SOME STUDIES ON IMAGE ENHANCEMENT AND FILTERING

A THESIS

SUBMITTED TO THE DELHI TECHNOLOGICAL UNIVERSITY
FOR THE AWARD OF THE DEGREE OF

DOCTOR OF PHILOSOPHY

ISHA SINGH

Department of Information Technology
DELHI TECHNOLOGICAL UNIVERSITY
DELHI, INDIA

2020

SOME STUDIES ON IMAGE ENHANCEMENT AND FILTERING

By

ISHA SINGH

(2K12/PHD/IT/01)

Submitted in fulfillment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

Supervisor

Prof. O.P. Verma
Department of Electronics & Communication Engineering
Delhi Technological University
New Delhi

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
DELHI, INDIA

2020

DECLARATION

I hereby declare that the thesis entitled "**Some Studies on Image Enhancement and Filtering**" to be submitted in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in the Department of Information Technology of the Delhi Technological University, Delhi is an authentic record of my own work carried out under the supervision of Prof. O. P.Verma. The thesis has not been submitted either in part or whole to any university or institute for the award of any degree or diploma.

Place: DTU, Delhi	Isha Singh
Date:	(2k12/PHDIT/01)

CERTIFICATE

This is to certify that the thesis entitled "**Some Studies on Image Enhancement and Filtering**" being submitted by **Isha Singh** to the Department of Information Technology, Delhi Technological University, Delhi, for the award of the degree of Doctor of Philosophy, is a record of bonafide research work carried out by her under my guidance and supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

(Supervisor)

Prof. O.P.Verma
Professor
Department of Electronics and Communication Engineering
Delhi Technological University, Delhi

ACKNOWLEDGMENTS

Thanks to merciful Lord for everything in my life!!

This thesis work has been a roller-coaster journey and I am indebted to acknowledge a few people who made this journey successful.

First and foremost, it is a great pleasure to express my sincere gratitude and appreciation to my supervisor Prof. O. P. Verma. It is my honor to work under the supervision of such a kind man of values and great expertise. The completion of my thesis work was an uphill task and without the constant support and suggestions of Prof.Verma, this thesis would not have seen the light of the day. He is a professor of wide acclaim and profound knowledge whose valuable guidance and consistent scholarly inputs have helped me to form the basis of this work.

My extended thanks go to the faculty and staff of the Department of Information Technology, for their assistance and encouragement during my research work.

I feel fortunate to have friends like Rajni Sethi and Anshu Khurana, whose unbounded support in my personal and professional life made me complete this work. Thank you for inspiring me and occasionally pushing me to certain deadlines.

My sincere gratitude for my mother in law Mrs. Asha diwvedi for accepting an unemployed doctoral student in the family and supporting my choices.

I would like to thank a jewel personality my uncle Mr. Anurag Singh, who is a mentor of my life and has shaped my career. Without whom I would not have been what I am today.

I cannot forget to mention a special living creature, my pet dog Jack for uplifting my moods and filling me up with positive energies during the low phases.

To my Sister, Mrs. Anchal Singh, your love and compassion have helped me to sail through the tough times. The fruitful discussions with you have saved me in the darkest hours.

I am forever indebted to my parents Mrs. Munesh and Mr. S. P. Singh for their unfettered love, invaluable blessings and unwavering belief in me. I dedicate this thesis to my parents. Thanks to almighty for being there as my parents. This thesis is your dream come true. I owe them everything.

Finally, this is the word of acknowledgment I have saved for my dear husband Abhishek Diwvedi and my son Reyansh Singh Diwvedi. These two are the pillars of strength of my life. My heartfelt gratitude for my partner who witnessed this journey with me with deep insight and supported me unconditionally with his love. Your random naggings pushed me to reach my goal. My son Reyansh is the ray of sunshine in my life whose smile transforms my dull moments instantly. Thank you for being there with me round the clock.

New Delhi, 2020 Isha Singh

ABSTRACT

Images are one of the best sources for research & communication. Proper analysis of image data is essential for various fields like satellite images & remote sensing, Biometrics, Criminology, military surveillance, astrology & many more. Requirements of the users vary according to the nature of application in use.Image processing plays a critical role in the analysis of data and is an integral part for various applications.

Image denoising is one of the challenging branches of image processing. Impulse noise is among the prevalent noises that degrade the image quality and its subsequent noise suppression plays a pivotal role in the enhancement of images. The presence of impulse noise cannot be averted during the digitization, acquisition, and transmission of images. Many state of art filters are available in the literature to deal with impulse noise encountered in images. Filters having dual modes of detection and restoration exhibit superior performance in removing noise and thereby keeping the original information of the images intact. In real world sometimes the user is uncertain about his requirements therefore the characteristics of the employed filter for impulse noise removal should be adaptable to the indecisive features of an image.The denoising filter should be robust enough to handle the varying amounts of noise density and should be intuitive in nature.

This thesis work tries to cater to all the essential features required for an efficient filter. Incorporation of fuzzy logic makes the filter more versatile.Investigations performed in this thesis show that the proposed work excels in the quantitative as well as qualitative manner. Four schemes introduced in this thesis are :

(i) High-density impulse noise detection using FCM algorithm (HDIND)

(ii) Edge preserving fuzzy filter for the suppression of impulse noise in images (EFFSIN)

(iii) Heuristic analysis of neighboring pixels for impulse noise detection (SPHN)

(iv) Impulse noise removal in color image sequences using Fuzzy logic (INFL)

The initial three schemes namely HDIND, EFFSIN, and SPHN focus on the grayscale images and INFL is proposed for color image sequences. All the schemes incorporate an efficient detection criterion and after proper classification of noisy and noise-free pixels, performs the restoration procedure. The use of fuzzy logic in the methods has enhanced the decision making aspect of the algorithms to classify the noisy pixels present in an image. The simulation results are done in isolation for all the schemes deduce that HDIND and EFFSIN are robust in nature and their performance does not deteriorates with a rise in noise density. The edge-preserving nature of EFFSIN preserves the original image data and false alarm rates are reduced. SPHN provides good PSNR and MSE results. INFL is a Spatio-temporal filter that gives excellent performance. SSIM, PSNR, MSE, False alarm and Miss detection rates are used as quality measures to analyze the proposed mechanisms.

LIST OF PUBLICATIONS

Journals

1. **Isha Singh** and Om Prakash Verma. "High Density Impulse Noise Detection using Fuzzy C-means Algorithm." *Defence Science Journal, Vol. 66, No. 1, January 2016, pp. 30-36, DOI: 10.14429/dsj.66.8722,* 2016. (SCI)

2. **Isha Singh** and Om Prakash Verma, "Edge Preserving Fuzzy Filter for Suppression of Impulse Noise in Images", *International Journal of Engineering and Advanced Technology,Vol.8,Issue6,August2019,pp:16271633,DOI:10.35940/ijeat.F8206.08869,* 2019. (SCOPUS)

Conferences

3. Verma, Om Prakash, Shreya Arora, and **Isha Singh**. "A novel approach for salt and pepper noise removal based on heuristic analysis of neighboring pixels." *International Conference on Computing for Sustainable Global Development (INDIACom), DOI:10.1109/IndiaCom.2014.6828134*. IEEE, 2014.

Under Review

Isha Singh and Om Prakash Verma, "Impulse Noise Removal in Color Image Sequences using Fuzzy logic", a paper under review in Multimedia tools and Applications, Springer. (SCI)

LIST OF ABBREVIATIONS

Abbreviation	**Detail**
AFSM	Adaptive Fuzzy Switching Median filter
ASWM	Adaptive Switching Median filter
AFISD	Adaptive Fuzzy Inference System based directional filter
CB	Color Band
CEB	Contrast Enhancement Based filter
CMYK	Cyan Magenta Yellow Black
CWM	Center Weighted Median filter
DDB	Directional Difference based noise detector
EEP	Efficient Edge Preserving filter
EEPA	Efficient Edge Preserving Algorithm
EFFSIN	Edge Preserving Fuzzy filter for suppression of Impulse Noise in Images
ERC	Effective impulse detector based on Rank-Order Criteria
FA	False Alarm
FAPG	Fast Averaging Peer-Group filter
FAHDS	Fast Adaptive & selective mean filter for High- Density Salt & pepper noise
FCM	Fuzzy C-Means
FSM	Fuzzy Switching Median Filter
FINR	Fuzzy filter for Impulse Noise Removal

FSBH	Fuzzy svm-based Histogram filter
FRINR	Fuzzy filter for Impulse Noise Reduction
HDIND	High Density Impulse Noise Detection using FCM algorithm
HDCNN	High Density impulse noise detection & removal using deep Convolutional Neural Network
HEIND	Highly Effective Impulse Noise Detection
HSV	Hue Saturation Value
MD	Miss Detection
MDBUTM	Modified Decision Based Unsymmetric Trimmed Median filter
MED	Median
MSE	Mean square error
NAFSM	Noise Adaptive Fuzzy Switching Median filter
PDF	Probability Distribution Function
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
SDOD	Standard Deviation for obtaining the Optimal Direction
SSIM	Structural Similarity Index Measure
SPHN	Salt & Pepper noise removal based on Heuristic analysis of Neighboring pixels
TSM	Tri-State Median filter

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1

Introduction

Human beings are visual creatures. They have a remarkable ability to process and remember pictures at an incredible speed. Images are universal to everyone irrespective of their linguistic information. Images are easier to process by the human brain at a rapid rate as compared to words. Today's short attention world makes images as a perfect source of research and communication. But sometimes due to various factors present in an environment the information present in an image gets ruined. This visual information needs some processing so that it is better suited for further analysis. Digital image processing [1] comprises analyzing the received visual information by a digital computer.

The following sub-sections provides a brief idea about Image enhancement & filtering and noise in images.Problem definition is given in section 1.3 and Section 1.4 states the Scopes and objectives of the thesis.Thesis layout is discussed at the end of the chapter.

1.1 Image Enhancement and Filtering

Images can be distorted easily and therein Image Enhancement plays an eminent role to fetch the specific information out of an image. Under the umbrella of digital image processing, Image Enhancement is an important pre-processing step to enhance features of interest and to highlight particular data present in an image. Image enhancement is performed for various applications as per their requirement such as astronomy, biometrics, medical imaging, defense and military surveillance, remote sensing, criminology, forensics entertainment, and communication services, etc.

Image Enhancement covers a wide range of activities like fine-tuning of brightness and contrast, convolution, creating a balanced image, filtering in the frequency domain, augmentation of edges and many more [2,3]. Image enhancement is generally followed by Image Restoration techniques which involve the correction of artifacts produced in the image during the imaging process. This process aims to generate an image with maximum proximity to the real image characteristics.

Noise removal is among the prominent classes of image restoration and this thesis work is primarily based on image noise removal as discussed in the Problem Definition section. Moreover, this thesis studies the effects of efficient image noise removal for high-quality image enhancement.

1.2 Noise in Images

Image acquisition digitizes an image captured from a real-world source. Every step in the acquisition process may bring up some irregular variations in the estimation of pixel values. These variations are known as *noise* [4].

Suppose you want to share a photograph of your dog with your companion over the internet. This is accomplished by snapping a picture with a traditional camera, putting the film grain into print, scanning the print into a computer and then finally mailing it. Over this short process due to various factors present in an environment, noise is introduced in an image as shown in

Fig.1.1. Dust particles and air present in the environment may interfere with the light reaching the lens of the camera. Film grain noise and photoelectric noise of the scanner further deteriorate the quality of an image [5]. Transmission of images over the internet often undergoes loss in bit information which introduces noise in images.

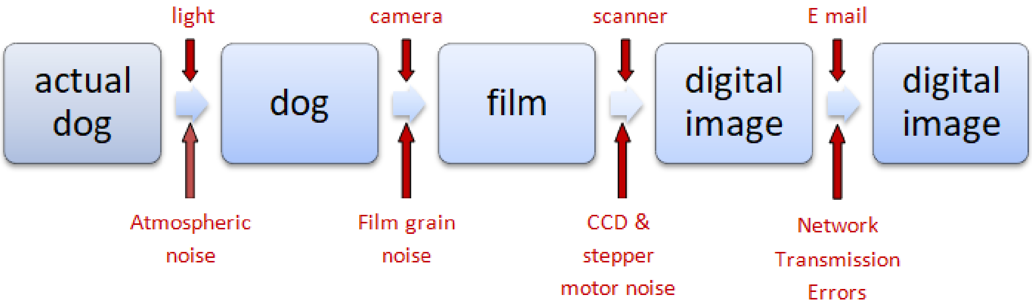


Figure 1.1: Noise at each step of image acquisition

Before proceeding further, let's get familiar with the notation and terminology used for digital images.

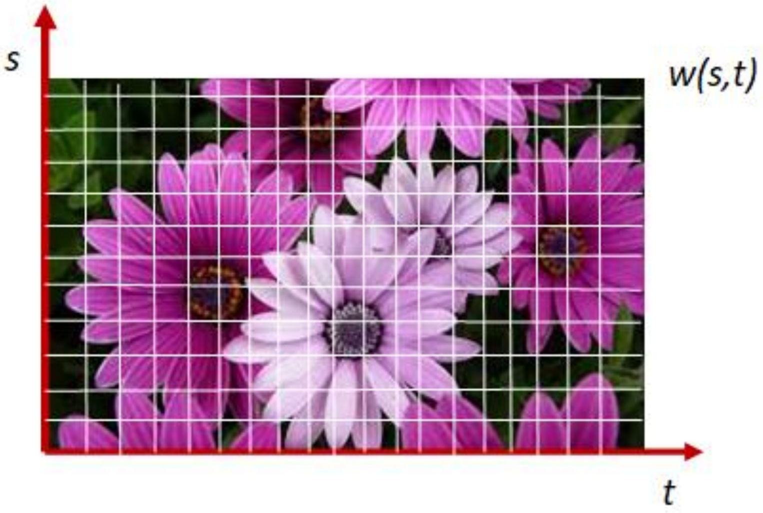


Figure 1.2: A digital image with resolution s × t

An image *I* is depicted as a two-dimensional function *w(s,t)*, where *s and t* denote the spatial (plane) coordinates. *w* is the intensity at position *(s,t)* which is also known as gray value or amplitude at that position. For a digital image *w(s,t)* should have finite and discrete values.

$$I = w(s,t) \tag{1.1}$$

The digital images are also known as bit-maps or raster-scan images as they are composed of an array/grid/matrix of picture elements referred to as pixels. Pixels are the smallest unit on screen [6].

Generally, images are an outcome of the estimation of some physical phenomena such as light, energy, heat or distance, etc and can be assessed by any numerical form.

Grayscale images are based on the measurement of light intensity or brightness. Minimum brightness implies black and maximum brightness is white. For efficient storage, pixels are quantized to P levels such that $w(s,t) \epsilon \{0,1,....P\text{-}1\}$ and P is modeled as 2^k. Since the human ocular system has some limitations, generally for grayscale images k=3 i.e 8 bits. For more critical applications like medical scans may use 12 to 16 bits per pixel.

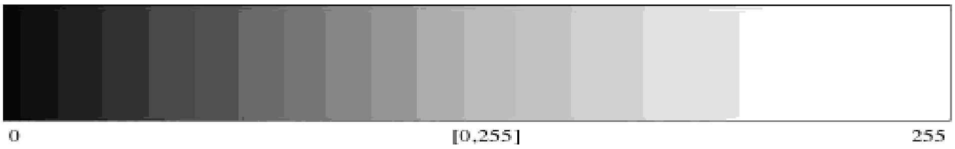


Figure 1.3: 256 gray levels for an 8-bit image

Color images are based on the measurement of light intensity and chrominance. A color image is comprised of pixels, where each pixel has three values for red, green, and blue components. Red, green and blue abbreviated as RGB are known as primary colors as with various compositions of RGB any color can be created. For an eight-bit image, each primary color has 256 levels and each color pixel takes three bytes i.e 24 bits to get store in memory. The available standard color models are RGB, CMYK and HSV.

HSV stands for hue, saturation, and value. CMYK stands for cyan, magenta, yellow and black.

A grayscale digital image is analyzed with a single 2D integer array and is comprised of three 2D arrays for each color band (R, G, B) in a color image [7].

Fig. 1.1 depicted how noise disseminates into images during their acquisition and transmission as a result of optical errors or instrumentation imperfections. But noise varies in its characteristics. To recognize the type of noise encountered in an image, various classes of noise are present [8-11] :

- Impulse (Salt & Pepper) Noise
- Gaussian Noise
- Quantization (Uniform) Noise
- Shot (Poisson) Noise
- Speckle Noise
- Rayleigh Noise
- Gamma Noise
- Exponential Noise

This thesis work is based on the efficient detection and removal of "*impulse noise*" from images.

Impulse Noise (Salt & Pepper)

Digital images are often impaired with impulse noise as a result of erroneous camera sensors, bit errors during transmission or imprecise memory locations. It is seen arbitrarily on images as black and white dots, where black is termed as *pepper* noise and white is referred to as *salt* noise [10]. Property of impulse noise is to produce extreme intensities that are 0 or 255 in an 8-bit image. The PDF (Probability distribution function) for an image affected with impulse noise is as follows:

$$D_i(U) = \begin{Bmatrix} D_a \ , & \mathrm{U} = a \\ D_b \ , & \mathrm{U} = b \\ 0 \ , \ otherwise \end{Bmatrix} \qquad (1.2)$$

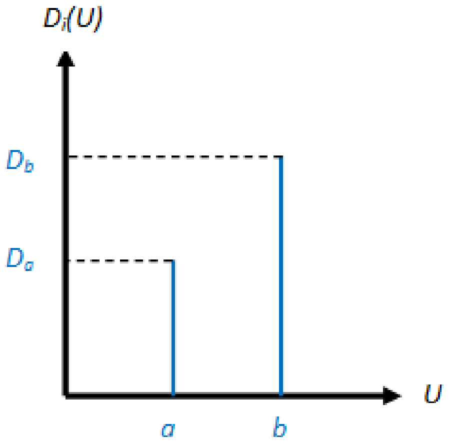


Figure 1.4: PDF for impulse noise

Where, a,b and U are the gray level values of pixels.

For salt noise, b > a and incase of pepper noise, a > b.

In 1975, with the invention of the CCD camera of resolution 100 × 100, the study of image denoising started broadly. Since then, there is a massive evolution in the approaches and methodologies used for noise suppression. Image restoration has the objective to produce an image as close as possible to the real image. This thesis work has thrust upon impulse noise removal from digital images. A general model for the image degradation/restoration process due to impulse noise is given below:

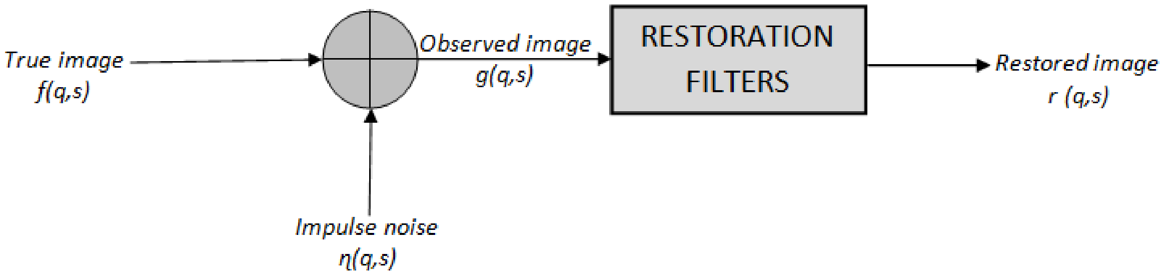


Figure 1.5: Model for impulse noise removal

$$g(q,s) = f(q,s) + n(q,s) \tag{1.3}$$

g(q,s) is the resultant image after the introduction of impulse noise *n(q,s)* in the original image *f(q,s).*

The type of restoration filter employed depends on the image characteristics [12,13]. There are two categories in which the filtering process can be performed:

1. Spatial domain

   Image pixels are directly manipulated in this case. Our study is focussed on the study of spatial filters only. *Chapter 2* provides a broad study of spatial filters.

2. Frequency domain

   Instead of direct manipulation of pixels, the Fourier transform of the image is computed and all the operations are performed over it. Inverse Fourier transform is computed for the resultant image.

Efficient noise detection and its restoration play an important pre-processing step for various applications like astronomy, medical imaging, forensics, defense services, remote sensing biometrics to name a few. These fields have images as their critical input data and any artifacts present in them must be removed before further processing. Various Filters are available in the literature to deal with the above-mentioned noises in images as noise can severely affect the data present in them. This thesis presents various algorithms for the detection of *impulse noise* from images and thereby its efficient restoration.

1.3 Problem Definition

In literature, there are numerous techniques to deal with impulse noise as it can alter the true information content of an image. This thesis attempts to address some major drawbacks present in existing methods used for impulse noise and proposes new algorithms for efficient impulse noise detection and removal. Following problems are conferred in this thesis:

- Most of the state of art filters emphasize the removal of impulse noise in various possible manners but lack the proper mechanism for impulse noise detection. Appropriate noise detection is an essential pre-requisite for noise restoration. Better the detection process, more superior will be the filtering result.
- With impulse noise removal, a lot of algorithms are not successful to preserve the fine details present in an image. The real image pixel data is modified as a result of the filtering process which results in various side effects on an image such as blurring, loss of edges, lack of required sharpness, etc. The quality of image data should not be compromised with the filtering process.
- There is a massive advancement in the technology used for impulse noise suppression in the last decade such as the incorporation of neural networks, soft computing, and fuzzy logic. But the formulation of such methods is so complex that in spite of good results their practical use becomes limited. The structure of denoising filters with these advancements should be easy to implement and the computation time must range in practical limits.
- Lastly, even if all the required parameters discussed above for a good filter are catered, most of the filters fail when there is a rise in noise density. The filter should be robust enough to handle even high-density noise levels with consistent performance.

1.4 Scopes and objectives of the thesis

Literature abounds of impulse noise filters. Many filters perform the restoration procedure directly without taking into account that the pixel is affected by noise or noise-free. This produces further artifacts in the image in the form of a blur, loss of original pixel information, edge distortion and many more. Techniques which perform detection first and filtering later have shown promising results but lack in structural quality and are complex to build, thereby resulting in high computation timings. The objectives of this thesis after the broad analysis of techniques present in literature are as follows:

- To devise a reliable detection scheme prior to restoration procedure to efficiently remove impulse noise from images and preserve image details.
- To work towards improved detectors using fuzzy logic which aids in better computational complexity and is easy to implement.
- To explore the utility of restoration mechanisms after proper detection to improve the qualitative and quantitative results.
- To create a robust filter for impulse noise removal whose performance is not degraded with a rise in noise density.

Apart from grayscale images, we also attempt to remove impulse noise from color videos with a Spatio-temporal filter. In the case of video sequences, various methods consider the spatial relationship within the frame under process or the temporal relationship between the frames is analyzed. But for efficient noise removal fusion of temporal filtering and spatial filtering is required.

1.5 Thesis Layout

The thesis is organized as follows:

- **Chapter 1**: **Introduction**

  It provides a fundamental idea about the concepts of noise and image enhancement. Problem formulation and thesis objectives are also discussed.

- **Chapter 2: Literature Review**

This chapter presents a brief study of the existing techniques present in literature used for impulse noise suppression.

- **Chapter 3: High-density Impulse noise detection using FCM algorithm**

The objective of this chapter is to introduce a novel scheme for impulse noise detection which uses the FCM algorithm. This algorithm is compared with many state of art techniques and shows superior results. Experimental results are given at the end of the chapter.

- **Chapter 4: Edge preserving fuzzy filter for suppression of impulse noise**

Another novel technique devised by this thesis work is presented in this chapter. This is an expansion of the work presented in the previous chapter. Along with the efficient detection of noise, it preserves the edges present in an image thereby improving the qualitative and quantitative results. This chapter is concluded with comparative analysis and results.

- **Chapter 5: Heuristic analysis of neighboring pixels for impulse noise detection**

This chapter introduces a new filter SPHN based on heuristic analysis of neighboring pixels for impulse noise. SPHN maintains the fine balance between image basic structure and blurring. Simulation results and conclusions are provided at the end.

- **Chapter 6: Impulse Noise Removal in color image sequences using Fuzzy Logic**

This chapter introduces a fuzzy filter for impulse noise suppression in color image sequences. A brief literature study of filters used for video is provided followed by simulation results and analysis. Lastly, conclusions are presented.

- **Chapter 7: Conclusion and future work**

The work is summarized in this chapter. The thesis concludes with the contributions of the work done in this thesis along with the discussion of their future scopes.

Chapter 2

Literature Review

This chapter presents a brief study of the existing techniques present in literature used for impulse noise suppression. The initial section discusses about spatial filters and its categories. In section 2.2, non-linear filters are studied in detail. The last section provides the details of all the performance measures used in this thesis work.

2.1 Spatial Filters

Spatial filters directly manipulate the image pixels. The spatial plane is the image plane itself. Spatial filters are classified into two categories:

- Linear filters
- Non- linear filters

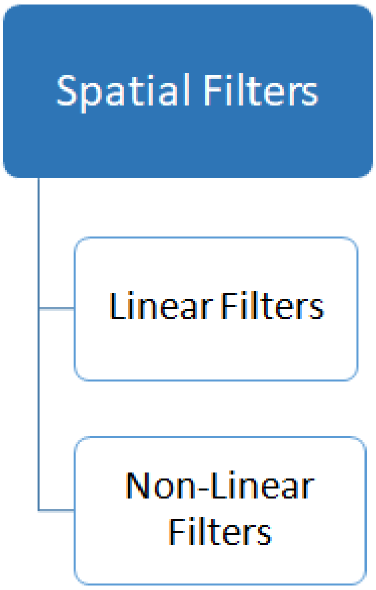


Figure 2.1: Spatial filters categories

2.1.1 Linear Filters

In the initial phases of image denoising, linear filters were employed extensively [14]. Simple structure and decent performance lead to their acceptance by various applications. But in the presence of noise, their performance degraded as they are likely to produce blur and loss of fine details present in an image. Linear filters have the principle to perform filtering without detection. Irrespective of the pixel status whether it is corrupted with noise or not, linear filters perform its operations.

In mathematical terms, A linear operator '*Q*', maps any signal *x* into signal *y*, using the relation *y=Q(x)*. For *Q* to be a linear operator it should satisfy both the principles of superposition and proportionality.

Mean filters are the most common linear filters available for impulse noise removal. Many variations of these filters are also available.

- **Mean Filter**

It is the simplest linear filter also known as an averaging filter [15]. It computes the average value in the window and uses it to replace the pixel value at the center. Usually, the window size is $(2T+1)^2$, where *T* should be a positive integer. A window size of 3×3,5×5,7×7 etc is used for computation. Let m×n be the window size, $W_{m\times n}$ is the sliding window centered around *(i,j)*, X stands for the noisy image and Y represents the restored image.

$$Y_{i,j} = \frac{1}{m \times n} \sum_{(u,v)\in W_{m,n}} X_{u,v} \qquad (2.1)$$

This filter is easy to implement but does not provide efficient detail preservation in the presence of impulse noise.

- **Weighted mean filter**

A weighted mean filter is an extension of the basic averaging filter [16,17]. Instead of giving equal weights to all the neighboring pixels, this filter assigns more weights to the pixels closer to the center pixel in processing. Weights can be computed in many ways such as distance computation etc. Mean is computed from the product of weight and pixel value for the restored pixel value. Better performance than basic mean filter is accomplished in terms of smoothing of an image but lags in preserving the basic details of an image.

Linear filters provide ease of use and simple design but they do not provide substantial image detail preservation and fail terribly in impulse noise suppression of high density [18]. Over the years many modifications have been done with the basic principle of averaging filter to meet up the current requirements in the area of impulse noise removal. Few nonlinear techniques are the outcome of such variations in mean filters and many more are discussed in the next section of nonlinear filters.

2.1.2 Non –Linear filters

The last few decades have shown massive evolution in the development of nonlinear filters. They surmount the shortcomings of linear filters efficiently. In literature, there are various classes of nonlinear filters [19], as shown in Fig. 2.2.

The broad categories of non-linear filters are namely Order statistics filters, Non- linear mean filters, Morphological filters, Homomorphic filters, and polynomial filters.

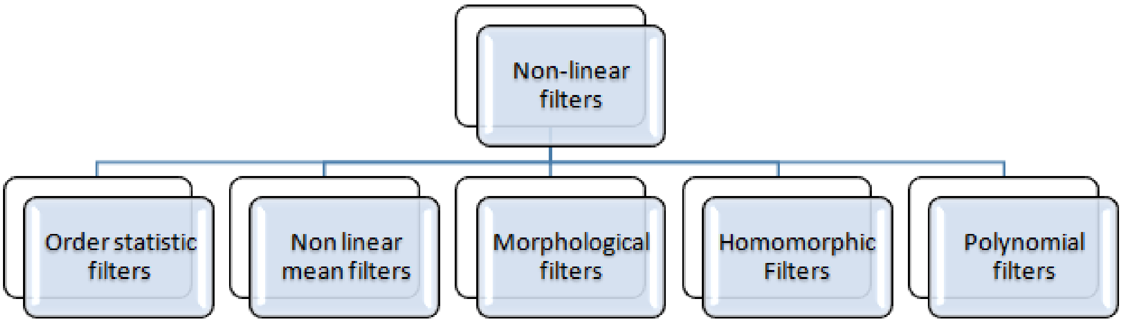


Figure 2.2: Classification of non-linear filters

Order statistic filters are the standard filters among non-linear filters [20]. They assign ranking or some order to the pixels comprised within the current processing window and then restore the pixel value derived from its ranking response. Prominent filters in this category are Median filter and its several variations, Max-Min filter, Midpoint filter, stack filters, etc. Order statistic filters are robust in nature and efficiently deal with impulse noise [21]. This thesis work has a principle focus on order statistic filters.

Non-linear mean filters follow the statistical approach too but they have been categorized in a different class of family of non-linear filters as they are derived from the modification in the basic linear mean filter. Along with median filters, non-linear mean filters have considerable importance in the field of noise removal. Many variations of mean filters that are nonlinear in nature are also present in literature [22], such as geometric mean filters, harmonic mean filters,contraharmonic mean filters, etc.

Morphological filters focus on geometric features of an image rather than analytical features. There is a vast group of filters utilizing the concept of morphological operations [23]. Biomedical studies, shape analysis, and other image enhancement applications employ morphological based filters.

Homomorphic filters are one of the oldest means of image enhancement primarily used for multiplicative noise, signal-dependent noise, contrast enhancement in images and seismic signal processing [1,21,22]. It involves a nonlinear mapping of images into a different domain like frequency domain then with the help of Fourier transform and use of linear techniques performing certain operations followed by inverse mapping into the original domain.

With the rise in the high processing power of computers, polynomial or Volterra filters are also used for certain applications in image processing. Quadratic image filters are a special subclass of these filters [24].

As discussed in previous sections, this thesis work aims to study spatial filters and order statistic filters in general, the next section provides a brief study of some notable nonlinear filters based on statistical information.

2.2 Study of non-linear filters

Among the big family of nonlinear filters, order statistics filters play a vital role in impulse noise removal. Order statistic filters were introduced by Bovik in 1980'[25]. The prime benefit of using order statistics is fast computation and speed. Since its inception, the Median filter is the most prevalent filter that falls in this category [26]. Its elementary structure makes it suitable for various applications. The basic principle of median filter has been incorporated by numerous algorithms and its many variations are present in literature. A brief study of the median filter and its variations is as follows:

Median filter [27,28]

The median filter is the leading mechanism for impulse noise removal which uses order statistics. It reduces the intensity variation between the pixels.

$$Y_{i,j} = MEDIAN_{u,v\,\in\,w(m,n)}\left(X_{u,v}\right) \tag{2.2}$$

All the pixels in filtering window $W_{m,n}$ are sorted and its median value is determined to replace the processing pixel value.

It has an uncomplicated structure and gives modest results for low noise levels which makes the median filter as a preferred algorithm of all. But the noise level is an important parameter for any filter performance. With the rise in noise levels performance of median filter degrades as in general, it operates similar to low pass filter that intercepts the high-frequency elements leading to blurring in images. Several modifications and extensions of median filter have evolved during the last decades of research for impulse noise removal such as Weighted median filter [29], Progressive switching median filter [26], Center-weighted median [30], Tri-state median filter [31], Direction based adaptive weighted switching median filter [32], Vector median filters [33], Switching weighted vector median filter [34], Modified switching median filter [35], Modified weighted mean filter [36], etc.

To surmount the drawbacks of median filters, adaptive median filters were introduced in 1988 by Lin [28]. It exhibited superior performance as compared to the basic median filter but faced issues with the correct detection of positive and negative impulses. Hwang and Haddad [37], modified this algorithm to overcome the existing problems. Image characteristics alter from one point to another and adaptive median algorithms employ the idea of changing the filter size depending upon the image characteristics. A series of adaptive median algorithms have evolved since their inception and used extensively for impulse noise removal. A few examples from the recent decade are [38-44].

Weighted median filters is another popular class of median filters. The basic methodology is the same as that of the weighted mean filter [29] discussed in the previous section. The only difference is instead of mean value, the median value is selected for restored pixel value. These types of filters are much more flexible than the basic median filter. Initially, Arce et al.

examined the weighted filters with weights having real positive values and in 1998 [45] with negative weight values. Center-weighted filter [46] (CWM) is distinctive in its nature as it assigns maximum weight exclusively to the central pixel of the filter which makes it more efficient. As a subclass of weighted median filters, CWM efficiently deals with salt noise but with pepper noise, its performance deteriorates.

The standard median filter has shown convergence property that is it is likely to have the output of the median filter as one of the input samples without changing its value. This property of median filters has given rise to another category of median filters known as recursive median filters. Recursive median filters [47-50] use the previously filtered values as their input. Faster convergence and high noise attenuation levels are achieved with the use of recursive median filters.

Decision-based filters is another well-known class of median filters which classify the noisy pixels and uncorrupted pixels and then process them accordingly. Proper detection of corrupted pixels increases the efficacy of the median filter. In the absence of any decision-based noise detection algorithm before median filtering, all the pixels are considered noisy and this compromises the fine details of an image. To conquer these effects, a detection operation for impulse noise before filtering is performed. Although this classification is a tedious task, recent decades have seen the evolution of such algorithms with the fusion of median filters with other methodologies namely fuzzy logic, neural networks or combination of weighted median filters with decision-based filters and so on. Several algorithms based on decision methodology combined with other practices exist in literature such as Tri-state median filter (TSM) [31], An efficient edge-preserving algorithm (EEP) [51], Fuzzy switching median filter (FSM) [52], Noise adaptive fuzzy switching median filter (NAFSM) [53], A highly effective impulse noise detection Algorithm (HEIND) [54], Contrast enhancement based filter (CEB) [55], MDBUTM [56], Adaptive switching median (ASWM) filter [57], Quadrant based spatially adaptive fuzzy filter [58], A fuzzy impulse noise filter based on boundary discriminative noise detection [59] and many more [60].

In the detection phase, different methods based on thresholding, histogram-based and clustering are used. Each has its own limitations and advantages, but the clustering technique yields much better results. Clustering is a grouping of a large number of data into clusters of smaller groups of similar data. Various algorithms present in literature have incorporated different clustering techniques such as modification of advanced boundary discriminative noise detection algorithm [61], efficient techniques for denoising of speckle and highly corrupted impulse noise images [62] and removal of impulse noise with an adaptive alpha-trimmed Mean fuzzy filter [63]. Various clustering techniques are present which can be used depending on the type of data to be classified and as per the requirement of the procedure.

The fusion of median filters with other techniques have enhanced the qualitative as well as quantitative results for impulse noise detection and removal [64].

This thesis work has derived new algorithms for impulse noise suppression using median filtering and fuzzy logic. Fuzzy logic has upgraded the performance of our work as discussed in upcoming chapters.

2.3 Performance Measures

To assess the performance of images, many metrics are available for research [65-70]. The performance measures used in our work for the evaluation of the filters for impulse noise removal are Peak signal to noise ratio (PSNR), Mean square error (MSE), Structural similarity index (SSIM), False alarm rate (FA) and Miss detection rate (MD).

2.3.1 Mean square error (MSE)

Given an original image *f* and the reconstructed image *I* of size M×N pixels.

MSE is calculated using:

$$MSE(f,I)=\frac{\sum_{r=1}^{3}\sum_{q=1}^{L}\sum_{p=1}^{M}\left[I(p,q,r)-f(p,q,r)\right]^{2}}{3\times L\times M} \tag{2.3}$$

Lower the value of MSE better the quality of the reconstructed image. Ideally, it should be zero but practically it's not feasible.

2.3.2 Peak signal to noise ratio (PSNR)

Using PSNR we can estimate the quality of the reconstructed image after noise removal. It is defined as the ratio of peak signal power to noise power. The basic idea is to compute a single number that reflects the quality of the reconstructed image.

PSNR is related to MSE by the following equation:

$$PSNR(f,I)=10\log\left(\frac{1}{MSE(f,I)}\right) \tag{2.4}$$

The higher the value of PSNR better is the similarity between two images.

2.3.3 Structural Similarity Index Measure (SSIM)

SSIM is calculated using the following equation [68,69] :

$$SSIM = \frac{(2\mu_I \mu_f + c_1) + (2\sigma_{If} + c_2)}{(\mu_I^2 + \mu_f^2 + c_1) + (\sigma_I^2 + \sigma_f^2 + c_2)} \tag{2.5}$$

c₁ and *c₂* as constants to stabilize the division with weak denominator. μ_I, μ_f ,σ_I, σ_f, *and* σ_{If} denote the average intensity, standard deviation, and cross-covariance values for images *I* and *f*, respectively. c_1 and c_2 have default values of $c_1 = (0.01\times255)^2$ and $c_2=(0.03\times255)^2$.

Three comparison measurements to wit luminance(*l*), contrast(*C*) and structure(*S*) comparisons are performed between the two images of the same size. *If the SSIM* index results in a decimal value *1*, *the* absolute structural similarity is implied between two images, whereas *0* indicates an absence of structural similarity.

$$l(I, f) = \frac{2\mu_I \mu_f + c_1}{\mu_I^2 + \mu_f^2 + c_1} \tag{2.6}$$

$$C(I, f) = \frac{2\sigma_I \sigma_f + c_2}{\sigma_I^2 + \sigma_f^2 + c_2} \tag{2.7}$$

$$S(I, f) = \frac{\sigma_{If} + c_3}{\sigma_I \sigma_f + c_3} \tag{2.8}$$

$$c_3 = \frac{c_2}{2} \tag{2.9}$$

SSIM is formulated as a weighted combination of above described comparative measures:

$$SSIM(I, f) = \left[l(I, f)^{\theta_1} \cdot C(I, f)^{\theta_2} \cdot S(I, f)^{\theta_3} \right] \tag{2.10}$$

Variables θ_1, θ_2 and θ_3 define the relative importance of the three components.

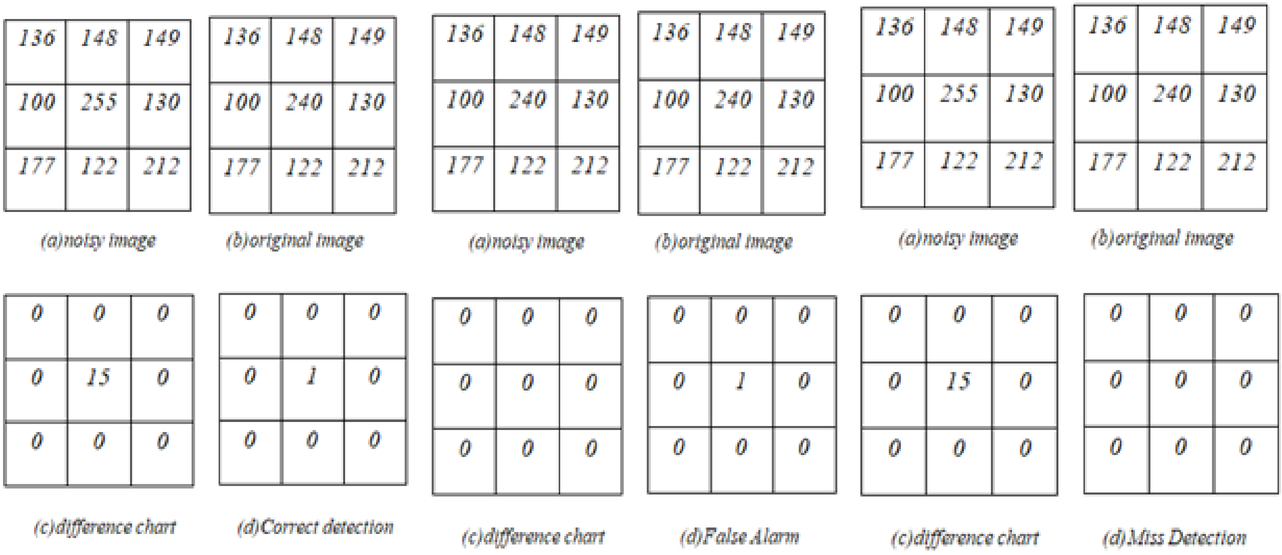
When θ_1, θ_2, $\theta_3 = 1$ *and* $c_1 = c_2$, *SSIM* is deduced as in equation (2.5).

2.3.4 False alarm rate (FA)

False alarm rate arises when a noise-free image pixel is identified as noisy. A difference based method is used for the evaluation of our work. Noise matrix is generated based on differences computed between the noise chart and the real image. In Fig. 2.3, ‘1’ in the noise chart indicates the presence of noise, whereas ‘0’ implies noise-free. From Fig. 2.3, it is clear that False Alarm arises when a noise-free pixel is identified as noisy and marked as ‘1’ in the noise chart. For any algorithm, a false alarm rate should be low.

2.3.5 Miss detection rate (MD)

Miss detection takes place when a pixel corrupted with noise is identified as noise-free. The difference based evaluation method is used for miss detection also, with the same methodology as shown in Fig. 2.3. For miss detection it is marked as ‘0’ is in the noise chart. Miss detection rates should be low too for optimal results.



136	148	149
100	255	130
177	122	212

(a)noisy image

136	148	149
100	240	130
177	122	212

(b)original image

136	148	149
100	240	130
177	122	212

(a)noisy image

136	148	149
100	240	130
177	122	212

(b)original image

136	148	149
100	255	130
177	122	212

(a)noisy image

136	148	149
100	240	130
177	122	212

(b)original image

0	0	0
0	15	0
0	0	0

(c)difference chart

0	0	0
0	1	0
0	0	0

(d)Correct detection

0	0	0
0	0	0
0	0	0

(c)difference chart

0	0	0
0	1	0
0	0	0

(d)False Alarm

0	0	0
0	15	0
0	0	0

(c)difference chart

0	0	0
0	0	0
0	0	0

(d)Miss Detection

Figure 2.3: Difference based method for FA and MD

Chapter 3

High Density Impulse noise detection using FCM algorithm

3.1 Preliminaries

Impulse noise in images is removed more effectively if the primary detection of noise is performed explicitly. It is one of the main objectives of this thesis work. We have emphasized the proper detection of noise first and then using the noise removal filters. As discussed in the section of non-linear filters, the fusion of median filters with fuzzy logic has evolved in the last decades for the advancement of noise filtering algorithms. Fuzzy logic enhances the performance of the algorithm and therefore we have used Fuzzy C-means (FCM) algorithm to detect impulse noise in an optimum manner and then using the median filter for the restoration of a corrupted pixel. The algorithm is iterative in nature and preserves more image details in a high noise environment. FCM is initially used to cluster the image data. The application of the FCM algorithm in the detection phase provides an exquisite classification of uncorrupted pixels from noisy data, so that the pictorial information remains preserved. Experimental results show that the proposed algorithm abbreviated as HDIND significantly outperforms existing well-known techniques. Results show that with the increase in the percentage of noise density, the performance of the algorithm is not degraded. Furthermore, the varying window size in the two detection stages provides more efficient results in terms of low false alarm rate and miss detection rate. The simple structure of the algorithm to detect impulse noise makes it useful for various applications like satellite imaging, remote sensing, medical imaging diagnosis, and military surveillance. After the efficient detection of noise, the existing filtering techniques can be used for the removal of noise. In this work, we have used the standard median filter to reinstate the values of noisy pixels.

The next section presents a brief study of FCM algorithm. Section 3.3, explains the methodology used followed by experimental results in section 3.4. Conclusions are given at the end of the chapter.

3.2 Fuzzy C-means (FCM) algorithm

Fuzzy c-means is one of the most common methods used to cluster data. In 1973, Dunn developed this method [71] and in 1981, Bezdek revised it [72]. FCM is commonly utilized for pattern recognition. Fuzzy clustering is employed in various algorithms [73,74] for the classification of image data. The main advantage of fuzzy clustering over hard clustering is that it allows each pattern to belong to more than one cluster based on membership function or varying degrees of certainty. Fuzzy C-Means (FCM) is a well-known algorithm among the class of fuzzy clustering algorithms. It provides the best result for the overlapped data set. The basic steps of the FCM algorithm are explained briefly.

The main purpose of FCM is to minimize the objective function J_m:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , 1 \leq m \leq \infty \tag{3.1}$$

Where,

m is the fuzziness index, that should be a real number > 1 (usually 2)

u_{ij} denotes the extent of membership x_i has in j^{th} cluster

x_i represents the i^{th} assessed data

N denotes the number of data

C represents the number of clusters

c_j is the d-dimension center of the cluster, and

$\left\| x_i - c_j \right\|^2$ is the Euclidean distance between i^{th} data and the j^{th} cluster center.

This algorithm assigns membership value to each data point x_i corresponds to each cluster center c_j based on the distance between the cluster center and the data point. This distance is the Euclidean distance among i^{th} data and j^{th} cluster center. This distance decides the membership of data towards the specific cluster. The values of membership u_{ij} and cluster centers c_j are revised in each iteration by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \tag{3.2}$$

Where,

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{3.3}$$

This iteration will stop when $\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$

where ε is a termination criterion between 0 and 1 and k is the iteration number.

The *FCM* algorithm can be summarized as follows:

a) Initialize $U = \left[u_{ij} \right]$ matrix, $U^{(0)}$.

b) At *k*-step: calculate the center's vectors $C^{(k)} = \left[c_j \right]$ with $U^{(k)}$.

c) Update $U^{(k)}$, $U^{(k+1)}$.

d) If $\left\| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right\| < \varepsilon$ then STOP, otherwise return to step (b).

Matlab in-built function is used to perform FCM on image data. It uses an additional argument variable, options, to specify clustering parameters, introduce a stopping criteria, or set the iteration information display. Specify options as a vector:

- options(1): Exponent for the partition matrix U. Default: 2.0.
- options(2): Maximum number of iterations. Default: 100.
- options(3): Minimum amount of improvement. Default: 1e-5.
- options(4): Information displayed during iteration. Default: 1.

The advantage of using fuzzy c-means is that it gives the best result for overlapped data, which is in the form of pixels in images. It is comparatively better than other existing algorithms such as a k-means algorithm. FCM helps us to compute the threshold values to classify pixels as lower intensity pixels , higher intensity pixels and noise free pixels. This increases the efficiency of noise detection filter and improves the qualitative and quantitative results.

In this technique, FCM is incorporated in both the detection stages. The proposed method is presented in the next section.

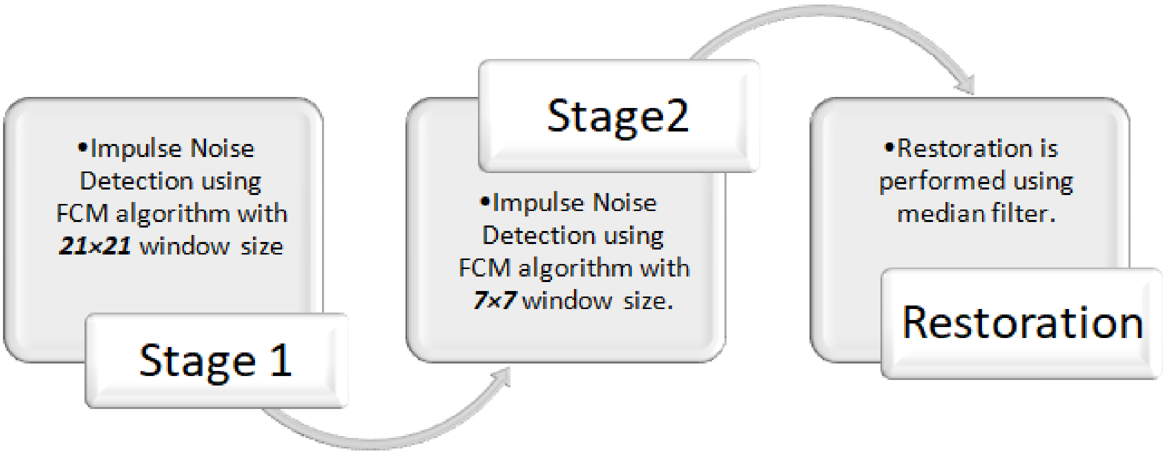

•Impulse Noise Detection using FCM algorithm with ***21×21*** window size
Stage 1
Stage2
•Impulse Noise Detection using FCM algorithm with ***7×7*** window size.
•Restoration is performed using median filter.
Restoration

Figure 3.1: Flow of HDIND

3.3 Methodology

The effective impulse noise removal primarily relies upon the detection mechanism.

The proposed algorithm introduces a strong detection method to identify the noisy pixels of an image which in turn reduces the miss detection and false alarm rates.

The pixels with higher and lower intensity are classified separately. The remaining pixels are categorized as noise-free.

In this technique for noise removal, the detection of impulse noise is carried out in two stages. Double stage detection efficiently locates the noise affected pixels and does not alter the value of noise-free pixels.

Stage I:

1) Center a 21 × 21 size window on every image pixel.
2) Pixel under inspection is denoted as *w(s,t).*
3) Let the three clusters formed after applying FCM be A, B, and C. After the formation of clusters find the maximum value present in each cluster respectively.
4) FCM classifies the adjacent 441 pixels unsupervisely into three clusters.
5) Compute the highest intensity value of the pixel in each cluster respectively, denoted as M1 , M2 and M3 such that M1 < M2 < M3.
6) Using the following equation, check whether the central pixel $w(s,t)$ is noisy or noise-free.

$$w(s,t)=\left\{\begin{matrix}\text{pepper noise ,} & \mathrm{w}(s,t)\le M1\\ \text{noise free ,} & \mathrm{M1<w}(s,t)\le M2\\ \text{salt noise ,} & \mathrm{M2<w}(s,t)\le M3\end{matrix}\right\} \qquad (3.4)$$

Cluster with the highest intensity value as *M1* is labeled as "*pepper noise*", the group with *M2* as its highest value is termed as "*noise-free*" and *M3* belongs to "*salt-noise*" group.

7) In case the pixel belongs to the “noise-free” group, it is not processed further and its original intensity value is retained otherwise, the pixel is evaluated again in the second stage of detection.

Stage II:

8) Window size is altered to 7 × 7.
9) Reiterate the Steps 2) to 6) in the same way.
10) If *w(s,t)* falls into the “*noise-free*” group, it is considered as an uncorrupted pixel and no transition is made in its original value otherwise,the noisy pixel is restored.

The median value computed via 7 × 7 window size, is utilized for the restoration of the corrupted pixel values.

Illustration of algorithm

For a better understanding of the proposed algorithm, an illustration of the work is presented as follows:

Instead of using a window size of 21×21 in the first detection Stage, working of the algorithm explained using a window size of 7×7.

All the 49 values of the window of size 7×7 are sorted in ascending order and using the FCM algorithm they are divided into three clusters. Let the three clusters be A, B, and C.

49	*87*	*155*	*255*	*54*	*64*	*81*
24	*255*	*132*	*163*	*0*	*255*	*0*
46	*39*	*145*	*0*	*156*	*119*	*0*
87	*46*	*141*	*0*	*155*	*255*	*117*
113	*104*	*77*	*125*	*0*	*136*	*116*
0	*0*	*99*	*72*	*119*	*255*	*255*
0	*0*	*255*	*0*	*97*	*54*	*29*

49	*87*	*155*	*255*	*54*	*64*	*81*
24	*255*	*132*	*163*	*0*	*255*	*0*
46	*39*	*145*	*0*	*156*	*119*	*0*
87	*46*	*141*	*87*	*155*	*255*	*117*
113	*104*	*77*	*125*	*0*	*136*	*116*
0	*0*	*99*	*72*	*119*	*255*	*255*
0	*0*	*255*	*0*	*97*	*54*	*29*

Figure 3.2. (a) 7 × 7 window (b) Restored value of the noisy pixel

Cluster A = {72, 77, 81, 87, 87, 97, 99, 104, 113, 116, 117, 119, 119, 125, 132, 136, 141,145, 155, 155, 156, 163}

Cluster B = {255, 255, 255, 255, 255, 255, 255}

Cluster C = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 29, 39, 46, 46, 49, 54, 54, 64}

The maximum value for each cluster is computed and sorted in ascending order such that *M1< M2< M3:*

M1= 64, M2=163, M3= 255

From Fig. 1, Pixel under consideration is $w(s,t)=0$. Using equation (3.4),

$$w(s,t)=\begin{Bmatrix}\text{pepper noise,} & \mathrm{w}(s,t)\le 64\\ \text{noise free ,} & 64<\mathrm{w}(s,t)\le 163\\ \text{salt noise ,} & 163<\mathrm{w}(s,t)\le 255\end{Bmatrix} \qquad (3.5)$$

From equation (3.5), it is clear that *0* belongs to the cluster having *Pepper noise*. Therefore, it will be evaluated further in second stage of detection, by using a window of size 7×7 in the same manner as illustrated above. If the pixel is again detected as noisy, it will be restored using the median filter.

After arranging all the pixel values of the 7×7 window in ascending order, Median value is computed as follows:

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 29, 39, 46, 46, 49, 54, 54, 64, 72, 77, 81, 87, **87**, 97, 99,104, 113, 116, 117, 119, 119, 125, 132, 136, 141, 145, 155, 155, 156, 163, 255, 255, 255,255, 255, 255, 255}

The median value of the selected window of size 7×7 is computed as 87. For the restoration of the pixel value in Fig. 3.2, the central pixel 0 is replaced by a computed median value 87.

The selection of the restoration method can be performed according to the application of images using several methods present in the literature. For example, the value of detected noisy pixel value “0” in Fig. 3.2, is restored by the mean filter as “97”.

In the future, the work can be expanded to the better restoration of pixels after efficient detection using the proposed algorithm.

Experimental results and comparison with existing techniques exhibit that the proposed detection algorithm detects noise more effectively. It even outperforms the techniques such as Contrast enhancement based Filter (CEB) [55], MDBUTM [56], A new adaptive switching median (ASWM) filter [57], Detail-preserving variation algorithm with decision-based structure for impulse noise [75], Impulse noise removal at high and low levels using nonlinear filter [76], Fuzzy mathematical morphology based impulse noise suppression (FMM) [77], etc.

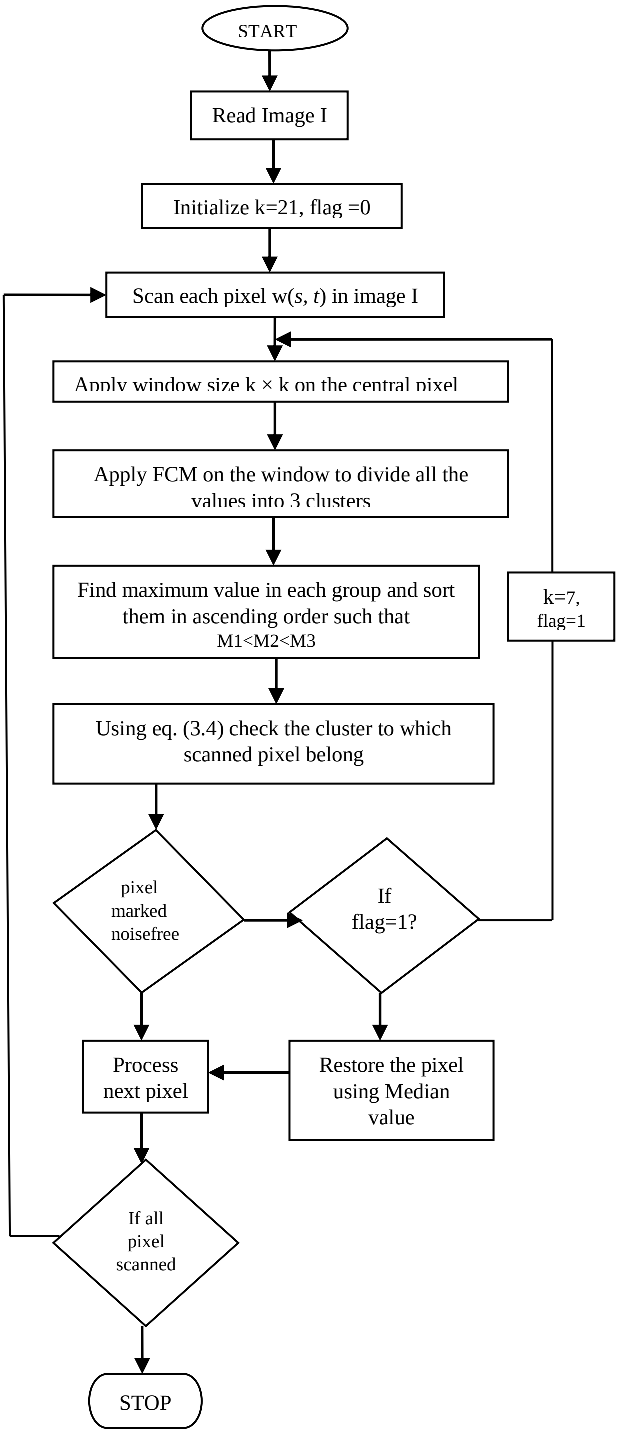

START
Read Image I
Initialize k=21, flag =0
Scan each pixel w(s, t) in image I
Apply window size k × k on the central pixel
Apply FCM on the window to divide all the values into 3 clusters
Find maximum value in each group and sort them in ascending order such that M1<M2<M3
k=7, flag=1
Using eq. (3.4) check the cluster to which scanned pixel belong
pixel marked noisefree
If flag=1?
Process next pixel
Restore the pixel using Median value
If all pixel scanned
STOP

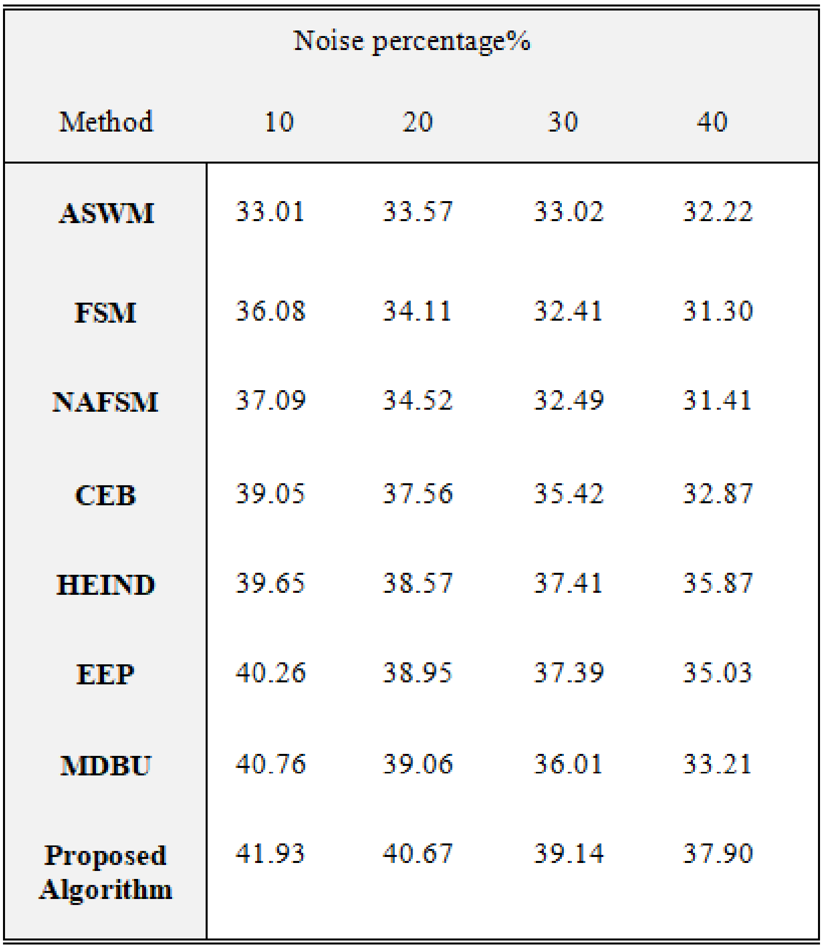
Figure 3.3: Algorithm flowchart

The algorithm flowchart is given in Fig. 3.3.

3.4 Experimental Results

This section analyzes our work in comparison with some existing techniques to demonstrate its effectiveness. For the performance analysis, two similarity measures PSNR and MSE that is Peak Signal to Noise Ratio and Mean square error are selected. Lower MSE values provide better results. Higher the value of PSNR better is the similarity between the original image and the restored image. The proposed algorithm is compared with the few existing techniques in Table 3.1 & 3.2 for PSNR and MSE values. The values are consistent with increasing noise density and it outperforms other methods. It is derived that HDIND is a novel filter for images altered with high noise.

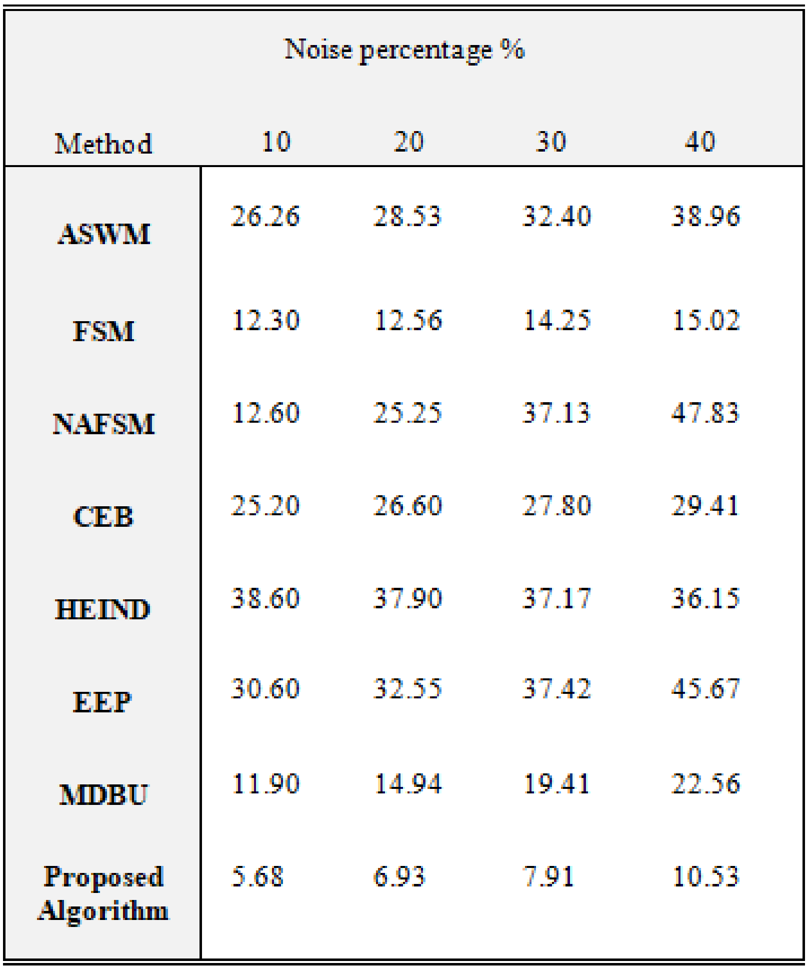
TABLE 3.1
COMPARISON OF PSNR VALUES FOR "*Lena*" IMAGE



	Noise percentage%			
Method	10	20	30	40
ASWM	33.01	33.57	33.02	32.22
FSM	36.08	34.11	32.41	31.30
NAFSM	37.09	34.52	32.49	31.41
CEB	39.05	37.56	35.42	32.87
HEIND	39.65	38.57	37.41	35.87
EEP	40.26	38.95	37.39	35.03
MDBU	40.76	39.06	36.01	33.21
Proposed Algorithm	41.93	40.67	39.14	37.90

TABLE 3.2

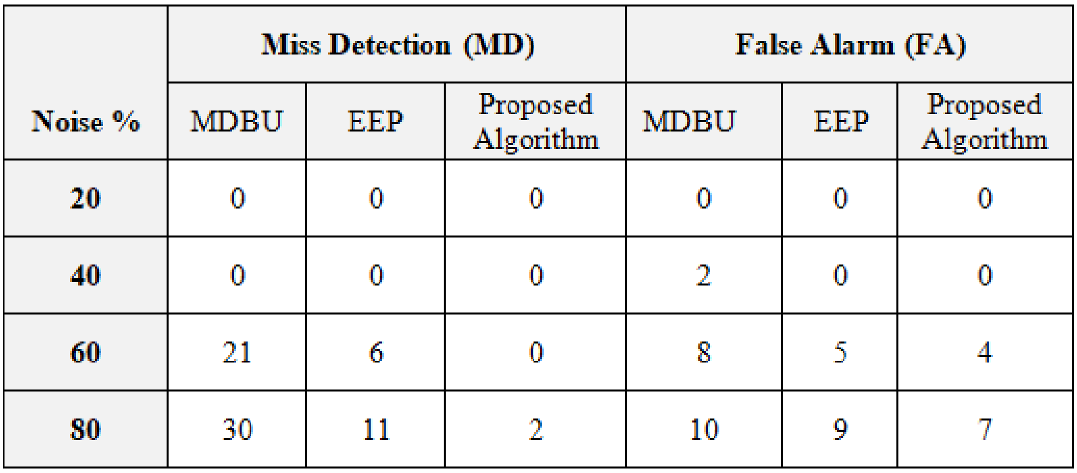
COMPARISON OF MSE VALUES FOR "*Lena*" IMAGE



	Noise percentage %			
Method	10	20	30	40
ASWM	26.26	28.53	32.40	38.96
FSM	12.30	12.56	14.25	15.02
NAFSM	12.60	25.25	37.13	47.83
CEB	25.20	26.60	27.80	29.41
HEIND	38.60	37.90	37.17	36.15
EEP	30.60	32.55	37.42	45.67
MDBU	11.90	14.94	19.41	22.56
Proposed Algorithm	5.68	6.93	7.91	10.53

TABLE 3.3

COMPARISON OF MD & FA VALUES FOR "*Lena*" IMAGE



	Miss Detection (MD)			**False Alarm (FA)**		
Noise %	MDBU	EEP	Proposed Algorithm	MDBU	EEP	Proposed Algorithm
20	0	0	0	0	0	0
40	0	0	0	2	0	0
60	21	6	0	8	5	4
80	30	11	2	10	9	7

Miss detection (MD) and False alarm (FA) rates are computed for more reliability of the proposed algorithm(PA). As discussed in section 2.3, the difference based method is used for the computation of MD and FA rates. Table 3.3 provides the MD and FA values for the "*Lena*" image. The algorithm provides low MD and FA rates which increases the quality of the proposed filter.

The advantage of the proposed algorithm is that its performance is not degraded with the increasing noise level. It can easily handle high noise levels up to 80%. The application of FCM helps to classify the salt and pepper noisy pixels separately in an efficient manner. This, in turn, improves our detection process. The algorithm utilizes the median value of the selected window to restore detected noisy pixels. It gives low miss detection rates as well as false alarm rates as shown in Table 3.3. The feasibility of the proposed algorithm is also observed from visual results at various noise levels. To assess the image results, the standard grayscale test image "*Lena*" of size 512× 512 is used.

In Fig. 3.4, using MATLAB Results of different filters, including the proposed algorithm, is presented at a noise level of 30%. From the visual results, it is also noticed that the nature of the restored image using our technique is better as compared to the quality of restored images using other existing algorithms. Our algorithm preserves more edge details and fine details present in the image.

From Fig. 3.5, it is derived that our algorithm performance is not degraded with an increasing percentage of noise density. For the better reconstruction of the image, it provides high PSNR values and lowers MSE values as compared to existing techniques in literature at different noise densities. Some more results are incorporated for image "*Mandril*" in Fig. 3.6, "*Pirate*" in Fig. 3.7, "*Woman Blonde*" in Fig. 3.8, and "*livingroom*" in Fig. 3.9, at noise levels of 30%, 40% & 50% respectively.

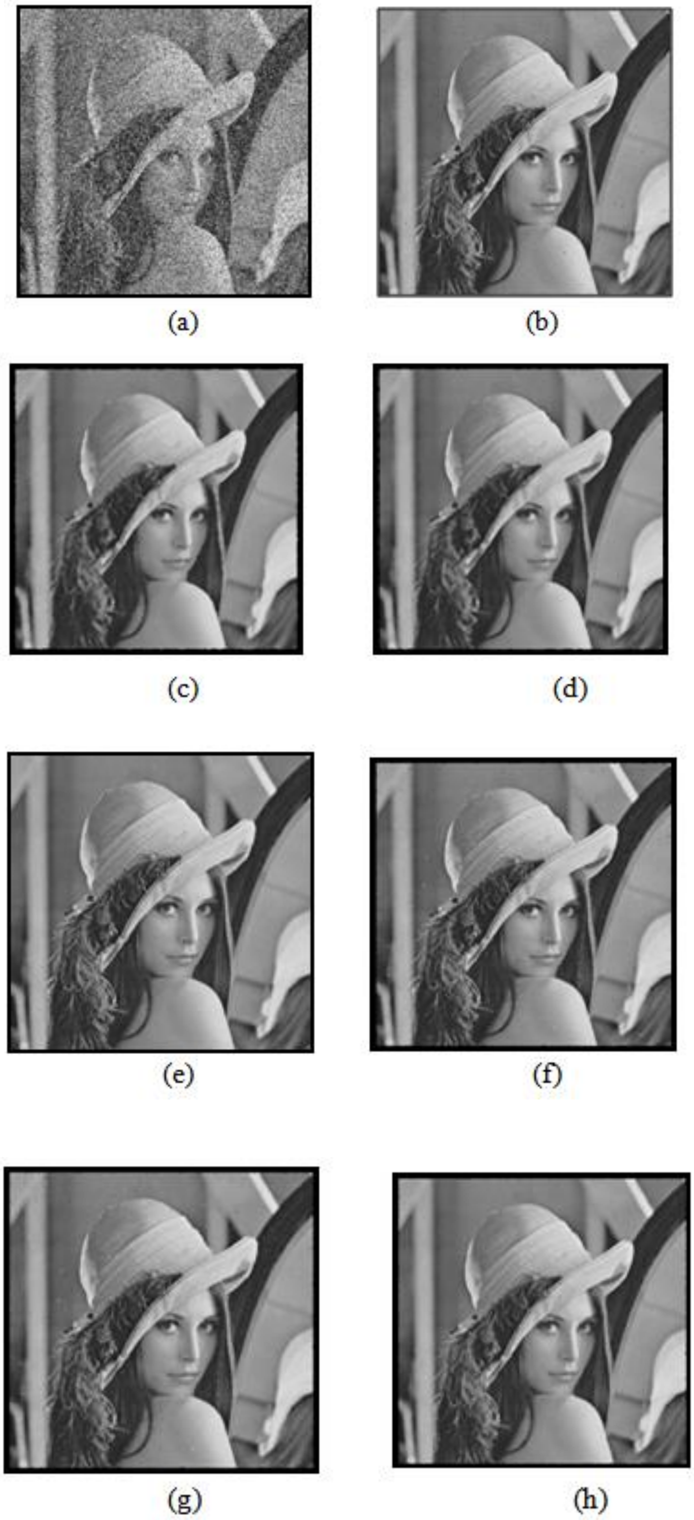


Figure 3.4 : Results for image “*lena*” using various methods (a) 30% Noisy image (b) ASWM (c) HEIND (d) EEP (e) MDBU (f) CEB (g) NAFSM (h) Proposed algorithm

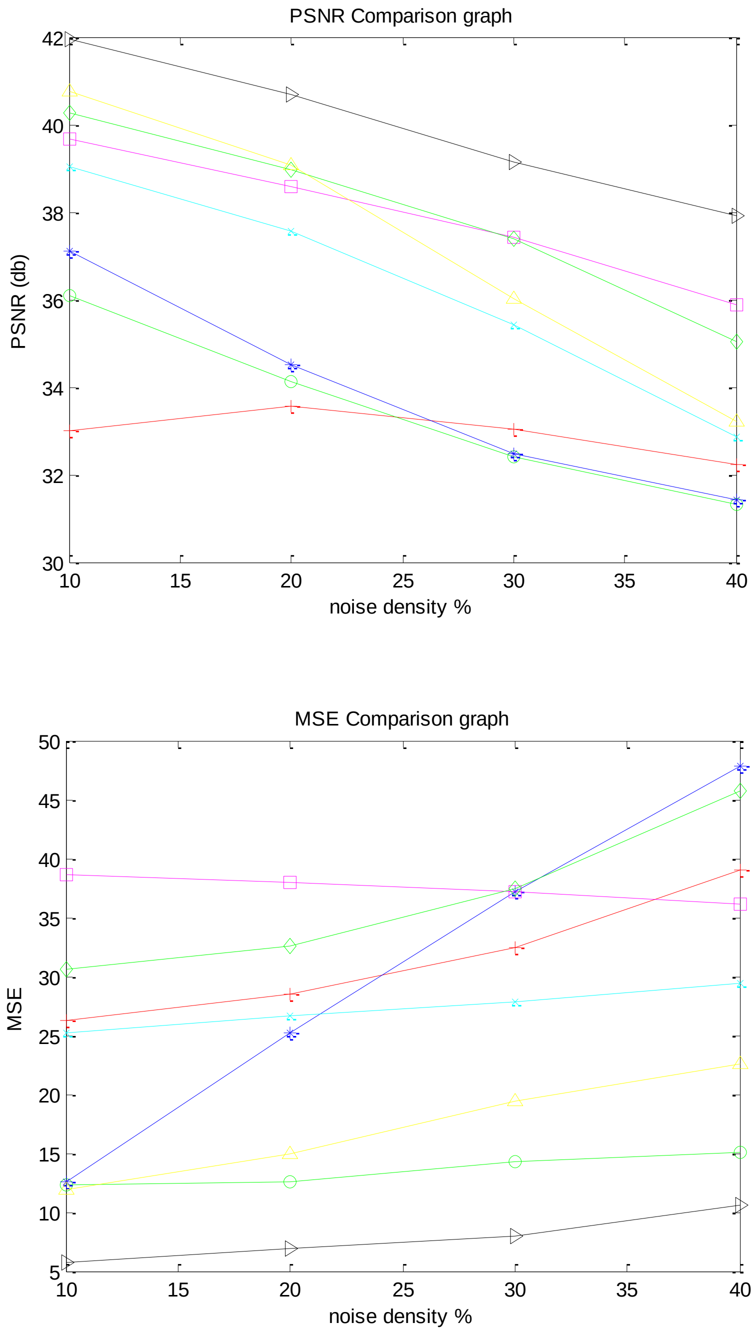


Figure 3.5 : PSNR and MSE graphs for 30% noisy image "*lena*" using various methods

Where,

Red (+) = ASWM , Green (o) =FSM, Blue (*) =NAFSM , Cyan (×) =CEF

Magenta (□) =HEIND ,Green (◊) =EEPA,Yellow (Δ) = MDBU,

Black (▶) = PROPOSED ALGORITHM

The PSNR and MSE values for different images after applying the proposed algorithm are presented in Table 3.4.

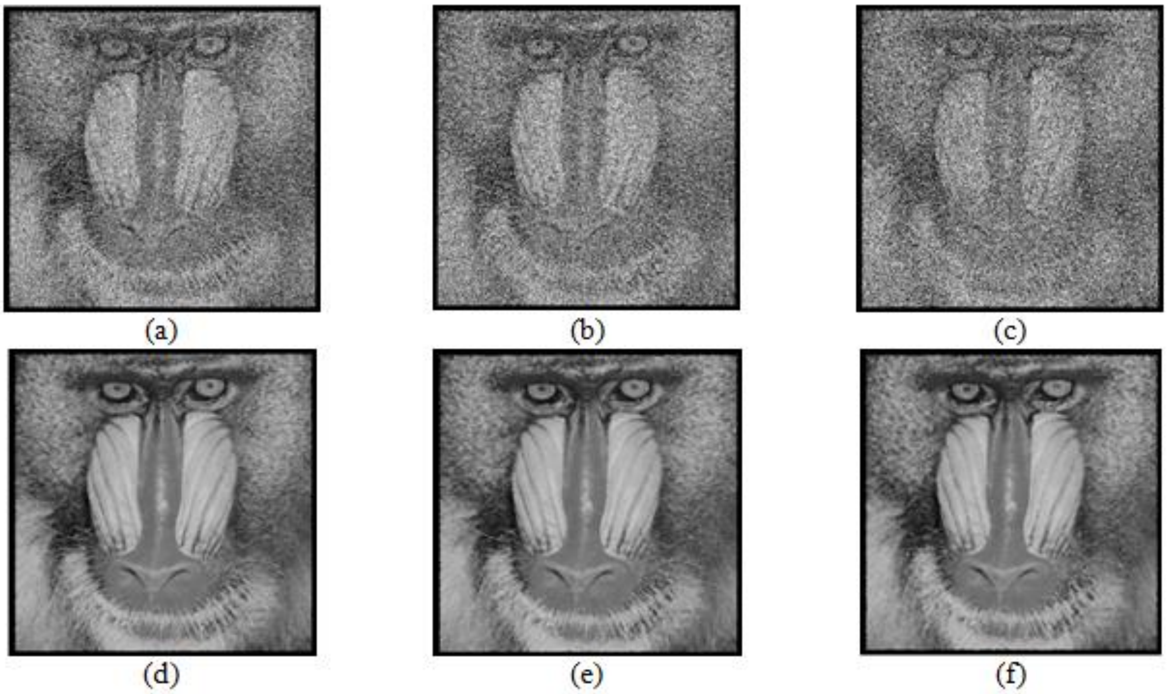
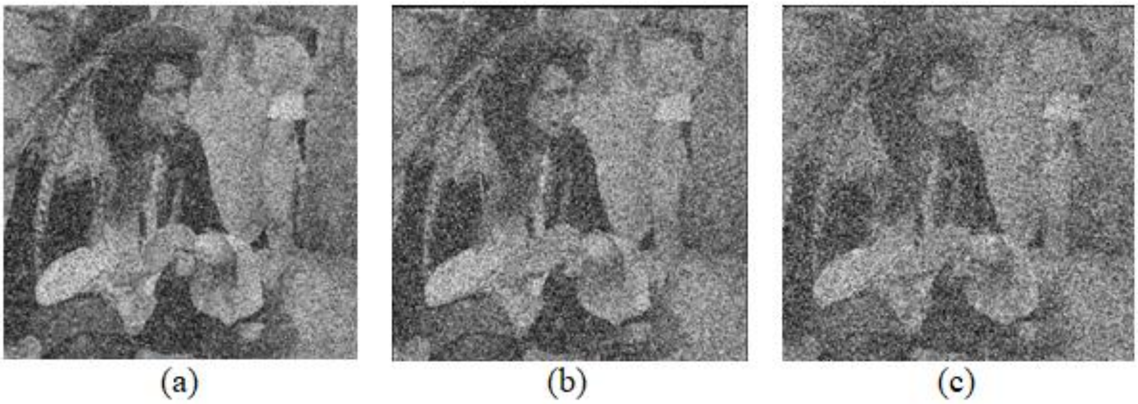


Figure 3.6 : Results for "*Mandril*" image (a) Image with 30% noise (b) Image with 40% noise (c) Image with 50% noise (d) Removed noise image (30%) (e) Removed noise image (40%) (f) Removed noise image (50%)



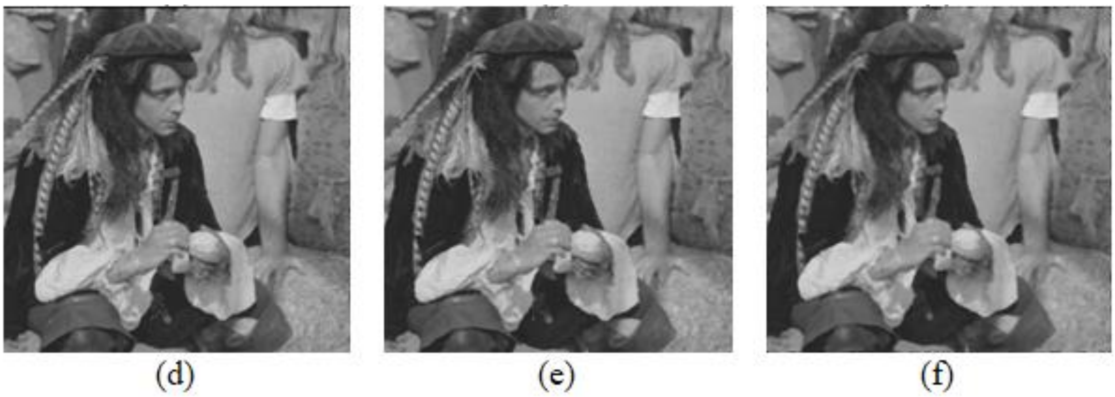


Figure 3.7 : Results for "*Pirate*" image (a) Image with 30% noise (b) Image with 40% noise (c) Image with 50% noise (d) Removed noise image (30%) (e) Removed noise image (40%) (f) Removed noise image (50%)

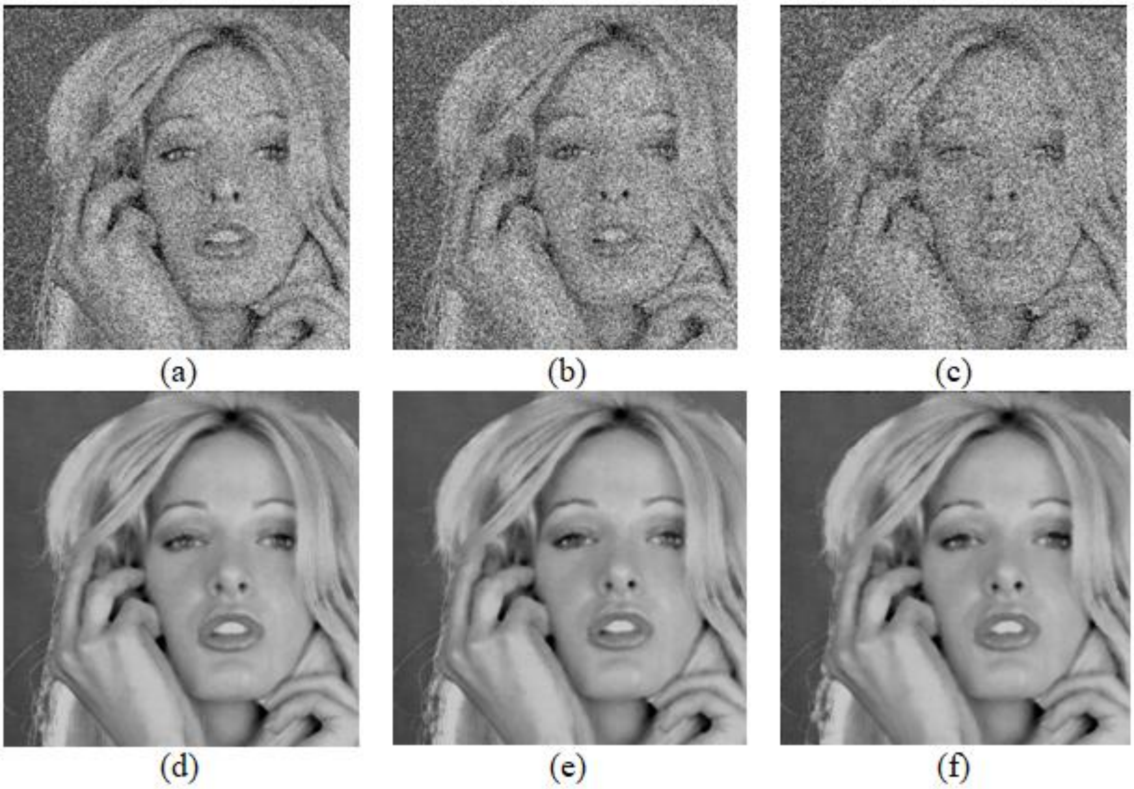


Figure 3.8 : Results for "*Woman Blonde*" image (a) Image with 30% noise (b) Image with 40% noise (c) Image with 50% noise (d) Removed noise image (30%) (e) Removed noise image (40%) (f) Removed noise image (50%)

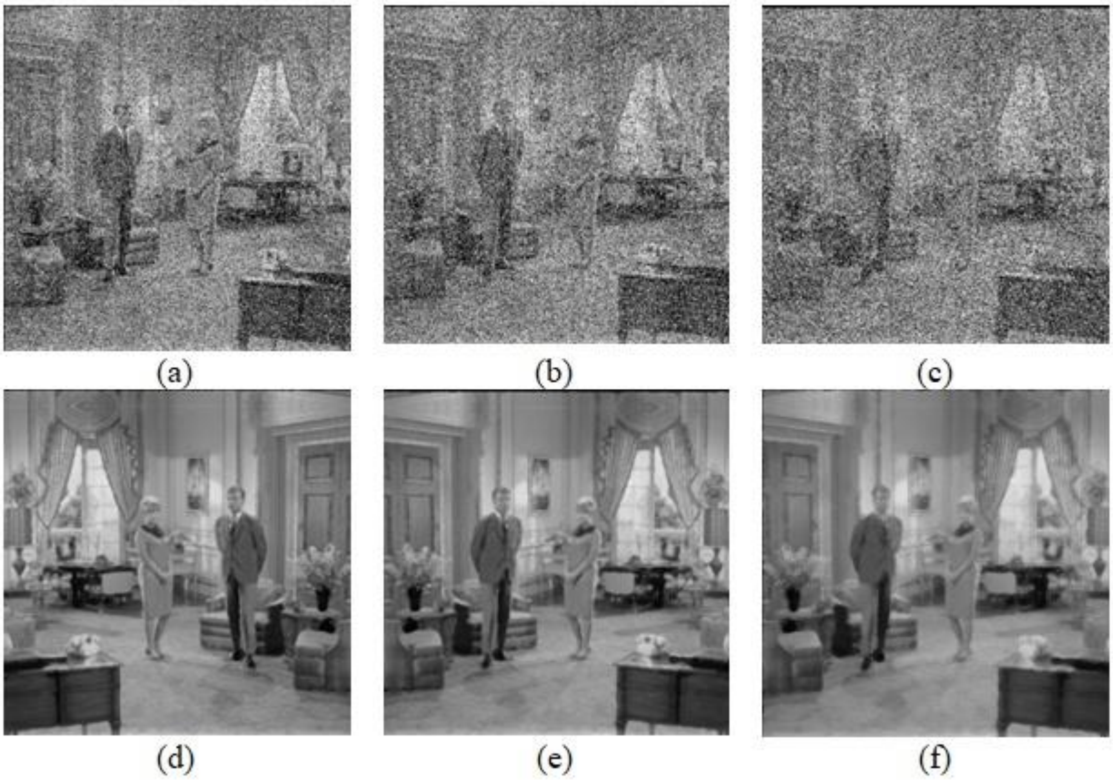


Figure 3.9 : Results for "*Livingroom*" image (a) Image with 30% noise (b) Image with 40% noise (c) Image with 50% noise (d) Removed noise image (30%) (e) Removed noise image (40%) (f) Removed noise image (50%)

TABLE 3.4

PSNR & MSE VALUES FOR DIFFERENT IMAGES USING PROPOSED ALGORITHM

Noise percentage%	30		40		50	
Image	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**
Mandril	34.68	22.11	33.52	28.89	32.48	36.66
Pirate	37.59	11.32	36.35	15.04	34.84	21.29
Woman blonde	37.91	10.34	36.92	13.19	35.89	16.89
Living room	36.59	14.24	35.46	18.49	33.96	26.09

3.5 Conclusions

This chapter introduced an effective detection scheme for impulse noise. The performance of the algorithm is not degraded with the increasing percentage of noise density. The detection phase is divided into two stages. It is iterative and minimizes the false alarm rate as well as zero miss detection rates. The application of fuzzy c-means (FCM) to detect the noisy pixels makes this algorithm a novel technique. Extensive simulation results and comparison with other filters indicate that HDIND surpasses several methods. After efficient detection of noise, any restoration technique present in the literature can be incorporated. It is easy to understand as it has an uncomplicated structure and intuitive in nature.

In the future, the work can be expanded by incorporating better restoration techniques along with the efficient detection scheme presented in this chapter. The next chapter presents an extension of this work by using a more robust framework for the restoration of noisy pixels.

Chapter 4

Edge Preserving Fuzzy Filter for Suppression of Impulse Noise in Images

4.1. Premiliaries

Impulse noise in images culminates in the loss of valuable information. This chapter introduces an efficient fuzzy filter for suppression of impulse noise in images (EFFSIN). In the previous chapter, HDIND [78] recognized the significance of proper detection of noise and its subsequent restoration. But even with an impressive detection procedure, HDIND is not able to classify the edge pixels properly. The motivation behind this work was to classify the edge pixel and noisy pixel before restoration of pixel, so that we don't have to compromise on edge information. The algorithm employs an exclusive image restoration method to perform the cross-check on pixels.To enhance the performance of the filter, an additional set of rules is used prior to the final restoration of the noisy pixel. Firstly, impulse noise is effectively detected by the proposed algorithm with the utilization of Fuzzy C-Means (FCM) in two stages. Only legitimate pixels are restored after the fair distinction of the noisy pixel from edge pixel. The algorithm is presented in detail in the subsequent section. Proper restoration of the noisy pixel, following effective detection by EFFSIN leads to low miss detection (MD) and false alarm (FA) rates. Simulation results depict the efficacy of the proposed filter regarding peak-signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

4.2. EFFSIN Algorithm

The basic characteristics of an image are impaired with the intrusion of impulse noise. Impulse noise is an alias as Salt and pepper noise in literature. The original pixel values are altered by the extreme intensity values of the dynamic range usually 0 or 255. The proposed work is dynamic in nature which significantly removes impulse noise from images without compromising on the existing edge details. Even if the pixels are detected as noisy, the algorithm performs restoration after checking whether the identified noisy pixel is impulse noise or an edge value whose true value is *0* or *255*.

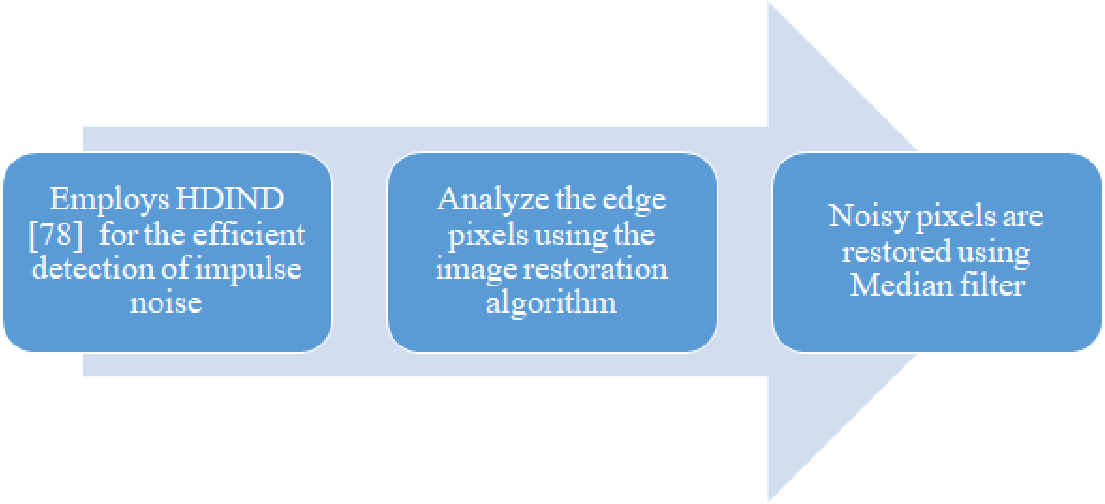

Employs HDIND [78] for the efficient detection of impulse noise
Analyze the edge pixels using the image restoration algorithm
Noisy pixels are restored using Median filter

Figure 4.1: Flow of EFFSIN

4.2.1 Impulse Noise Detection

The proposed algorithm performs impulse noise detection in two phases. The initial phase attempts to detect noise on a global basis therefore a bigger window size of 21×21 is used to extract relevant information from a noisy image.

Phase I

1. Center a 21×21 size window on every image pixel.
2. Pixel under inspection is referred to as $v(l^i,l^j)$ i.e value at i^{th} location and j^{th} location in the spatial domain.
3. *FCM* classifies the adjacent *441* pixels unsupervisely into three clusters.
4. Compute the highest intensity value of the pixel in each cluster respectively, denoted as H_1, H_2 and H_3 such that $H_1<H_2<H_3$.
5. Cluster with the highest intensity value as H_1 is labeled as "*pepper noise*", the group with H_2 as its highest value is termed as "*noise-free*" and H_3 belongs to "*salt-noise*" group. The following equation is used to evaluate that the pixel is affected by impulse noise or not.

$$v(l^i,l^j) = \left\{ \begin{array}{ll} pepper-noise, & \mathrm{v}(l^i,l^j) \le H_1 \\ noise-free, & H_1 < v(l^i,l^j) \le H_2 \\ salt-noise, & H_2 < v(l^i,l^j) \le H_3 \end{array} \right\} \tag{4.1}$$

6. In case the pixel belongs to the "*noise-free*" group, it is not processed further and its original intensity value is retained otherwise, the pixel is evaluated again in the second phase of detection.

Phase II

1) For the analysis of the noisy pixel on a local basis, the window size is altered to 7×7.
2) *FCM* classifies the adjacent *49* pixels unsupervisely into three clusters.
3) Reiterate the steps 4) and 5) in a similar manner.
4) If $v(l^i,l^j)$ falls into the "*noise-free*" group, it is considered as an uncorrupted pixel and no transition is made in its original value otherwise, the noisy pixel is restored.

4.2.2 Impulse Noise Restoration

An image is comprised of many fine details and sometimes the true value of a pixel or edge values is identified as noise. To minimize the false alarm and miss detection rates, it is important to differentiate between an edge pixel and noisy pixel. The double-phase analysis of a pixel by the proposed algorithm helps to identify the noise affected pixels more robustly.EFFSIN also focuses on proper restoration of noisy pixels such that edges are preserved and a high-quality image is perceived at the user end. For proper classification of noisy pixels from edge pixels, the following steps are performed at the restoration stage:

1. For a noisy pixel, consider the clusters generated by FCM with a 7×7 window, having the highest intensity value of H_1 and H_3 respectively.
2. In each cluster, count the number of *0* and *255* respectively and its value is stored in a function denoted by "*COUNT1"* for the number of *0* in case of "*pepper-noise*" and "*COUNT2"* for the number of *255* in case of "*salt-noise*".
3. If $COUNT1 \leq 2$ *or* $COUNT2 \leq 2$ then the pixel under evaluation is considered as noisy and will be restored.
4. If $COUNT1 \geq 3$ *or* $COUNT2 \geq 3$ then take a 3 × 3 window placed on processing pixel to check if its a noisy pixel or an edge.
5. The following two cases are considered for image restoration by taking the eight adjacent pixels of the center pixel i.e N_8 into account :

Case I: If N_8 has all *0* or *255* then do not perform any restoration as it can be an edge value.

Case II: If N8 has *0* or *255* greater than one then expand window size to 5 × 5 to check for any possible edge

v^{24}	v^{9}	v^{10}	v^{11}	v^{12}
v^{23}	v^{8}	v^{1}	v^{2}	v^{13}
v^{22}	v^{7}	v^{0}	v^{3}	v^{14}
v^{21}	v^{6}	v^{5}	v^{4}	v^{15}
v^{20}	v^{19}	v^{18}	v^{17}	v^{16}

Figure 4.2: 5 × 5 window

Let the pixel under processing is “v^0” as shown in Fig. 4.2. The adjacent pixels are denoted as v^1 to v^{24}.

The following rules are utilized to cross-check whether the pixel needs to be restored or not.

Rule 1:

IF *(*v^0 *is* ***NOISY) AND (***v^2 *and* v^{12} *are* ***NOISY) AND (***v^6 *and* v^{20} *are* ***NOISY) THEN*** *(*v^0 *is* ***EDGE)***

Rule 2:

IF *(*v^0 *is* ***NOISY) AND (***v^1 *and* v^{10} *are* ***NOISY) AND (***v^5 *and* v^{18} *are* ***NOISY) THEN*** *(*v^0 *is* ***EDGE)***

Rule 3:

IF *(*v^0 *is* ***NOISY) AND (***v^8 *and* v^{24} *are* ***NOISY) AND (***v^4 *and* v^{16} *are* ***NOISY) THEN*** *(*v^0 *is* ***EDGE)***

Rule 4:

***IF** (v^0 is **NOISY**) **AND** (v^7 and v^{22} are **NOISY**) **AND** (v^3 and v^{14} are **NOISY**) **THEN** (v^0 is **EDGE**)*

"*NOISY*" indicates the pixel value as *0* or *255*. Noisy Pixel value should be either *0* in all cases or *255* during the complete analysis of pixel with the above-mentioned rules. Edges indicate the significant regional transitions of intensity values in an image. Constant occurrence of noisy pixels with the same intensity value of either 0 or 255 implies the presence of a possible edge.

6. Pixel evaluated as "*NOISY*" via any of the rules stated in the preceding step is replaced with the median value of the 5×5 window and pixel evaluated as "*EDGE*" is not modified.

The rules presented in the work employ the idea to explore the nearest neighbourhood of the pixel in all possible directions ,to classify an edge pixel from noisy pixel , so that the information is well preserved and false alarm rates are reduced. With the reduction in false alarm rates the PSNR values also improve and results become more efficient.The separation of noisy pixels from possible edge pixels lowers the False Alarm rates and Miss detection rates by the proposed algorithm. It also increases the PSNR and SSIM values. The standard median filter is a well known powerful filter in the area of image processing and is used for various applications including impulse noise removal. If used directly it can lead to loss of fine details/edges present in an image.EFFSIN is a sturdy technique that uses the median filter after the analysis of impulse noise in an image that enhances its performance.

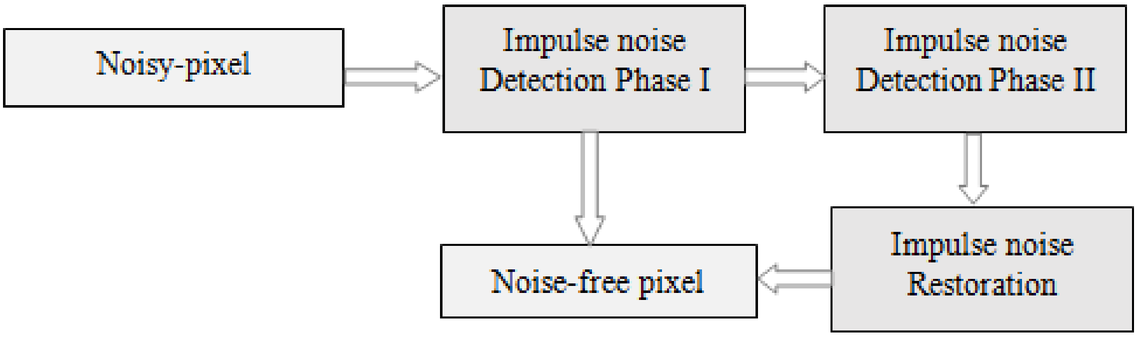
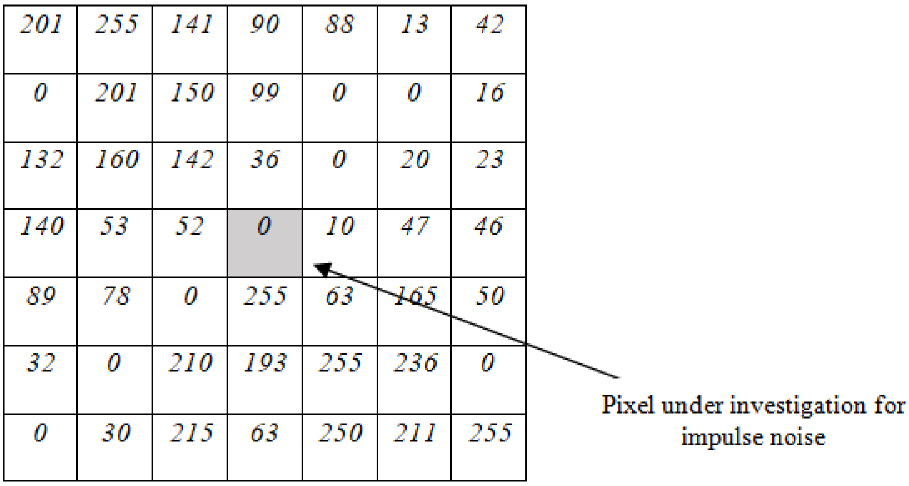

Noisy-pixel
Impulse noise Detection Phase I
Impulse noise Detection Phase II
Noise-free pixel
Impulse noise Restoration

Figure 4.3: Block Diagram of EFFSIN

4.2.3 Illustration of algorithm

An exemplification of the proposed work is presented for more clear perception.

A 7×7 window is used for demonstration of work instead of a 21 ×21 window for the first impulse noise detection phase for a crisp view. *FCM* classifies the adjacent 49 pixels unsupervisely in three clusters, let's say *Q*, *R*, and *S*.



201	255	141	90	88	13	42
0	201	150	99	0	0	16
132	160	142	36	0	20	23
140	53	52	0	10	47	46
89	78	0	255	63	165	50
32	0	210	193	255	236	0
0	30	215	63	250	211	255

Pixel under investigation for impulse noise

Figure 4.4: 7×7 window

Q={193,201,201,210,211,215,236,250,255,255,255,255}

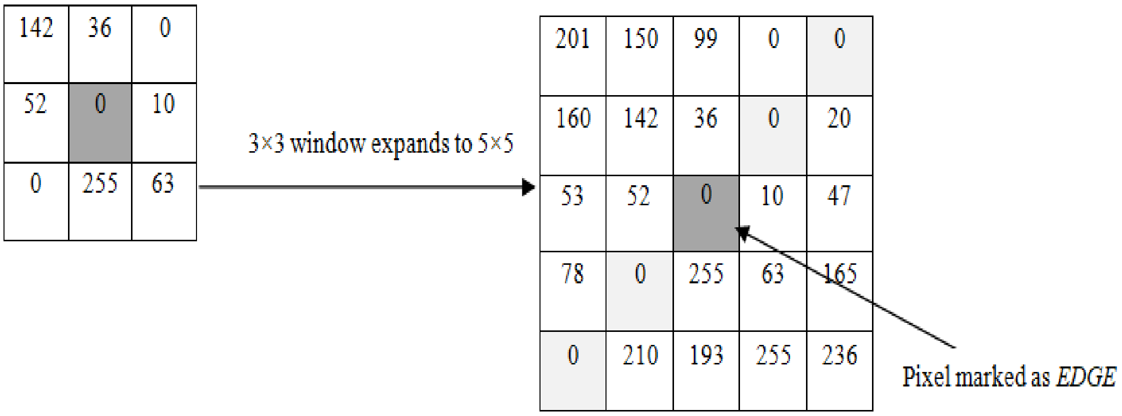
R={78,88,89,90,99,132,140,141,142,150,160,165}

S={0,0,0,0,0,0,0,0,0,10,13,16,20,23,30,32,36,42,46,47,50,52,53,63,63}

The highest value from each cluster is calculated and labeled as H_1, H_2 and H_3 such that $H_1<H_2<H_3$.Substitute the values of $H_1=63$, H_2 =165 and $H_3=255$ in equation (4.1).

$$v(l^i,l^j)=\left\{\begin{array}{c} pepper-noise,\quad \mathrm{v}(l^i,l^j)\le 63 \\ noise-free,\quad 63<v(l^i,l^j)\le 165 \\ salt-noise,\quad 165<v(l^i,l^j)\le 255 \end{array}\right\} \tag{4.2}$$

It is depicted from the above equation that the pixel under processing is "*pepper-noise*".Similarly, this pixel is again examined in the second detection phase with a window size of 7×7, for the presence of impulse noise. Likewise, it is again discovered as a pixel corrupted with noise in next phase. For impulse noise restoration, consider the clusters with the highest value of $H_1=63$ and $H_3=255$. The pixel is marked as "*pepper-noise*", therefore "*COUNT1"* is computed. *COUNT1=9*, in this case, *COUNT1* ≥ *3*, so 3×3 is centered on noisy pixels. According to *step (5)-case II* of impulse noise restoration, the window is expanded to 5×5.



142	36	0
52	0	10
0	255	63

201	150	99	0	0
160	142	36	0	20
53	52	0	10	47
78	0	255	63	165
0	210	193	255	236

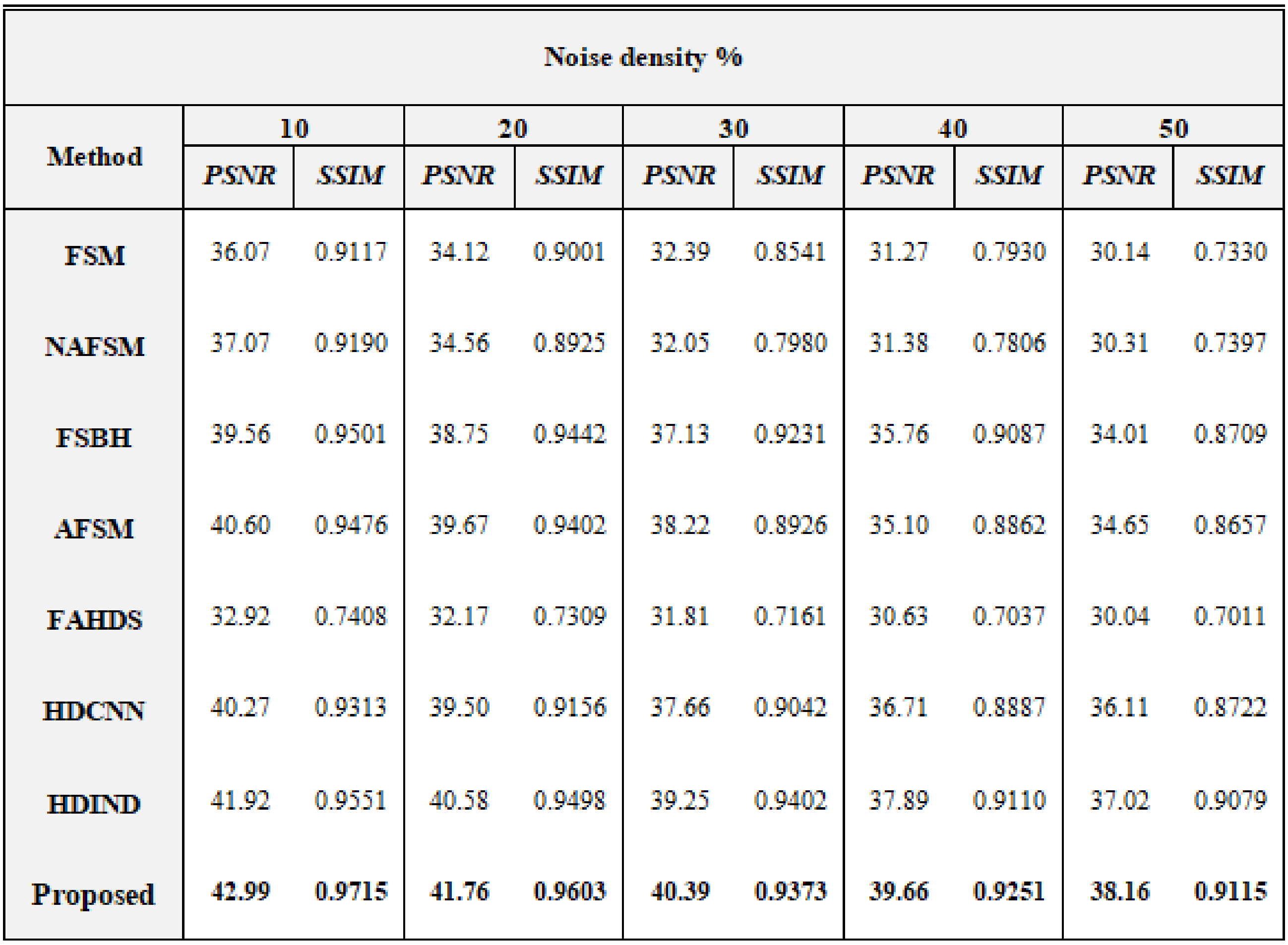
Figure 4.5: *Expansion of window size and the final result*

Analysis for any possible edge is performed using *Rule 1,2,3 and 4*.Rule 1 is true for present pixel therefore, it is marked as *EDGE* and no restoration is done.

4.3 Experimental results

EFFSIN is simulated for impulse noise of varying densities as well as examined in contrast with other state-of-art filters present in literature. The similarity measures employed for the performance evaluation of our work are Peak signal-to-noise ratio(*PSNR*), structural similarity index (*SSIM*), False alarm (FA) and Miss detection (MD) values.

TABLE 4.1: PSNR and SSIM results for image "*Lena*"



Method	Noise density %									
	10		20		30		40		50	
	PSNR	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*
FSM	36.07	0.9117	34.12	0.9001	32.39	0.8541	31.27	0.7930	30.14	0.7330
NAFSM	37.07	0.9190	34.56	0.8925	32.05	0.7980	31.38	0.7806	30.31	0.7397
FSBH	39.56	0.9501	38.75	0.9442	37.13	0.9231	35.76	0.9087	34.01	0.8709
AFSM	40.60	0.9476	39.67	0.9402	38.22	0.8926	35.10	0.8862	34.65	0.8657
FAHDS	32.92	0.7408	32.17	0.7309	31.81	0.7161	30.63	0.7037	30.04	0.7011
HDCNN	40.27	0.9313	39.50	0.9156	37.66	0.9042	36.71	0.8887	36.11	0.8722
HDIND	41.92	0.9551	40.58	0.9498	39.25	0.9402	37.89	0.9110	37.02	0.9079
Proposed	**42.99**	**0.9715**	**41.76**	**0.9603**	**40.39**	**0.9373**	**39.66**	**0.9251**	**38.16**	**0.9115**

TABLE 4.2: PSNR and SSIM results for various images using the Proposed filter

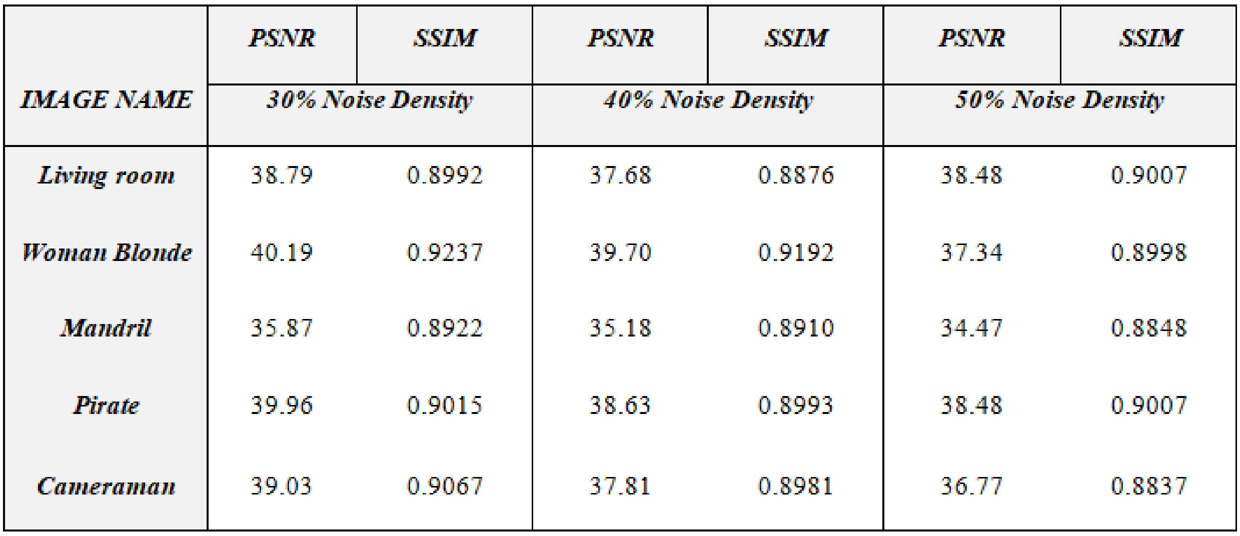
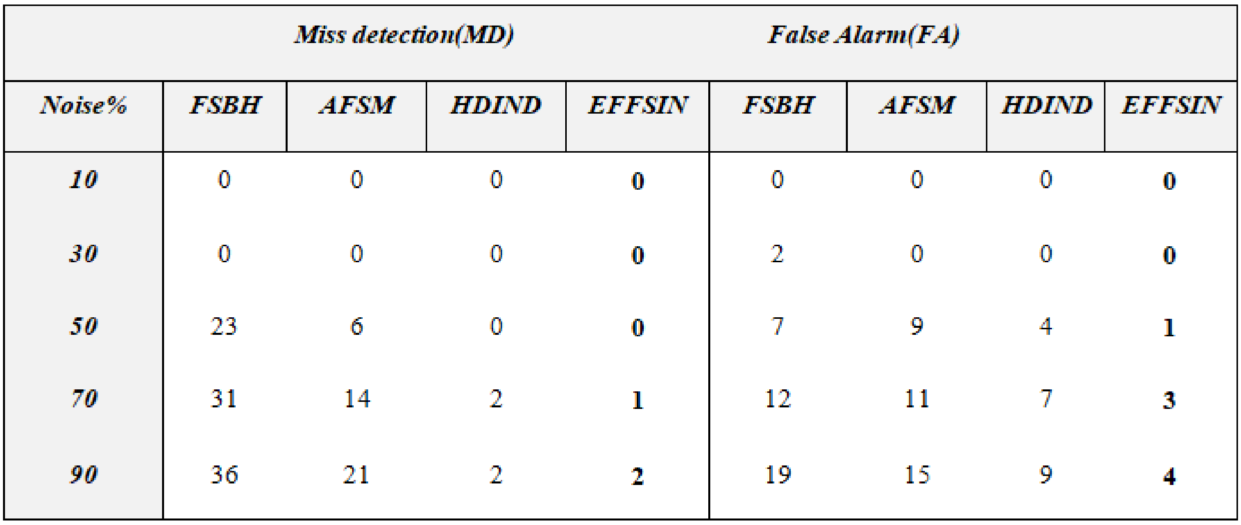


IMAGE NAME	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*
	30% Noise Density		*40% Noise Density*		*50% Noise Density*	
Living room	38.79	0.8992	37.68	0.8876	38.48	0.9007
Woman Blonde	40.19	0.9237	39.70	0.9192	37.34	0.8998
Mandril	35.87	0.8922	35.18	0.8910	34.47	0.8848
Pirate	39.96	0.9015	38.63	0.8993	38.48	0.9007
Cameraman	39.03	0.9067	37.81	0.8981	36.77	0.8837

TABLE 4.3: FA and MD values for image "*Lena*" using the Proposed filter



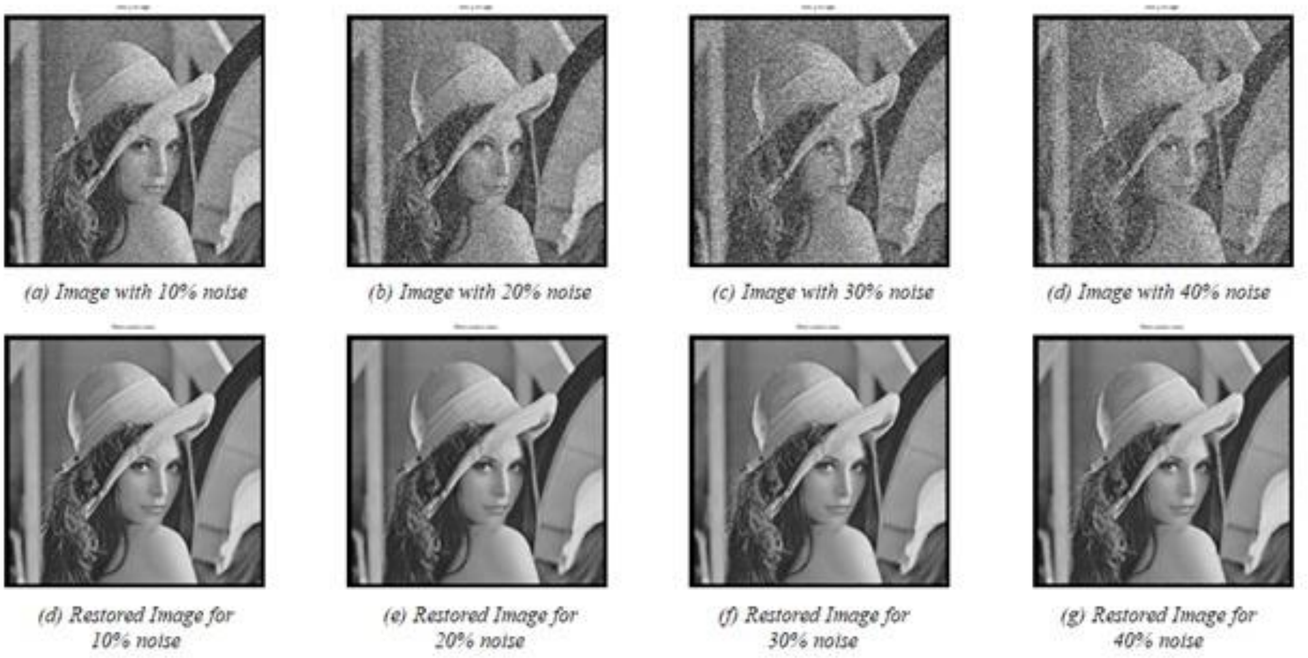
	Miss detection(MD)				*False Alarm(FA)*			
Noise%	***FSBH***	***AFSM***	***HDIND***	***EFFSIN***	***FSBH***	***AFSM***	***HDIND***	***EFFSIN***
10	0	0	0	**0**	0	0	0	**0**
30	0	0	0	**0**	2	0	0	**0**
50	23	6	0	**0**	7	9	4	**1**
70	31	14	2	**1**	12	11	7	**3**
90	36	21	2	**2**	19	15	9	**4**

Table 4.1 presents the analysis of PSNR and SSIM values for the "*Lena*" image using various algorithms at varying noise densities. EFFSIN excels with high SSIM and PSNR results in comparison to other methods. The proposed filter is robust in nature and its performance does not deteriorate with an increase in noise density.

Analysis of PSNR and SSIM values for various images using EFFSIN is presented in Table 4.2, for noise levels of 30%, 40%, and 50 %. EFFSIN achieves excellent results for different images as well.

There is a notable improvement in the results of *HDIND* [78] after the incorporation of edge detecting features in the impulse restoration phase. The proposed filter confers consistent performance with varying amounts of noise.

It is derived from Table 4.3 that the *FA* and *MD* values are reduced remarkably by EEFSIN.



(a) Image with 10% noise

(b) Image with 20% noise

(c) Image with 30% noise

(d) Image with 40% noise

(d) Restored Image for 10% noise

(e) Restored Image for 20% noise

(f) Restored Image for 30% noise

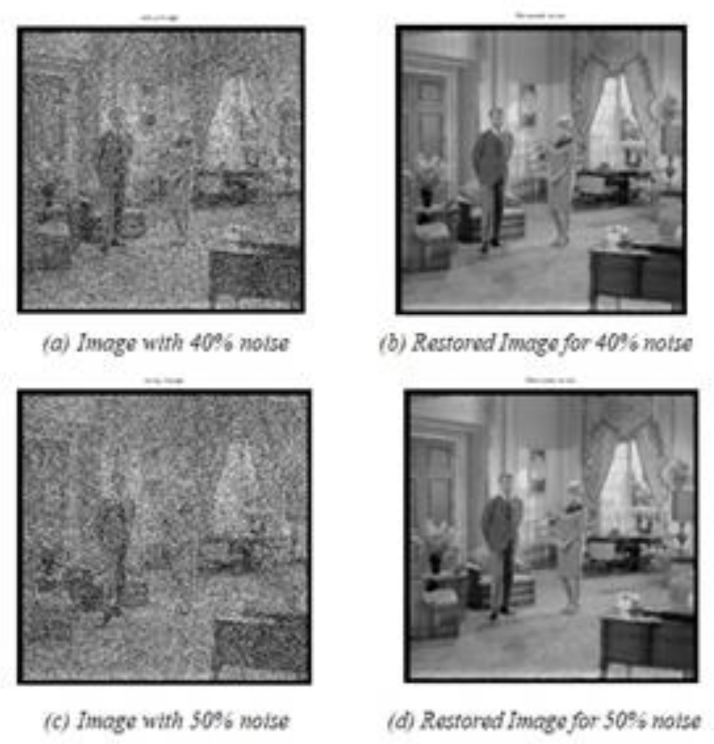
(g) Restored Image for 40% noise

Figure 4.6: Results for Lena image at different noise levels using EFFSIN

From the analysis of mathematical data in Table 4.1, Table 4.2 and Table 4.3, it is clear that *EFFSIN* significantly surpasses the contemporary techniques available for impulse noise. *HDCNN* [79] yields good *PSNR* and *SSIM* values but is complex in nature with high computation time. The proposed filter is elementary in nature with less computation time.

FAHDS [80] does not preserve the fine details of the image and provides low quantitative results. Similarly, *FSBH* [81] and *AFSM* [82] does not reduce the *FA* and *MD* values.

EFFSIN [83] is edge-preserving in nature which provides relatively low False alarm rates and miss detection rates thereby improving the *PSNR* and *SSIM* values. It also gives efficient results for different test images. The application of the median filter after the assessment of possible edges has enhanced the quality of work.



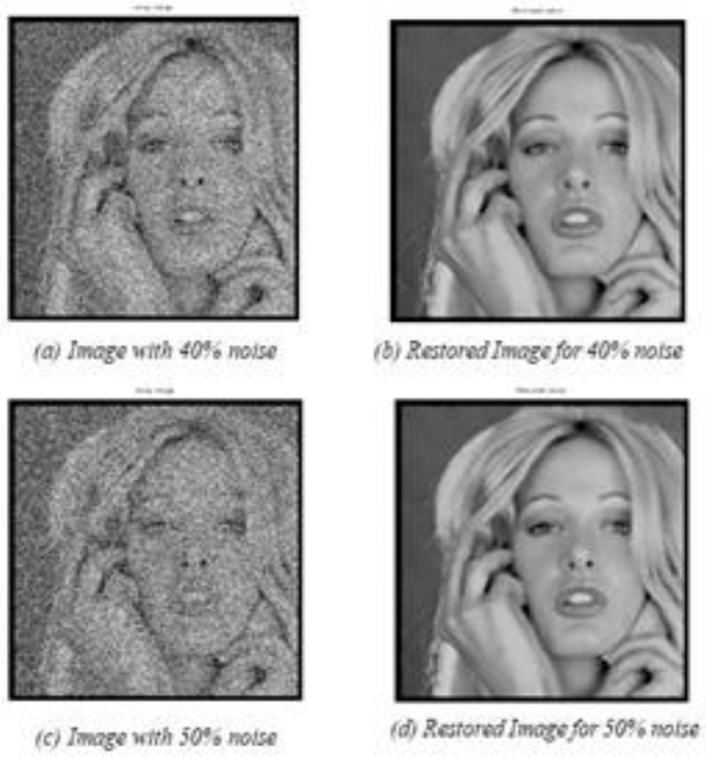
(a) Image with 40% noise

(b) Restored Image for 40% noise

(c) Image with 50% noise

(d) Restored Image for 50% noise

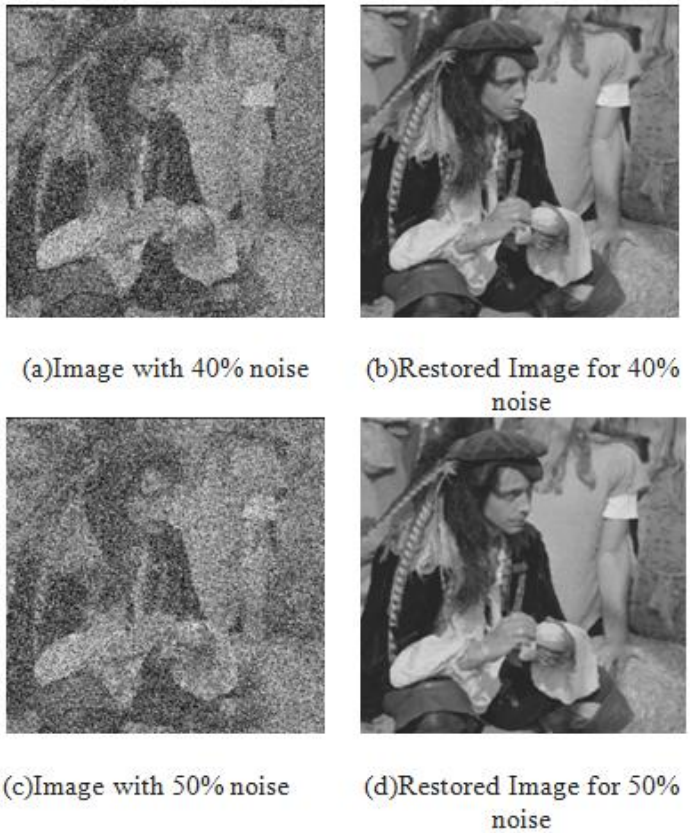
Figure 4.7 : Results for “*livingroom*” image by EFFSIN



(a) Image with 40% noise (b) Restored Image for 40% noise

(c) Image with 50% noise (d) Restored Image for 50% noise

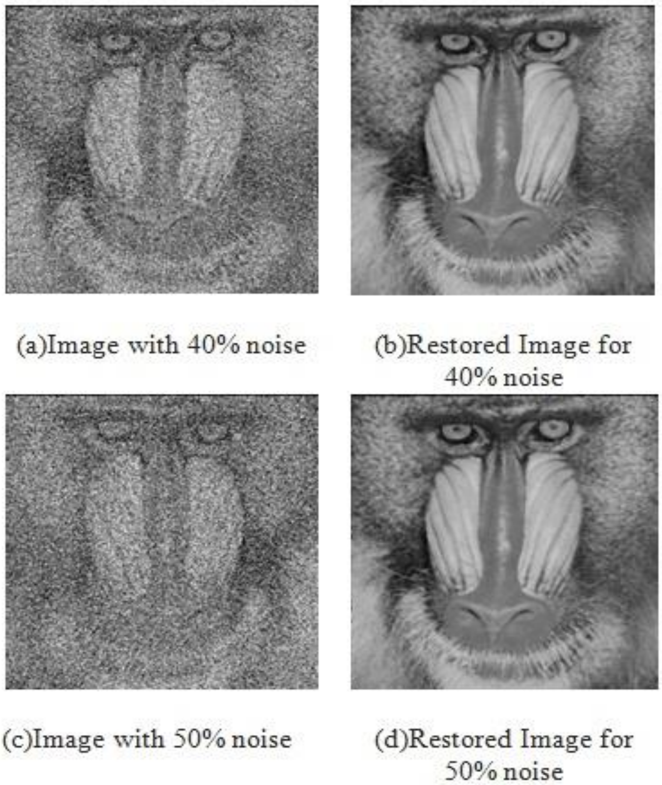
Figure 4.8 : Results for “*Womenblonde*” image using EFFSIN



(a)Image with 40% noise (b)Restored Image for 40% noise

(c)Image with 50% noise (d)Restored Image for 50% noise

Figure 4.9: Results for “*Pirate*” image using EFFSIN



(a)Image with 40% noise

(b)Restored Image for 40% noise

(c)Image with 50% noise

(d)Restored Image for 50% noise

Figure 4.10: Results for “*Mandril*” image using EFFSIN

The qualitative results for “*Lena*” in Fig. 4.6, “*livingroom*” in Fig. 4.7, “*Womenblonde*” in Fig. 4.8, “*Pirate*” in Fig. 4.9 and “*Mandril*” in Fig. 4.10, display the high conduct of the proposed filter for differing levels of impulse noise. All the simulation results are derived using MATLAB.

4.4 Conclusions

In this chapter, an edge-preserving algorithm for the suppression of impulse noise is introduced. The two-phase impulse noise detection and proper impulse noise restoration with the evaluation of possible edges make it a coherent technique for impulse noise suppression. The extensive simulation results predict that the proposed work makes a remarkable reduction in the False alarm rates. Moreover, it also excels quantitatively and qualitatively.

An unequivocal approach of the proposed work enhances its performance with less computation time. In the future, the work can be extended further for impulse noise detection and restoration in color images and sequences.

Chapter 5

A Novel Approach for Salt and Pepper Noise Removal based on Heuristic Analysis of Neighboring Pixels

5.1. Premialiaries

A robust framework for impulse noise detection is the primary objective of this thesis work. This chapter addresses this problem of efficient noise detection with another technique based on heuristic analysis of the neighboring pixels. It is a contemporary technique designed to suppress impulse noise in images, abbreviated as SPHN. It is a two-stage filter, in the first stage, the noisy pixels are detected. Only the pixels classified as noisy are subjected to the filtering stage. In the detector stage the proposed algorithm analyses the neighborhood of the pixel under consideration, based on this analysis it classifies the pixel as noisy or noise-free. The noisy pixels are then subjected to filtering. PSNR and MSE are used as performance measures for simulation results. SPHN outperforms the existing methodologies in literature. The comparative analysis is also given in the sections ahead. The next section describes the detailed functionality of the proposed filter.

5.2. SPHN Algorithm

SPHN is a double-stage filter, the first stage identifies the noisy and noise-free pixels, the second stage performs filtering only on the noisy pixels. Not subjecting the noise-free pixels to the filtering process helps to preserve fine details and avoids blurring. The details of the two stages are as follows:

5.2.1 Detection Stage

In the detection stage, the corrupted image X is subjected to a 3×3 window $W_{i,j}$ centered at pixel $x_{i,j.}$ The current pixel is compared to the maximum and minimum values l_{max} and l_{min} respectively within the current window; if the pixel is equal to either of the two values it is a noisy candidate.

$$x_{i,j} = \left\{ \begin{array}{ll} noise-free\ , & l_{\min} < x_{i,j} < l_{\max} \\ noise\ candidate\ , & x_{i,j} = l_{\min}\ or\ x_{i,j} = l_{\max} \end{array} \right\} \tag{5.1}$$

Where,

l_{max} = local maxima within the current window

l_{min}= local minima within the current window

Once identified as a noise candidate it is important to ascertain that the pixel is an isolated maximum or minimum value i.e. noise, and not a part of any edges. To do this, calculate the distance of all pixels in the window with the central pixel as below :

$$\delta_i = abs\ \left|(p_i - \xi)\right| \tag{5.2}$$

Where,

δ_i = distance of the ith pixel from the central pixel

p_i = ith pixel

ξ = central pixel in the 3×3 window

abs = absolute value

Next, we calculate the average δ_{avg} of all the distances δ_i.

By comparing this average distance δ_{avg} with the global maxima φ, we can classify the pixels into noisy or noise-free according to equation 5.3 :

$$x_{i,j} = \left\{ \begin{array}{ll} Noisy\ Pixel\ , & \delta_{avg} \geq 90\%\ \ of\ (\phi/2) \\ Noise{-}Free, & otherwise \end{array} \right\} \tag{5.3}$$

If according to the equation. 5.3 pixel is noisy, we process it further.

Then, the value of pixels p_1 & p_2 in the neighborhood ξ having the least distance δ_i (from ξ) are computed and calculate their average p_{avg}.

Finally, we categorize noisy and noise-free pixels according to equation 5.4.

$$x_{i,j} = \left\{ \begin{array}{ll} Noisy\ Pixel\ , & \rho_{avg} \geq 90\%\ \ of\ (\phi/2) \\ Noise{-}Free\ , & otherwise \end{array} \right\} \tag{5.4}$$

If the central pixel is noisy it undergoes the filtering stage, else if it is noise-free no further processing is needed. The value of global maximum value needed to determine whether a pixel is noisy or not at the detector stage of SPHN will be 255. As we are working with 8-bit grayscale images.

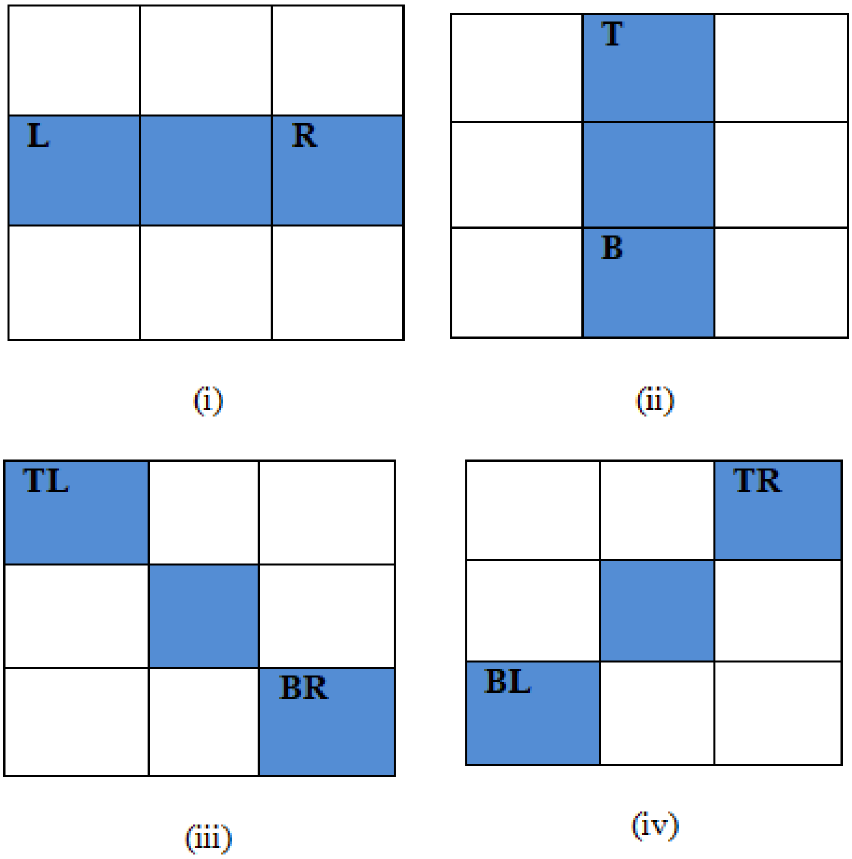
5.2.2 Filtering Stage

When the noisy pixels are identified, they are subjected to filtering.

In the filtering stage, the first step is to calculate the mean η_i of pixels in all directions

η_1 (L to R), η_2 (T to B), η_3 (TL to BR) and η_4 (TR to BL).

Next, the distance of each pixel p_i from its mean η_i is calculated. Now the average of the distances η_{avgi} for each direction *i*, as shown in Fig. 5.1 is calculated.



(i) (ii) (iii) (iv)

Figure 5.1: Directions used to calculate the distances

The next step is to assign weights *(w_1 to w_8)*. Pixels in the direction with minimum η_{avgi} get the highest weight 4.

Pixels in the direction of the second minimum η_{avgi} gets weight 3.

Similarly, we assign weights 2 and 1 to directions with 3rd and 4th minimum η_{avgi} respectively.

After assigning the weights the restoration of the corrupted pixels is done according to equation 5.5:

$$y_{i,j} = \begin{Bmatrix} median\{w_i \lozenge x_{i,j}\} , & if\ x_{i,j}\ is\ noisy \\ x_{i,j}\ , & if\ x_{i,j}\ is\ noise-free \end{Bmatrix} \qquad (5.5)$$

Where,

l_{max}=local maxima of the present window

l_{min} = local minima of the present window

ξ = central pixel in the window

δ_i = distance of the ith pixel from the central pixel

$\eta_1, \eta_2, \eta_3, \eta_4$ = mean of pixels in all directions

p_i = ith pixel

δ_{avg} = average of all distance values

ϕ = global maxima for the image

ρ_{avg} = average of pixels nearest to the central pixel ξ

η_{avg} = distance average

$\lozenge$ = repetition operator

$w_1\ to\ w_8$ = weights assigned to pixels

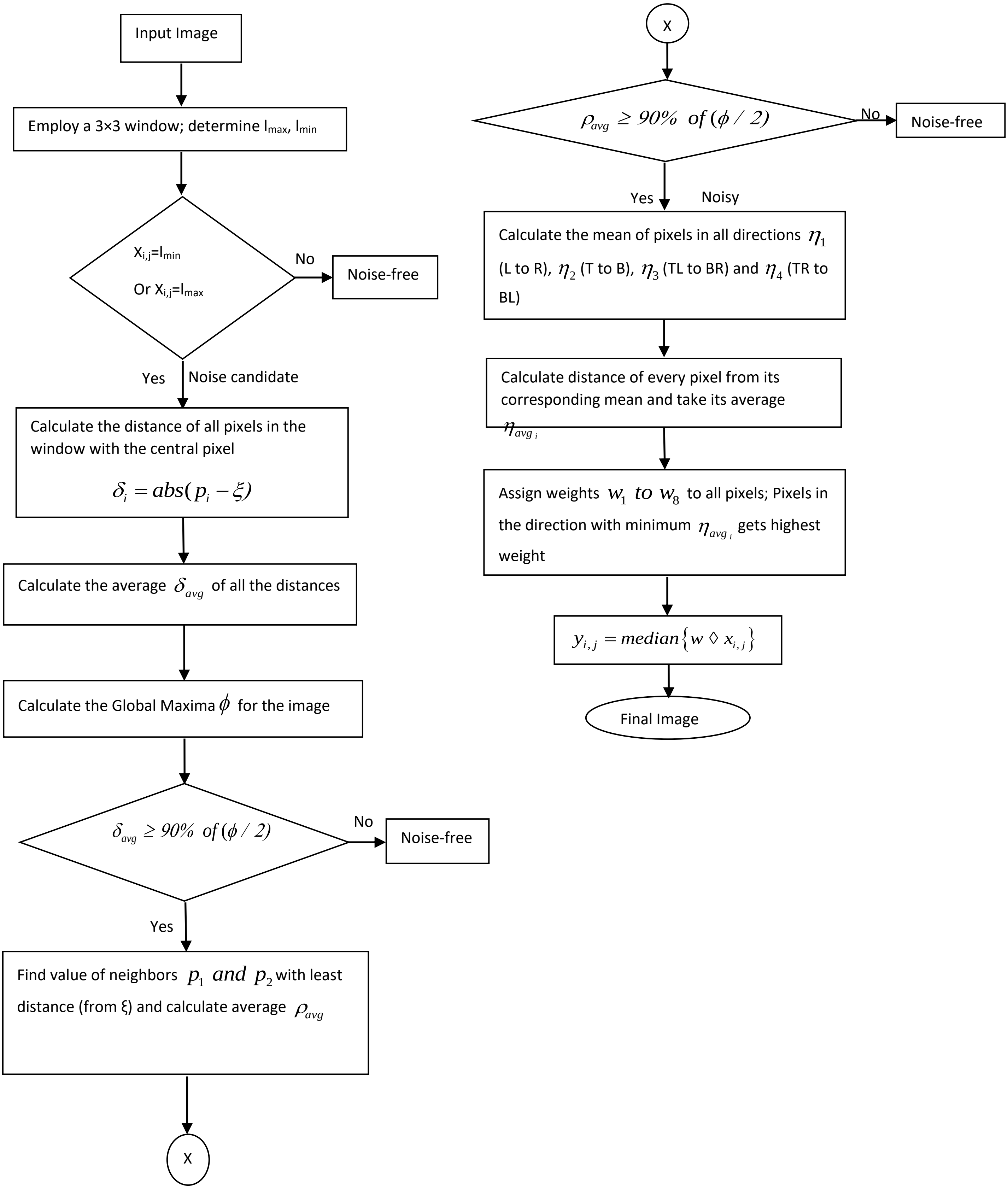

Input Image
Employ a 3×3 window; determine I_{max}, I_{min}
$X_{i,j}=I_{min}$ Or $X_{i,j}=I_{max}$
No
Noise-free
Yes
Noise candidate
Calculate the distance of all pixels in the window with the central pixel
$\delta_i = abs(p_i - \xi)$
Calculate the average δ_{avg} of all the distances
Calculate the Global Maxima ϕ for the image
$\delta_{avg} \geq 90\%$ of $(\phi / 2)$
No
Noise-free
Yes
Find value of neighbors p_1 and p_2 with least distance (from ξ) and calculate average ρ_{avg}
X
X
$\rho_{avg} \geq 90\%$ of $(\phi / 2)$
No
Noise-free
Yes
Noisy
Calculate the mean of pixels in all directions η_1 (L to R), η_2 (T to B), η_3 (TL to BR) and η_4 (TR to BL)
Calculate distance of every pixel from its corresponding mean and take its average η_{avg_i}
Assign weights w_1 to w_8 to all pixels; Pixels in the direction with minimum η_{avg_i} gets highest weight
$y_{i,j} = median\{w \lozenge x_{i,j}\}$
Final Image

Figure 5.2: Algorithm Flowchart

5.3. Simulation results

The proposed algorithm (SPHN) is evaluated by comparing it with other well-known algorithms for both grayscale and color images. For this purpose, salt and pepper noise of varying intensity, ranging from 10% to 90%, was added to the original image and the performance was analyzed in terms of visual quality, MSE, and PSNR.

Quantitatively, the relative performance of the algorithms is measured in terms of the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

The PSNR and MSE values for SPHN filter, for various images, at varying noise intensities are presented in Table 5.1. The mathematical results imply that SDOD has consistent performance in terms of PSNR and MSE values. It is able to handle even high noise levels with efficient results.

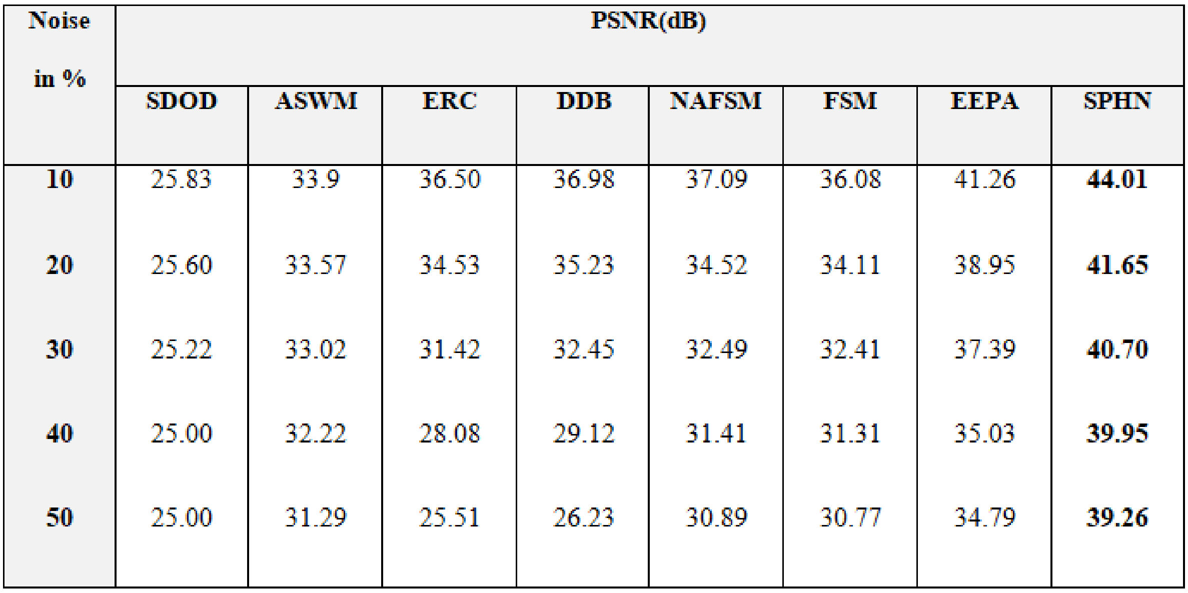
The PSNR and MSE values of the SPHN [84] filter are compared against the existing algorithms such as SDOD [85], ASWM [57], FSM [52], ERC [86] filter, DDB [87] filter, NAFSM [53] and EEPA [51] at variable noise levels, and are depicted in Table 5.2.

SDOD gives the lowest results for PSNR. ERC provides good results at lower noise density levels but degrades in its performance at higher levels. NAFSM and FSM are consistent in performance but fail to provide high qualitative results. The quality of images using EEPA also deteriorates above noise level 40%. The results are analyzed for different images and it is depicted that SDOD outperforms other filters with efficient and consistent results. The simple structure of SDOD makes it easy to implement and utilize it for various applications.

Table 5.1: PSNR and MSE values for SPHN filter for standard images at varying noise intensities

Type of image	Noise in %								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
PSNR(dB)									
Peppers	87.84	80.55	77.60	70.95	69.00	66.91	64.40	65.17	65.54
Lena	72.01	71.65	69.70	68.95	66.26	64.03	62.68	62.92	63.33
Baboon	77.13	76.29	69.98	64.90	61.97	60.42	59.70	58.52	60.01
Pirate	85.16	75.62	71.06	68.27	65.85	64.58	61.47	61.53	63.26
Cameraman	69.56	67.42	66.78	65.93	65.67	65.15	63.70	63.71	63.72
MSE									
Peppers	0.01	0.05	0.07	0.08	0.09	0.13	0.23	0.19	0.18
Lena	0.03	0.04	0.07	0.08	0.15	0.25	0.35	0.33	0.35
Cameraman	0.07	0.14	0.17	0.18	0.19	0.19	0.27	0.25	0.27
Baboon	0.01	0.02	0.03	0.05	0.07	0.09	0.13	0.12	0.13
Pirate	0.07	0.08	0.09	0.13	0.16	0.22	0.46	0.45	0.47

Table 5.2: Comparisons of restoration results in PSNR (dB) for different algorithms corrupted by varying Noise Intensities for Test Image "*Lena*".

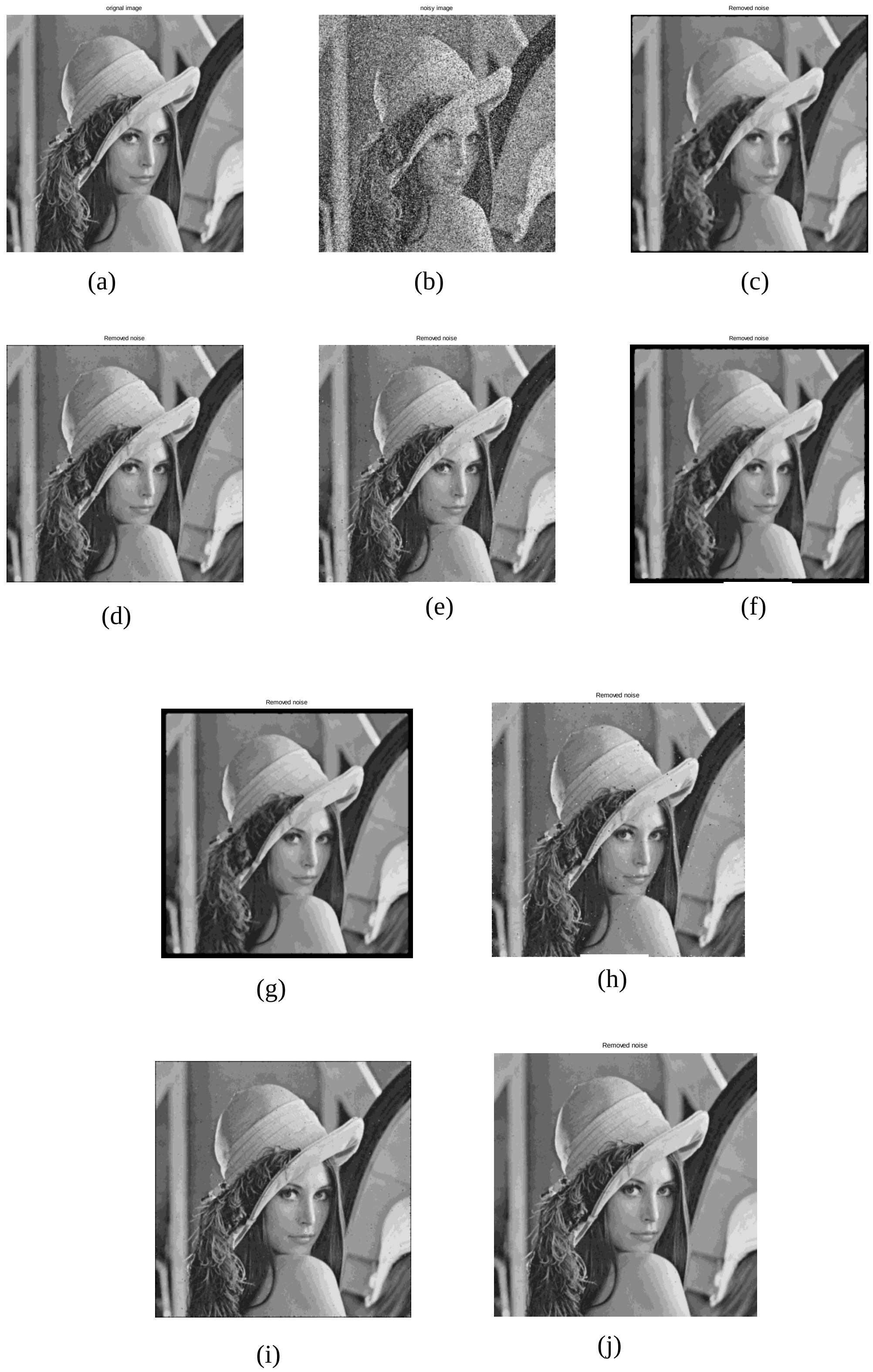


Noise in %	PSNR(dB)							
	SDOD	ASWM	ERC	DDB	NAFSM	FSM	EEPA	SPHN
10	25.83	33.9	36.50	36.98	37.09	36.08	41.26	**44.01**
20	25.60	33.57	34.53	35.23	34.52	34.11	38.95	**41.65**
30	25.22	33.02	31.42	32.45	32.49	32.41	37.39	**40.70**
40	25.00	32.22	28.08	29.12	31.41	31.31	35.03	**39.95**
50	25.00	31.29	25.51	26.23	30.89	30.77	34.79	**39.26**

For the analysis of visual quality, the restored images by various denoising methods for 30% noisy image "*Lena*" is shown in Fig. 5.3.

Some more results for the image "*Pirate*" in Fig. 5.4, image "*Peppers*" in Fig. 5.5, and image "*Cameraman*" in Fig. 5.6, are provided at noise levels of 20%, 40%, and 60% respectively.

The visual results for SDOD for various images depict the robust nature of the algorithm. The images are seen with a significant reduction in impulse noise after the application of the SDOD filter. It can be observed from the restoration images, the proposed SPHN filter produces results, which are better both qualitatively and quantitatively for grayscale images.



(a) (b) (c)

(d) (e) (f)

(g) (h)

(i) (j)

Figure 5.3: Restoration results of different methods in restoring corrupted image "Lena" (a) Original noise-free image, (b) corrupted image with 30% impulse noise, (c) SDOD (d) ASWM (e) FSM (f) ERC (g) DDB (h) NAFSM (i) EEPA (j) SPHN

5.4. Conclusions

This chapter presents an SPHN filter for the suppression of impulse noise. The Proposed algorithm was implemented using Matlab 7.9 and several test images namely "Lena", "Cameraman", "Peppers", "Pirate and "Baboon" were used. The performance of the proposed filter on various images at varying noise levels was analyzed and also compared with many existing filters. It can be deduced that SPHN works well at both low and high noise densities and also in comparison to some popular techniques. We have used only grayscale images for our research but this technique can be expanded for color images and sequences.

Chapter 6

Impulse Noise Removal in color image sequences using Fuzzy Logic

6.1 Preliminaries

This thesis work primarily deals with grayscale images for impulse noise removal. This Chapter addresses the impulse noise suppression from videos. A two-step fuzzy filter for impulse noise from an image sequence in RGB color space is introduced. The primary filter recognizes the noisy pixels with the measure of noise present and afterward rectifies them. The output of the first filter acts as an input to the second filter and further refines the outcome to give the final output. Excellent alignment is seen between noise removal and structure conservation due to the classifying nature of the algorithm. To minimize blurring, noisy pixels are exclusively rectified and noise-free pixels remain intact. The proposed filter is a 3-D Spatio-temporal filter that considers spatial, temporal as well as the color information. A pixel of one color component is compared to its neighboring pixels within the same frame and with the corresponding pixels in neighboring frames. It is likewise compared with the pixels of another two color components. Peak Signal to Noise Ratio (*PSNR*) , structural similarity index (*SSIM*), Miss Detection (*MD*), and False Alarm

(*FA*) rates are utilized as a performance metric. The experimental result of several color image sequences demonstrates the efficacy of the proposed fuzzy filter both qualitatively as well as quantitatively. A brief study of impulse noise removing filters used for color video sequences is presented in the next section.

6.2. Literature survey

A video sequence is the broadcasting of visual images or frames at a certain frame rate per second unit of time. The number of frames displayed per second depends on the type of application and user requirements. These frames or images get impaired with image noise due to various anomalies. Spatial filters play a substantial role to suppress impulse noise from images. They exploit the image plane itself for their processing. Most of the filters developed for video are 2-D spatial filters that filter the individual frames of the video progressively.To expel noise, an assortment of linear and nonlinear filters have been presented by different researchers. It is seen that non-linear techniques excel more as compared to the linear procedures as a result of their proficiency in recognizing and expelling noise. To eliminate noise from images, many filters have employed fuzzy logic in the past [88]. Fuzzy filters provide both edge preservation and precision [89]. Several filters are found in literature like the Fuzzy Directional (FD) filter [90,91], FRIN [92] which uses noise adaptive mean absolute difference along with fuzzy logic, fuzzy filter with type-2 fuzzy inference systems [93], Fuzzy two-step filter [94], Fuzzy random impulse noise removal [95] and Region adaptive fuzzy filter [96] to name a few. Median (*MED*) filter [97] is a notable non-linear filter utilized as a foundation in numerous filters since it disposes off the noise and performs the filtering process effectively even in the soft area of an image. Various varieties of median filters, namely Center-weighted median [46], Tri-state median filter [31], Direction based adaptive weighted switching median filter [32], Vector median filters [33], Switching weighted vector median filter [34], Modified switching median filter [35], CAWMF [36], Decision-based unsymmetric trimmed median filter [56], Iterative grouping median filter [98] and many more. A combination of fuzzy logic and median filter has proved as an efficient tool for noise removal in images. Some state-of-art filters based on this combination are fuzzy inference structure-based directional median filter [99], Fuzzy Directional Median Filter [100], cluster-based adaptive fuzzy switching median filter [82], Fuzzy reasoning-based directional median filter [101], adaptive fuzzy median filter [102,103] and many more. The use of fuzzy logic and median filters in our work has enhanced the noise removal capacity alongside the conservation of image details [104,105]. All the above-mentioned filters are also known as intra-frame filters as they consider only one frame at a time and remove the noise independent of the neighboring frames. By considering only one frame at a time they neglect the temporal relation between frames of the video.On the contrary, Temporal filters are inter-frame filters that analyze the relationship between the subsequent frame and neglect the spatial correlation [106,107].

Spatio-temporal filters are those filters that take into account both the spatial as well as temporal correlation and remove noise from video by considering the current frame as well as the next frame [108,109]. A study of various Spatio-temporal filtering schemes to remove noise in the video can be found in the paper by Brailean et al [110]. Spatio-temporal filters also employ various systems like fuzzy systems, neural networks, Neuro-fuzzy systems [111], optimization techniques [112], and so on. Such as, a paper by Lee et.al [113], which is a noise adaptive filter designed for low light levels. It requires extra hardware implementation using various modules which increase the overall system cost. Another paper by Szczepanski [114] introduces a Spatio-temporal based on digital paths which provide good results but is quite computation intensive in nature. Chervyakov et.al [115] proposed a method based on Lorentz function and utilized adaptive filtering for video denoising. These types of algorithms which include double processing stages excel better than other algorithms having a single module for noise detection. Therefore, in our work, we have introduced two-phase detection of impulse noise. The result of the primary filter acts as input to the secondary filter so that results are more refined and accurate.

Fuzzy systems play an important role in the methodologies devised for image processing. For real-time applications as discussed above, fuzzy logic provides more efficient results and is less computationally intensive as compared to other systems. Therefore, in our work, we propose a Spatio-temporal filter that removes impulse noise from the video using fuzzy logic to enhance the decision making capability of our algorithm. Nevertheless, a variety of methods are present in literature to remove noise from a color image sequence, most of them are either unable to provide desirable results or provide results only with low noise density. The proposed work calculates the degree with which the pixel is noisy and then removes the noise from it accordingly. Hence, only the noisy pixels are processed, which maintains the basic image structure. For noise removal in a color image sequence, along with spatial and temporal information, color information in other bands is required as well. Hence, the noise is corrected by considering the current frame, the neighboring frames, and the values of the other color components. Five membership functions are defined which provide complete spatial, temporal, and color information about the pixel. A Fuzzy rule is then applied to each pixel to check if it is degraded by noise or not. If the pixel is observed to be noisy, it is restored using the median value computed exclusively from the pixels free from noise in their neighborhood otherwise it is not changed. It is a robust technique that can handle the high impulse noise presence up to 70%. The video denoising performance is evaluated by the peak signal-to-noise ratio (*PSNR*), Structural similarity index (*SSIM*), false

Alarm (FA), and Miss Detection (MD) rates. The experimental results on several videos proved the efficacy of the proposed filter as compared to other well-known methods.

The next section defines the proposed Spatio-temporal filter to minimize impulse noise. Results and analysis, parameter selection and comparison with other filters are depicted in section 6.4. Lastly, the conclusion is provided in Section 6.5.

6.3 The Proposed Algorithm

The proposed work is devised for the color image sequence impaired with impulse noise.Mathematically, noisy sequence f_n is represented in equation (6.1):

$$f_n(s,t,CB,N^F) = \left\{ \begin{array}{l} f_o(s,t,CB,N^F),\ \ with\ probability\ 1-p \\ N(s,t,CB,N^F),\ \ with\ probability\ p \end{array} \right\} \tag{6.1}$$

Where p represents the probability between [0,1] with which a pixel component value is noisy, f_n is the noisy sequence, f_o is the original (noise-free) sequence, N is the noise value of the pixel, *(s,t)* corresponds to the pixel location in video sequence, *CB* corresponds to one of the three color bands, *CB=1, 2, 3* for Red, Blue and Green components respectively and N^F corresponds to the frame number.

For example, $f(s,t,1,N^F)$ corresponds to the red component of the pixel at s_{th} row and t_{th} column in the current frame N^F. This notation is followed throughout the chapter.

FILTERING STEPS:

The impulse noise from the color image sequences is removed in two progressive steps of filtering. The primary filter detects the noisy pixels and degree of noise. The output of the primary filter acts as an input to the secondary filter where it is further refined to give the final output. The degree of noise is calculated by using a fuzzy rule which requires the calculation of five fuzzy membership functions.

The five membership functions, namely; Big, Dissimilar, Terminal, High and Variant are defined.

Big, Dissimilar and Terminal membership function gives spatial information about a given pixel.

High and Variant membership functions give temporal information about the pixel. A Fuzzy rule operates on every pixel to check if it is noisy or noise-free. Depending on the outcome, the algorithm proceeds

6.3.1 Primary Filter

The main objective of the primary filter is the proper detection and rectification of noisy pixels. The pixels, altered due to noise is detected in this step along with the noise amount with which the pixels are corrupted. After detection, the median value calculated from noise-free pixels in the present window restores the noisy pixels Consider one color component, say a red component of the current frame N^F in the noisy image sequence. The extent of noise present in a pixel at position (s, t) is attained via the fuzzy rule 1.

$f(s,t,1, N^F)$ is central pixel under evaluation, N^F denotes the current frame,

$N^F +1$ is the next frame and N^F *-1 its* preceding frame. $f(s,t,1, N^F -1)$ denotes the relating pixel in the preceding frame, $f(s,t,1, N^F +1)$ is the relating pixel in the next frame and *med* $(s,t,1, N^F)$ is the median value computed from noise-free pixels in the current window. Mathematically the amount of noise that exists in the red component of a pixel at position *(s, t)* in the current frame N^F is evaluated as:

$$N_1\left(s,\ \mathrm{t},\ 1,\ N^F\right)=min\left\{\begin{array}{l}\mu_b(D(s,\ t,\ 1,\ N^F)),\ \mu_d\left(s,\ \mathrm{t},\ 1,\ N^F\right),\ \mu_{ter}\left(\ \mathrm{s},\ \mathrm{t},\ 1,\ N^F\right),\\ \mu_h\left(\ \mathrm{s},\ \mathrm{t},\ 1,\ N^F\right),\ \mu_v\left(s,\ \mathrm{t},\ 1,\ N^F\right)\end{array}\right\} \tag{6.2}$$

Where μ_b, μ_d, μ_{ter}, μ_h, μ_v *are* five membership functions that represent the five fuzzy sets Big, Dissimilar, Terminal, High and Variant respectively. *D* denotes the distinction in the central pixel value and the median value calculated from uncorrupted pixels within the current window. Mathematically,

$$D\left(s,t,1,N^F\ \right)\ =\ |\ f\left(s,t,1,N^F\right)-med\left(s,t,1,N^F\right)| \tag{6.3}$$

Likewise, the difference in blue and green components is calculated by using values 2 and 3 respectively for *CB* in the above equation. The same is followed throughout the algorithm.

The median value is estimated only from the noise-free pixels present within the window to modify the noisy pixel under interest. For R, G, and B color components, the corresponding median value is derived. To attain the value for the median from noise-free pixels, an algorithm presented in [89] is used. This median value is used instead of a simple median value due to its superior performance in both qualitative as well as quantitatively. The algorithm is described as:

1) Take a 3×3 window size to center on the present pixel to be processed in the noisy frame.

2) Every pixel of the window is organized into a vector and is sorted in ascending order to compute the median value.

3) The difference between the obtained median value and every window pixel is computed.

4) Pixels carrying the difference value less than or equal to the value of δ are organized into a new vector.

5) The new vector is sorted in ascending order to get the new median value (*med*).

The value used for (delta) δ is 5. This is intuitive in nature and as per maximum PSNR value it was chosen experimentally. Larger deviation from the median value imply presence of noise.

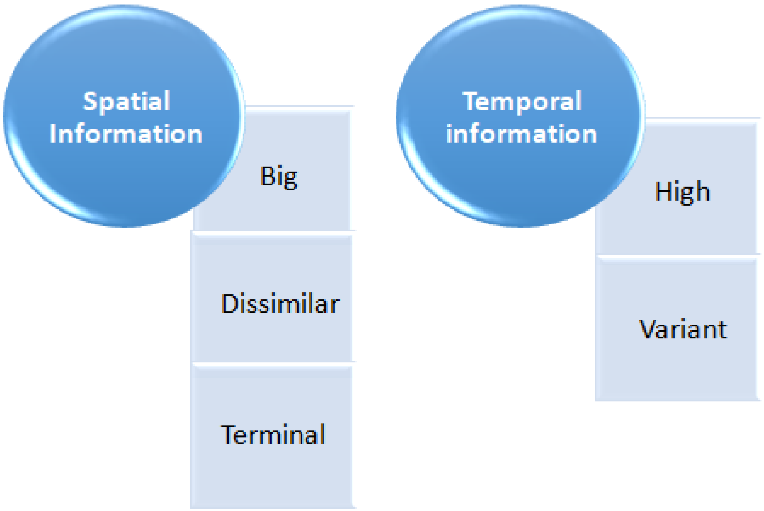


Figure 6.1: Membership Functions

Fuzzy Set: Big

The value of noisy pixels will show a larger deviation from the median value. This variation is denoted by μ_b, which is used for the fuzzy set "**Big**", that illustrates the huge difference. After calculating the difference *D*, μ_b is calculated for each color component as:

$$\mu_b(D(s,t,1,N^F)) = \begin{cases} 1, & D(s,t,1,N^F) \geq \alpha_2 \\ \dfrac{D(s,t,1,N^F) - \alpha_1}{\alpha_2 - \alpha_1}, & \alpha_1 \leq D(s,t,1,N^F) < \alpha_2 \\ 0, & D(s,t,1,N^F) < \alpha_1 \end{cases} \qquad (6.4)$$

The values of α_1 and α_2 in the above equation are derived experimentally.

Fuzzy Set: Dissimilar

Let K be the count of neighboring pixels in the present frame that are similar to the given pixel. The more the number of similar pixels, the less the chances of a given pixel being noisy. For this, a similarity criterion is used to decide whether the given pixel is similar to its neighboring pixels or not. Also, if we want to find the similarity of a red component of a pixel, we will find its interaction with the other 2 color components as well to ascertain the extent of the similarity.

Similarity Criterion: Red Component

To discover whether the red component of a pixel at location *(s,t)* matches with its adjacent pixels at the location *(s + q, t+ r),* we have to find the following differences

The variations of blue and green color components with respect to the red color component is estimated using the following equations:

$$\mathrm{d}_{\mathrm{rg}}\left(s,\ \mathrm{t},\ 1, N^F\right) = \left|\mathrm{f}\left(s,\ \mathrm{t},\ 1,\ N^F\right) - \mathrm{f}\left(s,\ \mathrm{t},\ 2,\ N^F\right)\right| \tag{6.5}$$

$$\mathrm{d}_{\mathrm{rb}}\left(s,\ \mathrm{t},\ 1, N^F\right) = \left|\mathrm{f}\left(s,\ \mathrm{t},\ 1,\ N^F\right) - \mathrm{f}\left(s,\ \mathrm{t},\ 3,\ N^F\right)\right| \tag{6.6}$$

In the same manner, the variation of blue and green color components with respect to red color component for neighboring pixels *(s + q,t+ r)* is estimated using the following equations:

$$\mathrm{d}_{\mathrm{rg}}\left(s+q,\ \mathrm{t}+r, 1,\ N^F\right) = \left|\mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 1,\ N^F\right) - \mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 2,\ N^F\right)\right| \tag{6.7}$$

$$\mathrm{d}_{\mathrm{rb}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) = \left|\mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 1,\ N^F\right) - \mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 3,\ N^F\right)\right| \tag{6.8}$$

The variation amongst the above-combined components at the pixel locations *(s, t)* and *(s+q, t+r)* is derived using the following equations:

$$\Delta_{\mathrm{rg}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) = \left|\mathrm{d}_{\mathrm{rg}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) - \mathrm{d}_{\mathrm{rg}}\left(s,\ \mathrm{t},\ 1, N^F\right)\right| \qquad (6.9)$$

$$\Delta_{\mathrm{rb}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) = \left|\mathrm{d}_{\mathrm{rb}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) - \mathrm{d}_{\mathrm{rb}}\left(s,\ \mathrm{t},\ 1, N^F\right)\right| \qquad (6.10)$$

Likewise, the variations amongst the central pixel and its adjacent pixels within the current window for the same color are computed as**:**

$$\Delta_{\mathrm{r}}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right) = \left|\mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 1,\ N^F\right) - \mathrm{f}\left(s,\ \mathrm{t},\ 1,\ N^F\right)\right| \qquad (6.11)$$

Now the red component of a pixel located at position *(s, t)* in the present frame N^F is considered similar to that at *(s + q, t+ r)* within the same frame in process N^F provided that the variations Δ_{rg} $(s + q,t + r, 1, N^F)$, Δ_{rb} $(s+ q,t+ r, 1, N^F)$ and Δ_r $(s+ q, t+ r, 1, N^F)$ as computed in *equations (6.9), (6.10) and (6.11)* respectively have their value below than the parameter λ_1.

Consider a window of size w × w (say 3×3). Then i and j in equation (6.7), (6.8), (6.9), (6.10) and (6.11) will range from -1 to +1. There will be a total of 9 pixels in the window, including the central pixel. K is the number of pixels having alike values except for the central pixel. That is, K is the value for similar pixels in the neighborhood with respect to processing pixel. In the window of size 3×3, *K* will range from 0 to 8. More are the count of comparable pixels in the neighborhood; less will be the chances of the central pixel being noisy. This number *K* is decided by comparing differences calculated in *equations (6.9), (6.10) and (6.11)* with parameter λ_1 estimated experimentally.

Now if *Δ_{rg}(s+ q, t+ r, 1, N^F), Δ_{rb}(s+ q,t+ r, 1, N^F) and Δ_r(s+ q, t+ r, 1, N^F)* are less than a parameter λ_1 we say that condition A meets, then

$$K_r(s,t,1,N^F) = \begin{cases} K_r(s,t,1,N^F)+1, & condition\,A\,meets \\ K_r(s,t,1,N^F), & otherwise \end{cases} \tag{6.12}$$

For a window of size 3×3, *μ (K_r (s, t, N^F))* can be computed as:

$$\mu(K_r(s,t,1,N^F)) = \begin{cases} 1, & K_r(s,t,1,N^F) \le 2 \\ 0.5, & K_r(s,t,1,N^F) = 3 \\ 0, & K_r(s,t,1,N^F) \ge 4 \end{cases} \tag{6.13}$$

A pixel having a high count of comparable neighboring pixels and whose intensity level is similar to the median is more likely to be noise-free. It is denoted by the membership function '***μ**$_d$*', which represents the fuzzy set "**dissimilar**". It indicates the number of neighboring pixels with homogeneous intensity levels. After calculating the difference '*D'* and the amount of the pixels having comparable values for the central pixel in the current window (taking 3×3 here), '***μ**$_d$*' is calculated for each color component as:

$$\mu_d(s,\, t,\, 1,\, N^F) \;=\; \max\Big(\mu\big(K_r\big(s,\, t, 1,\, N^F\big)\big),\, \mu\big(D\big(s,\, t,\, 1,\, N^F\big)\big)\Big) \tag{6.14}$$

Equation (6.15) is used to calculate the membership function *μ (D (s, t, 1, N^F)).*

$$\mu(D(s,t,1,N^F)) = \begin{cases} 1, & D(s,t,1,N^F) \ge \lambda_2 \\ \dfrac{D(s,t,1,N^F) - \lambda_1}{\lambda_2 - \lambda_1}, & \lambda_1 \le D(s,t,1,N^F) < \lambda_2 \\ 0, & D(s,t,1,N^F) < \lambda_1 \end{cases} \tag{6.15}$$

Here D (s, t, 1, N^F) is the same as that calculated in equation (6.3) and the max operator used is the Cartesian co-product. Hence, if the value of a pixel is closer to the greater part of the pixels in the present window as well as comparable to the median value, thereupon it is recognized as a noise-free pixel. The parameters λ_1 and λ_2 are obtained experimentally. Similarly, the equations (6.5) - (6.15) can be defined for the other 2 color components.

Fuzzy Set: Terminal

Now let us take a window of size w× w (say 3×3). There are 9 pixels in this window if we arrange these nine pixels in increasing order of their intensity; we will get 2 terminal pixels and one median pixel. The 2 terminal pixels within the window are denoted by f_{min} having the smallest value and f_{max} having the highest value respectively and the median pixel is denoted by f_{med}. The closer the value of the central pixel of these terminal values f_{min} and f_{max}, the higher its probability of being noisy and if the value of the pixel is nearby to f_{med}, its probability of being noise-free rises. This is defined by the third membership function '$\boldsymbol{\mu_{ter}}$' which represents the fuzzy set "**terminal**". μ_{ter} is calculated for each color component as:

$$\mu_{ter}(f)=\left\{\begin{array}{ll} 1, & f = f_{\min}\ or\ f = f_{\max} \\ \dfrac{f-f_{\min}}{f_{med}-f_{\min}}, & f_{\min} < f < f_{med} \\ \dfrac{f_{\max}-f}{f_{\max}-f_{med}}, & f_{med} < f < f_{\max} \\ 0, & f = f_{med} \end{array}\right\} \quad (6.16)$$

Where f represents the pixel value of a given color component of the given frame.

Till now we have only focused on the color and spatial information. But since we are removing noise from a color image sequence, we also need to take care of the temporal information. If we need to ascertain the amount of noise existing in a component of a pixel in one frame, we need to consider its neighboring frames (previous and next frame) too. Therefore the next three membership functions $\mu_{h},$ $\mu_{v},$ *and* μ_{tem} are employed to calculate the amount of noise present in a component of the pixel in one frame by comparing the current frame with its neighboring frames.

Fuzzy Set: High

A noisy pixel component in one frame will not only have a high difference with its corresponding component neighboring pixels in the same frame but also with the relating component of the pixel on the equivalent spatial position in the preceding or upcoming frame. This approach is adopted in obtaining the fuzzy set "**high**" defined by the membership function μ_h. That is, if a component (let us say Red) of a pixel at *(s, t)* in the ongoing frame N^F has a high positive difference with either the relating component (Red) of a pixel on the equivalent spatial position *(s, t)* in the preceding frame N^F *-1* or with the relating component (Red) of a pixel on the equivalent spatial position *(s, t)* in the later frame N^F *+1*, it will be considered as High. The parameters γ_1 and γ_2 are again obtained experimentally. Let,

$$\mu(\mathrm{R}_0(s,t,1,N^F)) = \begin{cases} 1, & \mathrm{R}_0(s,t,1,N^F) \geq \gamma_2 \\ \dfrac{\mathrm{R}_0(s,t,1,N^F) - \gamma_1}{\gamma_2 - \gamma_1}, & \gamma_1 \leq \mathrm{R}_0(s,t,1,N^F) < \gamma_2 \\ 0, & \mathrm{R}_0(s,t,1,N^F) < \gamma_1 \end{cases} \qquad (6.17)$$

Where R_0 *(s,t, 1,* N^F*)* in equation (6.17) can either be replaced by R_{11}*(s,t, 1,* N^F*) or* R_{12}*(s,t, 1,* N^F*)* as defined below:

$$\mathrm{R}_{11}(\mathrm{s, t, 1}, N^F) = \left|\mathrm{f}\left(\mathrm{s, t, 1}, N^F\right) - \mathrm{f}\left(\mathrm{s, t, 1}, N^F\text{-}1\right)\right| \qquad (6.18)$$

$$\mathrm{R}_{12}(\mathrm{s, t, 1}, N^F\ ) = \left|\mathrm{f}\left(\mathrm{s, t, 1}, N^F\right) - \mathrm{f}\left(\mathrm{s, t, 1}, N^F+1\right)\right| \qquad (6.19)$$

Then, using *equation (6.17), (6.18) and (6.19),* μ_h is calculated for the red color component as:

$$\mu_{\mathrm{h}}\left(s,\ \mathrm{t},\ 1,\ N^F\right)\ =\ \max\begin{pmatrix}\mu\ \left(\mathrm{R}_{11}\left(s,\ \mathrm{t},\ 1,\ N^F\right)\right), \\ \mu\ \left(\mathrm{R}_{12}\left(s,\ \mathrm{t},\ 1,\ N^F\right)\right)\end{pmatrix} \tag{6.20}$$

The Max operator used is the Cartesian co-product.

Fuzzy Set: Variant

Similarity Criterion: Red Component

To find the similarity of the red component of a pixel at position *(s, t)* in the present frame N^F with the relating component of the pixel at the equivalent spatial position *(s, t)* within the preceding frame N^F *-1*, we have to compute the following differences.

The differences between neighboring pixels *(s+ q, t+ r)* of the pixel component in the present frame N^F and the neighboring pixels *(s+ q, t+ r)* of the relating component of the pixel on the equivalent spatial position within the preceding frame N^F *-1* is determined as:

$$\mathrm{d_r}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right)\ =\ \left|\mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 1,\ N^F\right)-\mathrm{f}\left(s+q,\ \mathrm{t}+r,\ 1,\ N^F-1\right)\right| \tag{6.21}$$

The second difference is computed from the following equation:

$$\mathrm{R}_3\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right)=\left|\mathrm{d_r}\left(s+q,\ \mathrm{t}+r,\ 1, N^F\right)\ -\ \mathrm{R}_{11}(\mathrm{s, t, 1}, N^F)\ \right| \tag{6.22}$$

Where R_{11} (s, t, 1, N^F) as defined in *equation (6.17) and (6.18).*

Provided that the given pixel component at *(s, t)* in the current frame N^F is akin to the corresponding pixel component at the equivalent spatial position *(s, t)* in the preceding frame N^F *-1,* then the neighboring pixels *(s+ q, t+ r)* of the pixel component in the ongoing frame N^F will also be similar to the neighboring pixels *(s+ q, t+ r)* of the corresponding pixel component at the equivalent spatial position in the preceding frame N^F *-1*. To find such some neighboring pixels around the given pixel a window of size w × w (*say 3×3)* is selected. We call this number as *k*. That is, *k* denotes the number of neighboring pixels *(s+ q, t+ r, 1,* N^F*)* in the current frame that is identical to corresponding neighboring pixels *(s+ q, t+ r, 1,* N^F *-1)* in the previous frame for the same color component. In the window of size 3×3, k will range from 0 to 8. This number k is calculated for each color component separately by using the parameter λ_1. This parameter λ_1 is the same as that used before. For the Red component,

$$k_r(s,t,1,N^F) = \begin{Bmatrix} k_r(s,t,1,N^F)+1, & R_3(s+q,t+r,1,N^F) < \lambda_1 \\ k_r(s,t,1,N^F), & otherwise \end{Bmatrix} \tag{6.23}$$

For a window of size $3\times3\ \mu\left(\mathrm{k_r}\left(s,\ \mathrm{t},\ 1, N^F\right)\right)$ is formulated as:

$$\mu(k_r(s,t,1,N^F)) = \begin{Bmatrix} 1, & k_r(s,t,1,N^F) \le 2 \\ 0.5, & k_r(s,t,1,N^F) = 3 \\ 0, & k_r(s,t,1,N^F) \ge 4 \end{Bmatrix} \tag{6.24}$$

Similarly, the other two membership functions $\mu\left(\mathrm{k}_g\left(s,\ \mathrm{t},\ 1, N^F\right)\right)$ and $\mu\left(\mathrm{k}_b\left(s,\ \mathrm{t},\ 1, N^F\right)\right)$ can be computed. More are the number of similar neighboring pixels; more are the chances of central pixel in the current frame being similar to the relating pixel in the previous frame and hence less will be the chances of the central pixel being noisy. A pixel with a high number of such similar neighboring pixels for the same color component and at least one of the other two components can be considered as noise-free. This is defined by the membership function μv. μv is used to express a fuzzy set "**variant"**, that implies the extent of similarity of the pixel at the center in the current frame to its corresponding pixel on the equivalent spatial position within the preceding frame. After calculating the number of pixels having comparable values for the pixel at the center in the current window has a size of w × w (taking 3×3 here), μv is calculated from the red color component as:

$$\mu_v\left(s,\ \mathrm{t},\ 1,\ N^F\right)\ =\ min\ \left(\mu\left(k_r\right),\ max\left(\mu\left(k_g\right),\ \mu\left(k_b\right)\right)\right) \tag{6.25}$$

For simplicity we have used k_r for k_r (s, t, 1, N^F), k_g for k_g (s, t, 1, N^F) and k_b for k_b (s, t, 1, N^F). The Max operator used is the Cartesian co-product and the min operator used is the Cartesian product.

Thus, using equation (6.4), (6.14), (6.16), (6.20) and (6.25) and applying them in equation number (6.2), we will get $N_1(s,t,1,\ N^F)$, that is:

$$N_1\left(s,\ \mathrm{t},\ 1,\ N^F\right)=min\left\{\begin{array}{c}\mu_b(D(s,\ t,1,\ N^F)),\ \mu_d\left(s,\ \mathrm{t},\ 1,\ N^F\right),\\ \mu_{ter}\left(\ \mathrm{s},\ \mathrm{t},\ 1,\ N^F\right),\\ \mu_h\left(\ \mathrm{s},\ \mathrm{t},\ 1,\ N^F\right),\ \mu_v\left(s,\ \mathrm{t},\ 1,\ N^F\right)\end{array}\right\} \tag{6.26}$$

the AND operator is used to implement the "*minimum"* operation. As we know N_1 *(s, t, 1,* N^F*)* represents the measure of noise that exists in the red component of a pixel at position *(s, t)* in the ongoing frame N^F. Its value should be zero for the noise-free pixel and the noisy pixel; it should have some value between 0 and 1.

Correction Term

The correction term is a term that is added to the original value of the pixel to make it noise-free. So if the pixel has been already classified as noise-free the correction term will be zero otherwise, for noisy pixel, correction term will have some positive or negative value. For the red component, the correction term is calculated using the following equation:

$$\begin{aligned}\Delta \mathrm{f}\left(s,t,1,N^{F}\right) &= N_1(s,t,1,N^{F})\times \\ &\left(\mathrm{med}\left(s,t,1,N^{F}\right) - \mathrm{f(s,t,1,}N^{F})\right)\end{aligned} \tag{6.27}$$

In the absence of noise, the pixel is noise-free hence the correction term will be zero and for extremely noisy pixels ($N_1=1$), the value of correction term is equivalent to the variation among the median (*med*) of noise-free pixels in the selected window and the noisy pixel.

This correction term obtained is then combined with the original value of the pixel. The pixels free from noise are left untouched and highly corrupted pixels are restored by using the median value computed from the adjacent noise-free pixels. Other pixels having an average noise density are modified according to the extent of noise present in them.

Rule 1:

IF ((| $f\,(s,t,1,N^F)$ - $med\,(s,t,1,N^F)$ | is ***Big)***

AND (the degree of similarity of $f\,(s,\ t,\ 1,\ N^F)$ to its neighboring pixel in N^F is ***Dissimilar)***

AND ($f\,(s,\ t,\ 1,\ N^F)$ is ***Terminal)***

AND ((either the difference between the $f(s,\ t,\ 1,\ N^F)$ in N^F and $f(s,t,1,\ N^F\ \text{-}1)$ in N^F *-1* is ***High)***

OR (the difference between $f\,(s,\ t,\ 1,\ N^F)$ in N^F and $f\,(s,\ t,\ 1,\ N^F\ +1)$ in N^F *+1* is ***High))***

AND (the degree of similarity $f(s,t,1,\ N^F)$ to its neighboring pixels in N^F *-1* is ***Variant)***

THEN $f\,(s,\ t,\ 1,\ N^F)$ is noisy.

Thus the result of Primary filter is:

$$\mathrm{f}_1\left(s,t,1,N^F\right) = \mathrm{f}\left(s,t,1,N^F\right) + \Delta\mathrm{f}\left(s,t,1,N^F\right) \qquad (6.28)$$

6.3.2 Secondary Filter

Now the output of the primary filter acts as an input to the secondary filter to give the final output. This step refines the output of the Primary filter by taking into account the interactions among the color components of the same frame as well as the interaction of the given frame with its previous frame as well as the next frame.

The objective of the secondary filter is the detection and rectification of residual noisy pixels. The noisy pixels are detected along with the amount with which they are corrupted. After detection, the pixels affected by noise are corrected via the median of the adjacent noise-free pixels.

Again in the secondary filter, the degree of noise is calculated by using a fuzzy rule. Consider one color component, say a red component of the current frame N^F in the resulting image sequence of the first step f_1. The extent of noise present in a pixel at location*(s, t)* is measured by the fuzzy *Rule2*:

***Rule 2*:**

IF (| f_1 *(s, t, 1,* N^F*)* - f_1 *(s, t, 2,* N^F*)* | is **Big**)

AND(| f_1*(s, t, 1,* N^F*)* - f_1 *(s, t, 3,* N^F*)*| is **Big**)

AND ((either the difference between f_1 *(s, t, 1,* N^F*)* in N^F and f_1 *(s, t, 1,* N^F *-1)* in N^F - *1* is **High**)

OR (the difference between *f $_1$ (s, t, 1, N^F)* in N^F and *f $_1$ (s,t, 1, N^F +1)* in N^F *+1* is **High**))

THEN *f_1 (s, t, 1, N^F)* is noisy.

Where *f(s,t,1, N^F)* is central pixel under evaluation, N^F denotes the current frame, N^F *+1* is the next frame and N^F *-1 its* preceding frame. *f(s, t,1, N^F -1)* denotes the relating pixel in the preceding frame, *f (s,t , 1, N^F +1*) is the corresponding pixel in the next frame, *f_1(s,t,2, N^F)* is the relating pixel from the blue component and *f_1(s,t,3, N^F)* is the corresponding pixel from the green component. Mathematically the amount of noise existing in the red color component of a pixel at position *(s, t)* in the current frame N^F is evaluated as:

$$N_2\left(s,\ t,\ 1,\ N^F\right)=min\left\{\begin{array}{l}\mu_b\left(\zeta_{rg}\left(\ s,\ t,\ 1,N^F\ \right)\right),\\ \mu_b\left(\zeta_{rb}\left(\ s,\ t,\ 1,N^F\ \right)\right),\\ \mu_h\left(\ s,\ t,\ 1,\ N^F\ \right)\end{array}\right\} \tag{6.29}$$

Where μ_b *and* μ_h *are* the same membership functions used above that represent the fuzzy sets “Big” and “High” respectively.

ζ_{rg} signifies the distinction between the red and green component and ζ_{rb} is the distinction between red and blue components and are computed as follows

$$\zeta_{rg}\left(\ s,\ t,\ 1,N^F\ \right)\ =\left|f_1\left(s,\ t,\ 1,\ N^F\right)-f_1\left(s,\ t,\ 2,\ N^F\right)\right| \tag{6.30}$$

$$\zeta_{rb}\left(\ s,\ t,\ 1,N^F\ \right)\ =\left|f_1\left(s,\ t,\ 1,\ N^F\right)-f_1\left(s,\ t,\ 3,\ N^F\right)\right| \tag{6.31}$$

After calculating these differences, we will compute the fuzzy membership value μ_b by comparing these differences with parameters β_1 and β_2. The values of these 2 parameters are obtained experimentally.

$$\mu_b\left(\zeta_{rg}\left(s,\ \mathrm{t},\ 1, N^F\right)\right)=\left\{\begin{array}{cc} 1, & \zeta_{rg}\left(s,\ \mathrm{t},\ 1, N^F\right)\geq \beta_2 \\ \dfrac{\zeta_{rg}\left(s,\ \mathrm{t},\ 1, N^F\right)-\beta_1}{\beta_2-\beta_1}, & \beta_1\leq\zeta_{rg}\left(s,\ \mathrm{t},\ 1, N^F\right)<\beta_2 \\ 0, & \zeta_{rg}\left(s,\ \mathrm{t},\ 1, N^F\right)<\beta_1 \end{array}\right\} \quad (6.32)$$

Similarly, other membership functions $\mu_b(\zeta_{rb}\left(s,\ \mathrm{t},\ 1, N^F\right))$ are derived. Now Let

$$\mu(\mathrm{R}_1(s,t,1,N^F))=\left\{\begin{array}{cc} 1, & \mathrm{R}_1(s,t,1,N^F)\geq\gamma_2 \\ \dfrac{\mathrm{R}_1(s,t,1,N^F)-\gamma_1}{\gamma_2-\gamma_1}, & \gamma_1\leq\mathrm{R}_1(s,t,1,N^F)<\gamma_2 \\ 0, & \mathrm{R}_1(s,t,1,N^F)<\gamma_1 \end{array}\right\} \quad (6.33)$$

Where R_1 (s, t, 1, N^F) in equation (6.33) can either be replaced by R_{21}(s, t, 1, N^F) or R_{22}(s, t, 1, N^F) as defined

$$\mathrm{R}_{22}(\mathrm{s, t, 1}, N^F) \ = \ \left|\mathrm{f}_1\left(s,\ \mathrm{t},\ 1,\ N^F\right) \ - \ \mathrm{f}_1\left(s,\ \mathrm{t},\ 1,\ N^F+1\right)\right| \quad (6.34)$$

$$\mathrm{R}_{21}(\mathrm{s, t, 1}, N^F) \ = \ \left|\mathrm{f}_1\left(s,\ \mathrm{t},\ 1,\ N^F\right) \ - \ \mathrm{f}_1\left(s,\ \mathrm{t},\ 1,\ N^F-1\right)\right| \quad (6.35)$$

Then, using equation (6.33), (6.34) and (6.35), μ_h is calculated from the red color component as:

$$\mu_{\mathrm{h}}\left(s,\ \mathrm{t},\ 1,\ N^F\right) \ = \ \max\left\{\begin{array}{c} \mu\left(R_{11}\left(s,t,1,N^F\right)\right) \\ \mu\left(R_{21}\left(s,t,1,N^F\right)\right) \end{array}\right\} \quad (6.36)$$

The Max operator used is the Cartesian co-product.The AND operator is used to implement the "*minimum*" operation. *N_1 (s, t, 1. N^F)* denotes the amount of noise existing in the red component of a pixel at *(s, t)* position in the present frame *N^F*. Its value should be zero for the noise-free pixel and the noisy pixel; it should have some value between 0 and 1. The correction term is derived for the red component using *equation (6.37):*

$$\begin{aligned}\Delta \mathrm{f}_2\left(s,t,1,N^F\right) &= N_1(s,t,1,N^F)\times \\ &\left(\mathrm{med}\left(s,t,1,N^F\right) - \mathrm{f}_1\left(s,t,1,N^F\right)\right)\end{aligned} \tag{6.37}$$

This correction term obtained is then combined with the original pixel value. The pixels free from noise are left untouched and highly corrupted pixels are restored through the median value of the adjacent noise-free pixels.

Other pixels having an average noise density are modified according to the amount of noise existing in them. Thus the result of the secondary filter is:

$$\mathrm{f}_2\left(s,t,1,N^F\right) = \mathrm{f}\left(s,t,1,N^F\right) + \Delta \mathrm{f}\left(s,t,1,N^F\right) \tag{6.38}$$

6.4 Experimental Results

A color image sequence comprising of *T* frames and *M* × *N* × 3 arrays of the pixel at location *(s, t)* can be seen as an assembly of three scale frames analogous with RGB components. *PSNR* (peak signal to noise ratio), MSE (Mean square Error) and SSIM (Structural similarity index measure) are selected as the measure of performance.

***Parameter Selection**:*

After the evaluation of our work on the following color image sequences: “Salesman”, “Tennis”, “Bus”, “Foreman”, “Flower”, and “Deadline” for impulse noise level ranging from ‘*p = 5%* to *p = 80%’*. The values of the membership function parameters were determined experimentally by taking the average of the *PSNR* result of ten frames of theses six color image sequences, corrupted by impulse noise at different noise levels. The parameters $\boldsymbol{\alpha_1 = 10, \alpha_2 = 10.5, \delta = 5, \lambda_1 = 0.1, \lambda_2 = 0.3, \gamma_1 = 10.5, \gamma_2 = 30, \beta_1 = 5}$ ***and*** $\boldsymbol{\beta_2 = 5.5}$ were selected experimentally as per the maximum value of the PSNR. These values provide optimal results to remove impulse noise from video sequences while restoring the basic structure of the

The efficacy of the proposed work is analyzed to that of *FRINR* [94], *FAPG*[116], and *AFISD* [99]. A denoising model based on Convolutional neural networks (*DMCNN*) [117], Structure-Adaptive Fuzzy Estimation for Random-Valued Impulse Noise Suppression (*SAFEIN*) [118] and Fast adaptive switching technique of impulsive noise removal in color images (*HFBFT*) [119]. All the results are simulated on standard image sequences available using MATLAB R2018b.

The proposed algorithm is examined on the following color image sequences: *"Salesman", "Tennis", "Bus", "Foreman", "Flower",* and "*Deadline*" for impulse noise levels *5%, 10%, 15%, 20%, 25%, and 30%* respectively. The result of this experiment in terms of PSNR for the 10th frame of all the standard sequences is presented in Fig. 6.2. Results deduce that the proposed method can handle distinct image sequences for varying noise levels. It provides the highest PSNR results as compared to other state of the art techniques.

From Fig. 6.2 , we can deduce that with the rise in noise density levels the performance of our work is not degraded. Apart from this, our work can be utilized for various kinds of image sequences in different fields like astrology, medical analysis, satellite imagery, military surveillance, and many more. It is multifunctional in nature. Although the latest advancements in noise removal based on neuro-fuzzy systems and optimization techniques, like DMCNN [117] and SAFEIN [118] excel in terms of good

PSNR results, they lag in terms of processing time. In real-world applications along with the noise removal capacity of the filter, the time taken by the algorithm to generate the final result is quite critical. Computation timings are also computed for all the methods in Table 6.1.

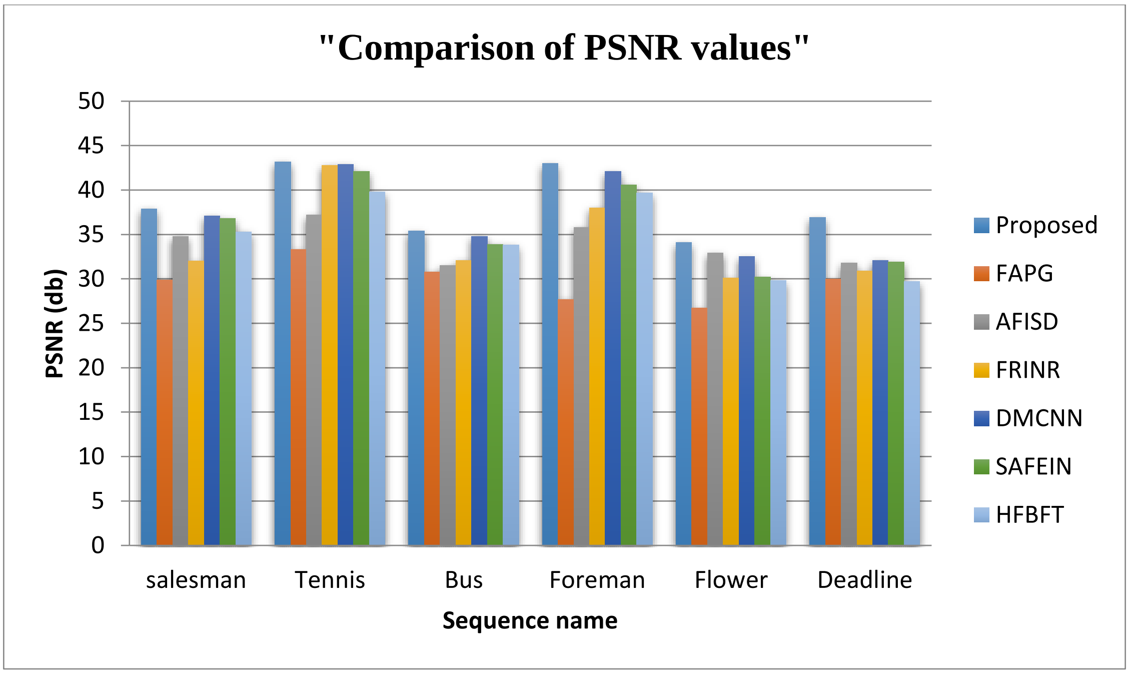


Figure 6.2 : Comparison graph of the PSNR values of the proposed filter with various methods

Table 6.1: Comparison of Computation time (in seconds) for various methods

	Noise Density %					
Method	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
FRINR [94]	11.73	11.98	12.97	13.80	15.79	17.99
SAFIN [118]	14.20	14.83	15.99	17.50	18.34	19.95
DMCNN[117]	15.52	16.28	18.93	19.52	20.70	21.43
FAPG [116]	10.94	11.46	11.95	13.06	14.13	15.89
HFBFT [119]	12.05	13.15	13.91	14.73	16.23	16.88
AFISD [99]	11.61	12.17	13.70	15.62	16.37	17.13
Proposed	**10.01**	**10.93**	**11.21**	**12.18**	**13.77**	**14.29**

Table 6.2: Comparison of SSIM values of various filters for different sequences

Noise %	**FRINR [94]**	**SAFEIN [118]**	**DMCNN [117]**	**FAPG [116]**	**HFBFT [119]**	**AFISD [99]**	**Proposed**
"Salesman"							
30	0.7308	0.7381	0.7601	0.6808	0.7091	0.6944	0.7626
40	0.7261	0.7304	0.7515	0.6771	0.7022	0.6873	0.7592
50	0.7140	0.7229	0.7394	0.6698	0.6945	0.6809	0.7403
60	0.7069	0.7153	0.7202	0.6583	0.6792	0.6793	0.7389
70	0.7002	0.7115	0.7096	0.6439	0.6773	0.6707	0.7311
"Tennis"							
30	0.9799	0.9759	0.9821	0.9792	0.9643	0.9788	0.9887
40	0.9700	0.9643	0.9795	0.9722	0.9611	0.9753	0.9814
50	0.9612	0.9586	0.9610	0.9669	0.9507	0.9667	0.9773
60	0.9514	0.9439	0.9545	0.9549	0.9313	0.9543	0.9744
70	0.9378	0.9403	0.9493	0.9467	0.9217	0.9416	0.9615
"Bus"							
30	0.9773	0.9811	0.9857	0.9761	0.9777	0.9815	0.9893
40	0.9681	0.9765	0.9764	0.9688	0.9706	0.9751	0.9819
50	0.9593	0.9710	0.9668	0.9604	0.9634	0.9644	0.9761
60	0.9516	0.9674	0.9517	0.9413	0.9572	0.9577	0.9591
70	0.9453	0.9501	0.9461	0.9339	0.9484	0.9411	0.9503
"Foreman"							
30	0.9537	0.9508	0.9578	0.9414	0.9479	0.9560	0.9601
40	0.9482	0.9472	0.9473	0.9377	0.9338	0.9513	0.9547
50	0.9407	0.9346	0.9417	0.9284	0.9249	0.9437	0.9501
60	0.9319	0.9279	0.9264	0.9114	0.9216	0.9331	0.9483
70	0.9202	0.9223	0.9202	0.9010	0.9177	0.9300	0.9446
"Deadline"							
30	0.9651	0.9577	0.9699	0.9606	0.9663	0.9689	0.9749
40	0.9518	0.9445	0.9641	0.9569	0.9585	0.9620	0.9698
50	0.9463	0.9389	0.9516	0.9493	0.9504	0.9550	0.9611
60	0.9372	0.9314	0.9437	0.9399	0.9470	0.9467	0.9560
70	0.9288	0.9266	0.9312	0.9318	0.9322	0.9378	0.9515

The results in ‘Table 6.1’ predict that the proposed method provides rapid output as compared to other methods in terms of processing time. The methods based on fuzzy logic like FAPG [94] provide faster results as compared to other methods. Though the qualitative results of the filters like DMCNN [117] and SAFIN [118] are good, they are slower in computation. This is a major drawback for techniques having complex structures and they are not apt for practical use. In the case of neural networks also, there is training difficulty and higher computation timings due to dependence on input qualities.

The proposed work has a simple structure and the use of fuzzy logic has enhanced the output of the filter. The computation timings are in practical range and the algorithm can be used with applications used for various multimedia devices.

Another performance metric used for the evaluation of our work is SSIM. The analysis of the SSIM values in ‘Table 6.2’, for various sequences predicts that the proposed algorithm is capable to handle varying amounts of noise levels. It is a robust method that maintains the basic structure of an image. The high SSIM results signify the better reliability of the proposed work. After noise removal, the proposed algorithm can preserve the true image pixels which lead to higher SSIM values.

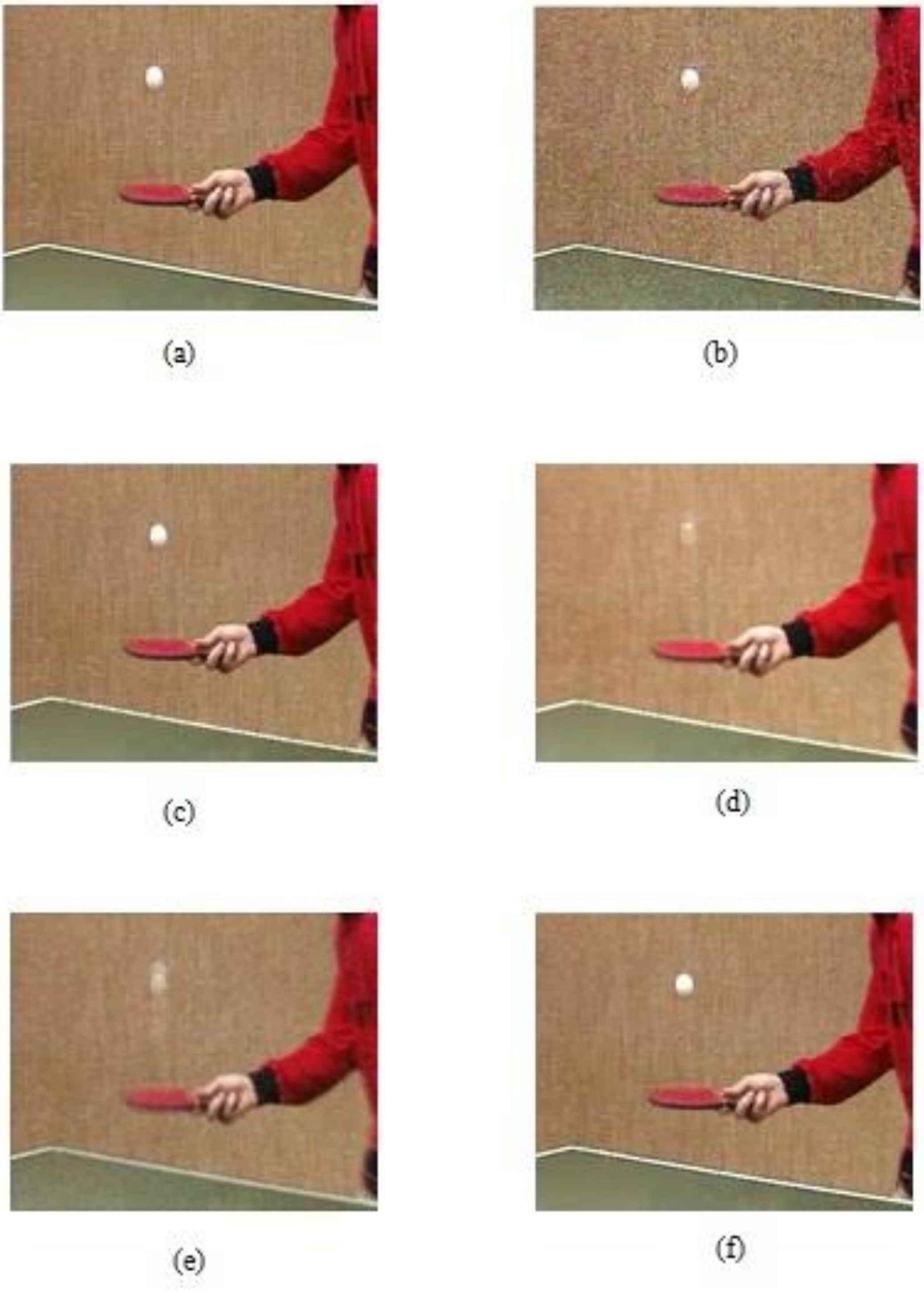
Various techniques face issues regarding over-detection and under detection of noisy pixels. False Alarm (FA) means when a noise-free pixel is marked as noisy. Miss detection exists (MD) when a noisy pixel is identified as noisy. MD and FA rates are calculated to analyze the efficacy of our filter in terms of its image conservation feature. False Alarm means when an un-corrupted pixel is marked as noise. Miss detection exists when a pixel with noise is detected as un-corrupted. We have used a difference based method [78] to compute the MD and FA rates. It generates a noise chart after computing the differences between the original image frame and the noise chart. The results are computed frame by frame with video sequences and average values are presented in this paper. To preserve the original image data, the values of MD and FA should be as low as possible. The results are shown in ‘Table 6.3’.

Results show that the proposed method surpasses other methods. The MD and FA rates are quite low which aids in the better reconstruction of the original image sequence. With the rise in noise levels, the chances of Miss detection and false alarm cases rise, but the results depict that the proposed filter is consistent in performance. Even at high noise levels above 50%, the filter detects noise in an optimum manner.

The extensive mathematical simulation results of the proposed work in ‘Fig 6.2’,‘Table 6.1’, ‘Table 6.2’ and ‘Table 6.3’, in terms of *PSNR, Computation time, SSIM, MD* and *FA values* respectively, validate the efficiency of the filter. All the quality metrics employed for the evaluation depict that the proposed filter is a potent method to remove impulse noise from a color image sequence. It can be utilized for various applications in the real world. It is rapid and feature conserving in nature. Simple structure, fast computation time, and high denoising capacity help to reduce the costs as well. It is practical and cheap as compared to other methods, which require separate hardware installations for various modules.

TABLE 6.3: MD & FA rates

Noise%	***Miss detection (MD)***				***False Alarm (FA)***			
	SAFEIN [118]	**DMCNN [117]**	**HFBFT [119]**	**Proposed**	**SAFEIN [118]**	**DMCNN [117]**	**HFBFT [119]**	**Proposed**
10	0	0	0	0	0	0	0	0
30	1	1	2	0	1	0	1	0
50	3	2	4	0	1	1	2	1
70	3	5	5	1	4	3	5	2
90	5	6	7	3	5	4	5	3



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.3 : 30th frame of the sequel “Tennis” (a) original (b) p=5% noisy (c) FRINR (d) AFISD (e) FAPG and (f) Proposed work

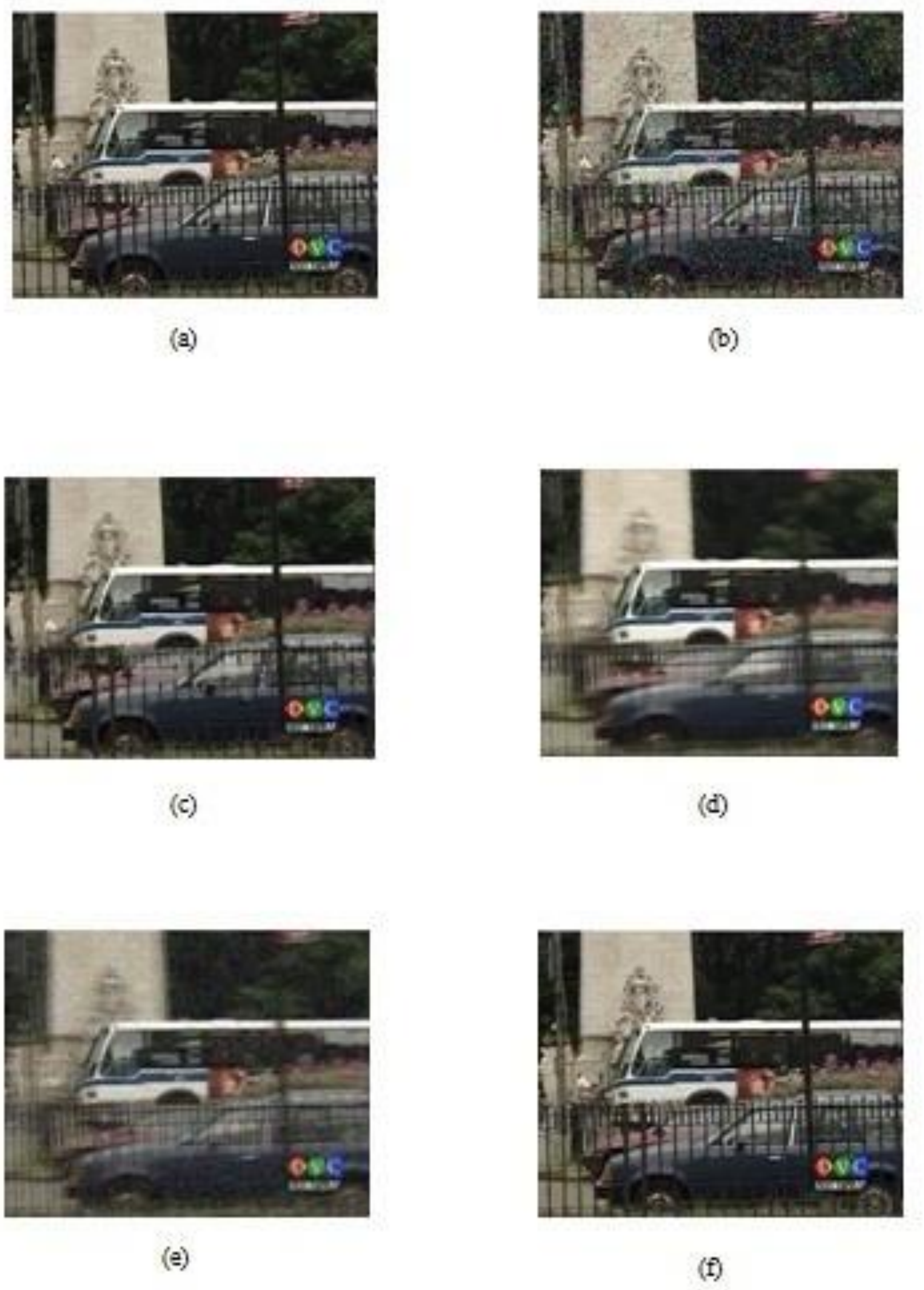
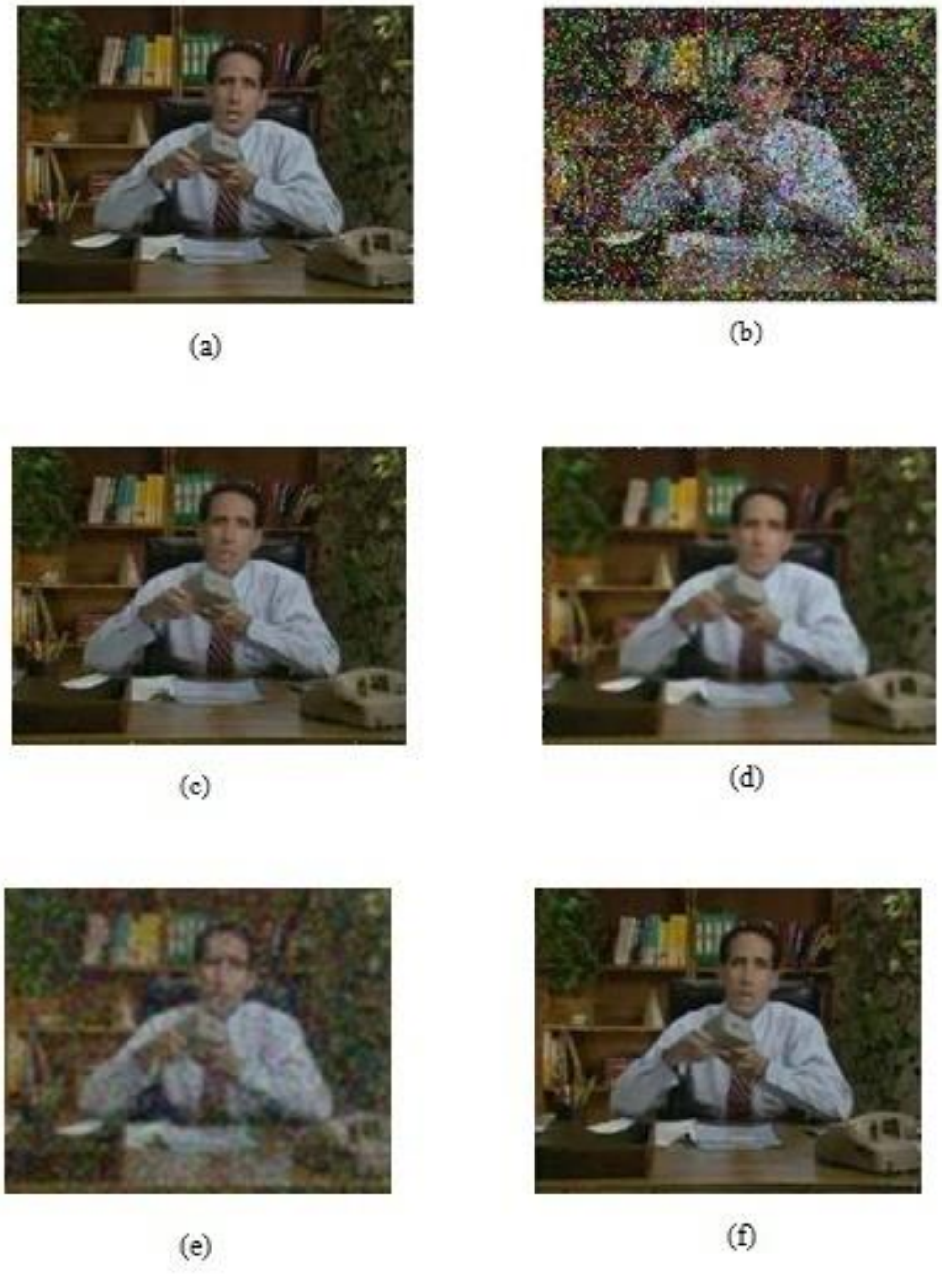


Figure 6.4 : 30th frame of the sequel “Bus” (a) original (b) p=10% noisy (c) FRINR

(d) AFISD (e) FAPG and (f) Proposed work



(a) (b) (c) (d) (e) (f)

Figure 6.5 : 30th frame of the sequel “Salesman” (a) original (b) p=20% noisy (c) FRINR (d) AFISD (e) FAPG and (f) Proposed work

Figures 6.3, 6.4 and 6.5 respectively represent the 30th frame of the sequel "Tennis" (p = 5%), the 30th frame of the sequel "Bus" (p = 10%) and the 30th frame of the sequel "Salesman" (p = 20%), the original frame, the noisy frame and the results deduced using *FRINR, AFISD, FAPG,* and *Proposed* method.

The proposed method outperforms these methods, both in objective measure and optical evaluation.

6.5 Conclusions

A new filtering schema for color image sequences debased by impulse noise is introduced in this chapter. To retain the details in an optimum manner, the noise is expelled well ordered. The assessment of noisy color components depends on fuzzy rules which incorporate the data from spatial and temporal neighbors including other color bands. Every color band is filtered independently. Two fuzzy subfilters are successively used to detect and filter the impulse noise. Since an impressive portion of the noise is filtered in the initial step, the remaining noise is easier to detect in the second step and hence the details are well preserved. Also, more reliable neighbors are used for comparison. Only noisy pixels are modified and noise-free pixels are left untouched, which minimizes the blurring. Furthermore, only noise-free pixels are used for filtering noisy pixels. The proposed method is characterized by superior results and stability in the surroundings with a broad area of impulse noise corruption. The experimental results demonstrate that the proposed filter result in an excellent alignment between noise removal and detail preservation. The proposed work surpasses other state-of-the-art techniques, both visually and mathematically.

Chapter 7

Conclusions and Future work

This thesis work has a prime focus on impulse noise suppression from images. Schemes for noise removal are devised for more enhanced results and optimum performance. The formulation of algorithms presented in this thesis contributes to better research of images used for multiple fields like satellite imaging, medical analysis, biometrics, astronomy, military surveillance and many more. These techniques are also compared with the state of art algorithms. Proposed algorithms excel in subjective as well as perceptible ways.

7.1 Summary of the work

To address the problem of impulse noise in images the first scheme called “High-density impulse noise detection using FCM algorithm” i.e. HDIND devised in this work concentrates on effective noise detection first and then its subsequent removal. The use of the FCM algorithm in our work has enhanced the detection schema and furthermore reduced the false alarm and miss detection rates. The algorithm provides good results for most of the images but with edge pixels rise in false alarm rates are seen. The work is extended to enhance the performance of the filter.

To remove the artifacts introduced in an image and reduce the false alarm rates in the case of HDIND, another scheme known as “Edge preserving fuzzy filter for suppression of impulse noise in images” is proposed. The EFFSIN deals with edge pixels more efficiently thereby reducing the false alarm rates as well. Double detection using FCM and edge-preserving nature of EFFSIN makes it a robust filter for impulse noise in images.

Another technique introduced in this thesis is a Novel Approach for Salt and Pepper Noise Removal based on Heuristic Analysis of Neighboring Pixels (SPHN). SPHN is a double-stage filter; the first stage identifies the noisy and noise-free pixels and the second stage performs filtering only on the noisy pixels. Not subjecting the noise-free pixels to the filtering process helps preserve fine details and avoids blurring.

All the above three schemes introduced till now remove impulse noise only from grayscale images. Lastly, we have attempted to remove noise from color image sequences.

A fuzzy filter for impulse noise suppression in color image sequences is a 3-D Spatio-temporal filter that considers spatial, temporal as well as the color information. It assesses noisy color components depending on fuzzy rules which incorporate the data from spatial and temporal neighbors including other color bands. Excellent alignment is seen between noise removal and structure conservation due to the classifying nature of the algorithm. It also outperforms other existing techniques present in literature.

The contributions of the work done in this thesis and their future scope are discussed in the next section.

7.2 Contributions and Future work

Images are a vital source of research for us. Image enhancement techniques and noise removal are among the dominant class of image processing. The contributions made by this thesis work are:

- We devised the algorithms for impulse noise suppression with potent detection schemes before the restoration of pixels. This decisive nature enhanced the performance of our filters.
- The proposed algorithms are robust in nature and capable to handle even the high-density noise thereby preserving the basic structure of the image.
- We developed the algorithms having a simple structure and intuitive nature which are more practical to be used with real-time applications.
- We created the algorithms using Fuzzy logic which helps us to use the techniques for various fields like defense & military surveillance, satellite images & remote sensing, astrology, criminology, medical image analysis and many more, where the user requirements and image dat are imprecise in nature.

In the future, schemes developed in this work for grayscale images can be expanded for color images and sequences. The work has explored the utility of median filters for restoration but in future other effective alternatives can be explored for the reconstruction of images.

Bibliography

[1] R.C.Gonzalez and R.E. Wood, “Digital Image Processing”, Addison Wisley, Second Edition 1992, First Impression 2009.

[2] A.Vyas, S. Yu, and J. Paik, “Image enhancement,” in Signals and Communication Technology, 2018.

[3] A.K.Jain, “Fundamentals of digital image processing”, Prentice-Hall of India,1989.

[4] S.Sridhar,“ Digital Image Processing”, Oxford University Press, 2011.

[5] F. Naderi, A. A. Sawchuk, "Estimation of images degraded by film-grainnoise", *Appl. Optics.*, vol. 17, pp. 1228-1237, Apr. 1978.

[6] A.Vyas, S. Yu, and J. Paik, “Fundamentals of digital image processing,” in Signals and Communication Technology, 2018.

[7] Goutsias, J, Vincent, L., and Bloomberg, D. S. (eds.), “Mathematical Morphology and Its Applications to Image and Signal Processing”, Kluwer Academic Publishers, Boston, MA. 2000.

[8] Bovick A, “Handbook of Image and Video Processing,” Academic Press, New York,2000.

[9] Patil, J. & Jadhav S., “A Comparative Study of Image Denoising Techniques,” International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, No. 3, 2013.

[10] Pawan Patidar, Manoj Gupta, Sumit Srivastava, and Ashok Kumar Nagawat, “Image De-noising by Various Filters for Different Noise”, International Journal of Computer Applications, Volume 9– No.4, November 2010.

[11] Mr. Salem Saleh Al-Amri, Dr. Khamitkar S.D, “A comparative study of Removal Noise from Remote sensing Image”, IJCSI International Journal of Computer Science Issues, Vol.7, Issue.1, January 2010.

[12] A. Vyas, S. Yu, and J. Paik, “Image restoration,” in *Signals and Communication Technology*, 2018.

[13] J. F. Abramatic, “Digital image restoration.,” *Fundam. Comput. Vis.*, 1983.

[14] Maria Petrou and Panagiota Bosdogianni, “Image Processing the Fundamentals”, John Wiley and Sons, 1st edition, 2003.

[15] S.Jayaraman, S.Esakkirajan, T.Veerakumar, “ Digital Image Processing”, Tata McGraw Hill Education Private Limited, 2009

[16] N Chithirala, A Radhakrishnan, PO. Amritanagar, "Weighted Mean Filter for Removal of High-Density Salt and Pepper Noise", *Advanced Computing and Communication Systems (ICACCS)*, pp. 1-4, 2016.

[17] Zhou Dengwen and ShenXiaoliu, — “Image Denoising Using Weighted Averaging”, International Conference on Communications and Mobile Computing, pp.400-403, 2009.

[18] William K. Pratt, “Digital Image Processing”, 4th edition, John Wiley & Sons, Inc., 2007.

[19] I. Pitas and A. Venetsanopou, Nonlinear Digital Filters: Principles and Application. Norwell, MA: Kluwer, 1990.

[20] Y. H. Lee, S. Tantaratana, "Decision-based order statistic filters", *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 406-420, Mar. 1990.

[21] J. Astola, P. Kuosmanen, Fundamentals of Nonlinear Digital Filtering, FL, Boca Raton: CRC, 1997.

[22] B. Chanda and D. Dutta Majumder,“ Digital Image Processing and Analysis”, Prentice-Hall of India, 1st edition, 2002.

[23] M. González-Hidalgo, S. Massanet, D. Ruiz-Aguilera, and A. Mir, “Improving salt and pepper noise removal using a fuzzy mathematical morphology-based filter,” Appl. Soft Comput., vol. 63, pp. 167–180, Feb. 2018.

[24] G. L. Sicuranza, “Quadratic filters for signal processing,” Proc. IEEE, 1992.

[25] John C. Russ, “The Image Processing Hand Book”, CRC Press, 4th Edition, 2002.

[26] U. Erkan, L. Gökrem, and S. Enginoglu, “Different applied median filter in salt and pepper noise,” Comput. Elect. Eng., vol. 70, pp. 789–798, Aug. 2018.

[27] W. K. Pratt, “Median filtering,” Tech. Rep., Image Proc. Inst., Univ. Southern California, Los Angeles, Sep. 1975.

[28] H. M. Lin and A. N. Wilson, Jr., “Median filters with adaptive length”, IEEE Trans. Circuits Svst., vol. 35, no. 6, June 1988.

[29] D.R.K. Brownrigg, “The weighted median filter, ACM Communications”, Vol. 8, No. 12, pp1834-1838, Dec 1999.

[30] S.J. Ko and Y.H. Lee, “Center weighted median filters and their applications to image enhancement,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, Vol. 38, No. 9, pp. 984-993, 1991.

[31] K.K. Ma, T. Chen, and L.H. Chen, “Tri-state median filter for image denoising,” IEEE TRANSACTIONS ON IMAGE PROCESSING, Vol. 8, No. 12, pp. 1834–1838, 1999.

[32] M.S. Nair and P.M.A. Mol, “Direction based adaptive weighted switching median filter for removing high-density impulse noise,” COMPUTERS AND ELECTRICAL ENGINEERING, Vol. 39, pp. 663-689, 2013.

[33] J. Astola, P. Haavisto, and Y. Neuvo, “Vector median filters,” PROCEEDINGS OF THE IEEE, Vol. 78, No. 4, pp. 678–689, 1990.

[34] J. Xu, L. Wang, and Z. Shi, “A switching weighted vector median filter based on edge detection,” SIGNAL PROCESSING, Vol. 98, pp. 359-369, 2014.

[35] G. Wang, D. Li, W. Pan, and Z. Zang, “Modified switching median filter for impulse noise removal,” SIGNAL PROCESSING, Vol. 90, No. 12, pp. 3213-3218, 2010.

[36] Roy, J. Singha, L. Manam, R. H. Laskar, "Combination of adaptive vector median filter and weighted mean filter for removal of high-density impulse noise from color images", *IET Image Process.*, vol. 11, no. 6, pp. 352-361, Jan. 2017.

[37] H. Hwang, R. A. Haddad, "Adaptive median filters: New algorithms and results", *IEEE Trans. on Image Processing*, vol. 4, pp. 499-502, 1995.

[38] K, Vigneshwari and Kalaiselvi, K, Adaptive Median Filter Based Noise Removal Algorithm for Big Image Data (International Journal of Advanced Studies of Scientific Research, Vol. 3, No. 10, Jan 2019.

[39] PAN T, WU X B, ZHANG W W, et al. “Application of Improved Adaptive Median Filter Algorithm in Image Denoising”, Journal of Logistical Engineering Unversity, 31(5):93-95,2015.

[40] WANG X K, LI F. “Improved Adaptive Median Filtering”, Computer Engineering and Applications, 46(3):175-176,2010.

[41] ZHU Y H, ZHANG J C, JIANG T “Iterative Adaptive Median Filter for Impulse Noise Cancellation”, SHANDONG SCIENCE, 23(4):52-54,2010.

[42] Ha R., Liu P., Jia K., “An Improved Adaptive Median Filter Algorithm and Its Application.” In: Pan JS., Tsai PW., Huang HC. (eds) Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, vol 64. Springer, Cham,2017.

[43] Khan, S., Lee, D. An adaptive dynamically weighted median filter for impulse noise removal. EURASIP J. Adv. Signal Process., 67,2017.

[44] Jiayi, C. H. E. N., Z. H. A. N. Yinwei, and C. A. O. Huiying. "Adaptive sequentially weighted median filter for image highly corrupted by impulse noise." *IEEE Access*,2019.

[45] Arce, Gonzalo R. "A general weighted median filter structure admitting negative weights." *IEEE Transactions on signal processing* 46, no. 12, 3195-3205,1998.

[46] Sun, Tong, Moncef Gabbouj, and Yrjö Neuvo. "Center-weighted median filters: some properties and their applications in image processing." *Signal processing* 35, no. 3 , 213-229,1994.

[47] Taha, Aina Qistina Md, and Haidi Ibrahim. "Reduction of Salt-and-Pepper Noise from Digital Grayscale Image by Using Recursive Switching Adaptive Median Filter."

In *Symposium on Intelligent Manufacturing and Mechatronics*, pp. 32-47. Springer, Singapore, 2019.

[48] Saleh, Abdurahman E., Faraj J. Zbida, and Salem A. Salem. "Modified Enhanced Recursive Median Filter to Image Denoising." In *International Conference on Technical Sciences (ICST2019)*, vol. 6, p. 04. 2019.

[49] Filter, Recursive Switching Adaptive Median. "Reduction of Salt-and-Pepper Noise from Digital Grayscale Image by Using Recursive Switching Adaptive Median Filter." In *Intelligent Manufacturing and Mechatronics: Proceedings of the 2nd Symposium on Intelligent Manufacturing and Mechatronics-SympoSIMM 2019, 8 July 2019, Melaka, Malaysia*, p. 32. Springer, 2019.

[50] Qiu, Gnoping. "An improved recursive median filtering scheme for image processing." *IEEE Transactions on Image Processing* 5, no. 4, 646-648,1996.

[51] P. Y. Chen and C. Y. Lien.: An efficient edge-preserving algorithm for removal of salt-and-pepper noise, IEEE Signal Processing Letters, vol. 15, pp. 833–836, 2008.

[52] K. K. V. Toh, H. Ibrahim, and M. N. Mahyuddin.: Salt-and-pepper noise detection and reduction using fuzzy switching median filter, IEEE Trans. Consumer Electron., vol. 54, no. 4, pp. 1956–1961, 2008.

[53] Kenny Kal Vin Toh.: Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction, IEEE signal processing letters, vol. 17, no. 3 pp 281-244,2010.

[54] Fei Duan and Yu-Jin Zhang.: A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters" IEEE Signal Processing Letters. vol. 17, pp. 647–650,2010.

[55] Umesh Ghanekar, Awadhesh Kumar Singh, and Rajoo Pandey.: A Contrast Enhancement-Based Filter for Removal of Random Valued Impulse Noise, IEEE Signal Processing Letters, vol. 17, no.1. pp. 47–50, 2010.

[56] S. Esakkirajan, T. Veerakumar, Adabala N. Subramanyam, and C. H. PremChand.: Removal of High-Density Salt and Pepper Noise through Modified Decision Based Unsymmetric Trimmed Median Filter, IEEE signal processing letters, vol. 18, no. 5 pp 287-290, 2011.

[57] Smail Akkoul, Roger Ledee, Remy Leconge, and Rachid Harba, "A New Adaptive Switching Median Filter", IEEE Signal Processing Letters, pp. 587-590. 2010.

[58] Nadeem, M., Ayyaz Hussain, Asim Munir, M. Habib, and M. Tahir Naseem. "Removal of Random Valued Impulse Noise from Grayscale images using Quadrant based Spatially Adaptive Fuzzy Filter." *Signal Processing*, 2019.

[59] Om Prakash Verma and Shweta Singh, "A Fuzzy Impulse Noise Filter Based on Boundary Discriminative Noise Detection", J Inf Process Syst, Vol.9, No.1, March 2013.

[60] Anuj Goel and Vikas Mittal, — "Removal of Impulse Noise Using Fuzzy Techniques: A Survey", International Journal of Applied Engineering Research, Vol.7 No.11, 2012.

[61] Huang, Yanwei; Qi, Binglu; Chen, Shaobin, "Modification of advanced boundary discriminative noise detection algorithm",10th IEEE International Conference on Control and Automation (ICCA-13), pp.961 – 966. 2013.

[62] Santosh, D.H.H.; Kumar, V.N.L.; Ramesh, D.R.; Prasad, P.S, " Efficient techniques for denoising of speckle and highly corrupted impulse noise images", 3rd IEEE International Conference on Electronics Computer Technology (ICECT-11), pp.253 – 257, 2011.

[63] Ahmed, F. and Das, S. "Removal of High-Density Salt-and-Pepper Noise in Images With an Iterative Adaptive Fuzzy Filter Using Alpha-Trimmed Mean", IEEE Transactions on Fuzzy Systems, Volume:22 , Issue: 5,2014.

[64] Anuj Goel and Vikas Mittal, "Removal of Impulse Noise Using Fuzzy Techniques: A Survey", International Journal of Applied Engineering Research, Vol.7 No.11, 2012 *on Signal Processing Image Processing and Pattern Recognition*, 2013.

[65] M. Bass, "Visual performance" in Handbook of Optics, New York: McGraw-Hill, 1995.

[66] A. M. Eskicioglu, P. S. Fisher, "Image quality measures and their performance", *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959-2965, 1995.

[67] R. Dosselmann, X. D. Yang, "Existing and emerging image quality metrics", *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 1906-1913, 2006.

[68] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[69] Sara, Umme, Morium Akter, and Mohammad Shorif Uddin. "Image quality assessment through FSIM, SSIM, MSE, and PSNR—A comparative study." *Journal of Computer and Communications* 7, no. 3, 8-18,2019.

[70] The Wikipedia the Free Encylopedia Website. http://en.wikipedia.org

[71] J. C. Dunn: "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3:32-57,1973

[72] J. C. Bezdek: "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York 1981.

[73] Russo, Fabrizio. "Recent advances in fuzzy techniques for image enhancement." *IEEE Transactions on Instrumentation and measurement* 47, no. 6, 1428-1434,1998.

[74] Toh, Kenny Kal Vin, and Nor Ashidi Mat Isa. "Cluster-based adaptive fuzzy switching median filter for universal impulse noise reduction." *IEEE Transactions on Consumer Electronics* 56, no. 4 ,2560-2568,2010.

[75] Zhou, Y.Y.; Ye, Z.F.; Huang, J.J, "Improved decision-based detail-preserving variational method for removal of random-valued impulse noise" Image Processing, IET, Volume: 6, Issue: 7, pp: 976 – 985, 2012.

[76] Benazir, T.M.; Imran, B.M, "Removal of high and low-density impulse noise from digital images using the nonlinear filter" International Conference on Signal Processing Image Processing & Pattern Recognition (ICSIPR-13), pp.229 – 233. 2013.

[77] González-Hidalgo, Manuel, et al. "High-density impulse noise removal using fuzzy mathematical morphology." 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13). Atlantis Press, 2013.

[78] Singh I, Verma O.P, "High Density Impulse Noise Detection using Fuzzy C-means Algorithm", Defence Science Journal, Vol. 66, No. 1, pp. 30-36, 2016.

[79] H.Y.Khaw,F.C.Soon,J.H.Chauh,C.O,Chow,"High density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation",Vol:13,Issue:2,*IET image processing*,2019.

[80] S. B. S. Fareed and S. S. Khader, “Fast adaptive and selective mean filter for the removal of high-density salt and pepper noise,” IET Image Process., vol. 12, no. 8, pp. 1378–1387, 2018.

[81] A.Roy, J.Singhha, R.H.Lasker,” Removal of Impulse noise from gray images using fuzzy SVM based Histogram fuzzy filter”, Journals of circuits, systems, and computers, Vol.27, No09,2018.

[82] Hussain, A. & Habib, A new cluster-based adaptive fuzzy switching median filter for impulse noise removal, M. Multimedia Tools Appl Springer 76: 22001. May 2017.

[83] Singh I, Verma O.P, “Edge Preserving Fuzzy Filter for Suppression of Impulse Noise in Images”, International Journal of Engineering and Advanced Technology, Vol.8, Issue6, August 2019.

[84] Verma, Om Prakash, Shreya Arora, and Isha Singh. "A novel approach for salt and pepper noise removal based on heuristic analysis of neighboring pixels." *International Conference on Computing for Sustainable Global Development (INDIACom),* IEEE, 2014.

[85] Ali 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, July 2011.

[86] C. Butakoff and I. Aizenberg. “Effective Impulse Detector Based on Rank Order Criteria”. IEEE Signal Processing Letters, 11(3):363 – 366, March 2004.

[87] Zhang, Xuming, and Youlun Xiong. "Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter." *IEEE Signal processing letters* 16, no. 4, 295-298, 2009.

[88] M. Nachtegael, D. Van der Weken, D. Van de Ville and E. E. Kerre,“Fuzzy Filters for Image Processing,” 1st ed., vol. 122, Heidelberg,Physica Verlag, 386 pages,2003.

[89] Om Prakash Verma, Madasu Hanmandlu, Anil Singh Parihar and Vamsi Krishna Madasu, “Fuzzy Filters for Noise Reduction in Color Images”, ICGST-GVIP Journal, Volume 9, Issue 5, September 2009.

[90] A. J. Rosales-Silva, F. J. Gallegos-Funes, V. I. Ponomaryov, "Fuzzy Directional (FD) Filter for impulsive noise reduction in color video sequences", *Journal of Visual Communication and Image Representation*, vol. 23, no. 1, pp. 143-149, 2012.

[91] V. Ponomaryov, F. Gallegos-Funes, A. Rosales-Silva, "Fuzzy directional (FD) filter to remove impulse noise from colour images", *IEICE Trans. Fundament. Electron. Commun. C omput. Sci.*, vol. E93-A, no. 2, pp. 570-572, 2010.

[92] T. Mélange, M. Nachtegael, E. E. Kerre, "Fuzzy random impulse noise removal from color image sequences", *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 959-970, 2011.

[93] M. E. Yuksel, A. Basturk, "Application of type-2 fuzzy logic filtering to reduce noise in color images", *IEEE Comput. Intell. Mag.*, vol. 7, no. 3, pp. 25-35, July 2012.

[94] S. Schulte, V. De Witte, M. Nachtegael, D Van der Weken, E.E. Kerre, "Fuzzy two-step filter for impulse noise reduction from color images", *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3567-3578, Nov. 2011.

[95] T. Mélange, M. Nachtegael, S. Schulte, E. E. Kerre, "A fuzzy filter for the removal of random impulse noise in image sequences", *Image and Vision Computing*, vol. 29, no. 6, pp. 407-419, 2011.

[96] Roy A, Manam L, Laskar RH. Region adaptive fuzzy filter, “ an approach for removal of random-valued impulse noise”, IEEE T Ind Electron ; 65: 7268-7278,2018.

[97] B. I. Justusson. Median Filtering: Statistical Properties. Two-Dimensional Signal Processing-II, T. S. Hwang Ed. New York: Springer Verlag, 1981.

[98] Chen, Jiayi, Yinwei Zhan, Huiying Cao, and Gangqiang Xiong. "Iterative grouping median filter for removal of fixed value impulse noise." *IET Image Processing* 13, no. 6 946-953,2019.

[99] M. Habib, A. Hussain, S. Rasheed, and M. Ali, “Adaptive fuzzy inference system based directional median filter for impulse noise removal,” *INTERNATIONAL JOURNAL OF ELECTRONICS & COMMUNICATIONS,* Vol. 70, No. 5, pp. 689-697, 2016.

[100] Vahid Kiani Abbas Zohrevand,” A Fuzzy Directional Median Filter for Fixed-value Impulse Noise Removal” 2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), April 2019.

[101] C.-C. Kang, W.-J. Wang, "Fuzzy reasoning-based directional median filter design", *Signal Process.*, vol. 89, no. 3, pp. 344-351, 2009.

[102] Yan, Z.: Adaptive Fuzzy Median Filter for Images Corrupted by Impulsive Noise. In: Congress on Image and Signal Processing, 2008.

[103] V.ThirilogasundariV. SureshbabuS. AgathaJanet,” Fuzzy Based Salt and Pepper Noise Removal Using Adaptive Switching Median Filter”, Volume 38, Pages 2858-2865, Procedia Engineering, ScienceDirect 2012.

[104] S. Khatri, H. Kasturiwale, "Quality assessment of Median filtering techniques for impulse noise removal from digital images", *IEEE 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1-4, 2016.

[105] George, Ginu, Rinoy Mathew Oommen, Shani Shelly, Stephie Sara Philipose, and Ann Mary Varghese, "A survey on various median filtering techniques for the removal of impulse noise from digital image", *Conference on Emerging Devices and Smart Systems (ICEDSS)*, pp. 235-238. IEEE, 2018.

[106] L. Guo, O. C. Au, M. Ma, and Z. Liang, "Temporal video denoising based on multi hypothesis motion compensation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 10, pp. 1423–1429, 2007.

[107] R. Rajagopalan and M. T. Orchard, "Synthesizing processed video by filtering temporal relationships," IEEE Transactions on Image Processing, vol. 11, no. 1, pp. 26–36, 2002.

[108] S.-W. Lee, V. Maik, J.-H. Jang, J. Shin, and J. Paik, "Noise-adaptive Spatio-temporal filter for real-time noise removal in low light level images," IEEE Transactions on Consumer Electronics, vol. 51, no. 2, pp. 648–653, 2005.

[109] E. J. Balster and R. L. Ewing, "Combined spatial and temporal domain wavelet shrinkage algorithm for video denoising," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 2, pp. 220–230, 2006.

[110] J.C. Brailean, R. P. Kleihorst, S. N. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk. Noise reduction filters for dynamic image sequences: A review. Proceedings of the IEEE, 83(9):1236–1251, September 1995.

[111] Li, Yueyang & Chung, Fu-Lai & Wang, Shitong. (2008). A robust neuro-fuzzy network approach to impulse noise filtering for color images. Applied Soft Computing. 8. 872-884. 10.1016/j.asoc.2007.07.006.

[112] S. Huang, Y. Peng, C. Chang, K. Cheng, S. Huang, and B. Chen, "Restoration of Images With High-Density Impulsive Noise Based on Sparse Approximation and Ant-Colony Optimization," in IEEE Access, vol. 8, pp. 99180-99189, 2020, doi: 10.1109/ACCESS.2020.2995647.

[113] Seong-Won Lee, Vivek Maik, Jihoon Jang, Jeongho Shin, and Joonki Paik, "Noise-adaptive Spatio-temporal filter for real-time noise removal in low light level images," in IEEE Transactions on Consumer Electronics, vol. 51, no. 2, pp. 648-653, May 2005, doi: 10.1109/TCE.2005.1468014.

[114] Szczepanski, M. Fast Spatio-temporal digital paths video filter. J Real-Time Image Proc 16, 477–489 (2019). https://doi.org/10.1007/s11554-016-0561-7.

[115] N. I. Chervyakov, P. A. Lyakhov, A. R. Orazaev and M. V. Valueva, "A New Method of Cleaning Video from Impulse Noise," 2019 International Conference on Engineering and Telecommunication (EnT), Dolgoprudny, Russia, 2019, pp. 1-5, doi: 10.1109/EnT47717.2019.9030589.

[116] Malinski, L., Smolka, B, “Fast averaging peer-group filter for the impulsive noise removal in color images”, J.Real-Time Image Processing, 2015.

[117] T. Le, P. Lin, and S. Huang, "LD-Net: An Efficient Lightweight Denoising Model Based on Convolutional Neural Network," in IEEE Open Journal of the Computer Society,2020, doi:10.1109/OJCS.2020.3012757.

[118] Y. Chen et al., "Structure-Adaptive Fuzzy Estimation for Random-Valued Impulse Noise Suppression," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 2, pp. 414-427, Feb. 2018, doi: 10.1109/TCSVT.2016.2615444.

[119] Arnal, J ; Súcar, L., “Hybrid Filter Based on Fuzzy Techniques for Mixed Noise Reduction in color images”, Applications sciences 2020. https:// doi.org/ 10.3390/app10010243.https://www.mdpi.com/2076-3417/10/1/243.