World Applied Sciences Journal 17 (2): 218-227, 2012
ISSN 1818-4952

Artificial Neural Networks Evaluation as an Image Denoising Tool

Yazeed A. Al-Sbou

Department of Computer Engineering, Faculty of Engineering,
Mu'tah University, P.O. Box (7), Al-Karak, Postal Code 61710, Jordan

Abstract: Image denoising is a challenging task in the digital image processing research and application. This makes it imperative to find a robust method to comply that task. In this paper, a detailed performance evaluation of using the neural networks as a noise reduction tool is presented. The proposed approach includes using both mean and median statistical functions for calculating the output pixels of the training pattern of the neural network. This uses part of the degraded image pixel to generate the system training pattern. Different test images, noise levels and neighborhoods sizes are used. Based on using samples of degraded pixel neighborhoods as inputs, the output of proposed approach provided a good image denoising performance, which exhibited promising qualitative and quantitative results of the degraded noisy images in terms of PSNR, MSE and visual tests.

Key words: Image denoising · Neural networks · Pixel neighborhoods · Noise variance · PSNR · MSE

INTRODUCTION

Images are usually produced in order to display or record useful and important information. Unfortunately, these images may fail to represent the original required scene due to some imperfections added by poor image sensors, imperfect instruments and problems with data acquisition process, or transmission errors which will lead to a corrupted or degraded image. Therefore, it is common that images are contaminated by some noise due to numerous unavoidable reasons. Due to this, it is necessary to detect and remove the added noises present in the image taking into account preserving the image details while removing theses noises. This represents the main goal of image denoising approaches [1].

Image denoising is still a challenging problem in image processing [2]. A variety of denoising methods have been developed to remove the noise and recover the true image. Some may be implemented directly in the time domain and the rest in the other transform domains such as frequency and wavelet domains. Even though they may be very different in tools it must be emphasized that all of them share the same basic remark: denoising is achieved by averaging [3]. This averaging may be performed locally in time domain or in frequency domain where repeated structures in an image were averaged to reduce the (random) noise [4]. As examples on the approaches which use the averaging principle in time domain are: the Gaussian smoothing model [5], the anisotropic filtering [6, 7] and the neighborhood filtering [8] by the calculus of variations like the total variation minimization [9]. Moreover, examples on averaging in the frequency domain can be: the empirical Wiener filters [8] and wavelet thresholding methods [10, 11].

There has been a fair amount of research on wavelet thresholding and threshold selection for signal denoising because wavelet provides an appropriate basis for separating noisy signal from the image signal [12, 13]. The motivation of using the wavelet transform in image denoising is that the small coefficient are more likely due to noise and large coefficient due to important signal features, therefore, these small coefficients can be thresholded without affecting the significant features of the image [14]. The adaptive Wiener filter [15] is one of the most well-known denoising methods. It is an optimal estimator based on the mean squared error (MSE) and requires the estimation of the additive noise variance, which may be determined from the local variance calculated over a uniform moving average window in the degraded image [16]. However, still these approaches have problems on a heavy noisy network [17]. Additionally, wavelet based approaches are computationally expensive [18].

Corresponding Author: Yazeed A. Al-Sbou, Department of Computer Engineering, Faculty of Engineering, Mu'tah University, P.O. Box (7), Al-Karak, Postal Code 61710, Jordan.
Tel. +962 799 613 566.

Recently, some of the methods for image denoising have attempted to improve their solutions and reduce the computational complexity. To overcome limitations in image denoising techniques, few researchers have introduced intelligent techniques to image denoising. These intelligent approaches exhibit promising results for natural and non-natural (document) images [2, 17, 19].

Due to their wide use as tools for data processing, Artificial Neural Networks (ANNs) have been recently introduced and used as a new model to the image denoising problems [20, 21]. This type of intelligent solution presents some attributes that may produce better results in the image restoration process [22, 23]. These attributes are related to the flexibility and their parallel computing properties that have made them suitable for applications in pattern recognition, signal processing, image processing, computer vision and several other application areas.

The focus of this research is to denoise a source image affected by additive white Gaussian noise. However, it is a valid assumption for images obtained through transmitting, scanning or compression. In this paper, a new image denoising technique by using feed forward neural network is proposed for images with 256 x 256 pixels. The training patterns are generated from the noisy image by using the pixel values of different local neighbors. The neural network uses a Multilayer Perceptron (MLP) algorithm and trained with the back propagation training algorithm [24, 25]. The values of each pixel and its neighbors are passed through the trained neural network. The resulted denoised image is obtained from the output values of the neural network. This design is tested using several benchmark images including-Lena, Cameraman and House. It is observed that the new approach is capable to produce a denoised image from the noisy one.

The Proposed Approach: ANN is a mathematical or computational model that attempts to simulate the functional features of biological neural networks. These networks consist of an interconnected group of artificial neurons to process input information. ANN can be an adaptive system that changes its construction based on external or internal information that flows through the network during the learning phase. ANNs are usually used to model complex relationships between inputs and outputs which can be identified mathematically.

ANNs are popular due to their learning capabilities. Learning algorithms seek through the solution space in order to find a cost function that has the smallest possible measure of how far away from an optimal solution to the problem that we want to solve. In this research, a Back Propagation Neural Network (BPN) is used as the learning algorithm [25]. BPN follows supervised learning. Consider a neural network with n input and m output units with any number of hidden units as shown in Figure 1. When the input pattern x_i from a training set is presented to the network; it produces an output y_i different in general from the target t_i. y_i and t_i should be identical for $i = 1,..., p$, by using the back propagation learning algorithm. The error between y_i and t_i should be minimized as follows:

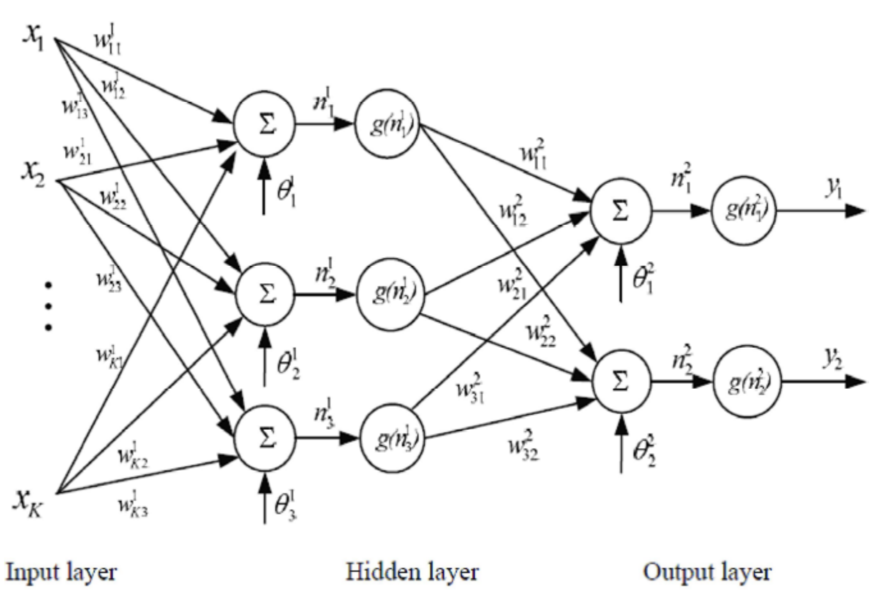

Fig. 1: Multilayer perceptron neural network architecture

$$e = \frac{1}{2}\sum_{i=1}^{p}\|y_i - t_i\|^2$$

Firstly, the NN starts to compute activations and signals of input, hidden and output neurons for the x_i. Then the error over the output neurons is calculated using equation (1). This error is used to compute the change in the hidden to output layer weights and the change in input to hidden layer weights (including all bias weights), such that a global error measure gets reduced. This procedure is repeated until the global error falls below a predefined threshold.

In this work, MLPs is used which are feed-forward neural networks [27]. MLPs are supervised networks so they need to be trained. They learn how to transform the input data into a desired output. Using some (one or two) hidden layers, MLP can approximate the input-output relationship. Most neural network applications involve MLPs as shown in Figure 1. This network has an input layer with k neurons, one hidden layer with three neurons and an output layer with two neurons. The input layer has vector of input variable values (x_1 ..... x_k). The input layer distributes the values to each of the neurons in the hidden layer. In the hidden layer, the value from each input neuron is multiplied by a weight (w_{ij}) and the resulting weighted values are added together producing a combined value n_i. The weighted sum n_i is fed into a transfer function, g. The outputs from the hidden layer are distributed to the output layer. In the output layer, the value from each hidden layer neuron is multiplied by another weight (w_{kj}) and the resulting weighted sum is fed into a transfer function, g, which outputs a value y_k. The y values are the outputs of the network.

In the conducted experiments, in order to design the training set for the neural network, the artificially degraded images are simulated by applying the degradation model. The image is first convoluted with a selected noise operation like Gaussian white noise and salt and pepper noise and then noise is added to it at different rate occurrence. A corrupted image can be given as:

$$y(i, j) = x(i, j) + n(i, j) \tag{2}$$

Where $y(i,j)$ is the observed value, $x(i,j)$ is the true (original) value and $n(i,j)$ is the noise perturbation at pixel (i,j). The learning phase of the MLP attempts to make it capture inherent space relations of degraded pixels and correspond them to the non-degraded pixels. The degraded image data is provided as input to the MLP and the non-degraded image as the corresponding output in the supervised learning process.

The noisy image has the dimension of N×M. In this work, image processing operations involve processing an image in sections, called blocks or neighborhoods, with sliding neighborhood operation rather than processing the entire image at once. A sliding neighborhood operation is an operation that is performed a pixel at a time, with the value of any given pixel in the output image being determined by the application of an algorithm to the values of the corresponding input pixel's neighborhood. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel, which is called the center pixel. The neighborhood is a rectangular block and as we move from one element to the next in an image matrix, the neighborhood block slides in the same direction. The center pixel is the actual pixel in the input image being processed by the operation. If the neighborhood has an odd number of rows and columns, the center pixel is actually in the center of the neighborhood. If one of the dimensions has even length, the center pixel is just to the left of center or just above center. For example, in a 2-by-2 neighborhood, the center pixel is the upper left one. To summaries how a sliding neighborhood operation is performed: Select a single pixel, determine the pixel's neighborhood, apply a function to the values of the pixels in the neighborhood. This function must return a scalar; find the pixel in the output image whose position corresponds to that of the center pixel in the input image then set this output pixel to the value returned by the function. These steps are repeated for each pixel in the input image.

In order to obtain the output pixel, it must be previously determined how this pixel is connected with its neighbors in the noisy image. Connectivity defines which pixels are connected to other pixels. A set of pixels in a binary image that form a connected group is called an object or a connected component. There are two types of pixels connectivity: 4-connected and 8-connected. In 4-connected, pixels are connected if their edges touch. This means that a pair of adjoining pixels is part of the same object only if they are both on and are connected along the horizontal or vertical direction. While in 8-connected, pixels are connected if their edges or corners touch. This means that if two adjoining pixels are on, they are part of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.

The training patterns set of the MLP are created by sequentially extracting 2x2, 3x3, 4x4 and 5x5 local neighborhoods from the degraded image and the pixel of the corresponding neighborhood center position in the original non degraded image. Here, the functions which are applied to values of the pixels in the neighborhood are averaging and median operations of pixels in the neighborhood. The result of this calculation is the value of the output pixel. During the training phase, the pixel values in each neighborhood are used as inputs to the MLP and the target is the output pixel (the result of the averaging or median operation). The process is repeated for all the pixels of the noisy image. Finally, a denoised image is generated that has the dimension of the noisy image.

In this work, the noisy images are generated from the original benchmark images by adding Gaussian white noise of different standard deviations (σ). The pixel values of the noisy image are normalized by dividing with 255 in order to accelerate the convergence of the neural network. The training patterns are generated as described above. The MLP was designed with 5-5-1 neural network (5 input neurons, 5 hidden neurons and 1 output neuron) is taken and the values of the weights are initially randomized in the interval (-0.1, +0.1). The NN is trained 5000 times (iterations) with the training patterns. The learning rate is set at 0.05 during the training phase. The Hyperbolic tangent sigmoid transfer function activation function is used in the hidden and output neurons. The denoised image is generated by applying the pixel values of the noisy image to the trained NN. The outputs of the NN are multiplied by 255 to form the denoised image.

Peak to Signal Noise Ratio (PSNR) and MSE are standard criteria reported in the literature for quantitative evaluation of the effectiveness of proposed image denoising methods. PSNR and MSE are metrics by which they measure the absolute difference between two signals, which are completely quantifiable. PSNR is the ratio between the reference signal and the distorted signal in an image, given in decibels. The higher the PSNR, the closer the distorted image is to the original. In general, a higher PSNR value should correlate to a higher quality image, but this is not always the case. MSE is the average squared difference between a reference image and a distorted image. It is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count. Both MSE and PSNR are very important in image and video quality monitoring and are computed as below,

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \tag{3}$$

$$PSNR = 10.\log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{4}$$

Where I and K are the original and the distorted images, respectively. m and n are the number of pixels in both images (dimensions of the images) and MAX equal to the maximum possible pixel value (in our case it is 2^8 - 1 = 255 for 8-bit images).

RESULTS AND DISCUSSION

The proposed procedure requires no prior knowledge of the noise; this represents a good advantage of the proposed approach compared to many of the current techniques that assume the noise model to be known like the Gaussian noise. That due to the fact that, in reality, this assumption may not always hold true due to the varied nature and sources of noise. The experimental results presented here are based on adding Gaussian noise with zero mean and different variance values to the natural images to demonstrate the performance of the proposed algorithm.

This section illustrates quantitative results of application of the neural network system as an image denoising methodology. The proposed technique was implemented using MATLAB 8.0. The obtained results show the PSNR and MSE of Cameraman, Lena and House image benchmarks for the noisy and denoised images with different noise levels. Figure 2 shows the original images of these image benchmarks. Figures 3-8 depict the 2x2, 3x3, 4x4 and 5x5 local neighborhoods PSNR and MSE using the mean and median functions to calculate the output pixels which compose the training pattern of the MLP with different noise variances.

Figure 3 illustrates the PSNR and MSE of the Cameraman image based on the mean function. From this figure, it can be noticed that as the local neighborhood size increases, the PSNR increases and the MSE decreases. This means that the 5x5 has the highest PSNR and the lowest MSE compared to the noisy, 2x2, 3x3 and 4x4 images PSNR and MSE, respectively. Also, the PSNR decreases as the added noise level increases while the MSE increases as the added noise increases for the 2x2, 3x3, 4x4 and 5x5 local neighborhoods. That is because as the noise which corrupts the original image increases, the capability of reconstructing the original one becomes

(a) (b) (c)

Fig. 2: The original images benchmark: (a) Cameraman, (b) Lena and (c) House.

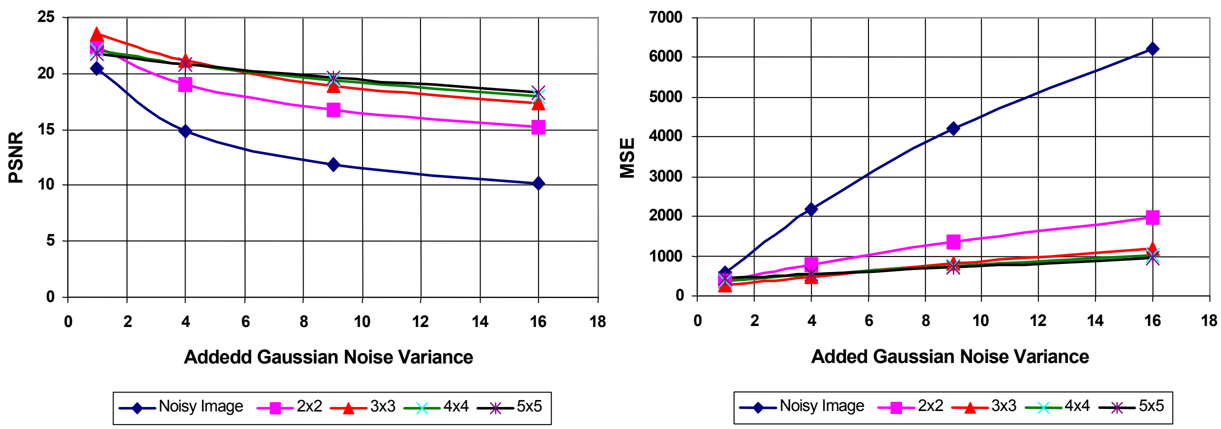

Fig. 3: The PSNR and MSE of the noisy and denoised Cameraman image with different noise levels and pixel neighborhood sizes based on the mean function.

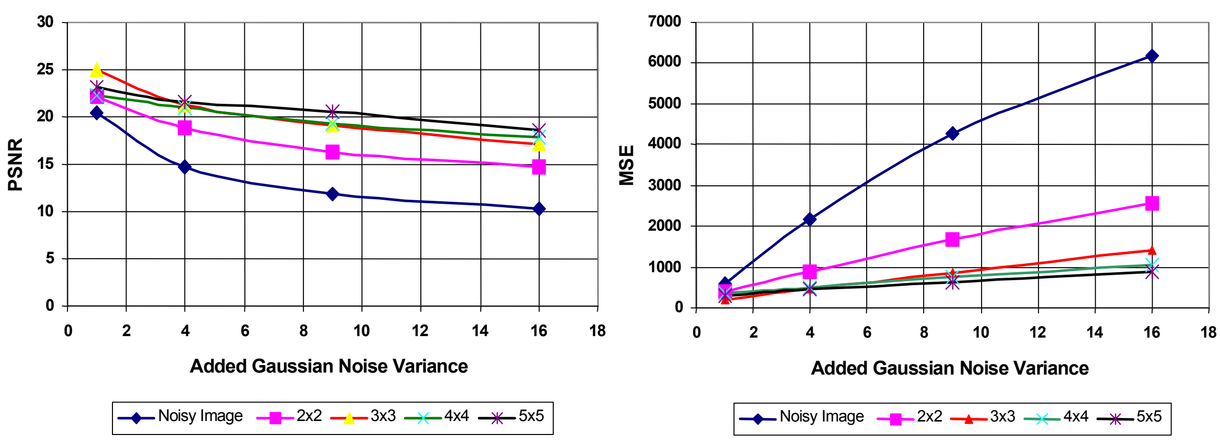

Fig. 4: The PSNR and MSE of the noisy and denoised Cameraman image with different noise levels and pixel neighborhood sizes based on the median function.

harder. A comparison has been made between the mean and median as functions to generate the denoised pattern sample for the MLP. For the Cameraman, the obtained results from both mean and median functions are shown in Figure 3 and Figure 4 for PSNR and MSE, respectively. These figures demonstrate that there are no clear differences between using mean or median functions to get the denoised image. This principle was applied to Lena and House test images in order to get more insight on the performance of the proposed denoising method. The results obtained from applying this method on these images are shown in Figures 5-8. These figures also reveal

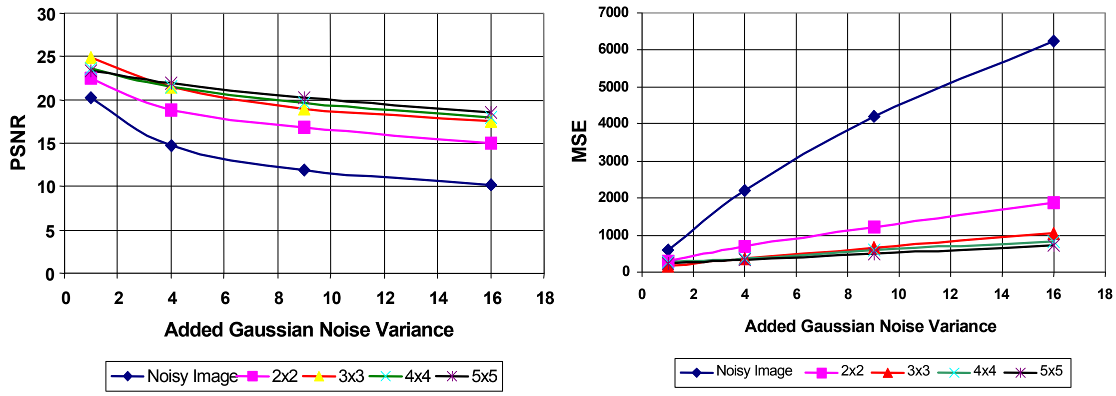

Fig. 5: The PSNR and MSE of the noisy and denoised Lena image with different noise levels and pixel neighborhood sizes based on the mean function

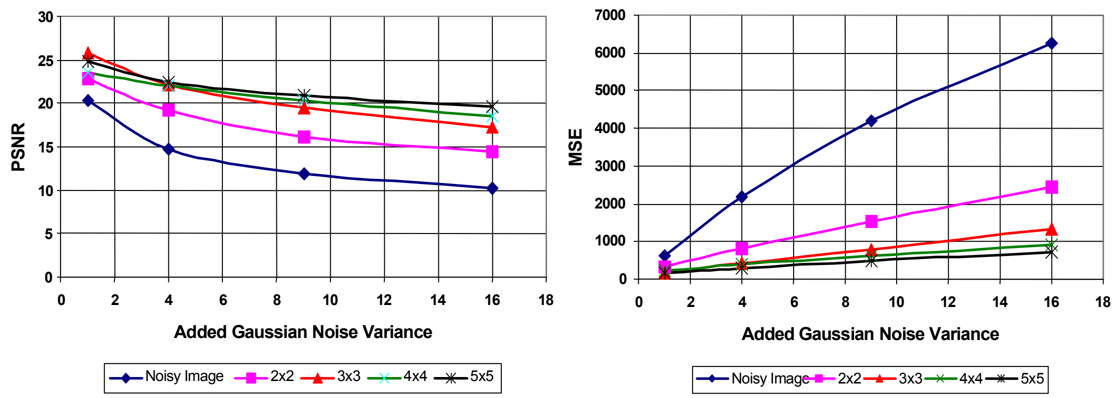

Fig. 6: The PSNR and MSE of the noisy and denoised Lena image with different noise levels and pixel neighborhood sizes based on the median function.

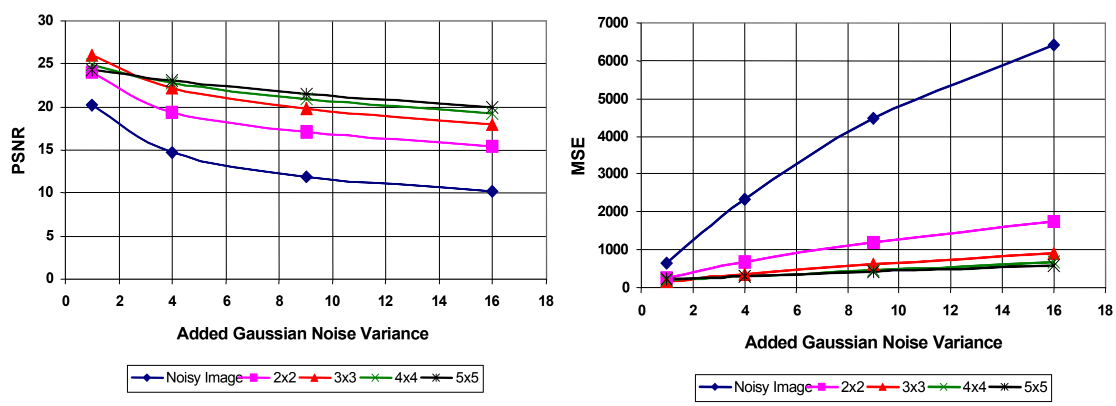

Fig. 7: The PSNR and MSE of the noisy and denoised House image with different noise levels and pixel neighborhood sizes based on the mean function.

that both mean and median functions provide nearly the same outputs when were used to generate the training pattern of the MLP.

A more detailed presentation about the effectiveness of the proposed technique can be obtained from calculating the difference between the both PNSR and MSE of the original image and the degraded version and the original and denoised images. In addition, a comparison between the mean and median functions can be accomplished. These results are summarized in Tables 1 and 2. As an example and due to limited space, these tables present the Cameraman image results only.

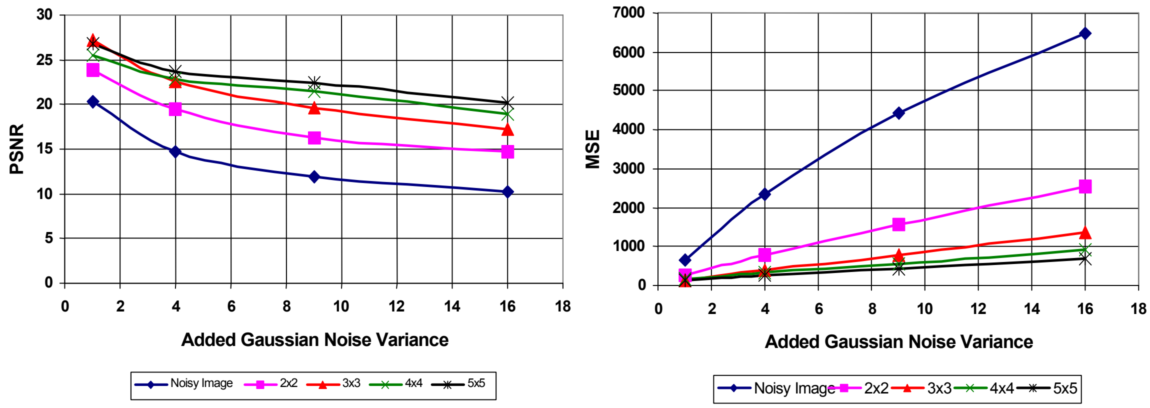

Fig. 8: The PSNR and MSE of the noisy and denoised House image with different noise levels and pixel neighborhood sizes based on the median function.

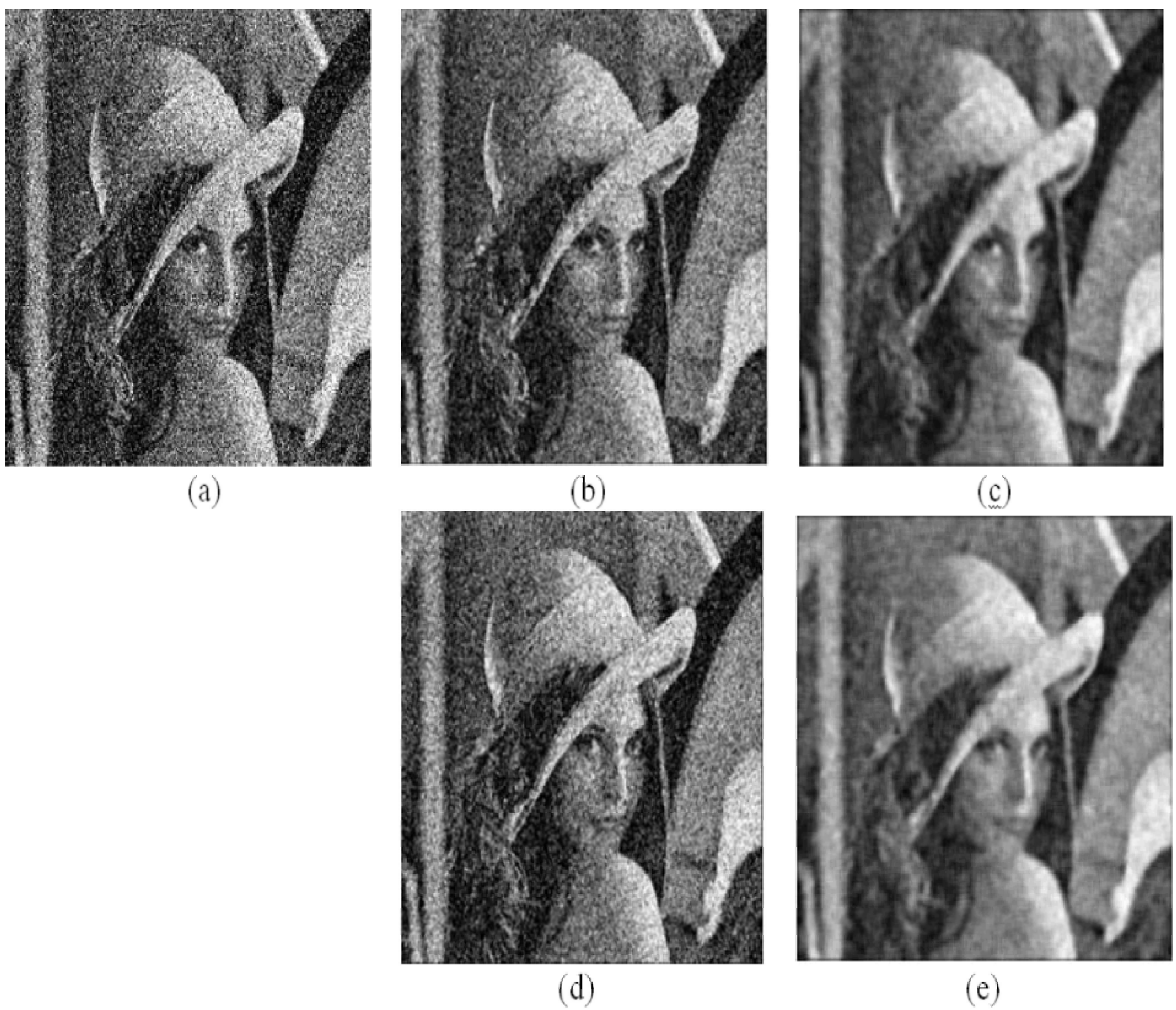

Fig. 9: (a) Lena degraded (noisy) image (noise variance = 4), (b) and (c) Denoised image based on the mean function using 2x2 and 5x5 neighborhood, respectively. (d) and (e) Denoised image based on the median function using 2x2 and 5x5 neighborhood, respectively.

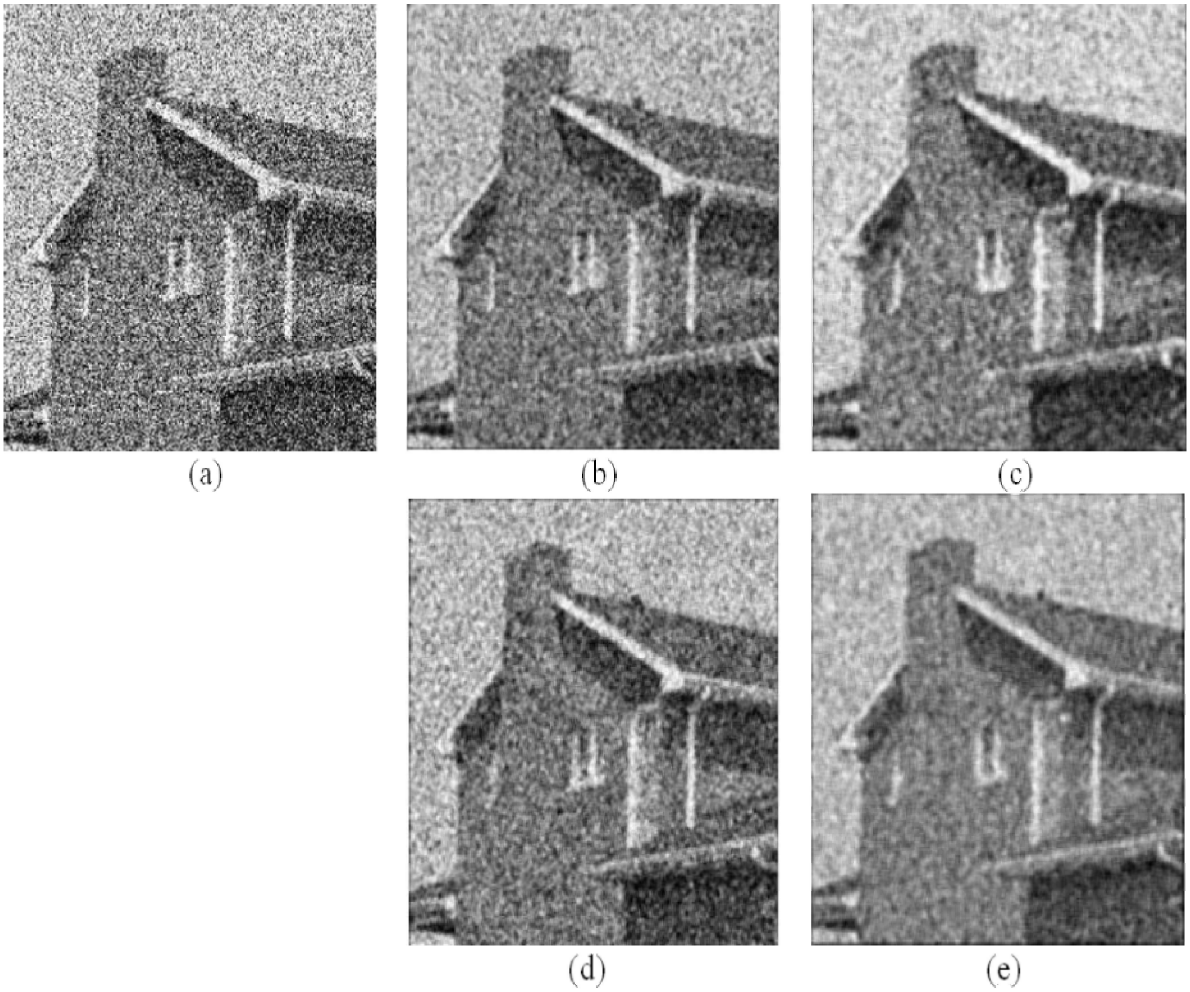

Fig. 10: (a) House degraded (noisy) image (noise variance = 9), (b) and (c) Denoised image based on the mean function using 2x2 and 5x5 neighborhood, respectively. (d) and (e) Denoised image based on the median function using 2x2 and 5x5 neighborhood, respectively.

From Table 1, it can be noticed that as the noise level increases as the error (i.e., difference) in the PSNR increases which means the more improvement has been made to the degraded image for both the mean and median functions. On the other hand, this is not the case for the neighborhood size. It is clear that the neighborhood size does not significantly affect the especially the 3x3, 4x4 and 5x5 neighborhood sizes. That is because as the neighborhood size increases as the amount of available information to predict the original pixel are more. This justification is true mathematically and quantitatively (objectively) but not always true qualitatively (i.e. subjectively) as will be seen later in this paper. These observations are also true for the MSE as can be seen from Table 2.

Another test for the proposed procedure is to perform a subjective evaluation of the resulted denoised images. Figures 9 and 10 show degraded images with different noise levels and the denoised images using the proposed approach based on the mean and median functions using the 2x2, 3x3, 4x4 and 5x5 local neighborhoods. The obtained images showed that the NN approach led to a good restoration of the degraded images with different noise reduction levels. In both mean and median functions the 2x2 and 3x3 neighborhoods provide little noise reduction. On the other hand, the 4x4 and 5x5 provide better noise reduction but with some blurs. Therefore, it can be seen that as the size of the local neighborhoods increases as the restored (denoised) image becomes more blurred as shown in Figures 9 and 10 for Lena and House images.

Table 1: The absolute error between the PNSR of the original image and the degraded version and the original and denoised images of the Cameraman image using the: (a) Mean and (b) Median functions

(a)	Pixel neighborhood size				
	2x2	3x3	4x4	5x5	
Noise var. (%)	1	2.01	3.16	1.78	1.31
	4	4.27	6.38	6.04	6.02
	9	4.85	7.02	7.47	7.70
	16	4.93	7.13	7.76	8.15
(b)	Pixel neighborhood size				
	2x2	3x3	4x4	5x5	
Noise var. (%)	1	1.78	4.64	1.92	2.74
	4	4.14	6.58	6.26	6.78
	9	4.43	7.33	7.43	8.78
	16	4.48	6.91	7.66	8.30

Table 2: The absolute error between the MSE of the original image and the degraded version and the original and denoised images of the Cameraman image using the: (a) Mean and (b) Median functions

(a)	Pixel neighborhood size				
	2x2	3x3	4x4	5x5	
Noise var. (%)	1	225.6	311.1	205.0	161.9
	4	1365.1	1670.8	1631.2	1628.9
	9	2867.7	3401.4	3483.4	3521.0
	16	4247.8	5028.6	5186.6	5275.8
(b)	Pixel neighborhood size				
	2x2	3x3	4x4	5x5	
Noise var. (%)	1	207.1	394.5	219.1	283.9
	4	1285.6	1711.1	1674.9	1732.2
	9	2598.1	3406.2	3514.4	3646.4
	16	3631.4	4774.9	5121.3	5288.6

CONCLUSIONS

In this paper, an evaluation of using the neural networks as a tool for image denoising was performed. Both MLP and BPN were used. The evaluation also included both mean and median functions to be used as a function to return a pixel which corresponds to a pixel in the output image whose position corresponds to that of the center pixel in the input image neighborhood and with a value returned by the function (i.e., the mean or median of the neighborhood pixels of the center pixel). In this work, different test images, noise levels and neighborhoods sizes were used. The evaluation was based on the PSNR, MSE and subjective (visual) methods. From the obtained experimental results, the proposed approach exhibited outcomes of noise reduction and image quality improvements, with different noise levels, which qualify it to be suitable for image processing and denoising. In addition, both mean and median functions provided comparable results for different neighborhood pixel sizes. Nevertheless, as the local neighborhoods size increases as the resulted denoised image becomes blurred. Moreover, as the local neighborhoods size increases as the PSNR of the denoised image becomes better. This concludes that, the larger the neighborhoods size, the better the PSNR but the less image details preserved.

REFERENCES

1. Motwani, M.C., M.C. Gadiya and R.C. Motwani, 2004. Survey of Image Denoising Techniques. In the Proceedings of the 2004 GSPx Conference, Santa Clara, CA.
2. Saba, T., A. Rehman and G. Sulong, 2010. An Intelligent Approach to Image Denoising. Journal of Theoretical and Applied Information Technology.
3. Buades, A., B. Coll and M. Morel, 2005. Image Denoising by Non-local Averaging. In the Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing, pp: 25-28.
4. Mahmoudi, M. and G. Sapiro, 2005. Fast Image and Video Denoising via Non-local Means of Similar Neighborhoods. IEEE Signal Processing Letters, 12(12).
5. Lindenbaum, M., M. Fischer and A. Bruckstein, 1994.. On Gabor Contribution to Image-Enhancement. Pattern Recognition, 27: 1-8.
6. Perona, P. and J. Malik, 1990. Scale Space and Edge Detection Using Anisotropic Diffusion. IEEE Trans. Patt. Anal. Mach. Intell., 12: 629-639.
7. Alvarez, L., P.L. Lions and J.M. Morel, 1992. Image Selective Smoothing And Edge Detection By Nonlinear Diffusion (II). SIAM Journal of Numerical Analysis, 29: 845-866.
8. Yaroslavsky, L.P. and M. Eden, 1996. Fundamentals of Digital Optics. Birkhauser.
9. Rudin, L. and S. Osher, 1992. Nonlinear Total Variation Based Noise Removal Algorithms. Physica D, 60: 259-268.
10. Coifman, R.R. and D. Donoho, 1995. Translation-invariant De-noising. Wavelets and Statistics, pp: 125-150.

11. Kazubek, M., 2003. Wavelet Domain Image Denoising by Thresholding and Wiener Filtering. IEEE Signal Processing Letters, 10: 324-326.
12. Grace Chang, S., B. Yu and M. Vattereli, 2000. Adaptive Wavelet Thresholding For Image Denoising And Compression. IEEE Transactions of Image Processing, 9: 1532-1546.
13. Jansen, M., 2001. Noise Reduction by Wavelet Thresholding. Springer Verlag, New York Inc.
14. Kaur, L., S. Gupta and R.C. Chauhan, 2002. Image Denoising Using Wavelet Thresholding. In the Proceedings of the 2002 Indian Conference on Computer Vision, Graphics and Image Processing.
15. Kumar, S., P.M. Gupta and A.K. Nagawat, 2010. Performance Comparison of Median and Wiener Filter in Image De-Noising. International Journal of Computer Applications, 12.
16. Jin, F., P.W. Fieguth, L. Winger and E. Jernigan, 2003. Adaptive Wiener Filtering of Noisy Images and Image Sequences. In the Proceedings of the 2003 International Conference on Image Processing.
17. Dalong, L., S. Simske and M. Mersereau, 2007. Image Denoising Through Support Vector Regression. In the Proceedings of the 2007 IEEE International Conference on Image Processing, 4: 425-428.
18. Buades, A., B. Coll and J. Morel, 2008. Nonlocal Image and Movie Denoising. International Journal of Computer Vision, 76: 123-139.
19. Kulkarni, A.D., 2001. Computer Vision and Fuzzy-Neural Systems. Prentice Hall.
20. Cheema, T.A., I. Qureshi and M. Naveed A. 2007. Blur and Image Restoration of Nonlinearly Degraded Images Using Neural Networks Based on Modified Nonlinear ARMA Model. The Arabian Journal for Science and Engineering, 32: 67-83.
21. Ahmed, S., M. Shahjahan and K. Md.- Murase, 2010. An image Denoising Technique Using Feedforward Neural Network. In the Proceedings of the 2010 International Symposium on Intelligent Systems (iFAN 2010), pp: 25-26.
22. Wu, Y.D., Q.Z. Zhu, S.X. Sun and H.Y. Zhang, 2006. Image Restoration Using Variational PDE-Based Neural Network. Journal of Neurocomputing, 69: 2364-2368.
23. Yan, L., L. Wang and K.H. Yap, 2004. A Noisy Chaotic Neural Network Approach To Image Denoising. In the Proceedings of the 2004 International Conference on Image Processing (ICIP).
24. Haykin, S. and R. Neurais, 2001. Princípios e Prática. Second Edition, Porto Alegre, Bookman.
25. Yu, C.C. and B.D. Liu, 2002. A Backpropagation Algorithm With Adaptive Learning Rate And Momentum Coefficient. In the Proceedings of the 2002 IEEE World Congress on Computational Intelligence, pp: 1218-1223.