Neuro Fuzzy Network and Wavelet Gabor For Face Detection

Dr. Raidah Salim
Computer Science Department, Science College, Basrah University

Abstract

This paper presents a face detection technique based on two techniques: ***wavelet Gabor filter*** for extract features from the localized facial image and ***neuro fuzzy system*** used as classifier depending on the features that extract , where it is used to determine the faces in the input image by draw boxes around the faces. The neurofuzzy network will be train on 128 image (69 face and 59 non face, size of each image 16*27 pixel in gray scale , this mean it trained to choose between two classes “face” and “non-face” images.

Our approach has been tested on eight common images with different face number in image and different number of fuzzy set. We got the best detection rate is 89.3% in case threshold equal 0.2 and in case number of fuzzy set equal 2. The stages of this work are implemented in MATLAB 7.0 environment.

Key Words- Face detection, Gabor wavelet, Neurofuzzy Network.

1. Introduction

In recent years, face recognition has attracted much attention and its research has rapidly expanded by not only engineers but also neuroscientists, since it has many potential applications in computer vision communication and automatic access control system. Especially, face detection is an important part of face recognition as the first step of automatic face recognition *which mean* segments the face areas from the background. However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating condition and facial expression.

Many novel methods have been proposed to resolve each variation listed above. For example, the *template-matching methods* are used for face localization and detection by computing the correlation of an input image to a standard face pattern. The *feature invariant approaches* are used for feature detection of eyes, mouth, ears, nose, etc. The *appearance-based methods* are used for face detection with *eigenface, neural network* and *information theoretical approach.* Nevertheless, implementing the methods altogether is still a great challenge[1].

Recently, several researchers introduced ***Neural Network*** (NN) and ***Fuzzy Logic*** (FL) systems for different tasks of image processing. Both NN and FL systems are aimed at exploiting human-like knowledge processing capability. Where, NN concentrate on the structure of human brain, i.e., on the ***hardware*** whereas FL system concentrate on ***software*** . A common way to apply a learning algorithm to fuzzy system is to represent it in a special NN like architecture, usually the ***NeuroFuzzy*** (NF) network has a number of layers, and each layer represents one or more steps of FL system[2][3]. Researchers proposed many different models of neurofuzzy for face detection, in the following recent works:

- J. S. Taur and C. W. Tao propose a neuro-fuzzy classifier (NEFCAR) that utilizes positive and negative rules with different rule importance to create the decision boundaries between different classes. The proposed classifier is applied to two applications. The first one is the Fisher iris data classification, and the second one is an on-line face detection and recognition application. Good classification results are obtained in both applications. In the on-line face detection and recognition system, two NEFCAR's are utilized: a two-class and a multi-class NEFCAR's are adopted to detect the face and recognize the face, respectively. The color of skin and the motion information are taken into consideration heuristically to improve the effectiveness of the face location algorithm[4].
- X. Zhu and D. Ramanan present a unified model for face detection, pose estimation, and landmark estimation in real-world, cluttered images. this model is based on a mixtures of trees with a shared pool of parts; in the model every facial

landmark as a part and use global mixtures to capture topological changes due to viewpoint. The researchers show that tree-structured models are surprisingly effective at capturing global elastic deformation, while being easy to optimize unlike dense graph structures. They present extensive results on standard face benchmarks, as well as a new "in the wild" annotated dataset, that suggests a system advances the state-of-the-art, sometimes considerably, for all three tasks. Though their model is modestly trained with hundreds of faces, it compares favorably to commercial systems trained with billions of examples (such as Google Picasa and face.com)[5]

- Olivier J. Bernier, Jean-Emmanuel Viallet, and Michel present approach, which obtains state of the art results, is based on a new neural network model: the Constrained Generative Model (CGM). Generative, since the goal of the learning process is to evaluate the probability that the model has generated the input data, and constrained since some counterexamples are used to increase the quality of the estimation performed by the model. To detect side view faces and to decrease the number of false alarms, a conditional mixture of networks is used[6].

In this paper we are interested by using neurofuzzy network in order to achieve image classification and wavelet Gabor filter for extract features from the localized facial image instead neural network. We used Matlab to programming our work.

The outline of this contribution is as follows: in the section 2 will be explain *Need for Neuro–Fuzzy Integration*why while the section 3 the wavelet Gabor filter will be discussed and a brief description of the NF network which is used in our work with the basic steps of its learning algorithm will be given in section 4. Section 5 will describe the strategy that is used to initialization of neuro fuzzy network. In the section 6 describe training and testing the proposed NF schemes and the results and test using the proposed NF schemes will be described also in this section Finally, section 7 contains some remarks and conclusions.

2. Neural network and fuzzy system

Both neural networks and fuzzy systems are dynamic, parallel processing systems that estimate input–output functions. They estimate a function without any mathematical model and *learn from experience* with sample data. A fuzzy system adaptively infers and modifies its fuzzy associations from representative numerical samples. Neural networks, on the other hand, can *blindly* generate and refine fuzzy rules from training data [7]. Hayashi and Buckley [8] proved that 1) any rule-based fuzzy system may be approximated by a neural net and 2) any neural net (feedforward, multilayered) may be approximated by a rule-based fuzzy system. Jang and Sun [9] have shown that fuzzy systems are functionally equivalent to a class of radial basis function (RBF) networks, based on the similarity between the local receptive fields of the network and the membership functions of the fuzzy system.

Fuzzy systems can be broadly categorized into two families. The first includes linguistic models based on collections of IF–THEN rules, whose antecedents and consequents utilize fuzzy values. It uses fuzzy reasoning and the system behavior can be described in *natural* terms. The *Mamdani* model [10] falls in this group. The second category, based on *Sugeno*-type systems [11], uses a rule structure that has fuzzy antecedent and *functional* consequent parts.

Neural networks, like fuzzy systems, are excellent at developing human-made systems that can perform information processing in a manner similar to our brain. In fact, the concept of ANN's was inspired by *biological neural networks* (BNN's), which are inherently nonlinear, highly parallel, robust and fault tolerant. A BNN is capable of 1) adapting its synaptic weights to changes in the surrounding environment; 2) easily handling imprecise, fuzzy, noisy, and probabilistic information; and 3) generalizing to unknown tasks. ANN's attempt to mimic these characteristics, often using principles from nervous systems to solve complex problems in an efficient manner. Fuzzy logic is capable of modeling vagueness, handling uncertainty, and supporting human-type reasoning.

A neural network is widely regarded as a black box that reveals little about its predictions. Extraction of rules from neural nets enables humans to understand this prediction process in a better manner. Rules are a form of knowledge that human experts can easily verify, transmit, and expand. Representing rules in *natural* form aids in enhancing their comprehensibility for humans. This aspect is suitably handled using fuzzy set-theoretic concepts.

The relation between neural networks and linguistic knowledge is bidirectional [12]. Therefore 1) neural network-based classification systems can be trained by numerical data and linguistic knowledge and 2) fuzzy rule-based classification systems can be designed by linguistic knowledge and fuzzy rules extracted from neural networks.

Fuzzy logic and neural systems have very contrasting application requirements. For example, fuzzy systems are appropriate if sufficient expert knowledge about the process is available, while neural systems are useful if sufficient process data are available or measurable. Both approaches build nonlinear systems based on bounded continuous variables, the difference being that neural systems are treated in a numeric quantitative manner, whereas fuzzy systems are treated in a symbolic qualitative manner. Fuzzy systems, however, exhibit both symbolic and numeric features. For example, when treated as collections of objects encapsulated by linguistic labels they lend themselves to symbolic processing via rule-based operations, while by referring to the definitions of the linguistic labels their membership functions are also suitable for numeric processing. Therefore, the integration of neural and fuzzy systems leads to a symbiotic relationship in which fuzzy systems provide a powerful framework for expert knowledge representation, while neural networks provide learning capabilities and exceptional suitability for computationally efficient hardware implementations. The significance of this integration becomes even more apparent by considering their disparities. Neural networks do not provide a strong scheme for knowledge representation, while fuzzy logic controllers do not possess capabilities for automated learning.

Neuro-fuzzy computing [13], which is a judicious integration of the merits of neural and fuzzy approaches, enables one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in data-rich environments into the system. The modeling of imprecise and qualitative knowledge as well as the transmission of uncertainty are possible through the use of fuzzy logic. Besides these generic advantages, the neuro–fuzzy approach also provides the corresponding application specific merits.

Neuro-fuzzy hybridization [13,14] is done broadly in two ways: a neural network equipped with the capability of handling fuzzy information [termed *fuzzy-neural network* (FNN)] and a fuzzy system augmented by neural networks to enhance some of its characteristics like flexibility, speed, and adaptability [termed *neural-fuzzy system* (NFS)].

In an FNN, either the input signals and/or connection weights and/or the outputs are fuzzy subsets or a set of membership values to fuzzy sets[15]. Usually, linguistic values such as *low*, *medium*, and *high*, or fuzzy numbers or intervals are used to model these. Neural networks with fuzzy neurons are also termed FNN as they are capable of processing fuzzy information.

A neural-fuzzy system (NFS), on the other hand, is designed to realize the process of fuzzy reasoning, where the connection weights of the network correspond to the parameters of fuzzy reasoning[16]. Using the backpropagation-type learning algorithms, the NFS can identify fuzzy rules.

3. Wavelet Gabor filter

Gabor filters are believed to function similarly to the visual neurons of the human visual system. From an information theoretic viewpoint, Okajima [17][18] derived Gabor functions as solutions for a certain mutual-information maximization problem.

Among various wavelet bases, Gabor functions provide the optimal resolution in both the time (spatial) and frequency domains, and the Gabor wavelet transform seems to be the optimal basis to extract local features for several reasons :

- Biological motivation: The simple cells of the visual cortex of mammalian brains are best modeled as a family of self-similar 2D Gabor wavelets[19].

- Mathematical and empirical motivation: Gabor wavelet transform has both the multi-resolution and multi-orientation properties and are optimal for measuring local spatial frequencies. Besides, it has been found to yield distortion tolerance space for pattern recognition tasks[19].

The Gabor receptive field can extract the maximum information from local image regions. For face recognition applications, by experiment we found that the number of Gabor filters 40 filters (5 scales and 8 orientations) is the best because it is get us best results in our, in the following the function Gabor wavelets to generate one filter (these steps are part of our programs):

```
function GW= GaborWavelet (R, C, Kmax,
                              f, u, v, Delt2);
k = ( Kmax / ( f ^ v ) ) * exp( i * u * pi/8 );

kn2 = ( abs( k ) ) ^ 2;
GW = zeros ( R , C );
for m = -R/2 + 1 : R/2
   for n = -C/2 + 1 : C/2
       GW(m+R/2,n+C/2) = ( kn2 / Delt2 ) *
        exp( -0.5 * kn2 * ( m ^ 2 + n ^ 2 ) /
       Delt2) * ( exp( i * ( real( k ) * m +
       imag (k) * n ) ) - exp ( -0.5 * Delt2 ) );
   end
end
```

4. Neuro Fuzzy Network

The architecture of the NF network introduced by Liman P. Maguire[20] based on Takaj fuzzy inference system consists of three layers; they represent an input layer, fuzzification and rule layer, and output layer respectively. Figure 3 shows the structure of NF network .

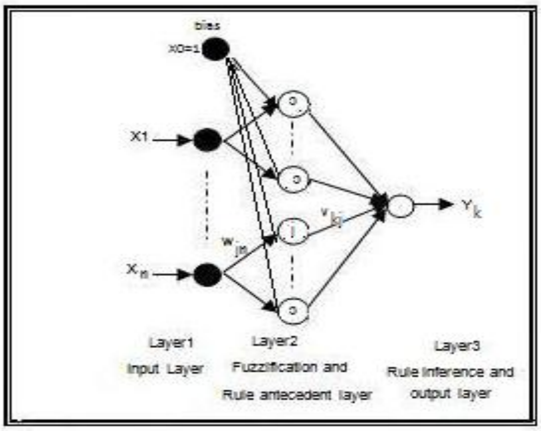

Figure 2: Structure of NeuroFuzzy Network

In order to derive a learning algorithm for a NF network with a gradient descent technique, the inference rule must use differentiable membership function type, for example in this work the Gaussian membership function will be used.

The adjusted parameters in the NF network can be divided into two categories based on ***if*** (antecedent) part and ***then*** (consequent) part of the fuzzy rules. For example in the antecedent part, the mean and variance are fine-tuned, whereas in the consequent part, the adjusted parameters are the consequence weights.

The gradient descent based on BP algorithm is employed to adjust the parameters in NF network by using training patterns. Moreover, the algorithm which is used for NF architecture is explained, both feed forward phase and the BP of errors as the follow:

Forward Phase

This phase computes the activation values of all the nodes in the network from the first to third layers.

1. Input layer**:** The nodes in this layer only transmit input values (crisp values) to the next layer directly without modification. Thus,

$$Net_i = x_i \qquad \forall i = 1..N \qquad \textbf{(1)}$$

Where, N^1 is a number of neurons in the input layer.

2. ***Fuzzification and rule antecedent layer*****:** The output function of this node is the degree that the input belongs to the given membership function. Hence, this layer acts as the fuzzifier. Each membership function is Gaussian and an input signal activates only M neighboring membership functions simultaneously. For a Gaussian-shaped membership function, the activation function for each node is:

$$f(netj) = \mu_i = e^{-netj^2} \qquad \textbf{(2)}$$

Where:

$$netj = w_{ji} x_i + w_{j0}$$

w_{ji} *Weights from neuron j in hidden layer to neuron I in input layer*

x_i *Input variable from neuron i in input layer to bias neuron*

w_{j0} *Weights from neuron j in*

hidden layer to bias neuron

$f(netj)$ *output of neuron j in hidden layer*

***3.Combination and Defuzzification layer*:** This layer performs defuzzification to produce a crisp output value. Among the commonly used defuzzification strategies, the COG method yielded the best result. In this layer, linear output activation function is used.

Each neuron in output layer determine the output value by calculate output weights average for hidden layer as follow :

$$y_k = \frac{netk}{\sum \mu_j} \qquad \textbf{(3)}$$

Where $netk = \sum_j \mu_j v_{kj}$

v_{kj} *weights from neuron k in output layer to neuron j in hidden layer*

Backward Phase

The goal of this phase is to minimize the error function:

$$E = \tfrac{1}{2}(y-d)^2 \qquad \textbf{(4)}$$

Where, d is the desired output.

The learning algorithm in NF is realized by adjusting connection weights of the neurons in output layer and hidden layer.

$$v_{kj}^{new} = v_{kj}^{old} + \Delta v_{kj} \qquad \textbf{(5)}$$

$$\Delta v_{kj} = -\eta \cdot \delta_k^{\ v} \qquad \textbf{(6)}$$

$$= -\eta \frac{\partial E}{\partial v_{kj}}$$

$$= -\eta \frac{\partial E}{\partial y_k} \cdot \frac{\partial f(netk)}{\partial netk} \cdot \frac{\partial netk}{\partial v_{kj}}$$

Then $\Delta v_{kj} = \eta(d_k - y_k) \cdot \dfrac{1}{\sum u_j} \mu_j$

While, adaptation of centers and widths of membership functions is as follows:

$$w_{ji}^{new} = w_{ji}^{old} + \Delta w_{ji} \qquad \textbf{(7)}$$

$$\Delta w_{ji} = -\eta \cdot \delta_{ji}^{w} \qquad \textbf{(8)}$$

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}$$

$$= -\eta \Sigma \left[\frac{\partial E}{\partial y_k} . \frac{\partial f(netk)}{\partial netk} \right] . \frac{\partial netk}{\partial \mu_j} \frac{\partial f(netj)}{\partial netj} \frac{\partial netj}{\partial w_{ji}}$$

$$= -\eta(\Sigma \delta_k v_{kj}) f'(netj) x_i$$

Then :

$$\Delta w_{ji} = -\eta(\Sigma \delta_k v_{kj})(-2 * netj * e^{-netj^2} * x_i)$$

5. Initialization of neuro fuzzy network

In any problem, each input variable ***(InputVariable)***, there are number of fuzzy set ***(fuzzySet_No)***, and the number of fuzzy rules or number of neuron***(Hidden_No)*** in hidden layer determine as follow:

$$Hidden\_No = fuzzySet\_NoforInputVariable1 + \dots + fuzzySet\_NoforInputVariablen..$$

To initialize network's weights we suggest the following strategy where this strategy give the best initial weights because we added some experience to the generate process which represent the training data itself :

- ***Steps to generate initial weights from n input layer to n in hidden layer which represent center and variance for membership function:***

$k = -0.9$

$step = 2/(fuzzySet\_No * InputVariable\_No$

for i = 1 : $fuzzySet\_No$

for j = 1 : $InputVariable\_No$

$W_{ji} = k$

$W_{j0} = 1$

$step = step + k$

end

End

- ***Steps to generate initial Weights from hidden layer to neuron in output layer which represent the parameters of consequent for fuzzy rule base.***

for k = 1 : $OutputVariable\_No$

for j = 1 : $Hidden\_No$

$w_{kj} = 0.9$

end

end

6. Training and Testing

In our work, the neurofuzzy network will be train on 128 image (69 face and 59 non face) as shown in figure 3 and figure 4 respectively, size of each image 16*27 pixel in gray scale. Then input layer consist of 2160 neuron and one neuron in output layer (0.0 for face and -0.9 for non face),

and hidden layer consist of neuro depend on number of fuzzy set for each input variable.

BP algorithm will be improved by using adaptive learning rate as follow:

$$\eta_t = \eta_0 (1 - \frac{t}{T}) \qquad \textbf{(9)}$$

Where η_0 :initial learning rate

T : maximum learning cycle

t : current learning cycle

We test our network with different learning and momentum rate then we find that the best values of them are 0.6 and 0.1 respectively.

The training is stopped after 1000 epochs and it is fail if the error is increased for more than five sequence epochs.

Neurofuzzy will be trained three times (each time with fuzzy set number different: FuzzySet no.=1, FuzzySet no.=2, FuzzySet no.=3) and the learning parameters are kept the same in all times. The final values of performance measure obtained from training are listed in table 1 and figure 5 shows the performance measure charts of neurofuzzy network.

Table 1: Final Performance Measure (MSE) of neurofuzzy network

Fuzzy set no.	MSE
1	0.003131
2	0.001373
3	0.002161

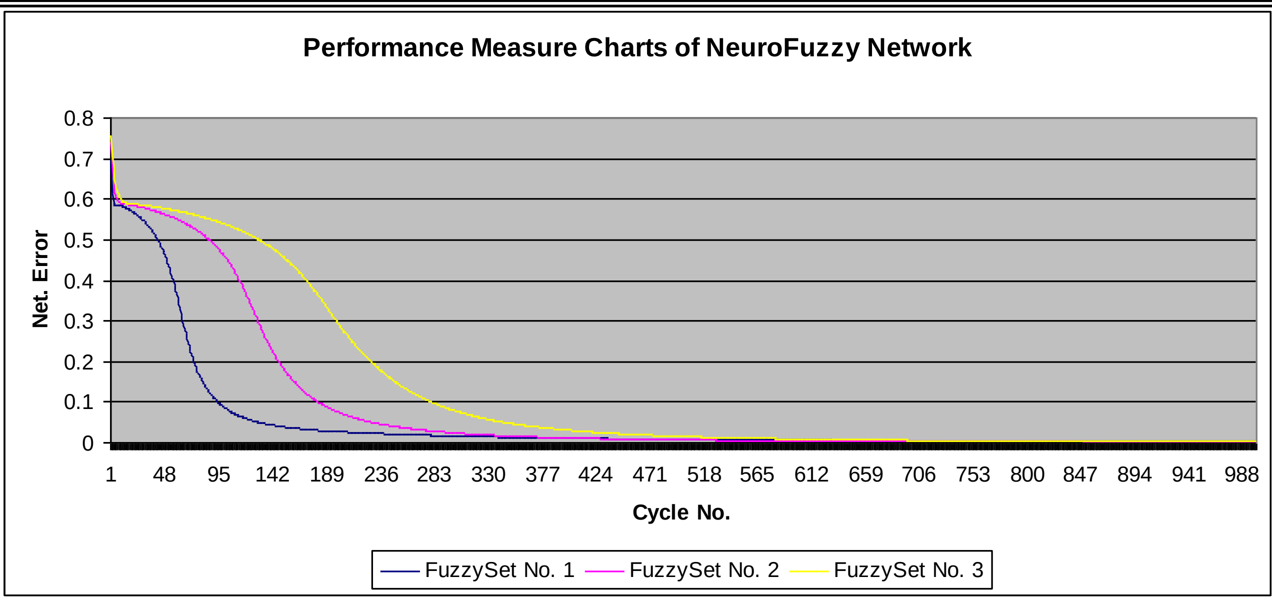

Figure 3: Performance Measure Charts of neurofuzzy network

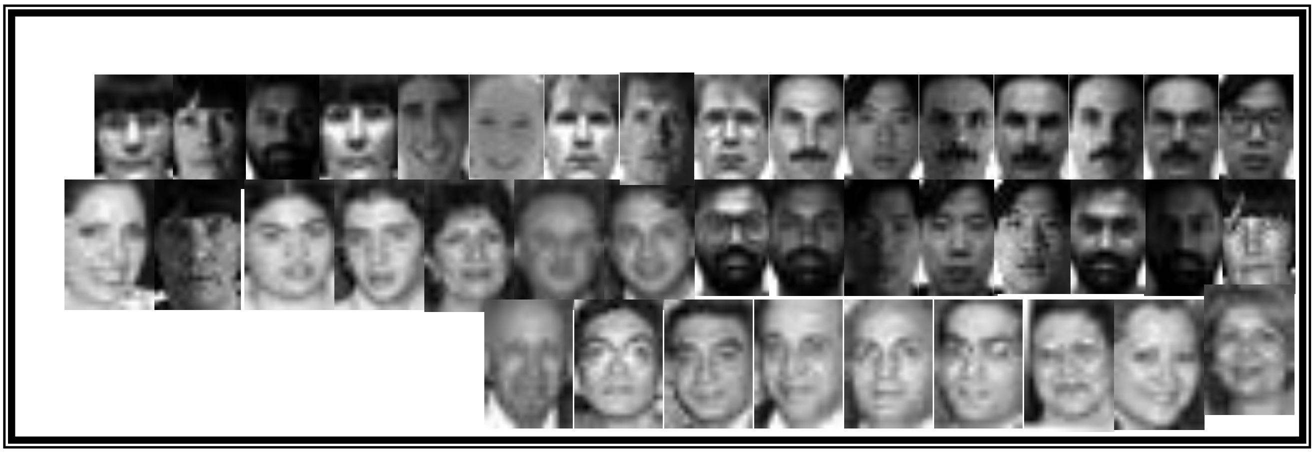

Figure 4: face images

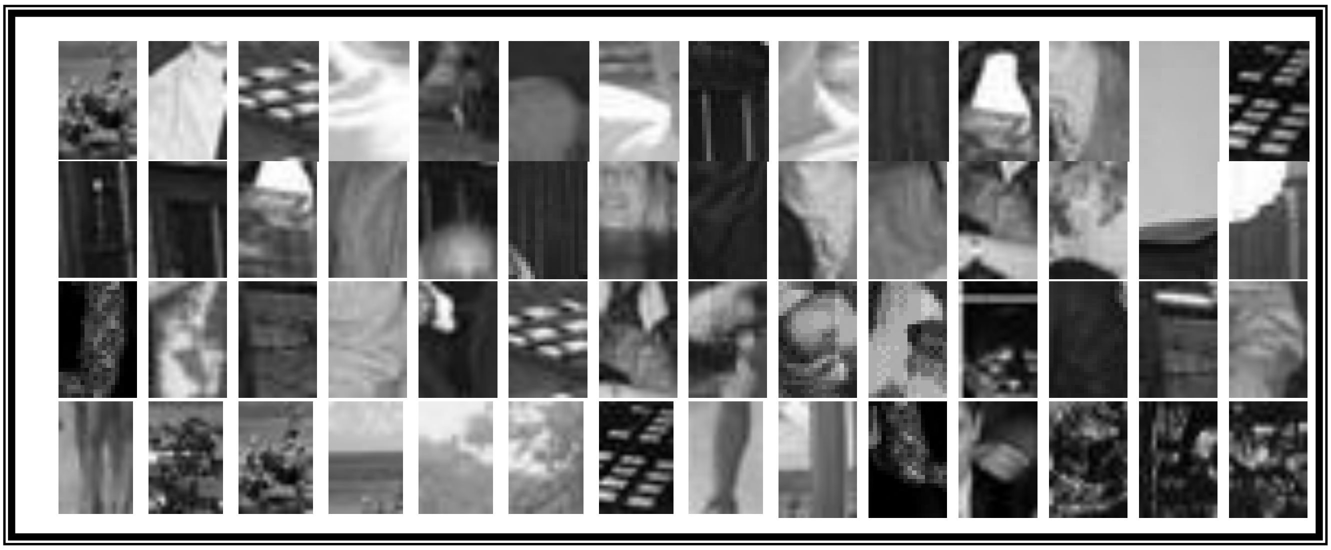

Figure 5: no face images

Table 2: Results of neuro-fuzzy network in case threshold=0.2

Image no.	No. of face in image	No. of detect face in case FSet1	No. of detect face in case FSet2	No. of detect face in case FSet3	Detection rate in case FSet1	Detection rate in case FSet2	Detection rate in case FSet3
1	7	7	7	6	**100%**	**100%**	85.6
2	15	12	15	15	80%	**100%**	**100%**
3	8	8	8	8	**100%**	**100%**	**100%**
4	56	52	56	55	92.9%	**100%**	98.2%
5	5	2	2	4	40%	40%	80%
6	1	0	0	0	0%	0%	0%
7	3	2	2	1	66.7%	66.7%	33.3%
8	3	3	2	2	**100%**	66.7%	66.7%
	103	86	92	91	83.5%	**89.3%**	88.3%

Table 3: Results of neuro-fuzzy network in case threshold=0.35

Image no.	No. of face in image	No. of detect face in case			Detection rate in case		
		FSet1	FSet2	FSet3	FSet1	FSet2	FSet3
1	7	7	7	6	**100%**	**100%**	85.7%
2	15	11	13	14	73.3%	86.7%	93.3%
3	8	8	8	8	**100%**	**100%**	**100%**
4	56	53	54	55	94.6%	96.4%	98.2%
5	5	2	3	3	40%	60%	60%
6	1	0	0	0	0%	0%	0%
7	3	0	1	2	0%	33.3%	66.7%
8	3	3	2	2	**100%**	66.7	66.7%
	103	84	88	90	81.6%	85.4%	**87.4%**

Table 4: Results of neuro-fuzzy network in case threshold=0.4

Image no.	No. of face in image	No. of detect face in case			Detection rate in case		
		FSet1	FSet2	FSet3	FSet1	FSet2	FSet3
1	7	7	6	6	**100%**	85.7%	85.7%
2	15	12	13	14	80%	86.7%	93.3%
3	8	8	8	8	**100%**	**100%**	**100%**
4	56 .	53	56	55	94.6%	**100%**	98.2%
5	5	2	3	3	40%	60%	60%
6	1	0	1	0	0%	**100%**	0%
7	3	1	1	2	33.3%	33.3%	66.7%
8	3	2	2	2	66.7%	66.7%	66.7%
	103	85	90	90	82.5%	**87.4%**	**87.4%**

Table 5: Results of neuro-fuzzy network in case threshold=0.5

Image no.	No. of face in image	No. of detect face in case			Detection rate in case		
		FSet1	FSet2	FSet3	FSet1	FSet2	FSet3
1	7	7	6	6	**100%**	85.6%	85.6%
2	15	12	13	13	80%	86.7%	86.7%
3	8	8	8	8	**100%**	**100%**	**100%**
4	56	54	56	55	96.4%	**100%**	98.2%
5	5	2	1	2	40%	20%	40%
6	1	0	1	0	0%	**100%**	0%
7	3	1	1	1	33.3%	33.3%	33.3%
8	3	2	2	2	66.7%	66.7%	66.7%
	103	86	88	87	83.5%	**85.4%**	84.5%

The tables (2-5) shown the result of 8 test images by using neurofuzzy network with different value of threshold (0.2, 0.35, 0.4, 0.5) and different number of fuzzy set(1, 2, 3).

7. Conclusions

Since both of neural network and fuzzy logic have their limitations and advantages, therefore the neural network and fuzzy logic are combined together to pass these limitations and use their advantages and the neurofuzzy architecture has been proposed and applied for face detection. The neurofuzzy architecture is based on Takage fuzzy logic system and neural network algorithms. In common Takage NF architecture, full connections are used to connect the neurons of fuzzification layer (membership functions). In our problem this architecture of connections will cause to have a huge number of connections therefore a new architecture is presented where only the neurons that facing each other will be connected and additional one that connects all the neurons of fuzzification layer together to cover all the possibilities.

In this contribution, NF scheme have been presented for face detection, from tables 2-5 shown that:

- Number of detect face and average detection rate in different values of threshold and number of fuzzy set threshold.
- The best detection rate is 89.3% in case threshold equal 0.2 and in case number of fuzzy set equal 2.
- We notice that if threshold is too small (like equal to 0) or too big (like >0.5) even the faces detection in images the false detection increase.
- Value of threshold has own great effect on the detect operation , where to detect the faces in any image there are appropriate number of fuzzy set and appropriate threshold determine by trail and error therefore we can use genetic algorithm to determine the best number of fuzzy set and threshold.
- Table 2 shown that the best result at fuzzy set = 2 where average of detection rate is 89.3%.
- Table 3 shown that the best result at fuzzy set = 3 where average of detection rate is 87.4%
- Table 4 shown that the best result at fuzzy set = 2 and fuzzy set = 3 where average of detection rate is 87.4%
- Table 5 shown that the best result at fuzzy set = 2 where average of detection rate is 85.4%

References

1. B. Moghaddam and A. Pentland, **"*Probabilistic visual learning for object recognition*,"** IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no.7. pp. 696-710, July, 1997.

2. S. Horikawa, T. Furuhashi, and Y. Uchikawa, ***"On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm"***, IEEE Trans. Neural Networks, vol. 3, pp. 801-806, 1992.

3. L.-X. Wang, "***Adaptive Fuzzy systems and control Design and stability analysi"s***, Prentice Hall, New Jersey, 1994.

4. J. S. Taur and C. W. Tao , **" *A New Neuro-Fuzzy Classifier with Application to On-Line Face Detection and Recognition*",** Journal of VLSI Signal Processing Systems , Volume 26 Issue 3, Nov. 2000 , Pages 397 - 409.

5. X. Zhu, D. Ramanan. "***Face detection, pose estimation and landmark localization in the wild***", *Computer Vision and Pattern Recognition* (CVPR) Providence, Rhode Island, June 2012.

6. Olivier J. Bernier, Jean-Emmanuel Viallet, and Michel Collobert**, "*A Fast and Accurate Face Detector Based on Neural Networks* ",** IEEE transaction pn pattern analysis and machine intelligence, Vol. 23, No. 1, 2001

7. B. Kosko, "***Neural Networks and Fuzzy Systems***", Englewood Cliffs, NJ: Prentice-Hall, 1991.

8. Y. Hayashi and J. J. Buckley, ***"Approximations between fuzzy expert systems and neural networks***", *Int. J. Approx. Reas.*, vol. 10, pp. 63–73, 1994.

9. J. S. R. Jang and C. T. Sun, ***"Functional equivalence between radial basis function networks and fuzzy inference systems"***, *IEEE Trans. Neural Networks*, vol. 4, pp. 156–159, 1993.

10. E. H. Mamdani and S. Assilian, "**An experiment in linguistic synthesis with a fuzzy**

logic controller". *Int. J. Man-Mach. Stud.*, vol. 7, pp. 1–13, 1975.
11. T. Takagi and M. Sugeno, ***"Fuzzy identification of systems and its application to modeling and contro"***, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, 1985.
12. H. Ishibuchi, M. Nii, and I. B. Turksen, ***"Bidirectional bridge between neural networks and linguistic knowledge: Linguistic rule extraction and learning from linguistic rules***", in *Proc. IEEE Int. Conf. Fuzzy Syst. FUZZ-IEEE'98*, Anchorage, AK, May 1998, pp. 1112–1117.
13. S. K. Pal and S. Mitra, ***"Neuro-fuzzy Pattern Recognition: Methods in Soft Computing"***, New York: Wiley, 1999.
14. Sushmita Mitra , ***"Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework"***, IEEE transactions on neural networks, Vol. 11 No. 3,
15. S. Mitra and S. K. Pal, ***"Fuzzy multilayer perceptron, inferencing and rule generation"***, *IEEE Trans. Neural Networks*, vol. 6, pp. 51–63, Jan. 1995.
16. D. Nauck, F. Klawonn, and R. Kruse, ***"Foundations of Neuro–Fuzzy Systems"***. Chichester, U.K.: Wiley, 1997.

17. Avinash Kaushal, J P S Raina, "***Face detection using neural network & Gabor wavelet transform***", IJCST vol. 1 issue 1, sep. 2010.

18. L. Shen and L. Bai, ***"A review of Gabor wavelets for face recognition"***, Patt. Anal. Appl. 9: 273-292, 2006.
19. Ming-Husan Yang, David J.Kriegman, and Narendra Ahuja, "***Detecting Faces in Images: A Survey "***, IEEE transaction on pattern analysis and machine intelligence, vol.24 no.1, January 2002.

20. Da Ruan, ***"Intelligent Hybrid System: Fuzzy Logic, Neurl Networks and Genetic Algorithms"***, Academic publishers, 1997.

المستخلص

قدم هذا البحث لتقنية كشف الوجه استنادا الى اثنين من التقنيات: مرشح جابور المويجي لاستخراج ميزات صورة ونظام عصبي مضبب يستخدم كمصنف اعتمادا على الميزات التي استخلصت، حيث يتم استخدامه لتحديد الوجوه في صورة مدخلة برسم مربعات حول الوجوه . دربت الشبكة العصبية المضببة على 128 صورة (69 و 59 وجه وجه غير، حجم كل صورة 16 * 27 بكسل في مقياس الرمادية، وهذا يعني أنها دربت مدربة على الاختيار بين فئتين صورة "وجه" و "غير وجه" .

وقد تم اختبار طريقتنا على ثماني صور مشتركة بعدد مختلف من الوجوه وعدد مختلف من المجاميع المضببة. وقد حصلنا على أفضل معدل اكتشاف 89.3٪ في حالة عدد المجاميع المضببة المساوية لـ 2. تم تنفيذ مراحل هذا العمل في بيئة MATLAB 7.0.