International Journal of Applied Engineering Research.
ISSN 0973-4562 Volume 10, Number 11 (2015), pp. 29651-29679

http://www.ripublication.com

A Hetero Dictionary Learning Based Noise Removal in Gray and Color Images

P. Jothibasu

Asst. Professor and Research Scholar,
EIE Department, R.M.K. Engineering College,
Kavaraipettai, Chennai-601206, India.
E-mail: jothibasuin@gmail.com

P. Rangarajan

Professor, EEE Department,
R.M.D. Engineering College,
Kavaraipettai, Chennai-601206, India.

Abstract

Dictionary learning used to remove a noise in images. Image contains edges and details. In order to better learning of details and edges from images, two types of dictionary or Hetero types of dictionary for coarse and details part of images were proposed in this paper. The proposed algorithm results were compared with spatial domain algorithms WMF, PSMF, DBA, NAFSM and sparse domain algorithm DL-INR. For two dictionaries two different soft thresholds used to maximize the dictionary learning.

AMS subject classification:
Keywords: Dictionary Learning, Augmented Lagrangian Multiplier, Fixed value impulse noise, Sparse representation.

1. Introduction

Image carries a lots of information and contains many redundant pixels. During compression these redundant pixels are discarded and in preprocessing like segmentation, de-noising, object identification redundant pixels are very useful. When image is corrupted by noise, these redundant pixels helps to identify corrupted pixels original value. Image may be corrupted during acquisition, processing, transmission and storage. Types

of image noises are gaussian, fixed value impulse noise, random value impulse noise and speckle noise. Noise in the image can be removed using spatial domain algorithms, wavelet domain algorithms and sparse domain algorithms [1]. Spatial domain algorithms are specifically designed fixed value impulse noise and random value impulse noise removal [3] [4] [21]. Spatial domain color image de-noising [7] [8]. Wavelet domain filters were capable of filtering gaussian noise in the gray and color images [5], [6]. Data dependent transforms are capable of transforming image into transform coefficients and basis values [26]. Recently sparse representation and sparse approximation based de-noising algorithms like K-SVD based on $\ell 2$ minimization based de-noising algorithm for gaussian noise [18] [19] [20] and [30]. Compressive sensing based algorithms [16], [33] and [25]. Sparse approximation not only used for de-noising, it has find many post processing applications such as text detection [27], pedestrian detection [28], facial expression detection [26], face recognition [23], image compression [24] and many more applications. Most of the applications requires a optimization algorithms [2]. to learn a dictionary from the noisy image or feature extraction from the image [17] and [20]. other methods using discriminate dictionaries [10], using adaptive kernals [11], augmented lagrangian multiplier based dictionary learning methods were implemented for fixed value impulse noise, random value impulse noise and gaussian impulse noise [12], [13], [14], [32] and [15].

Organization of this paper as follows, section 2 discuss about noise model, equation for corrupted image and previous work related this paper. Section 3 explains block diagram, algorithm, separation of detail and coarse parts of image batches. Section 4 discuss about gray and color image results, dictionary, PSNR and SSIM values. Section 5 discuss about parameters involved in this paper and setting optimum value for those parameters and computational analysis of algorithms. Section 6 concludes the paper.

2. Review of Previous Work

In this section, subsection 2.1 discuss about the noise model and defines the noise pixels. Subsection 2.2 discuss about the $\ell 1$ and $\ell 2$ minimization. Subsection 2.3 discuss about the modification in equation of $\ell 1 - \ell 1$ minimization for efficient implementation and improvements required for dual dictionary learning.

2.1. Noise Model

Let Im be the noise corrupted image of size $W \times W$ and ρ be the probability of noise intensity. Then corrupted image is represented in equation 1.

$$Im(s) = \begin{cases} Im_d(s), & \varepsilon \leq \rho \\ Im_0(s), & \varepsilon > \rho \end{cases} \tag{1}$$

where s denotes two dimensional indices of image Im, $Im_0(s)$ denotes noise free pixels in the image, $Im_d(s)$ stands for noise corrupted image pixels, s varies from 1 to W and ε- is a random number with range of values [0,1]. For Salt and Pepper Noise, $Im_d(s)$ takes

either Im_{min} or Im_{max}. For Random value noise, $Im_d(s)$ takes any value in the range $[Im_{min}, Im_{max}]$, which is independently and identically distributed. $\widehat{Im}$ - recovered de-noised image from corrupted image. Noisy image dimensions were $W \times W$.

2.2. Review of previous work

From noisy image, image batches were extracted by overlapping batches of size $\sqrt{M} \times \sqrt{M}$. Total number of batches from $W \times W$ image is $L = (W - \sqrt{M} + 1)^2$. M is the number of rows in the dictionary (A) and N is the number of columns in the dictionary (A). M and N value determination is explained in subsection 5 of section 5.1. Each batch size, $\sqrt{M} \times \sqrt{M}$ is reshaped into column vector $b_i = [Im(s_{i1}), Im(s_{i2}), \ldots, Im(s_{iM})] \, \varepsilon \, R^M$. These batches are represented as B. $B = [b_1, b_2, .b_i, ..b_L] \, \varepsilon \, R^{M \times L}$
b_i is represented by over complete dictionary A and sparse representation coefficient X. $b_i = A_i X_i, i = 1, 2, 3, \ldots, L$, where $A = [a_1, a_2, \ldots, a_W] \, \varepsilon \, R^{M \times W}$.

$$\min_{A,X} \|X\|_0 + \alpha \, \|B - AX\|_2 \tag{2}$$

Where $X = [X_1, X_2, \ldots, X_L] \, \varepsilon \, R^{W \times L}$ is the sparse co-efficient.

In equation 2 first term represents the sparse representation of X, which counts the few non-zero coefficient in the X, i.e sparse solution for X. $\|X\|_0$ is computationally intensive and $\|B - AX\|_2$ represents root mean square error between B and AX. Each column in A is called atom or basis, which represented as $a_j \, \epsilon \, R^M$, each atom is normalized as $\|a_j\|_2 = 1$. RMSE is susceptible to outliers such as salt and pepper impulse noise, so it is modified as $\ell 1$ norm equation, equation 3 is robust enough for outliers.

$$\min_{A,X} \|X\|_0 + \alpha \, \|B - AX\|_1 \tag{3}$$

Based on learned dictionary values $\widehat{A}$ and sparse coefficients $\widehat{X}$, de-noised image batches $\widehat{b}$ are constructed as in equation 4.

$$\widehat{b} = \widehat{A}\widehat{X} \tag{4}$$

2.3. Modification of $\ell 1 - \ell 1$ minimization

Equation for de-noising algorithms presented in [13], [14], [29] contains redundant similar terms and non-adaptive terms, which were reasons for high computational time and inaccurate results. Equations were modified by deriving the new equation for Augmented Lagrangian Multiplier Y, sparse coefficient X by modifying the iterative shrinkage algorithms. The equation for X and γ are given below.

$$X^k = SHRINK(X^{k-1} + \frac{A^{k-1}Y^K}{\gamma U^{k-1}}, \frac{1}{\gamma U^{k-1}}) \tag{5}$$

$$\gamma = max(eig(A^{k^T} A)) \tag{6}$$

To simplify equation for Y, assume U and τ as in equation 7 and 8.

$$U = -A^{k-1}X^{k-1} + b + \frac{Y^{k-1}}{\mu^{k-1}} \tag{7}$$

$$\tau = \frac{\alpha}{\mu^{k-1}} \tag{8}$$

Equation for Y rewritten as Equation for Y is rewritten as

$$PROJ(U, \tau) = U - SHRINK(U, \tau) \tag{9}$$

To improve the execution speed and fast learning adaptive threshold [9] were introduced then fixed threshold and shrink as in equation 10.

$$SHRINK(U, \tau) = \begin{cases} U - 0.5\tau^2/U, & |U| > \tau \\ 0.5U^3/\tau^2, & |U| \geq \tau \end{cases} \tag{10}$$

Equation 9 and equation for Y is written as in equation 11

$$Y = \mu PROJ(U, \tau) = \mu U - \mu SHRINK(U, \tau) \tag{11}$$

Equation 10 is substituted in 9, to yield

$$= \mu^{k-1}U - \mu^{k-1}SHRINK(U, \tau)$$

$$= \mu^{k-1}U - \begin{cases} \mu^{k-1}U - \mu^0.5\tau.\tau^2, & |U| > \tau \\ 0.5\mu^{k-1}U^3/\tau^2, & |U| \geq \tau \end{cases} \tag{12}$$

Again rearranging equation 12 we obtain 13. Constraints for 13 are not changed. We reduce the large amount calculation required to calculate the Y and also the overhead required for finding Y from using PROJ and SHRINK. For large amount of image batches and number of iterations, it is essential to have optimized equations and algorithms. The straight forward equation for Y is given in equation 13

$$Y = \begin{cases} 0.5\tau^3, & |U| > \tau \\ 0.5\mu^{k-1}U^3/\tau^2, & |U| \geq \tau \end{cases} \tag{13}$$

$\tau = \frac{\alpha}{\mu^{k-1}}$, Substituting the value τ in the equation 13, we obtain 14

$$Y = \begin{cases} 0.5\alpha^3/\mu^{k-1}U, & |U| > \tau \\ 0.5\mu^{k-1}U^3/\alpha^2, & |U| \geq \tau \end{cases} \tag{14}$$

$$A^k = A^{k-1} + \beta Y^k(X^{k^T}) \tag{15}$$

3. Block Diagram and Algorithm

In this section, subsection 3.1 discuss about block diagram and implementation of algorithm in detail. Subsection 3.2 explains the algorithm and modification to improve the quality of the de-noising results. In subsection 3.3, image batch separation and separate dictionary for detail and coarse parts of image is discussed. Subsection 3.4 discusses about initial dictionary values for two different spatial parts of image.

3.1. Block Diagram

In proposed algorithm, entire image cannot be processed as it is, memory requirement is very large to process a entire image. Image pixels were converted into batches. Batch size depends on the number of rows or number of elements in the basis or atom of the dictionary. Before image pixels were converted into batches, noisy pixels in the image identified using impulse noise detector. Impulse pixels were identified which will be used to calculate accurate DC value of image batches. Block diagram 1 shown in the figure contains many blocks. The first block converts 256×256 image in to $\sqrt{M} \times \sqrt{M}$ image batches. DC value of each batch is calculated using noise free pixels in the image batches. The DC value is subtracted from each image batch, which will avoid scaling of dictionary values. Scaling a dictionary is difficult and to find a suitable scale requires a significant amount of computation time.

After subtracting a DC value, batches were separated into detail batches and coarse batches depends on the pixel values in the batch. If image batches from iso-pixel region, pixels in the batch are equal. If image batch contains edges and lines, pixel value in the batch varies. Based upon the pixel value in the batches, it is divided into two groups. If pixel value in the batch is equal, it is considered as a coarse batch and pixel values are differed by the threshold value T, it is called detail batch. To learn a dictionary from these two batches, two separate initial dictionary values were used. After de-noising using efficient and modified $\ell 1 - \ell 1$ minimization based algorithm, two different dictionaries learned and de-noised coarse and detail batches were combined to form a de-noised image batches. From de-noised image batches, de-noised image is reconstructed.

3.2. Algorithm

The proposed algorithm contains two parts, one for detail batches and another for coarse batches. For detail batch initial dictionary is Haar transform coefficients and for coarse batch initial dictionary is DCT transform coefficients. Random values are not used as a initial values of dictionary for coarse batches. Coarse batches contains very less information or edges. For two different dictionaries and two different learning methods are required. For detail batch dictionary, learning should be fast and accurate, which is achieved by using adaptive soft threshold function. Parameter values for two different learning are different.

Algorithm 1 is for detail part of image batches and algorithm 2 is for coarse part of image batches.

Algorithm 1: Modified $\ell 1 - \ell 1$ Minimization using Augmented Lagrangian Multiplier for details part of image batches

$\gamma^0 = max(eig(A^{0^T} A^0))$ - Initial value
b - DC values of subtracted image batches
Initialize $X_0 = 0, Y_0 = 0, \mu_0 = 0.006$
A_0= DCT Dictionary or Walsh or Haar coefficients
WHILE(Stoping criterion is not satisfied, continue looping)
U=-$A^{k-1}X^{k-1}$+b+$\frac{Y^{k-1}}{\mu^{k-1}}$, τ=$\frac{\alpha}{\mu^{k-1}}$

$$Y^k = \begin{cases} 0.5\tau^3, & |U| > \tau \\ 0.5\mu^{k-1}U^3/\tau^2, & |U| \geq \tau \end{cases}$$

$X^k = SHRINK(X^{k-1} + \frac{A^{k-1}Y^K}{\gamma U^{k-1}}, \frac{1}{\gamma U^{k-1}})$ $A^k = A^{k-1} + \beta Y^k(X^{k^T})$
$A^k = A^k.diag(\|a_0^k\|_2^{-1}, \|a_1^k\|_2^{-1}, ... \|a_N^k\|_2^{-1})$ γ =*max(eig*$(A^{k^T} A^k)$*)*
$\mu^k = 1.01\mu^{k-1}$
ENDWHILE

Algorithm 2: Modified $\ell 1 - \ell 1$ Minimization algorithm for coarse part of image batches

$\gamma^0 = max(eig(A^{0^T} A^0))$ - Initial value
b - DC values of subtracted image batches
Initialize $X_0 = 0, Y_0 = 0, \mu_0 = 0.006$
A_0= DCT Dictionary values
WHILE(Stoping criterion is not satisfied, continue looping)
U=-$A^{k-1}X^{k-1}$+b+$\frac{Y^{k-1}}{\mu^{k-1}}$, τ=$\frac{\alpha}{\mu^{k-1}}$

$$Y^k = \begin{cases} \alpha & U > \tau \\ \mu^{k-1}U & \tau \geq U \geq -\tau \\ -\alpha & U < -\tau \end{cases}$$

$X^k = SHRINK(X^{k-1} + \frac{A^{k-1}Y^K}{\gamma U^{k-1}}, \frac{1}{\gamma U^{k-1}})$ $A^k = A^{k-1} + \beta Y^k(X^{k^T})$
$A^k = A^k.diag(\|a_0^k\|_2^{-1}, \|a_1^k\|_2^{-1}, ... \|a_N^k\|_2^{-1})$ γ =*max(eig*$(A^{k^T} A^k)$*)*
$\mu^k = 1.01\mu^{k-1}$
ENDWHILE

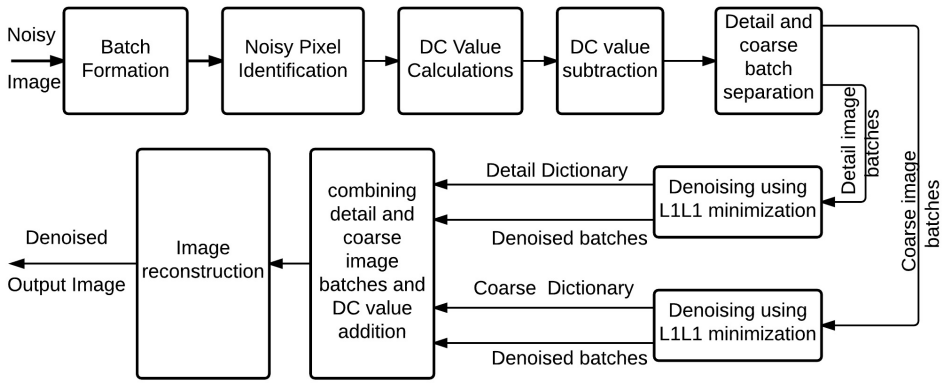

Figure 1: Block diagram for the proposed de-noising method for salt and pepper and random value impulse noises.

3.3. Image batch separation

In order to separate $L = (W - \sqrt{M} + 1)^2$ batches into detail and coarse batches, difference between minimum and maximum pixel values in the image batch is calculated. If difference is greater than the threshold T, batch is classified as detail batch, otherwise it is classified as coarse batch.Number of pixels in each batch is M. Among M pixels $b_{i,max}$ is maximum value of the pixel and $b_{i,min}$ is minimum value of the pixel. Difference value between $b_{i,max}$ and $b_{i,min}$ is compared with threshold T for batch separation. The comparison given in equation 16. This process is repeated for L patches.

$$b_{i,max} - b_{i,min} < T \tag{16}$$

3.4. Initial Dictionary Values

In this paper, two different initial dictionary values considered for detail and coarse batches of images. Detail batches contains edges and lines, for those batches Walsh, DCT, Random, Haar and Hadamard transform coefficients were considered as a initial values. For coarse batch, DCT and slant transforms coefficients were considered as a initial values.

4. Expeirments and Results

The proposed algorithm based on hetero dictionary learning and efficient $\ell 1 - \ell 1$ based de-noising algorithm implemented in MATLAB for gray and color images. The gray standard images used to demonstrate the algorithm are Lena, Airplane, and House. The color images are Barbara, Baboon and Peppers. The proposed algorithm contains many parameters, their values are assigned based on experiments. The detailed explanation of those experiments were given section 5. Size of the images used for comparison is

Table 1: PSNR Values for Lena image corrupted by fixed value impulse noise.

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	29.37	26.33	24.81	23.16	21.43	19.75	15.96	12.03
PSMF	30.03	28.00	26.21	25.43	23.85	22.53	19.33	13.53
DBA	32.96	31.18	29.23	27.39	26.08	24.49	22.55	20.51
NAFSM	37.72	34.82	32.17	30.13	28.63	27.39	25.77	24.22
DL-INR	29.83	28.68	27.36	25.56	24.26	22.90	21.04	17.84
HD-NR	39.03	36.25	33.68	31.32	29.93	28.57	26.57	25.41

256×256. Image metrics used for comparison of algorithms performance were peak-signal to noise ratio (PSNR), mean square error (MSE), and structural similarity index (SSIM) [22]. The equation for metrics are given in 17, 18, and 19.

$$\text{PSNR} = 10 \log_{10} \frac{(2^b - 1)^2}{MSE} \tag{17}$$

$$\text{MSE} = \frac{\sum_{m=0}^{N-1} \left(\sum_{n=0}^{M-1} (Im(m,n) - Im_0(m,n))^2 \right)}{N^2} \tag{18}$$

$$\text{SSIM} = \frac{(2\mu_0\mu_{I_0} + C_1)(2\sigma_{I_0,0} + C_2)}{(\mu_0^2 + \mu_{I_0}^2 + C_1)(\sigma_0^2 + \sigma_{I_0}^2 + C_2)} \tag{19}$$

μ_0-Mean intensities of original image, μ_{I_0} -Mean Intensities of restored image, $\sigma^2_{I_0,0}$-Co-variance of original and restored image, σ_0 - Standard deviation of the original image, σ_{I_0} - Standard deviation of the restored image image.

4.1. Grag image Results

The proposed algorithm tested for the following three standard gray images Lena, Airplane and House. Fig. 2 shows the de-noising results of the proposed algorithm and other algorithms. Fig. 2 (a) is original Lena gray image. Fig. 2 (b) is the 60 percentage of the pixels corrupted by the fixed value impulse noise. Fig. 2 (c), (d), (e), (f), (g) and (h) are the de-noising results of WMF, PSMF, DBA, NAFSM, DL-INR and the proposed algorithm (HD-NR).

Fig. 2 (c) is the de-noising result of WMF, which contains the patches of block and white pixels. For 60 percentage of pixels corrupted by the fixed value impulse noise, window size is 7 and due to large window size, patches occurs in the image. Fig. 2 (d) shows the de-noising results for PSMF algorithm, in the image edges were smoothed and blurred. Fig.2 (e) shows the results of DBA, in DBA when noise pixel is not available in the given filtering window, which creates a streaks in the image, which is visible in the de-noised image. Fig. 2 (f) shows the result of NAFSM algorithm, results of NAFSM is

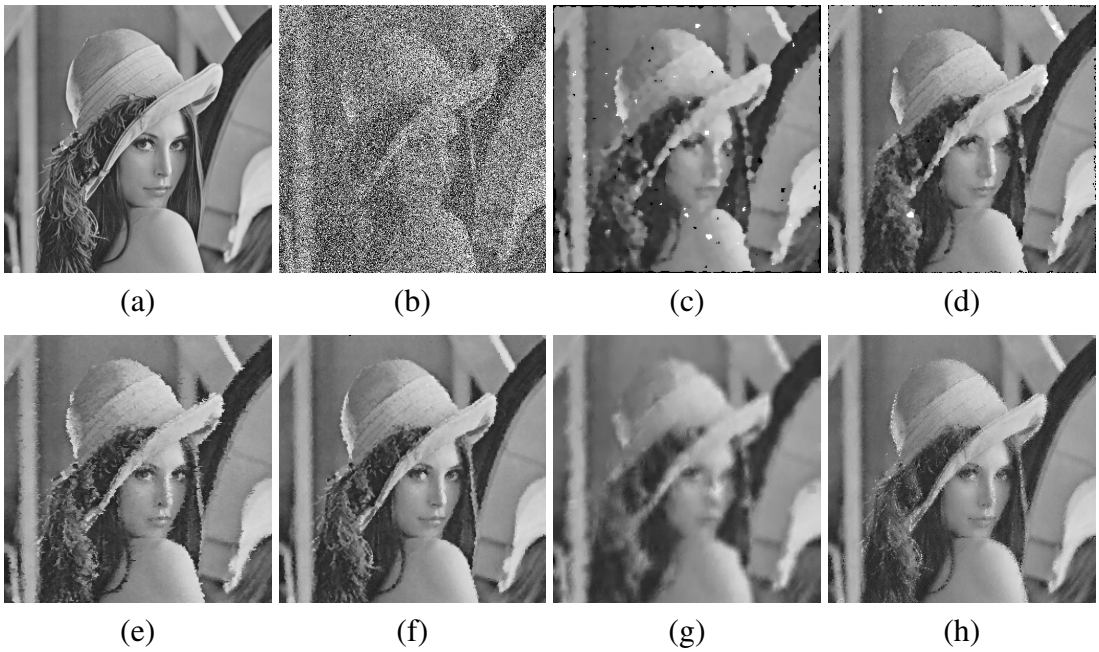

Figure 2: Denoising results for 60 % of pixels corrupted by the fixed value impulse noise (a) Original Lena Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm.

Table 2: SSIM values for Lena image

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	0.8921	0.8229	0.8086	0.7607	0.7291	0.6703	0.4639	0.2094
PSMF	0.9173	0.8828	0.8779	0.8725	0.8448	0.7983	0.6950	0.4088
DBA	0.9448	0.9329	0.9103	0.8772	0.8329	0.7784	0.6992	0.5909
NAFSM	0.9859	0.9695	0.9496	0.9245	0.8963	0.8655	0.8145	0.7525
DL-INR	0.9004	0.8812	0.8410	0.7818	0.7314	0.6669	0.5763	0.4222
HD-NR	0.9874	0.9743	0.9567	0.9324	0.9139	0.8807	0.8378	0.7662

better than other spatial domain algorithm. NAFSM algorithm's PSNR value shown in Table 1 is less than the proposed algorithm. Sparse approximation based DL-INR results shown in Fig. 2 is highly smoothed and blurred. The proposed algorithm's results are better than other algorithms in terms of PSNR and detail preserving.

Fig. 3 shows the de-noising results for the standard Airplane gray image. Fig. 3 (a) is the original Airplane gray image. Fig. 3 (b) is the 70 percentage of the pixels corrupted gray image. Fig. 3 (c) is the de-noising results of WMF, which contains the black and white pixel patches. White and black batches increases with noise density. Fig. 3 (d) is reult of PSMF algorithm, part of image is corrupted beyond recognition. Fig. 3 (e) and (f) are the result of DBA and NAFSM algorithms. Fuselage letters and Tails are not

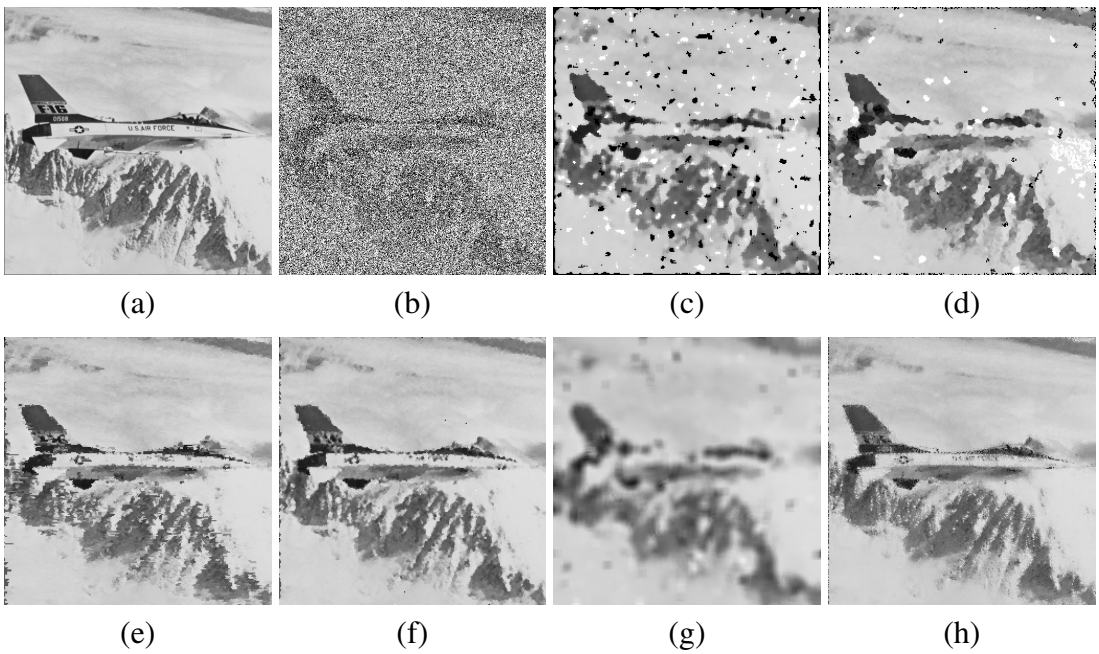

Figure 3: Denoising results for 70 % of pixels corrupted by the fixed value impulse noise (a) Original Airplane Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm.

Table 3: PSNR values for airplane image corrupted by fixed value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	27.54	24.27	22.79	20.71	18.94	17.49	14.53	10.77
PSMF	26.05	23.89	21.93	21.36	20.74	19.64	17.92	14.01
DBA	31.27	28.96	26.87	25.57	24.30	22.89	21.33	20.09
NAFSM	35.96	32.26	30.15	28.13	26.60	25.31	23.84	22.34
DL-INR	27.20	26.30	24.81	22.96	21.69	20.80	19.55	16.45
HD-NR	36.87	33.96	31.58	29.16	27.44	26.42	24.39	23.73

visible and edges were blurred and box effect appears. Fig. 3 (g) is de-noised image by DL-INR, which has very less details and edges of original image. Fig. 3 (h) is de-noised result of the proposed algorithm, which preserves the details and edges, letters than other algorithms.

Fig. 4 shows the de-noising results for the standard House gray image. Fig. 4 (a) is the original House gray image. Fig. 4 (b) is the 70 percentage of the pixels corrupted gray image. Fig. 4 (c) is the de-noising results of WMF, which contains the black and white pixel patches. White and black batches increases with noise density. Fig. 4 (d) due to increased noise density, black and white patches visible in the PSMF algorithm results. Fig. 4 (e) is de-noising result of DBA, in this image streaks are very much visible. Fig. 4

Table 4: SSIM values for airplane image

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	0.9073	0.8317	0.8176	0.7731	0.7296	0.6709	0.4732	0.2008
PSMF	0.7313	0.6627	0.6470	0.6926	0.7363	0.7284	0.6698	0.4086
DBA	0.9545	0.9392	0.9162	0.8907	0.8592	0.8112	0.7555	0.6895
NAFSM	0.9875	0.9713	0.9553	0.9329	0.9077	0.8783	0.8337	0.7764
DL-INR	0.8945	0.8806	0.8414	0.7853	0.7286	0.6722	0.5979	0.4211
HD-NR	0.9947	0.9858	0.9768	0.9416	0.9105	0.8980	0.8475	0.8141

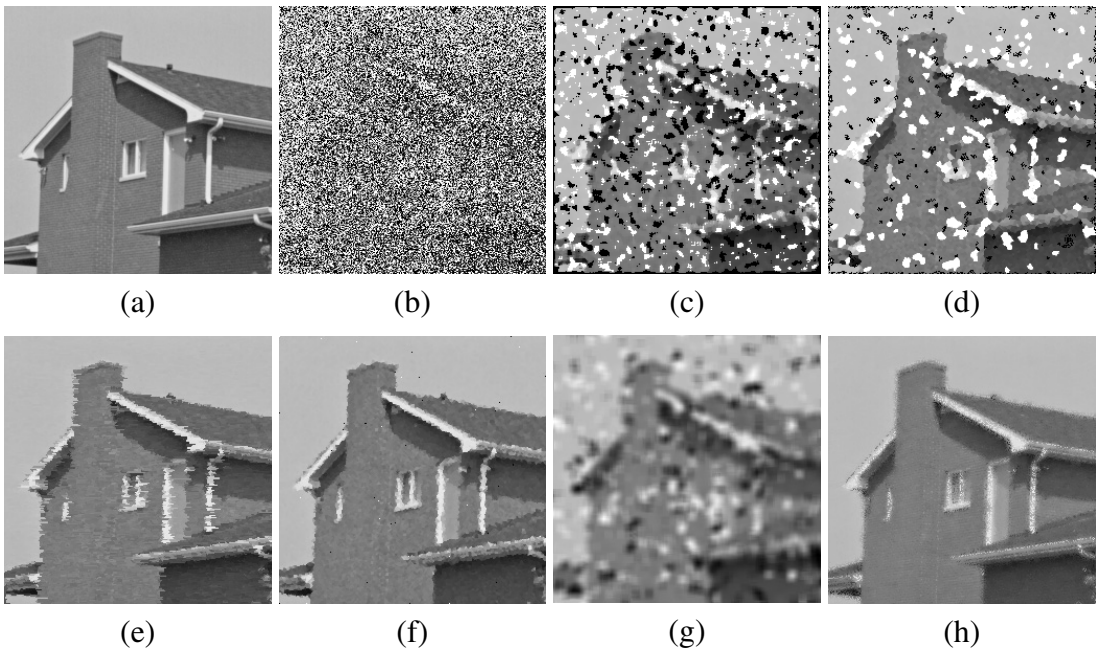

Figure 4: Denoising results for 80 % of pixels corrupted by the fixed value impulse noise (a) Original House Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm

(f) is NAFSM result, which is better than other spatial domain algorithms, but PSNR value is less than the proposed algorithm. Fig. 4 (f) is result of DL-INR algorithm, which contains many batches due to inaccurate DC value calculation. Fig. 4 (g) is result of the proposed algorithm, which gives better edge and details preservation and PSNR value.

Table 1, 3 and 5 are PSNR comparison of the proposed algorithm with WMF, PSMF, DBA, NAFSM and DL-INR algorithms. The proposed algorithm's PSNR value is better than the other algorithm for fixed value impulse noise corrupted image. Table 2, 4 and 6 is the structural similarity index of algorithms. The proposed algorithm performs better by preserving the details and details of the image, the proposed algorithm has better

Table 5: PSNR values for house image corrupted by fixed value impulse noise

Noise % / Algorithm	10	20	30	40	50	60	70	80
WMF	32.89	28.23	25.59	23.25	21.41	19.08	16.03	11.85
PSMF	35.74	31.45	28.96	26.54	25.06	23.12	19.82	14.28
DBA	36.88	35.06	32.54	30.67	29.05	27.62	26.14	23.94
NAFSM	42.16	38.30	35.35	33.54	31.70	30.10	28.90	27.10
DL-INR	34.92	33.53	30.98	28.69	26.59	25.04	22.79	19.00
HD-NR	43.78	39.74	36.18	34.83	32.40	31.73	29.66	28.78

Table 6: SSIM values for house image

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	0.9158	0.8611	0.8482	0.7995	0.7912	0.7082	0.5035	0.2008
PSMF	0.9719	0.9517	0.9339	0.9108	0.8813	0.8332	0.7259	0.4336
DBA	0.9473	0.9410	0.9214	0.8986	0.8690	0.8377	0.7958	0.7314
NAFSM	0.9869	0.9716	0.9548	0.9342	0.9095	0.8818	0.8496	0.8006
DL-INR	0.9497	0.9374	0.8992	0.8497	0.8041	0.7519	0.6728	0.4971
HD-NR	0.9925	0.9871	0.9609	0.9467	0.9134	0.8921	0.8651	0.8254

SSIM value than other algorithms. Table 7, 8 and 9 are the PSNR value comparison table of algorithm for random value corrupted images. The proposed algorithm PSNR values are higher than other algorithms and PSNR value difference is higher than PSNR difference for fixed value impulse noise.

Fig. 5 and Fig. 6 are the dictionaries for gray image corrupted by the fixed value impulse noise. Fig. 5 is dictionaries during different iteration of the algorithms for detail part of image batches. For detail part image dictionary changes are visible during algorithm iterations. Changes in dictionary values is minimal for coarse part of image batches, which not shown in figure. Fig. 6 is dictionary for various noise percentage

Table 7: PSNR values for lena image corrupted by random value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	30.19	26.90	26.12	24.66	22.65	21.15	19.06	16.96
PSMF	28.61	26.22	24.72	23.63	22.52	21.34	19.09	16.83
DBA	27.20	22.35	19.23	17.23	15.50	14.21	13.19	12.28
NAFSM	19.36	16.19	14.39	13.27	12.25	11.52	10.83	10.23
DL-INR	30.08	28.88	27.10	25.15	23.36	21.60	19.63	17.76
HD-NR	37.52	33.45	30.73	28.49	26.65	24.85	23.37	22.65

Table 8: PSNR values for airplane image corrupted by random value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	28.45	24.82	23.81	22.21	19.99	18.25	15.73	13.77
PSMF	26.78	24.68	23.27	21.40	20.38	18.42	15.89	13.63
DBA	25.20	20.41	17.50	15.44	13.81	12.58	11.45	10.54
NAFSM	17.93	15.05	13.26	11.92	11.01	10.21	9.59	9.01
DL-INR	27.21	26.39	25.35	23.44	21.65	19.63	16.90	14.49
HD-NR	36.03	34.55	32.32	28.20	20.58	18.53	16.00	15.16

Table 9: PSNR values for house image corrupted by random value impulse noise

Noise % / Algorithm	10	20	30	40	50	60	70	80
WMF	33.78	30.02	28.05	25.80	23.59	21.60	19.50	17.34
PSMF	29.77	27.10	25.39	24.52	23.42	21.47	19.49	17.01
DBA	27.72	22.65	19.49	17.26	15.65	14.30	13.17	12.26
NAFSM	19.34	16.20	14.46	13.30	12.35	11.50	10.90	10.30
DL-INR	34.55	33.69	30.12	28.13	25.30	23.21	20.91	18.42
HD-NR	42.42	38.43	34.26	31.44	28.36	26.47	24.88	24.23

of pixels. For lower noise percentage, final dictionary values are different from initial dictionary values, for higher percentage difference is less.

4.2. Extension to Color Images

RGB image is converted to YCbCr form. Because obvious advantages associated with converting RGB image in to luminance and chrominance component. Luminance component carries more information than chrominance component. Luminance channel is denoted by Y and Chrominance channels are Cb and Cr as in JPEG image compression standard. [34]. Conversion to YCbCr and viceversa given in matrix form in equation 20 and 21. Conversion from RGB to YCbCr decorrelates the RGB channel and signal to noise ratio of luminance component is higher than chrominance component. The proposed algorithm implemented for luminance and chrominance component separately. The PSNR calculation for color image given in equation 22.

$$A_{YCbCr} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.17 & -0.33 & 0.50 \\ 0.50 & -0.42 & -0.08 \end{bmatrix} \tag{20}$$

$$B_{RGB} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/\sqrt{6} & 0 & -1/\sqrt{6} \\ 1/3\sqrt{2} & -\sqrt{2}/3 & 1/3\sqrt{2} \end{bmatrix} \tag{21}$$

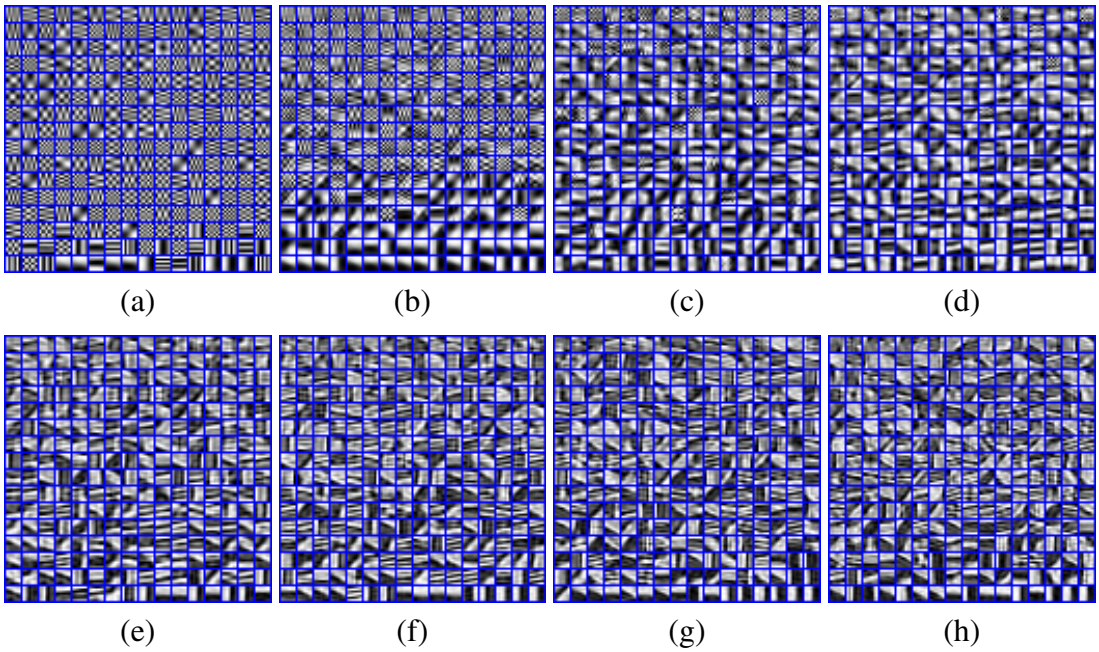

Figure 5: Gray image dictionary matrix during algorithm iterations (a) K=1 (b) K=3 (c) K=6 (d) K=9 (e) K=13 (f) K=16 (g) K=19 (h) K=22

$$\text{PSNR} = 10\log_{10}\frac{(2^b-1)^2}{3*MSE} \tag{22}$$

The proposed algorithm and other algorithms are implemented in MATLAB for color image processing. Three standard color images were used for comparison are Barbara, Baboon and Peppers. Fig. 7 is de-noising result of the proposed algorithm and other algorithms for standard Barbara color image of 10 percentage of pixels corrupted by random value impulse noise. Fig. 7 (a) is original image, Fig. 7 (b) is 10 percentage of pixels corrupted by the random value impulse noise. Fig. 7 (c), (d), (e), (f), (g) and (h) are the de-noising output of WMF, PSMF, DBA, NAFSM, DL-INR and the proposed algorithm. Fig. 7 (c) and (d) contains few unfiltered noisy pixels and Fig. 7 (e) and (f) has many unfiltered noisy pixels, which is visible throughout the images. Spatial domain algorithms results were shown in Fig. 7 (g) and (h). Both algorithm results are similar but the proposed algorithm has better PSNR value than DL-INR algorithm.

Fig. 8 is de-noising result of standard baboon color image. Fig. 8 (a) is original image, Fig. 8 (b) is 20 percentage of pixels corrupted by random value impulse noise. Fig. 8 (c), (d), (e), (f),(g) and (h) are the de-noising result of WMF, PSMF, DBA, NAFSM, DL-INR and the proposed algorithm. Fig. 8 (c), (d), (e) and (f) are the spatial domain algorithm results, Fig. 8 (g) and (h) are sparse domain algorithm results. Algorithm DBA, NAFSM leaves few unfiltered pixels in the image. Output of WMF is smoothed and blurred. PSMF and DL-INR has better results than other algorithm, their PSNR value is less than the proposed algorithm.

Fig. 9 is de-noising result of standard Peppers color image. Fig. 9 (b) is 30 percentage

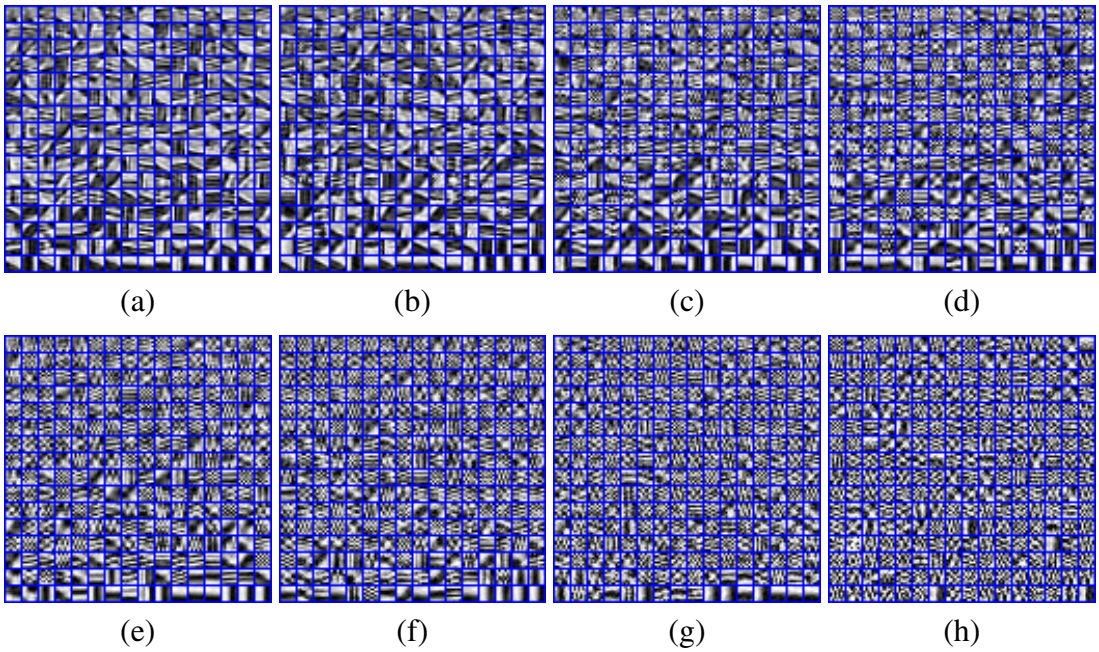

Figure 6: Gray image dictionary matrix for different noise densities (a) NP=10 (b) NP=20 (c) NP=30 (d) NP=40 (e) NP=50 (f) NP=60 (g) NP=70 (h) NP=80

pixel corrupted by the random value impulse noise. Fig. 9 (c), (d), (e), (f),(g) and (h) are the de-noising result of WMF, PSMF, DBA, NAFSM, DL-INR and the proposed algorithm. Output of DBA shows streaking effect for image beyond 30 percentage pixels corrupted by the noise. Output of NAFSM and PSMF has many unfiltered noisy pixels. DL-INR output is smoothed and its PSNR value is less than the proposed algorithm. WMF has better output, it has less PNSR value than HD-NR.

PSNR values for the algorithm is tabulated in Table 10, 11, 12, 13, 14 and 15. Table.10, 11, 12 is tabulation for PSNR value and image is corrupted by the random value impulse noise. Table 14 and 15 is tabulation for PSNR value of fixed value impulse noise corrupted color images. From Table 10, 11 and 12 the PSNR value of the proposed algorithm is significantly higher than other algorithms for all noise percentages. For fixed value impulse noise corrupted images, PSNR value higher by minimum of 0.5db than other algorithms as shown in Table 14 and 15.

Fig. 10 and 11 are color dictionary learned by the proposed algorithm. Fig. 10 is dictionary learned during different iterations of the proposed algorithm. For iterations K=1, learning from corrupted image is minimal, for K=16 maximum information is learned from corrupted image and after 16th iterations dictionary value not changes significantly. Fig. 11 is dictionary learned for various noise percentage of pixels corrupted by the fixed or random value impulse noises. For low noise percentage pixels corrupted by the fixed value impulse noise, Fig. 11 (a) shows that dictionary learning is high

Table 10: PSNR values for barbara color image corrupted by random value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	26.58	24.79	24.18	23.09	21.79	20.05	17.98	15.94
PSMF	27.27	25.06	23.74	22.85	21.75	20.11	17.84	15.57
DBA	25.48	21.17	18.52	16.53	14.94	13.64	12.53	11.58
NAFSM	18.92	15.84	14.08	12.88	11.90	11.06	10.44	9.810
DL-INR	30.47	28.16	25.39	22.74	19.87	17.23	15.50	14.67
HD-NR	35.93	32.29	29.52	27.25	25.24	23.54	22.18	21.23

Table 11: PSNR values for baboon color image corrupted by random value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	21.30	20.00	19.71	19.27	18.66	17.76	16.40	15.08
PSMF	21.69	20.67	19.89	19.27	18.66	17.58	16.12	14.77
DBA	23.70	19.78	17.10	15.24	13.75	12.49	11.44	10.59
NAFSM	18.84	15.82	14.00	12.77	11.79	11.05	10.37	9.770
DL-INR	24.69	22.41	20.75	19.40	17.76	16.20	15.02	14.36
HD-NR	27.87	25.39	23.70	22.52	21.64	20.66	19.88	19.17

Table 12: PSNR values for peppers color image corrupted by random value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	30.02	26.95	25.61	23.54	21.44	18.92	16.34	14.15
PSMF	27.17	25.00	23.58	22.31	20.99	19.02	16.26	13.89
DBA	23.70	19.78	17.10	15.24	13.75	12.49	11.44	10.59
NAFSM	17.13	14.44	12.89	11.72	10.89	10.12	9.540	8.990
DL-INR	29.72	27.20	24.52	21.99	18.76	15.78	13.92	12.94
HD-NR	34.70	31.12	28.7	26.22	24.18	22.48	20.9	19.9

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Denoising results for 10 % of pixels corrupted by the random value impulse noise (a) Original Barbara Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm.

and noisy pixels percentage increases learning from corrupted image decreases. For 80 percentage of pixels corrupted image the learning is insignificant. Learning decreases when noise increases more than 30 percentage of image pixels.

5. Discussion

This algorithm has many parameters, like batch size (N), number of atoms in dictionary (M), dictionary learning rate (β), fidelity term (α), coefficient learning rate (μ), maximum eigen value (γ), and number of iterations (K). To obtain the best PSNR value from the algorithm, setting the optimum value for the above parameters is important. Subsection 5.1 discuss about the parameters and its optimum values. Subsection 5.2 discusses about the computation time required for the algorithm and compares the computation time of other algorithm.

5.1. Parameter selection

The foremost parameter in this paper is dictionary size. The dictionary size is number of elements in a basis or atom is M and number of atoms in a dictionary is N. To find a best M and N value, M and N values are varied from 36 to 196 and 36 to 256 respectively. From Fig. 12, for M=81 the proposed algorithm has highest PSNR value of 43.5 and for other M values PSNR value is less than the highest value. From Fig. 13, for $N = 144$ the

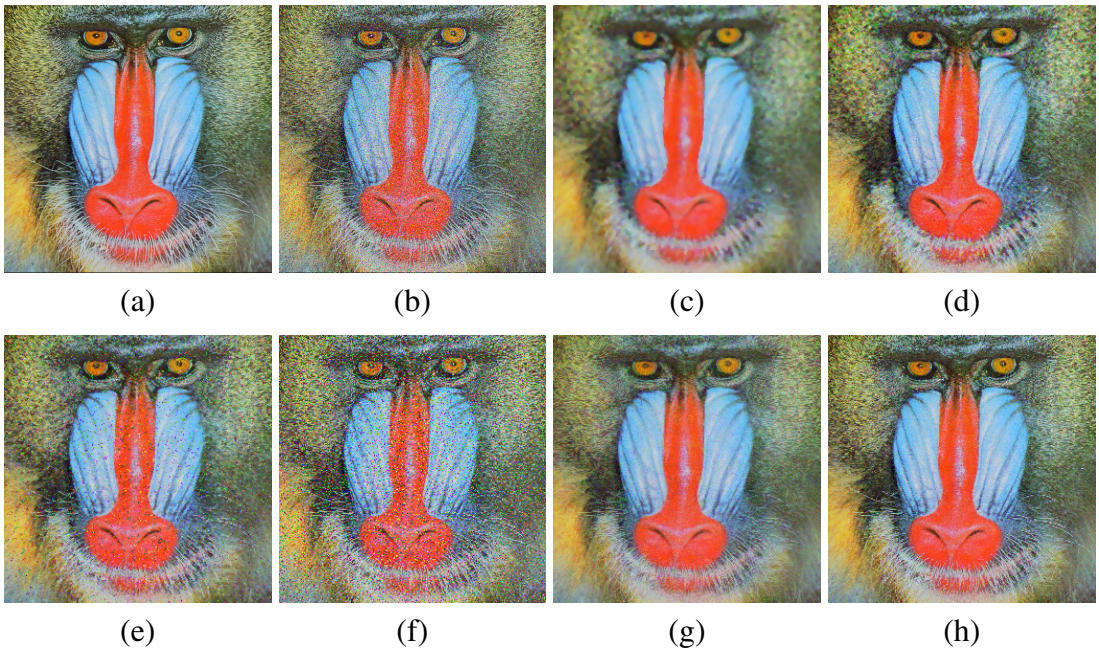

Figure 8: Denoising results for 20 % of pixels corrupted by the random value impulse noise (a) Original Baboon Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm.

proposed algorithm has highest PSNR value of 43.52 and for other values of N, PSNR value is less than the highest value. For this algorithm M and N values are fixed as 81 and 144.

The next parameter to be tuned is number of iterations(K). The number of iterations required for de-noising detail batch and coarse batch of the noisy image is determined by plotting number of iterations (K) against the PSNR value for each iteration. Fig. 14 shows the PSNR values for the various number of iterations. Number of iterations varied from 1 to 50. For K value 21 PSNR value is maximum than any other value. The number of iterations for the algorithm is fixed as 21.

μ value is 0.007 after calculating PSNR value from Fig. 15. For μ value 0.007, PSNR value is high for all images. μ value is calculated for House, Girl, Airplane, Barbara, Boat and Baboon gray images.

β is a learning parameter for dictionary A. Value of β determines the rate of learning or updation rate for the dictionary. To improve PSNR value using β and fast learning of dictionary from the noisy image. Fig. 16 is graph between β and PSNR value. β varied from 0.001 to 0.010. For β value 0.004 image has highest PSNR value than any other β value. so β value is fixed at 0.004.

Image is divided into detail part and coarse part. Detail part of image contains high frequency pixels or many edges. This part of image batches are called detail part image batches. Coarse part of image batches are called as coarse part image batches. To separate a coarse part image batches from detail part image batches, threshold value T

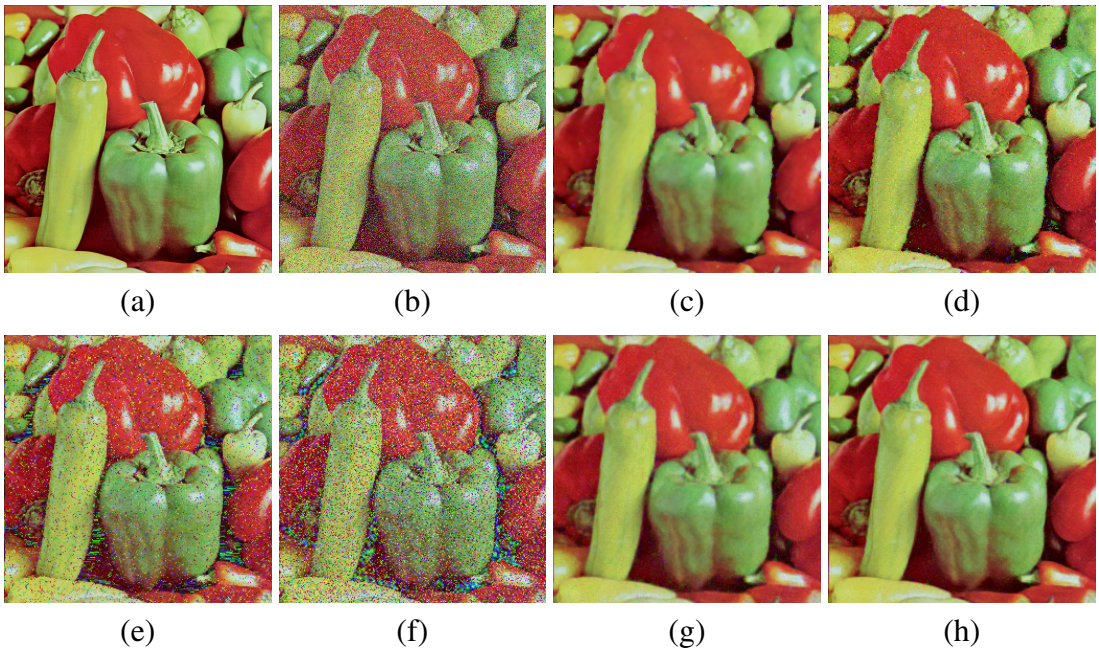

Figure 9: Denoising results for 30 % of pixels corrupted by the random value impulse noise (a) Original Lena Image (b) fixed value impulse noise corrupted image (c) WMF (d) PSMF (e) DBA (f) NAFSM (g) DL-INR (h) The proposed algorithm.

is used. If noise free pixels minimum and maximum value difference is less than the threshold value T, then the batch is classified as coarse part image batches, if difference is greater than the threshold T, it is called detail part image batches. The threshold value is determined after plotting the PSNR value against the threshold value T as shown in Fig. 17. The threshold value is fixed at $T = 45$ from figure.

α value used to balance between number of non-zero sparse coefficient and accurate sparse representation of the image as in equation 2. In this algorithm two separate dictionaries were used for detail part batches of the image and coarse part batches of the image. Table 16 shows the α value for the detail part batches of gray image. Table 17 shows the α value for the coarse part batches of the image. These α values are determined by iterating the algorithm for various values of α value by steps of 0.001.

Maximum eigen value is represented by greek alphabet γ and it is depends on initial dictionary value assigned. Initial dictionary values considered here are DCT, DST, Walsh, Haar, Hadamard and random values. For detail parts of batches, may have any one of the initial values. For coarse parts of batches can have DCT or Haar transform coefficient as a initial values. Fig. 18 and 19 shows γ value for various iterations of algorithm. When DCT matrix as a initial matrix, Fig. 19 is a plot between number of iterations and γ value for coarse and details parts of image batches. Details part image batches has a high variation in the γ value than coarse part of image batches. Coarse parts of image batches shows less variation in the γ value. Fig. 18 is plot for γ when initial dictionary values for both coarse and details parts of image batches is random value. From plots details

Table 13: PSNR values for barbara color image corrupted by fixed value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	26.32	24.86	24.15	22.82	22.63	20.28	16.62	11.79
PSMF	28.66	26.50	25.23	24.06	23.11	21.46	18.33	12.93
DBA	29.33	28.12	26.71	25.43	24.12	22.87	21.45	19.81
NAFSM	34.39	31.17	29.09	27.47	26.14	24.93	23.87	22.52
DL-INR	31.20	28.40	25.95	23.75	21.40	18.61	15.70	14.61
HD-NR	34.92	31.87	29.72	27.83	26.96	25.73	24.65	23.12

Table 14: PSNR values for baboon color image corrupted by fixed value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	21.15	19.88	19.56	18.95	18.68	17.49	14.86	11.19
PSMF	21.86	20.68	19.96	19.38	18.75	17.91	16.12	12.30
DBA	23.41	22.79	21.91	20.91	19.88	18.98	18.05	16.93
NAFSM	29.22	25.95	24.06	22.57	21.49	20.33	19.34	18.39
DL-INR	24.41	22.28	20.88	19.77	18.62	17.14	15.04	14.18
HD-NR	29.98	26.80	24.62	23.39	22.10	20.88	20.24	19.58

parts of image dictionary varies for larger range than coarse parts of image dictionary.

X represents the sparse coefficient of the image. Table 18 shows the average number of non-zero coefficients in the sparse coefficient matrix. The number of non-zero coefficient in a column or atom of the sparse matrix is 45.55 for 10 percentage of pixels are corrupted by impulse noise. Almost 30 percentage of the coefficients are non-zero and if 80 percentage of pixels are corrupted by the impulse noise, number of non-zero coefficient is 0.20. It shows that ability to learn from noisy image is decreases with increase in noise pixel percentage in the image. For optimal learning of dictionaries at least 50 percentage of pixels to be noise free.

Table 15: PSNR values for peppers color image corrupted by fixed value impulse noise

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	29.41	27.08	25.72	23.75	22.87	20.46	16.30	11.58
PSMF	27.33	25.06	23.10	22.17	21.41	20.61	17.66	12.61
DBA	27.10	26.77	25.04	23.87	23.28	22.03	20.47	18.60
NAFSM	27.89	27.26	26.46	25.83	24.99	24.17	23.25	22.11
DL-INR	29.71	27.64	25.36	23.31	20.81	17.93	14.14	12.88
HD-NR	31.67	29.49	27.46	26.71	25.86	24.87	23.93	22.79

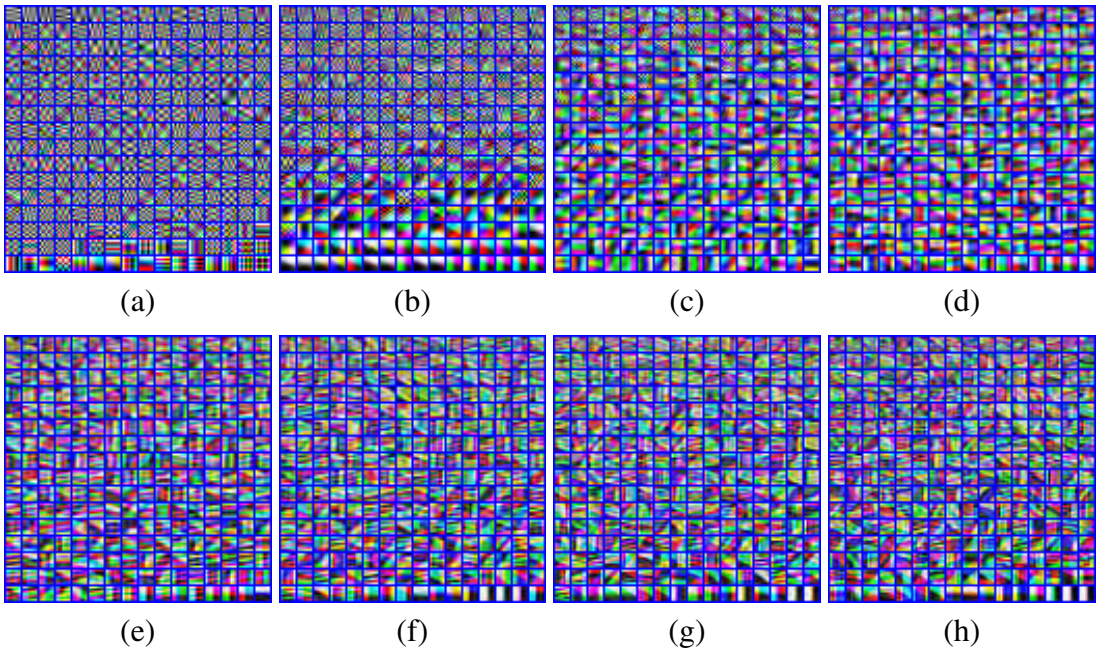

Figure 10: Color Dictionary matrix during algorithm iterations (a) K=1 (b) K=3 (c) K=6 (d) K=9 (e) K=13 (f) K=16 (g) K=19 (h) K=22.

Table 16: α value for detail part batches of gray images

Noise %	10	20	30	40	50	60	70	80
α	0.8456	0.7569	0.6995	0.6345	0.5423	0.4683	0.3584	0.2977

Table 17: α value for coarse gray image batches

Noise %	10	20	30	40	50	60	70	80
α	0.7869	0.7369	0.6795	0.6045	0.5123	0.4283	0.3284	0.2877

Table 18: Sparse coefficient for gray image

Noise %/ Image	10	20	30	40	50	60	70	80
Lena	45.55	42.34	39.52	33.76	24.60	13.23	2.550	0.200
Airplane	45.55	42.34	39.52	33.76	24.60	13.23	2.550	0.200
House	34.95	30.07	26.63	22.26	15.34	8.280	1.360	0.170
Barbara	43.40	39.62	36.51	30.86	20.81	11.73	1.870	0.210
Baboon	53.45	44.58	38.69	30.39	17.54	9.000	1.420	0.210
Peppers	47.55	43.81	40.42	33.49	24.77	14.07	2.470	0.210

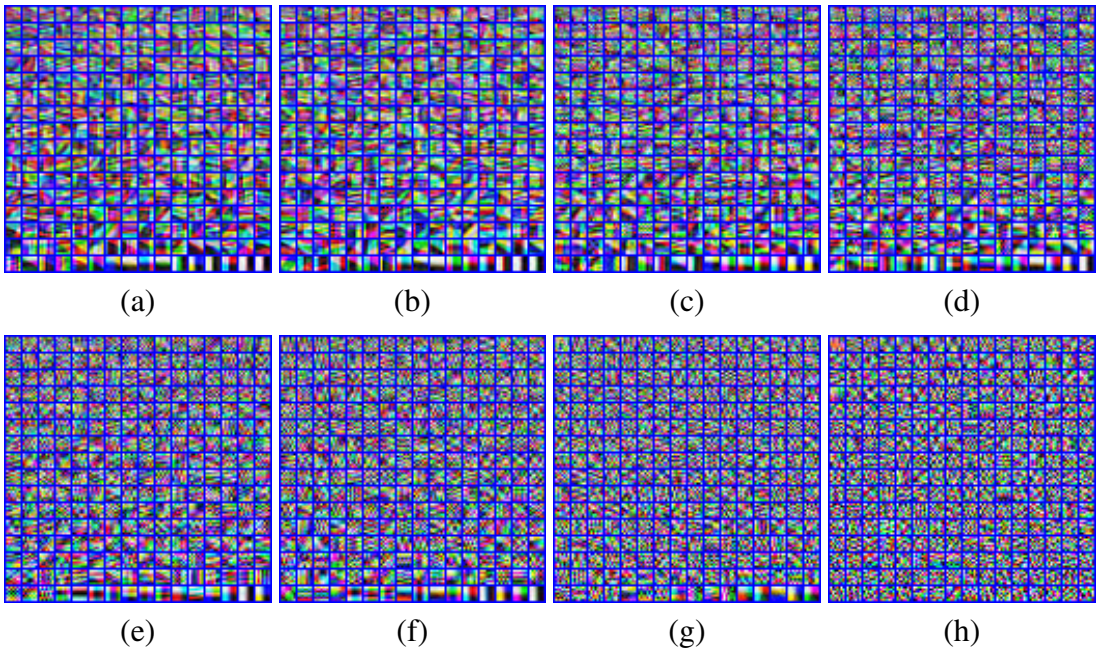

Figure 11: Dictionary matrix for different noise density in the image (a) NP=10 (b) NP=20 (c) NP=30 (d) NP=40 (e) NP=50 (f) NP=60 (g) NP=70 (h) NP=80

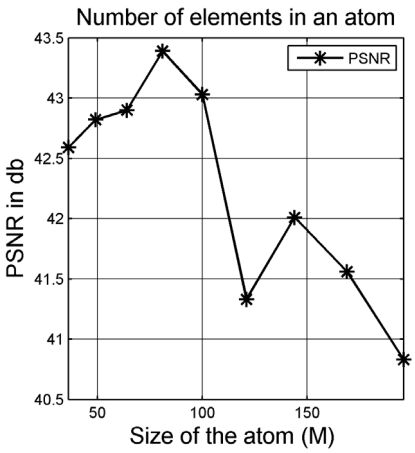

Figure 12: Number of elements in an atom or basis based on PSNR value

5.2. Computation Time

The proposed algorithm was implemented in MATLAB2007 with computer containing an Intel Core2 Duo T7500 processor at 2.00 GHz speed and 2GB DDR RAM. Average time was calculated for lean image of 10 experiments per noise level, for both impulse and random value noise being tested and tabulated in Table 21. Image size was restricted

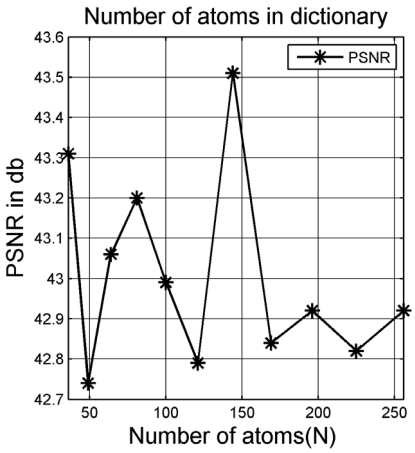

Figure 13: Number of atoms in a dictionary A based on PSNR value

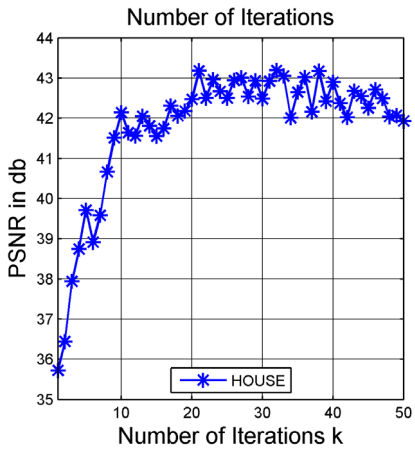

Figure 14: Determination of number of iterations required for the algorithm

Table 19: Sparse coefficient for color image

Noise %/ Image	10	20	30	40	50	60	70	80
Lena	39.08	36.29	33.54	28.15	19.20	11.03	1.800	0.200
Airplane	26.68	23.44	21.14	17.19	10.67	5.650	0.860	0.090
House	34.21	27.45	23.97	19.38	12.75	6.790	1.040	0.130
Barbara	44.85	42.64	38.31	32.84	21.72	12.61	2.220	0.210
Baboon	56.64	46.30	38.84	29.87	17.38	8.880	1.350	0.190
Peppers	44.35	40.79	38.67	33.24	23.08	12.84	2.080	0.200

to uniform 256×256. From Table 21, NAFSM algorithm has very less time than other algorithms, due to impulse noise detection in spatial domain is simplified to 0 and 255

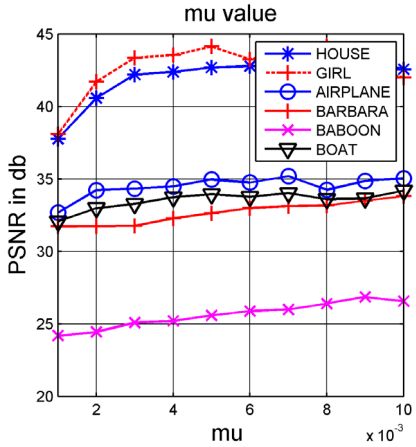

Figure 15: Plot to determine the μ value for soft thresholding

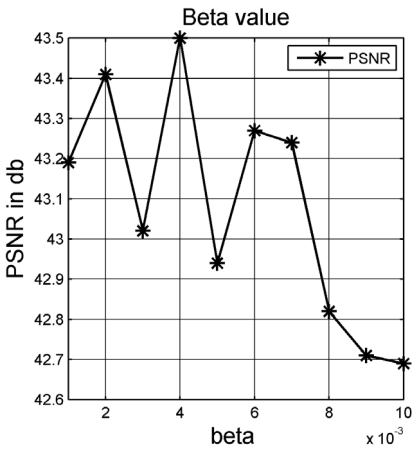

Figure 16: Plot to determine the β value for dictionary learning

Table 20: Sparse coefficient for gray image (coarse part)

Noise %/ Image	10	20	30	40	50	60	70	80
House	8.82	8.49	8.00	7.52	7.15	5.65	3.30	0.97

values, it performs poorly for random value impulse noise. NAFSM, DBA and WMF has lesser execution time than other algorithm, but their PSNR value is significantly less than the proposed algorithm. PSMF has moderate computation time than spatial domain algorithms, its PSNR value is far less than those algorithms. The proposed algorithms computation time is less than the DL-INR algorithm.

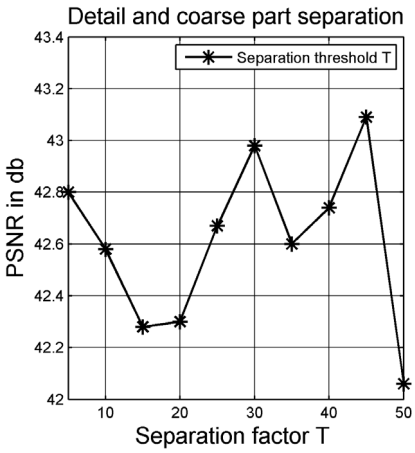

Figure 17: Threshold value to separate detail and coarse image batches

Table 21: Computation time

Noise %/ Algorithm	10	20	30	40	50	60	70	80
WMF	3.460	9.480	9.070	9.090	14.58	17.04	14.92	14.62
PSMF	18.58	14.49	21.48	14.57	23.18	16.59	20.35	15.27
DBA	2.370	2.400	2.390	2.390	2.400	2.360	2.370	2.400
NAFSM	0.510	0.870	1.240	1.620	1.990	2.390	2.730	3.090
DL-INR	369.63	368.95	372.54	364.69	364.68	364.07	361.66	362.24
HD-NR	98.16	96.98	102.34	103.14	96.56	91.94	92.82	99.00

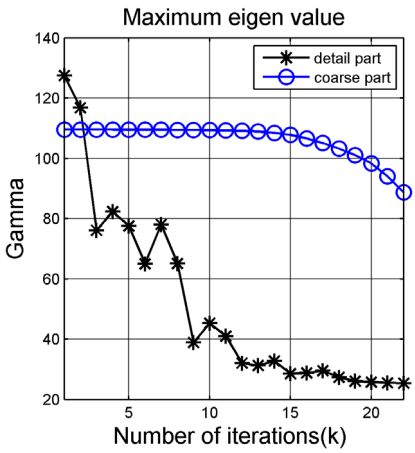

Figure 18: Maximum eigen value for random initial value as dictionary value

6. Conclusion

Two different dictionaries for coarse and details parts of image, improves the PSNR values, preserves the edges and details. Two different dictionaries requires two different

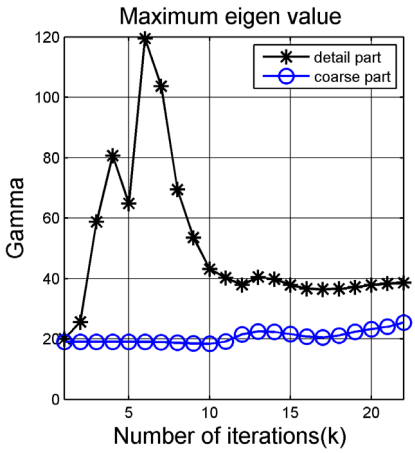

Figure 19: Maximum eigen value for DCT matrix as a initial dictionary value

soft thresholding methods. Two different initials values used for coarse and detail parts of image. Two different dictionaries methods improves a quality of image and also gives idea about complex dictionary implementation for sparse representation. The proposed algorithm performs in terms of metrics like PSNR and SSIM than other algorithms in sparse and spatial domain.

References

[1] J. Astola, P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Newyork, 1997.

[2] David. C. Lay, *Linear Algebra and Its Applications*, Addision-Wesley, Reading, MA, 1994.

[3] Z. Wang and D. Zhang,"Progressive switching median filter for the removal of impulse noise from highly corrupted images", *IEEE Trans. Circuits Syst.,* vol. 46, pp. 78–80, 1999.

[4] T. Chen and H. R. Wu, "Adaptive impulse detection using Center-Weighted median filters", *IEEE Signal processing letters,* vol.8, pp. 1–3, Jan 2001.

[5] Portilla, J. Strela, V. Wainwright, MJ. and Simoncelli, EP, "Image Denoising using scale mixtures of Gaussian in the wavelet domain", *IEEE Transactions Image processing*, vol.12, pp. 1338–1351, 2003.

[6] Starck, J.-L., Candes, E.J., Donoho, D.L,"The curvelet transform for image denoising",*IEEE Transactions Image processing*, vol.11, no.6, pp. 670–684, 2002.

[7] Bogdan Smolka, Andrzej Chydzinski, "Fast detection and impulsive noise removal in color images", *Real-Time Imaging*, vol.11, no.5, pp. 389–402, 2005.

[8] Takanori Koga and Noriaki Suetake, "Random-valued Impulse Noise Reduction in Color Image by Using Switching Vector Median Filter with MST-based Noise Detector", vol.38, no.3, pp. 283–288, 2011.

[9] Mehdi Nasri, Hossein Nezamabadi-pour, "Image denoising in the wavelet domain using a new adaptive thresholding function", *Neurocomputing*, vol.72, no.4, pp. 1012–1025, 2009.

[10] Mairal, Julien, et al., "Discriminative learned dictionaries for local image analysis". *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2008.

[11] P. Bouboulis , K. Slavakis and S. Theodoridis, "Adaptive kernel-based image denoising employing semi-parametric regularization", *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1465–1479, 2010.

[12] S. Wang , Q. Liu , Y. Xia , P. Dong, J. Luo , Q. Huang and D. D. Feng, "Dictionary learning based impulse noise removal via $\ell 1 - \ell 1$ minimization", *Signal Process.*, vol. 93, no. 9, pp. 2696–2708, 2013.

[13] Q. Liu, J. Luo, S. Wang, M. Xiao and M. Ye, "An augmented Lagrangian multiscale dictionary learning algorithm", *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 1–16, 2011.

[14] Q. Liu , S. Wang , J. Luo , Y. Zhu and M. Ye, "An augmented Lagrangian approach to general dictionary learning for image denoising", *J. Vis. Commun. Image Represent.*, vol. 23, no. 5, pp. 753–766, 2012.

[15] S.H. Chan, R. Khoshabeh, K.B. Gibson, P.E. Gill, T.Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration", *IEEE Trans. On Image Proce.*, Vol.20, pp. 3097–3111, 2011.

[16] D. L. Donoho, "Compressed sensing", *IEEE Trans. Inf. Theory,* vol. 52, no. 4, pp. 1289–1306, 2006.

[17] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Method's*, Athena Scientific, Reimont, MA, 1982.

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

[19] M. Aharon, M. Elad and A. M. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them", *J. Linear Algebra Appl.*, vol. 416, pp. 48–67, 2006.

[20] M. Elad, *Sparse and redundant representations from theory to applications in signal and Image processing*, Springer, 2010.

[21] 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*, Volume 24, Issue 7, pp. 956–967, October 2013.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity", *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[23] Allen Y. Yang, Zihan Zhou, Arvind Ganesh, S. Shankar Sastry, Yi Ma, "Fast L1-Minimization Algorithms For Robust Face Recognition", *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3234–3246, 2013.

[24] Bryt, Ori, and Michael Elad, "Compression of facial images using the K-SVD algorithm", *Journal of Visual Communication and Image Representation*, vol.19, no.4, pp. 270–282, 2008.

[25] A. Gramfort, C. Poupon, M. Descoteaux, "Denoising and fast diffusion imaging with physically constrained sparse dictionary learning", *Medical Image Analysis*, Vol.18, pp. 36–49, 2014.

[26] M.R. Mohammadi, E. Fatemizadeh, M.H. Mahoor, "PCA based dictionary building for accurate facial expression recognition via sparse representation", *J. Vis. Commun. Image. Represent.*, vol.25, pp. 1082–1092, 2014.

[27] M. Zhao, S. Li, J. Kwok, "Text detection in images using sparse representation with discriminative dictionaries", *Image and Vision Computing*, Vol.28, pp. 1590–1599, 2010.

[28] Ran Xu, Jianbin Jiao, Baochang Zhang, Qixiang Ye, "Pedestrian detection in images via cascaded l1 norm minimization learning method", *Pattern recognition*, Vol.45, pp. 2573–2583, 2012.

[29] Xiao, Y. and Zeng, T.Y. and Yu, J. and Ng, M.K, "Restoration of images corrupted by mixed Gaussian-impulse noise via $\ell 1 - \ell 0$ minimization", *Pattern recognition*, Vol.44, pp. 1708–1720, 2011.

[30] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", *IEEE Transactions on Signal Processing*, vol.54, 4311–4322, 2006.

[31] Rauhut. H, Schnass. K, Vandergheynst. P, "Compressed Sensing and Redundant Dictionaries", *IEEE Transactions on Information Theory*, vol.54, pp. 2210–2219, 2008.

[32] Gribonval. R, Schnass. K, "Dictionary identification: sparse matrix-factorization via $\ell 1$-minimization", *IEEE Transactions on Information Theory*, Vol.56, pp. 3523–3539, 2010.

[33] W. Yin, S. Osher, D. Gold farb, J. Darbon, "Bregman iterative algorithms for l1 minimization with application to compressed sensing", *SIAM Journal on Imaging Sciences*, 1, 2008, pp. 143–168.

[34] "Digital Compression and Coding of Continuoustone Still Images", Part 1, Requirements and Guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, Nov. 1993.

Biography

P. Jothibasu was born in 1978 in India. He received B.E. from Bharathiyar University and M.E. from Anna university. Presently he is doing part time doctoral degree in Anna University, Chennai. Currently he is working as an Assistant Professor in Electronics and Instrumentation Engineering Department, R.M.K Engineering College, Chennai. His research interests are signal processing, Image processing and embedded systems.

P. Rangarajan was born in 1969 in India. He received B.E in Electrical and Electronics Engineering from Bharthiyar University, M.E and Ph.D from Anna University. He is presently working as a Professor in Electrical and Electronics Engineering Department, R.M.D. Engineering College, Kavaraipettai, Chennai. His research interests are VLSI, VLSI system Design, signal processing and Image processing. He has successfully guided many doctoral degree scholars. He also carried out research projects funded by national and international Organizations.