Arab Gulf Journal of Scientific Research, 27 (3): 156- 167 (2009)

Research Paper Ref. No 2489

Improved Smoothing Filters Effects Using Mamdani Neuro Fuzzy Network

تحسين أداء مرشحات التنعيم باستخدام الشبكة العصبية المضببة (ممداني)

Sarah B. Aziz and Maytham A. Shahed

سارة بهنام عزيز و ميثم أبو الهيل شهيد

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

E-mail:maythamalbasri@yahoo.com

Abstract: The results of the noise removal have a strong influence on the quality of the other image processing techniques. And since the main goal for any noise removal method is the preservation of the edges and image information, this influence is the principal drawback of the smoothing filters. In this paper, a modified Mamdani neurofuzzy network scheme has been proposed and presented to improve effects of smoothing filters in removing impulsive noise. The number of connections is reduced to be equal to the number of membership functions plus one. A training strategy for the presented neurofuzzy is based on artificial image which reduces the time of computation. The Analysis shows that the proposed neurofuzzy works well in suppressing impulsive at different noise ratios for grayscale and truecolor images.

Keywords: *Impulsive noise, Median filter, Average filter, Mamdani neurofuzzy, Adaptive back-propagation algorithm.*

المستخلص: إن نتائج تقنيات إزالة الضوضاء له تأثير قوي على نوعية تقنيات معالجة الصور الأخرى، حيث إن الهدف الرئيسي لأي تقنية أزاله الضوضاء هو الحفاظ على حواف ومعلومات الصورة، ويعتبر هذا التأثير المشكلة الرئيسية لمرشحات التنعيم. البحث الحالي يستخدم الشبكة العصبية المضببة (ممداني) لتحسين انجازية مرشحات التنعيم (المرشحان الوسيط والمعدل)، حيث يعرض تطوير لمعمارية الشبكة العصبية المضببة من خلال تقليل عدد ارتباطات الشبكة إلى عدد دوال الانتماء المستخدمة زائداً واحد بدلاً من عدد دوال الانتماء مرفوعة لعدد مدخلات الشبكة. إضافة إلى تقليل زمن تدريب الشبكة وذلك باستخدام صورة اصطناعية للتدريب بدلاً من صورة حقيقية مما يقلل زمن الاحتساب المستخدم ويزيد من سرعة التقنية المقترحة. تم اختبار التقنية المقترحة على مجموعة من الصور العادية والملونة والتي تم تشويهها بضوضاء impulsive وبنسب متفاوتة وأثبتت التقنية المقترحة قدرة عالية في حذف الضوضاء مقارنة مع نتائج مرشحات التنعيم التقليدية.

كلمات مدخلية: *الضوضاء النبضية، مرشح الوسيط، مرشح المعدل، الشبكات العصبية المضببة، خوارزمية التعليم المتكيف.*

INTRODUCTION

Currently, image processing has numerous applications, which all result from the interaction between fundamental scientific researches on the one hand and the development of new technology on the other hand. The scientific community has become aware of soft computing techniques in image processing quite recently. Soft computing consists of several new components, such as fuzzy set theory, neural networks, and genetic algorithms. In the past years, these techniques have proven their richness and power in the field of image processing. This fact is evident from the increasing number of publications and patents, and the fact that image processing is mentioned as a topic in most international conferences about soft computing. Because soft computing is an

emerging field consisting of complementary elements of fuzzy logic, neural computing, evolutionary computation and machine learning, is characterized by a strong learning and cognitive ability and good tolerance of imprecision and uncertainty, soft computing techniques have found wide applications in society. Needless to say, image processing is one of the applications. The areas in which soft computing is a factor in image processing include but are not limited to the following topics: noise reduction in images, edge detection and segmentation, feature extraction, image enhancement, image compression, similarity between images, image analysis, medical imaging, visualization, and fuzzy wavelets.

This paper focuses on noise reduction and improves the effects of smoothing filters in removing different ratios of impulsive noise from grayscale and truecolor images by presenting an effective technique based on mamdani neurofuzzy network scheme. An artificial image will be used as training data set instead of real image in order to reduce the time of training. The quality in both terms of *"Mean Square Error"* (MSE) and *"Signal to Noise Ratio"* (SNR) are better than those obtained by the conventional image filtering techniques.

Smoothing Filters

Smoothing filters are used for noise reduction. Noise reduction is also called low pass filter since it leaves the low frequency components unchanged (Gomes and Velho, 1997; Baker, 2005). The proposed filter scheme in this work is to improve the effects of two kinds of smoothing filters. These are median filter and average filter (Aziz, 2006; Gonzales, *et al*.2009). The median filter (MF) is a nonlinear operator that arranges the pixels in a local window according to their intensity values and replaces the value of the pixel in the result image by the middle value in this order.The Average Filter (AF) also called Mean Filter is a linear filter. It takes the average of intensity values in an (N×N) region of each pixel as the new pixel value. The normalization factor (N×N) preserves the range of values of the original image. If we replace each pixel in the noisy image by the average of its local neighbors within an (N×N) mask then the image will be smoothed because averaging elements will reduce the noise mean towards zero. AF is a simple filter for all the types of noises but it is more effective in removing additive noise.

In the truecolor image, each pixel can be described by the base colors Red, Green, and Blue (RGB). By these color triplets, all color can be created. Consequently, the color image is usually represented by triplet of matrixes RGB. Therefore, the smoothing filter will be applied on matrix of each color separately then combined the three resulted grayscale images to get the filtered true color image (Aziz, 2006). The principle is shown in Figure (1). In this paper, the MF and AF will be applied by using a local window of size 3×3.

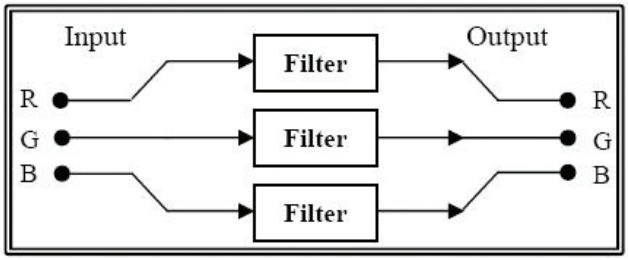


Fig. 1. RGB Component filtering.

NeuroFuzzy Network

The architecture of the NeuroFuzzy (NF) network based on Mamdani 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 NF network (Koivo, 2001; Kasabov, *et al*. 2001; Qin, 2005; Aziz, 2006).

The proposed architecture is based on Mamdani NF which has five layers and three inputs. The scheme has one output and the number of membership functions is generated randomly at each run in the range between 7 and 13. In common Mamdani NF architecture, full connections are used to connect the neurons of fuzzification layer (membership functions), thus the number of connections is equal to number of membership functions powered to the number of inputs. i.e. number of connections is 343 when the number of membership functions is 7 and

2197 when it is 13. As noticed, the number of connections is huge and that will cause impossible learning in training. A suggested modification to the NF architecture can alleviate this drawback, where only the neurons that facing each other were connected and additional one that connects all the neurons of fuzzification layer together. Thus, due to this only 14 connections as maximum are used. The schematic diagram of the NF architecture with our suggested modification is shown in Figure (3).

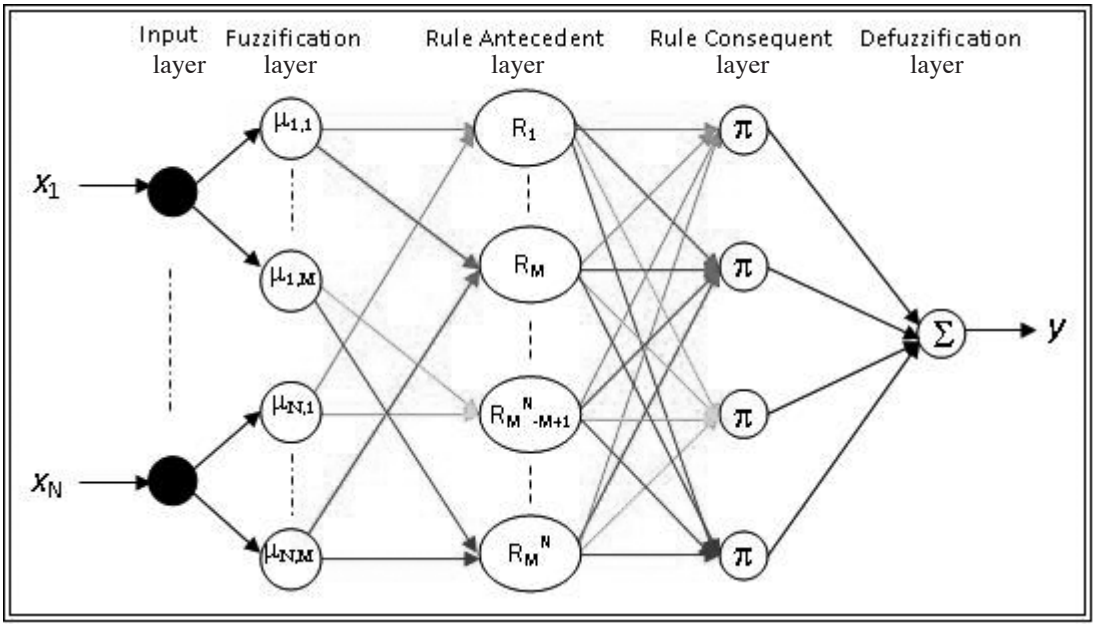


Fig. 2. Structure of Neuro Fuzzy (NF) network.

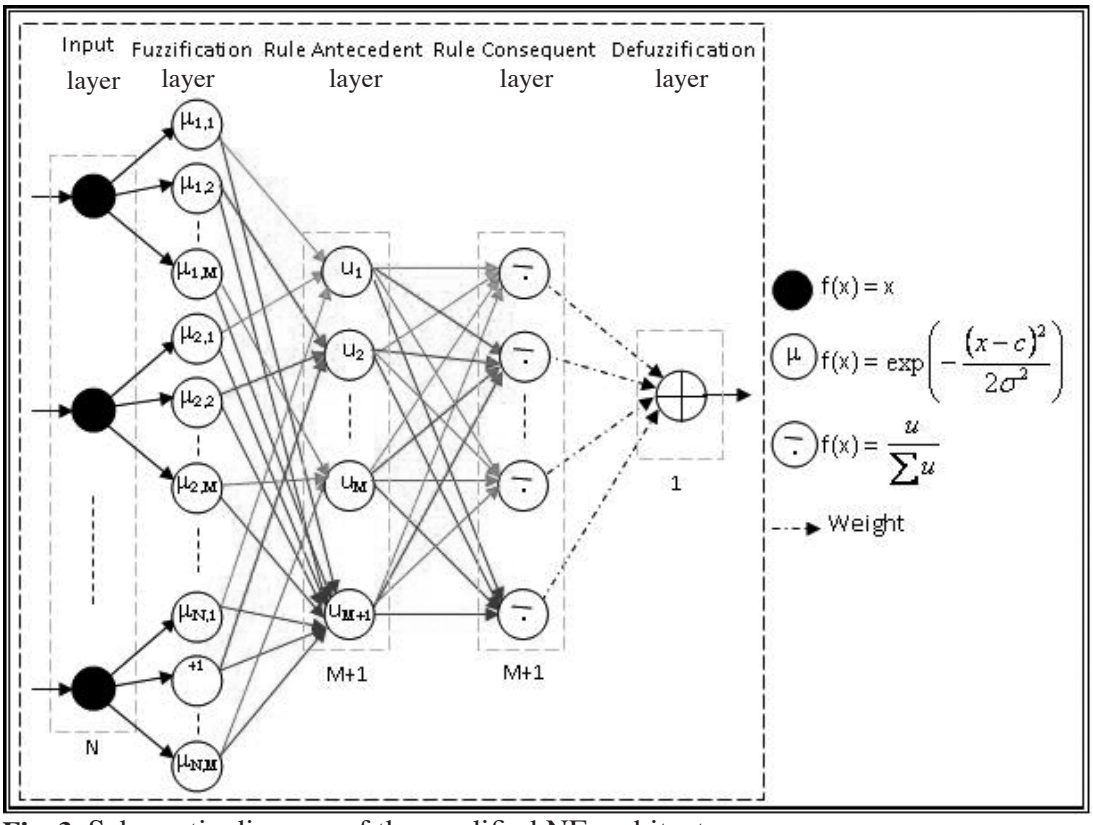


Fig. 3. Schematic diagram of the modified NF architecture.

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 Back-propagation (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.

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

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

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

***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. For a Gaussian-shaped membership function, the activation function for each node is as in equation (2):

$$\mu_{(i,j)}=\exp(-Net_i\text{-}C_{(i,j)})2/2x\sigma^2_{(i,j)}$$
$$\forall_i=1..N \textit{ and } j=1..M \qquad \textbf{(2)}$$

***Rule Antecedent layer*:** The implication method is performed by this layer and applied using the products. Where the output of each node is calculated as in equation (3):

$$u_M^{(i\text{-}j)}{}_{+j}=\prod\prod \qquad \textbf{(3)}$$
$$\forall_i=1..N \textit{ and } j=1..M$$

***Rule Consequent layer*:** The normalization process is applied in this layer by implementing equation (4).

$$\bar{u}_k = \frac{u_k}{\sum_{k=1}^{M^N} u_k} \quad \forall k = 1..M^N \qquad \textbf{(4)}$$

***Combination and Defuzzification layer*:** This layer performs defuzzification to produce a crisp output value. Among the commonly used defuzzification strategies, the center of gravity (COG) method yielded the best result. In this layer, linear output activation function is used (equation 5):

$$y = \sum_{k=1}^{M^N} \bar{u}_k \times w_k \qquad \textbf{(5)}$$

Backward Phase

The goal of this phase is to minimize the error which is calculated as in equation (6).

$$E = 0.5 \times (y - d)^2 \qquad \textbf{(6)}$$

Where, d is the desired output.

The learning algorithm in NF is realized by adjusting connection weights of the neurons beside the centers and widths of membership functions. Thus, the adaptation of weights of neurons is calculated using equations (7-9).

$$w_k^{new} = w_k^{old} + \Delta w_k \quad \forall k = 1..M^N \qquad \textbf{(7)}$$

Where,

$$\Delta w_k = -\eta \times \delta_k^w \qquad \textbf{(8)}$$

$$\delta_k^w = \frac{\partial E}{\partial w} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial w} = (y - d) \times \frac{u_k}{\sum_{k=1}^{M^N} u_k} \qquad \textbf{(9)}$$

While, adaptation of centers and widths of membership functions is calculated as in equations (10, 12-13) for centers and equations (11, 14-15) for widths.

$$c_{(i,j)}^{new} = c_{(i,j)}^{old} + \Delta c_{(i,j)} \quad \forall i = 1..N \text{ and } j = 1..M \tag{10}$$

And

$$\sigma_{(i,j)}^{new} = \sigma_{(i,j)}^{old} + \Delta\sigma_{(i,j)} \quad \forall i = 1..N \text{ and } j = 1..M \tag{11}$$

Where:

$$\Delta c_{(i,j)} = -\eta \cdot \delta_{(i,j)}^{c} \tag{12}$$

$$\delta_{(i,j)}^{c} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial \mu} \times \frac{\partial \mu}{\partial c} = \frac{\left(2 \times (y-d) \times (w_k - y) \times u_k \times (Net_i - c_{(i,j)})\right)}{\left(\sigma_{(i,j)}^{3} \times \sum_{k=1}^{M''} u_k\right)} \tag{13}$$

And

$$\Delta\sigma_{(i,j)} = -\eta \times \delta_{(i,j)}^{\sigma} \tag{14}$$

$$\delta_{(i,j)}^{\sigma} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial \mu} \times \frac{\partial \mu}{\partial \sigma} = \frac{\left(2 \times (y-d) \times (w_k - y) \times u_k \times (Net_i - c_{(i,j)})^2\right)}{\left(\sigma_{(i,j)}^{3} \times \sum_{k=1}^{M''} u_k\right)} \tag{15}$$

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:

- 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:
  The output of the nodes except the last one is computed as in equation (16):

$$u_j = \prod_{i=1}^{N} \mu_{(i,j)} \quad \forall j = 1..M \tag{16}$$

  Where, N is number of inputs (N = 3) and M is number of membership functions.
  While, the output of the last node will be calculated as in equation 3. Therefore, the carrying out of the last two layers of NF network will be achieved as in equations (17 and 18).

$$\overline{u}_j = \frac{u_j}{\sum_{j=1}^{M+1} u_j} \quad \forall j = 1..M+1 \tag{17}$$

And

$$y = \sum_{j=1}^{M+1} \overline{u}_j \times w_j \tag{18}$$

- BP algorithm will be improved by using adaptive learning rate. Therefore, the parameters are updated under the following conditions which are listed as in equation (19):

$$\begin{aligned} &\text{if } (newerror/olderror) > 1.04 \text{ then} \\ &\quad New values of parameters are discarded \\ &\quad lr = lr \times lr\_dec \\ &\text{else} \\ &\quad Update Parameters \\ &\quad \text{if } newerror < olderror \\ &\quad\quad lr = lr \times lr\_inc \end{aligned} \tag{19}$$

- Weights will be updated by using equation (7). Where, is calculated as in equation (20).

$$\frac{\partial E}{\partial y} = (y - d) \tag{20}$$

- The adaptation of the other parameters (centers and widths of membership functions) are affected by the type of connections. The actual effect gets from the term of centers and widths errors (δ^C and δ^σ). Thus, centers and widths of membership functions are adapted as in equations 11 and 12 respectively. Where, δ^C and δ^σ are calculated as in equations (21 and 22).

$$\delta^{\mu}_{(i,j)} = \frac{\partial E}{\partial y}\times\frac{\partial y}{\partial u}\times\frac{\partial u}{\partial \mu}\times\frac{\partial \mu}{\partial c} = \left(2\times(y-d)\times(w_j-y)\times\bar{u_j}\times(Net_i-c_{(i,j)})\right)\Big/\sigma^2_{(i,j)} + \left(2\times(y-d)\times(w_{M+1}-y)\times\bar{u}_{M+1}\times(Net_i-c_{(i,j)})\right)\Big/\sigma^2_{(i,j)} \quad \forall i=1..N \text{ and } j=1..M \tag{21}$$

And

$$\delta^{\sigma}_{(i,j)} = \frac{\partial E}{\partial y}\times\frac{\partial y}{\partial u}\times\frac{\partial u}{\partial \mu}\times\frac{\partial \mu}{\partial \sigma} = \left(2\times(y-d)\times(w_k-y)\times\bar{u_j}\times(Net_i-c_{(i,j)})^2\right)\Big/\sigma^3_{(i,j)} + \left(2\times(y-d)\times(w_{M+1}-y)\times\bar{u}_{M+1}\times(Net_i-c_{(i,j)})^2\right)\Big/\sigma^3_{(i,j)} \quad \forall i = 1..N \text{ and } j= 1..M \tag{22}$$

The initial values of each of the learning rate (***lr***), weights, 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 respectively and the initial values of centers of membership functions of each input are set by using equation (23).

$$c_{[i,j]}=(j-1)/(M-1) \quad \forall i=1..N \text{ and } j=1..M \tag{23}$$

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

Data Sets

In order two implement the NF network tow 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 of size 24 24 × 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 4). 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. The number of training patterns is equal to the number of rows multiplied by the number of columns, i.e. it is 256 in case of using artificial image, the time comparing to use a normal image is decreased about 85% and more depends on the size of the normal image.

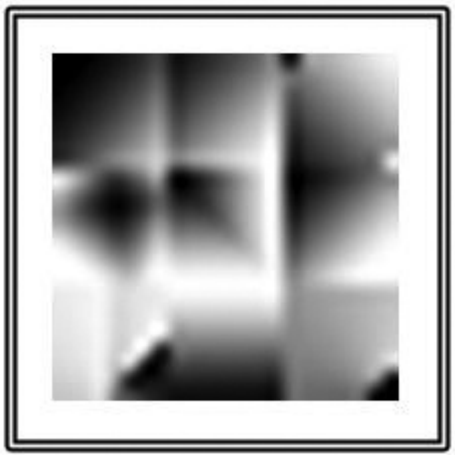


Fig. 4. The Original artificial image.

On the other side, the testing data set includes ten images. Five of them are grayscale images and the other five are truecolor images. The original copies of them with their sizes are shown in Appendix. To construct the patterns of training, the artificial image in Figure (4) will be corrupted by impulsive noise of ratio 20% and applied the AF and MF on it one at each time (Figure 5 A-C).

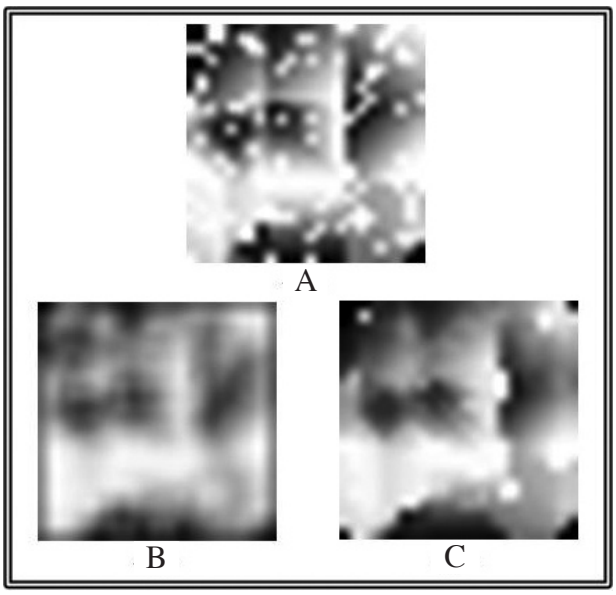


Fig. 5. (A-C) The artificial image.
A. Corrupted with 20% Impulsive Noise
B. Result image from applied AF
C. Result image from applied MF

There is a need to convert pixels values from the 8-bit 256 gray value format to a double (floating point) gray value format before forming the training patterns from these four images by scanning them row by row, in order to set the pixels values in the range [0, 1], so it can be processed by the NF scheme.

Each training pattern contains three inputs and one output. A pixel from the corrupted image (Figure 5-A) with its corresponding pixel from the image that resulted by applied AF (Figure 5-B) and its corresponding pixel from the image that resulted by applied MF (Figure 5-C) will be formed the inputs of the pattern. The output of the pattern will be formed from the corresponding pixel from the original image (Figure 4). By the same way, the patterns for a test image will be constructed from its corrupted, AF result, and MF result copies by scanning them row by row and gathering the corresponding pixels after converting them from 8-bit 256 gray values format to a double format. The output of NF scheme that obtained from processing the patterns of any test image will be in a sequence form of floating point values in range [0, 1]. This output should be converted to 8-bit 256 gray values and rearranged as matrix with size of the test image itself.

The Implementation of NF

The proposed NF scheme will be trained using the training data set that is constructed from the artificial image as explained previously. The training will be repeated five times and the learning parameters will be kept the same at each time. For each training run, the performance measure which is represented by the error (equation 6) will be listed in Table (1) beside the number of membership function that is generated for each run, while Figure (6) shows the convergence of performance measure for these five training runs. After training, the test images can be applied on the NF scheme to get the results.

Table 1. Final performance measure of the proposed NF scheme.

Training No.	Number of membership functions	Performance
1	11	0.00832
2	9	0.00833
3	7	0.00926
4	8	0.00876
5	13	0.00882

RESULTS

The idea of any kind of signal processing is to achieve some kind of desired effect on the perceived information content in the signal. In the noise removing which is a part of image enhancement and restoration, the idea is to process the data to improve illumination, or to remove defects like noise or dirt on a film. A mechanism

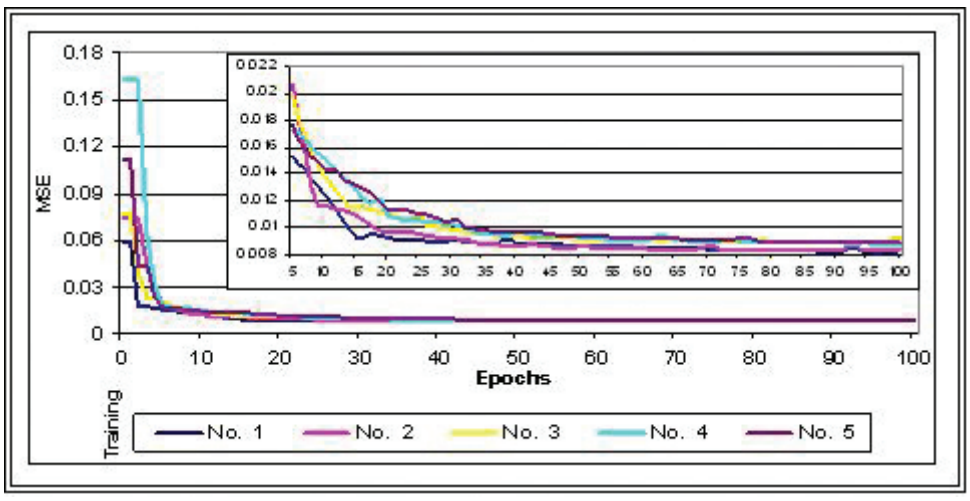


Fig. 6. Performance measure charts of the proposed NF scheme.

for assessing the damage done to the observed picture, is important to evaluate the quality of the processed output (Kokaram, 2004).

Two simple measurement units will be used to measure the observed error. They are ***Mean Squared Error*** (MSE) and ***Signal to Noise Ratio*** (SNR) (Kokaram, 2004; Aziz, 2006). These two measurement units will be used to show the results as follows:

The set of test images that are corrupted by five different ratios 10%, 15%, 20%, 25%, and 30% of impulsive noise separately will be filtered by the proposed NF scheme. For each noise ratio, the MSE beside SNR of each ten test images (five grayscale and five truecolor) resulted by the different NF schemes from the five testing times will be calculated and their average for each image will be reported together with that of corresponding average filter and median filter and lists in Figures (7), (8), (9), (10), and (11) respectively. Also, the figures of grayscale image (Albert) of 15% and truecolor image (Boats) of 25% impulsive noise obtained from NF scheme will be shown together with that obtained by the corresponding AF and MF in Figures (12) and (13) respectively.

Finally, the computation time for each test image which is calculated as in equation (24) will be listed in Table (2).

$$\text{Computation Time} = Z*(5*T+(N*M)-1) \quad \textbf{...(24)}$$

Where, T is the computing time for each layer of NF scheme, five is represented the number of layers in the NF scheme, N and M are the sizes of the image, while Z is equal to 1 in case of grayscale images and 3 in case of truecolor images.

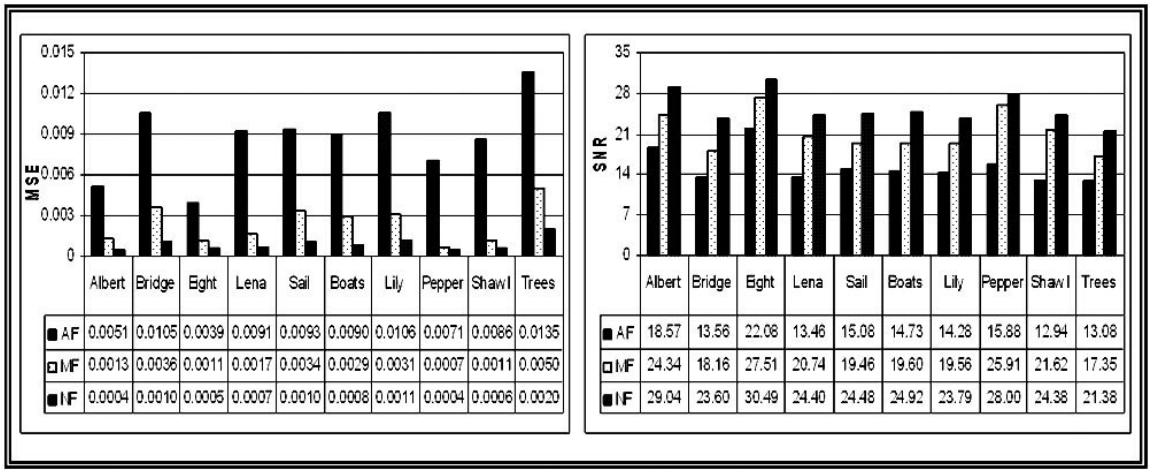


Fig. 7. The average (MSE/SNR (dB)) of the test images that corrupted by 10% impulsive noise and filtered by NF scheme together with the ordinary AF and MF.

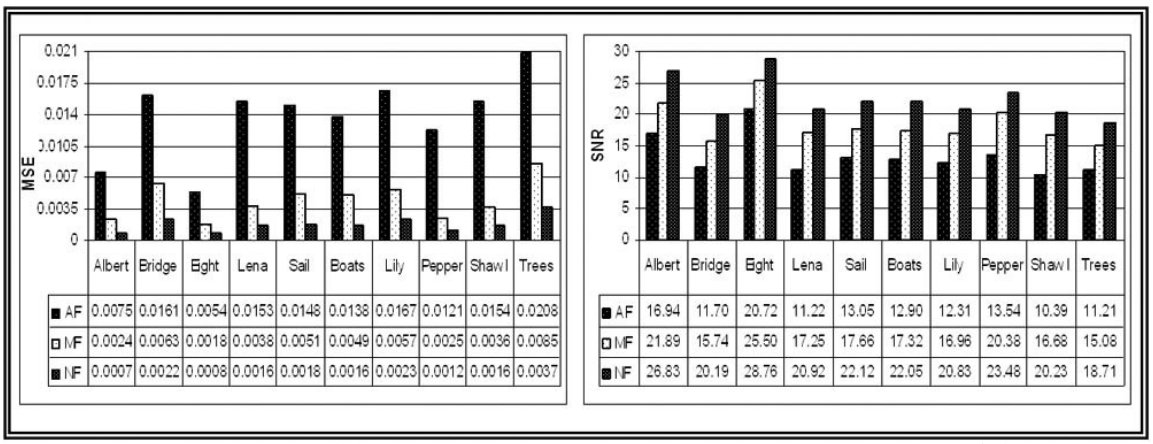


Fig. 8. The average (MSE/ SNR (dB)) of the test images that corrupted by 15% impulsive noise and filtered by NF scheme together with the ordinary AF and MF.

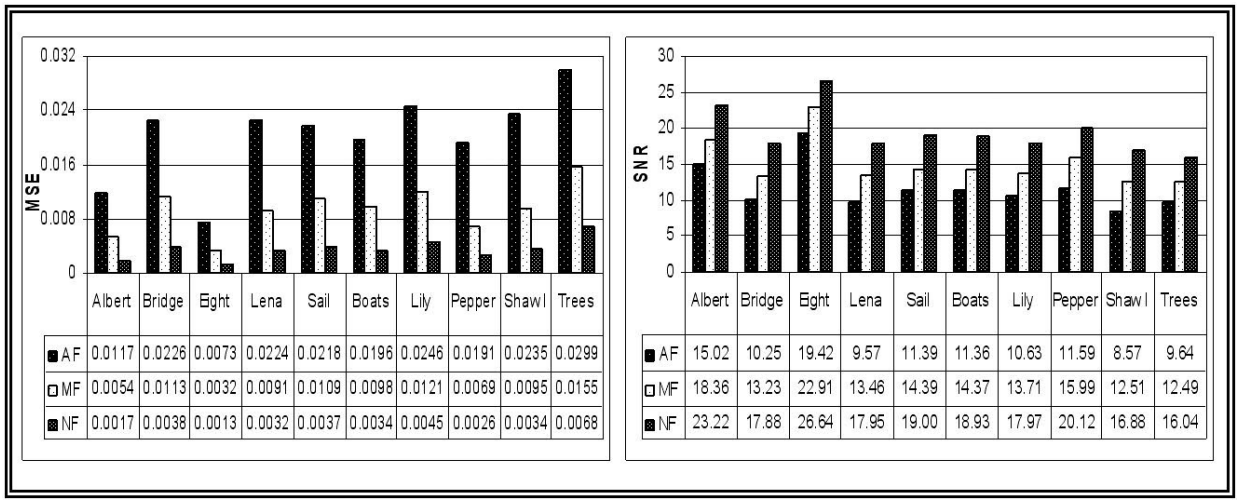


Fig. 9. The average (MSE/ SNR (dB)) of the test images that corrupted by 20% impulsive noise and filtered by NF scheme together with the ordinary AF and MF.

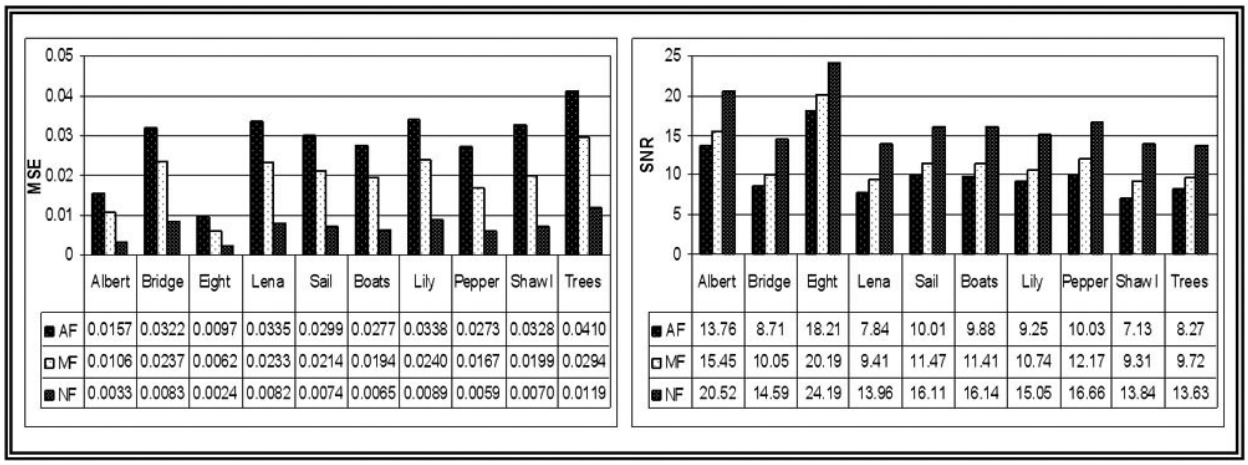


Fig. 10. The average (MSE/ SNR (dB)) of the test images that corrupted by 25% impulsive noise and filtered by NF scheme together with the ordinary AF and MF.

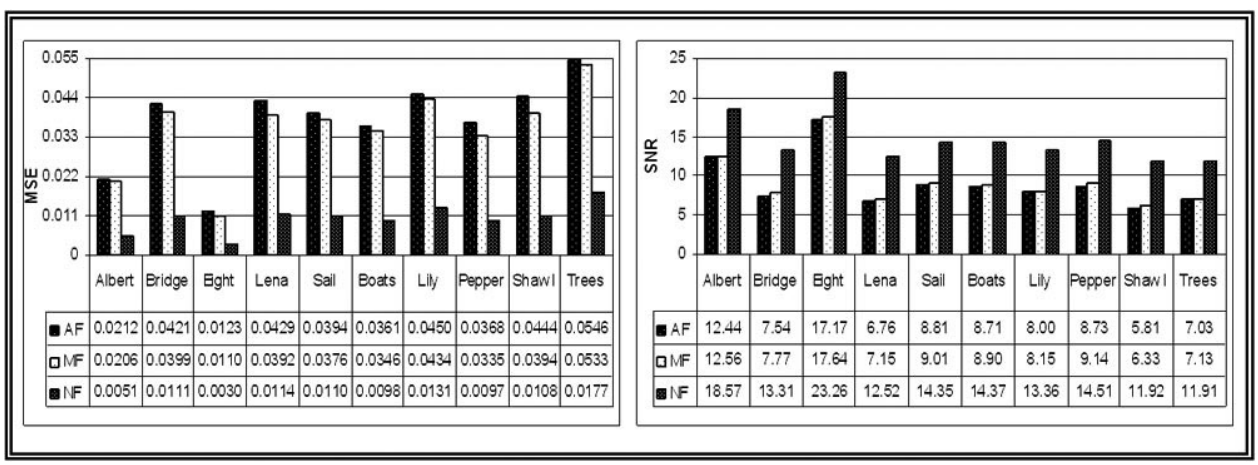


Fig. 11. The average (MSE/ SNR (dB)) of the test images that corrupted by 30% impulsive noise and filtered by NF scheme together with the ordinary AF and MF.

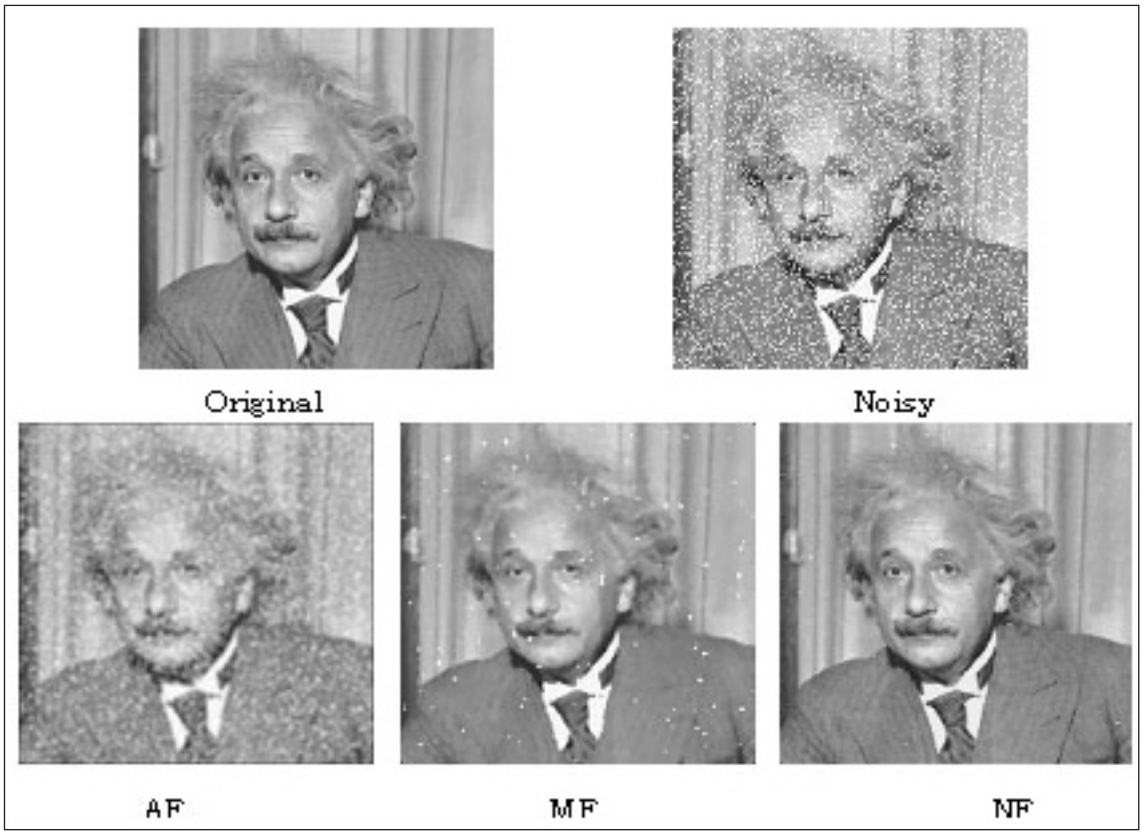


Fig. 12. The output of the Albert grayscale image corrupted by 15% impulsive noise that obtained by NF scheme together with the outputs of AF and MF.

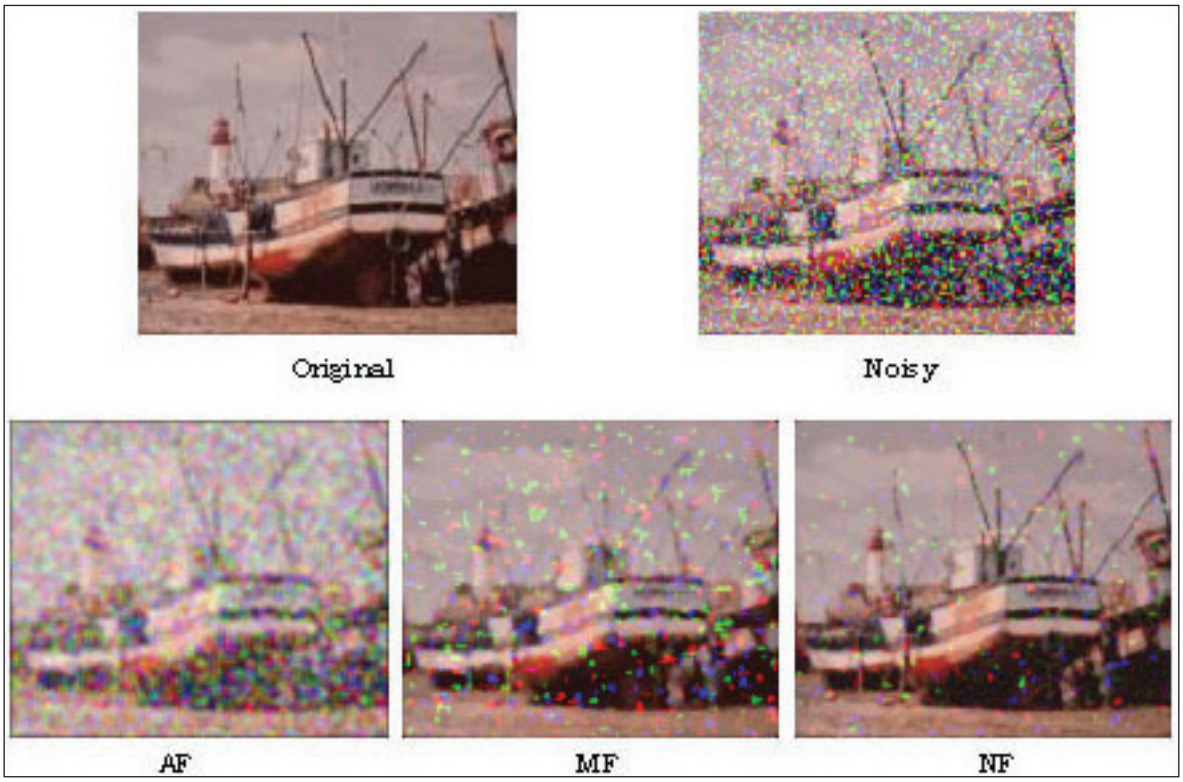


Fig. 13. The output of the Boats true color image corrupted by 25% impulsive noise that obtained by NF scheme together with the outputs of AF and MF.

Table 2. Time computation for test images.

	Image	Computation Time
Grayscale	Albert	5T+38024
	Bridge	5T+38024
	Eight	5T+74535
	Lena	5T+35720
	Sail	5T+38024
Truecolor	Boats	15T+43620
	Lily	15T+128337
	Pepper	15T+149172
	Shawl	15T+111744
	Trees	15T+151695

DISCUSSION AND CONCLUSIONS

The main objective of this paper is to improve the effects of smoothing filters using one of the most common techniques in the soft computing is the NF networks that based on Mamdani fuzzy logic system and neural network algorithms. A modified Mamdani NF architecture is proposed and 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. This is reduced the number of connections to the number of the number of membership functions plus one. The time of computation is reduced also by using an artificial image in training.

The results presented show that our proposed NF filter gives good performance in reducing impulsive noise while preserving image details for both grayscale and truecolor images this is clearly seen in visual and in terms of MSE and SNR when compared with other common AF and MF for various noise ratios. In addition, the proposed NF presents a quite stable performance over a wide variety of images.

REFERENCES

Aziz, SB. (2006) *Image Filtering Based on Soft Computing Techniques*. Ph. D. thesis, Basrah University.

Baker, G (2005) *Image Noise*. Website: www.cs.mu.oz.au/~gavinb/download.php?file=noise.pdf 2005. visited January, 2008.

Gomes, J, and Velho, L (1997) *Image Processing for Computer Graphics*. Springer-Verlag Inc, New York.

Gonzales, R, Woods, R, and **Eddine, S** (2009) *Digital Image Processing using MATLAB*, (2nded.). Gatesmark publishing, LLC.

Kasabov, N, Kim, J, Gray, A, and **Watts, M** (2001) *FuNN - A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition*. Information Science Dept., Otago University, Dunedin, New Zealand. Website: http://www.aut.ac.nz/resources/research/research_institutes/kedri/downloads/pdf/funn96_20.pdf

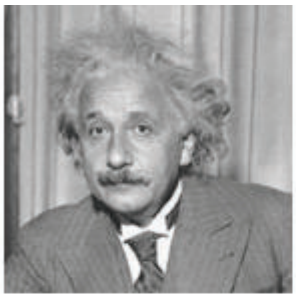
Koivo, H (2001) *Soft Computing In Dynamical Systems*. PhD thesis, Helsinki University of Technology.

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

Qin, H (2005) *Nonlinear Adaptive Noise Cancellation for 2-D Signals with Neuro-Fuzzy Inference System*. M.Sc. thesis, Guelph University.

Ref. No. (2489)
Rec. 29/05/2008
In-revised form: 3/10/2009

Appendix



A. Albert (195×195)

F. Boats (113×111×3)

B. Bridge (195×195)

G. Lily (230×186×3)

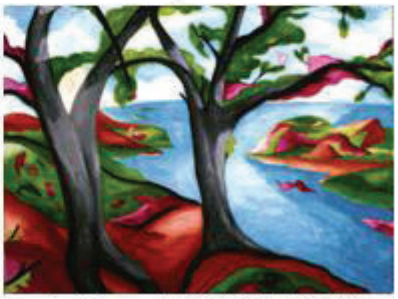
C. Eight (308 × 242)

H. Pepper (225 × 221 × 3)

D. Lena (189 × 189)

I. Shawl (193 × 193 ×

E. Sail (195 × 195)



J. Trees (262 × 193 × 3)

Original copies of the testing images set
(A-E): The grayscale images.
(F-J): The truecolor images.