Removal of Noise in an Image using Boundary Detection Technique

M. Sarvesh, M. Sivagami and N. Maheswari
School of Computer Science and Engineering, VIT, Chennai, India

E-mail: msivagami@vit.ac.in

Abstract. Image data recorded by devices contain mistakes or noises due to geometry and brightness values of the pixels. These mistakes are rectified using proper mathematical models. Examples of models include contrast and edge enhancement, pseudo-colouring, shadow removal, noise filtering, sharpening, and magnifying. Applying filters to remove a noise may lead to loss of some important data like edges in the image. The proposed work removes the noise from the image using local neighbourhood processing and preserves the edges using boundary-based approach. The noise filtered image is tested against the parameter entropy to validate the result.

1. Introduction
Noise is a random signal that distorts the original image. The noises come from sources in the vicinity of image capturing devices or imperfections in the image capturing device like misaligned lenses, weak focal length of camera, scattering etc. Noise in images is any random surge in brightness or intensity and is generally due to electronic noise. It can be introduced by the sensor and circuitry of a digital camera or scanner. It is an unwelcome side-effect of image capture that obscures the desired information. Noises are of many types and origins. Impulse noise is an unwanted, almost instantaneous sharp sounds. Salt and Pepper noise is a kind of Impulse Noise which may be caused due to faulty storage locations, improper functioning of pixel elements in the camera or scanner sensors, digitization errors. Image Processing accentuates the features of the raw image. It is done by working on a 2-D array of pixel values. The various techniques involved are mentioned below.

Image Pre-processing & Enhancement
In image pre-processing, noise can be removed for better image analysis. Normally the recorded image may have errors relative to brightness and geometry of the pixels. These errors are rectified by applying a proper mathematical model. Image enhancement also is an important image pre-processing approach. It enhances the visual appearance of an image by altering certain pixel values to accentuate certain pixel features. Edge Enhancement, Contrast Enhancement, Pseudo-Colouring, Sharpening are all examples of image enhancement techniques. These processes are useful in feature extraction and analysis of an image in an image processing application.

Contrast Stretching
Some images might not have varying intensity levels. They might only have few pixel values that form a narrow range. This may be due to poor illumination of the scene. Contrast Stretching is used to stretch the range of pixel values so that they better represent the image.

**Histogram Modification**

Histogram gives valuable information about the image. By modifying the histogram, image characteristics can be modified. Histogram equalization redistributes pixel values in such a way that every pixel occurs nearly the same amount of times in an image. This increases contrast at peaks and reduces contrast at tails.

**Noise Filtering**

Noise Filtering removes unnecessary information from the image. Some filters like low pass, high pass, mean, median etc. are used in filtering. There are different types of noises and each noise has a different mathematical model. Based on the model, different algorithms are created to tackle the problem of noise filtering. These models are described in the upcoming sections.

### 1.1. Types of Noise

There are many types of noise. Depending on the way they are distributed in an image, they are classified into the following. The following section is courtesy of [1].

**Poisson Noise**

Poisson noise is also known as Shot noise. It is a type of electronic noise which can be modelled by a Poisson function. The poisson function with parameter \( \lambda = f(x) \) is represented by

\[
P(f(x = k)) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (#1)
\]

**Exponential Noise**

This noise is a special case of erlang distribution. It is represented by,

\[
P(z) = \begin{cases} ae^{-z}, & z \geq 0 \\ 0, & z \leq 0 \end{cases} \quad (#2)
\]

**Salt & Pepper Noise**

An image having dark spots in bright regions and bright spots in dark regions is said to contain Salt and Pepper noise. If \( a = 0 \) (black) and \( b = 1 \) (white) then probability distribution is represented by

\[
P(z) = \begin{cases} ae^{-z}, & z \geq 0 \\ 0, & z \leq 0 \end{cases} \quad (#3)
\]

**Gaussian Noise**

The noise is a statistical noise and it has a probability density function which is equal to the normal distribution. This noise is a set of values taken from a Gaussian distribution that has zero mean and is added to every pixel. The distribution is represented as

\[
P(z) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (#4)
\]

Where \( \mu \) refers mean of random variable of \( z \), \( z \) refers grey level, \( \sigma^2 \) = variance of \( z \)

**Multiplicative Noise**

The noise is any random signal which will be multiplied into some other signal during the various activities like image capturing, sending or receiving the images or any other image processing activities. It is represented by

\[
\{g(n,m) = f(n,m) \ast u(n,m) + k(n,m)\} \quad (#5)
\]
where, \( g(n,m) \) denotes the observed image, \( k(n,m) \) is the additive component and \( u(n,m) \) is the multiplicative component of the noise. Here \( n \) and \( m \) are the axial and lateral indices of the image.

**Rayleigh Noise**
The Rayleigh noise in an image has Rayleigh distribution. The Rayleigh distribution is given below

\[
P(z) = \begin{cases} 
\frac{2}{b} (z-a) e^{-\frac{(z-a)^2}{2b^2}}, & z \geq a \\
0, & z < 0 
\end{cases} \quad \#(6)
\]

**Erlang Noise**
The Probability Distribution Function of Erlang noise is represented below

\[
P(z) = \begin{cases} 
\frac{a^b z^{b-1}}{(b-1)!} e^{-az}, & z \geq 0 \\
0, & z < 0 
\end{cases} \quad \#(7)
\]

The proposed work takes into account impulse noise. The different models for Salt and Pepper impulse noise is described below.

1.2. Detection of Impulse-Noise
Noise is modelled as salt and pepper impulse noise. The corrupted pixels has only two values, the maximum (255) or the minimum (0) for 8-bit images. The impulse noise in the image is modelled as 4 models and removed the noise[2].

**Model 1**
The pixels are corrupted randomly but their distribution is equal in the image and can be determined by a fixed probability given by the following sets of equations.

\[
f(x) = \begin{cases} 
\frac{p}{2}, & x = 0 \\
1-p, & x = l_{i,j} \quad \#(8) \\
\frac{p}{2}, & x = 255 
\end{cases}
\]

**Model 2**
Noise is like Model 1 but with unequal probabilities of salt (255) and pepper (0).

\[
f(x) = \begin{cases} 
\frac{p_1}{2}, & x = 0 \\
1-p, & x = l_{i,j} \quad \#(9) \\
\frac{p_2}{2}, & x = 255 
\end{cases}
\]

where \( p = p_1 + p_2 \) and \( p_1 \neq p_2 \)

**Model 3**
Impulse noise could be also modeled by two fixed ranges that will be at both ends with a length of \( m \) each, respectively instead of two fixed values.
Model 4 is similar to Model 3, but the densities of low-intensity impulse noise and high-intensity impulse noise are unequal. That is

$$f(x) = \begin{cases} \frac{p}{2m}, & 0 \leq x < m \\ 1 - p, & x = l_{i,j} \\ \frac{p}{2m}, & 255 - m < x \leq 255 \end{cases}$$

(10)

Where $p = p_1 + p_2$ and $p_1 \neq p_2$

The goal is to define a filtering algorithm for the above noise models.

1.3. Types of Filters

Linear Filters like the mean filter take the mean value of every window and update the central pixel with this value. This leads to smoothing of the image meaning the edges are not preserved. For edge preservation and more accurate noise filtering, non-linear filters like the median filter are used. Conventional Median Filters replace every pixel value of the image with the median of the working window matrix without considering corrupted or uncorrupted pixels. They are simple non-linear filters and mainly used to remove salt and pepper noise. The main drawback is they remove image details when the noise is more than 40%.

Wiener Filter uses a statistical based approach for filtering. For using this filter, the spectral property of the noise and original signal is to be known beforehand. It minimises the root mean squared error (RMSE) of the image. It is used to remove poisson and speckle noise.

Gaussian Filter is a filter that uses 2-D convolution operation to remove noise and blur from images. It is done by convoluting each point by a gaussian kernel and then summing all results to produce the output image. It works very well for gaussian noise.

2. Related Work

Pei-Eng Ng et al [2] proposed a Switching Median Filter with Boundary Discriminative Noise for Extremely Corrupted Images. Boundary Discriminative Noise Detection (BDND) algorithm takes into account a local window and classifies the pixels into three groups - lower intensity pixel noise, higher intensity pixel noise and uncorrupted pixel. The pixel under consideration (centre pixel of the window) is considered "uncorrupted" provided it falls within the boundaries defined. If not, it is considered corrupted. To determine the boundaries between the three groups, an algorithm is defined and applied to every pixel in the image. The boundary is defined based on the intensity differences in the sorted order in a vector. The maximum value of intensity difference between 0 and med in the difference vector is taken as b1 from the corresponding sorted vector and likewise the maximum value between med and 255 is taken as b2. The pixel is considered corrupted if it does not fall inside the range of b1 and b2. A binary matrix is formed which contains only 0’s and 1’s stating uncorrupted and corrupted respectively for the corresponding pixels in the image. The BDND process consists of two iterations. The first iteration takes a large local window (21 x 21) and examines the image for corrupted pixels. If the pixel value falls into the “corrupted” category, only then the second iteration is invoked. The second iteration takes a smaller local window (3x3) to further examine the pixel and confirm if it is corrupted. This work has been done and analysed with the four models of noise as mentioned in the above section.
Ms. P.H. Sangave et al [3] uses the same boundary conditions as above to classify corrupted pixels but uses different window sizes and modified the correction algorithm by using distance and weights of uncorrupted pixels from the center pixel.

Pragati Agarwal et al [4] talked about different types of Linear and Non-Linear filters and talks in detail about the advantages and disadvantages of different filters.

Prashant Dwivedy et al [1] compared the performance of different filters for different noises and talked about the mathematical models of different noises and filters.

K.S. Srinivasan et al [5] proposed “A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises”. A 3 × 3 window was taken, and it was sorted by row, column and right diagonal. If the processed pixel is not 0 or 255, it is considered uncorrupted. If it is corrupted, it is replaced with the median value. If median value is also corrupted (255), then the last processed pixel value is taken for replacement.

Debkumar Chowdhury et al [6] talked about the different noise models and noises.

M. S. A. Alias et al [7] proposed “Salt and Pepper Noise Removal by Using Improved Decision Based Algorithm”. If the value of pixel is either 0 or 255, the processing window is checked. If the window contains some noisy pixels or none of the pixels in the ‘+’ neighbors are noisy, remove the noisy pixels and median of the remaining pixels are taken as the replacement value. If processing window contains all noisy pixels, then midpoint of all pixels is taken.

P. Goyal et al [8] did Impulse noise removal with zero’s padding by median based adaptive filter. The boundaries of the image are padded with zeros and the local windows taken covers every pixel in the original image. Median values along row and column are then taken as boundary conditions for classifying pixels as corrupted or uncorrupted. Corrupted pixels are replaced by the median value of the window.

Anwarsha et al [12] analysed the variant of median filters impact in impulse noise removal. From literature survey, in most of the work median filter is applied for impulse noise removal and the results are good. So, the proposed work also uses the same median filter to remove the noises in the image. But the main area to be explored is in the definition of the boundaries for classifying noisy pixels. Thus, it is a very important part of the filtering process.

3. Proposed Method

The proposed work introduced the boundary based corrupted pixel detection and employed median filter for making the corrupted pixel as uncorrupted pixel for salt and pepper noise. The proposed model has two modules. They are

1. Noise Detection using boundary detection.
2. Noise removal using median filter.

The proposed methodology removes the noise in the image as a process of multiple iterations with these two modules. Every iteration takes into account a local window (3 × 3) and defines boundaries b1 and b2. b1 and b2 are defined based on contrast value of the window.

Contrast is defined as the difference between the maximum and minimum pixel value. b1 is defined as contrast/2 and b2 is defined as median+contrast/2. If the pixel value falls outside the range of b1 and b2, it is identified as ”corrupted” and the pixels which fall within the boundaries are termed as ”uncorrupted”. The binary matrix is obtained with corrupted pixels as 1 and uncorrupted pixels as 0. The corrupted pixel is changed by the median value of the window in the corresponding image and uncorrupted pixel values are left unchanged. Since median filter is proven to be one of the best filters for salt and pepper noise, median filter was chosen for the proposed algorithm.

The boundary detection b1 and b2 is calculated for every window of the image. Then the median filter of the corresponding window is applied on the corrupted pixels of the same window and hence the new image is obtained. The obtained output image of this step is now taken as the input for the next iteration and process is repeated. The termination criteria of this iterative process is determined using the entropy of the image. Generally, entropy serves as a measure of disorder. It relates the amount of uncertainty about an event with a given probability distribution. As the level of disorder rises, the entropy rises and events become less predictable. This process is terminated when the difference in entropy in three consecutive iterations is less than 0.001 meaning the events are now predictable and the filtering process can no longer produce a better result than the one obtained.
The proposed algorithm takes into account the maximum and minimum value of the local window and defines contrast and boundaries as

\[
\text{contrast} = \max - \min \quad (12)
\]

\[
b_1 = \frac{\text{contrast}}{2} \quad (13)
\]

\[
b_2 = \text{med} + \frac{\text{contrast}}{2} \quad (14)
\]

For each pixel value not in the range of \(b_1\) and \(b_2\), the binary matrix is updated with the value of 1 and the rest of the uncorrupted pixels are taken as 0. After traversing the entire image and updating the binary matrix, the corrupted pixels are updated with the median value of the local 3 \(\times\) 3 window. A new image is thus formed, and this image will be an input for the next iteration. The process is repeated until the entropy difference between consecutive iterations is less than 0.0001 for two or three times consistently.

3.1. Algorithm

1. Take a 3 \(\times\) 3 window centered on the current pixel of the image.
2. Sort the values of pixel elements in the window and obtain the median.
3. Obtain minimum and maximum value in the window.
4. Contrast is defined as maximum - minimum value.
5. Boundaries \(b_1\) and \(b_2\) are defined as \(b_1 = \text{contrast}/2\) and \(b_2 = \text{med} + \text{contrast}/2\)
6. If the current pixel falls within the boundaries \(b_1\) and \(b_2\), a binary matrix is updated with the value of 0. Else, the binary matrix is updated with the value 1 signalling corrupted pixel.
7. After traversing the entire image and forming the binary matrix, the binary matrix is traversed and the local window centered around the corrupted pixel is taken and the median value replaces the corrupted pixel value.
8. The process is repeated on the image until difference in entropy of consecutive iterations is less than 0.0001

3.2. Flowchart
Figure 1. Flowchart

The Figure 1 shows the diagrammatic representation of the proposed algorithm for noise removal.

4. Result Analysis
The proposed algorithm was tested on various test images obtained from reference [9] and their image quality 3 metrics were measured and compared with existing filters and algorithms. The tests were done using MATLAB R2018a in Windows 10 Environment. The following are some of the existing algorithms it was tested against.

4.1. Weighted Median Filter (WMF) [10]
In the weighted Median Filter, the corrupted pixel is replaced by the assigning different weights to the neighbourhood pixels. The new window is then sorted after convolution with different weights and the median value of that window is taken to replace the corrupted pixel. The results were obtained by taking only the '+' neighbours of the central pixel.

4.2. Adaptive Median Filter (WMF) [11]
The adaptive Median Filter starts with a fixed window size of $3 \times 3$. If the window size is deemed insufficient, 14 the window size is increased and the filtering process is executed. The median of the corresponding local window 15 centered around the corrupted pixel is calculated and used to replace the corrupted pixel. The boundary for checking window size is defined by

\[ a_1 = z_{med} - z_{min} \text{ and } a_2 = z_{med} - z_{max} \]  

If $a_1 < 0$ and $a_2 < 0$, go to step B. Or else, increase window size by 2. Maximum window size is $(7 \times 7)$. 

\[ (15) \]
STEP (B)

\[ b_1 = z_{xy} - z_{\min} \text{ and } b_2 = z_{xy} - z_{\max} \] (#16)

If \( b_1 > 0 \) and \( b_2 < 0 \), leave pixel unchanged. Or else, replace the pixel by \( z_{\med} \).

4.3. Boundary Discriminative Noise Detection (BDND) [3]

BDND uses an iterative procedure with 2 window sizes of 21 × 21 in the first iteration and 3 × 3 in the second iteration to classify the pixels. Boundaries \( b_1 \) and \( b_2 \) are defined as discussed earlier [3]. If the pixel under consideration is termed as corrupted in both the iterations, only then a correction algorithm to replace the pixel is implemented.

The two metrics used to validate the result are Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE).

Mean Square Error (MSE) as the name suggests, is a mean of the all the squared errors. A higher value of MSE suggests there is a lot of distortion in the image. Peak signal-to-noise ratio (PSNR) is a ratio between the maximum possible value of a signal (peak signal) and the value of the noise that affects it. Since many signals have a very high dynamic range of values, the PSNR is generally represented on the logarithmic decibel scale. A lower value of PSNR suggests there is a lot of noise in the image.

The mathematical formulae for the two for an \( M \times N \) image are

\[
MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x, y) - g(x, y)|^2 \] (#17)

\[
PSNR = 10 \log_{10} \frac{S^2}{MSE} \] (#18)

where

\[
S = \text{Intensity}_{\max} \] (#19)

The algorithm is repeated for multiple iterations until the difference in value of entropy between consecutive 4 iterations is consistently small and nearly constant.

\[
H(X) = -\sum_{i=0}^{255} P_i \log(P_i) \] (#20)

\[
P_i = \frac{N_i}{N} \] (#21)

where \( N_i \) refers the number of pixels with grey level \( i \) and \( N \) refers the total number of pixels of the image.

The proposed technique introduces the noise from 10% to 50% and the results have been obtained for the corresponding noises in the image. The value of PSNR of Lena Image has been taken from the research work[3] for the comparison of results.
### Table 1. PSNR of Lena Image.

| ND  | WMF  | AMF  | BDND [3] | PA  |
|-----|------|------|----------|-----|
| 10% | 36.06| 40.59| 42.29    | 41.13|
| 20% | 33.95| 39.69| 38.91    | 39.11|
| 30% | 32.16| 38.78| 36.54    | 37.75|
| 40% | 30.70| 37.68| 34.18    | 36.62|
| 50% | 29.55| 36.69| 31.12    | 35.61|

### Table 2. MSE of Lena Image.

| ND  | WMF  | AMF  | BDND [3] | PA  |
|-----|------|------|----------|-----|
| 10% | 16.11| 5.67 | 3.83     | 5.13|
| 20% | 26.18| 6.99 | 8.35     | 7.98|
| 30% | 39.52| 8.60 | 14.40    | 10.92|
| 40% | 55.35| 11.09| 24.80    | 14.17|
| 50% | 72.10| 13.94| 50.23    | 17.88|

### Table 3. PSNR of Cameraman Image.

| ND  | WMF  | AMF  | PA  |
|-----|------|------|-----|
| 10% | 37.68| 42.06| 43.72|
| 20% | 35.10| 41.00| 41.43|
| 30% | 33.08| 39.89| 39.84|
| 40% | 31.45| 38.69| 38.48|
| 50% | 30.31| 37.60| 37.19|

### Table 4. MSE of Cameraman Image.

| ND  | WMF  | AMF  | PA  |
|-----|------|------|-----|
| 10% | 11.09| 4.04 | 2.76|
| 20% | 20.10| 5.17 | 4.68|
| 30% | 32.00| 6.67 | 6.75|
| 40% | 46.58| 8.78 | 9.23|
| 50% | 30.31| 11.30| 12.43|

### Table 5. PSNR of Baboon Image.

| ND  | WMF  | AMF  | PA  |
|-----|------|------|-----|
| 10% | 33.40| 38.59| 38.32|
| ND   | WMF     | AMF     | PA     |
|------|---------|---------|--------|
| 10%  | 29.75   | 9.00    | 9.57   |
| 20%  | 40.18   | 12.10   | 15.02  |
| 30%  | 52.34   | 16.51   | 20.9   |
| 40%  | 66.66   | 21.74   | 27.51  |
| 50%  | 80.23   | 28.14   | 34.86  |

**Table 6.** MSE of Baboon Image.

ND- Noise Density, WMF-Weighted Median Filter, AMF-Adaptive Median Filter, BDND- Boundary Discriminative Noise Detection, PA-Proposed Algorithm

**Figure 2.** Original Lena Image, 10% Noise, 30% Noise, 50% Noise

**Figure 3.** 10% Noise Filtered Image, 30% Noise Filtered Image, 50% Noise Filtered Image
Figure 4. Original Cameraman, 10% Noise, 30% Noise, 50% Noise

Figure 5. 10% Noise Filtered Image, 30% Noise Filtered Image, 50% Noise Filtered Image

Figure 6. Original Mandrill, 10% Noise, 30% Noise, 50% Noise

Figure 7. 10% Noise Filtered Image, 30% Noise Filtered Image, 50% Noise Filtered Image

5. Conclusion
The proposed algorithm was tested for salt and pepper noise and the results were tabulated. The results of proposed approach for noise filtering in an image up to 50% noise density are satisfactory. For future work, more noise models could be explored and the filtering process could be extended to work for many more noise models.

References
[1] P. Dwivedy, A. Potnis, S. Soofi, and P. Giri, “Performance comparison of various filters for removing different image noises,” in 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Oct 2017, pp. 181–186.
[2] P.-E. Ng and K.-K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1506–1516, June 2006.

[3] P. H. Sangave and G. P. Jain, “Impulse noise detection and removal by modified boundary discriminative noise detection technique,” in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Dec 2017, pp. 715–719.

[4] P. Agarwal and J. S. Verma, “A survey of linear and non-linear filters for noise reduction,” in *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, Aug 2013.

[5] K. S. Srinivasan and D. Ebenezer, “A new fast and efficient decision-based algorithm for removal of high-density impulse noises,” *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 189–192, March 2007.

[6] D. Chowdhury, S. Gupta, D. Roy, D. Sarkar, C. C. Chattopadhyay, and S. K. Das, “A quantum study on digital image noises and their in-depth clusterization,” in *2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)*, Nov 2017, pp. 1–7.

[7] M. S. A. Alias, N. Ibrahim, and Z. M. Zin, “Salt and pepper noise removal by using improved decision based algorithm,” in *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, Dec 2017, pp. 487–492.

[8] P. Goyal, V. Chaurasia, and O. P. Meena, “Impulse noise removal with zero’s padding by median based adaptive 24 filter,” in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, Oct 2017, pp. 78–81.

[9] “Standard Test Images,” http://www.imageprocessingplace.com/root files V3/image databases.htm.

[10] D. R. K. Brownrigg, “The weighted median filter,” *Commun. ACM*, vol. 27, no. 8, pp. 807–818, Aug. 1984. [Online]. Available: http://doi.acm.org/10.1145/358198.358222

[11] P. Lei, “Adaptive median filtering,” in *Seminar Report, Machine Vision*, vol. 140, 2004.

[12] Anwarshah, Javed Iqbalbangash, Abdul waheedkhan,Imranahmed, Abdullahkhan, Asfandyarkhan, Arshadkhan “Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images” Journal of King Saud University – Computer and Information Sciences (2020) Available online https://doi.org/10.1016/j.jksuci.2020.03.007.