IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 23, 2019, accepted November 13, 2019, date of publication November 18, 2019,
date of current version December 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953924

An Iterative Mean Filter for Image Denoising

UĞUR ERKAN[1], DANG NGOC HOANG THANH[2], LE MINH HIEU[3], AND SERDAR ENGİNOĞLU[4]

[1]Department of Computer Engineering, Faculty of Engineering, Karamanoğlu Mehmetbey University, 70100 Karaman, Turkey
[2]Department of Information Technology, Hue College of Industry, Hue 530000, Vietnam
[3]Department of Economics, University of Economics - The University of Da Nang, Da Nang 550000, Vietnam
[4]Department of Mathematics, Faculty of Arts and Sciences, Çanakkale Onsekiz Mart University, 17100 Çanakkale, Turkey

Corresponding author: Dang Ngoc Hoang Thanh (dnhthanh@hueic.edu.vn)

ABSTRACT We propose an Iterative Mean Filter (IMF) to eliminate the salt-and-pepper noise. IMF uses the mean of gray values of noise-free pixels in a fixed-size window. Unlike other nonlinear filters, IMF does not enlarge the window size. A large size reduces the accuracy of noise removal. Therefore, IMF only uses a window with a size of 3×3. This feature is helpful for IMF to be able to more precisely evaluate a new gray value for the center pixel. To process high-density noise effectively, we propose an iterative procedure for IMF. In the experiments, we operationalize Peak Signal-to-Noise Ratio (PSNR), Visual Information Fidelity, Image Enhancement Factor, Structural Similarity (SSIM), and Multiscale Structure Similarity to assess image quality. Furthermore, we compare denoising results of IMF with ones of the other state-of-the-art methods. A comprehensive comparison of execution time is also provided. The qualitative results by PSNR and SSIM showed that IMF outperforms the other methods such as Based-on Pixel Density Filter (BPDF), Decision-Based Algorithm (DBA), Modified Decision-Based Untrimmed Median Filter (MDBUTMF), Noise Adaptive Fuzzy Switching Median Filter (NAFSMF), Adaptive Weighted Mean Filter (AWMF), Different Applied Median Filter (DAMF), Adaptive Type-2 Fuzzy Filter (FDS): for the IMAGESTEST dataset – BPDF (25.36/0.756), DBA (28.72/0.8426), MDBUTMF (25.93/0.8426), NAFSMF (29.32/0.8735), AWMF (32.25/0.9177), DAMF (31.65/0.9154), FDS (27.98/0.8338), and IMF (33.67/0.9252); and for the BSDS dataset – BPDF (24.95/0.7469), DBA (26.84/0.8061), MDBUTMF (26.25/0.7732), NAFSMF (27.26/0.8191), AWMF (28.89/0.8672), DAMF (29.11/0.8667), FDS (26.85/0.8095), and IMF (30.04/0.8753).

INDEX TERMS Salt-and-pepper noise, image denoising, noise removal, image restoration, image processing, nonlinear filter.

I. INTRODUCTION

Image noise usually occurs during signal acquisition and transmission. Image denoising is a procedure of removing noise from an image. The primary purpose of image denoising is to preserve image structures such as details, edges, and textures. The images acquired after denoising are used for post-processing tasks such as image segmentation, feature extraction, image analysis, image classification, and pattern recognition. Noise removal with image structure preservation is vital for improving the accuracy and performance of other post-processing tasks [1]–[9].

There are several types of noise formulated on images [10]–[12]. Being one of the common types of noise, impulse noise has two types: salt-and-pepper noise (SPN) [13]–[15] and random-valued impulse noise (RVIN) [16]. For SPN, pixels disturbed by noise hold a maximum or a minimum gray value. This issue causes a severe decrement of image quality [17]–[22].

One of the well-known methods for removing SPN is Median Filter (MF) [23]. MF uses a fixed-size window, and the median value of the pixels in the window is assigned to the center pixel. MF is relatively successful for low noise densities, but when we employ MF to remove medium-density and high-density SPN, MF works ineffectively [16]–[17]. To overcome the drawback, Adaptive Median Filter (AMF) has been proposed [24]. The goal of AMF is to use a dynamic adaptive window. The window size is enlarged gradually until the adaptive conditions are fulfilled. For the high-density noise, the window size needs to be large enough. This matter reduces the accuracy and processing

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

speed of the filter. Unlike AMF, Adaptive Weighted Mean Filter (AWMF) [25] uses a weighted mean to evaluate a new gray value for the center pixel. AWMF focuses on decreasing the number of errors occurring in the detection of noisy pixels of AMF. Hence, AWMF works more effectively than AMF. However, AWMF still has the drawback of AMF [26].

Recently, two-stage filters have been designed to eliminate the disadvantages of MF, AMF, and AWMF. In the first stage, noisy pixels are detected, and in the second stage, only the gray values of the detected noisy-pixels are replaced by new gray values [27]–[29]. Based-on Pixel Density Filter (BPDF) consists of two stages. BPDF sets a new gray value by considering the repetition of pixels in an adaptive window [30]. This method proves highly effective in the presence of low-density SPN, but its effectiveness tends to reduce for medium-density and high-density noise. Decision-Based Algorithm (DBA) is another well-operating method for low-density noise. DBA is based on a decision-making method exploiting pixels in the window [31]. The disadvantage of DBA is the fact that it uses a fixed-size window. Modified Decision-Based Unsymmetric Trimmed Median Filter (MDBUTMF) is listed among the well-known and effective filters for removing SPN [32]. The major disadvantage of this method is that when there are no noise-free pixels in the window, it will use the mean of gray values of all pixels in a window to assign a new gray value to the center pixel. Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) uses a dynamic adaptive window [33]. For NAFSMF, a new gray value is determined by a fuzzy decision-making method. NAFSMF works effectively for high-density SPN removal as well. Different Applied Median Filter (DAMF) operates competently for all SPN densities [34]. Another common filter is Adaptive Type-2 Fuzzy Filter (FDS, the fuzzy denoising for SPN) [35]. FDS is developed based on the theory of fuzzy decision. Adaptive Iterative Fuzzy Filter (AIFF) is an effective filter that is based on a fuzzy detector and a weighted mean filter [36]. Finally, Probabilistic Decision Based Filter (PDBF) – another proficient filter – relies on the patch else trimmed median [37].

As we mentioned above, modern denoising filters use a dynamic adaptive window to promote denoising success in the presence of high-density noise. This practice reduces the accuracy and processing speed. Our goal is only to use a window with a fixed size of 3×3 and the mean of grey values of noise-free pixels. This method will find out a new gray value closer to the original gray value of the center pixel. However, the denoising success for high noise densities will be reduced. To overcome this drawback, we integrate the filter by an iterative procedure.

The outline of the work is organized as follows: **Section II** presents some basic mathematical notions, definitions, and the algorithm of Iterative Mean Filter (IMF). In **Section III,** we perform the image denoising experiments, compare the obtained results with other state-of-the-art SPN denoising methods, and discuss the efficacy of the compared methods. **Section IV** offers a conclusion.

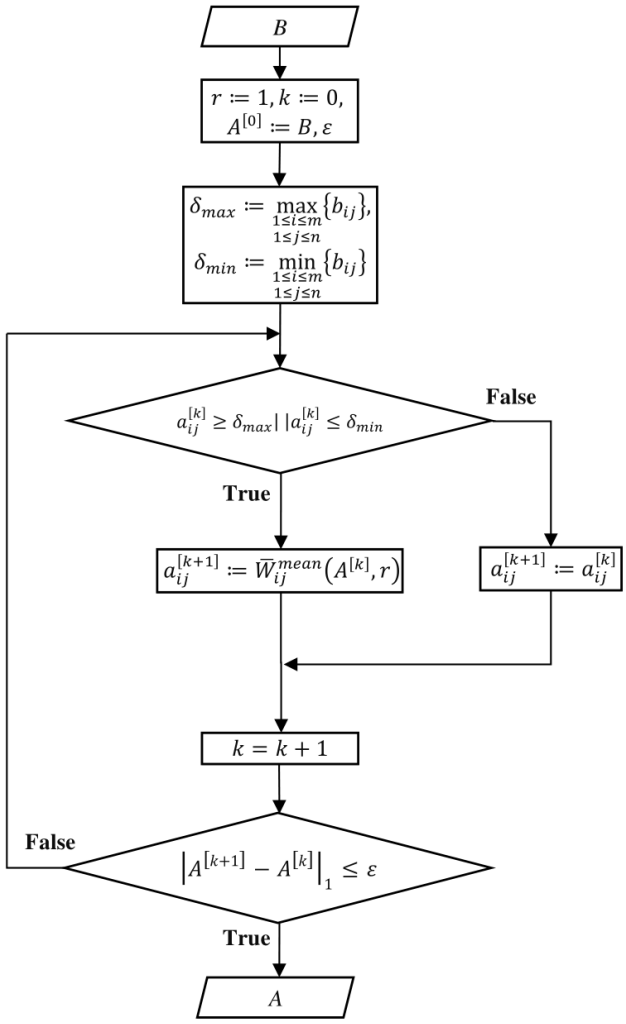

FIGURE 1. The flowchart of IMF.

II. ITERATIVE MEAN FILTER

A. DEFINITIONS AND NOTIONS

Through the article, let $U := \left[u_{ij}\right]_{m\times n}$ be a ground truth (noise-free) image such that $\delta_{min} \le u_{ij} \le \delta_{max}$, for all pixel locations $(i, j) \in I = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$. Here, m is the numbers of pixels in a row of U, n is the numbers of pixels in a column of U, and $[\delta_{min}, \delta_{max}]$ is the range of gray values of U. For example, $\delta_{min} = 0$ and $\delta_{max} = 255$, for an 8-bit grayscale image. We note that, for natural images, the number of pixels achieving the boundary values $\delta_{min}, \delta_{max}$ are very small.

Definition 1: Let $U := \left[u_{ij}\right]_{m\times n}$ be a ground truth image. If

$$b_{ij} := \begin{cases} \delta_{min}, & \textit{with probability } p \\ \delta_{max}, & \textit{with propability } q \\ u_{ij}, & \textit{with probability } 1-(p+q) \end{cases} \tag{1}$$

then $B := \left[b_{ij}\right]_{m\times n}$ is called a corrupted (noisy) image by SPN of U with $p + q$ noise level (or noise density or noise ratio), where p, q, and $p + q \in [0, 1]$. Here, if the value of the noise level is even, then $p = q$.

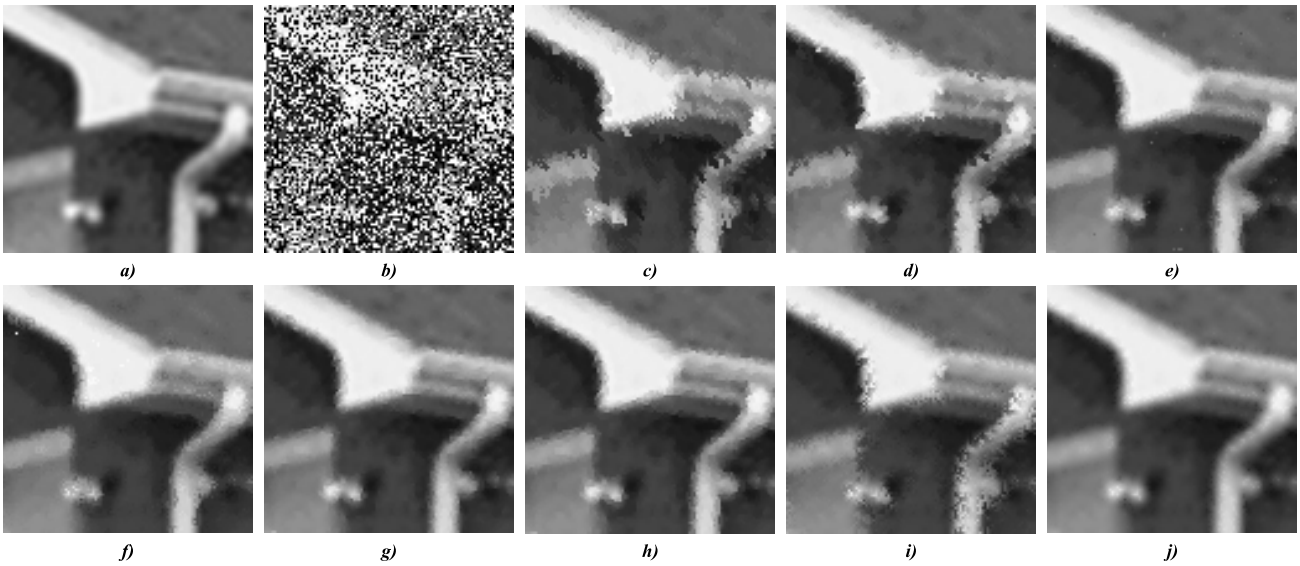

FIGURE 2. Denoising results for the House image of for a part of 140 × 140 pixel with a SPN ratio of 60%. PSNR, SSIM, VIF, IEF and MSSIM values of the results of the method: (b) Noisy image (7.40 dB, 0.0137, 0.0462, 1, 0.9103), (c) BPDF (27.41 dB, 0.8759, 0.3544, 102.29, 0.9870), (d) DBA (30.38 dB, 0.9228, 0.4548, 203.89, 0.9810), (e) MDBUTMF (35.02 dB, 0.9452, 0.5831, 477.55, 0.9888), (f) NAFSMF (33.98 dB, 0.9387, 0.5056, 480.53, 0.9894), (g) AWMF (37.80 dB, 0.9754, 0.7201, 1100.84, 0.9899), (h) DAMF (36.51 dB, 0.9704, 0.7005, 810.94, 0.9898), (i) FDS (29.77 dB, 0.9171, 0.4393, 167.74, 0.9780), and (j) IMF (39.77 dB, 0.9805, 0.7443, 1843.05, 0.9910).

Definition 2: Let $r \geq 1$ be an integer number and $A = [a_{ij}]_{m\times n}$ be an image. Then, in image A, the indices set of a window with a size of $(2r+1)\times(2r+1)$ centered at a pixel location (i,j), denoted by $W_{ij}(A,r)$, is defined as follows:

$$\left\{(i^*,j^*) \in I : |i^*-i| \leq r, |j^*-j| \leq r\right\} \tag{2}$$

Definition 3: Let $r \geq 1$ be an integer number and $A = [a_{ij}]_{m\times n}$ be an image. Then, in image A, the strict indices set of a window with a size of $(2r+1)\times(2r+1)$ centered at a pixel location (i,j), denoted by $W^*_{ij}(A,r)$, is defined as follows:

$$\left\{(i^*,j^*) \in W_{ij}(A,r) : a_{i^*j^*} \neq \delta_{min},\ a_{i^*j^*} \neq \delta_{max}\right\} \tag{3}$$

Definition 4: Let $A = [a_{ij}]_{m\times n}$ be an image. Then, constrained mean of $W_{ij}(A,r)$, denoted by $\bar{W}^{mean}_{ij}(A,r)$, is defined by

$$\begin{cases} a_{ij} & W^*_{ij}(A,r) = \varnothing \\ \dfrac{1}{card(W^*_{ij}(A,r))} \displaystyle\sum_{(i^*,j^*)\in W^*_{ij}(A,r)} & W^*_{ij}(A,r) \neq \varnothing \end{cases} \tag{4}$$

where *card* (.) is the set cardinality, i.e. the number of pixels.

Definition 5: Let $A := [a_{ij}]_{m\times n}$ and $A^* := \left[a^*_{ij}\right]_{m\times n}$ be two images. Then, l_1-distance (or Manhattan distance [38]) between A and A^* is defined as follows:

$$\left|A - A^*\right|_1 := \sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij} - a^*_{ij}\right|, \tag{5}$$

where $|.|$ denotes absolute value.

B. ITERATIVE MEAN FILTER ALGORITHM

The goal of IMF is based on MF. MF works effectively for low noise densities. The processing speed of MF is very high because it uses a fixed-size window instead of an adaptive window as in AMF, AWMF, MDBUTMF, and NAFSMF.

For the study, we focus on two characteristics: (1) MF is only effective in low-density noise, and (2) the weighted mean of AWMF is better than the median of AMF. The first characteristic can be explained as follows: because MF uses a small fixed-size window, there are no noise-free pixels in the window in the presence of high noise density. Hence, MF avails of gray values of all noisy pixels to restore a gray value for the center pixel. If all noisy pixels in the window are only salt pixels or only pepper pixels, the center pixel will remain to be noisy as well. The second characteristic is the improved accuracy of AWMF – relying on weighted mean – in comparison with AMF – operationalizing median – even though both adopt the same technique for noise detection. Inferring from a variety of comparisons in [25], AWMF outperforms AMF, particularly for high noise densities.

To propose IMF, first of all, we utilize the advantage of MF by using a window with a size of 3 × 3. Next, we incorporate this advantage with the constrained mean of a window instead of using the median. This way will give a higher accuracy to evaluate a new gray value for the center pixel. However, like MF, because we only consider a fixed-size window, IMF will work ineffectively at high noise densities. Hence, we propose to combine the method with an iterative procedure. The iterative procedure will guarantee that all noisy pixels will be processed. The stop condition of the iterative procedure bases on the ℓ_1-distance.

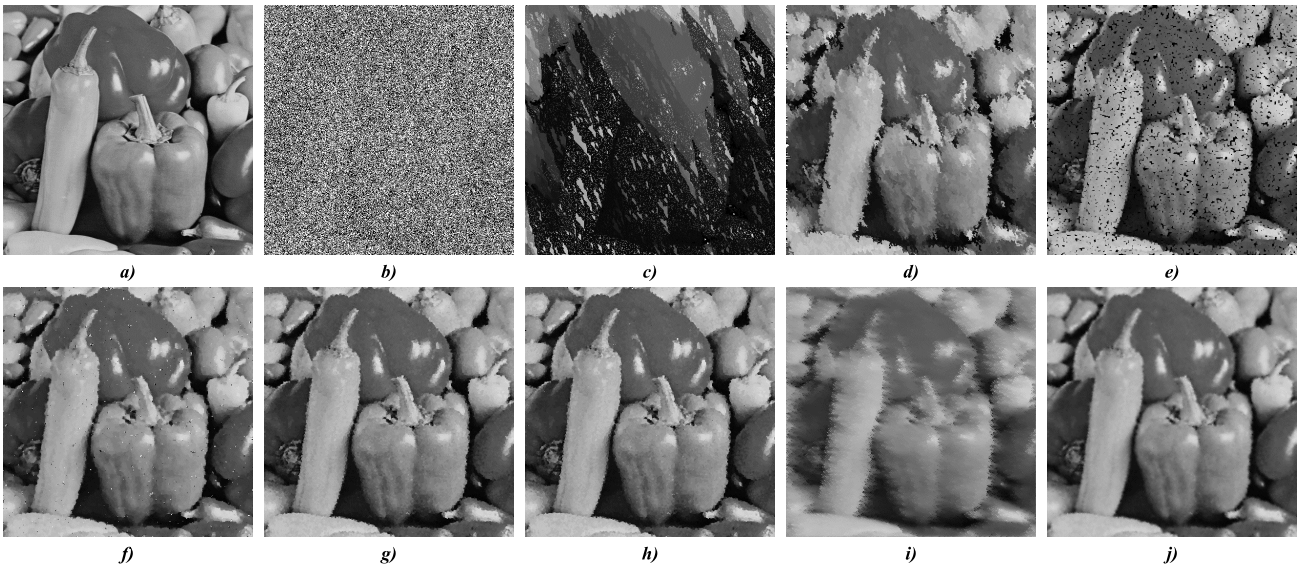

FIGURE 3. Denoising results for the Peppers image with the size of 512 × 512 pixels with a SPN ratio of 90%. PSNR, SSIM, VIF, IEF and MSSIM values of the results of the method: (b) Noisy image (7.40 dB, 0.0137, 0.0196, 1, 0.8078), (c) BPDF (9.21 dB, 0.1927, 0.0307, 2.14, 0.1836), (d) DBA (18.72 dB, 0.5272, 0.0699, 20.40, 0.9455), (e) MDBUTMF (16.91 dB, 0.4029, 0.0814, 12.88, 0.9568), (f) NAFSMF (23.60 dB, 0.6499, 0.1430, 60.34, 0.9848), (g) AWMF (26.23 dB, 0.7123, 0.2013, 109.22, 0.9925), (h) DAMF (25.87 dB, 0.7049, 0.1999, 102.62, 0.9921), (i) FDS (18.15 dB, 0.5085, 0.0618, 16.97, 0.9251), and (j) IMF (27.88 dB, 0.7700, 0.2289, 151.30, 0.9938).

Algorithm 1 Iterative Mean Filter (IMF)

Input: A noisy image $B := \left[b_{ij}\right]_{m\times n}$
Output: A restored image $A := \left[a_{ij}\right]_{m\times n}$

Initialize $r := 1$, $k := 0$, $A^{[0]} := B$, ε.
Compute : $\delta_{max} := \max\limits_{\substack{1\le i\le m\\ 1\le j\le n}} \left\{b_{ij}\right\}$, $\delta_{min} := \min\limits_{\substack{1\le i\le m\\ 1\le j\le n}} \left\{b_{ij}\right\}$.
Repeat
  For each pixel (i, j) of an image $A^{[k]}$ at a step k
    If $\left(a_{ij}^{[k]} \ge \delta_{max} \,||\, a_{ij}^{[k]} \le \delta_{min}\right)$
      Define : $W_{ij}^{*}\left(A^{[k]}, r\right)$.
      Update : $a_{ij}^{[k+1]} := \bar{W}_{ij}^{mean}\left(A^{[k]}, r\right)$
    Else
      Assign : $a_{ij}^{[k+1]} := a_{ij}^{[k]}$
    End
  End
Until $\left|A^{[k+1]} - A^{[k]}\right|_1 \le \varepsilon$

Details of IMF are presented in Algorithm 1. In every iteration step, windows with a size of 3 × 3 whose center pixel is noisy will be considered. The constrained mean of a window will be the new gray value of the center pixel. In the case of all pixels in the current window are noisy, the gray value of the center pixel of the window will not be changed. The center pixel will be processed in the next iteration steps. This is different from the way processed by an adaptive window of AWMF or AMF. Because enlarging the size of an adaptive window until there is at least a noise-free pixel will cause an issue in which gray values of very far away from the center pixel are also used for evaluating the gray value of the center pixel. For IMF, far pixels have lower weights. For the way of enlarging a window, all pixels have the same weight. In other words, the influence of far pixels on the new gray value of the center pixel in the case of IMF is smaller. Therefore, IMF can lead to devoid artifacts and sharpen edges as that it will be shown in the experimental section. The iterative procedure will be stopped if there is no change in the Manhattan distance between the images of two consecutive iteration steps. The corresponding flowchart is also presented in Figure 1.

Similar to nonlinear filters such as MF, AMF, and AWMF, IMF also uses the same condition based on the boundary values to detect noise: $a_{ij} \ge \delta_{max}$ or $a_{ij} \le \delta_{min}$. This can cause an issue for the synthetic images, in that all pixels owning boundary values $\delta_{max}, \delta_{min}$ will be treated as noisy pixels. This is not only the limit of IMF, but that is also the drawback of many other nonlinear filters for SPN. However, in the article, we only focus on natural images that are very important in practical applications. For natural images, the number of noise-free pixels acquiring the boundary values is very small. In the experimental section, we will show that IMF still works effectively for this case.

We also must notice that, because we consider gray values of pixels in the form of integer numbers, tolerance can be set to zero, $\varepsilon = 0$. If we consider gray values of pixels in the form of real numbers, tolerance can be set to a small enough number, for example, $\varepsilon = 10^{-6}$.

III. EXPERIMENTAL RESULTS

A. IMAGE QUALITY ASSESSMENT METRICS

In order to assess image quality after denoising, we use common error metrics such as Peak Signal-to-Noise Ratio

FIGURE 4. Denoising results by IMF for the Lena image with the size of 512 × 512 pixels with different SPN ratios. a) (15.43 dB, 0.1758, 0.1362, 1, 0.9910), b) (12.42 dB, 0.0848, 0.0887, 1, 0.9789), c) (10.68 dB, 0.0533, 0.0673, 1, 0.9646), d) (9.42 dB, 0.0364, 0.0528, 1, 0.9494), e) (43.48 dB, 0.9913, 0.8974, 518.49, 1), f) (40.18 dB, 0.9796, 0.8106, 457.49, 0.9999), g) (37.05 dB, 0.9675, 0.7343, 425.55, 0.0.9999), h) (35.40 dB, 0.9541, 0.6653, 398.35, 0.9998), i) (8.44 dB, 0.0263, 0.0422, 1, 0.9333), j) (7.65 dB, 0.0182, 0.0325, 1, 0.9141), k) (6.99 dB, 0.0139, 0.0278, 1, 0.8979), l) (6.42 dB, 0.0092, 0.0213, 1, 0.8803), m) (33.98 dB, 0.9383, 0.5917, 357.47, 0.9997), n) (32.49 dB, 0.9183, 0.5135, 306.95, 0.9994), o) (31.23 dB, 0.8953, 0.4353, 266.22, 0.9990), p) (29.70 dB, 0.8623, 0.3445, 210.75, 0.9982).

(PSNR) [39], Visual Information Fidelity (VIF) [40], Image Enhancement Factor (IEF) [41], Structural Similarity (SSIM) [39], and Multiscale SSIM (MSSIM) [42].

PSNR is defined as [39]:

$$PSNR(U, V) := 10 \log_{10} \left(\frac{255^2}{MSE(U, V)} \right) \quad (6)$$

TABLE 1. PSNR results of the methods for three traditional images with different SPN ratios.

Image	Filters	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lena	BPDF	39.89	35.67	32.99	30.62	28.29	25.89	22.90	17.85	10.98
	DBA	41.64	37.45	34.65	32.49	30.15	28.09	25.70	23.04	19.78
	MDBUTMF	40.48	35.41	31.56	30.49	30.94	30.70	29.28	26.05	16.60
	NAFSMF	38.83	35.65	33.81	32.30	30.99	29.90	28.57	27.13	23.47
	AWMF	39.02	37.50	36.16	34.88	33.57	32.06	30.56	28.81	26.20
	DAMF	42.97	39.29	36.84	34.94	33.21	31.64	30.22	28.53	25.93
	FDS	41.40	37.25	34.49	31.67	28.99	26.54	23.95	21.39	18.30
	IMF	**43.48**	**40.18**	**37.05**	**35.40**	**33.98**	**32.49**	**31.23**	**29.70**	**27.42**
Peppers	BPDF	38.44	34.89	32.58	30.49	28.50	26.10	23.18	18.07	9.21
	DBA	39.71	36.24	33.82	31.81	29.76	27.60	25.26	22.61	18.72
	MDBUTMF	40.64	35.23	31.15	30.14	30.99	30.93	29.41	25.97	16.91
	NAFSMF	39.60	36.33	34.36	32.85	31.68	30.55	28.97	27.37	23.60
	AWMF	37.74	36.86	35.58	34.40	33.10	31.82	30.21	28.57	26.23
	DAMF	41.52	37.89	35.67	33.95	32.55	31.31	29.79	28.28	25.87
	FDS	40.65	36.90	34.32	31.72	29.32	26.83	24.11	21.37	18.15
	IMF	**41.83**	**38.59**	**36.65**	**35.14**	**33.90**	**32.69**	**31.43**	**30.01**	**27.88**
House	BPDF	43.82	39.54	36.38	33.54	30.74	27.41	23.90	19.61	13.12
	DBA	47.60	42.43	38.12	35.55	32.80	30.38	27.22	24.07	19.99
	MDBUTMF	47.44	38.03	32.20	31.51	33.62	35.02	33.03	27.05	15.64
	NAFSMF	44.44	41.37	39.05	37.37	35.82	33.98	32.18	29.93	24.56
	AWMF	44.85	44.16	43.06	41.49	39.85	37.80	35.50	33.11	29.55
	DAMF	51.06	46.03	42.62	40.07	38.26	36.51	34.62	32.47	28.76
	FDS	47.43	42.85	39.39	35.78	32.83	29.77	26.47	23.90	20.59
	IMF	**51.89**	**48.39**	**45.84**	**43.73**	**42.06**	**39.77**	**37.76**	**34.99**	**31.60**

where MSE stands for Mean Square Error defined as:

$$MSE\,(U,V) := \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(u_{ij}-v_{ij}\right)^2 \tag{7}$$

$U := \left[u_{ij}\right]$ is a ground truth, $V := \left[v_{ij}\right]$ is an evaluated image. V can be a restored image A or a noisy image B.

VIF is defined as [40]:

$$VIF\,(U,V) := \frac{\sum_{j\in subbands} I\left(\vec{C}^{N,j};\vec{F}^{N,j}|s^{N,j}\right)}{\sum_{j\in subbands} I\left(\vec{C}^{N,j};\vec{E}^{N,j}|s^{N,j}\right)} \tag{8}$$

where

$$I\left(\vec{C}^{N,j};\vec{E}^{N,j}|s^{N,j}\right) := \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{M}\log_2\left(1+\frac{s_i^2\lambda_k}{\sigma_n^2}\right) \tag{9}$$

$$I\left(\vec{C}^{N,j};\vec{F}^{N,j}|s^{N,j}\right) := \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{M}\log_2\left(1+\frac{g_i^2 s_i^2\lambda_k}{\sigma_u^2+\sigma_n^2}\right) \tag{10}$$

where $I(\vec{C}^{N,j};\vec{E}^{N,j}|s^{N,j})$ and $I(\vec{C}^{N,j};\vec{F}^{N,j}|s^{N,j})$ represent the information that can ideally be extracted by the brain from a particular subband in the reference image (ground truth) U and the evaluated image V, respectively; $\vec{E}^{N,j}$ is a vector of N components of the visual signal for the j-subband at the output of the Human Visual System (HVS) of a ground truth U; $\vec{F}^{N,j}$ is a vector of N components of the visual signal for the j-subband at the output of HVS of an evaluated image V; $\vec{C}^{N,j}$ is a vector of N components of the random field from the j-subband in a ground truth U; σ_n^2 is the variance of visual noise; σ_u^2 is the variance of Gaussian noise of the distortion model; λ_k are eigenvalues of a covariance matrix; g_i is deterministic scalar field and s^N is the maximum likelihood estimate of S^N (or a realization of S^N for a particular reference image), S^N is a vector of N elements of a random field of positive scalars, $s^{N,j}$ is the j-subband of s^N. The random field used is Gaussian scale mixtures (GSMs). We must notice that the subbands are extracted from the Natural Scene Statistics (NSS) model by the wavelet decomposition.

TABLE 2. SSIM results of the methods for three native images of the MATLAB library with different SPN ratios.

Image	Filters	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lena	BPDF	0.9846	0.9652	0.9409	0.9090	0.8664	0.8055	0.7142	0.5257	0.2770
	DBA	0.9887	0.9743	0.9553	0.9308	0.8976	0.8542	0.7935	0.7083	0.5809
	MDBUTMF	0.9867	0.9469	0.8498	0.8184	0.8635	0.8883	0.8641	0.7838	0.3898
	NAFSMF	0.9837	0.9664	0.9485	0.9283	0.9057	0.8808	0.8477	0.8037	0.6806
	AWMF	0.9822	0.9738	0.9634	0.9504	0.9341	0.9115	0.8827	0.8424	0.7682
	DAMF	0.9902	0.9788	0.9655	0.9494	0.9304	0.9064	0.8770	0.8370	0.7620
	FDS	0.9894	0.9759	0.9573	0.9293	0.8858	0.8280	0.7441	0.6379	0.5020
	IMF	**0.9913**	**0.9796**	**0.9675**	**0.9541**	**0.9383**	**0.9183**	**0.8953**	**0.8623**	**0.8058**
Peppers	BPDF	0.9746	0.9471	0.9185	0.8814	0.8410	0.7826	0.6997	0.5463	0.1927
	DBA	0.9791	0.9551	0.9286	0.8959	0.8570	0.8063	0.7451	0.6564	0.5272
	MDBUTMF	0.9804	0.9318	0.8280	0.7873	0.8356	0.8513	0.8247	0.7450	0.4029
	NAFSMF	0.9783	0.9561	0.9335	0.9085	0.8834	0.8558	0.8186	0.7704	0.6499
	AWMF	0.9611	0.9570	0.9427	0.9222	0.8980	0.8663	0.8283	0.7801	0.7123
	DAMF	0.9815	0.9606	0.9389	0.9131	0.8866	0.8541	0.8180	0.7719	0.7049
	FDS	0.9825	0.9627	0.9396	0.9079	0.8687	0.8110	0.7355	0.6360	0.5085
	IMF	**0.9858**	**0.9684**	**0.9504**	**0.9299**	**0.9091**	**0.8846**	**0.8572**	**0.8219**	**0.7700**
House	BPDF	0.9937	0.9853	0.9722	0.9539	0.9259	0.8759	0.8027	0.6738	0.4467
	DBA	0.9972	0.9923	0.9843	0.9718	0.9538	0.9228	0.8730	0.8083	0.6982
	MDBUTMF	0.9950	0.9473	0.8138	0.7859	0.8799	0.9452	0.9406	0.8637	0.4133
	NAFSMF	0.9910	0.9825	0.9733	0.9638	0.9534	0.9387	0.9193	0.8868	0.7719
	AWMF	0.9933	0.9922	0.9904	0.9875	0.9831	0.9754	0.9624	0.9405	0.8922
	DAMF	0.9982	0.9954	0.9913	0.9858	0.9793	0.9704	0.9569	0.9350	0.8839
	FDS	0.9976	0.9937	0.9865	0.9732	0.9524	0.9171	0.8652	0.8028	0.7116
	IMF	**0.9995**	**0.9982**	**0.9955**	**0.9921**	**0.9880**	**0.9805**	**0.9707**	**0.9498**	**0.9128**

IEF is defined as [41]:

$$IEF\,(U,V,B) := \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(b_{ij}-u_{ij}\right)^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(v_{ij}-u_{ij}\right)^2} \tag{11}$$

where, $U := [u_{ij}]$ is a ground truth, $V := [v_{ij}]$ is an evaluated image, and $B := [b_{ij}]$ is a noisy image.

SSIM is defined as [39]:

$$SSIM(U,V) := \frac{(2\mu_U\mu_V + C_1) + (2\sigma_{UV} + C_2)}{(\mu_U^2 + \mu_V^2 + C_1) + (\sigma_U^2 + \sigma_V^2 + C_2)} \tag{12}$$

where μ_U, μ_V, σ_U, σ_V, and σ_{UV} are the average intensities, standard deviations, and cross-covariance of a ground truth image U and an evaluated image V, respectively. Also, $C_1 := (K_1L)^2$ and $C_2 := (K_2L)^2$ are two constants such that $K_1 := 0.01$, $K_2 := 0.03$ and $L := 255$ for 8-bit grayscale images.

The MSSIM is defined as [42]:

$$MSSIM\,(U,V) := (l_M\,(U,V))^{\alpha M}\prod_{j=1}^{M}\left(c_j\,(U,V)\right)^{\beta_j} \times \left(s_j\,(U,V)\right)^{\gamma_j} \tag{13}$$

where

$$l_M\,(U,V) := \frac{2\mu_U\mu_V + C_1}{\mu_U^2 + \mu_V^2 + C_1} \text{ on scale } M,$$

$$c_j\,(U,V) := \frac{2\sigma_U\sigma_V + C_2}{\sigma_U^2 + \sigma_V^2 + C_2} \text{ on each scale } \quad j = 1,..,M,$$

$$s_j\,(U,V) := \frac{\sigma_{UV} + C_3}{\sigma_U\sigma_V + C_3} \text{ on each scale } \quad j = 1,\ldots,M,$$

$C_1 := (K_1L)^2$, $C_2 := (K_2L)^2$, $C_3 := \frac{C_2}{2}$, $K_1 := 0.01$, $K_2 := 0.03$ and $L = 255$ for 8-bit grayscale images; j represents a resolution scale after each low-pass filtering and downsampling, and M represents the total number of scales; $\alpha, \beta_j, \gamma_j$ are used to adjust the relative importance of different components. In the experiments, we use five scales as in [42], i.e., $M = 5$. MSSIM is better than SSIM in terms of its correlation with a human judgment of the images [42].

Note that, values of SSIM, VIF, MSSIM are in the range of [0, 1]. A higher value of PSNR, IEF, SSIM, VIF or MSSIM indicates a better image quality.

TABLE 3. PSNR, SSIM, VIF, IEF and MSSIM values of denoising results for the 20 traditional test images with different SPN ratios.

Algorithm	Criterion	10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean
	PSNR	36.96	33.52	31.00	28.86	26.84	24.60	21.98	17.75	10.50	25.78
	SSIM	0.9800	0.9560	0.9258	0.8873	0.8365	0.7670	0.6687	0.5047	0.2477	0.7526
BPDF	VIF	0.8138	0.6785	0.5585	0.4502	0.3461	0.2486	0.1580	0.0771	0.0327	0.3737
	IEF	222.92	183.97	146.55	115.35	84.61	58.47	36.44	14.83	3.19	96.26
	MSSIM	0.9997	0.9983	0.9980	0.9978	0.9972	0.9899	0.9841	0.9545	0.6372	0.9508
	PSNR	37.92	34.71	32.31	30.26	28.35	26.44	24.32	21.98	18.97	28.36
	SSIM	0.9852	0.9663	0.9422	0.9111	0.8702	0.8169	0.7457	0.6487	0.5125	0.8221
DBA	VIF	0.8489	0.7241	0.6114	0.5062	0.4052	0.3094	0.2173	0.1345	0.0620	0.4243
	IEF	323.06	256.93	211.02	169.75	128.36	95.04	65.11	42.80	22.71	146.09
	MSSIM	0.9997	0.9996	0.9992	0.9974	0.9975	0.9959	0.9929	0.9840	0.9647	0.9923
	PSNR	37.17	32.80	29.44	28.44	28.78	28.55	27.35	24.29	16.24	28.12
	SSIM	0.9811	0.9354	0.8328	0.7948	0.8358	0.8504	0.8189	0.7309	0.3814	0.7957
MDBUTMF	VIF	0.8193	0.6635	0.5009	0.4397	0.4263	0.3917	0.3250	0.2199	0.0726	0.4288
	IEF	306.59	145.86	86.16	91.83	138.51	177.82	155.03	70.66	11.31	131.53
	MSSIM	0.9995	0.9991	0.9947	0.9945	0.9939	0.9920	0.9911	0.9894	0.9665	0.9912
	PSNR	36.09	33.28	31.52	30.16	29.02	27.97	26.83	25.46	22.37	29.19
	SSIM	0.9765	0.9528	0.9276	0.9003	0.8704	0.8367	0.7958	0.7401	0.6190	0.8466
NAFSMF	VIF	0.7829	0.6682	0.5769	0.4991	0.4262	0.3559	0.2853	0.2098	0.1213	0.4362
	IEF	211.85	205.64	196.27	186.30	173.49	156.79	133.84	104.06	49.77	157.56
	MSSIM	0.9997	0.9987	0.9983	0.9979	0.9976	0.9970	0.9961	0.9942	0.9963	0.9973
	PSNR	36.35	35.01	33.84	32.67	31.46	30.16	28.68	27.01	24.69	31.10
	SSIM	0.9751	0.9648	0.9515	0.9349	0.9137	0.8863	0.8490	0.7968	0.7089	0.8868
AWMF	VIF	0.7855	0.7301	0.6726	0.6128	0.5475	0.4772	0.4004	0.3088	0.1902	0.5250
	IEF	212.73	316.45	356.24	352.96	318.71	269.40	217.16	161.48	100.37	256.17
	MSSIM	0.9995	0.9991	0.9989	0.9984	0.9981	0.9975	0.9970	0.9967	0.9902	0.9973
	PSNR	39.61	36.34	34.17	32.43	30.98	29.66	28.27	26.71	24.36	31.39
	SSIM	0.9869	0.9718	0.9542	0.9335	0.9088	0.8795	0.8423	0.7908	0.7027	0.8856
DAMF	VIF	0.8720	0.7742	0.6884	0.6106	0.5371	0.4666	0.3935	0.3057	0.1886	0.5374
	IEF	562.92	443.72	364.80	313.91	268.81	231.04	193.41	148.89	90.89	290.93
	MSSIM	0.9997	0.9996	0.9994	0.9987	0.9991	0.9982	0.9980	0.9966	0.9927	0.9980
	PSNR	38.15	34.94	32.48	30.17	28.05	25.95	23.84	21.64	18.98	28.24
	SSIM	0.9857	0.9694	0.9474	0.9158	0.8718	0.8119	0.7335	0.6347	0.5142	0.8205
FDS	VIF	0.8494	0.7392	0.6260	0.5129	0.4030	0.3036	0.2137	0.1336	0.0652	0.4274
	IEF	334.14	279.67	214.58	159.10	112.13	79.63	55.21	36.61	21.84	143.66
	MSSIM	0.9997	0.9994	0.9994	0.9987	0.9973	0.9957	0.9884	0.9707	0.9495	0.9888
	PSNR	**39.76**	**36.84**	**34.92**	**33.39**	**32.11**	**30.81**	**29.55**	**28.09**	**26.10**	**32.40**
	SSIM	**0.9892**	**0.9752**	**0.9593**	**0.9411**	**0.9195**	**0.8931**	**0.8603**	**0.8143**	**0.7408**	**0.8992**
IMF	VIF	**0.8861**	**0.7954**	**0.6955**	**0.6225**	**0.5608**	**0.4970**	**0.4200**	**0.3230**	**0.2151**	**0.5573**
	IEF	**615.12**	**556.09**	**496.82**	**446.30**	**395.56**	**339.61**	**282.97**	**215.02**	**141.61**	**387.68**
	MSSIM	**0.9998**	**0.9997**	**0.9995**	**0.9988**	**0.9987**	**0.9985**	**0.9984**	**0.9975**	**0.9969**	**0.9986**

TABLE 4. PSNR, SSIM, VIF, IEF and MSSIM values of denoising results for the TESTIMAGES dataset with different SPN ratios.

Algorithm	Criterion	10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean
	PSNR	38.56	34.36	31.35	28.74	26.34	23.71	20.59	15.85	8.77	25.36
	SSIM	0.9855	0.9666	0.9407	0.9050	0.8552	0.7835	0.6769	0.4915	0.1989	0.7560
BPDF	VIF	0.8359	0.6991	0.5767	0.4647	0.3610	0.2616	0.1672	0.0837	0.0390	0.3876
	IEF	357.52	258.79	190.69	133.13	91.45	58.42	31.26	11.47	2.49	126.14
	MSSIM	0.9996	0.9986	0.9911	0.9886	0.9827	0.9720	0.9660	0.8173	0.4408	0.9063
	PSNR	40.47	36.38	33.47	30.97	28.61	26.27	23.81	21.01	17.46	28.72
	SSIM	0.9908	0.9780	0.9600	0.9348	0.8994	0.8490	0.7779	0.6741	0.5197	0.8426
DBA	VIF	0.8756	0.7550	0.6407	0.5319	0.4275	0.3286	0.2342	0.1452	0.0688	0.4453
	IEF	572.45	423.98	325.29	229.32	158.86	108.87	69.11	39.95	18.58	216.27
	MSSIM	0.9995	0.9991	0.9985	0.9978	0.9957	0.9930	0.8651	0.8325	0.8043	0.9428
	PSNR	32.76	29.86	27.14	26.40	26.87	26.68	25.60	22.64	15.45	25.93
	SSIM	0.9826	0.9330	0.8110	0.7764	0.8454	0.8829	0.8617	0.7773	0.4293	0.8111
MDBUTMF	VIF	0.8397	0.6773	0.5066	0.4516	0.4484	0.4244	0.3576	0.2486	0.0942	0.4498
	IEF	192.17	102.27	63.30	70.09	110.07	146.88	123.07	57.10	10.77	97.30
	MSSIM	0.9980	0.9973	0.9964	0.9954	0.9947	0.9908	0.9890	0.9877	0.9516	0.9890
	PSNR	37.15	34.15	32.17	30.54	29.22	27.92	26.53	24.80	21.36	29.32
	SSIM	0.9791	0.9602	0.9410	0.9206	0.8982	0.8717	0.8380	0.7879	0.6644	0.8735
NAFSMF	VIF	0.8020	0.6952	0.6086	0.5310	0.4590	0.3895	0.3174	0.2398	0.1437	0.4651
	IEF	304.35	285.25	267.13	242.27	212.34	180.34	144.26	104.37	45.16	198.39
	MSSIM	0.9985	0.9982	0.9979	0.9975	0.9959	0.9945	0.9905	0.9854	0.9770	0.9928
	PSNR	37.55	36.53	35.45	34.24	32.91	31.36	29.67	27.68	24.87	32.25
	SSIM	0.9808	0.9749	0.9671	0.9568	0.9430	0.9233	0.8951	0.8511	0.7674	0.9177
AWMF	VIF	0.8022	0.7571	0.7075	0.6515	0.5893	0.5207	0.4415	0.3464	0.2203	0.5596
	IEF	289.39	454.80	532.61	531.00	471.73	396.56	306.92	221.15	126.53	370.08
	MSSIM	0.9984	0.9974	0.9964	0.9961	0.9955	0.9952	0.9947	0.9903	0.9848	0.9943
	PSNR	40.96	37.26	34.71	32.76	31.19	29.73	28.21	26.40	23.66	31.65
	SSIM	0.9910	0.9815	0.9697	0.9553	0.9379	0.9159	0.8870	0.8430	0.7577	0.9154
DAMF	VIF	0.9015	0.8131	0.7295	0.6506	0.5780	0.5077	0.4326	0.3417	0.2171	0.5747
	IEF	726.33	576.52	469.80	386.60	318.24	267.07	217.56	162.43	89.57	357.12
	MSSIM	0.9995	0.9985	0.9981	0.9978	0.9971	0.9966	0.9905	0.9901	0.9747	0.9936
	PSNR	38.18	35.32	32.83	30.38	27.98	25.64	23.22	20.65	17.57	27.98
	SSIM	0.9880	0.9766	0.9595	0.9327	0.8924	0.8344	0.7547	0.6495	0.5168	0.8338
FDS	VIF	0.8574	0.7548	0.6447	0.5295	0.4199	0.3198	0.2288	0.1472	0.0764	0.4421
	IEF	309.39	302.72	253.48	189.21	135.08	93.59	61.97	37.86	19.72	155.89
	MSSIM	0.9980	0.9858	0.9828	0.9802	0.9789	0.9759	0.9737	0.9527	0.9093	0.9708
	PSNR	**42.19**	**38.93**	**36.76**	**34.94**	**33.42**	**31.89**	**30.29**	**28.50**	**26.11**	**33.67**
	SSIM	**0.9916**	**0.9830**	**0.9727**	**0.9604**	**0.9452**	**0.9253**	**0.8986**	**0.8590**	**0.7914**	**0.9252**
IMF	VIF	**0.9086**	**0.8277**	**0.7530**	**0.6803**	**0.6074**	**0.5337**	**0.4546**	**0.3676**	**0.2580**	**0.5990**
	IEF	**849.30**	**803.18**	**731.29**	**645.53**	**549.33**	**464.53**	**371.44**	**281.35**	**177.31**	**541.47**
	MSSIM	**0.9997**	**0.9994**	**0.9988**	**0.9984**	**0.9979**	**0.9970**	**0.9962**	**0.9958**	**0.9899**	**0.9970**

TABLE 5. PSNR, SSIM, VIF, IEF and MSSIM values of denoising results for the berkeley image dataset with different SPN ratios.

Algorithm	Criterion	10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean
	PSNR	35.38	31.83	29.36	27.33	25.52	23.64	21.45	17.92	12.12	24.95
	SSIM	0.9776	0.9506	0.9165	0.8731	0.8177	0.7455	0.6487	0.5015	0.2912	0.7469
BPDF	VIF	0.7488	0.6094	0.4944	0.3924	0.3014	0.2169	0.1390	0.0715	0.0334	0.3341
	IEF	165.06	141.88	119.13	98.11	77.91	60.43	41.56	21.70	6.55	81.37
	MSSIM	0.9948	0.9940	0.9940	0.9909	0.9901	0.9592	0.9658	0.8952	0.6542	0.9376
	PSNR	35.84	32.67	30.37	28.41	26.64	24.90	23.08	21.09	18.55	26.84
	SSIM	0.9827	0.9612	0.9338	0.8984	0.8529	0.7945	0.7191	0.6218	0.4907	0.8061
DBA	VIF	0.7791	0.6496	0.5383	0.4390	0.3480	0.2634	0.1839	0.1119	0.0511	0.3738
	IEF	190.03	174.82	152.08	126.20	99.31	77.91	56.59	39.81	23.81	104.51
	MSSIM	0.9930	0.9922	0.9918	0.9909	0.9909	0.9875	0.9836	0.9782	0.9458	0.9838
	PSNR	34.15	30.54	27.68	26.66	26.68	26.28	25.24	22.94	16.49	26.29
	SSIM	0.9742	0.9230	0.8147	0.7716	0.8077	0.8170	0.7785	0.6902	0.3817	0.7732
MDBUTMF	VIF	0.7362	0.5851	0.4438	0.3836	0.3632	0.3292	0.2683	0.1816	0.0645	0.3728
	IEF	148.47	96.83	65.05	68.99	95.82	120.32	106.90	60.84	13.98	86.36
	MSSIM	0.9930	0.9914	0.9893	0.9871	0.9821	0.9768	0.9704	0.9583	0.9579	0.9785
	PSNR	33.83	31.08	29.38	28.07	26.99	26.02	25.02	23.82	21.14	27.26
	SSIM	0.9697	0.9412	0.9115	0.8794	0.8441	0.8042	0.7563	0.6944	0.5712	0.8191
NAFSMF	VIF	0.7065	0.5899	0.5031	0.4296	0.3640	0.3016	0.2379	0.1716	0.0956	0.3777
	IEF	120.43	123.84	124.23	121.10	116.12	110.05	97.50	79.85	40.46	103.73
	MSSIM	0.9917	0.9904	0.9901	0.9892	0.9882	0.9877	0.9871	0.9834	0.9819	0.9877
	PSNR	34.16	32.68	31.44	30.26	29.10	27.85	26.50	24.99	23.01	28.89
	SSIM	0.9708	0.9577	0.9417	0.9220	0.8974	0.8654	0.8225	0.7627	0.6646	0.8672
AWMF	VIF	0.7035	0.6422	0.5817	0.5217	0.4505	0.3867	0.3170	0.2480	0.1476	0.4443
	IEF	125.16	179.17	203.07	205.77	194.06	173.09	145.63	116.31	80.79	158.12
	MSSIM	0.9940	0.9930	0.9917	0.9916	0.9907	0.9878	0.9852	0.9846	0.9821	0.9890
	PSNR	37.01	33.81	31.72	30.05	28.65	27.36	26.08	24.66	22.64	29.11
	SSIM	0.9837	0.9663	0.9460	0.9217	0.8931	0.8588	0.8155	0.7566	0.6582	0.8667
DAMF	VIF	0.7463	0.6887	0.6091	0.5049	0.4469	0.3430	0.3063	0.2207	0.1213	0.4430
	IEF	271.77	249.60	222.51	199.32	176.75	155.69	132.93	108.63	70.66	176.43
	MSSIM	0.9946	0.9942	0.9929	0.9916	0.9901	0.9867	0.9841	0.9826	0.9802	0.9886
	PSNR	35.66	32.62	30.37	28.39	26.55	24.81	23.08	21.23	18.93	26.85
	SSIM	0.9820	0.9631	0.9383	0.9041	0.8574	0.7963	0.7183	0.6222	0.5040	0.8095
FDS	VIF	0.7725	0.6598	0.5558	0.4553	0.3612	0.3249	0.1961	0.1255	0.0631	0.3905
	IEF	160.85	153.83	134.25	110.19	88.18	69.68	53.29	38.97	25.17	92.71
	MSSIM	0.9944	0.9942	0.9936	0.9934	0.9926	0.9890	0.9717	0.9670	0.9252	0.9801
	PSNR	**37.11**	**34.22**	**32.33**	**30.86**	**29.62**	**28.46**	**27.31**	**26.04**	**24.40**	**30.04**
	SSIM	**0.9840**	**0.9674**	**0.9483**	**0.9260**	**0.8996**	**0.8673**	**0.8261**	**0.7710**	**0.6878**	**0.8753**
IMF	VIF	**0.7881**	**0.6883**	**0.6364**	**0.5345**	**0.4673**	**0.4002**	**0.3307**	**0.2547**	**0.1620**	**0.4736**
	IEF	**281.71**	**281.12**	**269.34**	**256.03**	**238.42**	**219.03**	**196.45**	**170.70**	**135.75**	**227.62**
	MSSIM	**0.9951**	**0.9945**	**0.9946**	**0.9942**	**0.9945**	**0.9890**	**0.9955**	**0.9880**	**0.9794**	**0.9917**

B. DATASETS AND TEST CASES

We implement IMF algorithm on MATLAB R2019a. To assess denoising quality of the proposed method, we use 20 native images with the same size of 512 × 512 pixels of the MATLAB Library: Lena, Cameraman, Barbara, Baboon, Peppers, Living Room, Lake, Plane, Hill, Pirate,

TABLE 6. Execution time comparison of the methods.

Noise Density	BPDF	DBA	MDBUTMF	NAFSMF	AWMF	DAMF	FDS	IMF
10%	0.98	3.96	1.89	1.26	3.78	0.19	0.99	0.53
20%	1.65	4.10	2.88	2.51	3.10	0.32	1.76	1.03
30%	2.47	4.16	3.86	3.75	2.76	0.49	2.83	2.41
40%	3.36	4.19	4.74	4.93	2.71	0.66	4.16	3.14
50%	4.05	4.07	6.07	6.19	2.52	0.78	5.80	4.14
60%	4.93	4.30	7.13	7.49	2.47	0.96	7.95	5.07
70%	5.65	4.31	7.96	8.79	2.43	1.13	10.40	6.07
80%	6.41	4.32	8.69	9.93	2.66	1.31	13.44	8.85
90%	7.12	4.31	9.09	11.01	3.16	1.58	15.78	10.01
Mean	**4.07**	**4.19**	**5.81**	**6.21**	**2.84**	**0.83**	**7.01**	**4.58**

Boat, House, Bridge, Elaine, Flintstones, Flower, Parrot, Dark-Haired Woman, Blonde Woman, Einstein; 40 images with the same size of 600 × 600 pixels of the TESTIMAGES dataset [43] and 200 images of the BSDS dataset: *https://www2. eecs. berkeley. edu/research/projects/CS/vision/bsds/BSDS300/html/DataSet/images.html* of the UC Berkeley. All images of three datasets are grayscale. The images of the MATLAB library and the TESTIMAGES dataset are stored in the PNG format. For the BSDS dataset, the images are stored in the JPEG format. The size of the images of the BSDS dataset is 481 × 321 or 321 × 481. All images are published for use under a free license.

We use PSNR, SSIM, VIF, IEF, and MSSIM to assess image quality after denoising. We compare denoising results with ones of the following state-of-the-art denoising methods: BPDF [30], DBA [31], MDBUTMF [32], NAFSMF [33], AWMF [25], FDS [35], and DAMF [34]. We also must notice that AWMF works more effective than MF and AMF [18], so we do not need to compare denoising results with ones of MF and AMF.

We consider three test cases: evaluate denoising results of the methods by intuition; evaluate denoising results based on qualitative metrics for three datasets: the MATLAB library, the TESTIMAGES dataset, the BSDS dataset; and evaluate execution time of the methods.

C. DISCUSSION

1) THE FIRST TEST CASE

We test on three images of the MATLAB library: the house image, the peppers image, and the Lena image. We focus on the intuitive results assessment and qualitative assessment based on the metrics.

Firstly, we consider the house image. We added SPN with a noise level of 60%. Denoising results of the methods are shown in Figure 2. We cropped an image part with the size of 140 × 140 pixels to make it easy to distinguish. From the results, we can see that BPDF and DBA were unable to preserve structures and edges of images. FDS worked better, but the denoising result contains some defects, especially for the bottom-right region of the image. MDBUTMF, AWMF, and DAMF removed noise very well, but they created artifacts and made edges sharpen. The artifacts are visible on edges. NAFSMF also removed noise very well and avoided creating artifacts. However, the denoising result of NAFSMF was lost many details. IMF removed noise excellently. All noises were removed. No artifacts remain, and edges are smoothed naturally. On the other hand, IMF preserved the edges, details and other image structures very well. The PSNR, SSIM values (and VIF, IEF, MSSIM) of denoising result of IMF are the highest: BPDF (27.41 dB, 0.8759), DBA (30.38 dB, 0.9228), MDBUTMF (35.02 dB, 0.9452), NAFSMF (33.98 dB, 0.9387), AWMF (37.80 dB, 0.9754), DAMF (36.51 dB, 0.9704), FDS (29.77 dB, 0.9171), IMF (**39.77** dB, **0.9805**).

Secondly, we consider the pepper image. For this case, we add SPN with a noise level of 90%. Denoising results are presented in Figure 3. For this very high noise level, BPDF cannot work correctly. BPDF destroyed image structures, and we cannot see anything on the image. MDBUTMF removed noise completely, but it created many defects like ink. DBA and FDS removed noise completely, but they also created defects: a raindrop effect for DBA and a windblown-dust effect for FDS. A little noise remains on the results of the NAFSMF and DAMF. Besides, edges in the denoising results of AWMF and DAMF are very sharpened. IMF removed noise completely. Edges are smoothed naturally. By the PSNR, SSIM values (as well as VIF, IEF, MSSIM), denoising result of IMF is the best: BPDF (9.21 dB, 0.1927), DBA (18.72 dB, 0.5272), MDBUTMF (16.91 dB, 0.4029), NAFSMF (23.60 dB, 0.6499), AWMF (26.23 dB, 0.7123), DAMF (25.87 dB, 0.7049), FDS (18.15 dB, 0.5085), IMF (**27.88** dB, **0.7700**).

Thirdly, we consider the Lena image. In this case, we do not compare to other methods. We only consider the effectiveness of noise removal for various noise levels of IMF. We consider eight noise levels: 10%, 20%, 30%, 40%, 50%, 60%, 70% and 80%. Denoising results are shown in Figure 4. As we

can see, IMF can remove noise excellently and can preserve image structures very well even for very high noise levels. PSNR and SSIM values of denoising results with various noise levels of three above images are given in Table 1 and Table 2, respectively. It is easy to see that the quality of denoising results by both PSNR and SSIM of IMF are better than those of other methods.

2) THE SECOND TEST CASE

We assess denoising quality by the metrics. The assessments base on the average values of all images of the datasets. Tables 3, 4, 5 show the average PSNR value, the average SSIM value, the average VIF value, the average IEF value and the average MSSIM value of denoising results of the methods for all images of the MATLAB library (Table 1), the TESTIMAGES dataset (Table 2) and the UC-Berkeley dataset (Table 3), respectively. According to the acquired results, we can see that the denoising results of IMF are always better than those of the other compared methods.

3) THE THIRD TEST CASE

We assess the processing performance (i.e., execution time) of the denoising methods. Table 6 presents the execution time of the methods. The execution time depends on noise levels too much. DAMF is the fastest. FDS is the slowest. The difference in execution time of IMF, BPDF, and DBA is very small. IMF can work faster than FDS, NAFSMF and MDBUTMF. It must be noted that for noise levels up to 30%, IMF can remove noise very fast. It is only slower than DAMF.

As mentioned above, IMF is designed to remove noise in natural images, where dark and bright regions are usually not entirely black or completely white. As in the above test cases, even when there are some profoundly dark/bright regions in natural images (e.g., the Lena image, the Peppers image) containing some pixels of boundary values, IMF still removed noise very effectively and did not create any defects.

IV. CONCLUSION

An iterative mean filter (IMF) for SPN removal has been proposed. In IMF, we only consider a fixed-size window of 3×3 pixels, and we use the constrained mean of the window instead of median to evaluate new gray value for the center pixel. Hence, IMF works more effectively than the methods using dynamic adaptive windows. An iterative procedure has also been provided to integrate the power of removing high-density noise for IMF. From a vast number of tests, it is seen that IMF could remove noise excellently, and it can preserve image structures, edges, and details very well. We also can confirm that IMF outperforms other state-of-the-art compared SPN denoising methods.

For future work, we focus on study for an extensive IMF to remove the random-valued impulse noise (RVIN).

REFERENCES

[1] Q.-Q. Chen, M.-H. Hung, and F. M. Zou, "Effective and adaptive algorithm for pepper-and-salt noise removal," *IET Image Process.*, vol. 11, no. 9, pp. 709–716, 2017.
[2] G. Wang, D. Li, W. Pan, and Z. Zang, "Modified switching median filter for impulse noise removal," *Signal Process.*, vol. 90, no. 12, pp. 3213–3218, 2010.
[3] Y. Wang, J. Wang, S. Xiao, and L. Han, "An efficient adaptive fuzzy switching weighted mean filter for salt-and-pepper noise removal," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1582–1586, Nov. 2016.
[4] P.-Y. Chen and C.-Y. Lien, "An efficient edge-preserving algorithm for removal of salt-and-pepper noise," *IEEE Signal Process. Lett.*, vol. 15, pp. 833–836, 2008.
[5] F. Duan and Y.-J. Zhang, "A highly effective impulse noise detection algorithm for switching median filters," *IEEE Signal Process. Lett.*, vol. 17, no. 7, pp. 647–650, May 2010.
[6] X. Wang, S. Shen, G. Shi, Y. Xu, and P. Zhang, "Iterative non-local means filter for salt and pepper noise removal," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 440–450, Jul. 2016.
[7] D. Xiangyu, M. Yide, and D. Ming, "A new adaptive filtering method for removing salt and pepper noise based on multilayered PCNN," *Pattern Recognit. Lett.*, vol. 79, pp. 8–17, Aug. 2016.
[8] D. Li, S. Yan, X. Cai, Y. Cao, and S. Wang, "An integrated image filter for enhancing change detection results," *IEEE Access*, vol. 7, pp. 91034–91051, 2019.
[9] E. López-Rubio, "Restoration of images corrupted by Gaussian and uniform impulsive noise," *Pattern Recognit.*, vol. 43, no. 5, pp. 1835–1846, 2010.
[10] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 60–65.
[11] K. B. Khan, A. A. Khaliq, M. Shahid, and J. A. Shah, "Primjena ponderiranog stupnjevanog filtra u izoštravanju rendgenskih slika uz postojanje Poissonova šuma," *Tehnicki Vjesnik*, vol. 23, no. 6, pp. 1755–1762, 2016.
[12] K. B. Khan, M. Shahid, H. Ullah, E. Rehman, and M. M. Khan, "Adaptive trimmed mean autoregressive model for reduction of Poisson noise in scintigraphic images," *IIUM Eng. J.*, vol. 19, no. 2, pp. 68–79, 2018.
[13] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, Sep. 2005.
[14] Z. Li, G. Liu, Y. Xu, and Y. Cheng, "Modified directional weighted filter for removal of salt & pepper noise," *Pattern Recognit. Lett.*, vol. 40, pp. 113–120, Apr. 2014.
[15] T. Bai, J. Tan, M. Hu, and Y. Wang, "A novel algorithm for removal of salt and pepper noise using continued fractions interpolation," *Signal Process.*, vol. 102, pp. 247–255, Sep. 2014.
[16] S. Wang, "Dictionary learning based impulse noise removal via ℓ_1-ℓ_1 minimization," *Signal Process.*, vol. 93, no. 9, pp. 2696–2708, 2013.
[17] K. Panetta, L. Bao, and S. Agaian, "A new unified impulse noise removal algorithm using a new reference sequence-to-sequence similarity detector," *IEEE Access*, vol. 6, pp. 37225–37236, 2018.
[18] F. Taherkhani and M. Jamzad, "Restoring highly corrupted images by impulse noise using radial basis functions interpolation," *IET Image Process.*, vol. 12, no. 1, pp. 20–30, 2018.
[19] M. González-Hidalgo, S. Massanet, D. Ruiz-Aguilera, and A. Mir, "Improving salt and pepper noise removal using a fuzzy mathematical morphology-based filter," *Appl. Soft Comput.*, vol. 63, pp. 167–180, Feb. 2018.
[20] J. Wu and C. Tang, "An efficient decision-based and edge-preserving method for salt-and-pepper noise removal," *Pattern Recognit. Lett.*, vol. 32, no. 15, pp. 1974–1981, 2011.
[21] X. Qi, B. Liu, and J. Xu, "A neutrosophic filter for high-density salt and pepper noise based on pixel-wise adaptive smoothing parameter," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 1–10, Apr. 2016.
[22] S. Wang, "Dictionary learning based impulse noise removal via ℓ_1-ℓ_1 minimization," *Signal Process.*, vol. 93, no. 9, pp. 2696–2708, 2013.
[23] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.
[24] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.
[25] P. Zhang and F. Li, "A new adaptive weighted mean filter for removing salt-and-pepper noise," *IEEE Signal Process. Lett.*, vol. 21, no. 10, pp. 1280–1283, Jun. 2014.
[26] S. B. S. Fareed and S. S. Khader, "Fast adaptive and selective mean filter for the removal of high-density salt and pepper noise," *IET Image Process.*, vol. 12, no. 8, pp. 1378–1387, 2018.

[27] J. Chen, Y. Zhan, H. Cao, and X. Wu, "Adaptive probability filter for removing salt and pepper noises," *IET Image Process.*, vol. 12, no. 6, pp. 863–871, 2018.
[28] G. Pok and K. H. Ryu, "Efficient block matching for removing impulse noise," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1176–1180, Aug. 2018.
[29] X. Zhang and Y. Xiong, "Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter," *IEEE Signal Process. Lett.*, vol. 16, no. 4, pp. 295–298, Feb. 2009.
[30] U. Erkan and L. Gökrem, "A new method based on pixel density in salt and pepper noise removal," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 26, no. 1, pp. 162–171, 2018.
[31] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision-based algorithm for removal of high-density impulse noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, Mar. 2007.
[32] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 287–290, May 2011.
[33] K. K. V. Toh and N. A. M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 281–284, Mar. 2010.
[34] U. Erkan, L. Gökrem, and S. Enginoğlu, "Different applied median filter in salt and pepper noise," *Comput. Elect. Eng.*, vol. 70, pp. 789–798, Aug. 2018.
[35] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3170–3176, Feb. 2018.
[36] F. Ahmed and S. Das, "Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1352–1358, Oct. 2014.
[37] G. Balasubramanian, A. Chilambuchelvan, S. Vijayan, and G. Gowrison, "Probabilistic decision based filter to remove impulse noise using patch else trimmed median," *AEU-Int. J. Electron. Commun.*, vol. 70, no. 4, pp. 471–481, 2016.
[38] A. Peiravi and H. T. Kheibari, "A fast algorithm for connectivity graph approximation using modified Manhattan distance in dynamic networks," *Appl. Math. Comput.*, vol. 201, nos. 1–2, pp. 319–332, 2008.
[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[40] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Jan. 2006.
[41] I. Djurović, "Combination of the adaptive Kuwahara and BM3D filters for filtering mixed Gaussian and impulsive noise," *Signal, Image Video Process.*, vol. 11, no. 4, pp. 753–760, 2017.
[42] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Conf. Rec. Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402.
[43] N. Asuni and A. Giachetti, "TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms," in *Proc. Eurograph. Italian Chapter Conf.*, 2014, pp. 1–3.

UĞUR ERKAN received the B.Sc. degree in computer engineering from Selçuk University, Turkey, in 2001, the M.Sc. degree in mathematics and the Ph.D. degree in mechatronics engineering from Gaziosmanpaşa University, Turkey, in 2012 and 2017, respectively.

He is currently an Assistant Professor with the Department of Computer Engineering, Karamanoğlu Mehmetbey University, Karaman, Turkey. His research interests are in image processing, image denoising, image restoration, image enhancement, fuzzy sets, and pattern recognition.

DANG NGOC HOANG THANH received the bachelor's degree and the M.Sc. degree in applied mathematics from Belarusian State University, in 2008 and 2009, respectively, and the Ph.D. degree in computer science from Tula State University, Russia, in 2016.

He is currently a Lecturer/Researcher with the Hue College of Industry, Vietnam. He has over 50 works on international peer-reviewed journals and conference proceedings, three book chapters, one book, and one European Patent. His research interests are image processing, computer vision, machine learning, computational mathematics, and optimization. He is a member of scientific organization INSTICC, Portugal, ACM, USA, and IAENG, Taiwan. He is also a member of international conferences committee, such as IEEE ICCE 2018, Vietnam, IWBBIO, Spain, IEEE ICIEV, USA, IEEE ICEEE, Turkey, ICIEE, Japan, ICoCTA, Australia, and ICMTEL, U.K.

LE MINH HIEU received the bachelor's degree, the M.Sc. degree in applied mathematics, and the Ph.D. degree in computational mathematics from Belarusian State University, in 2009, 2010, and 2018, respectively.

He is currently a Lecturer with the University of Economics - The University of Da Nang, Vietnam. He has over 30 works on international peer-reviewed journals and conference proceedings. His research interests are finite difference schemes, nonlinear PDEs, image processing, machine learning, financial mathematics, and computational mathematics. In 2018, he was awarded a certificate for excellent researchers by the Danang People's Committee.

SERDAR ENGİNOĞLU received the B.Sc. degree in mathematics from Atatürk University, Erzurum, Turkey, in 1998, and the M.Sc. and Ph.D. degrees in mathematics from Gaziosmanpaşa University, Tokat, Turkey, in 2009 and 2012, respectively.

He is currently an Assistant Professor with the Department of Mathematics, Faculty of Arts and Sciences, Çanakkale Onsekiz Mart University, Çanakkale, Turkey. His current research interests are in soft set theory, fuzzy set theory, soft matrices, soft analysis, image processing, and decision-making.