Version of Record: https://www.sciencedirect.com/science/article/pii/S0165168418303979
Manuscript_aa811b50a14b3f565d457410f338e919

A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images

Mehdi Mafi *, Harold Martin, Mercedes Cabrerizo, Jean Andrian, Armando Barreto, and Malek Adjouadi

Center for Advanced Technology and Education, Department of Electrical and Computer Engineering, Florida International University

A B S T R A C T

This review article provides a comprehensive survey on state-of-the-art impulse and Gaussian denoising filters applied to images and summarizes the progress that has been made over the years in all 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 practical implications and domains of application. A comprehensive comparison is performed to cover all the denoising methods in details and the results they yield. With this extensive review, researchers in image processing will be able to ascertain which of these denoising methods will be best applicable to their research needs and the application domain where such methods are contemplated for implementation.

Key words: Image processing, Impulse noise, Gaussian noise, Denoising filters.

1. Introduction

1.1. Review

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 (salt and pepper) 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 [1-3]. Its noise model is assumed to be 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 [4-6].

$$I(c) = \begin{cases} c_{min} & Probability\ Ps \\ c_{max} & Probability\ Pp \\ c_{min} < c < c_{max} & Probability\ 1 - Pp - Ps \end{cases} \quad (1)$$

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 [5-10]. Accordingly, the noisy image can be expressed as

$$I_n(i,j) = I(i,j) + n(i,j) \quad (2)$$

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

Existing surveys of denoising techniques focus on edge and structure preservation and in the classification results of these denoising filters when applied in the spatial domain, the transform domain, or with regards to their statistical categories [11-18]. The spatial domain filters are further divided into linear (such as mean and Wiener filters) and non-linear (median, anisotropic diffusion, non-local means, and combination of domain and range (bilateral and partial differential equation)) [19-25]. Transform domain filters are further partitioned into curvelet, contourlet, data adaptive (i.e. independent component analysis (ICA) and non-data adaptive (subdivided into two categories)), spatial-frequency domain (low-pass filters), wavelet domain (i.e. linear Wiener filters), shrinkage rules (hard and soft thresholding filters), non-linear non-adaptive and non-linear adaptive thresholding, wavelet coefficients, and non-orthogonal wavelet transform filters [19, 20, 26-51].

1.2. Structural metrics

The following sections will cover the types of filters that can be used to remove impulse and Gaussian noise from images.

* Corresponding author.

Email addresses: mmafi002@fiu.edu (M. Mafi), hmart027@fiu.edu (H. Martin), cabreriz@fiu.edu (M. Cabrerizo), andrianj@fiu.edu (J. Andrian), barreto@fiu.edu (A. Barreto), adjouadi@fiu.edu (M. Adjouadi).

Said filters can also be used to remove other types of noise, to varying degrees, depending on their structure, the filtering procedure, and the intensity of the noise. Their performance can be measured using several structural metrics such as correlation coefficient (β), weighted distance (WD)(using $L_1 - norm$), normalized cross correlation (NK), signal to noise ratio (SNR), mean square error (MSE), peak signal to noise ratio (PSNR), mean absolute error (MAE), image enhancement factor (IEF), structural similarity index (SSIM), universal quality index (QI), average difference (AD), maximum difference (MD), structural content (SC), correlation quality (CQ), image fidelity (IF), Laplacian mean square error (LMSE), peak mean square error (PMSE), normalized absolute error (NAE), normalized mean square error (NMSE), and $L_p - norm$ [52-57]. Equations (3) through (20) formulate these aforementioned structural metrics when used in two-dimensional (2-D) images.

In all of these equations, x represents the original image, y is the denoised image, n is the noisy image, $\bar{x}$ is the mean value of the image, $\bar{y}$ is the mean value of the denoised image, M and N are the image dimensions, σ_x is standard deviation of the original image, σ_y is the standard deviation of the denoised image, and (i, j) are the pixel indices.

$$\beta = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[x(i,j)-\overline{x(i,j)}]\times[y(i,j)-\overline{y(i,j)}]}{\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[x(i,j)-\overline{x(i,j)}]^2\times[y(i,j)-\overline{y(i,j)}]^2}} \quad (3)$$

$$NK = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} xy}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} x^2} \quad (4)$$

$$MSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(x(i,j)-y(i,j))^2}{M\times N} \quad (5)$$

$$PSNR = 10log\frac{(\max(x))^2}{MSE} \quad (6)$$

$$MAE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|(x(i,j)-y(i,j)|}{M\times N} \quad (7)$$

$$IEF = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(n(i,j)-x(i,j))^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(y(i,j)-x(i,j))^2} \quad (8)$$

$$\text{SSIM} = \frac{(2\bar{x}\bar{y}+C1)(2\sigma xy+C2)}{(\bar{x}^2+\bar{y}^2+C1)(\sigma x^2+\sigma y^2+C2)} \quad (9)$$

In equation (9), C_1 and C_2 are two variables which depend on the pixels' dynamic range. Constants c_1=0.01 and c_2=0.03 are standard empirical choices for these measures.

$$\text{QI} = \frac{4\sigma xy\bar{x}\,\bar{y}}{(\sigma x^2+\sigma y^2)+[(\bar{x}^2)+(\bar{y}^2)]} \quad (10)$$

$$AD = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} x(i,j)-y(i,j)}{M\times N} \quad (11)$$

$$MD = max\,|(x(i,j)-y(i,j)| \quad (12)$$

$$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} y(i,j)^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} x(i,j)^2} \quad (13)$$

$$CQ = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} x(i,j)y(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} x(i,j)} \quad (14)$$

$$\text{IF} = 1 - \left(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j)-y(i,j))^2}{\sum_{i=1}^{M}\sum_{j=1}^{N} x(i,j)^2}\right) \quad (15)$$

$$\text{LMSE} = \frac{\sum_{i=1}^{M-1}\sum_{j=2}^{N-1}[F\{x(i,j)\}-F\{y(i,j)\}]^2}{\sum_{i=1}^{M-1}\sum_{j=2}^{N-1}[F\{x(i,j)\}]^2} \quad (16)$$

Where $F\{x(i,j)\} = x(i+1,j) + x(i-1,j),\ x(i,j+1),\ x(i,j-1),\ 4x(i,j)$.

$$\text{PMSE} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j)-y(i,j))^2}{M\times N\sum_{i=1}^{M}\sum_{j=1}^{N}(\max\{x(i,j)\})^2} \quad (17)$$

$$\text{NAE} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\ |F\{x(i,j)\}-F\{y(i,j)\}|}{\sum_{i=1}^{M}\sum_{j=1}^{N}|F\{x(i,j)\}|} \quad (18)$$

Where $F\{x(i,j)\}$ is defined in 3 ways; $x(i,j)$ or $x(i,j)^{\frac{1}{3}}$ or $H\{(u^2+v^2)^{\frac{1}{2}}\}x(u,v)$ (in the cosine transform domain).

$$\text{NMSE} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(F\{x(i,j)\}-F\{y(i,j)\}^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}(\text{F}\{x(i,j)\})^2} \quad (19)$$

Where $F\{x(i,j)\}$ is defined in 3 ways; $x(i,j)$ or $x(i,j)^{\frac{1}{3}}$ or $H\{(u^2+v^2)^{\frac{1}{2}}\}x(u,v)$ (in cosine transform domain).

$$\text{NMSE} = \{\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}|x(i,j)-y(i,j)|^p\}^{\frac{1}{p}} \quad (20)$$

Where $p = 0,1,2,3,....k$

1.3. Block diagram of the proposed classification

The overall filter classification block diagram is as shown in Fig. 1. Impulse and Gaussian denoising filters are further detailed in sections II and III, respectively.

2. Impulse Denoising Filters

2.1. 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. The filter output is a non-linear function of its input(s).

2.1.1. Median 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 [19, 20].

2.1.1.1. Adaptive filters

Adaptive median filters (AMF) start by computing the median of the moving window by using all but its center pixel [58]. Then a determination is made whether the median value and the center pixel are within the maximum and minimum values of the selected window. If the center pixel is equal to the maximum or minimum window's value, it is replaced by the median value. However, if the median is also equal to one of such extrema values, the window size is consequently increased, and the process is repeated from the beginning.

The filter proposed in [59] involving "Median-type Noise Detectors and Detail-preserving Regularization", consists of

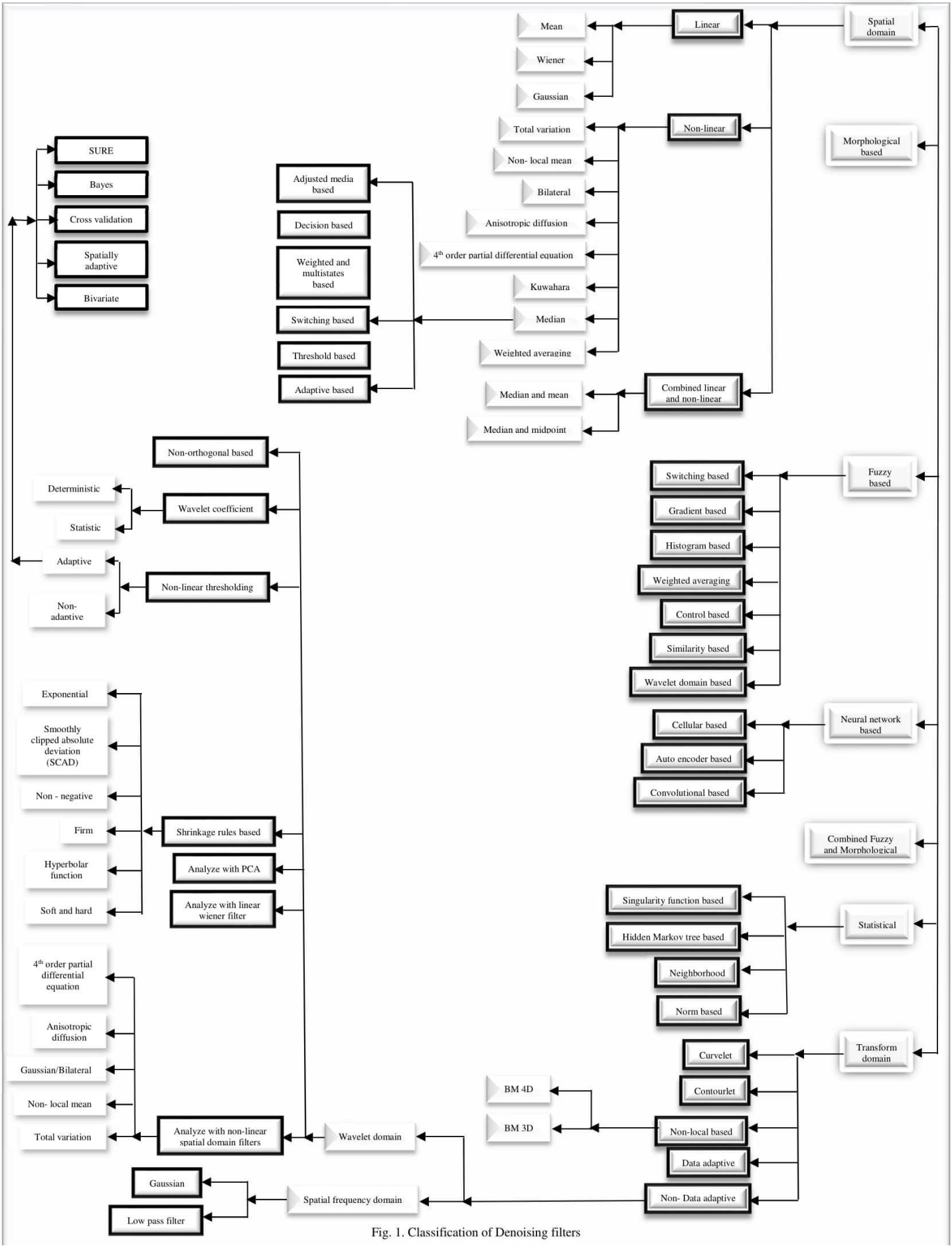

Fig. 1. Classification of Denoising filters

two phases: noise detection and noise replacement. The noise detection phase applies the AMF filter described in [58] and the replacement phase introduces an iterative minimization, based on a variation of methods [60] and [61], to restore the noisy image. This minimization uses an edge preserving potential function (21) in order to preserve a larger amount of edges [62].

$$\varphi = \sqrt{\alpha + t^2} \quad \alpha > 0, \qquad \varphi = |t|^2 \quad 1 < \alpha \leq 2 \qquad (21)$$

The value of α as detailed in [62] is selected as a trade-off value between noise suppression and detail preservation, and the function φ is increasing as a function of t and is even as gradients could be both positive or negative. The authors of [59] claim to obtain better results than those obtained when using either the progressive median filter (PSMF) [63], the multi states median filter (MSMF) [64], the noise adaptive soft switching median filter (NASMF) [65], the directional-difference based switching median filter (DDBSMF) [66], or the improved switching median filter (ISMF) [67].

Simple adaptive median filters (SAMF) have noise detection and noise cancelation phases [68]. The noise detection phase marks the noisy pixels in $\alpha(x, y)$ by finding the maximum and minimum values of the dynamic range of the image. These noisy pixels are set to 1 (in $\alpha(x, y)$) and the noise-free pixels are set to 0 [59]. Lastly, the cancellation phase outputs the filtered pixel values as define by (22)

$$y_{ij} = [1 - \alpha(x, y)]f(x, y) + \alpha(x, y).m(x, y) \qquad (22)$$

where $f(x, y)$ is the input image and $m(x, y)$ is the median value of the selected window ($W = 2R_{min} + 1$), with R_{min} being a small integer given as follows:

$$R_{min} = \frac{1}{2}\sqrt{\frac{7}{1-\eta}} \qquad (23)$$

Where η represents the noisy pixels ratio over the entire image. If the number of noise-free pixels in the selected window is less than 8, the size of the window is increased by 2 and the median value is updated.

2.1.1.2. Threshold filters

Threshold filterers, as their name implies, use predefined thresholds in their noise detection phase. The pixelwise Median Absolute Deviation filter (PWMAD) [69] uses a moving window for which it calculates the median and the absolute deviation (by subtracting each pixel in the selected window from median). With these values, the median of all the absolute deviations is calculated, as an estimator of the variance, to better estimate image details. Finally, the pixel-wise MAD values are iteratively subtracted from the absolute deviations. In [70], the absolute difference between the window's center pixel and the median is compared against a predefined threshold and in [71] the absolute difference between the window's center pixel and the rank order is the one compared against said threshold. However, [69] compares the result of the aforementioned iterative process against a threshold and the restored noisy pixels are obtain as per (22). If the output of the iterative process is greater than the threshold, $\alpha(x, y)$ is set to 1, otherwise $\alpha(x, y)$ is set to 0.

2.1.1.3. Switching filters

Filters that use a switching process to select the optimal output, either in the noise detection phase or in the noise correction phase, are considered switching filters. The filter proposed in [63] (PSMF) is made up of two stages: impulse detection and noise filtering. In the n^{th} iteration of each pixel, the median value of the selected window (not smaller than 3×3) is calculated and if the absolute difference between the median and the current pixel value is smaller than the predefined threshold, the pixel is considered uncorrupted, otherwise, it is corrupted. The noise filtering phase is similar to the impulse detection phase. However, this phase overwrites corrupted pixels with the median of the selected window, if the number of uncorrupted pixels (M) within said window is not even or is 0, or with $\frac{M_L + M_R}{2}$, where M_L is the left median ($\frac{M}{2}$ largest value) and M_R is the right median ($\frac{M}{2} + 1$ largest value), otherwise. As in previous filters, uncorrupted pixels are left unchanged.

The authors of [65] also propose a switching filter (NASMF) implemented in three phases and using three different filter types ("non-filtering", "standard median filtering", and "fuzzy weighted median filter"). Non-filtering is applied to uncorrupted pixels and standard median filtering with fuzzy weighted median filters are used to detected noisy pixels and edges. The first noise detection phase determines the corrupted and uncorrupted pixels by using four equal rectangular blocks (from Weber-Fechner Law [72]). If the absolute differences between the average pixel intensities of each block is greater than the threshold ($T = \frac{255}{32}$), the current block is non-homogeneous, and it should be further subdivided. Subsequent blocks are further subdivided as long as they are non-homogeneous, and their size is larger than 8x8. The corrupted pixels of each block lie within an interval around the center point of the estimated error (pixelwise difference between noisy image and estimated image) histogram. The second noise detection phase of the algorithm determines the isolated and non-isolated impulse noise by using the membership value (μ) of two groups ("low" (uncorrelated) and "high" (correlated)) of the uncorrupted neighboring pixels around the center of the selected window. If the ratio of $\frac{\mu_{low}}{\mu_{high}}$ is between 1/3 and 3, the pixel is isolated, otherwise it requires further discrimination on non-isolated impulse noise and edge pixels. The last step (third phase) is to check whether the number of correlated pixels in the enlarged decision window is greater than half of the total number of uncorrupted pixels in that selected window; if so, the pixel under consideration is determined to be an edge.

Directional difference-based switching median filters (DDBSMF) [66] calculate the absolute differences between south and north, southwest and northeast, west and east, northwest and southeast, and pick the direction of the minimum absolute differences. The sum of the pixels in that direction is divided by 2 and labeled (S). If the difference between the original pixel value and S is greater than the predefined threshold, the pixel being processed is replaced by S, otherwise it remains unchanged.

Impulse detector switching median filters (ISMF) [67] introduce an approach based on Laplacian operators and use median filters to remove the impulse noise. They use 4 single dimensional Laplacian kernels and calculate the minimum absolute value of the resulting image convolutions with said kernels. If result of the convolution is larger than a predefined threshold the pixel being processed is considered to be corrupted and is replaced with the median, otherwise it is uncorrupted, and its value remains unchanged.

The adaptive switching median filter (ASMF) proposed in [73] uses both detection and correction phases. The first phase computes the median value at the n^{th} iteration (stopped at 3^{th} iteration as proposed by the paper) of the selected window. The pixel at the center of the window is not considered to be noisy only if the previously computed median is between the maximum and minimum values of the current window and the absolute difference between said pixel and the median is less than a predefined threshold. The second phase starts by calculating the median value of the uncorrupted pixels in the window (S^n), defined as a sequence of windows of size ($W^n \times W^n$) where $W^n = 3+2n$. Pixels are considered corrupted and replaced by the median value if W^n is greater than or equal to a predefined maximum window size (9 is the value proposed in the paper) or the number of uncorrupted pixels within said window is greater than a predetermined integer value (associated with each $S^n(q^n)$). If the pixel under processing is not determined to be corrupted the window size is increased by 1 (n=n+1) and the process is repeated.

The sorted switching median filter (SSMF) in [74] uses the image's histogram. It consists of three stages: detection, sorting, and filtering. The detection phase finds the upper and lower bounds of the corrupted image, based on its histogram, and checks whether the pixel's intensity is within those bounds (uncorrupted). The sorting stage sorts the uncorrupted pixels within the selected window for the filtering phase to compute their median value and uses it to replace the corrupted pixels.

2.1.1.4. Weighted and multi-states filters:

Weighted and multi-states-based filters use different threshold comparison and consequently different states for noise detection and correction. The Tri-states median filter (TSMF) introduced in [75] is a combination of a centered weighted median filter (CWMF) [76] and a standard Median filter. It compares the input pixels and the output of the standard median filter to mark each pixel with one of three possible states using two different thresholds. Corrupted pixels are identified based on these three states and are replaced with the centered weighted median filter (WMF) results or the standard median filter while uncorrupted pixels are left unchanged.

MSMF [64] is an adaptive filter and a modified version of the center weighted median filter (CWMF) introduced in [76]. It defines comparison thresholds as the absolute difference between the median values of pixels obtained from the CWM filter (with different center weights) [76] and the center pixel of the selected window. The three states used by MSMF are defined based on the comparison between the absolute differences and a predefined threshold. Finally, the corrupted pixels are replaced by the median values obtained from CWMF.

The directional weighted median filter (DWMF) proposed in [77] calculates the sum of absolute differences between the gray scale values of the pixels along four directions (north-south, northeast-southwest, northwest-southeast, and east-west) in the selected window. It sets weights (w) for all absolute differences, w=2 for the two closest pixels and w=1 for others. Finally, if the minimum weighted absolute difference is less than a predefined threshold (T) the pixel at the center of the window is considered noisy and its replacement value is obtained according to (22) where m is the weighted median value of the pixels within the window (excluding the center) and α is 1. The values of non-noisy pixels are also adjusted according to (22), but α is 0. A decreasing threshold is also used to ensure the accuracy of the detection, making this an iterative process [67].

The directional weighted median filter (DWMF) described in [78] corrects the center pixel of a 7×7 sliding window. It does so by checking if its value is 0 or 255, in which case it is considered noisy. The filter then checks pixels considered noisy to determine whether they are edges, in which case they would need no further processing. The remaining noisy pixels are checked to determine if the smallest absolute distance along twelve different directions is less than or equal to a predefined iterative threshold (T). If the test succeeds, the pixel under consideration is determined noise-free. Once these checks are completed and a noise/not-noise mask is created, the image is filtered using the median filter described by equation (24), where $f(x, y)$ is the weighted median value of the smallest absolute distance along the twelve different predefined directions. This filter iteratively decreases the value of the predefined threshold (T).

$$y_{ij} = [1 - \alpha_{ij}]\hat{m}_{ij} + \alpha_{ij}.f_{ij} \quad (24)$$

where y_{ij} is the resulting denoised pixel, f_{ij} is the input pixel, $\alpha_{ij} = 1$ for noise-free pixels and $\alpha_{ij} = 0$ for noisy pixels, and $\hat{m}_{ij}$ is the directional-weighted-median filtered pixel.

2.1.1.5. Decision filters

Decision filters assume that corrupted pixels have a value of 0 or 255 and uncorrupted ones have a value between them. Decision based median filters, as proposed in [79], detect the corrupted pixel within the selected window and replace it with the median value of the window, whose size is proportional to the noise intensity. Some modifications on decision-based filters are new based decision Algorithm (NEDBF) [80] and decision base unsymmetrical median filter (DBUTMF) [79]. NEDBF scans the image by using a window of a predefined size and sorts the values within the window from lowest to highest while finding the minimum, maximum, and median values. If the center pixel is between minimum and maximum values, it is left unchanged. However, if it is 0 or 255 and not equal to the median, it is replaced with the median of the window, otherwise (0 or 255 and equal to the median) it is replaced with the neighborhood value. DBUTMF selects a window and detects the corrupted pixels while eliminating 0`s and 255`s from it; corrupted pixels are replaced with the

median value of the remaining (not equal to 0 or 255) pixels within the window.

2.1.1.6 Adjusted median filters

These types of filters use an adjusted median value to replace the corrupted pixels. The boundary discriminative noise detection filter (BDNDF) described in [81] performs up to two correction iterations. This filter starts by using a 21x21 moving window and if a pixel is considered corrupted it repeats its detection steps using a 3×3 window to confirm its findings. The following steps are used to detect the corrupted pixels: 1) the pixels within a 21×21 moving window are sorted in a vector (V_0) and the median value is computed. 2) The difference between adjacent elements of the sorted vector V_0 is computed and stored in a new vector (V_D). 3) A first boundary, b1, is defined as the point between 0 and the median (from V_0) after which the highest intensity change occurs (according to the adjacent differences stored in V_D). 4) The second boundary, b2, is defined as the point between the median and the maximum value (from V_0) after which the highest intensity change occurs (according to the adjacent differences stored in V_D). These two boundaries subdivide V_0 into three sets: the *lower-intensity cluster* [0, b1), the *middle-intensity cluster* [b1, b2], and the higher-intensity cluster (b2, 255]. 5) If the pixel being processed belongs to the middle-intensity cluster it is considered uncorrupted and the algorithm moves on to the next pixel. Otherwise, the algorithm imposes a 3×3 window (centered on it) and repeats steps 1 through 5. In the filtering phase, the filter starts with a 3x3 window and iteratively increments its size by 1 (in all sides) if the number of uncorrupted pixels within it is less than or equal to half the number of elements in a window of maximum size (according to [59]). Once a final window size is determined, the under-processed pixel is replaced with the adjusted median (the median computed excluding corrupted pixels) of pixels in the selected window.

An improvement on BDNDF (IBDNDF) [82] is to check the number of uncorrupted pixel (N) in the selected window, and if it equals zero or less than half the total number of pixels multiplied by the percentage of the uncorrupted pixel (1-noise level), where the window size is less than the predefined maximum, the size of the window is increased and the center pixel (i, j) is replaced with an adjusted median as in (25).

$$y_{i,j} = M + \frac{1}{D} \times \sum_{k=1}^{N} \frac{V_u(k) - M}{d(k)}, D = \sum_{m=1}^{N} \frac{1}{d(m)} \quad (25)$$

where V_u is a set of uncorrupted pixels, $d(k)$ is the distance between the pixels of k[th] value in the set and the pixel being processed, and M is the median value.

2.1.2. Weighted averaging filters

Weighted averaging filters employ a multicriteria weighted mean value to correct the corrupted pixels. The switching adaptive weighted mean filter (SAWMF) in [83] detects the noise in the center of the moving window using directional differences (dividing the window into four sub-windows). If the lowest absolute weighted mean of differences among the four quadrants is more than or equal to the predefined threshold, the center pixel is considered to be noisy. Once the noisy pixels are detected, a weighted mean of noise free pixels, based on the sum of degree of compatibility [84] between the center pixel and its neighbors in the selected window, is calculated and later used to replace the corrupted pixels. The authors of SAWMF also use the idea in [76] on weighted median filters with added improvement to increase the size of the window. Finally, the output of the filter is obtained according to (22) for which m is the weighted mean of the noise free pixels and α is 1 for noisy pixels and 0 for the noise-free pixels.

An Adaptive Weighted Mean Filter (AWMF), as proposed in [85], uses a moving window for which minimum, maximum, and weighted mean value (excluding pixels whose value is less than minimum and greater than maximum) are computed. If the extrema values of the selected window are not equal to those of the window with the next greater size, the size of the current window is increased by 1 (in each direction) and the process is repeated until the conditions are met or the maximum size, 39, is reached. Once the proper window size is obtained, the center pixel is replaced by the window's mean if its value is not between the extrema values, otherwise, it is left unchanged as it is deemed not corrupted.

Unbiased Weighted Mean Filters (UWMF), introduced in [86], identify the corrupted pixels using a moving window, and if all the pixels within the window are corrupted, they are replaced with 0 or 255, depending on the size of the window. However, if at least one pixel is uncorrupted, the corrupted pixels are replaced with the recalibrated weighted value of the uncorrupted pixel(s) in the window based on a spatial bias (Minkowski distance and spatial distance in x and y direction). The method presented in [86] claims to achieve better performance than the improved boundary discriminative noise detection filter (IBDND) [82], cloud model filter (CMF) [87], and the interpolation-based impulse noise filter (IBINRF) [88].

The method proposed in [87] is based on a cloud generator [89, 90] and has also two phases (noise detection and noise replacement). The first phase checks whether the central pixel ($x_{i,j}$) in the selected window, of size of $2N + 1$ (starting with N=0), is in the interval between $min\,(S_{min}, E_x + 3E_n)$ and $max\,(S_{max}, E_x - 3E_n)$, and if so it is considered uncorrupted; otherwise, the other pixels in the window are checked. If the number of uncorrupted pixels (n) is less than the predefined threshold, the size of the window is increased by 1 and the process is once more repeated; otherwise, the center pixel is considered corrupted. S_{min} and S_{max} are respectively the minimum and maximum values within the selected window, where E_x is the expectation value, and E_n is the entropy as defined in (26) and (27).

$$E_x = \frac{1}{n} \sum_{x_{i+s,j+t} \in W_{i,j}^{2N+1}} x_{i+s,j+t} \quad (26)$$

$$E_n = \sqrt{\frac{\pi}{2}} \times \frac{1}{n} \sum_{x_{i+s,j+t} \in W_{i,j}^{2N+1}} \left| x_{i+s,j+t} - E_x \right| \quad (27)$$

$W_{i,j}^{2N+1} = \{(i + s, j + t) | -N \leq s, t \leq N\}$ represents the weighted mean. The exponential weighted value replaces the corrupted pixel.

The interpolation-based impulse noise removal filter (IBINRF) in [88] detects the corrupted pixels within the

moving window and replaces them with 0 or 255 depending on how many there are. If an uncorrupted pixel exists within the window, the corrupted pixels are replaced with a weighted average of the pixels in the window based on their Euclidean distance from the central pixel. The weight of pixel (i,j) is determined in this case as:

$$\varphi_{x,y,i,j} = \frac{1}{\|(i,j)-(x,y)\|^k} \quad (28)$$

Where k is 4 and (x,y) is the processed pixel coordinates.

The Modified directional weighted filter (MDWF) presented in [91] is composed of two stages: 1) noise detection, and 2) signal/image restoration. The first stage determines any given pixel to be noisy if the minimum sum of the absolute intensity differences along four distinct directions is less than or equal to a predefined iterative threshold (T). While the second stage performs restoration using the weighted mean value of the remaining uncorrupted pixels within the window if at least one non-noisy pixel exists or the weighted mean value of whole window, if they are all corrupted. This filter iteratively decreases the value of the predefined threshold (T) to obtain better filtering results.

The Local and Global Image Information filter proposed in [92] has five stages: 1) noise detection, 2) noise detection rectification, 3) noise restoration, 4) post-restauration noise detection, and 5) noise restoration. The noise detection phase, as its name implies, detects noisy pixels and estimates the total number of them. It does so by sorting in descending order the minimums of four absolute weighted means (each sorted element corresponds to one image pixel). The noise detection rectification phase removes false alarms. The first noise restoration phase restores corrupted pixels using the weighted mean value of the set of remaining noise-free pixels. The second noise detection phase uses the MDWF [90] method for which the predefined threshold is set to $T=510\times0.8^{10}$. Lastly, the second noise restoration stage is identical to the first noise restoration and is used to remove any remaining and previously undetected noise.

The filter introduced in [93] uses a moving window to detect and correct noisy pixels. If all the pixels within the sliding window are corrupted, when correcting a noisy pixel, the size of said window is iteratively increased until at least one non-noisy pixel falls within its boundaries. Once a proper window size is achieved, the maximum and minimum values of non-noisy pixels within said window are used to calculate and average the absolute distances between each non-noisy pixel and these maximum and minimum values. Thereafter, different groups are defined using these averages (average absolute distance between noise-free pixels and maximum, minimum, and average values) and their ratio of occurrences to non-noisy pixels in existence is computed. Finally, the pixel under processing is replaced with the corresponding weighted values calculated from these ratios. This filter attains better performance than the decision filter [79], MDWF [91], and LGII [92].

2.1.3. Non-local mean filters

Non-local mean filters are used for Gaussian denoising. The iterative non-local mean filter (INMF) in [94] 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. The authors of [94] claim to achieve better performance than (SSMF) [74], (SAWMF) [83] and (CMF) [87].

2.2. Spatial combined linear and non-linear filters

2.2.1. Combined median and mean filters

Combined median and mean filters, as their name implies, are a combination of median and mean filters. The Improved Decision Based Algorithm Filter (IDBAF) described in [95] is a variation of the decision filters. It detects the corrupted pixels and replaces them with the median value of the window if it is different from 0 or 255, or with the mean of the neighboring pixels inside the window otherwise.

A Cascading algorithm combining a Decision-based Median Filter and an Asymmetric Trimmed Mean Filter DMF+UTMF is introduced in [96]. In this combined filter, the pixel being processed is replaced with the median value of the moving window if its value is 0 or 255, otherwise it is replaced with the mean of the window (excluding pixels of value 0 or 255).

Modified Decision Based Unsymmetrical Trimmed Median Filters (MDBUTMF) [97] are improvements on DBUTMF [79]. They detect corrupted pixels while checking the number of 0 and 1 pixels (in a normalized image) in the selected window. If all the pixels in the window are zeros and ones, the corrupted pixel is replaced with the window's mean; otherwise, the corrupted pixel is replaced with the median of the remaining pixels (ignoring 0 and 1 values). A survey on impulse denoising filter [98] shows a better performance of MDBUTMF [97] when compared against some standard median filters [9, 20], CWMF [76], weighted median filters (ASMF) [73], and adaptive switching median filters (DWMF) [77].

Cascade decision-based filtering algorithms, as the one presented in [55], are a combination of decision based median filters and direct based filters (DBF) (or modified asymmetric trimmed mean filters (MUTMF)). Detected corrupted pixels within the selected window are replaced with the median value of the window. If possible, they calculate the difference between the pixel's value along four cardinal directions (north and south, east and west, northeast and south west, northwest and southeast) and the mean value of such differences, then find the minimum and maximum edge difference, and replaces the central pixel (if corrupted) with a combination of the minimum and mean values. If the computation of the directional differences is not possible in the current window the pixels with a value different than 0 and 255 are stored in a one-dimensional array and used for further processing. If the number of pixels in the array is more than one, the window's central corrupted pixel is replaced with the mean of the pixels contained in the array and if said length is exactly one, the size

of the window is increased, and the processing window is moved to a new location before the process is repeated.

Decision based partial trimmed global mean filters (DBPTGMF) [99] treat corrupted pixels in one of four ways: 1) If the window contains only 0s, the corrupted pixel is replaced with salt noise trimmed global mean (remove saturated pixel values of 0 and calculates the mean of the image). 2) If the window only contains saturated pixels, the corrupted pixel is replaced with pepper noise trimmed global mean (remove pixels with 255 values and calculates the mean of the image). 3) If the window only contains both 0s and 255s, the corrupted pixels are replaced with both pepper and salt noise trimmed global mean (remove both 0 and 255 values from the image and calculates the mean value of the rest). 4) Otherwise, the corrupted pixel is replaced with the window's median (excluding 0 and 255 from the calculation). Trimmed-global mean filters, as presented in [100], replace corrupted pixels and extreme values, 0s and 1s, given a normalized image, with the moving window's median. However, if such window contains only extreme values, the corrupted pixel is replaced with the trimmed global mean of the image; that is, the average of the uncorrupted pixels in the entire image.

The non-linear decision-based method introduced in [101] uses a sliding window to detect corrupted pixels within it. This method replaces corrupted pixels with the sliding window's median (excluding 0s and 255s from the calculation). However, if all pixels within said window are corrupted, the pixel under consideration is replaced with the sliding window's mean.

Decision based adaptive neighborhood median filters (DBANMF) [102] offer better performance than peer group techniques [103] [104] used for color image denoising. The filter proposed in [102] uses a 3x3 moving window to detect and correct noisy pixels. A pixel is considered noisy if its value is either 0 or 255. If the center pixel of the moving window is noise free then no correction is performed, and the window is simply moved along to the next location. However, if the pixel is noisy, the north, south, east, and west neighbors are checked for potential corrupted pixels. If at least one of them is not corrupted, the pixel being processed is replaced by the median value of the uncorrupted neighbors. Otherwise, the diagonal pixels are checked, and the same process is repeated for them. However, if there are no uncorrupted pixels within the window, the pixel being processed is replaced by the window's mean.

Decision Based Unsymmetrical Trimmed Modified Winsorized Mean Filters (DBUTMWMF), presented in [105], are shown to have better performance than the cardinal B-spline algorithms in [106]. A DBUTMWM filter detects the corrupted pixels while checking the number of extreme points (0s and 1s in a normalized image) in the moving window, and if no extreme points are present, the corrupted pixels are replaced by the window's mean. However, if extreme points exist within the window, the Winsorized mean (the mean value of the window, with the extrema values (min and max) of the window added to the numerator) is used to replace the corrupted pixel's value.

The authors of [107] introduce a denoising filter that utilizes two cascading algorithms. The firsts filter is a combination of a decision based median filter and a modified decision based partial trimmed global mean filter (DBPTGMF), as in [99], while the second one is a combination of a decision based median filter and a modified decision based unsymmetrical trimmed median filter (MDBUTMF), as in [97]. The second algorithm reveals a better performance than the first one. Each of these algorithms are composed of two stages. The first stage is common for both approaches and detects the corrupted pixels while replacing them with the window's median. The second stage of algorithm1 detects the corrupted pixels while checking the number of extreme values (0s and 1s in a normalized image). If all the pixels in the window (except for the corrupted one at the center) are 0s, the corrupted pixel is replaced with a 0. If they are 1s, the corrupted pixel is replaced with a 1. If they are combination of 0`s and 1`s, the corrupted pixel is replaced with the mean value of the pixels within the window. If at least one pixel within the window is different from 0 or 1, the corrupted pixel is replaced with the window's median (computed excluding the 1s and 0s). The second stage of algorithm2 detects the corrupted pixels in the window, and if they are all 0s and 1s, the corrupted pixel is replaced with the window's mean; but if there exists at least one pixel with a value other than 1 or 0, the corrupted pixel is replaced with the window's median (computed excluding the 1s and 0s).

The Switching Adaptive Median and fixed Weighted Mean Filters (SAMFWMF) [108] is a considerable improvement of the AMFWMF [109] (which is applied to MRI images). It detects the corrupted pixels in the moving window by considering one of two scenarios: 1) For general grayscale images, if all of the pixels within are combinations of 0s and 1s or if the window's variance (σ) is much higher than its median (in this case $\sigma \geq 2Median$), the size of the window is progressively increased until it reaches a predefined maximum size. Thereafter, corrupted pixels are replaced with the median of the uncorrupted pixels within said window. 2) For images that could be binarized without losing much information (like checkerboards), it performs a pixel check (without checking the variance) while progressively increasing the windows size, similarly to that of the first scenario, and replaces the value of the corrupted pixels for the windows median. After the initial correction, all images are passed through a shrinkage window that checks the boundary pixels of the sliding window for one of the following conditions: 1) If all the pixels along the boundary are 1, the interior pixels are changed to 1. 2) If all the boundary pixels are 0, the interior pixels are changed to 0. 3) If they are a combination of both (ones and zeros), the window's size is shrunk by one and the process is repeated until the minimum size (3×3) is reached. After this process is completed, the remaining corrupted pixels are filtered by using a 2x2 moving window defined by (29).

$$y_{new} = \frac{\sum_{(x,y)\in S_{i,j}} \omega_{x,y}\, y_{x,y}}{N-1} \tag{29}$$

where y is the input image, N is 4 (the total number of pixels in the window), and $\omega_{x,y}$ is the weight of each neighboring pixel. In this equation, $\omega_{x,y}$ is defined according to the region and the three different probabilities of salt and pepper occurrence (0.5, 1, 2). This filter achieves better performance than the cascading algorithm [107], the filter in [93], IBDNDF

[82], DBUTMWMF [105], and UWMF [86] filters on different images and under different impulse noise intensities.

2.2.2. Combined median and midpoint filters

Cascading algorithms that combine decision-based median filter and asymmetric trimmed midpoint filtering DMF+UTMP, as in [96], replace the corrupted pixels by the window's median before sorting the non-extreme (not 0 or 255) pixels within the window in ascending value order into a 1D array for further processing. If the length of the array is larger than one, the corrupted pixel is replaced with the array's midpoint (average of the last and first element), or by the singular array element if its length is exactly one. However, if the array length is zero the corrupted pixel is replaced by the processed neighboring pixel value of the array.

The decision-based filter in [110] is based on median and midpoint filters. It uses a sliding window to detect corrupted pixels. If there are un-corrupted pixels within the window, excluding 0`s and 255`s, the corrupted center-pixel is replaced with the median value of the uncorrupted ones (again excluding 0 and 255). However, if all pixels within said window are corrupted or unusable, the corrupted center-pixel is replaced with the median value of the sliding window. This method attains better performance than the non-linear decision-based method proposed in [101].

2.2.3. Morphological based filters

Morphological operations are non-linear and not related to numerical values. Generalized directional morphological filters (GDMF), introduced in [111], are a combination of impulse noise detector and weighted rank order morphological filters. They use opening (dilation of erosion of a set A by a structuring element B) and closing (erosion of dilation of a set A by a structuring element B) to obtain the output of the directional morphological filter by a rank order function based on conventional directional morphological filters (DMF) [112, 113]. GDMFs use generalized directional opening and closing to remove impulse noise. Noise detection is achieved by using the difference between the image and opening/closing. If the difference between image and opening is larger than zero and the difference between image and closing equals zero, then salt noise is present. If the difference between image and opening equals zero and the difference between image and closing is larger than zero, then pepper noise present.

Open-close sequence filters (OCSF) [114] use erosion and dilation to detect and correct for noise. Impulse noise is detected by using the mathematical residues of calculating the opening and closing absolute distance from input image (I). If the opening absolute distance is greater than or equal to a predefined threshold and the closing absolute distance equals zero, then salt noise is detected. On the other hand, if the closing absolute distance is greater than or equal to a predefined threshold and the opening absolute distance equals zero, pepper noise is instead detected. Therefore, to restore the image, specialized open-close sequence algorithms are used to detected salt and pepper noise. One of the equations uses opening followed by closing, where opening removes the salt noise and closing removes additional pepper noise introduced by the opening, while another uses closing followed by opening, where closing removes the pepper noise and opening removes the additional salt noise introduced by the closing. Finally, a block smart erase algorithm (BSE) replaces the corrupted pixels with the neighboring pixel's median.

The filter proposed in [115] uses morphological operations and is composed of 3 distinct steps. The first detects the corrupted pixels (0 or 255) in the selected window and if all pixels within it are 0s or 255s, the corrupted pixel is replaced by the window's mean; otherwise, it is replaced with the window's median (computed excluding pixels equal to 0 or 255). The first step ignores the image's border while the second detects corrupted pixels (0 and 255 values) on it. If all the pixels along the borders are 0s or 255s the corrupted pixels are replaced with the mean, otherwise they are replaced with the median (computed excluding pixels equal to 0 or 255). The third step combines dilation and erosion morphological operators with 3×3 structuring elements (known as opening and closing).

2.2.4. Fuzzy filters

2.2.4.1. Switching based filters

The adaptive fuzzy switching filter (AFSF) presented in [116] is composed of 3 sub-units (impulse noise detection, pixel estimation, and pixel modification). The detection sub-unit convolves the image with four 5x5 Laplacian operators and uses the minimum absolute value of the four to detect impulse noise. Therefore, if the pixel under consideration is noisy, its minimum absolute value is large while if noise-free, an edge, or a flat region, its minimum absolute value should be small. The estimation sub-unit removes all pixels, for which the value of one-fourth of the neighboring pixels in the selected window equals one-fourth of the maximum or minimum values of the selected window and calculates the number of remaining pixels (k). If this number equals zero and the size of the window is 3×3, the size of window is incremented to 5×5 and the process is repeated. If it's larger than zero, the median value of the remaining pixels is calculated. If it equals zero, the average of the four pixels around the center pixel is calculated. The modification sub-unit defines a membership function based on the so-called S-function. Finally, the output of the AFSF is defined through equation (22), where m is the result of the estimation function and α is the membership function. Therefore, if α is 0, the pixel being processed is considered noise-free, if α is 1, it is considered noisy, and if α is between 0 and 1, it is somewhat corrupted.

2.2.4.2. Gradient based filters

The filters discussed in this section define their membership function based on the gradients along different directions. The fuzzy impulse noise detection and reduction method (FIDRM) proposed in [117] consists of a detection phase and a filtering phase. In its detection phase, the fuzzy rules for the fuzzy gradient values in eight different directions (north, northeast, east, southeast, south, southwest, west and northwest) are defined. A triangular membership function is defined, and the noisy pixels can be detected based on it and the fuzzy gradient values [118]. The detected noisy pixels are then appended to the level of noise histogram and only if the largest histogram

column is larger than or equal to a predefined threshold (somewhere in the $[2.5, 10]$ interval) is the image considered noisy and requires filtering. The filtering phase defines a trapezoidal membership function, based on a fuzzy set impulse noise, whose parameters are estimated based on the methods described in [119]. This membership function is used for filtering the noisy pixels through iterative fuzzy averaging of neighboring pixels. The results show that this method performs better than filters using fuzzy inference rules such as the Fuzzy Interference Rule by Else action Filter (FIREF) [120], dual-step FIRE filter (DSFIREF) [121], piecewise linear FIRE filter (PWLFIREF) [122], median or mean fuzzy filters such as fuzzy median filter (FMF) [123, 124], adaptive weighted fuzzy mean (AWFM) [125, 126], asymmetrical triangular Median fuzzy filter (ATMF) [127], fuzzy control based filters such as iterative fuzzy control based filter (IFCF) and its modification (i.e. modified IFCF (MIFCF)), extended IFCF (EIFCF), smoothing IFCF (SIFCF), sharpening and smoothing FCF (SSFCF), fixed point FCF (FFCF), adaptive fixed point FCF (AFCF) and adaptive average FCF (ACFCF) [128], histogram adaptive fuzzy filter (HAFF) [129], and fuzzy similarity based filter (FSBF) [130].

Gradient Detection Fuzzy Filters (GDFF) [131] use a set of membership functions, defined using gradients, to improve pixel denoising and edge preservation. The gradient-based membership functions and their multi-direction parameters $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$ [132-134] are set up to determine the pixel orientation along the edges and the denoised pixels are obtained using them in the moving window.

Fuzzy filters, such as the one in [135], are based on the approaches presented in [136-138]. The authors of [135] propose a fuzzy filter that detect the noisy pixels based on the intensity differences among neighboring pixels and corrects them accordingly. This method uses a triangular dual-parameters membership function for both detection and correction and each fuzzy rule has a particular neighboring pixels pattern.

2.2.4.3. Histogram based filters

The filter presented in [139] is an example of a histogram-based filter; it is composed of a fuzzy detection phase followed by a cancellation phase. Noisy pixels are identified in the detection phase through histograms if they are contained within its peaks. The minimum absolute distance between the pixel being processed and its neighboring pixels within the moving window is calculated, and a membership function is defined based on it and two thresholds [10, 20] and [22,32] are proposed. The output of the filter as defined in (22) in the cancelation phase, where m is the window's median, $\alpha = 0$ for non-noisy pixels, $\alpha = 1$ for completely noisy ones, and for $0 \leq \alpha \leq 1$ the pixel contains some noise and its replacement value is a linear combination of itself and m. This filter achieves better performance than rank-order based filters [140, 141], weighted fuzzy mean filters [125, 142], homogeneity level information-based filter [143], and tri-states median filter (TSMF) [75].

3. Gaussian Denoising Filters

3.1. Spatial filters

Spatial filters are defined in the spatial domain where the intensity of each pixel is changed according to its intensity and that of its neighbors.

3.1.1. Non-linear filters

Non-linear filters simply have outputs dictated by a non-linear function of their inputs.

3.1.1.1. Total variation filters

Total variation methods assume the total variation and, consequently assume that the integral of the signal gradient to be high. Therefore, by decreasing total variation, a denoised image with high similarity is obtained. The method in [144] implements total variation filters by developing the methods in [145] and [146]. However, the authors of [144] do a better job at explaining the concept of total variation. This method tries to reconstruct the original image from a noisy one by minimizing total variation through the denoising algorithm defined in (30), therefore reaching the Euler-Lagrange equation [147, 148].

$$min \frac{\|I-f\|^2}{2\lambda} + J(I) \tag{30}$$

where I is the clean image, f is the observed image, λ is the Lagrange multiplier, σ^2 is the estimated noise variance, and $J(I)$ is the total variation as defined in (31).

$$J(I) = \sum_{(i,j)=1}^{N} \left|(\nabla \mathrm{I})_{i,j}\right| \tag{31}$$

Improved total variation [149], an improvement on total variation [144], is applied to the image to smooth it and remove the remaining noise, especially from high frequency sub-bands. The filter proposed in [149] is based on dual information [150, 151] and presents a fast algorithm for minimizing the total variation through the Euler equation. Furthermore, [149] suggests an innovative solution to equation (32) to recover the original $N \times N$ image by using the following condition:

$$\min\{J(u): \|I - f\|^2 = N^2\sigma^2\} \tag{32}$$

Improvements to total variation are further described in [152-154]. The filter presented in [152] achieves smoother image variation and contour regularity by using higher total variation derivatives, higher isotropy, and higher anisotropic rotation invariance [155]. The total variation non-regularization filter (TVNLR) introduced in [153] aims to preserve image details while maximizing denoising. It introduces an improved non-local mean total variation in image f by solving (33)

$$\min ||Df||_1 + \alpha||f - Wf||_2^2 \tag{33}$$

here W is the weight of the non-central pixels of the selected window, and D is the set of vertical and horizontal finite difference operators. This filter also uses an augmented decomposed Lagrangian method to solve (33) and to determine the regularization parameters through the minimization of one of the parameters [156]. The results of this filter are better than those of tree-structured DCT [157].

In the traditional total variation algorithm [144], parameters only contain the absolute gradient values, therefore increasing the confidence of properly classifying edges as noise in smooth regions where the filter identifies edge pixels as noise. Therefore, the non-uniform total variation partition filter (NTVPF), as that in [154], use non-uniform partitions [158] where bivariate polynomial and divided image error controls [159] are used. Its first doubles the size of the image with bilinear interpolation and applies non-uniform partition, after control error and parameter initialization for total variation, before computing the mean square error. If the mean square error is smaller than the control error, the current part is recorded, all parts are combined, and the image is shrunk by half to reconstruct it. The authors of this paper claim to obtain better performance than the Wiener filter [160] as well as the wavelet filter [161].

The method in [162] is based on 3D-HIS mixed denoising and 2D spatial dimensions cube total variation. It achieves maximum smoothness (thanks to total variation) and utilizes more information for restoration. As a result, it obtains better performance than the PCABM4D [163], LRMR [164], and LRTV [165].

3.1.1.2. Anisotropic diffusion filters

Anisotropic diffusion filters reduce image noise and preserves edge details by using non-linear and space-variant transforms. The method described in [166] iteratively estimates local image variation, edge strength, and noise variation until satisfactory results are obtained. The authors of [167] introduce scale-space but have problems determining the location of edges at coarse scales and suffer from edge junction destruction. Algorithms that use a diffusion process and a new definition of scale-space are introduced in [22]. An appropriate diffusion coefficient (c) should be chosen to satisfy the criteria of sharp boundaries (immediate localization) and inter-region smoothing (piecewise smoothing). Anisotropic diffusion for image f is defined as follows:

$$f_t = div(c(x,y,t)\nabla f) = c(x,y,t)\Delta f + \nabla c.\nabla f \qquad (34)$$

where div is a divergence operator, ∇ and Δ are respectively the gradient and Laplacian operators, and $c(x,y,t)$ is a conduction coefficient that can be defined by a non-negative monotonically decreasing function (35) to properly estimate the edges.

$$c(x,y,t) = g(||\nabla f(x,y,t)||) \qquad (35)$$

If c is constant, equation (34) would be an isotropic heat equation described as $f_t = c\Delta f$. The authors of [22] further define possible choices for diffusion coefficients ($g(\nabla f)$).

The improvements on anisotropic diffusion from [168] and [169] are the result of modifications made to the anisotropic diffusion equation (34). They use a Gaussian kernel (G) to define their diffusion coefficients $(g(||\nabla(G * f)||)$. The modified anisotropic diffusion in [168] diffuses the image with a different diffusion function, and [169] introduced a non-linear partial differential equation to thin and preserve edges. Further improvement on anisotropic diffusion can be achieved by reducing the conduction coefficient at narrow peaks.

An improvement on anisotropic diffusion in [170] attains better noise removal by discriminating between fine details and noise while preserving edges and details by using local gradients and gray-level variance. If the gradient and variance are large, the diffusion coefficient function is small, and the gray-level is preserved by stopping the process. If they are both small, the diffusion coefficient function is large and hence the edges and details are preserved.

The authors of [171] achieve further improvements on anisotropic diffusion and improved edge quality by choosing appropriate conduction functions [22, 172], appropriate gradient thresholding and Gaussian kernel standard deviation parameters, and stopping criterion by evaluating edge quality in each iteration. They report a better performance than the anisotropic diffusion methods in [172, 173] by estimating gradient threshold parameters for each iteration. Further performance improvements are achieved by using stopping criterions other than those proposed in Mrazek-Navara (MN) [174] and Gilboa-Sochen-Zeevi (GSZ) methods [175, 176].

The improved anisotropic diffusion filter presented in [177], applies a Gaussian filter to the moving window to reduce noise while computing the local difference between its result (the output of the Gaussian filter I_t) and the pixel of interest (POI). If this difference is greater than a predefined threshold ($0.1 \times I_t$ as proposed in the paper), the POI is considered corrupted and is replaced with I_t; otherwise, it is left unchanged. An appropriate constant must be set to define the diffusion coefficients and control the smoothing process for high (over smoothing) and low (left unfiltered pixel) thresholds. Semi adaptive-based thresholds are used in the diffusion coefficient functions based on the noisy pixels' gradient value to achieve higher diffusion in smooth areas and lower diffusion on the boundaries. Improved anisotropic diffusion [177] is shown to better performance than the methods in [168, 170] and is based on the anisotropic model (EAD) from [178] with a modified anisotropic diffusion as given below:

$$f_t = div(\alpha.c(||\nabla f||).\nabla f)\ ,\ \alpha = 1 - \nabla G_\sigma * c(f) \qquad (36)$$

where c is the diffusion coefficient, α is the spatially adaptive term, and G_σ is the Gaussian kernel. The filter described in [177] achieves better performance than modified the Perona-Malik model (MPM) [179] (based on directional Laplacian and Hessian matrix) and UNKD model [180] with a modified anisotropic diffusion based on second order pixel difference Laplacian and gray-level variance.

3.1.1.3. Non-local mean filters

Non-local similarity-based filters use several similar patches to reconstruct the patch being processed. The non-local mean (NLM) filter presented in [23] uses non-local averaging to preserve image edges, and self-predictions to replaces the noisy pixel with the mean weighted average of the pixels with similar Gaussian neighbors. In other word, NLMs compare the patch being processed against its neighboring patches to find similar ones. The average of all pixels in the image (I) is a

non-local mean. The main idea here is that similarities among different parts of the image exist and they can be used to aid in the restoration process. The denoised pixel values are obtained as

$$y[d](j) = \sum_{j \in I} W(i,j) d(j) \quad (37)$$

where y is an estimated value, d is the discrete noisy image $d = \{d(i) | i \in I\}$ and W is the set of weights ($\sum_j W(i,j) = 1$) described as follows:

$$W(i,j) = \frac{1}{z(i)} e^{-\frac{||d(N_i) - d(N_j)||^2_{2,a}}{h^2}} \quad (38)$$

where $||d(N_i) - d(N_i)||^2_{2,a}$ is the weighted Euclidean distance, a is the standard deviation, N_k is a fixed size neighborhood window centered at k, h is the degrees of the filter, and $z(i)$ is the normalizing constant (39).

$$z(i) = \sum_j e^{-\frac{||d(N_i) - d(N_i)||^2_{2,a}}{h^2}} \quad (39)$$

The authors of [181] postulate that Non-Local Mean (NLM) filters can emerge from a Bayesian approach with new arguments. This improves NLMs by introducing a spatially adaptive dictionary and a new statistical distance measure to compare patches. The patches in the local dictionary are discarded if the difference between their mean value and the mean of the central patch is too large or if the F-test ratio is above a set threshold. It also improves the distance used for patch comparison by using a new statistical distance measure which, unlike the non-local mean [23], is valid for both overlapping and statistically dependent patches.

The authors in [182] introduce an iterative based NLM filter to track noise to further improve adequate weight selection by using a multivariate robust M-function [183] and make use of a robust estimator (replacing the Euclidean distance by the Mahalanobis distance) for correlated noise (color) removal, and an adaptive principal component [184] as a post-processing step to remove the remaining noise from areas that had limited iterations. This implementation leads to small search windows (31×31), better results, and significant noise reduction for every iteration; leading to more accurate weight estimation and, perhaps, reduced computational cost. In order to improve the NLM filter weights from [185], a Gaussian filter is used as a preprocessing step and the weights are computed as below:

$$w(p,q) = e^{-\frac{\hat{d}(p,q)}{h(x,y)}} \quad (40)$$

where $\hat{d}(p,q)$ is the spatial distance between neighboring pixels (p,q) and the center pixel, and $h(x,y)$ is a decay parameter [186].

In [187], the exponential term of the weight function in (38) is squared, the weight pre-classification (according to the similarity of the patches) is done by dividing each image patches into 4 sub-patches, and the ratio of sum of diagonal sub-patches is used to pre-classify non-local patches. These changes yielded improvements on the non-local mean [23]. The proposed weight kernel generates a gentle decline in weights when the distance between image patches is relatively small and sharp weights reductions when such distance increases. This improves the influence of similar pixel on the filtering results. The authors in [188] pick the pixels with highest weight without using Euclidean distance measurement.

The authors in [189] reduce the NLM [23] algorithm's time complexity by projecting the image's patches into a global feature space and performing a statistical t-test to reduce its dimensionality. The proposed algorithm is composed of 3 steps: 1) a global feature space that stored all possible neighborhood pixels is created; 2) a statistical t-test is performed to reduce the dimensionality of the feature space 3) the non-local means algorithm is implemented using the selected feature points to calculate the similarity measures of image neighborhoods.

NLM is sensitive to pixels that are far from each other within a patch (symmetric locations) and consequently sets the same weight for them, leading to a reduced denoising performance. A geometric distance between image pixels is used in [190] to obtain improved patch weights and better performance than Wiener filters, the NLM filter on [23], and the global NLM filter in [191].

3.1.1.4. Bilateral filters

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) [24]. The non-iterative method proposed in [24] improves edge preservation and reduces blurring better than previously introduced methods like [192]. The difference between the Gaussian-bilateral filter and the original image is passed through a wavelet domain block where the resulting wavelet reconstruction is combined with the output of Gaussian-Bilateral filter to preserve image details and sharpen boundaries from the sub-band approximation, in [193].

The improved bilateral filter presented in [194] achieve good high-noise rejection performance, without additional computational cost, by adding a preprocessing step that replaces the observed image with an average defined by (41).

$$\bar{f}(i) = \frac{1}{(2L+1)^2} \sum_{j \in \{-L,L\}^2} f(i-j) \quad (41)$$

where L controls the amount of smoothing ($L = 1$ is proposed in [194]) and $f(i)$ is the corrupted image. This change yields to better results with low noise thresholds by combining the filters described in [192] and [195]. The minimized MSE between the denoised image (resulting from the denoising operation) and clean image (original image before noise is introduced) should be calculated, but the clean image is often unavailable. Therefore, to solve this problem, the Stein's unbiased risk estimate (SURE) [196] is used. Finally, to minimize the mean square error (MSE) and obtain satisfactory denoising estimations, the optimal weights are obtained and used to filter the noisy image. The time complexity of the filter is reduced by using the faster algorithms presented in [197] and [198].

3.1.1.5. Fourth-Order Partial Differential filters

Fourth order partial differential equations (4th PDE) are used to optimize noise removal and edge preservation by minimizing the cost function (absolute value of the image's Laplacian). The fourth order partial differential equations (4th PDE) provided by [25] performs very well, avoiding the blocky artifacts that appear in the early stages of diffusion when smother areas diffuse faster than less smooth ones. This filter offers a tradeoff between noise removal and edge preservation but resolves the ramp edge artifact problem from [199]. The minimization of the cost function is equivalent to smoothing the image and is a special form of the general variation problem in [200]. This means that the global the cost function minimum is obtained when $|\nabla^2 f| = 0$ (f is observed image and ∇^2 is the Laplacian operator). The Euler equation, numerically and iteratively solved through (42), is obtained by applying the Laplacian to the partial differential equations and the energy function.

$$f_t = -\nabla^2[c(|\nabla^2 f|)\nabla^2 f] \quad (42)$$

here c is the diffusion coefficient.

The solution of the Euler equation is obtained as (time) $t \to \infty$, but the solution process may be stopped early to obtain an optimal tradeoff between noise removal and edge preservation.

The improved PDE proposed in [201] achieves higher detail preservation, even in low SNR conditions. Unlike the total variation method in [144], where the total variation norm of the observed image function (f) must be minimized, the study in [201] proposes a 4th order PDE that allows for the minimization of the total variation norm of ∇f. These two different functions are defined to calculate the signal oscillations occurring within the noisy image. The first is rotational invariant and its minimum values are subject to noise level constraints obtained from a Lagrangian function. The noise level constraint is recovered and f would satisfy the non-linear partial differential equation by solving the Lagrangian function and hence the Lagrange multiplier. Thereafter, a parabolic equation is defined for the nonlinear partial differential equation for $t > 0$ (time as an evolution parameter) and the Lagrange multiplier is obtained by using the same idea as in [144] to maintain the parabolic equation in steady state. Applying the Lagrangian to the second function (used to calculate the signal oscillations occurring within the noisy image), through a process like the one discussed before, yields the final non-linear PDE.

Some piecewise planarity conditions for 4th vibrational denoising in continuum coverage to a piecewise planar image are introduced in [202] in order to achieve better image restoration, while the authors of [203] propose an improved 4th PDE to set the diffusivity functions that controls the diffusion along the gradient direction, achieve fast convergence filtering, and obtain better edge preserving performance. The function proposed in [203] has an anisotropic behavior and is based on two second order derivatives along the direction and level of the gradient. An adaptive 4th order PDE regularization (43), related to anisotropic diffusion, is introduced in [204] to avoid edge blurring and preserve edge details.

$$f_t = -\left(\alpha \frac{f_{xx}}{|\nabla^2 f|_\varepsilon}\right)_{xx} - \left(\alpha \frac{f_{xy}}{|\nabla^2 f|_\varepsilon}\right)_{yx} - \left(\alpha \frac{f_{yx}}{|\nabla^2 f|_\varepsilon}\right)_{xy} - \left(\alpha \frac{f_{yy}}{|\nabla^2 f|_\varepsilon}\right)_{yy} - \lambda(f - f_0) \quad (43)$$

For which f is the input image, ε is the small positive parameter ($|\nabla^2 f|_\varepsilon = \sqrt{|\nabla^2 f|^2 + \varepsilon}$), and $\alpha = \frac{1}{1+k|\nabla G * f_0|^2}$ is the diffusivity function where k is the threshold parameter and G is the Gaussian function.

3.1.1.6. Kuwahara filters

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 [205]. However, Kuwahara filters have some important limitations as presented in [206]: 1) the sub-regions' square shape causes the output to look blocky, which can be solved by changing the shape of the sub-regions; and 2) the filter's output is not uniquely defined for instances where two sub-windows yield the same standard deviation.

The proposed output of N equal circular sub-regions presented in [206] is a weighted average based on means and standard deviations.

3.1.2. Linear filters

3.1.2.1. Mean filters

Mean filters, as in [19] and [20], use a moving window to detect the corrupted pixels and replace them with the average value of their neighboring pixels.

3.1.2.2. Gaussian filters

Gaussian filters, as in [20], aim to preserve edge detail. They resemble mean filters but use a Gaussian distribution function to achieve discrete approximations and softer frequency responses. The authors of [207] provide an improved on Gaussian filter by using a non-local mean filter that can preserve subtle grey-values and edge information. Their filter selects pixels which are similar to the pixel being corrected and applies a weighted average function within a small fixed window. The research shown in [208] also uses Gaussian filters to perform denoising.

3.1.2.3. Wiener filters

Wiener filter are linear and mean square error filters that can be used for Gaussian denoising and are often applied to images in their frequency domain [19, 20]. The authors of [210] propose an improvement to BM3D [209]. They generate improved Wiener filters by optimizing the SSIM (structured similarity index measure) between reconstructed and clean images instead of using the mean square error (MSE).

3.2. Neural network-based filters

3.2.1. Cellular 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 [211] proposed a

combination of a nonlinear transform domain filter and a cellular neural network. In this case, the spectral coefficients of the noisy image for the moving window are calculated over an orthogonal transform and the inverse transform of the pointwise multiplication of these coefficients with the filter's coefficients is reported as the output spectral coefficients. The optimal filter is obtained by minimizing the average difference between the filtered data's spectral coefficient and those of the original one (average loss). The inverse transform of the output spectral coefficients is estimated as the value of the central pixel of the moving window and is improved by using a trained cellular neural network estimator, with universal binary neurons (UBN) [212] and Boolean decision functions. Multi-valued neurons (MVN) [213] are also used to get parametric descriptions of multiple valued functions which are similar to Boolean threshold functions.

3.2.2. Auto encoder-based filters

Autoencoders learn to perform efficient representation, or encoding, of a given data set (through dimensional reduction) using unsupervised learning. The authors in [214] propose a combination of sparse coding and deep networks pre-trained with denoising auto-encoder (DA) as an alternative to training pure DAs. A denoising auto-encoder is used as the main block and is trained to minimize the reconstruction loss between the original image and its approximation, as defined by a sigmoid activation function. Another DA can be trained to follow the previously trained one by using the hidden layer output of the first DA as an input; creating a stacked denoising auto encoder (SDA) [215]. The authors of the study in [214] propose a 5-layer (a tradeoff between layer count and training time) stacked sparse denoising auto encoder (SSDA) to denoise the images with hidden layers of size proportional to the input. This implementation claims to obtain better result than Bayesian least squares- Gaussian scale mixture BLS-GSM [216] and the k means singular value decomposition K-SVD [217].

3.2.3. Convolutional Neural Network based filters

Convolutional neural networks are deep and feedforward artificial neural networks that use a variation of multilayer perceptrons and preprocessing. The Trainable Nonlinear Reaction Diffusion (TRND), presented in [218], used supervised training to train a dynamic nonlinear reaction diffusion model with time-dependent parameters (linear filters and influence functions). The proposed iterative non-linear diffusion model uses the difference between the reaction and the diffusion (as defined in (44) for Gaussian denoising terms.

$$f_t = f_{t-1} - \left(\sum_{i=1}^{N_k} \bar{k}_i^t * \varphi_i^t(k_i^t * f_{t-1}) + \lambda^t(f_{t-1} - I)\right) \quad (44)$$

where f is the noisy image, k_i is a set of linear filters, N_k is the number of filters, λ^t is a positive reaction force weight used for the training phase implementation, I is the original image, $\bar{k}_i$ is a convolution kernel (180° rotation in k_i), and φ_i^t is the influence function as a linear combination of radial basis functions (RBF) [219].

The method proposed in [218] is similar to the feedback convolutional neural network (CNN) presented in [220], 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. This method also uses a greedy pre-training phase [221, 222] as a good initialization for the joint training [219] that follows, where the cost function is minimized in all iterations as pre-training. Both these training methods use the gradient based algorithm (L-BFGS) presented in [223]. The TRND method presented in [218] claims to obtain better performance than Block matching and 3D filtering (BM3D) [209], learned simultaneous sparse coding (LSSC) [224], Expected Patch Log Likelihood-Gaussian mixtures [225], optimization-Markov random fields (opt-MRF) [226], Regression Tree Fields (RTF) [227], cascade of shrinkage fields (CSF) [219], Weighted nuclear norm minimization (WNNM) [228], Active Random Field (ARF) [229], and optimization-gradient descent (opt-GD) [230].

The deep convolutional neural network method (DnCNN) introduced in [231] can be thought of as a generalization of TRND [218] that: 1) achieve easier training, by replacing the influence function with a rectified linear unit (ReLU) training model [232]; 2) increase architectural depth (number of convolution layers) to improve the image modeling capacity; and 3) combine batch normalization to improve performance. Such model uses a mean square error loss function and three types of layers: convolutional with Rectified Lineal Unit activation function as the first layer to produce 64 feature maps, convolutional with batch normalization and Rectified Linear Unit activation function (layer 2 through $D-1$ where D is the number of layers) with batch normalization [233], and purely convolutional (last layer) to reconstruct the output. To reduce boundary artifact due to size mismatches between different input images, the input image is padded with zeros before the first convolution stage. Residual learning and batch normalization are associated with Gaussian distributions and can complement each other in the presence of Gaussian noise. The proposed integration has benefits such as lower convergence speed of residual learning with batch normalization, faster training, and performance improvements. The method in [231] achieves better performance than TRND [218] and multi-layer perceptron (MLP) [234]. It can also be used for images with non-Gaussian noise distribution, unknown Gaussian noise levels, and for establishing a single model for several image denoising tasks.

The convolutional neural network (CNN) presented in [235] is made up of 7 layers; Dilated Convolution+ReLU first, Dilated Convolution+Batch normalization in the middle, and Dilated Convolution last. The Dilated filter is used to enlarge the receptive field to 33×33 and increase the network's depth. Such enlargement catches context information easily thereby aiding image reconstruction. Batch normalization and residual learning are used to speed up training speeds and achieve faster model transformations. The authors also use small training samples, from the 400 Berkeley segmentation dataset (BSD) images, with zero padding to avoid boundary artifacts. The loss function used (for residual learning) is based on that of the ADAM optimizer from [236]. Their proposed filter

achieves better performance than BM3D [209], WNNM [228], MLP [234], and TRND [218].

The denoising method introduced in [237] uses TensorFlow (a popular machine learning python library [238]) to implement a 20 layer fully conventional neural network. It's trained using 160 mini-batches of 128×128 images. Each layer has 63 3×3 kernels and forwards its results through ReLU activations to the next layer up the chain. Additionally, each layer produces an extra convolutional result (using a 3x3 kernel) that is added along with the result of all other layers and the input image to produce the denoised results. However, as they perform class-specific denoising, the authors use classifiers pretrained on VGG16 [239] to asses which class the noisy image belong to before they can be adequately filtered. This filter achieves better performance than BM3D [209], multilayer perceptron (MLP) [234], TRND [218], and IRCNN [235].

3.3. Fuzzy based filters

3.3.1. Weighted averaging filters

The fuzzy rule-based filter proposed in [124] is a weighted average filter with non-linear weights. The authors of [124] introduce a gradient based non-linear multi-dimensional step-like function for which the mean square error is minimized. This gradient-based method updates the filter's weights according to the cost function's gradient. The updated weights are defined by this gradient based-method [240] as a summation of the previous weights and the derivative of the mean square error with respect to weights multiplied by a small positive coefficient. The fuzzy rules are based on the local variance and the membership function is defined according to the difference between central and neighboring pixels and the size of the weights. This causes the area in the selected window to be flat and the weights to take large values, if the local variance around the central pixel is small, or to be edge/textures and the weights to assume large values, otherwise. The modified fuzzy filter introduced in [241] is based on a weighted average filter where its weights are defined by an 8-neighbor fuzzy exponential membership function.

3.3.2. Control based filters

The authors of [128] 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 S-type fuzzy function uses the difference between neighboring pixels and the central pixel of the selected window, defined as "positive" or "negative" and "smallness" or "largeness". Its output is defined as follows:

$$y = \frac{\sum_{i=0}^{6} c_i w_i \lambda_i}{\sum_{i=0}^{6} w_i \lambda_i} \tag{45}$$

here i is the rule index, c_i is the central point, w_i is the width of the membership function, and λ_i is the degree of activity. The modified IFCF (MIFCF) avoids edge blurriness 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) propose the following modifications to SFCF: 1) using two 5 bits (LUT) membership functions (leading to easier implementations), 2) changing the sequence of algorithm computations, 3) left and right shift of multiplication and division parameters.

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.

3.3.3. Similarity based filters

The fuzzy similarity-based filter (FSBF) presented in [130] 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. Lastly, pixel correction is obtained through centroid and simplified defuzzification.

3.3.4. Wavelet domain-based filters

The method presented in [242] 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 Asymmetrical Triangular Median Fuzzy Filter (ATMF) [127] or Asymmetrical Triangular Moving Average Filter (ATMAF) [127]). The final denoised image is obtained through an inverse wavelet transform.

3.4. Combined Fuzzy and morphological filters

The method introduced in [243] uses fuzzy closing and opening mathematical image morphology [244] based on image erosion and dilation [245]. The authors in [243] propose an adaptive structuring element instead of a fixed one, as fixed structuring element may not be the best option to process all the image pixels as they may remove thin elements or displace contours. Therefore, morphological amoebas (or operators) [246] are used because of their ability to adapt their shape and temporarily alter their cells. Their closing and opening are used based on morphological dilation and erosion as they are equivalent to t-morphological operators for grey-level images [245, 246]. These two morphological operators are based on alternate filters [229] and [230], used when the filters $C(O(F))$, $O(C(F))$, $O(O(F))$ and $C(C(F))$ are not increasing and independent. In these cases ([247] and [248]), the fuzzy morphological amoeba at pixel x is set through the following steps: 1) the classical morphological amoeba is computed, 2) for each pixel y and x, two windows centered at y and x are defined and sub-images f_x and f_y are constructed,

3) the degree of similarity between f_x and f_y is computed, and 4) weights $\in [0,1]$ are proportionally assigned to pixel y of the amoeba that has the lowest degree of similarity. Therefore, a fuzzy morphological amoeba is defined (using the degree of similarity between f_x and f_y) and consequently, the fuzzy morphological dilation, erosion, closing, and openings will be as defined by the fuzzy morphological amoebas. Lastly, alternate filters for Gaussian denoising can be defined from the fuzzy morphological closing and opening.

3.5. Statistical filters

3.5.1. Singularity function-based filters

The method proposed in [249] 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). A discrete signal with *n* singular points can be represented, based on the singular function space (SFS) definitions, as a weighted sum of *n* singularity functions in SFS and *n* singularity spectrum functions in SSFS. The first part of the reconstructed image represents the denoised image and the second part is the components recovered by the 2-D SFA. The noise spectral data is thus a combination of the recovered spectral data and additional spectral error. The low frequency components with high SNR are kept in order to avoid reconstruction errors. In [249], four partial areas of a noisy image are independently translated into the k-space frequency domain where they are reconstructed through a 2-D SFA and are used to obtain the final noise-free version by averaging the resulting four images.

3.5.2. Hidden Markov tree-based filters

Markov trees 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 [250] uses non-parametric hidden Markov trees to denoise images. The authors of [250] use Daubechies-4 orthogonal wavelet, 5th order steerable pyramid decomposition, and hierarchical Dirichlet processes-hidden Markov trees (HDP-HMT) to restore the corrupted images in an empirical Bayesian fashion. HDP-HMTs can be adapted to different images and wavelets by using Monte Carlo learning. The corrected images retain important non-Gaussian dependencies thanks to uncorrelated coefficients. A hierarchical Dirichlet process is suggested as a flexible method to share mixture components among groups of related data [251]. The hierarchical Dirichlet processes is further developed into HDP-HMT [250] to model the global statistics of wavelet coefficient. A collapsed Gibbs sampler, as described in [251], is run on the noisy wavelet tree to infers the posterior distribution of HDP-HMT parameter by alternating between sampling assignment, or state space of hidden Markov tree, and global transition probabilities. Then, 500 posterior distribution samples are collected after ignoring the first 1000 burn-in iterations. Thereafter, the conditional mean of the true coefficients as well as the reduced denoising, a linear least square, are obtained by using the belief propagation algorithm [252, 253]. Finally, the denoised image is obtained by reverting the wavelet transform and combining observed scaling coefficients with the posterior mean of each detail coefficient.

3.5.3. Neighborhood based filters

An unsupervised, information-theoretic, adaptive filter (UINTA) is proposed in [17] based on the statistical relationship of the pixel being processed and its neighbors. Therefore, for each pixel-neighborhood pair, the entropy of the density distribution (joint entropy between pixels and their neighbors) is reduced through an iterative method that computes the statistical properties of the data and consequently reduces the noise present in the image spectrum. The proposed method uses gradient descent to minimize the entropy and learning is stopped before steady state is reached, obtaining the scale randomness of the filtered image.

3.5.4. Norm based filters

The PCABM4D method from [163] is based on sparse regularization. It uses a 3D-overcomplete wavelet dictionary and solves sparse regularization minimization by using an iterative Chambolle method.

The modified low rank matrix recovery (LRMR) method presented in [164] explores the hyperspectral image (HSI) low rank property by using a linear spectral mixing model [254, 255]. It first builds an HSI restoration model and solves it using "Go Decomposition" (GoDec) [165] and augmented Lagrange multiplier (ALM) numerical optimization. The optimization problem is as described in [164] and by (46):

$$min_{L,S} ||X - L - S||_F^2 \;\; s.t \;\; rank(L) \leq r, \;\; card(S) \leq k \qquad (46)$$

where X is the clean image, L is the low-rank matrix, S is the sparse error matrix, r and k are the upper bound of the **L** rank's and the cardinality of **S.**

The authors of [165] introduce a modified low rank matrix factorization (LRTV) denoising method. In this method, the nuclear norm is used as the low rank property, TV regularization is used for its spatial piecewise smoothness, and the l_1- norm is used to detect sparse noise. The LVTR sorting model is as (47):

$$min_{X,S \in R^{m \times n}} ||X||_* + \tau ||X||_{HTV} + \lambda ||S||_1 \; s.t. \;\; ||Y - X - S||_F^2 \leq \varepsilon \;\; rank(X) \leq r \quad (47)$$

for which X is the clean image, Y is the noisy image, S is the sparse noisy image, $||X||_{HTV}$ is the band-by-band HSI TV norm, τ controls the nuclear norm to TV norm balance, λ restricts the sparse noise sparsity, and ε is noise variance.

3.6. Transform domain filters

3.6.1. Curvelet filters

Curvelets based on the theory of multiscale geometry (using scale, orientation and position) are introduced in [26], yielding better performance on edge boundaries than other mature wavelet image denoising methods. The curvelet transform method uses ridgelet and curvelet [256, 257, 258] representations to improve their translation onto a Cartesian grid. The curvelet frame has sparse representation for singularities supported on 2-D curves and forms a sparse basis for pseudo differential and Fourier operators. Radon transform, and 1-D wavelet transforms are used to calculate

the ridgelets. Digital Radon transforms can be obtained by applying interpolation in Fourier space (2-D FFT, reversible Cartesian to polar conversion, and 1-D $IFFT$). Pseudo polar grids are used for Cartesian to polar conversion with square geometry level sets of pseudo-radial variables and to translate Cartesian samples onto recto polar grids. A 1-D over-complete wavelet pyramid is taken along the radial variable in Radon space to complete ridgelet transform and apply it to the Radon transform. The wavelet coefficients are computed in each Fourier space sub-band (with respect to Nyquist rate) to avoid computing the Fourier transform for each radial line. The trivial reconstruction shows that ridgelet coefficients need to be added together to reconstruct the Fourier coefficient. The proposed digital ridgelet and orthogonal wavelet pyramids have constant orientations and elements while being scale independent [256]. The proposed digital ridgelet pyramid decomposes the image into overlapping blocks of pixels to block artifacts. Discrete Curvelet Transform of Continuum Functions (DCT-CF) are used to obtain the digital curvelet transform. Following wavelet decomposition, the sub-bands of a filter bank are represented by different levels of a ridgelet pyramid. In curvelet transform, spatial partitioning (spatial band-pass filtering to isolate the scales) and multiscale ridgelet transform are applied to each sub-band separately. Thereafter, a so-called "$\grave{a}$ trous" filtering is used for sub-band implementation and the algorithm decomposes the image [259] to obtain the final filtered image through hard thresholding rules on the curvelet transform result.

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.

3.6.2. Contourlet filters

The proposed Contourlet method in [27] 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. Thus, unlike other approaches that first develop a transform in the continuous domain and then discretize the sampled data, the method in [27] starts with a discrete-domain construction and then studies its convergence to an expansion in the continuous domain.

The double filter proposed in [260] constructs a Laplacian pyramid filter (LP) [261] and a directional filter bank (DFB) [262] to find and link discontinuities and, consequently, obtain a sparse expansion of natural images with contours. LP filters are used to transform images into a set of LP band-passes and DFB are used to further decompose this set into a number of sub-bands; consequently, storing the directional information. Multiscale and multidirectional decomposition is used to reconstruct the filtered image. Multiscale decomposition has an oversampling drawback, as it uses LP to effectively apply a low pass filter to the original image.

In [263], tight frames with bounds equal to 1 are obtained when the LP has orthogonal filters, therefore, a dual frame operator is used for the linear reconstruction of the directional decomposition. This decomposition uses a single-level binary tree decomposition (2^l sub-band) with wedge-shaped frequency bands DFB that represent the image as a set of multi-scale directional sub-bands to simplify the rules and avoid input image modulation; proposing a new two-block reconstruction in [264].

The contourlet filters have the following features: 1) they are defined on rectangular grids (one of its differences from curvelet filters), 2) they have 2-D frequency partitions on centric squares, and 3) they have fast iterative filter bank algorithms. Parabolic scaling [265] and directional vanishing moment (DVM) [266] are two methods used by contourlet to approximate a smooth discontinuity curve and piece-wise smooth images with smooth contours [267- 269]. Denoising is achieved by improving the contourlets approximation and by preserving the most significant coefficients.

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.

3.6.3. Non-local based filters

Non-local filters process all pixels in the image to find how similar they are to the pixel at the center of the moving window being processed.

3.6.3.1 BM3D

An enhanced sparse representation-based filter in transform domain (BM3D) is proposed in [209]. It groups similar image patches (2-D blocks) into 3D matrices similar as in [270-272] to obtain the correct/denoised signal by using the similarity between them. Collaborative filtering is used to generate multiple 3-D groups estimations in the following stages: 3-D group transformation, transform spectrum shrinkage (through hard thresholding or Wiener filtering), and the inverse 3-D transform. A sparse representation of groups that make shrinkage effective is generated to attenuate the present Gaussian noise. The proposed algorithm has two steps, basic input noise estimation and final noise estimation. The first step performs a block-wise noise estimation using grouping and 3-stages of collaborating filtering while the second step uses aggregation by computing the weighted average of all block-wise estimates to approximate the correct image.

This second step performs block-wise estimates of each blocks by using grouping and collaborative Wiener filtering. Grouping finds the locations of blocks similar to the currently processed one and combines them into two 3-D matrices, one containing groups from the noisy image and another with the groups from the basic estimate. Collaborative Wiener filtering applies 3-D transforms to both groups and perform Wiener filtering on the noisy group while the inverse 3-D transform is applied to obtain the block estimation. Thereafter, the final estimation of the corrected image is performed by using a weighted average of the block-wise estimates. The method proposed in [209] claims to obtain better performance than BLS-GSM [216], K-SVD [217], pointwise SA-DCT [273], and prelim. 3D-DFT BM3D [274].

3.6.3.2 BM4D

The BM4D filter proposed in [275] is an extension of BM3D. In BM3D, the data is represented by blocks of pixels whereas in BM4D, it is represented by voxel cubes. BM4D is applied to the groups formed by stacking mutually similar cubes and uses the correlation between group voxels and the non-local correlation of pixels of different cubes. The denoised estimate of each grouped cube is obtained through an inverse 4D transformation and they are adaptively aggregated at their original locations. Like BM3D, BM4D is composed of two steps with three parts each: 1) grouping, 2) 3-stages collaborative filtering (4D transform, hard thresholding for step 1 with noisy inputs and Wiener filtering for step 2 with basic volumetric estimation inputs, inverse 4D transform), and 3) aggregation. The authors of [275] claim that the method achieves better result than OB-NLM3D-WM [276], 3-D DCT ODCT3D [277], PRI-NLM3D [277], and O13-NLM3D [278].

3.6.4. Data adaptive filters

Data adaptive filters use a common representation of the whole image that minimizes the global reconstruction error. The method proposed in [29] introduces the application of sparse coding (related to independent component analysis (ICA)) for image wavelet-like extraction while using soft thresholding [36] operators on sparse coding to further reduce noise. The proposed sparse code shrinkage [27] uses three distinct stages to achieve denoising: 1) determine the orthogonal transformation matrix M by any sparse methods ($s = Mx$; where x is the input data and s is the a vector representing the linearity transformed components), 2) the sparse noisy components $s(t)$ are computed for each x observation while shrinkage is applied to the non-linearity functions for every observation and to each components, and 3) perform the reverse step 1 (x=$M^T s$) to obtain new estimates of input.

The method proposed in [30] uses an iterative fixed-point method to obtain higher convergence speed. It is based on higher order statistics, like kurtosis (a measure of data independency), and uses the FASTICA algorithm [279]-[283] that implements them.

3.6.5. Non-data adaptive filters

Non-data adaptive filters utilize the local properties of the noisy image (such as local windows and local blocks) to approximate the denoised one.

3.6.5.1. Spatial frequency domain filters

Low pass filters: The lowpass filters presented in [9] and [20] remove the high frequency signals present in the image that exceed a specified cut-off frequency. Because Gaussian noise is largely made up of high frequency components, this type of filter is suitable to remove it. As this kind of filter causes image smoothing when the high frequency components are removed, its cut-off frequency can be increased to retain image details and reduce smoothing.

Gaussian filters: The authors in [284] use the relationship between Gaussian filters, images, and noise statistics to design an optimal filter. They define an error term, after frequency domain filtering based on the frequency domain Gaussian filter, as a combination of signal distortions and smoothed noise. According to the signal and noise spectrums, if the filter variance is 0, the filter does not affect the signal and the signal to noise ratio of the output is equal to that of the input. By increasing the filter's variance, and consequently decreasing its cut-off frequency, until such a point where the maximum output signal to noise ratio is obtained the out-of-band noise is removed without introducing any further distortion of the input signal. However, if the variance is increased past the optimal point, the filter will not only remove the out-of-band noise but will also introduce image distortions, leading to a decrease in the output's signal to noise ratio. Therefore, in order to design an optimal filter, the authors define a process to obtain the optimal variance.

3.6.5.2. Wavelet and Nonlinear spatial domain filters

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:

4th Order Partial differential equation: The method introduced in [285] uses dual-tree wavelet transforms to decompose the input image into sub-images. The dual-tree wavelets have better reconstruction performance than discrete wavelet transform. A 4th order PDE with L^2 –curvature gradient is applied on each sub-band before applying adaptive thresholding to approximate each sub-band and obtain the denoised image by applying inverse wavelet transform.

Anisotropic: The wavelet-based multiscale anisotropic diffusion method (WMSAD) presented in [286] removes common problems from traditional anisotropic diffusion filters like high noise levels and lack of reliable PDE edge stopping criterion. In their implementation, the noisy image is decomposed into four levels using dyadic wavelet transform (DWT) [287, 288] to get its wavelet components. Based on the region's texture and smoothness, each scale's wavelet transform is classified as regular or irregular by using a modified version of the multi-scales singularity detection algorithm presented in [289]. Optimal MMSE-based filtering is applied to the wavelet transform components, to reduce overall noise without affecting edge sharpness, before filtering the components with anisotropic diffusion and reconstructing the final denoised image through an inverse DWT. The authors of this method claim to obtain better result than complex wavelet transform - Bivariate shrinkage filter (CDWT-BSF) [290], Gaussian scale mixture (GSM) [216], and efficient wavelet-based image denoising (EWID) [291]. The method presented in [292] uses a stationary wavelet transform (SWT) which, unlike DWT, does not down-sample after each step to get the wavelet coefficients. Instead, said coefficients are smoothed by adaptive anisotropic diffusion with regularized P-M [293] before producing the denoised image through an inverse SWT with the diffusion details.

They claim to obtain better results rather than WMSAD [286], Prob-shrink [294], and standalone regularized P-M [168].

Gaussian and Bilateral: The method proposed in [295] uses Gaussian (good for flat regions) and bilateral filtering (to preserves the edges and sharp boundaries) to remove noise from the images. They start by calculating the difference between the noisy image and the Gaussian and bilateral filtering results. Thereafter, a wavelet transform with Bayesian thresholding is applied to said difference to estimate the details that were removed by the previous step before finally obtaining the denoised image by summing (applying inverse wavelet transform) the reconstructed image and the output of Gaussian/bilateral filter. This method claims to get better result than bilateral filtering (BF), multiresolution bilateral filtering (MRBF) [296], non-local mean (NLM) [23], BLS-GSM [216], and kernel based denoising [297].

Non-local mean: The method proposed in [298] uses non-local means (NLMs) with wavelet domain Wiener filtering to respectively remove high and low frequency noise. A Wiener filter is used to compensate for the poor low frequency noise removal performance of NLMs (which are less sensitive to small image variations) [299]. The filtering process starts by estimating the noise variance in three orthogonal wavelet domain levels; to be used to approximate the maximum likelihood estimator [300]. The Wiener filter is then used as an MMSE estimator to estimate each sub-band's noise-free coefficients from the noisy ones before reconstructing the final image. This method claims to obtain better performance than wiener filtering [35] or NLM [23] alone.

The method in [301] uses wavelet domain non-local means to decompose the input image into sub-images by using wavelet transforms. It uses shrinkage rules, as discussed in [294], for the given wavelet coefficients to propose a method that obtains each sub-band global bandwidth and noisy coefficients local bandwidth. The N-local mean is applied to each coefficient by using its local bandwidth before using inverse wavelet transforms to reconstruct the denoised image. The authors claim to obtain better results than total variation [144], NLM [23], Prob-shrink [294], MRBF [296], BLS-GSM [216], and BM3D [209].

Total variation: The method introduced in [302] uses wavelet transforms to decompose the input image into reduced noise Low frequency-Low frequency (LL) sub-bands. This less noisy bands are filtered with a single iteration of a variation filter. The horizontal, vertical, and diagonal edges of the LL sub-band need to be later computed to estimate the High frequency-Low frequency (HL), Low frequency-High frequency (LH) and High frequency-High frequency (HH) sub-bands. This method, which is an improvement on the total variation method, claims to obtain better performance than total variation [144], Visushrink [37], Sure shrink [303], and Wiener filter [19, 20]. The method depicted in [304] uses dual-tree complex wavelet transform (DT-CWT) [305] in order to remove noisy coefficients, before applying an improved version of total variation [149] to remove the remaining noise. They also apply soft thresholding [36] with SURE estimation [303] to the sub-bands.

Wiener filters: The method proposed in [34] uses two different wavelet transforms for standard wavelet shrinkage with hard thresholding [36] (using trustworthy coefficients to estimate a pilot signal) to design an empirical Wiener filter. The authors of [33] introduce a filter that minimizes the MSE to obtain the optimal first order FIR Wiener filter [306]. Exponentially decaying autocorrelation is used for the filter's design to avoid complexity and sensitivity problems that arise from increasing the filter's order. They also adapt the FIR filter to the local statistic to decrease the negative effects of the wavelet coefficient's non-constant autocorrelation structure. Denoising via block Wiener filtering in wavelet domain [31] uses orthogonal wavelet decomposition (statistical properties of the coefficients) and Wiener filters to achieve denoising. This process divides the sub-bands into blocks and calculates their variance as an estimate of the noisy image. The local variances of the sub-band's uncorrupted wavelet coefficients are later used for the Wiener filter. The method in [35] is based on a simplified Wiener filter that uses the multiplication of Wiener coefficients and the neighborhood coefficients to obtain the local expected square error (LESE) and an optimal threshold level used as a preprocessing step that can improve the wiener filter's performance. This threshold should be less than the result of multiplying the noise variance by a constant $(1+\sqrt{\frac{2}{N}})$, where N is size of the window. A partially denoised subset of the neighborhood of coefficients is obtained by applying the previously computed threshold and a second convolution pass is done to obtain better performance, making the noise clusters smaller.

PCA as a step in denoising: Wavelet domain principal component analysis (PCA) denoising is used in [307]. Through L-steps decomposition, the image is decomposed into $2^L \times 2^L$ sub-bands obtained by multiplying the input signal by the Hurwitz matrix [308]. A Karhunen-Loeve (KL) transform (akin to PCA) is applied to the resulting matrix to completely de-correlate the frequency sub-bands and cluster the energy of the signal into a few large coefficients. This transform uses eigenvectors and performs better at separating the noise from the signal of interest. Two-phases thresholding (containing hard thresholding [36]) and an exponential shrinkage function [309] are applied to each of the sub-band's coefficients.

Shrinkage rules-based filter: A classification of different shrinkage rules is performed in [310]. These rules separate noisy wavelet coefficients from non-noisy ones using thresholding. Among examples of these types of filters are:

- ***Soft and hard thresholding***: Hard thresholding [36] keeps the absolute wavelet coefficient (c) values if they are greater than a set threshold (λ) and soft thresholding (used to remove the artifacts introduced by hard thresholding) reduces the value of the wavelet coefficients (c) as $sgn(c)(|c|-\lambda)$ if their absolute value is above said threshold (λ) or sets them to zero otherwise.
- ***Hyperbola function thresholding***: Hyperbola function thresholding is introduced in [40] to reduce

the attenuation of large wavelet coefficients caused by previous methods. If the absolute wavelet coefficients (c) are above a set threshold (λ), they are reduced as $sgn(c)\sqrt{c^2-\lambda^2}$, otherwise they are set to zero.

- ***Firm thresholding***: An improvement on hard and soft thresholding with better MSE is presented in [311] where two thresholds are used. Here, the wavelet coefficients (c) are left unchanged if their absolute values are above a predefined threshold (λ_2), reduced as $sgn(c)\frac{\lambda_2(|c|-\lambda_1)}{\lambda_2-\lambda_1}$ if they are between (λ_2) and (λ_1), or set to zero otherwise.
- ***Non-negative Garrote thresholding***: The authors of [312] introduce an improvement on previous thresholding method; resulting on a reduction of the number of thresholding operations. In this instance, if the absolute value of wavelet coefficients (c) are above a threshold (λ), they are reduced as $c-\frac{\lambda^2}{c}$, or set to zero otherwise.
- ***Smoothly clipped absolute deviation (SCAD) thresholding***: The thresholding method introduced in [313] leaves the wavelet coefficients unchanged if their absolute value is larger than a threshold $(\alpha\lambda)$. However, if their absolute values are between $(\alpha\lambda)$ and (2λ), they are reduced as $\frac{(\alpha-1)c-\alpha\lambda sgn(c)}{\alpha-2}$, or as $sgn(c)\max(0,|c|-\lambda)$ otherwise. The constant $\alpha=3.7$ is empirically chosen in this case.
- ***Exponential thresholding***: The authors in [314] introduce improvements on continuous exponential thresholding. This new exponential thresholding is defined as $sgn(c)(e^{\frac{|c|}{\lambda}}-1)$ for which λ is the threshold level and c is the wavelet coefficient. This method has better PSNR, keeps more image details, and preserves more wavelet coefficients than previous methods.

Non-linear thresholding-based filters: These methods specify thresholds (λ) for their shrinkage rules. Among such filters are:

- ***Non-adaptive thresholding estimation-based filters:*** Visushrink thresholding [37], or universal thresholding, is a non-adaptive threshold estimation defined by $\sigma\sqrt{2logn}$, where σ is the noise variance and n is number of image pixels. This achieves good performance when reconstructing noise-free pixels to obtain smooth images.
- ***Adaptive thresholding filters-SURE:*** Sub-band universal and SURE (Stein`s unbiased risk estimation) thresholding is proposed in [303] as described in (48); where it is applied to the non-sparse coefficients case.

$$\lambda = argmin_{t\geq 0}\left[N-2[1:N]+\sum_{x,y=1}^{N}(\min(|c|,t))^2\right] \quad (48)$$

Here, N is the number of coefficients (c) in each sub-band. A weighted sum of non-linear thresholding-based denoising functions and MSE (estimated by SURE) minimization are used for denoising [315] along with an inter-scale orthonormal wavelet thresholding algorithm. The SURE-LET based filters described in [316] and [317] use linear extension of thresholds to rapidly solve the linear systems of equations. In [316], the SURE-LET approach increases the performance of the un-decimated orthonormal wavelet transform sub-band minimization. The method proposed in [317] applies inter-scale multichannel SURE-LET thresholding, depending on coefficients across multiple channels, after the non-redundant non-orthonormal wavelet transform. The local linear SURE-based method (LLSURE) introduced in [318] is also based on local linear models like [319], $LLSURE(y,r,\sigma^2)$ with y as the input signal, r as the window radius, and σ^2 being the noise variance. This method consists of 3 stages: 1) perform a local neighborhood (every position) affine transform of the input, 2) determine the coefficients through an MSE estimate (SURE) minimization, and 3) obtain the final results by averaging all the filtered output patches. The reconstruction of multi-scale edge-preserving decomposition is achieved by using $LLSURE$ k-times while increasing σ^2 at each iteration. The authors of [318] claim that the LLSURE approach has better performance than SURE-LET.

- ***Adaptive thresholding filters-Bayes:*** Bayesian threshold estimation [41, 42] is as defined in (49), and is based on Gaussian distribution and a Bayesian mathematical process. In (47) $\sigma_G^2=\frac{1}{N}\sum_{x,y=1}^{N}G_{x,y}^2$, N is the number of coefficients $(G_{x,y})$ in each sub-band, and σ_n^2 is the noise variance. The methods proposed in [38-40] are other Bayesian approaches used to estimate a similar threshold

$$\lambda=\frac{\sigma_n^2}{\sqrt{\max(\sigma_G^2-\sigma_n^2,0)}} \quad (49)$$

- ***Adaptive thresholding filters - Cross validation:*** Cross validation is used to minimize the mean integrated square error (MISE) between the wavelet shrinkage function and the true function (the function used for the Wavelet Transform). 2^k-fold cross validation, introduced in [43], is a k-dimensional (k=2 for 2-dimensional images) cross validation over all indices in a sub-grid. A general cross validation (GCV) approach without knowledge of noise variance, based on the method proposed in [320], is proposed in [44]. Iterative GCV (IGCV), introduced in [321], is yet another improvement on GCV for image denoising. This iterative method shows to be faster, more effective, and less time consuming than non-iterative ones.
- ***Adaptive thresholding filters-Spatially adaptive:*** Spatial adaptive wavelet thresholding is proposed in [322] and [323]. The method in [322] is based on edge detection and image segmentation. It classifies

Table 1:
Adaptive thresholding estimation-based filters comparison

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

the image into three regions (edge, texture, and non-texture (smooth)) and defines a different threshold for each region based on its noise power and a constant (0.4 for edges, 1.4 for textures, and 1.8 for non-textures). On the other hand, the method introduced in [323] models the wavelet coefficients as random variables with generalized Gaussian distributions (GGD). This model estimates unknown pixel-level parameter using context modeling (a common technique used in image compression to adapt the coder to changing image characteristics) and uses this spatial mixture of distribution to adapt the thresholding parameters for each coefficient based on noise variance ($\sigma_n^2 = \frac{median(|y_i|)}{0.6745}$) [37]. Translation-invariant (TI) denoising is proposed in [324] to remove artifacts. Because TI denoising is equivalent to thresholding the over-complete representation of the non-subsampled filter bank up to some scaling in the threshold, spatial adaptive thresholding can be extended as an over-complete expansion that separates the coefficients into 2^{2s} sets of uncorrelated coefficients, where *s* is the level of decomposition.

- ***Adaptive thresholding filters-Bivariate****:* The bivariate Laplacian probability density function (PDF) in [325] is a modification of Bayesian estimation that models the interdependency of adjacent wavelet coefficients to obtain a non-linear threshold. It defines its shrinkage function based on noise variance [37], marginal variance, and the noisy models of wavelet coefficients. The four non-Gaussian bivariate distributions, introduced in [326], are modifications of Bayesian estimators that model the interdependency of adjacent wavelet coefficients to obtain the bivariate shrinkage functions. The authors of [326] claim that their method achieves better performance than Visu, SURE, and Bayes shrinking. The locally adaptive denoising method described in [290] uses the bivariate shrinkage function from [325] and is based on both orthogonal and dual-tree complex wavelet transform (DTCWT) [305]. The method proposed in [327] uses dual contourlet transforms (DCT) (an improved wavelet transform) and dual tree complex wavelet transform (DTCWT) before using a bivariate threshold function on the coefficients like in [290]. For comparative purposes, Table 1 illustrates the peak signal to noise ratio comparison of filters based on adaptive thresholding estimation for the Lena image.

Wavelet coefficient-based filters: These filters can be categorized as deterministic, statistically-based or of non-orthogonal type.

- ***Deterministic based filters***: The wavelet-based method presented in [45] uses hierarchical nonlinear approximation on trees, where the wavelet coefficients are ordered according to their size, to select the largest coefficients first and the small ones later. Based on this ordered tree, an optimal tree is defined that minimizes the wavelet coefficients' sum of squares before a hierarchical representation of wavelet decomposition is established to estimate the noise-free coefficients through hard thresholding.

- ***Statistical based filters****:* The authors in [46] present connections between shrinkage methods and maximum A posteriori (MAP) estimation that are used to perform Bayesian estimation. They introduce new complexity priors (based on Rissanen`s universal prior on integers) and estimate a minimum description length (MDL). This method leads to the development of shrinkage rules based on generalized Gaussian distribution (GGD) complexity prior. A modification of the wavelet coefficients (w_I) is proposed in [48] by using shrinkage rules with Bayesian marginal probabilities, obtained by a combination of smoothness and geometrical constraints. These probabilities are approximated by a stochastic sampling of Markov random fields (MRF) with the smoothness measures based on wavelet coefficient approximations of the local Hölder exponent. However, as the iterative training time required to fit a Hidden Markov tree (HMT) is a considerable drawback, a simplified HMT is proposed in [47]. This method uses the inherent self-similarity of natural images to predict the statistical behavior of images at fine scales. The simplified HMT is specified by two models (iHMT and uHMT), both based on Hidden Markov tree (HMT) for wavelet coefficients. The iHMT model reduces the model's complexity by setting nine easily trained parameters that are independent from the image's size and the number of wavelet scales. The Bayesian universal HMT (uHMT) model fixes the parameters with no training. Finally, the authors of [47] also propose a modification of the shift-invariant [324, 328] denoising algorithm (fast shift-invariant HMT) to reduce its computational complexity by using the redundancies in the wavelet representation between shifts of the image to remove artifacts that exists in orthogonal wavelet transform.
- ***Non-orthogonal wavelet-based filters****:* The method introduced in [49] uses non-decimated wavelet transforms, instead of orthogonal ones, and shift invariance. It uses a combination of Baylkin`s algorithm and wavelet denoising with N wavelet coefficients (for redundancy) and applies it to 1-D signals such as Blocks, Bumps, and Doppler. Translation invariant (TI) modifications are proposed in [50], using multiwavelets, and in [51], using

minimum description length (MDL). The test signals in [50] are 1-D Blocks, Bumps, Heavisine and Doppler signals and the test signal in [51] is defined as a 1-D synthetic signal.

4. Comparison

Table 2 summarizes the key points and limitations one ought to consider in the implementation of the numerous filters covered in this review article. Table 3 compares the Peak Signal to Noise Ratio (PSNR), correlation, and Structural Similarity Index Metric (SSIM) of some of the discussed impulse denoising filters. Moving windows of 3x3, 5x5, and 9x9 sizes were respectively used for images with 10%, 60%, and 90% noise intensity. SAMFWMF [108] uses an adaptive window that starts at 3x3 and can be as large as the corresponding size for the current noise level.

Table 4 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 images used in [231]. Table 5 compares the Peak Signal to Noise Ratio (PSNR) of some of the discussed Gaussian denoising filters. The methods in the table 5 are based on the traditional methods (spatial non-linear filters) and the PSNR is calculated on the basis of one testing image, the Parrot image as in [177], and is deemed sufficient for this type of comparison involving PSNR. Table 6 compares the Structural Similarity Index Metric (SSIM) of some of the discussed Gaussian denoising filters. The Bird image is used for comparison in table 6 [177] and is also deemed sufficient for this type of comparison. Table 7 compares the average Peak Signal to Noise Ratio (PSNR) [237] of some of the discussed Gaussian denoising filters on one-thousand test images from the PASCAL dataset [329]. Most of the filters present in Table 7 are based on convolutional neural networks (CNN).

5. Conclusion

Through this literature survey, we have presented a comprehensive survey of impulse and Gaussian denoising filters which are applied to digital images to reduce their impulse-Gaussian noise combination. We use a random noise model comprised of impulse (salt and pepper) and Gaussian noise. We have explained the noise models and the respective denoising filters, as well as classified them according to their types and domains 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 model and denoising method they could use and which metrics they would contemplate using for their own research as a preprocessing step when dealing with noisy images.

ACKNOWLEDGMENTS

We are grateful for the continued support from the National Science Foundation (NSF) under NSF grants CNS-1532061, CNS-133892, CNS-1551221, CNS-1429345. We also greatly appreciate the support of the Ware Foundation. This work is also supported through a graduate research fellowship (NSF-GRFP) for Mr. Harold Martin.

REFERENCES

[1] A. C. Bovik, Hand Book of Image and Video Processing, 2nd ed., Cambridge, MA, USA: Academic press, 2000.

[2] K.Prathiba, R. Rathi, C. Seldev Christopher, Random valued Impulse Denoising using Robust Direction based Detector, in: Proceedings of the IEEE International Conference on Information & Communication Technologies (ICT), 2013.

[3] G. E. Healey, R. Kondepudy, Radiometric CCD Camera Calibration and Noise Estimation, IEEE Trans. Pattern Anal. Mach. Intell., 16 (3) (1994) 267-276.

[4] 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.

[5] Y. Chen, M. Das, An Automated Technique for Image Noise Identification Using a Simple Pattern Classification Approach, in: Proceedings of the IEEE 50th International Symposium on Circuits Systems, 2007.

[6] F. Li, J. Fan, Salt and Pepper Noise Removal by Adaptive Median Filter and Minimal Surface Inpainting, in: Proceedings of the IEEE 2th International Congress on Image and signal processing, 2009.

[7] W. Liu and W. Lin, Additive White Gaussian Noise Level Estimation in SVD Domain for Images, IEEE Trans. Image Process., 22 (3) (2013) 872-883.

[8] J. Nakamura, Image Sensors and Signal Processing for Digital Still Cameras, Boca Raton, FL, USA: CRC Press, 2005.

[9] 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.

[10] C. Liu, IEEE, Richard Szeliski, Sing Bing Kang, C. Lawrence Zitnick, William T. Freeman, Automatic Estimation and Removal of Noise from a Single Image, IEEE Trans. Pattern Anal. Mach. Intell., 30 (2) (2008) 299-314.

Table 2:
Summary of the key points and limitations in the implementation of the numerous filters reviewed

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

Table 3:
PSNR, Correlation and SSIM comparison of some Impulse denoising filters

		Lena			***Cameraman***		
DMFUTMF [96]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [59]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [117]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [79]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [99]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [107]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [93]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [105]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [82]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [88]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [86]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [94]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 [108]	*Noise level*	*Correlation*	*PSNR*	*SSIM*	*Correlation*	*PSNR*	*SSIM*
	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 4:
PSNR comparison of some Gaussian denoising filters

Gaussian noise (standard deviation)	*Type of denoising filters*						
	BM3D [209]	WNNM [228]	CSF [219]	MLP [234]	TNRD [218]	DnCNN-S [231]	DnCNN-B [231]
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 5:
PSNR comparison of some Gaussian denoising filters-continued for other methods

Gaussian noise (standard deviation)	*Type of denoising filters*						
	Catte [168]	TV [144]	EAD [178]	CTD [170]	MPM [179]	Improved Anisotropic diffusion 1 [177]	Improved Anisotropic diffusion 2 [177]
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 6:
SSIM comparison of some Gaussian denoising filters

Gaussian noise (standard deviation)	*Type of denoising filters*						
	Catte [168]	TV [144]	EAD [178]	CTD [170]	MPM [179]	Improved Anisotropic diffusion 1 [177]	Improved Anisotropic diffusion 2 [177]
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

Table 7:
PSNR comparison of some Gaussian denoising filters

Gaussian noise (standard deviation)	*Type of denoising filters*				
	BM3D [209]	MLP [234]	TNRD [218]	IRCNN-S [235]	FCNN [237]
0.1	34.26	34.29	-	-	34.87
0.15	32.10	-	32.35	32.66	32.79
0.25	29.62	29.95	29.90	30.25	30.33
0.35	28.14	28.49	-	-	28.88
0.50	26.61	26.98	26.91	27.22	27.32
0.65	25.64	26.07	-	-	26.30
0.75	25.12	25.54	-	-	25.75

[11] C. Pal, A. Chakrabarti, R. Ghosh, A Brief Survey of Recent Edge-Preserving Smoothing Algorithms on Digital Images, Procedia Computer Science, (2015) 1-40.

[12] A. Buades, B. Coll, J-M. Morel, A review of image denoising algorithms, with a new one, Multiscale Model Simul. J.: A SIAM Interdisciplinary J, 4 (2) (2005) 490–530.

[13] P. Jain, V. Tyagi, A survey of edge-preserving image denoising methods, Information System Frontiers, 18 (1) (2016) 159–170.

[14] M. C. Motwani, M. C. Gadiya, R. C. Motwani, F. C. Harris, Survey of image denoising techniques, in: Proceedings of the Global Signal Processing Expo. Conference (GSPx), 2004, pp. 27-30.

[15] J. Mohan, V. Krishnaveni, Yanhui Guo, A survey on the magnetic resonance image denoising methods, Biomed. Signal process. control, 9 (2104) 56-69.

[16] P. Jain, V. Tyagi, Spatial and frequency domain filters for restoration of noisy images, IETE J. Education., 54 (2) (2013) 108–116.

[17] 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.

[18] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. Weinberger, Universal discrete denoising: Known channel, IEEE Trans. Inform. Theory, 51 (1) (2005) 5-28.

[19] A. K. Jain, Fundamentals of digital image processing, 1st ed., Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[20] R. C. Gonzalez, R. E. Woods, Digital image processing, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2006.

[21] I. Pitas, & A. N. Venetsanooulos, Nonlinear digital filters: Principles and applications, Berlin, Germany: Springer Science & Business Media, 1990.

[22] P. Perona, J. Malik, Scale-Space and Edge Detection Using Anisotropic Diffusion, IEEE Trans. Pattern Anal. Mach. Intell, 12 (7) (1990) 629-639.

[23] 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), 2 (2005) 60- 65.

[24] C. Tomasi, R. Manduchi, Bilateral Filtering for Gray and Color Images, in: Proceedings of the IEEE International Conference on Computer vision (ICCV), 1998.

[25] Y.L. You, M. Kaveh, Fourth-order partial differential equation for noiseremoval, IEEE Trans. Image Process., 9 (10) (2000) 1723–1730.

[26] J.L. Starck, E.J. Candes, D.L. Donoho, The Curvelet transform for image denoising, IEEE Trans. Image Process., 11 (6) (2002) 670–684.

[27] M.N. Do, M. Vetterli, The contourlet transform: an efficient directional mul-tiresolution image representation, IEEE Trans. Image Process., 14 (12) (2005) 2091–2106.

[28] A. Jung, An introduction to a new data analysis tool: Independent component analysis, in: Proceedings of the workshop GK. Nonlinearity, 2001.

[29] A. Hyvarinen, E. Oja, P. Hoyer, J. Hurri, Image feature extraction by sparse coding and independent component analysis, Presented at the IEEE 4th International Conference on Pattern Recognition, 1998.

[30] 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.

[31] V. Strela, Denoising via block Wiener filtering in wavelet domain, In: Proceeding Springer, European congress of mathematics, 2001, pp. 619-625.

[32] H. Choi, R. Baraniuk, Analysis of wavelet-domain wiener filters, in: Proceedings of the IEEE International Symposium on time-frequency and time-scale analysis, 1998, pp. 613-616.

[33] H. Zhang, A. Nosratinia, R. O. Wells, Image denoising via wavelet-domain spatially adaptive FIR Wiener filtering, in: Proceedings of the IEEE International Conference on Acoust Speech, Signal Processing, 2000, pp. 2179–2182.

[34] 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, 3169, 1997, pp. 389–399.

[35] M. Kazubek, Wavelet domain image denoising by thresholding and wiener filtering, IEEE Signal Process. Lett., 10 (11), 2003, pp. 324–326.

[36] D. L. Donoho, De-noising by soft-thresholding, IEEE Trans. on Inform. Theory, 41 (3) 1995 613–627.

[37] D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika, 81 (1994) 425–455.

[38] F. Ruggeri, B. Vidakovic, A Bayesian decision theoretic approach to wavelet thresholding, Journal American Statistical Association, 93 (1998) 173–179.

[39] F. Abramovitch, T. Sapatinas, B. W. Silverman, Wavelet thresholding via a Bayesian approach, Journal Roy Statistic Society (B), 60 (4) (1998) pp. 725–749.

[40] B. Vidakovic, Nonlinear wavelet shrinkage with Bayes rules and Bayes factors, Journal American Statistical Association, 93 (441) (1998) 173–179.

[41] H. A. Chipman, E. D. Kolaczyk, E., R. E. McCulloch, Adaptive Bayesian wavelet shrinkage, Journal of the American Statistical Association, 92 (440) (1997) 1413–1421.

[42] S. G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Process., 9 (9) (2000) 1532–1546.

[43] G. P. Nason, Wavelet shrinkage using cross-validation, Journal Roy Statistic Society (B), 58 (1996) 463–479.

[44] N. Weyrich, G. T. Warhola, Wavelet shrinkage and generalized cross validation for image denoising, IEEE Trans. Image Process., 7 (1) (1998) 82–90.

[45] R. G. Baraniuk, Optimal tree approximation with wavelets, In: Proceeding SPIE Tech. Conference on Wavelet Applications in Signal and Image Processing VII, 3813, 1999, pp. 196-207.

[46] P. Moulin and J. Liu, Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors, IEEE Trans. Inform. Theory, 45 (3) (1999) 909-919.

[47] J. K. Romberg, H. Choi and R. G. Baraniuk, Bayesian wavelet domain image modeling using hidden Markov models, IEEE Trans. Image Process., 10 (7) (2001) pp. 1056-1068.

[48] M. Malfait and D. Roose, Wavelet based image denoising using a Markov Random Field a priori model, IEEE Trans. Image Process., 6 (4) (1997) pp. 549–565.

[49] 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.

[50] T. D. Bui and G. Chen, Translation-invariant denoising using multiwavelets, IEEE Trans. Signal Process., 46 (12) (1998) 3414-3420.

[51] I. Cohen, S. Raz and D. Malah, Translation invariant denoising using the minimum description length criterion, Signal Process., 75 (3) (1999) 201-223.

[52] J. L. Mateo, A. Fernandez-Caballero, Finding out general tendencies in speckle noise reduction in ultrasound images, Expert systems with applications, 36 (4) (2009) 7786-7797.

[53] A. M. Eskicioglu, P. S. Fisher, Image Quality Measures and Their Performance, IEEE Trans. Com, 43 (12) (1995) 2959-2965.

[54] A. M. Eskicioglu and P. S. Fisher, A survey of quality measures for gray scale image compression, In: Proceeding of the Space and Earth Science Data Compression Workshop (NASA Conference Publication 3191), 1993, pp. 49-61.

[55] 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, 2012, pp. 108 – 117.

[56] Z. Wang, A. C. Bovik, H. Rahim Sheikh, E. P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Trans. Image Process., 13 (4) (2004) 600-612.

[57] Z. Wang, A. C. Bovik, A Universal Image Quality Index, IEEE Signal Process. Lett., 9 (3) (2002) 81-84.

[58] H. Hwang and R. A. Haddad, Adaptive Median Filters: New Algorithms and Results, IEEE Trans. Image Process., 4 (4) (1995) 499-502.

[59] 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., 14 (10) (2005) 1479-1485.

[60] M. Nikolova, A variational approach to remove outliers and impulse noise, Journal of Mathematical Imaging and Vision, 20 (2004) 99–120.

[61] M. Nikolova, Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers, SIAM Journal on Numerical Analysis, 40 (2002) 965–994.

[62] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, Deterministic edge-preserving regularization in computed imaging, IEEE Trans. Image Process., 6 (2) (1997) 298–311.

[63] 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., 46 (1) (1999) 70-80.

[64] 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., 48 (8) (2001) 784-789.

[65] H-L. Eng, K-K. Ma, IEEE Trans. Image Process., 10 (2) (2001) 242-251.

[66] Y. Hashimoto, Y. Kajikawa, and Y. Nomura, Directional Difference-Based Switching Median Filters, Electronics and Communications in Japan, part 3, 85 (3) (2002) 22-32.

[67] S. Zhang, M. A. Karim, A New Impulse Detector for Switching Median Filters, IEEE Signal Process. Lett., 9 (11) (2002) 360-363.

[68] 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., 54 (4) (2008) 1920-1927.

[69] V. Crnojevic´, V. ˇSenk and Ž. Trpovski, Advanced Impulse Detection Based on Pixel-Wise MAD, IEEE Signal Process. Lett., 11 (7) (2004).

[70] T. Sun and Y. Neuvo, Detail-preserving median based filters in image processing, Elsevier, Pattern Recognition Letters, 15 (1994) 341–347.

[71] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, A New Efficient

Approach for the Removal of Impulse Noise from Highly Corrupted Images, IEEE Trans. Image Process., 5 (6) (1996) 1012–1025.

[72] J.-S. Lim, Two-Dimensional Signal and Image Processing, Englewood

Cliffs, NJ, USA: Prentice-Hall, 1990.

[73] V.V. Khryaschev, 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.

[74] 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, 24 (7) (2013) 956–967.

[75] T. Chen, K.-K. Ma, and L.-H. Chen, Tri-State Median Filter for Image Denoising, IEEE Trans. Image Process., 8 (12) (1999) 1834-1838.

[76] S.-J. Ko and Y.H. Lee, Center weighted median filters and their applications to image enhancement, IEEE Trans. Circuits Syst., 38 (9) (1991) 984–993.

[77] Y. Dong, S. Xu, A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise, IEEE Signal Process. Lett., 14 (3) (2007).

[78] C. T Lu, T. C. Chou, Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter, Pattern Recognit. Lett., 33 (10) (2012) 1287–1295.

[79] 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), 2010, pp. 409-413.

[80] K. S. Srinivasan, D. Ebenezer, A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises, IEEE Signal Process. Lett., 14.(3) (2007) 89-192.

[81] P.-E. Ng, K.-K. Ma, A Switching Median Filter With Boundary Discriminative Noise Detection for Extremely Corrupted Images, IEEE Trans on Image Process., 15 (6) (2006) 1506-1516.

[82] 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, 22 (3) (2013) 1223-1232.

[83] X. Zhang and Y. Xiong, Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter, IEEE Signal Process. Lett., 16 (4) (2009) 295-298.

[84] Y. S. Choi and R. Krishnapuram, A robust approach to fuzzy enhance-ment based on fuzzy logic, IEEE Trans. Image Process. , 6 (6) (1997) 808–825.

[85] P. Zhang, F. Li, A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise, IEEE Signal Process. Lett., 21 (10) (2014).

[86] C. Kandemir, C. Kalyoncu, Ö. Toygar, A weighted mean filter with spatial-bias elimination for impulse noise removal, Digital Signal Processing, 46 (2015) 164–174.

[87] Z. Zhou, Cognition and Removal of Impulse Noise with Uncertainty, IEEE Trans. Image Process., 21 (7) (2012) 3157-3167.

[88] C. Kalyoncu1, Ö. Toygar, H. Demirel, Interpolation-based impulse noise removal, IET Image Process., 7 (8) (2013) 777-785.

[89] D. Li and Y. Du, Artificial Intelligent with Uncertainty, Boca Raton, FL, USA, CRC Press, 2007.

[90] D. Li, C. Liu, W. Gan, A new cognitive model: Cloud model, International Journal of Intelligent System, 24 (3) (2009) 357–375.

[91] Z. Li, G. Liu, Y. Xu , Y. Cheng , Modified directional weighted filter for removal of salt & pepper noise, Pattern Recognit. Lett. 40 (2014) 113–120.

[92] Z. Li, Y. Cheng, K. Tang, Y. Xu, D. Zhang, A salt & pepper noise filter based on local and global image information, Neurocomputing, 159 (2015) 172–185.

[93] 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 Recognit. Lett., 80 (2016) pp. 188–199.

[94] X. Wang, S. Shen, G. Shi, Y. Xu, P. Zhang, Iterative non-local means filter for salt and pepper noise removal, Journal of Visual Communication and Image Representation, 38 (2016) 440–450.

[95] M. S. Nair, K. Revathy, R. Tatavarti, An Improved Decision Based Algorithm for Impulse Noise Removal, in: Proceedings of the IEEE Congress on Image and Signal Processing, 2008, pp. 426–431.

[96] 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), 2009, pp. 1-6.

[97] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, C. H. PremChand Removal of High Dendity Salt and Pepper noise Trhough Modified Decision based Unsymmetrical Trimmed Median Filter, IEEE Signal Process. Lett., 18 (5) (2011).

[98] S. Vishaga, Sreejith. L. Das, A Survey on Switching Median Filters for Impulse Noise Removal, in: Proceedings of the IEEE International Conference Circuit, Power and Computing Technologies (ICCPCT), 2015.

[99] 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), 2012, pp.1-5.

[100] T. Veerakumar, S. Esakkirajan, IIa. Vennila, An Approach to Minimize Very High Density Salt and Pepper Noise through Trimmed Global Mean, International Journal of Computer Applications, 39 (12) (2012).

[101] T. M. Benazir, B. M. Imran, Removal of High and Low Density Impulse Noise From Digital Images Using Non Linear Filter, in: Proceedings of the International Conference on Signal Processing, Image Process and Pattern Recognition (SIPP), pp. 1–5, 2013.

[102] A. K. Samantaray, P. Mallick, Decision Based Adaptive Neighborhood Median Filter, Procedia Computer Science, 48 (2015) 222-227.

[103] B. Smolka, A. Chydzinski, Fast detection and impulsive noise removal in color images, Real-Time Imaging, 11 (5-6) (2005) 389–402.

[104] J. G. Camrena, V. Gregori, S. Morillas, A. Sapena, Some improvements for image filtering using peer group techniques, Image and Vision Computing, 28 (1) (2010) 188-201.

[105] 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, 54 (2015) 595-604.

[106] 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 Processing, 7 (6) (2013) 1145-1157.

[107] 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, 2015, pp.96-101.

[108] 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., 27 (11) (2018) 5475-5490.

[109] 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, Philadelphia, USA, 2017.

[110] M. S. A. Alias, N. Ibrahim, Z. M. Zin, Salt and Pepper Noise Removal by Using Improved Decision Based Algorithm, in: Proceedings of the IEEE 15th Student Conference on Research and Development (SCOReD), 2017.

[111] 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.

[112] R. Stevenson, G. Arc, Morphological Filters: Statistics and Further Syntactic Properties, 1EEE Trans. Circuits Syst., 34 (11) (1987) 1292- 1305.

[113] J. Song, E. J. Delp, The Analysis of Morphological Filters with Multiple Structuring Elements, Computer Vision, Graphics and Image Processing, 50 (3) (1990) 308-328.

[114] D. Ze-Feng, Y. Zhou-Ping, X. You-Lun, High Probability Impulse Noise-Removing Algorithm Based on Mathematical Morphology, IEEE Signal Process. Lett., 14 (1) (2007).

[115] 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, 2012, pp. 7-14.

[116] H. Xu , G. Zhu, H. Peng, D. Wang, Adaptive fuzzy switching filter for images corrupted by impulse noise, Pattern Recognition Letters, 25 (15) (2004) 1657–1663.

[117] S. Schulte, M. Nachtegael, V. De Witte, D. Van der Weken, E. E. Kerre, A Fuzzy Impulse Noise Detection and Reduction Method, IEEE Trans. Image Process., 15 (5) (2006) 1153-1162.

[118] E. E. Kerre, Fuzzy Sets and Approximate Reasoning, Xian, China: Xian Jiaotong University Press, 1998.

[119] V. Zlokolica, W. Philips, Motion and detail adaptive denoising of video, In: Proceeding of the IS&T/SPIE Symposium on Electronic Imaging, 5298 (5) (2004) 403–412.

[120] F. Russo and G. Ramponi, A fuzzy filter for images corrupted by impulse noise, IEEE Signal Process. Lett., 3 (6) (1996) 168–170.

[121] F Russo, G. Rumponi, Removal of impulse noise using a FIRE filter, in: Proceedings 3rd IEEE International Conference Image Processing, 1996, pp. 975–978.

[122] F. Russo, Fire operators for image processing, Fuzzy Sets and Systems, 103 (2) (1999) 265–275.

[123] K. Arakawa, Median filter based on fuzzy rules and its application to image restoration, Fuzzy Sets and Systems, 77 (1) (1996) 3–13.

[124] K. Arakawa, Fuzzy rule-based image processing with optimization, Fuzzy Techniques in Image Processing. Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 52 (2000) 222–247.

[125] C.-S. Lee, Y.-H. Kuo, and P. T. Yu, Weighted fuzzy mean filters for image processing, Fuzzy Sets and Systems, 89 (2) (1997) 157–180.

[126] C. S. Lee and Y. H. Kuo, Adaptive fuzzy filter and its application to image enhancement, Fuzzy Techniques in Image Processing. Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 52 (2000) 172–193.

[127] H. K. Kwan, Fuzzy filters for noise reduction in images, Fuzzy Filters for Image Processing. Studies in Fuzziness and Soft Computing, Berlin, Heidelberg, 122 (2003) 25–53.

[128] 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, 52 (2000) 194–221.

[129] J.-H.Wang, H.-C. Chiu, An adaptive fuzzy filter for restoring highly corrupted images by histogram estimation, in: Proceedings of the Natl. Sci. Council ROC (A), 23 (5),1999, 630–643.

[130] 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, 122 (2003) 54–71.

[131] 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.

[132] T.-H. Yu, A Fuzzy Logic-Based Predictor for Predictive Coding of Images, IEEE Trans. Fuzzy Syst., 6 (1) (1998) 153-162.

[133] D. L. Donoho, I. M. Johnstone, 1994b. Threshold selection for wavelet shrinkage of noisy data, In: Proceeding of the IEEE 16th International Conference on Engineering in Medicine and Biology Society. Engineering Advances: New Opportunities for Biomedical Engineers, 1 (1994) A24-A25.

[134] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, Wavelet Shrinkage: Asymptotia, Journal of the Royal Statistical Society (B), 57 (1995) 301-369.

[135] F. Russo, G. Ramponi, A fuzzy filter for images corrupted by impulse noise, IEEE Signal Processing Lett., 3 (1996) 168–170.

[136] F. Russo, G. Ramponi, Nonlinear fuzzy operators for image processing, Signal Processing, 38 (1994) 429440.

[137] F. Russo, G. Ramponi, Introducing the fuzzy median filter, in: Proceedings of the 7th Europ. Signal Processing Conference EUSIPCO-94, 1994, pp. 963-966.

[138 A**.** Taguchi, H. Takashima, and Y. Murata, Fuzzy filters for image smoothing, in: Proceedings of the SPIE Conference on Nonlinear Image Processing V, 1994, pp. 332-339.

[139] Wenbin Luo, Member, IEEE, Efficient Removal of Impulse Noise from Digital Images, IEEE Trans Consum. Electron., 52 (2) (2006) 523-527.

[140] E. Abreu, M. Lightstone, S. K. Mitra, K. Arakawa, A new efficient approach for the removal of impulse noise from highly corrupted images, IEEE Trans. Image Process., 5 (6) (1996) 1012-1025.

[141] R. C. Hardie, K. E. Barner, Rank conditioned rank selection filters for signal restoration, IEEE Trans. Image Process., 3 (2) (1994) 192-206.

[142] C.-S. Lee, Y.-H. Kuo, and P.-T. Yu, Weighted fuzzy mean filters for heavy-tailed noise removal, in: Proceedings of the 3rd International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society, 1995, pp. 601-606.

[143] G. Pok, J.-C. Liu, A. S. Nair, Selective removal of impulse noise based on homogeneity level information, IEEE Trans. Image Process., 12 (1) (2003) 85-92.

[144] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: non-linear phenomena, 60 (1-4) (1992) 259–268.

[145] L. I. Rudin, Images, numerical analysis of singularities and shock filters, Caltech, C.S. Dept., CA, USA, Tech. Rep. 5250:TR:87, 1987.

[146] S. Osher and L.I. Rudin, Feature oriented image enhancement using shock filters, SIAM Journal on Numerical Analysis, Anal. 27 (4) (1990) 919-940.

[147] l. Alvarez, P.L. Lions and J.M. Morel, Image selective smoothing and edge detection by nonlinear diffusion, SIAM Journal on Numerical Analysis. 29 (3) (1992) 845-866.

[148] S. Osher and J. A. Sethian, Fronts propagation with curvature dependant speed: Algorithms based on a Hamilton-Jacobi formulation, Journal of Computational Physics, 79 (1988) 12-49.

[149] A. Chambolle, An Algorithm for Total Variation Minimization and Applications, Journal of mathematical imaging and vision, 20 (1-2) (2004) 89-97.

[150] T. F. Chan, G. H. Golub, P. Mulet, A nonlinear primal-dual method for total variation-based image storation, SIAM Journal of Scientific Computing, 20 (6) (1999) 1964–1977.

[151] J. L. Carter, Dual methods for total variation—Based image restoration, Ph.D. dissertation, U.C.L.A., CA, USA, 2001.

[152] Yue Hu, Mathews Jacob, Image Recovery Using Improved Total Variation Regularization, IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011.

[153] 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.

[154] L. Hu, U KinTak, Removal of Gaussian Noise with Non-uniform Total Variation Partition, IEEE 10th Conference on Information, Communications and Signal Processing (ICICS), 2015.

[155] S. Esedoglu, S. Osher, Decomposition of images by the anisotropic rudin-osher-fatemi model, Communication on Pure Applied Mathematic, 57 (12) (2004) 1609–1626.

[156] C. Li, W. Yin, Y. Zhang, TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Algorithm, [online] available at http://www.caam.rice.edu/ ~optimization/L1/TVAL3/, Rice University, Houston, TX, 2009.

[157] L. He, H. Chen, L. Carin, Tree-structured compressive sensing with variational Bayesian analysis, IEEE Signal Process. Lett., 17 (3) (2010) 233–236.

[158] U Kin Tak, Z. Tang, D. Qi., A non-uniform rectangular partition coding of digital image and its application, in: Proceedings of the IEEE International Conference on Information and Automation, 2009, pp. 995-999.

[159] U KinTak, X. He, B. Yang, D. Qi, Z. Tang, A Novel Image Denoising Algorithm Based on Non-uniform Rectangular Partition and Interpolation, in: Proceedings of the IEEE International Conference on Multimedian Communications, 2010, pp. 9-12.

[160] T. Dan, S. Jianjun, F. Xiaosheng, S. Zhenkang, The Design of Wavelet Domain Wiener Filter and Its Application in Image Denoising, Systems Engineering and Electronics, 23 (6) (2001) 4-7.

[161] Y. Xu, J. B. Weaver, D. M. Healy, Jr., J. Lu, Wavelet transform domain filters: a spatially selective noise filtration technique, IEEE Trans. Image Process., 3 (6) (1994) 747-758.

[162] L. Sun, B. Jeon, Y. Zheng, Z. Wu, A Novel Weighted Cross Total Variation Method for Hyperspectral Image Mixed Denoising, IEEE Access, 5 (1) (2017) 27172-27188.

[163] 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.*, 40 (1) (2014) 60-66.

[164] H. Zhang, W. He, L. Zhang, H. Shen, Q. Yuan, Hyperspectral image restoration using low-rank matrix recovery, *IEEE Trans. Geosci. Remote Sens.*, 52 (8) (2014) 4729_4743.

[165] W. He, H. Zhang, L. Zhang, H. Shen, Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration, *IEEE Trans.Geosci. Remote Sens.*, 54 (1) (2016)178-188.

[166] A. Rosenfeld, M. Thurston, Edge and Curve Detection for Visual Scene Analysis, IEEE Trans. Comput. vol. C-20, 5 (1971) 562-569.

[167] A. P. Witkin, Scale-space filtering, In: Proceeding of the International Conference on Artificial Intelligence, 1983, pp. 1019-1021.

[168] F. Catte, P.-L. Lions, J.-M. Morel, T, Coll, Image Selective Smoothing and Edge detection by Non-linear Diffusion, SIAM Journal of Numerical Analaysis, 29 (1) (1992) 182-193.

[169] 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 Pro-cessing, 1999.

[170] S.-M. Chao, D.-M. Tsai, An improved anisotropic diffusion model for detail- and edge-preserving smoothing, Pattern Recognition Letters, 31 (13) (2010) 2012–2023.

[171] C. Tsiotsios, M. Petrou, On the choice of the parameters for anisotropic diffusion in image processing, Pattern Recognition, 46 (5) (2013) 1369–1381.

[172] M.J. Black, G. Sapiro, D. H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. Image Process., 7 (3) (1998) 421–432.

[173] F. Voci, S. Eiho, N. Sugimoto, H. Sekiguchi, Estimating the gradient threshold in the Perona–Malik equation, IEEE Signal Process. Mag., 21(3) (2004) 39–46.

[174] P. Mrazek, M. Navara, Selection of optimal stopping time for nonlinear Diffusion Filtering, International Journal of Computer Vision, 52 (2-3) (2003) 189-203.

[175] G. Gilboa, N. Sochen, Y.Y. Zeevi, Estimation of optimal PDE-based denoising in the SNR sense, IEEE Trans. Image Process., 15 (8) (2006) 2269–2280.

[176] G. Gilboa, Nonlinear scale space with spatially varying stopping time, IEEE Trans. Pattern Anal. Mach. Intell., 30 (12) (2008) 2175–2187.

[177] J. Xu, Y. Jia, Z. Shi, K. Pang, An improved anisotropic diffusion filter with semi-adaptive threshold for edge preservation, Signal Processing, 119 (2016) 80–91.

[178] V.B. Surya Prasath, A. Singh, Well-posed inhomogeneous nonlinear diffusion scheme for digital image denoising, Hindawi, Journal of Applied Mathematic, (2010), Article ID 763847.

[179] Y.Q. Wang, J. Guo, W. Chen, W. Zhang, Image Denoising using Modified Perona-Malik Model based on Directional Laplacian, Signal Process, 93 (9) (2013) 2548-2558.

[180] N.U. Khan, K.V. Arya, M. Pattanaik, Edge Preservation of Impulse Noise Filtered images by Improved Anisotropic Diffusion, Multimedia Tools and Application, 73 (1) (2014) 573-597.

[181] Charles Kervrann, J´er.ome Boulanger, and Pierrick Coup, Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal, International Conf. Scale Space and Variational Methods in Computer Vision, (2007) 520-532.

[182] B. Goossens, H. Luong, A. Pi ˇzurica, W. Philips, An improved non-local denoising algorithm," Presented at the International Workshop on Local and Non-Local Approximation in Image Processing, 2008.

[183] A. Dauwe, B. Goossens, H. Luong, W. Philips, A Fast Non-Local Image Denoising Algorithm, in: Proceedings of the SPIE Electronic Imaging, 6812 (2008).

[184] D. D. Muresan, T. W. Parks, Adaptive Principal Components and Image Denoising, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2003.

[185] Y. Zhan, M. Ding, F. Xiao, X. Zhang, An Improved Non-local Means Filter for Image Denoising, IEEE International Conference on Intelligent Computation and Bio-Medical Instrumentation, 2011.

[186] P. Coupe, P. Yger, S. Prima, P. Hellier, C. Kervrann, C. Barillot, An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images, IEEE Trans. Med. Imag., 27 (4) (2008) 425 –441.

[187] R. Lai, X.-X. Dou, Improved Non-local Means Filtering Algorithm for Image Denoising, IEEE 3rd International Congress on Image and Signal Processing (CISP), 2010.

[188] N. He, K. Lu, Images Denoising by Improved Non-Local Means Algorithm, 2th International Conference on Theoretical and Mathematical Foundations of Computer Science (ICTMF), 2011, pp. 33-39.

[189] M. G. Maruf, M. R. El-Sakka, Improved Non-Local Means Algorithm Based on Dimensionality Reduction, International Conference on Image Analysis and Recognition, 2015, pp. 43-50.

[190] 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, 25th Iranian Conference on Electrical Engineering (ICEE), 2017.

[191] Hossein Talebi, and Peyman Milanfar, "Nonlocal image editing, IEEE Trans, Image Process., 23 (10) (2104) 4460-4473.

[192] N. Himayat and S.A. Kassam, Approximate performance analysis of edge preserving filters, IEEE Trans. signal process., 41 (9) (1993) 2764–2777.

[193] B. K. Shreyamsha Kumar, Image denoising based on gaussian/bilateral filter and its method noise thresholding, Signal, Image and Video Processing, 7 (6) (2013) 1159–1172.

[194] K. N. Chaudhury, K. Rithwik, Image denoising using optimally weighted bilateral filters: A sure and fast approach, in: Proceedings IEEE International Conference on image Procesing (ICIP), 2015, pp. 108–112.

[195] M. Aleksic, M. Smirnov, and S. Goma, Novel bilateral filter approach: Image noise reduction with sharpening," In: Proceeding of the SPIE Conference on Digital Photography II, 2006.

[196] C. Stein, Estimation of the mean of a multivariate normal distribution," the Annals of Statistics, 9 (6) (1981) 1135-1151.

[197] K. N. Chaudhury, D. Sage, and M. Unser, Fast O(1) bilateral filtering using trigonometric range kernels, IEEE Trans. Image Process., 20 (12) (2011) 3376-3382.

[198] K. N. Chaudhury, Acceleration of the shiftable O(1) algorithm for bilateral filtering and nonlocal means, IEEE Trans. Image Processing, 22 (4) (2013) 1291-1300.

[199] Y.-L.You, W. Xu, A. Tannenbaum, M. Kaveh, Behavioral analysis of anisotropic diffusion in image processing, IEEE Trans. Image Process., 5 (11) (1996) 1539–1553.

[200] G. Strang, Introduction to Applied Mathematics. Cambridge, MA: Wellesley-Cambridge, 1986.

[201] 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, 12 (12) (2003) 1579-1590.

[202] S. Kim, H. Lim, Fourth-order Partial Differential Equations for Effective Image Denoising, UAB Conference on Differential Equations and Computational Simulations, Electronic Journal of Differential Equations, vol.17, 2009, pp. 107–121.

[203] M. R. Hajiaboli, An Anisotropic Fourth-Order Partial Differential Equation for Noise Removal, Springer, International Conference on Scale Space and Variational Methods in Computer Vision, 2009, pp. 356-367.

[204] X. Liu, L. Huang, Z. Guo, Adaptive fourth-order partial differential equation filter for image denoising, Applied Mathematics Letters, 24 (8) (2011) 1282-1288.

[205] M. Kuwahara, K. Hachimura, S. Ehiu, and M. Kinoshita, Processing of ri-angiocardiographic images, Digital Processing of Biomedical Images. New York: Plenum, 1976, pp. 187–203.

[206] G. Papari, N. Petkov, P. Campisi, Artistic Edge and Corner Enhancing Smoothing, IEEE Trans. Image Process., 16 (10) (2007) 2449-2469.

[207] M. Wang, S. Zheng，X. Li, X. Qin, A new image denoising method based on Gaussian filter, in: Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering (ISEEE), 2014.

[208] M. Gupta, H. Taneja, L. Chand, Performance Enhancement and Analysis of Filters in Ultrasound Image Denoising, Procedia Computer Science, 132 (2018) pp. 643-652.

[209] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Processing, 16 (8) (2007) 2080–2095.

[210] M. Hasan, M. R. El-Sakka, Improved BM3D image denoising using SSIM-optimized Wiener filter, J. Image Video Process., 2018.

[211] R. Oktem, K. Egiazarian, I. Aizenberg, N. Aizenberg, Transform Domain Denoising Using Nonlinear Filtering and Cellular Neural Networks, IEEE International Conference on Image Processing (ICIP), 1998.

[212] I. N. Aizenberg, Processing of noisy and small detailed gray-scale images using Cellular Neural Networks, journal of Electronic Imaging, 6 (3) (1997) 272-285.

[213] N. N. Aizenberg, I. N. Aizenberg, CNN based on multi-valued neuron as a model of associative memory for grayscale images, in: Proceedings of the IEEE 2nd International Workshop on Cellular Neural Networks and their Applications, 1992, pp. 36-41.

[214] 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), 2012, pp. 341–349.

[215] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, The Journal of Machine Learning Research, 11 (2010) 3371–3408.

[216] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. , 12 (11) (2003) 1338–1351.

[217] M. Elad and M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process., 15 (12) (2006) 3736–3745.

[218] Y. Chen, T. Pock, Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration, IEEE Trans. Pattern Anal. Mach. Intell., 39 (6) (2017) 1256-1272.

[219] 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.

[220] V. Jain and S. Seung, Natural image denoising with convolutional networks, In Advances in Neural Information Processing Systems, 21 (2009) 769–776.

[221] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, Greedy layerwise training of deep networks, In Advances in Neural Information Processing Systems, 2007, pp. 153–160.

[222] V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 807–814.

[223] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization," Mathematical Programming, 45 (1-3) (1989) 503–528.

[224] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, Non-local sparse models for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2272–2279.

[225] D. Zoran and Y. Weiss, From learning models of natural image patches to whole image restoration, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 479–486.

[226] Y. Chen, T. Pock, R. Ranftl, and H. Bischof, Revisiting loss-specific training of filter-based MRFs for image restoration, in: Proceedings of the Conference on Pattern Recognition, 2013, pp. 271–281.

[227] J. Jancsary, S. Nowozin, C. Rother, Loss-specific training of nonparametric image restoration models: A new state of the art, Springer, in Proceedings of the European Conference on Computer Vision (ECCV), 2012, pp. 112–125.

[228] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2014.

[229] A. Barbu, Training an active random field for real-time image denoising, IEEE Trans. Image Process., 18 (11) (2009) 2451–2462.

[230] J. Domke. "Generic methods for optimization-based modeling, in Proceedings of the 15 International Conference on Artificial Intelligence and Statistics, 2012, pp. 318–326.

[231] 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., 26 (7) (2017) 3142-3155.

[232] 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), 2012, pp. 1097–1105.

[233] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning (ICML), 2015, pp. 448–456.

[234] H. C. Burger, C. J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with BM3D?, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2392–2399.

[235] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: Proceedings of the IEEE Conference on Vision and Pattern Recognition (*CVPR)*, 2017, pp. 3929–3938.

[236] D. Kingma, J. B. Adam, A method for stochastic optimization, in: Proceedings of the International Conference for Learning Representations (ICLR), 2015.

[237] T. Remez, O. Litany, R. Giryes, A. M. Bronstein, Class-Aware Fully Convolutional Gaussian and Poisson Denoising, IEEE Trans. Image Process., 27 (11) (2018) 5707-5722.

[238] M. Abadi et al., TensorFlow: Large-scale machine learning on heterogeneous distributed systems. [Online]. Available: https://arxiv.org/abs/1603.04467, 2015.

[239] K. Simonyan, A. Zisserman., Very deep convolutional networks for large-scale image recognition, [Online]. Available: https://arxiv.org/abs/1409.1556, 2014.

[240] K. Arakawa, Fuzzy rule-based signal processing and its application to image restoration, IEEE Journal on Select. Areas in Comm., 12 (9) (1994) 1495-1502.

[241] T. Rahman, M. R. Haque, L. J. Rozario, M. S. Uddin, Gaussian noise reduction in digital images using a modified fuzzy filter, in Proceedings of the 17th IEEE International Conference on Computer and Information Technology (ICCIT), 2014, pp. 217-222.

[242] A. Kethwas, B. Jharia, Image de-noising using Fuzzy and Wiener filter in Wavelet domain, IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015.

[243] M. Gonz´alez-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 (2016) 660-671.

[244] 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 (1998) 53–68.

[245] M. Nachtegael, 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, 52 (2000) 3–57.

[246] R. Lerallut, E. Decencire, F. Meyer, Image filtering using morphological amoebas, Image and Vision Computing, 25 (4) (2007) 395–404.

[247] J. Serra, Image Analysis and Mathematical Morphology, vols. 1,2, Academic Press, London (1982, 1988).

[248] P. Soille, "Morphological Image Analysis," 2ed ed., Berlin, Heidelberg: Springer, 1999.

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

[250] J. J. Kivinen, E. B. Sudderth†, M. I. Jordan, Image Denoising with Nonparametric Hidden markove Trees, IEEE International Conference on Image Processing (ICIP), 2007.

[251] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, Hierarchical Dirichlet processes, Journal of the American Statistical Association, 101 (476) (2006) 1566–1581.

[252] A. S. Willsky, Multiresolution Markov models for signal and image processing, in: Proceedings of the IEEE, 90 (8) (2002) 1396-1458.

[253] M. S. Crouse, R. D. Nowak, R. G. Baraniuk, Wavelet–based statistical signal processing using hidden Markov models, IEEE Trans. Sig. Proc., 46 (4) (1998) 886–902.

[254] 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.*, 54 (3) (2016) pp. 1818_1833.

[255] L. Sun, B. Jeon, Y. Zheng, Z. Wu, Hyperspectral image restoration using low-rank representation on spectral difference image, *IEEE Geosci. Remote Sens. Lett.*, 14 (7) (2017) pp. 1151_1155.

[256] E. J. Candès, Harmonic analysis of neural netwoks," Elsevier, Applied and Computational Harmonic Analysis, 6 (2) (1999) 197–218.

[257] E. J. Candès, Monoscale ridgelets for the representation of images with edges, Dept. Statist., Stanford Univ., Stanford, CA, Tech. Rep., 1999. for publication.

[258] E. J. Candès, D. L. Donoho, Curvelets, Curvelets—A surprisingly effective nonadaptive representation for objects with

edges, in Curve and Surface, C. Rabut, A. Cohen, L. L. Schumaker, Eds. Nashville, TN: Vanderbilt Univ. Press, 2000.

[259] J.-L. Starck, F. Murtagh, and A. Bijaoui, Image Processing and Data Analysis: The Multiscale Approach, Cambridge, U.K.: Cambridge Univ. Press, 1998.

[260] M. N. Do, M. Vetterli, Pyramidal directional filter banks and curvelets, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2001.

[261] P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans. Commun., 31 (4) (1983) 532–540.

[262] R. H. Bamberger and M. J. T. Smith, A filter bank for the directional decomposition of images: Theory and design, IEEE Trans. Signal Proc., 40 (4) (1992) 882–893.

[263] M. N. Do and M. Vetterli, Framing pyramids, IEEE Trans. Signal Proc., 51 (9) (2003) 2329–2342.

[264] M. N. Do, Directional multiresolution image representations, Ph.D. dissertation, Swiss Federal Institute of Technology, Lausanne, Switzerland, Dec. 2001.

[265] M. Vetterli and J. Kova˘cevi´c, Wavelets and Subband Coding, Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

[266] S. Mallat, A Wavelet Tour of Signal Processing, 3rd ed., Academic Press: Cambridge, MA, 1999.

[267] E. J. Candès, D. L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities, Communication on Pure and Applied Mathematics, vol. LVII (2004) 219–266.

[268] L. Demanet, Second-generation curvelets, M.S. thesis, Universit´e Catholique de Louvain, Belgium, 2002.

[269] D. L. Donoho, M. Vetterli, R. A. DeVore, I. Daubechies, Data compression and harmonic analysis, IEEE Trans. Inform. Th., 44 (6) (1998) 2435–2476.

[270] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[271] T. Kohonen, Self-Organizing Maps, ser. Information Sciences, 2nd ed. Heidelberg, Germany: Springer, vol. 30, 1997.

[272] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, Chichester, U.K.: Wiley, 1999.

[273] A. Foi, V. Katkovnik, K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images, IEEE Trans. Image Process., 16 (5) 2007.

[274] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, Image denoising with block-matching and 3D filtering, Presented at the SPIE Electronic Imaging: Algorithms and Systems V, 2006.

[275] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, Alessandro Foi, Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction, IEEE Trans. Image Process., 22 (1) (2013) 119-133.

[276] P. Coupé, P. Hellier, S. Prima, C. Kervrann, and C. Barillot, 3D wavelet subbands mixing for image denoising, Hindawi, International Journal of Biomedical Imaging, 2008 (2008) 1–11.

[277] J. V. Manjón, P. Coupé, A. Buades, D. L. Collins, and M. Robles, New methods for MRI denoising based on sparseness and self-similarity, Medical Image Analysis, 16 (1) (2012) 18–27.

[278] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images, IEEE Trans. Med. Imag., 27 (4) (2008) 425–441.

[279] A. Hyvarinen，E. Oja, A fast fixed-point algorithm for independent component analysis, IEEE Neural Computation，9(7) (1997) 1483-1492.

[280] A.Hyvarinen, A family of fixed-point algorithms for independent component analysis, in: Proceedings of the international Conference on Acoustics， Speech and Signal Processing，1997, pp. 3917-3920.

[281] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks, 10 (3) (1999) 626-634.

[282] Z. Shi，H. Tang, Y. Tang, A new fixed-point algorithm for independent component analysis, Neural Computing, 56 (2004) 467-473.

[283] R. Prasad, H. Saruwatari, K. Shikano, Blind Separation of Speech by Fixed-Point ICA with Source Adaptive Negentropy Approximation, IEICE Trans. Fundamentals, vol. E88-A (7) (2005) 1683~1692.

[284] S. K. Kopparapu, M. Satish, Identifying Optimal Gaussian Filter for Gaussian Noise Removal, IEEE 3th International Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, 2011.

[285] G. Santhanamari , J. S. Varuna Viveka, B. Purushothaman, U. Shanthini, M. Vanitha5, A New Image Denoising Algorithm Based on Adaptive threshold and Fourth Order Partial Diffusion Equation, IEEE International Conference on Computational Intelligence & Computing Research (ICCIC), 2012.

[286] J. Zhong, H. Sun, Wavelet-Based Multiscale Anisotropic Diffusion With Adaptive Statistical Analysis for Image Restoration, IEEE Trans. Circuits Syst.—I, 55 (9) (2008) 2716-2725.

[287] S. Mallat and S. Zhong, Characterization of signals from multiscale edges, IEEE Trans. Pattern Anal. Mach. Intell., 14 (7) (1992) 710–732.

[288] S. Mallat, W. L. Hwang, Singularity detection and processing with wavelets, IEEE Trans. Inf. Theory, 38 (2) (1992) 617–643.

[289] J. Zhong, R. Ning, D. Conover, Image denoising based on multiscale singularity detection for cone beam CT breast imaging, IEEE Trans. Med. Imaging, 23 (6) (2004) 696–703.

[290] L. Sendur and Ivan. W. Selesnick, Bivariate shrinkage with local variance estimation, IEEE Signal Process. Lett., 9 (12) (2002) 438–441.

[291] Z. Cai, T. H. Cheng, C. Lu, K. R. Subramanian, Efficient wavelet based image denoising algorithm, IEEE Electron. Lett., 37 (11) (2001) 683–685.

[292] W. Zhiming, B Hong, Z. Li, Image Denoising by Anisotropic Diffusion in Wavelet domain, IEEE 3th International Conference on Measuring Technology and Mechatronics Automation, 2011.

[293] X. Zhang, W. Chen, Wavelet Domain Diffusion for DWI Images, IEEE 2nd International Conference on Bioinformatics and Biomedical Engineering, 2008, pp. 2149-2152.

[294] A. Pizurica, W. Philips, Estimating the Probability of the Presence of a Signal of Interest in Multiresolution Single and Multiband Image Denoising, IEEE Trans. Image Processing, 15 (3) (2006) 654~665.

[295] B. K. Shreyamsha Kumar, Image denoising based on gaussian/bilateral filter and its method noise thresholding, Signal, Image and Video Processing, 7 (6) (2013) 1159-1172.

[296] M. Zhang, B. K. Gunturk, Multiresolution bilateral filtering for image denoising, IEEE Trans. Image Process., 17 (12) (2008) 2324–2333.

[297] Laparra,V.,Gutierrez, J., Camps-Valls,G., Malo, J., Image denoising with kernels based on natural image relations, Journal of Machin Learning Research, 11 (2010) 873–903.

[298] L. Lin, K. Lingfu, Image Denoising Base on Non-local Means with Wiener Filtering in Wavelet Domain, IEEE 5th International

Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009.
[299] Y.-L. Liu, J. Wang, C. Xi, A Robust and Fast Non-Local Means Algorithm for Image Denoising, Journal of Computer Science and Technology, 23 (2) (2008) 270-279.
[300] M. K. Mihcak, I. Kozintsev, K. Ramchandran, P. Moulin, Low complexity image denoising based on statistical modeling of wavelet coefficients, IEEE Signal Processing Lett., 6 (12) 300–303.
[301] S. J. You, N. I. Cho, An adaptive bandwidth nonlocal means image denoising in wavelet domain, EURASIP Journal on Image and Video Processing (2013).
[302] N. Bhoi, S. Meher, Total Variation based Wavelet Domain Filter for Image Denoising, IEEE 1th International Conference on Emerging Trends in Engineering and Technology, 2008.
[303] D. L. Donoho and I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, Journal of American Statistical Association, 90 (432) (1995) 1200-1224.
[304] M. Mafi, S. Tabarestani, M. Cabrerizo, A. Barreto, M. Adjouadi, Denoising of Ultrasound Images Affected by Combined Speckle and Gaussian Noise, IET Image Process., doi: 10.1049/iet-ipr.2018.5292, 2018.
[305] Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury, The Dual-Tree Complex Wavelet Transform, IEEE Signal Proc. Magazin, 2005.
[306] M. K. Mihcak, I. Kozintsev, and I. Ramchandran, Spatially adaptive statistical modeling of wavelet image coefficientas and its application to denoising, in: Proceedings IEEE International Conference Acoustics, Speech, and Signal Processing, 1999.
[307] S. Bacchelli, S. Papi, Image denoising using principal component analysis in the wavelet domain, Journal of Computational and Applied Mathematics, 189 (1-2) (2006) 606–621.
[308] M. Barnabei, L. B. Montefusco, Recursive properties of Toeplitz and Hurwitz matrices, Linear Algebra and its Application, 274 (1-3) (1998) 367–388.
[309] S. Bacchelli, S. Papi, Filtered wavelet thresholding methods, J. Comput. Appl. Math., 164-165 (2004) 39–52.
[310] R. D. da Silva, R. Minetto, W. R. Schwartz , H. Pedrini, Adaptive edge-preserving image denoising using wavelet transforms, Pattern Analysis and Applications, 16 (4) (2013) 567-580.
[311] H.Y. Gao, A. G. Bruce, WaveShrink with firm shrinkage, Statistica Sinica, 7 (1997) 855–874.
[312] H.Y. Gao, Wavelet shrinkage denoising using the nonnegative garrote, Journal of Computational and Graphical Statistics, 7 (4) (1998) 469–488.
[313] A. Antoniadis, J. Fan, Regularization of wavelet approximations, Journal of American Statistical Association, 96 (455) (2001) 939–967.
[314] Guoquan Xing, Yuxia Zhang, YuYan Ruolin Ruan, An Exponential Threshold Function Wavelet Denoising Method Based on Gaussian Distribution, IEEE 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 2016.
[315] F. Luisier, T. Blu, M. Unser, A New SURE Approach to Image Denoising: Interscale Orthonormal Wavelet Thresholding, IEEE Trans. Image Process., 16 (3) (2007) 593-606.
[316] Thierry Blu, Florian Luisier, The SURE-LET Approach to Image Denoising, IEEE Trans. Image Process., 16 (11) (2007) 2778 - 2786.
[317] F. Luisier, T. Blu, SURE-LET Multichannel Image Denoising: Interscale Orthonormal Wavelet Thresholding, IEEE Trans. Image Process, 17 (4) (2008) 482-492.
[318] T. Qiu, A. Wang, N. Yu, Aimin Song, LLSURE: Local Linear SURE-Based Edge-Preserving Image Filtering, IEEE Trans. Image Process., 22 (1) (2013) 80-90.
[319] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell., 35 (6) (2013) 1397-1409.
[320] M. Jansen, M. Malfait, A. Bultheel, Generalized cross validation for wavelet thresholding, Signal Processing, 56 (1) (1997) 33–44.
[321] L. Zhang, J. Chen, T. Zhu, Image denoising based on iterative generalized cross-validation and fast translation invariant, Journal of Visual Communication and Image Representation, 28 (2015) 1–14.
[322] 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, 1997, pp. 374–377.
[323] S. G. Chang, B. Yu, M. Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising, IEEE Trans. Image Processing, 9 (2000) 1522–1531.
[324] R. R. Coifman, D. L. Donoho, Translation invariant denoising, Wavelets and Statistics, vol. 103, Lecture Notes in Statistics (1995) pp. 125–150.
[325] L. Sendur, I. W. Selesnick, A Bivariate Shrinkage Function for Wavelet-based Denoisng, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002.
[326] L. Sendur, I. W. Selesnick, Bivariate Shrinkage Functions for Wavelet-Based Denoising Exploiting Interscale Dependency, IEEE Trans. Signal Processing, 50 (11) (2002) 2744 - 2756.
[327] D. Min, Z. Jiuwen, M. Yide, Image denoising via bivariate shrinkage function based on a new structure of dual contourlet transform, Signal Processing, 109 (2015) pp. 25–37.
[328] J. K. Romberg, H. Choi, and R. G. Baraniuk, Bayesian tree-structured image modeling using wavelet-domain hidden Markov models, IEEE Trans. Image Process., 10 (7) (2001) 1056-1068.
[329] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman., *The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.* [Online]. Available: http://www.pascalnetworks.org/challenges/VOC/voc2010/workshop/index.html, 2010