International Journal of Mathematics and Computational Science
Vol. 2, No. 1, 2016, pp. 1-7
http://www.aiscience.org/journal/ijmcs
ISSN: 2381-7011 (Print); ISSN: 2381-702X (Online)

An Efficient Image Denoising Approach Based on Dictionary Learning

Mohammadreza Karimipoor[1, *], Vahid Abolghasemi[2], Saideh Ferdowsi[2]

[1]Department of Electronics, Faculty of Engineering, Shahrood Science and Research Branch, Islamic Azad University, Shahrood, Iran
[2]Faculty of Electrical Engineering and Robotics, University of Shahrood, Shahrood, Iran

Abstract

In this paper, a denoising method based on dictionary learning has been proposed. With the increasing use of digital images, the methods that can remove noise based on image content and not restrictedly based on statistical properties has been widely extended. The major weakness of dictionary learning methods is that all of these methods require a long training process and a very large storage memory for storing features extracted from the training images. In the proposed method, using the concept of sparse matrix and similarities between samples extracted of similar images and adaptive filters the training process of dictionary based on ideal images have been simplified. Finally Images are checked based on its content by implicit optimization of memory usage and image noise will be removed with a minimum loss of stored samples in existing dictionary. At the end, the proposed method is implemented and results are shown its capabilities in comparison with other methods.

Keywords

Denoising, Sparsity, Clustering, Kmeans, Dictionary Learning

Received: November 10, 2015 / Accepted: December 9, 2015 / Published online: January 5, 2016

1. Introduction

Many researches have been conducted in the field of denoising so far. In general, images denoising systems include preprocessing steps, feature extraction and noise removal.

In [1] two new methods for eliminating 3- dimensional medical images that use scattering features and self-similarity of images are introduced. The proposed methods are based on thresholding the discrete cosines transform of 3-dimensional moving window. Both high performance speed and acceptable quality have made these methods suitable for many clinical and research works. In Glorious method [2], an estimation of the image without noise is obtained using a hard thresholding. A collection of 3-dimensional DCT blocks with 4×4×4 sizes has been used to remove the local noise and is eliminated noise thresholding, so an estimation of noiseless image is obtained. In [3], a noiseless method has been proposed which is called structural adaptive scattered noise deletion (SASD) and takes advantage of fractal of images. Here, one similarity criteria is defined based on local average and on a list of the revised structural similarity to find a collection of similar pieces that have been arranged in 3 – dimensional arrays. Also an effective and simple structural-adaptive window following method simple is proposed to achieve the sparse representation of these arrays. Components of the noise are obtained using structural adaptive arrays and two-dimensional wavelet conversion. This method shows the image details well and maintains the properties, image contrast, Soft Structures and creates very slight discontinues.

The simulation indicated that this method is obtained for images with a well enough structure abundantly and appropriate balance between the image contrast and flatness

* Corresponding author
E-mail address: karimipoorm@yahoo.com (M. Karimipoor)

of image; therefore, this method is potentially very useful for healing highly destroyed image. In [4] three methods of medical images noise deletion based on variable mode decomposition VMD, experimental mode decomposition EMD and discrete wave transform DWT have been proposed and compared with each other. This paper has used thresholding wavelet in wavelet transform, lowest mode of variable mode decomposition VMD and also has applied the experimental mode decomposition for noise deletion of destroyed images with Gaussian noise. A comparison has been conducted on a data collection containing brain images, prostate tissue and retina images, and results reveal that EMD and VMD methods are better than DWT method based on canonical thresholding. A combination of soft thresholding and proportional contraction method is used in [5]. According to images and SNRs, it can be easily seen that this method is better because of the use of new high-frequency coefficients. An adaptive filtering method has been performed in [6]. This filter is based on neutral collection method (neutrosophic) of average filter. The efficiency of this filter has been compared with the average filter and methods of classic non local average denoising (NLM). This filter operates better than the average typical filter for different levels of noise. In [7], a non- local average estimator is adaptively used for denoising images amplitude. In this way, the corrected ways contain 2 smoothing parameters. Amount of weights are based on a reliable estimation to minimize noise power of image. A new method of hybrid average as denoising algorithm has been presented in [8]. In short, the results of comparative study reveal that the current method presents an appropriate SNR and PSNR methods with a low RMSE. The proposed algorithm while eliminating noise preserves the original structure and details. The proposed method has been compared with average, mean, combined average, wavelet thresholding and winner filter. In this article it is clear that quality evaluation criteria, while maintaining structural details of the proposed procedure operate better than all other methods. Moreover, experimental results show that not only noise is removed point wisely, but also the details and image edges are preserved. [9] Discusses about deleting Gaussian noise from magnetic resonance images. The proposed algorithm based on Contourlet transform has been more effective than the wavelet method to noise removal, especially for Gaussian noise elimination.

The results of quantitative and qualitative analysis show that the proposed algorithm is better than the conventional wavelet method in terms of visual and PSNR. A method based on sporadic and sparse representations on taught dictionaries has been used in [10]. To create dictionary, 2 training options are considered; using the spoiled image or training on a qualified image database collection. A method of combined denoising image based on wavelet and sporadic representation model has been presented that is named SWK-SVD. Comprehensive and adaptive dictionaries have been obtained by training on the approximation of the image and high frequency wavelet coefficients. This approach will lead to noise deletion operation both in PSNR and visual effects with high noise. A method based on clustering building blocks of image is performed in [12]. Since clustering has a special place in the noise deletion methods, effective preprocessing method tools for clustering is proposed in this paper.

After this preprocessing, general clustering is performed and representatives from clusters with no noise will come into the final dictionary. Blocks of image noise are analyzed in the dictionary and de-noised version is obtained. This method

Enhances the speed of learning dictionary and will improve the quality of the results for de-noising applications goals.

In this paper, a method based on sparse representation with the purpose of high speed and proper quality has been proposed which largely covers the last objections.

As shown in the introduction, denoising methods based on content are all widely used and are now continually expanding. The rest of the article is as follows. Fist, proposed method theory and its principles are expressed. Then the results of simulations are compared with other papers. The last part of the article contains summarizing the results, conclusions and will be terminated.

2. Roposed Method

2.1. Preprocessing

In the first step, the main purpose should be giving a basic knowledge to the system to remove appropriate noise. With this view, at this stage first the image is divided into a series of non-overlapping windows based on the dimensions of the image. These square shaped overlapping windows will be analyzed individually in terms of existence of noise criteria. According to [13], the window dimension is selected 8*8. In this way, first the input image is changed to the extent that its row and column be divisible by 8 and then a very high number of non-overlapping 8*8 windows are broken.

2.2. Batching Algorithm

In the proposed method, operations categorization is directly done on the preprocessing step windows and no side feature can be detected. The selected batch algorithm is Kmeans which is considered as one of the most common and widely used no guide algorithms among the batching algorithms. This algorithm is based on Euclidean distance and a random selection of the starting points, directing them to accumulation centers and elimination of less important centers. The algorithm tries to detect the center and divide the N- dimensional space into some subspaces such that the number of vectors would be of equal ratio in this space. Obviously, due to the inconsistent existing form, number of points in the batches or centers are not completely equal and are always slightly different.

A Kmeans is considered a better that can select the number of samples within each cluster almost equal while finding suitable sites. In this way, we can make sure that the selected centers are ideal. Therefore, the space is divided into a number of appropriate spaces. At first, the conducted simulation in Kmeans algorithm in MATLAB software was used by default. We had 4096 windows for a typical 512×512 image. Windows extracted from this image were separately applied to Kmeans algorithm and centers were determined. If we assume that evaluates centers are total pixels of the present window of the image. Given the number of 4096 present windows we almost need 64 centers. After calculating centers by Kmeans algorithm of MATLAB, the number of windows of each cluster was calculated. Also it was determined due to the high dimensionality of input vector, the Kmeans software is not operating properly and change variance of the existing vector in each cluster is very high.

Ideally, if the number of vectors in each cluster be equal, variance of these numbers should be zero. Calculation method was such that 8*8 patches extracted from the image are ordered for 64 vectors. Finally, these vectors are arranged below each other. For instance, the number of 4096×64 vectors is obtained from a 512×512 image and is applied to Kmeans as an input vector. To fix this fault, we designed unique Kmeans algorithm. First in this method 4096 number of 64-vectors were arranged based on mean value. To calculate each 64- window, the present pixels are added together and divided into the total number. Then, according to the average of 4096 windows, 16 equal subspaces were selected based on the average. Also, the center of each category is considered as its total average. In this way, the windows are divided based on their brightness which is in fact average standard. Next, variance of 16 obtained packages is estimated. Then windows with less average and appropriate averages are categorized in a group.

Table 1. First eight samples from the eight centers selected by the proposed batching algorithm.

center	1	2	3	4	5	6	7	8
1	47.33	46.84	47.03	47.17	47.36	46.53	47.2	46.8
2	48.47	47.86	47.95	47.38	47.27	47.06	47	47.6
3	51.34	48.58	47.36	47.45	46.23	46.66	48.3	49.6
4	61.95	56.3	50.66	48.5	47.91	48.63	49.6	50.2
5	52.48	52.83	53.3	54.38	54.78	54.02	53.6	53
6	56.19	55.66	54.83	54.91	55.23	55.34	55.2	56.1
7	58.88	59.17	57.86	56.06	56.23	56.69	57.8	58.9
8	65.36	63.91	61.59	60.55	61.89	63.7	66.6	68.1

Also, windows with higher variance are put in separate sets.

This is why the low variance is considered as equality of that window and constant background. Moreover, higher variance is higher energy of window and the relatively higher changes. In this way, we can divide 4096 available windows in the basic image into 16 different batches without reducing the information. Then these batches of 16 samples are divided into 4 different groups based on the variance of each category. Table 1 shows a few samples of its first eight members. Two solutions were considered by specifying these 64 batches.

Storing all the samples with the center of the batch which saving them but consumes a very high volume memory does not look a wise way.

Saving the centers and removal of samples within the cluster that shows a sharp drop of information and efficiency reduction.

To overcome these paradoxes a method based on sparse matrix was selected.

Dates of manuscript submission, revision and acceptance should be included in the first page footnote. Remove the first page footnote if you don’t have any information there.

2.3. Storing Batches Based on Sparse

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Do not use abbreviations in the title unless they are unavoidable.

Given that our categorization standard was mean and variance. First, the center of each batch was considered equal to the average of available vectors. It can be shown that selecting the average is the most optimal choice in terms of mean squared error criteria. The volume of calculations on batching stage conducted based on random selection algorithm of beginning points and processing in MATLAB is much better. So, the amount of calculations in this algorithm is considered negligible and insignificant. Even if the input vectors have dimensions higher than 4196, the proposed method can easily explain it. After calculating each of the 64 batches based on average and variance, instead of storing the 8*8 windows directly, their difference with the Center was selected as the storage standard. For example, if difference between the third point of array and the third points of 64-vector is equal to 200 and pixel numerical value of one of windows is reported 198, then the difference of these two numbers i.e. 2 will be selected as storage standard.

If the rate of changes is less than 2% of the center value in the pixel or the threshold level was lower than 3 pixels, amount of this element would be negligible and almost equal to zero. In other words, in the range of 200 pixels, numbers between 196 to 214 pixels are assumed like 200 and the value store of them is zero. In this case, it was revealed that almost 32% of stored pixels of normal image will be zero. In this regard, the volume of computing and processing will be considerably reduced and the resulting matrix of existing vectors per cluster of a matrix will be totally sparse. According to the conducted evaluations in this regard if the matrix has relatively large changes such as the face or the bustling sights, the stored pixels will about 25%. In this case, still computing reduction will be very evident and significant and again the resulting matrix will be a sparse matrix. So instead of storing a 4096*64 matrix, characteristics of 64 batch centers and a sparse matrix are stored. Furthermore, this matrix indicates the ideal and noiseless changes with respect to the origin. Table 2 demonstrates an example of a center and a number of related patches. Also, table 3 shows the resulting difference as a sparse matrix.

Table 2. The first eight samples of a center and 7 samples in its batch that has been selected by the proposed algorithm.

Number	1	2	3	4	5	6	7	8
1	47.33	46.84	47.03	47.17	47.36	46.53	47.2	46.8
2	52	48	50	43	43	44	45	45
3	51	48	46	47	52	51	57	51
4	45	42	45	41	38	42	41	43
5	41	42	37	44	45	44	47	41
6	42	46	41	44	57	47	46	47
7	48	40	38	41	45	44	41	41
8	48	48	44	48	44	52	47	47

Table 3. First eight samples of a center and 7 samples in the batch.

Number	1	2	3	4	5	6	7	8
1	47.33	46.84	47.03	47.17	47.36	46.53	47.2	46.8
2	-4.67	0	0	4.17	4.36	0	0	0
3	-3.67	0	0	0	-4.64	-4.47	-9.8	-4.2
4	0	4.84	0	6.17	9.36	4.53	6.2	3.8
5	6.33	4.84	10.03	3.17	0	0	0	5.8
6	5.33	0	6.03	3.17	-9.64	0	0	0
7	0	6.84	9.03	6.17	0	0	6.2	5.8
8	0	0	3.03	0	3.36	-5.47	0	0

Due to the conducted simulation, the amount of computational load of algorithm at this stage is approximately 17 times less than the Kmeans algorithm of MATLAB. It must be mentioned that, with regard to this fact that no appropriate assessment for memory usage was found in this case and also we could not analytically prove the used memory with respect to Kmeans of the basis for the proposed algorithm, RAM changing value in implementing the system while running was used. It showed that the occupying amount of space while creating kmeans command of MATLAB was considerably more than the proposed algorithm. By determining the batch with 64 Center obtained from safe image we will enter the next phase, namely, how to create a dictionary and then apply it to a new image.

2.4. Dictionary Training

After determining the sampling algorithm and the used non-overlapping windows, the next stage will be dictionary training. At this stage, a lot of images of different types are transferred to the system. Then the images, assuming being ideal, will be stored for a code word by the system. Code word of each image contains extracted 64 centers from the original image, difference sparse vector related to each center. In this case, after changing a sample image we will have about 4096 windows and the 64- dimensions sparse vector. In this case, due to the small size of differences in storage volume, a significant saving is created.

2.5. Using Dictionary and Noise Removal

After the dictionary training phase, a new noised picture is entered. Assuming that the probability density function of noise is unclear, obviously there is no background of image texture and form is available.

In this case, the input image is separated into the same number of windows similar to the dictionary training. The goal is that by comparing the split and stored windows in the dictionary, any window be replaced by a low- noise window or we can obtain a definition of the information content of that window.

After splitting the image to non- overlapping windows that does not differ from the dictionary training stage, the image is window likely compared with the existing information of dictionary that have been approximately considered. After analyzing the image noise and splitting it for different overlapping windows, it is necessary that these windows be compared with the existing words in the dictionary that are in fact the same patch and distance vector from the patch center. Finally a decision is made about the about the most similarities to what is contained in the dictionary. At the first step, distance between the patch to the total centers is calculated by point wise criteria, and the selected centers will be calculated for an alternative. Supposing the two possible candidate centers is regard to the differences and distances. The next decision making principle will be the vector of the center and saved patch in that. Comparing all existing patches in that center with this patch will eventually lead to selecting best candidate in terms of replacement. For this replacement, in addition to the patch stored in the dictionary, the difference between the patch and surrounding patches should be necessarily considered so that minimum change in boundaries discontinues was created by this replacement. Otherwise, the discontinuity in boundary point may lead to window mode in the image. If the comparison has been correctly done and also an ideal replacement be conducted, we naturally will expect full recovery of the original image. Here there are to ambiguous and principal points, the first point is that basically there is no original image in the dictionary. So, obviously some replacements are not considered ideal.

The second case, the mistake possibility in replacement operation should always be considered. Hence, the replacement should be designed such that we encounter with not much of a change in the original image while a mistake occurs. For this reason, while the replacement is occurring, the relaxation algorithm is used. This means that when the original image is replacing in noisy image, a percentage of the recovered image is combined with a part of the old image and final image is considered as a new image. In this way, the image tissue destruction is not so obvious in cases that some replacements are wrong. In addition to that, the safe part replaced with the noisy image will result in reducing effective amplitude of the noise. In this regard, it will considerably enhance the image quality.

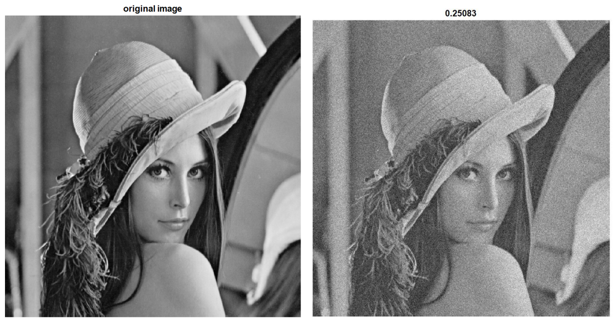
2.6. Adaptive Gaussian Filter

The proposed Adaptive Gaussian filter has been designed such that the difference between the existing candidate and the closest found candidates is calculated and noise parameters be calculated for each window based on this difference. Finally, assuming Gaussian, buffering operation of noise parameters are calculated for the entire image. A Gaussian filter is designed by the calculated parameters for Gaussian filter. Eventually this filter is applied to the image. Naturally, due to the adaptability of such a filter, we have the Gaussian noise parameters with very appropriate accuracy. So it will have optimal performance in the image. The adaptive filter can efficiently filter the existing noise in the image, with condition of noise type specificity i.e. Gaussian. This adaptive filtering technique is in fact a combination of the dictionary training method and adaptive filtering that can be used as a method along with dictionary based training techniques.

3. Results

3.1. Main Results

The results of the current research are as follows. Figure 1, part a, shows the idealized image, and part b of this image indicates an image infected with noise. Figure 2 (b) is related to a denoising image. Figure 1 (b) shows an image after comparing with the trained dictionaries. Image 1 (a) is proposed image. Image 2 (b) is output of the Gaussian adaptive filter. The level of image quality enhancement SSIM criteria has been shown above the new image.



a) b)

Fig. 1. a) main image. B: image noised with white noise of amplitude 20.

a) b)

Fig. 2. Denoising image with Adaptive Gaussian filter B: output of the proposed technique.

3.2. Closer Examination of the Proposed Method

Ambiguous parameters of the algorithm that should be carefully examined and their amount would have a direct impact on the performance of the method are as follows. Relaxation factor that has been assumed 0.4 in fist implementation and represents that 0.4 of the new image is combined with 0.6 of the noise image and final image is created. The window size is assumed 8*8. Table 4 shows changing value of SSIM based on noise and relaxation level. Clearly, the proposed method with relaxation has had the best result in all cases of low noise numbers. In these cases a better result is shown by =0.3. In cases with higher noise amplitude α=0.4. In some cases, Adaptive Gaussian filter results are comparable with the proposed method by relaxation. It can be concluded that α=0.35 is a right choice.

Table 5 with relaxation factor of 0.35 shows the effect of window sizes. It can clearly be concluded that window with sizes 4 and 8 have the best results. Also, with respect to the running time, size 8 has obviously less runtime and Windows with size 8 are the best choice.

Table 4. Changes of SSIM according to noise and relaxation level.

The noise	Relaxation	Image noise	Proposed method	Proposed method with relaxation	Adaptive Gaussian filter
5	0.3	0.6518	0.5307	0.698	0.6033
5	0.4	0.6518	0.4892	0.597	0.5072
5	0.5	0.6518	0.4464	0.4828	0.4415
5	0.6	0.6518	0.4294	0.4418	0.413
5	0.7	0.6518	0.4238	0.4288	0.4037
10	0.3	0.4361	0.4738	0.5857	0.546
10	0.4	0.4361	0.4457	0.5459	0.4852
10	0.5	0.4361	0.4132	0.4511	0.4244
10	0.6	0.4361	0.3961	0.411	0.3966
10	0.7	0.4361	0.3909	0.3977	0.3872
15	0.3	0.3217	0.4207	0.497	0.4898
15	0.4	0.3217	0.4077	0.495	0.4583
15	0.5	0.3217	0.3778	0.4176	0.4045
15	0.6	0.3217	0.3632	0.3798	0.3776
15	0.7	0.3217	0.3579	0.3659	0.3681
20	0.3	0.2504	0.3892	0.4338	0.4595
20	0.4	0.2504	0.3826	0.4581	0.4395
20	0.5	0.2504	0.3589	0.3988	0.3924
20	0.6	0.2504	0.3455	0.3632	0.3667
20	0.7	0.2504	0.3396	0.3485	0.3564

Table 5. Changes of SSIM according to noise and window sizes.

The noise as %	Window dimension	Image noise	Proposed method	Proposed method with relaxation	Adaptive Gaussian filter	run time
5	4	0.6525	0.6171	0.7026	0.6169	42
5	8	0.6525	0.5311	0.7025	0.5964	8
5	16	0.6525	0.4236	0.6896	0.5674	3.2
5	32	0.6525	0.3644	0.6705	0.5426	1
10	4	0.4355	0.5113	0.5759	0.5494	42
10	8	0.4355	0.4676	0.5782	0.5415	8
10	16	0.4355	0.4192	0.5564	0.5245	3.2
10	32	0.4355	0.3664	0.5304	0.5051	1
15	4	0.3211	0.453	0.4889	0.5016	42
15	8	0.3211	0.4242	0.501	0.4924	8
15	16	0.3211	0.4065	0.4908	0.4819	3.2
15	32	0.3211	0.3563	0.4704	0.4656	1
20	4	0.2518	0.4066	0.4249	0.4613	42
20	8	0.2518	0.3893	0.4423	0.4553	8
20	16	0.2518	0.3879	0.4367	0.4423	3.2
20	32	0.2518	0.3633	0.4263	0.4311	1

Table 6. Impact of threshold on the amount sparsity and method accuracy.

The noise	difference	Sparsity	Image noise	Proposed method	Proposed method with relaxation	Adaptive Gaussian filter
5	0.02	27.894	0.6325	0.5311	0.7025	0.5964
10	0.02	27.894	0.4655	0.4676	0.5782	0.5415
15	0.02	27.894	0.3411	0.4242	0.501	0.4924
20	0.02	27.894	0.2818	0.3893	0.4423	0.4553
5	0.04	47.201	0.6325	0.5011	0.5872	0.5298
10	0.04	47.201	0.4655	0.4417	0.4788	0.4361
15	0.04	47.201	0.3411	0.4102	0.4267	0.4461
20	0.04	47.201	0.2818	0.3711	0.369	0.386
5	0.06	59.006	0.6325	0.441	0.5653	0.4771
10	0.06	59.006	0.4655	0.3971	0.4233	0.3511
15	0.06	59.006	0.3411	0.3646	0.3825	0.3822
20	0.06	59.006	0.2818	0.3024	0.3484	0.3579
5	0.08	66.244	0.6325	0.3971	0.4766	0.3893
10	0.08	66.244	0.4655	0.3511	0.4196	0.3082
15	0.08	66.244	0.3411	0.2961	0.3584	0.3739

The noise	difference	Sparsity	Image noise	Proposed method	Proposed method with relaxation	Adaptive Gaussian filter
20	0.08	66.244	0.2818	0.2711	0.2979	0.3448
5	0.1	71.21	0.6325	0.3511	0.4425	0.3153
10	0.1	71.21	0.4655	0.3324	0.4144	0.2468
15	0.1	71.21	0.3411	0.2845	0.2899	0.3382
20	0.1	71.21	0.2818	0.2214	0.2484	0.3188

Table 6 indicates the effect of threshold on the amount of sparsity and method accuracy. Obviously, raising the threshold and removing distance from the center improves the sparsity, but as well as, will increase error decrease quality.

It is obvious that amount of 0.02 and in some cases the amount of 0.04 fit the threshold level and higher levels does not have the necessary efficiency.

Finally, after various simulations and due to the random nature of noise, the relaxation factor between 0.3 to 0.4, and window with size 8*8 and threshold of 2 was detected more appropriate. In addition, in the field of boundaries maintenance and continuity, Adaptive Gaussian filter operate better dictionary-based training method. Whereas, in terms of SSIM criteria, the dictionary-based training is better than the other methods.

4. Conclusion

Two denoising methods were proposed in this paper. The first method that was directly based on dictionary training using sparse matrix properties, there was no need to have a noise probability density function. Also, this method can be directly applied to any noise. The second method, which is based on adaptive filter design, the type of noise probability density function must probably be specified. The reason is that the designed Adaptive filter based on the probability density function be applied to the image and have the required performance. Both methods have been examined on different levels of noise and relaxation parameter that is considered an important parameter. Moreover, their efficiency has been shown compared to the conventional filtering methods. The dictionary-based method is better in terms of SSIM standard, but creates discontinuity at the boundaries, while the designed Adaptive Gaussian filter did not have this problem.

References

[1] Manjon. J, Coupe. C, Buades. A, Collins. L, Robles. M, New methods for MRI denoising based on sparseness and self-similarity, 2012, ELSEVIER, Medical Image Analysis 16 (2012) 18–27.

[2] Guleryuz, O. G, Weighted over complete denoising. 2003, In: Proceedings of the Asilomar Conference on Signals and Systems.

[3] Bao. L, Robini. M, Liu. W, Yuemin. Z Structure-adaptive sparse denoising for diffusion-tensor MRI, 2013, ELSEVIER, Medical Image Analysis 17 (2013) 442–457.

[4] Golshan. H, Hasanzadeh. R, Yousefzadeh. Sh, An MRI denoising method using image data redundancy and local SNR estimation, 2013, Elsevier, Magnetic Resonance Imaging 31 (2013) 1206–1217.

[5] Lahmiri. S, Boukadoum. M, Biomedical Image Denoising Using Variational Mode Decomposition, 2014, IEEE, 978-1-4799-2346-5.

[6] Mohan. J, Krishnaveni. V, A Neutrosophic Approach of MRI Denoising, 2011, IEEE, 978-1-61284-861-7.

[7] Hong. J, Luisier. F, and Wolfe. P. J, Magnitude MR Image Denoising Via Cure-Optimized Non-Local Means, 2012, IEEE, 978-1-4577-1858-8.

[8] Mustafa, Z.A.; Abrahim, B.A.; Kadah, Y.M., "K11. Modified Hybrid Median filter for image denoising," in Radio Science Conference (NRSC), 2012 29th National, vol., no., pp.705-712, 10-12 April 2012.

[9] Satheesh. S, Prasad. K, Medical Image Denoising Using Adaptive Threshold Based ON Contourlet Transform, March 2011, Advanced Computing: An International Journal (ACIJ), Vol. 2, No. 2.

[10] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. IP, 54(12): 3736–3745.

[11] Huibin Li; Feng Liu, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries in Wavelet Domain," in Image and Graphics, 2009. ICIG '09. Fifth International Conference on, vol., no., pp.754-758, 20-23 Sept. 2009.

[12] Hui Luo, Hongliang Chu, Yao Xu, Clustering-Based Image Sparse Denoising in Wireless Multimedia Sensor Networks. Journal of Circuits, Systems, and Signal Processing, March 2015, Volume 34, Issue 3, pp 1027-1040.

[13] Weisheng Dong; Xin Li; Zhang, D.; Guangming Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.457-464, 20-25 June 2011.