International Journal of Bio-Science and Bio-Technology
Vol.9, No.2 (2017), pp. 39-48
http://dx.doi.org/10.14257/ijbsbt.2017.9.2.04

Hardware based Brain MR Image De-Noising using Weiner filter with Discrete Haar Wavelet Transform

Srinivasan Aruchamy, Partha Bhattacharjee and Goutam Sanyal

CSIR-CMERI, National Institute of Technology, India
srinivaspsg@gmail.com, partha11cmeri@gmail.com, nitgsanyal@gmail.com

Abstract

Noise is a serious issue in any brain MR image analysis. Additive noises like Gaussian noise, salt and pepper noise and multiplicative noise like speckle noise are most common noises make MR image to suffer in diagnosis. In brain image analysis, MR image de-nosing plays an important role. Image de-nosing step improves the image quality by removing unwanted noise present in the image by applying some transformation techniques without losing the useful information. In the proposed work an attempt has been made to study different noise models like additive random noise, impulse noise, multiplicative noise and haar discrete wavelet transform combination with weiner filter has been presented. An attempt has been made to implement the same in hardware platform and study the performance of the implemented algorithm. Results were compared with several performance metrics like PSNR, Mean square error (MSE), Absolute Mean square error (AMBE), Structural similarity Index (SSIM). It has been implemented in a single board computer (raspberry pi) open source software platform OpenCV.

Keywords*: Haar wavelet transform, Weiner filter, median filter, salt-and-pepper noise, Gaussian noise, speckle noise*

1. Introduction

An MRI machine uses the magnetic field and radiofrequency pulses to form detailed images of the inner slices of the brain tissues. A good gray level contrast in the brain structures will help to analyze the brain MR Images precisely. Generally, gray level contrasts of the MR images are not superior due to more number of imaging sensors are used in small unit area in the imaging devices. Imaging devices are more sensitive to the noise [1]. Image de-noising play a significant role in improving the quality of the image. Image de-noising is a pre-processing step in which the unwanted pixels (noise) are removed from the original image and improves the image structure by preserving as much inner details as possible. Consider the below image f(x,y) got corrupted due to the noise n(x,y) and produces the noisy image g(x,y).

$$g(x,y) = f(x,y) + n(x,y)$$

This noise may be Gaussian, salt and pepper and speckle [2]. Image de-nosing techniques can be classified in to two different categories special domain and transform domain. The spatial domain techniques operates directly on the pixels values on the image, different classical filter namely lee, Gaussian, mean, median average filters were used. As it operates on image directly it makes use of the similarities in pixel information. In the transform domain the input image is transformed into other domains, in which relationship of transformed co-efficient are studied in detail for removing the noise. A different method of image de-noising has been documented in the literature. Hossein T and Peymen M[3] proposed a global image de-nosing algorithm where they have taken every pixel one by one and

ISSN: 2233-7849 IJBSBT

probable likelihood with all the other pixels. The above task they achieved by doing statistical analysis and approximation using Nyström extention [4]. Wangmeng Zuo *et.al.*, [5], gave more emphasis on the study of visual quality of the resultant image after de-nosing because removing noise will suppress the useful information as well. So, they proposed an algorithm for preserving the gradient of the image histogram which will enhance the texture of the image while removing the noise. Varsha A and Preetha B [6] proposed a Dual tree complex wavelet transform through cross validation technique. The performance of the proposed method has been evaluated based on PSNR and mean structural similarity and coefficient of correlation. Bouteldja M.A *et.al.*, [7] proposed a evolutionary game theory method for image de-noising problem. They considered pixels in the images are autonomous players that search for to maximize a payoff function through a set of different strategies. Strategies are selected by applying probabilities on non-negative weights of the neighboring pixels. Salim lahmiri and Mounir Boukadoum [8] compared three different image de-noising techniques namely Variational mode decomposition (VMD), Empirical mode decomposition and Discrete wavelet transform. They performed the experiments with the Gaussian noise with MRI datasets and retina digital image set. They validated with the PSNR values and found that DWT is out performed compare to other two methods. Kaihua Gan *et.al.*, [9] proposed a method to preserve structure information during the image de-noising process. They proposed Non Local means de-noising algorithm based on edge detection. Sobel operator [17] is applied first with the noisy image. The resultant image is used to improve the weight function of non local means algorithm. They used Euclidean distance with edge structure to find the similarity of neighborhood pixels.

2. Proposed Method

In the proposed method the Dicom [10] format images are taken as input and single slice is extracted for further processing. Different noise models [11] have been generated and added to the input image, subsequently wiener filter and median filters are applied separately. Discrete haar wavelet transform is applied to the filtered image where the details wavelet co-efficient are studied and inverse transform is performed to reconstruct the image. The performance metrics were calculated for the output image. The schematic overview of the proposed method is shown in Figure 1. The experiments were carried out in Raspberry pi single board computer. According to the equation 1 g(x,y) is a noisy image, weiner filter takes each pixels and calculates local mean using the equation 2 and the variance using the equation 3, and minimize the mean square error between the desired image f^(x,y) and the original image f(x,y)

$$\mu = \frac{1}{MN}\sum_{x=1,y=1}^{MN} g(x,y) \tag{2}$$

$$\sigma^2 = \frac{1}{MN}\sum_{x=1,y=1}^{MN}(g(x,y)-\mu)^2 \tag{3}$$

The image f^(x,y) is calculated using the (4) by taking each pixel from the image g(x,y).

$$f\hat{}(x,y) = \mu + \frac{\sigma^2-\sigma_k^2}{\sigma^2}\,(g(x,y)-\mu) \tag{4}$$

The discrete haar wavelet function is described by

$$\Psi(\mathrm{x}) = \begin{cases} +1 & \text{when } 0 \le \mathrm{x} < \frac{1}{2} \\ -1 & \text{when } \frac{1}{2} < \mathrm{x} \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\varphi(x) = \begin{cases} 1 & \text{when } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Where, $\Psi(x)$ wavelet function $\varphi(x)$ scaling function

$\psi_{m,n}(x) = 2^{n/2}\psi(2^n x - m)$, n=0,..... m = 0,.....$2^n$ - 1.
$\varphi_{m,n}(x) = 2^{n/2}\varphi(2^n x - m)$, n=0,.... m = 0,.....$2^n$ - 1.

Where, $\psi_{m,n}$ are average or sum coefficients. $\varphi_{m,n}$ are difference or detail coefficient.

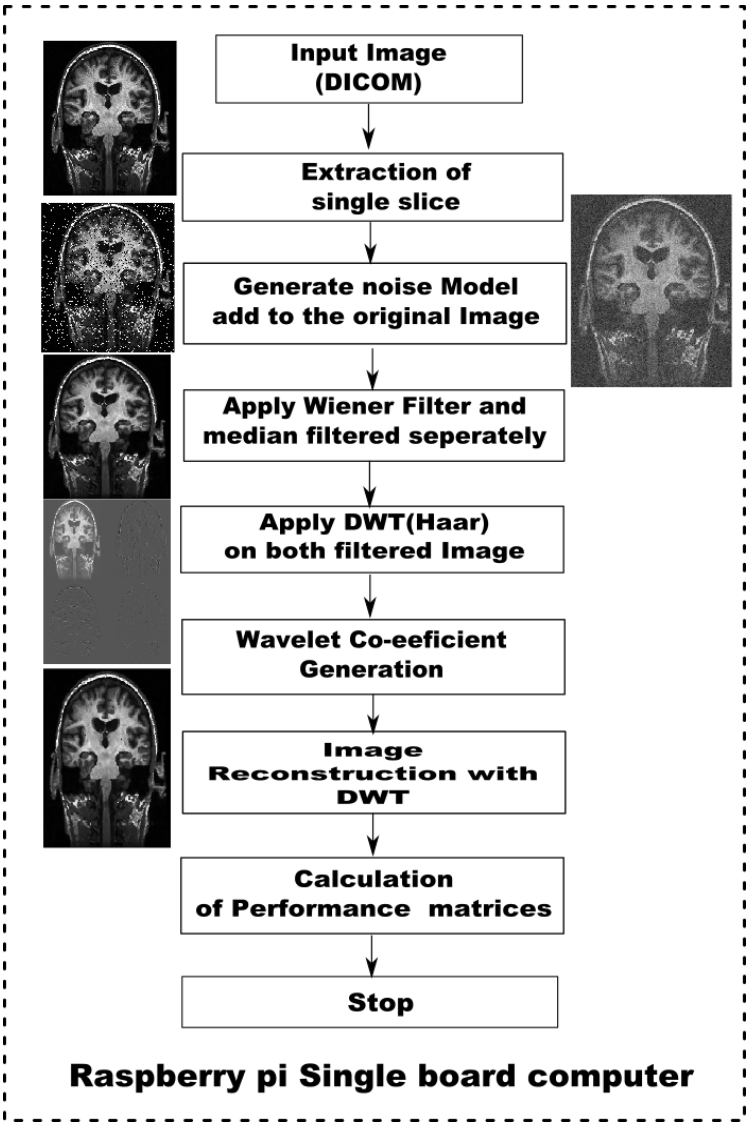

Figure 1. Schematic Overview of Proposed Method

For an image I, which is of the size M x N and having intensity f (x,y) is transformed to wavelet domain through the mathematical expression.

$$W_{\emptyset}(j0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \emptyset_{j0,m,n}(x,y) \quad (7)$$

$$W_{\psi}^{i}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \emptyset_{j,m,n}^{i}(x,y), i \quad = \{H,V,D\} \tag{8}$$

Where, H, V,D are horizontal, vertical and diagonal directions respectively. The Figure 2 depicts the 2D wavelet transform

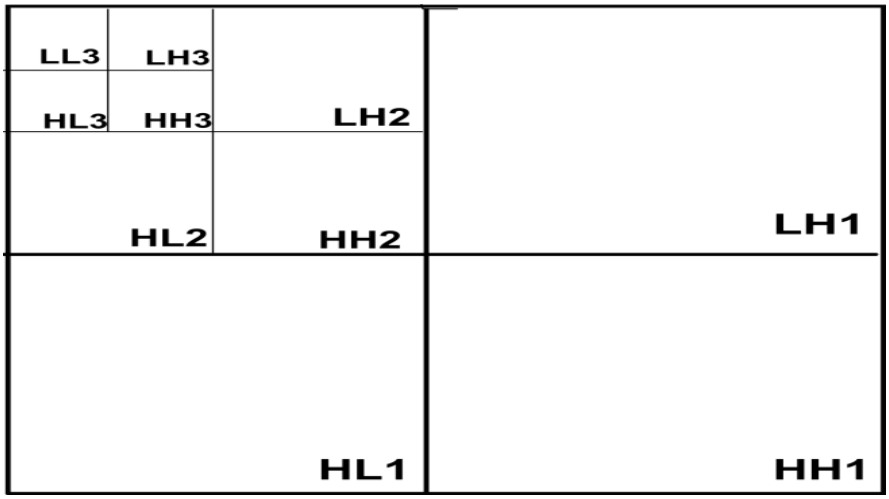

Figure 2. 2D Wavelet Decomposition

3. Experimental Setup

The proposed method was implemented in Raspberry Pi single board computer [12] for ensuring faster execution. The program code is written in Opencv with C++ libraries [13]. The Raspberry Pi single board computer has following features: Clock frequency of 900MHz, 1GB of RAM and 16GB of SD card Memory with Linux operating environment. HDMI interface is used for host communication. The complete experimental setup is shown in the Figure 3.

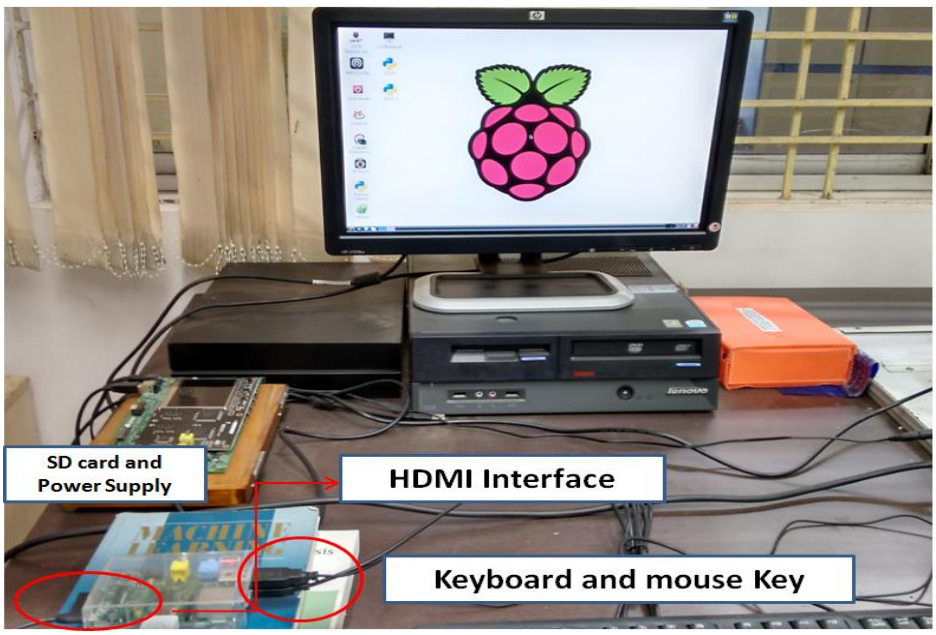

Figure 3. Experimental Setup

The Brain MR images are taken from publically available database OASIS [14]. It contains the ground truth data collected from several patients. The efficacy of the proposed algorithm was evaluated by following parameters.

4. Results and Discussion

4.1. Absolute Mean Brightness Error (AMBE)

The Absolute Mean Brightness Error (AMBE) is calculated using the difference between original and enhanced image AMBE(X,Y)=|XM-YM|,[15]where XM is the mean of the input image and YM is the mean of the output image. Smaller value of AMBE indicates lesser loss of information during enhancement.

4.2. Peak Signal to Noise Ratio (PSNR)

A large value of Peak Signal to Noise Ratio (PSNR) indicates better contrast enhancement in the output image as shown in Fig.6.The PSNR [15] has been computed as,

$$PSNR = 10\log_{10}(L-1)^2/MSE \tag{9}$$

Where, MSE is the mean square error and it is defined as

PSNR is used to assess the degree of contrast enhancement. Greater PSNR indicates better image quality.

$$MSE = \frac{1}{m \times n \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2} \tag{10}$$

4.3. Mean of Structural Similarity Index (mSSIM)

SSIM [16] is used to measure the relationship between two images by structural information. This metric gives better result than PSNR. This SSIM is expressed as follows

$$SSIM(x,y) = \frac{(2\mu x\mu y + c1)(2\sigma xy + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \tag{11}$$

Where, μx and μy are the average of the x and y. $\sigma_x^2$and σ_y^2 are the variance of x and y. is the co-variance of x and y. The mean of the SSIM is calculated using

$$mSSIM = \frac{1}{M}\sum_{i=0}^{M-1} SSIM(x_i, y_i) \tag{12}$$

Table 1. PSNR AMBE mSSIM Values for Different Noise (Salt and pepper) Level

Noise level in db	Salt and pepper								
	20			30			40		
Image No	PSNR	AMBE	MSSIM	PSNR	AMBE	MSSIM	PSNR	AMBE	MSSIM
1	10.9	83.6	0.18	8.66	58.69	0.05	9.03	68.54	0.02
2	12.5	85.2	0.16	10.99	82.56	0.07	10.09	75.62	0.01
3	13.4	94.5	0.24	11.52	107.15	0.08	10.62	76.84	0.02

4	13.0	138.0	0.25	10.66	56.44	0.07	10.41	93.52	0.02
5	12.2	75.0	0.18	10.25	78.51	0.06	9.93	72.92	0.01
6	11.2	108.2	0.24	8.93	64.71	0.07	9.22	47.87	0.02
7	12.6	78.4	0.22	10.86	98.55	0.07	10.15	82.51	0.03
8	13.5	65.5	0.25	11.75	76.09	0.07	10.7	74.96	0.03
9	12.8	77.8	0.26	11.09	69.52	0.09	10.29	57.76	0.03
10	11.9	83.4	0.16	10.42	65	0.06	9.71	61.57	0.01
11	13.2	90.5	0.24	11.52	107.15	0.08	10.62	76.84	0.02
12	12.9	136.0	0.25	11.66	56.44	0.07	10.41	93.52	0.03
13	12.3	85.6	0.15	11.00	81.54	0.07	10.09	75.62	0.01
14	13.4	94.5	0.24	11.52	107.15	0.08	10.62	76.84	0.02
15	13.3	132.0	0.25	10.66	54.44	0.07	10.1	83.52	0.03
16	9.9	73.6	0.11	7.63	52.69	0.04	8.03	60.54	0.02
17	11.5	80.2	0.17	9.99	72.56	0.07	9.09	70.62	0.01
18	12.3	91.5	0.22	10.52	105.15	0.08	11.62	78.84	0.02
19	13.2	132.0	0.21	9.61	56.49	0.08	10.44	92.78	0.02
20	12.7	74.0	0.16	10.51	77.12	0.04	9.13	70.92	0.01
21	11.3	103.2	0.22	8.93	61.71	0.07	10.21	43.87	0.02
22	12.8	77.4	0.32	10.16	97.45	0.07	11.15	86.51	0.03
23	13.5	65.5	0.25	11.75	76.09	0.07	10.7	74.96	0.03
24	12.8	77.8	0.26	11.09	69.52	0.09	10.29	57.76	0.03
25	11.9	83.4	0.16	10.42	65	0.06	9.71	61.57	0.01

Table 2. PSNR AMBE mSSIM Values for Different Noise (Gaussian Noise) Level

Noise level in db	Gaussian Noise								
	10			20			30		
Image No	PSNR	AMBE	MSSIM	PSNR	AMBE	MSSIM	PSNR	AMBE	MSSIM
1	9.70	88.83	0.32	7.55	36.91	0.33	6.60	50.27	0.33
2	10.29	64.11	0.30	7.21	48.68	0.32	6.11	56.12	0.33
3	11.41	69.11	0.43	8.07	66.48	0.43	6.71	63.71	0.43
4	10.65	72.40	0.47	7.20	52.44	0.46	5.94	64.35	0.45
5	10.51	63.36	0.32	7.68	92.27	0.34	6.44	47.67	0.35
6	9.51	64.46	0.44	6.95	44.37	0.44	5.87	56.14	0.43
7	10.61	65.69	0.41	7.72	88.96	0.40	6.54	54.77	0.40
8	11.44	59.39	0.48	8.13	89.59	0.47	6.85	64.20	0.46
9	11.04	58.80	0.47	7.80	101.16	0.47	6.41	60.58	0.46
10	10.37	66.91	0.29	7.74	89.07	0.31	6.60	50.25	0.32

11	9.58	69.40	0.37	7.20	49.44	0.38	5.01	59.35	0.38
12	11.43	60.36	0.39	7.66	89.27	0.32	5.44	43.67	0.37
13	9.59	61.23	0.40	7.43	50.54	0.44	5.12	55.82	0.44
14	10.11	63.69	0.45	7.62	86.96	0.43	6.14	52.77	0.39
15	11.31	58.42	0.46	7.83	87.59	0.44	6.15	63.24	0.45
16	11.52	64.25	0.35	7.37	93.3	0.35	7.07	46.58	0.3
17	10.52	65.35	0.47	6.64	45.4	0.45	6.5	55.05	0.38
18	11.62	66.58	0.44	7.41	89.99	0.41	7.17	53.68	0.35
19	12.45	60.28	0.51	7.82	90.62	0.48	7.48	63.11	0.41
20	12.05	59.69	0.5	7.49	102.19	0.48	7.04	59.49	0.41
21	11.38	67.8	0.32	7.43	90.1	0.32	7.23	49.16	0.27
22	10.59	70.29	0.4	6.89	50.47	0.39	5.64	58.26	0.33
23	12.44	61.25	0.42	7.35	90.3	0.33	6.07	42.58	0.32
24	10.6	62.12	0.43	7.12	51.57	0.45	5.75	54.73	0.39
25	12.32	59.31	0.49	7.52	88.62	0.45	6.78	62.15	0.4

Different Noise levels for salt and pepper noise and Gaussian noise with corresponding performance parameters like PSNR, AMBE, and MSSIM are shown in the Table 1 and 2. From the table it is clear that with the increase of noise level increase the PSNR value decrease gradually. Figure 4 to 6 shows the sample output for different noise model with different stages.

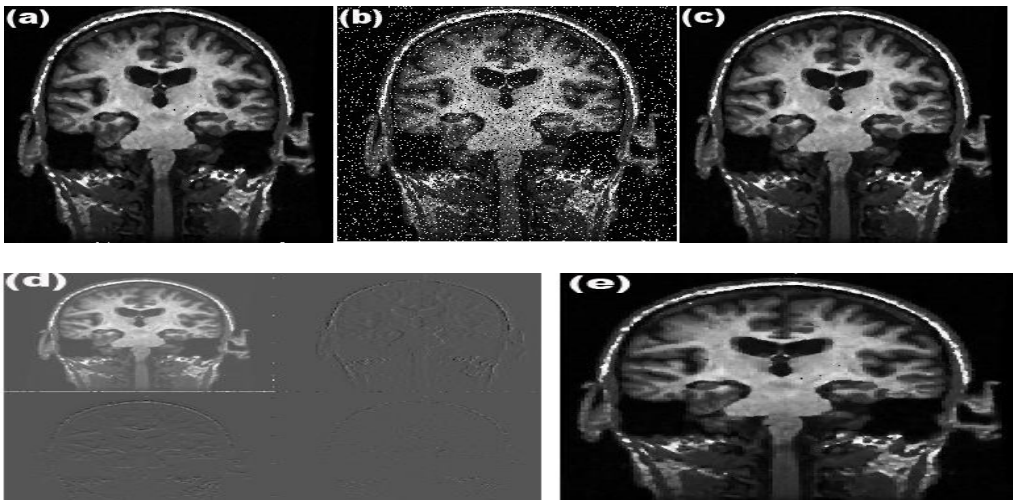

Figure 4. (a) Input Image (b) Noisy(salt-pepper) Image (c) Median Filtered Image (d) Wavelet Decomposition (e) Inverse Wavelet Transformed Image

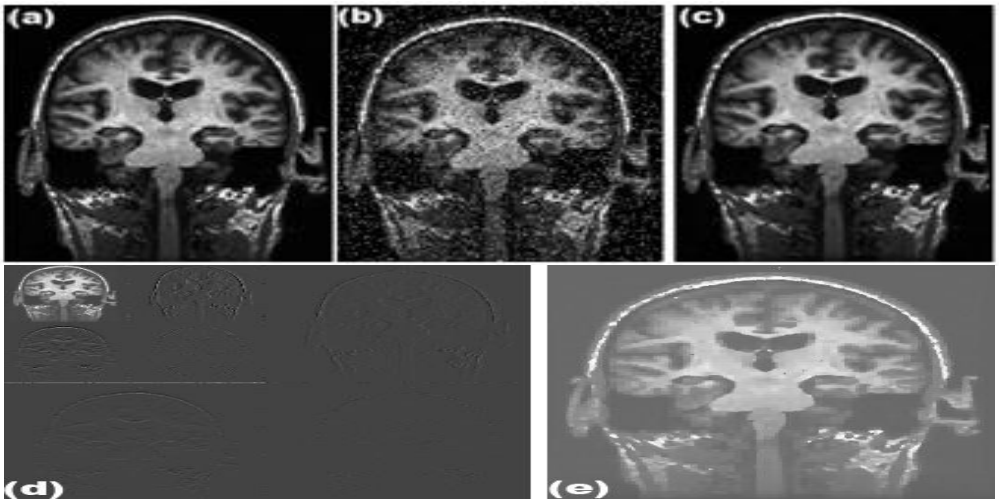

Figure 5. (a) Input Image (b) Noisy(spekle) Image (c) Median Filtered Image (d) Wavelet Decomposition (e) Inverse Wavelet Transformed Image

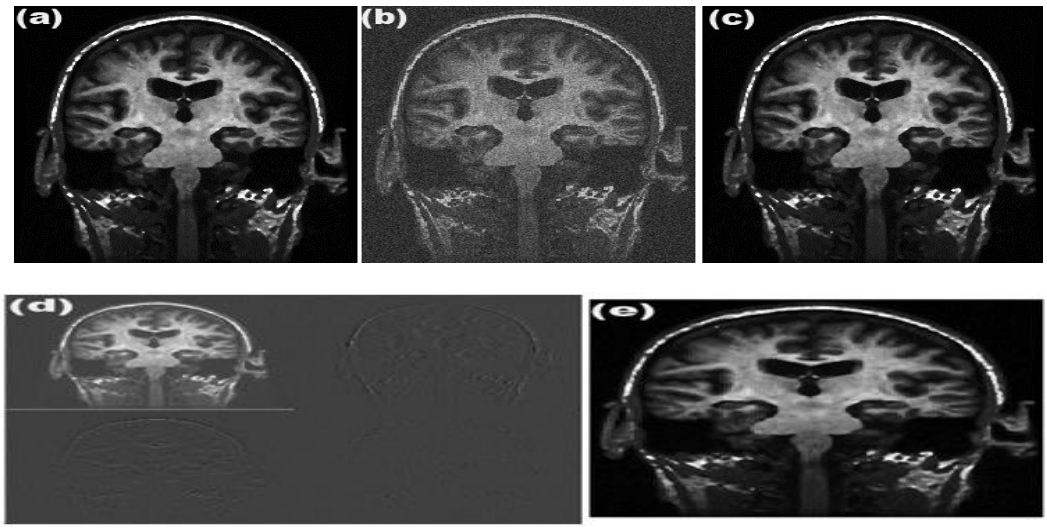

Figure 6. (a) Input Image (b) Noisy (gaussian) Image (c) Median Filtered Image(d) Wavelet Decomposition (e) Inverse Wavelet Transformed Image

5. Conclusion

An attempt has been made to study the noise models like additive random noise, impulse noise, multiplicative noise and haar discrete wavelet transform combination with weiner filter has been presented. An attempt has been made to implement the same in hardware platform and study the performance of the implemented algorithm. Results were compared with several performance metrics like PSNR, Mean square error (MSE), Absolute Mean square error (AMBE), Structural similarity Index (SSIM). It has been implemented in a single board computer (raspberry pi) open source software platform OpenCV.

References

[1] L. Shao, R. Yan, X. Li and Y. Liu, "From Heuristic Optimization to Dictionary Learning: A Review and Comprehensive Comparison of Image Denoising Algorithms", IEEE Transaction On cybernetics, vol. 44, no. 7, July **(2014)**.

[2] A. Kumar Boyat and B. Kumar Joshi, "A Review Paper: Noise Models in Digital Image Processing", An International Journal Signal & Image Processing(SIPIJ), vol. 6, no. 2, **(2015)** April.
[3] H. Talebi and P. Milanfar, "Global Image Denoising", IEEE Transaction on Image Processing, vol. 23, no. 2, **(2014)** February, pp. 755.
[4] P. Drineas and M. W. Mahoney, "On the Nystrom Method for Approximating a Gram Matrix for Improved Kernel-Based Learning", Journal of Machine Learning Research, vol. 6, **(2005)**, pp. 2153-2175.
[5] W. Zuo, Member, IEEE, L. Zhang, Member, IEEE, C. Song, D. Zhang, Fellow, IEEE and H. Gao, "Gradient Histogram Estimation and Preservation for Texture Enhanced Image Denoising", IEEE Transactions on Image Processing, vol. 23, no. 6, **(2014)** June.
[6] A. Varsha and P. Basu, "An improved dual tree complex wavelet transform based image denoising using GCV thresholding", Computational Systems and Communications (ICCSC), 2014 First International Conference on IEEE, **(2014)**.
[7] M. Abdou Bouteldja, M. Baadeche and M. Batouche, "A novel approach for image denoising based on evolutionary game theory", Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on IEEE, **(2014)**.
[8] S. Lahmiri and M. Boukadoum, "Biomedical image denoising using variational mode decomposition", Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE, IEEE, **(2014)**.
[9] K. Gan, J. Tan and L. He, "Non-Local means image denoising algorithm based on edge detection", Digital Home (ICDH), 2014 5th International Conference on IEEE, **(2014)**.
[10] http://dicom.nema.org/.
[11] P. Kamboj and V. Rani, "A brief study of various noise model and filtering techniques", Journal of Global Research in Computer Science, vol. 4, no. 4, **(2013)** April.
[12] https://www.adafruit.com/datasheets/pi-specs.pdf.
[13] G. Bradski and A. Kaehler – Learning OpenCV : Computer Vision with the OpenCV; O'Reilly Media; 1st edition, **(2008)**.
[14] http://www.oasis-brains.org/.
[15] A. Kaur, L. Kaur and S. Gupta, "Image Recognition using Coefficient of Correlation and Structural SIMilarity Index in Uncontrolled Environment", International Journal of Computer Applications (0975 – 8887), vol. 59, no. 5, **(2012)** December.
[16] D. Zhang, H. Gao, W. Zuo, L. Zhang and C. Song, "Gradient Histogram Estimation and Preservation for Texture Enhanced Image Denoising", IEEE Transactions on Image Processing, vol. 23, no. 6, **(2014)** June.
[17] R. S. Gonzalez and P. Wintz, "Digital image processing", **(1977)**.

