Feed Forward Back Propagation Algorithm For Eliminating Gaussian Noise And Impulse Noise

R. Pushpavalli[1] and G. Sivarajde[2]
Research scholar[1] and Professor[2]
Department of Electronics and Communication Engineering
Pondicherry Engineering College, Puducherry-605 014, India.

Abstract

Digital Images are contaminated by noise during acquisition and/or transmission over communication channel. Eliminating impulse noise and gaussian noise from the images without damaging their boundaries and fine details is an important and a challenging task in the image processing applications. A nonlinear technique based on decision mechanism for suppressing impulse noise and gaussian noise from digital images is proposed in this paper. The proposed algorithm, called, Feed Forward Back Propagation (FFBP) algorithm performs quite well in the presence of multiple noise while preserving the image features satisfactorily. The proposed intelligent algorithm is carried out in two stages. In first stage the corrupted image is filtered by applying a decision based nonlinear filter. This decision based nonlinear filtered output image and noise images are suitably combined with a feed forward neural network in the second stage. The parameters of the feed forward neural network are adaptively optimized by training of three well known images. This is quite efficient in eliminating uniform noise and impulse noise. Simulation results show that the proposed filter is superior in terms of eliminating impulse noise as well as preserving edges and fine details of digital images and results are compared with other existing nonlinear filters. The performance of the filtering technique has been evaluated by applying it on several test images corrupted by different levels of impulse noise and the results obtained are presented and compared with that of the existing filtering techniques.

1. Introduction

Image noise is random variation of brightness or colour information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information [1-3]. The magnitude of image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radioastronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing (a noise level that would be totally unacceptable in a photograph since it would be impossible to determine even what the subject was). Fat-tail distributed or "impulsive" noise is sometimes called salt-and-pepper noise or spike noise. An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc. It can be mostly eliminated by using dark frame subtraction and interpolating around dark/bright pixels. Dead pixels in an LCD monitor produce a similar, but non-random, display. The noise caused by quantizing the pixels of a sensed image to a number of discrete levels is known as quantization noise or uniform noise. It has an approximately uniform distribution. Though it can be signal dependent, it will be signal independent if other noise sources are big enough to cause dithering, or if dithering is explicitly applied. In this paper, elimination of gaussian noise and impulse noise on digital images is proposed.

These techniques aim to achieve optimal performance over the entire image. A good noise filter is required to satisfy two criteria, namely, suppressing the noise and preserving the useful information in the signal. Unfortunately, a great majority of currently available noise filters cannot simultaneously satisfy both of these criteria. The existing filters either suppress the noise at the cost of reduced noise suppression performance [4-34]. In order to address these issues, many neural networks have been investigated for denoising gaussian noise and impulse noise on digital images.

The feed forward back-propagation learning algorithm is suitable one to implement and computationally efficient in which its complexity is linear in the synaptic weights of the neural network [35-40]. Back propagation is a common method of training artificial neural networks algorithm so as to minimize the objective function. It is a multi-stage dynamic system optimization method. The input-output relation of a feed forward adaptive neural network can be viewed as a powerful nonlinear mapping. Conceptually, a feed forward adaptive network is actually a static mapping between its input and output spaces. Even though, intelligent techniques required certain pattern of data to learn the input. This filtered image data pattern is given through nonlinear filter for training of the input. Therefore, intelligent filter performance depends on conventional filters performance. This work aims to achieving good de-noising without compromising on the useful information of the signal.

In this paper, a novel structure is proposed to eliminate the impulse noise and gaussian noise and also preserves the edges and fine details of digital images; a feed forward neural architecture with back propagation learning algorithm is used and is referred as feed forward back propagation algorithm for restoring digital images. The proposed filtering operation is passed out in two stages. In first stage the corrupted image is filtered by applying a decision based nonlinear filtering technique. This filtered image output data sequence and noisy image data sequence are suitably combined with a feed forward neural (FFN) network in the second stage. The internal parameters of the feed forward neural network are adaptively optimized by training of the neural network with feed forward back propagation algorithm.

The rest of the paper is organized as follows. Section 2 explains the structure of the proposed filter and its building blocks. Section 3 discusses the results of the proposed filter on different test images. Conclusion is presented in section 4.

2. Proposed filter

A feed forward neural network is a flexible system trained by heuristic learning techniques derived from neural networks can be viewed as a 3-layer neural network with weights and activation functions.

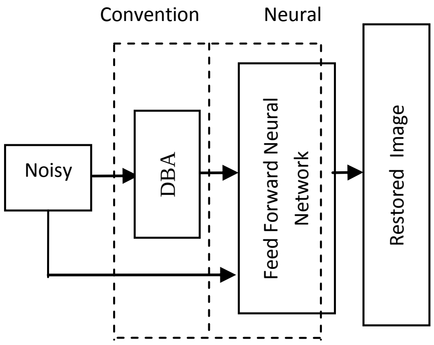

Fig. 1 Block diagram of proposed filter

Fig. 1 shows the structure of the proposed impulse noise removal filter. The proposed filter is obtained by appropriately combining output image from new decision based nonlinear filter with neural network. Learning and understanding aptitude of neural network congregate information from decision based nonlinear filter to compute output of the system which is equal to the restored value of noisy input pixel.

The neural network learning procedure is used for the input-output mapping which is based on learning the proposed filter and the neural network utilizes back propagation algorithm. The special class of filter is described in section 2.1.

2. Decision Based Algorithm

The filtering technique proposed in this paper employs a decision mechanism to detect the presence of impulse noise and gaussian noise on the test image. The pixels inside the sliding window are identified as corrupted or not. If the pixel is corrupted, based on the type of noise, the corrupted central pixel is replaced by either median filter or nonlinear mean filter. Median filter is defined as

$$MF = [med\{F(i,j)\}], \quad (i,j) \in S_{mn}(s,t) \in S_{mn} \qquad (2.1.1)$$

where, *MF* represents median filter, *F(i,j)* represents processing pixel, S_{mn} represents the filtering window. Mean filter simply computes the average of pixels within the filtering window. So mean filter is defined as

$$MF_{avg} = [mean\{F(i,j)\}], \quad (i,j) \in S_{mn}(s,t) \in S_{mn} \qquad (2.1.2)$$

where, MF_{avg} represents midpoint filter, *F(i,j)* represents processing pixel, S_{mn} represents the filtering window. This filter is a combination of statistics and mean filter. It is very useful for enhancement if image is corrupted with impulse noise and gaussian noise. Pixel inside the window is separated as impulse noise and remaining pixels. The remaining pixels (without impulse noise) inside the filtering window are arranged in ascending order and average value is calculated for filtering.

Consider an image of size M×N having 8-bit gray scale pixel resolution. The proposed filtering algorithm as applied on noisy image is described in steps as follows:

Step 1) A two-dimensional square filtering window of size 3 x 3 is slid over the noisy image.

Step 2) As the window move over the noisy image, at each point the central pixel inside the window is checked whether the pixel is corrupted by impulse noise or not.

Step 3) If it is corrupted by impulse noise, then the central pixel is replaced by median pixel value.

Step 3) If the central processing pixel is not corrupted by impulse noise, the pixels within the filtering window are sorted out excluding impulse noise and then nonlinear mean filter is performed on sorted pixels within the moving window.

Then the window is moved to form a new set of values, with the next pixel to be processed at the centre of the window. This process is repeated until the last image pixel is processed. It may be noted that the filtering is performed by either taking the median or the mean value of the pixels of the filtering window. Moreover, the mean filtering on the remaining pixels (without impulse noise) sample is performed only on processing pixels. As a result, the pixels in the filtered image do not cause any noticeable visual degradation. The performance of the proposed filter is superior to other existing filters in terms of eliminating multiple noise and preserving edges and features of images.

2.2 Feed Forward Neural Network

In feed forward neural network, back propagation algorithm is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in dynamic nonlinear systems. This network is popular general nonlinear modeling tool because it is very suitable for tuning by optimization and one to one mapping between input and output data. The input-output relationship of the network is as shown in Fig.2. In Fig.2 x_m represents the total number of input image pixels as data, n_{kl} represents the number of neurons in the hidden unit, k represents the number hidden layer and l represents the number of neurons in each hidden layer. A feed forward back propagation neural network consists of three layers.

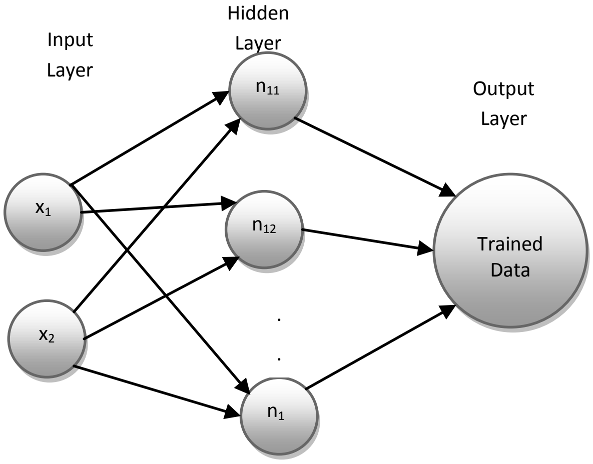

Fig.2 Feed Forward Neural Network Architecture

The first layer is referred as input layer and the second layer is represents the hidden layer, has a tan sigmoid (tan-sig) activation function is represented by

$$\phi(yi)=\tanh(vi) \qquad (2.2.1)$$

This function is a hyperbolic tangent which ranges from -1 to 1, y_i is the output of the *i*th node (neuron) and v_i is the weighted sum of the input and the second layer or output layer, has a linear activation function. Thus, the first layer limits the output to a narrow range, from which the linear layer can produce all values. The output of each layer can be represented by

$$Y_{Nx1} = f(W_{NxM} X_{M,1} + b_{N,1}) \qquad (2.2.2)$$

where Y is a vector containing the output from each of the N neurons in each given layer, W is a matrix containing the weights for each of the M inputs for all N neurons, **X** is a vector containing the inputs, b is a vector containing the biases and f(·) is the activation function for both hidden layer and output layer.

The trained network was created using the neural network toolbox from Matlab9b.0 release. In a back propagation network, there are two steps during training. The back propagation step calculates the error in the gradient descent and propagates it backwards to each neuron in the hidden layer. In the second step, depending upon the values of activation function from hidden layer, the weights and biases are then recomputed, and the output from the activated neurons is then propagated forward from the hidden layer to the output layer. The network is initialized with random weights and biases, and was then trained using the Levenberq-Marquardt algorithm (LM). The weights and biases are updated according to

$$Dn+1 = Dn - [J^T J + \mu I]^{-1} J^T e \qquad (2.2.3)$$

where *Dn* is a matrix containing the current weights and biases, *Dn+1* is a matrix containing the new weights and biases, e is the network error, *J* is a Jacobian matrix containing the first derivative of e with respect to the current weights and biases. In the neural network case, it is a K-by-L matrix, where K is the number of entries in our training set and L is the total number of parameters (weights + biases) of our network. It can be created by taking the partial derivatives of each in respect to each weight, and has the form:

$$J = \begin{bmatrix} \frac{\partial F(x_1, w)}{\partial w_1} \cdots \frac{\partial F(x_1, w)}{\partial w_w} \\ \frac{\partial F(x_1, w)}{\partial w_1} \cdots \frac{\partial F(x_1, w)}{\partial w_w} \end{bmatrix} \qquad (2.2.4)$$

where F(xi,L) is the network function evaluated for the i-th input vector of the training set using the weight vector L and wj is the j-th element of the weight vector L of the network. In traditional Levenberg-Marquardt implementations, the jacobian is approximated by using finite differences, Howe ever, for neural networks; it can be computed very efficiently by using the chain rule of calculus and the first derivatives of the activation functions. For the least-squares problem, the Hessian generally doesn't needs to be calculated. As stated earlier, it can be approximated by using the Jacobian matrix with the formula:

$$H = J^T J \qquad (2.2.5)$$

I is the identity matrix and μ is a variable that increases or decreases based on the performance function. The gradient of the error surface, g, is equal to *JTe*.

2.3 Training of the Feed Forward Neural Network

Feed forward neural network is trained using back propagation algorithm. There are two types of training or learning modes in back propagation algorithm namely sequential mode and batch mode respectively. In sequential learning, a given input pattern is propagated forward and error is determined and back propagated, and the weights are updated. Whereas, in Batch mode learning; weights are updated only after the entire set of training network has been presented to the network. Thus the weight update is only performed after every epoch. It is advantageous to accumulate the weight correction terms for several patterns. Here batch mode learning is used for training.

In addition, neural network recognizes certain pattern of data only and also it entails difficulties to learn logically to identify the error data from the given input image. In order to improve the learning and understanding properties of neural network, noisy image data and filtered output image data are introduced for training. Noisy image data and filtered output data are considered as inputs for neural network training and noise free image is considered as a target image for training of the neural network. Back propagation is pertained as network training principle and the parameters of this network are then iteratively tuned. Once the training of the neural network is completed, its internal parameters are fixed and the network is combined with noisy image data and the nonlinear filter output data to construct the proposed technique, as shown in Fig.3. While training a neural network, network structure is fixed and the unknown images are tested for given fixed network structure respectively. The performance evaluation is obtained through simulation results and shown to be superior performance to other existing filtering techniques in terms of impulse noise elimination and edges and fine detail preservation properties.

The feed forward neural network used in the structure of the proposed filter acts like a *mixture* operator and attempts to construct an enhanced output image by combining the information from the noisy image and decision based algorithm. The rules of mixture are represented by the rules in the rule base of the neural network and the mixture process is implemented by the mechanism of the neural network. The feed forward neural network is trained by using back propagation algorithm and the parameters of the neural network are then iteratively tuned using the Levenberg–Marquardt optimization algorithm, so as to minimize the learning error, *e*. The neural network trained structure is optimized and the tuned parameters are fixed for testing the unknown images. The internal parameters of the neural network are optimized by training. Fig.3 represents the setup used for training and here, based on definition, the parameters of this network are iteratively optimized so that its output converges to original noise free image and completely removes the noise from its input image. The well known images are trained using this neural network and the network structure is optimized. The unknown images are tested using optimized neural network structure.

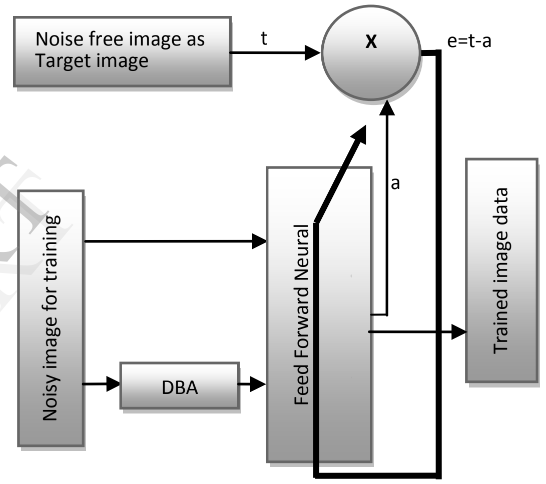

Fig.3 Training of the Feed forward Neural Network

In order to get effective filtering performance, already existing neural network filters are trained with image data and tested using equal noise density. But in practical situation, information about the noise density of the received signal is unpredictable one. Therefore; in this paper, the neural network architecture is trained using denoised three well known images which are corrupted by adding different noise density levels and also the network is trained for different hidden layers with different number of neurons. Impulse noise density with 0.1 and Gaussian noise density with zero mean and σ=200 gave optimum solution for both lower and higher level noise corruption. Therefore images are corrupted with this mentioned level of noise is selected for training. Then the performance error of the given trained data and trained neural network structure are observed for each network. Among these neural network Structures, the trained neural network structure with the minimum error level is selected (10^{-3}) and this trained network structures are fixed for testing the received image signal.

Network is trained for 10 different architectures and corresponding network structure is fixed. PSNR is measured on Lena test image for all architectures with various noise densities. Among these, based on maximum PSNR values; selected architectures is summarized in table 4 for Lena image corrupted with 10% impulse noise and gaussian noise with zero mean and σ=200. Finally, the maximum PSNR value with the neural network architecture of impulse noise density of 0.1 and Gaussian noise density of μ=0 and σ=200 and single hidden layers with 16 neurons has been selected for training. Fig.4 shows the images which are used for training. Three different images are used for network. This noise density level is well suited for testing the different noise level of unknown images in terms of quantitative and qualitative metrics. The image shown in Fig.4 ($a_{1,\ 2\ \text{and}\ 3}$) are the *noise free training image: cameraman, Baboonlion and ship*. The size of an each training image is 256 x 256. The images in Fig.4 ($b_{1,2\ \text{and}\ 3}$) are the *noisy training images* and is obtained by corrupting the noise free training image by impulse noise of 10% and gaussian noise with the density of μ=0 and σ=200. The image in Fig.4 ($c_{1,2\ \text{and}\ 3}$) are the trained images by neural network. The images in Fig.4 (b) and (a) are employed as the *input* and the *target (desired)* images during training, respectively.

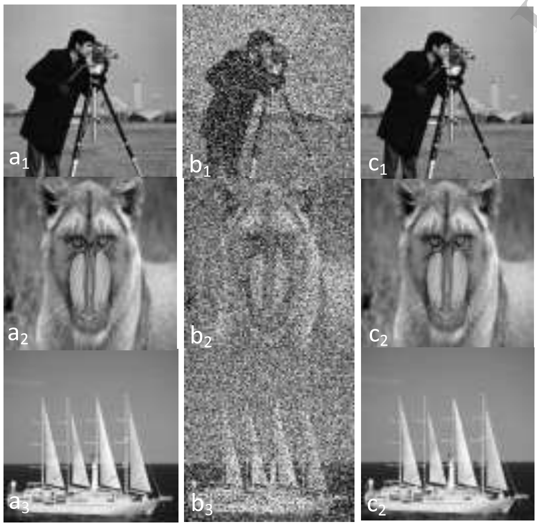

Fig.4 Performance of training image: ($a_{1,2\ \text{and}\ 3}$) original images, ($b_{1,2\ \text{and}\ 3}$) images corrupted with 45% of noise and ($c_{1,\ 2\ \text{and}\ 3}$) trained images

2.4 Testing of unknown images using trained structure of neural network

The optimized architecture that obtained the best performance for training with three images has 196608 data in the input layer, single layer with 16 neurons and one output layer. The network trained with 10% impulse noise and gaussian noise with the density of μ=0 and σ=200 shows superior performance for testing under various noise levels. The chosen network has been extensively tested for several images with different level of impulse noise. Fig.5 shows the exact procedure for taking corrupted data for testing the received image signals for the proposed filter. In order to reduce the computation time in real time implementation; in the first stage, a special class of filter is applied on unknown images and then pixels (data) from the outputs of noisy image and an decision based algorithm are obtained. Noisy image data and filtered image output data are applied as inputs for optimized neural network structure for testing.

At the same time, noise free pixels from input are directly taken as output pixels. The tested pixels are replaced in the same location on corrupted image instead of noisy pixels. The most typical feature of the proposed filter offers excellent line, edge, and fine detail preservation performance and also effectively removes impulse noise from the image. Usually conventional filters are giving denoised image output and then these images are enhanced using these conventional outputs as input for neural filter while these outputs are combined with the network. Since, networks need certain pattern to learn and understand the given data.

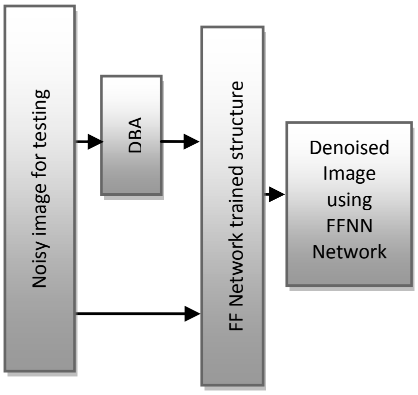
Noisy image for testing
DBA
FF Network trained structure
Denoised Image using FFNN Network

Fig.5 Testing of the images using optimized feed forward adaptive neural network structure

2.5 Filtering of the noisy image

The noisy input image is processed by sliding the 3x3 filtering window on the image. This filtering window is considered for the nonlinear filter. The window is started from the upper-left corner of the noisy input image, and moved rightwards and progressively downwards in a *raster scanning* fashion. For each filtering window, the nine pixels contained within the window of noisy images are first fed to the decision based algorithm. Next, the center pixel of the filtering window on noisy image, the output of the conventional filtered output is applied to the appropriate input for the neural network. Finally, the restored image is obtained at the output of this network.

3. Results and Discussion

The performance of the proposed filtering technique for image quality enhancement is tested for various levels of impulse noise and gaussian noise densities. Four images are selected for testing with size of 256 x 256 including *Baboon, Lena, Pepper and Ship.* All test images are 8-bit gray level images. The experimental images used in the simulations are generated by contaminating the original images by impulse noise and gaussian noise with different level of noise density. The experiments are especially designed to reveal the performances of the filters for different image properties and noise conditions. The performances of all filters are evaluated by using the *peak signal-to-noise ratio* (PSNR) criterion, which is defined as more objective *image* quality measurement and is given by the equation (3.1)

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \qquad (3.1)$$

where,

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|(x(i,j)-y(i,j)\right|^2 \qquad (3.2)$$

Here, M and N represents the number of rows and column of the image and $x(i, j)$ and $y(i, j)$ represents the original and the restored versions of a corrupted test image, respectively. The experimental procedure to evaluate the performance of a proposed filter is as follows: The noise density is varied gaussian noise from σ=100 to σ=600 with 100 increments and impulse noise of 10%. For each noise density step, the four test images are corrupted by impulse noise with that noise density. This generates four different experimental images, each having the same noise density. These images are restored by using the operator under experiment, and the PSNR values are calculated for the restored output images. By this method ten different PSNR values representing the filtering performance of that operator for different image properties, then this technique is separately repeated for all noise densities to obtain the variation of the average PSNR value of the proposed filter as a function of noise density. The entire input data are normalized in to the range of [0 1], whereas the output data is assigned to one for the highest probability and zero for the lowest probability.

Table 1
PSNR obtained by applying proposed filter on Lena image corrupted with 10 % of impulse noise with uniform noise σ=200.

S.No	Neural network architecture				PSNR
	No. of hidden layers	No. of neuron in each hidden layer			
		Layer 1	Layer2	Layer3	
1	1	6	-	-	26.9785
2	1	9	-	-	26.7231
3	1	16	-	-	**28.8684**
4	1	17	-	-	28.1124
5	1	20	-	-	26.0933

The architecture with two hidden layers and each hidden layer has 2 neurons yielded the best performance. The various parameters for the neural network training for all the patterns are summarized in Table 2 and 3. In Table 2, Performance error is nothing but Mean square error (MSE). It is a sum of the statistical bias and variance. The neural network performance can be improved by reducing both the statistical bias and the statistical variance. However there is a natural trade-off between the bias and variance.

Table 2
Optimized training parameters for feed forward neural network

S.No	Parameters	Achieved
1	Performance error	0.001
2	Learning Rate (LR)	0.1
3	No. of epochs taken to meet the performance goal	3000
4	Time taken to learn	2896 seconds

Learning Rate is a control parameter of training algorithms, which controls the step size when weights are iteratively adjusted. The learning rate is a constant in the algorithm of a neural network that affects the speed of learning. It will apply a smaller or larger proportion of the current adjustment to the previous weight If LR is low, network will learn all information from the given input data and it takes long time to learn. If it is high, network will skip some information from the given input data and it will make fast training. However lower learning rate gives better performance than higher learning rate. The learning time of a simple neural-network model is obtained through an analytic computation of the Eigen value spectrum for the Hessian matrix, which describes the second-order properties of the objective function in the space of coupling coefficients. The results are generic for symmetric matrices obtained by summing outer products of random vectors.

Table.3
Bias and Weight updation in optimized training neural network

Hidden layer			
		Weight	**Bias**
1^{st} Hidden layer	Weights from $x_{1\&2}$ to n_{11}	14.11;0.003	-13.26
	Weights from $x_{1\&2}$ to n_{12}	4.642;-1.590	-17.28
	Weights from $x_{1\&2}$ to n_{13}	-21.10;-0.608	25.94
	Weights from $x_{1\&2}$ to n_{14}	52.98;3.762	-25.37
	Weights from $x_{1\&2}$ to n_{15}	52.98;3.762	-151.7
	Weights from $x_{1\&2}$ to n_{16}	-0.711;0.107	14.43
	Weights from $x_{1\&2}$ to n_{17}	654.9;0.717	1.636
	Weights from $x_{1\&2}$ to n_{18}	-2.000;-0.006	1.837
	Weights from $x_{1\&2}$ to n_{19}	-426.7;1.883	425.9
	Weights from $x_{1\&2}$ to n_{110}	10.34;-0.008	-1.761
	Weights from $x_{1\&2}$ to n_{111}	25.48;0.570	-24.91
	Weights from $x_{1\&2}$ to n_{112}	778.3;-0.019	2.168
	Weights from $x_{1\&2}$ to n_{113}	21.12;0.607	-25.92
	Weights from $x_{1\&2}$ to n_{114}	17.83;-16.71	-0.509
	Weights from $x_{1\&2}$ to n_{115}	665.8;0.768	1.061
	Weights from $x_{1\&2}$ to n_{116}	3.354;-0.016	-1.535
Output layer	Weights from n_{21} to o	0.112	227.1
	Weights from n_{22} to o	-231.9	
	Weights from n_{23} to o	-88.18	
	Weights from n_{24} to o	-0.685	
	Weights from n_{25} to o	-0.013	
	Weights from n_{26} to o	230.5	
	Weights from n_{27} to o	-1047	
	Weights from n_{28} to o	-0.290	
	Weights from n_{29} to o	0.484	
	Weights from n_{210} to o	0.094	
	Weights from n_{211} to o	0.642	
	Weights from n_{212} to o	-0.294	
	Weights from n_{213} to o	-88.79	
	Weights from n_{214} to o	-0.0454	
	Weights from n_{215} to o	357.9	
	Weights from n_{216} to o	0.231	

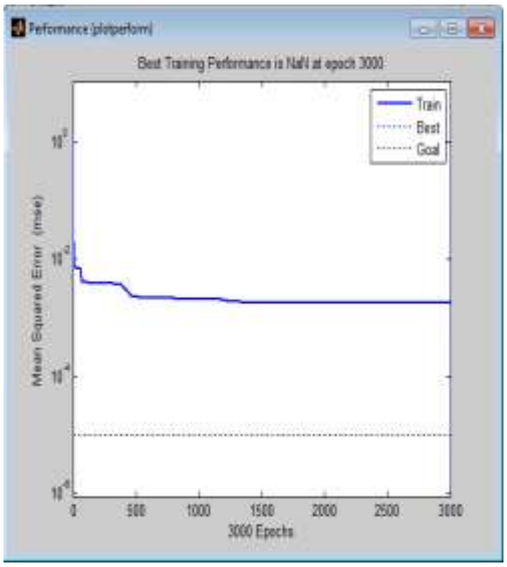

Fig.6 Performance error graph for feed forward neural network with back propagation algorithm

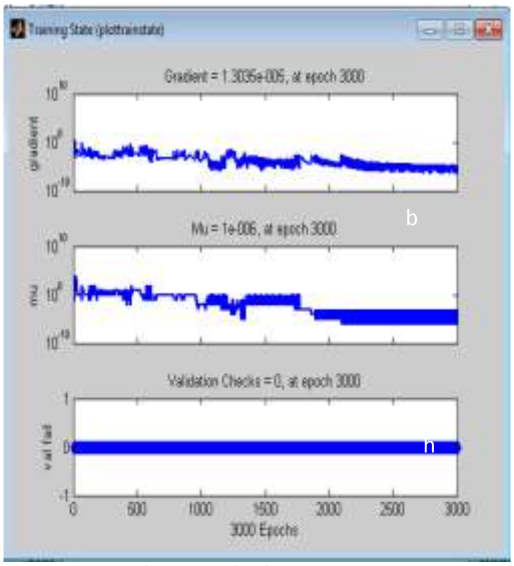
Training State (plottrainstate)
Gradient = 1.3035e-005, at epoch 3000
gradient
Mu = 1e-006, at epoch 3000
mu
Validation Checks = 0, at epoch 3000
val fail
1
0
-1
0
500
1000
1500
2000
2500
3000
3000 Epochs
b
n

Fig. 7 Performance of gradient for feed Forward neural network with back propagation algorithm

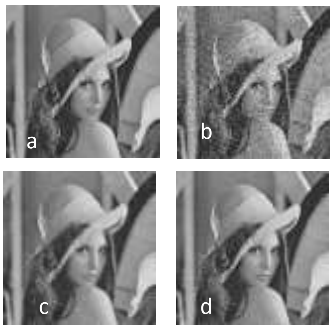

Fig.8 Subjective performance illustration of the proposed filtering technique compared with existing technique: (a) Original Lena image, (b) Lena image Corrupted by 10% impulse noise and gaussian noise with zero mean and σ^2=200, (c) Restored by multiple noise elimination, (d) Restored by the proposed filter

Table I
Psnr values obtained using different filtering technique on lena image corrupted with various densities of impulse noise

Noise		MNE	Neural based MNE
Gaussian noise	Impulse noise		
100	10%	28.9294	**29.0134**
200	10%	28.5258	**28.8684**
300	10%	28.2816	**28.6561**
400	10%	28.0382	**28.4084**
500	10%	27.9112	**27.9440**
600	10%	27.6408	**27.7532**
700	10%	27.4216	**27.5891**

Lena test image contaminated with the impulse noise of various densities are summarized in Table 3 for quantitative metrics for different filtering techniques and compared with the proposed filtering technique and is graphically illustrated in Fig.9. This graphical illustration shows the performance comparison of the proposed intelligent filter. This qualitative measurement proves that the proposed filtering technique outperforms the other filtering schemes.

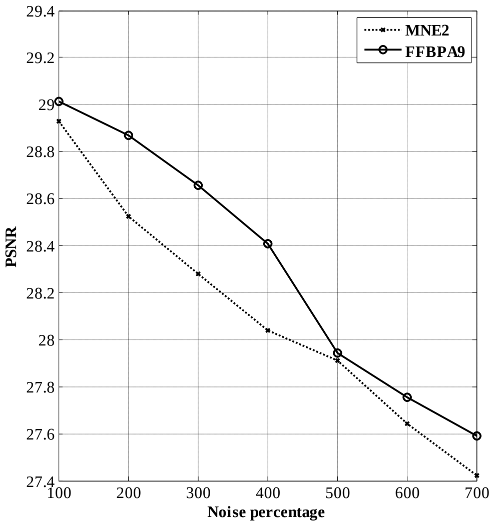

Fig.9 PSNR obtained by applying the proposed algorithm and compared with different technique on Lena image corrupted with various densities of gaussian noise and impulse noise of 0.1

The PSNR performance explores the quantitative measurement. In order to check the performance of the feed forward neural network, percentage improvement (PI) in PSNR is also calculated for performance comparison between conventional filters and proposed neural filter for Lena image and is summarized in Table 4. The PSNR performance explores the quantitative measurement. In order to check the performance of the feed forward neural network, percentage improvement (PI) in PSNR is also calculated for performance comparison between conventional filters and proposed neural filter for Lena image and is summarized in Table 4. This PI in PSNR is calculated by the following equation 8.

$$PI = \left[\frac{PSNR_{CF} - PSNR_{NF}}{PSNR_{CF}} x100 \right] \tag{8}$$

where PI represents percentage in PSNR, $PSNR_{CF}$ represents PSNR for conventional filter and $PSNR_{NF}$ represents PSNR values for the designed neural filter. Here, the conventional filters are combined with neural network which gives the proposed filter, so that the performance of conventional filter is improved.

Table 4
Percentage improvement in PSNR obtained on Lena image corrupted with different level of impulse noise

Impulse Noise %	Gaussian noise	Proposed filter (PF)	Conventional filter (CF)	PI for Proposed filter
10	100	29.01	28.92	0.29
10	200	28.86	28.52	1.20
10	300	28.65	28.28	1.32
10	400	28.40	28.03	1.32
10	500	27.94	27.91	0.13
10	600	27.75	27.64	0.41
10	700	27.58	27.42	0.61

In Table 4, the summarized PSNR values for conventional filters namely NF1 and NF2 seem to perform well for human visual perception when images are corrupted up to 10% of impulse noise and uniform noise with zero mean and σ=200. These filters performance are better for quantitative measures when images are corrupted up to 10% of impulse noise and and uniform noise with zero mean and σ=400. In addition to these, image enhancement is nothing but improving the visual quality of digital images for some application. In order to improve the performance of visual quality of image using these filters, image enhancement as well as reduction in misclassification of pixels on a given image is obtained by applying Feed forward neural network with back propagation algorithm. The summarized PSNR values in Table 4 for the proposed neural filter appears to perform well for human visual perception when images are corrupted up to 10% of impulse noise and uniform noise with zero mean and σ=200. These filters performance are better for quantitative measures when images are corrupted up to 10% of impulse noise and uniform noise with zero mean and σ=400. PI is graphically illustrated in Fig.10.

Digital images are nonstationary process; therefore depends on properties of edges and homogenous region of the test images, each digital images having different quantitative measures. Fig.11 illustrate the subjective performance for proposed filtering Technique for Baboon, Lena, Pepper and Rice images: noise free image in first column, images corrupted with 10% impulse noise and uniform noise with zero mean and σ=200 in second column, Images restored by proposed Filtering Technique in third column. This will felt out the properties of digital images.

Performance of quantitative analysis is evaluated and is summarized in Table.5. This is graphically illustrated in Fig.12. This qualitative and quantitative measurement shows that the proposed filtering technique outperforms the other filtering schemes. The qualitative and quantitative performance of Pepper and Rice images are better than the other images for low noise level. But for higher noise levels, the Pepper image is better. The Baboon image seems to perform poorly for higher noise levels.

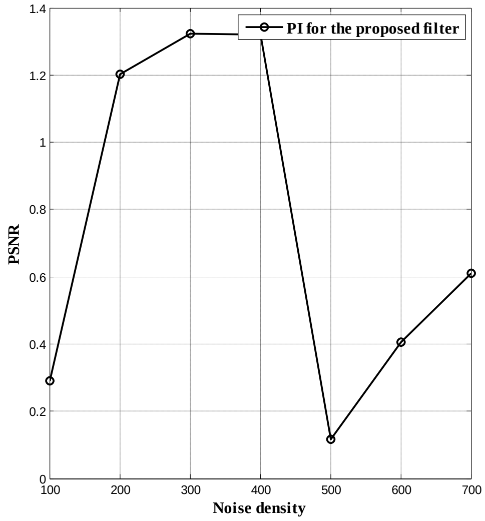

Fig.10 PI in PSNR obtained on Lena image for the proposed filter corrupted with various densities of mixed impulse noise

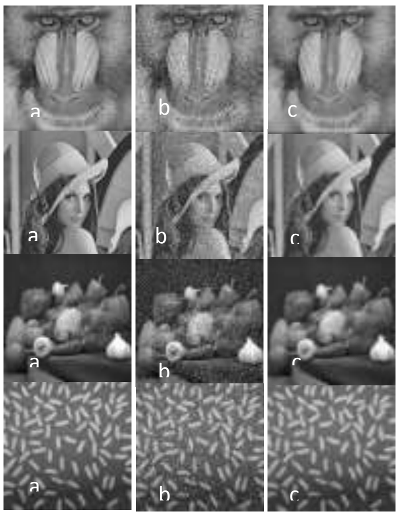

Fig.11 Performance of test images:($a_{1,2,.., 4}$) original images,($b_{1,2,...,4}$) images corrupted with 10% of impulse noise and gassian noise with zero mean and σ=200 and ($c_{1, 2,...,4}$) images enhanced by proposed filter

Based on the intensity level or brightness level of the image, it is concluded that the performance of the images like pepper, Lena, Baboon and Rice will change. Since digital images are nonstationary process. The proposed filtering technique is found to have eliminated the impulse noise completely while preserving the image features quite satisfactorily. This novel filter can be used as a powerful tool for efficient removal of impulse noise from digital images without distorting the useful information in the image and gives more pleasant for visual perception.

Table 5
PSNR obtained for the proposed filter on different test images with various densities of random valued impulse noise

Noise		Baboon	Lena	Pepper	Rice
Gaussian noise	Impulse noise				
100	10%	24.6603	29.0134	34.2204	32.4657
200	10%	23.9232	28.8684	33.9674	31.9801
300	10%	23.5716	28.6561	33.6540	31.6732
400	10%	23.3680	28.4084	33.3411	31.3290
500	10%	22.8399	27.9440	32.8306	30.8722
600	10%	22.5754	27.7532	32.5430	30.5411
700	10%	22.1822	27.5891	31.2307	29.2145

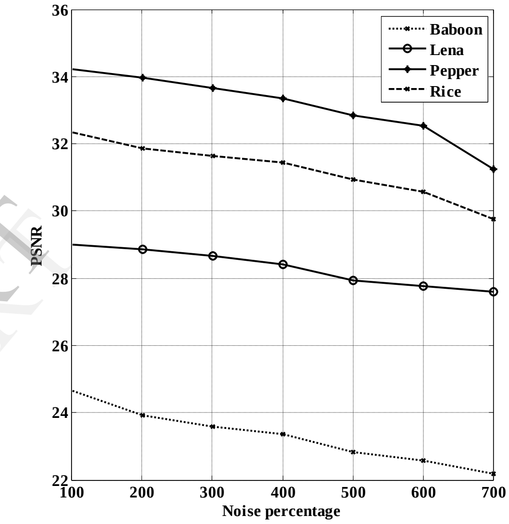

Fig. 12 PSNR obtained by applying proposed filter technique for different images corrupted with various densities of mixed impulse noise

In addition, it can be observed that the proposed filter for image restoration is better in preserving the edges and fine details than the other existing filtering algorithm. It is constructed by appropriately combining a two nonlinear filters and a neural network. This technique is simple in implementation and in training; the proposed operator may be used for efficiently filtering any image corrupted by impulse noise and gaussian noise of virtually any noise density. It is concluded that the proposed filtering technique can be used as a powerful tool for efficient removal of impulse noise from digital images without distorting the useful information within the image**.**

5. Conclusion

A feed forward back propagation algorithm is described in this paper for image denoising. This filter is seen to be quite effective in preserving image boundary and fine details of digital images while eliminating multiple noises. The efficiency of the proposed filter is illustrated applying the filter on various test images contaminated different levels of noise. This filter outperforms the existing filters in terms of objective and subjective measures. So that the proposed filter output images are found to be pleasant for visual perception.

5. References

[1] J.Astola and P.Kuosmanen Fundamental of Nonlinear Digital Filtering. NewYork:CRC, 1997.

[2] I.Pitasand .N.Venetsanooulos, Nonlinear Digital Filters:Principles Applications. Boston, MA: Kluwer, 1990.

[3] W.K. Pratt, Digital Image Processing, Wiley, 1978.

[4] T.Chen, K.-K.Ma,andL.-H.Chen, “Tristate median filter for image denoising”, IEEE Trans.Image Process., 1991, 8, (12), pp.1834-1838.

[5] Zhigang Zeng and Jun Wang, “Advances in Neural Network Research and Applications”, Lecture notesm springer, 2010.

[6] M. Barni, V. Cappellini, and A. Mecocci, “Fast vector median filter based on Euclidian norm approximation”, IEEE Signal Process. Lett., vol.1, no. 6, pp. 92– 94, Jun. 1994.

[7] Sebastian hoyos and Yinbo Li, “Weighted Median Filters Admitting Complex -Valued Weights and their Optimization”, IEEE transactions on Signal Processing, Oct. 2004, 52, (10), pp. 2776-2787.

[8] E.Abreu, M.Lightstone, S.K.Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images”,IEEE Trans. Image Processing, 1996, 5, (6), pp. 1012–1025.

[9] T.Sun and Y.Neuvo, “Detail preserving median filters in image processing”, Pattern Recognition Lett., April 1994, 15, (4), pp.341-347.

[10] Zhang and M.- A. Karim, “A new impulse detector for switching median filters”, IEEE Signal Process. Lett., Nov. 2002, 9, (11), pp. 360–363.

[11] Z. Wang and D. Zhang, “Progressive Switching median filter for the removal of impulse noise from highly corrupted images”, IEEE Trans. Circuits Syst. II, Jan. 2002, 46, (1), pp.78–80.

[12] H.-L. Eng and K.-K. Ma, “Noise adaptive soft – switching median filter,” IEEE Trans.Image Processing, , Feb. 2001, 10, (2), pp. 242–25.

[13] Pei-Eng Ng and Kai - Kuang Ma, “A Switching median filter with boundary Discriminative noise detection for extremely corrupted images”, IEEE Transactions on image Processing, June 2006, 15, (6), pp.1500-1516.

[14] Tzu – Chao Lin and Pao - Ta Yu, “salt – Pepper Impulse noise detection”, Journal of Information science and engineering, June 2007, 4, pp189-198.

[15] E.Srinivasan and R.Pushpavalli, “ Multiple Thresholds Switching Median Filtering for Eliminating Impulse Noise in Images”, International conference on Signal Processing, CIT, Aug. 2007.

[16] R.Pushpavalli and E.Srinivasan, “Multiple Decision Based Switching Median Filtering for Eliminating Impulse Noise with Edge and Fine Detail preservation Properties”, International conference on Signal Processing, CIT , Aug. 2007.

[17] Yan Zhouand Quan-huanTang, “Adaptive Fuzzy Median Filter for Images Corrupted by Impulse Noise”, Congress on image and signal processing, 2008, 3, pp. 265 – 269.

[18] Shakair Kaisar and Jubayer AI Mahmud, “ Salt and Pepper Noise Detection and removal by Tolerance based selective Arithmetic Mean Filtering Technique for image restoration”, IJCSNS, June 2008, 8,(6), pp. 309 – 313.

[19] T.C.Lin and P.T.Yu, “Adaptive two-pass median filter based on support vector machine for image restoration ”, Neural Computation, 2004, 16, pp.333-354,

[20] Madhu S.Nair, K.Revathy, RaoTatavarti, "An Improved Decision Based Algorithm For Impulse Noise Removal", Proceedings of International Congress on Image and Signal Processing - CISP 2008, IEEE Computer Society Press, Sanya, Hainan, China, May 2008, 1, pp.426-431.

[21] V.Jayaraj and D.Ebenezer,“A New Adaptive Decision Based Robust Statistics Estimation Filter for High Density Impulse Noise in Images and Videos”, International conference on Control, Automation, Communication and Energy conversion, June 2009, pp 1 - 6.

[22] Fei Duan and Yu – Jin Zhang,“A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters”, IEEE Signal processing Letters, July 2010, 17,(7), pp. 647 – 650.

[23] R.Pushpavalli and G.Sivaradje, “Nonlinear Filtering Technique for Preserving Edges and Fine Details on Digital Image”, International Journal of Electronics and Communication Engineering and Technology, January 2012, 3, (1),pp29-40.

[24] R.Pushpavalli and E.Srinivasan, “Decision based Switching Median Filtering Technique for Image Denoising”, CiiT International journal of Digital Image Processing, Oct.2010, 2, (10), pp.405-410.

[25] R.Pushpavalli, E. Srinivasan and S.Himavathi, “A New Nonlinear Filtering technique”, 2010 International Conference on Advances in Recent Technologies in Communication and Computing, ACEEE, Oct. 2010, pp1-4.

[26] R.Pushpavalli and G.Sivaradje, “New Tristate Switching Median Filter for Image Enhancement” International Journal of Advanced research and Engineering Technology, January-June 2012, 3, (1), pp.55-65.

[27] A.Fabijanska and D.Sankowski, “ Noise adaptive switching median-based filter for impulse noise removal from extremely corrupted images”, IET image processing, July 2010, 5, (5), pp.472-480.

[28] S.Esakkirajan, T,Veerakumar, Adabala.N Subramanyam, and C.H. Premchand, "Removal of High Density Salt & pepper Noise Through Modified Decision based Unsymmetric Trimmed Median Filter", IEEE Signal processing letters, May 2011, 18, (5), pp.287-290.

[29] A.L.Betker,T.Szturm, Z. oussavi1,"Application of Feed forward Back propagation Neural Network to Center of Mass Estimation for Use in a Clinical Environment", IEEE Proceedings of Engineering in Medicine and Biology Society, April 2004, Vol.3, 2714 – 2717.

[30] Chen Jindu and Ding Runtao Ding, "A Feed forward neural Network for Image processing", *in IEEE proceedings of ICSP*, pp.1477-1480, 1996.

[31] Wei Qian, Huaidong Li, Maria Kallergi, Dansheng Song and Laurence P. Clarke, "Adaptive Neural Network for Nuclear Medicine Image Restoration", *Journal of VLSI Signal Processing, vol.* 18, 297–315, 1998, Kluwer Academic Publishers.

[32] R.Pushpavalli, G.Shivaradje, E. Srinivasan and S.Himavathi, " Neural Based Post Processing Filtering Technique For Image Quality Enhancement", *International Journal of Computer Applications*, January-2012.

[33] E.Srinivasan, R.Selvam and D.Ebenezer, "A Nonlinear variable Cut-off Highpass Filter Algorithm for VLSI Implementation", Proc. in International conference on Information, Communication and Signal Processing (ICICS'99), NTU, Singapore, 3E2.4, Dec. 1999.

[34] K. Vasanth, S. Karthik and Sindu Divakaran, "Removal of Salt & Pepper Noise using Unsymmetrical Trimmed Variants as Detector",European Journal of Scientific Research ISSN 1450-216X Vol.70 No.3 (2012), pp. 468-478, © EuroJournals Publishing, Inc. 2012.

[35] Gaurang Panchal , Amit Ganatra, Y P Kosta and Devyani Panchal, "Forecasting Employee Retention Probability using Back Propagation Neural Network Algorithm", *Second International Conference on Machine Learning and Computing*,2010, pp.248-251.

[36] Sudhansu kumar Misra, Ganpati panda and Sukadev mehar, "Cheshev Functional link Artificial neural Networks for Denoising of Image Corrupted by Salt & Pepper Noise", International journal of rcent Trends in Engineering, may 2009, 1, (1), pp.413-417.

[37] Weibin Hong, Wei Chen and Rui Zhang, " The Application of Neural Network in the Technology of Image processing", Proceddings of the International Conference of Engineers and Computer Sciences, 2009, 1.

[38] A new methos of denoising mixed noise using limited Grayscale Pulsed Coupled Neural Network", Cross Quad-Regional Radio Science and Wireless Technology Conference, 2011, pp.1411-1413.

[39] Shamik Tiwari, Ajay kumar Singh and V.P.Shukla, "Staistical Moments based Noise Classification using Feed Forward back Peopagation neural Network", International journal of Computer Applications, March 2011, 18, (2), pp.36-40.

[40] Anurag Sharma, " Gradient Descent Feed Forward Neural Networks for Forecasting the Trajectories", International Journal of Advanced Science and Technology, September 2011, 34, pp.83-88.