Applying Non-Linear Filters to Image With Different Types of Noises

Ravindranath Reddy 19BCE0474 Yelamanchi Jahnavi 20BCE0413

November 25, 2021

Contents

1 Abstract

Image pixels get damaged with noises that occur due to transmission errors, malfunctioning pixel elements in the camera sensors, faulty memory locations, and timing errors in analog-to- digital conversion. Our aim is to compare removal of various types of noises along with standard averaging filters and a hybrid median filter. We will be demonstrating that hybrid median filter provides best filtering for taken images. For all window filters there is some problem. That is **edge treating.** If you place window over an element at the edge, some part of the window will be *empty*. To fill the gap, signal should be extended. For this purpose, we have to perform various pre-processing operations to extend the image.

2 Literature Review

2.1 Survey of the Existing Models/Work

2.1.1 Removing Gaussian noise

This gives an algorithm that initially calculates the amount of noise from the image. In the second stage, the centre pixel is replaced by the mean value of the some of the surrounding pixels based on a threshold value. Noise removal with edge preservation and computational complexity are two conflicting parameters. The proposed method is a solution for these requirements. The performance of the algorithm is tested and compared with standard mean filter, wiener filter, alpha trimmed mean filter K- means filter, bilateral filter and recently proposed trilateral filter. Experimental results show the superior performance of the proposed filtering algorithm compared to the other standard algorithms in terms of both subjective and objective evaluations.

2.1.2 Removing Salt and Pepper (Impulse) noise

This paper discusses various techniques to remove salt and pepper noise. This paper illustrates some of the common averaging filters that we have learned in the curriculum. After Mean filter the modified Mean-Median is introduced as a new filter which is used to remove the efficient noise from the image. To overcome the Mean-Median filter the SMF is used for removing impulse noise. SUMF is used to remove high density salt & pepper noise from digital images. WMF removes the discontinuity in underlying regression function whereas the Decision Based Filter is efficient to de-noise the low level, it fails in medium and high density because the restored pixel which is the median value of the selected window is also a corrupted pixel value. To overcome this problem the new weighted median filter was introduced which gives satisfactory result in reducing high level salt and pepper noise.

2.1.3 Removing Poisson (Shot) noise

This paper uses the method of regulating the variation of pixel intensities to reduce Poisson noise. This new model uses total variation regularization which also preserves edges. The result of this method is that the strength of the regularization is signal independent which is same as Poisson noise. This model preserves low contrast features in regions of low intensity.

2.2 Ambiguities identified in Survey

For all window filters there is some problem. That is **edge treating.** If you place window over an element at the edge, some part of the window will be empty. To fill the gap, signal should be extended. For hybrid median filter there is good idea to extend image symmetrically. In other words we are adding lines at the top and at the bottom of the image and add columns to the left and to the right of it. A hybrid median filter has the advantage of preserving corners and other features that are eliminated by the 3 x 3 and 5 x 5 median filters. With repeated application, the hybrid median filter does not excessively smooth image details (as do the conventional median filters), and typically provides superior visual quality in the filtered image. One advantage of the hybrid median filter is due to its adaptive nature, which allows the filter to perform better than the standard median filter on fast-moving picture information of small spatial extent.

3 Overview of Proposed System

3.1 Introduction

Hybrid median filter This is another type of the non-filter and advanced version of the median filter. The impulse noise removing is greatly improved by hybrid median filter. Here the median value of X, + shaped neighbours can be calculated and median value of that these are added to original median value.

3.2 Framework

B = hmf (A, n) performs hybrid median filtering of the matrix A using an n x n box. Hybrid median filter preserves edges better than a square kernel (neighbouring pixels) median filter because it is a three-step ranking operation. Three median values are calculated: MR is the median of horizontal and vertical R pixels, and MD is the median of diagonal D pixels. The filtered value is the median of the two median values and the central pixel C: median ([MR, MD, C]). Y = median of (MR, MD, C).

3.3 System Model

Algorithm for Hybrid mean filter

1. Place a cross-window over element
2. Pick up elements
3. Order elements
4. Take the middle element
5. Place a +-window over element
6. Pick up elements
7. Order elements
8. Take the middle element
9. Pick up result in point 4, 8 and element itself
10. Order elements
11. Take the middle element.

3.4 System Design Parameters

- Add noise (salt & pepper or gaussian or Poisson)
- Apply averaging filter onto image with noise
- Apply median filter onto image with noise
- Apply hybrid median filter onto image with noise
- Calculate MSE, PSNR & SSIM for each noise type
- Calculate MSE, PSNR & SSIM for each noise type with averaging filter
- Calculate MSE, PSNR & SSIM for each noise type with median filter
- Calculate MSE, PSNR & SSIM for each noise type with hybrid median filter

4 Implementation

4.1 List of MATLAB functions

- **imnoise:** to add different types of noises to the image.
- **imfilter:** to apply different filters to the image.
- **median:** to apply median to the sub-image.
- **imshow:** to display the image.
- **immse:** to calculate MSE.
- **impsnr:** to calculate PSNR.
- **imssim:** to calculate SSIM.

4.2 MATLAB Script

```
I=imread('watch.png'); %importing image
IB=imnoise(I,'salt & pepper'); %added noise figure(1)
subplot(1,2,1) %added 1st image to output imshow(I)

%{displays the grayscale image I in a figure. imshow uses the default display range for the image
data type and optimizes figure, axes, and image object properties for image display.%}

%showing imported image title('Original image') subplot(1,2,2) %added 2nd image with salt
and pepper noise to the output imshow(IB) title('Salt & Pepper Noisy image')

%AVG filter on salt and pepper noise
N=ones(3)/9; %forming 3x3 array for filter
If1=imfilter(IB,N); %adding avg filter to img with salt and pepper noise figure(2) imshow(If1)
%showing img with avg filter
title(' Salt & Pepper Noisy image filtered by a 3 by 3 average filter')

%MEDIAN filter on salt and pepper noise
img_med=IB; [r,c]=size(IB);
F_SP = zeros(r+2,c+2); %initialize array of zeroes with padding for i=2:r-1 for j=2:c-1 flt=[IB(i-1,j-
1),IB(i-1,j),IB(i-1,j+1),IB(i,j-1),IB(i,j),IB(i,j+1), IB(i+1,j-1),IB(i+1,j),IB(i+1,j+1)];
        %taking sub array of image for median filtering process
        F_SP(i,j)=median(flt); %applying median to the elts on sub-img array end end figure;
imshow(F_SP,[])
%{displays the grayscale image I, scaling the display based on the range of pixel values in I.
imshow uses [min(I(:)) max(I(:))] as the display range. imshow displays the minimum value
in I as black and the maximum value as white%} title('Noisy image filtered by a 3 by 3
median filter');
%showing image in output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I=imread('watch.png'); %importing image IC=imnoise(I,'gaussian'); %added noise

figure(4) subplot(1,2,1)
imshow(I) title('Original
image')

subplot(1,2,2) imshow(IC)
title('Gaussian noisy image')
```

```
%AVG filter on gaussian noise
N=ones(3)/9; %forming 3x3 array for filter If2=imfilter(IC,N); figure(5) imshow(If2)
title('Gaussian Noisy image filtered by a 3 by 3 average filter')

%MEDIAN filter on gaussian noise
img_med=IC; [r,c]=size(IC);
F_SP3=zeros(r+2,c+2); %initialize array of zeroes with padding for i=2:r-1 for
j=2:c-1 flt3=[IC(i-1,j-1),IC(i-1,j),IC(i-1,j+1),IC(i,j-1), IC(i,j),IC(i,j+1),
            IC(i+1,j-1),IC(i+1,j),IC(i+1,j+1)];
            %taking sub array of image for median filtering process
            F_SP3(i,j)=median(flt3);
            %applying median to the elts on sub-img array end end
figure; imshow(F_SP3,[]) title('Noisy image filtered by a 3 by 3 median
filter');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

IK=imnoise(I,'poisson'); %added noise figure(7)
subplot(1,2,1) imshow(I) title('Original image')

subplot(1,2,2) imshow(IK)
title('Poisson noisy image')

%AVG filter on Poisson noise
N=ones(3)/9;%forming 3x3 array for filter
If3=imfilter(IK,N); figure(8) imshow(If3)
title('Poisson Noisy image filtered by a 3 by 3 average filter')

%MEDIAN filter on Poisson noise img_med=IK;
[r,c]=size(IK); F_SP4=zeros(r,c); for i=2:r-1 for j=2:c-1 flt4=[IK(i-1,j-1),IK(i-
1,j),IK(i-1,j+1),IK(i,j-1), IK(i,j),IK(i,j+1),
            IK(i+1,j-1),IK(i+1,j),IK(i+1,j+1)];
            %taking sub array of image for median filtering process
            F_SP4(i,j)=median(flt4);
            ,%applying median to the elts on sub-img array end end
figure; imshow(F_SP4,[]) title('Noisy image filtered by a 3 by 3 median
filter');

%HYBRID MEDIAN FILTER
I=imread('watch.png'); %importing image
IB=imnoise(I,'salt & pepper'); %adding salt and pepper noise figure(1)
subplot(1,2,1) imshow(I) title('Original image')

subplot(1,2,2) imshow(IB) title('Salt &
Pepper Noisy image')

%implementing hybrid filter
[r,c]=size(IB); F_SP=zeros(r,c); for i=2:r-1 for j=2:c-1 flt=[IB(i-1,j),IB(i,j-
1),IB(i,j),IB(i,j+1),IB(i+1,j)]; flt1=[IB(i-1,j-1),IB(i,j),IB(i+1,j+1),IB(i-1,j+1),IB(i+1,j-1)]; flt2=I(i,j);
a=median(flt); b=median(flt1); h=[a b flt2];
            F_SP6(i,j)=median(h); end
end

figure; imshow(F_SP6,[]);
title('salt and pepper Noisy Image filtered by a 3 by 3 hybrid median filter');
```

```
I=imread('watch.png'); %importing image
IC=imnoise(I,'gaussian'); %adding gaussian noise figure(4)
subplot(1,2,1) imshow(I) title('Original image')

subplot(1,2,2) imshow(IC)
title('Gaussian noisy image')

%implementing hybrid filter
[r,c]=size(IC); F_SP=zeros(r,c); for i=2:r-
1 for j=2:c-1 flt0=[IC(i-1,j),IC(i,j-
1),IC(i,j),IC(i,j+1),IC(i+1,j)]; flt11=[IC(i-
1,j-1),IC(i,j),IC(i+1,j+1),IC(i-
1,j+1),IC(i+1,j-1)]; flt21=I(i,j);
x=median(flt0); y=median(flt11); h=[x y
flt21];
          F_SP7(i,j)=median(h); end
end figure(5); imshow(F_SP7,[]);
title('Gaussian Noisy Image filtered by a 3 by 3 hybrid median filter');

I=imread('watch.png'); %importing image
IK=imnoise(I,'poisson'); %adding Poisson noise figure(7)
subplot(1,2,1) imshow(I) title('Original image')
subplot(1,2,2) imshow(IK) title('Poisson noisy image')
%hybrid median filter
[r,c]=size(IK); F_SP=zeros(r,c); for i=2:r-1 for j=2:c-1 flt0=[IK(i-1,j),IK(i,j-
1),IK(i,j),IK(i,j+1),IK(i+1,j)]; flt12=[IK(i-1,j-1),IK(i,j),IK(i+1,j+1),IK(i-1,j+1),IK(i+1,j-1)]; flt22=I(i,j);
P=median(flt0); Q=median(flt12); h=[P Q flt22];
          F_SP8(i,j)=median(h); end
end figure(6); imshow(F_SP8,[]);
title('Poisson Noisy Image filtered by a 3 by 3 hybrid median filter');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%values of ratios to compare fprintf('**ERROR
COMPARISION**\n\n');

err_IB= immse(I,IB); fprintf('\nThe MSE of the salt and pepper noise image is %0.4f\n',err_IB);
err2_IB= psnr(I,IB);
fprintf('\nThe PSNR ratio of the salt and pepper noise image is %0.4f\n',err2_IB); err3_IB= ssim(IB,I);
fprintf('\nThe SSIM of the salt and pepper noise image is %0.4f\n',err3_IB);

err_If1= immse(I,If1);
fprintf('\nThe MSE of the salt and pepper noise image filtered by average filter is %0.4f\n',err_If1);
err2_If1= psnr(I,If1); fprintf('\nThe PSNR ratio of the salt and pepper noise image filtered by average
filter is %0.4f\n',err2_If1);
err3_If1= ssim(If1,I); fprintf('\nThe SSIM of the salt and pepper noise image filtered by average filter is
%0.4f\n',err3_If1);

err_img_med= immse(I,img_med); fprintf('\nThe MSE of the salt and pepper noise image filtered by
median filter is %0.4f\n',err_img_med); err2_img_med= psnr(I,img_med); fprintf('\nThe PSNR ratio of
the salt and pepper noise image filtered by median filter is %0.4f\n',err2_img_med); err3_img_med=
ssim(img_med,I); fprintf('\nThe SSIM of the salt and pepper noise image filtered by median filter is
%0.4f\n',err3_img_med);

err_IC= immse(I,IC); fprintf('\nThe MSE of the gaussian noise image is %0.4f\n',err_IC); err2_IC=
psnr(I,IC); fprintf('\nThe PSNR ratio of the gaussian noise image is %0.4f\n',err2_IC); err3_IC=
ssim(IC,I); fprintf('\nThe SSIM of the gaussian noise image is %0.4f\n',err3_IC);
```

```
err_If2= immse(I,If2); fprintf(’\nThe MSE of the gaussian noise image filtered by average filter is
%0.4f\n’,err_If2); err2_If2= psnr(I,If2); fprintf(’\nThe PSNR ratio of the gaussian noise image filtered by
average filter is %0.4f\n’,err2_If2); err3_If2= ssim(If2,I); fprintf(’\nThe SSIM of the gaussian noise
image filtered by average filter is %0.4f\n’,err3_If2);
```

```
err_img_med1= immse(I,img_med); fprintf(’\nThe MSE of the gaussian noise image filtered
by median filter is %0.4f\n’,err_img_med); err2_img_med= psnr(I,img_med); fprintf(’\nThe
PSNR ratio of the gaussian noise image filtered by median filter is %0.4f\n’,err2_img_med);
err3_img_med= ssim(img_med,I); fprintf(’\nThe SSIM of the gaussian noise image filtered by
median filter is %0.4f\n’,err3_img_med);
```

```
err_IK= immse(I,IK);
fprintf(’\nThe MSE of the Poisson noise image is %0.4f\n’,err_IK); err2_IK= psnr(I,IK);
fprintf(’\nThe PSNR ratio of the Poisson noise image is %0.4f\n’,err2_IK); err3_IK= ssim(IK,I);
fprintf(’\nThe SSIM of the Poisson noise image is %0.4f\n’,err3_IK);
```

```
err_If3= immse(I,If3); fprintf(’\nThe MSE of the Poisson noise image filtered by average filter
is %0.4f\n’,err_If3); err2_If3= psnr(I,If3);
fprintf(’\nThe PSNR ratio of the Poisson noise image filtered by average filter is %0.4f\n’,err2_If3);
err3_If3= ssim(If3,I);
fprintf(’\nThe SSIM of the Poisson noise image filtered by average filter is %0.4f\n’,err3_If3);
```

```
err_img_med2= immse(I,img_med); fprintf(’\nThe MSE of the Poisson noise image filtered
by median filter is %0.4f\n’,err_img_med); err2_img_med2= psnr(I,img_med);
fprintf(’\nThe PSNR ratio of the Poisson noise image filtered by median filter is
%0.4f\n’,err2_img_med); err3_img_med2= ssim(img_med,I); fprintf(’\nThe SSIM of the
Poisson noise image filtered by median filter is %0.4f\n’,err3_img_med);
```

5 Analysis

5.1 Image-1

Table 1: **Salt and pepper noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	1135.6316	17.5784	0.3275
Average filter	224.0999	24.6264	0.6236
Hybrid median filter	51.7384	30.9556	0.8214

Table 2: **Gaussian noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	1135.6316	17.5784	0.3275
Average filter	224.0999	24.6264	0.6236
Hybrid median filter	51.7384	30.9556	0.8214

Table 3: **Poisson noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	1135.6316	17.5784	0.3275
Average filter	224.0999	24.6264	0.6236
Hybrid median filter	51.7384	30.9556	0.8214

5.2 Image-2

Table 1: **Salt and pepper noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	955.7281	18.3275	0.6813
Average filter	162.8328	26.0134	0.9125
Hybrid median filter	126.1165	27.1231	0.9315

Table 2: **Gaussian noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	630.9675	20.1307	0.7424
Average filter	119.0279	27.1231	0.9315
Hybrid median filter	126.1165	27.3743	0.9351

Table 3: **Poisson noise**

Image Type Ratio	**MSE**	**PSNR**	**SSIM**
With noise only	126.1165	27.1336	0.9318
Average filter	125.8111	27.1231	0.9642
Hybrid median filter	119.0279	30.1817	0.9315

6 Inference

After successfully applying the filters to the images with different types of noises, we have found the MSE and PSNR for all the results and have mentioned them above. As seen, for all types of noises, MSE value of the hybrid median filter comes to be the lowest which means the error is least. At the same time, the Peak Signal to Noise Ratio is the highest. This result suggests that Hybrid Median Filter comes out to be the best for all the cases. Analysing for other filters, in case of Salt and Pepper, median filter is better than mean filter. In case of Gaussian noise, both mean and median filter gives almost the same result though median is slightly better than mean. For Poisson noise, again the median filter comes out to be better than the mean filter.

6.1 Original Image-1

Original image

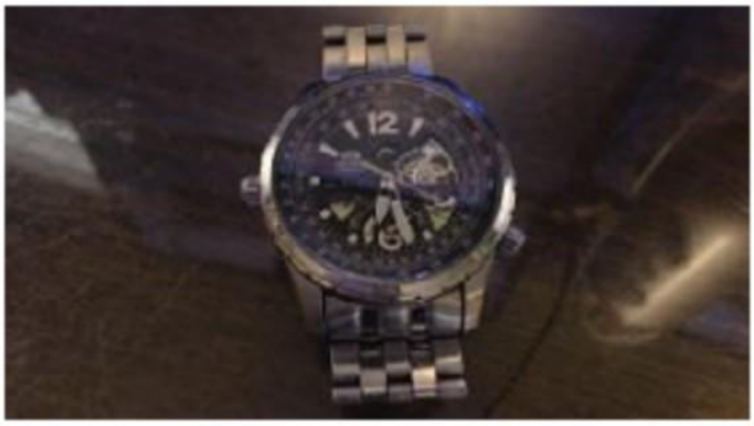
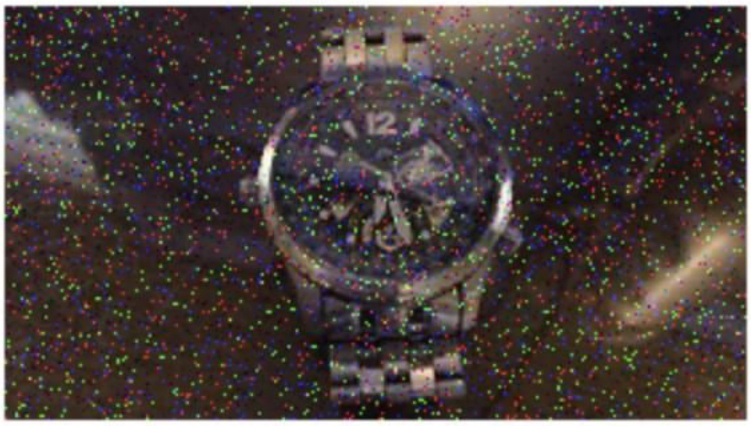
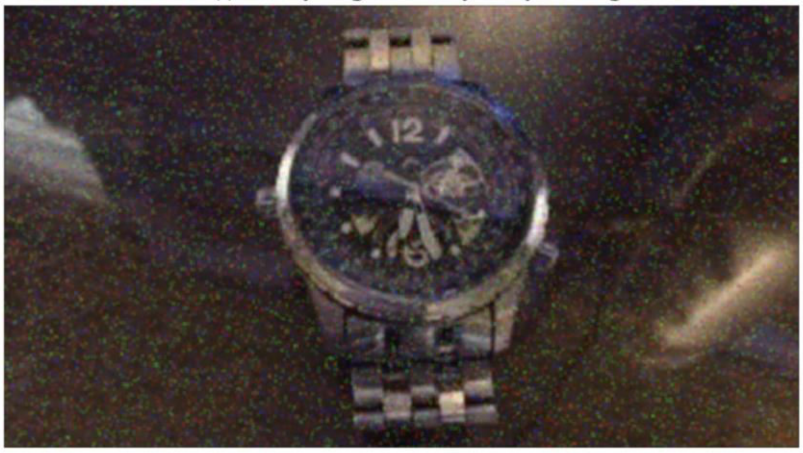


Image with Salt and Pepper noise

Salt & Pepper Noisy image



Salt & Pepper Noisy image filtered by a 3 by 3 average filter



Salt & pepper Noisy image filtered by a 3 by 3 median filter

Image with Gaussian noise

Gaussian noisy image

Gaussian Noisy image filtered by a 3 by 3 average filter

Gaussian Noisy image filtered by a 3 by 3 median filter

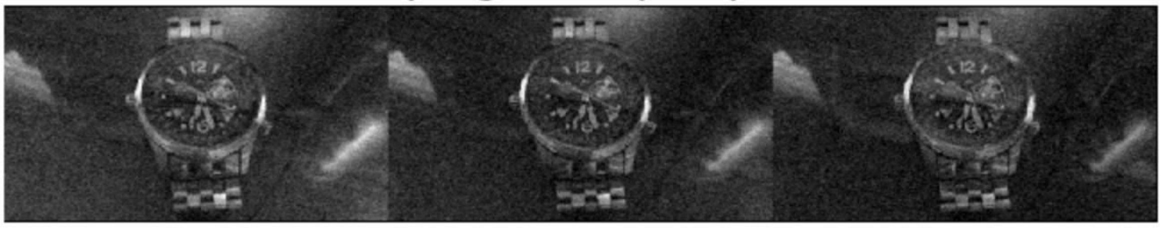
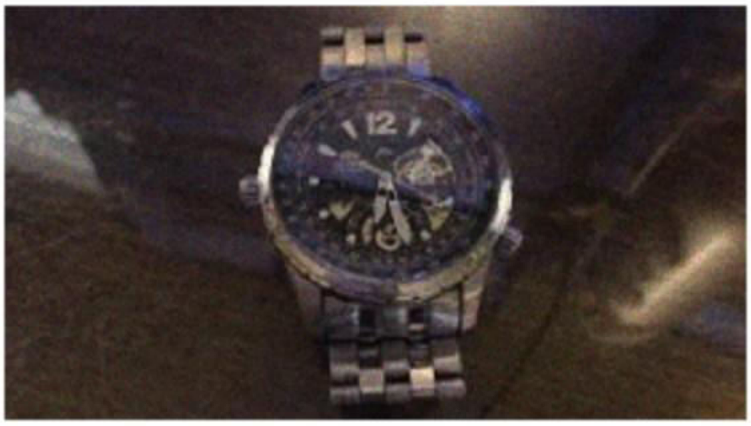
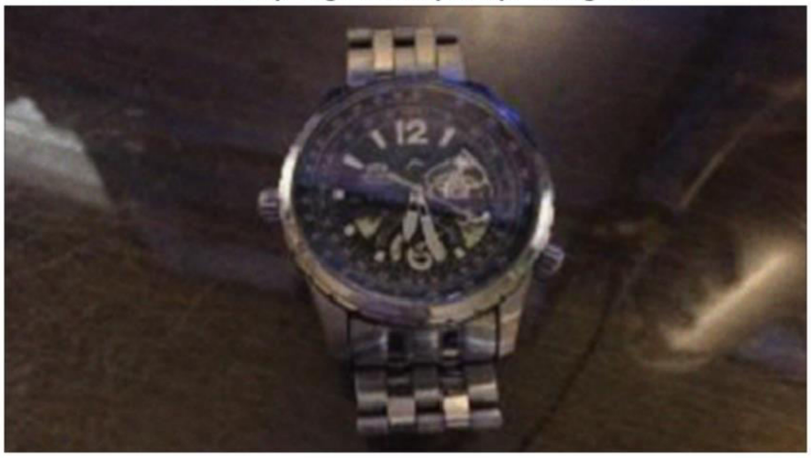


Image with Poisson noise

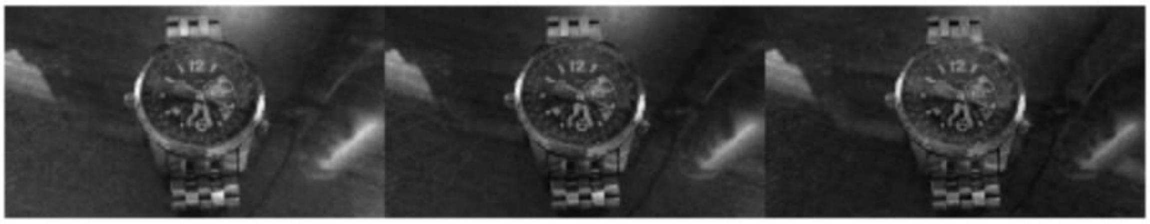
Poisson noisy image



Poisson Noisy image filtered by a 3 by 3 average filter

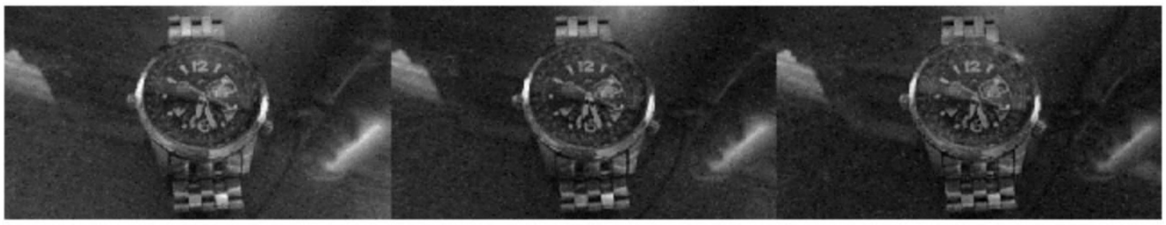


Poisson Noisy image filtered by a 3 by 3 median filter

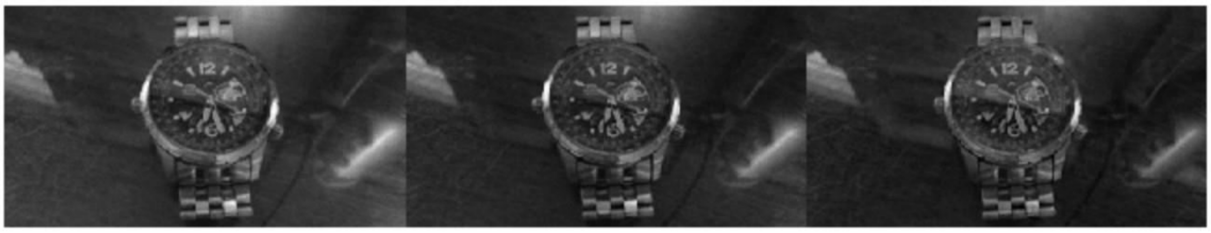


Noisy images with Hybrid median filter

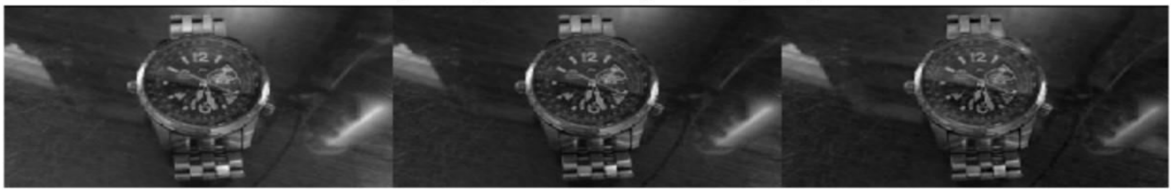
Gaussian Noisy Image filtered by a 3 by 3 hybrid median filter



Poisson Noisy Image filtered by a 3 by 3 hybrid median filter

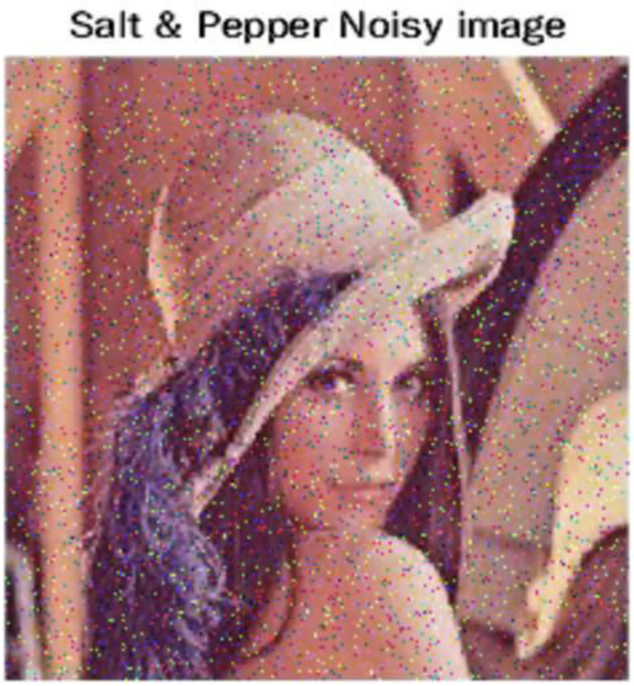


salt and pepper Noisy Image filtered by a 3 by 3 hybrid median filter

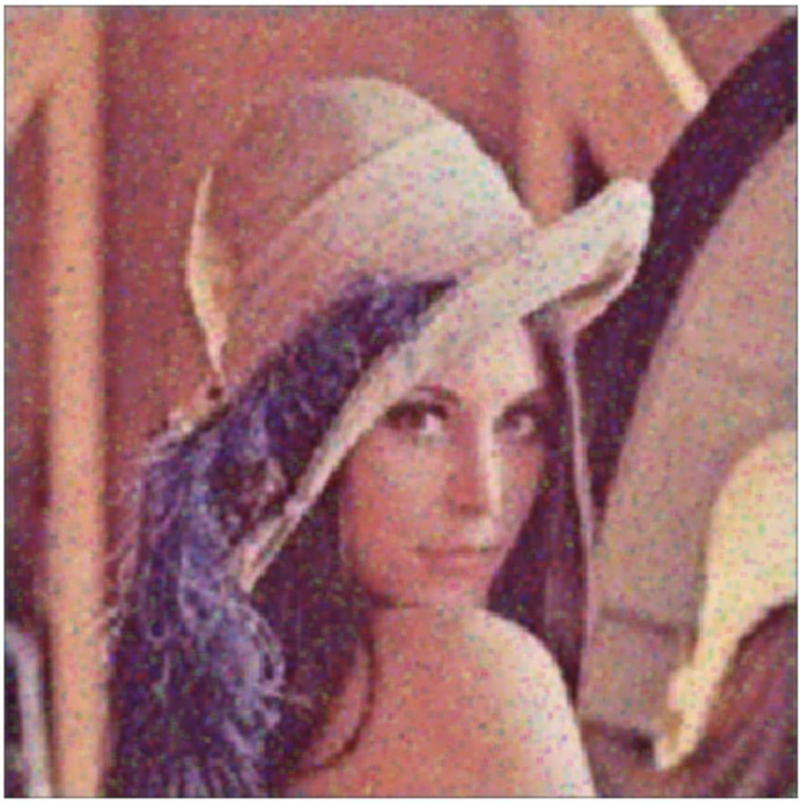


6.2 Original image-2

Image with Salt and Pepper noise



Salt & Pepper Noisy image filtered by a 3 by 3 average filter



Salt & pepper Noisy image filtered by a 3 by 3 median filter

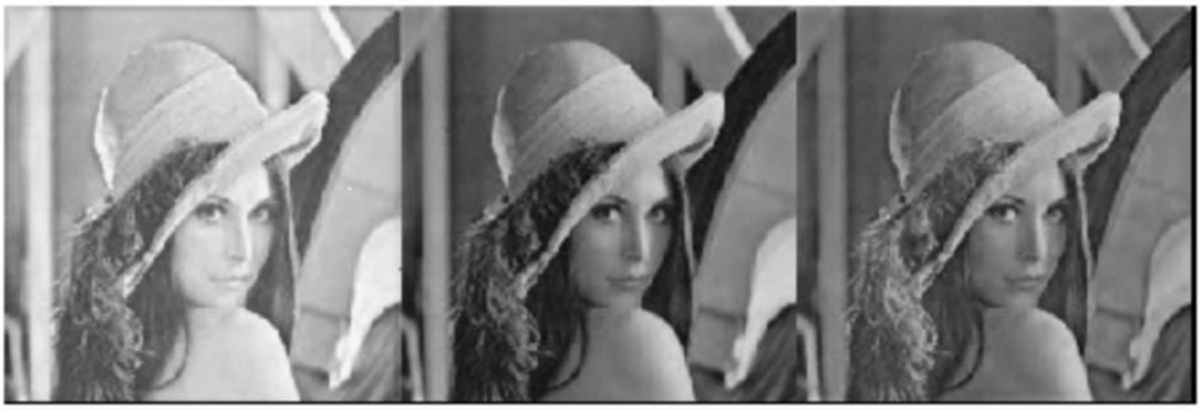
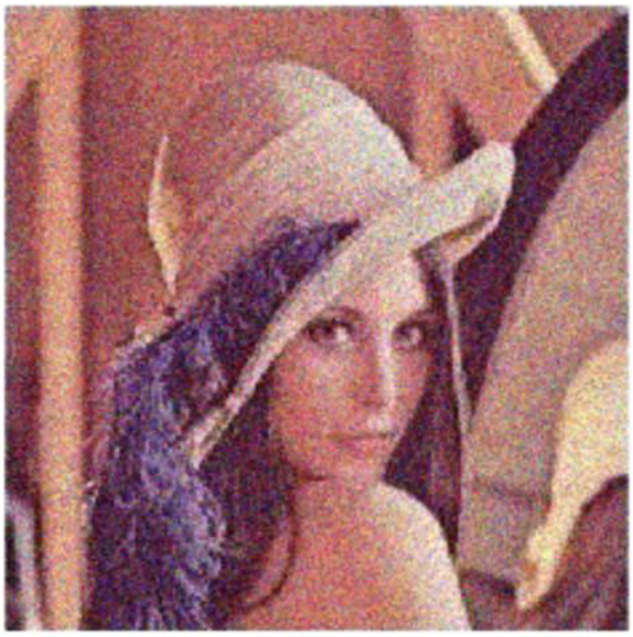
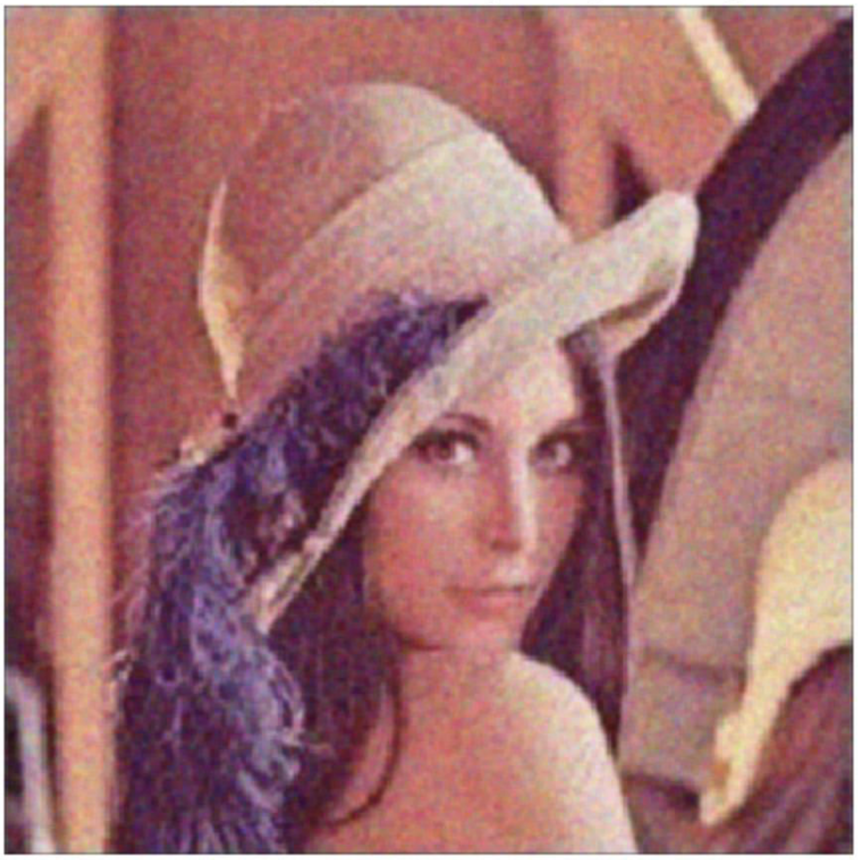


Image with Gaussian noise

Gaussian noisy image



Gaussian Noisy image filtered by a 3 by 3 average filter



Gaussian Noisy image filtered by a 3 by 3 median filter

Image with Poisson noise

Poisson noisy image

Poisson Noisy image filtered by a 3 by 3 average filter

Poisson Noisy image filtered by a 3 by 3 median filter

Noisy images with Hybrid median filter

salt and pepper Noisy Image filtered by a 3 by 3 hybrid median filter

Gaussian Noisy Image filtered by a 3 by 3 hybrid median filter

Poisson Noisy Image filtered by a 3 by 3 hybrid median filter

7 References

1. V.R.Vijaykumar, P.T.Vanathi, P.Kanagasabapathy, *Fast and Efficient Algorithm to Remove Gaussian Noise in Digital Images*, International Journal of Computer Science

2. Triet Le, Rick Chartrand, Thomas J. Asaki, *A Variational Approach to Reconstructing Images Corrupted by Poisson Noise*, Journal of Mathematical Imaging and Vision

3. Deepalakshmi R, Sindhuja A, *Survey on Salt and Pepper Noise Removal Techniques*, International Journal of Innovative Research in Science, Engineering and Technology

4. M. Narasimha Rao, S.R. Nageswarareddy, A. Leela Prasad, V. Dilip Kumar, Ch. Sathyam, P.S.N.M. Vinay, *Hybrid Median Filter for Impulse Noise Removal,* International Journal of Engineering Research & Technology

5. Linear and Non-Linear Filters for Image Processing using MATLAB
   https://researchrepository.murdoch.edu.au/id/eprint/30815/1/whole.pdf