Multimedia Tools and Applications (2022) 81:31529–31552
https://doi.org/10.1007/s11042-022-12847-7

A systematic review of state-of-the-art noise removal techniques in digital images

Nishant Bindal[1] · **Rajanbir Singh Ghumaan**[1] · **Prateek Jeet Singh Sohi**[1] · **Nikhil Sharma**[1] · **Hemdutt Joshi**[1] · **Bharat Garg**[1]

Received: 2 January 2021 / Revised: 23 February 2022 / Accepted: 9 March 2022 /
Published online: 9 April 2022

Abstract

Digital Image processing is a subcategory of digital signal processing that lays emphasis on the study of processing techniques used for enhancement or restoration. De-noising of images corrupted with various types of noises falls into this category. De-noising is mainly performed to enhance the understandability of an affected image. Images captured with faulty equipment or being transmitted over long distances are highly prone to be depraved by impulse noise, so, various techniques are presented for removal of this noise from images. Each of the presented technique has its own merits and demerits. This paper presents a comprehensive comparative analysis of these techniques over a wide range of noise densities. All the filtering techniques are implemented in MATLAB and simulated with standard benchmark image data and qualitative metrics namely Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are evaluated and compared. Therefore, this paper presents a comprehensive comparative analysis of various state-of-the-art noise removal techniques.

Keywords Median filters · Mean filters · Salt and pepper noise · Impulse noise · Image restoration and denoising

✉ Nishant Bindal
nishant.bindal1999@gmail.com

Rajanbir Singh Ghumaan
ghumaan1999@gmail.com

Prateek Jeet Singh Sohi
sohipjss@gmail.com

Nikhil Sharma
nikhil.sharma4@yahoo.com

Hemdutt Joshi
hemdutt.joshi@thapar.edu

Bharat Garg
bharat.garg@thapar.edu

[1] Thapar Institute of Engineering and Technology Patiala, Punjab, India

1 Introduction

The corruption of an image with noise may lead to loss of features that otherwise may prove to be useful. Noise in an image may be introduced as a result of defect in image capturing instrument, faulty processing techniques, interference while transmission to large distances and so on. To visually distinguish whether an image has impulse noise present in it, the viewer should look for abnormally light or dark pixels. These light and dark pixels correspond to maximum (255) and minimum (0) values a pixel can acquire (In the case of 8-bit image, 0 and 255 are the minimum and maximum values possible). The process of removing these noises with the help various image processing techniques is generally referred as de-noising.

Over the course of time various state of the art de-noising filters for restoring images corrupted with salt & pepper noise are proposed. The main target of this review paper is to cater the reader the knowledge about various different filters presented and various categories that these filters can be clustered under. The major contributions of the papers are as follows.

- We have summarized the working techniques of most of the state-of-the-art filters.
- For better understanding by the reader, visual representation of selected benchmark images (these images were corrupted with salt and pepper noise at a density of 95%) filtered using various techniques are also given.
- All in all this paper servers to be a guide for bringing researchers upto speed with the major developments in this domain of salt and pepper noise removal that has happened in the last quarter century, from 1995 to 2021.

The remainder of the paper is organized as follows. Section 3 explains about salt & pepper noise, its effects and different filtering techniques. Section 4 presents the reader with the simulation environment and comparative analysis performed on selected filtering algorithms. This section presents comparative analysis to the reader using different tables and plots for these filter over a wide range of noise density, namely, low which is up to 30%, medium, between 30% and 90% and high, that is above 90%. At the end in Section 5 the reader is provided the conclusion derived from Section 4.

2 Evaluation parameters

For quality analysis of various filter, different quality assessment metrics are used. Broadly, these quality assessment can be subjective or quantitative. The subjective quality analysis is based on the capability of human eye to draw out structural information from a given image. However, the quantitative quality metrics utilizes mathematical expressions to compute the quality of a given image. The Mathematical class is simple to implement and provide a uniform predict quality metrics. In this method, mathematical formulas are applied to compute amount of noise in the noisy/de-noised image over the original image. In this paper, following metrics are used for quality analysis.

2.1 Mean squared error (MSE)

The mean squared error or mean squared deviation (MSD) is the average of square of difference between the processed and the accurate pixel values. The mathematical expression of MSE is given by (1).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{i,j} - y_{i,j})^2 \tag{1}$$

where x and y are original and corrupted/reconstructed images respectively, while M and N are the number of rows and columns in the image.

2.2 Root mean square error (RMSE)

The RMSE can be calculated as the root of the average of squared difference between the actual and the restored pixel value. It can also be calculated by taking square root of MSE.

$$RMSE = \sqrt{\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{i,j} - y_{i,j})^2} \tag{2}$$

2.3 Peak signal-to-noise ratio (PSNR)

The PSNR is simply a ratio between the maximum possible signal power and the power of noise signal that alters the quantitative representation of an image. PSNR is often deployed to measure the restoration accuracy of a technique. The PSNR can be thought of as an approximate perception of humans towards reconstruction quality. The mathematical expression of PSNR in decibel (dB) is given by (3).

$$PSNR = 10log_{10}(MAX^2/MSE) \tag{3}$$

where, MSE and MAX are the mean square error and maximum value of the signal, respectively.

2.4 Structural similarity index (SSIM)

As the name suggests SSIM is used for measuring structural similarity between restored/processed image and actual image. SSIM is the measurement of image quality done by taking original/uncompressed image as reference.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

where $\mu_x(\sigma_x)$ and $\mu_y(\sigma_y)$ are the mean (standard deviation) in the x and y directions, respectively. While the constant C_1 and C_2 are selected such that maximum (minimum) value of SSIM is 1 (0).

2.5 Image enhancement factor (IEF)

IEF is the ratio of MSE of actual and corrupted image to that of actual and reconstructed image as given in (5). It is an accurate measure to estimate enhancement in the quality of de-noised image.

$$IEF = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (\hat{y}_{i,j} - x_{i,j})^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (y_{i,j} - x_{i,j})^2} \tag{5}$$

Denotions used in (3) to (5):

M, N -dimensions of the image

$x_{i,j}$ - actual image without any form of noise
$y_{i,j}$ - de-noised image
$\hat{y}_{i,j}$ - corrupted image.
MAX - function that gives the maximum allowed pixel value of the image.

This paper mainly revolves around Impulse Noise and its de-noising methods. The impulse noise can also be called as data drop noise because it statistically drops the intensity of the pixel value either to maximum or minimum possible value. Hence, is also referred as "Salt and Pepper Noise". This paper tries to address the notable work by many researchers on impulse noise removal techniques.

3 Different salt & pepper noise removal filters

3.1 Pre-requisite concept

The term "window" or "kernel" refers to the neighbourhood of the pixel of predefined size. Usually, three window sizes are taken in filtering process: 3×3, 5×5 and 7×7. In de-noising, the noise effect reduces with considering higher window size but the image losses its features. At low noise densities small processing window can have better performance. In high noise density large window size gives better results but adds blurring effect on the images.

Filtering is a technique to enhance an image. In this technique, the pixel in restored image is calculated by performing some set of operations on the noisy as well as its neighbouring pixels. There are some operations that are applied in image filtering such as smoothing, sharpening, and edge enhancement. Filtering techniques can be performed in either spatial or frequency domain. The operations are applied directly on pixels of an image in spatial domain. On the other hand, in frequency domain, the filtering actions are performed by mapping the spatial domain into frequency domain using Fourier transform or any other transform of the image function. However, this paper presents various state-of-the-art spatial filtering techniques as illustrated in Fig. 1. The various filtering techniques for eliminating salt & pepper noise classified into following classes.

3.2 Linear filters

In these filtering techniques, the restored pixel is computed as a linear combination of the pixels values of specific sized window centred around the noisy pixel. These techniques are generally accompanied by blurring effect and have poor performance in removing noises such as signal dependent noise. The most common type of linear filter is mean or averaging filter discussed below.

3.2.1 Mean filter

Mean filter is one of the simplest filters. It is a linear filter based on the technique of averaging. The processing window is usually of a square shape. This filter computes the mean of the processing window around the central noisy pixel is computed and the central pixel value is replaced by the computed value. Average value of the corrupted image is calculated in the area defined by S_{xy} [1, 5, 6, 24].

$$f(x, y) = \frac{1}{MN} \sum_{(s,t)\in S_{xy}} g(s, t) \tag{6}$$

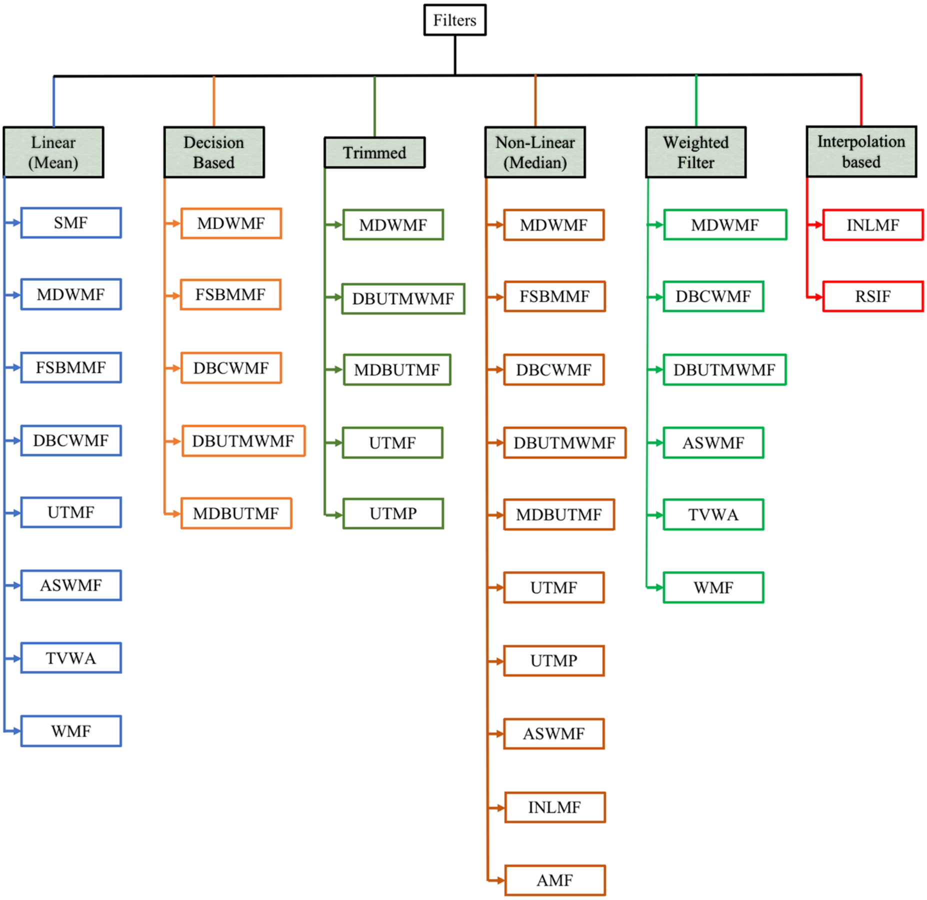


Fig. 1 Classification flowchart of various salt and pepper noise removal techniques

where, $f(x, y)$ is the de-noised image, g(x,y) is the original image of M×N dimension.

3.2.2 Weighted mean filter

This can be taken as the advancement in the simple mean filtering technique. The main difference between the mean filter and weighted mean filter is that, the specified pixels within a local neighbourhood are multiplied by their weights specified in the weight matrix [29, 32]. The mathematical expression of this filter is as follows:

$$f(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) g(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)} \tag{7}$$

3.3 Non-Linear filters

Several non-linear filters [2–4, 7, 10–12, 16, 18–20, 29, 31] are developed to provide a better performance over linear filters and overcome their shortcomings. As the name suggests, filters of this class do not satisfy superposition. This section provides a comprehensive review on various non-linear filters present in the literature.

3.3.1 Median filter

It can also be referred to as order statistics filter. It is the simple and most powerful non-linear filter. Like the mean filter, it also removes noise by smoothing the images. It replaces the pixel value of an image with the median value of the selected window or kernel. This also lowers the variation of intensity between different pixels of an image. Median of a given window/vector is calculated by, firstly arranging them in ascending order and then by choosing the middle value of the rearranged window/vector. If the number of pixels in the selected window/vector is even, then the median value is calculated by averaging the two middle pixel values. Following is the equation of this filter.

$$f(x, y) = Median[g(s, t)] \tag{8}$$

Where S_{xy} is the area over which the median is calculated. Median filters are more efficient in removing salt & pepper noise as compared to linear filters. The major advantage of median filters over linear filters is that, the former can effectively eliminate the effect of very large magnitudes of noise from the input image. Standard median filter is usually more efficient at lower noise density. At higher noise, increase in the size of processing window results in blurred output whereas small windows have poor noise suppression.

3.3.2 Switched median filter

This is a modification in the standard median filter. Noise detection algorithm used in the filter decides corrupted pixel, uncorrupted pixel and how the correction algorithm use median equation [12, 20, 29, 31].

3.3.3 Progressive switched median filter

This filter makes decision based on predefined fixed threshold value for detection of the pixel. The major challenge is to fix the threshold value for accurately detecting the noisy pixels. This filter cannot preserve the edges properly at high densities because this filter does not consider the local details while performing its operations [12, 20, 29, 31].

3.3.4 Trimmed median filter

This filter uses decision-based algorithms. It first uses 3×3 window for de-noising. If the processing window has no non-noisy pixels then central pixel is replaced by mean of the window, otherwise replaced by the median of the window [3].

3.3.5 Un-symmetric trimmed median filter

In this filter, the trimming is done on corrupted pixels on either side of the sorted window unsymmetrically. The median of the trimmed window is calculated and it replaces the processing pixel. Hence, it will not work if the trimmed window generated is empty, which mainly occurs at high noise density. Therefore, it does not work well at high noise densities and it also fails to preserve the edge details [11].

3.3.6 Min filter

In this filter, the pixel value is replaced by the value of the neighbourhood pixel having minimum intensity level. Due to its procedure, it is also known as the 0^{th} percentile filter. This means it fits the darkest pixels in the image.

3.3.7 Max filter

In this filter, the pixel value is replaced by the most brightest pixel in the neighbourhood of the corrupted processing pixel. It is also known as 100^{th} percentile filter. It removes pepper noise from an image.

3.4 Decision based algorithms (DBA)

At high noise density, the corrupted central pixel is replaced by the noise-free neighbouring pixel. Streaking effect also arises when repeated replacement with the neighbourhood pixel occurs [3, 4, 11, 26].

3.4.1 Improved DBA

This filter is designed to avoid streaking effect observed in DBA. Here, value of the corrupted central pixel is calculated by computing the median of the unsymmetrically trimmed processing window. However, the filter has poor or very low efficiency at high noise density.

3.5 Adaptive filters

As the name suggests, the behaviour of these [1, 2, 12, 16, 19, 21, 32]. type of filters varies with the statistical characteristic variation of the image region as well as the filter region. These filters are hard to implement and give better performance than other filters. The steps involved in this de-noising algorithm are:

1. Analysis: In this step, similar image blocks are collected in groups.
2. Processing: The obtained groups are filtered by thresholding.
3. Synthesis: The filtered blocks will then be used for the final image reconstruction that is computed by the operation on the filtered blocks.

3.5.1 AMF

Most commonly used adaptive filter is the “Adaptive Median Filter” [16]. It preserves the details of an image while filtering the image. It also performs well on images containing high density salt and pepper noise. One of the interesting feature of this filter is that it can change its window size during its operation depending on predefined conditions. The steps involved in this filter are as follows:

1. It calculates extreme values of the noise affected window.
2. Checks whether the calculated median itself is not an extreme value ($i.e.$ Salt & Pepper noise).
3. If the median value is also a corrupted value, then the process is reiterated with the increased window size and median is calculated. The window size can be increased to predefined max window size. Otherwise, move to the next step.
4. Checks whether the value of the processed pixel is corrupted or not. If corrupted, then replace that pixel with previously calculated median otherwise the pixel remains unchanged.

The AMF works well at low as well as medium noise densities. It introduces blurring effect in the image as it uses variable window size which is sometimes large that leads to blurring effect in the reconstructed image.

3.5.2 ARMF

The authors proposed the Adaptive Riesz Mean Filter (ARMF [8]). The main concept used is pixel similarity for pixel de-noising. It performs operation on image after converting it into double form. It uses Riesz mean based on the similarity. It performs better than other different similarity or distance functions (such as p-norm, Euclid and Hamming). It is also optimised to do all these operations. There is also a scope of improvement by using noise detection mask or another mean and using the same pixel similarity concept.

3.5.3 NRSMMF

It [25] is a unique technique which chooses operations to be performed according to the noise density of the image. Basic operations are: Mean, Median, Pre-processed values. Efficient in restoring images corrupted by very high density noise. It uses three noise density ranges. For noise density less than equal to 25%, it checks the series of 4 conditions for restoring pixel value. If the noise density is less than 85% and greater than 25%, it calculates the mean of previously processed pixels in 5×5 window. Else if the noise density is greater than 85%, calculate the mean of previously processed pixels in 5×5 window. This filter makes accurate decision according to the information in particular window.

3.5.4 AWM3F

The motive of development of this [23] algorithm was to filter very high density SAP noise(i.e., greater than 85%). This algorithm uses symmetric padding and adaptive window size concept. In the first step, this algorithm tries to select the window size in which there are enough non-noisy pixel for de-noising. The basic operation is to find: minimum, maximum and middle pixel values in the window and also finding out their weights. Here, two highly correlated groups of noise free pixels are computed using minimum and maximum value of the chosen window. If not found, its size is increase by one. Maximum size can be 7×7.

3.5.5 AFMF

This [9] filter can be considered as a development in the Adaptive Median Filter (AMF). It computes the frequency median instead of amplitude median to restore the pixel values of the noisy pixel. This modification produces better results than AMF because the restored pixel value is very closer to the original pixel value. The reason can be the operations performed in the frequency domain. This will exclude noisy pixels from evaluating the restored pixel value and focusses on the uniqueness of the restored value.

3.6 Combinational filters

These filters utilize combined linear and non-linear methods to restore the noisy pixels. Based on the noise density or the availability of the noisy free pixels in the processing window, these filters use either linear approach such as mean, interpolation etc. or use non-linear techniques such as median for de-noising. The following subsections discuss various combinational filters proposed in recent past.

3.6.1 Modified decision based un-symmetric trimmed median (MDBUTM) filter

In this filter, median of non-noisy pixels in the processing window is calculated and the corrupted processing pixel is replaced with calculated median [11]. Further, if all the pixels values in a window are corrupted, then mean of the processing window is calculated to replace central corrupted pixel. However, this method leads to the poor performance of the filter because this is equivalent to assigning any random value to the corrupted pixel rather than the value ‘0’ or ‘255’.

3.6.2 Decision based coupled window median filter (DBCWMF)

This filter chooses the window size according to some specified conditions. The strategy used to find the value of corrupted pixel is median. If there is any uncorrupted pixel in the processing window of size 3×3, then the corrupted pixel is replaced by the median value of all non-noisy pixels present in the given processing window. If there are no noise-free pixels available, then the window size is gradually (stepwise) increased to maximum of 9×9 for median calculation. But, if there is still no uncorrupted pixel in the 9×9 window then mean of all pixels in the 9×9 window is calculated for replacing noisy pixels. The main reason for better performance of this filter is its variable window size. In this way, it provides better quality metrics than MDBUTMF [11]. However, the large window size is also responsible for the blurring effect.

3.6.3 Noise density sensitive mean-media filter

A noise density range sensitive (NDRS) filtering algorithm is presented in [25] where the noisy pixel is replaced by mean, median or pre-processed values based on value of noise density. The NDRS filter provides a unique approach restoring corrupted pixels even at very high noise densities. A four stage median filtering algorithm (FSMA) is presented in [15]. The FSMA algorithm performs median filtering with smaller window size, large window size, running average and replacement by previously processed pixels at first, second, third and fourth stages, respectively. The algorithm moves to next stages only when the current stage fails to de-noise the candidate noisy pixel. The FSMA algorithm can effectively de-noise the image by considering the first two stages at low noise density while it may requires all stages at higher noise density. This filter eliminates SAP noise along with better edge preservation as it considers noise-free pixels while estimating the value of noisy pixels.

3.6.4 Fast switching based mean-median (FSBMM)

This filter is another state-of-the-art filter which uses two strategies - mean and median in a combinational way [29]. The unprocessed pixels are replaced by the mean of previously processed pixels. This is a very good approach for better performance of the filter but there is a problem with this filter is that when the boundary pixels are left unprocessed by the median strategy then the value is replaced by the value of previously processed pixel in their respective row/column. Therefore, under high noise density, there will be replication of the pixels resulting in low image quality. So, this filter has low performance under high noise density conditions.

3.6.5 Un-symmetric trimmed mid-point filter (UTMP)

This filter is an extension of the DMF filter and is noise-density range specific filter. In this filter, the median strategy is used to evaluate the value of the de-noised pixel and this performs well under the high noise density condition.

3.6.6 Un-symmetric trimmed median filter (UTMF)

It is also an extension of DMF filter to achieve noise density range specific filtering. This filter evaluates the value of the de-noised pixel using mean strategy which has better de-noising performance at lower noise densities. But this filter fails to adapt at every noise density range.

3.6.7 Adaptive trimmed filters

An adaptive trimmed median (ATM) filter is presented in [13] can effectively de-noise image corrupted with high noise density. This filter restores noisy pixel by computing median of noise-free pixels of adaptive size window and by computing via interpolation based procedure at low and high noise densities, respectively. Further, this algorithm de-noise candidate pixel using nearest processed pixel procedure for the rare scenarios especially at the boundary where de-noising using interpolation is not good enough. A multi procedure min-max average pooling based filter (MMAPF) is presented in [22]. In this filter, after preprocessing, the noisy image divided into two parts and passed through multiple layers of max and min pooling. Finally, the algorithm combines the processed images from the previous procedures and performs average pooling to remove all residual noise. An adaptive min-max value based weighted median (AMMWM) filter is presented in [14]. This filter first computes two highly correlated groups of noise-free pixels using minimum and maximum value of the current window and then determines weighted medians of these groups to restore the noisy pixel. The filter increases the window size if the current window fails to provide any noise-free pixels. However the filter can restores noisy pixel at high noise density, it exhibits high computational complexity.

3.6.8 Recursive spline interpolation filter (RSIF)

The RSIF filter [28] is similar to MDBUTMF [11] but this filter takes the median of uncorrupted pixels using cubic spline interpolation technique. Since, cubic spline interpolation is a mathematically superior technique, it produces better result in low and medium noise density ranges. However, this filter fails to produce good results under high noise density conditions due to lack of information for estimating the de-noising value. There is one more constraint that is to be satisfied to make the filter work is that there should be at least two uncorrupted pixels available in the window for applying interpolation which are rarely available at high noise densities. Another major drawback of this filter is that it requires more execution time.

3.6.9 Adaptive switching weighted median filter

This filter uses mainly two conditions to detect the noisy pixel $i.e.$ the pixel in the window should either be '255' or '0' whose window mean is not equal to '255' and '0' respectively [12]. It also uses the concept of variable window size. If the number of non-noisy available pixels in the processing window of size 3×3 is odd then the corrupted pixel is replaced by median of the trimmed window. Otherwise, the non-noisy pixel with maximum occurrence or the nearest non-noisy pixel (if there is no single value with maximum repletion available) in the window is repeated and the corrupted pixel is replaced by the median of the resultant window. If the entire 3×3 window is corrupted, then the above process is performed with a processing window of size 5×5. There is a major problem in this algorithm that leads to streaking effect under high noise density as the nearest uncorrupted pixel is used for median calculation, mostly leading to direct substitution.

3.6.10 Three-valued weighted filter (TVWA)

The TVWA filter [17] is another weighted filter. This filter initially takes a 3×3 window and divides all the noisy pixels into three groups on the basis of their correlation with the middle, maximum and minimum pixel values. The weighing factor used here is variable according to the chosen window and determined by the length of these three groups. After that, the corrupted pixel is replaced by the weighted mean of the window. Since, this algorithm uses variable window size, so if the 3×3 window is completely corrupted, then the window size is increased to 5×5 with the processing corrupted pixel as the central pixel and the above process is repeated. The maximum window size is 7×7. The main problem of this filter is the blurring effect due to large window size.

3.6.11 Iterative non-local median filter (INLM)

This filter uses a non-local weighted mean approach combined with the iterative approach to find the value of the corrupted pixel [30]. The algorithm works in three steps. First, create a matrix having locations of noisy pixels. The values '0' or '1' in this matrix denote the noisy pixels and noise-free pixels respectively. The noisy pixels are filtered through this matrix called N_{Map}. The strategy used to filter is the switching based median filter. Here, patch making is done for noisy pixels with the help of N_{Map}. On the basis of earlier respective noisy pixels with the similarities are patched together for INLM processing [30].

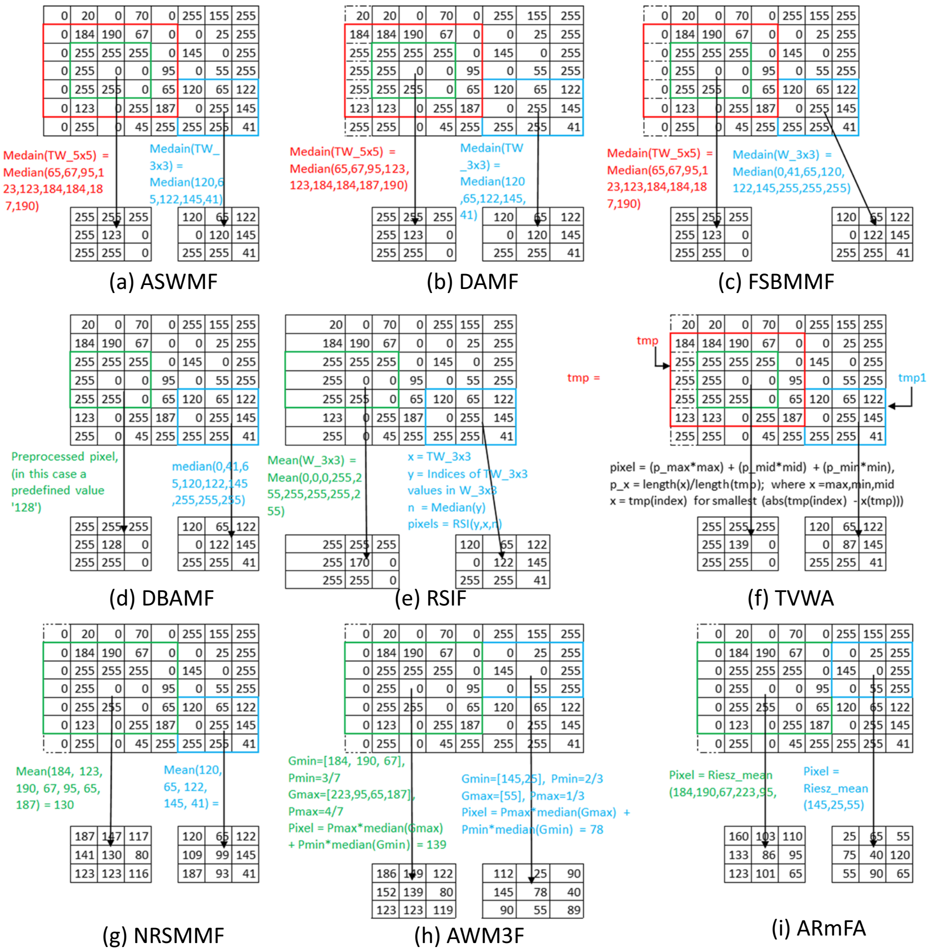


Fig. 2 An illustration of de-noising using (a) ASWMF, (b) DAMF, (c) FSBMMF, (d) DBAMF (e) RSIF (f) TVWA, (g) NRSMMF, (h) AWM3F, (i)ARmF filters. *Operations in black are for both arrow indicators, ^ value has been rounded to nearest whole number

3.6.12 Decision-based coupled window median filter

This filter uses variable window sizes to remove noise from the image [4]. It is a unique approach for high density saturated impulse noise removal. This filter uses the following steps for noise removal. At first, it checks whether the pixel is corrupted or not. Then, a 3×3 window is taken and the noisy pixel is replaced by the median of the window excluding the corrupted pixels that have value '0' or '255'. If the processing window has no noise-free pixel then the window size is increased using the method $2n+1$ where n is incremented from '1' to '4'. The value of the corrupted pixel is replaced by the mean of median of the window depending on the type of noise present in the window. Since, it uses the concept of variable window size, its performance is better than DBUTMF [11] in high noise density range. Due to variable window size the quality of the image is better at high noise density. This filter has

Table 1 Mathematical formulation of various filters for removing salt and pepper noise

Filters	Equation	Explanation
FSBMMF	$\begin{cases} S_{med}, \text{if} X_{min} < S_{med} < X_{\max} \\ X_{i,j-1}; \text{if i=1 or i=N} \\ X_{i-1,j}; \text{if j=1 or j=N} \\ mean(\text{four PPP in} S_{i,j}) \end{cases}$	• S_{med} is the median of $S_{i,j}$, where $S_{i,j}$ is 3 × 3 windows $(median(W^{nf}))$ • $X_{i,j}$ is the central pixel of $S_{i,j}$, W^{nf} is the noise free processing window • PPP:- Previously processed pixels.
DBUTMWMF	$\begin{cases} W_{mean}(TW_{i,j}); \text{ if} W_{i,j} \text{has NNP} \\ mean(W_{i,j}); \text{Otherwise} \end{cases}$	• W_{mean} is winsorized mean, $TW_{i,j}$ is trimmed processing window • $W_{i,j}$ is 3 × 3 processing window
MDBUTMF	$\begin{cases} median(TW_{i,j}); \text{if} W_{i,j} \text{has NNP} \\ mean(W_{i,j}); \text{Otherwise} \end{cases}$	• NFP - Noise free pixels • $W_{i,j}$ is processing window of size 3 × 3 • $TW_{i,j}$ is trimmed processing window of size 3 × 3 • NNP :- Non noisy pixel(s).
UTMF	$\begin{cases} mean(TW_{i,j}); \text{ if} W_{i,j} \text{has NNP} \\ \text{PPP; Otherwise} \end{cases}$	• $TW_{i,j}$ is trimmed window of size 3 × 3 • PPP:- Previously processed pixel
UTMP	$\begin{cases} median(TW_{i,j}); \text{if} W_{i,j} \text{has NNP} \\ \text{PPP; Otherwise} \end{cases}$	• TW_3 is trimmed window of size 3 × 3 • PPP:- Previously processed pixel
ASWMF	$\begin{cases} median(Z) \text{if} W_{3\times3} \text{has NNP} \\ median(Y) \text{if size of Z even} \end{cases}$	• Z: Odd sized vector from Trimmed window of initial size 3 × 3, with horizontal and vertical values repeated two times, Y: Vector having most repeated values in Z(if any), else nearest NNP. • If $W_{3\times3}$ has no NNP then size is increased to 5 × 5 anthe pixels in nearest + direction are repeated four times, in W direction three times and others two times. and converted to vector Z.
TVWA	$(p_{max} \times max) + (p_{mid} \times mid) +$ $(p_{min} \times min)$	• p_x = length(x)/length(tmp); where x = max, min, mid • x = tmp(index) for smallest (abs(tmp(index) - x(tmp))) • max, mid and min are the maximum, minimum and middle values in tmp where tmp is the trimmed 1D vector of the processing window
INLMF	$\frac{\sum_{k,l\in\Omega_{i,j}} w_{i,j,k,l}^{(t)} Y_{k,l}}{\sum_{k,l\in\Omega_{i,j}} w_{i,j,k,l}^{(t)}}$	$w_{i,j,k,l}^{(t)} = \exp \frac{\lVert P(X^{(t-1)}) - P(Y_{k,l}) \rVert_2^2}{h^2}$ • $X_{i,j}$ is the input image initialised using prefiltered results

Table 1 (continued)

Filters	Equation	Explanation
DBCWMF	$\begin{cases} median(TW_n) \text{if} W_n \text{has NNP } \forall n < 5 \\ mean(W_1);\ \text{Otherwise} \end{cases}$	• TW_n is trimmed processing window of size (2n + 1)×(2n + 1) • W_n is processing window of size (2n + 1)×(2n + 1) • NNP:- Non noisy pixel(s).
MDWMF	$DWM(TW_{7\times7})$	• DWM: Directional Weighted Median Operation. • Total number of directions possible w.r.t center pixel are twelve • $TW_{7\times7}$ is trimmed window of size 7 × 7.
Mean Filter	$\frac{1}{mn}\sum_{(s,t)\in S} g(s,t)$	• g(s,t) is the original image
Weighted Mean Filter	$\frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)*g(s+x,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$	• g(s,t) is the original image
Adaptive Median Filter	$median(W_{i,j})$	• $W_{i,j}$ is the processing window of user defined size (most probably 3×3)
RSIF	$P_{i,j} = \begin{cases} RCSI(y,x,n) \\ mean(W_{3\times3}), \text{if} W_n \text{has no NNP} \end{cases}$	• NNP:- non noisy pixel(s), RSCI:- Recursive Cubic Spline Interpolation, • y = indices of pixels in x, x = non noisy pixels in $W_{3\times3}$, • $W_{3\times3}$ is the processing window of size 3 × 3

some limitations. Increasing the processing window size, the blurring or smudging effect is introduced in the final image. But performs better than other filters in its class because of dynamically increasing the processing window size for processing the corrupted pixel.

3.6.13 Decision based unsymmetrical trimmed modified winsorized mean filter

This filter removes the salt and pepper noise by using the mean of the trimmed and a sorted 1D array [27]. Firstly, a 3×3 window is selected with the noisy pixel at the center, it is converted into a sorted 1D array. At last, the noise is removed from the latter array. This results in two cases being generated, in first case the some of the pixels the resultant array are left (the unsymmetrical winsorized operation is applied by replacing the smallest and the largest values from the trimmed window) and the processed noisy pixel is replaced by the mean of the array obtained after the unsymmetrical winsorized operation hence eliminating the noise from image. In the second case the remaining array point leads to null value, *i.e.* the initial window had no noise-free pixel and to accommodate this case the mean value of the initial window is taken and is placed at the central pixel (processing pixel) of the processing window selected. At high noise densities $\geq 95\%$ the performance of the filter reduces as more number of selected windows will be completely noisy and hence there mean will be just a random value. Figure 2 illustrates the denoising process of various filters.

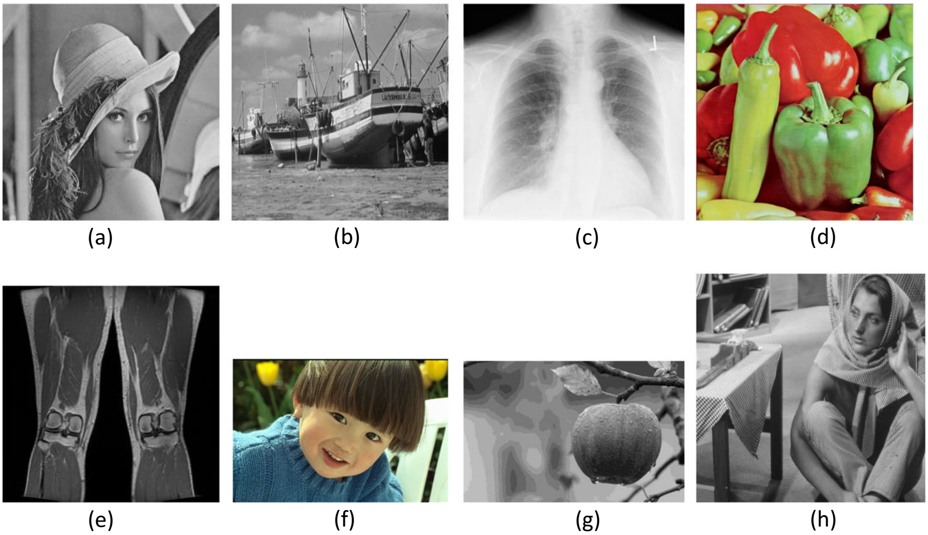


Fig. 3 Benchmark images used in simulation for qualitative analysis.* * Kodak images considered for simulation are not shown here

3.7 Filter selection

The aforementioned filtering techniques and their mathematical formulations are summarized in Table 1. These filters are selected after a detailed analysis of various researches in this field. Basic criteria of selection are listed as follows:

- Complexity of the method (Basic to advance mathematics).
- Time complexity.
- Simulation results based on different performance metrics.

The specific filters used in this study had shown remarkable simulation results on varying noise densities.

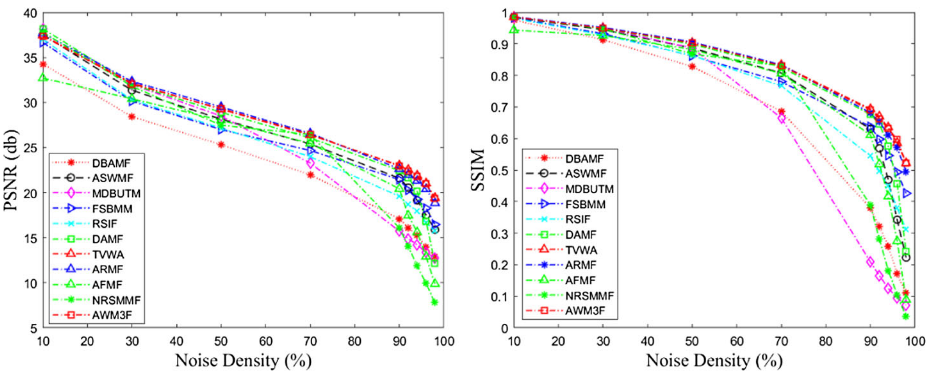
The next section presents a comparative analysis of these state-of-the-art salt and pepper noise removal techniques.

4 Simulation environment and comparative analysis

The following subsections give the qualitative and quantitative analysis of some state-of-the-art salt & pepper noise removal techniques on different benchmark images at varying noise densities. All the existing filtering techniques are implemented in MALTAB and simulated with benchmark inputs. Initially, the SAP noise of varying noise density (wide range: 10% to 90% and very high noise density range: 91% to 99%) is added in these benchmark images and then resulting noisy images are filtered via these filtering techniques and the quality metrics are extracted. Finally, both quantitative and qualitative analysis is done based on the extracted quality metrics and reconstructed images, respectively. Some benchmark images

Table 2 Average PSNR and SSIM of filtered Lena image using exiting filters at different noise density levels

Metrics	Filters/Noise	10	30	50	70	90	92	94	96	98
PSNR	DBAMF	34.27	28.43	25.31	21.98	17.1	16.11	15.28	13.98	12.91
	ASWMF	37.53	31.35	28.15	25.39	21.6	20.55	19.14	17.51	15.88
	MDBUTM	38.19	31.98	28.49	23.28	15.73	14.89	14.21	13.37	12.75
	FSBMM	36.66	30.14	26.99	24.67	21.44	20.27	19.26	18.27	16.46
	RSIF	36.99	30.3	27.15	23.99	19.59	18.66	17.93	16.68	15.72
	DMAF	38.19	32.01	28.91	26.07	22.44	21.67	20.15	16.84	12.13
	TVWA	37.45	32.16	29.24	26.4	23.02	22.54	21.91	21.04	19.42
	ARMF	37.68	32.34	29.5	26.58	22.64	21.99	21.31	20.4	18.83
	AFMF	32.7	30.41	28.02	25.46	20.42	17.46	15.61	12.91	9.87
	NRSMMF	37.8	31.79	27.48	26.44	16.08	14.01	11.87	9.93	7.79
	AWM3F	37.36	32.06	29.31	26.46	22.95	22.39	21.8	21	19.37
SSIM	DBAMF	0.9755	0.9129	0.8282	0.6858	0.3799	0.3211	0.2591	0.171	0.1103
	ASWMF	0.9844	0.9445	0.8867	0.8075	0.6303	0.5689	0.469	0.3425	0.2228
	MDBUTM	0.9858	0.949	0.8869	0.665	0.2086	0.1652	0.1251	0.0948	0.0718
	FSBMM	0.9829	0.9318	0.8617	0.7799	0.6379	0.5977	0.546	0.4934	0.4263
	RSIF	0.9807	0.9282	0.863	0.7667	0.5444	0.4961	0.4504	0.3793	0.3116
	DMAF	0.9858	0.9492	0.8987	0.8264	0.676	0.6456	0.5756	0.4576	0.2419
	TVWA	0.9851	0.9507	0.9032	0.833	0.6936	0.6698	0.6352	0.5859	0.522
	ARMF	0.9862	0.9519	0.9059	0.8336	0.6811	0.6519	0.6114	0.5702	0.4952
	AFMF	0.9435	0.9263	0.8821	0.806	0.6114	0.5184	0.4167	0.2746	0.0891
	NRSMMF	0.9855	0.9475	0.8681	0.8295	0.3884	0.2811	0.182	0.1046	0.0354
	AWM3F	0.9853	0.9495	0.903	0.8304	0.6898	0.6635	0.6311	0.597	0.5218



(a) (b)

Fig. 4 PSNR and SSIM value plots of Lena image filtered using various filters at different noise density levels

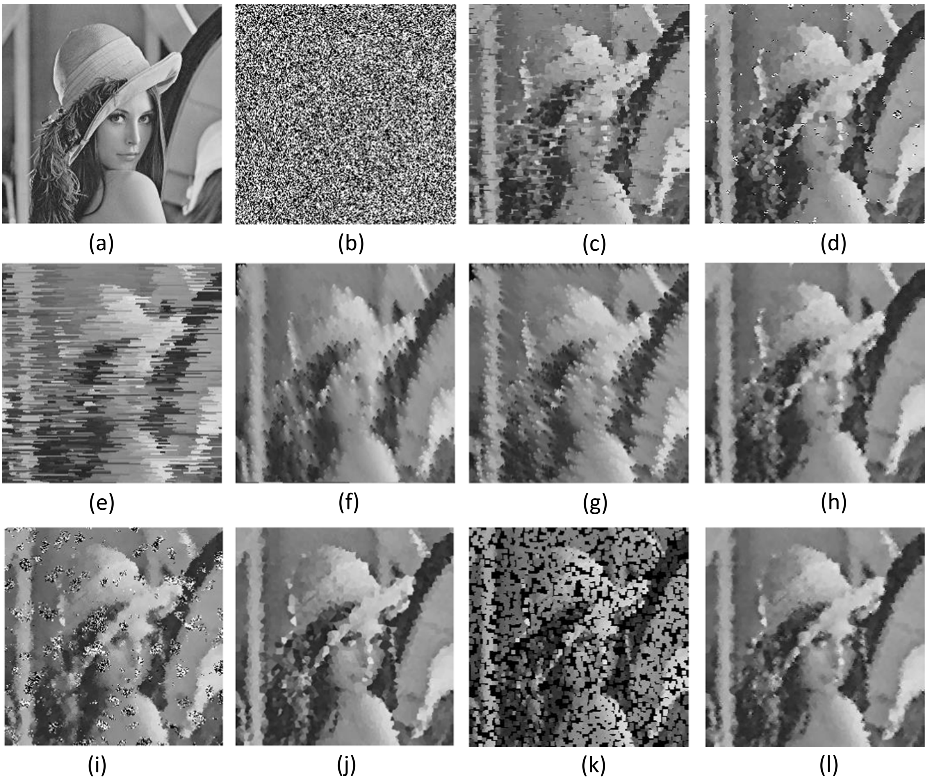


Fig. 5 (a) Original Image (b) Noisy Image (95%) (c) ASWMF (d) DAMF (e) DBAMF (f) FSBMMF (g) RSIF (h) TVWA (i) AFMF (j) ARMF (k) NRSMMF (l) AWM3F

considered for analysis are in Fig. 3 along with Kodak (set of 24 images) images. Simulation results of the following images are discussed:

- Grey scale Lena image.
- High Resolution Apple image.
- Medical image lungs.
- Kodak Images.

4.1 Simulation result analysis with grey scale images

The Table 2 encapsulates the average PSNR and SSIM values for different state of the art filters for Lena image (256×256) with the noise density varying from wide range (10% to 90%) to very high range(92% to 98%). From the quality metrics, it can clearly observed that the AWM3F [23] has best noise removal capabilities as compared to other filters analysed.

For better comparative analysis, Fig. 4 provides the plots for Table 2. Accordingly it can be noted that ARmf [8] has better noise removal at low noise density and as the noise density increases the performance of FSBMMF [29] rises to peak at medium noise density, and afterwards is clearly over-taken by AWM3F [23] at higher noise densities.

Further for the qualitative analysis, the filtered images using different techniques (corrupted with noise density of 95%) are shown in Fig. 5. It can be observed from these filtered

Table 3 Average PSNR and SSIM of filtered apple images using exiting filters at different noise density levels

Metrics	Filters/Noise	10	30	50	70	90	92	94	96	98
PSNR	DBAMF	43.63	37.72	34.25	31.18	26.35	25.62	24.54	23.03	20.56
	ASWMF	46.08	40.24	36.79	33.82	28.74	26.88	24.17	20.8	17.35
	MDBUTM	46.74	40.88	35.37	25.89	16.57	15.73	14.9	14.07	13.29
	FSBMM	45.5	38.84	35.31	32.65	28.63	27.84	26.89	25.4	23.23
	RSIF	44.63	37.77	34.63	31.96	28.41	27.68	26.86	25.51	23.48
	DMAF	46.74	40.91	37.62	34.72	30.29	28.72	25.1	19.33	12.52
	TVWA	46.87	41.41	38.21	35.06	31.09	30.51	29.88	28.91	27.05
	ARMF	47.61	42.07	38.68	46.87	43.57	41.41	39.75	38.21	36.7
	AFMF	42.08	39.51	36.66	33.62	23.5	20.28	16.91	13.76	10.19
	NRSMMF	46.68	40.68	32.53	34.66	17.21	14.9	12.64	10.38	8.14
	AWM3F	46.86	41.43	38.24	35.07	31.04	30.52	29.85	28.86	27.11
SSIM	DBAMF	0.9996	0.9979	0.9939	0.9828	0.9185	0.9001	0.8695	0.8137	0.7053
	ASWMF	0.9998	0.9988	0.9968	0.992	0.9394	0.8818	0.7544	0.5518	0.3274
	MDBUTM	0.9998	0.9989	0.9938	0.8948	0.3959	0.3387	0.2756	0.2149	0.1596
	FSBMM	0.9997	0.9981	0.9949	0.987	0.9366	0.9204	0.897	0.8553	0.7817
	RSIF	0.9997	0.9981	0.9949	0.9858	0.9373	0.9227	0.9001	0.8623	0.7899
	DMAF	0.9998	0.9989	0.9973	0.993	0.9667	0.9526	0.9072	0.7386	0.3819
	TVWA	0.9998	0.9991	0.9975	0.9933	0.9688	0.9618	0.9505	0.9316	0.8888
	ARMF	0.9998	0.9992	0.9978	0.9937	0.968	0.9602	0.9486	0.9279	0.8905
	AFMF	0.9993	0.9985	0.9966	0.9904	0.8782	0.7849	0.63	0.4491	0.2601
	NRSMMF	0.9998	0.9988	0.9843	0.9921	0.4366	0.3179	0.219	0.1394	0.0752
	AWM3F	0.9998	0.999	0.9976	0.9934	0.9682	0.9609	0.9499	0.93	0.8913

images that image filtered using AWM3F [23] has preserved the more face details as compared to others. Hence we can conclude that AWM3F [23] has best performance over grey scale Lena image.

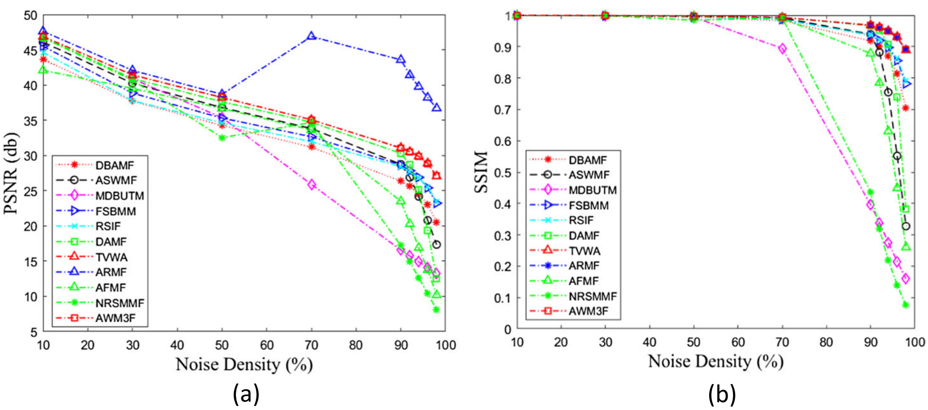


Fig. 6 PSNR and SSIM plots of high resolution apple image filtered via various filters at different noise density levels

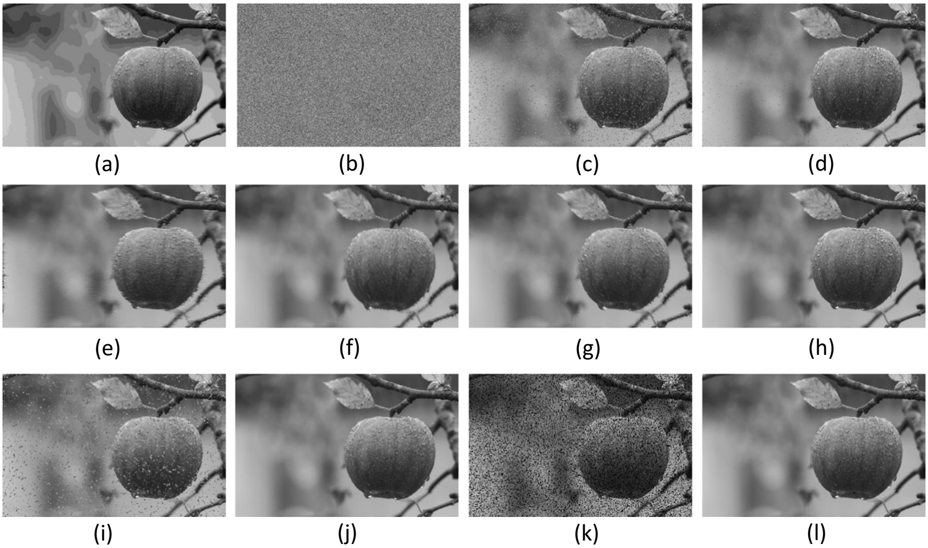


Fig. 7 (a) Original Image (b) Noisy Image (95%) (c) ASWMF (d) DAMF (e) DBAMF (f) FSBMMF (g) RSIF (h) TVWA (i) AFMF (j) ARMF (k) NRSMMF (l) AWM3F

Table 4 PSNR and SSIM of filtered Kodak images using exiting filters at different noise density levels

Metrics	Filters/Noise	10	30	50	70	90	92	94	96	98
PSNR	DBAMF	32.26	28.63	26.11	23.75	20.41	19.83	19.16	18.24	16.79
	ASWMF	32.82	29.66	27.22	24.98	21.88	21.01	19.57	17.61	15.14
	MDBUTM	33.05	29.98	27.28	22.29	14.96	14.21	13.47	12.73	12.01
	FSBMM	32.98	29.19	26.67	24.63	22.09	21.55	20.85	20.12	18.56
	RSIF	32.04	28.44	26.2	24.17	21.51	21.03	20.48	19.71	18.37
	DMAF	32.64	29.64	27.43	25.32	22.37	21.66	20.1	16.86	11.58
	TVWA	32.5	30.09	28.22	25.96	22.84	22.21	21.6	20.87	19.39
	ARMF	33.04	30.47	28.41	26.04	22.89	22.39	21.79	21.1	19.99
	AFMF	29.97	28.33	26.47	24.45	19.8	17.73	15.25	12.63	9.57
	NRSMMF	30.48	28.07	25.33	24.76	16.37	14.55	12.66	10.71	8.68
	AWM3F	32.92	30.36	28.4	26.08	23.34	22.86	22.4	21.76	20.5
SSIM	DBAMF	0.9717	0.9158	0.8389	0.7362	0.5723	0.5462	0.5157	0.4753	0.4103
	ASWMF	0.9745	0.9316	0.8726	0.789	0.6268	0.5762	0.4952	0.3778	0.2544
	MDBUTM	0.9738	0.9343	0.8651	0.606	0.1914	0.167	0.1477	0.133	0.124
	FSBMM	0.9768	0.9244	0.8526	0.7661	0.6346	0.6145	0.5914	0.5656	0.5311
	RSIF	0.9705	0.908	0.8359	0.7473	0.6163	0.5976	0.5763	0.5528	0.5202
	DMAF	0.9741	0.9362	0.8843	0.8087	0.667	0.638	0.5851	0.4696	0.2358
	TVWA	0.9772	0.9421	0.895	0.8195	0.6757	0.6516	0.625	0.5947	0.5546
	ARMF	0.9781	0.9464	0.9013	0.8255	0.6787	0.6552	0.629	0.5987	0.5598
	AFMF	0.9288	0.9093	0.8627	0.7792	0.5925	0.5249	0.4202	0.2735	0.1038
	NRSMMF	0.9696	0.9317	0.8564	0.8117	0.4033	0.3071	0.2188	0.1456	0.0839
	AWM3F	0.9775	0.9419	0.8945	0.8196	0.6861	0.6657	0.6433	0.6158	0.5738

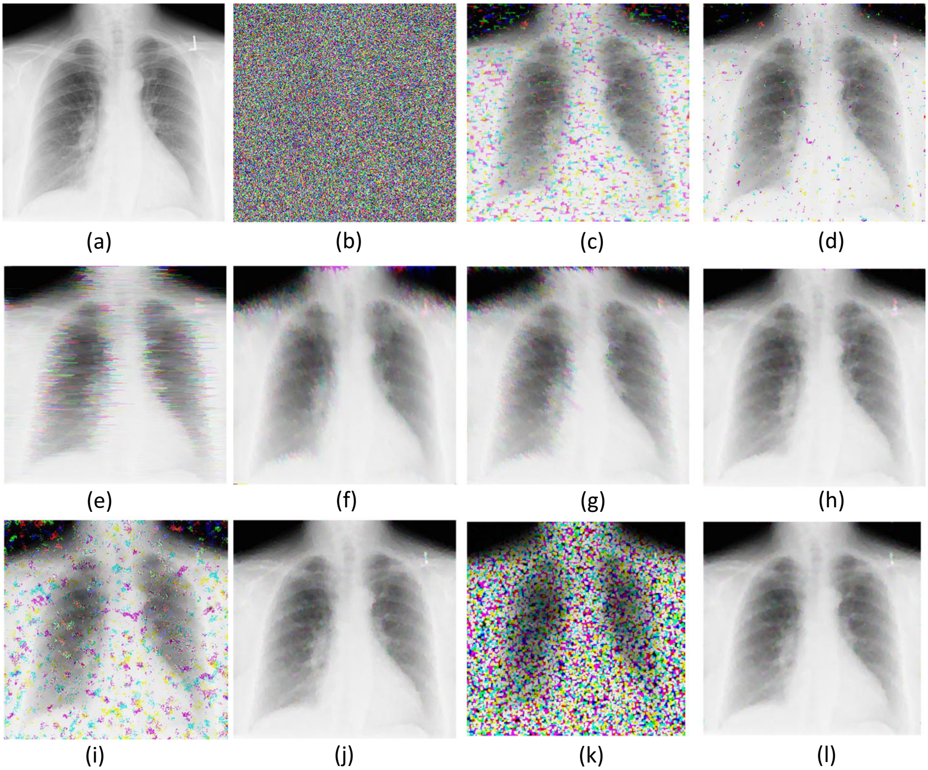


Fig. 8 (a) Original Image (b) Noisy Image (95%) (c) ASWMF (d) DAMF (e) DBAMF (f) FSBMMF (g) RSIF (h) TVWA (i) AFMF (j) ARMF (k) NRSMMF (l) AWM3F

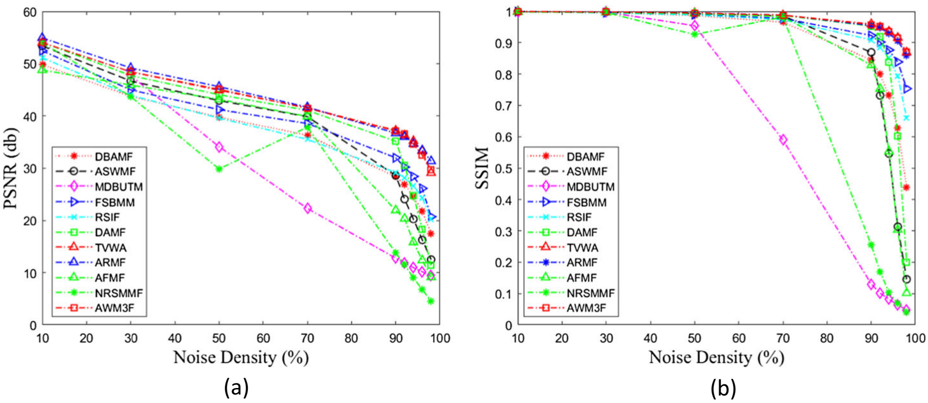


Fig. 9 PSNR and SSIM value plots of lungs medical image filtered using various filters at different noise density levels

Table 5 Average PSNR and SSIM of filtered lung images using exiting filters at different noise density levels

Metrics	Filters/Noise	10%	30%	50%	70%	90%	92%	94%	96%	98%
PSNR	DBAMF	49.8	43.71	39.77	36.25	28.32	26.81	24.61	21.79	17.39
	ASWMF	53.36	46.66	42.91	39.85	28.56	24.1	20.27	16.25	12.51
	MDBUTM	54.07	47.71	34.06	22.32	12.77	11.83	10.99	10.17	9.4
	FSBMM	52.43	44.96	41.19	38.54	31.94	30.16	28.38	26.09	20.74
	RSIF	51.11	43.92	39.56	35.47	29.13	28.16	26.59	24.39	20.19
	DMAF	54.05	47.71	44.05	41.05	35.1	30.46	24.8	18.3	11.39
	TVWA	54	48.41	44.89	41.55	37.18	36.22	35.16	33.23	29.06
	ARMF	54.91	49.16	45.58	41.66	36.65	35.94	34.62	33.37	31.3
	AFMF	48.76	45.88	43.16	39.88	21.97	20.38	15.82	12.37	9.13
	NRSMMF	53.97	43.71	29.81	37.79	13.78	11.51	9.02	6.76	4.59
	AWM3F	54	48.42	45.06	41.45	37.09	36.58	34.58	32.43	29.6
SSIM	DBAMF	0.9991	0.9953	0.9868	0.9655	0.8451	0.7991	0.732	0.6265	0.4374
	ASWMF	0.9995	0.9975	0.9934	0.9854	0.8692	0.7316	0.5456	0.3133	0.1457
	MDBUTM	0.9996	0.9979	0.9542	0.5889	0.1307	0.1016	0.082	0.0639	0.0486
	FSBMM	0.9994	0.996	0.9895	0.9777	0.9226	0.9017	0.8765	0.8384	0.7534
	RSIF	0.9992	0.9953	0.9887	0.9736	0.9086	0.884	0.8508	0.7921	0.6604
	DMAF	0.9996	0.9979	0.9945	0.9872	0.9531	0.9201	0.837	0.601	0.2
	TVWA	0.9996	0.9981	0.9951	0.988	0.9597	0.9491	0.9361	0.9156	0.8688
	ARMF	0.9996	0.9983	0.9957	0.9884	0.9548	0.9469	0.929	0.9057	0.8594
	AFMF	0.998	0.9967	0.9932	0.9831	0.8283	0.7531	0.553	0.3035	0.1019
	NRSMMF	0.9996	0.9964	0.9256	0.9822	0.255	0.1702	0.1034	0.0691	0.0416
	AWM3F	0.9996	0.9981	0.9953	0.9879	0.9579	0.952	0.9346	0.9136	0.8705

4.2 Simulation result analysis with high resolution apple image

Table 3 consists of PSNR and SSIM values of high resolution apple images filtered using some of the best existing filters with noise density varying from wide range (10% to 90%) to very high range (92% to 98%). The quality metrics from this table suggest that ARmF [8] gives significant performance as compared to other existing filters.

For better comparative analysis, Fig. 6 presents the plot for PSNR and SSIM of apple images for varying noise density as given in Table 3. Considering these given figures and tables we can see that ARmF [8] provides better result at all levels of noise density and shows significant improvement in medium noise density and carries it forward at higher noise density.

Finally for the subjective analysis, the reconstructed filtered images are shown in Fig. 7. It can be seen from the images that the ARmF [8] is able to preserve more details as compared to other filters used.

4.3 Analysis on medical images

This subsection provides a qualitative and quantitative analysis on a medical image (Lungs image) in terms of PSNR and SSIM from 10% to 98% noise density with variable intervals. The quality metrics are summarized in Table 4. Analysing the given performance

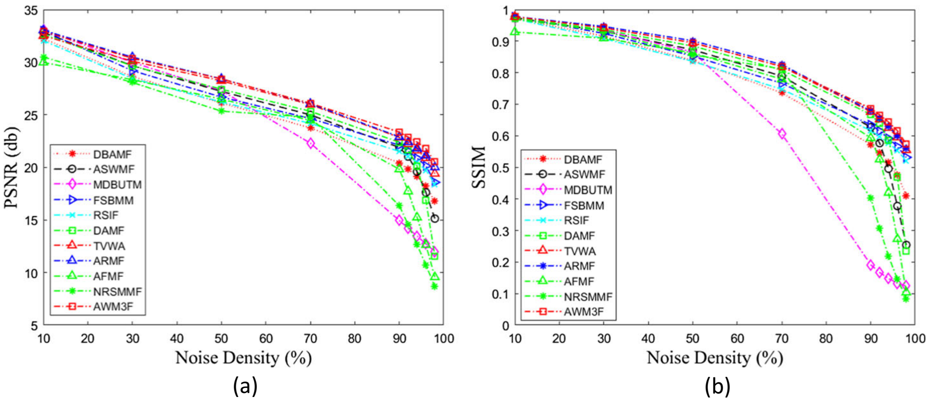


Fig. 10 PSNR and SSIM value plots of Kodak images filtered using various filters at different noise density levels

metrics, we can clearly see that at noise density $< 75\%$ ARmF [8] has slightly better performance as compared to AWM3F [23]. But, as the noise density increases beyond this level AWM3F [23] provides a better performance as compared to others.

Finally, the filtered images using various filters are shown for subjective analysis in Fig. 8. From simulation images it can be observed that the maximum details are retained by the one filtered using AWM3F [23] filtering technique. Hence, it can be clearly seen that AWM3F [23] has the best overall performance among other filters in consideration as can be seen from Fig. 9.

4.4 Analysis on Kodak images

This subsection provides analysis on Kodak images in terms of PSNR and SSIM from 10% to 98% noise density at variable intervals. Simulation results are presented in Table 5 and Fig. 10. From Fig. 10 it can be noted that, initially ARmF [8] performs better till 55% of noise density. Afterwards, AWM3F [23] has a slightly better performance as compared to other filters.

5 Conclusion

This paper summarizes the various state-of-the-art salt & pepper noise removal techniques and their comparative analysis. Through the medium of this communication, the reader can expect successful classification of various noise types and various filtering techniques based on their underlying algorithms to eliminate salt & pepper noise. The characteristic of different filters with their mathematical expressions are summarized and shown at one place in the table to have quick comparative analysis. Further, quality metrics of some filters are evaluated and summarized in the tables. The plots of quality metrics at varying noise density show that the TVWA and AWM3F performs better that the existing filtering techniques. Finally, the reconstructed/filtered image are shown for qualitative analysis. We can conclude that, TVWA are the best technique out of the ones shown at all noise densities. From the results it can seen that the accuracy of AWM3F is comparable with TVWA at all noise density ranges. Both the filtering techniques uses three values (middle, minimum and maximum values) of the chosen window size along with their weights for restoring pixel value. AWM3F uses two highly correlated groups of noise free pixels for computation whereas TVWA doesn't. Since the performance of adaptive filters especially the ones with noise density sensitive behaviour, the authors feel that research in this subcategory of filters will lead to better filtering techniques. The future work will also involve how to apply these algorithms in real life applications especially in de-noising images with depth effect implementation, and 3d-point de-noising. At last it is also important to optimize these algorithms in terms of speed and space.

Funding This work is supported by Thapar Institute of Engineering Technology (TIET), Patiala, India.

Declarations

Competing interests The authors report no declarations of interest.

Conflict of Interests The authors declare no conflict of interest.

References

1. Ahmed F, Das S (2013) Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean. IEEE Trans Fuzzy Syst 22(5):1352–1358
2. Balasubramanian G, Chilambuchelvan A, Vijayan S, Gowrison G (2016) An extremely fast adaptive high-performance filter to remove salt and pepper noise using overlapping medians in images. Imaging Sci J 64(5):241–252
3. Balasubramanian G, Chilambuchelvan A, Vijayan S, Gowrison G (2016) Probabilistic decision based filter to remove impulse noise using patch else trimmed median. AEU Int J Electron Commun 70(4):471–481
4. Bhadouria VS, Ghoshal D, Siddiqi AH (2014) A new approach for high density saturated impulse noise removal using decision-based coupled window median filter. Sig Image Video Process 8(1):71–84
5. Deivalakshmi S, Palanisamy P (2010) Improved tolerance based selective arithmetic mean filter for detection and removal of impulse noise. In: 2010 5th international conference on industrial and information systems. IEEE, pp 309–313
6. Deivalakshmi S, Palanisamy P (2016) Removal of high density salt and pepper noise through improved tolerance based selective arithmetic mean filtering with wavelet thresholding. AEU Int J Electron Commun 70(6):757–776
7. Dong Y, Xu S (2007) A new directional weighted median filter for removal of random-valued impulse noise. IEEE Signal Process Lett 14(3):193–196
8. Enginoğlu S, Erkan U, Memiş S (2019) Pixel similarity-based adaptive riesz mean filter for salt-and-pepper noise removal. Multimed Tools and Appl 78:35401–35418, 12
9. Erkan U, Enginoğlu S., Thanh D, Le Minh H (2020) Adaptive frequency median filter for the salt and pepper denoising problem. IET Image Process 14:1291–1302, 05
10. Erkan U, Gökrem L, Enginoğlu S (2018) Different applied median filter in salt and pepper noise. Comput Electric Eng 70:789–798
11. Esakkirajan S, Veerakumar T, Subramanyam AN, PremChand C (2011) Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Processing Letters 18(5):287–290
12. Faragallah OS, Ibrahem HM (2016) Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise. AEU Int J Electron Commun 70(8):1034–1040
13. Garg B (2020) Restoration of highly salt-and-pepper-noise-corrupted images using novel adaptive trimmed median filter. Signal image and video processing
14. Garg B (2020) An adaptive minimum-maximum value-based weighted median filter for removing high density salt and pepper noise in medical images. Int J of Ad Hoc and Ubiquitous Computing 35(2):84–95

15. Garg B, Arya K (2020) Four stage median-average filter for healing high density salt and pepper noise corrupted images. Multimed Tools Appl 79(43):32305–32329
16. Hwang H, Haddad RA (1995) Adaptive median filters: new algorithms and results. IEEE Trans Image Process 4(4):499–502
17. Lu C-T, Chen Y-Y, Wang L-L, Chang C-F (2016) Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window. Pattern Recogn Lett 80:188–199
18. Lu C-T, Chou T-C (2012) Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter. Pattern Recogn Lett 33(10):1287–1295
19. Meher SK, Singhawat B (2014) An improved recursive and adaptive median filter for high density impulse noise. AEU Int J Electron Commun 68(12):1173–1179
20. Ng P.-E., Ma K.-K. (2006) A switching median filter with boundary discriminative noise detection for extremely corrupted images. IEEE Trans Image Process 15(6):1506–1516
21. Roy A, Laskar RH (2017) Non-casual linear prediction based adaptive filter for removal of high density impulse noise from color images. AEU Int J Electron Commun 72:114–124
22. Satti P, Sharma N, Garg B (2020) Min-max average pooling based filter for impulse noise removal. IEEE Sig Process Lett 27:1475–1479
23. Sharma N, Sohi PJS, Garg B (2021) An adaptive weighted min-mid-max value based filter for eliminating high density impulsive noise, Wireless Personal Communications. [Online]. Available: https://doi.org/10.1007/s11277-021-08314-5
24. Singh KM, Bora PK, Singh SB (2002) Rank-ordered mean filter for removal of impulse noise from images. In: 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT'02., vol 2. IEEE, pp 980–985
25. Sohi PJS, Sharma N, Garg B, Arya KV (2021). In: Sharma MK, Dhaka VS, Perumal T, Dey N, Tavares JMRS (eds) Noise density range sensitive mean-median filter for impulse noise removal, Innovations in Computational Intelligence and Computer Vision. Singapore, Springer, pp 150-162
26. Srinivasan K, Ebenezer D (2007) A new fast and efficient decision-based algorithm for removal of high-density impulse noises. IEEE Signal Process Lett 14(3):189–192
27. Vasanth K, Manjunath T, Raj SN (2015) A decision based unsymmetrical trimmed modified winsorized mean filter for the removal of high density salt and pepper noise in images and videos. Procedia Computer Science 54:595–604
28. Veerakumar T, Esakkirajan S, Vennila I (2014) Recursive cubic spline interpolation filter approach for the removal of high density salt-and-pepper noise, Signal. Image and Video Processing 8(1):159–168
29. Vijaykumar V, Mari GS, Ebenezer D (2014) Fast switching based median–mean filter for high density salt and pepper noise removal. AEU Int J Electron Commun 68(12):1145–1155
30. Wang X, Shen S, Shi G, Xu Y, Zhang P (2016) Iterative non-local means filter for salt and pepper noise removal. J Vis Commun Image Represent 38:440–450. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S104732031630027X
31. Zhang S, Karim MA (2002) A new impulse detector for switching median filters. IEEE Signal Process Lett 9(11):360–363
32. Zhang P, Li F (2014) A new adaptive weighted mean filter for removing salt-and-pepper noise. IEEE Signal Process Lett 21(10):1280–1283