7 Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition

The combination of Soft Computing techniques allows the improvement of intelligent systems with different hybrid approaches. In this work we consider two parts of a Modular Neural Network for image recognition, where a Type-2 Fuzzy Inference System (FIS 2) makes a great difference. The first FIS 2 is used for feature extraction in training data, and the second one to find the ideal parameters for the integration method of the modular neural network. Once again Fuzzy Logic is shown to be a tool that can help improve the results of a neural system, when facilitating the representation of the human perception.

7.1 Method for Image Recognition

At the moment, many methods for image recognition are available. But most of them include a phase of feature extraction or another type of preprocessing closely related to the type of image to recognize (Melin and Castillo, 2005) (Starovoitov et al., 2002) (Chuang et al., 2000). The method proposed in this paper can be applied to any type of images, because the preprocessing phase does not need specific data about the type of image (Melin et al., 2007) (Mendoza and Melin, 2007).

Even if the method was not designed only for face recognition, we have made the tests with the ORL face database (AT&T Laboratories Cambridge) composed of 400 images of size 112x92. There are 40 persons, with 10 images of each person. The images are taken at different times, lighting and facial expressions. The faces are in upright position of frontal view, with slight left-right rotation. Figure 7.1 shows the 10 samples of one person in ORL database.

To explain the proposed steps of the method, we need to separate it them in two phases: the training phase in figure 7.3 and the recognition phase in figure 7.4.

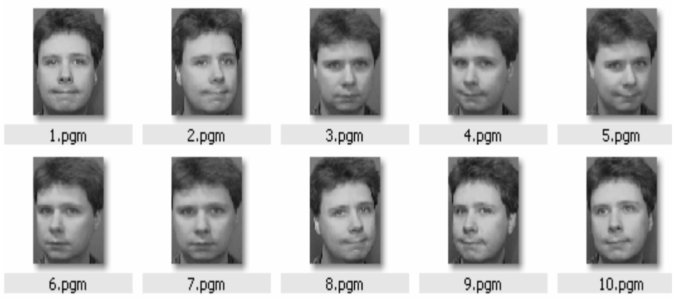

Fig. 7.1. Set of 10 samples of a person in ORL

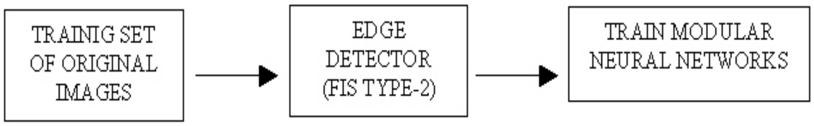
TRAINIG SET OF ORIGINAL IMAGES
EDGE DETECTOR (FIS TYPE-2)
TRAIN MODULAR NEURAL NETWORKS

Fig. 7.2. Steps in Training Phase

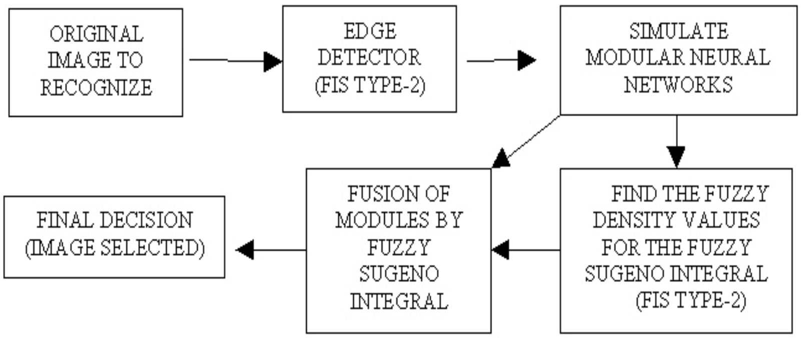
ORIGINAL IMAGE TO RECOGNIZE
EDGE DETECTOR (FIS TYPE-2)
SIMULATE MODULAR NEURAL NETWORKS
FIND THE FUZZY DENSITY VALUES FOR THE FUZZY SUGENO INTEGRAL (FIS TYPE-2)
FUSION OF MODULES BY FUZZY SUGENO INTEGRAL
FINAL DECISION (IMAGE SELECTED)

Fig. 7.3. Steps in Recognition Phase

7.2 Type-2 Fuzzy Inference System as Edge Detector

In previous work we presented an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks (Mendoza and Melin, 2006). In the proposed technique, it is necessary to apply Sobel operators to the original images, then use a Fuzzy Inference System Type-2 to generate the vector of edges that would serve like input data in a neural network. Type-2 Fuzzy Logic enables us to handle uncertainties in decision making and recognition in a more convenient way and for this reason was proposed (Castillo et al., 2007).

For the Type-2 Fuzzy Inference System, 3 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with (1), to which we will call DH and DV respectively.

The Sobel edges detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{7.1}$$

Where Sobely y Sobelx are the Sobel Operators throughout x-axis and y-axis.

If we define I as the source image, g_x and g_y are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as (2) and (3).

$$g_x = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \tag{7.2}$$

$$g_y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \tag{7.3}$$

Where *gx* and *gy* are the gradients along axis-x and axis-y, and * represents the convolution operator.

The other input is a filter that calculates when applying a mask by convolution to the original image. The low-pass filter hMF (4) allow us to detect image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise (Miosso and Bauchspiess, 2001).

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7.4}$$

Then the inputs for FIS type 2 are: DH=g_x, DV=g_y, M= *hMF*I,* where * is the convolution operator, and de output is a column vector contains the values of the image edges, and we can represent that in graphics shown in figure 7.4.

The Edges Image is smaller than the original because the result of convolution operation is a central matrix where the convolution has a value. Then in our example, each image with dimension 112x92 is reduced to 108x88.

The inference rules and membership function parameters allow to calculate a gray value between -4.5 and 1.5 for each pixel, where the most negative values corresponds

corresponds to the dark tone in the edges of the image. Then if we see the rules, only when the increment value of the inputs DH and DV are low the output is HIGH or clear (the background), in the rest of rules the output is LOW or dark (the edges). The complete set of fuzzy rules is given as follows (Castro et al., 2006):

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH) (1)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW) (1)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW) (1)
4. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW) (1)
5. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW) (1)

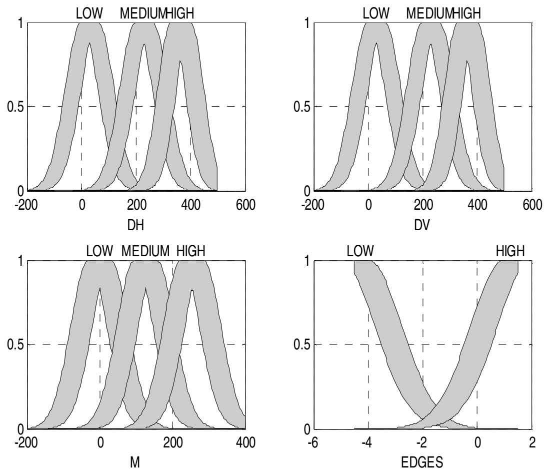

Fig. 7.4. Membership Function for the Type-2 FIS Edge Detector

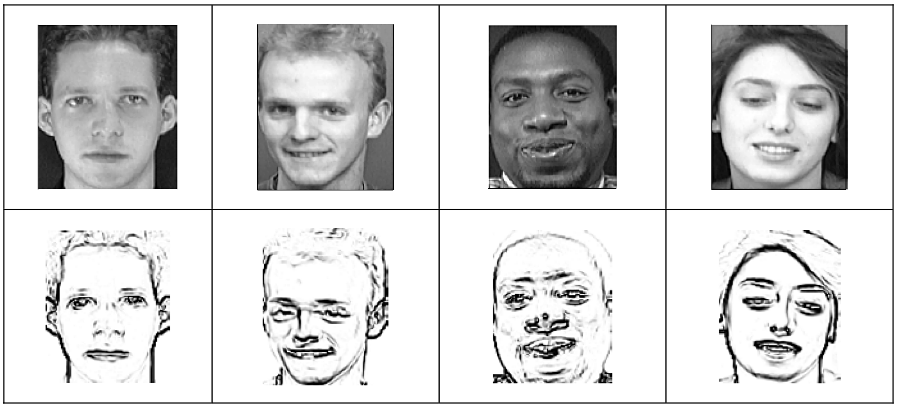

Fig. 7.5. Examples of edge detection with the Type-2 FIS method

The edge detector allows us to ignore the background color. We can see in this database of faces, different tones present for the same or another person. Then we eliminate a possible influence of a bad classification by the neural network, without losing detail in the image. Another advantage of edge detector is that the values can be normalized to a homogenous value range, independently the light, contrast or background tone in each image. At the examples in figure 7.5, all the edges in the images have a minimum value of -3.8 and a maximum value of 0.84. In particular for neural network training, we find these values to make the training faster: the mean of the values is near 0 and the standard deviation is near 1 for all the images.

7.3 The Modular Structure

The design of the Modular Neural Network consists of 3 monolithic feedforward neural networks (Sharkey, 1999), each one trained with a supervised method with the first 7 samples of the 40 images. Then the edges vector column is accumulated until the number of samples to form the input matrix for the neural networks as it is in the scheme of figure 7.7. Once the complete matrix of images is divided in 3 parts, each module is training with a correspondent part, with some rows of overlap.

The target to the supervised training method consist of one identity matrix for each sample, building one matrix with dimensions 40x(40*number_of_samples), as shown in figure 7.8.

Each Monolithic Neural Network has the same structure and is trained under the same conditions, like we can see in the next code segment:

```
layer1=200; layer2=200;
layer3=number_of_subjects;
net=newff(minmax(p),[layer1,layer2,layer3],{'tan
sig','tansig','logsig'},'traingdx');
net.trainParam.goal=1e-5;
net.trainParam.epochs=1000;
```

The average number of epochs to meet the goal in each module is of 240, and the required time of 160 seconds.

7.4 Simulation Results

A program was developed in Matlab that simulates each module with the 400 images of the ORL database, building a matrix with the results of the simulation of each module. These matrices are stored in the file “mod.mat” to be analyzed later for the combination of results. We can observe that in the columns corresponding to the training data, the position with a value near one is the image selected correctly. However in the columns that correspond to the test data this doesn’t always happens, reason why it is very important to have a good combination method to recognize more images.

According to exhaustive tests made in the simulation matrices, we know that recognition of the images that were used for the training of the neural networks is of the

100%. Therefore the interest is focused on the recognition of the samples that do not belong to the training set, is to say samples 8,9 and 10. The parameters for the Sugeno Fuzzy Integral that will be inferred will be the Fuzzy Densities, a value between 0 and 1 for each module, which determines the rate for each module. The parameter lambda, according to the theory of fuzzy measures depends on the values of the fuzzy densities, and is calculated by searching for the roots of a polynomial. After the simulation of an image in the Neural Network, the simulation value is the only known parameter to make a decision, then to determine the fuzzy density for each module is the unique available information. For this reason we analyze the values in many simulations matrix and decide that each input to the FIS Type-2 corresponds to the maximum value of each column corresponding to the simulation of each module of each one of the 400 images. The process to recognize each one of the images is shown in figure 6.

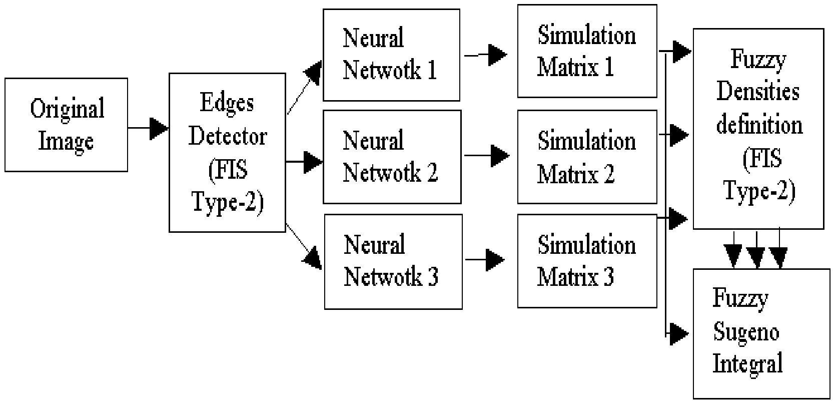

Fig. 7.6. Process of recognition using the type-2 fuzzy modular approach

Then each output corresponds to one fuzzy density, to be applied for each module to perform the fusion of results later with the Fuzzy Sugeno Integral. The inference rules found fuzzy densities near 1 when de maximum value in the simulation is between 0.5 and 1, and near 0 when the maximum value in the simulation is near 0. The fuzzy rules are shown below and membership functions in Figure 7.7.

1. If (max1 is LOW) then (d1 is LOW) (1)
2. If (max2 is LOW) then (d2 is LOW) (1)
3. If (max3 is LOW) then (d3 is LOW) (1)
4. If (max1 is MEDIUM) then (d1 is HIGH) (1)
5. If (max2 is MEDIUM) then (d2 is HIGH) (1)
6. If (max3 is MEDIUM) then (d3 is HIGH) (1)
7. If (max1 is HIGH) then (d1 is HIGH) (1)
8. If (max2 is HIGH) then (d2 is HIGH) (1)
9. If (max3 is HIGH) then (d3 is HIGH) (1)

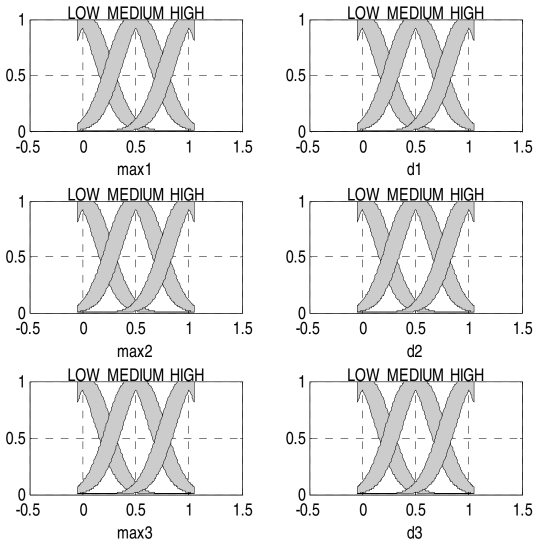

Fig. 7.7. Membership functions for the FIS to find fuzzy densities

Although the rules are very simple, allows to model the fuzziness to rate de modules when the simulation result don't reach the maximum value 1.

However some of the images don't reach the sufficient value in the simulation of the three modules, in these cases, do not exists enough information to select an image at the modules combination, and the image is wrongly selected.

In order to measure of objective form the final results, we developed a method of random permutation, which rearranges the samples of each person before the training. Once a permutation is made, the modular neural networks are trained and combined four times to obtain the sufficient information to validate the results. The average recognition rate is of 96.5%.

We show in Table 7.1 the summary of simulation results for each of the modules and the average and maximum results of the modular network (after fusion or combination of the results).

Table 7.1. Summary of the simulation results with the hybrid approach

Permu-	Image Recognition (%)					
tation	Train 1	Train 2	Train 3	Train 4	Average	Maximum
1	92.75	95	92.2	93.25	93.3	95
2	96.5	95.25	94.25	95.5	95.375	96.5
3	91.5	92	93.75	95.25	93.125	95.25
4	94.5	94.5	93.25	94	94.0625	94.5
5	93.75	93.5	94	96	94.3125	96
					94.035	**96.5**

7.5 Summary

We have shown in this chapter that the combination of Soft Computing techniques allows the improvement of intelligent systems with different hybrid approaches. In this chapter we considered two parts of a Modular Neural Network for image recognition, where a Type-2 Fuzzy Inference System (FIS 2) help us improves the performance results in image recognition. The first FIS 2 was used for feature extraction in training data, and the second one to find the ideal parameters for the integration method of the modular neural network.