Least Square Filter and MLMVN Network for Image Denoising

مرشح المربعات الصغرى وشبكة MLMVN العصبية لاجل رفع الضوضاء من الصورة

Zeyad Farooq Lutfi
Mustansiriyah University- Science College

Abstract

Image noise removal is one of the image processing used to repaint the image and restore image information affected by a noise types. One of image problems is the noise affect in the image. To solve this problem, a new denoising technique was proposed in order to reduced or remove the noise effect in the images. A new proposed filter was design using the Least Square interpolation to calculate the pixels value for pixels affected by noise. The proposed system will used to replace the noisy image pixels by results of proposed filter under control of the neural network. The MLMVN neural network was used in this proposed system due the ability of this neural network to optimization and accuracy in decision making. The proposed filter was work with good results for fully and partially noising region in images.

المستخلص:

إزالة ضوضاء الصورة هي واحدة من معالجة الصور المستخدمة لإعادة رسم الصورة واستعادة المعلومات المتضررة بأنواع الضوضاء. وحدة من مشاكل الصورهو تأثير الضوضاء فى الصورة. لحل هذه المشكلة ، اقترحت تقنية جديدة لتقليل الضوضاء ولخفض أو إزالة تأثير الضوضاء في الصور. تصميم الجديد المقترح مرشح يستخدم مربعات الصغرى للاستيفاء في حساب قيمة بكسل المتأثرين بالضوضاء. النظام المقترح سيبدل قيم بكسلات الصورة المتضررة بالضوضاء مع نتائج المرشح المقترح تحت سيطرة الشبكة العصبية. واستخدمت الشبكة العصبية MLMVN فى هذا النظام المقترح بسبب قدرة هذه الشبكة العصبية على التحسين ودقة في صنع القرار. مرشح المقترح يعمل بنتائج جيدة للمناطق الصورة الكاملة والجزئية المتضررة بالضوضاء .

1. Introduction

Image processing is a field that continues to grow, with new applications being developed at an ever-increasing pace. It is a fascinating and exciting area to be involved in today with application areas ranging from the entertainment industry to the space program. One of the most interesting aspects of this information revolution is the ability to send and receive complex data that transcends ordinary written text. Visual information, transmitted in the form of digital images, has become a major method of communication for the 21st century. [1]

Many techniques for noise reduction replace each pixel with some function of the pixel's neighborhood. Because 1D features and 2D noise usually have common frequency components, they are not separable in the frequency domain. Hence, linear filters seldom can meet goals 1 and 2 simultaneously. Linear filters tend either to amplify the noise a long with the 1D features, or to smooth out the noise and blur the 1D features. To minimize the conflict between goals 1 and 2 above, researchers have introduced a number of adaptive noise reduction algorithms.[2

In the most applications, it is very important to remove impulse noise from image data, since the performances of subsequent image processing tasks are strictly dependent on the success of image noise removal operation. However, this is a difficult problem in any image processing system because the restoration filter must not distort the useful information in the image and preserve image details and texture while removing the noise. [3]

Noise modeling in images is greatly affected by capturing instruments, data transmission media, image quantization and discrete sources of radiation. Different algorithms are used depending on the noise model. Most of the natural images are assumed to have additive random noise, which is modeled as a Gaussian. [1]

Automated techniques for identification of image noise are of considerable interest, because once the type of noise is identified from the given image, an appropriate algorithm can then be used to de-noise it. Only a few researchers have addressed this issue to date. However, many algorithms proposed are pretty complicated, because their main goal is to estimate the statistical parameters of the noise.[1]

In this paper, an image denoising system was proposed to reduce or remove three types of noise (Gaussian noise, Salt and Pepper noise, and uniform noise) from noisy image. The denoising operation proposed depending on the Least Square Polynomial calculations. Also, the MLMVN neural network was used to controlling the denoising operation in order to increase the efficiency of the denoising operation results.

2. Image Denoising

In the past three decades, a variety of denoising methods have been developed in the image processing and computer vision communities. Although seemingly very different, they all share the same property: to keep the meaningful edges and remove less meaningful ones. A categorize the existing image denoising work by their different natural image prior models and the corresponding representation of natural image statistics.[4]

Wavelets. When a natural image is decomposed into multiscale oriented subbands. To enforce the marginal distribution to have high kurtosis, we can simply suppress low-amplitude values while retaining high-amplitude values, a technique known as coring. The joint distributions of wavelets were found to be dependent. A joint coring technique is developed to infer the wavelet coefficients in a small neighborhood across different orientation and scale subbands simultaneously. The typical joint distribution for denoising is a Gaussian scale mixture (GSM) model. In addition, wavelet-domain hidden Markov models have been applied to image denoising with promising results. Although the wavelet-based method is popular and dominant in denoising, it is hard to remove the ringing artifacts of wavelet reconstruction. In other words, wavelet-based methods tend to introduce additional edges or structures in the denoised image.[4]

Anisotropic Diffusion. The simplest method for noise removal is Gaussian filtering, which is equivalent to solving an isotropic heat diffusion equation, a second order linear PDE. As a result, for high gradient pixels, c(x, y, t) is small and therefore gets less diffused. For low gradient pixels, c(x, y, t) has a higher value and these pixels get blurred with neighboring pixels. Compared to simple Gaussian filtering, anisotropic diffusion smooths out noise while keeping edges. However, it tends to over-blur the image and sharpen the boundary with many texture details lost.[4]

More advanced partial differential equations (PDEs) have been developed so that a specific regularization process is designed for a given (user-defined) underlying local smoothing geometry, preserving more texture details than the classical anisotropic diffusion methods.[4]

FRAME & FOE. As an alternative to measuring marginal or joint distributions on wavelet coefficients, a complete prior model over the whole image can be learnt from marginal distributions. Thus, it is natural to use a Bayesian inference for denoising or restoration. [4]

Bilateral Filtering. An alternative way of adapting Gaussian filtering to preserve edges is bilateral filtering, where both space and range distances are taken into account. Fast bilateral filtering algorithm was also used. Bilateral filtering has been widely adopted as a simple algorithm for denoising, e.g., video denoising. However, it cannot handle speckle noise and it also has the tendency of over smoothing and edge sharpening.[4]

Nonlocal Methods. If both the scene and camera are static, we can simply take multiple pictures and use the mean to remove the noise. This method is impractical for a single image, but a temporal mean can be computed from a spatial mean–as long as there are enough similar patterns in the single image. We can find the similar patterns to a query patch and take the mean or other statistics to estimate the true pixel value, e.g.[4]

Nonlocal methods are an exciting innovation and work well for texture-like images containing many repeated patterns. However, compared to other denoising algorithms that have $O(n^2)$ complexity where n is the image width, these algorithms have $O(n^4)$ time complexity, which is prohibitive for real-world applications.[4]

Conditional Random Fields. Recently, conditional random fields (CRFs) have been a promising model for statistical inference. Without an explicit prior model on the signal, CRFs are flexible at modeling short and long range constraints and statistics. Since the noisy input and the clean image are well aligned at image features, CRFs, in particular Gaussian conditional random fields (GCRFs) can be well applied to image denoising.[4]

3. Image Noise Estimation

Image-dependent noise can be estimated from multiple or a single image. Estimation from multiple images is an over-constrained problem. Estimation from a single image, however, is an under-constrained problem and further assumptions have to be made for the noise. In the image denoising literature, noise is often assumed to be additive white Gaussian noise (AWGN). A widely used estimation method is based on mean absolute deviation. The noise level is estimated from the gradient of smooth or small textured regions, and the signal-dependent noise level is estimated for each intensity interval. The literature researchers proposed three methods to estimate noise levels based on training samples and the statistics (Laplacian) of natural images. A generalized expectation maximization algorithm is proposed to estimate the spectral features of a noise source corrupting an observed image. [4]

Estimation of the standard deviation is done by considering only regions in the window for each pixel having the frequent label. The mean and variance are found for all the regions for each pixel and then standard deviation is calculated. For the additive noise, the average of the square root of the variance identified for each of the homogeneous region in the noisy image gives you the estimate of the standard deviation of the additive noise affecting the image. For the multiplicative noise, the variance increases with an increase in mean value in a parabolic manner. So the standard deviation is estimated a little different. The standard deviation is found for each of the homogeneous region and it is divided by the corresponding mean of the homogeneous region to get a corresponding value. This corresponding value is got for all the homogeneous regions present in the noisy image. The average of all the values thus got constitutes the estimated standard deviation of the multiplicative noise. [1]

Techniques for noise estimation followed by noise reduction have been proposed, but they tend to be heuristic. For example, a set of statistics of film grain noise are used to estimate and remove the noise produced from scanning the photographic element under uniform exposures. A signal-dependent noise is estimated from the smooth regions of the image by segmenting the image gradient with an adaptive threshold. The estimated signal-dependent noise is applied to the whole image for noise reduction. This work was further extended by associating a default film-related noise model to the image based on its source identification tag. The noise model is then adjusted using the image statistics. In certain commercially available image enhancement software, such as Neat ImageTM1, the noise level can be semi automatically estimated by specifying featureless areas to profile noise. Neat Image TM also provides calibration tools to estimate the amount of noise for a specific camera and camera setting; pre-calibrated noise profiles for various cameras are also available to directly denoise images. [4]

4. The Least Squares Estimation Method [5]

The method of least squares is a standard approach to the approximate solution of over determined systems, i.e. sets of equations in which there are more equations than unknowns. "Least squares" means that the overall solution minimizes the sum of the squares of the errors made in solving every single equation.

The most important application is in data fitting. The best fit in the least-squares sense minimizes the sum of squared residuals, a residual being the difference between an observed value and the fitted value provided by a model.

Least squares problems fall into two categories: linear least squares and nonlinear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem occurs in statistical regression analysis; it has a closed-form solution. The non-linear problem has no closed solution and is usually solved by iterative refinement; at each iteration the system is approximated by a linear one, thus the core calculation is similar in both cases.

The least-squares method was first described by Carl Friedrich Gauss around 1794. Least squares correspond to the maximum likelihood criterion if the experimental errors have a normal distribution and can also be derived as a method of moments estimator.

The objective consists of adjusting the parameters of a model function to best fit a data set. A simple data set consists of n points (data pairs) (x_i, y_i), i = 1, ..., n, where x_i is an independent variable and y_i is a dependent variable whose value is found by observation. The model function has the form f(x,β), where the m adjustable parameters are held in the vector β. The goal is to find the parameter values for the model which "best" fits the data. The least squares method finds its optimum when the sum, S, of squared residuals

$$S = \sum_{i=1}^{n} r_i^2 \qquad \text{....(1)}$$

is a minimum. A residual is defined as the difference between the value predicted by the model and the actual value of the dependent variable

$$r_i = f(x_i, \beta) - y_i. \qquad \text{....(2)}$$

An example of a model is that of the straight line. Denoting the intercept as β_0 and the slope as β_1, the model function is given by $f(x, \beta) = \beta_0 + \beta_1 x$. A data point may consist of more than one independent variable. For an example, when fitting a plane to a set of height measurements, the plane is a function of two independent variables, x and z, say. In the most general case there may be one or more independent variables and one or more dependent variables at each data point. The minimum of the sum of squares is found by setting the gradient to zero. Since the model contains m parameters there are m gradient equations.

$$\frac{\partial S}{\partial \beta_j} = 2\sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0,\ j = 1,\dots,m \quad \dots(3)$$

and since $r_i = y_i - f(x_i, \beta)$ the gradient equations become

$$-2\sum_i \frac{\partial f(x_i, \beta)}{\partial \beta_j} r_i = 0,\ j = 1,\dots,m \quad \dots(4)$$

The gradient equations apply to all least squares problems. Each particular problem requires particular expressions for the model and its partial derivatives.

A generalization to approximation of a data set is the approximation of a function by a sum of other functions, usually an orthogonal set:

$$f(x) \approx f_n(x) = a_1\phi_1(x) + a_2\phi_2(x) + \cdots + a_n\phi_n(x). \quad \dots(5)$$

with the set of functions $\{\phi_j(x)\}$ an orthonormal set over the interval of interest, say [a, b]. The coefficients $\{a_j\}$ are selected to make the magnitude of the difference $\|f - f_n\|^2$ as small as possible.

5. Multilayer Neural Network Based On Multi-Valued Neurons [6]

A multilayer neural network based on multi-valued neurons (MLMVN) has been introduced, investigated and developed. This network consists of multi-valued neurons (MVN). That is a neuron with complex-valued weights and an activation function, defined as a function of the argument of a weighted sum. This activation function was proposed in 1971 in the pioneer paper of N. Aizenberg et al. The multi-valued neuron was introduced based on the principles of multiple-valued threshold logic over the field of complex numbers formulated.

The most important properties of MVN are: the complex-valued weights, inputs and output lying on the unit circle, and the activation function, which maps the complex plane into the unit circle. It is important that MVN learning is reduced to the movement along the unit circle. The MVN learning algorithm is based on a simple linear error correction rule and it does not require differentiability of the activation function.

Different applications of MVN have been considered during recent years, e.g.: MVN as a basic neuron in the cellular neural networks, as the basic neuron of the neural-based associative memories, as the basic neuron in a variety of pattern recognition systems, and as a basic neuron of the MLMVN.

The MLMVN outperforms a classical multilayer feedforward network and different kernel-based networks in the terms of learning speed, network complexity, and classification/prediction rate tested for such popular benchmarks problems as the parity n, the two spirals, the sonar, and the Mackey-Glass time series prediction. These properties of MLMVN show that it is more flexible and adapts faster in comparison with other solutions.

6. Multilayer Neural Network Based On Multi-Valued Neurons[6]

A continuous-valued MVN has been introduced in [6]. It performs a mapping between n inputs and a single output using n+1 complex-valued weights

$$f(x_1, \dots, x_n) = P(w_0 + w_1 x_1 + \dots + w_n x_n), \quad \dots(6)$$

where $X = x_1, \dots, x_n$ is a vector of complex-valued inputs (a pattern vector) and $W = w_0, w_1, \dots, w_n$ is a weighting vector. P is the activation function of the neuron:

$$P(z) = \exp(i\ (\arg z)) = e^{i\arg z} = \frac{z}{|z|}. \quad \dots(7)$$

where $z = w_0 + w_1 x_1 + \dots + w_n x_n$ is a weighted sum, arg z is an argument of the complex number z, Arg z is a main value of the argument of the complex number z and |z| is its modulo.

The MVN learning is reduced to the movement along the unit circle. This movement does not require differentiability of the activation function. Any direction along the circle always leads to the target. The shortest way of this movement is completely determined by an error that is a difference between the desired and actual outputs. The corresponding learning rule is:

$$W_{r+1} = W_r - \frac{C_r}{(n+1)}(\varepsilon^q - e^{i\arg z})\bar{X} = W_r - \frac{C_r}{(n+1)}(\varepsilon^q - \frac{z}{|z|})\bar{X} \quad \dots(8)$$

where $\bar{X}$ denotes vector with the complex-conjugated elements to input pattern vector X, W_r is a current weighting vector, W_{r+1} is a weighting vector after correction, C_r is a learning rate. A modified learning rule is:

$$W_{r+1} = W_r - \frac{C_r}{(n+1)|z_r|}\left(\varepsilon^q - \frac{z}{|z|}\right)\bar{X} \quad ..(9)$$

where z_r is a current value of the weighted sum.

A multilayer feedforward neural network based on multi-valued neurons (MLMVN) has been proposed in [6]. It refers to the basic principles of the network with a feedforward dataflow through nodes proposed by D. E. Rumelhart and J. L. McClelland. The most important is that there is a full connection between the consecutive layers (the outputs of neurons from the preceding layer are connected with the corresponding inputs of neurons from the following layer). The network contains one input layer, m-1 hidden layers and one output layer. Let us use here the following notations. Let km T be a desired output of the k^{th} neuron from the m^{th} (output) layer; Y_{km} be an actual output of the k^{th} neuron from the m^{th} (output) layer. Then the global error of the network taken from the k^{th} neuron of the m^{th} (output) layer is calculated as follows:

$\delta^*_{km} = T_{km} - Y_{km}$ ...(10)

The square error functional for the s^{th} pattern $X_s = x_1, ..., x_n$ is as follows:

$$E_s = \sum_k (\delta^*_{km})^2 (W),$$

...(11)

where δ^*_{km} is a global error taken from the k^{th} neuron of the m^{th} (output) layer, E_s is a square error of the network for the s^{th} pattern, and W denotes all the weighting vectors of all the neurons of the network. The mean square error functional for the network is defined as follows:

$$E = \frac{1}{N}\sum_{s=1}^{N} E_s .$$

...(12)

where N is a total number of patterns in the training set.

The backpropagation learning algorithm for the MLMVN was used; the errors of all the neurons from the network are determined by the global errors of the network (eqn.10). Finally, the MLMVN learning is based on the minimization of the error functional (eqn.12). It is fundamental that the global error of the network consists not only of the output neurons errors, but of the local errors of the output neurons and hidden neurons. It means that in order to obtain the local errors for all neurons, the global error must be shared among these neurons.

7. The Proposed Least Squares Interpolation Filter

In this section, the Least Square filter design will explain. This filter was proposed to repaint image depend on the least square calculation method. The result from this filter will used to remove or reduce the noise in the noisy images. The main equations of this proposed filter are develop to the eqn.(1) and (2) applied to the suggested mask. The Leas square developed equations are:

$S = \sum_{i=1}^{n} r_i^2$ and $r_i = ((f(x_i) - y_i) + (M - y_i))/2$ ...(13)

where M is the average pixels for the suggested mask.

The Least Squares interpolation resulted values that used to replace the selected pixels in the original image depend on the net of neighbors pixels effect. The replacing operation will covered all pixels on the suggested mask.

The suggested mask design to gives the best pixels effect calculation. The suggested mask size is (5x5) pixels. Figure (1) shows an example of the design mask. The mask will apply to whole image until denoising operation complete.

X0	X1	X3	X4	X5
X6	X7	X8	X9	X10
X11	X12	X13	X14	X15
X16	X17	X18	X19	X20
X21	X22	X23	X24	X25

100	120	125	110	100	130	132	125	116	110
120	124	110	110	110	120	122	129	110	110
123	124	141	111	110	120	132	128	110	120
114	125	120	112	115	140	124	130	111	115
112	126	123	111	110	150	120	155	113	115
134	123	114	111	110	160	140	150	112	115
145	110	110	112	114	110	120	165	115	118
121	100	120	113	113	120	125	145	110	120

Fig.(1) Example of The Least Squares Interpolation Mask

As shown in Fig.(1), the mask points to selected named by x0, x1, x2, x3, ...x15. These selected pixels will replaced by the resulted values from the Least Square calculation. (Note: the replacing operation will control by the MLMVN network).

8. The Proposed Image Denoising System

Image restoration processing like denoising average noise from images is a very difficult problem to resolve. In this research we describe a strategy that can be used for solving such image noise problems. This strategy is used to evaluate the proposed filter to denoise image (explain in section 7) control by neural network explain in section 5 and 6. A single neural network (the discrete-valued MLMVN) with the original backpropagation training scheme was used to identify both smoothing operator and its parameter on a single observed noisy image. Where, the discrete-valued MLMVN had such a drawback as discrete inputs which results in quantization error of pattern vectors. The structure of the neural network used in this proposed system was explained bellow.

As explain in section 7, the proposed filter was design using the principles and theory of the Least polynomial interpolation to estimate the pixels values to replace in the mask window of the filter. For each mask window, the proposed system will use the MLMVN to test each pixel in the window if pixel noise or not noise.

In this paper we apply MLMVN to identify noise and its parameters, which is a key problem in image denoising. The MLMVN network was learned to cover noise types like (Gaussian, salt & pepper, and uniform noise). In this proposed system, we used the continuous-valued MLMVN (further simply MLMVN) describe in [6] to solve both the noise and its parameters identification problems in order to overcome the disadvantages mentioned above. The modification of the MLMVN results in significant improvement of the functionality.

In the same time, the proposed system applied the proposed Least Square filter to calculate the new replacement pixels values. If the pixel is assigned as noise with the noise type, the proposed system will replace the pixel by the proposed filter result. Else, the pixel will leave without replace. The proposed filter and MLMVN network will apply to whole image. Finally, the proposed system recomposes the resulted image and calculates the RMSE measure to check the variation between the original image and resulted image.

The main proposed system steps are as follows:

- Apply the proposed mask on the image with windows of size 5x5 pixels (mask window) and calculated the average for each mask image.
- Apply MLMVN neural network on the image mask.
- Begin Least Square filter calculation depends on the image masks.
- If MLMVN decision id noise replace with result of the proposed Least Square filter.
- Calculate RMSE

Figure 2 shows the block diagram of the proposed system block:

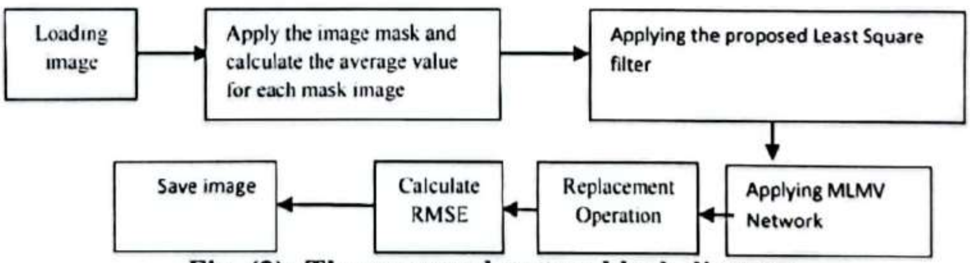


Fig. (2): The proposed system block diagram

In this proposed system, the RMSE measure was used to calculate the change amount in the noisy image after denoising operation was applied. The RMSE can calculate from:

$$RMSE = \sqrt{\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[\hat{X}(x,y)-X(x,y)]^2}{N^2}} \quad .....(14)$$

where $\hat{X}$, X represent the denoised and noise images, respectively.

MLMVN Neural Network Structure

As point above the MLMVN network used in the proposed system was explain in[6].But we provide some development on the input nodes number and the noise types due the increase the mask design properties and to increase the performance of the neural network. In this proposed system, we learned the neural network with the three types of noise with the there parameters.

The MLMVN has two hidden layers consisting of 10 and 25 neurons, respectively, and the output layer which consists of the same number of neurons as the number of classes, i.e. types of noise (3 types). Therefore, the structure of network is 25,10, 25, 3 (input layer, hidden layer 1, hidden layer 2, .output layer). Each neural element of the output layer has to classify a parameter of the corresponding type of noise, and reject the un-noised image.

The MVN activation function (eqn. 7) for the output layer neurons has a specific form: the equal subdomains (non-overlapping sectors) of the complex plane are reserved to classify a particular noise and its parameters and to reject un-noise image pixels.

7. Experimental Results & Discussions

In this research a Least Square interpolation calculation was used with Multilayer Neural Network Based On Multi-Valued Neurons to denoising images with a full noise or partial noise. The proposed filter and MLMVN network give very good results in removing the fully/partially noise from images. Also, a good RMSE values are resulted. Fig. (3) shows some proposed system results. The RMSE of the different samples denoising by proposed Least Square filter is compared in Table1.

For instance, the networks used have specific architecture with no universal learning algorithm, thus each neuron was trained separately. Another disadvantage is the use of too many spectral coefficients as features (quarter of image size). Thus the learning process was heavy.

Table 1 RMSE Results from applies the proposed Least Square filter

Sample name	RMSE (for Gaussian noise)	RMSE (for Salt &Pepper noise)	RMSE (for Uniform noise)	Size
S1	33.42	34.45	35.14	640x480
S2	36.18	37.34	34.55	600x800
S3	36.12	36.39	35.19	600x800
S4	38.89	39.10	39.21	512x 512
S5	36.13	35.95	36.85	512x512
S6	39.10	38.87	39.54	600x800

References

[1] Mohammed Ghouse, Dr. M. Siddappa," Adaptive Techniques Based High Impulsive Noise Detection And Reduction Of A Digital Image", Journal Of Theoretical And Applied Information Technology, 2011 Jatit & Lls.

[2] GIRISH BHOPALE," IMAGE NOISE REDUCTION USING MATHEMATICAL MORPHOLOGY SIZE DISTRIBUTIONS", Journal of Theoretical and Applied Information Technology, 2010 JATIT.

[3] Mahdipour Hossein-Abad Hadi, Khademi Morteza, Sadoghi Yazdi Hadi, "Least mean square algorithm tuned by fuzzy c-means for impulsive noise suppression of gray-level images", International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 3, No. 4, December, 2010

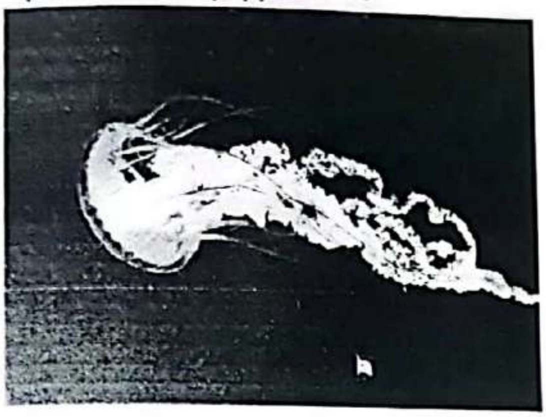
[4] Ce Liu, Richard Szeliski, Sing Bing Kang, C. Lawrence Zitnick, *and* William T. Freeman, "Automatic Estimation and Removal of Noise from a Single Image", IEEE Transactions on Pattern Analysis and Machine Intelligence October, 2006.

[5] Khalid Ali Hussien and Jolan Rokan Naif, " The Least Squares Interpolation Polynomial for Images", 2010

[6] Igor Aizenberg, Dmitriy Paliy, Claudio Moraga, and Jaakko Astola, "Blur Identification Using Neural Network for Image Restoration", Computational Intelligence, Theory and Applications 2006, Part 17, 441-455, DOI: 10.1007/3-540-34783-6_45.

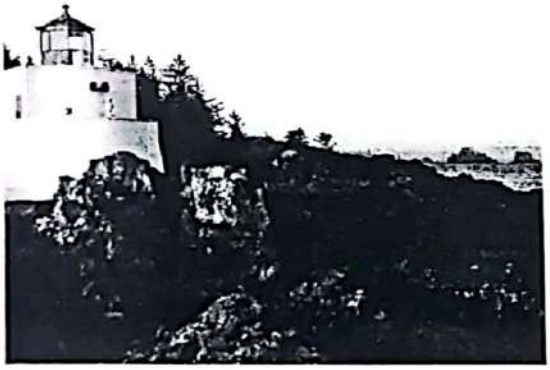
Sample1 –salt and pepper noise

Sample 1- After Denoising



Sample2(S2) -Guassain Noise

After denoising operation



Sample 3(S3)-Uniform noise

After denoising operation

Fig. 3 Some Test Images processed In the Proposed System.