

10-7-2019

## Machine Learning And Image Processing For Noise Removal And Robust Edge Detection In The Presence Of Mixed Noise

Mehdi Mafi  
mmafi002@fiu.edu

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

---

### Recommended Citation

Mafi, Mehdi, "Machine Learning And Image Processing For Noise Removal And Robust Edge Detection In The Presence Of Mixed Noise" (2019). *FIU Electronic Theses and Dissertations*. 4354.  
<https://digitalcommons.fiu.edu/etd/4354>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

MACHINE LEARNING AND IMAGE PROCESSING FOR NOISE REMOVAL  
AND ROBUST EDGE DETECTION IN THE PRESENCE OF MIXED NOISE

A dissertation submitted in partial fulfillment of

the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Mehdi Mafi

2019

To: Dean John Volakis  
College of Engineering and Computing

This dissertation, written by Mehdi Mafi, and entitled Machine Learning and Image Processing for Noise Removal and Robust Edge Detection in the Presence of Mixed Noise, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

---

Mercedes Cabrerizo

---

Armando Barreto

---

Jean Andrian

---

Naphtali David Rishe

---

Malek Adjouadi , Major Professor

Date of Defense: October 7, 2019

The dissertation of Mehdi Mafi is approved.

---

Dean John Volakis  
College of Engineering and Computing

---

Andrés G. Gil  
Vice President for Research and Economic Development  
and Dean of the University Graduate School

Florida International University, 2019

© Copyright 2019 by Mehdi Mafi

All rights reserved.

## DEDICATION

I want to dedicate this dissertation:

To my dear parents Mrs. Parvin Mafi and Mr. Mohammad Mafi, who instilled in me the virtues of perseverance and commitment and relentlessly encouraged me to strive for excellence. There is no way for me to express utmost gratitude and thanks to you two. Through the good times and the bad, you have always been there for me guiding me on the right path. You two are the best role-models and parents I could have ever asked for.

To my very kind sisters - Mina, my little Melika, for their continuous support and encouragement. I just want you to know you mean the world to me. Richly blessed is how I feel having sisters just like you.

## ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my major advisor Prof. Malek Adjouadi who is one of the best professors around the world for his guiding and supporting me over the past 4 years through all the steps of my doctoral studies and research. He has set an example of excellence as a researcher, mentor, instructor, and role model. I have learned a lot from him, not only the fundamentals of my research area but also, I have learned life lessons, self-confidence and handling of tough professional situations. And I wish to give a special thank you to Dr. Mercedes Cabrerizo who I could rely on her help and advice whenever I encountered any problem.

Furthermore, I would like to thank my dissertation committee members Prof. Armando Barreto, Prof. Jean Andrian and Prof. Naphtali David Rishe for their help, support and accessibility.

It was a great pleasure for me to be part of a professional research team at the Center for Advanced Technology and Education (CATE). It was a great opportunity for me to improve my technical skills and knowledge which were precious experiences for my future career. Also, I should sincerely thank my student colleagues who helped me during my Ph.D. program.

We are grateful for the continued support from the National Science Foundation (NSF) support: NSF grants CNS-1920182, CNS-1551221, CNS-1532061, and CNS 1338922. The support from the Ware Foundation is also greatly appreciated. Furthermore, I should acknowledge the support from the university graduate school for offering me the Dissertation Year Fellowship (DYF) to support the writing stage of my dissertation and for publishing the findings of this research.

ABSTRACT OF THE DISSERTATION

MACHINE LEARNING AND IMAGE PROCESSING FOR NOISE REMOVAL  
AND ROBUST EDGE DETECTION IN THE PRESENCE OF MIXED NOISE

by

Mehdi Mafi

Florida International University, 2019

Miami, Florida

Professor Malek Adjouadi-Major Professor

The central goal of this dissertation is to design and model a smoothing filter based on the random single and mixed noise distribution to significantly attenuate the effect of noise while preserving edge details. Only then could robust, integrated and resilient edge detection methods be deployed to overcome the ubiquitous presence of random noise in images. Random noise effects are modeled as those that could emanate from impulse noise, Gaussian noise and speckle noise.

In the first stage of this dissertation, a thorough evaluation of methods is performed based on an exhaustive review on the different types of methods which focus on impulse noise and Gaussian noise along with related denoising filters that were designed to counter their effects. These include spatial filters (linear, non-linear and a combination of them), transform domain filters, neural network-based filters, numerical-based filters, fuzzy-based filters, morphological filters, statistical filters, and supervised learning-based filters.

In the second stage, switching adaptive median and fixed weighted mean filter (SAMFWMF), which is a combination of linear and non-linear filters, is introduced in

order to detect and remove impulse noise. Then, a robust edge detection method is applied relying on an integrated process including non-maximum suppression, maximum sequence, thresholding and morphological operations. The results are obtained on MRI and natural images.

In the third stage, a combination of transform domain-based filter which is a combination of dual tree – complex wavelet transform (DT-CWT) and total variation, is introduced in order to detect and remove Gaussian noise as well as mixed Gaussian and speckle noise. Then, a robust edge detection is applied in order to track the true edges. The results are obtained on ultrasound and natural images.

In the final stage, a smoothing filter based on a feed-forward convolutional network (CNN) is introduced to assume a deep architecture supported through a learning algorithm, an  $l_2$  loss function minimization, a regularization method, and batch normalization, all integrated in order to detect and remove impulse noise as well as mixed impulse and Gaussian noise. This process is followed with the deployment of a robust edge detection in order to track true edges in the different images considered. The results are obtained on natural images for both specific and non-specific noise-levels.

The significance of this work is evidenced through its many critical applications in (1) image segmentation, (2) object identification, (3) feature matching in stereo vision, (4) pattern recognition, (5) classification, (6) deriving structural and functional measurements in medical imaging, and (7) biometrics.

## TABLE OF CONTENTS

CHAPTER	PAGE
1. CHAPTER I .....	1
INTRODUCTION .....	1
1.1. Research Objectives .....	1
1.2. Random Noise .....	4
1.3. Structure .....	6
2. CHAPTER II .....	7
LITERATURE REVIEW.....	7
2.1. Impulse Noise Filtering .....	7
2.1.1 Spatial Non-Linear Filters .....	7
2.1.2 Spatial Combined Linear and Non-Linear Filters .....	9
2.1.3 Morphological Based Filters .....	10
2.1.4 Fuzzy Filters .....	10
2.2. Gaussian Noise Filtering .....	10
2.2.1 Spatial Non-Linear Filters .....	10
2.2.2 Spatial Linear Filters .....	12
2.2.3 Neural Network Based Filters .....	13
2.2.4 Fuzzy Filters .....	14
2.2.5 Combined Fuzzy and Morphological Filters .....	15
2.2.6 Statistical Filters .....	15
2.2.7 Transform Domain Based Filters .....	16
2.3. Mixed Impulse and Gaussian Noise Filtering .....	21

2.3.1	Spatial Non-Linear Filters .....	21
2.3.2	Spatial Combined Linear and Non-Linear Filters .....	21
2.3.3	Fuzzy Filters .....	22
2.3.4	Statistical Filters .....	22
2.3.5	Supervised Learning Algorithm Based Filters .....	24
2.3.6	Numerical Method Based Filters.....	24
2.3.7	Morphological Operation Based Filters .....	24
2.3.8	Transform Domain Based Filters .....	24
3.	CHAPTER III.....	27
	THEORY AND METHODOLOGY .....	27
3.1.	Spatial Filter Design for Impulse Denoising.....	27
3.1.1	Proposed Method for Impulse Denoising .....	29
3.1.1.1	Structure of the Method .....	32
3.1.1.2	Evaluation Measures .....	36
3.1.1.3	Experimental Evaluation in the Presence of Impulse Noise .....	39
3.1.2	Edge Detection After Spatial Filtering.....	42
3.1.2.1	Continuity in Edge and Thresholding in Grayscale Images.....	44
3.1.2.1.1	Non-Maximum Suppression .....	44
3.1.2.1.1	Maximum Sequence and Thresholding.....	45
3.1.2.2	Morphological Operations .....	48
3.2	Wavelet-Based Filter for Gaussian and Combined Gaussian-Speckle Denoising.....	51

3.2.1	Proposed Method for Gaussian and Combined Gaussian- Speckle Denoising.....	54
3.2.1.1	Description of the Method.....	54
3.2.1.2	Evaluation Measures .....	59
3.2.1.3	Structure of the Method.....	60
3.2.2	Edge Detection After Wavelet-Based Filtering .....	61
3.3	Design CNN Filter for Mixed Impulse and Gaussian Denoising .....	61
3.3.1	Proposed Method for Mixed Impulse and Gaussian Denoising .....	64
3.3.1.1	Evaluation Measures .....	64
3.3.1.2	Related Works on Denoising.....	65
3.3.1.3	Batch Normalization and Network Parameters .....	66
3.3.1.4	Network Model.....	66
3.3.2	Edge Detection After CNN Filtering .....	69
4.	CHAPTER IV.....	70
	RESULTS AND DISCUSSIONS .....	70
4.1.	Denoising Filters Comparisons .....	70
4.1.1	Impulse Denoising Filters .....	70
4.1.2	Gaussian Denosing Filters.....	70
4.1.3	Mixed Impusle and Gaussian Denosing Filters.....	72
4.2.	Impusle Denoising Based on Spatial Filter .....	73
4.2.1	Implementation on Natural Images .....	73
4.2.2	Implementation on Magnetic Resonance Imaging.....	83
4.2.3	Results After Edge Detection .....	86

4.3	Gaussian and Combined Gaussian - Speckle Denoising Based on Wavelet Filter .....	92
4.3.1	Combined Gaussian and Speckle Denoising.....	92
4.3.2	Gaussian Denoising.....	95
4.3.3	Results After Edge Detection.....	96
4.4	CNN Filtering.....	98
4.4.1	Mixed Impulse and Gaussian Denoising .....	98
4.4.2	Impulse Denoising.....	105
4.4.3	Results After Edge Detection.....	110
5.	CHAPTER V.....	113
	SUMMARY & CONCLUSIONS .....	113
	LIST OF REFERENCES .....	118
	VITA .....	136

## LIST OF TABLES

TABLE	PAGE
Table 1 – Peak signal to noise ratio comparison of Impulse denoising filters based on adaptive thresholding estimation for the Lena image example .....	19
Table 2 – Summarizes the key points and limitations one ought to consider in the implementation of the numerous filters for the Impulse and Gaussian filtering .....	19
Table 3 – Essential points and limitations to consider in the implementation of the numerous filters for the mixed Impulse and Gaussian filtering.....	26
Table 4 – Maximum window size of adaptive median filter in different noise levels on different images a) natural images b) images which contain significant black and white regions with clear edges like checkerboards.....	32
Table 5 – PSNR and Correlation comparison of proposed method with DT-CWT and stationary wavelet transform (SWT) on different images in the presence of different combined Speckle and Gaussian noise levels.....	57
Table 6 – PSNR, Correlation, and Structural Similarity Index Metric (SSIM) comparison of some of the discussed impulse denoising filters .....	71
Table 7 – Averaged PSNR comparison of some of the discussed Gaussian denoising filters (based on machine learning techniques).....	72
Table 8 – PSNR comparison of some of the discussed Gaussian denoising filters (based on spatial non-linear methods).....	72
Table 9 – Structural Similarity Index Metric (SSIM) comparison of some of the discussed Gaussian denoising filters.....	72
Table 10 – Averaged peak signal to noise ratio (PSNR) comparison of some of the discussed mixed impulse and Gaussian denoising filters .....	73
Table 11 – Averaged image perceptual quality index (FSIM) comparison of some of the discussed mixed impulse and Gaussian denoising filters .....	73
Table 12 – Averaged peak signal to noise ratio (PSNR) comparison of some of the discussed mixed impulse and Gaussian denoising filters .....	74
Table 13 – Averaged image perceptual quality index (FSIM) comparison of some of the discussed mixed impulse and Gaussian denoising filters .....	74
Table 14 – Execution time after proposed spatial filtering process.....	75

Table 15 – Correlation ( $\beta$ ) measured in comparison to the different initial adaptive median window sizes for the proposed spatial filter.....	76
Table 16 – Peak signal to noise ratio (PSNR) measured in comparison to the different initial adaptive median window sizes for the proposed spatial filter.....	76
Table 17 – Computed structural metrics using the checkerboard for different initial adaptive median window sizes for the proposed spatial filter .....	76
Table 18 – Correlation ( $\beta$ ) comparison for different denoising filters to the proposed spatial filter .....	77
Table 19 – Peak signal to noise ratio (PSNR) comparison for different denoising filters against the proposed spatial filter .....	77
Table 20 – Structural similarity (SSIM) comparison for different denoising filters against the proposed spatial filter.....	78
Table 21 – Computed structural metrics using the checkerboard for comparing the results obtained using different denoising filters to the proposed spatial filter.....	78
Table 22 – Correlation ( $\beta$ ) and the PSNR measures, comparing other Impulse denoising filters with and without fixed weighted mean filter as a post-processing step.....	78
Table 23 – FOM comparison between the proposed edge detection and the Canny edge detection algorithm.....	79
Table 24 – FOM comparison between proposed edge detection algorithm after proposed spatial filter denoising process with Canny edge detection algorithm after the same denoising process, and the proposed edge detection algorithm after UWMMF [39] denoising process with and without fixed weighted mean filter as a post processing step. ....	79
Table 25 – Summary of acronyms and the corresponding methodologies.....	80
Table 26 – Correlation ( $\beta$ ) measures for different filters against the proposed spatial filter (results for the proposed filter are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level) .....	84
Table 27 – Structural similarity index (SSIM) measures for different filters against the proposed spatial filter (results for the proposed filter are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level)..	84
Table 28 – Correlation ( $\beta$ ) measure, comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined Speckle and Gaussian noise intensities.....	93

Table 29 – PSNR measure, comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined Speckle and Gaussian noise intensities .....	93
Table 30 – Feature similarity index (FSIM), comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined Speckle and Gaussian noise intensities.....	93
Table 31 – FOM comparison between proposed wavelet-based filter denoising process with gradient-based edge detection process, and Lee-diffusion [39] with the same edge detection process.....	93
Table 32 – Average PSNR comparison for different mixed Impulse and Gaussian denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images.....	100
Table 33 – Average feature similarity index (FSIM) comparison for different mixed Impulse and Gaussian denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images .....	100
Table 34 – Average PSNR comparison for different mixed Gaussian and salt and pepper Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset.....	100
Table 35 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and salt and pepper denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset .....	100
Table 36 – Average PSNR comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on 12 test images.....	101
Table 37 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on 12 test images.....	102
Table 38 – Average PSNR comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images.....	102
Table 39 – Average feature similarity index (FSIM) comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images....	102
Table 40 – Average PSNR comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on BSD100 dataset.....	103

Table 41 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on BSD100 dataset.....	103
Table 42 – Average PSNR comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset.....	104
Table 43 – Average feature similarity index (FSIM) comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset. ....	104
Table 44 – Average peak signal to noise ratio (PSNR), average structural similarity index (SSIM), and averaged FSIM comparison between proposed CNN filter and AMFWMF [179] denoising process in presence of different Impulse noise intensities.....	109

## LIST OF FIGURES

FIGURE	PAGE
Figure 1 – Overall filter classification block diagram for the Impulse and Gaussian filtering.....	20
Figure 2 – Overall filter classification block diagram for the mixed Impulse and Gaussian filtering.....	25
Figure 3 – Essential processing steps for Impulse denoising and edge detection .....	29
Figure 4 – Flowchart of the Impulse denoising .....	37
Figure 5 – Correlation comparison between both switching methods on a) Lena b) Checkerboard.....	40
Figure 6 – Edge boundaries of different images after applying the spatial filter with switch 1 .....	41
Figure 7 – Correlation comparison between two states of fixed mean filter (with and without weights) on image Lena .....	42
Figure 8 – Different grayscale images with different kernels on image Camera man .....	44
Figure 9 – Performance of the maximum sequence to remove noisy pixels and track the edge lines .....	47
Figure 10 – Continuity along the edge lines in the image after applying maximum sequence.....	48
Figure 11 – Edge detection with different thresholding methods on image Lena.....	49
Figure 12 – Binary formatted morphological operations.....	51
Figure 13 – Morphological operations on Lena image and the specified area within the white rectangle is compared in the two different conditions .....	51
Figure 14 – Essential steps for Gaussian and combined Gaussian-Speckle denoising.....	54
Figure 15 – Stopping criteria for total variation .....	59
Figure 16 – Performance of the Wavelet-based algorithm with and without DT-CWT in the presence of combined Speckle and Gaussian noise on image Lena .....	59
Figure 17 – Essential steps for proposed DCNN based denoising .....	64

Figure 18 – Proposed DCNN model.....	68
Figure 19 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Lena .....	80
Figure 20 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Coins .....	81
Figure 21 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Camera man .....	81
Figure 22 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Lena .....	82
Figure 23 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Coins .....	82
Figure 24 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Camera man .....	83
Figure 25 – Edge boundaries and similarity of different MRI images after applying the proposed filter in the presence of high intensity noise .....	85
Figure 26 – Comparison in the presence of 80% impulse noise intensity on MRI images .....	85
Figure 27 – Comparison in the presence of 90% impulse noise intensity on MRI images .....	86
Figure 28 – Results obtained from the proposed method for different impulse noise levels on different MRI images.....	86
Figure 29 – Comparison on the camera man in the presence of 80% impulse noise intensity ....	88
Figure 30 – Comparison on the camera man in the presence of 90% impulse noise intensity ....	88
Figure 31 – Comparison on the Coins in the presence of 80% impulse noise intensity .....	89
Figure 32 – Comparison on the Coins in the presence of 90% impulse noise intensity .....	89
Figure 33 – Comparison on the Lena in the presence of 80% impulse noise intensity .....	90
Figure 34 – Comparison on the Lena in the presence of 90% impulse noise intensity .....	90
Figure 35 – Application of proposed spatial filter (using switch 2) and the proposed edge detection algorithm on the checkerboard image .....	91

Figure 36 – Application of the proposed spatial filter (using switch 1) and the proposed edge detection algorithm on the Lena image in presence of 95% Impulse noise intensity .....	91
Figure 37 – Comparison of the denoising filters in the presence of Speckle ( $\sigma = 0.1$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image. ....	94
Figure 38 – Comparison of the denoising filters in the presence of Speckle ( $\sigma = 0.2$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image. ....	95
Figure 39 – Application of the proposed wavelet-based filter in the presence of combined Speckle and Gaussian noise intensities on a medical ultrasound image .....	95
Figure 40 – Application of the proposed wavelet-based filter in the presence of different Gaussian noise intensities on different natural images .....	96
Figure 41 – Edge detection after applying the proposed wavelet-based filter in the presence of combined Speckle ( $\sigma = 0.2$ ) and Gaussian ( $\sigma = 0.1$ ) noise on a medical ultrasound image.....	97
Figure 42 – Edge detection after applying the proposed wavelet-based filter in the presence of Gaussian noise .....	97
Figure 43 – 12 Test Images .....	98
Figure 44 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 20 and 50 percent salt and pepper impulse noise on test image “Vase” .....	105
Figure 45 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 20 and 30 percent random value impulse noise on test image “Flower” .....	106
Figure 46 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 10, 40 percent salt and pepper impulse noise, and 10 percent random value impulse noise on test image “Boat” .....	106
Figure 47 – Application of the proposed CNN filter in the presence of different mixed Gaussian and salt and pepper Impulse noise intensities on different natural images .....	107
Figure 48 – Application of the proposed CNN filter in the presence of different mixed Gaussian and random value Impulse noise intensities on different natural images .....	108
Figure 49 – Application of the proposed CNN filter in the presence of different mixed Gaussian, salt and pepper Impulse, and random value Impulse noise intensities on different natural images .....	109
Figure 50 – Comparison of the denoising filters in the presence of 90 percent salt and pepper impulse noise on test image “Lena” .....	110

Figure 51 – Application of the proposed CNN filter in the presence of different salt and pepper Impulse noise intensities on different testing images .....	111
Figure 52 – Edge detection after applying the proposed CNN filter (specific and non-specific noise-level) in the presence of mixed Gaussian and salt and pepper Impulse noise.....	112
Figure 53 – Edge detection after applying the proposed CNN filter (non-specific noise-level) in the presence of mixed Gaussian and random value Impulse noise .....	112
Figure 54 – Edge detection after applying the proposed CNN filter (specific and non-specific noise-level) in the presence of mixed Gaussian, salt and pepper Impulse, and random value Impulse noise .....	112

## 1. CHAPTER I

### INTRODUCTION

#### 1.1. *Research Objectives*

Edge detection is a challenging nontrivial problem but is a task that remains essential for object identification, image segmentation, feature extraction, among other essential image processing tasks. A treatise on the “Theory of Edge detection is presented in [1] describing what constitutes a full primal sketch and defining what constitutes an intensity change over a wide-range of scales, and what optimal smoothing filter could effectively be used. Earlier experiments on information processing in the visual system was pioneered by Hubel and Wiesel (1962, 1968) [2] and later by Campbell & Robson (1968) [3]. Hubel and Wiesel, through inserted microelectrodes into the primary visual cortex of anesthetized cat and monkey, identified what they named as simple cells, complex cells and hyper-complex cells. Through the discovery of these cells, they elicited new understanding on how collectively they could construct composite edge representations of visual information from simple features extracted through orientation tuned line/slit detectors, motion detectors, and angle/corner detectors. Campbell & Robson took a different direction, and through experiments involving a variety of grating patterns over a wide range of spatial frequencies, suggest “the existence within the nervous system of linearly operating independent mechanisms selectively sensitive to limited ranges of spatial frequencies,” akin to Fourier transforms.

There are several methods and well-known operators that are commonly used to detect edges in images, and their success is often weighted as a function of the application at hand. Contentious issues remain with thresholding, contrast and scale issues for which an edge

point is deemed to be a true edge point. When we deal with images, pertinent details can be useful when analyzing specific imaging data, but the concern has always been in delineating what really constitute edge data with a high degree of similarity and correlation in contrast to other background and noise data. The challenge is further amplified when the images are degraded by noise, affecting significantly the structural metrics and the signal to noise ratio measure.

In this dissertation smoothing filters based on types of noise distributions that we have to contend with are created with the aim to attenuate the effect of noise while preserving as much edge details as possible. The smoothing filters are based on a combination of linear and non-linear filters for impulse denoising, combination of transform domain based and non-linear based filters for Gaussian as well as mixed Gaussian and speckle denoising, convolutional neural network (CNN) with very deep architecture (deep learning) for impulse and mixed impulse and Gaussian denoising. The results obtained are contrasted to other well-known denoising filters by using different structural metrics and evaluation measures that would gauge the degree of edge preserving by means of correlation and signal to noise ratio (SNR). Then, a robust and integrated edge detection method that is resilient to the presence of noise in images, is applied in order to detect the true edges. There is however a tradeoff (or a balance) that needs to be struck between the image smoothing operation which is to attenuate the effect of noise and the edge detection process that should preserve edge details and minimize any presence of potential noise points. The smoothing algorithm is thus designed based on the assumed model of the random mixed noise. The edge detection phase, on the other hand, is based on the gradient that is applied to the smoothed/denoised image with additional processing steps that are created for

optimized thresholding, edge thinning to overcome the blur introduced through smoothing, appropriate edge tracking to overcome discontinuities, and maximized sequencing to eliminate any remaining noise points due to predefined thresholds in the presence of high density noise.

Edge detection methods that are immune or resilient to noise allow for enhanced image segmentation, object recognition, feature extraction and pattern classification. The significance of edge detection is evidenced through its many critical applications in (1) object identification, (2) feature matching in stereo vision [4, 5], (3) pattern recognition and classification, (4) deriving structural and functional measurements in medical imaging [6], (5) biometrics [7], and (6) image segmentation, among many other real-world applications that can be contemplated. For illustrative examples that highlight the significance of this work, our experience with medical images clearly show that effective edge detection could improve (1) segmentation of tumors in PET images which have low resolution and suffer from inherent noise in the image [8]; (2) Estimating anatomic liver volumes towards selective internal radiation treatment (SIRT) [9], where the ratio of tumor to liver is essential in determining the radiation dose.

In more specific terms, the significance of the work of this dissertation could also be measured in terms of (1) allowing researchers that deal with noisy images to be familiar with different types of methods and algorithms that best investigate the types of noise that they confront, (2) providing denoising method for images containing single and mixed noise, and (3) introducing an integrated process with its related software that maximize the preservation of edge details and minimizes the signal to noise ratio. Moreover, the significance of such a design is also elevated in view of its simplicity of use, flexibility,

and extensibility with users still having the full potential to add and/or improve the software modules in seeking an optimal outcome under different circumstances and conditions of the application at hand.

## 1.2. *Random Noise*

The term noise is used to describe any unwanted and/or random phenomenon that may degrade an image, distorting its original content and burdening any preprocessing step that may be undertaken. There exists a plethora of noise sources that can affect images, some of which are controllable by the potential means of undoing (reversing) their effect, while others are extremely difficult to formulate and hence less obvious for overcoming their effect. Some common sources of noise include image sensors, scanners, optic defects, relative motion, shot noise, atmospheric turbulence, among others. Impulse noise is caused by A/D converter saturation, transmission errors, memory errors, and faulty pixels in camera sensors resulting in black pixels in white regions and white pixels in black regions [10-11]. Impulse noise, also known as salt and pepper noise, is represented by equation (1), where  $c_{\min}$  and  $c_{\max}$  are minimum and maximum values which are 0 and 255 in the standard 8-bit pixel resolution images [12].

$$I(c) = \begin{cases} I_{\min} & \text{Probability } P_s \\ I_{\max} & \text{Probability } P_p \\ I_{\min} < c < I_{\max} & \text{Probability } 1 - P_p - P_s \end{cases} \quad (1)$$

In this model,  $c$  denotes the uncorrupted pixels, and where the corrupted pixels are assigned probability  $P_s$  for salt and  $P_p$  for pepper. In this normalized representation of the image, 0 being the minimum intensity denoted by  $I_{\min}$  and 1 being the maximum intensity denoted by  $I_{\max}$ .

Also, the random value model appears as uniformly-distributed random numbers between a minimum and maximum interval  $[n_{min}, n_{max}]$  and is expressed by equation (2).

$$I_c(normalized) = \begin{cases} n & \text{Probability } p \\ c & \text{Probability } 1 - p \end{cases} \quad (2)$$

In this model,  $c$  denotes the uncorrupted pixels, and the corrupted pixels are assigned probability  $P$ .

Gaussian noise, represented by a Gaussian distribution function, is additive and independent, and is caused by 3 common factors: amplifier noise, shot noise, and grain noise of film [13-15]. Accordingly, the noisy image can be expressed as

$$I_n(i, j) = I(i, j) + n(i, j) \quad (3)$$

Where  $I_n$  represents the noisy image,  $I$  is the original (noise free) image and  $n$  is the additive noise on a pixel basis.

Speckle noise is one type of noise that is multiplicative and independent. It is the result of interference between returning light from rough surfaces and the aperture creating a granular shape pattern in the camera sensor. This type of noise affects both the resolution and contrast in ultrasound images. The general model of speckle noise [16] contains multiplicative and additive components, but in ultrasound images, the additive part can be overlooked [17]. Using a logarithmic transform, the multiplicative noise is converted to additive noise. Speckle noise is assumed to have a generalized gamma (GG) distribution as in (4).

$$p_z(z) = \frac{\gamma z^{\gamma v - 1}}{\alpha^{\gamma v} \Gamma(v)} e^{-\left(\frac{z}{\alpha}\right)^{\gamma}} \quad z \geq 0, \alpha, v, \gamma > 0 \quad (4)$$

Where  $z$  is a random variable which represents a pixel value.  $\Gamma(v)$  is the gamma function,  $v$  and  $\gamma$  are shape parameters, and  $\alpha$  is a scale parameter.

### 1.3. *Structure*

Chapter 1 introduces the research objectives of this dissertation and relates the importance of image denoising and edge detection. The mathematical foundation of random noise in images is presented, and a structure of the main research themes covered are discussed subsequently.

Chapter 2 provides an exhaustive literature review describing the impulse, Gaussian, and mixed impulse and Gaussian denoising filters and a classification of them is shown to depend on the method used.

Chapter 3 introduces filters designed to remove random noise. These filters include 1) spatial filter for impulse denoising 2) wavelet-based filter for Gaussian, speckle, and mixed Gaussian and speckle denoising 3) Convolutional neural network (CNN) filter with very deep architecture for impulse and mixed Impulse and Gaussian denoising. Finally, gradient based edge detection methods are applied in order to detect the true edges.

Chapter 4 provides the implementation results of 1) spatial filter for impulse denoising as applied on MRI and natural images, 2) wavelet-based filter for Gaussian and mixed Gaussian and speckle denoising also applied on MRI and natural images, and 3) specific and non-specific noise-level convolutional neural network (CNN) for impulse and mixed impulse and Gaussian denoising applied on natural images. Finally, after each of the filtering methods, edge detection results are given to gauge the merits of each of the denoising methods.

Chapter 5 provides concluding remarks, highlighting the merits of the proposed denoising and edge detection methods, and provides a perspective on future research endeavors.

## 2. CHAPTER II

### LITERATURE REVIEW

This chapter provides a comprehensive survey on state-of-the-art impulse denoising filters [18], Gaussian denoising filters [18], and mixed impulse and Gaussian [19] applied to images and summarizes the progress that has been made over the years in several applications involving image processing. The random noise model in this survey is assumed to be comprised of impulse (salt and pepper) and Gaussian noise. Different noise models are addressed, and different types of denoising filters are studied in terms of their performance on digital images and in their various domains of application. A comparison is performed to cover relevant denoising methods and the results they yield.

#### *2.1 Impulse Noise Filtering*

##### *2.1.1 Spatial Non-Linear Filters*

Spatial filters are obviously defined in the normal 2-D image space, where the intensity of each pixel is adjusted based on its original value and that of its neighbors. In this case, the filter output is a non-linear function of its inputs. They can be divided into median filters, weighted averaging filters, and non-local mean filters.

Median filters convolve a window of a determined size (referred to as a moving window) over the image to determine whether the pixel at its center is corrupted or not. When a pixel is deemed corrupted, its value is replaced with the current window's median value. When the noise intensity increases, the size of the window must be increased to compensate for this intensity increase. The median filters can be used in different formats, allowing for specific improvements. Such filters include: 1) The adaptive median filters exploit the adaptive property of the scanning window as exemplified in the adaptive median

filter (AMF) [20], the filter in [21] which is a combination of AMF and an iterative minimization, and the simple adaptive median filter (SAMF) [22]; 2) Threshold filters, which use predefined thresholds in their noise detection phase such as pixel wise median absolute deviation filter (PWMAD) [23]; 3) Switching filters that use a switching process to select the optimal output, either in the noise detection phase or in the noise correction phase. Such filters include progressive median filter (PSMF) [24], noise adaptive soft-switching median filter [25], directional difference-based switching median filters (DDBSMF) [26], Impulse detector switching median filters (ISMF) [27], adaptive switching median filter (ASMF) [28] and sorted switching median filter (SSMF) [29]; 4) Weighted and multi-states-based filters that use different threshold comparisons and consequently different states for noise detection and correction. The tri-states median filter (TSMF) [30], MSMF [31], directional weighted median filter (DWMF) [32] are different types of weighted and multi-states filters; 5) Decision filters, which assume that corrupted pixels have a value of 0 or 255 and uncorrupted ones have a value between them. They include decision based median filters [33], new based decision algorithm (NEDBF) [34], and decision base unsymmetrical median filter (DBUTMF) [33]; and 6) Adjusted median filters are types of filters that use an adjusted median value to replace the corrupted pixels, and boundary discriminative noise detection filter (BDNDF) [35], and IBDNDF [36] which is an improvement on BDNDF.

Weighted averaging filters employ a multi-criteria weighted mean value to correct the corrupted pixels. They include switching adaptive weighted mean filter (SAWMF) [37], adaptive weighted mean filter (AWMF) [38], unbiased weighted mean filters (UWMF)

[39], the cloud generator-based filter in [40], and interpolation-based impulse noise removal filter (IBINRF) in [41].

Non-local mean filters are used for Gaussian denoising. The iterative non-local mean filter (INMF) in [42] is a combination of a switching median filter and a non-local mean filter. It identifies the corrupted pixel within the selected window and replaces it with the median value of the uncorrupted ones. Once the corrupted pixels are identified and replaced, the filter further improves the approximation of the corrupted pixel's value by using an iterative non-local means (NLM) algorithm.

### 2.1.2 *Spatial Combined Linear and Non-Linear Filters*

They can be divided into the following types: combined median and mean filters, and combined median and mid-point filters.

Combined median and mean filters, as their name implies, assume a certain combination of median and mean filters. Such filters include decision based algorithm filters (IDBAF) as described in [43], cascading algorithm combining a decision-based median filter and an asymmetric trimmed mean filter DMF+UTMF as introduced in [44], modified decision based unsymmetrical trimmed median filters (MDBUTMF) as applied in [45], cascade decision-based filtering algorithms [46], decision based partial trimmed global mean filters (DBPTGMF) [47], decision based adaptive neighborhood median filters (DBANMF) [48], decision based unsymmetrical trimmed modified winsorized mean filters (DBUTMWMF) [49], and denoising filter in [50] that utilizes two cascading algorithms, which are special types of spatial combined linear and non-linear filters.

Combined median and mid-point filters, as their name implies, is likewise a combination of median and mid-point filters. They include cascading algorithms that

combine decision-based median filter and unsymmetric trimmed midpoint filtering DMF+UTMP as described in [44].

### *2.1.3 Morphological Based Filters*

Morphological operations are non-linear and not related to numerical values. They include generalized directional morphological filters (GD MF) [51], Open-close sequence filters (OCSF) [52], and the proposed denoising and enhancement filter for salt and pepper noise [53].

### *2.1.4 Fuzzy Based Filters*

These types of filters are based on fuzzy rules and they include 1) switching based filters such as adaptive fuzzy switching filter (AFSF) [54]; 2) Gradient based filters such as fuzzy impulse noise detection and reduction method (FIDRM) as proposed in [55], and gradient detection fuzzy filters (GDFF) [56] [57]; and 3) Histogram based filters explored in [58].

## *2.2 Gaussian Noise Filtering*

### *2.2.1 Spatial Non-Linear Filters*

Spatial Non-linear filters simply have outputs dictated by a non-linear function of their inputs. They can be divided into total variation filters, anisotropic diffusion filters, non-local mean filters, bilateral filters, fourth order partial differential filters, and Kuwahara filters. Total variation methods make use of the total variation measure and assume that the integral of the signal gradient to be high. Therefore, by decreasing total variation, a denoised image with high similarity is obtained. Total variation was first introduced in [59] with the assumption that spurious effects in the image contribute greatly to this variation measure. An improved total variation method is applied to the image to smooth it and

remove the remaining noise [60], especially from high frequency sub-bands. Also, other types of improvements to total variation are further described in [61-63] with a non-uniform total variation partition filter (NTVVPF) introduced in [63].

Anisotropic diffusion filters reduce image noise and preserves edge details by using non-linear and space-variant transforms. Algorithms that use a diffusion process and a new definition of scale-space are introduced in [31]. Then, improvements on anisotropic diffusion are presented in [65] and [66]. A specific improvement on anisotropic diffusion as shown in [67] attains better noise removal by discriminating between fine details and noise while preserving edges and details by using local gradients and gray-level variance. Another type of improvement on anisotropic diffusion filter presented in [68] applies a Gaussian filter to the moving window to reduce noise.

Non-local similarity-based filters use several similar patches to reconstruct the patch being processed. The non-local mean (NLM) filter presented in [69] uses non-local averaging to preserve image edges and self-predictions to replace noisy pixels with the mean weighted average of the pixels with similar Gaussian neighbors. The authors of [70] postulate that non-local mean (NLM) filters can emerge from a Bayesian approach with new arguments and the authors in [71] reduce the NLM [69] algorithm's time complexity. The exponential term of the weight function of NLM is improved in [72]. Also, there are other improvements on NLM such as iterative based NLM filter in [73] as well as with the method presented in [74].

Bilateral filters are non-linear denoising methods aimed at edge preservation. Corrupted pixels are replaced with the weighted Gaussian-based average of its neighboring pixels depending on domain and range distance to reduce phantom edge color. Bilateral filters are

composed of a combination of domain (responsible for geometry closeness) and range (responsible for similarity intensification) filter(s) [75]. The improved bilateral filter presented in [76] achieve good high-noise rejection performance. Also, the time complexity of the filter is reduced by using the faster algorithms presented in [77] and [78].

Fourth order partial differential equations (4<sup>th</sup> PDE) are used to optimize noise removal and edge preservation by minimizing the cost function (absolute value of the image's Laplacian). The 4<sup>th</sup> PDE provided in [79] performs very well, avoiding the blocky artifacts that appear in the early stages of diffusion when smoother areas diffuse faster than the less smooth ones. The improved PDE proposed in [80] achieves higher detail preservation, even in low SNR conditions. The authors of [81] propose an improved 4<sup>th</sup> PDE to set the diffusivity functions that controls the diffusion along the gradient direction, achieving fast convergence filtering with better edge preserving performance.

Kuwahara filters are non-linear denoising filters that preserve edge details. They divide the 3x3 moving window into four sub-windows, calculate their means and standard deviations, and use the mean of the window with the smallest standard deviation to replace the corrupted pixel at the center of the original window [82]. However, Kuwahara filters have some important limitations as discussed in [83].

### 2.2.2 *Spatial Linear Filters*

Spatial linear filters include mean filters, Gaussian filters and Wiener filters. Mean filters, as in [16] and [84], use a moving window to detect the corrupted pixels and replace them with the average value of their neighboring pixels. Gaussian filters, as in [84], aim to preserve the edges. They are similar to mean filters but use the Gaussian distribution function to achieve discrete approximations and softer frequency responses. They are linear

and mean square error stationary filters that can be used for Gaussian denoising and are often applied to images in the frequency domain [16, 84].

### 2.2.3 *Neural Network-Based Filters*

These filters can be divided into cellular-based filters, auto-encoder-based filters, and convolutional neural network-based filters.

Cellular neural networks are parallel computing algorithms similar to neural networks (NNs). However, unlike NNs, cellular neural networks allow communication only between neighboring units. The method in [85] proposed a combination of a nonlinear transform domain filter and a cellular neural network.

Auto encoders learn to perform efficient representation, or encoding, of a given data set (through dimensional reduction) using unsupervised learning. The authors in [86] propose a combination of sparse coding and deep networks pre-trained with denoising auto-encoder (DA) as an alternative to training pure DAs.

Convolutional neural networks are deep and feedforward artificial neural networks that use a variation of multilayer perceptron and preprocessing. The Trainable Nonlinear Reaction Diffusion (TRND), presented in [87], used supervised training to train a dynamic nonlinear reaction diffusion model with time-dependent parameters (linear filters and influence functions). The method proposed in [87] is similar to the feedback convolutional neural network (CNN) presented in [88], where each iteration (stage) of the proposed diffusion process uses convolutional operations of a set of linear filters and can thus be thought of as a convolutional network. Also, the deep convolutional neural network method (DnCNN) introduced in [89] can be thought of as a generalization of TRND [87] and are shown to achieve better performance than TRND-based filters.

#### 2.2.4 *Fuzzy Filters*

These fuzzy filters can be divided into weighted averaging based filters, control-based filters, similarity-based filters, and wavelet domain-based filters. The fuzzy rule-based filter proposed in [90] is a weighted average filter with non-linear weights. The authors in [90] introduce a gradient based non-linear multi-dimensional step-like function for which the mean square error is minimized.

The authors in [91] propose an iterative fuzzy control-based filter (IFCF) whose membership function is defined by 7 triangular-shape fuzzy sets based on the S-type fuzzy function. The modified IFCF (MIFCF) avoids the blurring of edges and tunes the membership function used by IFCF in each step. Extended IFCF (EIFCF) perform extra filtering in each iteration by compressing the membership function to further decrease the noise level. Smoothing fuzzy control-based filters (SFCF) use non-iterative IFCF-based filters to increase the filter's runtime by changing one of the rules and replacing the IFCF membership functions with smoother slope ones. Sharpening SFCFs (SSFCF) add two extra rules with an extra S-type or sigmoid function to smooth the noise and sharpen the edges at the same time. Fixed-point fuzzy control-based filters (FFCF) is a modification to SFCF. Adaptive fixed-point fuzzy control-based filters (AFCF) are modified versions of FFCF used to reduce hardware implementations cost. While adaptive c-average fuzzy control-based filters (ACFCF), nearly identical to AFCF, overcome better the effect of blurriness and perform with faster runtimes.

The fuzzy similarity-based filter (FSBF) presented in [92] defines its fuzzy rules based on the similarities between the central pixel and all the selected window templates depending on uniformity of the intensity and the template homogeneity. While the method

presented in [93] applies the Haar wavelet transformation to the noisy image and filters the wavelet coefficients through a Wiener filter (or through other kinds of fuzzy filters) with triangular membership functions like the asymmetrical triangular median filter (ATMF) [94] or the asymmetrical triangular moving average filter (ATMAF) [94]). The final denoised image is obtained through an inverse wavelet transform.

### 2.2.5 *Combined Fuzzy and Morphological Filters*

These filters are based on a combination of fuzzy and morphological filters. The method introduced in [95] uses fuzzy closing and opening mathematical image morphology [96] based on image erosion and dilation [97].

### 2.2.6 *Statistical Filters*

These can be divided into singularity function-based filters, Hidden Markov tree-based filters, and neighborhood-based filters. For example, the method proposed in [98] divides the input image into multiple sub-images and reconstructs it by using 2-D singularity function analysis (SFA) and inverse discrete Fourier transform (IDFT). Markov trees, on the other hand, are tree-like graphs composed of nodes, subset of variables, and links to which a learning algorithm is applied to model and predict meaningful descriptions. The method proposed in [99] uses non-parametric hidden Markov trees to denoise images. The neighborhood-based filters are statistical filters and they are based on the neighborhood of under-process pixels. An unsupervised, information-theoretic, adaptive filter (UINTA) is proposed in [100] based on the statistical relationship of the pixel being processed and its neighbors.

Norm based filters are another type of statistical filters. The concept of norm is a statistical average which can be defined as a function in which the size or length of each

vector (in a vector space) is set to be positive. It has different types such as sparsity based filters and low-rank approximation-based filters. The PCABM4D method from [101] is based on sparse regularization. It uses a 3D-overcomplete wavelet dictionary and solves sparse regularization minimization by using an iterative Chambolle-Pock method as the gradient method. The modified low rank matrix recovery (LRMR) method presented in [102] explores the hyperspectral image (HSI) low rank property by using a linear spectral mixing model [103, 104]. It first builds an HSI restoration model and solves it using “Go Decomposition” (GoDec) [105] and the augmented Lagrange multiplier (ALM) numerical optimization method. Total-variation-regularized low-rank matrix factorization (LRTV) denoising method [105] is another method in which the nuclear norm is used as the low rank property, total variation (TV) regularization is used for its spatial piecewise smoothness, while the  $l_1$ - norm is used to detect sparse noise.

### *2.2.7 Transform Domain Based Filters*

These filters can be divided into Curvelet filters, Contourlet filters, non-local-based filters, data-adaptive filters, and non-data adaptive filters. Curvelets based on the theory of multiscale geometry (using scale, orientation and position) are introduced in [106], yielding better performance on edge boundaries than other mature wavelet image denoising methods. The curvelet transformation consists of the following steps: 1) compute all curvelet thresholds, 2) obtain curvelet norms, 3) apply the curvelet transform, 4) apply hard thresholding to the curvelet coefficients, and 5) apply the inverse curvelet transform.

The proposed Contourlet method in [107] uses a 2-D transform to find image geometries from their discrete nature and provides sparse representation in both spatial and directional resolution; achieving better smooth curve edges performance than wavelets while keeping

contours and details. The Contourlet transformation can be summarized by the following steps: 1) apply Contourlet transform to multiscale decomposition and consequently obtain the number of scales and directions, 2) apply thresholding to the Contourlet coefficients (for each direction and for each scale), and 3) apply inverse Contourlet transform to obtain the denoised image.

Non-local filters process all pixels in the image to find how similar they are to the pixel being processed at the center of the moving window. They include: an enhanced sparse representation-based filter in transform domain (BM3D) [108], and the BM4D filter proposed in [109] which is an extension of BM3D.

Data adaptive filters use a common representation of the whole image that minimizes the global reconstruction error. The method proposed in [110] introduces the application of sparse coding (related to independent component analysis (ICA)) for image wavelet-like extraction while using soft thresholding [111] operators on sparse coding to further reduce noise. The method proposed in [112] uses an iterative fixed-point method to obtain higher convergence speed.

Non-data adaptive filters utilize the local properties of the noisy image (such as local windows and local blocks) to approximate the denoised one. They can be further divided into spatial frequency domain filters, wavelet and non-linear spatial domain filters. Low-pass filters and Gaussian filters are two types of spatial frequency domain filters. The low-pass filters presented in [15] [84] and [113] remove the high frequency signals present in the image that exceed a specified cut-off frequency. Also, the authors in [114] use the relationship between Gaussian filters, images, and noise statistics to design an optimal filter. Wavelet domain filters use orthogonal mathematical series to generate square

integrable function (Wavelets). These Wavelets are then used to transform a noisy image's domain into one where various noise removal steps can be applied.

Other filters use a combination of wavelet transforms and non-linear spatial filters to achieve denoising. Among these types of filters are: 1) 4<sup>th</sup> Order Partial differential equation [115], 2) Anisotropic like wavelet-based multiscale anisotropic diffusion method (WMSAD) presented in [116], 3) Gaussian and Bilateral [117], 4) Non-local mean [118] [119], 5) Total variation [120], 6) Wiener filters [121], 7) PCA-based denoising [122], and 8) Shrinkage rules-based filter. A classification of different shrinkage rules is performed in [123]. This type of classification includes soft and hard thresholding [111], hyperbola function thresholding [124], firm thresholding [125], non-negative Garrote thresholding [126], smoothly clipped absolute deviation (SCAD) thresholding [127], exponential thresholding [128], and non-linear thresholding-based filters. Also, these methods specify thresholds for their shrinkage rules, which include: non-adaptive thresholding estimation-based filters [129], adaptive thresholding filters-SURE [130-132], adaptive thresholding filters-Bayes [133][134], adaptive thresholding filters-cross validation [135-137], adaptive thresholding filters-spatially adaptive [138] [139], adaptive thresholding filters-bivariate [140-143].

For comparative purposes, Table 1 illustrates the peak signal to noise ratio comparison of filters based on adaptive thresholding estimation for the Lena image. 10) Wavelet coefficient-based filters. These filters can be categorized as deterministic [144], statistically-based [145-147] or of non-orthogonal type [148]. Table 2 summarizes the key points and limitations one ought to consider in the implementation of the numerous Impulse

and Gaussian filtering. Also, the overall filter classification block diagram is as shown in figure 1.

Table 1 – Peak signal to noise ratio comparison of impulse denoising filters based on adaptive thresholding estimation for the Lena image example

Method	Noise level	Correlation	PSNR	SSIM
Spatially adaptive	0.1	0.9281	26.6219	0.9245
LLSURE	0.1	0.9243	26.1790	0.9224
SURE-LET	0.1	0.9225	26.0652	0.9207
Bivariate	0.1	0.9201	25.9104	0.9189
Bayes	0.1	0.9187	25.7731	0.9178

Table 2 – Summary of the key points and limitations one ought to consider in the implementation of the numerous filters for impulse and Gaussian noise filtering

	<b>Key Points/Limitations</b>
<b>Spatial filters</b>	Averaging blurs edges and image.
<b>Total variation filters</b>	Inappropriate estimation of the number of iteration causes detail loss and over-smoothing.
<b>Non-local means filters</b>	Weight estimation complexity leads to increased computational requirements.
<b>Bilateral filters</b>	Small structures and details are removed by narrow spatial windows.
<b>Anisotropic diffusion filters</b>	Block effects result from removing features.
<b>Partial differential equation filters</b>	Increasing the order of filter produces artifacts.
<b>Morphological-based filters</b>	It uses small images as structuring elements and acts as a moving probe that samples each pixel in the image. Artifacts appear in the shape of structuring element as the window moves in a fixed direction across the image.
<b>Fuzzy-based filters</b>	Time and memory complexity are the main implementation limitations for these types of filters.
<b>Neural network-based filters</b>	Bad performance can result from inappropriate loss function and inappropriate or small training datasets.
<b>Singularity function-based filters</b>	Inappropriate frequency response, singular point determination, and thresholding lead to bad performance.
<b>Hidden Markov tree-based filters</b>	Inappropriate convergence and large number of unstructured parameters cause bad performance.
<b>Low rank approximation based filters</b>	Complexity and large dimension of the matrix in order to solve the problem cause to computational burden.
<b>Curvelet filters</b>	Poor performance in smooth area and induced Curvelet artifact production.
<b>Contourlet filters</b>	High computational complexity.
<b>Non-local based filters</b>	Lack of large amounts of matching blocks can result in artifacts. Edges blur after collaboration and aggregation steps specially in highly corrupted images.
<b>Data adaptive based filters</b>	Dimensionality reduction causes feature and information loss.
<b>Wavelet domain-based filters</b>	Inappropriate scaling and thresholding introduces image artifacts. Also, by avoiding detail blurring leads to information loss.
<b>Frequency domain-based filters</b>	Enhances entire structure (image and noise) without discrimination.

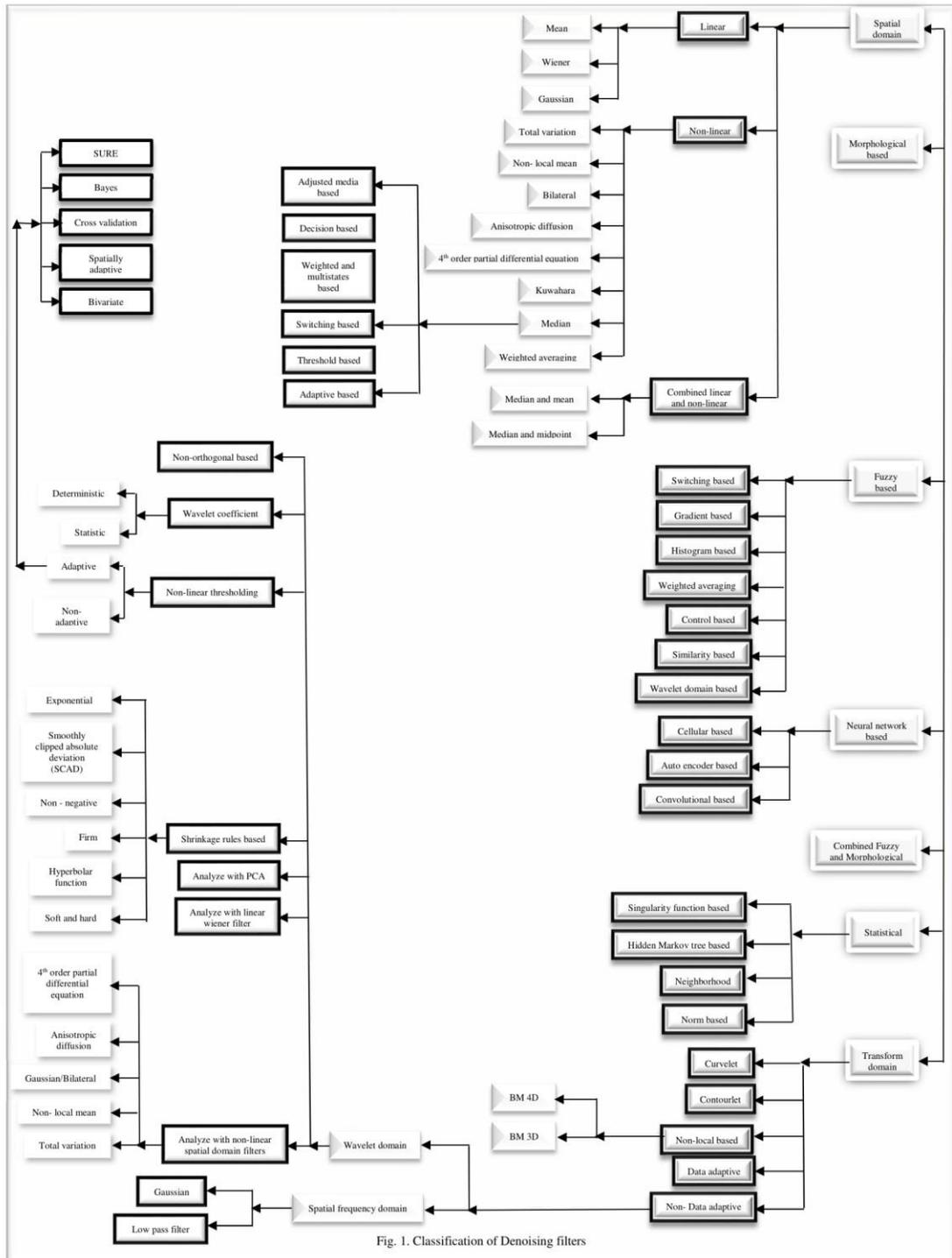


Fig. 1. Classification of Denoising filters

Figure 1 – Overall filter classification block diagram for the Impulse and Gaussian filtering

## 2.3 *Mixed Impulse and Gaussian Noise Filtering*

### 2.3.1 *Spatial Non-Linear Filters*

As indicated earlier, spatial filters are defined in the spatial domain of the image in which the intensity of each pixel is changed based on its intensity and the intensities of the neighborhood pixels. A non-linear filter is a type of filter in which its output is a non-linear function of its input. These filters can be divided into non-local mean based filters, adaptive median based filters, and total variation filters.

The non-local mean filters are based on the weighted mean value of the all pixels of the image as well as how similar they are to the pixel being processed. Patch based weighted mean filter (PVMF) [149] is a one type of such filters.

Adaptive median filters have good performance in eliminating impulse noise while averaging filters tend to have good performance in eliminating Gaussian noise. Their combination can hence be effective in eliminating these noises when mixed. An adaptive weighted mask [150] is used to remove such mixed noises based on the median filter.

Total variation methods are designed to remove spurious effects by gauging the total variation. Therefore, by decreasing total variation, we are also decreasing such spurious effects. The method proposed in [151] is based on the total variation and has two steps: noise detection and total variation minimization. Another total variation-based method is proposed in [152] which first detects the corrupted impulse noise then, applies the total variation in order to remove Gaussian and impulse noise, respectively.

### 2.3.2 *Spatial Combined Linear and Non-Linear Filters*

The non-linear median filter has good performance in removing impulse noise. If median filter is combined with linear filters, their combination will be effective in

removing the mixed impulse and Gaussian noise. The filter in [153], uses a combination of median filter [16][84], wiener filter [16][84] and bilateral filter in order to remove mixed noises.

### 2.3.3 *Fuzzy Filters*

These filters can be divided into weighted averaging filters, entropy-based filters, switching based filters, and Cardinality based filters.

The weighted averaging filters are based on weights. They apply weights to the under-processed pixels by using a sort averaging in which the summation of them are used as denominator. The weights can be obtained in different ways, one of them is based on the fuzzy logic. There are some methods which are based on the fuzzy weights [154-156].

The concept of entropy refers to a statistical measure of randomness and can analyze the texture of the image. Some filters use this concept in order to perform image restoration. The method proposed in [157] is based on the fuzzy entropy concept.

Some filters in noise detection phase are based on the switching process in order to select the optimal output. The method in [158] uses fuzzy switching filter and bilateral filter to remove the impulse and Gaussian noise, respectively.

Cardinality is defined as number of elements in a set. The method in [159] which is an improvement on simple fuzzy rule (SFR) [160] and vector median filter (VMF) method [161] uses the Cardinality concept.

### 2.3.4 *Statistical Filters*

This type of filters exploits key statistical parameters through norm-based filters, non-local similarity-based filters, and maximum likelihood estimation-based filters. The following are 3 different types that are predominant in the literature:

1) Sparsity based filters: Sparse approximation can be defined as a sparse vector in which a system of equations is solved. By adding key information as we prevent overfitting, sparse regularization is obtained. Both sparsity and sparse regularization select the best input variables (reduced input variables) in seeking the desired output variables. They include methods in [162] and [163], and weighted encoding with sparse non-local regularization (WESNR) [164].

2) In-painting based filters: Image in-painting is said to occur when there are damaged image pixels and missing image pixels. The image is reconstructed from background information. The proposed method in [165] is based on an in-painting filter design.

3) low-rank approximation-based filters: Low-rank approximation is a minimization problem based on the Frobenius norm in which the cost function calculates the fit between a given data and an approximating optimization variable, subject to a constraint that the approximating optimization variable has a reduced rank. The method in [166] is based on low rank approximation and uses weighted low rank model (WLRM) as weighted low-rank approximation (WLRA) [167] [168] or representation (WLRR) [165][170]. Another low rank approximation-based algorithm is defined in [171], based on Laplacian scale mixture (LSM) modeling and non-local low rank regularization.

In non-local similarity-based filter, several similar patches are used to reconstruct the under-processed patch. A non-local similarity-based filter is introduced in [172] which has several steps. Moreover, maximum likelihood estimation (MLE) is based on statistical model and is a special case of maximum a posteriori estimation (MAP). For parameter estimation, MLE finds the values that maximize the likelihood of them. The PARIGI method [173] is based on the MLE method.

### 2.3.5 *Supervised Learning Algorithm Based Filters*

Supervised learning is a machine learning task in which the output is known for the network in the process of labeling the training set. It iteratively makes predictions on the training data. The method in [174] is based on switching scheme with two noise detectors and two estimators for noise removal. Most of the noise is captured by the first detector and the rest remains hidden in the image details or close to the edges which are to be detected by the second one. Each detector has its own estimator which are based on median and median absolute deviation (MAD). Also, in order to build the detectors, genetic programming (GP) is sometimes used [175] [176].

### 2.3.6 *Numerical Method Based Filters*

Finite element method is a numerical method used to solve the problem in which a vibrational formulation, post processing and one or more solution algorithm are used. The method in [177] is based on one such finite element method [178-180].

### 2.3.7 *Morphological Operation Based Filters*

This is a non-linear operation using the morphology of features in an image not necessarily related to a numerical value. Dilation and erosion are two such morphological operators used in images. Dilation adds pixels to the boundaries in an image and erosion removes the pixels on the boundaries. The method in [181] is based on such morphological operations.

### 2.3.8 *Transform Domain Based Filters*

These filters are processed in the domain which is not their original domain. There are some transformation domain filters such as frequency, wavelet, Curvelet, and Framelet. The method in [182] proposed a frame-based [183] iterative algorithm for denoising.

The overall filter classification block diagram is as shown in figure 2. Also, Table 3 summarizes the key point and limitations of the numerous mixed Impulse and Gaussian filtering.

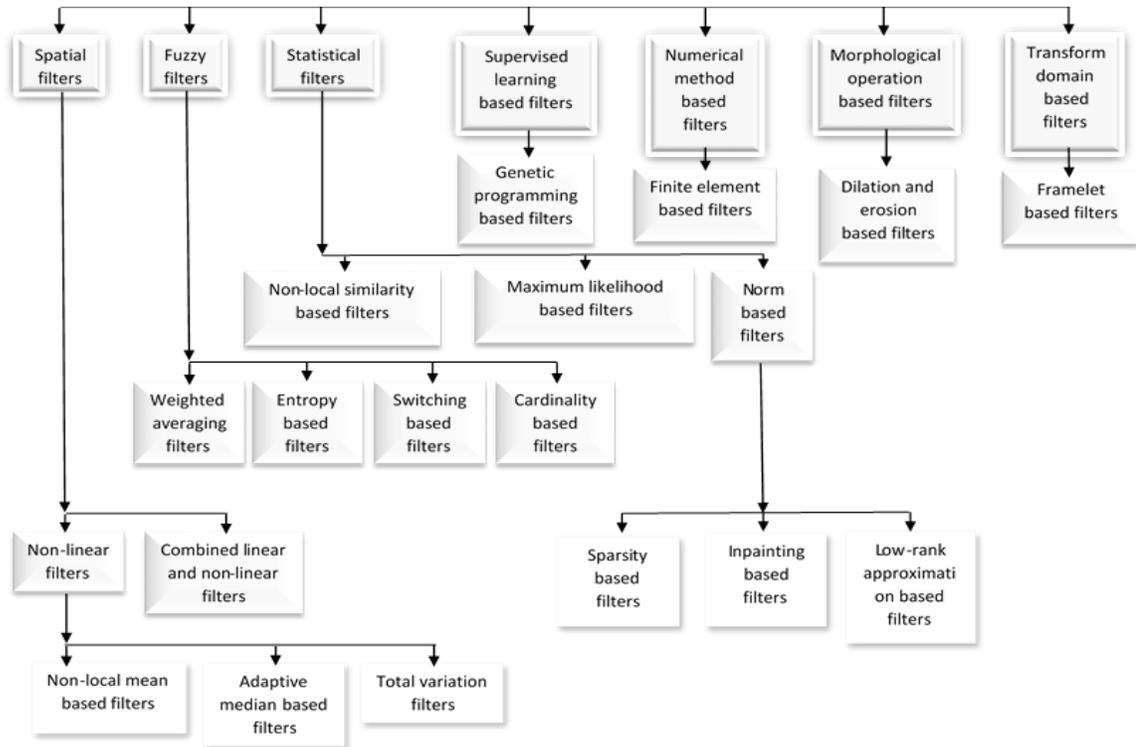


Figure 2 – Overall filter classification block diagram for the mixed impulse and Gaussian filtering

Table 3 – Essential points and limitations to consider in the implementation of the numerous filters for the mixed Impulse and Gaussian filtering

<i>Type of filter</i>	<i>Limitations</i>
<b>Spatial filters</b>	Averaging caused to blur the edges and consequently image
<b>Total variation filters</b>	Inappropriate estimation of the number of iteration caused to loss the details and over-smoothing
<b>Non-local means filters</b>	Complexity of weighting cause to computational burden
<b>Fuzzy based filters</b>	In the case of good mathematical descriptions and solutions, time and memory are two limitations for complete mathematical implementation
<b>Non-local similarity based filters</b>	Detection of the best patches. Complexity cause to computational burden. Difficulty obeying quality constraints.
<b>Maximum likelihood based filters</b>	Difficulty obeying quality constraints. Sensitive to choose initial values. It needs large samples to get optimal result. The numerical estimation is non-trivial. The mathematic is often non-trivial, particularly confidence intervals for the parameters is needed.
<b>Sparsity based filters</b>	Principled way to choose a solution for problem cause to computational burden and time consumption. Solving a noise-aware variant cause to sparse approximation and representation problem.
<b>Inpainting based filters</b>	Reproduction of large texture regions Unable to recover partially degraded image
<b>Low rank approximation based filters</b>	Complexity and large dimension of the matrix in order to solve the problem cause to computational burden. It could be a serious practical problem in the image.
<b>Genetic programming based filters</b>	Very remarkable computing resources required.
<b>Finite element based filters</b>	There is no general close-form solution (it can change in various parameters). The solutions are based on an approximation. It has inherent errors which can cause to corrupt the image.
<b>Morphological based filters</b>	It uses small images as structuring elements and acts as a moving probe that sample each pixel of image. it moves a fixed direction across the image, therefore, an artifact appears in the shape of structuring element.
<b>Framelet based filters</b>	Their orientation selectivity is limited to only two directions. Complexity cause to computational burden.

### 3. CHAPTER III

## THEORY AND METHODOLOGY

### *3.1 Spatial Filter Design for Impulse Denoising*

Noise is a ubiquitous and unwanted phenomenon that is inherent to many image acquisition and transmission sources. One such type of noise that degrades image quality is impulse (or salt and pepper) noise which appears as white and black pixels in the degraded image. In order to remove this type of noise, smoothing filters are often applied to the image to decrease the variance of the noise, while endeavoring to preserve as much as possible important details in the image. A standard course of action is to perform smoothing of the image first before some form of gradient is applied. With the knowledge that derivatives tend to amplify the presence of noise, a tradeoff needs to be negotiated between the objective of decreasing noise variance and the need for keeping all relevant image details.

There are several image impulse denoising and edge-preserving methods that have been proposed in the past as discussed in chapter II. In this study, a comparative assessment is provided contrasting the results obtained using the proposed approach with the results of the most recent and proven effective filters, which focus on the removal of impulse noise in images [184] [185]. These include: 1) improvement boundary discriminative noise detection (IBDND) [36] which is an improvement on BDND [35]. 2) Decision based unsymmetrical trimmed modified winsorized mean filter (DBUTMWMF) [49], which is based on two mean filtering steps. The authors report that the results they obtained using the method in [49] were better than those achieved using AMF [20], progressive switching median filter (PSMF) [24], decision based median filter (DBMF) [44, 33], improved

decision based filter (IDBA) [43], MDBUTMF [45], trimmed-global mean [186], adaptive cardinal B spline algorithm (ACBSA) [187] and the cascaded decision based median filter and unsymmetrical trimmed decision midpoint filter (CUDBMPF) [44]. 3) Two cascading algorithms were proposed in [50] with the first combining a decision based median filter and modified decision based partial trimmed global mean filter (DBPTMGF) [47] and the second combining DBMF and MDBUTMF [45]. The authors who proposed these cascading algorithms report a better performance than when using the AMF [20], the decision base asymmetric median filter (DBUTMF) [33], the decision based partial trimmed global mean filter (DBPTGMF) [47] as well as when using other cascading algorithms such as DMF+UTMF and DMF+UTMP [44]. 4) Unbiased-based weighted mean filter (UWMF) as described in [39], a weighted mean filter, which is based on the spatial bias, Minkowski distance and spatial distances in the  $x$  and  $y$  directions. The results using the UWMF show a better performance than when using AMF [20], the MDBUTMF [45], the improved boundary discriminative noise detection filter (IBDND) [36], cloud model filter (CMF) [40] as well as the interpolation-based impulse noise filter (IBINRF) [41]. Nonetheless, these denoising methods still encounter some challenges when faced with high impulse noise that include loss of image details, blurring of the image and unsmoothed edges, which make the edge detection process more difficult to attain reliably.

The Motivation in this endeavor is driven by the following two goals: 1) resolve the challenges still faced when using denoising methods in terms of keeping as much image details as possible, avoid blurring of the image, and preserve the sharper edges associated with boundaries; and 2) contend with these challenges even in the presence of high-intensity impulse noise. Combining these two steps highlights the novelty of this proposed

method. Consequently, this study introduces a new denoising filter capable of preserving more edge details with high structural similarity to the original (noise-free) image even in the presence of high impulse noise. The results obtained, as will be discussed later, are contrasted to all other well-known denoising filters.

The proposed method, as illustrated in Figure 3, consist of five essential steps: 1) use of adaptive median and fixed weighted mean smoothing filters in combination in an effort to yield the highest structural metrics in comparison to current state-of-the-art filters; 2) perform edge detection using standard kernels; 3) extract edge routes based on the non-maximum suppression method; 4) fill the discontinuities and remove noisy pixels according to the maximum-sequence method, especially when using high predefined thresholds and under high-intensity noise levels; and 5) apply predefined thresholds and make use of specific morphologic operations to evaluate the results under different impulse noise intensities.

### 3.1.1 Proposed Method for Impulse Denoising

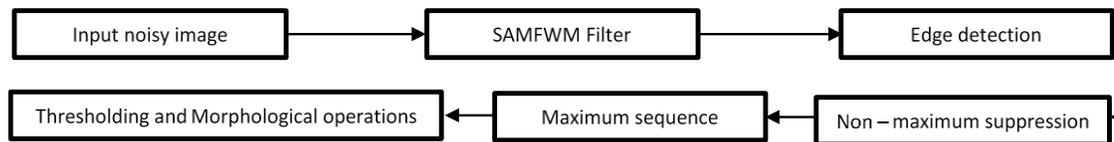


Figure 3 – Essential processing steps for impulse denoising and edge detection

With the proposed method, boundary edges of filtered images are assumed to have high correlation with the original images, and as such, edges should track the true routes even under high-intensity impulse noise. Most of the current leading filters ensure a good performance on impulse noise reduction, but they still do not perform well on boundaries, especially in the presence of impulse noise with high-intensity levels.

In the proposed method, there are two choices that can be made when using the switching adaptive median filter:

- **In switch 1**, within an initial sliding window, all pixels with 0 and 1 values are removed, and the median value of the remaining pixels with probability of  $1 - P_p - P_s$  as in (1) will replace the pixel being processed. If all of them are 0s, 1s or a combination of them, or if the variance of the pixels is much higher than the median value, then the size of the window is increased by 1 and the process is repeated until the window size reaches the predefined maximum window size. We assume the difference between pixel values is high when the variance is much higher than the median value (in this study  $\sigma > 2Median$ ), which could be an indication that an edge is present in that area. It thus checks the variance in bigger window sizes to validate whether such an edge does indeed exist or not. If there is an edge, the assumption is that the median value can detect it, otherwise the median value will be correlated to the texture found within the window.

- **In switch 2**, the 0s and 1s are not removed within the initial sliding window, and the median value of all pixels will replace the pixel being processed. In this switch, if all of the pixels are 0s, 1s or a combination of 0s and 1s, the window is simply increased by 1, and in the same way as in switch 1, the process continues until the window size reaches the predefined maximum window size. This case is designed specifically for images which contain significant black and white regions with clear edges like checkerboards or mesh like images; but, for other types of images, switch 1 is expected to yield better structural metrics than switch 2.

By increasing the size of the adaptive median filter, the structural metrics will be somewhat decreased, resulting in an image that is slightly blurred. However, the edges still

appear sharp. Therefore, there is a tradeoff to be made between the edges extracted and the quantitative values of the structural metrics. However, the pixel being processed will remain unchanged if all of the pixels in the selected window are 0s or 1s or a combination of them. There are special cases when a given texture would itself consist of 1s and 0s, for example a checkerboard. The problem for this latter case becomes more challenging in delineating such textures especially in the presence of impulse noise. When such combinations of 0s and 1s are found in several instances in the sliding window, the mean filter can be applied. This combination can smooth the image while maintaining high structural metrics and sharp edge boundaries even in the presence of high-intensity impulse noise. In order to avoid any lingering noise effects in the black and white regions (especially in relatively bigger regions) in which the mean filter could potentially change the intensities, an additional shrinkage window can be defined before applying the mean filter. This step, which removes 0s in white regions and 1s in black regions, can be very useful for textures that consist of a combination of black and white sections. The maximum size of the window would hence depend on the texture and noise level in the image being denoised.

Also, the structural metrics of fixed mean filter can be improved by assigning appropriate adaptive weights for the pixels in the selected window in accordance to the probabilities of noise occurrence. This window could contain all 0s ( $P_p$ ), all 1s ( $P_s$ ), or a combination of them together with the other pixels with probability of  $1 - P_p - P_s$ . The mean filter tends to introduce more blur in the image, which in turn could lead to loss of details. To prevent these side effects, the size of the mean filter should be kept small and fixed, as is done in the proposed method.

In this study, noise reduction is performed with the intent to preserve edge points in an optimal fashion.

### 3.1.1.1 Structure of the Method

The procedural steps of this method embed the two main components of switching adaptive median (SAM) filtering and fixed weighted mean (FWM) filtering with additional shrinkage window to make up the proposed denoising method we refer to as SAMFWMF. The adaptive median component is so called in that the window size can be dynamically changed according to table 4. Increasing the window size from its original 3x3 size is warranted only if the SAM step did not yield optimal results. Another adaptive additional window is set to overcome any the remaining noise in white and black regions. By doing so, we avoided blurring the final SAMFWMF image by increasing the size of the window in the SAM component (steps 1-5) rather than in the FWM component (steps 6 -11) of the following process:

Table 4 – Maximum window size of adaptive median filter in different noise levels on different images a) natural images b) images which contain significant black and white regions with clear edges like checkerboards

Window size	3×3	5×5	7×7	9×9	> 9×9
Noise level	< 40%	≥ 40%	> 70%	> 80%	> 90%
		≤ 70%	≤ 80%	≤ 90%	

(a)

Window size	3×3	5×5	≥7×7
Noise level	< 40%	≥ 40%	≥70%
		< 70%	

(b)

1. In the case of switch 1, if all of the pixels in the 3×3 window are 0s and 1s, or a combination of them, or if the variance ( $\sigma$ ) of the window is much higher than the median value (in this case  $\sigma > 2Median$ ) then, the size of the window is increased to a 4×4, then 5×5 and so on until it reaches the predefined maximum size. Otherwise,

- it leaves the window size unchanged. In the case of switch 2, it only checks if all of the pixels in the  $3 \times 3$  window are either 0s or 1s, or a combination of them, and if so, the size of the window is increased by 1; else, it leaves the window size unchanged. Then set the normalized pixels of the 2-D selected window as a  $1 \times N$  1-D vector ( $N=3$  to maximum value), and check if the pixel  $I(i, j)$  being processed is a corrupted pixel; that is to check if  $I(i, j) = 0$  or 1 (normalized value) in  $W_{1 \times N} = (\dots, I(i, j), \dots)$ .
2. Detect all pixels with 0 and 1 values, and in the case of switch 1 eliminate them, so the size of the window  $W_{1 \times N}$  is now decreased to a new size  $W_{1 \times N-k} = (\dots, I(i, j), \dots)$ , where  $k$  represents the number of corrupted pixels that were removed; and in the case of switch 2 where such pixels are not eliminated, the size of the window remains  $W_{1 \times N}$ .
  3. Switch 1 replaces the  $I(i, j)$  pixel value with the median value of the remaining  $N - k$  pixels in the vector window if at least one pixel remains in the reduced window, otherwise leaves  $I(i, j)$  unchanged. Switch 2 replaces the  $I(i, j)$  pixel value with the median value of the  $N$  pixels in the vector window.
  4. Leave uncorrupted pixels unchanged.
  5. Slide the window by one pixel and repeat the process consisting of steps 1-4 throughout the entire image, establishing at this stage the SAM filtered image.
  6. Starting from the predefined maximum size for the shrinkage window, we start by checking the boundary pixels of the selected window (filtering window). One of the following conditions has to be met: If they are all 1, the interior pixels are changed to 1. If all the pixels on the boundary are 0, then the interior pixels are changed to 0.

Otherwise (there is a combination of 0 and 1), the window is then shrunk by one and the process is repeated until the minimum size (3×3) is reached.

7. For the fixed mean filtered image, use a 2×2 window in a convolution manner, and check if the pixel being processed ( $I(i, j)$ ) within the vector window  $W_{1 \times 4} = (I(i, j), I(i, j + 1), I(i + 1, j), I(i + 1, j + 1))$  is found corrupted (i.e.,  $I(i, j) = 0$  or 1 (normalized value))
8. Using the weights selected on the basis of the two conditions described next, if salt or pepper (probability  $P_s$  or  $P_p$ ) is detected, the new processed pixel would be assigned the new value as in (5). Otherwise, it leaves the pixels unchanged.

$$M_{new}(i, j) = \frac{\sum_{(x,y) \in S_{new}(i,j)} \omega_{x,y} I_{x,y}}{N-1} \quad (5)$$

In this equation,  $N$  is 4,  $S_{new}(i,j) = \{I(i, j + 1), I(i + 1, j), I(i + 1, j + 1)\}$ , with indices  $(i, j)$  indicating the positions of the corrupted pixels, and  $(x, y)$  are the coordinates of the pixels around it. In this proposed method, when the detected corrupted pixel occurs as salt or pepper (with probabilities  $P_s$  or  $P_p$ ), the weights  $\omega_{x,y}$  are directly selected based on the probability of occurrence 1 or 0 for neighboring pixels, according to one of these conditions:

- Condition1: We assume the corrupted pixel with the probability of  $P_s$  or  $P_p$  occurs, and the probability of occurrence of 1 is more than that of 0 for the neighboring pixels (with the assumption that the window contains only 0 and 1). We will set  $\omega_{x,y} = 1$  for all pixels. In this case, if all the neighboring pixels are equal to 1, the value of the corrupted pixel changes to 1, otherwise, changes to a

value between 0 and 1 based on the assumption that the probability of changing neighboring pixels to a value between 0 and 1 is high.

- Condition 2: We assume the corrupted pixel with the probability of  $P_s$  or  $P_p$  occur, and the probability of occurrence of 0 is more than that of 1 in neighboring pixels (with the assumption that the window contains only 0 and 1). Then we will set  $\omega_{x,y} = 2$  for the east and south pixels and  $\omega_{x,y} = 1$  for the southeast pixel. In this case, if all the neighboring pixels are equal to 0, the value of the corrupted pixel changes to 0, otherwise, changes to a value between 0 and 1 based on the assumption that the probability of changing neighboring pixels to a value between 0 and 1 is high.
- Condition 3: We assume the corrupted pixel with the probability of  $P_s$  or  $P_p$  occurs, and there is a probability for neighboring pixels with value between 0 and 1 to exist. Then we will set  $\omega_{x,y} = 2$  for the east and south pixels and  $\omega_{x,y} = 1$  for the southeast pixel. If all of the neighboring pixels are equal or if the summation of the weighted neighboring pixels are greater or equal to the denominator (greater or equal to  $N-1$ ) as  $sum = \sum_{(x,y) \in S_{i,j}} \omega_{x,y} I_{x,y}$  if  $sum \geq N - 1$ , then we will set  $\omega_{x,y} = 1$  for all pixels. In this case, the value of the corrupted pixel changes to a value between 0 and 1 with the assumption that the probability of changing neighboring 0 or 1 pixels (if they exist) to a value between 0 and 1 is high and with the rest of the pixels still assuming values between 0 and 1. Also, in the case of equal neighboring pixel values, the corrupted pixel would be equal to the value of these neighboring pixels.

9. Replace the corrupted pixel with the mean value. Leave uncorrupted pixels unchanged.
10. Repeat steps 7-9 for the entire filtered image, resulting in the SAMFWM filtered image.
11. Check the level of impulse noise present and determine if the filter yields satisfactory results. If results are not satisfactory, increase the switching adaptive median  $3 \times 3$  window into  $5 \times 5$  and so on. Hence, as the intensity of the noise present is higher, set a new of dimension of the window as  $W_{new} = W_{old} + 2$ , and the process consisting of steps 1 through 11 is repeated until optimal results are obtained.

In this last step, optimization of the filtering results is reached when the evaluations measures, as described next, yield the highest values. Figure 4 shows a flowchart depicting the process..

#### 3.1.1.2 *Evaluation Measures*

To measure the degree of edge preserving and image structural metrics, standard measures are computed in order to compare the performance of different filters including the proposed method to gauge the quality of image after the smoothing process is performed.

The following measures are used in this study:

- Correlation coefficient ( $\beta$ ) [188] measures the amount of preserved details and edges after the denoising process.
- Structural similarity index (SSIM) [189] measures the difference between the original noise-free image and the denoised image after the denoising process.

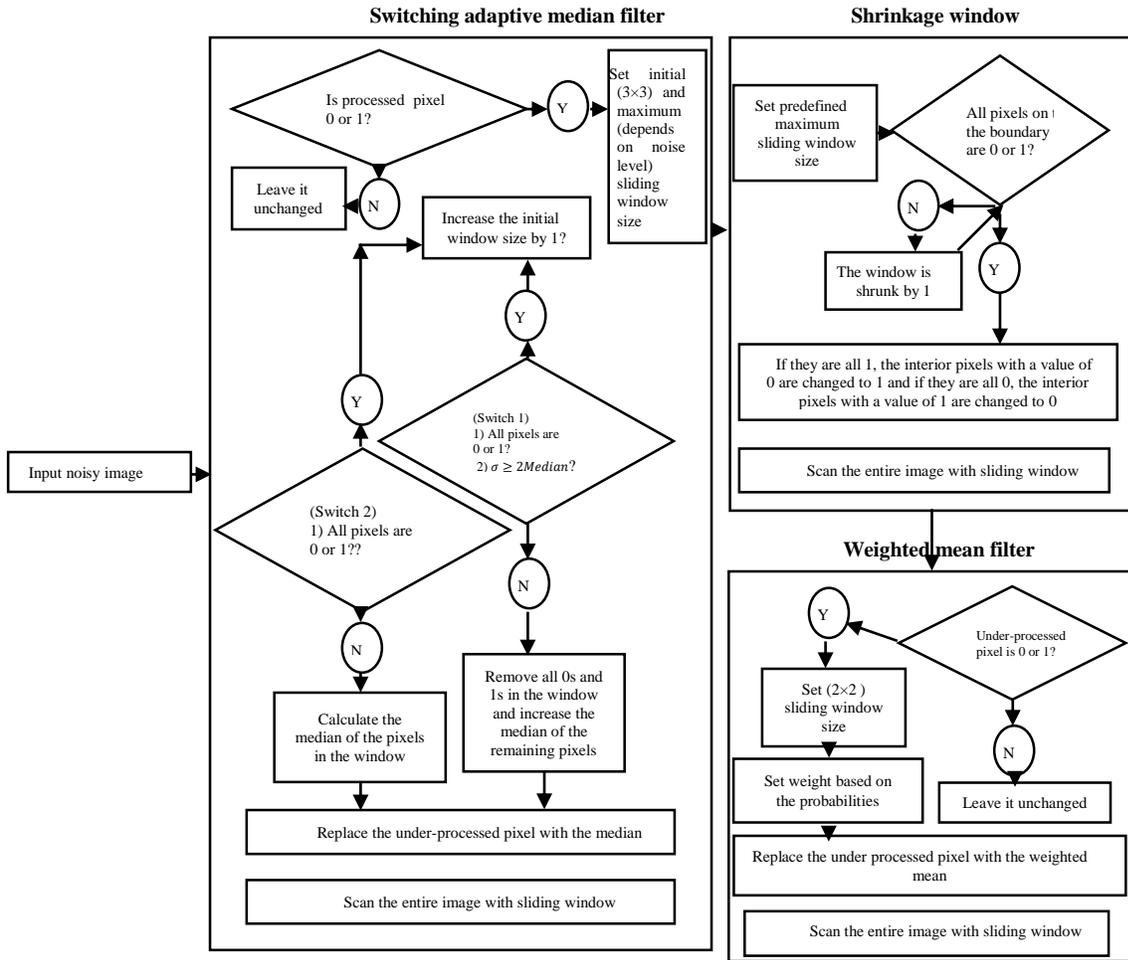


Figure 4 – Flowchart for impulse denoising

- Peak signal to noise ratio (PSNR) which measures the level of noise in the denoised image after the denoising process.
- Figure of merit (FOM) [190] which measures the edge detection performance.

Equations (6) through (9) provide the different formulations used for Correlation ( $\beta$ ), SSIM, PSNR and FOM, respectively. In all these formulations,  $x(i, j)$  represent the pixels in the original noise-free image,  $n(i, j)$  represent the pixels in the noisy image, and  $y(i, j)$  represent the pixels in the denoised image after the filtering process has been applied.

The correlation coefficient is defined as follows:

$$\beta = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i,j) - \overline{x(i,j)}] \times [y(i,j) - \overline{y(i,j)}]}{\sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i,j) - \overline{x(i,j)}]^2 \times [y(i,j) - \overline{y(i,j)}]^2}} \quad (6)$$

Where  $\overline{x(i,j)}$  and  $\overline{y(i,j)}$  represent the mean values of the  $x$  and  $y$  images, respectively.

The structural similarity index (SSIM) is measured as follows:

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{(\bar{x}^2 + \bar{y}^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (7)$$

Where  $\sigma_x$  and  $\sigma_y$  define the standard deviations in the  $x$  and  $y$  images, respectively, and  $\sigma_{xy}$  defines the standard deviation between the two images, while  $C_1$  and  $C_2$  are two variables which depend on the dynamic range of pixels often set in the (7) as  $C_1=0.01L$  and  $C_2=0.03L$ , where  $L$  is the dynamic range (here it is assumed 1 since pixels are normalized). The values of 0.01 and 0.03 are default values recommended by the inventors of the SSIM measure to stabilize the denominator and avoid a zero value in the denominator.

The peak signal to noise ratio (PSNR) measure is given as:

$$PSNR = 10 \log \frac{(\max(x))^2}{MSE} \quad (8)$$

Where  $MSE$  is the mean square error, and  $\max(x)$  defines the maximum intensity of the pixels in image  $x$ .

Noteworthy comparisons and evaluations of different edge detection methods are provided in [190] [191]. It should be noted that the main point in this proposed method is in edge detection evaluation after the denoising process has been accomplished; therefore, the figure of merit (FOM) [190] of the algorithm is measured to assess the merits of the

denoising process. FOM, which consists of several steps, is a metric that measures the distance between the detected edges with those of a reference image. The binary reference image is generated based on 3 steps: 1) white noise generator, 2) low pass filter (Gaussian PSF with width  $\sigma_g$ ), and 3) a zero-crossing detector. The test image is generated based on 8 steps: 1) white noise generator, 2) low-pass filter (Gaussian PSF with width  $\sigma_g$ ), 3) thresholding, 4) region labeling, 5) random grey level assignment (with standard deviation  $\sigma_h$ ), 6) low-pass filter (Gaussian PSF with width  $\sigma_p$ ), 7) Gaussian noise with variance  $\sigma_n^2$ , and 8) impulse noise (with noise level  $Im$ ); all this before applying the denoising process which is an extra step in order to evaluate the edge detection performance in the presence of impulse noise. The FOM measure is thus given as:

$$FOM(\sigma_c, \sigma_g, \sigma_h, \sigma_p, \sigma_n, Im) = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M g^2(n, m) \quad (9)$$

Where  $(N, M)$  is the size of the image,  $g(n, m)$  is the convolution between  $f(n, m)$  and the Gaussian PSF with width  $\sigma_c$ , and where  $f(n, m)$  is the difference between the binary image (with the detected edges) and the binary reference image.

### 3.1.1.3 Experimental Evaluation in the Presence of Impulse Noise

The cascading algorithm [50], IBDNDF [36], DBUTMWMF [49], UWWMF [39], considered as most effective when dealing with impulse noise, are compared to the SAMFWMF under different impulse noise intensity levels, and the aforementioned metrics are used for evaluation. It should be noted that in order to optimize the denoising of the image, when the impulse noise is increased, the size of the filter may be changed.

Using images of Lena (512×512), Camera man (256×256), Coins (300×246) and checkerboard (256×256) as the standard examples used in the literature for comparative

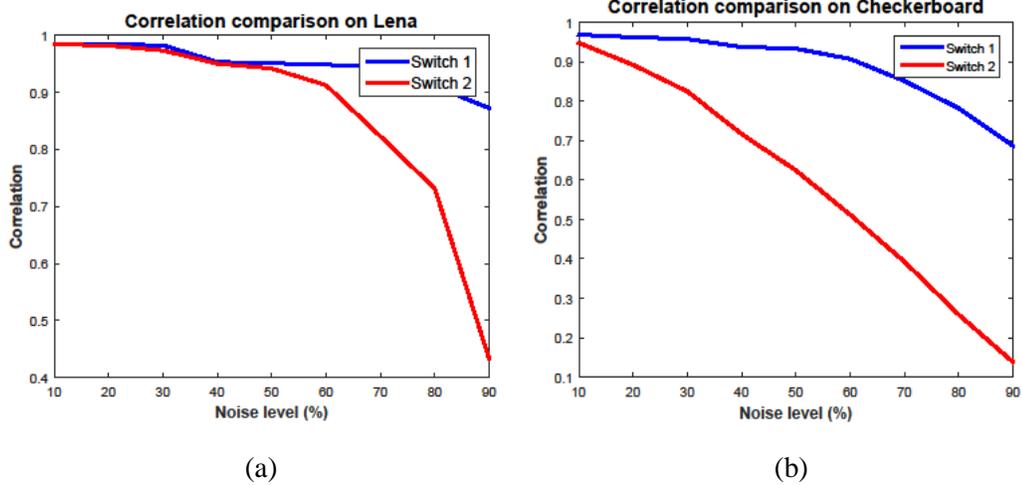


Figure 5 – Correlation comparison between both switching methods on a) Lena  
b) Checkerboard

purposes, Figure 5 shows respectively the correlation comparison between both switching methods on different images. The initial windows for both of them are equal to the maximum window size in the related noise intensity.

Figure 6 shows the edge boundaries of different images after applying the SAMFWMF with switch 1 in the presence of high noise intensities with different initial adaptive median windows and in contrast to the other well-known filters. The adaptive properties of the proposed median filter, in which its initialization is dynamic, adapts well to the smoothness of the edges. In the lower initial window size, sharpness of the edges is not easily attained; therefore, as the initial window size is increased, the edges appear smoother and sharper, Figure 6 (rows 7 and 8) exemplifies these observations. As the results for the SAMFWMF reveal, the intensity variations on the edges are sharper, and the structural similarity measures are higher than with other filters even when the impulse noise intensity is high. Furthermore, since the mean filter introduces blur in the image with some details lost as a

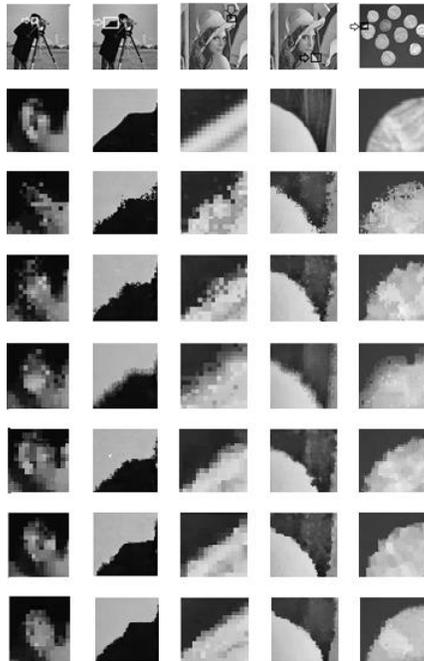


Figure 6 – Edge boundaries of different images after applying the spatial filter with switch 1. Results of filtering Camera man with 70% impulse noise intensity, Camera man with 90% impulse noise intensity, Lena with 80% impulse noise intensity, Lena with 90% impulse noise intensity and Coins with 90% impulse noise intensity in column 1 through 5, respectively. After applying the filter: Rows 1 through 8 are: Original image showing the specified area under scrutiny, original specified area, denoised results using, cascading algorithm [50], IBDNDF [36], DBUTMWMF [49] and UWMF [39], and SAMFWMF with initial adaptive median window size=3 (minimum size), and SAMFWMF with initial adaptive median window size=maximum size in that noise level

consequence, the proposed method maintains the size of the mean filter fixed but with specific weights given to the neighboring pixels during the smoothing process.

Figure 6 shows the comparison between other denoising methods with and without fixed weighted filter as a post-processing step in the presence of high impulse noise intensities.

Figure 7 shows the correlation comparison between the two states (with and without weights) of fixed mean filter (FM) with switch 1 in the presence of different noise intensities. The initial adaptive median window size for both of them are equal to the

minimum window size ( $3 \times 3$ ). In the first approach, weights ( $\omega_{x,y}$ ) are set for neighboring pixels, and in the second approach the method is run without setting these neighboring

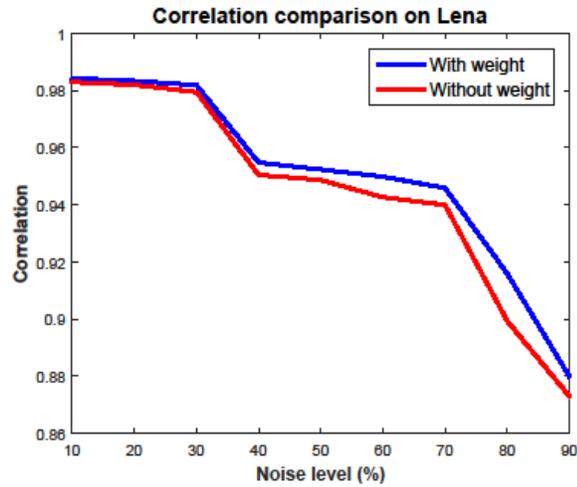


Figure 7 – Correlation comparison between two states of fixed mean filter (with and without weights) on image Lena

weights. As the figure shows, setting weights improves the structural metrics in the denoised images.

### 3.1.2 Edge Detection After Spatial Filtering

Edge detection is a challenging nontrivial problem but is a task that remains essential for object identification, image segmentation, feature extraction, pattern recognition among other essential image processing tasks. There are several methods and well-known operators that are commonly used to detect edges in images, and their success is often weighted as a function of the amount of image detail that was preserved and the application at hand. When we deal with images, pertinent details can be useful when analyzing specific imaging data, but the concern has always been in delineating what really constitute actual edge data with a high degree of similarity to the original noise-free image in contrast to

other background and noise data that burdens the edge detection process. The challenge is further amplified when the images are degraded by noise, affecting significantly their structural metrics.

Canny edge detection [192], perhaps one of the most useful and well-known method, is a multifaceted process that integrates Gaussian filtering for smoothing the image, intensity gradient, non-maximum suppression for edge thinning, thresholding and tracking of the edges to ensure edge connectivity and continuity. The holistically nested edge detection (HED) method [193], which is a robust edge detection method, uses convolutional neural networks and is based on image training and prediction through multi-scale and multi-level feature learning. Such edge detection methods and related edge operators extract quite successfully edge information and yield a good performance when dealing with clean images; however, their performance is degraded in the presence of impulse noise, especially when it is of high-intensity type. Such degradation could be overcome, but only with additional well thought out filtering steps. Neuro-fuzzy operator [194] is designed to detect edges in the presence of impulse noise, but its success is limited only for low intensity noise levels. A fast algorithm that detects edges in noisy images is proposed in [195], but preserving image details under different noise intensities was not its main focus.

Edge detection is a nontrivial process mainly due to the ambiguity associated with defining what constitutes an observable transition (differential thresholds or just-noticeable difference) between image intensities. For all practical purposes, first derivative operators are adequate in their use for edge detection and in determining local minima and maxima. Second derivative operators could be useful for localization purposes due to the zero

crossings. To determine the strength of an edge point, the gradient should be measured perpendicular to the edge direction.

There are several kernels that can be used for edge detection. First derivative operators, although weak in terms of localization, are nonetheless less sensitive to noise than their second derivative counterparts and are also less complicated in their implementation. Accordingly, for this study, any edge detection kernel could have been used, but in the implementation of the proposed method, a  $3 \times 3$  first order derivative kernel is used, and the

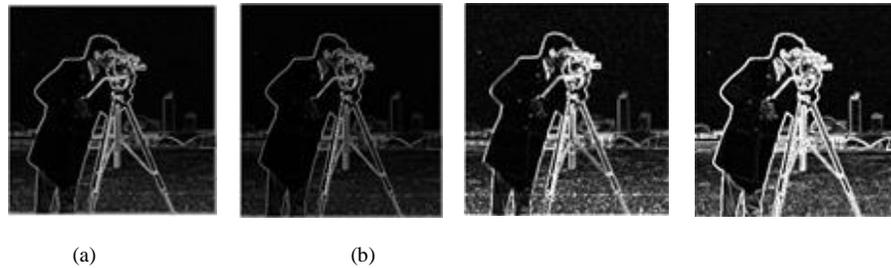


Figure 8 – Different grayscale images with different kernels on image Camera man a) First derivative with  $2 \times 2$  matrix b) First derivative with  $2 \times 2$  diagonal matrix c) Second derivative d) First derivative with  $3 \times 3$  matrix

results are satisfactory in terms of the evaluation metrics used in this study. Figure 8 shows the different grayscale images with different kernels.

### 3.1.2.1 *Continuity in Edges and Thresholding in Grayscale Images*

#### 3.1.2.1.1 *Non-Maximum Suppression*

This technique [192] is used for edge thinning in the grayscale image. Edge strength is compared with the neighboring pixels according to gradient direction, the whole process can be summarized as follows:

- Calculate the vertical and horizontal gradient.
- Calculate the angle of the gradient, and

- if the angle of gradient is 0 degree, the gradient magnitude is checked in the east and west directions, and if it is more than the magnitude of pixels in these directions, it is considered on the edge
- if the angle of gradient is 45 degrees, the gradient magnitude is checked in the northeast and southwest directions, and if it is more than magnitude of pixels in these directions, it is considered on the edge
- if the angle of gradient is 90 degrees, the gradient magnitude is checked in the north and south directions and if it is more than the magnitude of pixels in these directions, it is considered on the edge
- if the angle of gradient is 135 degrees, the gradient magnitude is checked in the northwest and southeast directions and if it is more than the magnitude of pixels in these directions, it is considered on the edge

#### 3.1.2.1.2 *Maximum-Sequence and Thresholding*

This technique is used to maintain edge continuity during the edge detection process and extract more edges at different threshold intensities while minimizing noise. The predefined threshold ( $T$ ) and edge point factor ( $\alpha$ ) are set to any value within the normalized range such that  $0 < T < 1$ ,  $0 < \alpha < 1$ . The value  $\alpha$  when used with  $T$  as  $(\alpha * T)$  is assigned such as to resolve the dilemma of selecting too high or too low of a threshold initially. Then the following steps are considered:

- The process starts by setting a threshold value for a starting (first) edge point ( $I_{startpoint}$ ) as  $(\alpha * T)$  or  $(T)$
- The next step is to check the value of  $I_{startpoint}$  in all four edge directions in a  $2 \times 2$

window ( $W_{2 \times 2} = I(i, j), I(i, j + 1), I(i + 1, j), I(i + 1, j + 1)$ ), if it is higher than (T) or ( $\alpha * T$ ).

- Find the maximum value of the neighboring pixels ( $I_{max}$ ) in direction of the edge.
- Increase the intensity of the maximum pixel ( $I_{max}$ ) which was found within the  $2 \times 2$  window to the value of (T).
- If  $I_{startpoint} \geq \alpha * T$  and  $I_{max} \geq \alpha * T$ , then  $I_{maxnew} = T$ ; or if  $I_{startpoint} \geq T$  and  $I_{max} \geq \alpha * T$  then  $I_{maxnew} = T$
- This process scans the entire image.

It is possible to set  $\alpha = 1$  and change the value of T to get the desired results, but this does not guarantee a noise-free outcome when a low value of T is chosen, especially when the probability of occurrence of the salt and pepper is high. By using the weights for the mean filter in order to affect the values of salt and pepper, the values of the noisy pixels may be changed to a value more consistent with their neighboring pixels, and if the value of T is less than that, the algorithm will assume the noisy pixels to be edge points. However, by using the maximum sequence method, this problem can be overcome. Such an algorithm can thus detect edges at any threshold level simply by changing the value of  $\alpha$ . If the pixel is considered part of the edge line, the algorithm will continue to track the line, but if it is a noisy point, the algorithm makes the pixel zero, creating a discontinuity. There are two choices that can be made: in the first choice, we can assume  $I_{startpoint} \geq T$ , while in the second choice, we assume  $I_{startpoint} \geq \alpha * T$ . For both choices, the algorithm tracks the edge line and maintains the continuity. In the second choice, maybe some points in the edge line may not be detected, because the algorithm selects only the neighboring



thresholding such as the hysteresis and Otsu methods which are quite interesting and effectual. The hysteresis method relies on two thresholds (low and high) and pixels above the high threshold are assumed edges and those below are not edges and those pixels in between these two thresholds are edges only if they are adjacent to other edges. Also, the Otsu method sets its own general threshold depending on the minimized variance of the two regions that are separated by the threshold [196]. This last approach is akin to finding that threshold maximizing interclass variance in a bimodal histogram. Figure 10 shows that how the maximum-sequence method makes continuity along the edge lines in the image. Figure 11 contrasts the results between different thresholding methods and the maximum-sequence method on the image. As the figure shows, the maximum-sequence method can detect more edges, and the edges it detects are thinner in the different threshold intensities in contrast to the other methods.

### 3.1.2.2 Morphological Operations

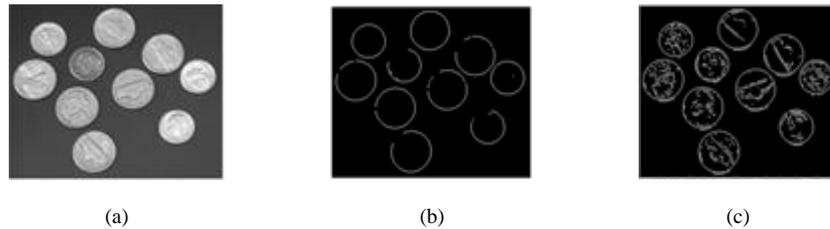


Figure 10 – Continuity along the edge lines in the image after applying maximum sequence

After applying the threshold on the image, the output would be a binary image. So, in order to improve the binary image, some morphological operations as shown in figure 12 are performed on the image. The objectives for using such operations are to remove unwanted edge points and improve the tasks of edge thinning and edge continuity. They also help in

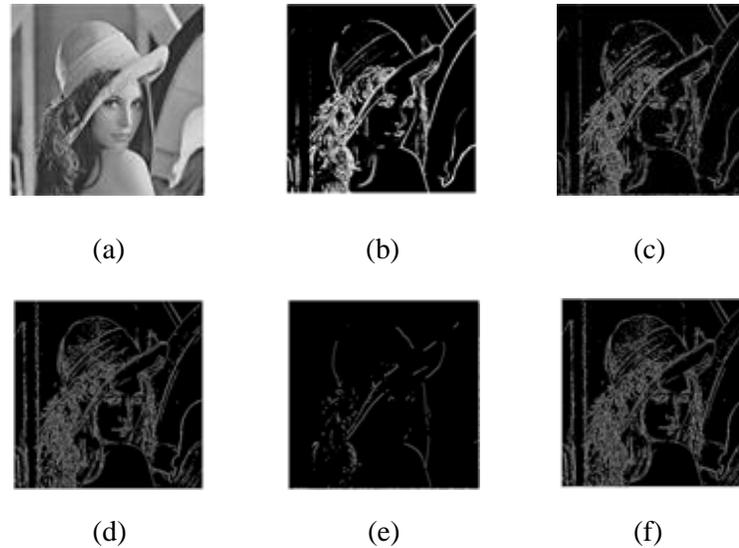


Figure 11 – Edge detection with different thresholding methods on image Lena a) Original image b) Otsu method c) Hysteresis method with predefined threshold value=0.01-0.1(normalized) d) PA with predefined threshold value (T=0.1, normalized) and edge point factor ( $\alpha = 0.6$ ) e) Hysteresis method with predefined threshold value=0.1-0.4 (normalized) f) PA with predefined threshold value (T=0.4, normalized) and edge point factor ( $\alpha = 0.15$ )

determining the true edge boundaries, especially for curved regions, which remain a challenging task for many of the edge operators.

Figure 12 shows examples of binary formatted morphological operations which are applied to the binary image.

- Figure 12-a shows that diagonal pixel which is attached to the lines or curves, is removed, bit “1” in the top-left corner can be put in different corners.
- Figure 12-b shows that H pixel(s) which lie between two lines, are removed; it removes at most two pixels.
- Figure 12-c and 12-d shows that the gap between two pixels will be filled with a single pixel and connects two vertical and horizontal lines.
- Figure 12-e shows the unwanted pixels removal which are attached to each other in a region, this process is under control of the user, somehow, the user can determine

the number of pixels that should be removed.

Some new developments were performed on the following morphological operations:

- Figure 12-f shows that vertical and horizontal pixel(s) which are attached to the lines or curves, are removed, also, it removes the pixels from left side, upside and downside.
- Figure 12-g shows how a single pixel which is attached to the lines or curves vertically or horizontally, are removed. The symbol '×' indicates that the pixel can be zero or one and the pixel can be attached left, right, up and down.
- Figure 12-h and 12-i shows that the gap between two pixels will be filled with a single pixel and connects the curves; the figures show examples of binary formats for horizontal up-right side curve connection and vertical down-left side curve connection, respectively. Also, it can connect the curve horizontally or vertically in different sides.
- Figure 12-j shows that the pixel(s) which lie in front of each other as parallel (double edges), are removed, they can be up to 3 pixels.
- Figure 12-k shows a pixel on the corner is removed to make the edge thinner. The symbol "×" indicates that the pixel can be zero or one and such a pixel can be situated in the different corners.

Figure 13 shows an example of before and after applying morphological operations on the image with respect to a particular area of the image.

$$\begin{array}{ccc}
 \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array} & \begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{array} \\
 \text{(a)} & \text{(b)} & \text{(c)} \\
 \\
 \begin{array}{ccc} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} & \begin{bmatrix} 0 & \dots & 0 \\ \vdots & 1 & \vdots \\ 0 & \dots & 0 \end{bmatrix} & \begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array} \\
 \text{(d)} & \text{(e)} & \text{(f)} \\
 \\
 \begin{array}{ccc} \times & 1 & \times \\ 1 & 1 & 1 \end{array} & \begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} & \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{array} \\
 \text{(g)} & \text{(h)} & \text{(i)} \\
 \\
 \begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} & \begin{array}{ccc} \times & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{array} \\
 \text{(j)} & \text{(k)}
 \end{array}$$

Figure 12 – Binary formatted morphological operations

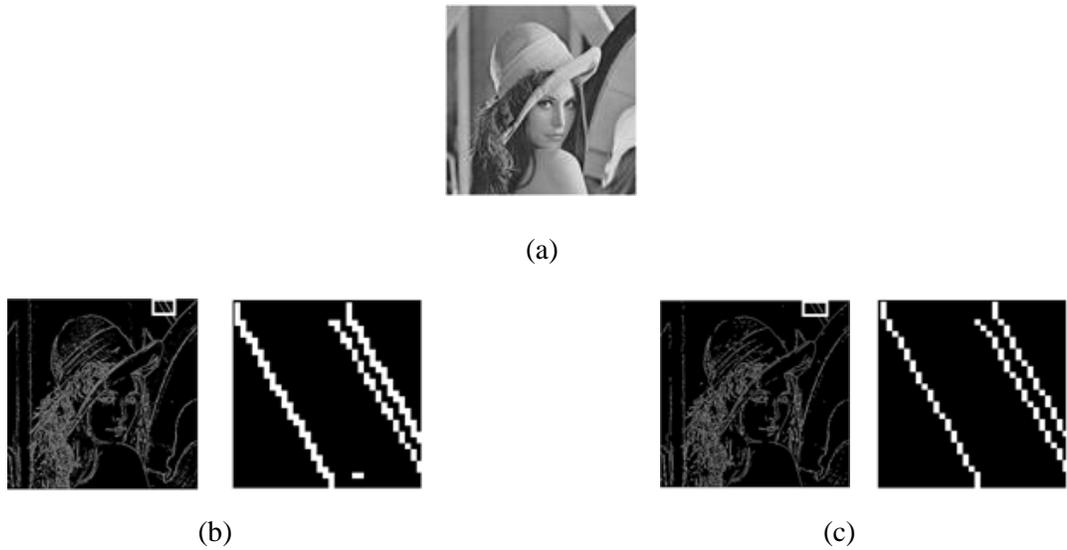


Figure 13 – Morphological operations on Lena image and the specified area within the white rectangle is compared in the two different conditions a) Original image b) Results before applying morphological operations c) Results after applying morphological operations

### 3.2 Wavelet-Based Filter for Gaussian and Combined Gaussian-Speckle Denoising

Other types of noise which can be appear in different forms and distributions such as impulse, speckle and Gaussian. As said before, speckle noise is one type of noise that is

multiplicative and independent and the general model of speckle noise [16] contains multiplicative and additive components, but in ultrasound images, the additive part can be overlooked [17]. Speckle noise is assumed to have a generalized Gamma (GG) distribution as in (3). Also, Gaussian noise is another type of noise that is also additive and independent, and the noisy image is as expressed in (2).

The wavelet and spatial filters can be used in order to reduce speckle, additive white Gaussian and salt and pepper noise in ultrasound images [197]. There are many filters in order to reduce Gaussian noise in the image which were discussed in chapter II. A combination of wavelet thresholding and Bilateral filter in the transform domain are often used in order to remove speckle noise in ultrasound images [198]. Modified total variation regularization is proposed in [199] in order to remove multiplicative noise. Furthermore, a combination of total variation, high-order total variation and a generalized Kullback-Leibler divergence method [200] is proposed in [201] in order to remove speckle noise. The Daubechies complex wavelet transform is used in order to remove speckle noise [202] in which imaginary component of complex scaling coefficient and shrinkage on complex wavelet coefficient are applied respectively, to detect edges and non-edges. Improved adaptive wavelet shrinkage is proposed in [203] based on correlation of the coefficients within and across the resolution scales.

There are some research studies reported on a combination of wavelet and total variation in order to remove speckle and Gaussian noise. A combination of wavelet and total variation is performed in studies [120, 204] in order to achieve low level Gaussian and speckle denoising on natural and ultrasound images. The method in [120] applied total variation [59] to LL sub-band of wavelet for one iteration and after inverse wavelet

transform, the remaining noise is removed by that one-iteration total variation. The method in [204] applied total variation [60] to LL sub-band of wavelet and used local variance of sub-bands for thresholding and finally, inverse wavelet transform is applied in order to obtain the reconstructed image. Also, there are several edge-preserving image speckle denoising methods that have been proposed in the past. The filters that are commonly used include the Frost filter [205], the Kuan filter [206], the Lee-diffusion filter [207], the Lee filter [208, 209] and the Geometry filter [210]. Most of the previous research studies were focused on removing speckle or Gaussian noise separately. The challenge is obviously amplified when these types of noise are combined.

Therefore, this study [211] introduces a new filter that combines the strengths of the dual complex wavelet domain filter [212] and improved total variation filter [60] in order to preserve edge details and overcome the presence of a Gaussian noise as well as combined speckle and Gaussian noise. The Motivation and resulting modifications that were made in this study can be explained through the following two main objectives: 1) Resolve the challenges faced with the use of denoising methods by keeping as much image details as possible, while avoiding blurring of the image, and hence preserving the sharper edges associated with boundaries in the presence of Gaussian noise and combined speckle noise and Gaussian noise. 2) Contend with these challenges even in the presence of high-intensity noise levels. Designing a denoising filter that integrates these two objectives highlights the novelty of the proposed method. As a practical application, and to assess the merits of our method, we applied the proposed denoising filter to natural and medical ultrasound images. Consequently, this study introduces a new denoising filter capable of preserving more edge details with high structural similarity to the original (noise-free) image even in the presence

of Gaussian noise as well as combined speckle noise and Gaussian noise even under high intensity levels. The results obtained, as will be discussed later, are contrasted to all other well-known denoising filters.

### 3.2.1 Proposed Method for Gaussian Denoising and Combined Gaussian and Speckle Denoising

#### 3.2.1.1 Description of the method

The block diagram of the fully integrated process is given in figure 14, showing the all steps of the denoising process. The iterated based structure of the total variation causes to miss some details and other textures. This is due to an inappropriate estimation of the number of iterations in the process [204]. On the other side, there is another challenge to predict the noisy coefficients and determine an appropriate threshold to remove them in the wavelet domain. The challenge is more complicated in the presence of random distribution of combined speckle and Gaussian noise in the image. In the case of using one of two wavelet or total variation algorithm independently, some high frequency noise components still remain in the image, which the algorithms are not able to suppress them alone. Some algorithms apply total variation only on LL sub-band of wavelet transform and remove the noise in other sub-bands with thresholding [204, 120], but, in the presence of combined noise, there is still noise in LH, HL and HH sub-bands, especially in higher noise intensity levels.

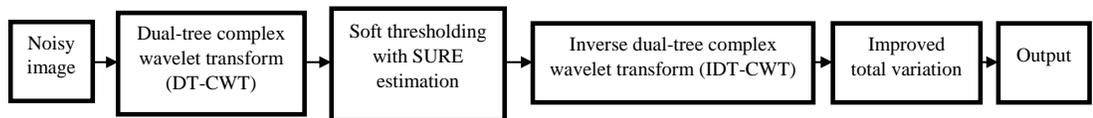


Figure 14 – Essential steps for Gaussian and combined Gaussian-speckle denoising

Therefore, in this study, a new structure is introduced which consists of two separated blocks: dual-tree complex wavelet transforms [212] and improved total variation [60]. In the proposed method, in addition to noise reduction and structural parameters extraction, boundary edges are assumed to yield high correlation with the original image, an important outcome especially in in ultrasound images.

In this integrated approach, noisy coefficients are removed using the dual-tree complex wavelet domain filter [212], which is a modification of the standard wavelet transform. The 2-D Discrete Wavelet Transform (DWT) coefficients and 2-D Discrete Wavelet Transform are as defined in (10) and (11).

$$C_i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \varphi_i(x, y), \quad i = LH, HL, HH \quad (10)$$

$$\varphi_i(x, y) = 2^j \varphi_i(2^j x - m, 2^j y - n) \quad (11)$$

Where  $I(x, y)$  is the  $M \times N$  input image,  $m$  and  $n$  are time shifts which control the different time points,  $j$  is scale factor which controls the frequency content and  $i$  is an index used for the three different wavelet functions. But, DWT has some problems such as oscillation, shift variance, aliasing and direction selectivity. Then, we switched to using complex wavelet transform (CWT) as defined in (12) which forms a Hilbert transform ( $90^\circ$  out of phase with each other). The CWT has another problem in that it cannot exactly express the Hilbert transform analytic.

$$C_c(x, y) = C_r(x, y) + jC_i(x, y), \quad \varphi_c(x, y) = \varphi_r(x, y) + j\varphi_i(x, y) \quad (12)$$

Finally, we switched to Dual tree- Complex Wavelet Transform (DT-CWT). It employs two real DWTs as expressed in (13), one is the real part (upper tree) and the second one is the imaginary part (lower tree).

$$\varphi(x, y) = \varphi_h(x, y) + j\varphi_g(x, y) \quad \varphi_g(x, y) = \mathcal{H}\varphi_h(x, y) \quad (13)$$

Where  $\mathcal{H}$  shows Hilbert transform. Each tree is divided in to low pass and high pass pairs. Real and imaginary parts are inverted and averaged to obtain the output. Then, the original output is recovered from either the real or the imaginary part as in (14).

$$\varphi(x, y) = [\varphi_h(x) + j\varphi_g(x)][\varphi_h(y) + j\varphi_g(y)] = \varphi_h(x)\varphi_h(y) - \varphi_g(x)\varphi_g(y) + j[\varphi_g(x)\varphi_h(y) + \varphi_h(x)\varphi_g(y)] \quad (14)$$

DT-CWT is implemented as two parallel-channel filter banks applied to the same data. In its structure, filters are purely real and meet the perfect reconstruction (PR) condition, and the phase shift of complex coefficients depends almost linearly on the displacement and as a result, this transform is shift invariant. The parallel trees are first applied to the rows, then, they are applied to the columns of the image. In the upper tree, sub-band signals are considered as the real part and in the lower tree, they are considered as the imaginary part of the transform. Each level of decomposition contains six complex high-pass sub bands and two complex low-pass sub bands. Low pass sub bands will iterate in the subsequent stages and high pass sub bands are the result of directional filtering of the signal in six different orientations ( $\pm 15^\circ$ ,  $\pm 45^\circ$ ,  $\pm 75^\circ$ ), which are set to provide directional selectivity. This will improve the accuracy of edge definition compared to real coefficients. In this study, the level of decomposition is set to 2, and soft shrinkage thresholding [106] with SURE estimation [130] are applied to the sub-bands as in (15) and (16), respectively.

$$\text{sgn}(c)(|c| - \lambda) \text{ if } |c| \geq \lambda, \text{ otherwise it is } 0 \quad (15)$$

$$\lambda = \underset{t \geq 0}{\text{argmin}} [N - 2[1:N] + \sum_{x,y=1}^N (\min(|c_{x,y}|, t))^2] \quad (16)$$

Where  $(c)$  is the absolute value of the wavelet coefficients,  $(\lambda)$  is a threshold and  $N$  is number of the coefficients  $(c_{x,y})$  in each sub-band. Then, inverse dual-tree complex wavelet transform is applied to the image in order to obtain the reconstructed image. The results obtained from the dual-tree complex wavelet transform are compared against the standard wavelet transform denoising filter in the presence of combined speckle noise and Gaussian noise on Lena and Cameraman as shown in table 5. The structural metrics include correlation and peak signal to noise ratio (PSNR).

Table 5 – PSNR and Correlation comparison of proposed method with DT-CWT and stationary wavelet transform (SWT) on different images in the presence of different combined speckle and Gaussian noise levels

	PSNR Lena	Correlation Lena	Correlation Cameraman	PSNR Cameraman
<b>DT-CWT</b>	21.5436	0.8966	0.8753	20.7149
<b>SWT</b>	20.8767	0.8766	0.8500	20.0452

(a)

	PSNR Lena	Correlation Lena	Correlation Cameraman	PSNR Cameraman
<b>DT-CWT</b>	20.7149	0.8753	0.8970	18.9431
<b>S-WT</b>	20.0452	0.8500	0.8779	18.4661

(b)

Finally, improved total variation [60], which is an improvement on total variation in [59], is applied to the image in order to smooth the image and remove the remaining noise, especially in high frequency sub-bands. The total variation minimization [59] is defined as in (17).

$$\min \frac{\|u-g\|^2}{2\lambda} + J(u) \quad (17)$$

Where  $u$  is the clean image,  $g$  is the observed image,  $\lambda$  is the Lagrange multiplier,  $\sigma^2$  is the estimated noise variance and  $J(u)$  is the total variation as defined in (18).

$$J(u) = \sum_{(i,j)=1}^N |(\nabla u)_{i,j}| \quad (18)$$

The improved total variation [60], which is based on dual information [213, 214], can be expressed by using the Euler-Lagrange equation given below.

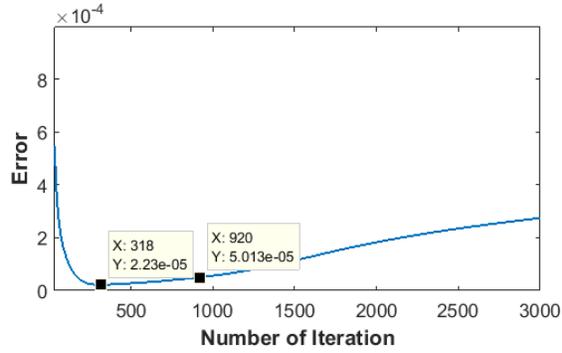
$$u = g - \pi_{\kappa\lambda}(g) \quad (19)$$

Where  $\pi_{\kappa\lambda}(g)$  is a non-linear orthogonal projection of  $g$  [204] and it is solved based on a fast minimization algorithm. Then, a new iterated way is suggested in [60] for image denoising to solve equation (20) in order to recover the original  $N \times N$  image.

$$\min\{J(u): \|u - g\|^2 = N^2\sigma^2\}, \pi_{\kappa\lambda}(g) = N\sigma \quad (20)$$

In this study, the stopping criterion has been set to  $5e-5$ , which empirically led to the lowest error. This value has been carefully chosen with over 3000 iteration of the algorithm. The minimum error has actually occurred at  $2.3e-5$ , but we selected the  $5e-5$  as our stopping criteria since it resulted in a visually smoother image. Figure 15-a shows the denoising error versus the number of iterations. In this figure, minimum error and desired threshold (stopping criteria) points are specified. Figure 15-b shows the resultant denoised image based on threshold= $2.3e-5$  (minimum error) and figure 15-c demonstrates the resulting image by selecting the desired threshold using  $5e-5$  as the stopping criteria for total variation.

Figure 16 shows the performance of the algorithm with and without DT-CWT in the presence of combined Speckle and Gaussian noise on Lena. As the figures show, the result is better when the DT-CWT is used.



(a)



(b)



(c)

Figure 15 – Stopping criteria for total variation a) Denoising error vs number of iteration with respect to minimum error and desired threshold b) Denoised image after minimum error c) Denoised image after desired threshold.



Figure 16 – Performance of the Wavelet-based algorithm with and without DT-CWT in the presence of combined Speckle ( $\sigma = 0.1$ ) and Gaussian ( $\sigma = 0.1$ ) noise on image Lena a) Original image b) Noisy image (c) Denoised without DT-CWT (d) Denoised with DT-CWT

### 3.2.1.2 Evaluation Measures

To measure the degree of edge preserving and to gauge the image structural metrics, correlation coefficient ( $\beta$ ), peak signal to noise ratio (PSNR), feature similarity index (FSIM) [215] and figure of merit (FOM) [190] are computed. The correlation coefficient

is defined as in (5). The peak signal to noise ratio (PSNR) and FOM measures are given by equations (7) and (8), respectively.

The feature similarity index (FSIM) [215] between the noisy image and the denoised image is provided in order to measure the degree of similarity and quality. This is based on the human visual system (HVS) which understands an image according to its low-level features. Phase congruency (PC) and gradient magnitude (GM) are used respectively as the primary and secondary features in FSIM. They have complementary roles in order to identify the image local quality. PC is a dimensionless measure of the significance of a local structure. Finally, the local quality map is obtained and then the PC is used as weighting function in order to achieve the desired quality score [215].

#### 3.2.1.3 *Structure of the Method*

Implementation of the proposed method assumes the following steps:

1. Load the input noisy image.
2. Apply dual-tree complex wavelet transform (DT-CWT) with 2-level decomposition. Farras filters are set for the first-stage of the dual-tree wavelet transform and 10-tap Kingsbury Q-shift filters for the next stage.
3. Soft shrinkage thresholding with SURE estimation on the noisy coefficients is applied. The equation (10) (sure shrink) is solved with the coefficients of the analysis and synthesis filters (previous step) to calculate the threshold  $\lambda$ . The number of coefficients for all stages is equal to 10 and we reached a value for  $\lambda = 1.4746$ . Then this  $\lambda$  is used to solve equation (9) in order to zero-out the noisy coefficients and keep the non-noisy ones.

4. Apply the inverse dual-tree complex wavelet transform (DT-CWT) in order to obtain the reconstructed image.
5. Improved total variation is applied in this step. The Lagrangian total variation minimization in equation (11) is solved, where  $J(u)$  is discrete total variation norm. It includes partial derivatives  $\sum_i(\sqrt{u(i)_x^2 + u(i)_y^2})$  and the regularization parameter ( $\lambda$ ) controls the amount of denoising. The algorithm linearly decreases the value of  $\lambda$  between two predefined maximum and minimum values. Then, the iterative algorithm is implemented. Finally, by reducing the total variation of the noisy image, the denoised image deemed a close match to the clean original image is obtained.

### *3.2.2 Edge Detection After Wavelet-Based Filtering*

The edge detection process is applied to the denoised image according to the defined process in section 3.1.2 with this difference that the maximum sequence block is removed from the process.

### *3.3 Design CNN Filter for Mixed Impulse and Gaussian Denoising*

The noise removal challenge is further amplified when the images are degraded by mixtures of impulse and Gaussian noise, significantly affecting the structural metrics of any given image. It is thus necessary to find a reliable process by which we could attenuate, and at best remove, the effects of such mixed noise. Therefore, a standard course of action is to perform an adequate smoothing technique to the image first before some form of gradient could be applied to preserve finer image details. Given that derivatives could amplify the effect of noise, a tradeoff must be negotiated between the task of decreasing

noise variance and that of keeping all relevant image details. This results in a subtle and difficult challenge in image processing especially when edge detection is to be performed on the resulting denoised image. Consequently, edge detection remains a challenging nontrivial problem although an essential preprocessing step for object identification, image segmentation, feature extraction, pattern recognition, and other relevant image processing tasks. When we deal with images, pertinent details can be useful when analyzing specific images for all types of real-world applications, but the concern has always been in delineating what really constitute an edge with high degree of similarity in contrast to other background and noise data that could be misidentified as real edges.

Several impulse and Gaussian denoising methods as well as mixed impulse and Gaussian denoising methods that have also been proposed which are discussed in chapter II. The methods uniquely used for impulse or for Gaussian denoising have shown good performance in the presence of the targeted noise, but it degrades when the two noises are mixed. Previously reported methods that have considered mixed impulse and Gaussian denoising filters are based on traditional methods that could not properly confront the problem when in the presence of high-intensity impulse and Gaussian noise, leading to loss of image details and excessive blurring burdening the edge detection process.

For a fair assessment of the proposed method against others that have focused on mixed impulse and Gaussian noise under different intensity levels, a comparative study is provided contrasting the results obtained using the proposed deep learning algorithm against those from the most recent and effective denoising filters. In this comparative assessment, we have focused our comparison against the low rank approximation algorithm (LSM-NLR) [166] which has proven to be effective for the removal of mixed impulse and

Gaussian noise in images. The LSM-NLR method is based on Laplacian scale mixture (LSM) modeling and non-local low rank regularization. In order to model the impulse noise in LSM, a MAP estimator is defined by the authors of [171]. For non-local rank regularization, a combination of the LSM model, the MAP estimator, and a low rank regularization model was used with the assumption that similar patches are interdependent, consequently resolving the denoising process by using an optimization algorithm. The author of this effective approach claims to obtain better result than the two-phase deblurring/denoising (TPD) method (or Cai1) [216], sparse and low-rank regularization denoising (SLR) [217], the well-known BM3D [108], non-locally centralized sparse representation [218], weighted encoding with sparse non-local regularization (WESNR) [164],  $l_0$ -nonlocal low rank, and  $l_1$ -nonlocal low rank.

We should note that the WESNR method does not have an impulse noise detection step due to its generated artifacts in high intensity noise levels. It shows that the image can be defined as a multiplication of sparse coding and a dictionary. In order to denoise the image, an optimal estimation of sparse code should be calculated by encoding the noisy image over the dictionary. Because of two different noise categories, the weight (close to 1 for pixels corrupted by Gaussian noise and smaller weights ( $w$ ) for pixels corrupted by impulse noise) is assigned to residuals; and therefore, an optimal estimation for sparse coding is defined in the presence of mixed noise based on sparse regularization. This method is claimed to outperform ROR-NLM [219], Cai [220],  $l_1 - l_0$  [162], TF [221], and M+BM3D [108].

This study's motivation can be explained by three objectives: 1) To determine new ways for overcoming the persisting problems experienced by previously reported denoising

methods in order to attenuate as much as possible the effects of noise, while keeping more of the image details; 2) To deblur the image in such a way as to yield an enhanced noise free image were the lack of noise is visually appreciable; 3) To preserve edge information yielding sharper and continuous edge boundaries, considered here as one of the main aims of this study.

Therefore, this study [222] introduces a new denoising filter capable of preserving more edge details while yielding high structural similarity to the original (noise-free) image even in the presence of high mixtures of impulse and Gaussian noise. The obtained results, as presented and discussed later, are contrasted to all other well-known denoising filters.

### 3.3.1 Proposed Method for Mixed Impulse and Gaussian Denoising

The proposed method, as illustrated in Figure 17, uses an end-to-end deep convolutional neural network (CNN) to achieve optimal denoising of mixed impulse and Gaussian noise

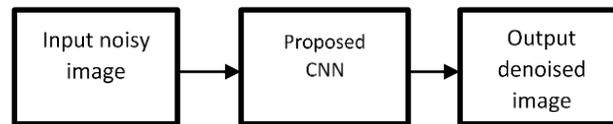


Figure 17 – Essential steps for proposed DCNN based denoising

and, consequently, directly estimates the original noise free image. Thereafter, batch normalization is applied to speed up and improve this denoising process. Finally, the network is trained for both specific and non-specific noise-levels denoising.

#### 3.3.1.1 Evaluation Measures

Standard structural metrics are computed to compare the performance of multiple filters against the proposed method and gauge the quality of the denoised image. The following metrics are used in this study:

- Feature similarity index (FSIM) [215] measures the quality of the denoised image based on the human visual system (HVS).
- Peak signal to noise ratio (PSNR) measures the level of noise remaining in the denoised image as equation (7).

### 3.3.1.2 *Related Works on Denoising*

There are some notable previous works which use CNNs for image denoising. The authors of [86] propose a combination of sparse coding and deep neural networks, pre-trained with denoising auto-encoders (DAs), as an alternative to training pure DAs. In [223], the authors use multilayer perceptron (MLP) for image denoising. The Trainable Nonlinear Reaction Diffusion (TRND) presented in [87] uses supervised training to train a dynamic nonlinear reaction diffusion model with time-dependent parameters (linear filters and influence functions) for Gaussian denoising. The method proposed in [87] is similar to the feedback convolutional neural network (CNN) presented in [88], and in both cases each iteration (stage) of the proposed diffusion process uses convolutional operations of a set of linear filters. The deep convolutional neural network method (DnCNN) introduced in [89] can be seen as a generalization of TRND [87] that: 1) is easier to train, by replacing the influence function with a rectified linear units (ReLU) [224]; 2) increases architectural depth (number of convolution layers) to improve image modeling capacity; and 3) incorporates batch normalization [225] to improve performance.

All the aforementioned networks are used solely for Gaussian denoising and all of them, except [89], were used to remove known/predefined noise levels. As Gaussian noise is additive, the network in [89] removes the noise by using residual learning, that is, learning the noise. When Gaussian and impulse noise are mixed the resulting interference is no

longer additive. Therefore, in this paper we cannot use residual learning to directly estimate the denoised image.

### 3.3.1.3 *Batch Normalization and Network Parameters*

During training, any change to a deep neural network layer's parameters causes a change in distribution of the following layer's input, referred to as an internal covariate shift. Batch normalization [225] can alleviate internal covariate shift by learning the normalization parameters of each part of the model and applying it to each training mini-batch. Batch normalization has several advantages that include faster convergence (preventing the gradient to be zero in backpropagation), flexibility for incorporating larger learning rates, independency from weight initialization, and lack of need for drop-out.

We have seen from previous work that the network's depth and patch size are dependent on the type and level noise present. Specifically, larger patch sizes are shown to exhibit better performance in the presence of higher noise levels [226]. The network presented in [87] used 10 convolutional layers (or 5 stages) with patches of  $61 \times 61$  to remove a predefined level of Gaussian noise. The network in [89] uses 17 layers of  $40 \times 40$  patches for specific-noise-level denoising and 20 layers of  $50 \times 50$  patches for non-specific noise-level Gaussian denoising.

### 3.3.1.4 *Network Model*

The proposed CNN model is a modified version of the VGG [227] and DnCNN [89] models aimed at overcoming the challenge of mixed impulse and Gaussian noise removal. The network's input is a noisy image  $y_i$ , produced by artificially injecting noise to a clean original ( $x_i$ ) image, and the network's output  $f(y_i)$  is an estimate of the original noise-free image. The network's loss function is the summation of the squared error between the

estimated and original noise-free images as formulated in (21). Finally, the network's parameters are updated by minimizing this loss function.

$$L = \sum_{i=1}^N \|f(y_i) - x_i\|_2^2 \quad (21)$$

Where  $N$  is number of training image sets  $(\{y_i, x_i\})$ .

The network used in this study resembles that used in [89] with few changes. It is composed of 3 different types of layers where: the 1<sup>st</sup> is a convolutional layer of 64  $3 \times 3 \times 1$  filters with ReLU non-linear activation functions [224] used to create 64 feature maps, the 2<sup>nd</sup> through second-to-last layers are batch normalized [225] convolutional layers of 64  $3 \times 3 \times 64$  filters with ReLU activations [224]; and the last is a convolutional layer made out of a single  $3 \times 3 \times 64$  kernel used to output the reconstructed image. The use of ReLU activations [224] on convolutional layers separates the mixed noise from the noisy observations through the hidden layers. Finally, the input images are directly padded with zeros to reduce boundary artifact [89] resulting from size mismatches between different input images. Figure 18 illustrates the considered network model.

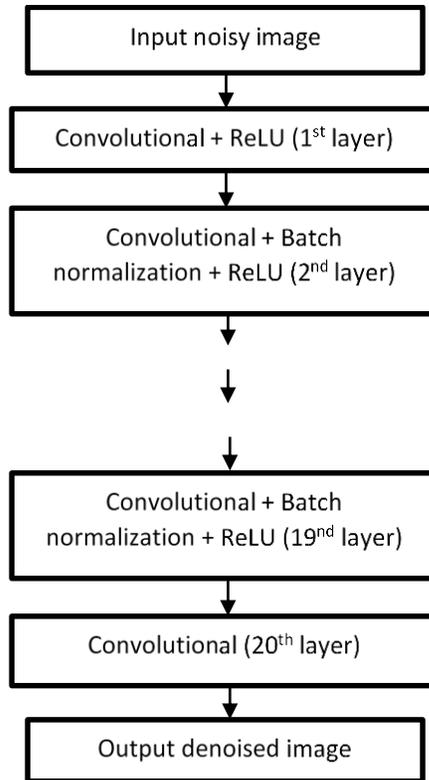


Figure 18 – Proposed DCNN model

In contrast to [89], this network attempts to predict the clean noise-free image directly instead of obtaining it from subtracting the predicted residuals from the noisy image. This is a result of the non-additive nature of the types of noise dealt with by this paper. Although we use batch normalization to prevent overfitting, we also implemented extra steps to further avoid it. Every time the network starts a new epoch (run through the training data) a new random seed is used to regenerate the noisy images. This extra step has proved to be a very helpful regularization technique as it prevents the network from seeing the same input image twice, or at least assign a very low probability to such event, allowing the network to better generalize.

We should indicate that we obtained optimal denoising results by using 20 layers with  $40 \times 40$  patches for both specific and non-specific noise-level denoising. Although either, Stochastic gradient descent (SGD)-momentum [228] and Adam gradient-based optimization [229] could have been used, previous convolutional neural networks [230-233] [89] have used stochastic gradient descent (SGD), and the performance of networks with and without batch normalization for both SGD-momentum and Adam are shown in [89]. This demonstrates that batch normalization can significantly improve the PSNR for SGD by increasing the number of epochs. Therefore, in this paper stochastic gradient descent (SGD)-momentum [228] is used.

Lastly, all the source-code written in the deployment of this research study is available at: “<https://github.com/wizquierdo/DnCNN>” for other researchers to perform comparative assessments and explore for any potential improvements that can be made to the proposed method.

### *3.3.2 Edge Detection After CNN Filtering*

The edge detection process is applied to the denoised image according to the defined process in section 3.1.2.

## 4. CHAPTER IV

### RESULTS AND DISCUSSIONS

#### 4.1 *Denoising Filters Comparisons*

##### 4.1.1 *Impulse Denoising Filters*

Table 6 compares the Peak Signal to Noise Ratio (PSNR), correlation, and Structural Similarity Index Metric (SSIM) of some of the relevant impulse denoising filters. Moving windows of 3x3, 5x5, and 9x9 sizes were respectively used for images with 10%, 60%, and 90% noise intensity.

##### 4.1.2 *Gaussian Denoising Filters*

Table 7 compares the averaged Peak Signal to Noise Ratio (PSNR) of some of the discussed Gaussian denoising filters. The images whose PSNR were average included the standard images of Camera man, House, Peppers, Starfish, Monarch, Airplane, Parrot, Lena, Barbara, Boat, Man, Couple. The methods compared in Table 4 are based on machine learning algorithms (neural networks, deep learning) and the PSNR values are calculated on the average of the 12 testing images similar to the standard images used in [89]. Table 8 compares the Peak Signal to Noise Ratio (PSNR) of some of the discussed Gaussian denoising filters. The methods indicated in table 8 are based on the traditional methods of spatial non-linear filters, and the PSNR and SSIM are calculated on the basis of one testing image, the Parrot image as in [68], deemed sufficient for this type of comparison involving PSNR and SSIM. Table 9 compares the SSIM measure of some of the discussed Gaussian

denoising filters. The Bird image is used for comparison in table 9 [68] and is also deemed sufficient for this type of comparison.

Table 6 – PSNR, Correlation, and Structural Similarity Index Metric (SSIM) comparison of some of the discussed impulse denoising filters

		<i>Lena</i>			<i>Camerman</i>		
	Noise level	Correlation	PSNR	SSIM	Correlation	PSNR	SSIM
DMFUTMF [44]	10 %	0.9659	26.1517	0.8783	0.9366	21.2120	0.7910
	60 %	0.8415	18.9888	0.5335	0.8048	16.0254	0.4616
	90 %	0.5772	11.2136	0.1004	0.3431	9.9221	0.0661
The algorithm [21]	10 %	0.9577	24.9189	0.9402	0.9328	20.9744	0.9094
	60 %	0.8981	19.0121	0.8411	0.8731	15.8791	0.6807
	90 %	0.7141	14.5981	0.4383	0.5621	12.5981	0.3973
FIDRM [55]	10 %	0.9681	25.6991	0.9432	0.9422	21.7111	0.9100
	60 %	0.9133	20.3459	0.8417	0.8741	16.3336	0.7519
	90 %	0.8103	16.2993	0.5221	0.6473	13.3112	0.4812
DBUTMF [33]	10 %	0.9715	26.8279	0.9469	0.9584	22.8511	0.9231
	60 %	0.9137	21.0671	0.8427	0.8821	17.2112	0.7823
	90 %	0.8593	17.8893	0.6421	0.7891	15.1173	0.5949
DBPTGMF [47]	10 %	0.9720	26.9001	0.9473	0.9587	22.8794	0.9240
	60 %	0.9163	21.3123	0.7852	0.8948	18.7242	0.7852
	90 %	0.8621	18.0001	0.6048	0.7846	15.4401	0.6048
Cascading [50]	10 %	0.9722	26.9181	0.9559	0.9580	22.8053	0.9220
	60 %	0.9286	22.7196	0.8190	0.8777	18.0692	0.7418
	90 %	0.8324	18.8145	0.5987	0.7413	14.5668	0.5234
The algorithm [234]	10%	0.9714	26.8173	0.9377	0.9576	22.7594	0.9003
	60%	0.9330	22.9959	0.7625	0.8960	18.5976	0.6813
	90%	0.8686	19.0761	0.6806	0.7895	14.8671	0.5676
DBUTMWMF [49]	10 %	0.9722	26.9401	0.9574	0.9581	22.8361	0.9227
	60 %	0.9366	23.2440	0.8599	0.8951	18.7293	0.7851
	90 %	0.8547	19.6696	0.6917	0.7726	15.4655	0.5640
IBDNDF [36]	10 %	0.9720	26.8841	0.9542	0.9507	22.6953	0.9174
	60 %	0.9339	23.0719	0.8369	0.8865	18.4298	0.7553
	90 %	0.8677	19.8145	0.6976	0.7859	15.3892	0.6074
IBINRF [41]	10 %	0.9725	26.9733	0.9586	0.9588	22.9110	0.9254
	60 %	0.9402	23.5241	0.8831	0.9032	19.1352	0.7947
	90 %	0.8725	20.0350	0.7496	0.7963	15.6989	0.6093
UWMF [39]	10 %	0.9725	26.9813	0.9580	0.9586	22.8891	0.9251
	60 %	0.9396	23.4561	0.8633	0.9021	19.0629	0.7929
	90 %	0.8709	19.9437	0.7039	0.7956	15.6536	0.6136
INMF [42]	10 %	0.9831	29.3112	0.9721	0.9802	26.2762	0.9531
	60 %	0.9472	24.2273	0.8701	0.9189	19.9385	0.8021
	90 %	0.8751	20.1331	0.7179	0.8107	15.9397	0.6268
SAMWMF [185]	10 %	0.9843	29.4961	0.9744	0.9821	26.5929	0.9576
	60 %	0.9478 – 0.9499	24.1751 – 24.3335	0.8641 – 0.8744	0.9165 – 0.9219	19.8201-20.0877	0.7940 – 0.8080
	90 %	0.8725 – 0.8800	20.0941 – 20.2852	0.7032 – 0.7253	0.8011 – 0.8122	15.8451 -16.0122	0.6116 – 0.6315

Table 7 – Averaged PSNR comparison of some of the discussed Gaussian denoising filters (based on machine learning techniques)

Gaussian noise (standard deviation)	Type of denoising filters						
	BM3D [108]	WNNM [235]	CSF [236]	MLP [223]	TNRD [87]	DnCNN-S [89]	DnCNN-B [89]
0.15	32.372	32.696	32.318	-	32.502	32.859	32.680
0.25	29.969	30.257	29.837	30.027	30.055	30.436	30.362
0.50	26.722	27.052	-	26.783	26.812	27.178	27.206

Table 8 – PSNR comparison of some of the discussed Gaussian denoising filters (based on spatial non-linear methods)

Gaussian noise (standard deviation)	Type of denoising filters						
	Catte [65]	TV [59]	EAD [237]	CTD [67]	MPM [238]	Improved Anisotropic diffusion 1 [68]	Improved Anisotropic diffusion 2 [68]
0.15	32.1392	33.9370	32.4884	33.2009	31.7202	33.4551	33.4537
0.20	29.3338	31.3270	30.8812	30.3440	30.7569	31.8628	32.0353
0.25	25.7468	28.5566	28.9374	26.3181	29.5313	30.3513	30.6436
0.30	22.6088	26.7680	26.7680	22.9874	28.3664	28.9608	29.3514
0.35	20.1988	23.9319	24.7122	20.2339	27.2262	27.6440	28.1019

Table 9 – Structural Similarity Index Metric (SSIM) comparison of some of the discussed Gaussian denoising filters

Gaussian noise (standard deviation)	Type of denoising filters						
	Catte [65]	TV [59]	EAD [237]	CTD [67]	MPM [238]	Improved Anisotropic diffusion 1 [68]	Improved Anisotropic diffusion 2 [68]
0.15	0.9731	0.9837	0.9745	0.9823	0.9663	0.9748	0.9720
0.20	0.9676	0.9799	0.9730	0.9705	0.9652	0.9724	0.9701
0.25	0.9513	0.9719	0.9696	0.9502	0.9635	0.9699	0.9680
0.30	0.9205	0.9581	0.9603	0.9144	0.9608	0.9666	0.9654
0.35	0.8794	0.9394	0.9455	0.8633	0.9567	0.9618	0.9614

#### 4.1.3 Mixed Impulse and Gaussian Denoising Filters

Tables 10 and 11 provide comparisons of the averaged peak signal to noise ratio (PSNR) and averaged image perceptual quality index (FSIM) of some of the discussed mixed impulse and Gaussian denoising filters on 12 images. The images Lena, F16, Leaves, Boat, Couple, Fingerprint, Hill, Man, Peppers, Painting and Average are used as standard images for comparative purposes. Tables 12 and 13 provide a comparison of the averaged peak signal to noise ratio (PSNR) and averaged image perceptual quality index (FSIM) of some of the discussed mixed impulse and Gaussian denoising filters. The images Lena, FG, Boat,

Hill, Peppers, Man, Couple, AP, Cloth, Vase, Bush, Flower were used for comparative purposes as shown in tables 12 and 13.

Table 10 – Averaged peak signal to noise ratio (PSNR) comparison of some of the discussed mixed impulse and Gaussian denoising filters

Gaussian noise (standard deviation)	Impulse noise (level)	Type of denoising filter			
		ROR-NLM [219]	Cai [220]	$l_1 - l_0$ [162]	WESNR [164]
0.1	30%	27.6027	29.8790	31.8109	31.3600
	40%	26.5590	28.9290	30.6754	30.6309
	50%	21.2990	27.8354	29.4290	29.6663
0.2	30%	25.1118	27.6600	28.9027	31.4636
	40%	24.1227	27.0627	28.1281	28.2509
	50%	21.4790	25.4827	27.1900	27.4809
0.25	30%	24.1327	26.7172	27.8636	27.9100
	40%	23.0354	26.2172	27.1436	27.3154
	50%	20.4409	25.4827	26.3172	26.5718

Table 11 – Averaged image perceptual quality index (FSIM) comparison of some of the discussed mixed impulse and Gaussian denoising filters

Gaussian noise (standard deviation)	Impulse noise (level)	Type of denoising filter			
		ROR-NLM [219]	Cai [220]	$l_1 - l_0$ [162]	WESNR [164]
0.1	30%	94.5000	95.6909	97.0154	96.7063
	40%	93.1700	94.5800	96.1927	96.1700
	50%	88.8263	89.5381	95.0400	95.3563
0.2	30%	88.2336	92.3518	93.7163	93.6018
	40%	86.0090	91.2409	92.7163	92.8263
	50%	80.9609	89.5381	91.5227	91.7854
0.25	30%	85.1118	90.5818	91.9881	92.2709
	40%	82.5336	89.5054	91.0718	91.2309
	50%	77.2509	87.8072	89.8200	90.1327

## 4.2 Impulse Denoising Based on Spatial Filter

### 4.2.1 Implementation on Natural Images

In this section, the results obtained using the proposed method after applying the steps of denoising (SAMFWMF) followed by edge detection is presented. The results of SAMFWMF are compared with the cascading algorithm [50], IBDNDF [36], DBUTMWMF [49] and UWMF [39] filters on different images and under different impulse noise intensity levels. Images of Lena (512×512), Camera man (256×256), Coins

Table 12 – Averaged peak signal to noise ratio (PSNR) comparison of some of the discussed mixed impulse and Gaussian denoising filters

Gaussian noise (standard deviation)	Impulse noise (level)	Type of denoising filter						
		TPD [216]	BM3D [108]	WESNR [164]	SLR [217]	L <sub>1</sub> -NLR [171]	L <sub>0</sub> -NLR [171]	LSM-NLR [171]
0.1	10%	28.78	3057	30.24	30.50	31.25	31.36	32.30
	20%	27.97	29.46	29.36	29.18	29.46	29.86	30.82
	30%	27.15	28.30	28.40	27.82	27.74	28.55	29.37
	40%	26.02	26.67	27.02	26.20	26.74	26.92	27.24
	50%	24.92	24.54	25.30	24.18	24.72	25.18	25.36
0.2	10%	27.96	27.96	27.69	27.62	28.62	28.90	29.22
	20%	25.84	27.21	27.09	26.79	27.50	27.78	28.27
	30%	25.29	26.36	26.42	25.86	26.21	26.74	27.28
	40%	24.37	24.97	25.24	24.54	24.83	25.37	26.08
	50%	23.42	23.22	23.86	22.88	23.19	24.00	24.62
0.3	10%	24.58	26.08	26.11	25.60	26.56	26.95	26.98
	20%	24.21	25.39	25.55	24.85	25.49	25.92	26.20
	30%	23.76	24.65	24.92	24.03	24.30	25.00	25.39
	40%	22.95	23.35	23.74	22.83	23.11	23.74	24.33
	50%	21.99	21.77	22.30	21.35	21.62	22.52	23.18
0.5	10%	22.19	23.66	23.16	22.83	23.83	24.14	24.27
	20%	21.94	22.99	22.59	22.18	22.73	23.20	23.61
	30%	21.59	22.29	21.84	21.44	21.59	22.31	22.85
	40%	20.87	21.05	20.78	20.35	20.73	21.14	21.95
	50%	19.83	19.62	19.35	19.08	19.34	20.08	20.73

Table 13 – Averaged image perceptual quality index (FSIM) comparison of some of the discussed mixed impulse and Gaussian denoising filters

Gaussian noise (standard deviation)	Impulse noise (level)	Type of denoising filter						
		TPD [216]	BM3D [108]	WESNR [164]	SLR [217]	L <sub>1</sub> -NLR [171]	L <sub>0</sub> -NLR [171]	LSM-NLR [171]
0.1	10%	95.59	97.11	96.65	97.11	97.48	97.57	97.63
	20%	94.77	96.43	96.06	96.29	96.63	96.78	96.99
	30%	93.78	95.43	95.25	95.15	95.32	95.78	96.17
	40%	91.89	93.38	93.69	93.16	93.25	93.93	93.88
	50%	89.60	89.58	90.99	89.63	90.23	90.96	91.00
0.2	10%	93.76	93.76	92.82	93.66	94.29	94.77	94.78
	20%	90.49	92.82	92.12	92.58	93.31	93.56	93.80
	30%	89.43	91.52	91.27	91.18	91.91	92.15	92.67
	40%	87.13	88.96	89.44	91.18	89.05	89.78	90.83
	50%	84.70	85.19	87.07	88.73	85.86	86.70	87.92
0.3	10%	86.98	90.11	90.19	85.14	91.06	91.67	91.68
	20%	85.12	88.81	89.32	90.15	89.66	89.86	90.42
	30%	84.92	87.33	88.30	88.73	87.91	87.96	89.17
	40%	82.45	84.48	85.95	87.06	84.47	85.06	86.59
	50%	79.89	80.73	82.93	84.32	81.13	81.93	83.78
0.5	10%	79.84	84.04	82.71	80.72	85.19	85.54	86.31
	20%	78.55	82.21	80.92	83.93	83.09	82.66	84.60
	30%	77.31	80.51	78.94	82.02	80.82	79.86	82.72
	40%	74.88	77.52	76.17	77.06	76.79	76.39	79.66
	50%	72.37	74.04	72.51	73.71	73.53	73.74	76.80

(300×246) and checkerboard (256×256) are standard examples used in the literature for comparative purposes.

Table 14 shows the results obtained on the execution time (in seconds) after the denoising process, comparing different denoising filters to the SAMFWMF. All the experiments were run on a PC with Intel(R) core (TM) 2 Quad CPU 2.67GHZ and 8G RAM. All filters except for UWMF [39] yielded high execution time in the presence of high-intensity impulse noise. As the results show, SAMFWMF has a high execution time because of the complicated nesting blocks. Most of the execution time of SAMFWMF is dedicated to switching adaptive median filter (more than 70%) and the rest of the time is dedicated to the shrinkage window and weighted fixed mean filters. By decreasing the initial adaptive median window size, the execution time is increased. Also, by decreasing the shrinkage window size, the execution time is decreased.

Table 14 – Execution time after proposed spatial filtering process

	10%	40%	80%	90%
UWMF [39]	0.309078	0.465255	1.002563	1.652859
IBDNDF [36]	0.622326	1.580908	5.242255	11.045418
DBUTMWMF [49]	1.839567	4.002603	3.621499	4.868026
Cascading algorithm [50]	5.268536	6.429431	8.824390	11.972124
<b>Proposed Algorithm</b>	<b>8.481578</b>	<b>8.564322-8.659768</b>	<b>8.576215-10.387456</b>	<b>8.553084-12.779870</b>

Tables 15 and 16, show respectively the results obtained on the correlation ( $\beta$ ) and the peak signal to noise ratio (PSNR) measured in comparison to the different initial adaptive median window sizes for the SAMFWMF. Note that in tables 15 and 16, higher numbers are associated with better results. All these metrics/measures are computed in the presence of 10 up to 90 percent impulse noise and switch 1 is used for SAMFWMF. Also, table 17 shows the results for the computed structural metrics using the checkerboard as a challenging example for different initial adaptive median window sizes for the SAMFWMF. In this case, switch 2 is used for the SAMFWMF, given the nature of the

checkerboard image used here as a challenge for denoising such type of input images. As Tables 15-17 show, by increasing the initial adaptive median window size (approaching to maximum predefined window size), the values of the structural metrics are decreased, but, the edges are sharper and smoother; therefore, there is a tradeoff between better image similarity with less noise and the need for sharper edges.

Table 15 – Correlation ( $\beta$ ) measured in comparison to the different initial adaptive median window sizes for the proposed spatial filter

	<i>Initial window=3×3</i>			<i>Initial window=5×5</i>			<i>Initial window=7×7</i>			<i>Initial window=9×9</i>		
	<i>Lena</i>	<i>Camera man</i>	<i>Coins</i>									
10%	0.9843	0.9821	0.9933	-	-	-	-	-	-	-	-	-
20%	0.9834	0.9790	0.9921	-	-	-	-	-	-	-	-	-
30%	0.9819	0.9753	0.9906	-	-	-	-	-	-	-	-	-
40%	0.9548	0.9345	0.9799	0.9528	0.9301	0.9785	-	-	-	-	-	-
50%	0.9524	0.9282	0.9770	0.9508	0.9231	0.9755	-	-	-	-	-	-
60%	0.9499	0.9219	0.9734	0.9478	0.9165	0.9720	-	-	-	-	-	-
70%	0.9459	0.9138	0.9690	0.9432	0.9083	0.9666	-	-	-	-	-	-
80%	0.9160	0.8663	0.9544	0.9136	0.8624	0.9501	0.9106	0.8571	0.9504	-	-	-
90%	0.8800	0.8122	0.9300	0.8773	0.8077	0.9287	0.8754	0.8071	0.9277	0.8725	0.8011	0.9274

Table 16 – Peak signal to noise ratio (PSNR) measured in comparison to the different initial adaptive median window sizes for the proposed spatial filter

	<i>Initial window=3×3</i>			<i>Initial window=5×5</i>			<i>Initial window=7×7</i>			<i>Initial window=9×9</i>		
	<i>Lena</i>	<i>Camera man</i>	<i>Coins</i>									
10%	29.4967	26.5927	31.7914	-	-	-	-	-	-	-	-	-
20%	29.2478	25.9157	31.1160	-	-	-	-	-	-	-	-	-
30%	28.8934	25.2241	30.3653	-	-	-	-	-	-	-	-	-
40%	24.9708	20.8165	26.8780	24.5841	20.5567	26.6311	-	-	-	-	-	-
50%	24.5506	20.4417	26.3245	24.4105	20.1538	26.1080	-	-	-	-	-	-
60%	24.3336	20.0870	25.7222	24.1757	19.8203	25.5401	-	-	-	-	-	-
70%	24.0080	19.6702	25.0923	23.8142	19.4124	24.7978	-	-	-	-	-	-
80%	21.9598	17.5974	23.2398	21.8666	17.5196	22.9917	21.7218	17.3890	22.9759	-	-	-
90%	20.2855	16.0121	21.2778	20.2264	15.9205	21.2165	20.1514	15.9198	21.2199	20.0942	15.8457	21.2177

Table 17 – Computed structural metrics using the checkerboard for different initial adaptive median window sizes for the proposed spatial filter

	<i>Initial window=3</i>			<i>Initial window=5</i>			<i>Initial window=13</i>		
	30%	50%	80%	30%	50%	80%	30%	50%	80%
$\beta$	0.9595	0.9307	0.8076	-	0.9295	0.8054	-	-	0.8007
PSNR	17.0082	14.6801	10.2579	-	14.6175	10.1606	-	-	10.0947
SSIM	0.8417	0.7953	0.6599	-	0.7915	0.6497	-	-	0.6492

Tables 18-20 show respectively the results obtained on the correlation ( $\beta$ ), the peak signal to noise ratio (PSNR) and the structural similarity (SSIM) measures, comparing different denoising filters to the SAMFWMF based on the minimum initial adaptive median window size to the maximum predefined window size. All these metrics/measures are computed in

the presence of 10 to 90 percent impulse noise and switch 1 is used for the proposed filter. In tables 18-20, higher numbers are again associated with better results. As the results show, SAMFWMF yielded better structural metrics. Table 21 shows the results for the computed structural metrics using the checkerboard example for comparing the results obtained using different denoising filters to the SAMFWMF. In this case, switch 2 is used for the SAMFWMF, with higher numbers indicating better results. As the results show, SAMFWMF has better structural metrics. Table 22 shows the results obtained on the correlation ( $\beta$ ) and the peak signal to noise ratio (PSNR) measures, comparing other denoising filters with and without fixed weighted mean filter as a post-processing step. All of these metrics/measures are computed in the presence of 10 to 90 percent impulse noise. As the results indicate, the structural metrics are increased when the fixed weighted mean filter is used as a post-processing step for other denoising filters.

Table 18 – Correlation ( $\beta$ ) comparison for different denoising filters to the proposed spatial filter

	Cascading algorithm [50]			IBDNDF [36]			DBUTMWMF [49]			UWMF [39]			SAMFWMF		
	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins
10%	0.9722	0.9580	0.9893	0.9720	0.9567	0.9885	0.9722	0.9581	0.9892	0.9725	0.9586	0.9897	<b>0.9843</b>	<b>0.9821</b>	<b>0.9933</b>
20%	0.9704	0.9529	0.9873	0.9704	0.9518	0.9861	0.9712	0.9543	0.9880	0.9717	0.9566	0.9886	<b>0.9834</b>	<b>0.9790</b>	<b>0.9921</b>
30%	0.9681	0.9472	0.9842	0.9682	0.9464	0.9836	0.9701	0.9508	0.9863	0.9704	0.9533	0.9869	<b>0.9819</b>	<b>0.9753</b>	<b>0.9906</b>
40%	0.9396	0.9024	0.9716	0.9404	0.9031	0.9724	0.9420	0.9080	0.9751	0.9441	0.9128	0.9763	<b>0.9528-0.9548</b>	<b>0.9301-0.9345</b>	<b>0.9785-0.9799</b>
50%	0.9348	0.8905	0.9646	0.9376	0.8950	0.9692	0.9397	0.9011	0.9722	0.9424	0.9085	0.9738	<b>0.9508-0.9524</b>	<b>0.9231-0.9282</b>	<b>0.9755-0.9770</b>
60%	0.9286	0.8777	0.9574	0.9339	0.8865	0.9637	0.9366	0.8951	0.9683	0.9396	0.9021	0.9698	<b>0.9478-0.9499</b>	<b>0.9165-0.9219</b>	<b>0.9720-0.9730</b>
70%	0.9199	0.8616	0.9481	0.9303	0.8799	0.9591	0.9323	0.8852	0.9639	0.9355	0.8927	0.9647	<b>0.9432-0.9459</b>	<b>0.9083-0.9138</b>	<b>0.9663-0.9690</b>
80%	0.8781	0.8067	0.9205	0.9018	0.8340	0.9442	0.8948	0.8295	0.9388	0.9063	0.8479	0.9497	<b>0.9106-0.9160</b>	<b>0.8571-0.8663</b>	<b>0.9504-0.9544</b>
90%	0.8324	0.7413	0.8846	0.8677	0.7859	0.9216	0.8547	0.7726	0.9117	0.8709	0.7956	0.9248	<b>0.8725-0.8800</b>	<b>0.8011-0.8122</b>	<b>0.9274-0.9300</b>

Table 19 – Peak signal to noise ratio (PSNR) comparison for different denoising filters against the proposed spatial filter

	Cascading algorithm [50]			IBDNDF [36]			DBUTMWMF [49]			UWMF [39]			SAMFWMF		
	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins
10%	26.918	22.805	29.666	26.884	22.695	29.379	26.940	22.836	29.656	26.981	22.889	29.862	<b>29.496</b>	<b>26.592</b>	<b>31.791</b>
20%	26.675	22.320	28.970	26.661	22.259	28.613	26.802	22.492	29.233	26.861	22.697	29.443	<b>29.247</b>	<b>25.915</b>	<b>31.116</b>
30%	26.345	21.834	28.034	26.382	21.821	27.929	26.656	22.206	28.700	26.681	22.403	28.870	<b>28.893</b>	<b>25.224</b>	<b>30.365</b>
40%	23.434	19.012	25.335	23.514	19.088	25.460	23.611	19.272	25.905	23.771	19.533	26.198	<b>24.584-24.758</b>	<b>20.556-20.816</b>	<b>26.631-26.878</b>
50%	23.109	18.515	24.406	23.318	18.759	25.006	23.448	18.986	25.449	23.646	19.332	25.734	<b>24.410-24.550</b>	<b>20.153-20.441</b>	<b>26.108-26.324</b>
60%	22.719	18.069	23.625	23.071	18.429	24.318	23.244	18.729	24.914	23.456	19.062	25.261	<b>24.175-24.333</b>	<b>19.820-20.087</b>	<b>25.540-25.722</b>
70%	22.242	17.555	22.790	22.847	18.191	23.803	22.981	18.372	24.385	23.181	18.677	24.451	<b>23.814-24.008</b>	<b>19.412-19.670</b>	<b>24.797-25.092</b>
80%	20.296	15.941	20.806	21.220	16.595	22.322	21.131	16.784	22.185	21.453	17.033	22.790	<b>21.721-21.959</b>	<b>17.389-17.597</b>	<b>22.975-23.239</b>
90%	18.814	14.566	19.107	19.814	15.389	20.719	19.669	15.465	20.563	19.943	15.653	20.980	<b>20.094-20.285</b>	<b>15.845-16.012</b>	<b>21.217-21.277</b>

Table 20 – Structural similarity (SSIM) comparison for different denoising filters against the proposed spatial filter

	Cascading algorithm [50]			IBDNDF [36]			DBUTMWMF [49]			UWMF [39]			SAMFWMF		
	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins	Lena	Camera man	Coins
10%	0.9559	0.9220	0.9342	0.9542	0.9174	0.9312	0.9574	0.9227	0.9348	0.9580	0.9251	0.9373	<b>0.9744</b>	<b>0.9576</b>	<b>0.9634</b>
20%	0.9401	0.8997	0.9225	0.9384	0.8947	0.9180	0.9468	0.9052	0.9270	0.9489	0.9128	0.9307	<b>0.9635</b>	<b>0.9426</b>	<b>0.9558</b>
30%	0.9209	0.8741	0.9072	0.9199	0.8671	0.9046	0.9354	0.8885	0.9174	0.9361	0.8955	0.9202	<b>0.9496</b>	<b>0.9225</b>	<b>0.9463</b>
40%	0.8893	0.8281	0.8695	0.8902	0.8227	0.8716	0.9016	0.8438	0.8815	0.9043	0.8497	0.8892	<b>0.9048-0.9156</b>	<b>0.8508-0.8630</b>	<b>0.8898-0.9022</b>
50%	0.8585	0.7867	0.8396	0.8662	0.7886	0.8515	0.8825	0.8141	0.8658	0.8848	0.8204	0.8706	<b>0.8861-0.8962</b>	<b>0.8221-0.8370</b>	<b>0.8725-0.8875</b>
60%	0.8190	0.7418	0.8046	0.8369	0.7553	0.8280	0.8599	0.7851	0.8442	0.8633	0.7929	0.8509	<b>0.8641-0.8744</b>	<b>0.7940-0.8080</b>	<b>0.8521-0.8670</b>
70%	0.7724	0.6875	0.7659	0.8099	0.7245	0.8039	0.8312	0.7454	0.8187	0.8331	0.7588	0.8225	<b>0.8343-0.8456</b>	<b>0.7604-0.7736</b>	<b>0.8240-0.8427</b>
80%	0.6895	0.6103	0.6829	0.7693	0.6699	0.7614	0.7614	0.6508	0.7151	0.7791	0.6921	0.7727	<b>0.7772-0.7995</b>	<b>0.6901-0.7129</b>	<b>0.7714-0.7958</b>
90%	0.5987	0.5234	0.5981	0.6976	0.6074	0.6939	0.6817	0.5640	0.6353	0.7039	0.6136	0.7018	<b>0.7032-0.7253</b>	<b>0.6116-0.6315</b>	<b>0.7011-0.7169</b>

Table 21 – Computed structural metrics using the checkerboard for comparing the results obtained using different denoising filters to the proposed spatial filter

	Cascading algorithm [50]			IBDNDF [36]			DBUTMWMF [49]			UWMF [39]			SAMFWMF		
	30%	50%	80%	30%	50%	80%	30%	50%	80%	30%	50%	80%	30%	50%	80%
$\beta$	0.9587	0.9270	0.7921	NaN	NaN	NaN	0.9076	0.8470	0.4251	0.9520	0.9202	0.7860	<b>0.9595</b>	<b>0.9295-0.9270</b>	<b>0.8002-0.8076</b>
PSNR	16.9202	14.5194	10.0525	NaN	NaN	NaN	12.7868	10.1671	6.8285	16.2001	13.9772	9.6609	<b>17.0082</b>	<b>14.6175-14.6801</b>	<b>10.0947-10.2579</b>
SSIM	0.7457	0.7027	0.4793	NaN	NaN	NaN	0.1626	0.1427	0.2089	0.7391	0.7467	0.6202	<b>0.8417</b>	<b>0.7915-0.7953</b>	<b>0.6492-0.6599</b>

Table 22 – Correlation ( $\beta$ ) and the PSNR measures, comparing other Impulse denoising filters with and without fixed weighted mean filter as a post-processing step

	Cascading Algorithm [50]				IBDNDF [36]				DBUTMWMF [49]				UWMF [39]			
	Without mean filter		With mean filter		Without mean filter		With mean filter		Without mean filter		With mean filter		Without mean filter		With mean filter	
	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR	$\beta$	PSNR
10%	0.9722	0.9840	26.918	29.406	0.9720	0.9838	26.884	29.364	0.9722	0.9841	26.940	29.452	0.9725	0.9843	26.981	29.431
20%	0.9704	0.9823	26.675	28.971	0.9704	0.9823	26.661	28.989	0.9712	0.9832	26.802	29.222	0.9717	0.9834	26.861	29.232
30%	0.9681	0.9799	26.345	28.432	0.9682	0.9805	26.382	28.634	0.9701	0.9820	26.656	28.745	0.9704	0.9821	26.681	29.018
40%	0.9396	0.9505	23.434	24.376	0.9404	0.9516	23.514	24.493	0.9420	0.9521	23.611	24.495	0.9441	0.9548	23.771	24.526
50%	0.9348	0.9455	23.109	23.968	0.9376	0.9484	23.318	24.222	0.9397	0.9487	23.448	24.398	0.9424	0.9511	23.646	24.326
60%	0.9286	0.9381	22.719	23.423	0.9339	0.9423	23.071	23.965	0.9366	0.9427	23.244	24.162	0.9396	0.9451	23.456	24.387
70%	0.9199	0.9306	22.242	22.930	0.9303	0.9401	22.847	22.543	0.9323	0.9436	22.981	23.843	0.9355	0.9455	23.181	24.005
80%	0.8781	0.9104	20.296	21.855	0.9018	0.9103	21.220	21.675	0.8948	0.9048	21.131	21.643	0.9063	0.9167	21.453	21.934
90%	0.8324	0.8375	18.814	19.049	0.8677	0.8612	19.814	20.065	0.8547	0.8654	19.669	20.088	0.8709	0.8792	19.943	20.339

The results in table 23 show the FOM comparison (with different input parameters) between the proposed edge detection (without SAMFWMF) and the Canny edge detection algorithm. Table 24 shows the FOM comparison (with different input parameters) between proposed edge detection algorithm after SAMFWMF denoising process with Canny edge detection algorithm after the same SAMFWMF denoising process, and the proposed edge detection algorithm after UWMF [39] denoising process with and without fixed weighted mean filter as a post processing step. We selected UWMF [39], because visually it has produced better results among the other filters used in the comparison. In table 24, in order

to evaluate the edge detection after the denoising processes, we insert an extra block in FOM process which injects impulse noise before the denoising process is applied. In tables 23 and 24, lower numbers in this case show improvement on the performance as FOM is monotonically increasing the noise variance and image blurring. The size of the input for FOM is 64×64. A 3×3 first-order derivative kernel is used for the proposed edge detection algorithm. In tables 23 and 24, there is an unexpected tendency in the results of the proposed edge detection which shows decreasing FOM values by increasing noise intensity. The study in [190] indicates that when operators are used (like Sobel) which can cause thickening of edges or missed edges, noise can improve the detector quality by decorrelation of the quantization error. However, the proposed edge detection algorithm (with and without applying SAMFWMF) has resulted in better structural metrics.

Table 23 – FOM comparison between the proposed edge detection and the Canny edge detection algorithm

$\sigma_g$	1		2	
$\sigma_h$	5	15	5	15
$\sigma_p$	0.5	1	0.5	1
$\sigma_n$	1	2	1	2
Canny	1.1627	1.4009	1.1223	1.3539
<b>Proposed Algorithm</b>	<b>0.0607</b>	<b>0.0377</b>	<b>0.0783</b>	<b>0.0237</b>

Table 24 – FOM comparison between proposed edge detection algorithm after proposed spatial filter denoising process with Canny edge detection algorithm after the same denoising process, and the proposed edge detection algorithm after UWMF [39] denoising process with and without fixed weighted mean filter as a post processing step

$\sigma_g$	1				2			
$\sigma_h$	5		15		5		15	
$\sigma_p$	0.5		1		0.5		1	
$I_m$	10%	30%	10%	30%	10%	30%	10%	30%
Canny	6.3127	7.8594	6.4761	7.8751	6.3561	7.9511	7.0012	7.9724
UWMF – With post processing	5.5767	4.4949	5.5123	4.6514	5.4337	4.8386	5.2711	4.2581
UWMF – Without post processing	4.8765	1.8743	4.6754	1.9876	4.1132	1.3241	3.9854	1.2190
<b>Proposed Algorithm</b>	<b>4.3306</b>	<b>0.6555</b>	<b>4.1241</b>	<b>0.3072</b>	<b>3.9849</b>	<b>0.3318</b>	<b>3.6782</b>	<b>0.3058</b>

Table 25 summarizes the acronyms and the corresponding methodologies. Figures 19-21 show a comparison of the denoising filters in the presence of 80% impulse noise on the images of Lena, Camera man, and the Coins. Figures 22-24 show a comparison of the

denoising filters in the presence of 90% impulse noise on the images of Lena, Camera man, and the Coins. The proposed filter with switch 1 is used in both figures. As the results show, SAMFWMF has better structural metrics, and by increasing the initial adaptive median window size (approaching to maximum predefined window size), the similarity is decreased, but the edges become sharper and smoother.

Table 25 – Summary of acronyms and corresponding methodologies

Acronym	Corresponding methodology
Switching Adaptive Median (SAM)	This is a technique for denoising and it switches between two states of adaptive median filter in which adaptive median filter is flexible and adapts itself to the predefined conditions
Fixed Weighted Mean (FWM)	This technique for denoising, calculates the averaging weighted mean of neighboring pixels in which the size of the selected window is fixed
Shrinkage window	This technique is used to improve the denoising in which the size of the window is shrunk according to predefined condition
Gradient based edge detection	This technique is used to detect the image edges in which a kernel obtain based on the gradient of the image and the kernel convolves with the image in order to edge detecting
Non-maximum suppression	This technique is used to track the edges based on the angle of gradient
Maximum sequence	This technique is used to keep the connectivity of the edges and remove the noisy pixels after edge detection
Thresholding	This technique is used to obtain a binary image from grayscale one
Morphological operation	This technique is used for trimming the binary image in order to better visualization

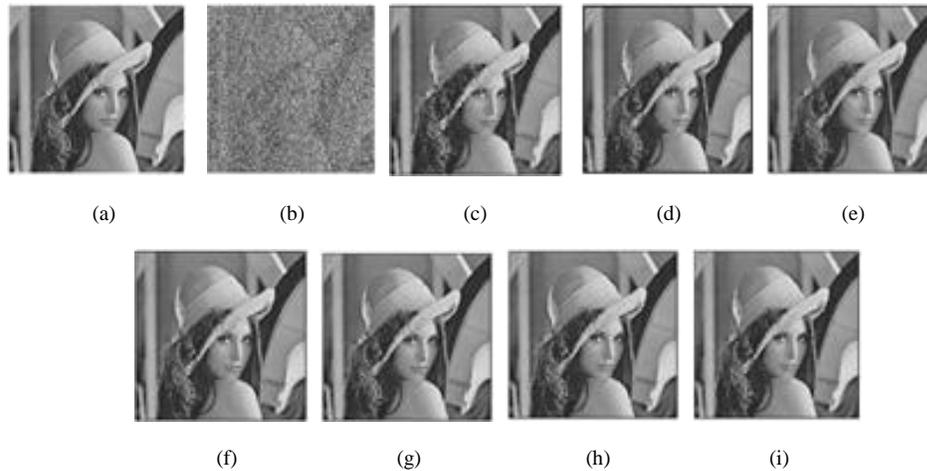


Figure 19 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Lena a) Original image b) Noisy image c) Cascading Algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWMF [39] g) SAMFWMF (initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7)

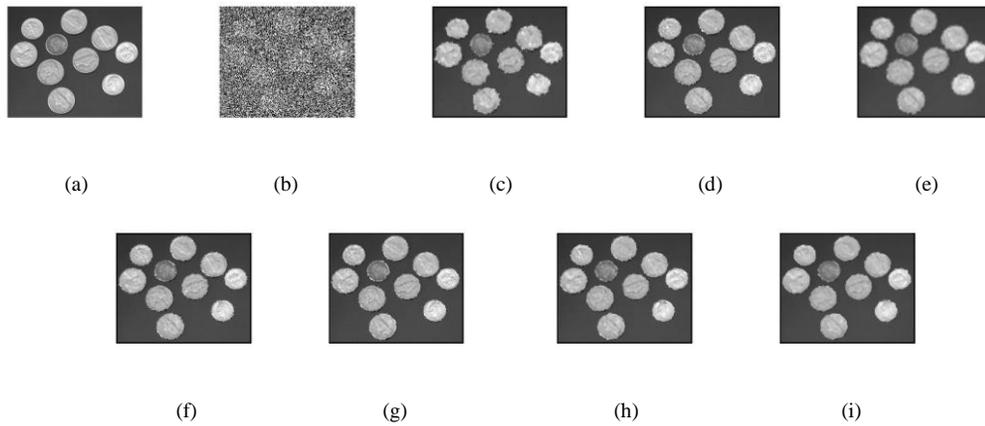


Figure 20 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Coins a) Original image b) Noisy image c) Cascading Algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWMMF [39] g) SAMFWMF (initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7)

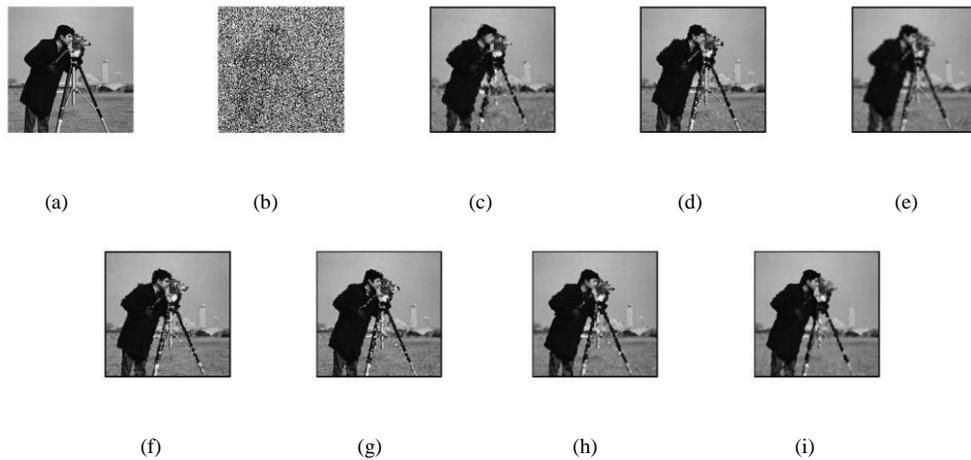


Figure 21 – Comparison of the denoising filters in the presence of 80% impulse noise on the image of Camera man a) Original image b) Noisy image c) Cascading Algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWMMF [39] g) SAMFWMF (initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7)

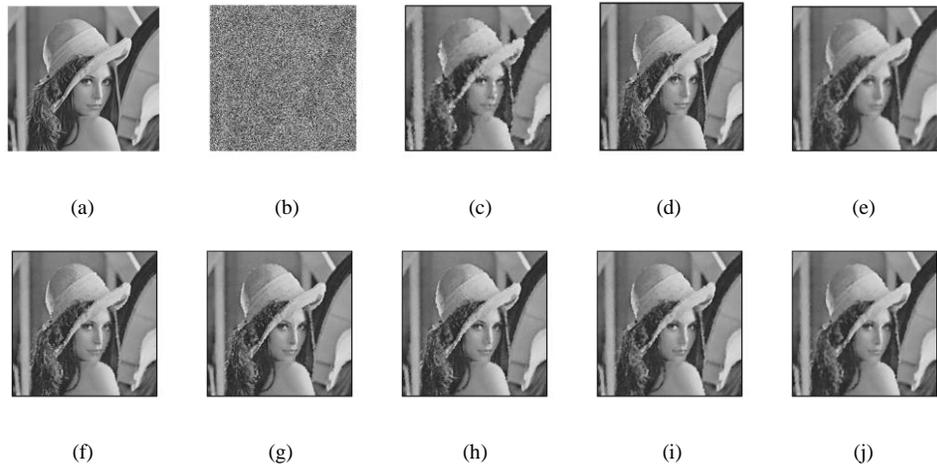


Figure 22 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Lena a) Original image b) Noisy image c) Cascading algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWMMF [39] g) SAMFWMF(initial window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7) j) SAMFWMF (initial adaptive median window size=9)

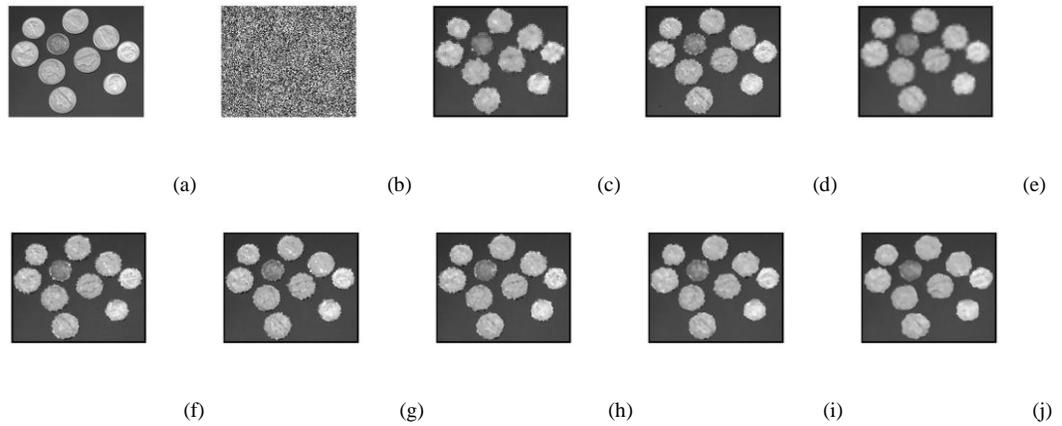


Figure 23 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Coins a) Original image b) Noisy image c) Cascading algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWMMF [39] g) SAMFWMF(initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7) j) SAMFWMF (initial window size=9)

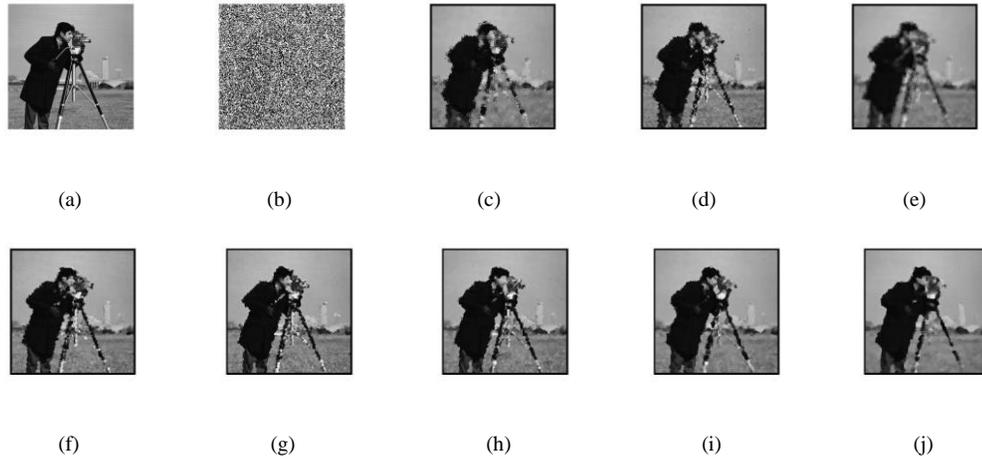


Figure 24 – Comparison of the denoising filters in the presence of 90% impulse noise on the image of Camera man a) Original image b) Noisy image c) Cascading Algorithm [50] d) IBDNDF [36] e) DBUTMWMF [49] f) UWFMF [39] g) PA PA(initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial window size=7) j) SAMFWMF (initial adaptive median window size=9)

#### 4.2.2 Implementation on Magnetic Resonance Imaging

To assess the merits of the proposed method on a different imaging modality, Magnetic Resonance Imaging (MRI) of brain are used for evaluation. For comparative purposes, the results obtained using the proposed method are compared with some of the most effective methods reported in the literature, namely IBDNDF [36], DBUTMWMF [49], UWFMF [39] and Lu`s three-values-weighted filter [234] under different impulse noise intensities. In this comparison, all the parameters chosen, such as initialization and regularization parameters, weights, and window sizes, are set according to their proposed optimal values for the specific noise level.

Tables 26 and 27 show the results obtained on the correlation ( $\beta$ ), and the structural similarity index (SSIM) measures, comparing different filters against the proposed filter (results for the proposed filter are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level). All these metrics are computed

in the presence of 10 to 90 percent impulse noise on images frequently used in the literature for denoising purposes of MRIs.

Table 26 – Correlation ( $\beta$ ) measures for different filters against the proposed spatial filter (results for the proposed filter are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level)

	<i>IBDNDF [36]</i>			<i>DBUTMWMF [49]</i>			<i>UWMF [39]</i>			<i>Lu's three-values-weighted [234]</i>			<i>AMFWMF</i>		
	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>
	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>man</i>	<i>image</i>	<i>image</i>			
10%	0.9720	0.9567	0.9971	0.9722	0.9581	0.9434	0.9725	0.9586	0.9983	0.9714	0.9576	0.9975	<b>0.9843</b>	<b>0.9821</b>	<b>0.9987</b>
20%	0.9704	0.9518	0.9929	0.9712	0.9543	0.8621	0.9717	0.9566	0.9909	0.9687	0.9525	0.9936	<b>0.9834</b>	<b>0.9790</b>	<b>0.9971</b>
30%	0.9682	0.9464	0.9801	0.9701	0.9508	0.7597	0.9704	0.9533	0.9495	0.9659	0.9480	0.9820	<b>0.9819</b>	<b>0.9753</b>	<b>0.9931</b>
40%	0.9404	0.9031	0.9422	0.9420	0.9080	0.7420	0.9441	0.9128	0.9013	0.9383	0.9079	0.9501	<b>0.9528-0.9548</b>	<b>0.9301-0.9345</b>	<b>0.9577-0.9627</b>
50%	0.9376	0.8950	0.9300	0.9397	0.9011	0.6110	0.9424	0.9085	0.8777	0.9359	0.9023	0.9412	<b>0.9508-0.9524</b>	<b>0.9231-0.9282</b>	<b>0.9501-0.9541</b>
60%	0.9339	0.8865	0.9187	0.9366	0.8951	0.4811	0.9396	0.9021	0.8477	0.9330	0.8960	0.9237	<b>0.9478-0.9499</b>	<b>0.9165-0.9219</b>	<b>0.9402-0.9439</b>
70%	0.9303	0.8799	0.8894	0.9323	0.8852	0.3257	0.9355	0.8927	0.6381	0.9301	0.8881	0.9097	<b>0.9432-0.9459</b>	<b>0.9083-0.9138</b>	<b>0.9281-0.9323</b>
80%	0.9018	0.8340	0.8327	0.8948	0.8295	0.2011	0.9063	0.8479	0.5914	0.9024	0.8430	0.8621	<b>0.9106-0.9160</b>	<b>0.8571-0.8663</b>	<b>0.8771-0.8851</b>
90%	0.8677	0.7859	0.7991	0.8547	0.7726	0.0859	0.8709	0.7956	0.3539	0.8686	0.7895	0.8315	<b>0.8725-0.8800</b>	<b>0.8011-0.8122</b>	<b>0.8401-0.8506</b>

Table 27 – Structural similarity index (SSIM) measures for different filters against the proposed spatial filter (results for the proposed filter are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level)

	<i>IBDNDF [36]</i>			<i>DBUTMWMF [49]</i>			<i>UWMF [39]</i>			<i>Lu's three-values-weighted [234]</i>			<i>AMFWMF</i>		
	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>	<i>Lena</i>	<i>Camera</i>	<i>MRI</i>
	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>image</i>	<i>man</i>	<i>man</i>	<i>image</i>	<i>image</i>			
10%	0.9542	0.9174	0.9807	0.9574	0.9227	0.4825	0.9580	0.9251	0.9818	0.9377	0.9003	0.9811	<b>0.9744</b>	<b>0.9576</b>	<b>0.9877</b>
20%	0.9384	0.8947	0.9681	0.9468	0.9052	0.4128	0.9489	0.9128	0.9606	0.8944	0.8529	0.9713	<b>0.9635</b>	<b>0.9426</b>	<b>0.9813</b>
30%	0.9199	0.8671	0.9535	0.9354	0.8885	0.3982	0.9361	0.8955	0.8994	0.8565	0.7923	0.9621	<b>0.9496</b>	<b>0.9225</b>	<b>0.9724</b>
40%	0.8902	0.8227	0.9098	0.9016	0.8438	0.3865	0.9043	0.8497	0.8646	0.8168	0.7423	0.9227	<b>0.9048-0.9156</b>	<b>0.8508-0.8630</b>	<b>0.9353-0.9424</b>
50%	0.8662	0.7886	0.8871	0.8825	0.8141	0.3702	0.8848	0.8204	0.8186	0.7894	0.7127	0.9017	<b>0.8861-0.8962</b>	<b>0.8221-0.8370</b>	<b>0.9181-0.9264</b>
60%	0.8369	0.7553	0.8525	0.8599	0.7851	0.3544	0.8633	0.7929	0.7693	0.7625	0.6813	0.8702	<b>0.8641-0.8744</b>	<b>0.7940-0.8080</b>	<b>0.8916-0.9001</b>
70%	0.8099	0.7245	0.8195	0.8312	0.7454	0.3317	0.8331	0.7588	0.5689	0.7437	0.6537	0.8441	<b>0.8343-0.8456</b>	<b>0.7604-0.7736</b>	<b>0.8627-0.8712</b>
80%	0.7693	0.6699	0.7715	0.7614	0.6508	0.2757	0.7791	0.6921	0.5775	0.7244	0.6191	0.8031	<b>0.7772-0.7995</b>	<b>0.6901-0.7129</b>	<b>0.8107-0.8261</b>
90%	0.6976	0.6074	0.7016	0.6817	0.5640	0.2161	0.7039	0.6136	0.4424	0.6806	0.5676	0.7543	<b>0.7032-0.7253</b>	<b>0.6116-0.6315</b>	<b>0.7656-0.7873</b>

Figures 25 shows, the edge boundaries and similarity of different natural and MRI images after applying the proposed filter in the presence of high intensity noise. Figures 26 and 27 show the same comparison in the presence of 80% and 90% impulse noise, respectively, on MRI images. Figure 28 show the results obtained from the proposed method for 20%, 40%, 60%, 80% and 90% impulse noise on different MRIs. As these figures show, the proposed algorithm has good performance in terms of keeping relevant detail while obtaining the highest similarity, least noise, and the preserving of edges, especially in high impulse noise environment.

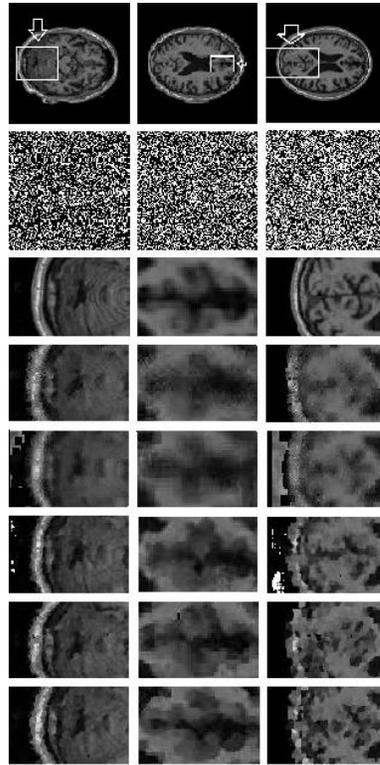


Figure 25 – Edge boundaries and similarity of different MRI images after applying the proposed filter in the presence of high intensity noise

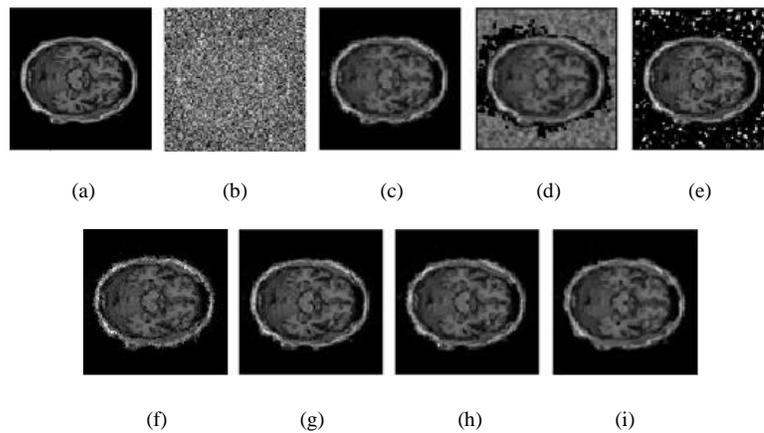


Figure 26 – Comparison in the presence of 80% impulse noise intensity on MRI images a) Original image b) Noisy image c) IBDNDF [36] d) DBUTMWMF [49] e) UWMF [39] f) Lu's three-values-weighted [234] g) AMFWMF (initial window size=3) h) AMFWMF (initial adaptive median window size=5) i) AMFWMF (initial adaptive median window size=7)

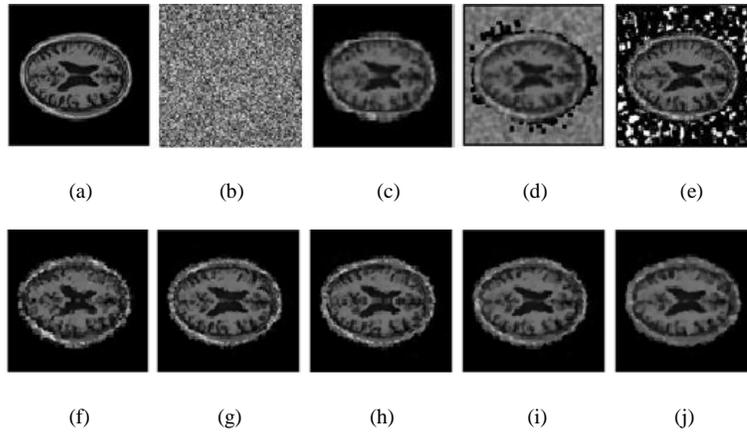


Figure 27 – Comparison in the presence of 90% impulse noise intensity on MRI images a) Original image b) Noisy image c) IBDNDF [36] d) DBUTMWMF [49] e) UWMF [39] f) Lu's three-values-weighted [234] g) AMFWMF (initial window size=3) h) AMFWMF (initial adaptive median window size=5) i) AMFWMF (initial adaptive median window size=7) j) AMFWMF (initial adaptive median window size=9)

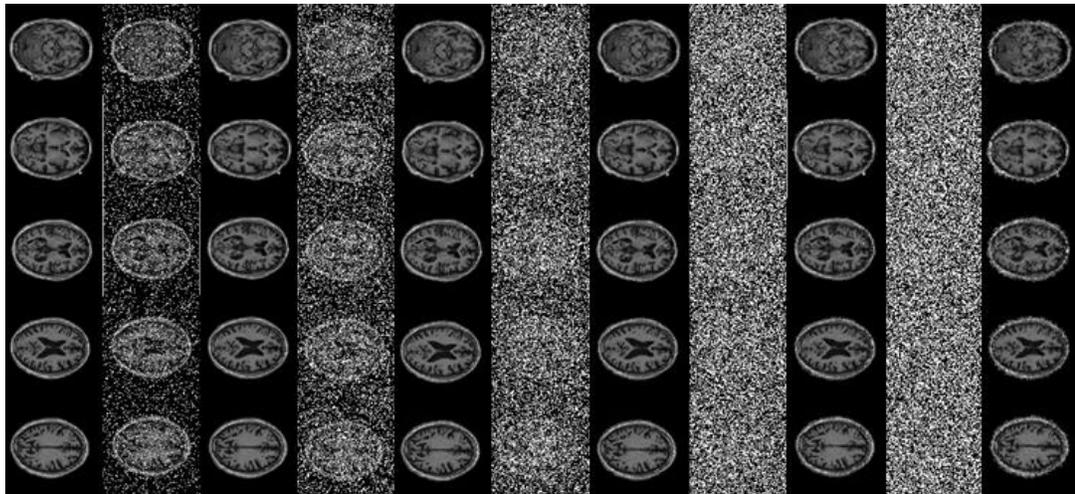


Figure 28 – Results obtained from the proposed method for different impulse noise levels on different MRI images. The 1<sup>th</sup> column is original MRI images, even columns (2<sup>nd</sup> through 10<sup>th</sup>) are respectively the original MRI with 20%, 40%, 60%, 80% and 90% impulse noise, odd columns (3<sup>rd</sup> through 11<sup>th</sup>) show the denoising results of their previous columns.

#### 4.2.3 Results After Edge Detection

To evaluate the performance of the proposed edge detection step after SAMFWMF process with switch 1, the results obtained are compared with other relevant denoising

filters. Also, the performance of the SAMFWMF with switch 1 after applying the Canny edge detector, which is one of the most powerful and most reliable edge detectors [190] [191], is evaluated. Figures 29 and 30 show these comparisons on the camera man in the presence of respectively, 80% and 90% impulse noise intensities. Figures 31 and 32 show these comparisons on the Coins in the presence of respectively, 80% and 90% impulse noise intensities. Figures 33 and 34 show these comparisons on the Lena in the presence of respectively, 80% and 90% impulse noise intensities.

For all these figures, the initial adaptive median window size for SAMFWMF is equal to the maximum predefined window size. In both figures, part (c) shows the results when applying the Canny edge detection step after SAMFWMF process, part (d) shows the results when applying the proposed edge detection step after the cascading algorithm [50] process, part (e) shows the results when applying the proposed edge detection step after the IBDNDF [36] process, part (f) shows the results when applying the proposed edge detection step after DBUTMWMF [49] process, part (g) shows the results when applying the proposed edge detection step after the UWMF [39] process, and part (h) shows the result when applying the proposed edge detection step after the SAMFWMF process. Figure 35 shows the results when applying the SAMFWMF (using switch 2) and the proposed edge detection algorithm on the checkerboard image. Figure 36 shows the proposed filter (using switch 1) with 95% impulse noise after and before edge detection.

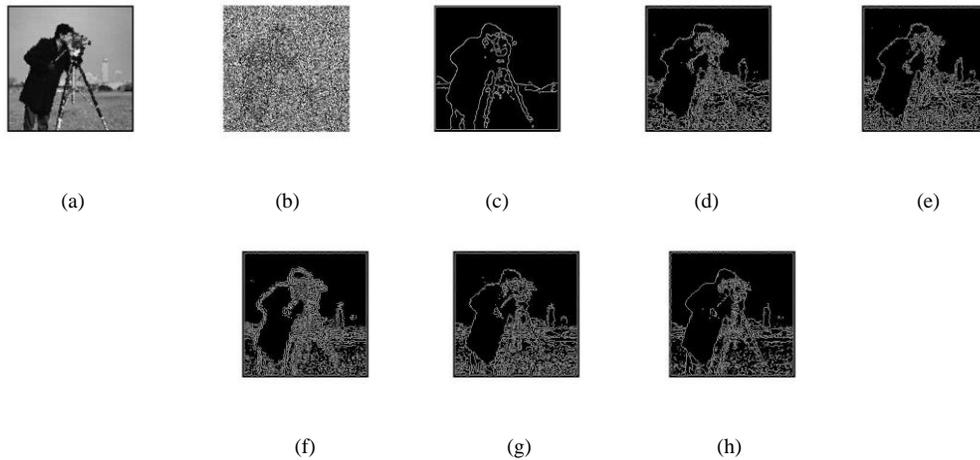


Figure 29 – Comparison on the camera man in the presence of 80% impulse noise intensity,  $T=0.2$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

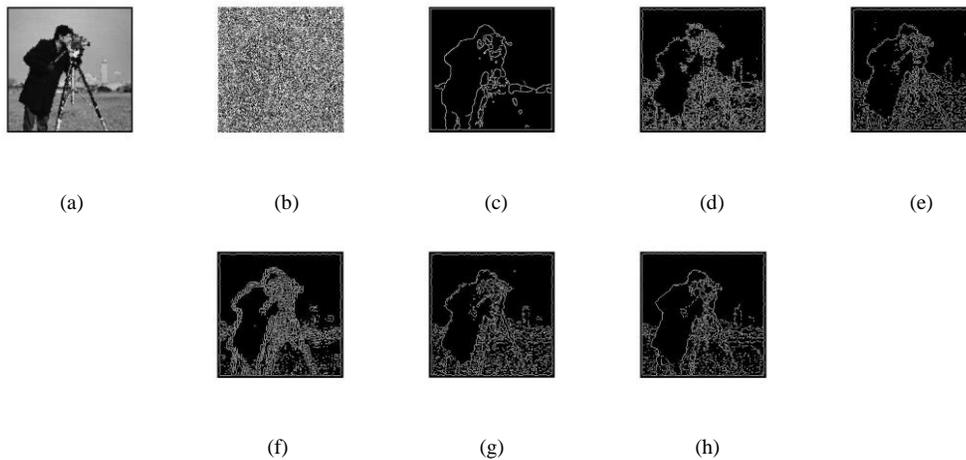


Figure 30 – Comparison on the camera man in the presence of 90% impulse noise intensity,  $T=0.2$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

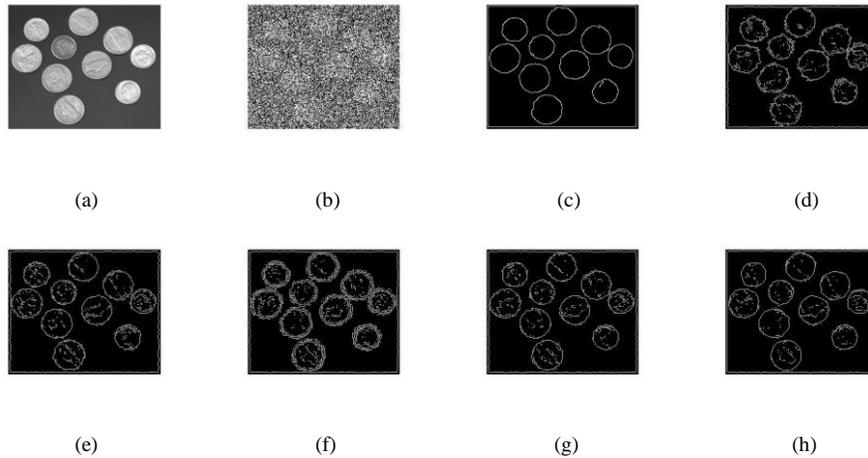


Figure 31 – Comparison on the Coins in the presence of 80% impulse noise intensity,  $T=0.3$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

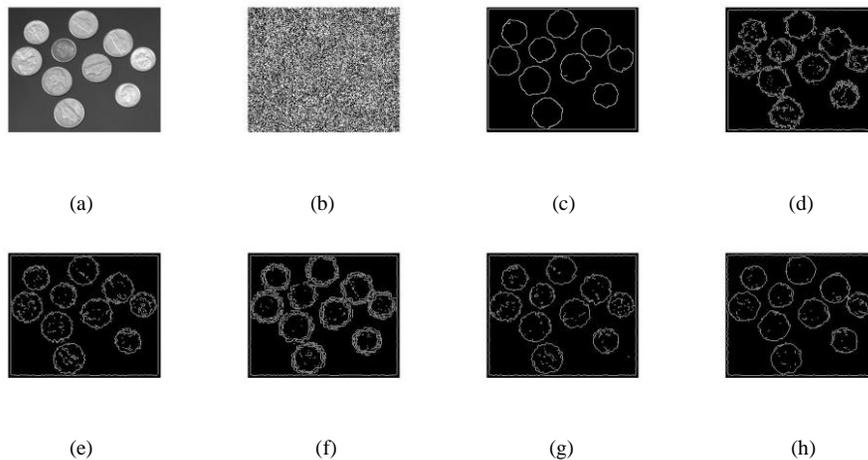


Figure 32 – Comparison on the Coins in the presence of 90% impulse noise intensity,  $T=0.3$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

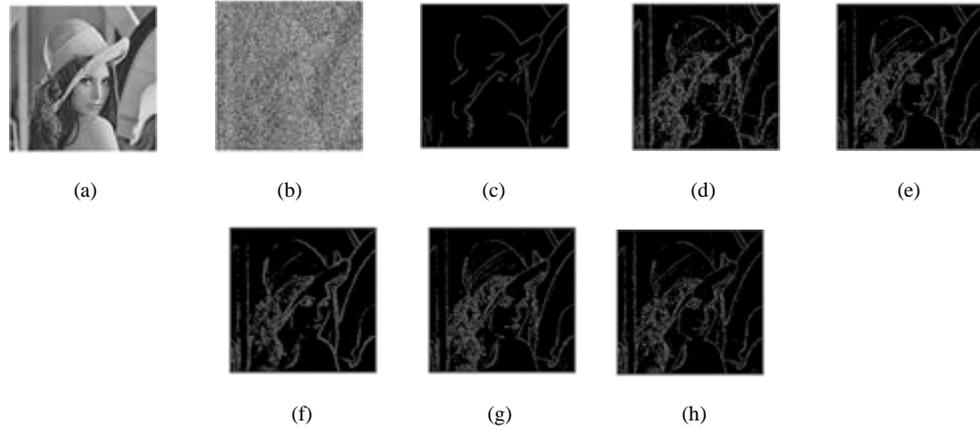


Figure 33 – Comparison on the Lena in the presence of 80% impulse noise intensity,  $T=0.3$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

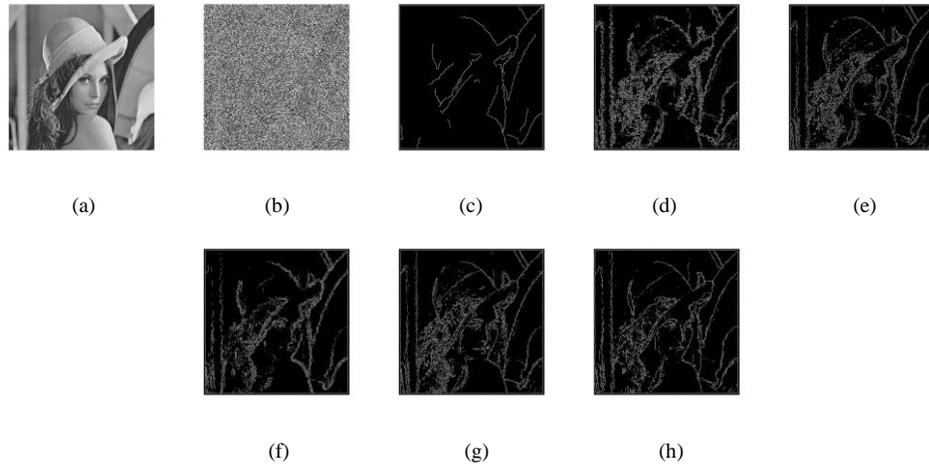


Figure 34 – Comparison on the Lena in the presence of 90% impulse noise intensity,  $T=0.3$  (normalized) a) Original image b) Noisy image c) Edge detection with canny (with  $\sigma=1$ , by increasing the  $\sigma$ , more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [50] process e) Proposed edge detection algorithm after IBDNDF [36] process f) Proposed edge detection algorithm after DBUTMWMF [49] process g) Proposed edge detection algorithm after UWMF [39] process h) Proposed edge detection algorithm after SAMFWMF process

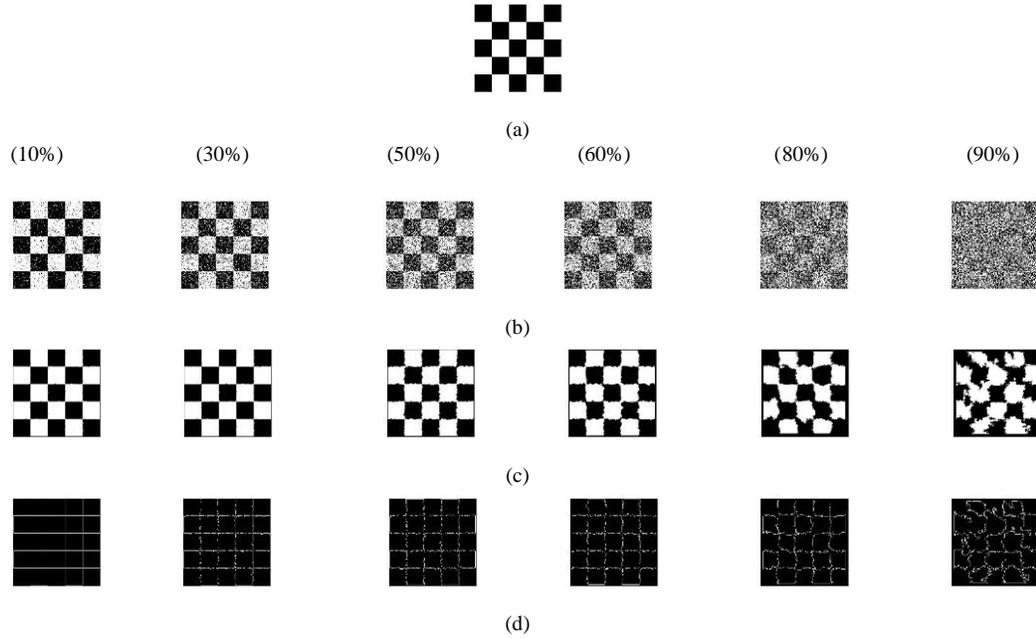


Figure 35 – Application of the proposed spatial filter (using switch 2) and the proposed edge detection algorithm on the checkerboard image a) Original image b) Noisy images c) After denoising d) After edge detection

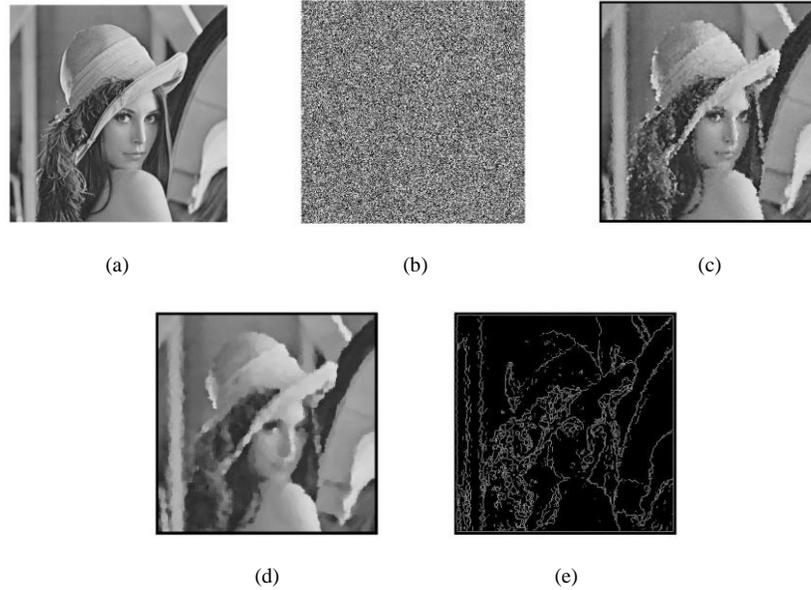


Figure 36 – Application of the proposed spatial filter (using switch 1) and the proposed edge detection algorithm on the Lena image in presence of 95% Impulse noise intensity a) Original image b) Noisy image c) after apply the filter with 95% impulse noise, initial window size of the adaptive median filter is 3 d) after apply the filter with 95% impulse noise, initial window size of the adaptive median filter is 13 e) Edge detection after apply the filter with 95% impulse noise, initial window size of the adaptive median filter is 13

As the results clearly demonstrate, the proposed algorithm has a better performance amongst all other methods in terms of keeping relevant details and for attaining the highest similarity, least noise, preservation of edges, and better edge tracking, especially in the presence of impulse noise even under high-intensity levels.

### *4.3 Gaussian and Combined Gaussian - Speckle Denoising Based on Wavelet Filter*

#### *4.3.1 Combined Gaussian and Speckle Denoising*

In this section, the results obtained using the proposed method after applying the steps of denoising are presented. These results are compared with some well-known despeckling filters reported in the literature, namely the Frost filter [205], Kuan filter [206], Lee-diffusion filter [207], Lee filter [208, 209], Geometry filter [210] and improved total variation filter [59] applied on different images and under different noise intensity levels.

In this comparison, images of Lena , Camera man and medical ultrasound images are standard examples used in the literature for quantitative and visual comparative purposes. Tables 28-30 show respectively the results obtained on the peak signal to noise ratio (PSNR), correlation ( $\beta$ ) and the feature similarity index (FSIM) measures, comparing different denoising filters against the proposed method. All these metrics/measures are computed in the presence of speckle noise with standard deviation 0.1, 0.2, 0.3 and Gaussian noise with standard deviation 0.05, 0.1, 0.2, 0.3. In Tables 28-30, higher numbers mean better results, indicating that the proposed method produced better structural metrics.

The results in Table 31 shows the FOM comparison after the denoising process, expressed as an extra block in the FOM. We selected Lee-diffusion [207], because it has shown better results among all other filters used in the comparison. In Table 31, lower numbers in this case show improvement on the performance as FOM is monotonically increasing with

noise variance and image blurring. The size of input for FOM is 64×64. The Sobel kernel is used for edge detection. However, the proposed method has resulted in better structural metrics.

Figure 37 shows the denoising filters comparison in the presence of speckle ( $\sigma = 0.1$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image. As the figure shows, the proposed method has better performance among all of others in terms of keeping relevant details and also in terms of obtaining the highest similarity and the least noise as well as better edge tracking, especially in high combined speckle and Gaussian noise intensities.

Table 28 – Correlation ( $\beta$ ) measure, comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined speckle and Gaussian noise intensities

Speckle / Gaussian Noise levels	Geometry filter [210]			Frost filter [205]			Lee filter [208,209]			Lee-diffusion filter [207]			Proposed Algorithm		
	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras
	0.1, 0.1	0.6156	0.7297	0.6359	0.9004	0.9134	0.8725	0.8219	0.8662	0.8559	0.9203	0.8942	0.9565	<b>0.9501</b>	<b>0.9312</b>
0.2, 0.1	0.5993	0.7261	0.6420	0.8727	0.8969	0.8840	0.7856	0.8462	0.8526	0.9025	0.8831	0.9249	<b>0.9422</b>	<b>0.9275</b>	<b>0.9411</b>
0.2, 0.05	0.7136	0.8036	0.7869	0.9127	0.9253	0.9349	0.8425	0.8855	0.9192	0.9265	0.9034	0.9693	<b>0.9474</b>	<b>0.9307</b>	<b>0.9731</b>

Table 29 – PSNR measure, comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined speckle and Gaussian noise intensities

Speckle / Gaussian noise levels	Geometry filter [210]			Frost filter [205]			Lee filter [208,209]			Lee-diffusion filter [207]			Proposed Algorithm		
	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras
	0.1, 0.1	8.5464	8.5193	7.4461	21.6528	19.6096	18.7944	19.2698	18.2299	18.5562	22.5657	19.0907	22.0761	<b>22.3043</b>	<b>19.2951</b>
0.2, 0.1	8.2559	8.3458	7.4086	20.6083	18.9099	19.3681	18.5428	17.6942	18.7525	21.9825	18.7237	21.2821	<b>21.4054</b>	<b>18.7749</b>	<b>19.8631</b>
0.2, 0.05	9.4056	9.4015	9.9133	22.0613	20.3137	21.0742	19.7286	18.8843	20.6442	22.9332	19.5416	23.7871	<b>22.4349</b>	<b>19.7529</b>	<b>21.8145</b>

Table 30 – Feature similarity index (FSIM), comparing other denoising filters against proposed wavelet-based denoising filter in presence of different combined speckle and Gaussian noise intensities

Speckle / Gaussian noise levels	Geometry filter [210]			Frost filter [205]			Lee filter [208,209]			Lee-diffusion filter [207]			Proposed Algorithm		
	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras	Lena	Cman	Ultras
	0.1, 0.1	0.9036	0.9277	0.9099	0.9437	0.9545	0.9398	0.9264	0.9377	0.9362	0.9711	0.9348	0.9639	<b>0.9714</b>	<b>0.9419</b>
0.2, 0.1	0.8992	0.9254	0.9087	0.9336	0.9477	0.9387	0.9155	0.9319	0.9356	0.9700	0.9316	0.9633	<b>0.9702</b>	<b>0.9439</b>	<b>0.9638</b>
0.2, 0.05	0.9235	0.9380	0.9348	0.9487	0.9569	0.9573	0.9265	0.9412	0.9547	0.9719	0.9389	0.9745	<b>0.9728</b>	<b>0.9465</b>	<b>0.9678</b>

Table 31 – FOM comparison between proposed wavelet-based filter denoising process with gradient-based edge detection process, and Lee-diffusion [39] with the same edge detection process

$\sigma_g$	1		2	
$\sigma_h$	3	10	3	10
$\sigma_p$	0.5	1	0.5	1
Gaussian	0.5	1	0.5	1
Lee-diffusion [207]	2.9446	2.4765	2.5985	2.4164
<b>Proposed Algorithm</b>	<b>0.1659</b>	<b>0.0206</b>	<b>0.1241</b>	<b>0.0122</b>

Figure 38 shows the denoising filters comparison in the presence of speckle noise ( $\sigma = 0.2$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image. As the figure shows, the proposed method has better performance among all of the other methods, again in terms of keeping relevant details, producing the highest similarity and the least noise as well as showing better edge tracking, especially in high combined speckle and Gaussian noise intensities. Figure 39 shows the results obtained from the proposed method in the presence of different combined speckle and Gaussian noise intensities on a medical ultrasound image.

Although the compared filters have shown very high performance in the presence of speckle noise alone, as these figures illustrate, their performance is weakened in terms of keeping the relevant details, obtaining the highest similarity, least noise, and preserving edges, especially in high intensity levels when there is a combined presence of speckle and Gaussian noise.

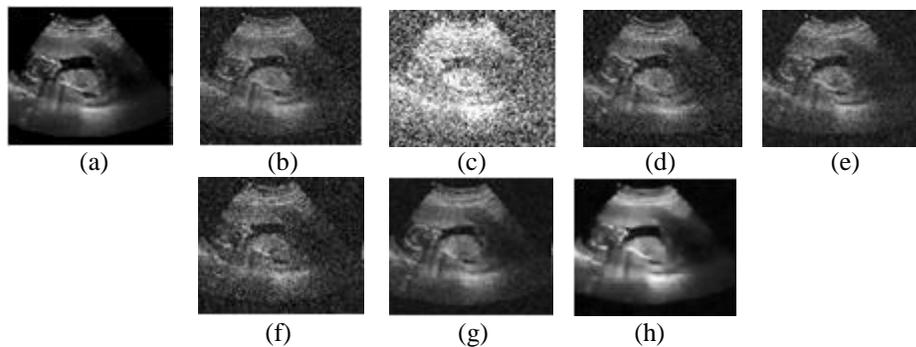


Figure 37 – Comparison of the denoising filters in the presence of speckle ( $\sigma = 0.1$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image a) Original image b) Noisy image c) Geometry filter d) Frost filter e) Kuan filter f) Lee filter g) Lee-diffusion filter h) Proposed filter

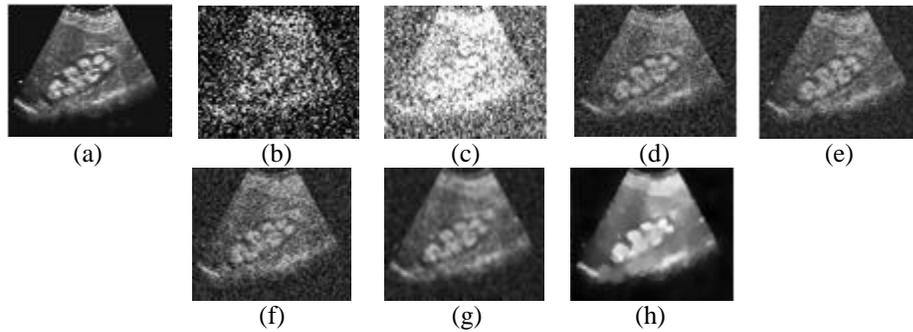


Figure 38 – Comparison of the denoising filters in the presence of speckle ( $\sigma = 0.2$ ) and Gaussian noise ( $\sigma = 0.1$ ) on a medical ultrasound image a) Original image b) Noisy image c) Geometry filter d) Frost filter e) Kuan filter f) Lee filter g) Lee-diffusion filter h) Proposed filter

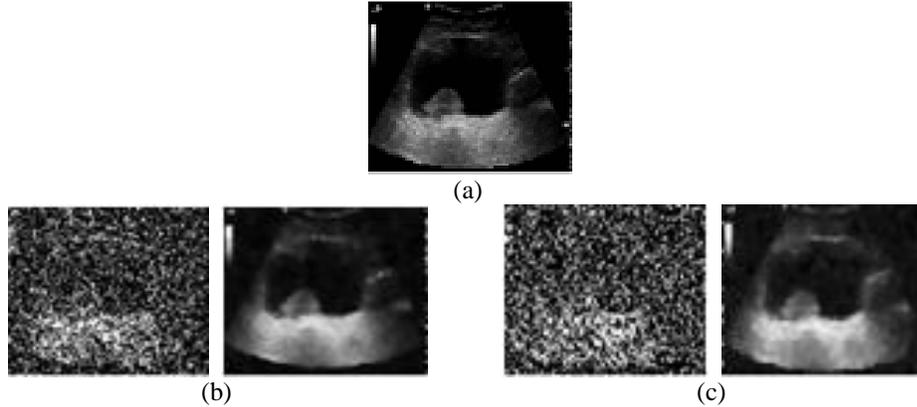


Figure 39 – Application of the proposed wavelet-based filter in the presence of combined Speckle and Gaussian noise intensities on a medical ultrasound image, a) original image b) Denoising in the presence of Noisy image with Speckle ( $\sigma = 0.2$ ) and Gaussian ( $\sigma = 0.2$ ) c) Denoising in the presence of Speckle ( $\sigma = 0.3$ ) and Gaussian ( $\sigma = 0.3$ )

#### 4.3.2 Gaussian Denoising

To evaluate the performance of the proposed filter in presence of Gaussian noise alone, the visual results are provided on different natural images. Figure 40 shows the results obtained from the proposed method in the presence of different Gaussian noise intensities on different natural images.



Figure 40 – Application of the proposed wavelet-based filter in the presence of different Gaussian noise intensities on different natural images a) column 1 through 3 are: original Lena image, noisy image corrupted with Gaussian noise ( $\sigma = 0.1$ ) b) a) column 1 through 3 are: original Man image, noisy image corrupted with Gaussian noise ( $\sigma = 0.15$ ) c) a) column 1 through 3 are: original Boat image, noisy image corrupted with Gaussian noise ( $\sigma = 0.2$ ) d) a) column 1 through 3 are: original Peppers image, noisy image corrupted with Gaussian noise ( $\sigma = 0.3$ )

#### 4.3.3 Results after Edge Detection

To evaluate the performance of the proposed filter after the edge detection step, the visual results are provided on different ultrasound and natural images. Figures 41 shows the results after edge detection in the presence of speckle noise ( $\sigma = 0.2$ ) and Gaussian noise

( $\sigma = 0.1$ ) on a medical ultrasound image. Figures 42 shows respectively results after edge detection in the presence of Gaussian noise ( $\sigma = 0.2$ ) and ( $\sigma = 0.3$ ) on Lena and Peppers images.

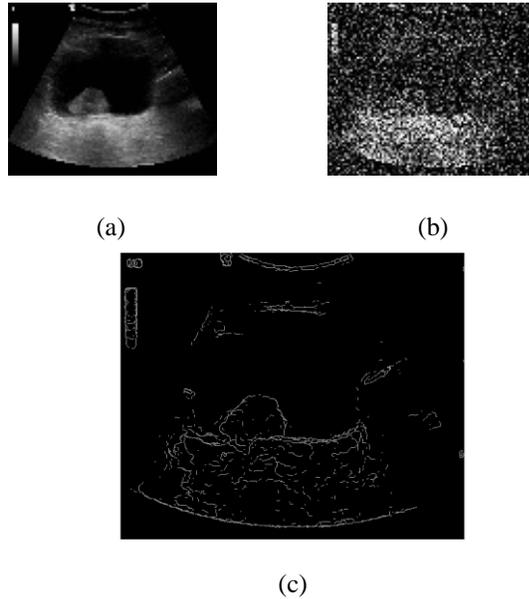


Figure 41 – Edge detection after applying the proposed wavelet-based filter in the presence of combined speckle ( $\sigma = 0.2$ ) and Gaussian ( $\sigma = 0.1$ ) noise on a medical ultrasound image,  $T=0.02$  (normalized)

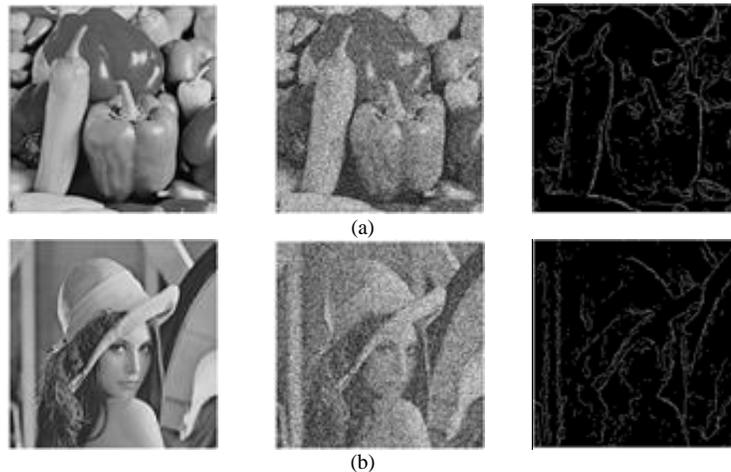


Figure 42 – Edge detection after applying the proposed wavelet-based filter in the presence of Gaussian noise a) Column 1 through 3 are: original peppers image, noisy image corrupted with Gaussian noise ( $\sigma = 0.2$ ), after edge detection ( $T=0.02$  (normalized)) b) Column 1 through 3 are: original peppers image, noisy image corrupted with Gaussian noise ( $\sigma = 0.3$ ), after edge detection ( $T=0.02$  (normalized))

#### 4.4 CNN Filtering

##### 4.4.1 Mixed Impulse and Gaussian Denoising

We have used 400 180×180 pixels images from the Berkeley segmentation dataset (BSD) to train the described network for both specific and non-specific noise-level removal, similar to studies reported in [87, 89, 171]. While for testing, we have used BSD100 (as in [171]) and the additional 12 images that [171] uses and are shown in Figure 43.



Figure 43 – 12 Test Images

As previously stated, the optimal results were obtained by using a 20-layer network with 40×40 patches for both specific and non-specific noise-level denoising. We have used stochastic gradient descent (SGD)-momentum [228] with an initial learning rate of 0.1 (which decreased over progressive epochs), weight decay of 0.0001, momentum of 0.9 and

mini-batches of size 128. The type of noise mixture affected the numbers of epochs the model needed for training.

We carried out all implementations in MATLAB 2017b using the MatConvNet package [89, 69] for convolutional neural networks on a PC with Nvidia Quadro M6000 GPU. The time required to train the network varied between 24 and 48 hours depending on the noise mixture (as different mixtures required different number of epochs).

Once the network was trained, the results obtained from the proposed denoising method are compared to the results obtained from the WESNR [164] method and the LSM-NLR [171] method on the different images and under the same mixed impulse and Gaussian noise intensities.

Tables 32 and 33 show the results of specific and non-specific noise-level removal for mixed Gaussian and impulse noise. They respectively show the results obtained from the average peak signal to noise ratio (PSNR) and the average feature similarity index (FSIM) [215] metric from the 12 test images shown in Figure 3 after the denoising process. Gaussian noise with standard deviation of 10, 20, 30, and 50, and 10, 20, 30, 40, and 50 percent salt and pepper impulse noise were introduced. Specific noise-level denoising required 125 epochs while non-specific noise-level denoising required 150; further increments of the number of training epochs yielded better results in both cases. The network was run 50 times for each noise level over the testing set and the means and standard deviations of the results were calculated.

Tables 34 and 35 show the results obtained for the average peak signal to noise ratio (PSNR) and the average feature similarity index (FSIM) measures on the BSD100 data set

images. The collection method, number of epochs, and noise levels introduced are equal to those of Tables 32 and 33.

Table 32 – Average PSNR comparison for different mixed Impulse and Gaussian denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	50%	28.95	29.75	29.89	30.60	<b>31.0823±0.0095</b>	<b>31.4845±0.0127</b>
$\sigma = 20$	50%	26.73	27.50	27.75	28.51	<b>28.6711±0.0096</b>	<b>28.9127±0.0112</b>
$\sigma = 30$	20%	26.80	26.59	27.09	28.31	<b>28.3351±0.0067</b>	<b>28.5869±0.0836</b>
	50%	24.52	25.85	26.13	26.70	<b>26.9530±0.1371</b>	<b>27.1718±0.0138</b>
$\sigma = 50$	10%	20.80	24.44	24.83	26.00	<b>26.1794±0.0063</b>	<b>26.4956±0.0063</b>
	50%	14.43	23.35	23.56	24.36	<b>24.5109±0.0136</b>	<b>24.8086±0.0116</b>

Table 33 – Average feature similarity index (FSIM) comparison for different mixed Impulse and Gaussian denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	50%	95.29	96.49	96.39	96.63	<b>0.9677±1.1773×10<sup>-4</sup></b>	<b>0.9703±1.0412×10<sup>-4</sup></b>
$\sigma = 20$	50%	91.99	93.32	93.38	93.76	<b>0.9409±1.8120×10<sup>-4</sup></b>	<b>0.9430±2.1311×10<sup>-4</sup></b>
$\sigma = 30$	20%	91.55	93.02	93.12	93.21	<b>0.9346±0.0014</b>	<b>0.9367±1.6123×10<sup>-4</sup></b>
	50%	88.80	90.02	90.04	91.06	<b>0.9133±3.1004×10<sup>-4</sup></b>	<b>0.9154±4.0265×10<sup>-4</sup></b>
$\sigma = 50$	10%	82.29	89.31	89.36	90.04	<b>0.9000±2.0388×10<sup>-4</sup></b>	<b>0.9005±2.1699×10<sup>-4</sup></b>
	50%	66.08	83.69	83.50	85.44	<b>0.8637±4.0668×10<sup>-4</sup></b>	<b>0.8661±4.4334×10<sup>-4</sup></b>

Table 34 – Average PSNR comparison for different mixed Gaussian and salt and pepper Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	50%	26.62	27.54	27.36	28.17	<b>29.0404±0.0046</b>	<b>29.4035±0.0055</b>
$\sigma = 20$	50%	24.81	25.86	26.12	26.88	<b>27.1435±0.0049</b>	<b>27.3621±0.0053</b>
$\sigma = 30$	20%	24.94	24.61	25.27	26.94	<b>27.0714±0.0031</b>	<b>27.2842±0.0021</b>
	50%	22.92	24.43	24.81	25.44	<b>25.7770±0.0060</b>	<b>25.9552±0.0044</b>
$\sigma = 50$	10%	19.82	22.13	22.80	24.05	<b>25.2762±0.0031</b>	<b>25.5064±0.0033</b>
	50%	14.44	22.22	22.66	23.96	<b>23.9800±0.0058</b>	<b>24.0689±0.0048</b>

Table 35 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and salt and pepper denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	50%	86.45	89.96	90.06	89.87	<b>0.8979±1.2083×10<sup>-4</sup></b>	<b>0.9085±9.5743×10<sup>-5</sup></b>
$\sigma = 20$	50%	80.61	83.25	83.79	83.83	<b>0.8528±2.1385×10<sup>-4</sup></b>	<b>0.8638±1.4583×10<sup>-4</sup></b>
$\sigma = 30$	20%	80.05	82.79	83.00	83.00	<b>0.8500±1.4967×10<sup>-4</sup></b>	<b>0.8551±1.0677×10<sup>-4</sup></b>
	50%	78.45	80.43	80.66	80.75	<b>0.8136±0.200</b>	<b>0.8170±1.8009×10<sup>-4</sup></b>
$\sigma = 50$	10%	73.45	76.71	78.05	80.51	<b>0.8060±1.3329×10<sup>-4</sup></b>	<b>0.8083±1.8824×10<sup>-4</sup></b>
	50%	63.90	75.22	75.60	75.66	<b>0.7627±3.1885×10<sup>-4</sup></b>	<b>0.7578±2.5994×10<sup>-4</sup></b>

Tables 36 and 37 show the average peak signal to noise ratio (PSNR) and the average feature similarity index (FSIM) of 12 test images in Figure 43 for non-specific noise-level denoising of mixed Gaussian and random value impulse noise. The collection method, number of epochs, and noise levels injected are equal to those of Tables 32 and 33.

Table 36 – Average PSNR comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)
$\sigma = 10$	10%	30.24	31.25	31.36	32.30	<b>33.0696±0.0049</b>
	20%	29.36	29.46	29.86	30.82	<b>31.2961±0.0100</b>
	30%	28.40	27.74	28.55	29.37	<b>30.3377±0.0116</b>
	40%	27.02	26.74	26.92	27.24	<b>29.4561±0.0080</b>
	50%	25.30	24.72	25.18	25.36	<b>28.6942±0.0423</b>
$\sigma = 20$	10%	27.69	28.62	28.90	29.22	<b>29.8603±0.3333</b>
	20%	27.09	27.50	27.78	28.27	<b>29.8453±0.0082</b>
	30%	26.42	26.21	26.74	27.28	<b>28.9769±0.0097</b>
	40%	25.24	24.83	25.37	26.08	<b>28.4833±0.0084</b>
	50%	23.86	23.19	24.00	24.62	<b>27.9477±0.0236</b>
$\sigma = 30$	10%	26.11	26.56	26.95	26.98	<b>27.3427±0.0097</b>
	20%	25.55	25.49	25.92	26.20	<b>28.2655±0.0115</b>
	30%	24.92	24.30	25.00	25.39	<b>27.8864±0.0104</b>
	40%	23.74	23.11	23.74	24.33	<b>27.4426±0.0100</b>
	50%	22.30	21.62	22.52	23.18	<b>27.0158±0.0112</b>
$\sigma = 50$	10%	23.16	23.83	24.14	24.27	<b>26.5025±0.0031</b>
	20%	22.59	22.73	23.20	23.61	<b>26.5615±0.0055</b>
	30%	21.84	21.59	22.31	22.85	<b>26.3021±0.0084</b>
	40%	20.78	20.73	21.14	21.95	<b>25.6629±0.0086</b>
	50%	19.35	19.34	20.08	20.73	<b>24.7693±0.0160</b>

Tables 38 and 39 show the average PSNR and average FSIM for the same 12 test images from Figure 43 after performing both non-specific and specific noise-level denoising for mixed Gaussian, salt and pepper impulse and random value impulse noise. We used Gaussian noise with standard deviation 10 and 20, salt and pepper impulse noise of 10 and 40percent, and random value impulse noise of 10 and 30 percent. The number of epochs for specific noise-level denoising is 30 and for non-specific noise-level denoising is 35.

Tables 40 and 41 show the PSNR and FSIM of the results with non-specific noise-level denoising for mixed Gaussian and random value impulse noise on the BSD100 dataset. In

these tables, we use combinations of Gaussian noise with standard deviation 10, 20, 30, and 50 and random value impulse noise of 10, 20, 30, 40, and 50 percent. The network was trained for 150 epochs while further training showed promise of results improvement.

Table 37 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)
$\sigma = 10$	10%	96.65	97.48	97.57	97.63	<b><math>0.9777+8.6987 \times 10^{-5}</math></b>
	20%	96.06	96.63	96.78	96.99	<b><math>0.9712+1.0816 \times 10^{-4}</math></b>
	30%	95.25	95.32	95.78	96.17	<b><math>0.9677+1.6653 \times 10^{-4}</math></b>
	40%	93.69	93.25	93.93	93.88	<b><math>0.9480+2.0952 \times 10^{-4}</math></b>
	50%	90.99	90.23	90.96	91.00	<b><math>0.9386+2.1346 \times 10^{-4}</math></b>
$\sigma = 20$	10%	92.82	94.29	94.77	94.78	<b><math>0.9608+1.0206 \times 10^{-4}</math></b>
	20%	92.12	93.31	93.56	93.80	<b><math>0.9543+1.1349 \times 10^{-4}</math></b>
	30%	91.27	91.91	92.15	92.67	<b><math>0.9451+2.2174 \times 10^{-4}</math></b>
	40%	89.44	89.05	89.78	90.83	<b><math>0.9366+2.0672 \times 10^{-4}</math></b>
	50%	87.07	85.86	86.70	87.92	<b><math>0.9278+2.7104 \times 10^{-4}</math></b>
$\sigma = 30$	10%	90.19	91.06	91.67	91.68	<b><math>0.9414+9.0921 \times 10^{-4}</math></b>
	20%	89.32	89.66	89.86	90.42	<b><math>0.9382+1.6833 \times 10^{-4}</math></b>
	30%	88.30	87.91	87.96	89.17	<b><math>0.9303+0.0020</math></b>
	40%	85.95	84.47	85.06	86.59	<b><math>0.9223+2.1602 \times 10^{-4}</math></b>
	50%	82.93	81.13	81.93	83.78	<b><math>0.9136+4.1212 \times 10^{-4}</math></b>
$\sigma = 50$	10%	82.71	85.19	85.54	86.31	<b><math>0.9112+8.6603 \times 10^{-4}</math></b>
	20%	80.92	83.09	82.66	84.60	<b><math>0.9075+1.7635 \times 10^{-4}</math></b>
	30%	78.94	80.82	79.86	82.72	<b><math>0.9012+2.2361 \times 10^{-4}</math></b>
	40%	76.17	76.79	76.39	79.66	<b><math>0.8927+2.9754 \times 10^{-4}</math></b>
	50%	72.51	73.53	73.74	76.80	<b><math>0.8827+3.8588 \times 10^{-4}</math></b>

Table 38 – Average PSNR comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	Random value level noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	40%	10%	27.25	28.00	28.36	29.20	<b><math>31.9400+0.0179</math></b>	<b><math>30.7781+0.0104</math></b>
$\sigma = 20$	10%	30%	25.27	25.74	25.87	26.10	<b><math>28.2207+0.0089</math></b>	<b><math>28.6746+0.0102</math></b>

Table 39 – Average feature similarity index (FSIM) comparison for different mixed Gaussian, salt and pepper impulse noise, and random value impulse denoising filter against the proposed CNN filter (for specific and non-specific noise-level) on 12 test images

Gaussian noise	Impulse noise	Random value level noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	40%	10%	93.53	95.06	95.14	96.09	<b><math>0.9738+1.0498 \times 10^{-4}</math></b>	<b><math>0.9665+1.1537 \times 10^{-4}</math></b>
$\sigma = 20$	10%	30%	90.30	90.56	90.56	91.32	<b><math>0.9348+2.6162 \times 10^{-4}</math></b>	<b><math>0.9379+2.0485 \times 10^{-4}</math></b>

Table 40 – Average PSNR comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)
$\sigma = 10$	10%	27.66	29.13	29.15	30.12	<b>31.9660±0.0020</b>
	20%	27.06	27.67	27.79	28.54	<b>30.1583±0.0029</b>
	30%	26.44	26.27	26.75	27.27	<b>29.0402±0.0029</b>
	40%	25.40	25.22	25.43	27.27	<b>28.1507±0.0040</b>
	50%	24.09	23.61	23.92	24.37	<b>27.4191±0.0023</b>
$\sigma = 20$	10%	25.67	26.97	27.22	27.64	<b>28.9212±0.0033</b>
	20%	25.47	26.07	26.21	26.73	<b>28.6438±0.0031</b>
	30%	25.04	25.03	25.35	25.86	<b>27.7893±0.0055</b>
	40%	24.13	23.85	24.24	24.85	<b>27.2935±0.0043</b>
	50%	22.98	22.43	23.01	23.45	<b>26.7856±0.0021</b>
$\sigma = 30$	10%	24.86	25.38	25.71	25.87	<b>26.6653±0.0171</b>
	20%	24.46	24.35	24.65	24.96	<b>27.2172±0.0164</b>
	30%	23.97	23.53	24.01	24.51	<b>26.7719±0.0057</b>
	40%	23.03	22.56	23.01	23.73	<b>26.3448±0.0062</b>
	50%	21.74	21.21	21.89	22.69	<b>25.9463±0.0054</b>
$\sigma = 50$	10%	22.67	23.09	23.19	23.40	<b>25.6711±0.0012</b>
	20%	22.21	22.38	22.60	23.08	<b>25.6309±0.0023</b>
	30%	21.47	21.36	21.79	22.48	<b>25.3034±0.0030</b>
	40%	20.50	20.58	20.82	21.85	<b>24.7095±0.0056</b>
	50%	19.13	19.14	19.65	20.62	<b>23.8956±0.0050</b>

Table 41 – Average feature similarity index (FSIM) comparison for different mixed Gaussian and random value Impulse denoising filter against the proposed CNN filter (non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)
$\sigma = 10$	10%	87.25	91.82	92.51	92.70	<b>0.9358±5.9362× 10<sup>-5</sup></b>
	20%	86.12	90.00	90.38	90.83	<b>0.9116±7.4322× 10<sup>-5</sup></b>
	30%	84.91	87.72	88.20	88.94	<b>0.8897±0.0016</b>
	40%	82.60	81.63	82.61	88.94	<b>0.8938±1.7099× 10<sup>-4</sup></b>
	50%	80.20	79.34	80.36	79.62	<b>0.8463±2.8516× 10<sup>-4</sup></b>
$\sigma = 20$	10%	79.44	87.89	87.24	87.44	<b>0.9003±8.3381× 10<sup>-5</sup></b>
	20%	78.62	83.82	85.23	85.38	<b>0.8859±1.7674× 10<sup>-4</sup></b>
	30%	77.81	82.49	82.98	83.35	<b>0.8674±2.0273× 10<sup>-4</sup></b>
	40%	46.03	78.05	80.20	80.44	<b>0.8508±1.7974× 10<sup>-4</sup></b>
	50%	74.59	75.84	76.26	76.83	<b>0.8345±2.1202× 10<sup>-4</sup></b>
$\sigma = 30$	10%	77.30	80.96	83.46	82.70	<b>0.8644±7.9881× 10<sup>-5</sup></b>
	20%	76.59	79.44	80.27	80.20	<b>0.8558±9.7496× 10<sup>-5</sup></b>
	30%	76.01	78.46	78.82	78.87	<b>0.8422±1.8048× 10<sup>-4</sup></b>
	40%	74.27	75.05	75.72	75.74	<b>0.8277±2.4142× 10<sup>-4</sup></b>
	50%	72.78	73.04	73.19	71.98	<b>0.8129±2.2593× 10<sup>-4</sup></b>
$\sigma = 50$	10%	74.93	76.17	76.15	76.19	<b>0.8259±4.5774× 10<sup>-5</sup></b>
	20%	73.78	74.19	74.38	74.88	<b>0.8128±1.4083× 10<sup>-4</sup></b>
	30%	72.65	73.77	73.51	73.34	<b>0.7997±9.9043× 10<sup>-5</sup></b>
	40%	70.52	70.27	70.34	70.00	<b>0.7866±3.1593× 10<sup>-4</sup></b>
	50%	67.97	69.01	69.13	68.66	<b>0.7734±3.2558× 10<sup>-4</sup></b>

Tables 42 and 43 show the PSNR and FSIM of the results of both non-specific and specific noise-level denoising for mixed Gaussian, salt and pepper impulse, and random value impulse noise images of the BSD100 dataset. Gaussian noise with standard deviation 10 and 20, salt and pepper impulse noise of 10 and 40 percent, and random value impulse noise of 10 and 30 percent. The network was trained for 30 epochs for specific noise-level denoising and for 35 epochs for non-specific noise-level denoising.

Table 42 – Average PSNR comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	Random value level noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	40%	10%	26.20	26.13	26.45	27.11	<b>29.0286±0.0044</b>	<b>29.0522±0.0044</b>
$\sigma = 20$	10%	30%	23.91	24.58	24.55	24.70	<b>27.0155±0.0057</b>	<b>27.3811±0.0042</b>

Table 43 – Average feature similarity index (FSIM) comparison for different mixed Gaussian, salt and pepper Impulse noise, and random value Impulse denoising filter against the proposed CNN filter (specific and non-specific noise-level) on BSD100 dataset

Gaussian noise	Impulse noise	Random value level noise	WESNR [56]	$l_1$ -NLR [58]	$l_0$ -NLR [58]	LSM-NLR [58]	Proposed CNN (Non-specific)	Proposed CNN (Specific)
$\sigma = 10$	40%	10%	86.23	86.60	86.09	88.44	<b>0.8962±1.1804× 10<sup>-4</sup></b>	<b>0.9055±7.0711× 10<sup>-5</sup></b>
$\sigma = 20$	10%	30%	80.65	79.24	80.27	79.72	<b>0.8592±1.4142× 10<sup>-4</sup></b>	<b>0.8551±1.5470× 10<sup>-4</sup></b>

Figure 44 shows the results comparison when removing Gaussian noise with standard deviation 20 and 50 percent salt and pepper impulse noise from test image “Vase”. Figure 45 shows the denoising comparison for Gaussian noise with standard deviation 20 and 30 percent random value impulse noise from “Flower”. Figure 46 present the performance in the presence of Gaussian noise with standard deviation 10, 40 percent salt and pepper impulse noise, and 10 percent random value impulse noise from “Boat”. As seen from these figures, the proposed CNN attains better performance than all other filters at preserving relevant image details, obtaining the highest similarity, and achieving the least amount of

remaining noise. These improvements lead to better edge tracking especially when dealing with high intensity mixtures of impulse and Gaussian noise.

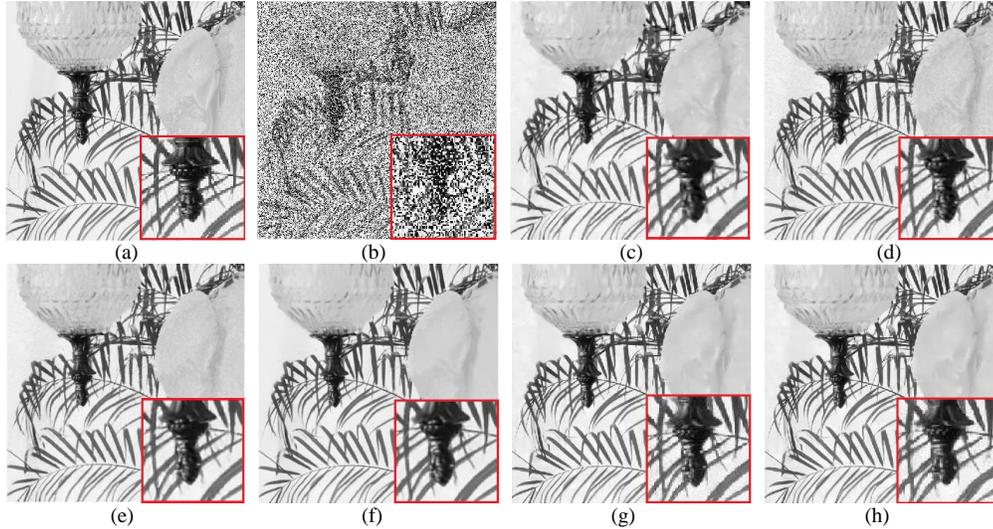


Figure 44 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 20 and 50 percent salt and pepper impulse noise on test image “Vase” image, a) Original image b) Noisy image; images denoised by (c) WESNR [56] (PSNR= 24.43dB, FSIM= 0.9235) d)  $l_1$  –NLR [58] (PSNR= 27.56dB, FSIM= 0.9442) e)  $l_0$  –NLR [58] (PSNR= 27.72dB, FSIM= 0.9464) f) LSM–NLR [58] (PSNR=29.24dB, FSIM= 0.9556) g) Non-specific noise-level proposed CNN (PSNR=29.17dB, FSIM= 0.9532) h) Specific noise-level proposed CNN (PSNR=29.58dB, FSIM=0.9586 )

Finally, Figures 47 through 49 showcase the denoising results of the proposed CNN for varying degrees of specific and non-specific noise-levels trained networks in the presence of different intensity mixture of Gaussian and impulse noise on multiple testing images.

#### 4.4.2 Impulse Denoising

To assess the merits of the proposed method [222] [239], different natural input images are used for evaluation. For comparative purposes, the results obtained using the proposed method are compared with SAMFWMF [185]. Again, all the parameters chosen for this comparison, such as initialization and regularization parameters, weights, and window sizes, are set according to their proposed optimal values for the specific noise level.

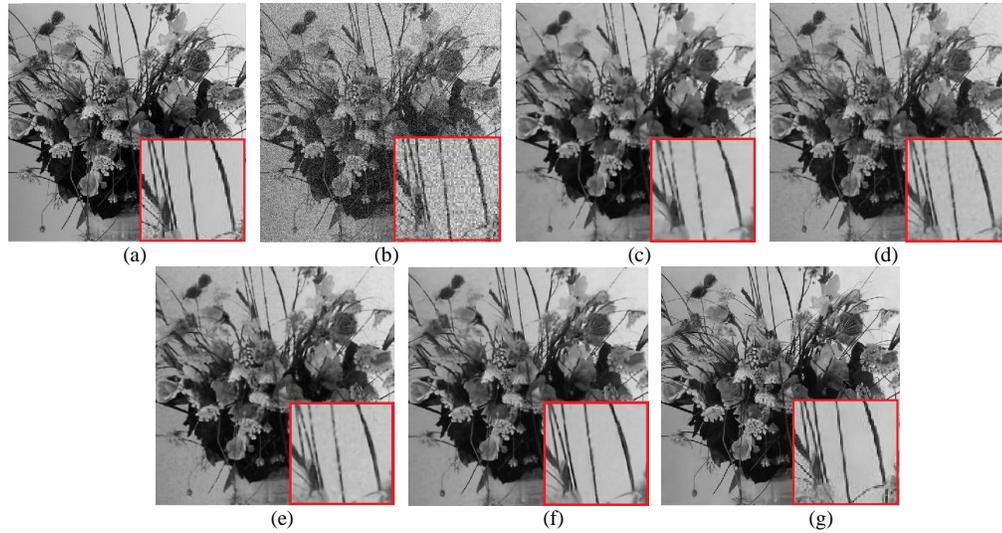


Figure 45 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 20 and 30 percent random value impulse noise on “Flower” image a) Original image b) Noisy image; images denoised by (c) WESNR [56] (PSNR= 23.04dB, FSIM= 0.8956) d)  $l_1$ -NLR [58] (PSNR=23.68 dB, FSIM=0.9120) e)  $l_0$ -NLR [58] (PSNR=23.51dB, FSIM=0.9071) f) LSM-NLR [58] (PSNR=24.36dB, FSIM=0.9156) g) Non-specific Noise-level proposed CNN (PSNR=27.07dB, FSIM=0.9482 )

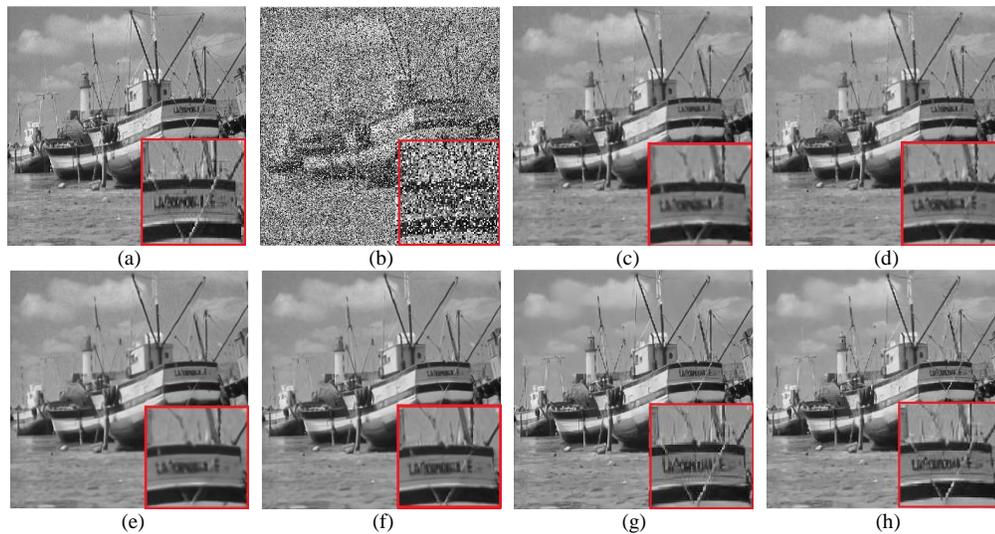


Figure 46 – Comparison of the denoising filters in the presence of Gaussian noise with standard deviation 10, 40 percent salt and pepper impulse noise, and 10 percent random value impulse noise on “Boat” image a) Original image b) Noisy image; images denoised by (c) WESNR [56] (PSNR= 27.32 dB, FSIM= 92.75 ) d)  $l_1$ -NLR [58] (PSNR=27.99dB, FSIM=0.9452) e)  $l_0$ -NLR [58] (PSNR=27.99dB, FSIM=0.9396) f) LSM-NLR [58] (PSNR=28.89dB, FSIM=0.9482) g) Non-specific Noise-level proposed CNN (PSNR= 30.97 dB, FSIM=0.9620 ) h) Specific noise-level proposed CNN (PSNR=30.82dB, FSIM=0.9646)

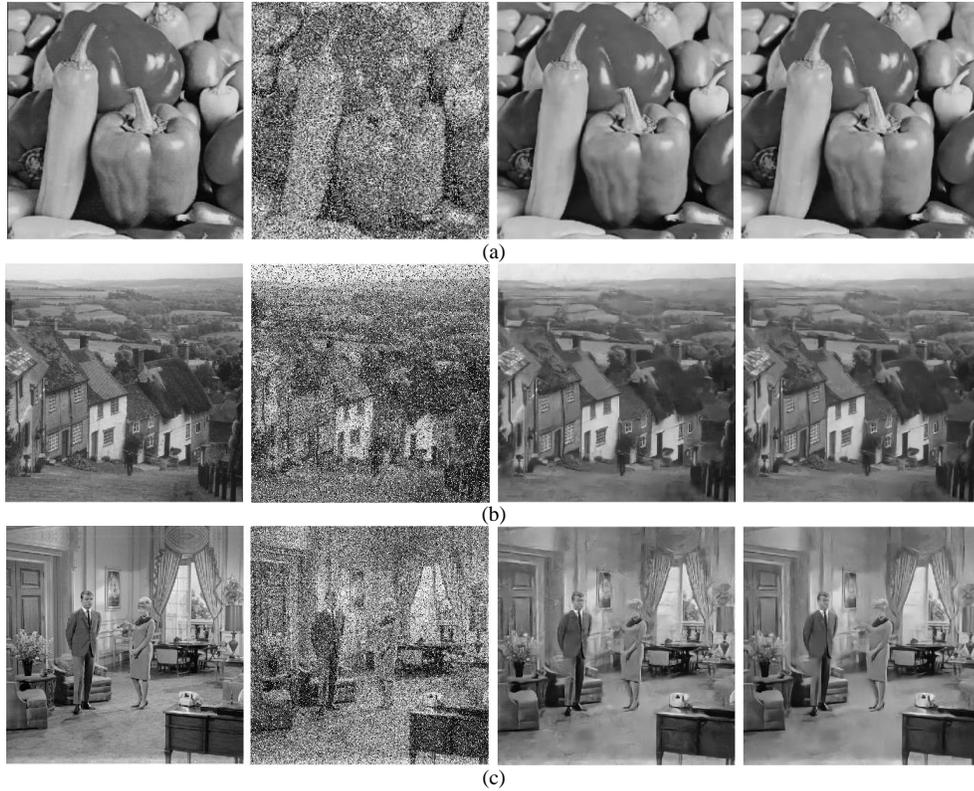


Figure 47 – Application of the proposed CNN filter in the presence of different mixed Gaussian and salt and pepper Impulse noise intensities on different natural images, columns 1 through 4 are: Original test image, corrupted image with mixed Gaussian and salt and pepper impulse noise, non-specific noise-level denoising, and specific noise-level denoising a) Test image “Fruits” corrupted with Gaussian (s.d.=10) and salt and pepper (50%) (Non-specific Noise-level: PSNR=33.28 dB, FSIM=0.9736) (Specific noise-level: PSNR=33.53 dB, FSIM=0.9743) b) Test image “Hill” corrupted with Gaussian (s.d.=30) and salt and pepper (20%) (Non-specific noise-level: PSNR=28.57 dB, FSIM=0.9200) (Specific noise-level: PSNR=28.77 dB, FSIM=0.9236) c) Test image “Couple” corrupted with Gaussian (s.d.=50) and salt and pepper (10%) (Non-specific noise-level: PSNR=26.05 dB, FSIM=0.8844) (Specific noise-level: PSNR=26.34 dB, FSIM=0.8830)

We also used the 400 images (180×180) from Berkley segmentation dataset (BSD) [89] in the training phase. We set as testing images (8 natural images (512×512) as shown in figure 51. We emphasize that the images that are used in the testing phase were not seen in the training phase. As we discussed before, the optimal results were obtained by using depth of 20 with 40×40 patch size for non-specific noise-level denoising. In this

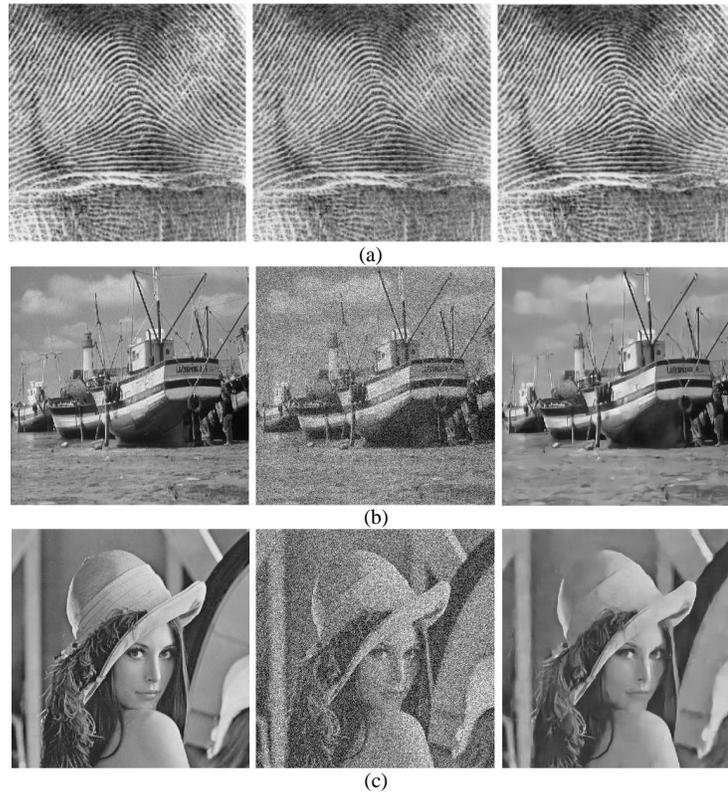


Figure 48 – Application of the proposed CNN filter in the presence of different mixed Gaussian and random value Impulse noise intensities on different natural images, columns 1 through 3 are: Original test image, corrupted image with mixed Gaussian and random value impulse noise, and non-specific noise-level denoising a) Test image “Finger print” corrupted with Gaussian (s.d.=10) and random value (10%) (PSNR=31.14 dB, FSIM=0.9906) b) Test image “Boat” corrupted with Gaussian (s.d.=30) and random value (30%) (PSNR=28.24 dB, FSIM=0.9211) c) Test image “Lena” corrupted with Gaussian (s.d.=50) and random value (50%) (PSNR=27.31 dB, FSIM=0.9107).

implementation stochastic gradient descent (SGD)-momentum [228] with weight decay of 0.0001 and momentum of 0.9 and a mini-batch of 128 is used. The 50 epochs are trained for our model. MatConvNet package [89] which is a MATLAB toolbox for convolutional neural network (CNN) is used in this case. All implementations are carried out using MATLAB 2017b on a PC with Nvidia GPU. The training time of the network is about 1 day. Tables 44 shows the results obtained on the averaged peak signal to noise ratio (PSNR), averaged structural similarity index (SSIM) and averaged FSIM measures,

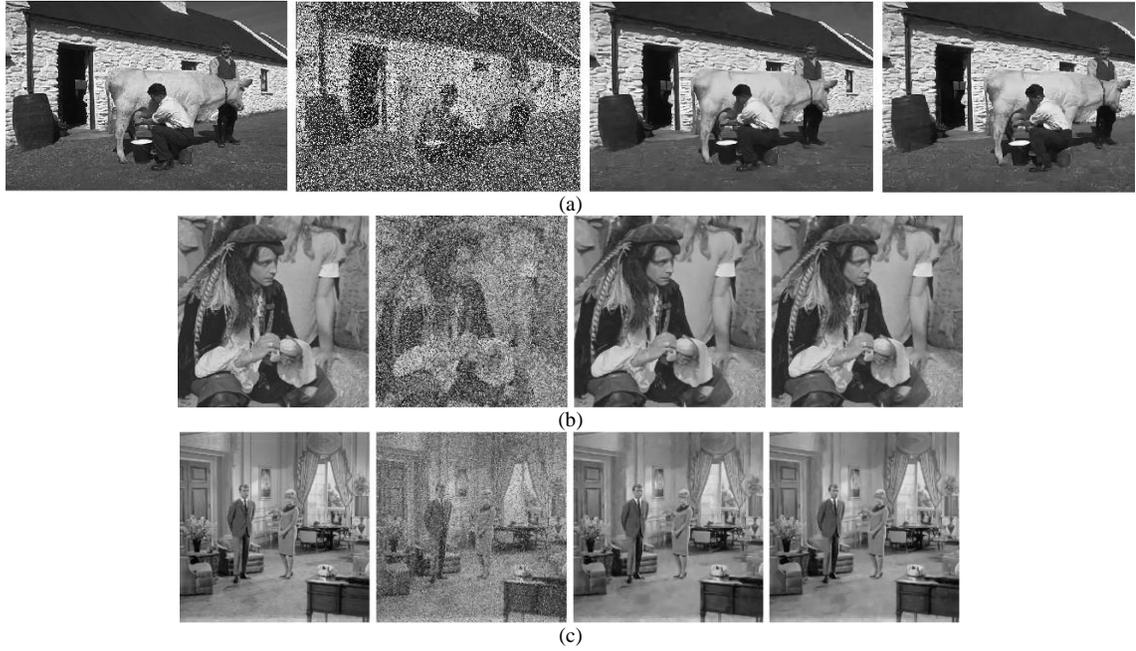


Figure 49 – Application of the proposed CNN filter in the presence of different mixed Gaussian, salt and pepper Impulse, and random value Impulse noise intensities on different natural images, columns 1 through 4 are: Original test image, corrupted image with mixed Gaussian and salt and pepper impulse and random value impulse noise, non-specific noise-level denoising, and specific noise-level denoising a) Test image “385039 of BSD100 dataset” corrupted with Gaussian (s.d.=10), salt and pepper (40%) and random value (10%) (Non-specific noise-level: PSNR=27.18 dB, FSIM=0.8902) (Specific noise-level: PSNR=27.09 dB, FSIM=0.8962) b) Test image “Man” corrupted with Gaussian (s.d.=10), salt and pepper (40%) and random value (10%) (Non-specific noise-level: PSNR=31.05 dB, FSIM=0.9585) (Specific noise-level: PSNR=30.86 dB, FSIM=0.9599) c) Test image “Couple” corrupted with Gaussian (s.d.=20), salt and pepper (10%) and random value (30%) (Non-specific Noise-level: PSNR=28.16 dB, FSIM=0.9266) (Specific noise-level: PSNR=28.55 dB, FSIM=0.9303)

Table 44 – Average peak signal to noise ratio (PSNR), average structural similarity index (SSIM), and averaged FSIM comparison between proposed CNN filter and AMFWMF [179] denoising process in presence of different impulse noise intensities

	<i>SAMFWMF[185]</i>			<i>Proposed CNN</i>		
	<i>PSNR</i>	<i>SSIM</i>	<i>FSIM</i>	<i>PSNR</i>	<i>SSIM</i>	<i>FSIM</i>
10%	28.8309	0.9698	0.9979	40.7748 ±0.0125	0.9869±4.4721×10 <sup>-5</sup>	0.9980±4.5241×10 <sup>-5</sup>
20%	28.6718	0.9549	0.9965	33.9328±0.0171	0.9780±7.6089×10 <sup>-5</sup>	0.9969±2.2361×10 <sup>-5</sup>
30%	26.3245	0.9437	0.9932	37.4581±0.0148	0.9684±7.9472×10 <sup>-5</sup>	0.9955±5.1042×10 <sup>-5</sup>
40%	23.9689	0.9052	0.9926	36.1030±0.0187	0.9574±9.2338×10 <sup>-5</sup>	0.9939±0.0013
50%	23.7025	0.8815	0.9902	34.7381±0.0229	0.9466±0.0112	0.9909±5.1042×10 <sup>-5</sup>
60%	23.9220	0.8442	0.9838	33.2798±0.0204	0.9269±1.8778×10 <sup>-4</sup>	0.9865±8.3351×10 <sup>-5</sup>
70%	23.2654	0.8331	0.9721	31.5983±0.0335	0.9027±2.4623×10 <sup>-4</sup>	0.9790±1.4749×10 <sup>-4</sup>
80%	21.5890	0.7865	0.9521	29.5832±0.0227	0.8641±3.6746×10 <sup>-4</sup>	0.9644±2.8266×10 <sup>-4</sup>
90%	20.0002	0.7201	0.9067	26.6888±0.0346	0.7835±5.1186×10 <sup>-4</sup>	0.9269 ±5.1186×10 <sup>-4</sup>

comparing different filters against the proposed filter (results for the SAMFWMF [185] are based on the minimum and maximum initial window size of the adaptive median filter for the related noise level. All these metrics are computed in the presence of 10 to 90 percent impulse noise intensities on images frequently used in the literature for the denoising purposes (i.e. “8 testing images”).

Figures 50 show the same comparison in the presence of 90% impulse noise on test image “Lena”. Figure 51 shows the proposed non-specific noise-level CNN denoising filter results in the presence of different noise intensities on different testing images. As these figures show, the proposed algorithm has good performance in terms of keeping relevant detail and obtaining the highest similarity, least noise, and preserving edges, especially in high impulse noise environments.

#### 4.4.3 Results After Edge detection

To evaluate the performance of the proposed filter after edge detection step, the visual results are provided on different natural images. Figures 52 through 54 shows the results

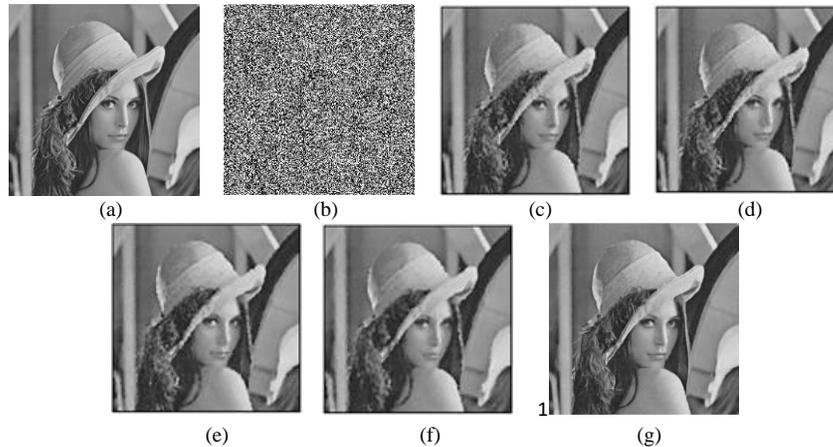


Figure 50 – Comparison of the denoising filters in the presence of 90 percent salt and pepper impulse noise on test image “Lena”, a) Original image b) Noisy image c) AMFWMF(initial window size=3) d) AMFWMF (initial adaptive median window size=5) e) ) AMFWMF (initial adaptive median window size=7) f) ) SAMFWMF (initial adaptive median window size=9) [185] g) Proposed bling CNN

after edge detection in the presence of different mixed Impulse noise and Gaussian noise intensities on different testing images.

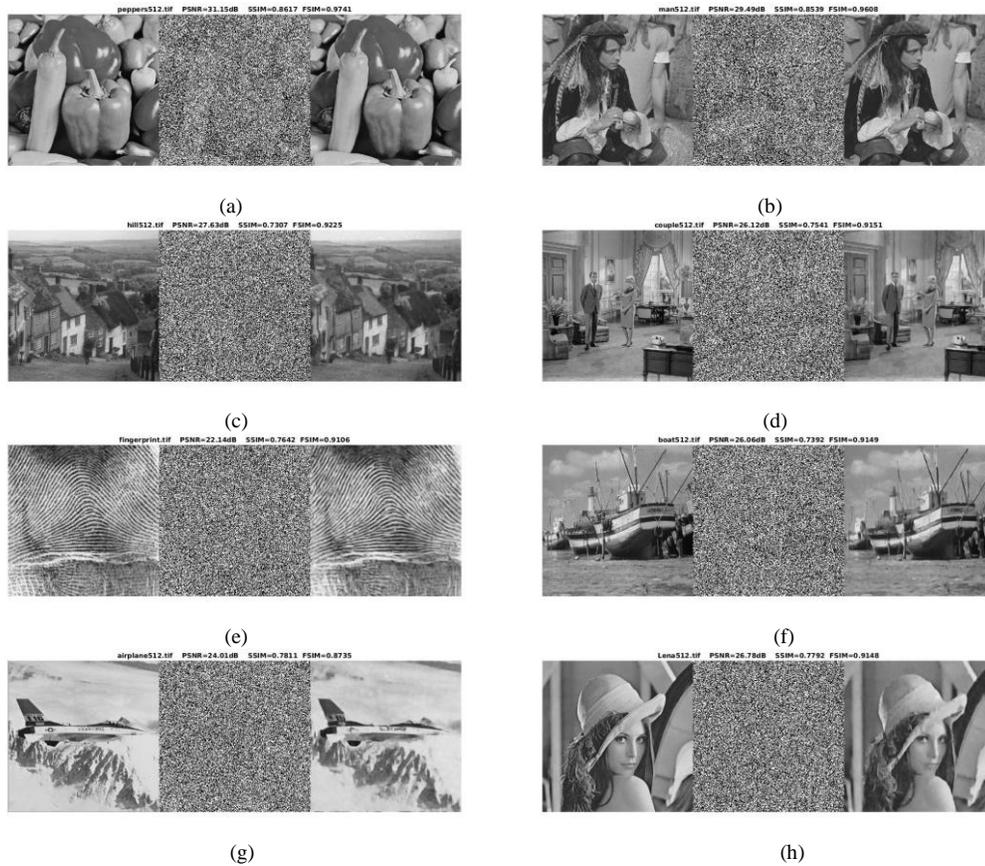


Figure 51 – Application of the proposed CNN filter in the presence of different salt and pepper Impulse noise intensities on different testing images, columns 1 through 3 are: Original test image, noisy image, and denoised image a) Test image “Fruits” corrupted with 80% impulse noise b) Test image “Man” corrupted with 80% impulse noise c) Test image “Hill” corrupted with 90% impulse noise d) Test image “Couple” corrupted with 90% impulse noise e) Test image “Finger print” corrupted with 90% impulse noise f) Test image “Boat” corrupted with 90% impulse noise g) Test image “Airplane” corrupted with 95% impulse noise h) Test image “Lena” corrupted with 95% impulse noise

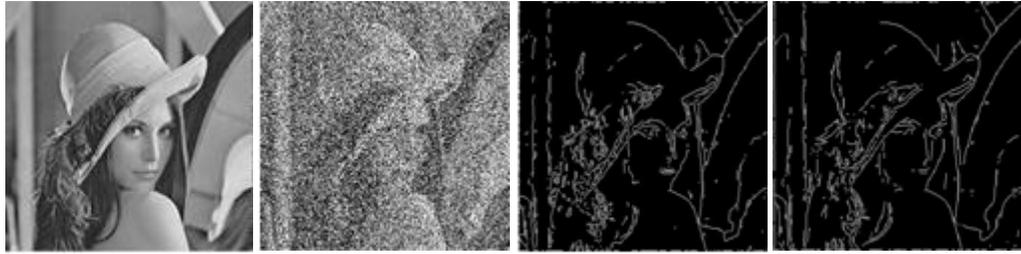


Figure 52 – Edge detection after applying the proposed CNN filter (specific and non-specific noise-level) in the presence of mixed Gaussian and salt and pepper Impulse noise, a) Original test image “Lena”, b) corrupted image with mixed Gaussian (s.d=50) and salt and pepper impulse (50%) c) edge detection after blind denoising (Normalized threshold=0.1) d) edge detection after certain level denoising (Normalized threshold=0.1)



Figure 53 – Edge detection after applying the proposed CNN filter (non-specific noise-level) in the presence of mixed Gaussian and random value Impulse noise, columns 1 through 3 are: Original test image “Fruits”, corrupted image with mixed Gaussian (s.d=50) and random value impulse (50%), edge detection after blind denoising (Normalized threshold=0.08), and edge detection after certain level denoising (Normalized threshold=0.08)



Figure 54 – Edge detection after applying the proposed CNN filter (specific and non-specific noise-level) in the presence of mixed Gaussian, salt and pepper Impulse, and random value Impulse noise, Original test image “Boat”, corrupted image with mixed Gaussian (s.d=20) and salt and pepper impulse (10%) and random value impulse (30%), edge detection after blind denoising (Normalized threshold=0.1), and edge detection after certain level denoising (Normalized threshold=0.1)

## 5. CHAPTER V

### SUMMARY & CONCLUSIONS

This dissertation contends with noise in digital images. The first Chapter introduces the research objectives and what we aim to achieve through the newly developed denoising method to be followed by effective edge detection. The goal is to preserve image details while minimizing the effects of noise. Chapter II provides the literature survey of the current literature in relation to the theme of this dissertation. This chapter presented a comprehensive retrospective on impulse, Gaussian, and mixed impulse and Gaussian denoising filters which are applied to digital images to reduce the effects of the different noise types and combinations. We considered a random noise model comprised of impulse (salt and pepper) and Gaussian noise. We have explained the noise models and denoising filters, as well as classified them according to their types and domain of application. The merits of each of the methods reviewed are assessed in comparison to other related methods in terms of their application domain and in terms of the different performance levels they achieve. This survey allows researchers to also gauge the progress in this challenging research endeavor and to ascertain which method and which metrics they would contemplate using for their own research as a preprocessing step when dealing with noisy images.

In chapter III, the relevant theory and related methodologies were presented. In the first subsection, a new combination of median and mean filter, we refer to as the switching adaptive median and mean filter (SAMFWMF), with additional shrinkage window, was introduced as a new smoothing filter to remove or minimize in an optimal fashion the

presence of impulse noise even at high intensity levels. The adaptive properties of the median filter are proven to control the similarity and edge smoothing as an option to adjust the smoothness and sharpness of the edges. Also, a shrinkage window is introduced in order to improve the denoising process, and the entire process is completed by applying a  $2 \times 2$  fixed weighted mean filter. The properties of the mean filter as set provide a considerable improvement on the denoising process while circumventing image blurring, especially under high impulse noise intensity levels. Also, the weights are set for the fixed mean filter based on probabilities of noise occurrence with the ability to remove the remaining noise in the image with the least effect on non-noisy pixels. Also, the switching property of the denoising filter introduced a new option which is able to denoise the images like the challenging case of the checkerboard even in the presence of high-intensity impulse noise. This combination of filters is shown to yield the best (i.e., highest) structural metrics than any other well-known denoising filter in the presence of different impulse noise intensities. Denoising under this method is shown to preserve edge details with good edge preserving capability as reflected through the highest structural similarity measure between the denoised image and the original noise-free image. This filtering method also allows edge detection algorithms to become immune and resilient to noise, enhancing image segmentation, object recognition, feature extraction, pattern classification, and deriving structural and functional measurements in medical imaging especially MRI and CT images.

Chapter IV- provides the results and discussions. A comparative assessment is also provided in chapter IV in terms the filter developed and its denoising performance in contrast to state-of-the-art filters that have been proven effective in the literature. The

comparative results that were presented indicate that the proposed method outperformed state-of-the-art methods and filters which were designed to remove this type of noise on both natural and medical images. In the edge detection phase, and after the smoothing process attained with SAMFWMF, we observe that our method preserved edge continuity and tracked well the boundaries, especially in high predefined thresholds in relation to the use of maximum-sequence, whose intent was to detect more edges at different threshold intensities while minimizing the effect of noise. This new approach led to a better performance in contrast to other common thresholding methods. For visual appreciation of the optimal outcome, several morphological operations were used on the final image. The results obtained proved that the proposed method yielded a better performance after edge detection even in the presence of high intensity impulse noise.

In the second subsection of Chapter IV, the focus was placed on a combination of the dual-tree complex wavelet and improved total variation introduced as a new smoothing filter to remove or minimize in an optimal fashion the presence of combined speckle and Gaussian noise on ultrasound images. Dual-tree complex wavelet transform is shown to yield better structural metrics than standard wavelet transforms denoising filter. Its combination with total variation filter is shown to yield better performance than dual-tree complex wavelet and total variation independently as well as other well-known de-speckling filters. Furthermore, this combination is shown to preserve image details and has good edge preservation capability as reflected by the highest similarity between the denoised image and the original noise-free image. This is extremely useful in order to enhance image segmentation, object recognition, feature extraction, pattern classification,

and deriving necessary structural or functional measurements in medical imaging as this practical implementation on ultrasound images has proven. The results obtained using the proposed method after applying the steps of denoising are presented in presence of combined Gaussian and speckle noise. These results are compared with some well-known de-speckling filters reported in the literature. Also, we observe that our method has very good performance in presence of Gaussian noise alone, especially in high intensities. Again, the results obtained proved that the proposed method yielded a good performance after edge detection even in the presence of high intensity noise.

Chapter V provides concluding remarks on the many denoising methods and relates the merits of each in accordance to their performance and domain of applications.

In the area of signal and image denoising, the theory and methodology could be used to detect and remove different kind of random noise. Such a generalized approach could be very useful for any challenging applications in signal and image processing often fraught with ubiquitous noise effects. This research topic can be further extended to various applications that could include medical imaging, biometrics, and telecommunication systems. The efficiency of the proposed method can be improved through further research. The ultimate plan is to implement the proposed methods in order to design denoising algorithms that can automatically the source of noise, formulate its effect, and model the filter design to minimize the presence of noise. A singular merit of the proposed filters is in their ability to remove the different noise types in isolation or in combination even in high intensity levels of noise, and still yield an output with high correlation and similarity to the noise free image.

The importance of image segmentation and object detection in different industries especially medical industry is obvious. The goal of segmentation is to simplify or change the representation of an image into something that is more meaningful and easier to analyze. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. Object detection deals with detecting instances of semantic objects of a certain class in digital images and videos. Their importance is further amplified when we deal with medical images. New algorithms for image segmentation could extend to object detection and identification. In the medical field, delineating certain anatomical structures could help localize tumors or diseased tissue from healthy tissue all in context to key anatomical landmarks. Such research could eventually extend to classification, diagnosis and for surgical planning.

## LIST OF REFERENCES

- [1] D. Marr and E. Hildreth, "Theory of Edge-Detection", *Proceedings of the Royal Society Series B-Biological Sciences*, vol. 207, no. 1167, pp. 187-217, 1980.
- [2] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, Lond. 160, pp. 106-154, 1962.
- [3] F. W. Campbell, and J. G. Robson, "Applications of Fourier analysis to the visibility of gratings," *J. Physiol.*, Lond. 197, pp. 551-556, 1968.
- [4] F. Candocia and M. Adjouadi, "A Similarity Measure for Stereo Feature Matching," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1460-1464, Oct. 1997.
- [5] M. Adjouadi and F. Candocia, "A Stereo Matching Paradigm Based on the Walsh Transformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1212-1218, Dec. 1994.
- [6] R. Duara, W. Barker, D. Loewenstein, M. T. Greig-Custo, R. Rodriguez, M. Goryawala, Q. Zhou, and M. Adjouadi, "Insights into cognitive aging and Alzheimer's disease using amyloid PET and structural MRI scans", *Clinical and Translational Imaging*, vol. 3, no. 1, pp. 65-74, Feb. 2015.
- [7] A.M. Guzman, M. Goryawala, J. Wang, A. Barreto, J. Andrian, N. Rishe and M. Adjouadi, "Thermal Imaging as a Biometrics Approach to Facial Signature Authentication," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 1, pp. 214-222, Jan. 2013.
- [8] R. Bhatt, M. Adjouadi, M. Goryawala, S. Gulec, and A. McGoron, "An algorithm for PET tumor volume and activity quantification: Without specifying camera's point spread function (PSF)," *Medical Physics*, vol. 39, no. 7, pp. 4187-4202, Jul. 2012.
- [9] M. Goryawala, S. Gulec, R. Bhatt, A. J. McGoron, M. Adjouadi., "A low-interaction automatic 3D liver segmentation method using computed tomography for selective internal radiation therapy," *Biomed Res Int.*, vol. 2014, Jul. 2014.
- [10] A. C. Bovik, *Hand Book of Image and Video Processing*, 2nd ed., Cambridge, MA, USA: Academic press, 2000.
- [11] G. E. Healey, R. Kondepudy, "Radiometric CCD Camera Calibration and Noise Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 3, pp. 267-276, 1994.
- [12] K. M Moon, M. D. Patil, B. Parmar, "Image Restoration Using Adaptive Switching Median Filter," in: *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*., Coimbatore, India, 2010.
- [13] W. Liu and W. Lin, "Additive White Gaussian Noise Level Estimation in SVD Domain for Images," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 872-883, 2013.

- [14] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*, Boca Raton, FL, USA: CRC Press, 2005.
- [15] R. E. Jacobson, S. F. Ray, G. G. Attridge, and N. R. Axford, *The Manual of Photography*, 9th ed., Waltham, MA, USA: Focal Press, 2000.
- [16] A. K. Jain, *Fundamental of Digital Image Processing*, (Prentice-Hall; Englewood Cliffs, NJ: 1989).
- [17] O. V. Michailovich, Tannenbaum, A. "Despeckling of Medical Ultrasound Images," *IEEE trans. Ultrason. Ferroelectr. Freq. Control*, vol. 53, no. 1, pp. 64 -78, 2006.
- [18] Mehdi Mafi, Harold Martin, Mercedes Cabrerizo, Jean Andrian, Armando Barreto, and Malek Adjouadi "A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images," *Signal Process.*, vol. 157, pp. 236 -260, 2019.
- [19] Mehdi Mafi, Walter Walter Izquierdo, and Malek Adjouadi "A survey on Mixed Impulse and Gaussian denoising filters," *under review*.
- [20] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. Image Process.*, vol. 4 no. 4, pp. 499-502, 1995.
- [21] R. H. Chan, C-W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-type Noise Detectors and Detail-preserving Regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479-1485, 2005.
- [22] H. Ibrahim, N. Sia Pik Kong, Theam Foo Ng, "Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1920-1927, 2008.
- [23] V. Crnojevic', V. Šenk and Ž. Trpovski, "Advanced Impulse Detection Based on Pixel-Wise MAD," *IEEE Signal Process. Lett.*, vol. 11, no. 7, 2004.
- [24] Z. Wang and D. Zhang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Circuits Syst.—II: Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 70-80, 1999.
- [25] H-L. Eng., K-K. Ma, "Noise Adaptive Soft-Switching Median Filter," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 242-251, 2001.
- [26] Y. Hashimoto, Y. Kajikawa, and Y. Nomura, "Directional Difference-Based Switching Median Filters," *Electronics and Communications in Japan*, part 3, vol. 85, no. 3, pp. 22-32, 2002.
- [27] S. Zhang, M. A. Karim, "A New Impulse Detector for Switching Median Filters," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 360-363, 2002.
- [28] V.V. Khryashev, A.L. Priorov, I.V. Apalkov, P.S. Zvonarev, "Image Denoising Using Adaptive Switching Median Filter," in: Proceedings of the IEEE 2th International Conference on Image processing (ICIP), 2005.

- [29] S.-J. Horng, L.-Y. Hsu, T. Li, S. Qiao, X. Gong, H.-H. Chou, M. K. Khan, "Using sorted switching median filter to remove high-density impulse noises," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 956–967, 2013.
- [30] T. Chen, K.-K. Ma, and L.-H. Chen, "Tri-State Median Filter for Image Denoising," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1834-1838, 1999.
- [31] T. Chen and H. R. Wu, "Space Variant Median Filters for the Restoration of Impulse Noise Corrupted Images," *IEEE Trans. Circuits Syst.—II: Analog Digit. Signal Process.*, vol. 48, no. 8, pp. 784-789, 2001.
- [32] Y. Dong, S. Xu, "A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, 2007.
- [33] K. Aiswarya, V. Jayaraj, D. Ebenezer, "A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noise in Images and Videos," in: Proceedings of the IEEE 2th International conference on Computer Modeling and Simulation (ICCMS), pp. 409-413, 2010.
- [34] K. S. Srinivasan, D. Ebenezer, "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 89-192, 2007.
- [35] P.-E. Ng, K.-K. Ma, "A Switching Median Filter With Boundary Discriminative Noise Detection for Extremely Corrupted Images", *IEEE Trans on Image Process.*, Vol. 15, No. 6, pp.1506-1516, Jun. 2006.
- [36] I. F. Jafar, R. A. AlNa'mneh, and K. A. Darabkh, "Efficient Improvements on the BDND Filtering Algorithm for the Removal of High-Density Impulse Noise," *IEEE Trans. on Image Processing*, vol. 22, no. 3, pp. 1223-1232, 2013.
- [37] X. Zhang and Y. Xiong, "Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter," *IEEE Signal Process. Lett.*, vol.16, no.4, pp.295-298, 2009.
- [38] P. Zhang, F. Li, "A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise," *IEEE Signal Process. Lett.*, vol. 21, no. 10, 2014.
- [39] C. Kandemir, C. Kalyoncu, Ö. Toygar, "A weighted mean filter with spatial-bias elimination for impulse noise removal," *Digital Signal Processing*, vol. 46, pp. 164–174, 2015.
- [40] Z. Zhou, "Cognition and Removal of Impulse Noise With Uncertainty," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3157-3167, 2012.
- [41] C. Kalyoncu, Ö. Toygar, H. Demirel, "Interpolation-based impulse noise removal," *IET Image Process.*, vol. 7, no. 8, pp. 777-785, 2013.
- [42] X. Wang, S. Shen, G. Shi, Y. Xu, P. Zhang, "Iterative non-local means filter for salt and pepper noise removal," *J. Vis. Commun. Image Rep.*, vol. 38, pp. 440–450, 2016.

- [43] M. S. Nair, K. Revathy, R. Tataavarti, "An Improved Decision Based Algorithm for Impulse Noise Removal," in: Proceedings of the IEEE Congress on Image and Signal Processing, pp. 426–431, 2008.
- [44] S. Balasubramanian, S. Kalishwaran, R. Muthuraj, D. Ebenezer, V. Jayaraj, "An efficient non-linear cascade filtering algorithm for removal of high density salt and pepper noise in image and video sequence," In: Proceeding IEEE International Conference Control, Automation, Communication and Energy Conservation (INCACEC), pp. 1-6, 2009.
- [45] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, C. H. PremChand, "Removal of High Dendity Salt and Pepper noise Through Modified Decision based Unsymmetrical Trimmed Median Filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, 2011.
- [46] A. Pattnaik, S. Agarwal, S. Chand, "A New and Efficient Method for Removal of High Density Salt and Pepper Noise through Cascade Decision Based Filtering Algorithm," Elsevier, In: Proceedings of the 2nd International Conference on Communication, Computing & Security, 6, pp. 108 – 117, 2012.
- [47] M. T. Raza, S. Sawant, "High density salt and pepper noise removal through decision based partial trimmed global mean filter," in: Proceedings IEEE International Conference Engineering (NUiCONE), pp.1-5, 2012.
- [48] A. K. Samantaray, P. Mallick, "Decision Based Adaptive Neighborhood Median Filter," *Procedia Computer Science*, vol. 48, pp. 222-227, 2015.
- [49] K. Vasanth, T. G. Manjunath, N. Raj, "A Decision Based Unsymmetrical Trimmed Modified Winsorized Mean Filter for the Removal of High Density Salt and Pepper Noise in images and Videos," *Procedia Computer Science*, vol. 54, pp. 595-604, 2015.
- [50] A. Dash , S. K. Sathua, "High Density Noise Removal By Using Cascading Algorithms," in: Proceedings of the IEEE 5th International Conference Advanced Computing & Communication Technologies, pp.96-101, 2015.
- [51] J. Oh, Luis F: Chuparro, "Ranked Directional Morphological Filtering of Impulse Noise in Images," in: Proceedings IEEE Conference Acoustics, Speech, and Signal Processing (ICASSP), 2000.
- [52] D. Ze-Feng, Y. Zhou-Ping, X. You-Lun, "High Probability Impulse Noise-Removing Algorithm Based on Mathematical Morphology," *IEEE Signal Process. Lett.*, vol. 14, no. 1, 2007.
- [53] K. Ratna Babu, K. V. N. Sunitha, "Image De-noising and Enhancement for Salt and Pepper Noise Using Improved Median Filter-Morphological Operations," in: Proceedings Springer International Conference Advances in Communication, Network, and Computing, pp. 7-14, 2012.
- [54] H. Xu , G. Zhu, H. Peng, D. Wang, "Adaptive fuzzy switching filter for images corrupted by impulse noise," *Pattern Recognition Lett.*, vol. 25, no. 15, pp. 1657–1663, 2004.

- [55] S. Schulte, M. Nachtgael, V. De Witte, D. Van der Weken, E. E. Kerre, "A Fuzzy Impulse Noise Detection and Reduction Method," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1153-1162, 2006.
- [56] L. Tang, H. Wang, B. Qi, "A New Fuzzy Logic Image De-noising Algorithm Based on Gradient Detection," in: Proceedings of the IEEE 4th International Conference on Fuzzy Systems and Knowledge Discovery, 2007.
- [57] F. Russo, G. Ramponi, "A fuzzy filter for images corrupted by impulse noise," *IEEE Signal Processing Lett.*, vol. 3, pp. 168–170, 1996.
- [58] Wenbin Luo, Member, IEEE, "Efficient Removal of Impulse Noise from Digital Images," *IEEE Trans Consum. Electron.*, vol. 52, no. 2, pp. 523-527, 2006.
- [59] L. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: non-linear phenomena*, vol. 60, no. (1-4), pp. 259–268, 1992.
- [60] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," *J. mathematical imag. vision*, vol. 20, no. (1-2), pp. 89-97, 2004.
- [61] Yue Hu, Mathews Jacob, "Image Recovery Using Improved Total Variation Regularization," IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011.
- [62] J. Zhang, S. Liu, R. Xiong, S. Ma, D. Zhao, "Improved Total Variation based Image Compressive Sensing Recovery by Nonlocal Regularization," IEEE International Symposium on Circuits and Systems (ISCAS), 2013.
- [63] L. Hu, U KinTak, "Removal of Gaussian Noise with Non-uniform Total Variation Partition," in: Proceedings of the IEEE 10th Conference on Information, Communications and Signal Processing (ICICS), 2015.
- [64] P. Perona, J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629-639, 1990.
- [65] F. Catte, P.-L. Lions, J.-M. Morel, T. Coll, "Image Selective Smoothing and Edge detection by Non-linear Diffusion," *SIAM Journal of Numerical Analysis*, vol. 29, no. 1, pp. 182-193, 1992.
- [66] Z. Lin, Q. Shi, "An anisotropic diffusion PDE for noise reduction and thin edge preservation," in: Proceedings of the 10th IEEE International Conference on Image Analysis and Processing, 1999.
- [67] S.-M. Chao, D.-M. Tsai, "An improved anisotropic diffusion model for detail- and edge-preserving smoothing," *Pattern Recognition Lett.*, vol. 31, no. 13, pp. 2012–2023, 2010.
- [68] J. Xu, Y. Jia, Z. Shi, K. Pang, "An improved anisotropic diffusion filter with semi-adaptive threshold for edge preservation," *Signal Process.*, vol. 119, pp. 80–91, 2016.

- [69] A. Buades, B. Coll, J-M Morel, “A non-local algorithm for image denoising,” in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 60- 65, 2005.
- [70] Charles Kervrann, Jérôme Boulanger, and Pierrick Coup, “Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal,” in: Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision, pp. 520-532, 2007.
- [71] M. G. Maruf, M. R. El-Sakka, “Improved Non-Local Means Algorithm Based on Dimensionality Reduction,” in: Proceedings of the International Conference on Image Analysis and Recognition, pp. 43-50, 2015.
- [72] R. Lai, X.-X. Dou, “Improved Non-local Means Filtering Algorithm for Image Denoising,” in: Proceedings of the IEEE 3rd International Congress on Image and Signal Processing (CISP), 2010.
- [73] B. Goossens, H. Luong, A. Pižurica, W. Philips, “An improved non-local denoising algorithm,” Presented at the International Workshop on Local and Non-Local Approximation in Image Processing, 2008.
- [74] M. Kazemi, E. Mohammadi, P. shahidi sadeghi, M. B. Menhaj, “A Non-Local Means Approach for Gaussian Noise Removal from Images using a Modified Weighting Kernel,” in: Proceedings of the 25th Iranian Conference on Electrical Engineering (ICEE), 2017.
- [75] C. Tomasi, R. Manduchi, “Bilateral Filtering for Gray and Color Images,” in: Proceedings of the IEEE International Conference on Computer vision (ICCV), 1998.
- [76] K. N. Chaudhury, K. Rithwik, “Image denoising using optimally weighted bilateral filters: A sure and fast approach,” in: Proceedings IEEE International Conference on image Processing (ICIP), 2015, pp. 108–112.
- [77] K. N. Chaudhury, D. Sage, and M. Unser, “Fast  $O(1)$  bilateral filtering using trigonometric range kernels,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376-3382, 2011.
- [78] K. N. Chaudhury, “Acceleration of the shiftable  $O(1)$  algorithm for bilateral filtering and nonlocal means,” *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1291-1300, 2013.
- [79] Y.L. You, M. Kaveh, “Fourth-order partial differential equation for noiseremoval,” *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1723–1730, 2000.
- [80] M. Lysaker, A. Lundervold, X.-C. Tai, “Noise Removal Using Fourth-Order Partial Differential Equation With Applications to Medical Magnetic Resonance Images in Space and Time,” *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1579-1590, 2003.
- [81] M. R. Hajiaboli, “An Anisotropic Fourth-Order Partial Differential Equation for Noise Removal, Springer,” in: Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision, pp. 356-367, 2009.

- [82] M. Kuwahara, K. Hachimura, S. Ehiu, and M. Kinoshita, "Processing of ri-angiocardigraphic images," (Digital Processing of Biomedical Images. New York: Plenum, pp. 187–203, 1976).
- [83] G. Papari, N. Petkov, P. Campisi, "Artistic Edge and Corner Enhancing Smoothing," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2449-2469, 2007.
- [84] R. C. Gonzalez, R. E. Woods, "Digital image processing," (3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2006).
- [85] R. Oktem, K. Egiazarian, I. Aizenberg, N. Aizenberg, "Transform Domain Denoising Using Nonlinear Filtering and Cellular Neural Networks," in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 1998.
- [86] J. Xie, L. Xu, E. Chen, "Image denoising and inpainting with deep neural networks," in: Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), pp. 341–349, 2012.
- [87] Y. Chen, T. Pock, "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256-1272, 2017.
- [88] V. Jain and S. Seung, "Natural image denoising with convolutional networks," *In Advances in Neural Information Processing Systems*, vol. 21, pp. 769–776, 2009.
- [89] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142-3155, 2017.
- [90] K. Arakawa, "Fuzzy rule-based image processing with optimization," *Fuzzy Techniques in Image Processing. Studies in Fuzziness and Soft Computing, Physica, Heidelberg*, vol. 52, pp. 222–247, 2000.
- [91] F. Farbiz and M. B. Menhaj, "A fuzzy logic control based approach for image filtering," *Fuzzy Techniques in Image Processing. Studies in Fuzziness and Soft Computing, Physica, Heidelberg*, vol. 52, pp. 194–221. 2000.
- [92] I. Kalaykov, G. Tolt, "Real-time image noise cancellation based on fuzzy similarity," *Fuzzy Filters for Image Processing. Studies in Fuzziness and Soft Computing, Springer, Berlin, Heidelberg*, vol. 122, pp. 54–71, 2003.
- [93] A. Kethwas, B. Jharia, "Image de-noising using Fuzzy and Wiener filter in Wavelet domain," in: Proceedings of the IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015.
- [94] H. K. Kwan, "Fuzzy filters for noise reduction in images," *Fuzzy Filters for Image Processing. Studies in Fuzziness and Soft Computing, Berlin, Heidelberg*, vol. 122, pp. 25–53, 2003.
- [95] M. González-Hidalgo, S. Massanet, A. Mir, D. Ruiz-Aguilera, "Gaussian Noise Reduction Using Fuzzy Morphological Amoebas," *Springer, International Conference on*

*Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 660-671, 2016.

[96] B. De Baets, “Fuzzy morphology: a logical approach,” *Uncertainty Analysis in Engineering and Science: Fuzzy Logic, Statistics, and Neural Network Approach*, vol. 11 of the series International Series in Intelligent Technologies, pp. 53–68, 1998.

[97] M. Nachtgael, E. E. Kerre, “Classical and fuzzy approaches towards mathematical morphology,” *Fuzzy techniques in image processing, Chapter 1. Studies in Fuzziness and Soft Computing*, Physica-Verlag, New York, vol. 52, pp. 3–57, 2000.

[98] M. Shafiee, M. R. Karami, K. Kangarloo, “Denoising by Averaging Reconstructed Images: Using Singularity Function Analysis,” in: Proceedings of the IEEE 8th Conference on Machine Vision and Image Processing (MVIP), 2013.

[99] J. J. Kivinen, E. B. Sudderth†, M. I. Jordan, “Image Denoising with Nonparametric Hidden Markov Trees,” in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2007.

[100] Suyash P. Awate and Ross T. Whitaker, “Image denoising with unsupervised, information-theoretic, adaptive filtering,” Whitaker Scientific Comput. Imaging Inst., School of Comput., Univ. Utah, Salt Lake, UT, Tech. Rep. UUCS-04-013, 2004.

[101] G. Chen, T. D. Bui, K. G. Quach, S. E. Qian, “Denoising hyperspectral imagery using principal component analysis and block-matching 4D Filtering,” *Can. J. Remote Sens.*, vol. 40, no. 1, pp. 60-66, 2014.

[102] H. Zhang, W. He, L. Zhang, H. Shen, Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729\_4743, 2014.

[103] J. Yang, Y. Q. Zhao, J. C. W. Chan, S. G. Kong, “Coupled sparse denoising and unmixing with low-rank constraint for hyperspectral image,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1818\_1833, 2016.

[104] L. Sun, B. Jeon, Y. Zheng, Z. Wu, “Hyperspectral image restoration using low-rank representation on spectral difference image,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1151\_1155, 2017.

[105] W. He, H. Zhang, L. Zhang, H. Shen, “Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178-188, 2016.

[106] J.L. Starck, E.J. Candes, D.L. Donoho, “The Curvelet transform for image denoising,” *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, 2002.

[106] M.N. Do, M. Vetterli, “The contourlet transform: an efficient directional multi-resolution image representation,” *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, 2005.

- [108] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [109] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, Alessandro Foi, "Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119-133, 2013.
- [110] A. Hyvarinen, E. Oja, P. Hoyer, J. Hurri, "Image feature extraction by sparse coding and independent component analysis," Presented at the IEEE 4<sup>th</sup> International Conference on Pattern Recognition, 1998.
- [111] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Inform. Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [112] H. Li, G. Ren, B-J Xiao, "Image Denoising Algorithm Based on Independent Component Analysis," Presented at the IEEE International Word Congress on Software Engineering, 2009.
- [113] M. P. Ekstrom, "Digital Image Processing Techniques," (Elsevier Science, 1984).
- [114] S. K. Kopparapu, M. Satish, "Identifying Optimal Gaussian Filter for Gaussian Noise Removal," in: Proceedings of the IEEE 3th International Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, 2011.
- [115] G. Santhanamari, J. S. Varuna Viveka, B. Purushothaman, U. Shanthini, M. Vanitha, "A New Image Denoising Algorithm Based on Adaptive threshold and Fourth Order Partial Diffusion Equation," in: Proceedings of the IEEE International Conference on Computational Intelligence & Computing Research (ICCIC), 2012.
- [116] J. Zhong, H. Sun, "Wavelet-Based Multiscale Anisotropic Diffusion With Adaptive Statistical Analysis for Image Restoration," *IEEE Trans. Circuits Syst.—I*, vol. 55, no. 9, pp. 2716-2725, 2008.
- [117] B. K. Shreyamsha Kumar, "Image denoising based on gaussian/bilateral filter and its method noise thresholding," *Signal, Image and Video Process.*, 7 (6) (2013) 1159-1172.
- [118] L. Lin, K. Lingfu, "Image Denoising Base on Non-local Means with Wiener Filtering in Wavelet Domain," in: Proceedings of the IEEE 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009.
- [119] S. J. You, N. I. Cho, "An adaptive bandwidth nonlocal means image denoising in wavelet domain," *EURASIP J. Image and Video Process.*, 2013.
- [120] N. Bhoi, S. Meher, "Total Variation based Wavelet Domain Filter for Image Denoising," in: Proceedings of the IEEE 1th International Conference on Emerging Trends in Engineering and Technology, 2008.
- [121] S. P. Ghael, A. M. Sayeed, R. G. Baraniuk, "Improved wavelet denoising via empirical wiener filtering," in: Proceedings of the SPIE, Wavelet Applications in Signal and Image Processing, vol. 3169, pp. 389–399, 1997.

- [122] S. Bacchelli, S. Papi, “Image denoising using principal component analysis in the wavelet domain,” *J. Comput. Appl. Mathematics*, vol. 189, no. 1-2, pp. 606–621, 2006.
- [123] R. D. da Silva, R. Minetto, W. R. Schwartz, H. Pedrini, “Adaptive edge-preserving image denoising using wavelet transforms,” *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 567-580, 2013.
- [124] B. Vidakovic, “Nonlinear wavelet shrinkage with Bayes rules and Bayes factors,” *J. American Statistical Association*, vol. 93, no. 441, pp. 173–179, 1998.
- [125] H.Y. Gao, A. G. Bruce, “WaveShrink with firm shrinkage,” *Statistica Sinica*, vol. 7, pp. 855–874, 1997.
- [126] H.Y. Gao, “Wavelet shrinkage denoising using the nonnegative garrote,” *J. Comput. Graphical Statistics*, vol. 7, no. 4, pp. 469–488, 1998.
- [127] A. Antoniadis, J. Fan, “Regularization of wavelet approximations,” *Journal of American Statistical Association*, vol. 96, no. 455, pp. 939–967, 2001.
- [128] Guoquan Xing, Yuxia Zhang, YuYan Ruolin Ruan, “An Exponential Threshold Function Wavelet Denoising Method Based on Gaussian Distribution,” in: Proceedings of the IEEE 12<sup>th</sup> International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 2016.
- [129] D. L. Donoho, I. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, pp. 425–455, 1994.
- [130] D. L. Donoho and I. M. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *J. American Statistical Association*, vol. 90, no. 432, pp. 1200-1224, 1995.
- [131] Thierry Blu, Florian Luisier, “The SURE-LET Approach to Image Denoising,” *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778 - 2786, 2007.
- [132] F. Luisier, T. Blu, “SURE-LET Multichannel Image Denoising: Interscale Orthonormal Wavelet Thresholding,” *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 482-492, 2008.
- [133] H. A. Chipman, E. D. Kolaczyk, E., R. E. McCulloch, “Adaptive Bayesian wavelet shrinkage,” *J. American Statistical Association*, vol. 92, no. 440, pp. 1413–1421, 1997.
- [134] S. G. Chang, B. Yu, M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, 2000.
- [135] G. P. Nason, “Wavelet shrinkage using cross-validation,” *J. Roy Statistic Society (B)*, vol. 58, pp. 463–479, 1996.
- [136] N. Weyrich, G. T. Warhola, “Wavelet shrinkage and generalized cross validation for image denoising,” *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 82–90, 1998.

- [137] L. Zhang, J. Chen, T. Zhu, “Image denoising based on iterative generalized cross-validation and fast translation invariant,” *J. Vis. Commun. Image Rep.*, vol. 28, pp. 1–14, 2015.
- [138] S. G. Chang, M. Vetterli, “Spatial Adaptive Wavelet Thresholding for Image Denoising,” in: Proceedings of the IEEE International Conference on Image Processing (ICIP), vol. 2, pp. 374–377, 1997.
- [139] S. G. Chang, B. Yu, M. Vetterli, “Spatially adaptive wavelet thresholding with context modeling for image denoising,” *IEEE Trans. Image Process.*, vol. 9, pp. 1522–1531, 2000.
- [140] L. Sendur, I. W. Selesnick, “A Bivariate Shrinkage Function for Wavelet-based Denoising,” in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002.
- [141] L. Sendur, I. W. Selesnick, “Bivariate Shrinkage Functions for Wavelet-Based Denoising Exploiting Interscale Dependency,” *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2744 - 2756, 2002.
- [142] L. Sendur and Ivan. W. Selesnick, “Bivariate shrinkage with local variance estimation,” *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 438–441, 2002.
- [143] D. Min, Z. Jiuwen, M. Yide, “Image denoising via bivariate shrinkage function based on a new structure of dual contourlet transform,” *Signal Process.*, vol. 109, pp. 25–37, 2015.
- [144] R. G. Baraniuk, “Optimal tree approximation with wavelets,” In: Proceeding SPIE Tech. Conference on Wavelet Applications in Signal and Image Processing VII, 3813, pp. 196-207, 1999.
- [145] P. Moulin and J. Liu, “Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors,” *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 909-919, 1999.
- [146] J. K. Romberg, H. Choi and R. G. Baraniuk, “Bayesian wavelet domain image modeling using hidden Markov models,” *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1056-1068, 2001.
- [147] M. Malfait and D. Roose, “Wavelet based image denoising using a Markov Random Field a priori model,” *IEEE Trans. Image Process.*, vol. 6, no. 4, pp. 549–565, 1997.
- [148] M. Lang, H. Guo, J.E. Odegard, and C.S. Burrus, R. O. Wells, “Nonlinear processing of a shift invariant DWT for noise reduction,” in: Proceedings SPIE Conference on Wavelet Applications, 2491, 1995.
- [149] H. Haijuan, L. Bing, and L. Quansheng, “Removing Mixture of Gaussian and Impulse Noise by Patch-Based Weighted Means,” *J. Sci. Comput.*, vol. 67, no. 1, pp. 103-129, 2016.

- [150] T. J. Su and C. I. Li, "An Adaptive Filtering Method for Mixed Noise of Images," in: Proceedings: IEEE International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 2012.
- [151] Y.M. Huang, M. K. Ng, and Y. W. Wen, "Fast Image Restoration Methods for Impulse and Gaussian Noises Removal," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 457-460, 2009.
- [152] P. Rodr'iguez, R. Rojas, and B. Wohlberg, "Mixed Gaussian-impulse Noise Image Restoration Via Total Variation," in: Proceedings IEEE International Conference on Acoustics, Speech, and Signal process., Kyoto, Japan, 2012.
- [153] S. K. Agarwal and P. Kumar, "Denoising of A Mixed Noise Color Image using New Filter Technique," in: Proceedings IEEE International Conference on Computational Intelligence and Communication Networks, Jabalpur, India, 2015.
- [154] J. Arnal, M. G. Sanchez, and V. Vidal, "Parallel Filter for Mixed Gaussian-Impulse Noise Removal," in: Proceedings IEEE International Conference on Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan, Poland, 2013.
- [155] S. Morillas, V. Gregori, and A. Hervás, "Fuzzy Peer Groups for Reducing Mixed Gaussian-Impulse Noise From Color Images," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1452-1466, 2009.
- [156] M. Jayasree and N. K. Narayanan, "A Novel Fuzzy Filter for Mixed Impulse Gaussian Noise from Color Images," International Conference on Signal, Networks, Computing, and Systems, pp. 53-59, 2016.
- [157] Y. Sun, H. Junwei, and Lu Jun, "An Information-fusion Edge Preserving Method in Image Filtering," in: Proceedings 16<sup>th</sup> IEEE Conference on Wireless Communications Networking and Mobile Computing, Chengdu, China, Sep. 2010
- [158] X, Guo and B, Guo, "A fuzzy filter for color images corrupted by mixed noise," in: Proceedings IEEE International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, China, Oct. 2014.
- [159] S. Chankhachon and S. Intajag, "Resourceful Method to Remove Mixed Gaussian-Impulse Noise in Color Images," in: Proceedings 12<sup>th</sup> IEEE International Conference on Computer Science and Software Engineering, Songkhla, Thailand, Jul. 2015.
- [160] J. G. Camarena, V. Gregori, S. Morillas, and A. Sapena, "A simple fuzzy method to remove mixed gaussian-impulsive noise from color images," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 971-978, 2013.
- [161] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," in: Proceedings of the IEEE International Conference, vol. 78, no. 4, pp. 678-689, 1990.
- [162] Y. Xiao, T. Zeng, J. Yu, and M. K. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via l1-l0 minimization," *Pattern recognit.*, vol. 44, pp. 1708-1720, 2011.

- [163] M. Filipović, and A. Jukić, “Restoration of Images Corrupted by Mixed Gaussian-Impulse Noise by Iterative Soft-hard Thresholding,” *Pattern Recognit.*, vol. 44, no. 8, pp. 1708-1720, 2011.
- [164] J. Jiang, L. Zhang, and J. Yang, “Mixed Noise Removal by Weighted Encoding With Sparse Nonlocal Regularization,” *IEEE Trans Image Process.*, vol. 23, no. 6, pp. 2651-2662, 2014.
- [165] M. Yan, “Restoration of Images Corrupted by Impulse Noise and Mixed Gaussian Impulse Noise using Blind Inpainting,” UCLA CAM Rep., pp. 11-72, 2011.
- [166] J. Jiang, J. Yang, Y. Cui, and L. Luo, “Mixed noise removal by weighted low rank model,” *Neurocomput.*, vol. 151, part 2, pp. 817-826, 2015.
- [167] H. Ji, C. liu, Z. Shen, and Y. Xu, “Robust video denoising using low rankmatrix completion,” in: Proceedings IEEE 23<sup>th</sup> International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.
- [168] I. markosky, “Low Rank Approximation: Algorithms, Implementation, Applications,” *Commun. Control Eng.*, 2012.
- [169] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in: Proceedings 27<sup>th</sup> International Conference on Mach. Learn., 2010.
- [170] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210-227, 2009.
- [171] T. Huang, W. Dong, X. Xie, G. Shi, and X. Bai, “Mixed Noise Removal via Laplacian Scale Mixture Modeling and Nonlocal Low-rank Approximation,” *IEEE Trans. image Process.*, vol. 26, no. 7, pp. 3171-3186, 2017.
- [172] N. Eslahi, H. Mahdavinataj, and A. Aghagolzadeh, “Mixed Gaussian-Impulse Noise Removal from Highly Corrupted Images via Adaptive Local and Nonlocal Statistical Priors,” in: Proceedings 9<sup>th</sup> IEEE Iranian Conference on Machine Vision and Image Process., Tehran, Iran, 2015.
- [173] J. Delon, A. Desolneux, T. Guillemot, “PARIGI: a Patch-based Approach to Remove Impulse-Gaussian Noise from Images,” *Image Process. On Line*, 5, 130–154, 2016.
- [174] R. P. Aher , and K. C. Jodhanle, “Removal of Mixed Impulse Noise and Gaussian Noise Using Genetic Programming,” in: Proceedings IEEE International Conference on Signal Process., Beijing, china, Oct. 2012.
- [175] J. R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection,” (Cambridge, MA,USA: MIT Press, 1992).
- [176] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, “Genetic Programming: An Introduction—On the Automatic Evolution of Computer Programs and Its Applications,” San Mateo, CA, USA: Morgan Kaufmann Publishers, 1998.

- [177] Bishnu and P. Lamichhane, "Finite Element Techniques for Removing the Mixture of Gaussian and Impulsive Noise," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2538-2547, 2009.
- [178] P. Ciarlet, "The Finite Element Method for Elliptic Problems," (Amsterdam, The Netherlands: North-Holland, 1978).
- [179] S. Brenner and L. Scott, "The Mathematical Theory of Finite Element Methods," (New York: Springer-Verlag, 1994).
- [180] A. Quarteroni and A. Valli, "Numerical Approximation of Partial Differential Equations," (Berlin, Germany: Springer-Verlag, 1994).
- [181] J. D. Mendiola-Santibañez, and I. R. Terol-Villalobos, "Filtering of mixed Gaussian and impulsive noise using morphological contrast detectors," *IET Image Process.*, vol. 8, no. 3, pp. 131-141, 2014.
- [182] Y. Shen, B. Han, and E. Braverman, "Removal of Mixed Gaussian and Impulse Noise Using Directional Tensor Product Complex Tight Framelets," *J. Math. Imag.*, vol. 54, no. 1, pp. 64-77, 2016.
- [183] B. Han, "Properties of discrete framelet transforms," *Math. Model. Nat. Phenom.*, vol. 8, pp. 18-47, 2013.
- [184] Mehdi Mafi, Harold Martin, Malek Adjouadi, "High Impulse Noise Intensity Removal in MRI Images", IEEE Signal Processing in Medicine and Biology Symposium (SPMB17)," Philadelphia, PA, USA, 2017.
- [185] Mehdi Mafi, Hoda Rajaei, Mercedes Cabrerizo, Malek Adjouadi, "A Robust Edge Detection Approach in the Presence of High Impulse Intensity through Switching Adaptive Median and Fixed Weighted Mean Filtering," *IEEE Transaction on Image Processing*, vol. 27, issue. 11, 2018, pp. 5475-5490.
- [186] T. Veerakumar, S. Esakkirajan, I. Vennila, "An Approach to Minimize Very High Density Salt and Pepper Noise through Trimmed Global Mean," *Int. J. Comput. Appl.*, vol. 39, no. 12, 2012.
- [187] P. S. Jayasree, P. Raj, P. Kumar, R. Siddavatam, S. P. Ghrera, "A Fast Novel Algorithm for Salt and Pepper Image Noise Cancellation using Cardinal B-Splines," *Signal Image and Video Process. J.*, vol.7, no. 6, pp.1145-1157, 2013.
- [188] J. L. Mateo, A. Fernandez-Caballero, "Finding out general tendencies in speckle noise reduction in ultrasound images," *Expert Syst. Appl.*, vol. 36, no.4, pp.7786-7797, 2009.
- [189] Z. Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, 2004.
- [190] V. D. Heyden, "Evaluation of edge detection algorithms," *IET 3th Conf. Image Process. its Appl.*, pp.618-622, 1989.

- [191] M. C. Shin, D. B. Goldgof, K. W. Bowyer, S. Nikiforou, "Comparison of Edge Detection Algorithms Using a Structure from Motion Task," *IEEE Trans. System, Man, and Cybernetics—Part B: Cybernetics*, vol. 31, pp. 589-601, 2001.
- [192] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.PAMI-8, no.6, 1986.
- [193] S. Xie, Z. Tu, "Holistically-nested edge detection," In *Proc. IEEE Int. Conf. Comput. Vision*, pp. 1395–1403, 2015.
- [194] M. E. Yukse, "Edge detection in noisy images by neuro-fuzzy processing," *Int. J. Electron. Commun.* vol.61 pp.82 – 89, 2007.
- [195] Y. Shi, Q. Zhao, F. Guo, Y. Zhu, "A Fast Edge Detection Model In Presence of Impulse Noise," in: Proceedings Springer 8<sup>th</sup> International Conference on Image and Graphics, Tianjin, China, vol.9217, pp. 64-74, 2015.
- [196] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no.1, pp. 62-66, 1979.
- [197] P. Singh, L. Jain, "Noise reduction in Ultrasound images using Wavelet and Spatial filtering Techniques," in: Proceedings IEEE 2<sup>th</sup> Int. Conf. Information Management in the Knowledge Economy, Chandigarh, India, 2013.
- [198] R. Vanithamani, G. Umamaheswari, "Wavelet based Despeckling of Medical Ultrasound Images with Bilateral filter," IEEE Region 10 Conf. TENCN, Bali, Indonesia, 2011.
- [199] Y.M. Huang, M.K. Ng, Y.W. Wen, "A New Total Variation Method for Multiplicative Noise Removal," *SIAM J. Imaging Sciences*, 2 (1), pp. 20–40, 2009.
- [200] J. Shi, S. Osher, "A nonlinear inverse scale space method for a convex multiplicative noise model," *SIAM J. Imaging Sciences*, 1, pp. 294–321, 2008.
- [201] S. Wang, T.Z. Huang, X.L. Zhao, J.J. Mei, J. Huang, "Speckle noise removal in ultrasound images by first- and second-order total variation," *Numerical algorithms*, pp. 1-21, 2017.
- [202] A. Khare, M. Khare, Y. Jeong, H. Kim, M. Jeon, "Despeckling of medical ultrasound images using Daubechies complex wavelet transform," *Signal process.*, 90, pp. 428-439, 2010.
- [203] P.N. Devi, R. Asokan, "An improved adaptive wavelet shrinkage for ultrasound Despeckling," *Sadhana*, 39 (4), pp. 971-988, 2014.
- [204] B.A. Abraham, Y. Kadah, "Speckle Noise Reduction Method Combining Total Variation and Wavelet Shrinkage for Clinical Ultrasound Imaging," in: Proceedings IEEE 1<sup>st</sup> Middle East Conf. Biomedical Engineering, Sharjah, United Arab Emirates, 2011.

- [205] V. S. Frost, J.A. Stiles, K. S. Shanmugan, J. C. Holtzman, “A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise,” *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-4 (2), pp. 157-166, 1982.
- [206] D.T. Kuan, A. Sawchuk, T. C. Strand, P. Chavel, “Adaptive restoration of images with speckle,” *IEEE Trans. on Acoustics, Speech and Signal Process.*, ASSp-35 (3), pp. 373-383, 1987.
- [207] Y. Yu, S. T. Acton, “Speckle reducing anisotropic diffusion,” *IEEE Trans. Image Process.*, 11 (11), pp. 1260-1270, 2002.
- [208] J. S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2 (2), pp. 165-168, 1980.
- [209] J. S. Lee, “Refined filtering of image noise using local statistics,” *Comput. Vision, Graphics Image Process.*, 15, pp. 380-389, 1981.
- [210] T. R. Crimmins, “Geometric Filter For Reducing Speckle,” *optical engineering*, 25 (5), 1986.
- [211] Mehdi Mafi, Solale Tabarestani, Mercedes Cabrerizo, Armando Barreto, Malek Adjouadi, “Denoising of Ultrasound Images Affected by Combined Speckle and Gaussian Noise,” *IET Image Processing Journal*, vol. 12, no. 12, pp. 2346-2351, 2018.
- [212] I. W. Selesnick, R. G. Baraniuk, N. G. Kingsbury, “The Dual-Tree Complex Wavelet Transform,” *IEEE Signal Proc. Magazin*, 22 (6), pp. 123-151, 2005.
- [213] T. F. Chan, G. H. Golub, P. Mulet, “A nonlinear primal-dual method for total variation-based image restoration,” *SIAM J. Sci. Comput.*, 20 (6), pp. 1964–1977, 1999.
- [214] J. L. Carter, “Dual methods for total variation—Based image restoration,” Ph.D. dissertation, U.C.L.A., 2001.
- [215] L. Zhang, L. Zhang, X. Mou, D. Zhang, “FSIM: A feature similarity index for image quality assessment,” *IEEE Trans. Image Process.*, 20 (8), pp. 2378–2386, 2011.
- [216] J. Cai, R. H. Chan, and M. Nikolova, “Two-phase methods for deblurring images corrupted by impulse plus gaussian noise,” *Inverse Probl. Imaging*, vol. 2, pp. 187-204, 2008.
- [217] H. Ji, S. Huang, Z. Shen, and Y. Xu, “Robust video restoration by joint sparse and low rank matrix approximation,” *SIAM J. Imag. Sci.*, vol. 4, no. 4, pp. 1122–1142, 2011.
- [218] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.
- [219] B. Xiong and Z. Yin, “A universal denoising framework with a new impulse detector and nonlocal means,” *IEEE Trans. Image Process*, vol. 21, no. 4, pp. 1663–1675, Apr. 2012.

- [220] J. F. Cai, R. Chan, and M. Nikolova, “Fast two-phase image deblurring under impulse noise,” *J. Math. Imaging Vis.*, vol. 36, pp. 46-53, 2010.
- [221] R. Garnett, T. Huegerich, C. Chui, and H. Wenjie, “A universal noise removal algorithm with an impulse detector,” *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747–1754, Nov. 2005.
- [222] Mehdi Mafi, Walter Izquierdo, Harold Martin, Malek Adjouadi, “Deep Convolutional Neural Network for Mixed Random Impulse and Gaussian Noise Reduction in Digital Images,” *under review*.
- [223] H. C. Burger, C. J. Schuler, S. Harmeling, “Image denoising: Can plain neural networks compete with BM3D?” in: Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition, RI, USA, pp. 2392–2399, Jun. 2012.
- [224] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Image net classification with deep convolutional neural networks,” in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105, 2012.
- [225] S. Ioffe, C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in: Proceedings of the conference on Machine Learning (ICML), pp. 448–456, Jul. 2015.
- [226] A. Levin, B. Nadler, “Natural image denoising: Optimality and inherent bounds,” in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2833–2840, 2011.
- [227] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in: Proceedings of the conference on Learning Representations, 2015.
- [228] J. Duchi, E. Hazan, Y. Singer, “Adaptive sub gradient methods for online learning and stochastic optimization,” *J. Mach. Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [229] D. Kingma, J. Ba, “Adam: A method for stochastic optimization,” in: Proceedings of the conference on Learning Representations, 2015.
- [230] J. Liu, M. Gong, K. Qin, and P. Zhang “Deep Convolutional Coupling Network for Change Detection Based on Heterogeneous Optical and Radar Images,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 3, pp. 545-559, Mar. 2018.
- [231] W. Shi, Y. Gong, X. Tao, and N. Zheng, “Training DCNN by Combining Max-Margin, Max-Correlation Objectives, and Correntropy Loss for Multilabel Image Classification,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 7, pp. 2896-2908, Jul. 2018.
- [232] W. Shi, Y. Gong, X. Tao, D. Cheng, and N. Zheng, “Fine-Grained Image Classification Using Modified DCNNs Trained by Cascaded Softmax and Generalized Large-Margin Losses,” *IEEE Trans. Neural Netw. Learning Syst.*, Mar. 2018.

- [233] D. Lian, L. Hu, W. Luo, Y. Xu, L. Duan, J. Yu, and S. Gao “Multiview Multitask Gaze Estimation With Deep Convolutional Neural Networks,” *IEEE Trans. Neural Netw. Learning Syst.*, Sep. 2018.
- [234] C. T. Lu, Y. Y. Chen, L. L. Wang, , C. F. Chang, “Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window,” *Pattern Recognition Letters*, vol. 80, pp. 188–199, 2016.
- [235] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [236] U. Schmidt and S. Roth, “Shrinkage fields for effective image restoration,” in: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [237] V.B. Surya Prasath, A. Singh, “Well-posed inhomogeneous nonlinear diffusion scheme for digital image denoising,” *Hindawi, Journal of Applied Mathematic*, Article ID 763847, 2010.
- [238] Y.Q. Wang, J. Guo, W. Chen, W. Zhang, “Image Denoising using Modified Perona-Malik Model based on Directional Laplacian,” *Signal Process*, vol. 93, no. 9, pp. 2548-2558, 2013.
- [239] Mehdi Mafi, Walter Izquierdo, Malek Adjouadi, “High impulse Noise Intensity Removal in Natural Images Using Deep Learning,” *under review*.

## VITA

MEHDI MAFI

Miami, Florida

2010-2012	M.S., Telecommunication Engineering- ICT Amirkabir University of Technology (Tehran polytechnic)
2012-2015	Lecturer at Islamic Azad University of Shar-e-Rey, Tehran, Iran
2013-2015	Board of Trustees of Graduates, Association of the Amirkabir University of Technology (Tehran polytechnic), Tehran, Iran
2015-2019	Ph.D. student, Graduate Assistant/ Graduate Research Fellow, Florida International University, Miami, FL, USA

## SELECTED PUBLICATIONS

M. Mafi, H. Martin, J. Andrian, A. Barreto, M. Cabrerizo, M. Adjouadi, "A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images," *Signal Process.*, vol. 157, pp. 236 -260, 2019.

M. Mafi, S. Tabarestani, M. Cabrerizo, A. Barreto, M. Adjouadi, "Denoising of Ultrasound Images Affected by Combined Speckle and Gaussian Noise," *IET Image Process.*, vol. 12, no. 12, pp. 2346-2351, 2018.

M. Mafi, H. Rajaei, M. Cabrerizo, M. Adjouadi. "A Robust Edge Detection Approach in the Presence of High Impulse Intensity through Switching Adaptive Median and Fixed Weighted Mean Filtering," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5475-5490, 2018.

M. Mafi, H. Martin, M. Adjouadi, "High Impulse Noise Intensity Removal in MRI Images," in: Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium (SPMB17)," Philadelphia, PA, USA, 2017.

M. Mafi, (2012), Introduction to Network Engineering, Taymaz pub., (in Persian).

M. Mafi, (2012), Introduction to C ++ Programming Language, Taymaz pub., (in Persian).

M. Mafi, (2012), New Methods of ICT in Management and Trading, Taymaz pub., (in Persian).

M. Mafi, (2013), New Methods of ICT in Management and Trading, Taymaz pub., (in English).