



REVUE ALGERIENNE DES SCIENCES

Section -A-

Sciences de la Nature et de la Vie

Sciences techniques

<http://univ-eltarf.dz/fr/>



Fast and efficient Preprocessing Algorithm for Removing Highly Impulsive Noise in Digital Images

Beddad Boucif^{a,*}, Hachemi Kaddour^a, Postaire Jack-Gerard^b

^a Laboratoire Technologies de Communications, Université de Saida Dr Tahar Moulay, Algérie.

^b Centre de Recherche en Informatique, Signal et Automatique de Lille, Université de Lille, France.

Information	Abstract
Keywords : Code Composer Studio Impulsive Noise Median Filter Spatial Information TMS320C6713 DSK, VM3224K2 Daughter Kit *Correspondance : boucif.beddad@univ-saida.dz	This work proposes a new approach to enhance corrupted images. Digital images suffer from different type of degradations. However, the main objective is to remove low and high impulsive noise density using a new developed median filter. The employed filter performs spatial information processing to determine which pixels in the image have been affected by impulsive noise. Thereafter, replace those pixels by the median value of the proposed 2-D areas that have the low variance value. The proposed algorithm has been optimized and implemented on a float-point TMS320C6713 Digital Signal Processor of Texas Instruments. A display module (LCD screen) is also used and connected to the DSP kit to display the output enhanced image. Performance of the proposed algorithm is compared with others nonlinear filters. The improved median filter is successfully tested with multiple grayscale images and gives better Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE) results..

1. Introduction

Image processing is a method for performing certain operations on a digital image to obtain an enhanced image or to extract hidden information that can be interpreted and used for several purposes. Impulse noise is one the most severe noise which usually affects the images. So, the researchers focus on the removal of this impulse noise. The Nonlinear filters have widely been exploited due to their success in removing the impulsive salt-and-pepper noise while preserving the fine details of the image [1] and can be performed more effectively with the front-end and subsequent processes, such as edge detection, segmentation, classification and recognition [3]. The general impulse noise removal process can be expressed in figure 1.

In this research, we will first explain the objective of image preprocessing, where the developed algorithms can reduce the effect of impulse noise and intensity inhomogeneities by using the modified median filter. The rest of this paper is structured as follows: Sections 2, 3 and 4 explain some related methods and materials used in this work. The Section 5 is dedicated to our proposed approach and system design, the hardware implementation is given in section 6. All experimental result and their optimizations

were presented in the Section 7 and 8. Finally general conclusion is noted.

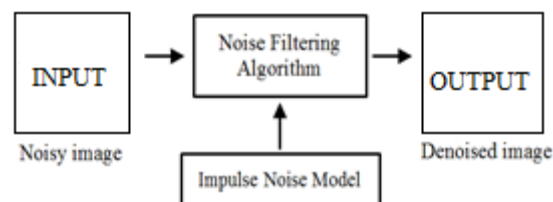


Fig. 1 : A General Impulse Noise Removal Process.

2. Literature Review

Various enhanced median filters are cited in the literature for removal of high density impulse noise in image. Recently, many improved techniques have been considered for that issue: Sharma et al propose a novel recursive median filter that detects noise and then it will be removed by using the two thresholds values (Minimum threshold and maximum threshold) [5]. Also Zheng-yang et al develop an adaptive median filter which determines a possible noise pixel according to the characteristic of the salt and pepper noise [7]. Lots of important enhanced methods have been done and executed as Li et al[8], Jubair et al

[9] [6], Tafti et al. [10], Anitha et al. [11], Sakthivel et al. [12], Varshney et al. [13], Das et al. [14], Arastehfar et al [15].

3. Image De-noising Based on Median Filter

3.1. Impulse Noise Model

Impulse noise is a phenomenon that affects an image during some process like acquisition, formation, storage and transmission. This type of noise is modeled as salt-and-pepper noise or random valued noise where if noisy pixels take just 0 or 255, the impulse noise will be called salt-and-pepper noise. Otherwise, the noise is called random valued noise. The impulse noise is mathematically expressed in the first equation (1):

$$w(i,j) = \begin{cases} R(i,j) & \text{with propability } B \\ I(i,j) & \text{with prpability } 1 - B \end{cases} \quad (1)$$

Where B is Noise density, I is a noise-free image, W represents noisy image and R takes random numbers 0 or 255 generated with the equal probability in the image.

3.2. Median Filter Overview

There are two basic techniques to image de-noising: the spatial filtering and the Transform domain filtering methods; we are very interesting by the first technique. The Standard median filter is one type of spatial filtering technique that replaces the center value in the mask with the median of all pixels values [2]. So in 2-D if X_{ij} is the input, the output Y_{ij} is given by:

$$Y_{ij} = \text{med}\{X_{ij}\} = \text{med}\{X_{(i+r)(j+c)}(r,c) \in A, (i,j) \in I\} \quad (2)$$

The Recursive Median Filter is developed from the median filter using previous output of median filter rather than pixel values of input image; also we can develop it to adaptive filter.

4. Digital Signal Processor Tools

4.1 TMS320C6713 DSK

The TMS320C6713DSK is an embedded platform of low-cost autonomous development and also is based on the high-performance that enables users to evaluate, develop and implement applications or new algorithm on hardware Kit. The simplified architecture of the TMS320C6713DSK is shown in the figure 2.

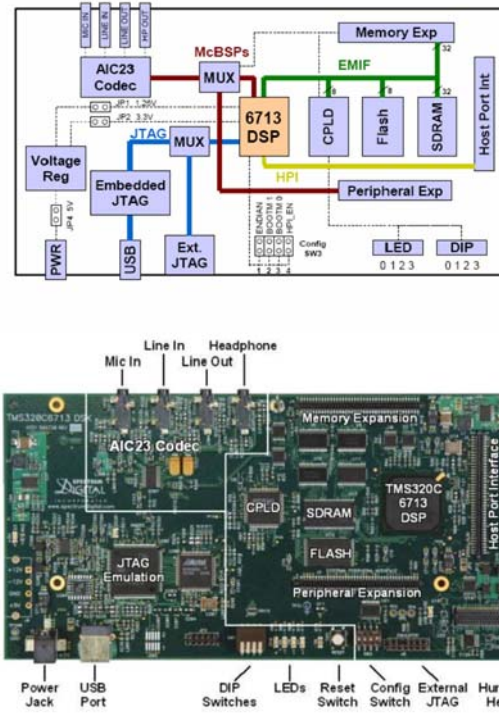


Fig. 2 : TMS320C6713 Development Starter Kit.

This platform contains various components that support some interesting features, the C6713DSK architecture include:

- JTAG (Joint Test Action Group) emulation through on-board JTAG emulator with USB host, external emulator
- A Texas Instruments TMS320C6713 DSP Core 32 Bit floating point processor that operating at 225 MHz
- Software board configuration through the registers implemented in CPLD, HPI boot mode, 32 Bit EMIF.
- 512 KB of non-volatile Flash memory (256 KB usable in default configuration) and 264KB of internal memory
- Standard expansion connectors for daughter card use
- Single voltage power supply (+5V),
- 04 user accessible LEDs and DIP switches, boot options
- An AIC23 stereo codec, Interface or external emulator
- 16 Mbytes of the external synchronous DRAM

4.2 VM3224K2 LCD Daughter Card

The Video Daughter card like VM3224K2 kit is a video in/out hardware module that provides developers to evaluate and develop image processing algorithms based on Texas Instruments' TMS320C6000 DSP Starter Kit. This product VM3224K2 acquires NTSC/PAL

analog video signal and displays digital video data on TFT LCD display screen. The VM3224 is embedded over to see the results of processing done in TMS320C6713 hardware. The daughter card is also compatible with Image processing algorithms, as we are able to display input and output segmented images. The TFT LCD panel uses an RGB565 pixel expression that is 320x240 in size. The LCD panel must provide pixel data periodically according to the pixel array pattern. Thus, the DSP stores image data in the RGB565 format, in order to display it on the TFT LCD. So, output obtained in the project is RGB which is first converted into RGB565 format which is a mandatory for this daughter kit in order to display the output image in the respective kit.

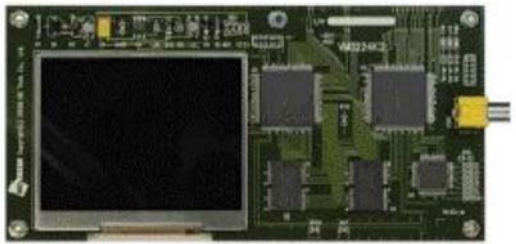


Fig. 3 : VM3224K2 Display Module..

4.3 Embedded Target For Texas Instrument C67x

Embedded IDE Link provides a connection between Matlab and CCS which the Simulink Embedded Target converts our Simulink model into a CCS Project. With IDE we can use Matlab and Simulink to analyze, shape and correct the behavior of running code. Another possible use for automation is to create Matlab scripts that verify and examine the algorithms running in their final run. The Relationship between MATLAB, CCS and DSP Texas Instrument.

5. Proposed Method

In this section, we are going to describe the different steps of our proposed algorithm that performs spatial information processing to determine which pixels in an image have been affected by impulse noise then replace this corrupted pixel only by the median value of selected area that have low variance value. The nine proposed areas are represented as:

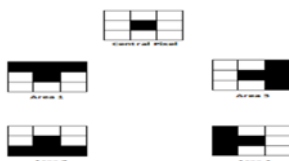


Fig. 4 : The fourth Proposed Neighborhood Areas.

The main various steps involved in our proposed method can be explained as follows:

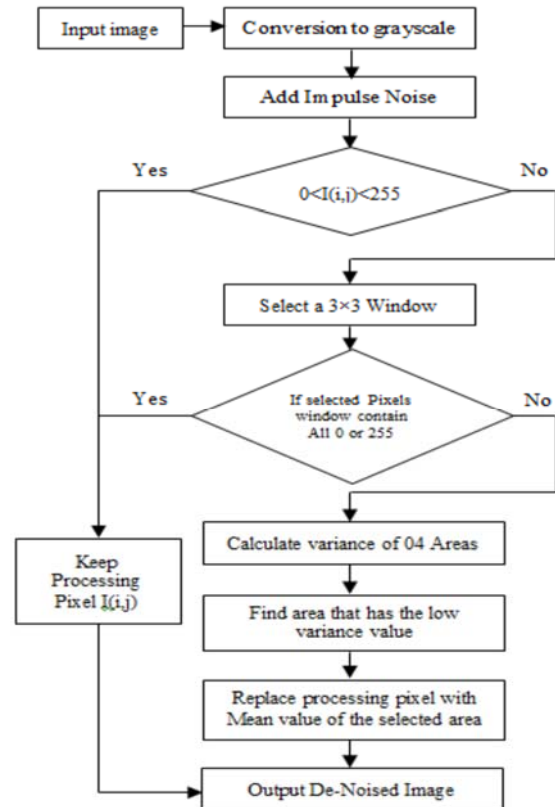


Fig. 5 : Flow Diagram of the Proposed Algorithm.

Step 1: Read noisy grayscale image $I[M \times N]$ that are corrupted by salt and pepper impulse noise.

Step 2: Check the processing pixel value in the whole image, if their value is between 0 and 255 ranges then keep the same processing value $I(i,j)$ else go to step 3.

Step 3: Select a 2-D moving window with 3×3 size.

Step 4: Check a 3×3 selected pixels in the current window, if all these selected pixels have as 0 or 255 values then keep the same processing value $I(i,j)$ else go to step 5.

Step 5: Calculate variance value of the fourth selected areas showed in figure 5.

Step 6: Find the area that has low variance value

Step 7: Replace the current processing pixel $I(i,j)$ with the Mean value of the selected area that have low variance value.

Step 8: Repeat steps two to seven until all pixels are processed and get better de-noised image as final output.

6. Hardware Implementation

In this work, a novel approach is implemented in real time on TMS320C6713 DSP and displayed by the VM3224K2 Daughter card Kit which is fully automatic and does not need any user interaction. Figure7 showed the proposed hardware implementation.

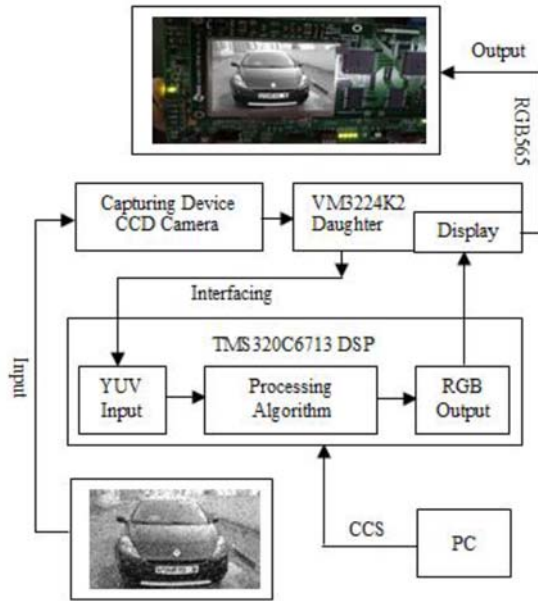


Fig. 6 : Proposed Hardware Implementation.

Once proposed algorithm was developed, it was completely verified in Matlab with multiple input images. Then the in-built functions of MATLAB were replaced by the user-defined functions in Matlab Simulink Model that he uses different Blocksets of "Video and Image

Processing Blocksets" library. The proposed C6713DSP Simulink model is shown in the figure

Blocks	Library	Quantity
Image from file	Sources	01
Median filter	Enhancement	01
Video viewer	Sinks	02
Embedded MATLAB function	User-Defined Functions	05
Image data type conversion	Conversions	04
Color space conversion	Conversions	01
PSNR	Statistics	03
Display	Sinks	06

Table 1. Matlab Simulink Blocks Set Used.

On Matlab Simulink, there is a library containing a specific blocks for each type of DSP card, from this library we add the C6713 block then we select processor type and also clock frequency value. Concerning the passage from DSP Simulink Model to the Code Composer Studio, MATLAB has developed a tool "Link for code composer studio" which allowed him to communicate with the Code Composer Studio for compilation, real time debugging, and linking program in Integrated Development Environment (IDE). So with using these tools, the proposed Simulink is translated into CCS project written in C language that could be compiled using a highly optimizing C/ C++ Compiler and run on the TMS320C6713 DSP. The next step is to compile C code creating, so the Build command will compile all the files that are included in this project and make an executable file then we load the executable file (.out).

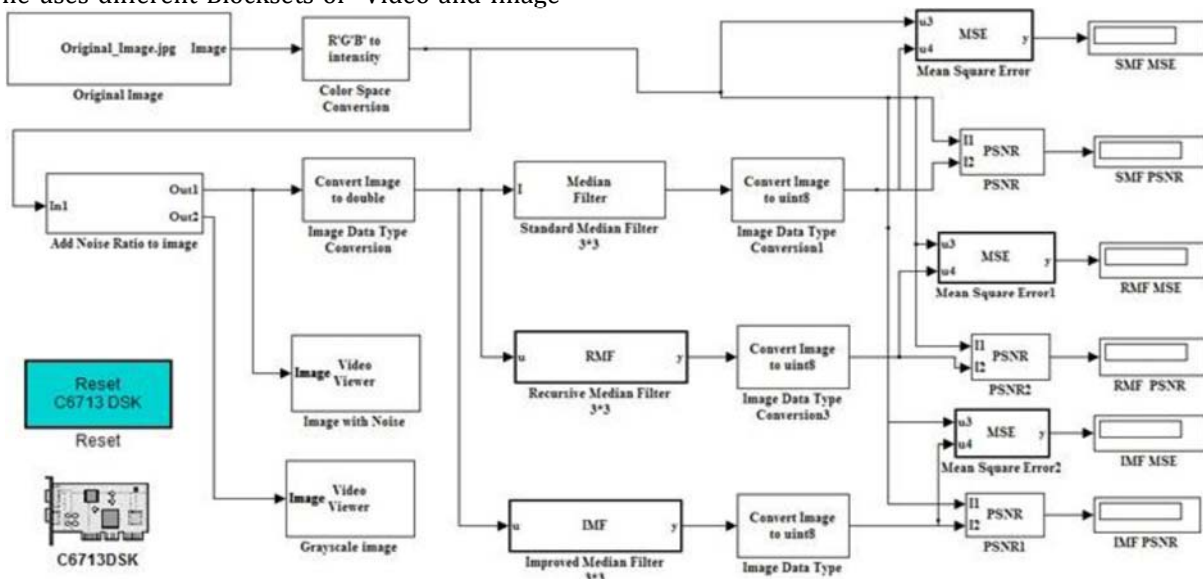


Fig. 7 : Flowchart of the Proposed Simulink Model.

4. Results and Discussion

In this work, two performance indicators have been used for comparison. Tables 2 and 3 show the Peak Signal-to-Noise Ratio and Mean Squared Error results. Figures 7 presents debugging results after successful execution in Code Composer Studio v3.1 and hardware implementation on TMS320C6713 DSP Processor.

In order to measure the quality of the filtered image, a comparison is made by using two performance metrics: Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR).

PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise and it expressed as:

$$PSNR=10\log_{10}\left(\frac{R^2}{MSE}\right) \quad (3)$$

In the previous equation, R equal 255 if we use image as uint8 data. The M and N are the numbers of rows and columns in image. MSE represents the square of the cumulative error and he is calculated according to the following equation:

$$MSE = \frac{\sum_{M,N}[I(i,j)-I_{noisy}(i,j)]^2}{M*N} \quad (4)$$

Peak Signal to Noise Ratio (PSNR)					Mean Square Error (MSE)			
Noise Ratio	SMF	RMF	AMF	IMF (Proposed)	SMF	RMF	AMF	IMF (Proposed)
0.00	27.33	29.50	30.77	32.30	119.29	72.30	53.72	38.03
0.02	26.79	29.12	30.21	32.09	135.01	78.98	61.14	39.99
0.04	26.37	28.70	29.84	30.91	148.81	87.09	66.72	52.51
0.06	25.41	27.71	28.53	29.96	185.81	109.60	90.33	65.46
0.08	25.08	26.84	27.79	28.99	200.82	133.92	107.24	81.88
0.10	24.85	26.96	26.89	28.61	211.98	130.19	132.42	89.31
0.12	24.73	26.26	26.35	28.08	217.53	153.10	150.05	100.88
0.14	24.09	25.63	25.51	27.24	252.76	177.46	181.95	122.49
0.16	23.03	24.66	24.84	26.52	322.32	221.55	212.37	144.54
0.18	22.20	24.49	24.38	26.05	391.11	230.51	236.49	161.26
0.19	21.75	23.75	23.90	25.27	433.27	273.87	263.91	192.87

Table 2. PSNR and MSE Values for Various Noise Density.

The proposed filter IMF is tested with digital images of size 512 x 512. These images are corrupted by Salt and Pepper noise at various noise ratios. For comparison purposes, lot of methods were chosen and indicated as

follows: SMF_ Standard Median Filter, RMF_ Recursive Median Filter [5], AMF_Adaptive Median Filter [7]. Also the experimental results obtained were well compared with the several previously published research methods.

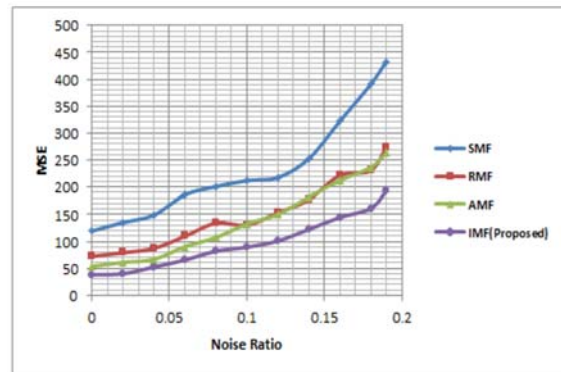
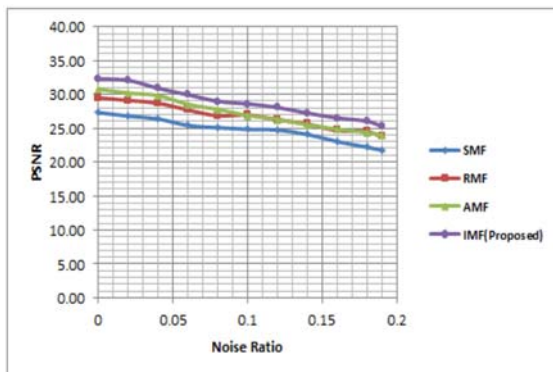


Fig. 8 : Graph comparison in terms of PSNR and MSE.

5. Optimization

For analyzing the performance of the proposed algorithm various optimization techniques were used in term of the DSP implementation as:

- Speed optimization done using Temporary variable to avoid repeated computations
- Code optimization using short data type instead of word data type
- Change Compiler setting to -g -o3.
- Inbuilt Instruction using intrinsic operators like ADD2, SUB2, LDW and STW

Fonctions	No. of CPU Cycles (Without Optimization)	No. of CPU Cycles (With Optimization)
RGB to Intensity	3923732	3026222
Convert image to uint8	3401868	2718414
Convert image to double	3401868	2718414
Add Noise Ratio	727897	648206
Improved Median Filter	6203673	5946233
Total	17659038	15057489

6. Conclusion

In this paper, an enhancement algorithm has been proposed to remove impulsive noise from corrupted digital images. The method is based on an improved median filter. The proposed scheme has been implemented practically on TMS320C6713 DSP and VM3224K2 Daughter Kit using integrated development environment (IDE) via code composer studio. The experiment results carried out are consistent and satisfactory compared with other nonlinear filter methods..

Référence :

- [1] S. Malviya and H. Amhia, "Image Enhancement Using Improved Mean Filter at Low and High Noise Density", *International Journal of Emerging Engineering Research and Technology*, vol. 2, No. 3, pp. 45-52, 2014.
- [2] G. Qiaoman, W. Guoxin and X. Xiaoli, "Application of adaptive median filter and wavelet transform to dongba manuscript images denoising", in *13th International Conference on Electronic Measurement & Instruments, ICEMI'20 7*, Oct 20, 20 7 - Oct 22, 2017, Yang-zhous, China, 2017.
- [3] M. Hsieh, et al, "Fast and efficient median filter for removing 1-99% levels of salt-and-pepper noise in images", *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1333-1338, 2013.
- [4] B. Sharmila, N. Karalan, D. Nedumaran, "Image Processing on DSP Environment Using OpenCV," *IJA CSSE. Image Process.* Vol 5, pp. 489~493, February 20 5.
- [5] A. Sharma, V. Chaurasia, "Image emoval of High Density Salt-And-Pepper Noise by ecursive Enhanced Median Filtering", *2nd International Conference on Emerging Technology Trends in Electronics Communication and Networking*,

- ET2ECN2014, Dec 26, 2014 - Dec 27, 2014, Surat, India, 2014.
- [6] M. I. Jubair, et al., " An Enhanced Decision Based Adaptive Median Filtering Tech-nique to Remove Salt and Pepper Noise in Digital Images", in *14th International Confer-ence on Computer and Information Technology, ICCIT'20* , Dec 22, 20 - Dec 24, 2011, Dhaka, Bangladesh, 2011.
- [7] G. Zheng-yang and Z. Le, "Fast Multi-exposure Image Fusion with Median Filter and Re-cursive Filter", *Tenth International Conference on Computational Intelligence and Securi-ty, CIS2014*, Canada, 2014, pp 44-46.
- [8] S. Li and X. Kang, "Image Enhancement Using Improved Mean Filter at Low and High Noise Density", *IEEE Transactions on Consumer Electronics*, vol. 58, No. 2, pp. 626-632, 2012.
- [9] M. I. Jubair, et al, " Evaluationof An Improved Adaptive Filtering Technique to emove High Density Salt-and-Pepper Noise Using Multiple Last Processed Pixels", *Global Jour-nal of Computer Science and Technology Graphics & Vision*, vol. 12, No. 14, pp. 21-26, 2012.
- [10] A. D. Tafti and E. Mirsadeghi, "A Novel Adaptive ecursive Median Filter in Image Noise Reduction Based on Using the Entropy": in *International Conference on Control System, Computing and Engineering, CSCE'20 2*, Nov 23, 20 2 - Nov 25, 2012, Penang, Malaysia, 2012, pp. 520-523.
- [11] S. Anitha and V. adha., "Preprocessing using Enhanced Median Filter for Defect Detec-tion in 2D Fabric Images", *International Journal of Engineering and Technology*, vol. 6, No. 2, pp. 707-717, 2014.
- [12] N. Sakthive, L. Prabhu, "Mean - Median Filtering For Impulsive Noise emoval", *Inter-national Journal of Basic and Applied Science*, vol. 2, No. 4, pp. 47-57, 2014.
- [13] P. Varshney and A. Tyagi, "An Enhanced ecursive Median Filter for Image Denoising", *International Journal of Advanced Research in Computer Science*, vol. 6, No. 1, pp. 201-203, 2015.
- [14] J. Das, et al, " Removal of Salt and Pepper Noise Using Selective Adaptive Median Filter," in *International Conference on Accessibility to Digital World, ICADW2016*, Dec 16, 2016 - Dec 18, 2016, Guwahati, India, 2016.
- [15] S. Arastehfar, A. A. Pouyan and A. Jalalian, "An enhanced median filter for removing noise from M images", *Journal of AI and Data Mining*, vol. , No. , pp. 3-17, 2013.