**Automatic Noise Detection and Removal**

**Using Hybrid Filter**

**A Mini Project Report**

Submitted in fulfilment of the requirements for

the award of the degree of

**INTEGRATED MASTER OF SCIENCE**

**In**

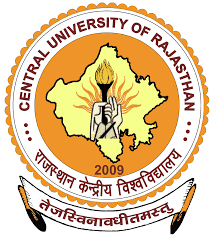
**COMPUTER SCIENCE**

**Submitted by**

Nitesh Sukhwani 2015IMSCS010

Under **the Guidance of**

Mr. Raviraj Choudhary

****

**Department of Computer Science**

**School of Mathematics, Statistics and Computational Science**

**CENTRAL UNIVERSITY OF RAJASTHAN**

**MAY-2018**

Declaration

I hereby declare that the project entitled **Automatic Noise detection and Removal using Hybrid filter** submitted for the Integrated M.Sc. (CS) degree is my original work conducted under the guidance of Mr. Raviraj Choudhary.

I further declare that to the best of my knowledge the project does not contain any part of any work that has been submitted for the award of any degree either in this university or in any other university without proper citation.

Name : Nitesh Sukhwani

Enrollment no.: 2015IMSCS010

Department of Computer Science

This is to certify that the statement made above by the candidate is true to the best of my knowledge.

--------------------

(Supervisor)

**ACKNOWLEDGEMENT**

I would like to take this opportunity to express my sincere gratitude towards my supervisor **Mr. Raviraj Choudhary**  for his invaluable guidance during the progress from the beginning to the completion of this project and all the encouragement and moral support.

I also wish to thank all the faculty members of Department of computer science , my friends who gave me valuable knowledge through their lectures. At this place, I have got plenty of fundamental and advanced knowledge for my career.

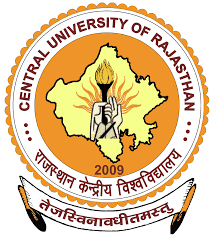
Nitesh Sukhwani

Central University of Rajasthan

Abstract

Image denoising is the manipulation of the image data to produce a visually high quality image. At present there are a variety of methods to remove noise from digital images. There are different types of filters like mean filter, median filter, bilateral filter, wiener filter etc. to remove a single type of noise such as salt and pepper noise, speckle noise, Gaussian noise etc. But if the image is corrupted with an unknown type of noise then we do not know which filters to use to remove the noise exactly. In this project we try to make a hybrid filter to detect the type of noise us ing the statistical behavior of the processed image to detect Noise type image and according to that a suitable filter will be applied to the image for denoising. The software used for simulation is MATLAB R2015a.

**Keywords:** bilateral filter, wiener filter, salt and pepper noise, speckle noise, gaussian noise.

****

**TABLE OF CONTENTS**

Title page

Declaration

Acknowledgement

Abstract

1. **Introduction**
   1. Overview
   2. Objective
   3. Significance
2. **Theory**
   1. Types of Noise
   2. Denoising algorithm used
3. **Methodology** 
   1. For noise type identification
   2. Hybrid filter for Salt & pepper noise
   3. Find best suitable existing filters
4. **Implementation**
   1. Identify the type of noise
   2. Hybrid filter for S&P noise
   3. comparative study of various filter
5. **Output**
   1. Objective 1 and objective 3
   2. Objective 2
6. **Conclusion**
7. **Future Work**
8. **Reference**

**INTRODUCTION**

**Overview:**

Digital images play an important role both in daily life applications such as satellite television, magnetic resonance imaging, computer tomography, geographical information systems and astronomy.

An automated technique for identification of image noise is really important because once the type of noise is identified from the given image an appropriate algorithm can then be used to de-noise it.

The method in this project uses a simple technique for identifying the type of noise present in an image which is required in many applications that call for superior de-noising performance. Thus denoising is often a necessary and the first step to be taken before the images are analyzed to extract the useful features from an image. Since poor de-noising often results from poor noise identification, a better noise identification technique is always preferred.

**Objective:**

1. In this project our objective is to identify the type of the noise with which image is corrupted for instance we have considered only three type of noise i.e Gaussian , speckle and salt & pepper noise.
2. If image is corrupted with salt and pepper noise it will detect the percentage of image corrupted with that noise and also identify the noisy pixel so based on that a hybrid filter will be used get denoised image.
3. Find the best suitable existing filter for the denoising using the comparative study of the PSNR values of denoised image.

**Significance:**

1. As in real life we do not know the type of the noise with which image is corrupted so this method is detect the noise type .
2. If we know the type of the noise we can apply best suitable filter or hybrid filter to get good quality denoised image.
3. If image is corrupted with salt and pepper noise it will detect the percentage of image corrupted with that noise and also identify the noisy pixel so based on that a hybrid filter will be used get denoised image.

**Theory**

**Type of noise:**

Noise is a disturbance that affects a signal and may distort the information carried by the signal. Many types of noises exist today. They are mainly classified as follows :

**Additive Noise**

The additive noise is primarily caused by thermal noise (fundamental noise), which comes from the reset noise of capacitors. Thermal noise is a random fluctuations present in all electronic systems.

The mathematical model given for additive noise type is:

f(i, j) = y(i, j) + w(i, j) (1)

where 1 ≤ i ≤ M, 1 ≤ j ≤ N.

Let M and N be the size of the original image y(i, j), w(i, j) be the noisy image and f(i , j) is the noisy image.

**Multiplicative Noise**

This kind of noise is also called as the speckle noise. This noise gives a magnified view of the area and there is a higher random variation observed. On the other hand, when this noise is applied to a darker region in the image, the random variation observed is not that much as compared to that observed in the brighter areas. Thus, this type of noise is signal dependent and distorts the image in a large way.

The mathematical model for multiplicative noise type is:

f(i, j) = y(i, j) \* w(i, j) (2)

where 1 ≤ i ≤ M, 1 ≤ j ≤ N

Let M and N be the size of the original image y(i, j), w(i, j) be the

noisy image and f(i , j) is the noisy image.

**Impulsive Noise**

Impulsive noise is sometimes called as salt-and-pepper noise or spike noise. This kind of noise is typically seen on digital images. It represents itself as randomly occurring white and black pixels. An image containing this type of noise will have dark pixels in bright regions and bright pixels in dark regions. It can be caused by dead pixels, analog-to-digital converter errors, bit errors in transmission.

*0 with probability r*

F ( I , J ) =

*255 with probability 1 – r*

**Denoising algorithm Used:**

There are many approaches to image denoising, but in this project we for finding most efficient existing filter we chooses mainly five filtering techniques which are explained below:

**1. Gaussian filter**

Gaussian filter is a filter whose impulse response is a Gaussian function. Gaussian filters are designed to give no overshoot to a step function input while minimizing the rise and fall time. This behavior is closely connected to the fact that the Gaussian filter has the minimum possible group delay. Mathematically, a Gaussian filter modifies the input image by convolution with a Gaussian function. The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian ('bell-shaped') hump.

**2 Adaptive median filter**

The adaptive median filtering has been introduced as an improvement to the standard median filtering, as we explained before that the Median filtering can detect the noise but in the same it can't differentiate between the fine details and the noise. So the main idea in the Adaptive Median Filter is to perform a spatial processing to determine which pixels in an image have been affected by impulse noise, and run the filter only in this pixel. The Adaptive Median Filter classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test.

**3.Median filter**

The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. The median of the pixel values in the window is computed, and the center pixel of the window is replaced with the computed median. Median filtering is done by, first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

**4.Homomorphic Filter**

Homomorphic filtering involves a nonlinear mapping to a different domain in which linear filter techniques are applied, followed by mapping back to the original domain. It simultaneously normalizes the brightness across an image and increases contrast. Here homomorphic filtering is used to remove multiplicative noise. Homomorphic filter despeckling methods take the advantage of logarithmic transformation, which converts multiplicative noise to additive noise. High boost butter worth filter is used here to reject the resulting additive noise.

**5.Wiener Filter**

Wiener filters are a class of optimum linear filters which involve linear estimation of a desired image data sequence from another related sequence. The wiener filtering method requires the information about the spectra of the noise and the original image and it works well only if the underlying signal is smooth. Wiener method implements spatial smoothing and its model complexity control correspond to choosing the window size [5- 8]. Wiener filtering is performed on the wavelet coefficients. In this model it is assumed that the wavelet coefficients are conditionally independent Gaussian random variables. The noise is also modeled as stationary independent zero-mean Gaussian variable

**Methodology:-**

**For noise type identification**

The basic principle consists of three main steps to identify the noise type.

**Step 1** : Extract respective noise samples from the given noisy image

**Step 2 :** Estimate statistical features such as Kurtosis & Skewness

**Step 3 :** Use a pattern classifier to identify the type of noise

**Step 1 : Extract respective noise sample**

Assume the original M×N image y(i, j) is contaminated by either additive or multiplicative type of noise. To start with it has been assumed that the type of noise is unknown, but it belongs to one of N known classes. For each type of noise, choose a simple linear or nonlinear spatial filter operator capable of removing most of the noise type from the image. first process the image through each filter operator to obtain:

gk(i,j) = Hk(i,j) f(i,j), 1≤ k ≤ N

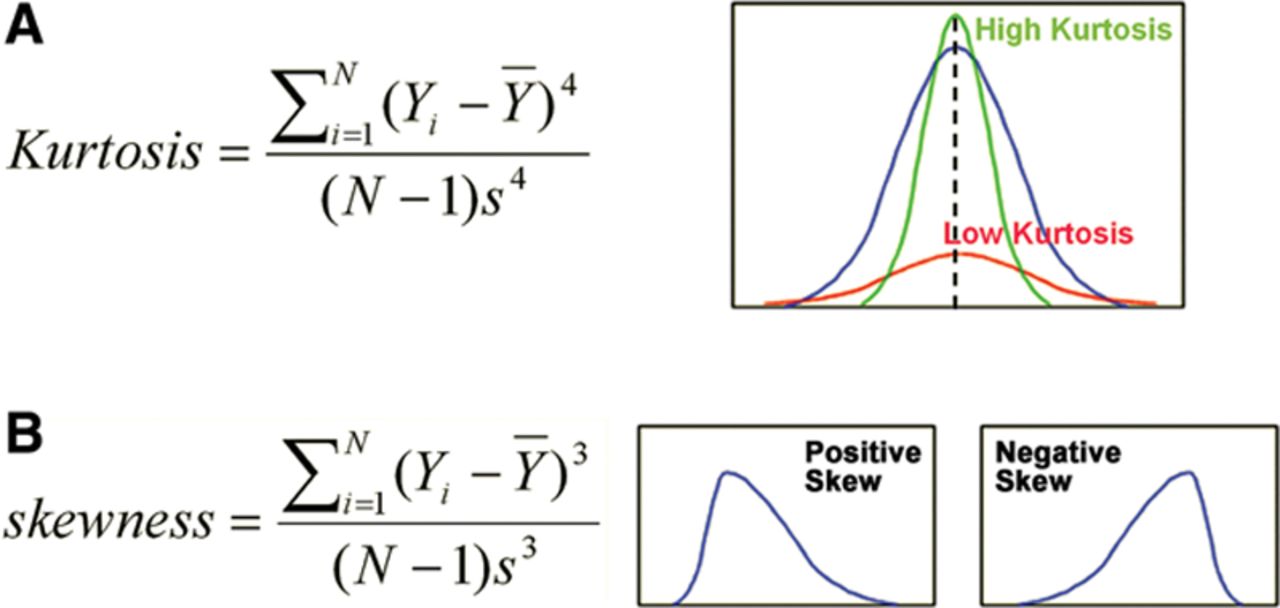
Next, estimate some simple statistical based features from wk(i,j), 1≤ k ≤ N, and then classify the noise into one of N known classes using similarity measures.

We select the filter wiener, homomorphic and median as they are best suited for Additive(gaussian), multiplicative and impulse type of noise respectively and then substract the original image from filtered image.

**Step 2: Estimate the statistical feature**

Estimate the statistical feature such as skewness and kurtosis of the given Noise sample image.

The 4th order moments Kurtosis and 3rd order moments skewness of a random variable Y is defined as



**Step 3: Find a pattern classifier to identify the type**

Suppose ω (i, j) is Salt-and-Pepper noise, then *Median filter*, should give the best estimate of y(i, j) and therefore ω = *f (I,j) - y* (*i j*) should be close to Salt-and-Pepper noise. A pattern classifier based on measuring the similarity of Kurtosis and Skewness values is used to evaluate how close *Wiener* ω is to Gaussian white noise, *Median* ω is to salt-and pepper noise and *Homo* ω is to speckle noise. To measure the similarities:

* First generate training sequences of Salt-and-Pepper noise.
* Then, filter each noise sequence through the Median filter.
* Then estimate the Kurtosis and Skewness of each filtered noise sequence and compute their average to yield the reference values of Kurtosis and Skewness for the Salt-and-Pepper noise
* the final measure of similarity between *Median* ω and actual Salt-and-Pepper noise. The same procedure is carried out to obtain the expected Kurtosis and Skewness values for Speckle noise and Gaussian white noise and based on these data we can identify the type of noise.

**Hybrid filter for Salt and Pepper Noise**

If our image is corrupted with S&P noise type the proposed algorithm calculate the percentage of noise image is contaminated with i.e. identify the noisy pixels and based on that processing can be done on only noisy pixel so it will give great results.

1. I = Noisy image
2. A = a matrix of same size of image with all zero values.
3. S= maximum allowed window size
4. R= starting window size
5. Count0 = no. of black pixel
6. Count1 = no. of white pixel

**Proposed algorithm:**

for identification of noisy pixels and percentage of noise

1. Take matrix of size R\*R
2. If ( A (i , j) != 1)
3. If (count0 + count1 != R^2) && ( I(i,j) == 0 || I (i , j) == 255)

A(i,j) = 1;

4. Repeat step 1,2 & 3 for entire image.

5. Increase window size while R<=S

6. Repeat step 1 to 5

7. Count = (no of 1’s in A matrix) / (resolution of image)

8. Return count and A;

**For removing the noise identified**

1. If ( A( i , j ) == 1)

{

X=median ( R )

If ( X!=0 && X!=255 )

{

I ( i, j )==X;

break();

}

Else

{

Increase window size while(R<=S)

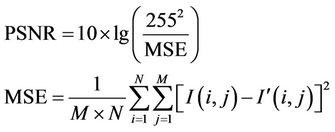
}

}

1. Repeat step 1 for entire pixels

**4.3 Find best suitable existing filter**

In this we denoised image using various filters such as median , wiener, adaptive median, gaussian and homomorphic filters. The quality of the reconstructed image is specified in terms of the Peak Signal-to-Noise Ratio (PSNR) which is calculated using the formula



Results show that the wiener give the best PSNR value for speckle type of noise , wiener is best suited for additive gaussian noise and adaptive median filter is best existing filter for salt and pepper noise.

**Implementations :-**

Objective 1: for identify the type of noise

I=imread('C:\Users\Nitesh Sukhwani\Desktop\10.jpg');

I=imresize(I,[255,255]);

%J=rgb2gray(I);

J=I; % for already grayscale imagee

K=imnoise(J,'salt & pepper',0.28);

K=im2double(K);

L=wiener2(K);

M=medfilt2(K,[3,3]);

% code for homomorphic filter

im=im2double(K);

fftlogim=fft2(log(im+0.01));

f=butter\_hp(I,15,1);

c=fftlogim.\*f;

h=real(ifft2(c));

h1=exp(h);

f1=abs(h1);

fm=max(f1(:));

N=(f1/fm);

L1=K-L;

M1=K-M;

for i=1:255

for j=1:255

N1(i,j)=K(i,j)/N(i,j);

end

end

%%---------------------------------------------

% calculating kurtosis and skewnss

L1\_kurt=my\_kurt(L1);

M1\_kurt=my\_kurt(M1);

N1\_kurt=my\_kurt(N1);

L1\_skew=my\_skew(L1);

M1\_skew=my\_skew(M1);

N1\_skew=my\_skew(N1);

% identify and removal of noise

if(M1\_kurt > N1\_kurt)

disp('image is affected with salt and pepper noise');

[x,y]=getnoise(K);

disp('NOise percentage =');

disp(y);

L = filter(K,x);

figure,

imshow(L);title('filtered image');

end

if(N1\_kurt > M1\_kurt)

disp('image is affected with gaussian noise');

K=wiener2(K);

figure,

imshow(K);title('filtered image');

end

[a,b]=getnoise(I);

if((b < 0.4)&&(M1\_kurt<N1\_kurt<L1\_kurt))

disp('Image is affected with speckle noise');

end

**Objective 2:Hybrid filter for salt and pepper noise**

1. Identification of noisy pixels and calculation of percentage of noise

function [a,b]= getnoise(R)

[r,s]=size(R);

K = [];

for i = 1 : r

for j = 1 : s

K(i,j) = 0;

end

end

for i = 2 : (r-1)

for j = 2 : (s-1)

count0 = 0;

count1 = 0;

for k = i - 1 : i + 1

for l = j - 1 : j + 1

if R(k,l) == 0

count0 = count0 + 1;

else

if R(k,l) == 255

count1 = count1 + 1;

end

end

end

end

if count0 > 0 || count1 > 0

if count0 + count1 ~= 9 && (R(i,j) == 0 || R(i,j) == 255)

K(i,j) = 1;

end

end

end

end

for i = 3 : (r-2)

for j = 3 : (s-2)

if K(i,j) == 0

count0 = 0;

count1 = 0;

for k = i - 2 : i + 2

for l = j - 2 : j + 2

if R(k,l) == 0

count0 = count0 + 1;

else

if R(k,l) == 255

count1 = count1 + 1;

end

end

end

end

if count0 > 0 || count1 > 0

if count0 + count1 ~= 25 && (R(i,j) == 0 || R(i,j) == 255)

K(i,j) = 1;

end

end

end

end

end

for i = 4 : (r-3)

for j = 4 : (s-3)

if K(i,j) == 0

count0 = 0;

count1 = 0;

for k = i - 3 : i + 3

for l = j - 3 : j + 3

if R(k,l) == 0

count0 = count0 + 1;

else

if R(k,l) == 255

count1 = count1 + 1;

end

end

end

end

if count0 > 0 || count1 > 0

if count0 + count1 ~= 49 && (R(i,j) == 0 || R(i,j) == 255)

K(i,j) = 1;

end

end

end

end

end

count = 0;

for i = 2 : (r-1)

for j = 2 : (s-1)

if K(i,j) == 1

count = count + 1;

end

end

end

a=K;

b=(count ./ ((r-1) .\* (s-1)));

end

1. Filtering for noisy pixel to get denoised image

function a=filter(R,K)

[r,s]=size(R);

A = [];

%I2 = I1;

for i = 2 : (r-1)

for j = 2 : (s-1)

a = 1;

if(K(i,j) == 1)

for k = i-1: i+1

for l = j-1: j+1

A(a) = R(k,l);

a = a + 1;

end

end

b = median(A);

if(b ~= 0 && b ~= 255)

R(i,j) = b;

K(i,j) = 0;

end

end

end

end

A = [];

for i = 3 : (r-2)

for j = 3 : (s-2)

a = 1;

if(K(i,j) == 1)

for k = i-2: i+2

for l = j-2: j+2

A(a) = R(k,l);

a = a + 1;

end

end

b = median(A);

if(b ~= 0 && b ~= 255)

R(i,j) = b;

K(i,j) = 0;

end

end

end

end

A = [];

for i = 4 : (r-3)

for j = 4 : (s-3)

a = 1;

if(K(i,j) == 1)

for k = i-3: i+3

for l = j-3: j+3

A(a) = R(k,l);

a = a + 1;

end

end

b = median(A);

if(b ~= 0 && b ~= 255)

R(i,j) = b;

K(i,j) = 0;

end

end

end

end

a=R;

end

objective3: comparative study of various filter using PSNR value of denoised image

I=imread('C:\Users\Nitesh Sukhwani\Desktop\10.jpg');

I=imresize(I,[255,255]);

%J=rgb2gray(I);

J=I; % for already grayscale imagee

J=imnoise(J,'speckle');

a1=medfilt2(J);

a2=adpmedian(J,5);

a3=wiener2(J);

h = fspecial('gaussian',5,1);

a4=conv2(J,h);

im=im2double(J);

fftlogim=fft2(log(im+0.01));

f=butter\_hp(I,15,1);

c=fftlogim.\*f;

h=real(ifft2(c));

h1=exp(h);

f1=abs(h1);

fm=max(f1(:));

a5=(f1/fm);

b1=PSNR(a1,I);

b2=PSNR(a2,I);

b3=PSNR(a3,I);

b4=PSNR(a4,I);

b5=PSNR(a5,I);

1. Function to calculate PSNR value

function [x] = PSNR(a,b)

sum=double(0);

[r,s]=size(a);

for i =2 : r-1

for j = 2 : s-1

sum = sum + (double(a(i,j))-double(b(i,j)))^2;

end

end

mse = sum ./ (r \* s);

%mse=sqrt(mse);

disp(mse);

x = 10 \* log10( double(256)^2 / mse);

end

1. Function for adaptive median

function f = adpmedian(g, Smax)

if (Smax <= 1) | (Smax/2 == round(Smax/2)) | (Smax ~= round(Smax))

error('SMAX must be an odd integer > 1.')

end

[M, N] = size(g);

% Initial setup.

f = g;

f(:) = 0;

alreadyProcessed = false(size(g));

% Begin filtering.

for k = 3:2:Smax

zmin = ordfilt2(g, 1, ones(k, k)); %order-statistic filtering.

zmax = ordfilt2(g, k \* k, ones(k, k));

zmed = medfilt2(g, [k k]);%median filtering.

processUsingLevelB = (zmed > zmin) & (zmax > zmed) & ...

~alreadyProcessed;

zB = (g > zmin) & (zmax > g);

outputZxy = processUsingLevelB & zB;

outputZmed = processUsingLevelB & ~zB;

f(outputZxy) = g(outputZxy);

f(outputZmed) = zmed(outputZmed);

alreadyProcessed = alreadyProcessed | processUsingLevelB;

if all(alreadyProcessed(:))

break;

end

end

% Output zmed for any remaining unprocessed pixels. Note that this

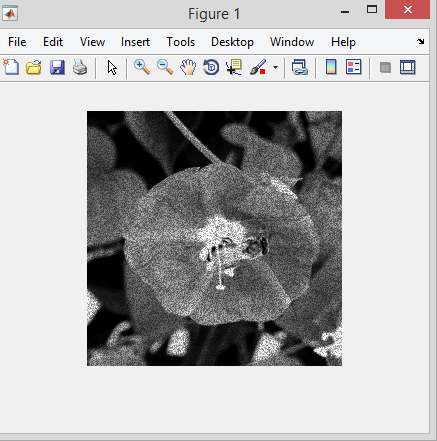
% zmed was computed using a window of size Smax-by-Smax, which is

% the final value of k in the loop.

f(~alreadyProcessed) = zmed(~alreadyProcessed);

**Output : (objective 1 and 3)**

**Input 1**

****

**Input Image(affected with speckle noise)**

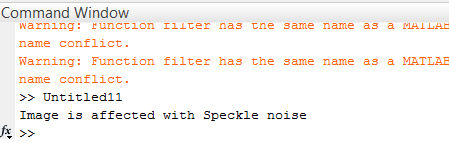
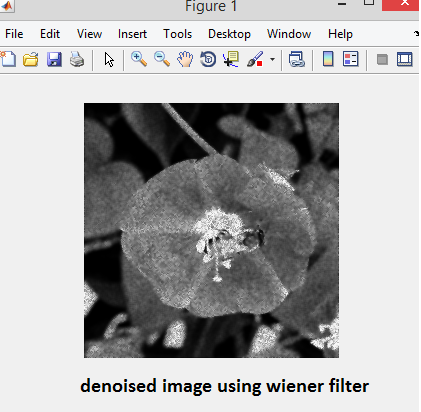
****

TABLE: PSNR value for differ filters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Filter | Median | adpmedian | wiener | gaussian | mean | homomorphic |
| PSNR | 17.66 | 17.10 | 17.80 | 17.48 | 13.58 | 10.87 |



**INPUT 2**

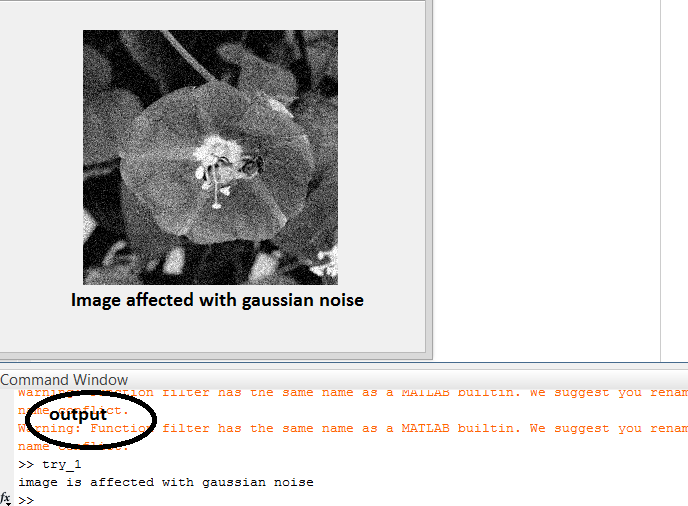
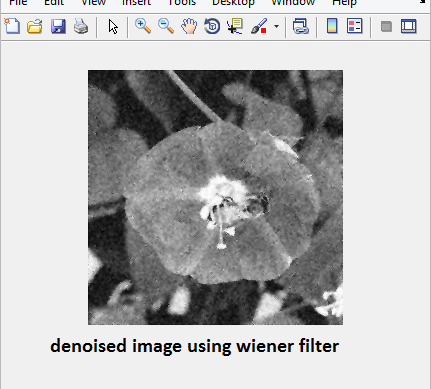


TABLE: PSNR value for differ filters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Filter | Median | adpmedian | wiener | gaussian | mean | homomorphic |
| PSNR | 13.34 | 13.08 | 13.36 | 13.26 | 12.63 | 11.86 |



**INPUT 3**

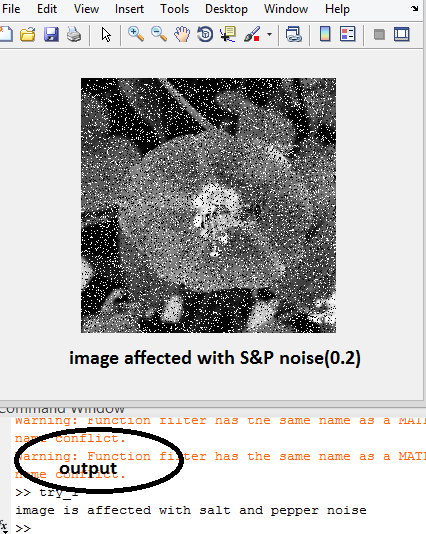
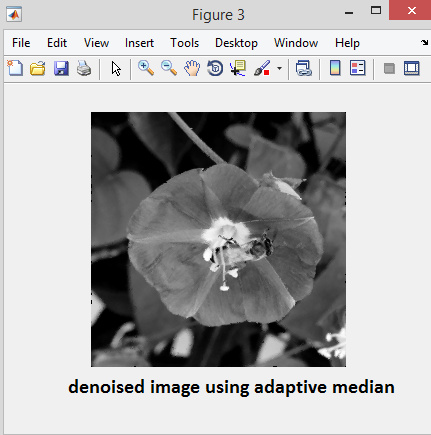


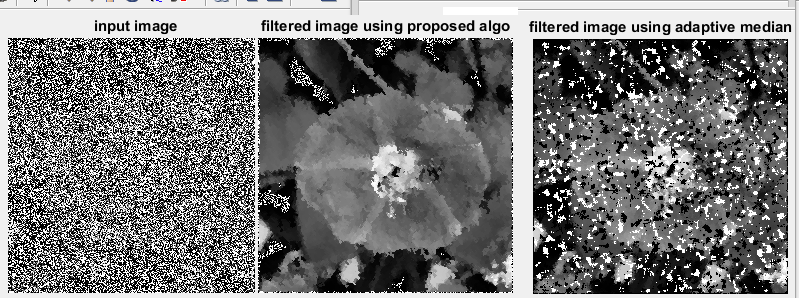
TABLE: PSNR value for differ filters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Filter | Median | adpmedian | wiener | gaussian | mean | homomorphic |
| PSNR | 17.86 | 18.10 | 14.01 | 15.26 | 16.27 | 10.89 |



**Objective 2**

**INPUT 1**

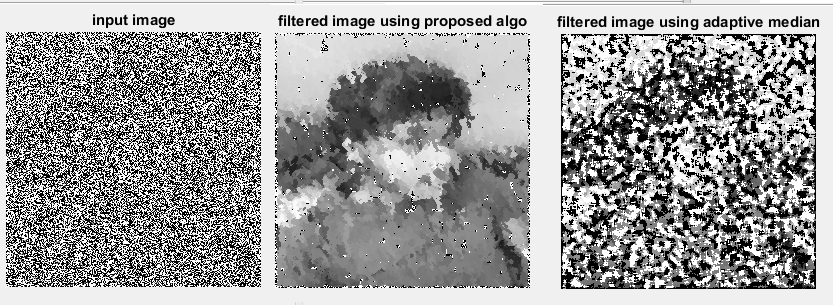


Noise=80% PSNR = 17.37 Noise detect=77.24% PSNR= 11.34

**INPUT 2**

****

Noise=60% PSNR=26.02 Noise detect=59.44% PSNR=21.28



Noise =93% PSNR=17.06 noise detect=88.48% PSNR=7.62

**Conclusion :-**

The experiments have shown that this algorithm provided the most favorable results by means of the PSNR improvements and visual observation over some benchmark methods. The proposed method appears to be better or competitive with some other recently developed techniques. In addition, the computational procedure and speed of the proposed method is very easy for computer implementation, and can be extended for filtering other types of noise in images.

Our comparative tests with 6 standard filters have shown that Wiener is best suitable for both speckle and additive gaussian noise and adaptive median is best existing filter for salt & pepper noise.

**Future Work:-**

1. The noise identification technique is working for limited parameters of gaussian type of noise ( 0 - 0.3) and default parameter of speckle noise so it can be extended to entire range of parameter.

2 In Comparative study for denoising with have taken into consideration only 6 filter this can also be extended to more no. of filter in order to best suitable existing filter.

3.Denoising of S&P affected images using proposed method is not working properly on binary image i.e. image with only two intensity value 0 & 255. So it can be extended to overcome this problem.

**References:-**

* Yixin Chen, Manohar Das, “ An Automated Technique for Image Noise Identification Using a Simple Pattern Classification Approach”, IEEE Trans, 2007
* Dr. P. Subashini , Bharathi.P. T “Automatic Noise Identification inImages using Statistical Features “
* M. C. Motwani, M. C. Gadiya, R. C. Motwani, F. C. Harris, Jr.,” Survey of image denoising techniques”, Proceedings of GSPx 2004, Santa Clara, CA, September 27-30, 2004.
* Amira A. Mahmoud, S. EL Rabaie, T.E Taha “Comparative Study of Different Denoising Filters for Speckle Noise Reduction in Ultrasonic B-Mode Images”
* Suman Shrestha “ Image denoising using adaptive based median filter”.
* Y. Zhao, D. Li, Z. Li, “Performance enhancement and analysis of an adaptive median filter”