0.0293	100
Sobel + IT2FLS	97.70	0.0314	100
Sobel + GT2 FLS	**98.11**	**0.0314**	**100**
Prewitt operator	2.80	0.0050	5.26
Prewitt + T1FLS	94.28	0.0348	100
Prewitt + IT2FLS	94.35	0.0304	100
Prewitt + GT2 FLS	97.58	0.0328	100

The results obtained for the ORL database are presented in Table 6; so, in this Table we can notice that the mean rate value is better when the Sobel GT2 fuzzy edge detector is applied with a value of 87.97. In these results we can also observe that the mean rate and max rate values obtained with the Prewitt + GT2 FLS were better than the Prewitt + IT2 FLS and Prewitt + T1 FLS.

Otherwise, the results achieved when the Cropped Yale database is used are shown in Table 7. In this Table we observed that the best performance (mean rate) of the neural network is obtained when the Sobel + GT2 FLS was applied; nevertheless, we can notice than the max rate values obtained by all the fuzzy edge detectors was of 100%.

7 Conclusions

In summary, in this paper we have presented two edge detector methods based on GT2 FS. The edge detection was applied in two image databases before the training phase of the monolithic neural network.

Based on the simulation results presented in Tables 6 and 7 we can conclude that the edge detection based on GT2 FS represent a good way to improve the performance in a face recognition system.

In general, the results achieved in the simulations were better when the fuzzy edge detection was applied; since the results were very low when the monolithic neural network was performed without edge detection; even so, when the traditional Prewitt and Sobel edge detectors were applied.

References

1. J. Canny, A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
2. I. Sobel, Camera models and perception. Ph.D. thesis, Stanford University, Stanford, CA, 1970
3. J.M.S. Prewitt, Object enhancement and extraction, ed. by B.S. Lipkin, A. Rosenfeld. in *Picture Analysis and Psychopictorics*, (Academic Press, NY, 1970), pp. 75–149
4. F. Perez-Ornelas, O. Mendoza, P. Melin, J.R. Castro, A. Rodriguez-Diaz, Fuzzy index to evaluate edge detection in digital images. PLoS ONE **10**(6), 1–19 (2015)
5. R. Kirsch, Computer determination of the constituent structure of biological images. Comput. Biomed. Res. **4**, 315–328 (1971)
6. L. Hu, H.D. Cheng, M. Zhang, A high performance edge detector based on fuzzy inference rules. Inf. Sci. **177**(21), 4768–4784 (2007)
7. Z. Talai, A. Talai, A fast edge detection using fuzzy rules, in *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, Mar 2011, pp. 1–5
8. C. Tao, W. Thompson, J. Taur, A fuzzy if-then approach to edge detection, in *Fuzzy Systems*, (1993), pp. 1356–1360
9. R. Biswas, J. Sil, An improved canny edge detection algorithm based on type-2 fuzzy sets. Procedia Technol. **4**, 820–824 (2012)

10. O. Mendoza, P. Melin, O. Castillo, An improved method for edge detection based on interval type-2 fuzzy logic. Expert Syst. Appl. **37**(12), 8527–8535 (2010)
11. O. Mendoza, P. Melin, G. Licea, A new method for edge detection in image processing using interval type-2 fuzzy logic, in *2007 IEEE International Conference on Granular Computing (GRC 2007)*, Nov 2007, pp. 151–151
12. O. Mendoza, P. Melin, G. Licea, Interval type-2 fuzzy logic for edges detection in digital images. Int. J. Intell. Syst. (IJIS) **24**(11), 1115–1133 (2009)
13. C.I. Gonzalez, P. Melin, J.R. Castro, O. Mendoza, O. Castillo, An improved sobel edge detection method based on generalized type-2 fuzzy logic. Soft. Comput. **20**(2), 773–784 (2014)
14. P. Melin, C.I. Gonzalez, J.R. Castro, O. Mendoza, O. Castillo, Edge-detection method for image processing based on generalized type-2 fuzzy logic. IEEE Trans. Fuzzy Syst. **22**(6), 1515–1525 (2014)
15. O. Mendoza, P. Melin, O. Castillo, Neural networks recognition rate as index to compare the performance of fuzzy edge detectors, in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, (2010), pp. 1–6
16. A. Doostparast Torshizi, M.H. Fazel Zarandi, Alpha-plane based automatic general type-2 fuzzy clustering based on simulated annealing meta-heuristic algorithm for analyzing gene expression data. Comput. Biol. Med. **64**, 347–359 (2015)
17. S.M.M. Golsefid, F. Zarandi, I.B. Turksen, Multi-central general type-2 fuzzy clustering approach for pattern recognitions. Inf. Sci. (Ny) **328**, 172–188 (2016)
18. G.E. Martínez, O. Mendoza, J.R. Castro, P. Melin, O. Castillo, Generalized type-2 fuzzy logic in response integration of modular neural networks, in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, (2013), pp. 1331–1336
19. J.M. Mendel, General type-2 fuzzy logic systems made simple: a tutorial. IEEE Trans. Fuzzy Syst. **22**(5), 1162–1182 (2014)
20. M.A. Sanchez, O. Castillo, J.R. Castro, Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. Expert Syst. Appl. **42**(14), 5904–5914 (2015)
21. C. Wagner, H. Hagras, Toward general type-2 fuzzy logic systems based on zSlices. IEEE Trans. Fuzzy Syst. **18**(4), 637–660 (2010)
22. J.M. Mendel, R.I.B. John, Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
23. D. Zhai, J.M. Mendel, Uncertainty measures for general type-2 fuzzy sets. Inf. Sci. **181**(3), 503–518 (2011)
24. D. Zhai, J. Mendel, Centroid of a general type-2 fuzzy set computed by means of the centroid-flow algorithm, in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, (2010), pp. 1–8
25. L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. **SMC-3**(1), 28–44 (1973)
26. L.A. Zadeh, *Fuzzy Sets*, vol. 8 (Academic Press Inc., USA, 1965)
27. F. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf. Sci. **178**(9), 2224–2236 (2008)
28. X. Liu, J.M. Mendel, D. Wu, Study on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements. Inf. Sci. **184**(1), 75–91 (2012)
29. J.M. Mendel, On KM algorithms for solving type-2 fuzzy set problems. IEEE Trans. Fuzzy Syst. **21**(3), 426–446 (2013)
30. C. Wagner, H. Hagras, Employing zSlices based general type-2 fuzzy sets to model multi level agreement, in *2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, (2011), pp. 50–57
31. J.M. Mendel, Comments on α-plane representation for type-2 fuzzy sets: theory and applications. IEEE Trans. Fuzzy Syst. **18**(1), 229–230 (2010)
32. J.M. Mendel, F. Liu, D. Zhai, α-Plane representation for type-2 fuzzy sets: theory and applications. IEEE Trans. Fuzzy Syst. **17**(5), 1189–1207 (2009)

33. R.C. Gonzalez, R.E. Woods, S.L. Eddins, *Digital Image Processing using Matlab*, (Prentice-Hall, 2004)
34. O. Mendoza, P. Melin, G. Licea, A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. Inf. Sci. **179**(13), 2078–2101 (2009)
35. The USC-SIPI image database, http://www.sipi.usc.edu/database/
36. A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
37. K.C. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
38. P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)