ADAPTIVE DIGITAL IMAGE DENOISING FOR GRAY SCALE AND COLOR IMAGES

A Thesis

submitted in partial fulfillment of the requirements for the

award of the degree of

DOCTOR OF PHILOSOPHY

in

Electronics and Communication Engineering

by

Amandeep Singh

(41600034)

Supervised by

Dr. Gaurav Sethi

Co-supervised by

Dr. G.S Kalra

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

2020

DECLARATION

I declare that the thesis entitled **"Adaptive Digital Image Denoising for Gray Scale and Color Images"** has been prepared by me under the guidance of **Dr. Gaurav Sethi,** Associate Professor, School of **Electronics and Electrical Engineering** at **Lovely Professional University, Punjab, India** and **Dr. G.S Kalra, Professor at CT Group of Institute , Punjab, India**. No part of this thesis has been included in or has formed the basis for the award of any Degree or Diploma or Fellowship of any institution or university anywhere previously.

I further declare that there is no falsification or manipulation in terms of research materials, equipment or processes, experiments, methods, models, modeling, data, data analysis, results or theoretical work.

I have checked the thesis using Turnitin for ensuring that there is no plagiarized material in my thesis and wherever any copyrighted material has been included, the same has been duly acknowledged.

Amandeep Singh
School of Electronics and Electrical Engineering
Lovely Faculty of Technology and Sciences
Lovely Professional University
Phagwara, -144411 (Punjab)

We endorse the above declaration of Ph.D. student

Date:

Dr. Gaurav Sethi	Dr. G.S Kalra
Associate Professor	Campus Director
(Ph.D. Supervisor)	(Ph.D. Co-Supervisor)

CERTIFICATE

We certify that **AMANDEEP SINGH** has prepared his thesis entitled "**Adaptive Digital Image Denoising for Gray Scale and Color Images**", for the award of the Ph.D. degree of the Lovely Professional University, under our guidance. He has carried out the work at the **School of Electronics and Electrical Engineering**, Lovely Professional University.

Dr. Gaurav Sethi
Associate professor
School of Electronics and Electrical Engineering
Lovely Faculty of Technology and Sciences
Lovely Professional University,
Phagwara, Kapurthala-144411 (Punjab)

Dr. G.S Kalra
Campus Director
CT Institute of Engg., Mgt & Technology Technical
Shahpur, Jalandhar-140401(Punjab)

ABSTRACT

There are numerous methods in the field of Image denoising that have emerged in the recent past. However, there are only a few that provide solutions with high accuracy and high proximity to the original image. Generally, to denoise an image two stage process is followed in which at first noise location is detected in image and secondly detected noise location is restored. The noise detection stage is very challenging as different noise model distort the image with different intensity values and strength. This leads to inaccurate detection of noisy pixels in the image i.e. over detection and under detection. After the detection of noisy pixel location it is required to replace the noisy pixel value with the estimated original pixel value. This task is performed by restoration stage and it is also a very challenging task due to presence of noise. The overall performance of denoising algorithm depend upon individual accuracy of detection stage and restoration stage. To overcome these challenges the new adaptive algorithms of detection stage and restoration stage has been proposed to over come the over and under detection of noisy pixels. As quality of restoration stage depends upon closeness of restored value to the original values of image. To achieve this further an algorithm has been developed to enhance the performance of existing algorithm, this algorithm ensures the high proximity of restored value to the original pixel value. One more crucial thing that also needs to addressed in image denoising is to ensure the operational stability of algorithms on wide range of data set. To overcome this issue the proposed algorithms are tested on wide range of standard data set of grayscale images and color images. To validate the performance of proposed algorithms on the wide range of dataset various qualitative parameters are also utilized i.e. Peak signal to noise ratio (PSNR), Structural similarity index (SSIM) and Image enhancement factor (IEF).These parameters plays a vital role to evaluate the performance of image denoising algorithms with other existing denoising algorithms for different level on noise affected images. Therefore, this thesis focuses on designing algorithm with high denoising with detail preservation. In the initial work the various algorithms for image denoising have been analyzed and also the research gaps related to the study have been identified. Secondly, the hybrid algorithm for achieving image denoising of gray scale and color image dataset has been accomplished. Finally, on the basis of complete denoising algorithm requirements and available research gaps in the design process of

image denoising algorithm, new adaptive image denoising algorithm have been presented in this work. The results validation of these algorithms has been accomplished and comparative analysis with the present state of the art is also done. The proposed hybrid method , spatially adaptive image denoising - enhanced noise detection (SAID-END) method, amalgamation of rank-order absolute difference trimmed global mean with progressive principal component analysis using performance booster method (ROAD-PPCA-PB) and amalgamation of spatially adaptive image denoising - enhanced noise detection with progressive principal component analysis using performance booster method (SAID-END-PPCA-PB) has respectively achieved 3.01%, 5.24%, 7.32% and 14.85% PSNR improvement over best performing recent state of the art iterative median filter algorithm (IMF) for gray scale standard Lena test image. Similarly, these algorithms have achieved 5.84%, 8.10%, 10.34% and 16.94% PSNR improvement respectively over best performing recent state of the art adaptive unsymmetric trimmed shock filter (AUTSF) algorithm for color standard Lena test image.

ACKNOWLEDGEMENT

High achievement always takes place in the framework of high expectation. The expectation was there and I begin with determined resolve and put in sustained hard work. It has been rightly said that every successful individual knows that his or her achievement depends on a community of persons working together but the satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible.

This is to acknowledge with gratitude the guidance and ameliorating suggestions from my supervisor, Dr. Gaurav Sethi, and co-supervisor Dr. G.S Kalra for their support and motivation throughout the thesis work.

I am grateful to the members of the School of Research Degree Programme (RDP) for being in my entire research progress review panel, providing useful suggestions during the period. I am indeed thankful to all anonymous reviewers of my research papers submitted to various International Journal and International conferences, due to which I was indeed able to improve upon the work containing herein.

I am indebted to Honorable Chancellor, Worthy Pro-Chancellor, the Vice-Chancellor, and the successive Deans, LPU, for facilitating the administrative issues involved and encouraging me throughout.

I express my heartfelt gratefulness to Head of School, of the School of Electronics and Electrical Engineering, and the staff of Electronics and Electrical Engineering School for their cooperation and support. I am also thankful to Dr. Kulbir Singh of Thapar Institute of Engineering & Technology, Patiala for the valuable advice and support.

No words of thanks are enough to express my deepest gratitude and sincerest love to my parents whose rock-solid faith in me kept me strong in all circumstances. I am also thankful to my wife Ramanpreet Kaur for her wholehearted endless support and patience. Last but not least, I bow my head before the Almighty God for showering his mercy on me for every moment.

Amandeep Singh

Table of Contents

List of Abbreviations

APP	Ambivalent Pixels Percentage
AUTSF	Adaptive unsymmetric trimmed shock filter
CCTV	Closed-circuit television
CD	Color image dataset
CWM	Center weighted median
CWMF	Center weighted median filter
DAMF	Decision Based Adaptive Median Filter
DBMF	Decision Based Median Filter
END	Enhanced noise detection
FDS	Fuzzy denoising for Impulse noise
FFT	Fast Fourier Transform
GCV	Generalized Cross Validation
GD	Grayscale image dataset
ID	Image denoised
IEF	Image enhancement factor
IIN	Iterative scheme-inspired network
IMF	Iterative median filter
IR	Resultant matrix
MCF	Modified cascaded filter
MF	Median filter
MSE	Mean square error
MSI	Maximum Similarity Index
MSIR	Maximum Similarity Index Range
NA	Number of algorithms
NL	Noise levels
NP	Number of parameters
PCA	Principle component analysis
PDF	Probability density function
PSMF	Progressive Switching Median Filter
PSNR	Peak signal to noise ratio

ROAD	Rank-order absolute difference
ROM	Rank-ordered mean
SAID	Spatially adaptive image denoising
SIR	Similarity Index Range
SMF	Standard median filter
SP	Selected parameter
SPN	Salt & Pepper noise
SSIM	Structural similarity index
TGM	Trimmed global mean
TRC	total number of results for CD
USC	University of South California

List of Tables

List of Figures

Chapter 1
Introduction

This chapter focuses on understanding of image, image denoising and image processing. In today's world technology is part of the lifestyle in one way or another. The impact of technology can easily be understood by simply realizing how often we are using it in our daily life. One of the most impact making technology is digital image processing. Digital image processing plays a vital role in various applications like face recognition, surveillance, medical imaging, robot vision, underwater imaging, satellite imaging etc. Image denoising is the primary preprocessing to image processing and also to almost all image analysis since impulse noise is an unwanted and inevitable noise that mixed with the original image in different situations, such as during image acquisition, storage and transmission. The Salt & Pepper noise (SPN) is the most common form of impulse noise in digital images. This noise can highly dilute the image quality as it occurs due to multiple sources such as the transmission of image, dust on the camera lens, faulty photosensors and faulty memory locations. Generally, faulty photosensors and faulty memory locations cannot be avoided as these occur due to the aging of electronic components. Due to these inevitable challenges image denoising is still a significant field and demanding for constant improvement. The process of image denoising becomes further challenging when an algorithm is required to denoise both grayscale and color images affected by SPN. To enhance the performance of image processing-based application it requires to keep the image noise-free and it should have proximity to the original image after denoising.

This chapter is organized into seven sections. Section I describes the brief overview of the area under study. Section II and Section III give the introduction to image denoising and motivation respectively. Further, Section IV provides various research gaps. Section V and Section VI hold the objectives and research methodology. Finally, Section VII presents the organization of thesis work respectively.

1.1 Image Denoising

In this digital era images are part of our daily life starting from a mobile phone screen unlock, social media to CCTV surveillance. Every day a massive number of images are captured and stored, but both these tasks are prone to noise. These images are considered as a vital source of information, so these are transmitted, stored and processed in high numbers. Any loss in image information can affect the overall performance of the application containing the image processing stage. So, day by day the demand for more conspicuous and accurate images is increasing. To fulfill this demand noise is required to be removed from the images. For this purpose, various algorithms are proposed and it is very difficult to choose one for the desired application. Every algorithm has its advantages and disadvantages. This demands more technological advancements in image denoising algorithms to maintain image quality[1]–[5]. Due to these reasons, image denoising is still a valid challenge for the researchers. The image denoising process is required to achieve maximum denoising while keeping the prime details of images like high-frequency components and object bounders. The image loses some important information in the denoising stage as there is a very less intensity difference between the object and its boundaries. It is worth taking a note here that image denoising is an inverse problem, so the solution may vary from problem to problem. Salt and Pepper is the noise which is having a major role in image distortion. This noise can reduce the image details, hence it is important to detect and remove the noise before providing the image to the image processing stage.

Salt and Pepper noise can have two possible values i.e, 0 or 255 [6]. The occurrence of impulse noise is random in image and can produce any pattern which makes it even more difficult to detect the location of noise and to predict the original value of the noisy pixel. This problem is considered as a classic problem in digital image processing but it is still attracting the attention of various researchers as the need for enhanced image visual clarity is always in demand. To achieve this various algorithms are available, hence it will be a challenging task to find a suitable method for both grayscale

and color images. As various methods were designed earlier for the grayscale image but in comparison, less work is performed with respect to color image denoising.

To perform image denoising it's important to understand image noise models. At each stage of the acquisition of a scene, disturbances (scratches, dust, faulty camera sensors, amplification, quantization) will deteriorate the quality of the image. These disturbances are grouped under the name of "image noise"[7]. The image noise can be categorized into two categories:

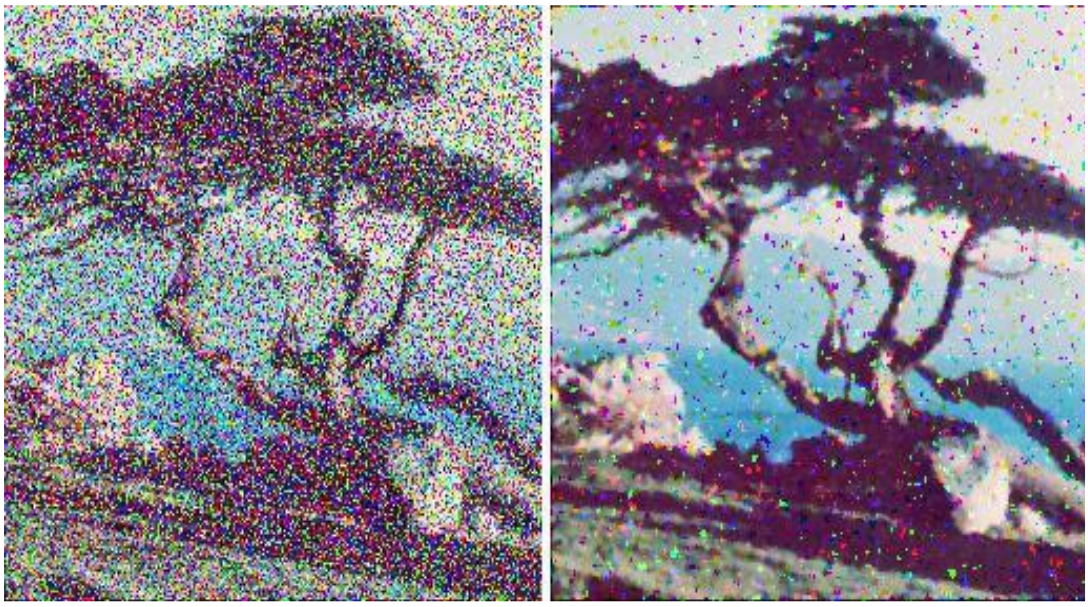
- Independent noise (we speak of random noise)
- The noise that depends on the distribution pattern.

We can represent the noise affected image with the following expression[8]

$$S(a,b) = w(a,b) + z(a,b) \quad (1)$$

S(a,b) is defined as the combination of the real image w(a,b) and noise z(a,b).The noise z(a,b) is frequently defined by its variance ($\sigma^2 z$). PSNR defines the quality of a denoised image from an image under the influence of noise. Similarly, $\sigma^2 w$ and $\sigma^2 s$ represent the respective variances of the actual and the processed image. The noise of the image is considered as a random field. Because of the different mechanism are involved in image acquisition, the level of noise is affecting the image can be different. So, an algorithm needed which can remove this wide range of noises level. The most common noise in a digital image is Salt and Pepper noise which is also referred as shot noise, binary noise and impulse noise. This is caused while capturing and sending the images because of impairments involved in the communication process. It has just two conceivable values (0 and 255) as already discussed. The likelihood of the occurrence of any of these two values is normally under 0.1. The noisy pixels are kept on the other hand to the minimum value or to the most extreme value, provide the picture a "Salt and Pepper" like resemblance[1], [6], [9]–[12]. Noise-free pixels in the affected image stay unaltered. For the image of 8-bit, the representation of Pepper noise in the digital image

is done by using value zero and for Salt noise reflection is achieved by using value 255. To represent the effect of noise on image, 40% Salt and Pepper noise affected image is shown in figure 1.1 (a).



(a) (b)

Figure 1.1. (a) 40% Salt and Pepper affected image (b) denoised image.

There are two fundamental ways to deal with picture denoising, spatial domain methods and frequency domain techniques. A standard approach to remove noise from picture information is to utilize spatial methods. Spatial methods can be further arranged into non-linear and linear methods. These techniques can be subdivided on the basis of situational awareness. Situational awareness can be used to make an image denoising algorithm adaptive which enhances the denoising performance. Generally, to reduce computational cost of algorithms and to make operation faster non adaptive algorithms are preferred. These algorithms perform operation at high speed at the cost of loss in proximity to original image. These methods work well in low to mid level of noise but quality of denoised image tend to reduce with increase in noise level in the image. Whereas the adaptive algorithms are used to improve the denoised image proximity to original image and to enhance the quality of image denoising in case of wide range of noise affected image. The adaptive nature of a algorithm can be implements at noise

detection and noise restoration levels in image denoising algorithm. Adaptive algorithms are complex to design and can have high computational cost also but these algorithms are focused to achieve high image denoising efficiency. Some of the denoising methods are as follows.

A. Non-Linear Filter

With non-linear filters, the noise is removed considering low frequency values as non-noisy and informative. Spatial methods utilize a low pass filter on window of pixels with the presumption that the noise involves the higher values of the frequency range. The large number of spatial methods remove noise to a significant value at the cost of blurring pictures which thus makes the edges in pictures undetectable. Lately, an method based on nonlinear center value method for example, weighted middle , rank adapted rank determination have been formed to overcome this downside[13]–[15].

B. Weighted Median Filter

The essential thought is to offer weight to every pixel. Each pixel is given a weight and this weight is increase with pixel. As per this weight the pixels are sorted into increasing order of values , and higher weight of pixel will provide more influence of the respective pixel in reconstruction of original value [16], [17].

C. Mean Filter

Mean filtering is essentially to replace every pixel value in a picture with the mean (`average') estimation of its neighbors, including itself. This has the impact of fading pixel values which are inconsistent of their environment. Mean filtering is generally considered as a convolution channel. Like different convolutions it is based around a window, which consider the shape and size of the area to be tested while computing the mean. Regularly a 3×3 square piece is utilized, considerably larger window size(e.g. 5×5 squares) can be utilized for more intensive smoothing[8], [18], [19].

D. Spatial-Frequency Filtering

Spatial-frequency filtering alludes utilization of low pass filters with Fast Fourier Transform (FFT). In frequency filtration methods [20]–[22] the removal of the noise is

accomplished by devolving a frequency domain filter and adjusting its cut-off frequency at a point when the noise levels are decorrelated from the required window in the frequency range. These techniques are tedious and rely upon the cut-off frequency and the filter bandwidth.

E. Non-Adaptive Threshold

VISUShrink [23]–[25] There are two types of thresholding: Soft and Hard thresholding. The Universal thresholding method i.e.VisuShrink is based on the Hard-thresholding and it is not appropriate for Soft-thresholding. VISUShrink is known to yield excessively smoothed pictures since its edge reconstruction can be much wider than edges in original image because of its reliance on the quantity of pixels in the picture.

F. Adaptive Threshold

SUREShrink[24], [25] utilizes a hybrid of the edge information and the SURE [Stein's Unbiased Risk Estimator] limit and performs superior to VISUShrink. BayesShrink limits the risk estimator work expecting better performance and consequently yielding required information along with edge preservation. BayesShrink outperform SUREShrink in the majority of the circumstances. Cross Validation[26] replaces wavelet coefficient with the weighted normalized neighborhood coefficients to limit Generalized Cross Validation (GCV) work giving ideal edge to each object. The presumption that one can separate noise from the window exclusively, is a challenging task the reconstruction of damaged pixel is incorrect when noise levels are high for the respective window sizes. Under these high noise condition, the spatial setup of neighboring wavelet coefficients can assume a vital part as noise in window arrangements. Reconstructed images tend to form significant components (artifacts e.g. straight lines, blur and patches), while high noise levels are randomly distributed.

1.2 Motivation

In image denoising, the designing of an efficient algorithm is a major challenge since it desires to have accurate noise detection, high proximity to original image and noise filtration for wide range of color and gray scale images. Thus, the initial work in the field of algorithm design with mentioned characteristics has been accomplished by

employing spatial domain and principle component analysis. Image denoising algorithm designs can be broadly divided into two stages, i.e., noise detection and noise filtration. However, due to vast image applications usage in today's world, algorithm are required to be designed for both grayscale and color image dataset.

The algorithm applicable for both grayscale and color images is required to be tested to wide range of image along with wide rage on noise levels. As in real life scenarios the noise level can be in any range from very low to high depending on various factors like environment, electronic component aging etc. To overcome these issues the proposed algorithm is tested on standard data set of color and grayscale images affected with wide range of noise.

1.3 Research Gaps

Research gaps identified as follows

- Although the spatial filters perform well on digital images but they have some constraints regarding resolution degradation. These filters operate by smoothing over a fixed window and it produces artifacts around the object. This leads to requirement of adaptive window algorithms.

- Existing filters so far have some disadvantages which are: loss of high frequency components for example thin lines, thin edges, blurring image fine textures in the image during the noise removal operation. So, an improved noise detection algorithm can be implemented which will not only focus on identification of noisy pixels also focus on maintaining the image structure edges and work without decreasing the features of image.

- Denoising is often required for proper image analysis, both by humans and machines. In the case of random-valued impulse noise, the performance of standard noise detectors is significantly reduced. In this case the improvement

is required in existing noise detector algorithms for efficient detection of noise.

- Numerous approaches are being presented for gray scale image denoising but still there is a vast research scope for color image denoising.

1.4 Research Objectives

The objectives of the work are as follows:

1. To perform comparative analysis of existing algorithms for denoising of gray scale and colored image to find algorithm which will give superior performance for the removal of Salt and Pepper noise from the affected images.

2. To improve the noise reduction procedure efficiency by developing adaptive detection stage.

3. To improve the noise reduction procedure efficiency by developing adaptive filtering stage.

4. To improve the algorithm for denoising procedure using noise pixel direction for achieving enhanced results.

1.5 Research Methodology

To achieve the proposed objectives of the work, a methodology layout is prepared in figure 1.2. In the research, we will present an efficient image denoising procedure for grayscale and color image noise reduction (The comparison will be done on various

algorithms for selection of grayscale and color image denoising procedure (objective 1), to make detection stage window size adaptive for denoising process (objective 2), to increase the performance by making restoration stage adaptive for denoising process (objective 3) and to achieve enhanced performance from designed algorithm (objective 4). All the designed algorithms are tested on wide range of standard dataset with low to high levels of noise.

The Flow chart for proposed methodology is given:

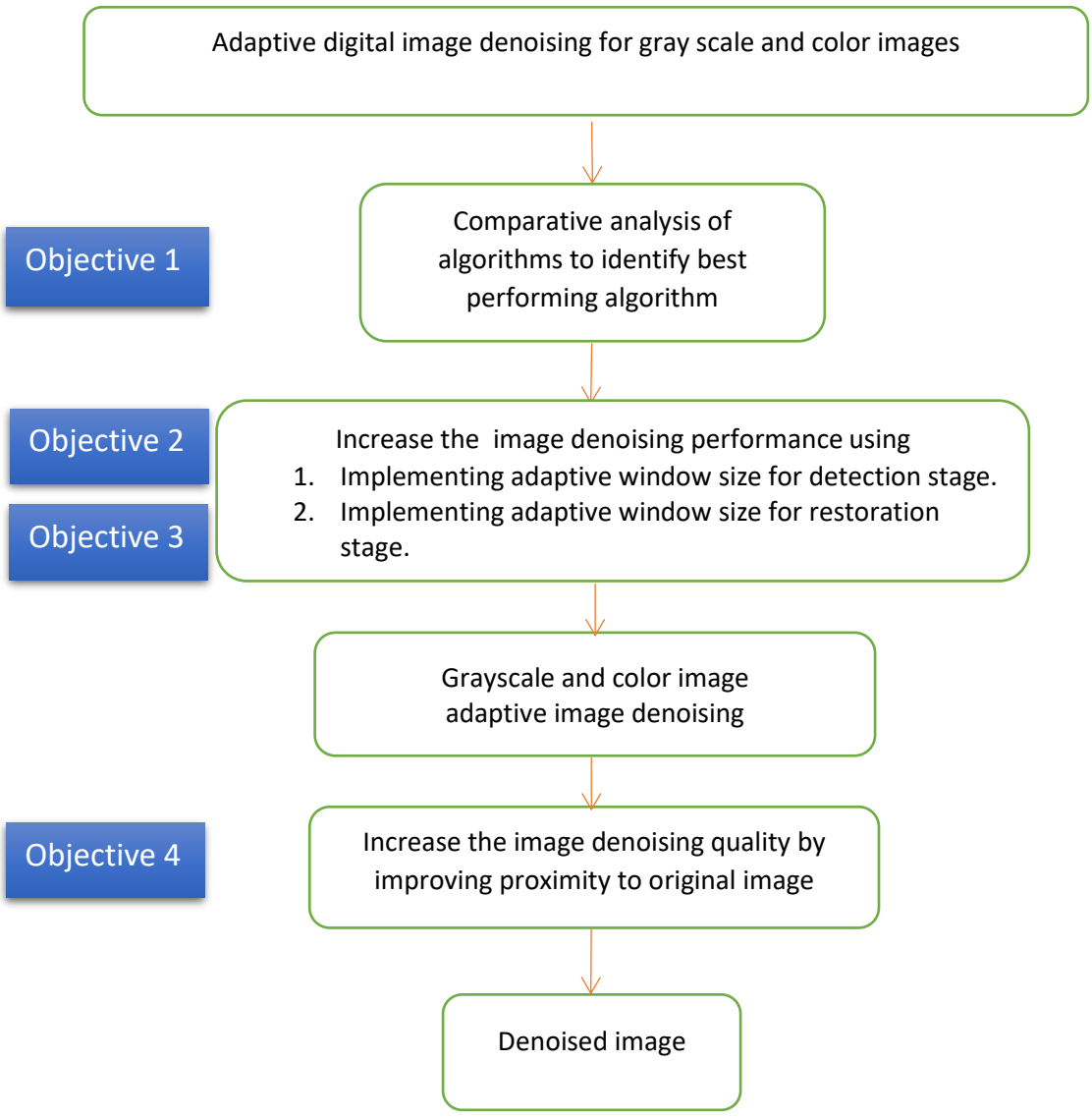


Figure 1.2. Methodology for the objectives

We are proposing an improvement in detection and filtering stage of denoising algorithm. To enhance the proximity of denoised pixel to the original pixel intensity of neighbor pixels are carefully selected by avoiding noise influenced pixels before replacing the noisy pixel value with mean or median of selected pixels.

1.6 Thesis Organization

This thesis is organized into six chapters as follows:

Chapter 1 provides the introduction to the image denoising systems and various applications. Further, the importance of designing efficient image denoising algorithm and challenges for color image denoising have been presented. Based on this discussion the motivation, research gaps, and objectives of the proposed work have been also given.

Chapter 2 delivers an exhaustive literature review for the proposed work. In this review, the different techniques for achieving image denoising over image affected with SPN are presented. Moreover, the challenges present in existing algorithms are discussed.

Chapter 3 gives the basic design process of hybrid denoising algorithm for both grayscale and color images. Various existing algorithms like ROAD-TGM, DBMF, CWMF, PSMF, MF has been tested and top performing algorithms were selected for proposed algorithm. The proposed algorithm is hybrid in nature, representing a combination of two existing algorithms. It is important to mention that the proposed algorithm incorporates the advantages of both existing algorithms, yielding better results.

Chapter 4 presents proposed detection stage and proposed filtration stage combined together as a spatially adaptive image denoising via enhanced noise detection method (SAID-END) is presented for grayscale and color images. The denoising is achieved using a two-stage sequential algorithm, the first stage ensures accurate noise estimation by eliminating over and under detection of noisy pixels. The second stage performs

image restoration by considering non-noisy pixels in estimation of the original pixel value.

Chapter 5 provide two proposed methods and these methods are two-stage sequential algorithm for noisy grayscale and color images. At first proposed methods enhances the accuracy of the noise detection stage by using spatial domain filter rank-order absolute difference trimmed global mean (ROAD-TGM) or SAID-END along with transform domain-based progressive principle component analysis (PCA) method. Then the performance booster algorithm is used to ensure the proximity of restored values to the original values.

Chapter 6 discuss the conclusions and future scope for the proposed work.

Chapter 2

Literature Survey

The Image denoising is crucial preprocessing stage for almost all the real time applications, image transmission and image analysis-based applications. The image denoising algorithm design is depended upon various factors like type and level of noise affecting the image. As these noise models are different from each other its practical not feasible to provide single stop solution. In the field of image denoising another factor is type of image color or grayscale, the process of denoising become more challenging when color image is required to be denoised. Grayscale images are more used in application of biomedical image processing and color images are used in applications for satellite imaging, segmentation, quality analysis etc. So, the new design of algorithm with high accuracy for wide range of color and grayscale images is required. To achieve this, analysis of existing algorithms has been reported in the literature, and this chapter provides a comprehensive review of the different existing algorithm for achieving high denoising accuracy with high proximity to original image. Moreover, the challenges, future perspectives and their applications are also discussed.

2.1 Introduction

In the field of image processing, digital images are often corrupted by several kinds of noises and artifacts. There are various reasons of generation of noises and artifacts in our digital world for example transmission of image in a noisy channel, faulty memory locations in hardware or during the process of image acquisition malfunctioning of pixels in camera sensors [27], [28]. In the digital images the major contribution to tamper the image quality is usually done by the impulse noise (it is a noise which corrupt the image with peak high and peak low values) [29], So it becomes important to remove the noise before the image is utilized further for analysis otherwise it will lead towards misinterpretation. The process of restoring the corrupted image, is known as image denoising. For the last few decades, researchers are dedicatedly working to achieve an effective denoising algorithm which will retain the important details of

image while removing the noise from digital image. To measure the performance or to evaluate the operational quality of algorithm quantitative and qualitative parameters are required, for this purpose various parameters are available, but most suitable or preferred parameters are Peak signal to noise ratio (PSNR)[30]–[32], Mean square error (MSE)[33], [34], Image enhancement factor (IEF)[11], Structural similarity index (SSIM)[35]–[37]. As we had already discussed that image is affected by several types of noises, but it is observed that impulse noise usually affects the images and the impact of impulse noise in the digital image is most severe in comparison to other noises. So, the researchers generally emphasize on the eliminating the impulse noise while ensuring the least loss to important details. Impulse noise is classified into two categories, first is the random-valued impulse noise and second is the salt-and-pepper noise also known as the fixed valued impulse noise [38], [39]. Moreover, the noise generation in the processes of acquisition (due to faulty camera sensor, amplifier, and quantization) of images is inevitable and it induce disturbances which can be discomforting for understanding and image processing. The purpose of noise filtering is to reduce intensity variations within each region of the image while respecting the integrity of scenes: transitions between homogeneous regions, significant elements of the image must be preserved for the best quality. An exhaustive literature review has been carried out related to the titled work to find out the current research aspects.

Salt and pepper noise is also referred as impulse noise, which is also additionally understood as unwanted black and white pixels. This occurs to received images because of impairments involved in the communication process. It has just two conceivable values (0 and 255) [40]. The likelihood of occurrence of any of these two values is normally under 0.1. Salt and pepper noise present in the most of the digital images due to flawed of pixel components in the camera sensors, broken storage areas, or timing blunders in the digitization procedure. Some other noise or artifacts do also affect the image like Gaussian noise generally produces impact on digital images by altering its gray values. Due to this reason the Gaussian noise model essentially characterized and designed by its normalize histogram or probability density function (PDF) with respect to the gray value. In the case of Gaussian distributed noise, the PDF is given by equation (1):

$$p_V(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2} \right) \quad (1)$$

Where μ is the mean, z represents the gray level and σ^2 is the variance[41]. The Gaussian noise mathematical model represents the correct approximation of real-world scenarios. Generally, Gaussian white noise follows two patterns of distorting pixels in the image one is Gaussian white noise with constant variance, mean and the other is Gaussian white noise of local variance as per image.

There are some different kind of abnormalities that are involved in old movies, video arrangements and pictures - to be specific: blotches and the stripes. The film materials of early pictures and videos where frequently corrupted by irregular pattern of stripes lines and blotches [42]. Old motion pictures and images are significant authentic record yet the greater part of those films deteriorates in visual quality amid the years and diminishing their value. These artifacts might show up because of damage to the film surface or they might come into existence while cleaning the concealing little zone of the outer layer of video tape. While digitizing these types of movie tapes substantial outcome in the picture will have patches with low level values those are not associated with the original pixels. Fundamental pictorial deformity is due to dust particles that get stuck with the film area of the thin top layer of film tape. Moreover, the effect of stripes and blotches also caused by the outside dust particle on lenses, present in the devices like camera and projector. The result of it shows up in the form of a tiny stripe of pixels of self-assertive form with almost dark value of respective pixels and sometimes it results in the form of blotches, which is defined as a piece or a tiny individual zone with comparative dark pixel shades. Due to which the effect of these two kinds of degradation in quality may remain permanent in film tapes and image acquisition process, so we need to design denoising algorithm to overcome these issues. Every pixel involved in stripe lines or blotches can contain different values than original, generally these are low level of intensity in the respective image. Spatial

filtering process is used when just additive noise exists in the image. As salt and pepper, strip lines and blotches are unavoidable due to their existence increases with aging of electronic components, pictures or tapes, so these are used for denoising performance evaluation in this thesis.

2.2 Denoising Methods

There are diverse classes of filtering methods present in spatial domain filtering [43], [44]. These methods are powerful when the picture is corrupted by impulse noise. A median filter has a place in the class of nonlinear filters. The median filter (MF) additionally takes after the sliding the window guideline like the mean filter [45], [46]. The median is more powerful contrasted with the mean value. In this way, a removing the exceptionally less similar pixel from a region has no impact on the central value altogether.

Subsequently the median value be the approximation of one of the pixels in the area, so the median filter does not create new implausible pixel value when the filter overlaps an edge. Henceforth the median filter is extraordinarily enhanced to protect object edges other than the mean filter. These central values help median filters in denoising uniform noise from a digital image. The primary disadvantage of SMF (standard median filter) is that it is powerful just for low noise densities in the image [47]. At high noise densities, SMF frequently display requirement for large window sizes and deficient when noise ratio is high in image. Median filter containing the high denoising power for low noise levels and it has been one of the most popular filters. As its various advancements have been proposed in form of multistate median filter [48], such as median filter based on homogeneity information decision based trimmed median filters to improve its performance [11]. The method for removal of impulse noise from highly corrupted images [49] was proposed to eliminate the noises from images where noise types are Gaussian and mixed noises. This method operates using a simple two-step method where it does switching between the output of an identity filter and a rank-ordered mean (ROM) filter. It always try to get the tradeoff between noise suppression and image fine texture preservation. It has low computational complexity as well. Simulation results show that it performs better in terms of noise suppression and preservation of fine details than other nonlinear filters with maximum 40% noise

density. The filter performs well on both grayscale and color images. The rank-ordered mean filter is another two-step method which performs noise detection and then filtering operation on the noisy pixels. This filter is adaptive where restoration is performed according to the current state of the noisy image. It is defined by the output of an operator which calculates the differences between the current pixel and the other sorted pixels where the center pixel is under evaluation. So, this method is s robust and simple but this does not preserve the image fine textures during restoration. The fuzzy filter based method [50] was proposed for enhancement of digital images corrupted by impulse noises. This technique adopts a two-step fuzzy reasoning method where the first subunit is noise detection module which detects the noises by considering luminance differences among neighboring pixels. The second subunit is restoration filter module which modifies the value of the corrected pixel to preserve the image fine textures. This filter performs better in terms of MSE than the other mentioned filter in [51]. A robust algorithm [52] was proposed for noise reduction in color images where the noise types are impulse and Gaussian. This filter has an added advantage that it can enhance the sharpness in the restored images. This operator is a smoothing one which is based on a random walk model and a fuzzy similarity measure between pixels connected by digital geodesic paths. Experimental results of this filter show that it is useful for segmentation of digital images. To some extent, this is also useful for reducing the mixed impulse noises and Gaussian noises in the digital images. The technique suffers from high computational complexity. Evolutionary method [53] was proposed for image enhancement and impulse noise reduction. The method uses the genetic algorithm to find a filter set for reducing impulse noises in digital images. The salt-and-pepper noise is used to corrupt the digital images while the algorithm works to suppress the noise from Lena image. There are m number of filters available, the number of possible ordered subsets of n filters is ‘mn’. This technique finds the proper type and order of filters using genetic algorithm. The operators of the genetic algorithm such as encoding, fitness evaluation, selection, cross over, mutation and elitism strategy are properly maintained in the algorithm to find the optimal solution. To find the filter which will be used for the all scenarios is difficult and time consuming to determine. That is why the algorithm is not suitable for fast operations. Standard deviation was [54] used to devised an algorithm for restoring digital images which are

distorted by random valued noise especially at high noise density. The detection method is performed by finding the optimum direction among the test window, by calculating the standard deviation in different directions in the filtering window. Results prove that the technique has superior performance, when compared to other existing methods, especially at high noise rates. An improved image denoising using wavelet transform method [55] was proposed which was using a dual step approach for denoising. In the first step it uses stationary wavelet based denoising and in continuation to second step, a spatial domain method, Non-local means, is used to remove the artifacts. This method works with standard method of noise detection and then removal of noise. Proposed technique concentrates on adequate selection of threshold to avoid reduction in noise pixel selection procedure and then use non local mean to fill the noise affected pixels. Now let's discuss some of the most effective and commonly used algorithms for image denoising.

Progressive Switching Median is a filter which works in two stages and it is a median based filter, [56]. Initially, an impulse detector is used to generate a sequence of binary window images. This binary window image predicts the location of noise in the observed image. Then iteration-based procedure is utilized to remove the noise. Filter performance for random value noise is very poor, but its performance is very good for fixed valued noises [57].

The Center weighted median (CWM) filter, which is a weighted median filter giving more weight just to the central values of every window [58]. This filter can save important details while removal of added noise from digital image. The factual properties of the CWM filter are examined. It is observed that the CWM filter can outperform the median filter. The median weight is typically required to be sure and odd for better execution [59]. CWM filter can maintain its performance with respect to different noises and artifacts.

The image denoising techniques are required to remove the noise existing in the image. But these techniques are generally designed for grayscale images. To make these techniques compatible with color images an iterative mechanism is required to be produced which can pipeline the component (Red, Green and Blue image) of color

image. So, by breaking the color image into 3 components a grayscale algorithm can be made compatible with them. Let us consider an example grayscale images which are having single matrix containing pixel intensity value from 0 to 255, whereas if we break a color image into its three basic color components then one by one separately processing can be applied to them and these separate images can be combined to produce color image at the end again. Now let's discuss the algorithms considered for this comparative analysis. Let's start with first one which is ROAD (Rank Ordered Absolute Difference) this method is quite efficient when image is affected by uniform noise[60]. This method is based on a window to detect the pixel affected by noise, so these detected pixels can be removed from image. Let s = (s1,s2) be the pixel position below the threshold value and Ωs(N) be the number of point in a (2N+1) × (2N+1) surrounding concentrated at s for creation of window.

$$\Omega s(N) = \{s + (i,j) \quad -N \leq i,j \leq N \tag{2}$$

Let us consider N=1. Hence Ωz denotes the set of points in a 3x3 identified surrounding of s. Given by expression (3)

$$\Omega z = \Omega s\ (1)/\{s\} \tag{3}$$

For every one point of y ϵ Ωz define $d_{x,y}$ as the complete difference in strength of the pixel among s and y. Absolute difference is expressed by following expression (4)

$$d_{x,y} = |u_s - u_y| \tag{4}$$

Sort the $d_{x,y}$ value in ascending order and describe the ROAD by following expression (5)

$$ROAD_m(s) = \sum_{i=1}^{m} r_i(s)\ 1 < i < m \tag{5}$$

Where $2 \leq m \leq (2N+1)^2 - 1$ and $r_i(s)$ is the smallest $d_{x,y}$ for y ϵ Ωz.

Let’s suppose value for N=1 and m=4, ROAD offers similarity or closeness with its neighbors which is presented in the form of pixel intensity value to its 4 nearest neighbors in 3×3 window. Rationale under this process noisy pixel will have high intensity difference with other neighbor pixel values in comparison the original pixel values will have lower differences of intensity values. So, the ROAD value would be higher in places where uncontaminated pixel existing and these are part of original picture. Moreover, these pixels are going to have large portion of the neighboring pixel of similar intensity so ROAD value will be high. ROAD value can be utilized to recognize a pixel infused by noise affect, this can be achieved by setting a specific threshold value. For the image denoising the ROAD value is more prominent than the threshold value when the pixel is considered to be affected by noise. It is recommended to utilize a 3×3 window and m=4 for the noise level under 25% generally 5×5 window and m=12. In the Trimmed Global Mean (TGM) Filter calculation begins by the identification of noisy pixels. On the likelihood that the preparing pixels assume P (i, j) is in the vicinity of 0 and 1, at that point the pixel in uncorrupted and if in the same event the range of event that the P (i, j) is 0 or 1 then it is determined as noisy pixels. For noisy pixels, we choose a window of size N X N and eliminate every noise affected pixel with in the chosen window, to achieve the stated elimination, it becomes important to locate the median of the rest of the pixels and replace the noise affected pixels with the median value. In the event that they chose window contains whole pixels are noisy pixels at that point noisy pixel is supplanted by trimmed global mean. Trimmed global mean is calculated by avoiding the involvement of noisy pixels in the calculation of restoration stage. So combinedly ROAD TGM is a two-phase calculation, in the primary stage the noisy pixels in the image applied for denoising is distinguished utilizing rank order absolute difference (ROAD) algorithm. In the subsequent phase, the degraded pixels are replaced by the median of the noiseless pixels in the chosen window. Trimmed global mean filter is utilized, if the chosen window comprises all of the pixels as noise affected pixels. Then TGM is calculated by eliminating the noise corrupted pixels from the window and mean of the uncorrupted pixels is obtained to replace the value with noisy pixel. In this manner, it doesn't take abundant handling time however still provides great outcomes for high noise density. The fundamental

points of interest of the ROAD TGM calculation are that it is anything but difficult to actualize in equipment and that it has low run time[8].

Decision Based Median Filter (DBMF) was proposed to deal with impulse noise (Salt and Pepper noise) as we already know this noise exists at two value 0 or 255. So, this algorithm checks the existence of noise from the beginning and identifies the noise locations and considers the values as original values if they exist between 1 and 254 and then this filter uses median value to restore the original value. Drawback of this filter is when high level of noise in introduced in image then its effectiveness is reduced.

Different Applied Median Filter (DAMF) [61] algorithm was developed to operate on a wide range of impulse noise. It can successfully denoise all range of impulse noise but its performance declines sharply as noise density becomes very high. Iterative median filter (IMF)[62] is using a fixed window base iterative mechanism for high-density impulse noise reduction. This is a very promising method and achieves the desired results. The use of a fixed window provides high speed operation of IMF algorithm but affects its accuracy when very high noise density is present in the window. Some other algorithms were also designed by using trimmed values to avoid noise effect on original value estimation. Adaptive unsymmetric trimmed shock filter (AUTSF)[63] is also a two-stage process for the detection and restoration of noisy image. This algorithm performs well on both color and grayscale images. Modified cascaded filter (MCF)[64] is a hybrid approach using trimmed median values to neglect the effect of noise on the restoration stage. This algorithm can operate well on color images affected by impulse noise. Fuzzy decision-based algorithms and supervised data-driven models were also developed to enhance the image denoising for impulse noise. Adaptive Type-2 Fuzzy Filter (FDS, fuzzy denoising for Impulse noise)[65] this is also two-stage algorithms, where first stage operates to classify pixel as good or bad and second stage, uses the weighted mean value for the restoration of noisy value. Iterative scheme-inspired network (IIN)[66] denoises on the basis of training data, the accuracy of the algorithm depends on size and type of images in the dataset. The comparative analysis of recent state of the art methods for grayscale image denoising is presented in Table 2.1. The mean PSNR and mean SSIM value of IMF method is higher than other methods in comparison which confirms it

superior performance. So, to claim high performance the proposed method is required to surpass the performance of IMF for gray scale images. On the similar ground comparative analysis of recent state of the art methods for color image denoising is presented in Table 2.2. The AUTSF methods has achieved the better mean PSNR value for color image denoising when tested for standard test image of Lena for the noise range of 10% to 90%.

Table 2.1. Comparative analysis of recent state of the art methods for grayscale image denoising.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	FDS [65]	PSNR	41.40	37.25	34.49	31.67	28.99	26.54	23.95	21.39	18.30	29.33
		SSIM	0.9894	0.9759	0.9573	0.9293	0.8858	0.8280	0.7441	0.6379	0.5020	0.83
	DAMF [61]	PSNR	42.97	39.29	36.84	34.94	33.21	31.64	30.22	28.53	25.93	33.73
		SSIM	0.9902	0.9788	0.9655	0.9494	0.9304	0.9064	0.8770	0.8370	0.7620	0.91
	IIN [66]	PSNR	–	31.43	29.50	27.62	26.39	–	–	–	–	28.74
		SSIM	–	–	–	–	–	–	–	–	–	–
	IMF [62]	PSNR	43.48	40.18	37.05	35.40	33.98	32.49	31.23	29.70	27.42	34.55
		SSIM	0.9913	0.9796	0.9675	0.9541	0.9383	0.9183	0.8953	0.8623	0.8058	0.92
Peppers	FDS [65]	PSNR	40.65	36.9	34.32	31.72	29.32	26.83	24.11	21.37	18.15	29.26
		SSIM	0.9825	0.9627	0.9396	0.9079	0.8687	0.8110	0.7355	0.6360	0.5085	0.82
	DAMF [61]	PSNR	41.52	37.89	35.67	33.95	32.55	31.31	29.79	28.28	25.87	32.98
		SSIM	0.9815	0.9606	0.9389	0.9131	0.8866	0.8541	0.8180	0.7719	0.7049	0.87
	IIN [66]	PSNR	–	27.23	26.21	24.82	23.98	–	–	–	–	25.56
		SSIM	–	–	–	–	–	–	–	–	–	–
	IMF [62]	PSNR	41.83	38.59	36.65	35.14	33.9	32.69	31.43	30.01	27.88	34.24
		SSIM	0.9858	0.9684	0.9504	0.9299	0.9091	0.8846	0.8572	0.8219	0.7700	0.90

Table 2.2. Comparative analysis of recent state of the art methods for color image denoising.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	AUTSF [63]	PSNR	41.51	38.12	35.91	34.20	32.60	31.05	29.22	27.38	24.29	32.70
	MCF [64]	PSNR	–	–	38.95	36.55	34.20	31.42	26.41	21.34	8.46	28.19

2.3 Simulation Process and Evaluation Parameters

In this area, we will discuss the parameters for evaluations and procedure of achieving our outcomes to demonstrate the picture reconstruction ability of the given algorithms. We simulated the algorithms in MATLAB and tried the execution on both 16 color images of USC South California miscellaneous dataset volume 3 [67] and 50 grayscale images of Brodatz texture dataset [68]. We included three types of image degradations that we had applied in the standard image data set for grayscale and color images of 256*256 size. Both the color and gray scale pictures are affected by 10% to 80% of noise levels with the step size of 10% and for artifacts like stripes it is starting from 2-

pixel wide strips to 9-pixel wide stripes, blotches are started from 4-pixel square to 81-pixel square for both artifacts we have 8 levels. For the statistical evaluation of denoising algorithm we are using 3 parameters PSNR, SSIM, IEF. These parameters are applied to the denoised image to achieve statistical values for interpretation of comparative analysis. Peak signal to noise ratio (PSNR) [69] can be mathematically expressed by equation (6).

$$\text{PSNR (dB)} = 10 * \log_{10}((225 \times 255))/\text{MSE}) \quad (6)$$

Where MSE is a mean square error between the denoised image and original image [70] .MSE is expressed by given equation(7).

$$\text{MSE} = \frac{1}{\text{p X q}} \sum_{\text{i,j}=1}^{\text{p,q}} (\text{f(i, j)} - \text{g(i, j)})^2 \quad (7)$$

Where 1<i<p and 1<j<q

The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processes such as data transmission and by data compression or we can say SSIM index is a way for analyzing the perceived quality of digital cinematic pictures and television, as well as other kinds of videos and digital images [71]. It is a full reference metric that requires two images from the same image capture a reference image and a processed image. SSIM is given by equation (8).

$$\text{SSIM(x; y)} = [\text{l(x; y)}]^{\alpha}[\text{c(x; y)}]^{\beta}[\text{s(x; y)}]^{\gamma} \quad (8)$$

here $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ control the relative significance.

$$\text{l(x;y)} = \frac{2\mu_x \mu_y + c1}{\mu^2{}_x + \mu^2{}_y + c1} \quad (9)$$

$$\text{c(x;y)} = \frac{2\sigma_x \sigma_y + c2}{\sigma^2{}_x + \sigma^2{}_y + c2} \quad (10)$$

$$\text{s(x;y)} = \frac{2\sigma_{xy} + c3}{\sigma_x \sigma_y + c3} \quad (11)$$

Where the variables used in equation (9), (10) and (11) are explained as μ_x and μ_y represent the means of the original and coded images, respectively, σ_x and σ_y represent the standard deviations of each of the signals and $\sigma_{xy}$is the covariance of the two photos. [51].

Image enhancement factor (IEF) [72] is the ratio of mean square error before filtering to the mean square error after filtering. It is expressed by the equation (12).

$$IEF = \frac{\sum_{ij} n_{ij} - r_{ij}}{\sum_{ij} x_{ij} - r_{ij}} \quad (12)$$

In equation (12) n is corrupted image, r is original image and x is restored image.

2.4 Conclusion

Due to the growing demand for devices with high image quality for both grayscale and color images, researchers are looking for solutions, which provide optimum results for each application. Image denoising algorithm provide this solution for salt & pepper noise for both grayscale and color images.

This chapter addresses the challenges involved in designing of image denoising algorithm for grayscale and color image dataset. Designing an optimal image denoising algorithm with accurate detection stage to provide maximum performance in comparison with other existing image denoising algorithms is one of the major challenges in the implementation of image denoising. The second major task in process of image denoising is to achieve proximity of denoised pixel value to the original pixel value of respective location. Therefore, different algorithms for achieving the high denoising characteristics for the wide range of image dataset with their advantages and disadvantages have been discussed in this study. To evaluate and validate the performance of image denoising algorithm statistically parameters are required. So, to achieve this, various parameters and techniques have been taken into consideration, such as PSNR, SSIM and IEF.

Chapter 3

Hybrid Denoising Algorithm for Both Grayscale and Color Images

Image quality is greatly affected by noise. Noisy images lead to inaccurate results as well as to segmentation and enhancement errors, making denoising vital to efficient image processing. Existing denoising methods do not work effectively on both grayscale and color images. Thus, a novel hybrid denoising algorithm is proposed herein, which works well on both image types. The proposed two-stage method involves: (i) noisy pixel detection; and (ii) noisy pixel restoration. In the first stage, the hybrid algorithm detects the noisy pixel position; while in the second stage, the corrupted value of the noisy pixel is restored using a hybrid algorithm. The proposed algorithm was used to process a variety of grayscale and color images. Comparison of these results with those of five other denoising algorithms showed that the proposed hybrid denoising algorithm outperformed these existing ones.

3.1 Introduction

In the field of image processing, digital images are often affected by several kinds of noise. There are various reasons for noise in the digital world. Some of the most common reasons include transmission of images in a noisy channel, faulty memory locations in the hardware, and malfunctioning of pixels in the camera sensors during image acquisition. The resultant noise is known as impulse noise because it corrupts the image; it has peak high and peak low values of 255 and 0, respectively. The second most prominent distortion involved in image deterioration is artifacts. The impulse noise generated during image acquisition (related to camera sensor noise produced by high temperature or low light). It is important to remove such noise, before the image is analyzed. Otherwise, it can lead to misinterpretation. The process of restoring the corrupt image is known as image denoising. The effectiveness of the image denoising process depends upon the percentage of noise the algorithm can eliminate from the

corrupt image and how close the restored pixel value is to the original pixel value. When the image denoising algorithm is not effective, important details, like edges, will not be preserved. This makes image denoising a challenging task. Over the last few decades, researchers have worked to achieve an effective and accurate denoising algorithm that can retain the important details of images, while removing the noise. To measure the performance of a denoising algorithm, various quantitative parameters are available, including the peak signal to noise ratio (PSNR), mean square error (MSE), and structural similarity index (SSIM). Although images are affected by several types of noise, it is impulse noise which largely affects images. Therefore, researchers have focused on eliminating impulse noise, while ensuring that important details are retained in the image. Impulse noise in the field of digital image processing is more commonly referred to as Salt and Pepper noise. Most existing denoising algorithms remove the noise by reducing the intensity variations within each region of the image; this produces a blurred effect which reduces the image clarity and affects other aspects of image processing, like segmentation and edge detection. To achieve higher levels of performance in image processing, we need to preserve the integrity of the original pixel values. Herein, we outline a novel hybrid algorithm for denoising. Its performance is compared with several existing algorithms, including the Median Filter (MF), Progressive Switching Median Filter (PSMF), Center Weighted Median Filter (CWMF), Decision-Based Median Filter (DBMF) and Rank-Ordered Absolute Differences Trimmed Global Mean Filter (ROAD-TGM). A large dataset of grayscale and color images having different noise levels was used to evaluate the performances of all these algorithms.

3.2 Related Work

It is necessary to explore noise models and their effects to design a better mechanism to counter noise effects in digital images. Herein, we describe various types of noise models as well as existing denoising algorithms as background to this study.

3.2.1 Digital Image Noise

Salt and Pepper noise (or impulse noise) is defined by unwanted black and white pixels. This noise occurs because of sudden and sharp changes in the digital signal during image acquisition. This noise has two defined levels (0 and 255) . The occurrence of these two values in the image is random. This noise primarily occurs at the time of image acquisition because of dust on the camera lens, faulty pixel components in the camera sensors and defective digital storage locations. Another prominent image impairments are strip lines and blotches these represent physical damage to surface of a image. These can also occur to an image digitally due to bulk damage of neighboring pixels generally due to overheating. These artifacts can be reparented mathematically by using equation 1.

$$M \times N \tag{1}$$

The strip line artifact can be represented by considering the M as width of line and N as length of line. If we consider the M=2 and N=10 then it will be considered as line have with of 2 pixels and length of 10 pixels. Similarly, for blotch artifacts the ratio of M and N will be proportionate as this artifact represent a block. In the case if M =5 and N = 5 then it will be considered as block or a patch representing artifact having width and length as 5 × 5 pixels. In this way by change values of M and N different size or patterns of strip lines and blotch artifacts can be represented in image. Typically for strip lines range of M=2 to M=9 and N=256 (total length of image). Blotch artifacts range is M=2 to M=9 and N=2 to N=9 in incremental eight steps.

3.2.2 Image Denoising Algorithms

A spatial filtering process is commonly used to remove the noise from affected images. Spatial domain filters use neighborhood operations on pixel intensity values for denoising the image. The various classes of spatial domain filtering methods are presented in the subsections.

A. Median Filter

The MF belongs to the class of nonlinear filters. It performs denoising by replacing the noisy pixel value with the median of the neighboring pixels. The median filter denoising performance can be enhanced by using a moving window, as in the mean filter. In contrast, the mean filter performs denoising by replacing the noisy pixel value with the mean value of the neighboring pixels. The median filter is more effective than the mean filter because it considers the middle intensity values of the neighboring pixels and is less affected by outliers. In this way, removing the least similar pixel in a region will not drastically change the denoised image from the original image. Therefore, the median filter is better at protecting sharp edges in the image than the mean filter. Using middle values of the surrounding pixels also ensures that the median filter corrects distributed noise in digital images. The primary disadvantage of the median filter is that the algorithm is only powerful at low noise densities. At high noise densities, the median filter frequently needs a larger window size and may not retain the important details of an image. Median filters are most popular for processing low noise level images at high denoising power. To improve its performance at low to mid noise levels, various advancements have been proposed, including the multistate median filter and homogeneity information decision-based trimmed median filters.

B. Progressive Switching Median Filter

The PSMF is an enhanced version of the median filter. Initially, an impulse detector is used to generate a sequence of binary window images. The binary window image predicts the location of noise in the observed image. An iterative procedure is applied to remove image noise. The performance of this filter for random-valued noise is very poor, but its performance for fixed-valued noise is very good.

C. Center Weighted Median Filter

The CWMF gives more weight to the central values of every window in the median filter, allowing it to retain important details while removing noise from digital images.

The CWMF performs better than a simple median filter and maintains its performance at different noise levels. The median weight is crucial to its execution.

D. Decision-Based Median Filter

The DBMF is designed to process noise-affected images by identifying the existence of impulse noise among the pixels. Firstly, it scans for impulse noise-affected pixels by evaluated whether the selected pixel value lies in between the range of the maximum and minimum possible values. This is because impulse noise-affected pixels can only have minimum and maximum values, i.e., 255 or 0. If the pixel value under evaluation is within the range of 1 to 254, then it is defined as a non-noisy pixel and no further processing is performed. While if the value of the pixel under evaluation does not fall within this range, then it is defined as a noisy pixel and is changed to the median value of the pixel values present within a selected window. When high density noise is present, it causes the calculated median value to be corrupted. In such cases, the neighborhood pixel value is used to replace the noisy pixel value. This generates a higher correlation between neighborhood pixels and noisy pixels that leads to better preservation of edges. The DBMF performs its operation within a fixed length window of pixel size 3×3, which markedly reduces its processing time compared with other filters. The main disadvantage of the DBMF algorithm is the occurrence of streaking, which occurs when the image is affected by high density noise. In this case, the noisy pixel is replaced by a neighborhood pixel value which is also noisy. Hence, this method is unable to recover details of edges satisfactorily under high noise densities.

E. Rank-Ordered Absolute Differences Trimmed Global Mean Filter

The ROAD-TGM is an algorithm which uses a trimmed global mean filter with rank-ordered absolute differences to filter scratches, random impulse noise, blotches and stripes. It involves two-stages of processing. In the first stage, the noisy pixels are identified using ROAD. In the second stage, the noisy pixels are changed to the median of the non-noisy pixels within a selected window. The TGM filter is used when the selected window contains only corrupted pixels. The algorithm uses a window of fixed

size in both filtering and detection stages. Its denoising performance is better than other existing algorithm, when restoring images affected by random-valued impulse noise. The ROAD-TGM algorithm also produces good noise filtering, when images are corrupted by the high levels of impulse noise.

3.3 Description of the Datasets

The datasets used herein consist of 50 grayscale images from the Brodatz texture dataset (Fig. 3.1) and 16 color images from the University of South California miscellaneous dataset volume 3 (Fig. 3.2). All simulations were carried out in MATLAB (MathWorks, Natick, MA, USA).

To establish the robustness of our proposed algorithm, two image degradations types (Salt and Pepper as well as artifacts) were added to images at varying levels (Fig. 3.3). Firstly, 10% noise was added, then 20% noise was added, with additional increments of 10% noise up to a noise level of 80% of the total pixels. Similarly, eight levels of artifacts are produced for straplines and blotch. Noise-affected data in both grayscale and color images were denoised using six different algorithms, i.e., the proposed algorithm, ROAD-TGM, DBMF, CWMF, PSMF and MF. By multiplying the grayscale image dataset (GD), total number of noise levels (NL), total number of algorithms (NA) and total number of parameters (NP) with each other, the total number of results for the GD (TRG) produced in our comparison of the proposed algorithm with existing algorithms could be calculated, i.e., 50(GD)*8(NL)*6(NA)*3(NP) = 7200 (TRG). Similarly, in the case of the color image dataset (CD), the total number of results for the CD (TRC) could be calculated, using 16(CD)*8(NL)*6(NA)*3(NP) = 2304 (TRC). Comparisons of the proposed algorithm with the five other existing algorithms are made using box plots. Values contained in each box were calculated by multiplying the GD, NL, selected algorithm for the box plot (SD) and the selected parameter for the box plot (SP), i.e., 50(GD)*8(NL)*1(SA)*1(SP) = 400 (GBP; total values in the box plots for the GD). Similarly, the total number of values in the box plots for the CD (CBP) was given by 16 (CD)*8(NL)*1(SA)*1(SP) = 128. Our methodology is shown schematically in Fig. 3.3.

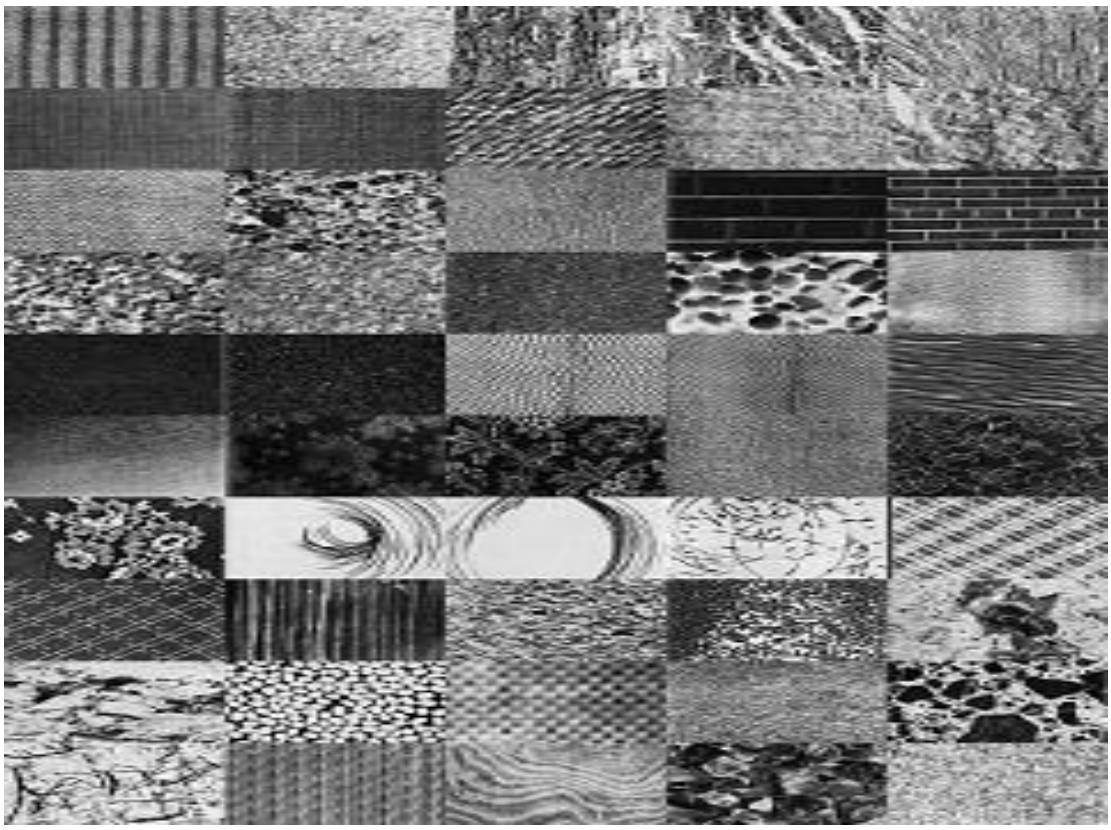


Figure 3.1. Brodatz texture dataset of grayscale images[68]

Figure 3.2. University of South California miscellaneous dataset of color images[73]

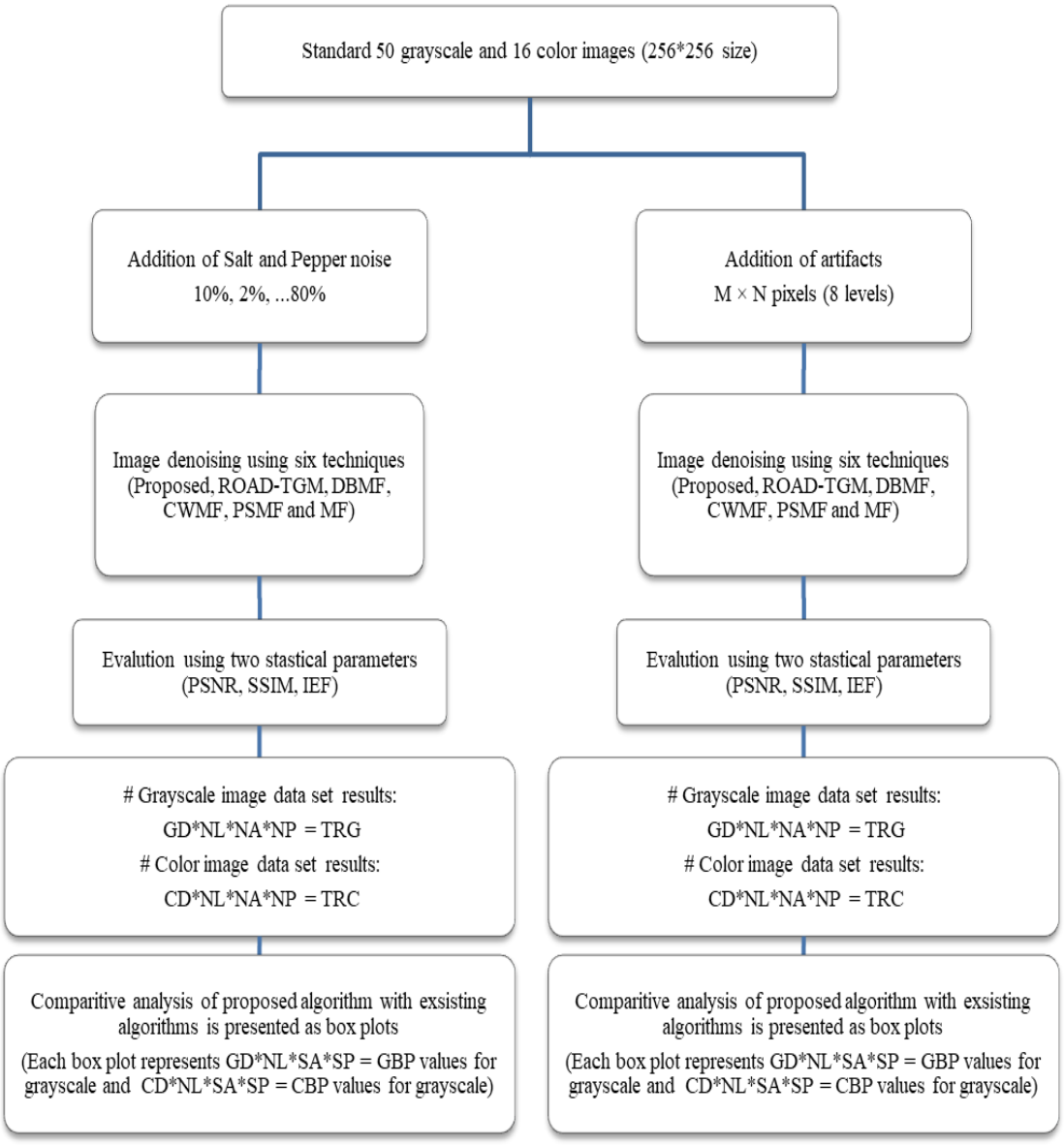


Figure 3.3. Flow chart showing the methodology used in this study. NB: Median Filter (MF), Progressive Switching Median Filter (PSMF), Center Weighted Median Filter (CWMF), Decision-Based Median Filter (DBMF) and Rank-Ordered Absolute Differences Trimmed Global Mean Filter (ROAD-TGM), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Grayscale image dataset (GD), Total number of noise levels (NL), Total number of algorithms (NA), Total number of parameters (NP), Total number of results for the GD (TRG), color image dataset (CD), Total number of results for the CD (TRC), selected algorithm for the box plot (SD) and the selected parameter for the box plot (SP).

3.4 Statistical Parameters

To evaluate the performance of the various denoising algorithms, both PSNR and SSIM values were used:

(a) Peak signal to noise ratio and image enhancement factor

The PSNR [29] can be mathematically expressed as:

$$\text{PSNR (dB)} = 10 * \log_{10}((225 \times 255))/\text{MSE}), \quad (2)$$

where MSE is the mean square error between the denoised image and original image . The MSE is given by:

$$\text{MSE} = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[f(x,y) - g(x,y)]^2, \quad (3)$$

where N and M are the dimensions of the images; $f(x,y)$ is the original image and $g(x,y)$ is the approximated image. IEF is ratio of mean square error before filtering to the mean square error after filtering.

(b) Structural Similarity Index

The SSIM is a metric that quantifies image quality degradation caused by processes, such as data transmission and data compression. The SSIM is used to analyze the perceived quality of digital stored or real time image as well as other kinds of videos and digital images. It is a full reference metric that requires two images for its calculation. The first image is the original image, which is used as the reference image; while the second image is the processed image. The SSIM measures luminance (l), contrast (c) and structure (s) of images x and y, giving:

$$\text{SSIM}(x;\ y) = [l(x;\ y)]^{\alpha}[c(x;\ y)]^{\beta}[s(x;\ y)]^{\gamma}, \quad (4)$$

here α, β and γ are the weights, set to α> 0, β> 0 and γ> 0, to control the relative importance of l, c and s, defined in Eq. (5)–(7) below.

$$l(x;y) = \frac{2\mu_x\mu_y + c1}{\mu^2{}_x + \mu^2{}_y + c1}, \quad (5)$$

$$c(x;y) = \frac{2\sigma_x\sigma_y + c2}{\sigma^2{}_x + \sigma^2{}_y + c2}, \quad (6)$$

$$s(x;y) = \frac{2\sigma_{xy} + c3}{\sigma_x\sigma_y + c3}, \quad (7)$$

where the variables μ_x and μ_y in Eq. (5) represent the means of the original and denoised images x and y, respectively. The variables σ_x and σ_y represent the standard deviations, while $\sigma^2{}_x$ and $\sigma^2{}_y$ are the variances of images x and y, respectively. The variable σ_{xy} is the covariance of these images.

In situations where the denominator is close to zero, the constants C1, C2 and C3 are introduced. For an 8-bit grayscale image composed of L = 256 gray levels, where L is the dynamic range of the pixel values; C1 =$(k_1L)^2$, C2 = $(k_2L)^2$ and C3 = C2/2, where k_1 and k_2 are constants and their default values are k_1 = 0.01 and k_2= 0.03. When C1 = C2 = 0, the metric is reduced to the universal quality index.

3.5 Algorithm Selection for the Hybrid Model

The proposed algorithm is hybrid in nature, representing a combination of two existing algorithms. It is important to mention that the proposed algorithm incorporates the advantages of both existing algorithms, yielding better results. The steps involved in selecting algorithms for the proposed method were as follows:

Step 1: Consider the most prevalent existing denoising algorithms, i.e., ROAD-TGM, DBMF, CWMF, PSMF and MF

Step 2: Select noise type (Salt and Pepper noise) and set the initial noise level at 10%

Step 3: Add noise (10%) to all grayscale images of the dataset

Step 4: Apply existing denoising algorithms for denoising these noisy images

Step 5: Calculate the PSNR, SSIM and IEF values for all denoised images for each respective algorithm

Step 6: Increase the noise level to 20% in Step 2 and repeat Step 3 to Step 5; keep incrementing the noise and repeating Step 6 until it reaches a level of 80%

Step 7: Calculate PSNR, SSIM and IEF values for all images for all noise levels

Step 8: Repeat Step 2 but change the impairment type to Artifacts

Step 9: Repeat Step 3 to Step 7 for Artifacts noise-affected images

Step 10: Compare all denoising algorithms for each noise type (Salt and Pepper as well as Artifacts) and for all noise levels (10%–80%) using PSNR, SSIM and IEF values

Step 11: Rank the existing denoising algorithms based on their PSNR, SSIM and IEF values

Step 12: Select the two top-performing denoising algorithms for inclusion in the proposed algorithm.

Clearly, all the above-mentioned steps also must be implemented for color images because the two top-ranked algorithms may be different for grayscale and color images. Based on our results for grayscale images, the ROAD-TGM and DBMF were the two top-performing algorithms, while ROAD-TGM and CWMF were the two top-performing algorithms for color images. These algorithm pairs were selected for the proposed hybrid algorithm for grayscale and color images. Once the two best performing algorithms were selected, the next step was to integrate these algorithms into the proposed method. The proposed hybrid scheme performs the denoising operation in two stages, involving (i) noisy pixel detection; and (ii) noisy pixel restoration, as outlined below.

I. Noisy pixel detection:

The first stage of the proposed method was to detect noisy pixels. As outlined above, noise was added to all images (grayscale and color) in incremental steps from 10% up to 80%. The proposed algorithm was applied to all noisy images. To understand the proposed algorithm process in detail, we firstly describe how to denoise grayscale images. The detection of noisy pixels in grayscale images is shown in the flow chart in Fig. 3.4.

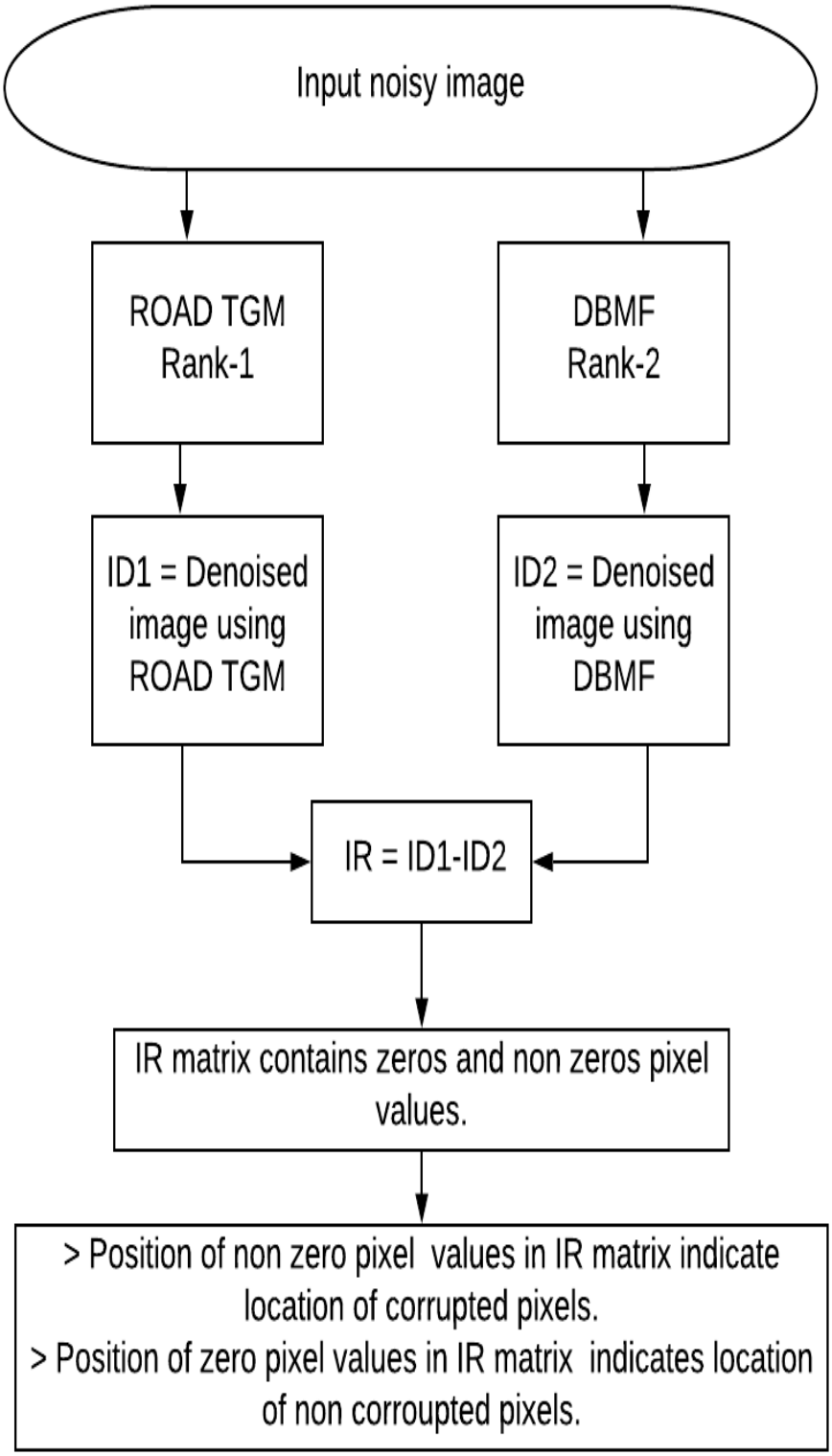


Figure 3.4. Flow chart of the proposed hybrid model. NB: Decision-Based Median Filter (DBMF) and Rank-Ordered Absolute Differences Trimmed Global Mean Filter (ROAD-TGM) and resultant matrix (IR).

As shown in Fig. 3.4, the noisy grayscale image was input into ROAD-TGM and DBMF, being the two top-ranked denoising algorithms selected for our hybrid algorithm. Output matrices of these two denoising algorithms (ROAD-TGM and DBMF) were ID1 and ID2, respectively. It is obvious that the values of pixels, which were non-corrupted in the original matrix (after mixing noise) remained unchanged in both ID1 and ID2. Meanwhile, the values of all corrupted pixel were changed in ID1 and ID2, with both denoising algorithms giving different denoised values of the corrupted pixels. The two matrices ID1 and ID2 were subtracted to get the resultant matrix 'IR'. In the matrix IR, if the value of a pixel is zero, this means that the pixel was not corrupted, and has the same value in ID1 and ID2. But any corrupted pixel in matrix IR will not be zero, as its corresponding values in ID1 and ID2 will not be the same. Therefore, the non-zero values in matrix IR identify noisy pixels. In this way, we can identify the positions of all noisy pixels in the images. An example of this process is outlined below.

172	163	168
168	158	164
160	165	152

(a)

172	0	168
168	158	255
160	0	152

(b)

172	0	168
168	158	162
160	162	152

(c)

172	166	168
168	158	255
160	170	152

(d)

0	-166	0
0	0	-93
0	-8	0

(e)

172	143	168
168	158	235
160	165	152

(f)

172	166	168
168	158	167
160	165	152

(g)

172	158	168
168	158	235
160	165	152

(h)

0	8	0
0	0	-68
0	0	0

(i)

Figure 3.5. (a) Original matrix, (b) Matrix corrupted by Salt and Pepper noise (highlighted pixels are corrupted by noise), (c) Matrix ID1 restored by ROAD-TGM from (b), (d) Matrix ID2 restored by DBMF from (b), (e)Resultant matrix IR (IR = ID1−ID2, for Salt and Pepper noise), (f) Matrix corrupted by non-extreme noise, (g) Matrix ID1 restored by ROAD-TGM from (f), (h) Matrix ID2 restored by DBMF from (f), (i) Resultant matrix IR (non-extreme). NB: DBMF, Decision-Based Median Filter; and ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter.

As shown in Fig. 3.5(a), a 3×3 matrix is the original matrix. This original matrix was corrupted with Salt and Pepper noise, as shown in Fig. 3.5(b). Noise-affected areas are highlighted in each subsequent matrix. The noisy matrix was then denoised by the two top-ranked algorithms, i.e., the ROAD-TGM and DBMF (NB: the proposed hybrid algorithm does not have any prior information about noise locations). Both the denoised matrices of ROAD-TGM (ID1) and DBMF (ID2) were subtracted to produce the resultant matrix (IR = ID1−ID2). Its zero-valued pixels indicate non-corrupted pixels, while its non-zero values indicate the location of noise-affected pixels. Hence, noise-affected areas in the 3×3 resultant matrix are given by IR(1,2), IR(2,3) and IR(3,2) in the case of Salt and Pepper noise. Similarly, in the case of non-extreme noise, the noise-affected areas are IR(1,2) and IR(2,3) (Fig. 3.5(i)). In this way, we were able to detect noisy pixels.

II. Noisy Pixel Restoration:

Once a noisy pixel is detected the next stage is restoration of the value of the noisy pixel, which should be as close as possible to the original value of the pixel (prior to the corruption). To illustrate this stage, we present two cases, detailing the process for Salt and Pepper and non-extreme noise-affected images.

Case 1: This case involved restoration of pixel values corrupted by Salt and Pepper noise. In the IR for Salt and Pepper noise (Fig. 3.5(e)), there are three corrupt pixels. We can compare the corresponding pixel positions of these corrupted pixels in matrices ID1 and ID2 for the Salt and Pepper noise case (Fig. 3.5(c) and (d), respectively). To show the process of noisy pixel restoration in the case of the Salt and Pepper noise, we explore the following two scenarios:

1. If only one of the two top-ranked algorithms (ROAD-TGM and DBMF) is able to detect and denoise the Salt and Pepper noise, then under such conditions, only

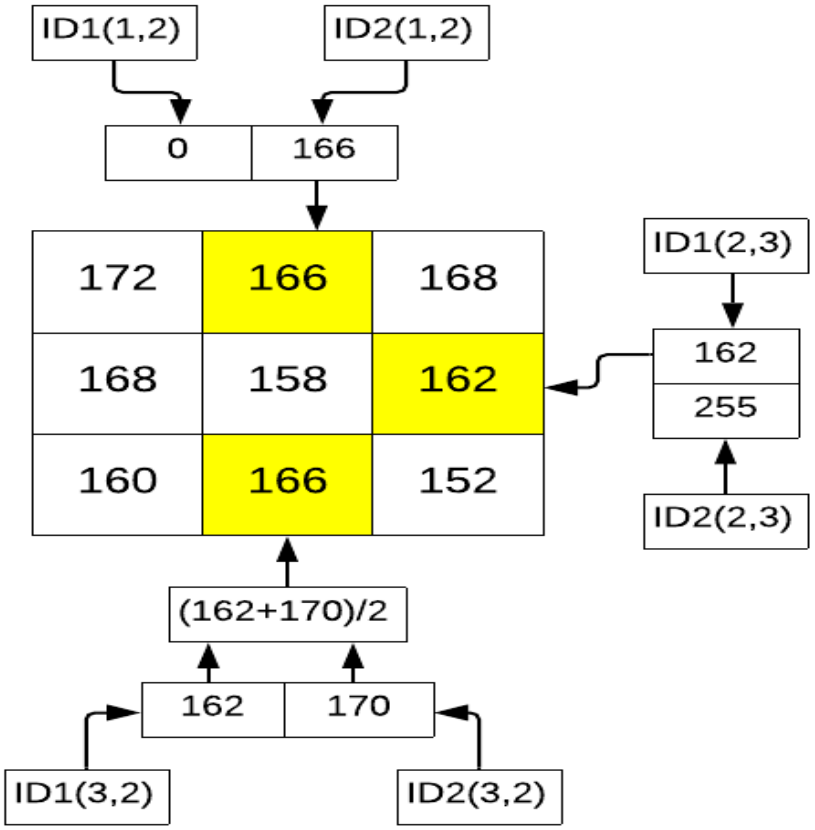


Figure 3.6. Denoised matrix (IH) restored using the proposed hybrid algorithm on Salt and Pepper noise. NB: ID1, matrix for image denoised by Rank-Ordered Absolute Differences Trimmed Global Mean Filter (ROAD-TGM); and ID2, matrix for image denoised by Decision-Based Median Filter (DBMF).

the value of the denoised pixel is used by the proposed algorithm to produce denoised matrix IH. In the example shown in Fig. 3.6, when the value of ID1(1,2) is equivalent to zero, it means the ROAD-TGM failed to detect the noise at this location. At this same location, the ID2 derived from the DBMF algorithm successfully denoised the pixel, shown as ID2(1,2) and revalued it as 166. Therefore, the proposed algorithm selects the value denoised by the DBMF algorithm (i.e., 166) for the denoised matrix, i.e., IH (1,2) = 166. Similarly, at pixel location ID1(2,3), the ROAD-TGM successfully denoised this pixel, replacing the noisy value (255) with a computed pixel value (162), while the

DBMF failed to detect noise at this location, yielding ID2(2,3) = 255. In this case, the proposed algorithm denoised the pixel location by using the denoised value of ROAD-TGM, i.e., ID1(2,3) = 162.

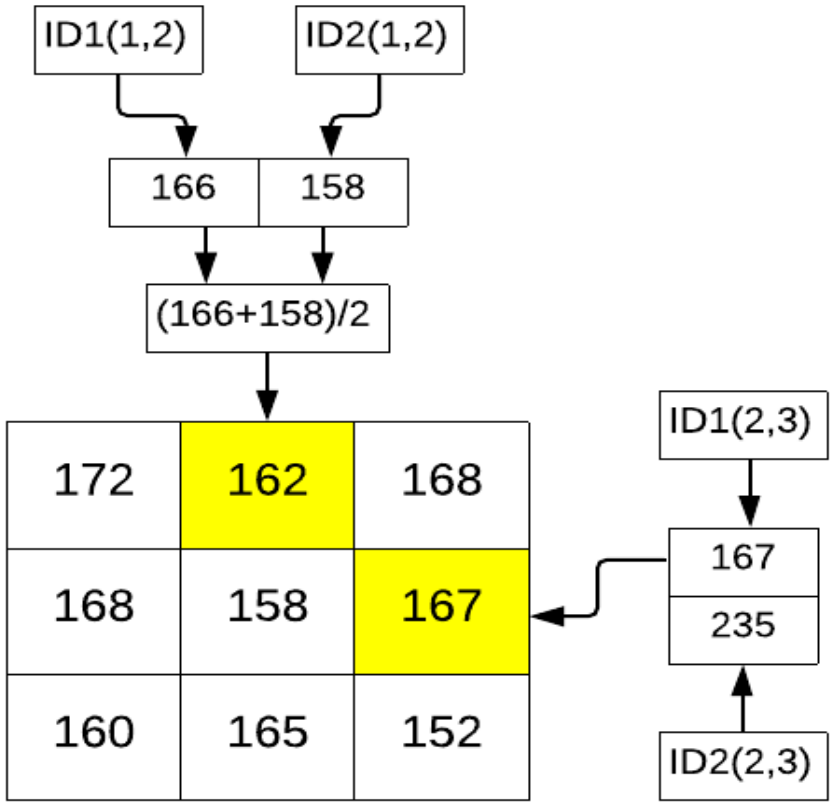


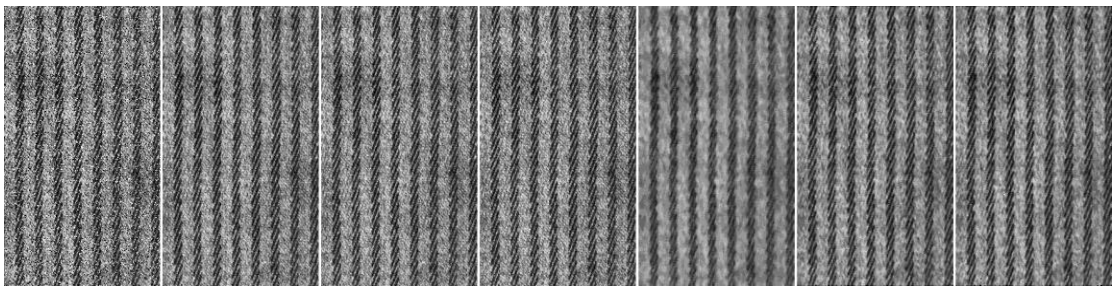
Figure 3.7. Denoised matrix (IH) restored by the proposed algorithm following non extreme noise addition. NB: ID1, matrix for image denoised by Rank-Ordered Absolute Differences Trimmed Global Mean Filter (ROAD-TGM); and ID2, matrix for image denoised by Decision-Based Median Filter (DBMF).

2. If both the algorithms (ROAD-TGM and DBMF) are able to detect and denoise the Salt and Pepper noise successfully, but the values restored by both algorithms after denoising are different because of their different correction functions, then the proposed algorithm replaces the noisy pixel with an average of the values computed by both ROAD-TGM and DBMF algorithms. In the example shown in Fig. 3.6, ID1(3,2) = 162 and ID2(3,2) = 170 are the two

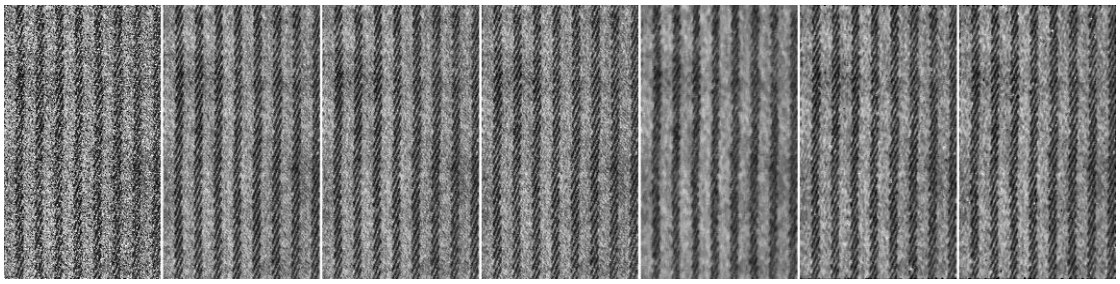
different values restored by algorithms ROAD-TGM and DBMF, respectively. To denoise the Salt and Pepper noise, the proposed algorithm replaces the noisy pixel location in the final denoised matrix IH with an average value of both algorithm results, i.e., IH (3,2) = (162+170)/2.

Case 2: This case involved restoration of pixels affected by non-extreme noise (1 to 254), as highlighted in Fig. 3.5(f). The denoised matrix (IH) computed by the proposed hybrid algorithm for this case is shown in Fig. 3.7. In matrix IR for the non-extreme noise case (Fig. 3.5(i)), there are two noise-affected pixels: IR (1,2) and IR (2,3). Herein, we consider the denoising process of the proposed algorithm for non-extreme noise under two scenarios:

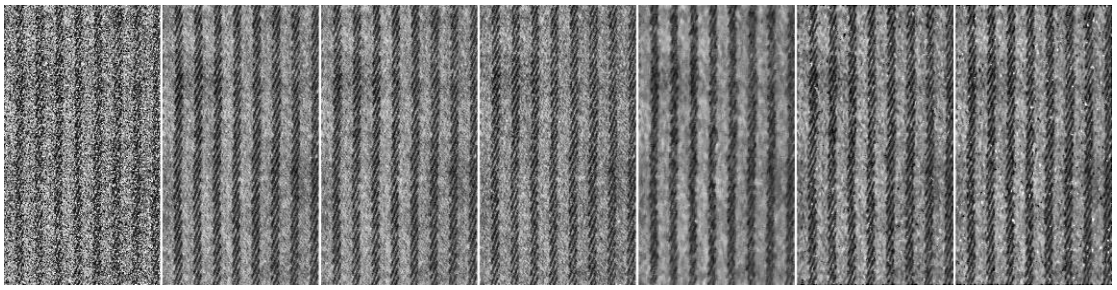
1. If only one of the top two-ranked algorithms (ROAD-TGM and DBMF) is able to detect and correct a pixel corrupted by non-extreme noise, then under these conditions the proposed algorithm restores the values in the final matrix IH by using the successfully denoised value of the respective algorithm. In the example shown in Fig. 3.7, values of the denoised matrix ID1(2,3) and ID2(2,3) produced by ROAD-TGM and DBMF algorithm are different. In this case, the proposed algorithm first subtracts the denoised values of both algorithms to calculate the difference(diff)value, i.e., diff = mod(ID1(2,3)−ID2(2,3)). If the calculated difference value is greater than or equivalent to 20, this means one of the algorithms failed to detect the non-extreme noise. To identify the successfully denoised pixel value, the proposed algorithm considers the immediate neighbors (ID1 (1,3) = 168, ID1(2,2) = 158 and ID2(2,3) = 152, which are close to ID1(2,3), and calculates a neighborhood mean, i.e., (168+158+152)/3 = 159.33 (using only non-corrupted neighboring values). The proposed algorithm replaces the noisy pixel in the final matrix IH with ID1(2,3) as the accepted denoised pixel value, while ID2(2,3) is ignored.



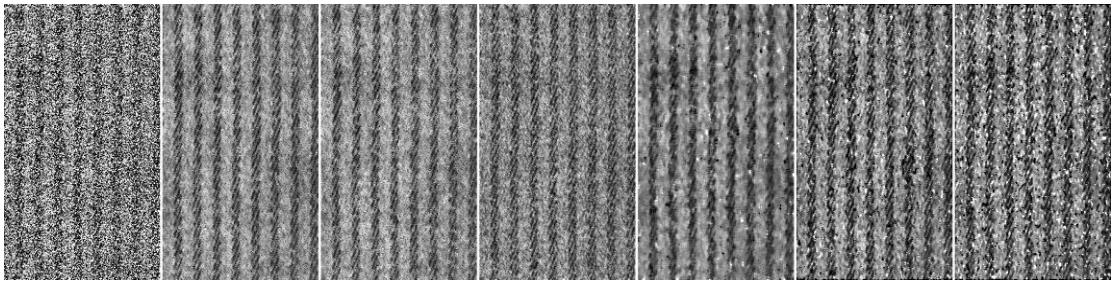
(aa) (ab) (ac) (ad) (ae) (af) (ag)



(ba) (bb) (bc) (bd) (be) (bf) (bg)



(da) (db) (dc) (dd) (de) (df) (dg)



(ea) (eb) (ec) (ed) (ee) (ef) (eg)

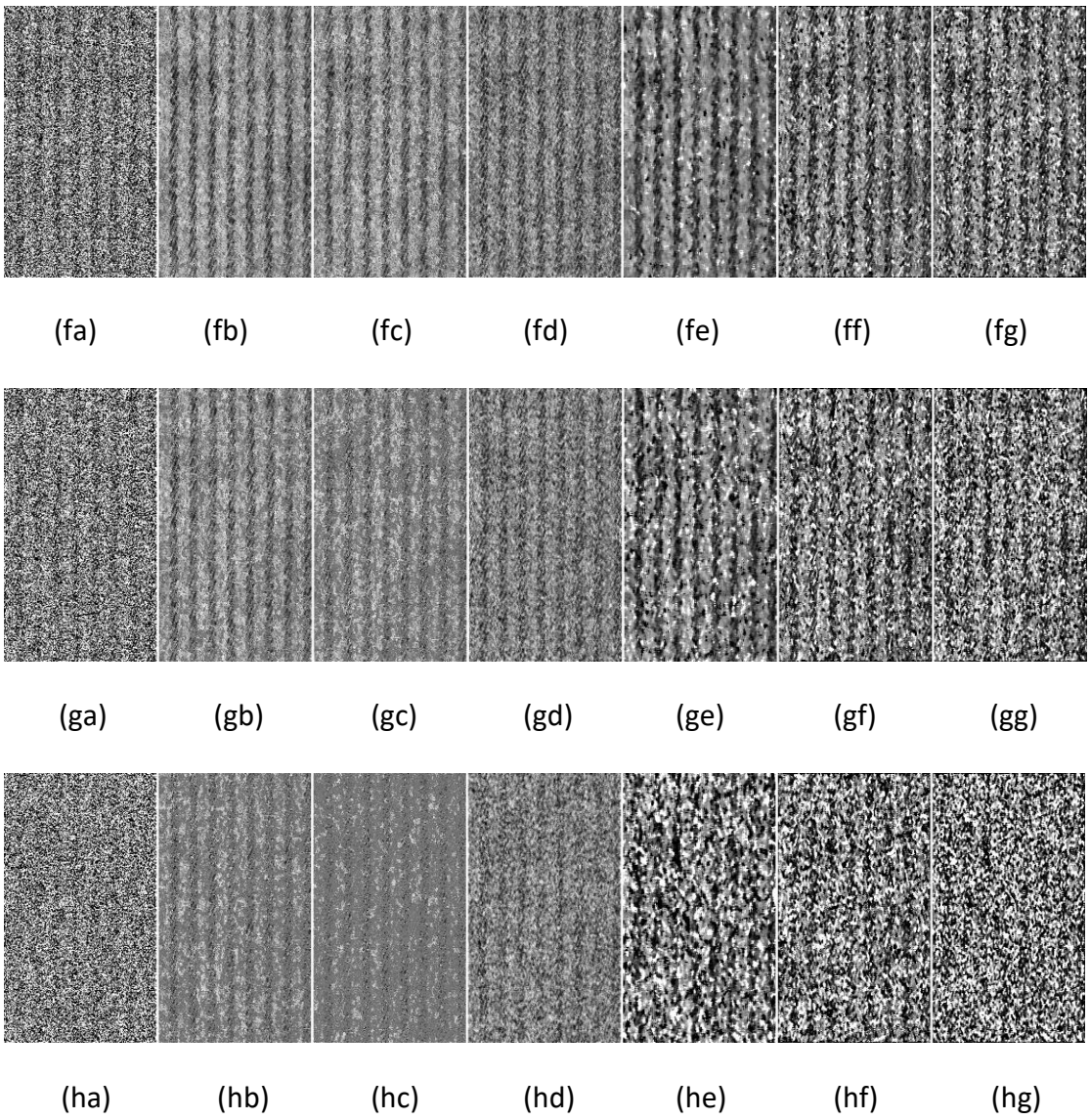


Figure 3.8. (aa–ha) Texture image corrupted by 10%–80% Salt and Pepper noise, (ab–hb) Image restored using proposed hybrid technique from (aa–ha), (ac–hc) Image restored using the ROAD-TGM from (aa–ha), (ad–hd) Image restored using the DBMF from (aa–ha), (ae–he) Image restored using the CWMF from (aa–ha), (af–hf) Image restored using the PSMF from (aa–ha), (ag–hg) Image restored using the MF from (aa–ha). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

2. If ROAD-TGM and DBMF are able to detect and denoise the non-extreme noise successfully, but the values restored by both algorithms are different because of the

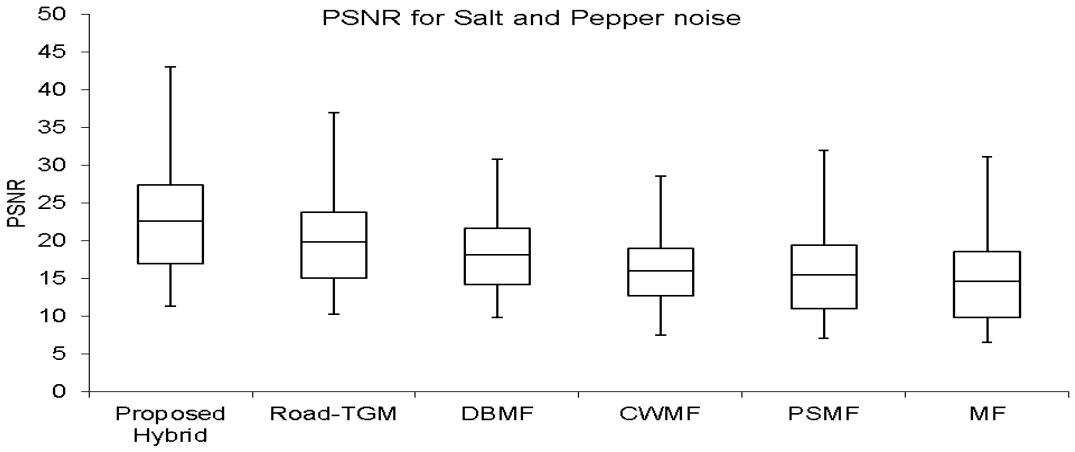
different functions used by each algorithm, then the proposed algorithm replaces the noisy pixel with an average computed from both ROAD-TGM and DBMF values. In Fig. 3.7, the two algorithms produced respective values of ID1(1,2) = 166 and ID2(1,2) = 158. The proposed algorithm computes the difference value (‘diff’value = 8). If the difference value is less than 20, this means that both algorithms detected and denoised the non-extreme noise successfully. Therefore, in the matrix IH, the proposed algorithm replaces the noisy pixel with the mean value, (166+158)/2 = 162.

Similarly, a hybrid model was prepared for color images, wherein the combination of ROAD-TGM and CWMF algorithms was used in the proposed hybrid algorithm. Same process for grayscale and color images can be repeated for strip lines and botch artifacts.

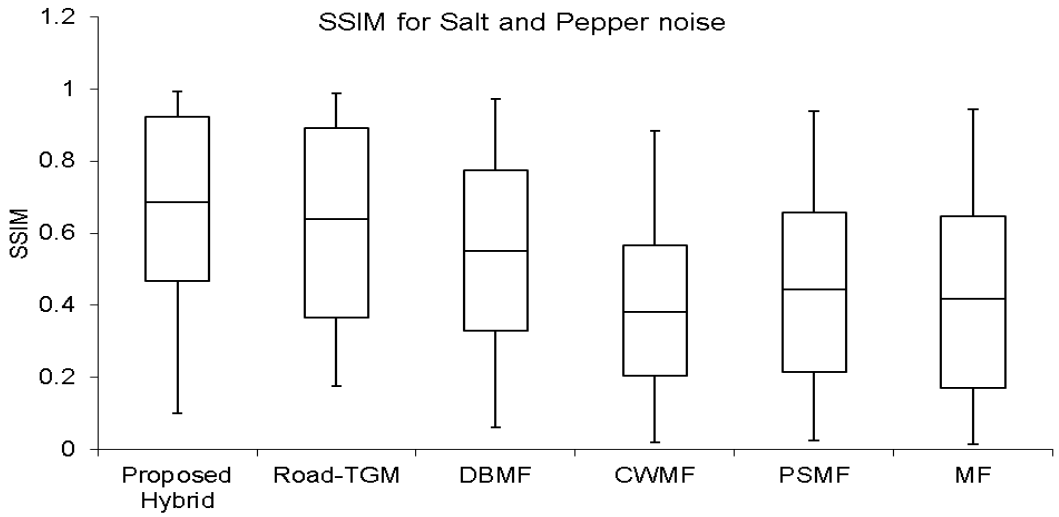
3.6 Results and Discussion

This section discusses the evaluation of the proposed hybrid method and those of existing algorithms. As discussed earlier, for each image, different noise types were incrementally added from 10% noise level to 80% noise level, forming sequences of increasingly corrupted images having either salt and pepper or artifacts types. A comparative analysis of the proposed algorithm with existing denoising algorithms on images corrupted by 10% to 80% Salt and Pepper noise is shown in Fig. 3.9–3.10. The PSNR, SSIM and IEF values were used to evaluate the performance of the proposed and existing algorithms on all grayscale images and color images. The PSNR, SSIM and IEF values for each image and for each noise level were calculated for the denoised image produced by both the proposed and existing algorithms; these are shown in box plots and table for all noise levels and for all images. The results prove that the hybrid model was better than all other algorithms, producing better PSNR, SSIM and IEF values in all cases. Typically, the overall mean PSNR value of the proposed hybrid model was 22.56, which has higher than values for all other models: ROAD-TGM (PSNR = 19.82), DBMF (PSNR = 18.02), CWMF (PSNR = 15.99), PSMF (PSNR = 15.45), and MF (PSNR = 15.18) (Fig. 3.9). The box plots also show that the proposed

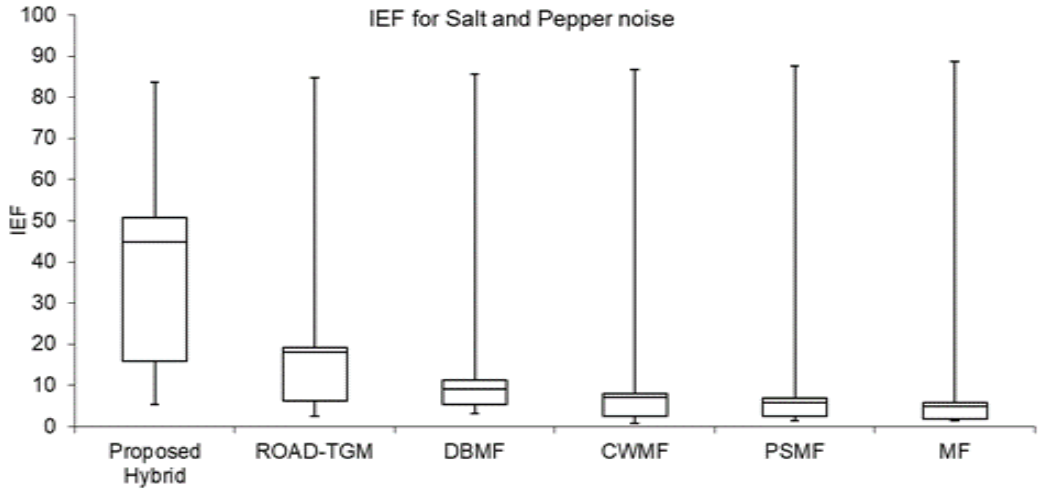
hybrid model outperformed all other algorithms in terms of the SSIM parameter. The mean SSIM value of the proposed hybrid model was 0.68, which was higher than values for all other models: ROAD-TGM (SSIM = 0.64), DBMF (SSIM = 0.55), CWMF (SSIM = 0.38), PSMF (SSIM = 0.44) and MF (SSIM = 0.41). Similarly, IEF values of proposed hybrid model was 44.76 which is much higher than other algorithms. The results of various denoising techniques on images corrupted by artifacts and noise are shown in Table 3.1 for grayscale images. Again, the performance of the proposed hybrid model was better than all other algorithms in terms of denoising the artifacts, producing better PSNR, SSIM and IEF values in all cases. The mean PSNR, mean SSIM and mean IEF values of the proposed hybrid model for strip lines were 24.46, 0.091, 16.82 and for blotch artifact values were 45.06, 0.99, 19.76 respectively. These were higher than those of all other algorithms. The Table 3.1 clearly demonstrate the superiority of the proposed hybrid model compared to other existing algorithms. Our comparative analysis of denoising algorithms on color images affected by Salt and Pepper noise and artifacts are shown in Fig. 3.10 and Table 3.2, respectively. In the case of impulse noise, the proposed hybrid model produced a PSNR value of 25.83, SSIM value of 0.76 and 72.23, which were higher than those of ROAD-TGM, DBMF, CWMF, PSMF, and MF algorithms. In the case of artifacts, the proposed algorithm outperformed the other existing algorithms, for blotch artifact the mean PSNR value of 40.91, SSIM value of 0.996 and IEF value of 26.12. similarly, for strip lines artifact values are 40.91, 0.96 and 51.45 as shown in Table 3.2.



(a)



(b)

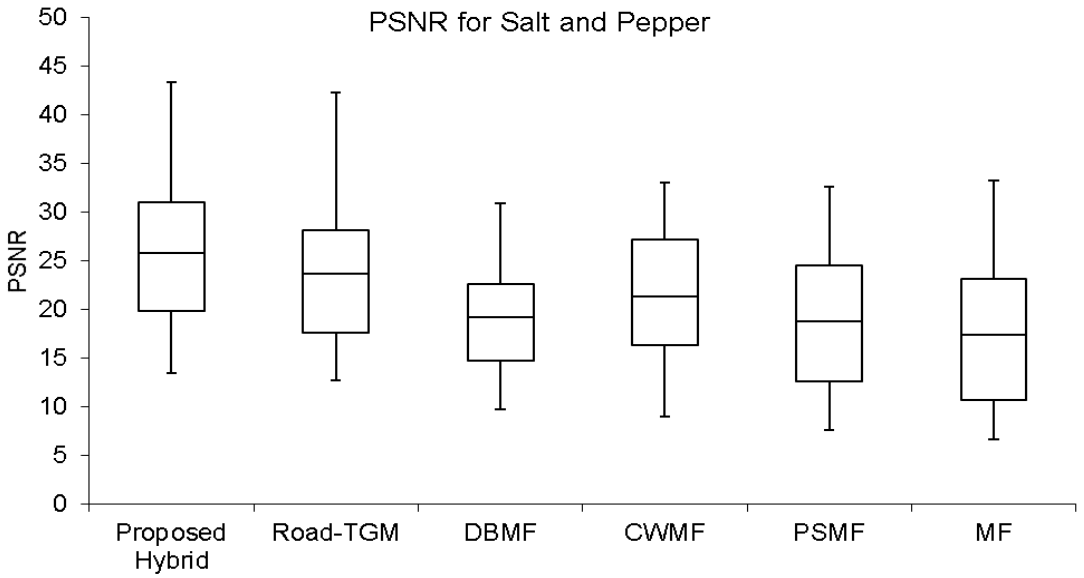


(c)

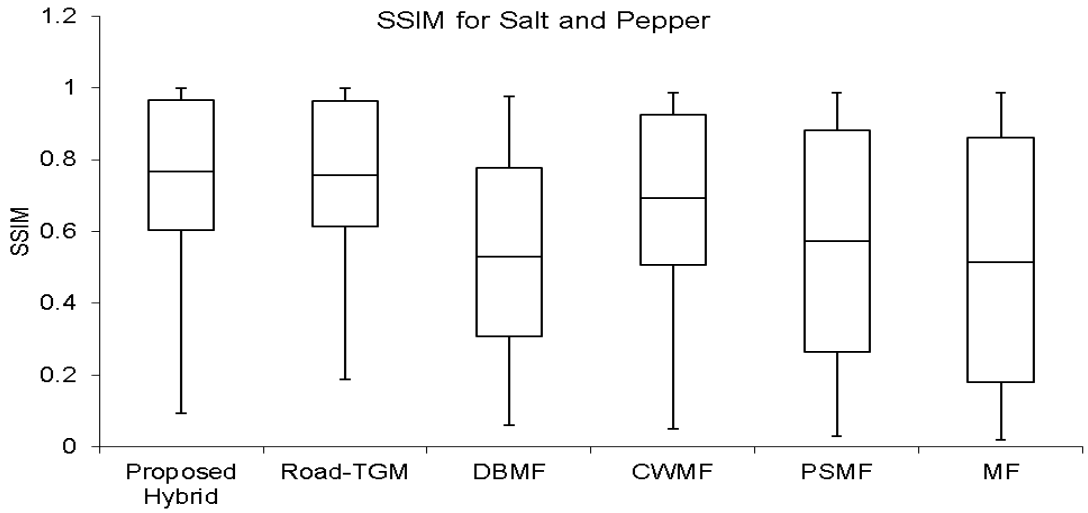
Figure 3.9. Comparative analysis of algorithms for 10% to 80% Salt and Pepper noise-affected grayscale image (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (C) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

Table 3.1. Comparative analysis of proposed hybrid algorithm with existing algorithms for grayscale images.

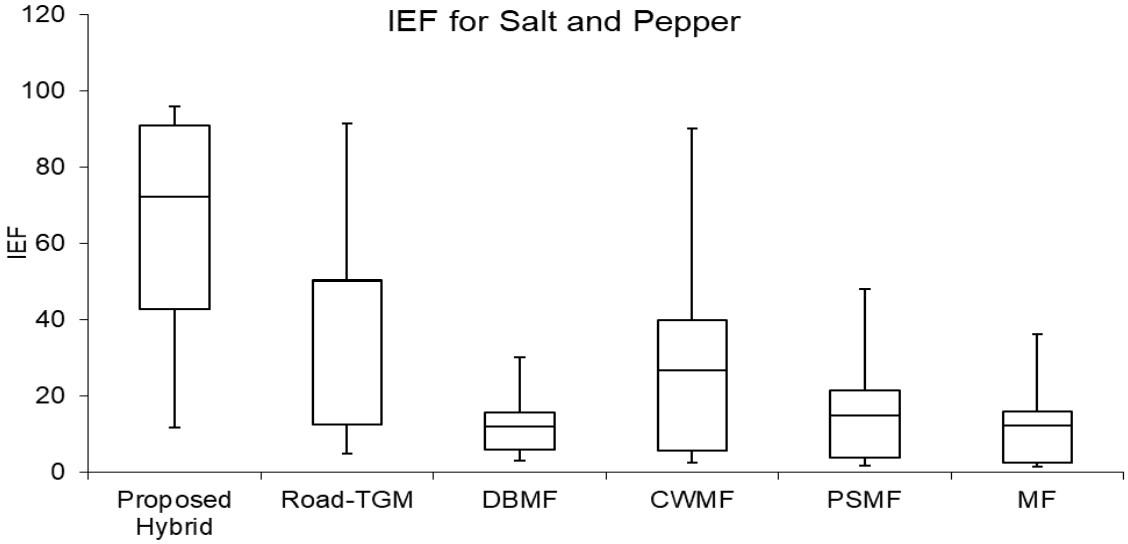
Image type	Noise type	Parameter	Proposed Hybrid	ROAD-TGM	DBMF	CWMF	PSMF	MF
Grayscale dataset 50 images	Salt and Pepper (10%-80%)	Mean PSNR	**22.568**	19.825	18.002	15.966	15.456	15.181
		Mean SSIM	**0.686**	0.640	0.551	0.383	0.444	0.419
		Mean IEF	**44.769**	17.960	9.158	7.007	5.736	4.818
	Blotchs (2x2 - 9x9)	Mean PSNR	**45.060**	44.214	36.299	19.442	22.636	22.713
		Mean SSIM	**0.999**	0.998	0.997	0.525	0.765	0.770
		Mean IEF	**19.768**	14.447	1.643	0.068	0.128	0.130
	Strip lines (2 - 9)	Mean PSNR	**24.464**	23.723	15.533	13.235	13.706	13.719
		Mean SSIM	**0.910**	0.875	0.784	0.394	0.578	0.582
		Mean IEF	**16.829**	10.688	1.278	0.781	0.841	0.844



(a)



(b)



(c)

Figure 3.10. Comparative analysis of algorithms for Salt and pepper noise-affected color image (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (C) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

Table 3.2. Comparative analysis of proposed hybrid algorithm with existing algorithms for color images.

Image type	Noise type	Parameter	Proposed Hybrid	ROAD-TGM	DBMF	CWMF	PSMF	MF
Color image dataset 16 images	Salt and Pepper (10%-80%)	Mean PSNR	**25.839**	23.700	19.170	21.323	18.733	17.403
		MeanSSIM	**0.767**	0.756	0.531	0.694	0.574	0.513
		Mean IEF	**72.233**	50.175	11.958	26.639	14.837	12.223
	Blotchs (2x2 - 9x9)	Mean PSNR	**58.833**	44.064	39.672	27.423	30.588	30.635
		MeanSSIM	**0.996**	0.994	0.994	0.916	0.958	0.959
		Mean IEF	**26.124**	15.670	1.533	0.164	0.292	0.294
	Strip lines (2 - 9)	Mean PSNR	**40.915**	27.876	16.300	15.402	15.048	15.051
		MeanSSIM	**0.960**	0.939	0.788	0.711	0.734	0.735
		Mean IEF	**51.452**	25.358	1.301	1.138	0.961	0.962

3.7 Conclusion

In this chapter, a novel hybrid denoising algorithm was built from the two best performing algorithms for grayscale and color images, based on a comparative analysis of ROAD-TGM, DBMF, CWMF, PSMF and MF. The hybrid algorithm selected ROAD-TGM and DBMF for grayscale images; and ROAD-TGM and CWMF for color images for (i) noisy pixel detection; and (ii) noisy pixel restoration. In the first processing stage, the hybrid algorithm detected the noisy pixel position, while in the second stage, the original value of the detected noisy pixel was restored using values of both inbuilt algorithms. To evaluate the performance of our proposed hybrid algorithm, different levels of noise from 10% to 80% were added to both grayscale and color images, and their PSNR, SSIM and IEF values were calculated after denoising. The PSNR, SSIM and IEF values indicated that the proposed denoising algorithm outperformed all five other algorithms tested.

Chapter 4

Spatially Adaptive Image Denoising Via Enhanced Noise Detection Method for Grayscale and Color Images

Keeping in view the variety of the applications, image denoising still remains the unexplored territory for the researchers. There are many pros and cons in existing denoising algorithms. The two prime cons of image denoising algorithms are (i) Over and under detection of noisy pixels (ii) Low performance at high noise levels. So, in order to overcome these existing issues, a spatially adaptive image denoising via enhanced noise detection method (SAID-END) is proposed for grayscale and color images. The denoising is achieved using a two-stage sequential algorithm, the first stage ensures accurate noise estimation by eliminating over and under detection of noisy pixels. The second stage performs image restoration by considering non-noisy pixels in estimation of the original pixel value. To enhance the accuracy while denoising high-density impulse noise and artifacts, both noise estimation and restoration stages are using a spatially adaptive window (window expands to spatially connected area), the size of the window depends upon the noise level in the vicinity of the reference noisy pixel. The two stages of the proposed method are referred to as (i) Enhanced adaptive noise detection (ii) Non-corrupted pixel sensitive adaptive image restoration. The proposed method is evaluated by two test steps to ensure its versatility and robustness. In the first step, the proposed method is tested on a wide standard data set of color and grayscale images affected by impulse noise and artifacts. The results of proposed method are compared with well-known methods compatible for denoising impulse noise and artifacts. In the second step, the results of proposed method are compared with the recent state of the art algorithms for traditional test images. The result shows that the proposed method outperforms the existing denoising methods when applied to grayscale and color images.

4.1 Introduction

Various applications like recognition, edge detection, medical imaging and satellite imaging require high-quality noise free images. So, it is a necessity to denoise the images as preprocessing to such applications. It is important to retain information such as edges, texture and structure details while performing the image denoising. Specifically, edges are extremely important in the biomedical field analysis like forensic examination and hairline cracks in bones. Noise and artifacts are two major contributors in image quality deterioration. In digital images, salt & pepper noise deteriorate the image quality by introducing extreme values 0 and 255[39]. The artifacts represent deterioration of image quality due to spots and scratches.

Median filter (MF)[74], [75] is one of the most popular standard non-linear filter to remove impulse noise. This filter performs well for low level of noise but the performance of filter reduces drastically as noise level increases. Modified forms of MF are commonly used and preferred for image denoising till date. Decision Based Median Filter (DBMF), Center Weighted Median Filter (CWMF), Progressive Switching Median Filter (PSMF), Different Applied Median Filter (DAMF), and Iterative Mean Filter (IMF) are some examples of methods modified from MF. DBMF [8], [76], [77] is an effective method for denoising low and mid noise density affected images. This method introduces blurring and artifacts at high noise levels. The CWMF [58], [78] algorithm provides better proximity to original values by providing more weight to center values of the window. This method achieves better visual performance but as more values get corrupted in the selected window due to high noise density, its performance dilutes. PSMF[11], [56], [61] is a two-stage cascaded process, first noise is detected and then restored iteratively. Pixel values of the current iteration are considered for calculation of pixel value in the next iteration. This method can denoise impulse noise and low level of blotches. DAMF[61] algorithm was developed to operate on a wide range of impulse noise. It can successfully denoise all range of impulse noise but its performance declines sharply as noise density becomes very high. IMF[62] is using a fixed window base iterative mechanism for high-density impulse noise reduction. This is a very promising method and achieves the desired results. The use of a fixed window provides high speed operation of IMF

algorithm but affects its accuracy when very high noise density is present in the window. Some other algorithms were also designed by using trimmed values to avoid noise effect on original value estimation. Rank Ordered Absolute Differences with Trimmed Global Mean filter (ROAD-TGM)[8] is a window base two-stage algorithm, where the first stage focus on noise detection and the second stage ensures the desired restoration. This algorithm uses TGM when all values of the selected window are noisy. Similarly, adaptive unsymmetric trimmed shock filter (AUTSF)[63] is also a two-stage process for the detection and restoration of noisy image. This algorithm performs well on both color and grayscale images. Modified cascaded filter (MCF)[64] is a hybrid approach using trimmed median values to neglect the effect of noise on the restoration stage. This algorithm can operate well on color images affected by impulse noise. Fuzzy decision-based algorithms and supervised data-driven models were also developed to enhance the image denoising for impulse noise. Adaptive Type-2 Fuzzy Filter (FDS, fuzzy denoising for Impulse noise)[65] this is also two-stage algorithm, where first stage operates to classify pixel as good or bad and second stage, uses the weighted mean value for the restoration of noisy value. Iterative scheme-inspired network (IIN)[66] denoises on the basis of training data, the accuracy of the algorithm depends on size and type of images in the dataset.

As discussed above various algorithms are available to denoise the noisy image, these algorithms work mainly on two-stage procedure (i) Detection of noisy pixel and (ii) Restoration of the noisy pixel. The success of such algorithms depends upon the individual performance of these respective stages[3], [79]–[82]. For the detection stage, the performance of the algorithm depends upon how accurately locations of corrupted pixels are detected. Detecting corrupted pixels locations in the presence of noise is a very challenging task that causes false detection once the noise level increases to a certain level. This leads to the problem of under and overdetection of noisy pixels. Similarly, for noisy pixels, restoration stage performance depends upon how close the algorithm restores the corrupted pixel value to the original value. As noise level increases in the image more of the neighboring pixels tend to get corrupted and it is very difficult to restore the desired values of the pixel[51], [83]–[85]. Therefore, in order to overcome the problem of over/under detection and to restore the value of corrupted pixels close to the

original values, the SAID-END method is proposed. The proposed method works exceedingly well in high noise scenarios for both grayscale and color images.

The main contributions of this chapter are as follows

1. We propose an enhanced adaptive noise detection algorithm to overcome the problem of over/under noise detection. The proposed method confirms the noisy pixel by using systematic thresholding and similarity index formulation.
2. We propose a non-corrupted pixel sensitive adaptive image restoration to increase the accuracy of the image restoration stage. This stage excludes the noisy values from contributing to original value estimation and it ensures the maximum number of noise-free pixels in the selected window. This process uses a spatially adaptive window with maximum non-corrupted pixel ratio criteria.

In the past decade, numerous contributions were made for denoising grayscale images and the challenges were addressed from diverse and many points of view. But significantly fewer contributions were made while addressing the issue of color image denoising[86], [87]. In this article, the focus is to provide a novel approach that is highly effective for both grayscale and color images. The proposed image denoising algorithm is applied to a variety of grayscale and color image data set. Experimental results demonstrate that it achieves high denoising performance in terms of Peak Signal-to-Noise Ratio (PSNR)[10], [61], [69], [88], [89], Image Enhancement Factor (IEF)[62], [64], [90] and Structural Similarity Index (SSIM)[61], [71], [91]–[94], that is superior than conventional denoising methods.

4.2 Preliminaries for Noisy Dataset Generation

Two data sets are used in the evaluation process of the proposed method first, datasets used in this analysis consists of 50 grayscale images from the Brodatz texture dataset (Fig. 3.1)[68] and 16 color images from the University of South California miscellaneous dataset volume 3 (Fig. 3.2)[67] as already discussed in chapter 3. The first data set is used to validate the performance on wide data set for both noise and artifact. Secondly, some commonly used traditional test images like Lena (grayscale and color) and Peppers (grayscale) are used for comparison of the proposed method with the recent state of the

art methods. All simulations were carried out in MATLAB (MathWorks, Natick, MA, USA). Sample of eight levels (10% to 80%) of salt and pepper noise affected grayscale and color images are shown in Fig 4.1 and Fig 4.2 respectively.

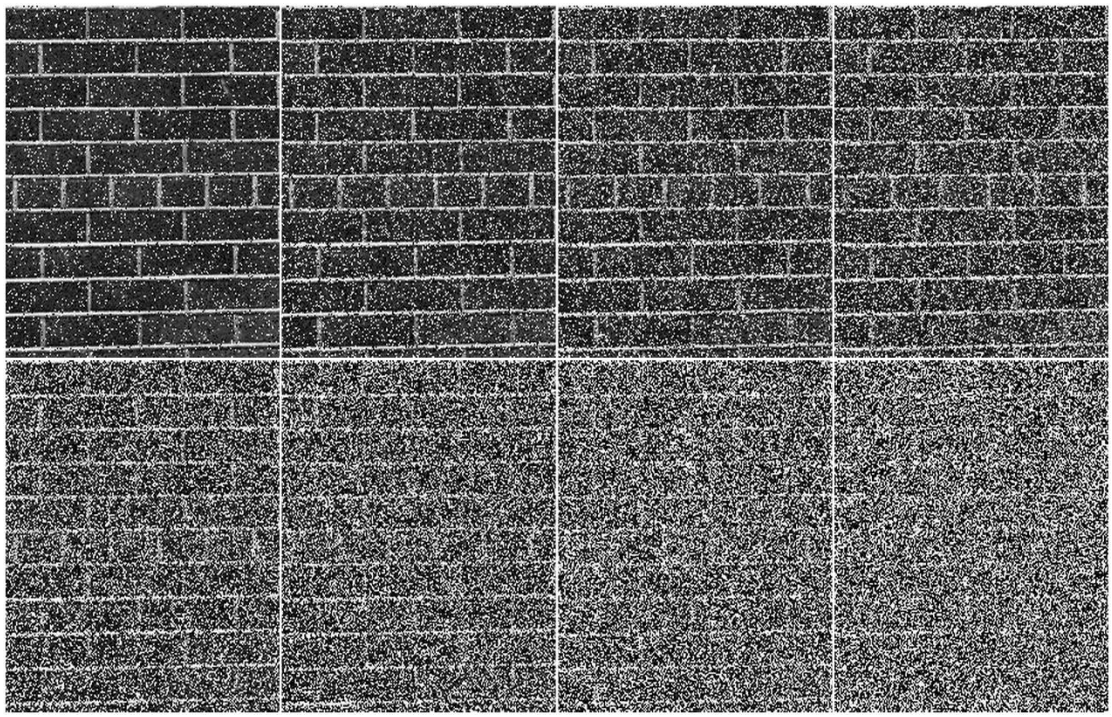


Figure 4.1. Eight levels of impulse noise affected Brick texture grayscale images.

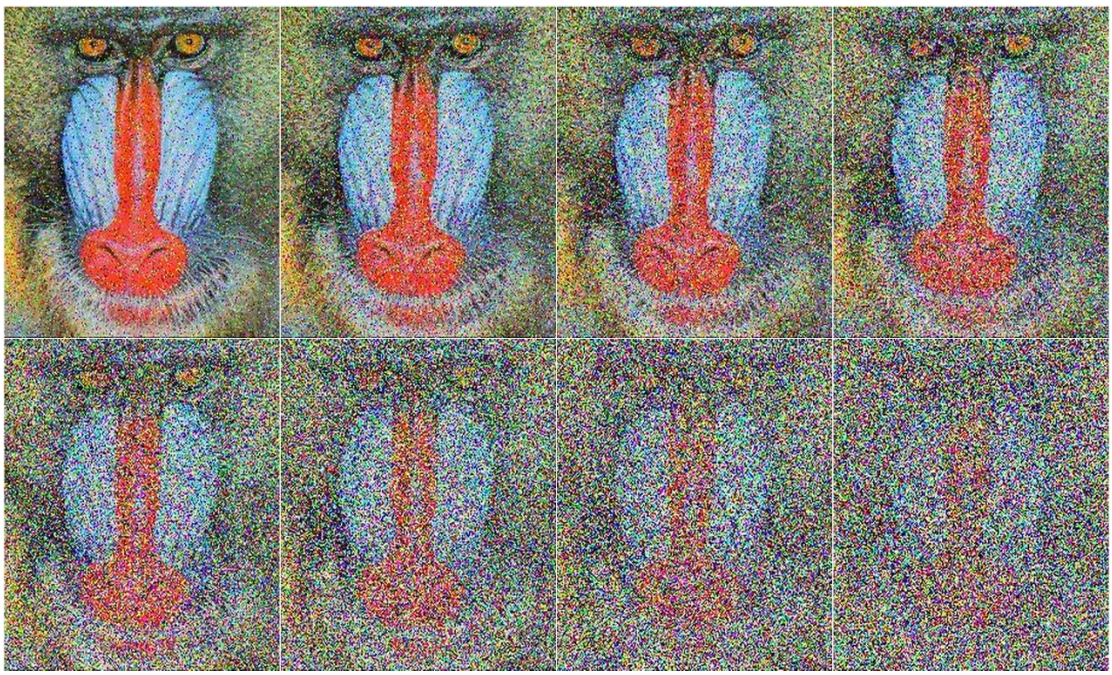


Figure 4.2. Eight levels of impulse noise affected Baboon color images.

Let's discuss first data set for test one, this case is to establish the robustness of the proposed method, two types of artifacts i.e., Strip lines and Blotches[19], [95] and one types of noise i.e., Salt & Pepper[96], [97] are added in the images at varying levels (Fig. 4.3).

Since the work is to propose the denoising algorithm which can denoise the images affected with any noise level (low to high noise level). So, in order to check the robustness of the proposed method, the noise and artifacts are added in the image in eight steps each as shown in (Fig. 4.3). First Salt and Pepper noise is added to an image in eight steps from 10% to 80% (10% is considered as low level 80% is considered as high noise level). Vertical and horizontal strip lines are added in eight steps starting from 2 pixels wide strip lines to 9-pixel wide strip line with an increment of one-pixel width. Similarly, Blotches artifacts are also introduced in an image with 8 levels starting from a square of 2*2 to 9*9 with an increment of one. The noise (Salt & Pepper) and artifacts (strip lines and blotches) are added one by one and then the proposed method is applied to noisy images to achieve a noise-free image. The protocol of adding noise and artifacts in the images is as follows.

- The noise (Salt & Pepper) having 10% of corruption level is added in the image (as shown in Fig. 4.3).
- Then, the noise level is increased by 10% to achieve a total noise level of 20% and this increased noise level of 20% is added in the image.
- Similarly, keep on increasing the noise level by 10% until the noise level reaches up to 80%.
- So, the database of corrupted images is created by adding noise level started from 10% to 80%.
- Similarly, create the data set of corrupted images using Strip lines and Blotches.
- The process of dataset creation & addition of different noise levels from 1 to 8 in both grayscale & color images is shown in Fig. 4.3.

Afterward, noise affected data in both grayscale and color images were denoised using existing well-known algorithms. ROAD-TGM[8], DBMF [8], [11], (CWMF)[17], [78],

(PSMF)[11], [56], (MF)[74] and proposed method. A comparison of the proposed method

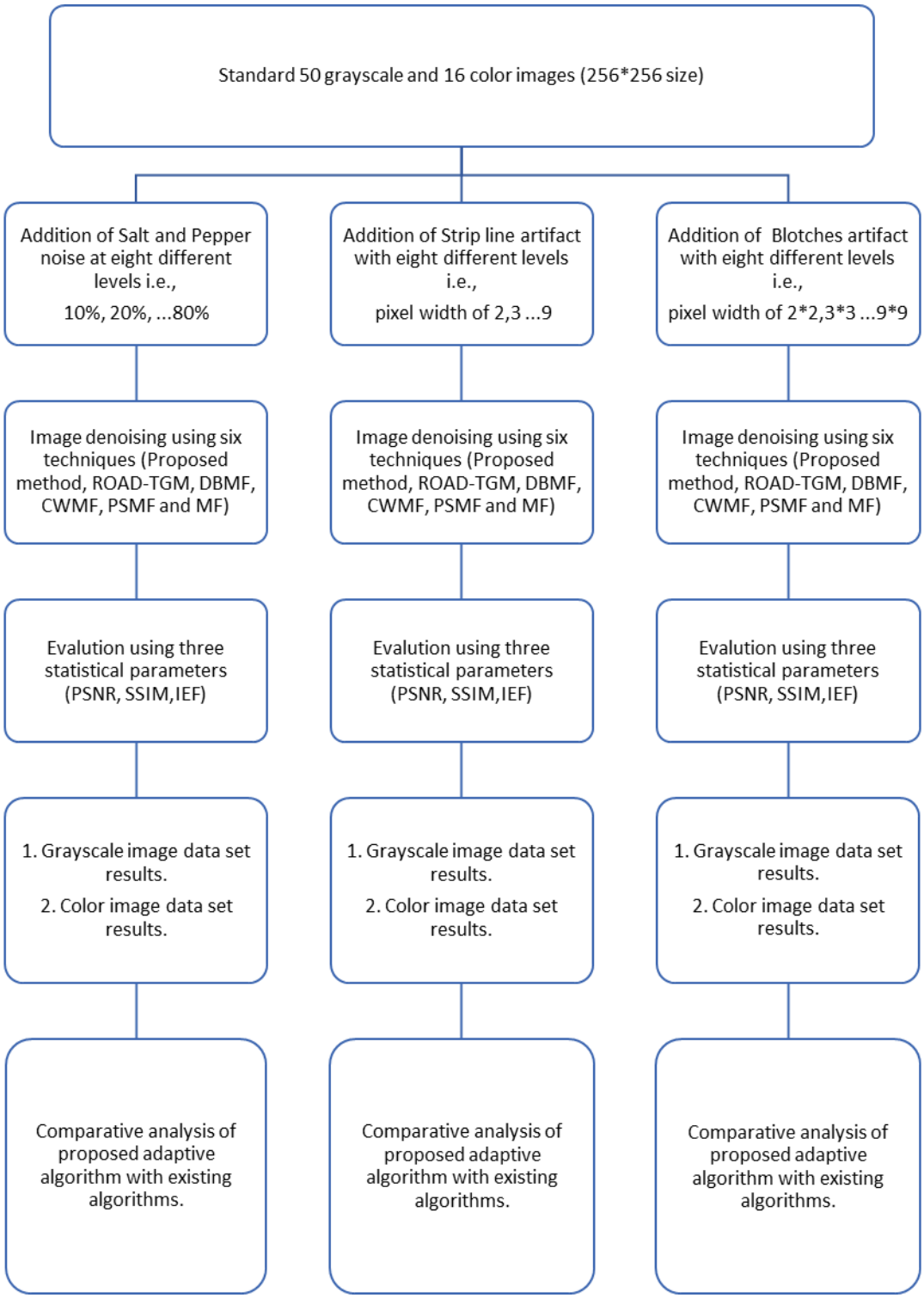


Figure 4.3. Explanation of test one Dataset, the procedure for addition of different noises, artifacts and description of protocol.

with the five other existing algorithms is done using box plots. Each box plot for the grayscale image dataset represents values obtained from 400 denoised images (50 images*8 levels of noise). Similarly, the color image data set each box plot represents parameter values obtained from 128 images (16 images*8 levels of noise). Our methodology is shown schematically in Fig. 4.3. For evaluation of the proposed method with the recent state of the art algorithms some commonly used traditional images i.e., Lena (grayscale and color) and Peppers (grayscale) are corrupted with 10% to 90% impulse noise. Then the proposed method along with the recent state of the art algorithms i.e., FDS, DAMF, IIN and IMF are applied to grayscale image dataset for performance comparison. For color image denoised, a performance comparison is drawn between AUTSF, MCF and proposed method. The grayscale image and color image comparison are drawn on the basis of PSNR and SSIM parameters as these are commonly preferred parameters.

4.3 Algorithm of Proposed Denoising Method.

In this chapter, the SAID-END method is proposed. The proposed method works on the concept of finding noisy pixels using systematic thresholding and spatially adaptive window hence overcoming the problem of under & overdetection. Secondly, the original value of the noisy pixel is restored adaptively by adjusting the statistical parameter median. The proposed method consists of two stages.

(A) Enhanced adaptive noise detection
(B) Non-corrupted pixel sensitive adaptive image restoration

Following is a detailed explanation of two stages of the proposed method.

A. Enhanced Adaptive Noise Detection

The detailed flowchart of the detection stage of the proposed method is shown in Fig 4.4. As mentioned earlier, the noise is added in the image ranges from 1 to 8 levels. Let assume the image having Salt and Pepper noise with a noise level of 10%.

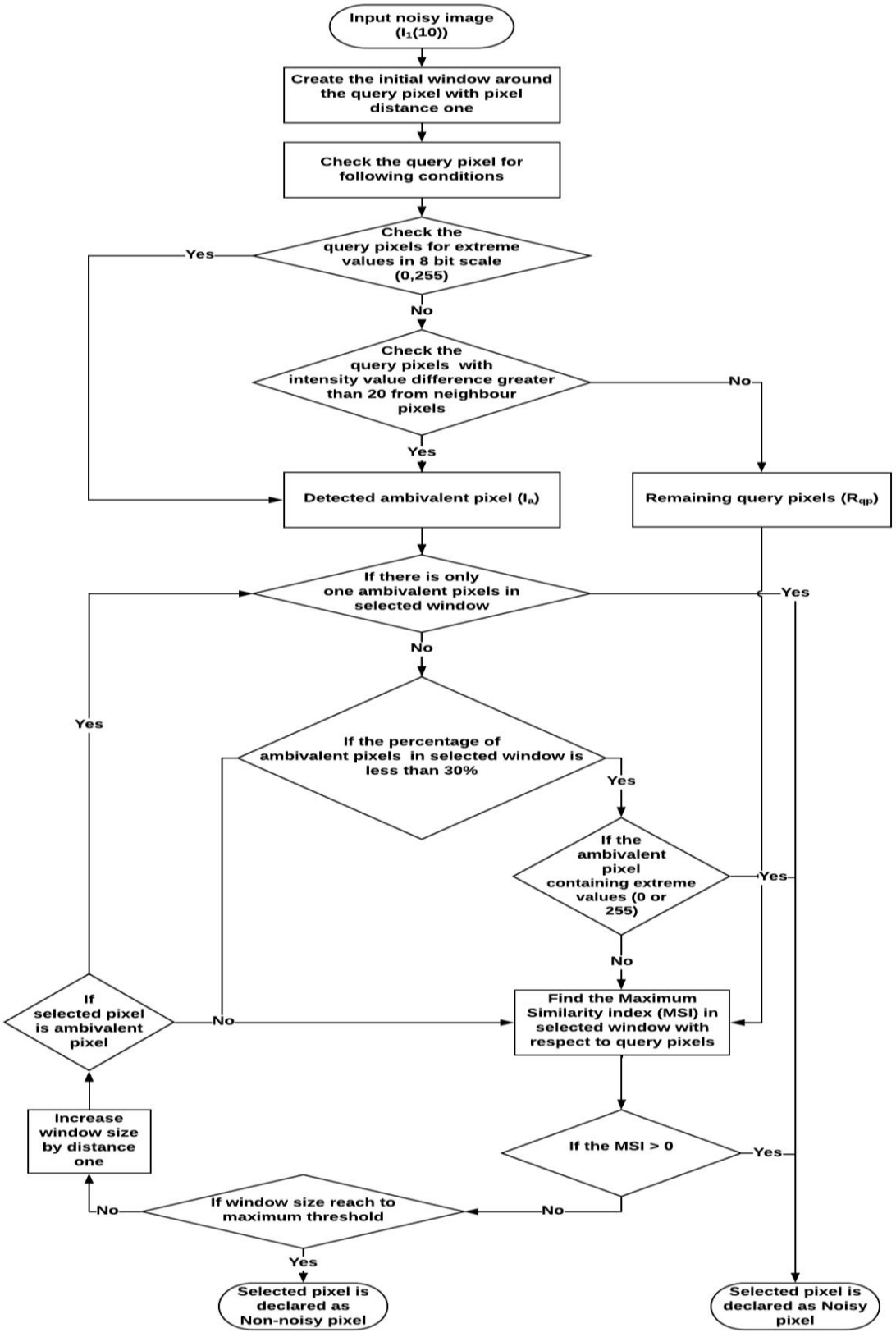


Figure 4.4. Flow chart of the detection stage of the proposed method.

The objective of this stage is to detect noisy pixels. Let's assume $I_i(j)$ as a noisy dataset of grayscale images.

$$I_i(j)\ where\ \begin{matrix} i{=}Number\ of\ images(1\ to\ 50\ for\ grayscale\ dataset) \\ j{=}Noiselevel(10\%,20\%\ldots80\%) \end{matrix} \quad (1)$$

The first image ($i = 1$) is having a noise level of 10% ($j = 10$). So, the first image having a noise level of 10% is denoted as $I_1(10)$ (flowchart is shown in Fig 4.4)

a) Consider the query pixel (first pixel) of $I_1(10)$ and create the window around query pixel with distance one (create a window on the immediate neighborhood).
b) Check whether the query pixel is having abrupt intensity values i.e., (i) 0 or 255 (ii) check if the intensity difference of query pixel with neighbor pixels in the window is greater than 20.
c) If any of the above condition is valid then, consider this query pixel as ambivalent pixel (doubtful to be the noisy pixel (I_a)). More confirmation is required to declare I_a as a noisy pixel.
d) Further, check if the selected window contains single or multiple ambivalent pixels.
e) If I_a is the only pixel in the window having extreme value or only pixel with intensity difference greater than 20 from surrounding pixels then it is a noisy pixel. If not then check how many pixels in the window are having extreme values and intensity difference greater than 20.
f) Then, count how many $I_a's$ are present in the selected window. Let's assume there are 's' ambivalent pixels I_{as} ($I_{as_1}, I_{as_2}, I_{as_3}$ ...). The formula for calculating Ambivalent Pixels Percentage (APP) is given below.

$$\%\ APP = \frac{I_{as}}{Total\ number\ of\ pixels\ in\ selected\ window} * 100 \quad (2)$$

g) The pixel I_a is considered noisy, if APP satisfy the following condition.

$$APP \leq D_{ref} \text{ Where } D_{ref} = 30\% \quad (3)$$

If APP does not satisfy the above criteria then check the similarity of I_a with the neighbor pixels.

h) To declare I_a as a noisy or non-noisy pixel, Calculate Maximum Similarity Index (MSI) [98]which is

$$MSI = [sum(MSIR)] - WP_{ref} \quad (4)$$

where MSIR is the Maximum Similarity Index Range which is calculated as.

$$MSIR = \frac{SIR_i}{S_{ref}} \quad (5)$$

where,

i $= 1,2, \ldots$ WP_{ref} (Window pixel reference)

$Similarity\ reference\ (S_{ref}) = 20$

$WP_{ref} = 30\%\ of\ total\ number\ of\ pixels\ in\ the\ window$

Similarity Index Range (SIR) i.e difference of I_a with all non-extreme pixels in the window NN_{ij} ($NN_{i1}, NN_{i2}, NN_{i3}$-----$NN_{in}$). Where 'i' represent window number and 'j' represent the pixel number of respective window (NN_{11} indicates the first pixel of the first window). So, to create a vector SIR equation is given below.

$$SIR = [d_1, d_2 \text{---} d_n] \quad (6)$$

$$d_1 = |NN_{11} - I_a| \quad (7)$$

Similarly, $d_2\ to\ d_n$ can be created by varying pixel represented by j in the above equation. Arrange SIR in ascending order.

i) Ambivalent pixel I_a can be declared as a noisy pixel. If MSI satisfies the following conditions.

$$Result = \begin{cases} noisy & if \quad MSI > 0 \\ non\ noisy & otherwise \end{cases} \tag{8}$$

j) If MSI of pixel is lesser than zero or ratio of corrupted pixels is greater than D_{ref} then expand the window and repeat step d to step j.

k) In case window expanded to maximum window size (pixel distance thirteen) while MSI remains less than zero, then the status of I_a will be fixed as non-noisy.

l) For $R_{qp}\ pixels(remaining\ query\ pixels)$ repeat steps h to j.

Let's understand the detection stage with the help of the following examples. Let's say the matrix of the original image having intensity values of pixels as shown in Fig. 4.5. The original matrix (prior to addition of noise is shown in Fig. 4.5) is corrupted by the different types of noises and artifacts to express the cases of detection stage. The following cases are discussed.

Case 1 (a), (b): When single-pixel is corrupted in the selected window having extreme values (0 or 255).

Case 1 (c), (d): When single-pixel is corrupted in the selected window having value between 0 and 255.

Case 2: When less than 30% of pixels in a selected window are ambivalent pixels.

Case 3: When more than 30% of pixels in a selected window are ambivalent pixels.

Case 4: Edge preservation.

Firstly, let's take an example of noisy image affected by Salt and Pepper noise shown in Fig. 4.6.

172	165	167	165	170
170	170	166	172	171
171	175	170	173	169
177	167	164	169	168
173	174	171	168	166

Figure 4.5. Original Matrix (without any noise).

Case 1 (a), (b): When single-pixel is corrupted in the selected window having extreme values (0 or 255).

As shown in Fig. 4.6, there are two windows in each window there is only one corrupted (ambivalent) pixel i.e., pixel having value '0' in window NN_1 and another pixel having extreme value '255' in window NN_2. So as per the proposed method, this pixel will be considered as noisy (as there is only one ambivalent pixel in each of the windows) shown in Fig. 4.6.

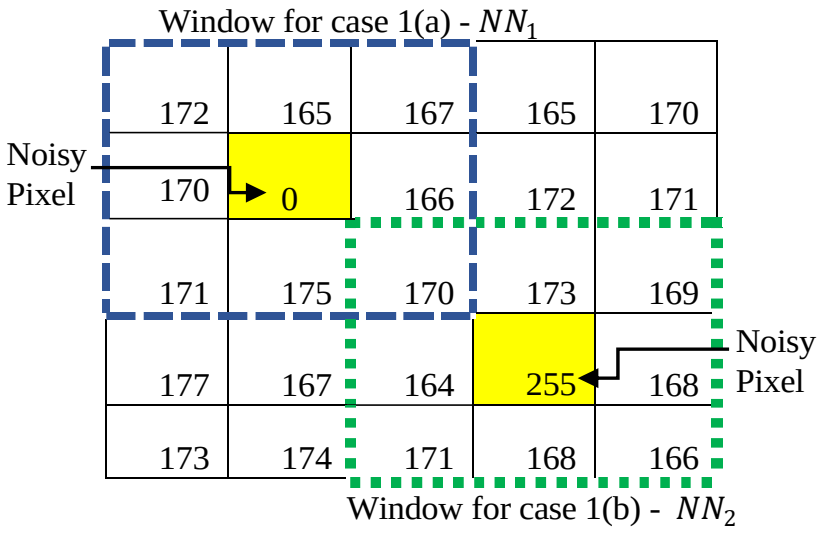


Figure 4.6. Matrix for noise detection case 1 (a) and (b).

Case 1 (c), (d): When single-pixel is corrupted in the selected window having value between 0 and 255.

Pixels of an image can take non-extreme values now let us consider the case 1 (c) (shown in Fig. 4.7) which consist of non-extreme value in the selected window. Pixel in window NN_3 is having value '100' which is having a difference greater than 20 (reference intensity difference threshold) from every neighbor pixel in the window. So, it will be considered as the odd man out and labeled as a noisy pixel. Case 1 (d) (shown in Fig. 4.7) takes care of those noises which can attain any value except from value neither extreme in nature nor having a difference greater than 20 from every neighbor pixel in the selected window. In such cases, it is difficult to decide the

Remaining Query Pixels (R_{qp} as shown in Fig. 4.4) is a noisy pixel or original pixel. To overcome this challenge distance base similarity is calculated for the R_{qp} and similarity conditions need to be satisfied for thirty percent pixels of the total number of pixels in the considered window. Otherwise, the pixel will be considered as a noisy pixel. Maximum Similarity Index (MSI) calculations are already discussed above (refer to equation no.4 - 8) in this chapter. In window NN_4 of case 1 (d) (Fig. 4.7) the remaining query pixel (R_{qp}) value is 145 which is not having an intensity difference of 20 from every other pixel in the window. So, it is important to check the similarity of query pixel with its non-extreme neighbors. Let us understand this example in detail by applying the equation number 4 and onwards respectively. In this case R_{qp} is having value 145 and $d_1\ to\ d_n$ are the values of its neighbor pixels. Let's start by creating the Similarity Index Range (SIR).

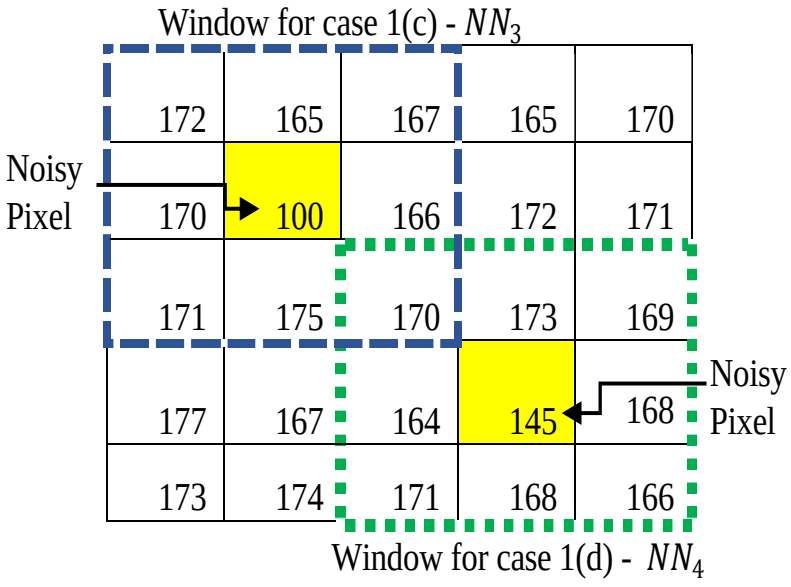


Figure 4.7. Matrix for noise detection case 1 (c) and case 1 (d**).**

$$SIR = [|170-145|, |173-145|, |169-145|, |164-145|, |168-145|, \ldots\ldots$$

$$|171-145|, |168-145|, |166-145|] \quad (9)$$

$$SIR = [25,28,24,19,23,26,23,21] \quad (10)$$

Arrange the SIR in ascending order as follows

$$SIR = [19,21,23,23,24,25,26,28] \quad (11)$$

Calculate 30% of the total number of pixels in the selected window, window NN_4 is a having dimensions 3x3 (total 9 pixels). So, 30 percent of 9 is 2.7 which can be rounded off to 3. Now consider the first 3 distance from SIR vector (as 3 is the result of 30 % of 9 pixels) and then divide SIR values with Similarity reference (S_{ref})which is defined as 20 as shown in the following operation.

$$MSIR = [19/20\,, 21/20\,, 23/20] \quad (12)$$

$$MSIR = [0.95,1.05,1.15] \quad (13)$$

$$MSI = [0.95 + 1.05 + 1.15] - 3 = 0.15 \quad (14)$$

As MSI is greater than zero in this example the respective R_{qp} pixel will be considered as

Case 2: When less than 30% of pixels in a selected window are ambivalent pixels.

To understand this case in detail let's consider an example in Fig. 4.8. In this case, a window is considered with the initial distance one from ambivalent pixels and is initially having 3x3 dimensions. Consider two ambivalent pixels in both windows (NN_1& NN_2) and the rest of the pixels are non-corrupted pixels. So specifically, for this case 7 pixels in each window are non-corrupted and two are query pixels out of total 9 pixels. The corrupted (ambivalent) pixels are highlighted with yellow color in both windows. The algorithm will calculate percentage of ambivalent pixels (APP refer to equation no.3) in the selected window. The percentage of ambivalent pixels should be less than the considered threshold value i.e., D_{ref}= 30% and $APP =$ 22% ((2/9)*100=22%). So, when the calculated percentage of ambivalent pixels is less than the considered threshold value then the ambivalent pixel will be considered as a noisy pixel.

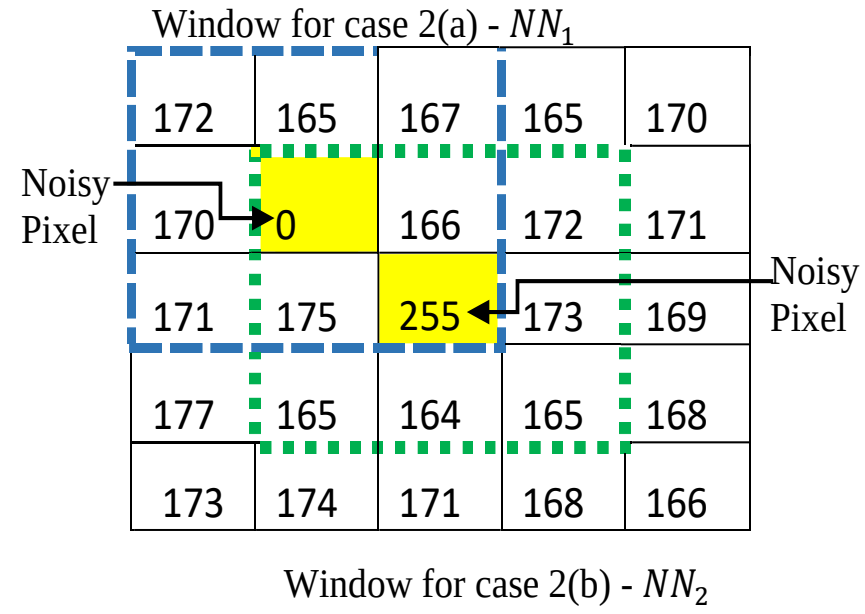


Figure 4.8. Matrix for multiple ambivalent pixels in the selected window.

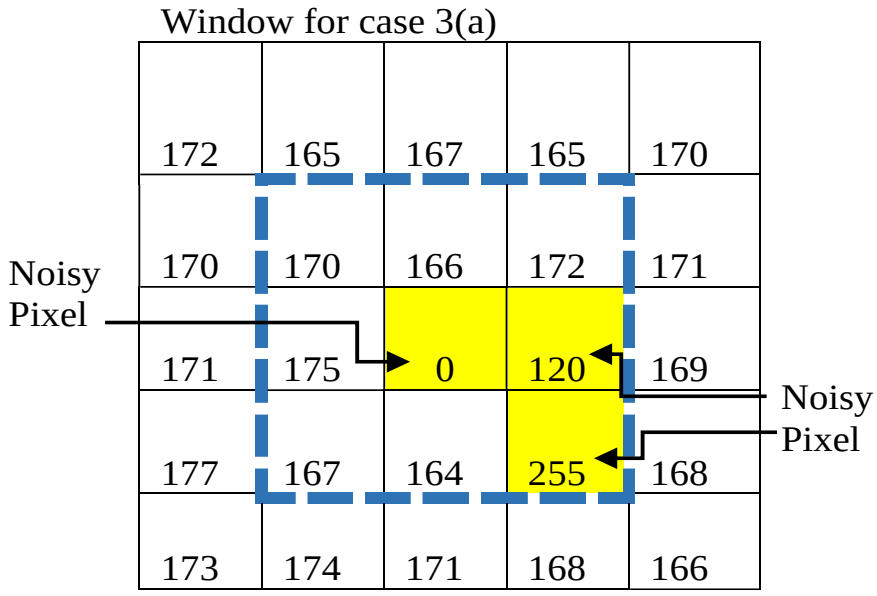


Figure 4.9. Matrix for multiple ambivalent pixels in selected window**.**

Case 3: When more than 30% of pixels in a selected window are ambivalent pixels.

In this example, three pixels are considered as ambivalent pixels out of total 9 pixels (initial window is of 3*3 dimensions) as shown in Fig. 4.9. The ambivalent pixel

percentage is 33% ((3/9)*100=33%) which is greater than the considered threshold value ($D_{ref} = 30\%$). In such a condition when the noise level is higher than the

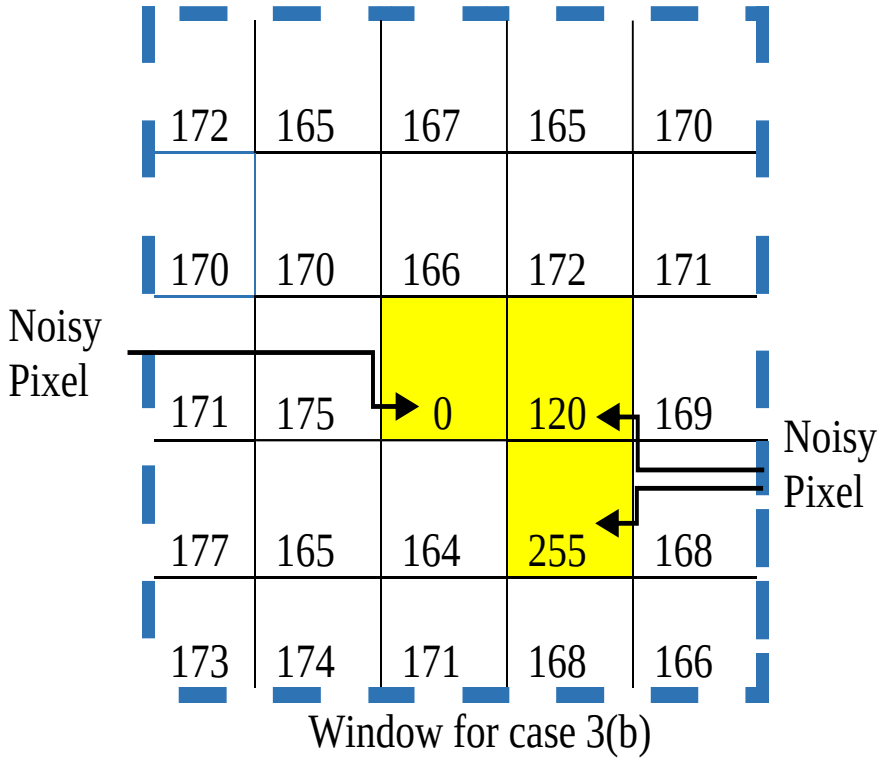


Figure 4.10. Matrix for multiple ambivalent pixels in selected window of 5*5 size.

threshold value, the algorithm will increase the window size by pixel distance one. The new window is having 5*5 dimensions shown in Fig. 4.10 and the total number of pixels are now 25 in the window. The algorithm will again calculate the Ambivalent Pixel Percentage (*APP* refer to equation no.3) for the considered case it will be 12% ((3/25)*100 =12%) which is less than the considered threshold value(D_{ref}= 30%), So query pixel will be declared as a noisy pixel.

Due to the high amount of noise present in cases where multiple noise pixels are existing in a window. Ambivalent pixel percentage can have value higher than 30%, in such cases algorithm will keep on increasing the window size by pixel distance one and repeat the ambivalent pixel percentage calculation again until ambivalent pixel percentage becomes less than 30%. In proposed method increasing the window size is restricted to maximum pixel distance thirteen, if till this limit of pixel distance thirteen

ambivalent pixel percentage remains higher than 30% then the pixel is considered as original pixel.

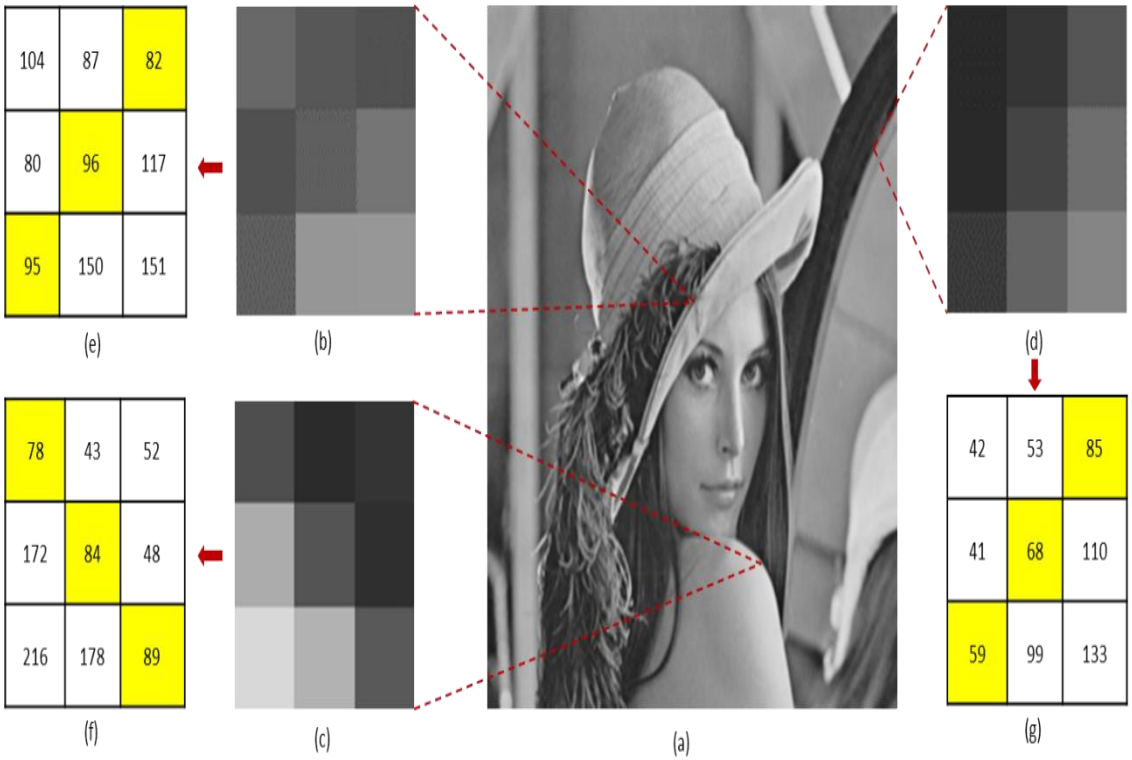


Figure 4.11. Edge preservation process (a) Original Lena image; (b) Zoomed hat edge; (c) Zoomed shoulder edge; (d) Zoomed mirror edge; (e) Intensity values of picture (b); (f) Intensity values of picture (c); (g) Intensity values of picture (d).

Case 4: Edge Preservation

To understand the edge and detail preservation process of the proposed method in natural conditions, let us consider an example of a standard Lena image. For this purpose, three edges from different locations are marked to represent different contrast situations and have been discussed case by case. These cases are as follows:

Case 4-A: Edge pixel on hat (shown in Fig.4.11 (a))

Case 4-B: Edge pixel on shoulder (shown in Fig.4.11 (b))

Case 4-C: Edge pixel on mirror (shown in Fig.4.11 (c)).

The direction of the edge is marked by the yellow color in Fig.4.11(e-g). The proposed method preserve edge by doing classification of edges as non-corrupted pixels.

Case 4-A: Edge pixel on hat

The first example is considered from Lena hat as shown in Fig. 4.11(b) and its respective intensity values are presented in Fig. 4.11(e). Let us form a 3x3 initial window by considering the center pixel ($\boldsymbol{R_{qp}}$ $is\ remaining\ query\ pixel$) as a base is shown in Fig.4.11(e) which is under evaluation (originally an edge pixel). To verify the pixel as noisy or non-noisy for this case we need to satisfy $\boldsymbol{MSI}$ equation. Let us compute MSI in steps (as per step h and i of the proposed algorithm) as mentioned in the proposed method.

Computation of Similarity Index Range ($\boldsymbol{SIR}$) from values of Fig.4.11(e) by using equation 6-7.

SIR =[$|104-96|$,$|87-96|$,$|82-96|$,$|80-96|$, $|117-96|$, $|95-96|$,……., $|150-96|$,$|151-96|$] (15)

$\text{SIR} = [8,9,14,16,21,1,54,55]$ (16)

Arrange the SIR in ascending order as follows

$\text{SIR} = [1,8,9,14,16,21,54,55]$ (17)

To ensure the detailed preservation we are using 30% criteria of window size. The rationale of using a 30% value is that the proposed method on this data set gives the best results when 30% criteria are used. This criterion is set to 30% to make the algorithm work well in considerably high noise conditions (as noise increases, the count of non-corrupted pixel reduces). Calculate 30% of the total number of pixels in the selected window, the window has initial dimensions of 3x3 (total 9 pixels). So, 30 percent of 9 is 2.7 which can be rounded off to 3.

Thirty percent criteria also ensure quality with an increase in window size (adaptive window). Now consider the first 3 distance from $\boldsymbol{SIR}$ vector (as 3 is the result of 30 %

of 9 pixels) and then divide $\boldsymbol{SIR}$ values with similarity reference ($\boldsymbol{S_{ref}}$) which is defined as 20. $\boldsymbol{S_{ref}}$ is a weight that provides tolerance to the algorithm with varying noise ratio, for this chapter it is set to 20 as this tolerance weight is working well on a large dataset considered in this analysis. MSI is calculated as shown in the following equations by using equations 4-5.

$$\text{MSIR} = [1/20\,, 8/20\,, 9/20] \tag{18}$$

$$\text{MSIR} = [0.05, 0.4, 0.45] \tag{19}$$

$$\text{MSI} = [0.05 + 0.4 + 0.45] - 3 = -2.1 \tag{20}$$

As $\boldsymbol{MSI}$ is less than zero in this example the respective $\boldsymbol{R_{qp}}$ pixel will be considered as a non-noisy pixel. So, the edge will be preserved (original intensity value will be kept as such). In this way proposed algorithm finds the pixels on edges as non-noisy.

Case 4-B: edge pixel on shoulder.

MSI is calculated from intensity values of Fig.4.11(f) on similar lines as in above-mentioned case by considering the initial window size as 3 x 3.

$$\text{SIR} = [|78 - 84|, |43 - 84|, |52 - 84|, |172 - 84|, |48 - 84|, |216 - 84|,$$

$$|178 - 84|, |89 - 84|] \tag{21}$$

$$\text{SIR} = [6, 41, 32, 88, 36, 132, 94, 5] \tag{22}$$

Arrange the SIR in ascending order as follows

$$\text{SIR} = [5, 6, 32, 36, 41, 88, 94, 132] \tag{23}$$

$$\text{SIR} = [5/20\,, 6/20\,, 32/20] \tag{24}$$

$$\text{MSIR} = [0.25, 0.3, 1.6] \tag{25}$$

$$\text{MSI} = [0.25 + 0.3 + 1.6] - 3 = -0.85 \tag{26}$$

As $\boldsymbol{MSI}$ is less than zero in this example the respective $\boldsymbol{R_{qp}}$ the pixel will be considered as a non-noisy pixel. So, the edge will be preserved.

Case 4-C: Edge pixel on mirror.

In this case, find out the MSI of intensity values given in Fig. 4.11(g) on a similar pattern as mentioned in the proposed method (step h and i).

SIR =[$|42-68|$,$|53-68|$,$|85-68|$,$|41-68|$,$|110-68|$, $|59-68|$,…,

$|99-68|$,$|133-68|$] (27)

$SIR = [26,15,17,27,42,9,31,65]$ (28)

Arrange the SIR in ascending order as follows

$SIR = [9,15,17,26,27,42,9,31,65]$ (29)

$MSIR = [9/20\,, 15/20\,, 17/20]$ (30)

$MSIR = [0.45,0.75,0.85]$ (31)

$MSI = [0.45 + 0.75 + 0.85] - 3 = -0.95$ (32)

As MSI is less than zero in this example the respective R_{qp} pixel will be considered as a non-noisy pixel. So the edge will be preserved (edge pixel will be kept in original form).

B. Non-Corrupted Pixel Sensitive Adaptive Image Restoration

Once the pixel is detected as noisy pixel (as shown in Fig. 4.6 – 4.10), next stage is to restore the original values of noisy pixels. For this purpose, the restoration stage is proposed. The flow chart of the restoration process is shown in Fig. 4.15. In case of high noise density fixed window size is a prime reason for the loss of edge information. To overcome this issue, noise level based adaptive window is preferred to ensure high amount of non-corrupted pixels in the window[99], [100]. The noise restoration stage uses the location of noisy pixels identified by noise detection stage. This stage ensures

a minimum of 70% non-corrupted pixels (maximum non-corrupted pixel ratio criteria) are used to estimate the original value to increase accuracy. The maximum non-corrupted pixel ratio criteria is implemented by an adaptive window with the condition on noise level less than 30%. The use of local information of non-corrupted in the window can preserve edge details to a certain extent. To understand the process of noise restoration stage in detail let us consider the following cases.

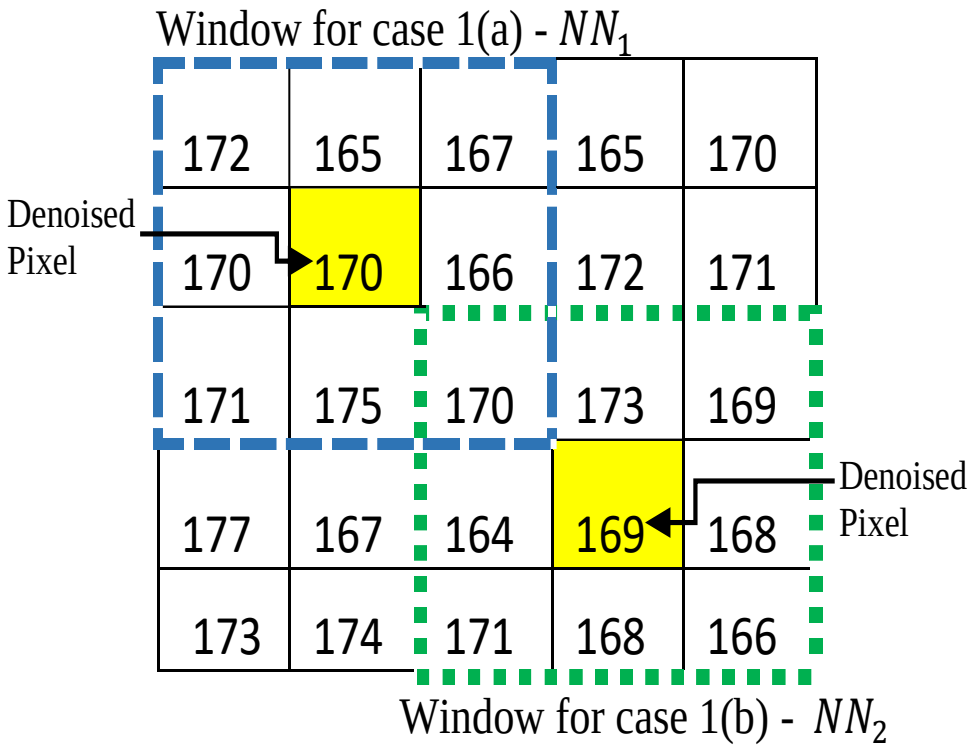


Figure 4.12. Matrix for single restored pixels in selected window.

Case 1: When a single pixel is noisy pixel in selected window

This case restores the original value of noise detected in case 1 (a)-(d) of the detection stage. In the considered matrix for noise restoration, noise location is already known as a result of detection stage. The algorithm will utilize the location of noisy pixel and create an initial window with distance one which results in 3*3 matrix. Algorithms utilizing only non-corrupted neighbor pixels to restore the value of noisy pixel (non-corrupted and corrupted pixels are already identified in the detection stage). The median of non-corrupted pixels in the selected window is taken and replaced with the value of

noisy pixel (noisy pixels in the selected window are not included in the median calculation). This process of restoration stage is shown in Fig. 4.15. Consider case 1 (a) window NN_1 (as shown in Fig. 4.6) and case 1 (c) window NN_3 (as shown in Fig. 4.7) in these windows noisy pixel value is replaced by median value 170 (calculated by taking a median of non-corrupted pixels of the respective window). For case 1 (b) window NN_2 (as sown in Fig. 4.6) and case 1 (d) window NN_4 (shown in Fig. 4.7) noisy pixel is restored with median value 169 (as integer value is required in the image, so round off operation is applied on decimal values). Final restored matrix from noisy matrix shown in Fig. 4.6 and Fig. 4.7 is presented in Fig. 4.12.

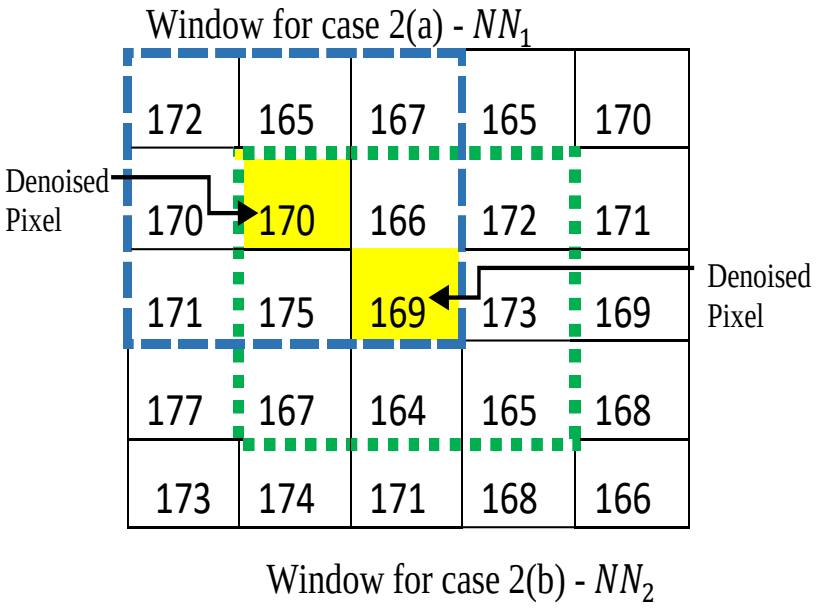


Figure 4.13. Matrix for multiple restored pixels in selected window.

Case 2: When less than 30% of pixels in a selected window are noisy pixels

Noisy pixels in a considered window NN_1 *and* window NN_2 of matrix shown in Fig. 4.8 are multiple. Let's consider case 2(a) window NN_1 (shown in Fig. 4.8) first, Algorithm will calculate percentage of noisy pixels in the selected window. Percentage of noisy pixels should be less than the considered threshold value of 30%, in the considered window it is 22% ((2/9)*100=22%). So, the median value of seven non-corrupted pixels is calculated as 170 and replaced with the noisy pixel in case 2(a) window NN_1. As now noisy pixel of case 2(a) window NN_1 is restored, now case 2(b)

window NN_2 will have only one noisy value which is filtered as case 1 of the restoration stage already discussed above. So median value of eight non-corrupted pixels in the window NN_2 is replaced with noisy pixel (median value 169). Restored matrix from noisy matrix shown in Fig. 4.8 is presented in Fig. 4.13.

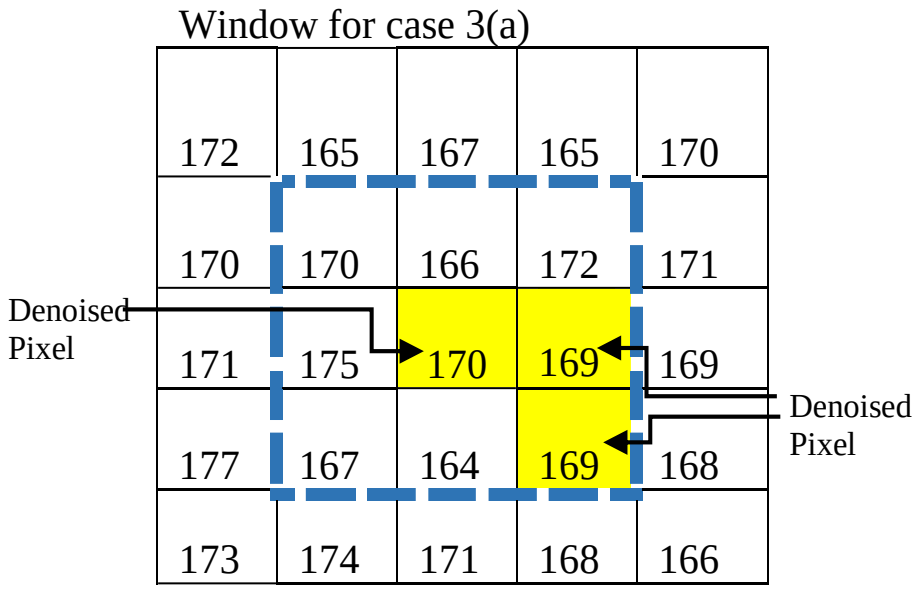


Figure 4.14. Matrix for multiple restored pixels in selected window.

Case 3: When more than 30% of pixels in a selected window are noisy pixels.

When noisy pixels in a considered window are high (greater than 30%) shown in Fig. 4.9. Then window size is increased by one and noisy pixel percentage in a selected window is calculated again (Noisy Pixel Percentage (NPP) calculation is similar to calculation of APP simply replace ambivalent pixels with detected noisy pixels in the equation no.2). This process is repeated until the noisy pixel percentage in the selected window is less than D_{ref} which is 30% and then the median is calculated from non-corrupted values. For denoising of the matrix shown in Fig. 4.9 it is required to increase the window size by one (as noise pixel percentage is greater than 30% in case 3(a)), the new window size will be of 5x5 and noisy pixel percentage is 12% ((3/25)*100=12). As 12% is less than considered a threshold value of 30%, so the median value of non-corrupted pixels from the current window is calculated as value '170' (out of 25 total values 23 are considered for median calculation) which is replaced with noisy pixel

value ‘0’. Now for noisy pixel having value ‘120’, again window is created with distance one (3x3) and now the percentage of noisy pixel is calculated as 12% ((2/9)*100=12%) which is less than considered threshold value 30%. Now, this pixel will be denoised as case 2 where the median is calculated by ignoring the other noisy pixels in the window. So, for this pixel median is calculated as 169 (actual value was 173 it is not possible to achieve the same value every time). For noisy pixel value 255 case 1 will be applicable as the other two pixels in the window are already restored. So the median is calculated as 169 with a similar procedure to case 1 of image restoration stage and replaced with noisy pixel (shown in Fig.4.14).

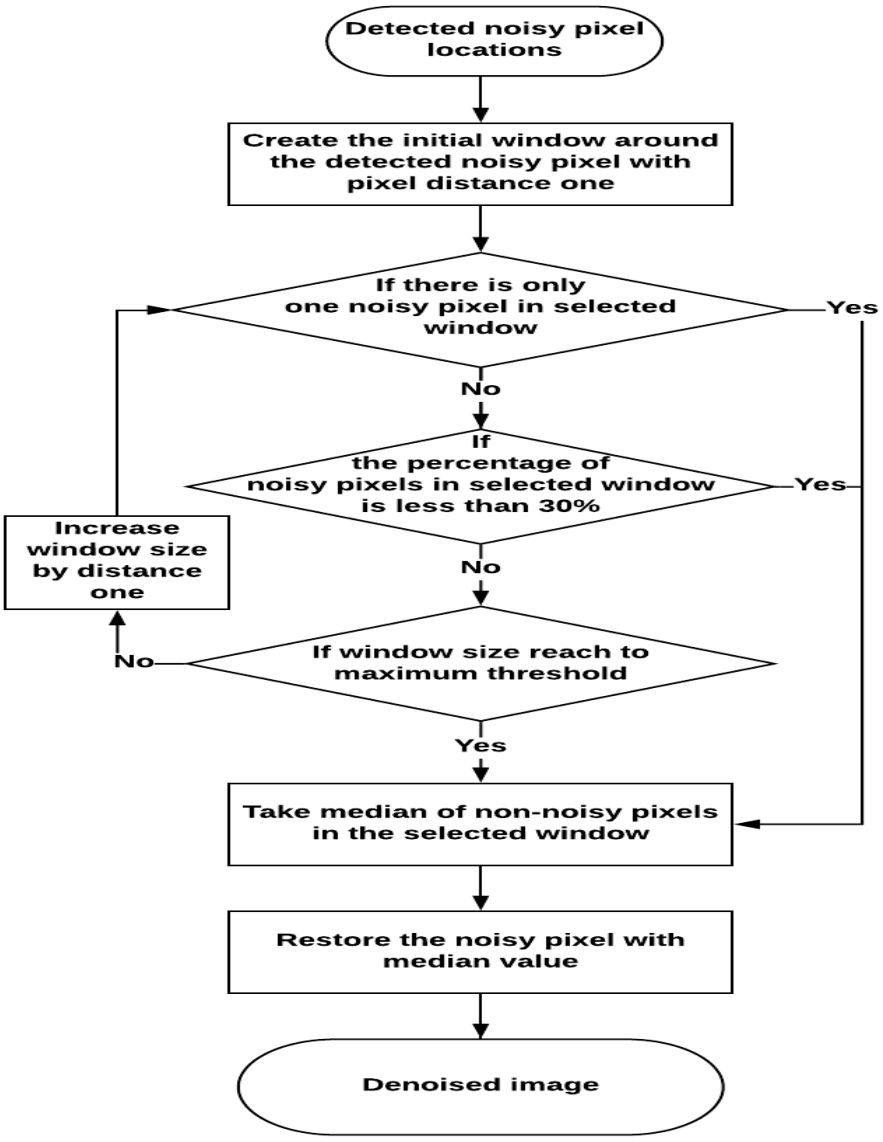
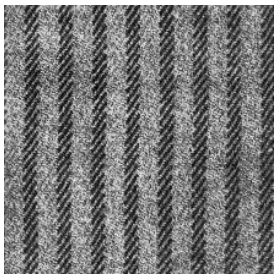


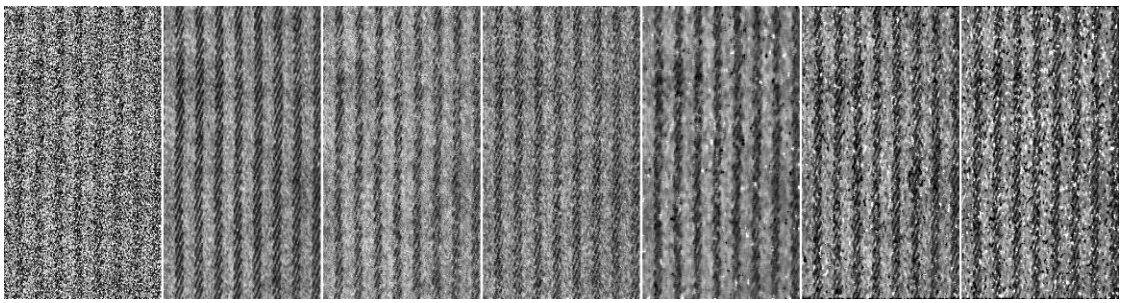
Figure 4.15. Noise restoration stage algorithm.

4.4 Results and Discussion

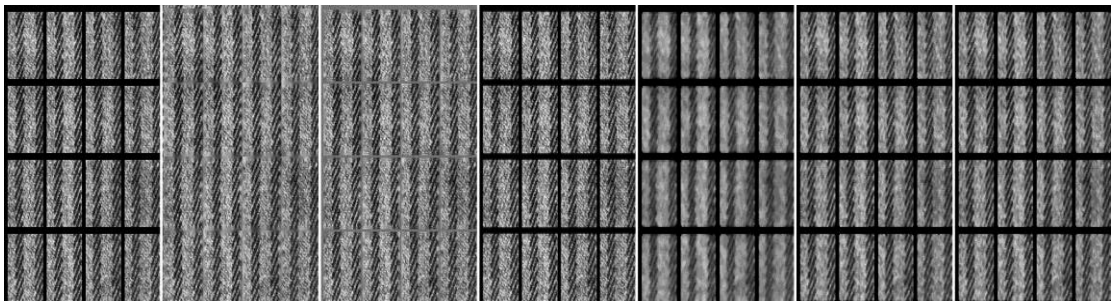
This section evaluates the performance of the proposed method and comparison is done with existing denoising algorithms. This comparison is carried out in two test stages. Let's discuss the first stage for the comparison of the proposed method with well-known methods based on the wide dataset. As mentioned earlier the noise (salt pepper) and artifacts (Strip lines & blotches) are added in images (Grayscale and color images). For each image, different noise levels were incrementally added from noise level one to noise level eight and 2 pixels to 9 pixels range is used for strip lines and blotches artifacts, forming sequences of increasingly corrupted images.



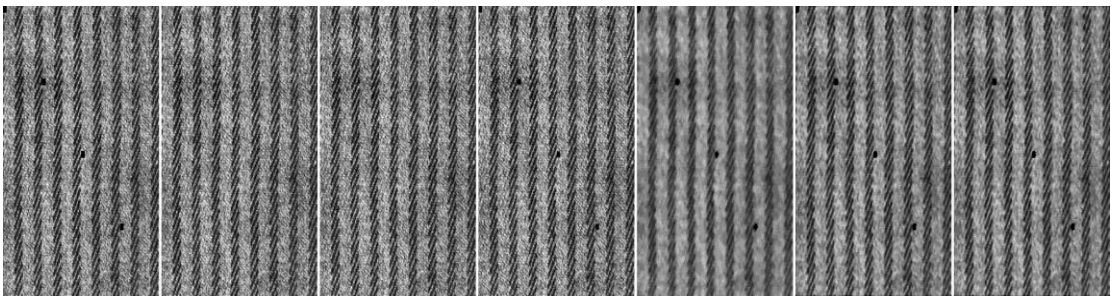
(a)



(aa) (ab) (ac) (ad) (ae) (af) (ag)



(ba) (bb) (bc) (bd) (be) (bf) (bg)



(ca) (cb) (cc) (cd) (ce) (cf) (cg)

Figure 4.16. (a) Original texture image, (aa) Texture image corrupted by 50% Salt and Pepper noise, (ba) Texture image corrupted by 6 pixel wide strip line artifact, (ca) Texture image corrupted by 6*6 pixel blotches artifacts, (ab–cb) Image restored using proposed method from (aa–ca), (ac–cc) Image restored using the ROAD-TGM from (aa–ca), (ad–cd) Image restored using the DBMF from (aa–ca), (ae–ce) Image restored using the CWMF from (aa–ca), (af–cf) Image restored using the PSMF from (aa–ca), (ag–cg) Image restored using the MF from (aa–ca). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

For the visual understanding of images before and after denoising, A set of grayscale images are present in Fig 4.16 and color images are present in Fig 4.17. A comparative analysis of the proposed method with existing denoising algorithms on grayscale images corrupted by level 1 to level 8 of noise and artifacts is shown in Fig. 4.18–4.20. The PSNR, SSIM and IEF values are used to evaluate the performance of the proposed and existing algorithms on all grayscale images and color images.

(a)

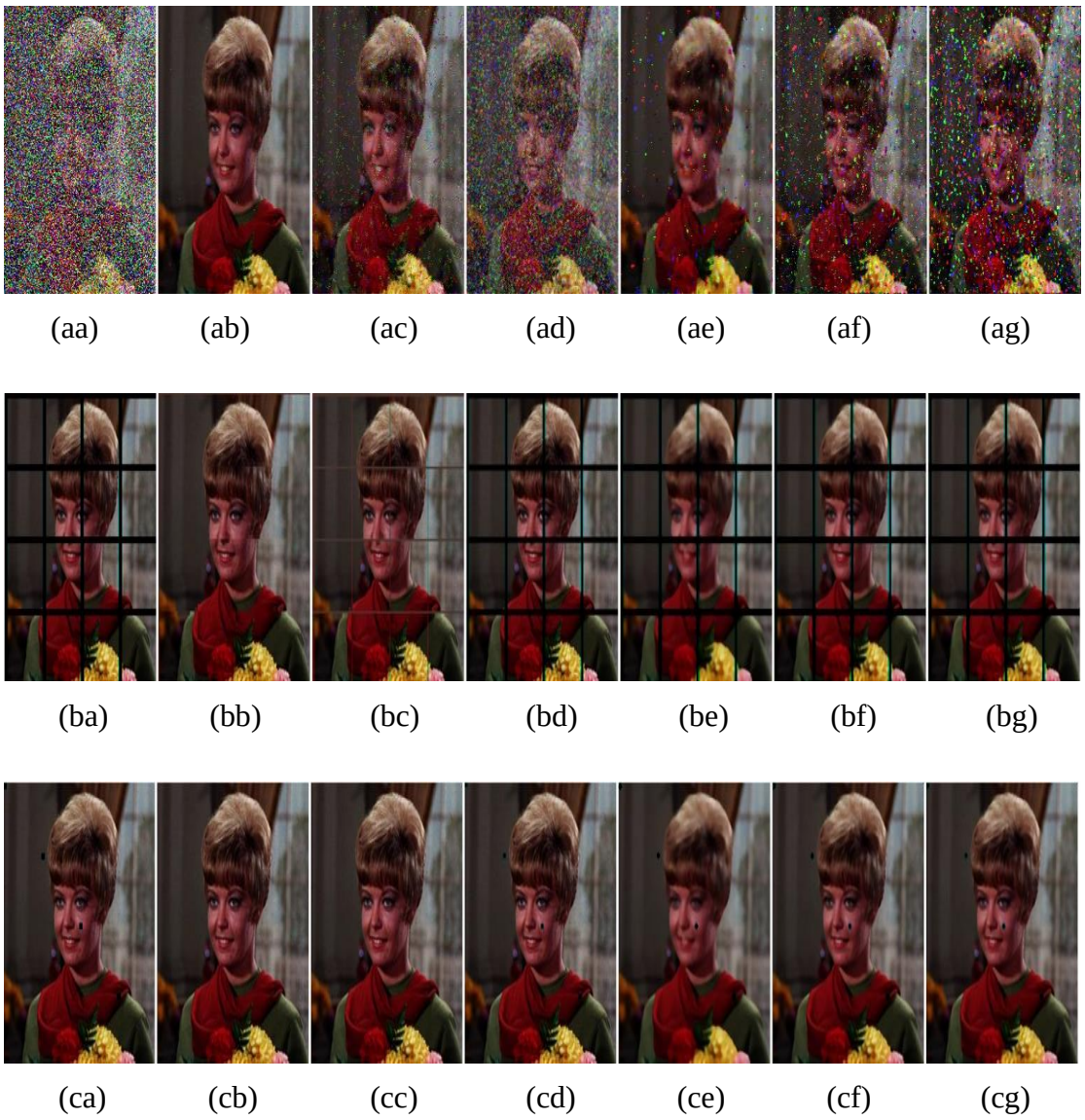


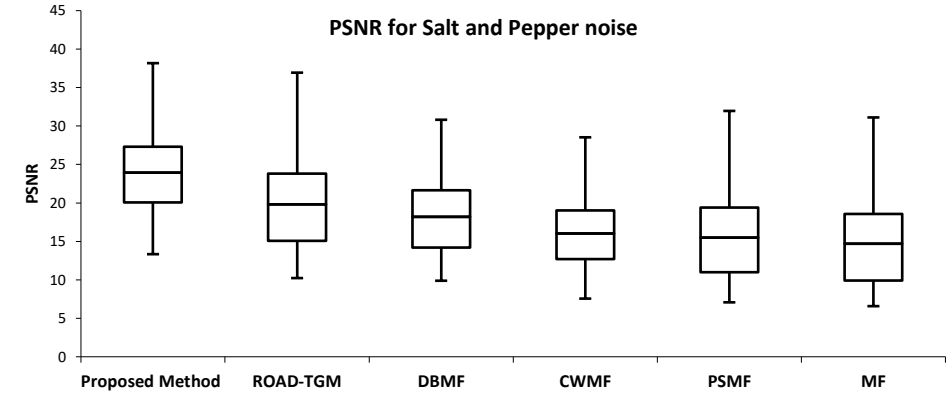
Figure 4.17. (a) Original color image, (aa) Color image corrupted by 50% Salt and Pepper noise, (ba) Color image corrupted by 6 pixel wide strip line artifact, (ca) Color image corrupted by 6*6 pixel size blotches artifacts,(ab–cb) Image restored using proposed method from (aa–ca), (ac–cc) Image restored using the ROAD-TGM from (aa–ca), (ad–cd) Image restored using the DBMF from (aa–ca), (ae–ce) Image restored using the CWMF from (aa–ca), (af–cf) Image restored using the PSMF from (aa–ca), (ag–cg) Image restored using the MF from (aa–ca). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

The PSNR, SSIM and IEF values for each image and for each noise and artifact level were calculated for the denoised image produced by both the proposed and existing algorithms. These results are shown in box plots representation for all noise levels and all images. The results prove that the proposed method is better than all other algorithms in comparison, producing better PSNR, SSIM and IEF values in all cases. Typically, the overall mean PSNR value of the proposed method is 23.95, which is higher than all other models: ROAD-TGM (PSNR = 19.82), DBMF (PSNR = 18.17), CWMF (PSNR = 16.03), PSMF (PSNR = 15.50), and MF (PSNR = 14.68).

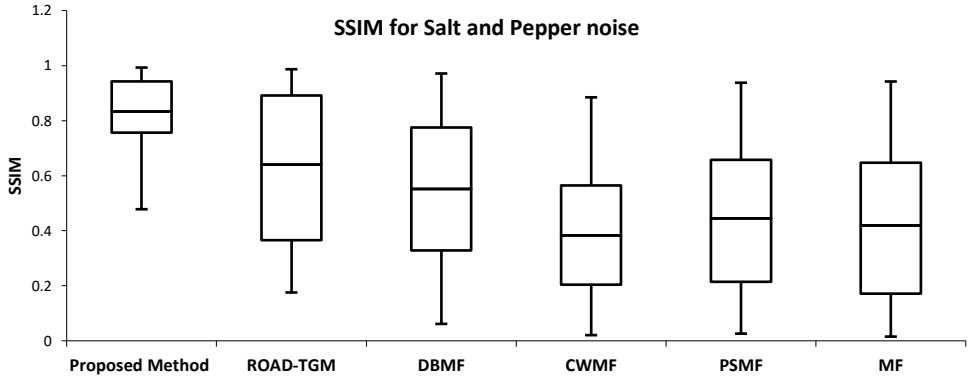
The box plots also show that the proposed method outperformed all other algorithms in terms of the SSIM parameter. The mean SSIM value of the proposed method is 0.83, which is higher than values for all other models: ROAD-TGM (SSIM = 0.63), DBMF (SSIM = 0.55), CWMF (SSIM = 0.38), PSMF (SSIM = 0.44) and MF (SSIM = 0.41). The mean IEF value of the proposed method is 47.12, which is higher than values for all other models: ROAD-TGM (IEF = 17.96), DBMF (IEF = 9.15), CWMF (IEF = 7.0), PSMF (IEF = 5.73) and MF (IEF = 4.81).

The results of various denoising techniques on images corrupted by strip lines and blotches artifacts are shown in Fig.4.19 and Fig 4.20. Again, the performance of the proposed method is better than all other algorithms, producing better PSNR, SSIM and IEF values in all cases. The mean PSNR, mean SSIM and mean IEF values of the proposed method for strip lines artifact are: PSNR= 25.73, SSIM=0.92, IEF=17.71 and for blotches artifacts are: PSNR= 45.51, SSIM=0.99, IEF=20.80, which are higher than all other algorithms considered in the comparison. The box plots clearly demonstrate the superiority of the proposed method compared to other existing algorithms.

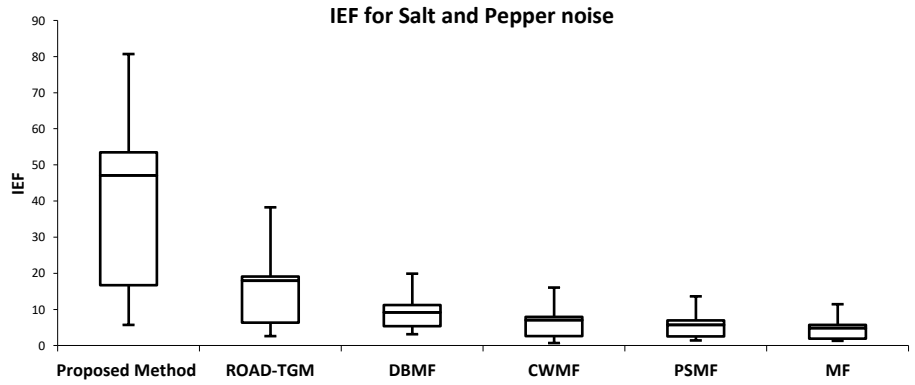
Our comparative analysis of denoising algorithms on color images affected by Salt and Pepper noise, Strip lines artifacts and Blotches artifacts are shown in Fig. 4.21 and Fig. 4.23, respectively. In the case of noise and artifacts, the proposed method produced a PSNR value of 36.37, SSIM value of 0.92 and IEF value of 72.23, which were higher than ROAD-TGM, DBMF, CWMF, PSMF, and MF algorithms.



(a)

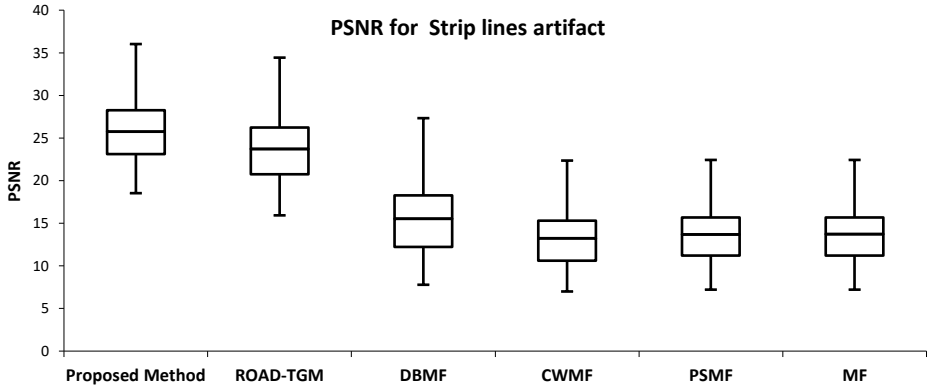


(b)

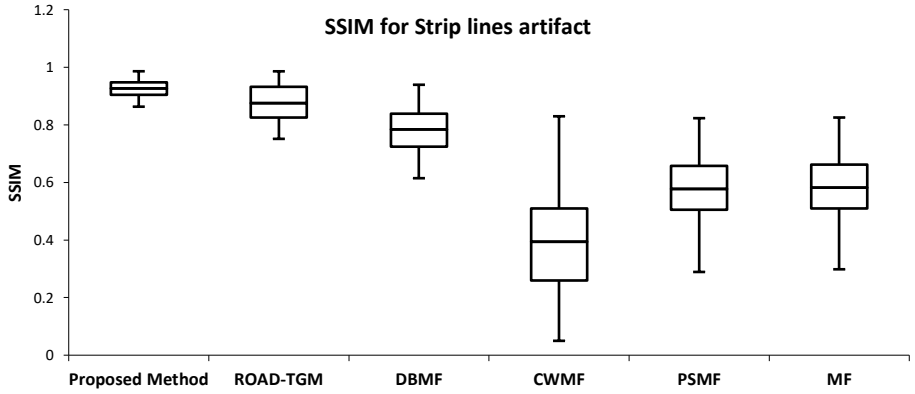


(c)

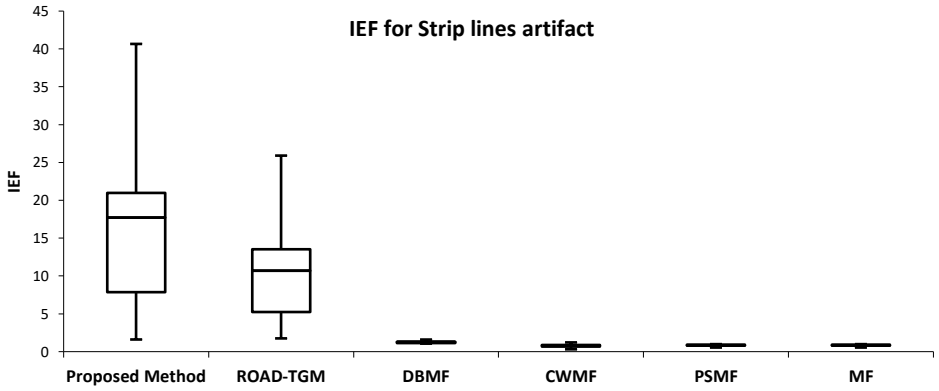
Figure 4.18. Comparative analysis of algorithms for 10% to 80% Salt and Pepper noise-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image enhancement factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.



(a)

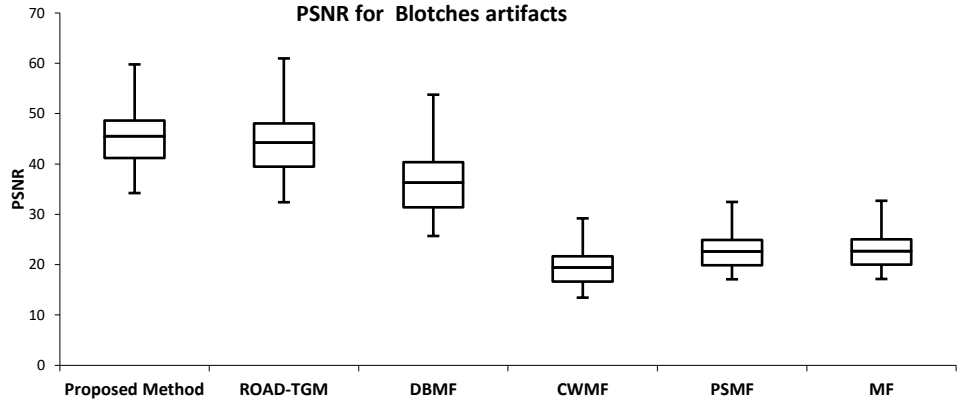


(b)

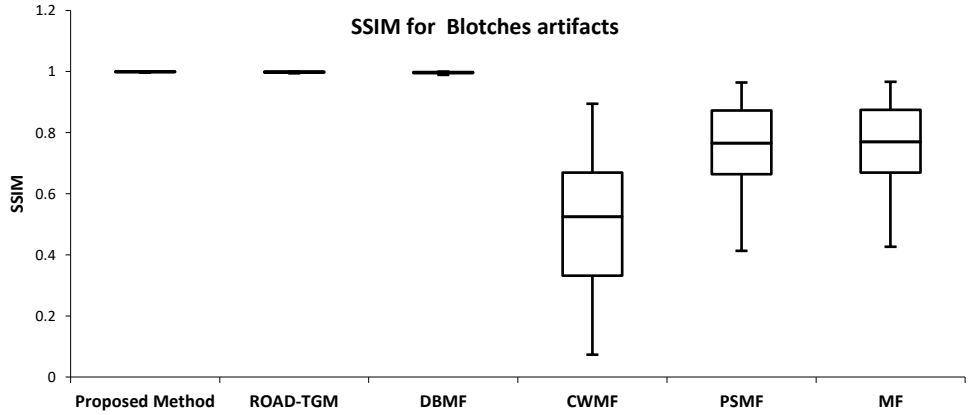


(c)

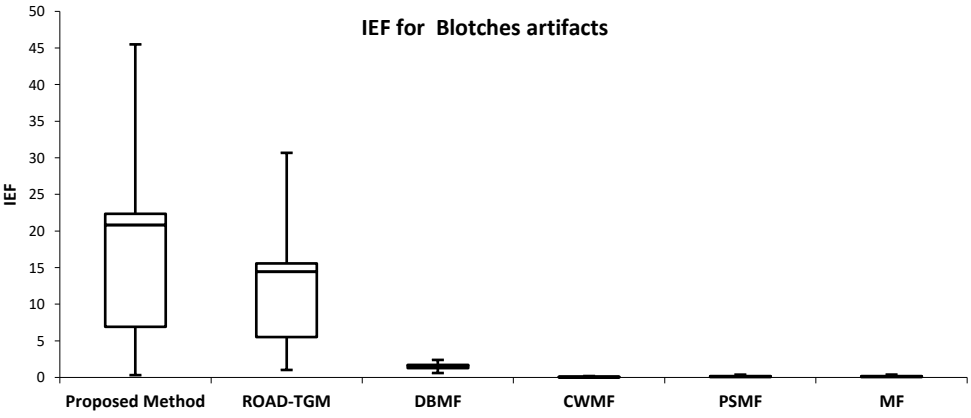
Figure 4.19. Comparative analysis of algorithms for 2 pixels to 9 pixels wide strip line-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.



(a)

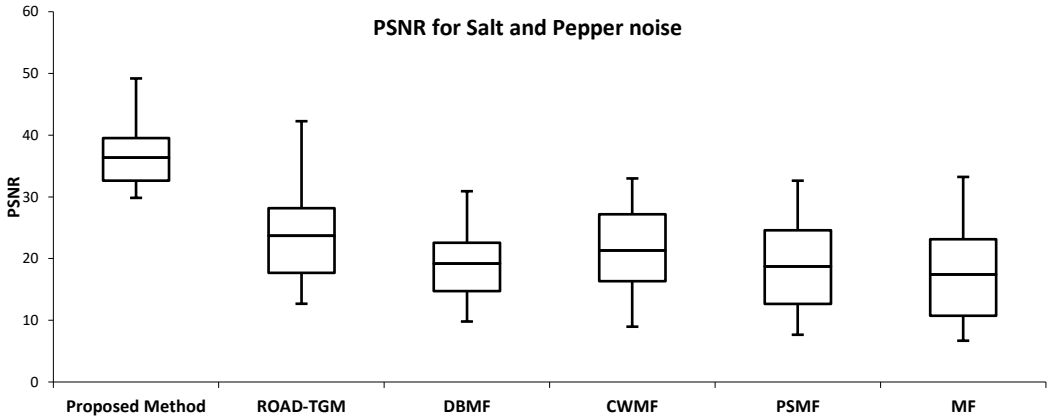


(b)

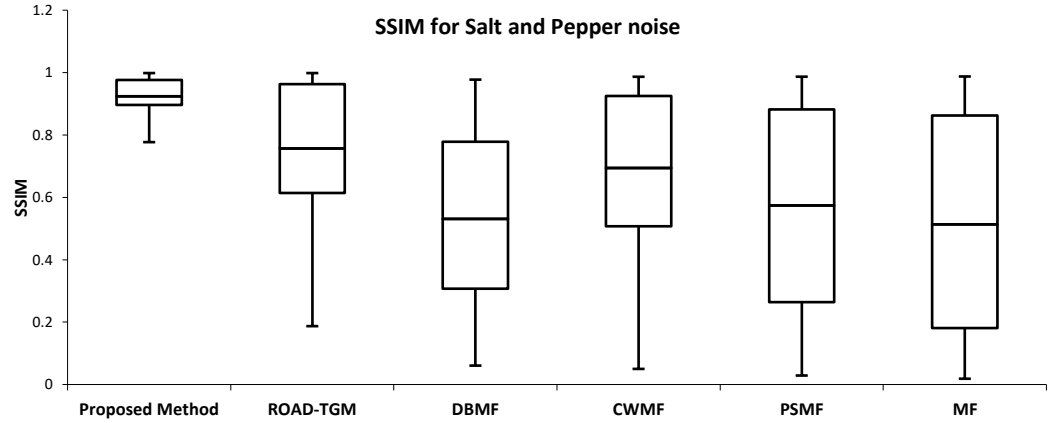


(c)

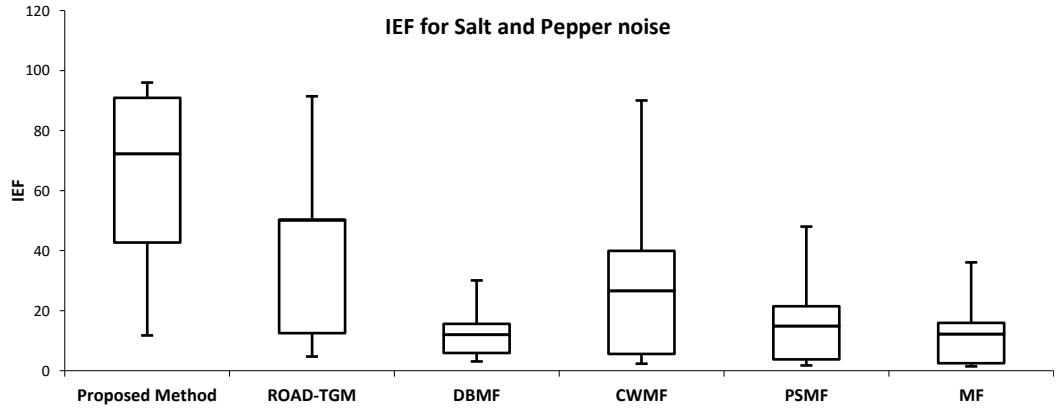
Figure 4.20. Comparative analysis of algorithms for 2*2 pixel to 9*9 pixel size blotches-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.



(a)

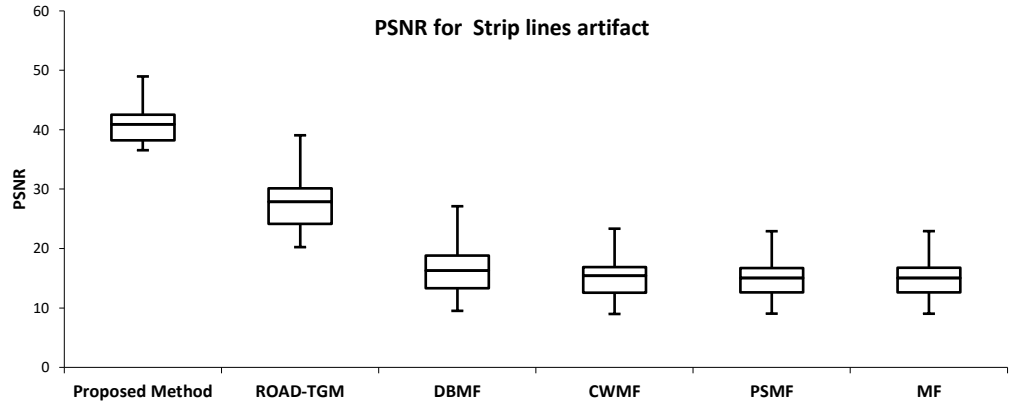


(b)

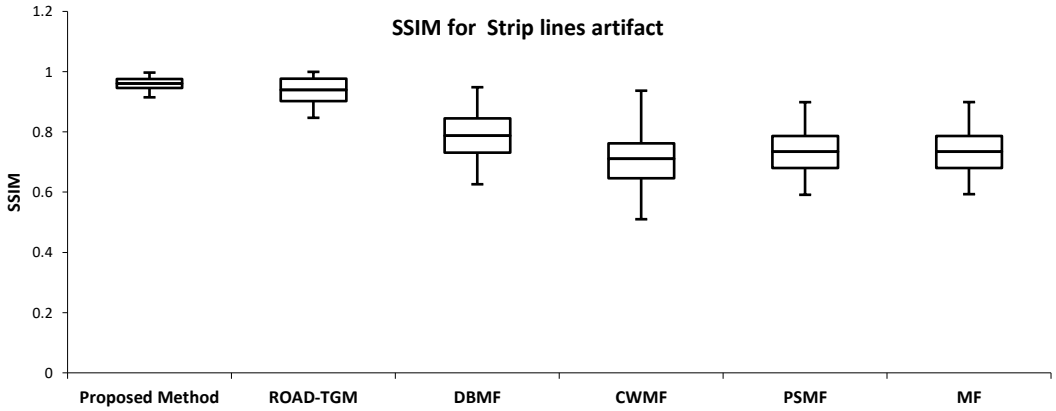


(c)

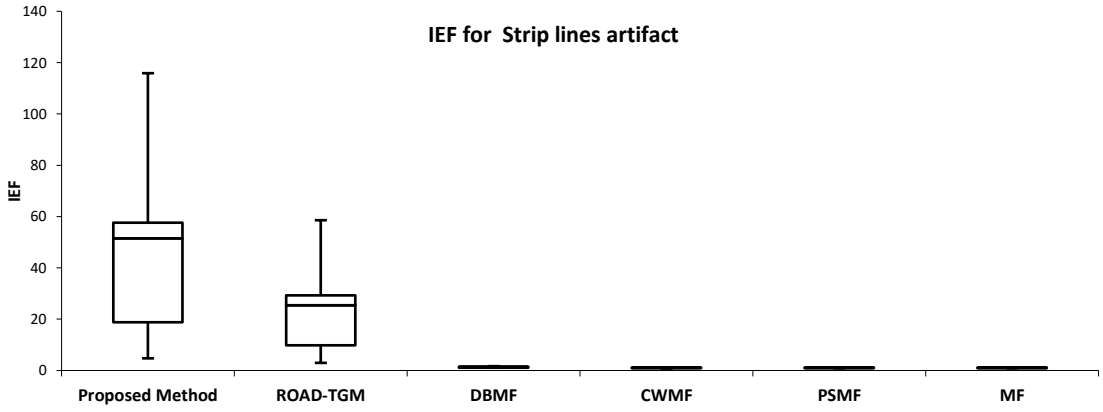
Figure 4.21. Comparative analysis of algorithms for 10% to 80% Salt and Pepper noise-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image enhancement factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.



(a)

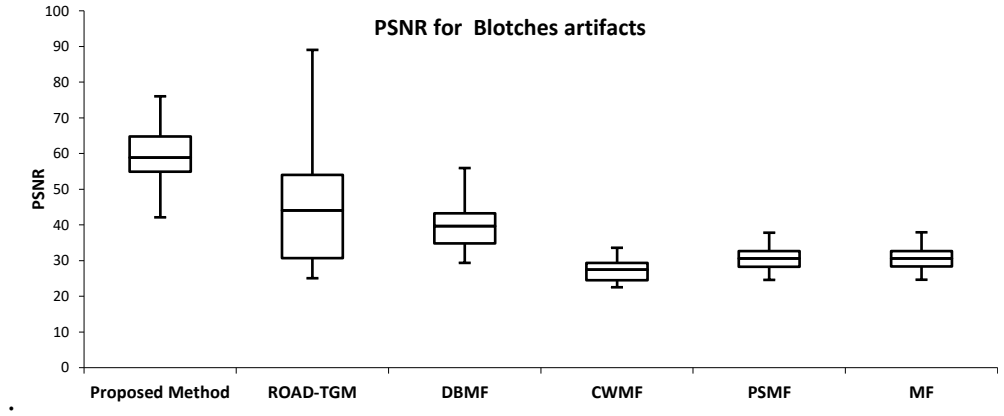


(b)

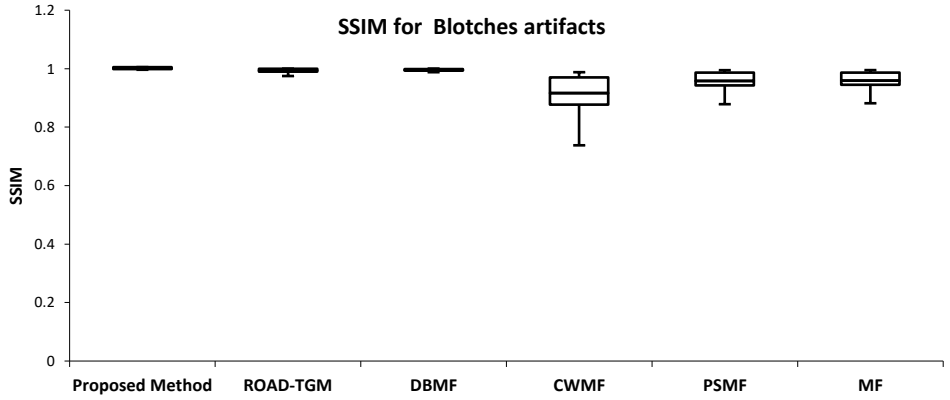


(c)

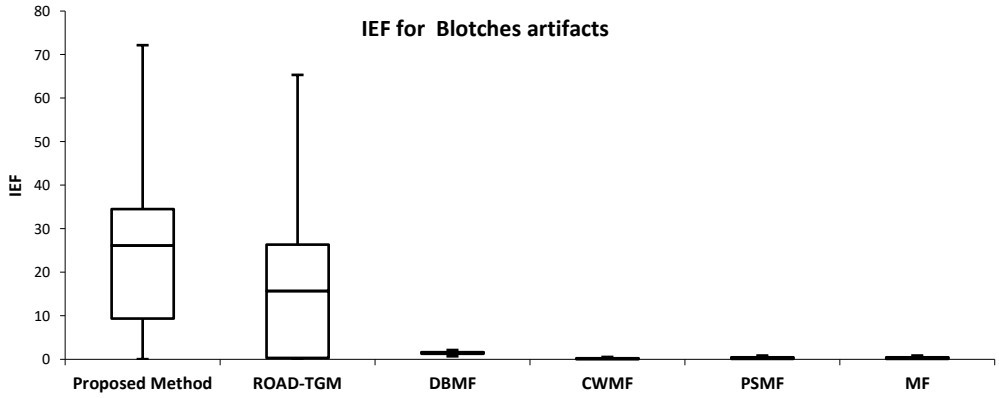
Figure 4.22. Comparative analysis of algorithms for 2 pixels to 9 pixels wide strip line-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.



(a)



(b)



(c)

Figure 4.23. Comparative analysis of algorithms for 2*2 pixel to 9*9 pixel size blotches-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

Table 4.1. Color image denoising comparison of the proposed method with the recent state of the art methods.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	AUTSF [63]	PSNR	41.51	38.12	35.91	34.20	32.60	31.05	29.22	27.38	24.29	32.70
		SSIM	–	–	–	–	–	–	–	–	–	–
	MCF [64]	PSNR	–	–	38.95	36.55	34.20	31.42	26.41	21.34	8.46	28.19
		SSIM	–	–	–	–	–	–	–	–	–	–
	SAID-END Method	**PSNR**	**44.20**	**41.34**	**38.99**	**36.66**	**34.32**	**32.93**	**31.96**	**30.11**	**27.82**	**35.35**
		SSIM	**0.9855**	**0.9740**	**0.9663**	**0.9544**	**0.9399**	**0.9295**	**0.9080**	**0.8721**	**0.8346**	**0.9294**

*For Table 4.1-4.2 The values in bold represent better PSNR and SSIM as compared to several state-of-the-art algorithms. ** For Table 4.1-4.2 '—' indicates value not available for denoising method.

In the case of Strip lines artifacts and Blotches artifacts, the proposed method outperformed the other existing algorithms, with a PSNR value of 40.91,58.83; SSIM value of 0.96, 0.99 and IEF value of 51.45,26.12 respectively.

The mean PSNR, mean SSIM and mean IEF values of the proposed method for strip lines artifact are: PSNR= 25.73, SSIM=0.92, IEF=17.71 and for blotches artifacts are: PSNR= 45.51, SSIM=0.99, IEF=20.80, which are higher than all other algorithms considered in the comparison. The box plots clearly demonstrate the superiority of the proposed method compared to other existing algorithms. Our comparative analysis of denoising algorithms on color images affected by Salt and Pepper noise, Strip lines artifacts and Blotches artifacts are shown in Fig. 4.21 and Fig. 4.23, respectively. In the case of noise and artifacts, the proposed method produced a PSNR value of 36.37, SSIM value of 0.92 and IEF value of 72.23, which were higher than ROAD-TGM, DBMF, CWMF, PSMF, and MF algorithms. In the case of Strip lines artifacts and Blotches artifacts, the proposed method outperformed the other existing algorithms, with a PSNR value of 40.91,58.83; SSIM value of 0.96, 0.99 and IEF value of 51.45,26.12 respectively.

In the second stage of comparison, the proposed method is evaluated with the recent state of art methods. This comparison is performed for both grayscale and color image

as discussed in Section II. For color image denoising comparison, colored Lena image affected with (10% to 90%) impulse noise is used as a test image. The main parameter in this comparison is PSNR as this parameter is commonly used by recent methods for color image denoising, but for proposed method SSIM results are also presented along with PSNR values in Table 4.1. The proposed method outperforms the recent state of the art methods by gaining superior mean PSNR value 35.35 for the noise range of 10% to 90% in comparison to denoising performance of AUTSF (32.70) and IMF (28.19). For color image denoising proposed algorithm obtained high mean SSIM value '0.93' for the image affected with low to high density of noise. The proposed method has also achieved high value of SSIM (0.9294) parameter which reflects good image quality.

Table 4.2. Grayscale image denoising comparison of the proposed method with recent state of the art methods.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	FDS [65]	PSNR	41.40	37.25	34.49	31.67	28.99	26.54	23.95	21.39	18.30	29.33
		SSIM	0.9894	0.9759	0.9573	0.9293	0.8858	0.8280	0.7441	0.6379	0.5020	0.83
	DAMF [61]	PSNR	42.97	39.29	36.84	34.94	33.21	31.64	30.22	28.53	25.93	33.73
		SSIM	0.9902	0.9788	0.9655	0.9494	0.9304	0.9064	0.8770	0.8370	0.7620	0.91
	IIN [66]	PSNR	–	31.43	29.50	27.62	26.39	–	–	–	–	28.74
		SSIM	–	–	–	–	–	–	–	–	–	–
	IMF [62]	PSNR	43.48	40.18	37.05	35.40	33.98	32.49	31.23	29.70	27.42	34.55
		SSIM	0.9913	0.9796	0.9675	0.9541	0.9383	0.9183	0.8953	0.8623	0.8058	0.92
	SAID-END Method	**PSNR**	**45.21**	**42.35**	**39.80**	**37.67**	**35.33**	**33.94**	**32.97**	**31.12**	**28.83**	**36.36**
		SSIM	**0.9957**	**0.9842**	**0.9765**	**0.9646**	**0.9501**	**0.9397**	**0.9182**	**0.8823**	**0.8448**	**0.94**
Peppers	FDS [65]	PSNR	40.65	36.9	34.32	31.72	29.32	26.83	24.11	21.37	18.15	29.26
		SSIM	0.9825	0.9627	0.9396	0.9079	0.8687	0.8110	0.7355	0.6360	0.5085	0.82
	DAMF [61]	PSNR	41.52	37.89	35.67	33.95	32.55	31.31	29.79	28.28	25.87	32.98
		SSIM	0.9815	0.9606	0.9389	0.9131	0.8866	0.8541	0.8180	0.7719	0.7049	0.87
	IIN [66]	PSNR	–	27.23	26.21	24.82	23.98	–	–	–	–	25.56
		SSIM	–	–	–	–	–	–	–	–	–	–
	IMF [62]	PSNR	41.83	38.59	36.65	35.14	33.9	32.69	31.43	30.01	27.88	34.24
		SSIM	0.9858	0.9684	0.9504	0.9299	0.9091	0.8846	0.8572	0.8219	0.7700	0.90
	SAID-END Method	**PSNR**	**43.41**	**40.13**	**38.13**	**36.76**	**35.08**	**33.84**	**32.91**	**31.18**	**29.83**	**35.70**
		SSIM	**0.9922**	**0.9863**	**0.9674**	**0.9433**	**0.9256**	**0.9009**	**0.8799**	**0.8521**	**0.8187**	**0.92**

Traditional Lena and Peppers grayscale images corrupted with low to high density of impulse noise (10% to 90%) are denoised using the proposed method. This comparison with FDS, DAMF, IIN and IMF methods is presented in Table 4.2, where '—' indicate

unavailability of value. To achieve fair performance comparison, only common test images and statistical paraments of the recent state of the art algorithms are used. Performance evaluation of grayscale images using parameter PSNR and SSIM, shows the superiority of the proposed method among the recent state of the art algorithms (refer to Table 4.2). The proposed method achieves mean PSNR value of 36.36 in comparison to lower values of the recent state of the art methods (FDS=29.33, DAMF=33.73, IIN=28.74, IMF=34.45) for grayscale Lena image. Again, on Lena grayscale test image the proposed method achieves better performance for SSIM parameter by obtaining the mean SSIM value 0.94 in comparison to the recent state of art algorithms (FDS=0.83, DAMF=0.91, IIN=N.A, IMF=0.92). Similarly, the proposed method performs well on the second traditional test image (Peppers). The proposed method repeats its success over the recent state of art methods for both the parameters PSNR and SSIM by achieving higher values. The proposed method achieves mean parameter (PSNR/SSIM) values (35.71/0.92) followed by FDS (29.26/082), DAMF (32.98/087), IIN (25.56/N.A) and IMF (34.24/0.90). Table 4.3 represents the comparison of proposed method with existing denoising techniques for Grayscale and Color image dataset.

Table 4.3. Grayscale and Color image denoising comparison of the proposed method with the existing methods.

Image type	Noise type	Parameter	Proposed Adaptive	Road-TGM	DBMF	CWMF	PSMF	MF
Grayscale dataset 50 images	Salt and Pepper (10%-80%)	Mean PSNR	**23.953**	19.825	18.002	15.966	15.456	15.181
		MeanSSIM	**0.834**	0.640	0.551	0.383	0.444	0.419
		Mean IEF	**47.125**	17.960	9.158	7.007	5.736	4.818
	Blotchs (2x2 - 9x9)	Mean PSNR	**45.515**	44.214	36.299	19.442	22.636	22.713
		MeanSSIM	**0.999**	0.998	0.997	0.525	0.765	0.770
		Mean IEF	**20.809**	14.447	1.643	0.068	0.128	0.130
	Strip lines (2 - 9)	Mean PSNR	**25.739**	23.723	15.533	13.235	13.706	13.719
		MeanSSIM	**0.926**	0.875	0.784	0.394	0.578	0.582
		Mean IEF	**17.714**	10.688	1.278	0.781	0.841	0.844
Color image dataset 16 images	Salt and Pepper (10%-80%)	Mean PSNR	**36.372**	23.700	19.170	21.323	18.733	17.403
		MeanSSIM	**0.924**	0.756	0.531	0.694	0.574	0.513
		Mean IEF	**73.678**	50.175	11.958	26.639	14.837	12.223
	Blotchs (2x2 - 9x9)	Mean PSNR	**60.010**	44.064	39.672	27.423	30.588	30.635
		MeanSSIM	**0.996**	0.994	0.994	0.916	0.958	0.959
		Mean IEF	**26.646**	15.670	1.533	0.164	0.292	0.294
	Strip lines (2 - 9)	Mean PSNR	**41.733**	27.876	16.300	15.402	15.048	15.051
		MeanSSIM	**0.965**	0.939	0.788	0.711	0.734	0.735
		Mean IEF	**52.481**	25.358	1.301	1.138	0.961	0.962

4.5 Conclusion

In order to overcome the performance issues of the existing denoising methods, a two-stage SAID-END denoising algorithm has been proposed. At first, the proposed method overcomes the issues of over/under detection of noisy pixels by using enhanced adaptive noise detection stage. This stage uses systematic thresholding with iterative similarity indexing to ensure the accurate categorization of noisy and non-noisy pixels. After the classification of pixels, the task was to reduce the impact of high-density noise on original value estimation which was achieved by using a non-corrupted pixel sensitive adaptive image restoration stage. This stage ensures the computation of restored value would be carried out only when a good amount of non-corrupted pixels are available in the window. This process has been implemented using an adaptive window mechanism with non-corrupted pixel ratio criteria. Once the non-corrupted pixel ratio criteria is satisfied, the original value of noisy pixel was restored using statistical measure i.e., median of non-corrupted pixel values. The two-stage test has been carried out on the proposed method to evaluate its performance. The first test was carried out to validate the operativity of the proposed method on a wide range of noise and artifacts affected dataset. The proposed method has shown better PSNR, SSIM and IEF performance when compared with some well-known algorithms for a wide dataset of color and grayscale images. The second test stage was performed to evaluate the proposed method in comparison to the recent state of art algorithms. The commonly referred traditional test images have been used to perform this comparison. The proposed algorithm has shown improved performance on the basis of PSNR and SSIM parameters. In the future, this work can be extended by increasing the proximity of restored value to the original value to achieve higher detail preservation.

Chapter 5

Amalgamation of ROAD-TGM or SAID-END with Progressive PCA Using Performance Booster Method for Detail Preserving Image Denoising

This chapter presents two algorithms ROAD TGM-PPCA-PB and SAID END-PPCA-PB each having a two-stage sequential method for denoising grayscale and color images. Firstly, discussion will be with respect to ROAD TGM-PPCA-PB and by replacing the ROAD TGM algorithm with SAID END algorithm we can achieve SAID END-PPCA-PB algorithm which is having highest performance for impulse noise denoising. At first ROAD TGM-PPCA-PB method enhances the accuracy of the noise detection stage by using spatial domain filter rank-order absolute difference trimmed global mean (ROAD-TGM) along with transform domain-based progressive principle component analysis (PPCA) method. Then the performance booster algorithm is used to ensure the proximity of restored value to the original value. Quite often in real-world applications images are corrupted with Salt & Pepper noise, Strip lines artifact and Blotches artifact. We observed the proposed method is capable of removing the above-mentioned noises and artifacts with comparatively better accuracy.

The proposed method uses the progressive PCA for its dimension reduction ability and local information of image restored by ROAD-TGM to provide enhanced noise detection performance. Before noise removal, a performance booster algorithm eliminates noisy values by using sequential hard thresholding and estimates the tentative original values automatically. Then algorithm decides the suitable value for the restoration of noise pixel by using a structural similarity index (SSIM) to ensure the proximity of restored image to the original image. The proposed algorithm is tested on a standard set of color and grayscale images to ensure the versatility of the proposed algorithm. The experiment shows that the proposed algorithms achieves high denoising performance for noise and artifact while maintaining the visually important details.

5.1 Introduction

It is important for applications of image processing and computer vision to obtain detailed visual information from visual imaging sensors[101], [102]. Acquiring useful detailed visual information is only possible if details of the image are free from noise. It's always a need of the hour to achieve a higher quality detail preservation image denoising method to keep the visual detail of the image intact. The images are usually deteriorated due to the effect of Salt & Pepper noise[6], [90], [43], [103], [104], Blotches[42], [95], [105] and Strip lines[19], [106] artifacts. Salt & Pepper noise tends to change any pixel color into either white or black. Blotch affects the images as a patch or set of pixels have unusual values than the surrounding. Similarly, Strip lines will be degrading the image quality in the form of vertical or horizontal lines.

The denoising techniques are generally bifurcated into two domain procedures i.e. Spatial domain and Transform domain. Spatial domain algorithms are mostly depending upon the knowledge of image properties like similarity of pixel intensity in a region. Natural images in the spatial domain can be divided into a set of pixel groups (groups based on the similarity of pixels) and each group of pixels has a different rage of pixel values. Some of the spatial domain filters are Median filter (MF)[43], [107], [108], Decision Base Median Filter (DBMF)[11], [90], [109], Center Weight Median Filter (CWMF)[58], [59], [110], Progressive Switching Median Filter (PSMF)[11], [61], [109], Rank Ordered Absolute Differences with Trimmed Global Mean Filter (ROAD-TGM)[8] and Iterative Mean Filter (IMF)[62]. MF is one of the basic yet powerful filter for a low level of noise. This filter introduces a blurring effect while denoising the high-level noise. MF is good for the removal of Salt & Pepper noise but it is unable to repeat the same performance for artifacts. DBMF is the filter that works in two stages. The first stage decides whether the pixel is noisy or non-noisy and the second stage replaces the noisy pixel with its median value. This filter maintains the quality of image details while denoising it for the low level of noise. This filter does not perform in the same manner for the high level of noise. CWMF is window base technique, the algorithm provides the more weightage to center pixel values of the window. This filter performs good for impulse noise like Salt & Pepper but does not achieve the same performance for artifacts. PSMF is a two-stage filter, the first stage

will identify the noisy and non-noisy pixels. The second stage filters the noisy pixel with an iterative mechanism due to which this method can preserve the edges of the image. The drawback of PSMF is it cannot maintain the same performance for noise apart from Salt & Pepper. ROAD-TGM also works with a two-stage procedure where stage one detects the noisy pixel by using ROAD algorithm and the second stage restores the noisy image with a median value of non-noisy pixels of the respective window. In this algorithm, restoration is performed by TGM method when the selected window does not have noise-free pixels. ROAD-TGM is also capable of detail preservation and can work on artifacts along with Salt & Pepper noise. IMF is one of the promising methods for Salt & Pepper noise, this method uses iterative process for denoising of image. It works on a fixed window which increase the operational speed but limits its proximity to original pixel in case of high noise.

The wavelets and principal component analysis (PCA)[111] are examples of transform base procedures. Some algorithms based on transforms are Bayes least squares with a Gaussian scale-mixture (BLS-GSM)[112], dictionary learning (DL)[113], [114] based denoising etc. BLS-GSM algorithm performs well for a different level of noise but due to a fixed wavelet basis, it does not maintain structural integrity. These stated problems were resolved in DL method by replacing fixed bases with adaptive bases and provides better spatial image representation. PCA is one of the prominent techniques used together with other various algorithms to enhance their performance due to its dimension reduction abilities. In the dimension reduction process, PCA reduces random values by considering certain principle values. An example of such an algorithm is linear minimum mean square error estimation (LMMSE)[115] with PCA. Moreover, the best denoising performance with detail preservation can be achieved by combining the advantages of both the spatial domain and transform domain-based techniques. Some efforts are already made in this field i.e. block-matching 3-D shape-adaptive principle component analysis (BM3DSAPCA)[116] where the performance of BM3D is enhanced by PCA. As the PCA method estimates the unknown level of noise in the image without considering any prior information, its estimation may deviate from the original noise value. So, it is required that a performance booster method for estimation of original value should be designed to ensure the high quality of the denoised image.

Although these algorithms produce good results of denoising for a wide range of noise levels but due to above-stated issues they introduce blurring effect in the resultant image. So, it becomes important to design a high-quality image denoising algorithm that can denoise the image by maintaining the high quality of visual information in the image.

A set of parameters is always preferred to ensure and evaluate image quality produced by the image denoising algorithm. These parameters are Peak Signal to Noise Ratio (PSNR)[40], [61], [117], Image Enhancement Factor (IEF)[19], [109], [118] and Structural Similarity (SSIM)[51], [61], [119], [120]. To conclude the issues of image denoising we can summarize the issues in two stages, first an algorithm with qualities of spatial domain and transform domain is need of the hour to achieve high noise detection accuracy. Second, a performance booster algorithm is required to ensure the proximity of randomly estimated values to the original value for enhancing the quality of the restoration stage of the denoising process.

The proposed algorithm addresses these issues by providing multi-domain base denoising with a performance booster algorithm to ensure high-quality denoising. The algorithm is based on ROAD-TGM with progressive PCA [114], [121], [122] to achieve detail preservation with accurate denoising. We are utilizing the ability of ROAD-TGM to denoise images affected by a wide range of noise and artifacts. But due to the estimation of restored values using the median and mean function in ROAD-TGM, the values may deviate from the original value with a significant margin which gives us a scope of increasing the proximity of estimated value to the original value. To achieve better prediction of the noisy values, the algorithm is also including the random value estimation ability of progressive PCA. To maintain the estimation of restored values close to original values, we propose a performance booster algorithm. The performance booster algorithm first uses the hard threshold to eliminate the extreme estimation and then uses a set of similarity mapping steps to maintain the proximity of

restored value with original value without having any prior information of original value.

This chapter makes two Substantial contributions:

1. In this chapter, we present an effective detection method which can estimate both noise (Salt & Pepper noise) and artifacts (Strip lines and Blotches). The proposed method is based on the amalgamation of ROAD-TGM or SAID END with progressive PCA.

2. For enhanced restoration, we propose a performance booster algorithm using a window base systematic thresholding and similarity index evaluation. The proposed algorithm can automatically identify the suitable value for noisy pixel restoration in the image. Due to which it enhances the proximity to the original values.

5.2 Preliminaries

This section briefly describes the image denoising methods used in this analysis as well as form the theoretical basis for this chapter. The image denoising algorithms are mainly categories as transform domain and spatial domain. The features of spatial and transform domain algorithms are effectively incorporate in the proposed algorithm to ensure desired performance.

A. ROAD-TGM (Ranked Order Absolute Difference with Trimmed Global Mean)

This method uses ROAD algorithm to detect the location of noisy pixels and restore it by using a median of non-noisy values. In case all of the pixels in the selected window are noisy then restoration is applied by using TGM[8]. Let's understand this process in detail. Consider $\boldsymbol{l} = (\boldsymbol{r}, \boldsymbol{c})$ be the location of the pixel under evaluation. Window size should always be odd as the window will be centered around the pixel $\boldsymbol{c_p}$. The set of pixels in a window size of $(\boldsymbol{2n+1}) \times (\boldsymbol{2n+1})$ are represented by $\boldsymbol{\varphi}$. Here $\boldsymbol{n}$ is positive integer $\boldsymbol{n \geq 1}$.

$$\boldsymbol{\varphi(n) = \{c_p + (i,j) : -n \leq i,j \leq n\}} \quad (1)$$

To be specific let's consider $\boldsymbol{n = 2}$ and under this condition the total number of pixels in the $\boldsymbol{5 \times 5}$ window.

$$\boldsymbol{\varphi^* = \varphi(5)} \quad (2)$$

To create the difference vector from a matrix $\boldsymbol{\varphi}^*$, the absolute difference of $\boldsymbol{c_p}$ intensity value in the window is calculated with its neighbors $\boldsymbol{\varphi^*_{l1,l2}}$.Absolute difference vector is represented as $\boldsymbol{D_{l1,l2}}$ and its give below.

$$\boldsymbol{D_{l1,l2}} = \left|\boldsymbol{\varphi^*_{l1,l2}} - \boldsymbol{c_p}\right| \quad (3)$$

Sort the difference vector $\boldsymbol{D_{l1,l2}}$ and arrange it in increasing order to create $\boldsymbol{ID}$. Then ROAD can be calculated by using equation 4.

$$\boldsymbol{ROAD_N(l)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{ID(l)} \quad (4)$$

Where $\boldsymbol{2 \leq N \leq 7}$

ROAD algorithm provides information about closeness of neighbor pixels to its center pixel $\boldsymbol{c_p}$. ROAD algorithms work on a basic concept that in natural images pixel intensity in a selected area is close to each other and any random noise proves to have high variation in comparison to original values.

Trimmed global mean (TGM) algorithm detects the noisy pixels by using a hard threshold. On grayscale image intensity level pixel $\boldsymbol{X(r,c)}$ is declared as corrupted pixels, if this pixel is not in the range of $\boldsymbol{0 \leq X(r,c) \leq 255}$. If all the neighbor pixels to $\boldsymbol{c_p}$ are non-corrupted then median of non-corrupted neighbor is calculated and replace the $\boldsymbol{c_p}$ location value with the calculated value. In the case of image affected by a higher level of noise, it is possible that no neighbor is have non-corrupted value. In this case $\boldsymbol{c_p}$ (noisy center pixel of the window) is replaced by TGM. TGM is calculated by ignoring all the corrupted values of an image and taking the mean of all non-corrupted values image. TGM is only preferred in this algorithm when the entire values of the selected window are corrupted.

B. Progressive PCA

The Progressive PCA is based on a two-stage sequential process. In the first stage, the algorithm divides the given image into equal or size varying partitions (equal size of partitions are preferred) to create non-overlapping subpartitions of the image. In the second stage of the algorithm, the PCA is progressively applied to subsets generated in the first stage[121]. For more understanding let's assume a set of patterns are given in variable $\boldsymbol{Im}$, expressed as below.

$$\boldsymbol{Im} = \left\{\boldsymbol{Im_1}, \boldsymbol{Im_2}, \boldsymbol{Im_{3,\ldots\ldots,}Im_N}\right\} \quad (5)$$

Each column of given variable $\boldsymbol{Im}$ is expressed as a column vector $\boldsymbol{Im_i(i = 1, 2, 3, \ldots\ldots, N)}$. As we already know that the task of first stage is to divide the given information into X y-dimensional non-overlapping subsets to form d by X matrix.

$$\boldsymbol{Im_i = \{Im_{i1}, Im_{i2}, Im_{i3,\ldots\ldots,} Im_{iX}\}} \tag{6}$$

With

$$\boldsymbol{Im_{ij} = \left(Im_{i((j-1)y+1)}, \ldots\ldots, Im_{i(jy)}\right)^T} \tag{7}$$

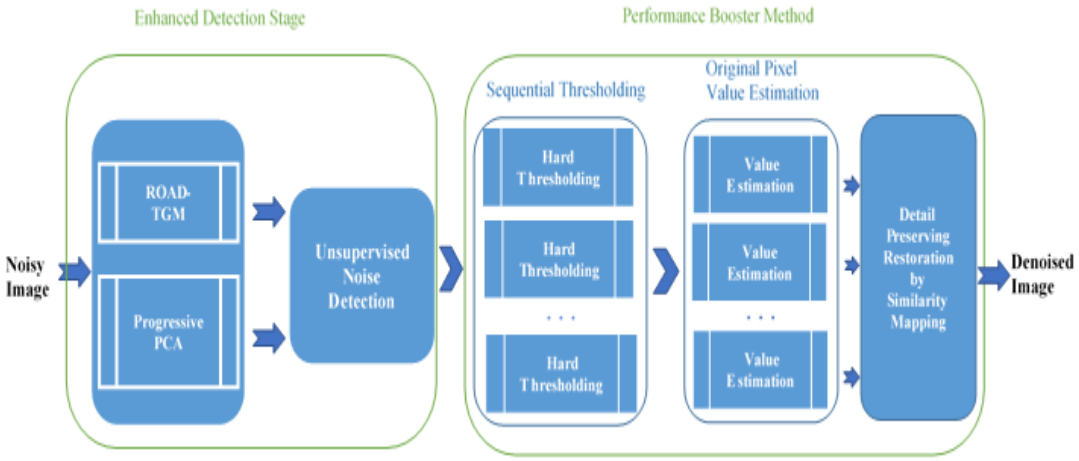


Figure 5.1. The flowchart of the proposed algorithm.

As considered $\boldsymbol{j^{th}}$ subset of $\boldsymbol{Im_i}$ where $\boldsymbol{i = 1, 2, 3, \ldots, N}$ and $\boldsymbol{j = 1, 2, 3, \ldots, X}$. After the competition of the first step, the PCA is applied to subsets progressively. The progressive approach is implemented by applying PCA to subsets from left to right.$\mathbf{PCA_1}$ is obtained by performing the second stage on the first subset $\boldsymbol{\{Im_{11}, Im_{21}, Im_{31,\ldots\ldots,} Im_{N1}\}}$ and obtain the respective reduced subset $\boldsymbol{\{Rim_{11}, Rim_{21}, Rim_{31,\ldots\ldots,} Rim_{N1}\}}$. The subset formed in $\mathbf{PCA_1}$ are shown in equation 8.

$$\boldsymbol{\left\{\begin{pmatrix} Rim_{11} \\ Im_{11} \end{pmatrix}, \begin{pmatrix} Rim_{21} \\ Im_{21} \end{pmatrix}, \begin{pmatrix} Rim_{31} \\ Im_{31} \end{pmatrix}, \ldots\ldots, \begin{pmatrix} Rim_{N1} \\ Im_{N1} \end{pmatrix}\right\}} \tag{8}$$

Then these generated subsets are used to argument the directionality of new subsets, which are going to be the base of the next step of the progressive stage to form $\mathbf{PCA_2}$. So, by using similar steps we can form $\mathbf{PCA}_{\boldsymbol{x}}$ where $\boldsymbol{x} \leq \boldsymbol{X}$.

5.3 Implementation of Proposed Algorithm

The proposed algorithms work in two sequential stages. In the first stage algorithm work on the primary objective of estimating the noise location in the image. Then the second stage works on identifying the suitable value for the restoration of noisy pixel locations to achieve higher efficiency of denoising and detail preservation. The denoised image is obtained as a result of a combination of these two stages, where stage one ensures the accurate detection of noisy pixels and stage two ensures the pixel value restoration similar to the original value to obtain high accuracy with least distortion to details.

A. Amalgamation of ROAD-TGM and Progressive PCA for Global Noise Detection

The key parameter in the process of image denoising is noise estimation. To denoise the image affected by the noise, it is required to identify the location of noise in the image. The various method has been devolved in the spatial domain and transform domain. To achieve high accuracy of noise location detection. But these algorithms do have the scope of improvement, as the problem of over and under detection exists. We address this issue by the proposed algorithm by verifying the detected noisy pixel location for high accuracy. The verification process carried out by using the individual resultant images of both the algorithm's ROAD-TGM and progressive PCA.

Noisy image α with noise level β ($1 \leq \beta \leq 8$) is given as input to ROAD-TGM and progressive PCA algorithms for produce denoising image as a response. For the image denoising analysis Salt & Pepper noise, Blotch artifacts and Strip lines artifacts are used to produce noisy dataset Ð containing noisy images α. The noisy dataset Ð is expressed in equation 9.

$$Ð = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n] \quad (9)$$

Let the subscript â and ê are denoised images produced by ROAD-TGM and progressive PCA algorithms respectively. To achieve enhanced performance, it is required to ensure the accuracy of the noisy pixel detection stage. The unsupervised noise detection stage is designed for validation of noise estimation and it is achieved by equation 10-11.

$$\Phi = |\hat{a} - \hat{e}| \quad (10)$$

$$\Phi_m = \{0, \quad 0 < \Phi ; \text{ otherwise } 1 \quad (11)$$

Where Φ is difference matric reflecting the two cases. The case one contains zero values indicating original values with no change and case two contains non zero values representing noisy pixel location. This works on a fact that every algorithm works on different parameters which lead to similar yet different restored values of the pixel. Therefore, noise estimation accuracy increases as noisy pixel may remain undetected by ROAD-TGM may get detected by progressive PCA algorithms or vice-versa will get reflected in matrix Φ. Φ can be converted into a binary map Φ_m by using equation 11. The binary map Φ_m is an effective way to indicate the verified noisy pixel location on image â and ê by Determine Φ_p according to (13);

$$\Phi_r = \Phi_m \times \hat{a} \quad (12)$$

$$\Phi_p = \Phi_m \times \hat{e} \quad (13)$$

The matrix Φ_p and Φ_r are the superimposed noise maps with respect to ROAD-TGM and progressive PCA respectively. Further, these superimposed noise maps produced by enhanced detection stage are used in the noise restoration stage for forming the non-overlapping windows. The enhanced detection stage process is concluded in Algorithm 1.

Algorithm 1 Enhanced Noise Detection Algorithm

Input: Noisy image data set Ð.

Output: The noise location map $\Phi_{r,}\ \Phi_p$;

1: Input selection: Noisy image α_1 with noise level β from data set Ð.

2: ROAD-TGM: Do ROAD-TGM on image to get â.

3: Progressive PCA: Do: Progressive PCA on the image to get ê.

4: **for** both image matrix*s* â, ê **do**

5: Determine Φ according to (11);

6: Determine Φ_m according to (12);

7: Determine Φ_r according to (13);

8: Determine Φ_p according to (14);

9: End for

10: Return $\Phi_{r,}\ \Phi_p$.

B. Adaptive Performance Booster Method

Once the detection stage identifies the location of the noisy pixel. Non-overlapping windows are formed by considering the noisy pixel location in the center of the matrix. Let's consider the total number of detected noisy pixels as ñ. Therefore, the noisy pixel locations of the image $\boldsymbol{\Phi_p}$ and $\boldsymbol{\Phi_r}$ are used to divided â and ê into ñ patches of 3 × 3 to ensure high-speed operation of the algorithm. The subscript š and ž represent ñ patches of image â and ê represented using equation 14-15.

$$š = [\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}, \ldots, \boldsymbol{p}_{ñ}] \quad (14)$$

$$ž = [\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}, \ldots, \boldsymbol{r}_{ñ}] \quad (15)$$

where both $\boldsymbol{p_1}$ ***and*** $\boldsymbol{r_1}$ are 3 × 3 dimensional patches of image $\boldsymbol{\Phi_p}$ and $\boldsymbol{\Phi_r}$ respectively.

To achieve a higher performance of the restoration stage algorithm, we need to decrease the difference between the original pixel value and restored pixel value. Therefore, the algorithm eliminates the noisy values for pixel restoration by using sequential thresholding. The sequential hard thresholding is important to eliminate the cases where ROAD-TGM or progressive PCA restored a noisy value due to all noisy neighbors. The center value of the patch matrix $\boldsymbol{p}_{ñ}$ and $\boldsymbol{r}_{ñ}$ are validated by sequential hard thresholding. The selected value is considered only if it satisfies the conditions given in equation 16-19 and similarly hard thresholding can be applied for $\boldsymbol{r}_{ñ}$.

$$\boldsymbol{p}_{ñ}(\boldsymbol{2},\boldsymbol{2}) \neq \boldsymbol{255} \quad (16)$$

$$\boldsymbol{p}_{ñ}(\boldsymbol{2},\boldsymbol{2}) \neq \boldsymbol{0} \quad (17)$$

$$\boldsymbol{p}_{ñ}(\boldsymbol{2},\boldsymbol{2}) \neq \boldsymbol{max}(ê) \quad (18)$$

$$\boldsymbol{p}_{ñ}(\boldsymbol{2},\boldsymbol{2}) \neq \boldsymbol{min}(ê) \quad (19)$$

These sequential thresholds eliminate noisy restored value in case of Salt & Pepper, Strip lines and Blotches. To maintain the denoising performance under influence of high level of noise, additional hard thresholding conditions are required, so in this case thresholding is computed by using equations 20-22.

$$(\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2})-\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})) < \emptyset \quad (20)$$

$$\sum(\boldsymbol{p}_{\tilde{n}} - \boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2})) < \emptyset \quad (21)$$

$$\sum(\boldsymbol{r}_{\tilde{n}} - \boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})) < \emptyset \quad (22)$$

Subscript $\emptyset = \mathbf{20}$ represents the threshold for the highly noise affected case in this simulation. To enhance the value of the restored pixel, the difference between the original and restored value is required to minimize. Subscript K is weight to control the pixel value estimation, for this simulation K is having a range from 1 to10. Algorithms produce a set of tentative (T) original values by using equations 23-26.

$$\boldsymbol{T_1} = |(|\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})|/\boldsymbol{K})+\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2})| \quad (23)$$

$$\boldsymbol{T_2} = |(|\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})|/\boldsymbol{K})-\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2})| \quad (24)$$

$$\boldsymbol{T_3} = |(|\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})|/\boldsymbol{K})+\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})| \quad (25)$$

$$\boldsymbol{T_4} = |(|\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})|/\boldsymbol{K})-\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2})| \quad (26)$$

The next set of tentative original values are computed by considering the deviation percentage into account using equation 31-38. The subscript $\boldsymbol{DP}_{\tilde{N}}$ represent the deviation percentage, utilized to form subscript $\boldsymbol{D_r}$ and $\boldsymbol{D_p}$ which represents the absolute difference vector with respect to ROAD-TGM and progressive PCA respectively. The vector $\boldsymbol{D_{vr}}$ and $\boldsymbol{D_{vp}}$ are form similar to equation 3. The combined dataset of all $\boldsymbol{D_r}$ and $\boldsymbol{D_p}$ values for ñ windows is form by using equation 27-30.

$$\boldsymbol{D_{vr}} = [\boldsymbol{D_{r1}}, \boldsymbol{D_{r2}}, \boldsymbol{D_{r3}}, \ldots, \boldsymbol{D}_{r\tilde{n}}] \quad (27)$$

$$\boldsymbol{D_{vp}} = [\boldsymbol{D_{p1}}, \boldsymbol{D_{p2}}, \boldsymbol{D_{p3}}, \ldots, \boldsymbol{D}_{p\tilde{n}}] \quad (28)$$

$$\boldsymbol{DP}_{\tilde{n}} = \left(\left|\boldsymbol{min}(\boldsymbol{D}_{\boldsymbol{vp}_{\mathbf{1},\tilde{n}}})/\boldsymbol{max}(\boldsymbol{D}_{\boldsymbol{vp}_{\mathbf{1},\tilde{n}}})\right| \times \mathbf{100}\right) / \boldsymbol{K} \quad (29)$$

$$\boldsymbol{DR}_{\tilde{n}} = \left(\left|\boldsymbol{min}(\boldsymbol{D}_{\boldsymbol{rp}_{\mathbf{1},\tilde{n}}})/\boldsymbol{max}(\boldsymbol{D}_{\boldsymbol{rp}_{\mathbf{1},\tilde{n}}})\right| \times \mathbf{100}\right) / \boldsymbol{K} \quad (30)$$

$$\boldsymbol{T_5} = |\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2}) + \boldsymbol{DR}_{\tilde{n}}| \quad (31)$$

$$\boldsymbol{T_6} = |\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) + \boldsymbol{DP}_{\tilde{n}}| \quad (32)$$

$$\boldsymbol{T_7} = |\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{DR}_{\tilde{n}}| \quad (33)$$

$$\boldsymbol{T_8} = |\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{DP}_{\tilde{n}}| \quad (34)$$

$$\boldsymbol{T_9} = |\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{DP}_{\tilde{n}}| + Ö \quad (35)$$

$$\boldsymbol{T_{10}} = |\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2}) - \boldsymbol{DR}_{\tilde{n}}| + Ö \quad (36)$$

$$\boldsymbol{T_{11}} = |\boldsymbol{p}_{\tilde{n}}(\mathbf{2},\mathbf{2}) + \boldsymbol{DP}_{\tilde{n}}| - Ö \quad (37)$$

$$\boldsymbol{T_{12}} = |\boldsymbol{r}_{\tilde{n}}(\mathbf{2},\mathbf{2}) + \boldsymbol{DR}_{\tilde{n}}| - Ö \quad (38)$$

Subscript Ö is the weight for generation of tentative original values, where Ö is having a range from 1 to 10. These tentative original pixel values are then evaluated by computing similarity $\boldsymbol{SF_r}$ (similarity with respect to ROAD-TGM) with its neighbor pixels by using equation 39-40.

$$\boldsymbol{SF_r} = \text{SSIM}\ (š, \boldsymbol{T_n}) \quad (39)$$

Where

$$\text{SSIM} = \frac{(2A_xA_y+B1)(2C_{xy}+B2)}{(A_x^2+A_y^2+B1)(C_x^2+C_y^2+B2)} \quad (40)$$

To evaluate the proximity between two images SSIM is computed. The evaluation of SSIM computed between tentative original pixel values T_n and all the immediate

neighbor's noisy pixel location available in š to ensures the proximity of restored value to the original value. The subscript A_x and A_y are the mean of two different images x and y. Constant B1 and B2 are utilized to maintain stability. C_{xy} is covariance computed from image x and y. The subscript C_x^2 and C_y^2 represents the variance of images x and y respectively. The T_n having maximum similarity is used to restore the noisy pixel location in image â which provide enhanced restoration with detail preservation. The performance booster algorithm stage process is concluded in Algorithm 2.

Algorithm 2 Performance Booster Algorithm

Input: Superimposed noise map images Φ_r, Φ_p.

Output: The noise location map SF_r

1: Input selection: Noise map images Φ_r, Φ_p.

2: Patch extraction: segment the image into ñ patches by using 3×3 window to form š and ž.

3: Sequential thresholding: Do sequential thresholding on š and ž to eliminate noisy values; If condition 16-12 satisfy, consider value for computing tentative original values; otherwise, no change.

4: Estimation set 1: Do original pixel value estimation to get tentative original value set 1 according to (23-26).

5: For preliminaries of estimation set 2, do.

6: Determine D_{vr} according to (28);

7: Determine D_{vp} according to (28);

8: Determine $\boldsymbol{DP}_{\tilde{n}}$ according to (29);

9: Determine $\boldsymbol{DR}_{\tilde{n}}$ according to (30);

10: End for

11: Estimation set 2: Do original pixel value estimation to get tentative original value set 1 according to (31-38).

12: Determine SF_r according to (39,40);

13: Return SF_r

5.4 Experimental Results and Analysis

In this section, the effectiveness of the proposed algorithm is verified in two stages first by comparing it with MF, PSMF, CWMF, DBMF and ROAD-TGM. The proposed algorithm is tested for a vast data set of gray and color images. These images are affected by eight increasing steps of noise to ensure the robustness of the algorithm. Both color and grayscale images are affected with Salt & Pepper noise, Strip line artifact and Blotch artifact to form noisy data set. The Salt & Pepper noise is applied to images in the range of 10% to 80 % with equal eight steps to form the noisy dataset-1. To form the noisy dataset-2 for images affected with Strip lines, a set of vertical and horizontal Strip lines are formed on the image with a width of 2 pixels to 9 pixels (equally separated eight steps). Similarly, for noisy dataset-3, Blotch artifacts are applied on images with eight steps of square shape starting from 2×2 pixels wide shape to 9×9 pixels wide shape. The raw dataset used in this analysis contains 16 color images and 50 grayscale images. The images employed in this experimental analysis are of 256 × 256 size. The color images are from the University of South California miscellaneous dataset volume 3[67] and grayscale images from the Brodatz texture[68] dataset. Secondly, the proposed method is tested for grayscale Lena and Peppers traditional test image affected with 10% to 90% Salt & Pepper noise. Then the proposed method is

compared with the recent state-of-the-art IMF algorithm for impulse noise reduction. This comparison is evaluated by computing PSNR and SSIM. The traditional images employed in the second experimental analysis are of 512 × 512 size.

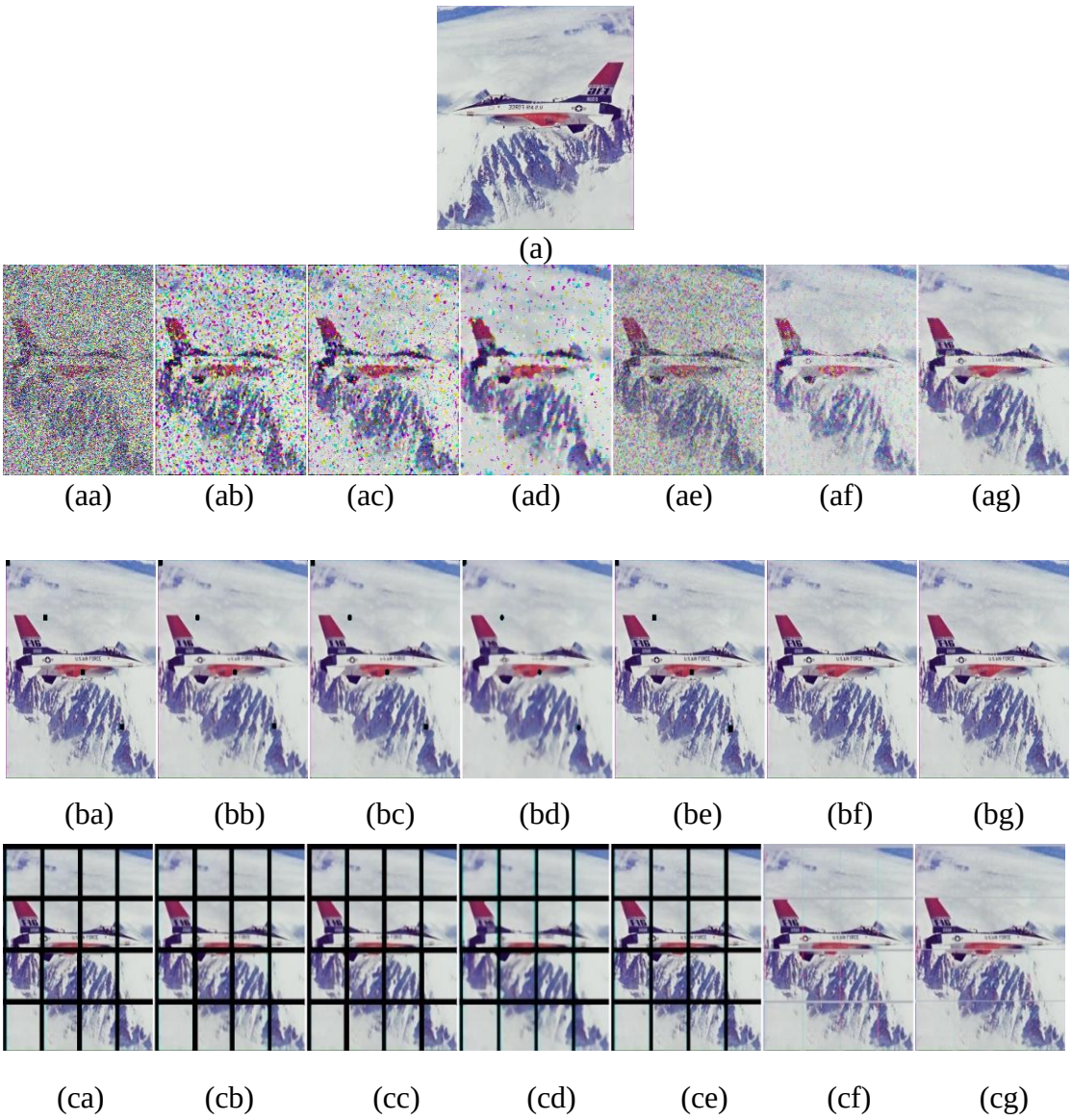
The experimental simulations are run with software MATLAB with 1.70 GHz Intel Core i5 with 8GB RAM. In this analysis, we set the threshold for high level of noise affected images Ø is set to 20. As discussed above weight variable is set as K=2 and weight for original tentative value is set to Ö = 5.

The denoising effect of the image denoising algorithm can be inspected by parameter PSNR. The PSNR is measured in decibels (dB) as it is computed by ratio of maximum possible power of a signal to the destructive noise power. High the value of PSNR indicates better performance and the low value of PSNR indicates poor performance for a denoising algorithm. The PSNR can be computed by using equation 41 and 42.

$$\text{PSNR}= 10\, log\, \frac{255^2}{MSE} \tag{41}$$

$$\text{MSE}= \frac{1}{I\times J}\sum_{x=1}^{I}\sum_{y=1}^{J}[a_d(x,y)-a_0(x,y)]^2 \tag{42}$$

Where MSE is mean square error and the original image is represented with subscript $a_0(x,y)$ with size I×J.



(a)

(aa) (ab) (ac) (ad) (ae) (af) (ag)

(ba) (bb) (bc) (bd) (be) (bf) (bg)

(ca) (cb) (cc) (cd) (ce) (cf) (cg)

Figure 5.2. (a) Original color image, (aa) Color image corrupted by 60% Salt & Pepper noise, (ba) Color image corrupted by 6*6 pixel Blotches artifacts, (ca) Color image corrupted by 6 pixel wide Strip line artifact,(ab–cb) Image restored using MF from (aa–ca), (ac–cc) Image restored using the PSMF from (aa–ca), (ac–cd) Image restored using the CWMF from (aa–ca), (ae–ce) Image restored using the DBMF from (aa–ca), (af–cf) Image restored using the ROAD-TGM from (aa–ca), (ag–cg) Image restored using the Proposed algorithm from (aa–ca). NB: MF, Median filter; PSMF, Progressive Switching Median Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; and ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter.

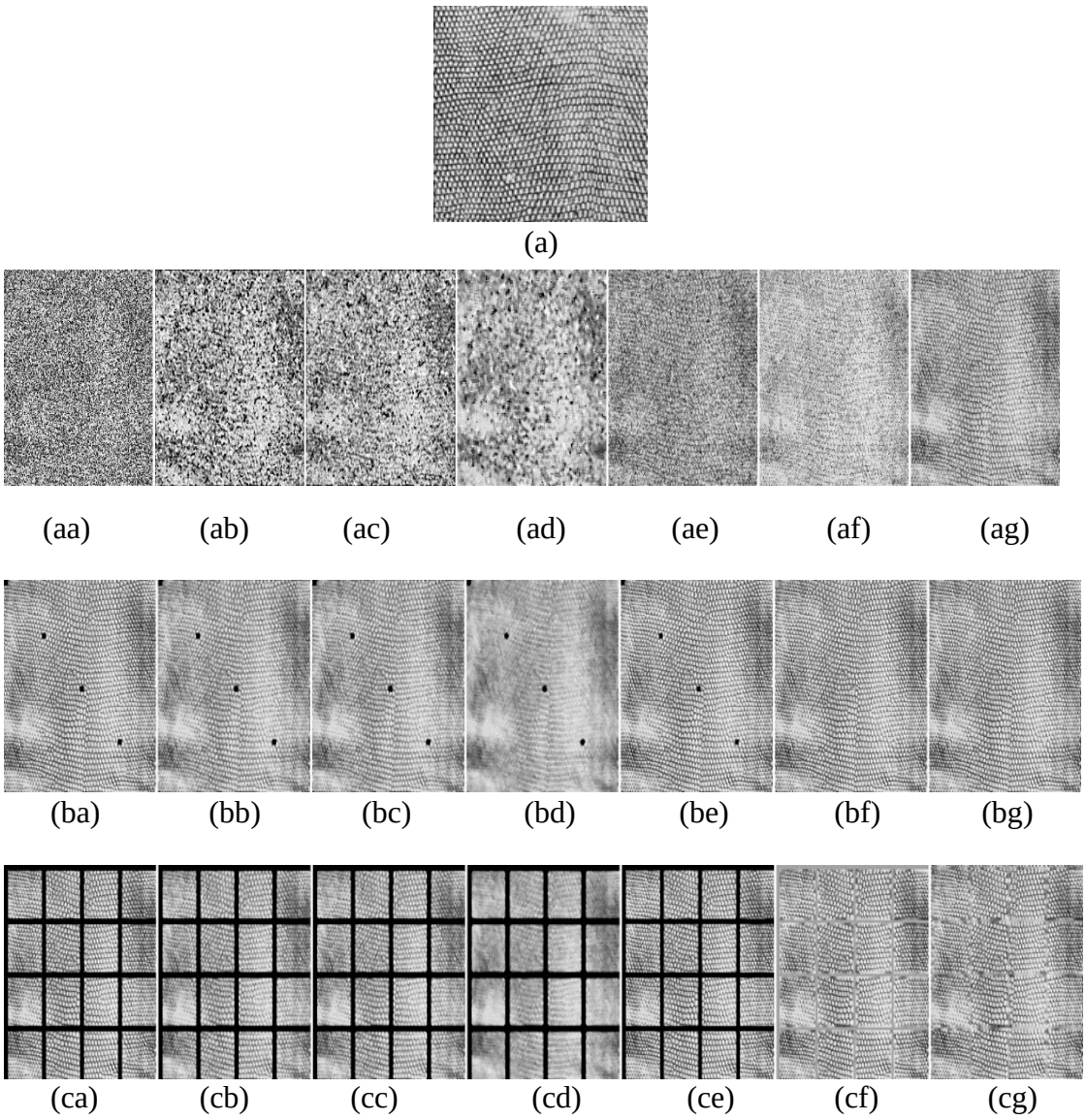


Figure 5.3. (a) Original texture image, (aa) Texture image corrupted by 60% Salt & Pepper noise, (ba) Texture image corrupted by 6*6 pixel Blotches artifacts, (ca) Texture image corrupted by 6 pixel wide Strip line artifact,(ab–cb) Image restored using MF from (aa–ca), (ac–dc) Image restored using the PSMF from (aa–ca), (ad–cd) Image restored using the CWMF from (aa–ca), (ae–ce) Image restored using the DBMF from (aa–ca), (af–cf) Image restored using the ROAD-TGM from (aa–ca), (ag–cg) Image restored using the Proposed algorithm from (aa–ca). NB: MF, Median filter; PSMF, Progressive Switching Median Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; and ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter.

The denoised image formed as a result of the denoising algorithm is represented as $a_d(x, y)$. To perceive the improvement of the denoised image the parameter IEF is computed using equation 45.

$$V_O = \frac{1}{I \times J} \sum_{x=1}^{I} \sum_{y=1}^{J} [a_n(x, y) - a_0(x, y)]^2 \quad (43)$$

$$V_D = \frac{1}{I \times J} \sum_{x=1}^{I} \sum_{y=1}^{J} [a_d(x, y) - a_0(x, y)]^2 \quad (44)$$

$$\text{IEF} = V_O / V_D \quad (45)$$

MSE is computed for calculation of IEF, as discussed earlier $a_0(x, y)$ is original image and $a_d(x, y)$ is a denoised image. The subscript $a_n(x, y)$ represents the noisy image. An increase in the value of IEF reflects the better performance of the denoising algorithm. The third parameter used to evaluate the performance of the proposed denoising algorithm based on the similarity of the denoised image with the original image. This parameter is SSIM and it is computed by using equation 40. The computed value of SSIM is having a range of zero to one. In this range value, one indicates perfect similarity which is only possible in the case of same images. To meet the objective detail preservation, evaluation of denoised image can be performed by analyzing the SSIM, the PSNR and IEF. For the first comparison (vast noise data set of color and grayscale image), a sample of pictorial results of color image and grayscale image denoising are presented for visual evaluation in Fig. 5.3 – Fig. 5.4, respectively. The proposed algorithm has less over-smoothing and better detail preservation than the algorithms in comparison. For Salt & Pepper noise ROAD-TGM reduces the noise to a considerable level with low blurring but it is still inferior to the proposed algorithm. For artifacts, proposed algorithm had achieved far better performance than other state-of-the-art algorithms in comparison. The proposed algorithm had achieved better noise estimation and accurate noise restoration. The result of PSNR, SSIM and IEF for color denoised images shown in Fig.5.3 are presented in Table 5.1, which shows the superiority of the proposed algorithm in terms of noise reduction and detail preservation. Similarly, for grayscale texture image (Fig. 5.4) results are shown in Table 5.2.

Table 5.1 Results of the PSNR, SSIM and IEF of the denoised color images (shown in Fig.5.3) with different types of algorithms.

Noise Type	Noise level	MF			PSMF			CWMF			DBMF			ROAD-TGM			**ROAD TGM-PPCA-PB**		
		PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	**PSNR**	**SSIM**	**IEF**
Salt & Pepper noise	10%	28.31	0.93	21.91	28.41	0.94	22.77	25.27	0.86	29.77	29.49	0.79	29.03	34.50	0.97	75.05	**39.07**	**0.98**	**90.93**
	20%	25.52	0.87	23.07	25.63	0.88	23.49	24.91	0.85	27.15	24.58	0.61	18.61	30.80	0.94	66.75	**35.51**	**0.96**	**77.81**
	30%	21.23	0.67	12.95	22.41	0.75	17.02	24.45	0.84	22.49	21.25	0.47	12.98	28.02	0.92	67.76	**34.26**	**0.94**	**72.22**
	40%	17.58	0.43	7.42	19.47	0.53	11.48	23.60	0.81	20.32	18.73	0.38	9.75	25.31	0.89	43.87	**34.12**	**0.92**	**69.19**
	50%	14.13	0.24	4.19	16.27	0.33	6.87	21.42	0.69	10.99	16.74	0.31	7.67	22.43	0.73	28.37	**33.83**	**0.90**	**63.89**
	60%	11.42	0.14	2.71	13.44	0.20	4.31	17.53	0.44	10.97	14.99	0.24	6.16	19.38	0.50	16.88	**33.30**	**0.85**	**59.43**
	70%	9.38	0.08	1.96	10.66	0.12	2.65	13.15	0.19	4.71	13.53	0.19	5.11	17.49	0.36	12.73	**32.58**	**0.83**	**50.34**
	80%	7.49	0.05	1.45	8.39	0.06	1.79	9.86	0.08	2.50	12.25	0.13	4.34	16.47	0.34	11.54	**31.90**	**0.80**	**41.07**
Blotchs Artifact	2x2	30.72	0.92	0.46	30.58	0.94	0.46	25.56	0.87	0.21	49.87	0.95	3.48	59.86	0.96	63.56	**64.82**	**0.97**	**77.84**
	3x3	28.50	0.90	0.41	29.51	0.93	0.40	24.54	0.85	0.17	44.27	0.93	2.16	58.00	0.95	63.45	**62.40**	**0.96**	**72.23**
	4x4	27.19	0.88	0.35	28.06	0.91	0.34	23.46	0.82	0.14	40.75	0.93	1.74	56.93	0.93	61.06	**60.57**	**0.95**	**66.63**
	5x5	26.82	0.85	0.29	27.70	0.90	0.28	22.34	0.80	0.11	38.21	0.91	1.53	54.25	0.93	56.09	**58.29**	**0.94**	**61.48**
	6x6	24.44	0.81	0.22	26.33	0.88	0.22	21.19	0.79	0.08	36.37	0.86	1.42	53.78	0.92	49.91	**55.22**	**0.93**	**50.96**
	7x7	23.03	0.79	0.15	25.93	0.87	0.15	20.04	0.77	0.05	34.89	0.84	1.35	51.83	0.91	47.47	**53.60**	**0.92**	**42.55**
	8x8	21.62	0.77	0.09	23.52	0.86	0.09	19.86	0.75	0.03	33.61	0.82	1.29	48.78	0.90	35.34	**51.41**	**0.91**	**36.64**
	9x9	20.19	0.75	0.04	22.11	0.85	0.04	17.68	0.74	0.01	32.52	0.80	1.26	47.16	0.89	32.07	**49.47**	**0.90**	**34.75**
Strip lines Artifact	2	16.60	0.78	0.99	16.59	0.78	0.99	21.19	0.82	2.74	19.64	0.84	1.92	36.64	0.96	95.99	**43.73**	**0.97**	**56.52**
	3	14.93	0.75	0.99	14.93	0.75	0.99	14.65	0.68	0.97	16.75	0.81	1.47	34.41	0.94	85.71	**42.05**	**0.96**	**48.85**
	4	13.74	0.73	0.99	13.73	0.73	0.99	13.52	0.66	0.96	15.03	0.78	1.31	31.92	0.91	64.19	**41.15**	**0.92**	**43.01**
	5	12.80	0.70	0.98	12.80	0.70	0.98	12.63	0.64	0.96	13.81	0.76	1.24	28.85	0.89	39.45	**40.52**	**0.91**	**39.86**
	6	12.03	0.68	0.98	12.03	0.68	0.98	11.89	0.61	0.95	12.86	0.73	1.19	27.11	0.85	31.67	**39.77**	**0.87**	**38.64**
	7	11.38	0.66	0.98	11.38	0.65	0.97	11.26	0.59	0.94	12.08	0.71	1.16	25.85	0.84	27.57	**39.64**	**0.86**	**37.18**
	8	10.82	0.63	0.97	10.82	0.63	0.97	10.72	0.57	0.93	11.42	0.68	1.13	24.91	0.80	25.30	**39.77**	**0.82**	**36.86**
	9	10.33	0.61	0.95	10.33	0.61	0.95	10.24	0.55	0.91	10.85	0.66	1.12	24.16	0.78	23.91	**39.88**	**0.80**	**34.18**

Table 5.2 Results of the PSNR, SSIM and IEF of the denoised grayscale texture images (shown in Fig.5.4) with different types of algorithms.

Noise Type	Noise level	MF			PSMF			CWMF			DBMF			ROAD-TGM			**ROAD TGM-PPCA-PB**		
		PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	PSNR	SSIM	IEF	**PSNR**	**SSIM**	**IEF**
Salt & Pepper noise	10%	17.32	0.78	2.85	17.17	0.77	2.48	13.74	0.35	2.46	22.78	0.95	6.16	24.78	0.97	9.44	**26.73**	**0.97**	**14.35**
	20%	16.13	0.73	2.64	15.95	0.71	2.38	13.48	0.34	2.36	19.52	0.89	5.65	21.06	0.92	7.90	**23.22**	**0.95**	**13.73**
	30%	14.80	0.66	2.54	14.82	0.65	2.35	13.14	0.32	2.33	16.94	0.81	4.67	18.71	0.86	6.99	**20.90**	**0.92**	**12.98**
	40%	13.24	0.57	2.25	13.43	0.57	2.04	12.70	0.30	2.02	14.90	0.71	4.00	16.82	0.78	6.10	**19.22**	**0.88**	**12.03**
	50%	11.57	0.46	1.83	12.11	0.48	1.96	11.96	0.26	1.94	13.26	0.60	3.94	15.44	0.69	5.54	**17.78**	**0.84**	**10.90**
	60%	9.84	0.33	1.68	10.46	0.36	1.58	10.94	0.20	1.57	12.28	0.48	3.69	14.13	0.59	4.88	**16.56**	**0.78**	**9.86**
	70%	8.49	0.24	1.55	9.04	0.25	1.39	9.63	0.15	1.38	12.25	0.39	3.34	13.03	0.47	4.66	**15.17**	**0.68**	**8.24**
	80%	7.07	0.13	1.29	7.61	0.15	0.74	7.90	0.08	0.74	11.98	0.30	3.21	12.68	0.33	4.45	**13.71**	**0.56**	**6.54**
Blotchs Artifact	2x2	18.32	0.61	0.10	19.42	1.71	0.10	16.32	0.36	0.04	44.06	0.94	1.69	47.81	0.96	4.68	**45.29**	**0.97**	**5.72**
	3x3	17.30	0.58	0.08	18.40	1.68	0.08	15.30	0.36	0.03	40.74	0.91	1.58	42.87	0.92	4.02	**43.10**	**0.94**	**5.00**
	4x4	16.27	0.53	0.07	17.37	1.63	0.06	14.27	0.36	0.03	38.10	0.88	1.58	39.76	0.88	3.89	**40.42**	**0.90**	**4.83**
	5x5	15.20	0.52	0.05	16.30	1.62	0.05	13.20	0.36	0.02	35.00	0.84	1.57	38.94	0.85	3.75	**39.72**	**0.87**	**4.62**
	6x6	13.12	0.48	0.03	14.22	1.58	0.03	11.12	0.36	0.01	32.52	0.79	1.38	37.83	0.82	3.65	**38.53**	**0.83**	**3.67**
	7x7	11.04	0.47	0.02	12.14	1.57	0.02	9.04	0.36	0.01	30.85	0.71	1.26	35.46	0.77	3.11	**36.75**	**0.80**	**2.88**
	8x8	10.97	0.42	0.01	12.07	1.52	0.01	8.97	0.36	0.00	29.63	0.66	1.23	33.72	0.73	2.58	**34.31**	**0.76**	**2.60**
	9x9	9.88	0.40	0.00	10.98	1.50	0.00	7.88	0.36	0.00	28.64	0.61	1.21	33.78	0.71	2.48	**34.78**	**0.75**	**1.88**
Strip lines Artifact	2	14.03	0.72	0.89	13.98	0.71	0.89	12.66	0.33	0.75	18.82	0.92	1.81	23.67	0.96	5.53	**25.71**	**0.97**	**10.23**
	3	12.95	0.69	0.88	12.91	0.68	0.87	11.13	0.30	0.72	16.02	0.88	1.42	21.46	0.93	5.45	**23.14**	**0.95**	**8.84**
	4	12.07	0.66	0.86	12.04	0.66	0.85	10.58	0.29	0.69	14.35	0.85	1.28	19.84	0.90	5.27	**21.68**	**0.94**	**8.51**
	5	11.37	0.64	0.83	11.34	0.63	0.83	10.11	0.28	0.65	13.13	0.82	1.21	19.06	0.87	5.11	**20.21**	**0.91**	**7.45**
	6	10.79	0.62	0.80	10.77	0.61	0.80	9.71	0.27	0.60	12.24	0.79	1.16	18.57	0.84	5.00	**19.28**	**0.89**	**7.30**
	7	10.29	0.60	0.76	10.26	0.59	0.75	9.34	0.26	0.54	11.51	0.76	1.14	18.05	0.81	4.96	**19.68**	**0.90**	**6.95**
	8	9.83	0.58	0.70	9.82	0.57	0.69	9.00	0.25	0.46	10.90	0.74	1.12	17.64	0.79	4.72	**19.71**	**0.90**	**6.15**
	9	9.43	0.55	0.60	9.42	0.55	0.59	8.69	0.24	0.44	10.37	0.71	1.10	17.30	0.76	4.54	**20.04**	**0.91**	**5.89**

Table 5.3 The mean value results of PSNR, SSIM and IEF of the denoised color and grayscale dataset with ROAD TGM-PPCA-PB and existing algorithms.

Image type	Noise type	Parameter	ROAD TGM-PPCA-PB	Road-TGM	DBMF	CWMF	PSMF	MF
Grayscale dataset 50 images	Salt and Pepper (10%-80%)	Mean PSNR	**25.489**	19.825	18.002	15.966	15.456	15.181
		MeanSSIM	**0.856**	0.640	0.551	0.383	0.444	0.419
		Mean IEF	**63.009**	17.960	9.158	7.007	5.736	4.818
	Blotchs (2x2 - 9x9)	Mean PSNR	**49.611**	44.214	36.299	19.442	22.636	22.713
		MeanSSIM	**0.999**	0.998	0.997	0.525	0.765	0.770
		Mean IEF	**22.889**	14.447	1.643	0.068	0.128	0.130
	Strip lines (2 - 9)	Mean PSNR	**26.732**	23.723	15.533	13.235	13.706	13.719
		MeanSSIM	**0.973**	0.875	0.784	0.394	0.578	0.582
		Mean IEF	**23.197**	10.688	1.278	0.781	0.841	0.844
Color image dataset 16 images	Salt and Pepper (10%-80%)	Mean PSNR	**39.282**	23.700	19.170	21.323	18.733	17.403
		MeanSSIM	**0.933**	0.756	0.531	0.694	0.574	0.513
		Mean IEF	**76.567**	50.175	11.958	26.639	14.837	12.223
	Blotchs (2x2 - 9x9)	Mean PSNR	**61.210**	44.064	39.672	27.423	30.588	30.635
		MeanSSIM	**0.996**	0.994	0.994	0.916	0.958	0.959
		Mean IEF	**27.179**	15.670	1.533	0.164	0.292	0.294
	Strip lines (2 - 9)	Mean PSNR	**42.568**	27.876	16.300	15.402	15.048	15.051
		MeanSSIM	**0.970**	0.939	0.788	0.711	0.734	0.735
		Mean IEF	**53.531**	25.358	1.301	1.138	0.961	0.962

Table 5.4 Results of the PSNR, SSIM of the denoised grayscale traditional images with proposed algorithms and state of art algorithms

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	FDS [65]	PSNR	41.40	37.25	34.49	31.67	28.99	26.54	23.95	21.39	18.30	29.33
		SSIM	0.9894	0.9759	0.9573	0.9293	0.8858	0.8280	0.7441	0.6379	0.5020	0.83
	DAMF [61]	PSNR	42.97	39.29	36.84	34.94	33.21	31.64	30.22	28.53	25.93	33.73
		SSIM	0.9902	0.9788	0.9655	0.9494	0.9304	0.9064	0.8770	0.8370	0.7620	0.91
	IIN [66]	PSNR	—	31.43	29.50	27.62	26.39	—	—	—	—	28.74
		SSIM	—	—	—	—	—	—	—	—	—	—
	IMF [62]	PSNR	43.48	40.18	37.05	35.40	33.98	32.49	31.23	29.70	27.42	34.55
		SSIM	0.9913	0.9796	0.9675	0.9541	0.9383	0.9183	0.8953	0.8623	0.8058	0.92
	Hybrid Method	PSNR	44.26	41.46	38.96	36.88	34.59	33.23	32.28	30.47	28.22	35.59
		SSIM	0.9927	0.9813	0.9736	0.9617	0.9473	0.9369	0.9155	0.8797	0.8423	0.94
	SAID-END Method	PSNR	45.21	42.35	39.80	37.67	35.33	33.94	32.97	31.12	28.83	36.36
		SSIM	0.9957	0.9842	0.9765	0.9646	0.9501	0.9397	0.9182	0.8823	0.8448	0.94
	ROAD TGM-PPCA-PB	PSNR	46.11	43.20	40.60	38.42	36.04	34.62	33.63	31.74	29.41	37.08
		SSIM	0.9967	0.9852	0.9863	0.9742	0.9596	0.9491	0.9274	0.8911	0.8532	0.95
	SAID END-PPCA-PB	**PSNR**	**49.34**	**46.22**	**43.44**	**41.11**	**38.56**	**37.04**	**35.98**	**33.96**	**31.47**	**39.68**
		SSIM	**0.9999**	**0.9883**	**0.9981**	**0.9859**	**0.9711**	**0.9605**	**0.9385**	**0.9018**	**0.8635**	**0.96**
Peppers	FDS [65]	PSNR	40.65	36.9	34.32	31.72	29.32	26.83	24.11	21.37	18.15	29.26
		SSIM	0.9825	0.9627	0.9396	0.9079	0.8687	0.8110	0.7355	0.6360	0.5085	0.82
	DAMF [61]	PSNR	41.52	37.89	35.67	33.95	32.55	31.31	29.79	28.28	25.87	32.98
		SSIM	0.9815	0.9606	0.9389	0.9131	0.8866	0.8541	0.8180	0.7719	0.7049	0.87
	IIN [66]	PSNR	—	27.23	26.21	24.82	23.98	—	—	—	—	25.56
		SSIM	—	—	—	—	—	—	—	—	—	—
	IMF [62]	PSNR	41.83	38.59	36.65	35.14	33.9	32.69	31.43	30.01	27.88	34.24
		SSIM	0.9858	0.9684	0.9504	0.9299	0.9091	0.8846	0.8572	0.8219	0.7700	0.90
	Hybrid Method	PSNR	42.50	39.29	37.33	35.99	34.34	33.13	32.22	30.53	29.20	34.95
		SSIM	0.9893	0.9834	0.9645	0.9405	0.9229	0.8982	0.8773	0.8496	0.8163	0.9158
	SAID-END Method	PSNR	43.41	40.13	38.13	36.76	35.08	33.84	32.91	31.18	29.83	35.70
		SSIM	0.9922	0.9863	0.9674	0.9433	0.9256	0.9009	0.8799	0.8521	0.8187	0.92
	ROADTGM - PPCA-PB	PSNR	44.28	40.93	38.89	37.50	35.78	34.52	33.57	31.80	30.43	36.41
		SSIM	0.9932	0.9873	0.9771	0.9527	0.9349	0.9099	0.8887	0.8606	0.8269	0.9257
	SAID END-PPCA-PB	**PSNR**	**48.26**	**44.62**	**42.39**	**40.87**	**39.00**	**37.62**	**36.59**	**34.67**	**33.16**	**39.69**
		SSIM	**0.9992**	**0.9971**	**0.9956**	**0.9708**	**0.9526**	**0.9272**	**0.9056**	**0.8770**	**0.8426**	**0.9409**

Table 5.5 Results of the PSNR, SSIM of the denoised color traditional images with proposed algorithms and state of art algorithms

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	**Mean**
Lena	AUTSF [63]	PSNR	41.51	38.12	35.91	34.20	32.60	31.05	29.22	27.38	24.29	32.70
		SSIM	–	–	–	–	–	–	–	–	–	–
	MCF [64]	PSNR	–	–	38.95	36.55	34.20	31.42	26.41	21.34	8.46	28.19
		SSIM	–	–	–	–	–	–	–	–	–	–
	Hybrid Method	PSNR	43.67	40.84	38.96	36.55	34.22	31.45	30.52	28.75	26.57	34.61
		SSIM	0.9658	0.9545	0.9470	0.9353	0.9211	0.9109	0.8898	0.8547	0.8179	0.9108
	SAID-END Method	PSNR	44.20	41.34	38.99	36.66	34.32	32.93	31.96	30.11	27.82	35.35
		SSIM	0.9855	0.9740	0.9663	0.9544	0.9399	0.9295	0.9080	0.8721	0.8346	0.9294
	ROAD TGM-PPCA-PB	PSNR	45.08	42.17	39.77	37.39	35.01	33.59	32.60	30.71	28.38	36.08
		SSIM	0.9934	0.9818	0.9740	0.9620	0.9474	0.9369	0.9153	0.8791	0.8413	0.9368
	SAID END-PPCA-PB	**PSNR**	**47.79**	**44.70**	**42.15**	**39.64**	**37.11**	**35.60**	**34.55**	**32.55**	**30.08**	**38.24**
		SSIM	**0.9993**	**0.9906**	**0.9828**	**0.9707**	**0.9559**	**0.9454**	**0.9235**	**0.8870**	**0.8488**	**0.9449**

Table 5.6 The mean value results of PSNR, SSIM and IEF of the denoised color and grayscale dataset with SAID END-PPCA-PB and existing algorithms.

Image type	**Noise type**	**Parameter**	**SAID END-PPCA-PB**	**Road-TGM**	**DBMF**	**CWMF**	**PSMF**	**MF**
Grayscale dataset 50 images	Salt and Pepper (10%-80%)	Mean PSNR	**31.861**	19.825	18.002	15.966	15.456	15.181
		MeanSSIM	**0.890**	0.640	0.551	0.383	0.444	0.419
		Mean IEF	**72.461**	17.960	9.158	7.007	5.736	4.818
	Blotchs (2x2 - 9x9)	Mean PSNR	**53.580**	44.214	36.299	19.442	22.636	22.713
		MeanSSIM	**1.000**	0.998	0.997	0.525	0.765	0.770
		Mean IEF	**29.756**	14.447	1.643	0.068	0.128	0.130
	Strip lines (2 - 9)	Mean PSNR	**29.138**	23.723	15.533	13.235	13.706	13.719
		MeanSSIM	**0.982**	0.875	0.784	0.394	0.578	0.582
		Mean IEF	**27.836**	10.688	1.278	0.781	0.841	0.844
Color image dataset 16 images	Salt and Pepper (10%-80%)	Mean PSNR	**43.996**	23.700	19.170	21.323	18.733	17.403
		MeanSSIM	**0.963**	0.756	0.531	0.694	0.574	0.513
		Mean IEF	**80.309**	50.175	11.958	26.639	14.837	12.223
	Blotchs (2x2 - 9x9)	Mean PSNR	**63.658**	44.064	39.672	27.423	30.588	30.635
		MeanSSIM	**0.997**	0.994	0.994	0.916	0.958	0.959
		Mean IEF	**29.625**	15.670	1.533	0.164	0.292	0.294
	Strip lines (2 - 9)	Mean PSNR	**46.399**	27.876	16.300	15.402	15.048	15.051
		MeanSSIM	**0.986**	0.939	0.788	0.711	0.734	0.735
		Mean IEF	**57.814**	25.358	1.301	1.138	0.961	0.962

Table 5.7 Comparison of SAID END-PPCA-PB with all proposed algorithms and existing algorithms

Image type	Noise type	Parameter	**SAID END-PPCA-PB**	ROAD TGM-PPCA-PB	Proposed Adaptive	Proposed Hybrid	Road-TGM	DBMF	CWMF	PSMF	MF
Grayscale dataset 50 images	Salt and Pepper (10%-80%)	Mean PSNR	**31.861**	25.489	23.953	22.568	19.825	18.002	15.966	15.456	15.181
		MeanSSIM	**0.890**	0.856	0.834	0.686	0.640	0.551	0.383	0.444	0.419
		Mean IEF	**72.461**	63.009	47.125	44.769	17.960	9.158	7.007	5.736	4.818
	Blotchs (2x2 - 9x9)	Mean PSNR	**53.580**	49.611	45.515	45.060	44.214	36.299	19.442	22.636	22.713
		MeanSSIM	**1.000**	0.999	0.999	0.999	0.998	0.997	0.525	0.765	0.770
		Mean IEF	**29.756**	22.889	20.809	19.768	14.447	1.643	0.068	0.128	0.130
	Strip lines (2 - 9)	Mean PSNR	**29.138**	26.732	25.739	24.464	23.723	15.533	13.235	13.706	13.719
		MeanSSIM	**0.982**	0.973	0.926	0.910	0.875	0.784	0.394	0.578	0.582
		Mean IEF	**27.836**	23.197	17.714	16.829	10.688	1.278	0.781	0.841	0.844
Color image dataset 16 images	Salt and Pepper (10%-80%)	Mean PSNR	**43.996**	39.282	36.372	25.839	23.700	19.170	21.323	18.733	17.403
		MeanSSIM	**0.963**	0.933	0.924	0.767	0.756	0.531	0.694	0.574	0.513
		Mean IEF	**80.309**	76.567	73.678	72.233	50.175	11.958	26.639	14.837	12.223
	Blotchs (2x2 - 9x9)	Mean PSNR	**63.658**	61.210	60.010	58.833	44.064	39.672	27.423	30.588	30.635
		MeanSSIM	**0.997**	0.996	0.996	0.996	0.994	0.994	0.916	0.958	0.959
		Mean IEF	**29.625**	27.179	26.646	26.124	15.670	1.533	0.164	0.292	0.294
	Strip lines (2 - 9)	Mean PSNR	**46.399**	42.568	41.733	40.915	27.876	16.300	15.402	15.048	15.051
		MeanSSIM	**0.986**	0.970	0.965	0.960	0.939	0.788	0.711	0.734	0.735
		Mean IEF	**57.814**	53.531	52.481	51.452	25.358	1.301	1.138	0.961	0.962

To ensure the robustness of algorithm evaluation was conducted on wide data set of 16 color images and 50 grayscales texture dataset, the results of the test are repeating the success of the proposed algorithm on wide data set as presented in Table 5.3. Table 5.3 contains the mean PSNR, SSIM and IEF parameters value of denoised color and grayscale image dataset with respect to noise type and noise level.

Similarly, a better performing enhanced algorithm can further be produced by replacing the ROAD-TGM with SAID-END algorithm (as already discussed in Chapter 4) in the design process. The results of the second comparison of the proposed algorithms with state-of-the-art algorithms for impulse noise reduction from the grayscale and color image are presented in Table 5.4 - Table 5.5 respectively. The proposed algorithms has outperformed the other algorithms in both the parameters SSIM and PSNR for images affected with salt & noise ranging from 10% to 90%. Performance comparison on large dataset for amalgamation of SAID-END with Progressive PCA using performance booster method is presented in Table 5.6. To

validate superior performance of SAID END-PPCA-PB the final comparison of SAID END-PPCA-PB with algorithms proposed in Chapter 3 to Chapter 5 and existing algorithm Table 5.7 is presented.

5.5 Conclusion

In this chapter, we have proposed ROAD END-PPCA-PB and SAID END-PPCA-PB which are not only the enhanced noise detection algorithm but also having accurate performance booster method. In these proposed algorithms the best features of both spatial domain and transform domain are integrated to achieve the high performance. Each noise location is verified by the amalgamation of ROAD-TGM or SAID END and progressive PCA to enhance the detection stage performance. To ensure the accurate restoration of the noisy pixel three-stage algorithm is designed using sequential hard thresholding to eliminate chances of noise involvement in the calculation. Followed by a tentative original value estimation stage to decrease the difference of estimated value and original value. Finally, restoration is achieved by similarity mapping which ensures only the most suitable value among tentative original values will be used to restore the noisy pixel location. The proposed algorithms work well on both color and grayscale image data set. Our future work will be focus on the improvement of the proposed method as well as its applications in the field of computer vision and visual tracking.

Chapter 6

Conclusions and Future Scope

In this chapter, the conclusions of the thesis work based on the analytical study carried out in the field of color and grayscale image denoising have been presented. The prime objective of the study was to evaluate the performance of these image denoising algorithms and identify the existing research gaps. On the basis of these gaps, new adaptive image denoising algorithm have been presented in this work.

6.1 Conclusions

In Chapter 1, a brief introduction to image denoising algorithms for the grayscale and color image dataset have been presented. The applications and advantages of these algorithms have been also discussed. Further, this chapter also gives the importance of image denoising day today life and, also the need for improvement in the image denoising algorithms. Based on this, an exhaustive literature survey of the various methods employed by the researchers to achieve effective image denoising for grayscale and color image dataset has been presented in Chapter 2.

In Chapter 3, the few techniques studied in literature have been analyzed and simulated on wide dataset of standard color and grayscale images. Also, the design of new hybrid image denoising model has been carried out and simulated for validation of results. The main focus was to understand the basic design process of image denoising algorithm for grayscale and color image dataset.

A new adaptive image denoising algorithm with adaptive detection and restoration stage have been presented in Chapter 4. This design has been thoroughly analyzed using PSNR, SSIM and IEF parameters. The adaptive algorithm achieves its high-quality image denoising performance by optimally maintaining tradeoff between quality and processing time. Lastly, Chapter 5 gives another two enhanced image

denoising algorithms to increase proximity of proposed algorithm with the original image. These algorithms use progressive principle component analysis along with performance booster algorithm. The basic design process of this algorithm is inspired by the algorithm presented in an earlier chapter.

The main contributions of this thesis work are summarized as:

- Comparative analysis of image denoising algorithm for grayscale and color images.
- Design and simulation of hybrid image denoising algorithm for grayscale and color image denoising.
- Design and simulation of spatially adaptive image denoising algorithm using enhanced noise detection method for grayscale and color images.
- Simulation of algorithm designed by amalgamation of ROAD-TGM and progressive PCA using performance booster method for detail preserving image denoising.
- Simulation of algorithm designed by amalgamation of SAID-END and progressive PCA using performance booster method for detail preserving image denoising.

The proposed hybrid method, SAID-END method, ROAD-PPCA-PB and SAID-END-PPCA-PB has respectively achieved 3.01% (PSNR=35.59), 5.24% (PSNR=36.36), 7.32% (PSNR=37.08) and 14.85% (PSNR=39.68) PSNR improvement over best performing recent state of the art IMF algorithm value (PSNR=34.55) for grayscale standard Lena test image. For the standard grayscale test image of Peppers, the respective algorithms have achieved 2.07% (PSNR=34.95), 4.26% (PSNR=35.70), 6.34% (PSNR=36.41) and 15.92% (PSNR=39.69) PSNR improvement over the recent state of the art IMF algorithm value (PSNR=34.24). Similarly, these algorithms have achieved 5.84% (PSNR=34.84), 8.10% (PSNR=35.35), 10.34% (PSNR=36.08) and 16.94% (PSNR=38.24) PSNR improvement respectively over best performing recent state of the art AUTSF algorithm value (PSNR=32.70) for color standard Lena test image.

6.2 Future Works

As discussed above, the proposed work is focused on the design and simulation of adaptive image denoising algorithm for grayscale and color image dataset. The work has been carried out extensively based on the research gaps identified in the literature. However, there are still a few dimensions of this work that need to be touched in the future. Hence, here are a few issues that can be addressed in future work:

- Design and performance evaluation of image denoising algorithm for biomedical applications.
- Design and development of image denoising algorithm with real time applications.
- Design and performance evaluation of performance booster algorithm with artificial intelligence.
- Design and performance evaluation of image denoising algorithm for video applications.
- Design and performance evaluation of image denoising algorithm for remote sensing applications.

List of Publications

Journal

[1] Amandeep Singh, Graurav Sethi and G.S Kalra, “Spatially Adaptive Image Denoising via Enhanced Noise Detection Method for Grayscale and Color Images," in IEEE Access, vol. 8, pp. 112985-113002, 2020, doi: 10.1109/ACCESS.2020.3003874.

[2] Amandeep Singh, Graurav Sethi and G.S Kalra, “Comparative Analysis of Color Image Denoising Techniques,” International Journal of Engineering Research and Technology, vol.13, pp. 2825-2831,2020.

[3] Amandeep Singh, Graurav Sethi and G.S Kalra, “Hybrid denoising algorithm for both grayscale and color images,” *Multimed Tools Appl (Springer)* ***(Under Review)***.

[4] Amandeep Singh, Graurav Sethi and G.S Kalra, “Amalgamation of ROAD-TGM and Progressive PCA Using Performance Booster Method for Detail Preservation Image Denoising,” ***(communicated)***

International Conference

[1] Amandeep Singh, Graurav Sethi and G.S Kalra, “Qualitative Performance Analysis of Gray Scale Image Denoising Techniques,” in *Proceedings – 3rd International Conference on Intelligent Circuits and Systems, ICICS 2020*, 2020. ***(Accepted)***

[2] Amandeep Singh, Graurav Sethi and G.S Kalra, “Statistical Evaluation of Color Image Denoising Techniques,” in *Proceedings – 3rd International Conference on Intelligent Circuits and Systems, ICICS 2020*, 2020. ***(Accepted)***

Bibliography

[1] N. K. Chaitanya and P. Sreenivasulu, “Removal of salt and pepper noise using Advanced Modified Decision based Unsymmetric Trimmed Median Filter,” in *2014 International Conference on Electronics and Communication Systems, ICECS 2014*, 2014.

[2] O. S. Faragallah and H. M. Ibrahem, “Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise,” *AEU - Int. J. Electron. Commun.*, vol. 70, no. 8, pp. 1034–1040, 2016.

[3] Z. Sun, B. Han, J. Li, J. Zhang, and X. Gao, “Weighted Guided Image Filtering with Steering Kernel,” *IEEE Trans. Image Process.*, vol. 29, pp. 500–508, 2020.

[4] M. Li and Y. Xu, “Improved non-local means algorithm for image denosing,” in *2019 IEEE 11th International Conference on Communication Software and Networks, ICCSN 2019*, 2019, pp. 358–361.

[5] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, “Weighted guided image filtering,” *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 120–129, Jan. 2015.

[6] Kamarujjaman, M. Mukherjee, and M. Maitra, “A new decision-based adaptive filter for removal of high density impulse noise from digital images,” in *2014 International Conference on Devices, Circuits and Communications, ICDCCom 2014 - Proceedings*, 2014.

[7] P. S. Windyga, “Fast impulsive noise removal,” *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 173–179, 2001.

[8] G. S. Kalra and S. Singh, “Efficient digital image denoising for gray scale images,” *Multimed. Tools Appl.*, vol. 75, no. 8, pp. 4467–4484, 2016.

[9] M. Nasri, S. Saryazdi, and H. Nezamabadi-Pour, “SNLM: A switching non-local means filter for removal of high density salt and pepper noise,” *Sci. Iran.*, vol. 20, no. 3, pp. 760–764, 2013.

[10] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[11] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, “Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter,” *IEEE Signal Process. Lett.*, vol.

18, no. 5, pp. 287–290, 2011.

[12] M. S. Nair and G. Raju, “A new fuzzy-based decision algorithm for high-density impulse noise removal,” *Signal, Image Video Process.*, vol. 6, no. 4, pp. 579–595, 2012.

[13] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, “Non-linear Convolution Filters for CNN-based Learning,” *Proc. IEEE Int. Conf. Comput. Vis.*, Aug. 2017.

[14] E. Santiago-Raḿırez, J. . A. Gonźalez-Fraga, and S. Ĺazaro-Marťınez, “Face recognition and tracking using unconstrained non-linear correlation filters,” *Procedia Eng.*, vol. 35, pp. 192–201, 2012.

[15] K. Aiswarya, V. Jayaraj, and D. Ebenezer, “A new and efficient algorithm for the removal of high density salt and pepper noise in images and videos,” in *ICCMS 2010 - 2010 International Conference on Computer Modeling and Simulation*, 2010, vol. 4, pp. 409–413.

[16] G. George, R. M. Oommen, S. Shelly, S. S. Philipose, and A. M. Varghese, “A Survey on Various Median Filtering Techniques For Removal of Impulse Noise From Digital Image,” in *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, 2018, pp. 235–238.

[17] R. H. Chan, C. Hu, and M. Nikolova, “An iterative procedure for removing random-valued impulse noise,” in *IEEE Signal Processing Letters*, 2004, vol. 11, no. 12, pp. 921–924.

[18] J. S. Silva, A. Silva, and B. S. Santos, “Image denoising methods for tumor discrimination in high-resolution computed tomography,” *J. Digit. Imaging*, vol. 24, no. 3, pp. 464–469, 2011.

[19] S. Manikandan and D. Ebenezer, “A nonlinear decision-based algorithm for removal of strip lines, drop lines, blotches, and missing and impulses in images and videos,” *Eurasip J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.

[20] P. J. Bex and W. Makous, “Spatial frequency, phase, and the contrast of natural images,” *J. Opt. Soc. Am. A*, 2002.

[21] C. F. Westin, H. Knutsson, and R. Kikinis, *Adaptive image filtering*. 2009.

[22] X. Kang, S. Li, and J. A. Benediktsson, “Spectral–Spatial Hyperspectral Image Classification With Edge-Preserving Filtering,” *IEEE Trans. Geosci. Remote*

Sens., vol. 52, no. 5, pp. 2666–2677, May 2014.

[23] N. Pierazzo, M. Lebrun, M. E. Rais, J. M. Morel, and G. Facciolo, "Non-local dual image denoising," in *2014 IEEE International Conference on Image Processing, ICIP 2014*, 2014, pp. 813–817.

[24] L. Şendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Trans. Signal Process.*, 2002.

[25] S. Kaur and N. Singh, "Image Denoising Techniques: A Review," *Int. J. Innov. Res. Comput. Commun. Eng.*, 2014.

[26] D. R. Roberts *et al.*, "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," *Ecography (Cop.).*, vol. 40, no. 8, pp. 913–929, Aug. 2017.

[27] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[28] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[29] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikainen, "Median Robust Extended Local Binary Pattern for Texture Classification," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1368–1381, Mar. 2016.

[30] J. Wang, Y. Chen, Y. Wu, J. Shi, and J. Gee, "Enhanced generative adversarial network for 3D brain MRI super-resolution," in *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, 2020.

[31] N. H. Harun, J. A. Bakar, Z. A. Wahab, M. K. Osman, and H. Harun, "Color Image Enhancement of Acute Leukemia Cells in Blood Microscopic Image for Leukemia Detection Sample," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2020, pp. 24–29.

[32] N. A. Binti Mohd Sharif *et al.*, "Performance of Image Enhancement Methods for Diabetic Retinopathy based on Retinal Fundus Image," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2020, pp. 18–23.

[33] S. Aja-Fernandez, C. Alberola-Lopez, and C.-F. Westin, “Noise and Signal Estimation in Magnitude MRI and Rician Distributed Images: A LMMSE Approach,” *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1383–1398, Aug. 2008.

[34] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, “Mean-square performance of a convex combination of two adaptive filters,” *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[35] A. Ignatov *et al.*, “NTIRE 2019 Challenge on Image Enhancement: Methods and Results,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2224–2232.

[36] J. Liu *et al.*, “Learning Raw Image Denoising With Bayer Pattern Unification and Bayer Preserving Augmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2070–2077.

[37] C. Huang, H. Liu, T. Chen, Q. Shen, and Z. Ma, “Extreme Image Coding via Multiscale Autoencoders with Generative Adversarial Optimization,” in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019, pp. 1–4.

[38] X. Zhang, Y. Lu, J. Liu, and B. Dong, “Dynamically Unfolding Recurrent Restorer: A Moving Endpoint Control Method for Image Restoration,” *7th Int. Conf. Learn. Represent. ICLR 2019*, May 2018.

[39] T. Veerakumar, B. N. Subudhi, S. Esakkirajan, and P. K. Pradhan, “Context model based edge preservation filter for impulse noise removal,” *Expert Syst. Appl.*, 2017.

[40] I. Irum, M. Sharif, M. Raza, and S. Mohsin, “A nonlinear hybrid filter for salt & Pepper noise removal from color images,” *J. Appl. Res. Technol.*, vol. 13, no. 1, pp. 79–85, 2015.

[41] R. Klette, *Concise Computer Vision*. London: Springer London, 2014.

[42] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner, “Detection of Missing Data in Image Sequences,” *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1496–1508, 1995.

[43] R. Chandel and G. Gupta, “Image Filtering Algorithms and Techniques: A Review,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 10, p. 2277, 2013.

[44] E. B. MacDougall, “Spatial Filtering,” *Econ. Geogr.*, vol. 46, p. 425, 1970.

[45] T. Chen, K. K. Ma, and L. H. Chen, “Tri-state median filter for image denoising,” *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1834–1838, 1999.

[46] C. C. Chang, J. Y. Hsiao, and C. P. Hsieh, “An adaptive median filter for image denoising,” in *Proceedings - 2008 2nd International Symposium on Intelligent Information Technology Application, IITA 2008*, 2008, vol. 2, pp. 346–350.

[47] T. Sun and Y. Neuvo, “Detail-preserving median based filters in image processing,” *Pattern Recognit. Lett.*, vol. 15, no. 4, pp. 341–347, 1994.

[48] T. Chen, “Space variant median filters for the restoration of impulse noise corrupted images,” *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 48, no. 8, pp. 784–789, 2001.

[49] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images,” *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 1012–1025, 1996.

[50] F. Russo and G. Ramponi, “A fuzzy filter for images corrupted by impulse noise,” *IEEE Signal Process. Lett.*, vol. 3, no. 6, pp. 168–170, 1996.

[51] R. Dosselmann and X. D. Yang, “A comprehensive assessment of the structural similarity index,” *Signal, Image Video Process.*, vol. 5, no. 1, pp. 81–91, 2011.

[52] B. Smolka, “On the new robust algorithm of noise reduction in color images,” *Comput. Graph.*, vol. 27, no. 4, pp. 503–513, 2003.

[53] U.-K. Cho, J.-H. Hong, and S.-B. Cho, “Evolutionary Image Enhancement for Impulsive Noise Reduction,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, pp. 678–683.

[54] A. S. Awad, “Standard deviation for obtaining the optimal direction in the removal of impulse noise,” *IEEE Signal Process. Lett.*, vol. 18, no. 7, pp. 407–410, 2011.

[55] B. N. Aravind and K. V. Suresh, “An improved image denoising using wavelet transform,” *Int. Conf. Trends Autom. Commun. Comput. Technol. I-TACT 2015*, 2016.

[56] Z. Wang and D. Zhang, “Progressive switching median filter for the removal of impulse noise from highly corrupted images,” *IEEE Trans. Circuits Syst. II*

Analog Digit. Signal Process., vol. 46, no. 1, pp. 78–80, 1999.

[57] S. T. Boo, H. Ibrahim, and K. K. V. Toh, "An improved progressive switching median filter," in *Proceedings - 2009 International Conference on Future Computer and Communication, ICFCC 2009*, 2009, pp. 136–139.

[58] S. J. Ko and Y. H. Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, 1991.

[59] T. Sun, M. Gabbouj, and Y. Neuvo, "Center weighted median filters: Some properties and their applications in image processing," *Signal Processing*, vol. 35, no. 3, pp. 213–229, 1994.

[60] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747–1754, 2005.

[61] U. Erkan, L. Gökrem, and S. Enginoğlu, "Different applied median filter in salt and pepper noise," *Comput. Electr. Eng.*, vol. 70, pp. 789–798, 2018.

[62] U. Erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, "An iterative mean filter for image denoising," *IEEE Access*, vol. 7, pp. 167847–167859, 2019.

[63] T. Veerakumar, B. N. Subudhi, S. Esakkirajan, and P. K. Pradhan, "Iterative Adaptive Unsymmetric Trimmed Shock Filter for High-Density Salt-and-Pepper Noise Removal," *Circuits, Syst. Signal Process.*, vol. 38, no. 6, pp. 2630–2652, Jun. 2019.

[64] B. Karthik, T. Krishna Kumar, S. P. Vijayaragavan, and M. Sriram, "Removal of high density salt and pepper noise in color image through modified cascaded filter," *J. Ambient Intell. Humaniz. Comput.*, 2020.

[65] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive Type-2 Fuzzy Approach for Filtering Salt and Pepper Noise in Grayscale Images," *IEEE Trans. Fuzzy Syst.*, 2018.

[66] M. Zhang, Y. Liu, G. Li, B. Qin, and Q. Liu, "Iterative scheme-inspired network for impulse noise removal," *Pattern Anal. Appl.*, vol. 23, no. 1, pp. 135–145, Feb. 2020.

[67] "Volume 3: Miscellaneous - USC Viterbi | Ming Hsieh Department of Electrical Engineering." [Online]. Available: https://minghsiehee.usc.edu/volume-3-

miscellaneous/. [Accessed: 15-Jun-2017].

[68] T.Randen, "Brodatz Textures," *Http://Www.Ux.Uis.No/~Tranden/Brodatz.Html*, 2007. [Online]. Available: http://www.ux.uis.no/~tranden/brodatz.html. [Accessed: 16-Jun-2017].

[69] Y. A. Vlasov and S. J. McNab, "Coupling into the slow light mode in slab-type photonic crystal waveguides," *Opt. Lett.*, vol. 31, no. 1, p. 50, Apr. 2006.

[70] H. L. Tan, Z. Li, Y. H. Tan, S. Rahardja, and C. Yeo, "A perceptually relevant mse-based image quality metric," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4447–4459, 2013.

[71] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[72] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision-based algorithm for removal of high-density impulse noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, 2007.

[73] USC Viterbi School of Engineering, "Volume 3: Miscellaneous - USC Viterbi | Ming Hsieh Department of Electrical and Computer Engineering," 2019. [Online]. Available: https://minghsiehece.usc.edu/volume-3-miscellaneous/. [Accessed: 25-Jan-2020].

[74] T. A. Nodes and N. C. Gallagher, "Median Filters: Some Modifications and Their Properties," *IEEE Trans. Acoust.*, vol. 30, no. 5, pp. 739–746, Oct. 1982.

[75] A. Noor, Y. Zhao, R. Khan, L. Wu, and F. Y. O. Abdalla, "Median filters combined with denoising convolutional neural network for Gaussian and impulse noises," *Multimed. Tools Appl.*, 2020.

[76] J. Lu, "M-PSK and M-QAM BER computation using signal-space concepts," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 181–184, 1999.

[77] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," *IEEE Signal Process. Lett.*, 2002.

[78] T. Chen and H. R. Wu, "Application of partition-based median type filters for suppressing noise in images," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 829–836, 2001.

[79] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2. pp. 490–

530, 2005.

[80] C. Kervrann and J. Boulanger, “Optimal spatial adaptation for patch-based image denoising,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[81] P. Chatterjee and P. Milanfar, “Is denoising dead?,” *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, 2010.

[82] A. Rahiman V and S. N. George, “Robust single image super resolution using neighbor embedding and fusion in wavelet domain,” *Comput. Electr. Eng.*, vol. 70, pp. 674–689, Aug. 2018.

[83] A. Buades, B. Coll, and J. M. Morel, “Image denoising methods. A new nonlocal principle,” *SIAM Rev.*, vol. 52, no. 1, pp. 113–147, 2010.

[84] B. Zhang and J. P. Allebach, “Adaptive bilateral filter for sharpness enhancement and noise removal,” *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 664–678, 2008.

[85] H. Talebi and P. Milanfar, “Global image denoising,” *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, Feb. 2014.

[86] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[87] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, “Image deblurring and denoising using color priors,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1550–1557.

[88] X. Wang, S. Shen, G. Shi, Y. Xu, and P. Zhang, “Iterative non-local means filter for salt and pepper noise removal,” *J. Vis. Commun. Image Represent.*, vol. 38, pp. 440–450, 2016.

[89] F. Ahmed and S. Das, “Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean,” *IEEE Trans. Fuzzy Syst.*, 2014.

[90] V. Gupta, D. K. Gandhi, and P. Yadav, “Removal of fixed value impulse noise using improved mean filter for image enhancement,” in *2013 Nirma University International Conference on Engineering, NUiCONE 2013*, 2013, pp. 1–5.

[91] D. Brunet, E. R. Vrscay, and Z. Wang, “On the mathematical properties of the structural similarity index,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp.

1488–1495, Apr. 2012.

[92] T. L. Lin *et al.*, “NR-Bitstream video quality metrics for SSIM using encoding decisions in AVC and HEVC coded videos,” *J. Vis. Commun. Image Represent.*, vol. 32, pp. 257–271, 2015.

[93] A. Shah *et al.*, “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.

[94] Z. Sun, B. Han, J. Li, J. Zhang, and X. Gao, “Weighted Guided Image Filtering with Steering Kernel,” *IEEE Trans. Image Process.*, 2020.

[95] H. Yous, A. Serir, and S. Yous, “CNN-based method for blotches and scratches detection in archived videos,” *J. Vis. Commun. Image Represent.*, vol. 59, pp. 486–500, Feb. 2019.

[96] W. Li, Y. Sun, and S. Chen, “A new algorithm for removal of high-density salt and pepper noises,” in *Proceedings of the 2009 2nd International Congress on Image and Signal Processing, CISP’09*, 2009, pp. 1–4.

[97] L. Xu, W. Jing, H. Song, and G. Chen, “High-Resolution Remote Sensing Image Change Detection Combined With Pixel-Level and Object-Level,” *IEEE Access*, vol. 7, pp. 78909–78918, 2019.

[98] C. T. Lu, Y. Y. Chen, L. L. Wang, and C. F. Chang, “Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window,” *Pattern Recognit. Lett.*, vol. 80, pp. 188–199, 2016.

[99] Z. Huang, Y. Zhang, Q. Li, T. Zhang, and N. Sang, “Spatially adaptive denoising for X-ray cardiovascular angiogram images,” *Biomed. Signal Process. Control*, 2018.

[100] L. Fan, X. Li, H. Fan, and C. Zhang, “An adaptive boosting procedure for low-rank based image denoising,” *Signal Processing*, 2019.

[101] R. Ma, H. Hu, S. Xing, and Z. Li, “Efficient and Fast Real-World Noisy Image Denoising by Combining Pyramid Neural Network and Two-Pathway Unscented Kalman Filter,” *IEEE Trans. Image Process.*, vol. 29, pp. 3927–3940, 2020.

[102] G. Yang, Z. Lu, J. Yang, and Y. Wang, “An Adaptive Contourlet HMM-PCNN Model of Sparse Representation for Image Denoising,” *IEEE Access*, vol. 7, pp.

88243–88253, 2019.

[103] R. G. Gavaskar and K. N. Chaudhury, “Fast Adaptive Bilateral Filtering,” *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 779–790, Feb. 2019.

[104] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[105] A. Ponomarenko, S. B. Goodwin, and G. H. J. Kema, “Septoria tritici blotch (STB) of wheat Septoria tritici blotch (STB) of wheat,” *Plant Heal. Instr.*, no. Figure 2, pp. 1–7, 2011.

[106] B. Münch, P. Trtik, F. Marone, and M. Stampanoni, “Stripe and ring artifact removal with combined wavelet—Fourier filtering,” *Opt. Express*, vol. 17, no. 10, p. 8567, May 2009.

[107] T. Bai and J. Tan, “Automatic detection and removal of high-density impulse noises,” *IET Image Process.*, vol. 9, no. 2, pp. 162–172, 2015.

[108] K. Kondo, M. Haseyama, and H. Kitajima, “An accurate noise detector for image restoration,” *IEEE Int. Conf. Image Process.*, vol. 1, pp. 321–324, 2002.

[109] G. M. Daiyan and M. A. Mottalib, “Removal of high density salt & pepper noise through a modified decision based median filter,” *2012 Int. Conf. Informatics, Electron. Vision, ICIEV 2012*, vol. 18, no. 5, pp. 565–570, 2012.

[110] T. Chen and H. R. Wu, “Adaptive impulse detection using center-weighted median filters,” *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 1–3, 2001.

[111] J. M. Kinser and J. M. Kinser, “Principle Component Analysis,” in *Image Operators*, First edition. | Boca Raton, FL: CRC Press/Taylor & Francis Group, [2019] |: CRC Press, 2018, pp. 111–126.

[112] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, 2003.

[113] M. Elad and M. Aharon, “Image denoising via learned dictionaries and sparse representation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 895–900.

[114] W. Zhao, Y. Lv, Q. Liu, and B. Qin, “Detail-Preserving Image Denoising via Adaptive Clustering and Progressive PCA Thresholding,” *IEEE Access*, vol. 6,

pp. 6303–6315, 2018.

[115] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, 2010.

[116] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering,” *Image Process. Algorithms Syst. Neural Networks, Mach. Learn.*, vol. 6064, p. 606414, 2006.

[117] K. Sebastian and S. Devi, “A novel model of feature extraction for lung cysts detection in CT image using Minutiae based Mumford and Shah functional model,” *Aust. J. Electr. Electron. Eng.*, vol. 16, no. 4, pp. 345–356, Oct. 2019.

[118] P. Karthikeyan and S. Vasuki, “Efficient decision based algorithm for the removal of high density salt and pepper noise in images,” *J. Commun. Technol. Electron.*, vol. 61, no. 8, pp. 963–970, 2016.

[119] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, “Conditional Probability Models for Deep Image Compression,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[120] B. Li and W. Xie, “Image denoising and enhancement based on adaptive fractional calculus of small probability strategy,” *Neurocomputing*, vol. 175, no. PartA, pp. 704–714, Jan. 2015.

[121] J. Liu, S. Chen, and Z. H. Zhou, “Progressive principal component analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3173, no. May 2014, pp. 768–773, 2004.

[122] J. J. Hong and J. Zhang, “Progressive PCA modeling for enhanced fault diagnosis in a batch process,” in *ICCAS 2010 - International Conference on Control, Automation and Systems*, 2010, pp. 713–718.