Impulsive and Poisson Noises Removal Using Takagi Neuro-Fuzzy Network

Sarah B. Aziz and Maytham A. Shahed

Computer Science Department, Science College,
Basrah University, Basrah, Iraq

Abstract :

A new Takagi neuro-fuzzy filter is presented for the noise reduction of images corrupted with impulsive and Poisson noises. In the current paper, the number of neuro-fuzzy connections is reduced to be equal to the number of membership functions. Also, the time of computation is reduced by using artificial image for training the presented neuro-fuzzy filter. The filter can be applied effectively to reduce heavy noise. Experimental results are obtained to show the feasibility of the proposed approach. These results are good when compared to other filters by numerical measures and visual inspection. The presented scheme is applied to grayscale and truecolor images. The presented scheme is efficient, fast, and can be extended by adding other filter as input, without smoothing an image.

Keywords: Impulsive noise; Poisson noise; Median filter; Average filter; Takagi Neuro-Fuzzy.

1. Introduction

Some images contain spurious cell values (much brighter or darker than their surroundings) that represent “noise” imposed by the imaging system or by later processing. Additive noise, probably the most common type and Poisson noise is an additive white noise its intensity values are drawn from Poisson distribution. Another common type of noise is the impulsive noise which is a random occurrence of white intensity values.

A wide variety of filtering algorithms have been developed to detect and remove noise, leaving as much as possible of the pure image. These include both temporal filters and spatial filters. For example, the average filter is very effective in filtering poisson noise, while the Median Filter works very well against impulse noise [1, 2].

In spite of the filters’ abilities in reducing noises from images, most of them have a significant disadvantage that they left the result image blurred.

Recently, the different techniques of soft computing; neural networks, fuzzy logic, and their combination have already been applied in several domains of image processing (e.g., filtering, interpolation, and morphology), and have numerous practical applications. These fuzzy and neuro-fuzzy (NF) filters, including VM-filter [3], the DD-filter [4], the weighted fuzzy mean filter [5], Fuzzy two-step filter [6], nonlocal fuzzy diffusion [7], RNF-filter [8], and IVM-filter [9].

In this paper, we will focus on neuro-fuzzy technique for image filtering. Most techniques in image noise reduction mainly deal with fat-tailed noise like impulsive noise. These techniques are able to outperform rank-order filter schemes (such as the median filter). Nevertheless, most of these techniques are not specifically designed for more than one type of noise or do not produce convincing results when applied to handle another types of noise (such as Poisson noise), which eliminated using average filter [1, 10]. The median and average filters will be used later for comparison.

This paper deals with noise reduction and presents an effective technique based on Takagi neuro-fuzzy network scheme to improve the effects of smoothing filters in removing different ratios of impulsive noise (10%, 15%, 20%, 25%, and 30% of corrupted pixels) and Poisson noise from grayscale and truecolor images. The suggested neuro-fuzzy architecture will be implemented, the time of training will be reduced by using an artificial image as training data set instead of real image.

The rest of this paper is organized as follows: First, Section 2 will describe smoothing filters and a brief explain of the NF network with the basic steps of its learning algorithm will be given in Section 3. Section 4 will describe the data sets used in training and testing the proposed NF scheme. Section 5 explains the proposed NF scheme and in Section 6 the experiments performed using the proposed NF scheme and compared their results with those obtained by the corresponding conventional median and average filters through computer simulations. Finally, Section 7 concludes this paper.

2. Smoothing filters

Smoothing filters are used for noise reduction. The proposed filter scheme in this work is to improve the effects of two kinds of smoothing filters. They are:

2.1 Median filter

The median filter (MF) is nonlinear filter; it is ranks the input values from the current filter window in numerical order and assigns the middle value to the output cell. Because the median value is not affected by the actual value of the noise cells, the median filter is particularly good at removing isolated random noise. It also preserves edges and line features better than the average filter, but does produce some blurring. Applications of the median filter require caution because median filtering tends to remove image details such as thin lines and corners while reducing noise [10,11].

2.2 Average filter

The average filter (AF) is linear filter. The output cell value calculated by this filter is the simple average (arithmetic mean) of the cells in the filter window. The averaging performed by the low pass filter removes some of the higher frequency features, while allowing the low-frequency features to "pass" through the filter unchanged (thus the term "low pass" filter). This has the effect of smoothing the raster image, emphasizing its larger-scale brightness trends [10,11].

For truecolor image, the smoothing filter will be applied on matrix of each color: red, green, and blue (RGB) separately then combined the three resulted grayscale images to get the filtered truecolor image [11]. The principle is shown in figure 1.

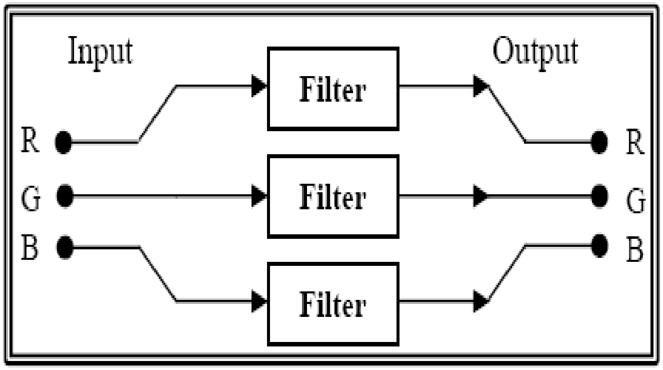


Figure 1: RGB Component Filtering

3. Neuro-Fuzzy network

The architecture of the NF network based on Takagi fuzzy inference system consist of five layers; they represent an input layer, fuzzification layer, rule antecedent layer, rule consequent layer, and combination and defuzzification layer respectively. Figure 2 shows the structure of Takagi NF network [11, 12, 13, 14].

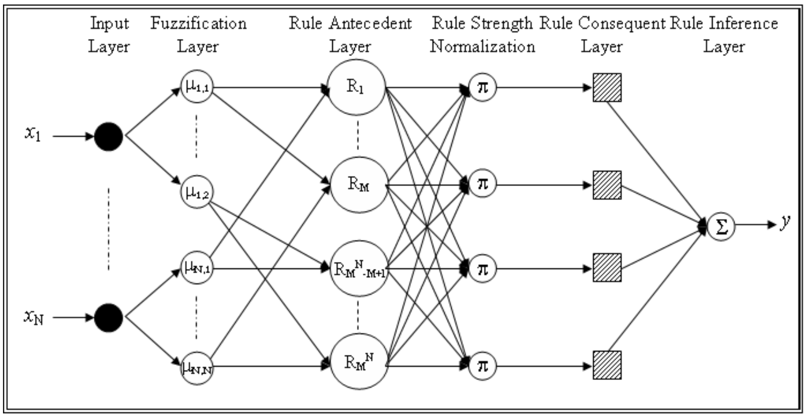


Figure 2: Structure of Takagi Neuro-Fuzzy Network

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 backpropagation (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.

❖ **Forward Phase**

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

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

3. ***Rule Antecedent layer*****:** The implication method is performed by this layer and applied using the products.
4. ***Rule Strength Normalization:*** Every node in this layer calculates the ratio of the k^{th} rule's firing strength to the sum of all rules firing strength.
5. ***Rule Consequent layer*****:** Every node in this layer will be multiplied with a node function.
6. ***Rule Inference layer*****:** The single node of this layer computes the overall output as the summation of all incoming signals.

❖ **Backward phase**

The goal of this phase is to minimize the error function. The learning algorithm in NF is realized by adjusting the parameters set of node function beside the centers and widths of membership functions.

4. Data sets

Two types of data sets will be used; the first data set for training phase and the other for testing phase.

The training data set is an artificial image 36×36 generated randomly or formed from the representative building blocks of a natural image which are including the flat regions, varying from bright regions to dark regions and edge areas that may be sharp and blurred (Figure 3).

Although, this kind of training data is not commonly used where it is somehow difficult to find the suitable artificial image for the specific application, the training time becomes short since the size of the artificial image is small.

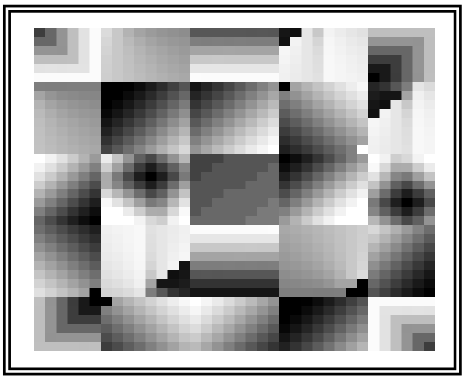


Figure 3: The original artificial image

On the other side, the testing data set includes ten images. Five of them are grayscale images and the other five are truecolor images.

5. Proposed NF scheme

The proposed architecture is based on Takagi NF which has five layers as explained in Section 3. The proposed scheme of NF has three inputs that represent the noisy pixel and its corresponding outputs from applied AF and MF, respectively. The scheme has one output and the number of membership functions is generated randomly at each run in the range between 9 and 11. This range of membership function is selected after several experiments, where it gave good training results.

In common Takagi NF architecture, full connections are used to connect the neurons of fuzzification layer (membership functions). In another words, the number of connections is equal to number of membership functions powered to the number of inputs. i.e. number of connections is 729 when the number of membership functions is 9 and 1331 when it is 11. As noticed, the number of connections is huge and that will cause non convergence in the training process. To solve this problem, only the neurons that facing each other were connected i.e. the number of connections is equal to the number of membership functions. The schematic diagram of the suggested NF architecture is shown in figure 4.

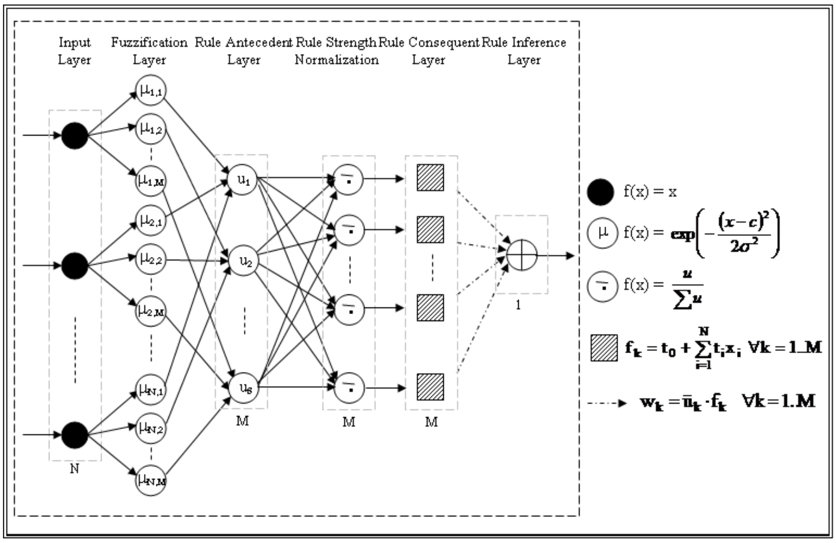


Figure 4: Schematic diagram of the suggested NF architecture

5.1 Data preparing

To construct the patterns of training, the artificial image in figure 3 will be corrupted by impulsive noise of ratio 20% for upper part and Poisson noise for lower part and applied the AF and MF on it one at each time (Figure 5).

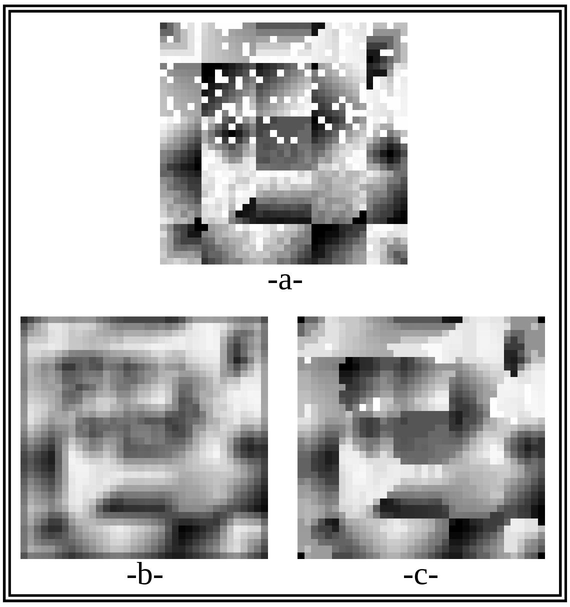


Figure 5: The artificial image

a. Image corrupted with 20% impulsive noise for upper part and Poisson noise for lower part
b. Result image from applied AF
c. Result image from applied MF

The above corrupted, AF result, MF result, and original 8-bit 256 gray value images are converted to a floating point images, with gray values in the range [0,1]. Then, each pattern that contains three inputs and one output will be formed from a pixel in the corrupted image and its corresponding pixels from AF result, MF result, and original images. This is done by scanning the four images row by row.

By the same way, the patterns of testing are constructed from testing images, where each pattern that contains three input values only will be formed. Then, the output of NF scheme of any test image which forms a sequence of floating point values in range [0, 1] should be converted to 8-bit 256 gray values and rearranged as matrix with size of the test image itself.

5.2 Training

There are several parameters to be adjusted by the gradient descent based on BP algorithms. These parameters are mean and variance of Gaussian memberships and the consequence weights. The BP algorithm that its forward and backward phases illustrated in section 3 will be used in training the NF network taking into consideration the following points:

1. Only the facing neurons in the fuzzification layer will be connected together instead of the full connections. Therefore, the output of the nodes in the Rule Consequent Layer (3rd layer) will be calculated as follows:

$$u_j = \prod_{i=1}^{N} \mu_{(i,j)} \qquad \forall\ j = 1..M \qquad \textbf{. . . (1)}$$

Where, N is number of inputs (N = 3) and M is number of membership functions.

2. BP algorithm will be improved by using adaptive *learning rate* (*lr*) [15]. Therefore, the parameters are updated under the following conditions.

$$\begin{aligned}
&if\ \ (new\,error\,/\,old\,error) > 1.04\ \ then\\
&\qquad New\ values\ of\ \ parameters\ are\ discarde\\
&\qquad lr = lr * lr_dec\\
&else \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{. . . (2)}\\
&\qquad Update\ Parameters\\
&\qquad if\ \ \ new\,error < \ old\ error\\
&\qquad\qquad lr = lr * lr_inc
\end{aligned}$$

The initial values of each of the learning rate, set of order parameters, and widths are generated randomly in range between 0 and 1; where, the width of each membership functions of all the inputs are the same while the values of ***lr_inc***, and ***lr_dec*** were set to 1.3, and 0.1, such it will be find experimentally that these values give and lead to smoothing and speeding convergence. The initial values of centers of membership functions of each input are set as follows:

$$c_{(i,j)} = (j-1)/(M-1) \qquad \forall\, i = 1..N\ \ and\ \ j = 1..M \qquad \textbf{. . . (3)}$$

The training is stopped after 100 epochs and it is fail if the error is increased for more than five sequence epochs.

The proposed NF scheme will be trained six times and the learning parameters are kept the same in all times. The final values of performance measure obtained from training the proposed NF six times are listed in table 1 and figure 6 shows their performance measure.

Table 1: Final performance measure of the proposed NF scheme

Training No.	Number of membership functions	Initial learning rate	Initial width of memberships	Performance
1	9	0.4504	0.3985	0.00358
2	11	0.1770	0.2943	0.00326
3	9	0.4711	0.3205	0.00339
4	10	0.2755	0.2984	0.00351
5	10	0.2514	0.3720	0.00338
6	11	0.4444	0.4492	0.00344

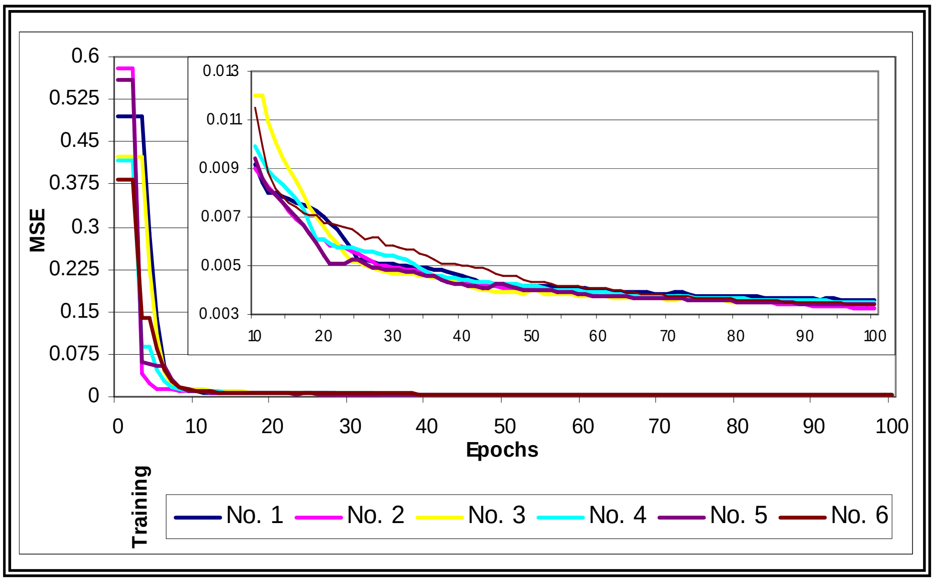


Figure 6: Performance measure charts of the proposed NF Scheme

6. Experiments and Results

To check the efficiency of the proposed technique, the ten test images (five grayscale and five truecolor) was artificially corrupted by impulsive noise with corruption rates of 10%, 15%, 20%, 25%, 30%, and Poisson noise separately. In all tests, the MF and AF will be applied by using mask size 3×3.

To check the efficiency, two criteria are used. They are **Mean Square Error (MSE)** and **Signal to Noise Ratio (SNR)** [11, 16] between original image and the result of filtering. For each test image, the MSE and SNR resulted by proposed NF, MF, and AF for five testing times will be calculated and their average will be reported. The results of impulsive noise are summarized in Figures 7, 8, 9, 10, and 11, while the results of Poisson noise are summarized in figure 12. The images of grayscale (Albert) of 15% impulsive noise filtering are presented in figure 13, and the images of truecolor (Shawl) of 25% impulsive noise filtering are presented in figure 14. The images of grayscale (Eight) of Poisson noise filtering are presented in figure 15, and the images of truecolor (Trees) of Poisson noise filtering are presented in figure 16. From the following figures, it is easy to see that the proposed NF scheme gives good results for both type of noise compared with the other filters.

Finally, a comparison between the values of MSE and SNR that are obtained by the proposed NF and other different algorithms for different impulsive noise ratios in the truecolor image (Boats) is summarized in figure 17.

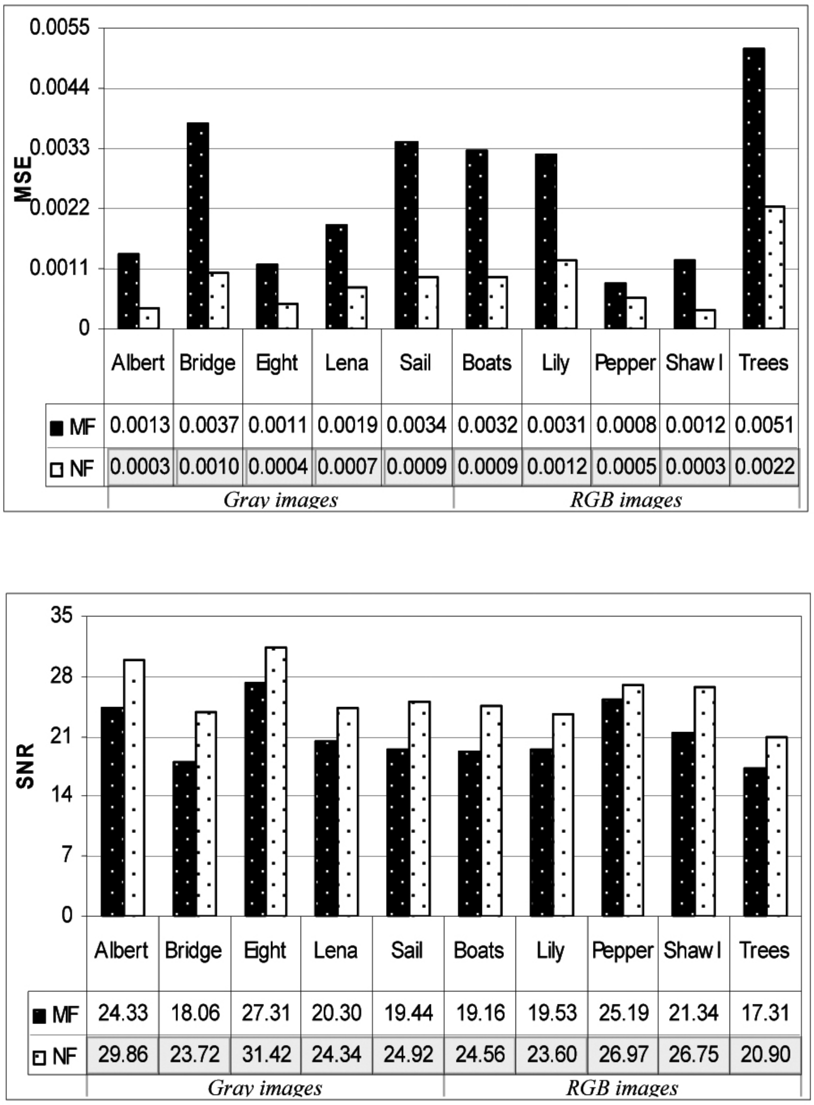
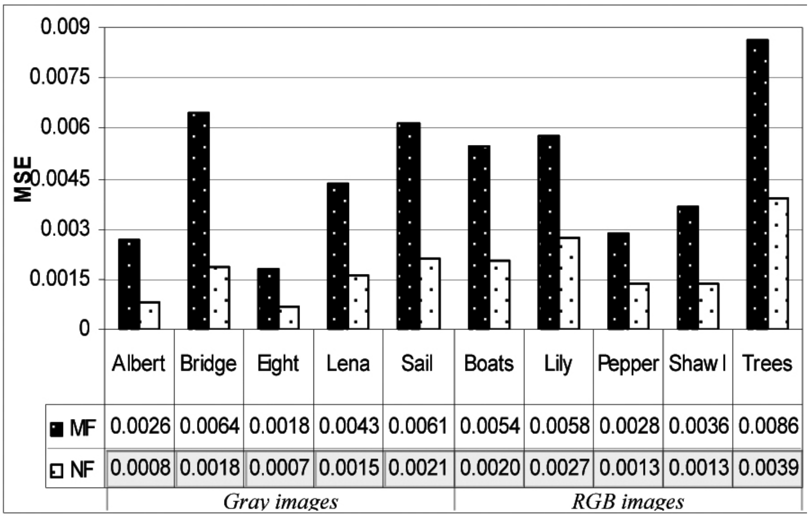


Figure 7: The average (MSE/SNR) of the test images that corrupted by 10% Impulsive noise and filtered by NF scheme together with the ordinary MF.



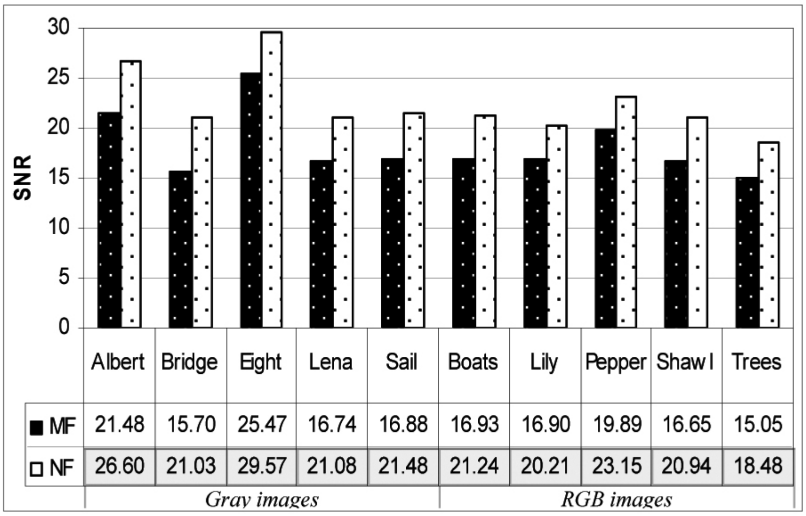


Figure 8: The average (MSE/SNR) of the test images that corrupted by 15% Impulsive noise and filtered by NF scheme together with the ordinary MF.

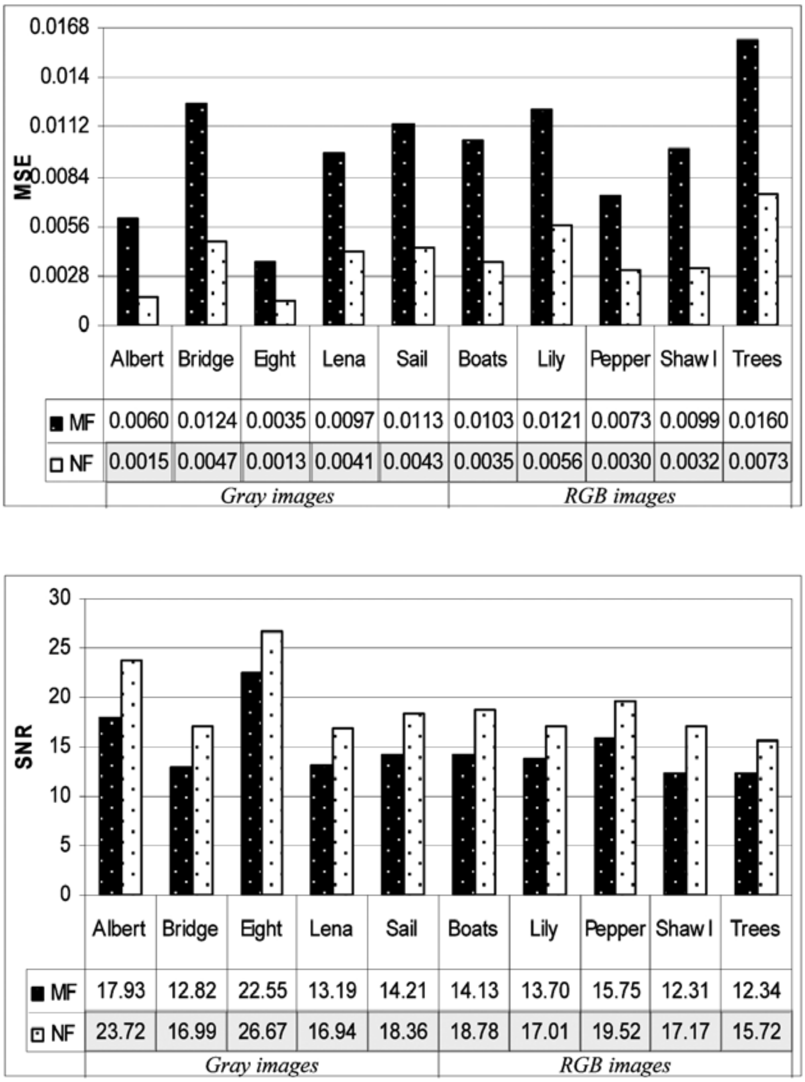
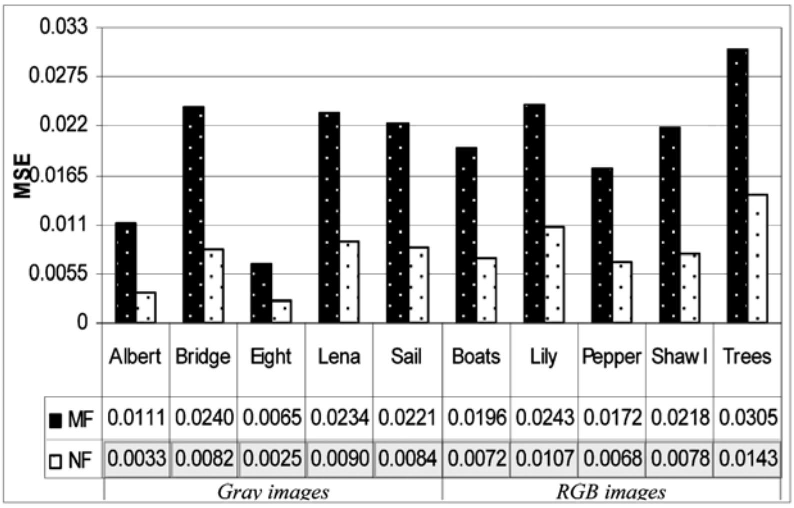


Figure 9: The average (MSE/SNR) of the test images that corrupted by 20% Impulsive noise and filtered by NF scheme together with the ordinary MF.



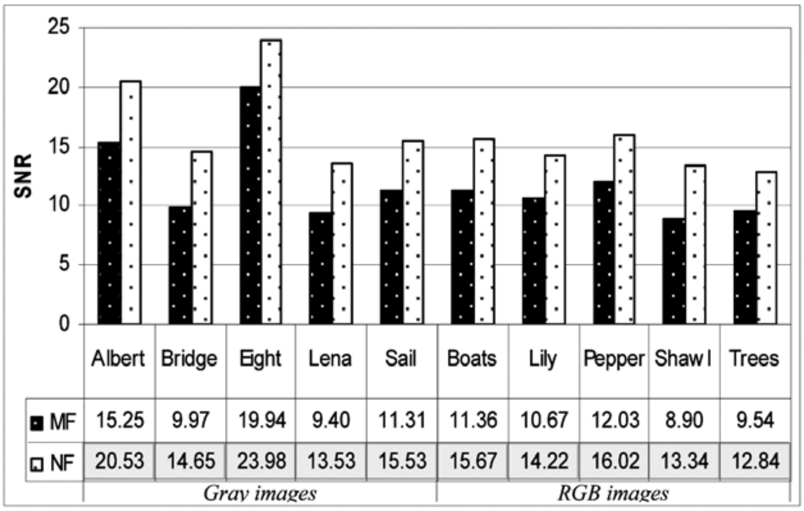
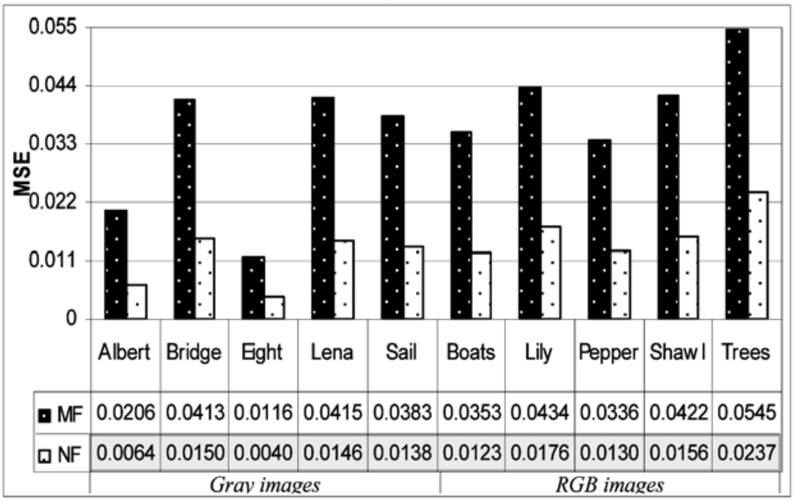


Figure 10: The average (MSE/SNR) of the test images that corrupted by 25% Impulsive noise and filtered by NF scheme together with the ordinary MF.



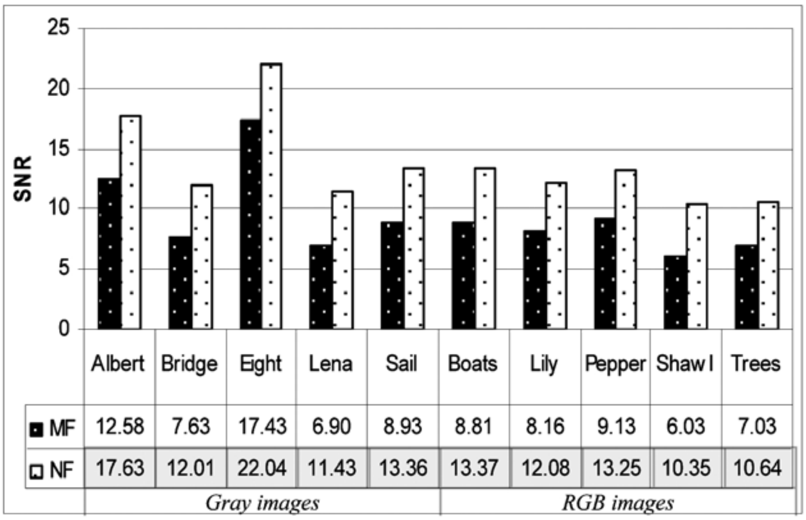
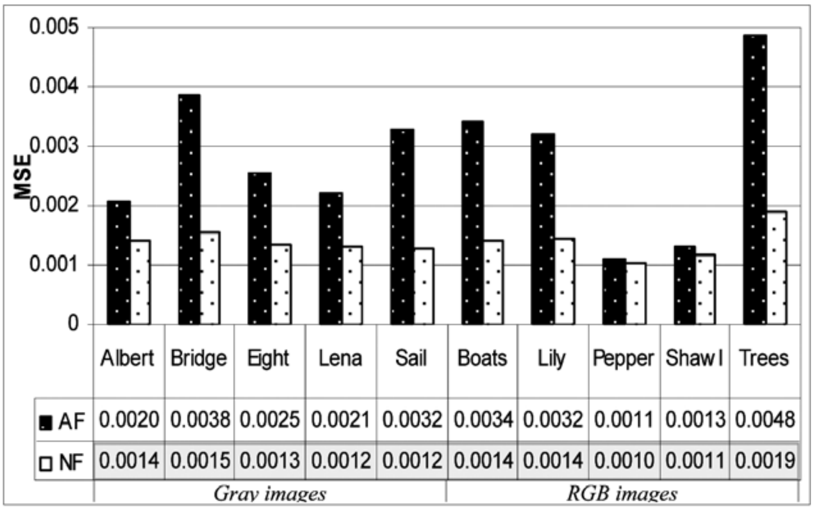


Figure 11: The average (MSE/SNR) of the test images that corrupted by 30% Impulsive noise and filtered by NF scheme together with the ordinary MF.



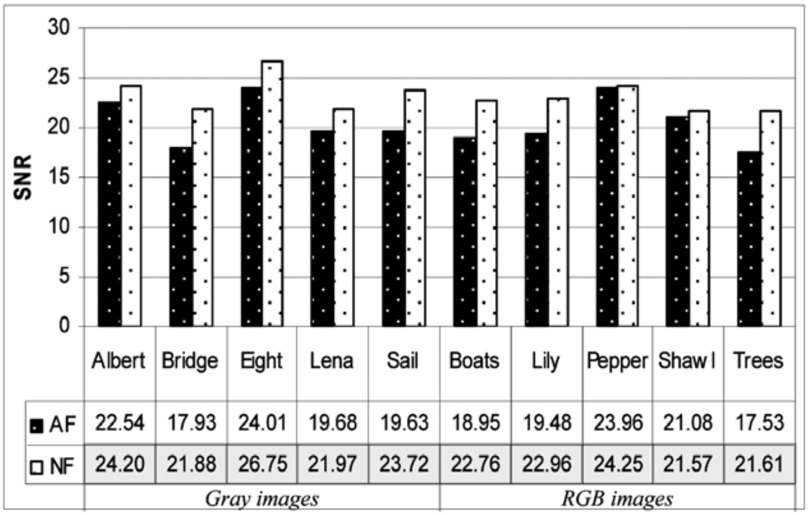
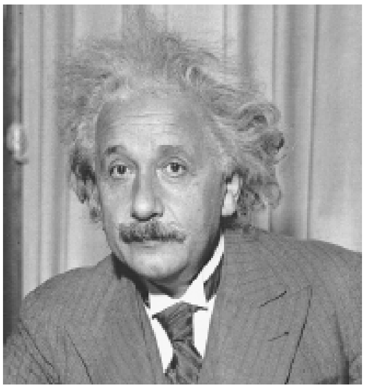
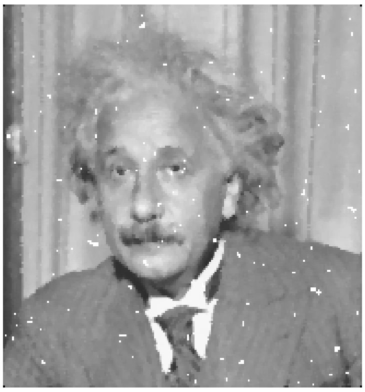


Figure 12: The average (MSE/SNR) of the test images that corrupted by Poisson noise and filtered by NF scheme together with the ordinary AF.

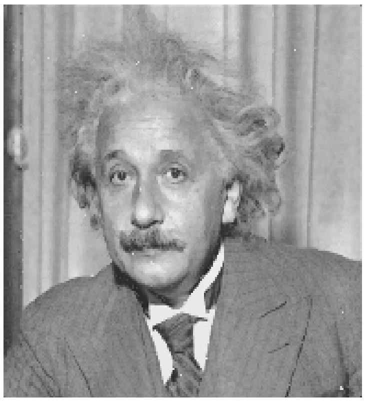


Original

Noisy



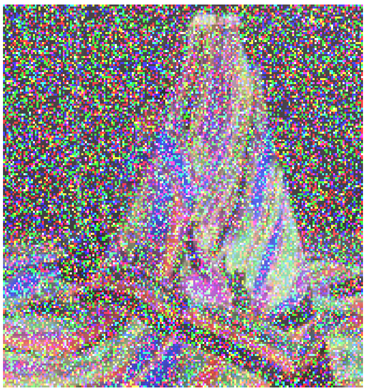
MF



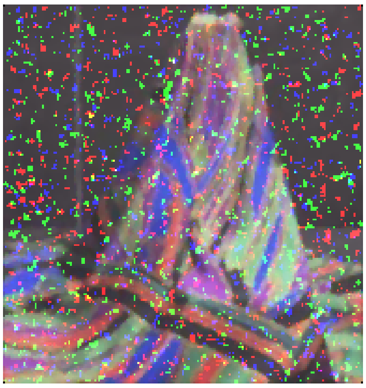
NF

Figure 13: The output of the Albert grayscale image corrupted by 15% Impulsive noise that obtained by NF scheme together with the output of MF.

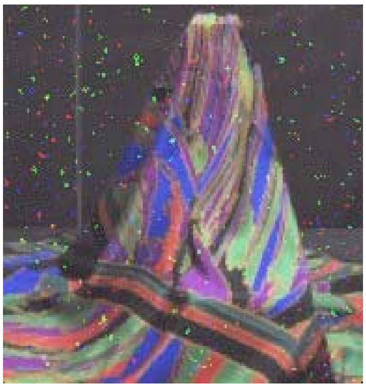
Original



Noisy



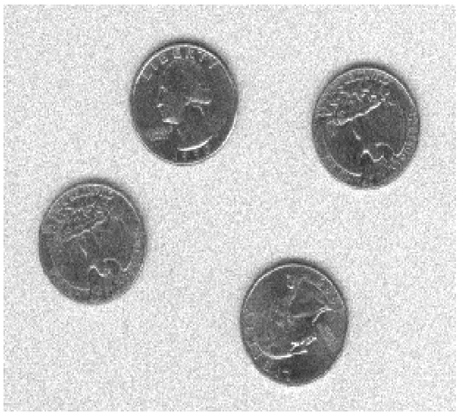
MF



NF

Figure 14: The output of the Shawl truecolor image corrupted by 25% Impulsive noise that obtained by NF scheme together with the output of MF.

Original

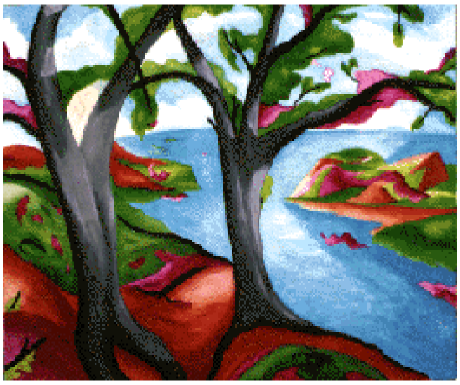


Noisy

MF

NF

Figure 15: The output of the Eight grayscale image corrupted by Poisson noise that obtained by NF scheme together with the output of AF.



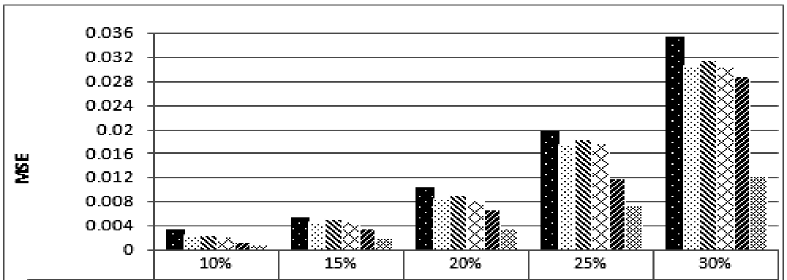
Original

Noisy

MF

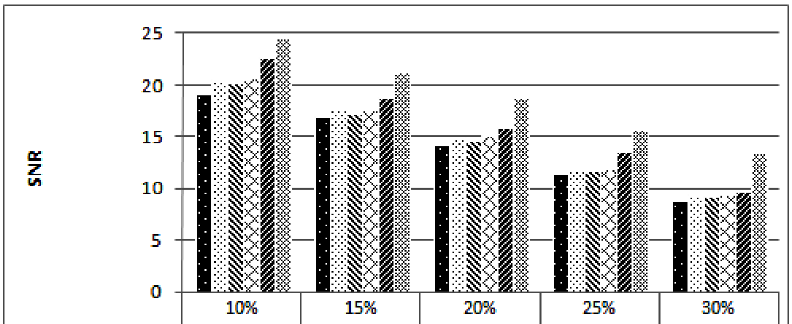
NF

Figure 16: The output of the Trees truecolor image corrupted by Poisson noise that obtained by NF scheme together with the output of AF.



	10%	15%	20%	25%	30%
MF	0.00326	0.00545	0.01037	0.01966	0.03538
VMF	0.00242	0.00462	0.00847	0.01748	0.03054
DDF	0.00254	0.00506	0.00914	0.01826	0.03158
RNF	0.00227	0.00460	0.00827	0.01761	0.03032
IVM	0.00144	0.00356	0.00681	0.01192	0.02879
Proposed NF	0.00094	0.00202	0.00356	0.00729	0.01236

Impulsive Noise Ratio



	10%	15%	20%	25%	30%
MF	19.16	16.93	14.13	11.36	8.81
VMF	20.45	17.64	15.01	11.87	9.44
DDF	20.24	17.25	14.68	11.68	9.30
RNF	20.74	17.66	15.12	11.84	9.48
IVM	22.71	18.78	15.96	13.53	9.70
Proposed NF	24.55	21.24	18.78	15.67	13.37

Impulsive Noise Ratio

Figure 17: A comparison between the values of MSE and SNR that are obtained by the proposed NF and other different algorithms for different impulsive noise ratios in the truecolor image (Boats)

7. Conclusions

We present a new technique to the problem of impulsive noise and Poisson noise reduction for grayscale and truecolor images based on Takagi neuro-fuzzy network.

A modified Takagi NF architecture is proposed and presented where only the neurons that facing each other will be connected. This is reduced the number of connections to the number of membership functions. The time of computation is reduced also by using an artificial image in training.

Several experiments were performed our proposed NF on different images with different impulsive noise ratios (10%, 15%, 20%, 25%, and 30% of corrupted pixels) and Poisson noise. These experiment results demonstrate that our proposed NF is superior, in terms of performance (both in visual and in MSE and SNR results when compared with other common AF and MF), to algorithms commonly used for impulsive and Poisson noise reduction for both type of image. It provides good results for impulsive noise elimination by the median filter and, also provides good results for Poisson noise elimination by the average filter. Table 2 shows the average of the improvement percentages of MSE and SNR for (five grayscale and five truecolor) test image for all test impulsive rates and Poisson noises.

Table 2: Average of the improvement percentages of MSE & SNR for all test images

	Poisson noise		**Impulsive noise**									
			Noise ratios									
			10%		**15%**		**20%**		**25%**		**30%**	
Criteria	*SNR*	*MSE*	*SNR*	*MSE*	*SNR*	*MSE*	*SNR*	*MSE*	*SNR*	*MSE*	*SNR*	*MSE*
GRAY	14.66	48.08	23.44	67.65	25.43	65.93	28.13	63.18	36.64	64.10	47.84	65.27
RGB	12.96	38.30	20.45	58.63	22.01	57.29	29.57	59.70	37.74	59.19	53.60	61.05
Average	13.81	43.19	21.95	63.14	23.72	61.61	28.85	61.44	37.19	61.65	50.72	63.16

On the other hand, and comparing to other filters, the proposed method achieves a high SNR and low MSE even when the noise level is high.

Experimental results show that the current technique impressively outperforms other techniques. It is fast, efficient, and can easily be extended by adding another input filter, such as adaptive or weighted median filter and winner filter to give good results. It is comparable with the best available filtering schemes

8. References

1. G. Baker, *Image Noise*. Website: www.cs.nu.oz.au/~gavinb/download.php?file=noise.pdf
2. Anonymous, *Image Processing Toolbox*, User's Guide, version 4, for use with MATLAB, Massachusetts: The Math Works Inc., 1998.
3. J. Astola, P. Haavisto, and Y. Neuvo, *Vector Median Filters*, Proceedings of the IEEE, vol. 78, no. 4, pp. 678-689, 1990.
4. D. Karakos, and P. Trahanias, *Combining vector median and vector directional filters: the directional-distance filters*, Proceeding of the IEEE Conference Image Processing, pp.171-174, 1995.
5. C.-S. Lee, Y.-H. Kuo, and P.-T Yu, *Weighted fuzzy mean filters for image processing*, Fuzzy Sets Systems, no. 89, pp. 157-180, 1997.
6. F. Farbiz, and M. Menhaj, *Fuzzy Techniques in Image Processing*, New York: Springer-Verlag, Studies in Fuzziness and Soft Computing, vol.52, pp. 194-221, 2000.
7. S. Schulte,V. De Witte, M. Nachtegael, D. Van Der Weken, and E. Kerre, *Fuzzy two-step filter for impulse noise reduction from color image*, IEEE Trans. On Image Processing, vol.15, no.5, 2006.
8. X. Liu,B. Li, Z. Su, X. Shi, and F. Liu, *Impulsive noise removal by nonlocal fuzzy diffusion*, Fourth International Conference on Image and Graphics, pp. 155-159, 2007.
9. Y. Li, F.-L Chung, and S Wang, *A robust neuro-fuzzy network approach to impulse noise filtering for color images*, Applied Soft Computing, vol. 8, pp. 872-884, 2008.
10. G. Deepti, M. Borker, and J. Savaswamy, *Impulse Noise Removal from Color Images with Hopfield Neural Network and Improved Vector Median Filter*, Sixth Indian Conference on Computer Vision, Graphics and Image Processing, 2008.
11. R. Gonzales, and R. Woods, *Digital Image Processing*, 2nd Edition, Englewood Cliffs, NJ: Prentice Hall, 2001.
12. S. B. Aziz, *Image filtering based on soft computing techniques*, PhD thesis, Basrah University, 2006.
13. H. Qin, *Nonlinear adaptive noise cancellation for 2-D signals with Neuro-Fuzzy inference system*, M.Sc. thesis, Guelph University, 2005.
14. H. Koivo, *Soft Computing in Dynamical Systems*, PhD thesis, Helsinki University of Technology, 2001.

15. N. kasabov, J. Kim, A. Gray, and M. Watts, *FuNN - A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition*, Information Science Dept., Otago University, Dunedin, New Zealand, 2001.

16. H. Demuth, and M. Beale, *Neural Network Toolbox*, User's Guide, version 4, for use with MATLAB, Massachusetts: The Math Works Inc., 1998.

17. A. Kokaram, *Introduction to Electrical Engineering: Digital Image and Video Processing*, Dept. of Electronic and Electrical Engineering, Trinity College, Dublin University, 2004.

إزالة الضوضاء النبضية وضوضاء باوسون
بإستعمال شبكة عصبية مضببة

سارة بهنام عزيز و ميثم أبو الهيل شهيد

قسم علوم الحاسبات، كلية العلوم، جامعة البصرة

البصرة، العراق

الملخص :

في هذا البحث تم تطبيق مرشح شبكي عصبي مضبب جديد من نوع Takagi لغرض إزالة وتقليص الضوضاء عن صور مشوه بضوضاء نبضية impulsive وأخرى من نوع باوسون Poisson. حيث تم خفض عدد الوصلات المستخدمة في الشبكة العصبية المضببة ليكون مساوٍ فقط لعدد دوال الانتماء membership functions، أيضاً تم تقليص الوقت اللازم لتدريب الشبكة المقدمة وذلك باستخدام صورة اصطناعية.

لقد أظهرت نتائج التجارب جدوى الطريقة المقترحة، حيث كانت النتائج أفضل من تلك التي يتم الحصول عليها من تطبيق المرشحات القياسية من حيث المقاييس الرقمية والمرئية.

طُبقت الطريقة المقترحة على كل من الصور الملونة وغير الملونة، وامتازت بالكفاءة والسرعة، مع إمكانية توسيعها بإضافة مرشح آخر دون الحاجة لإجراء أي عمليات تمهيدية على الصور.

كلمات المفاتيح: الضوضاء النبضية، ضوضاء باوسون، المرشح المتوسط، المرشح الوسيط، الشبكات العصبية المضببة