PROJECT-6

NON-LINEAR FILTERING

EE5356 Digital Image Processing
Dr. K. R. Rao

Submitted By:
Ashutosh Desai
UTA ID:1001388602
Email: - ashutoshrajesh.desai@mavs.uta.edu

March 12th 2018

EE 5356 - DIGITAL IMAGE PROCESSING - PROJECT 4

NON-LINEAR FILTERING

Read any 256x256 or 512x512 grayscale image. Add the following types of noise to it to generate 4 noisy images:

1. Gaussian noise
2. Poisson noise
3. Salt & pepper noise
4. Speckle noise

Apply the following spatial filters to the noisy images:

1. Arithmetic mean
2. Geometric mean
3. Harmonic mean
4. Contra-harmonic mean
5. Median filter
6. Min
7. Max
8. Mid-point
9. Alpha trimmed mean filter

Submit the following with your code:

1. Print
    a. the original image,
    b. the noisy images, and
    c. the results of all the filters on each noisy image.
2. Determine which type of filtering worked well for each type of noise.

References:

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, III edition, Prentice Hall, pages 322-325, 2008.
2. Gonzalez, Woods and Eddins, “Digital Image Processing with MATLB”, I edition, Prentice Hall, pages 160-164, 2009.
3. Practical image and video processing using MATLAB by Marques, Oge

MATLAB SCRIPT:

```
function[]=NonLinear()
clear all
close all
clc;

In_img = imread('D:\STUDY\DIP\Test img\goldhill256.bmp');
[row,col] = size(In_img);
figure;
imshow(In_img);
title('Orignal Image');


% speckle noise
Variance = 0.05;
Speckle_Image = imnoise(In_img,'speckle',Variance);
Spk_Img = zeros(row+2,col+2);
Spk_Img(2:1:row+1,2:1:col+1) = Speckle_Image(:,:);
Spk_Img(:,1) = Spk_Img(:,2);
Spk_Img(:,col+2) = Spk_Img(:,col+1);
Spk_Img(1,:) = Spk_Img(2,:);
Spk_Img(row+2,:) = Spk_Img(row+1,:);

% Gaussianian noise
Mean = 0;
Variance = 0.01;
GN = imnoise(In_img,'Gaussian',Mean,Variance);
GN_Img = zeros(row+2,col+2);
GN_Img(2:1:row+1,2:1:col+1) = GN(:,:);
GN_Img(:,1) = GN_Img(:,2);
GN_Img(:,col+2) = GN_Img(:,col+1);
GN_Img(1,:) = GN_Img(2,:);
GN_Img(row+2,:) = GN_Img(row+1,:);

% salt & pepper noise
G = 0.05;
SNP_N = imnoise(In_img,'salt & pepper',G);
SNP_N_img = zeros(row+2,col+2);
SNP_N_img(2:1:row+1,2:1:col+1) = SNP_N(:,:);
SNP_N_img(:,1) = SNP_N_img(:,2);
SNP_N_img(:,col+2) = SNP_N_img(:,col+1);
SNP_N_img(1,:) = SNP_N_img(2,:);
SNP_N_img(row+2,:) = SNP_N_img(row+1,:);
```

```
%Poissonon noise
PN = imnoise(In_img,'Poisson');
PN_Img = zeros(row+2,col+2);
PN_Img(2:1:row+1,2:1:col+1) = PN(:,:);
PN_Img(:,1) = PN_Img(:,2);
PN_Img(:,col+2) = PN_Img(:,col+1);
PN_Img(1,:) = PN_Img(2,:);
PN_Img(row+2,:) = PN_Img(row+1,:);


figure;
subplot(2,2,1),imshow(GN), title('Gaussian noise image');
subplot(2,2,2), imshow(PN), title('Poisson noise image');
subplot(2,2,3), imshow(SNP_N), title('Salt & pepper noise
image');
subplot(2,2,4), imshow(Speckle_Image),title('Speckle noise
image');

% arithmetic Mean filter
Arithmetic_Filter(In_img,GN_Img,PN_Img,SNP_N_img,Spk_Img);

% geometric filter
Geometric_Filter(In_img,GN_Img,PN_Img,SNP_N_img,Spk_Img);

%harmonic filter
Harmonic_Filter(In_img,GN_Img,PN_Img,SNP_N_img,Spk_Img);

%contraharmonic filter
ContraHarmonic_Filter(In_img,GN_Img,PN_Img,SNP_N_img,Spk_Img);

% Median filter
Median_Filter(GN,PN,SNP_N,Speckle_Image);

% Max filter
Max_Filter(GN,PN,SNP_N,Speckle_Image);

% Min filter
Min_Filter(GN,PN,SNP_N,Speckle_Image);

% mid-point filter
Mid_Point_filter(GN,PN,SNP_N,Speckle_Image);

% alpha trimmed Mean filter
Alphatrimmed_filt(GN,PN,SNP_N,Speckle_Image);

end
```

```
%Functions used:
% arithmetic filter
function[]=Arithmetic_Filter(ip_img,Gaussian_img,Poissonon_img,S
alandpr_img,Sp_img)
A=3;
B=3;
[row,col] = size(ip_img);
Varia= @(x) arith_calc(x(:));
AriMean_img = nlfilter(double(Gaussian_img),[A B],Varia);
figure;
subplot(2,2,1),imshow(uint8(AriMean_img(2:1:row+1,2:1:col+1)));
title('Gaussian Noise Arithmatic Filter');
AriMean_img = nlfilter(double(Poissonon_img),[A B],Varia);
subplot(2,2,2),imshow(uint8(AriMean_img(2:1:row+1,2:1:col+1)));
title('Poisson Noise Arithmatic Filter');
AriMean_img = nlfilter(double(Salandpr_img),[A B],Varia);
subplot(2,2,3),imshow(uint8(AriMean_img(2:1:row+1,2:1:col+1)));
title('S&P Noise Arithmatic Filter');
AriMean_img = nlfilter(double(Sp_img),[A B],Varia);
subplot(2,2,4),imshow(uint8(AriMean_img(2:1:row+1,2:1:col+1)))
title('Speckle Noise Arithmatic Filter');

end

% geometric filter
function[]=Geometric_Filter(ip_img,Gaussian_img,Poissonon_img,Sa
landpr_img,Sp_img)
A=3;
B=3;
[row,col] = size(ip_img);
Varia = @(x) geometric_calc(x(:));
Geo_Img = nlfilter(double(Gaussian_img),[A B],Varia);
figure;
subplot(2,2,1),imshow(uint8(Geo_Img(2:1:row+1,2:1:col+1)));
title('Gaussian noise geometric filtered image');
Geo_Img = nlfilter(double(Poissonon_img),[A B],Varia);
subplot(2,2,2),imshow(uint8(Geo_Img(2:1:row+1,2:1:col+1)));
title('Poissonon noise geometric filtered image');
Geo_Img = nlfilter(double(Salandpr_img),[A B],Varia);
subplot(2,2,3),imshow(uint8(Geo_Img(2:1:row+1,2:1:col+1)));
title('S&P noise geometric filtered image');
Geo_Img = nlfilter(double(Sp_img),[A B],Varia);
subplot(2,2,4),imshow(uint8(Geo_Img(2:1:row+1,2:1:col+1)));
title('Speckle noise geometric filtered image');
end

%harmonic filter
```

```
function[]=Harmonic_Filter(ip_img,Gaussian_img,Poissonon_img,Sal
andpr_img,Sp_img)
A=3;
B=3;
[row,col] = size(ip_img);
Varia = @(x) harmonic_calc(x(:));
harmonic_Image = nlfilter(double(Gaussian_img),[A B],Varia);
figure;
subplot(2,2,1),imshow(uint8(harmonic_Image(2:1:row+1,2:1:col+1))
);
title('Gaussianian noise harmonic filtered image');
harmonic_Image = nlfilter(double(Poissonon_img),[A B],Varia);
subplot(2,2,2),imshow(uint8(harmonic_Image(2:1:row+1,2:1:col+1))
);
title('Poissonon noise harmonic filtered image');
harmonic_Image = nlfilter(double(Salandpr_img),[A B],Varia);
subplot(2,2,3),imshow(uint8(harmonic_Image(2:1:row+1,2:1:col+1))
);
title('Salt & pepper noise harmonic filtered image');
harmonic_Image = nlfilter(double(Sp_img),[A B],Varia);
subplot(2,2,4),imshow(uint8(harmonic_Image(2:1:row+1,2:1:col+1))
);
title('Speckle noise harmonic filtered image');
end

%contraharmonic filter
function[]=ContraHarmonic_Filter(ip_img,Gaussian_img,Poissonon_i
mg,Salandpr_img,Sp_img)
A=3;
B=3;
[row,col] = size(ip_img);
Varia = @(x) contraharmonic_calc(x(:));
CH_img = nlfilter(double(Gaussian_img),[A B],Varia);
figure;

subplot(2,2,1),imshow(uint8(CH_img(2:1:row+1,2:1:col+1)));
title('Gaussianian noise contra-harmonic filtered image');
CH_img = nlfilter(double(Poissonon_img),[A B],Varia);
subplot(2,2,2),imshow(uint8(CH_img(2:1:row+1,2:1:col+1)));
title('Poissonon noise contra-harmonic filtered image');
CH_img = nlfilter(double(Salandpr_img),[A B],Varia);
subplot(2,2,3),imshow(uint8(CH_img(2:1:row+1,2:1:col+1)));
title('Salt & pepper noise contra-harmonic filtered image');
CH_img = nlfilter(double(Sp_img),[A B],Varia);
subplot(2,2,4),imshow(uint8(CH_img(2:1:row+1,2:1:col+1)));
title('Speckle noise contra-harmonic filtered image');
end
```

```
% Median filter
function[]=Median_Filter(Gaussian,Poisson,Salandpr,Speckimage)
A=3;
B=3;
Median_Img = medfilt2(Gaussian,[A B],'symmetric');
figure(7),
subplot(2,2,1),imshow(uint8(Median_Img));
title('Gaussianian noise median filtered image');

Median_Img = medfilt2(Poisson,[A B],'symmetric');
subplot(2,2,2),imshow(uint8(Median_Img));
title('Poissonon noise median filtered image');

Median_Img = medfilt2(Salandpr,[A B],'symmetric');
subplot(2,2,3),imshow(uint8(Median_Img));
title('Salt & pepper noise median filtered image');

Median_Img = medfilt2(Speckimage,[A B],'symmetric');
subplot(2,2,4),imshow(uint8(Median_Img));
title('Speckle noise median filtered image');
end

% Max filter
function[]=Max_Filter(Gaussian,Poisson,Salandpr,Speckimage)
A=3;
B=3;

Max_Img = ordfilt2(Gaussian,A*B,ones(A,B),'symmetric');
figure(8),
subplot(2,2,1),imshow(uint8(Max_Img));
title('Gaussianian noise max filtered image');

Max_Img = ordfilt2(Poisson,A*B,ones(A,B),'symmetric');
subplot(2,2,2),imshow(uint8(Max_Img));
title('Poissonon noise max filtered image');

Max_Img = ordfilt2(Salandpr,A*B,ones(A,B),'symmetric');
subplot(2,2,3),imshow(uint8(Max_Img));
title('Salt & pepper noise max filtered image');

Max_Img = ordfilt2(Speckimage,A*B,ones(A,B),'symmetric');
subplot(2,2,4),imshow(uint8(Max_Img));

title('Speckle noise max filtered image');
end
```

```
% Min filter
function[]=Min_Filter(Gaussian,Poisson,Salandpr,Speckimage)
A=3;
B=3;

Min_Img = ordfilt2(Gaussian,1,ones(A,B),'symmetric');
figure(9),
subplot(2,2,1),imshow(uint8(Min_Img));
title('Gaussianian noise min filtered image');

Min_Img = ordfilt2(Poisson,1,ones(A,B),'symmetric');
subplot(2,2,2),imshow(uint8(Min_Img));
title('Poissonon noise min filtered image');

Min_Img = ordfilt2(Salandpr,1,ones(A,B),'symmetric');
subplot(2,2,3),imshow(uint8(Min_Img));
title('Salt & pepper noise min filtered image');

Min_Img = ordfilt2(Speckimage,1,ones(A,B),'symmetric');
subplot(2,2,4),imshow(uint8(Min_Img));
title('Speckle noise min filtered image');
end

% mid-point filter
function[]=Mid_Point_filter(Gaussian,Poisson,Salandpr,Speckimage
)
A=3;
B=3;

fil_1 = ordfilt2(Gaussian, 1, ones(A,B), 'symmetric');
fil_2 = ordfilt2(Gaussian, A*B, ones(A,B), 'symmetric');
MPoint_img = imlincomb(0.5, fil_1, 0.5, fil_2);
figure(10),
subplot(2,2,1),imshow(uint8(MPoint_img));
title('Gaussianian noise mid-point filtered image');

fil_1 = ordfilt2(Poisson, 1, ones(A,B), 'symmetric');
fil_2 = ordfilt2(Poisson, A*B, ones(A,B), 'symmetric');
MPoint_img = imlincomb(0.5, fil_1, 0.5, fil_2);
subplot(2,2,2),imshow(uint8(MPoint_img));
title('Poissonon noise mid-point filtered image');

fil_1 = ordfilt2(Salandpr, 1, ones(A,B), 'symmetric');
fil_2 = ordfilt2(Salandpr, A*B, ones(A,B), 'symmetric');
MPoint_img = imlincomb(0.5, fil_1, 0.5, fil_2);
subplot(2,2,3),imshow(uint8(MPoint_img));
title('Salt & pepper noise mid-point filtered image');
```

```
fil_1 = ordfilt2(Speckimage, 1, ones(A,B), 'symmetric');
fil_2 = ordfilt2(Speckimage,A*B, ones(A,B), 'symmetric');
MPoint_img = imlincomb(0.5, fil_1, 0.5, fil_2);
subplot(2,2,4),imshow(uint8(MPoint_img));
title('Speckle noise mid-point filtered image');

end

% alpha trimmed Mean filter
function[]=Alphatrimmed_filt(Gaussian,Poisson,Salandpr,Speckimag
e)
    A=3;
    B=3;
    D = 4;
    alptri_img = imfilter(double(Gaussian), ones(A, B),
'symmetric');
    for G = 1:D/2
        alptri_img = imsubtract(alptri_img,
ordfilt2(double(Gaussian), G, ones(A,B), 'symmetric'));
    end
    for G = (G*B - (D/2) + 1):G*B
        alptri_img = imsubtract(alptri_img,
ordfilt2(double(Gaussian), G, ones(A,B), 'symmetric'));
    end
    alptri_img = alptri_img / (G*B - D);
    figure(11);
    subplot(2,2,1),imshow(uint8(alptri_img));
    title('Gaussianian noise alpha trimmed Mean filtered
image');
    alptri_img = imfilter(double(Poisson), ones(A,B),
'symmetric');
    for G = 1:D/2
        alptri_img = imsubtract(alptri_img,
ordfilt2(double(Poisson), G, ones(A,B), 'symmetric'));
    end
    for A = (A*B - (D/2) + 1):A*B
    alptri_img = imsubtract(alptri_img,
ordfilt2(double(Poisson), G, ones(A,B), 'symmetric'));
    end
    alptri_img = alptri_img / (A*B - D);
    subplot(2,2,2),imshow(uint8(alptri_img));
    title('Poissonon noise alpha trimmed Mean filtered image');
    alptri_img = imfilter(double(Salandpr), ones(A,B),
'symmetric');
    for G = 1:D/2
```

```
        alptri_img = imsubtract(alptri_img,
ordfilt2(double(Salandpr), G,ones(A,B), 'symmetric'));
    end
    for G = (A*B - (D/2) + 1):A*B
        alptri_img = imsubtract(alptri_img,
ordfilt2(double(Salandpr), G,ones(A, B), 'symmetric'));
    end
    alptri_img = alptri_img / (A*B - D);
    subplot(2,2,3),imshow(uint8(alptri_img));
    title('Salt & pepper noise alpha trimmed Mean filtered
image');
    alptri_img = imfilter(double(Speckimage), ones(A,B),
'symmetric');
    for G = 1:D/2
        atmImage = imsubtract(alptri_img,
ordfilt2(double(Salandpr), G,ones(A,B), 'symmetric'));
    end
    for G = (A*B - (D/2) + 1):A*B
        atmImage = imsubtract(alptri_img,
ordfilt2(double(Salandpr), G,ones(A,B), 'symmetric'));
    end
    alptri_img = alptri_img / (A*B - D);
    subplot(2,2,4),imshow(uint8(alptri_img));
    title('Speckle noise alpha trimmed Mean filtered image');
end

%Functions to calculate Varia:

function Varia = arith_calc(A)
    [M,N] = size(A);
    sum = 0;
    for i = 1:M
        for j = 1:N
            sum = sum + A(i,j);
        end
    end
    Varia = sum / (M * N);
end

function Varia = geometric_calc(A)
    [M,N] = size(A);
    prod = 1;
    for i = 1:M
        for j = 1:N
            prod = prod * A(i,j);
        end
    end
    Varia = prod ^ (1/(M * N));
end

function Varia = harmonic_calc(A)
    [M,N] = size(A);
    sum = 0;
    for i = 1:M
        for j = 1:N
            sum = sum + (1/A(i,j));
        end
    end
    Varia = (M * N)/sum;
end

function Varia = contraharmonic_calc(A)
    [M,N] = size(A);
    Q = 1;
    sum = 0;
    sum1 = 0;
    for i = 1:M
        for j = 1:N
            sum = sum + (A(i,j)^(Q+1));
            sum1 = sum1 + (A(i,j)^(Q));
        end
    end
    Varia = sum/sum1;
    end
```

OUTPUT:

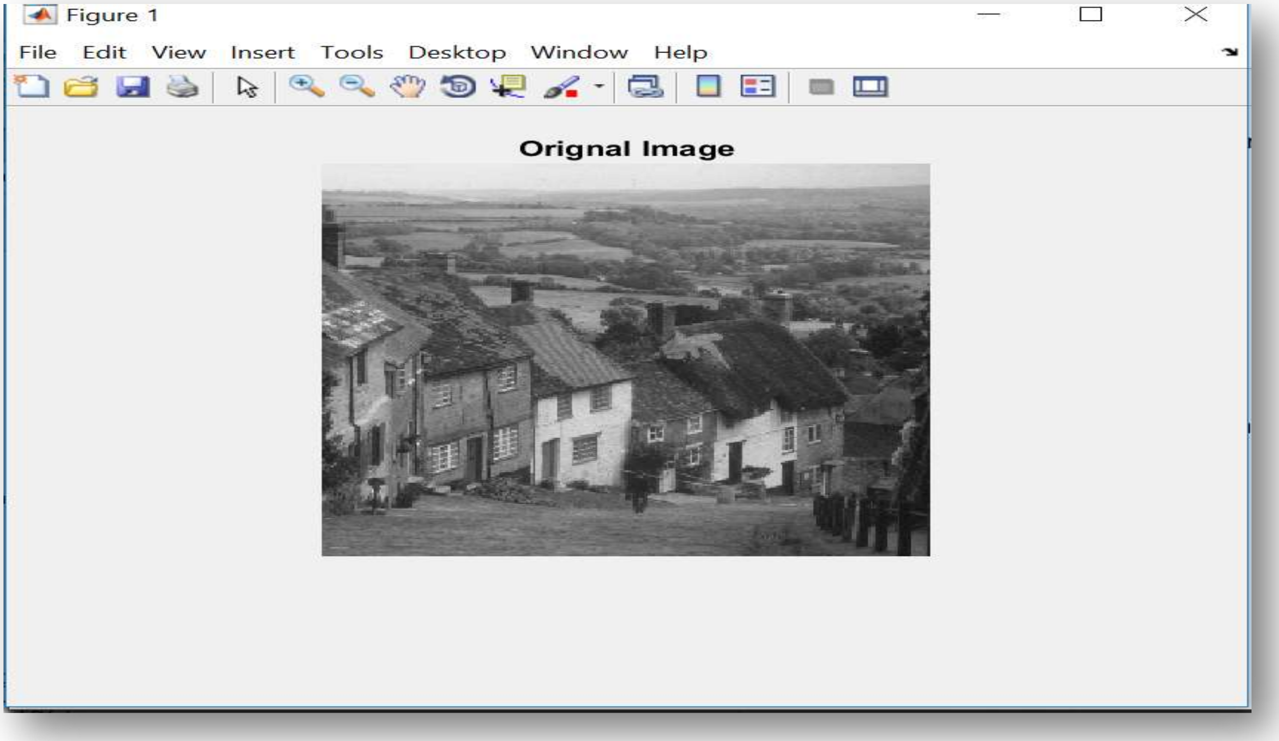


Figure 1
File Edit View Insert Tools Desktop Window Help
Orignal Image

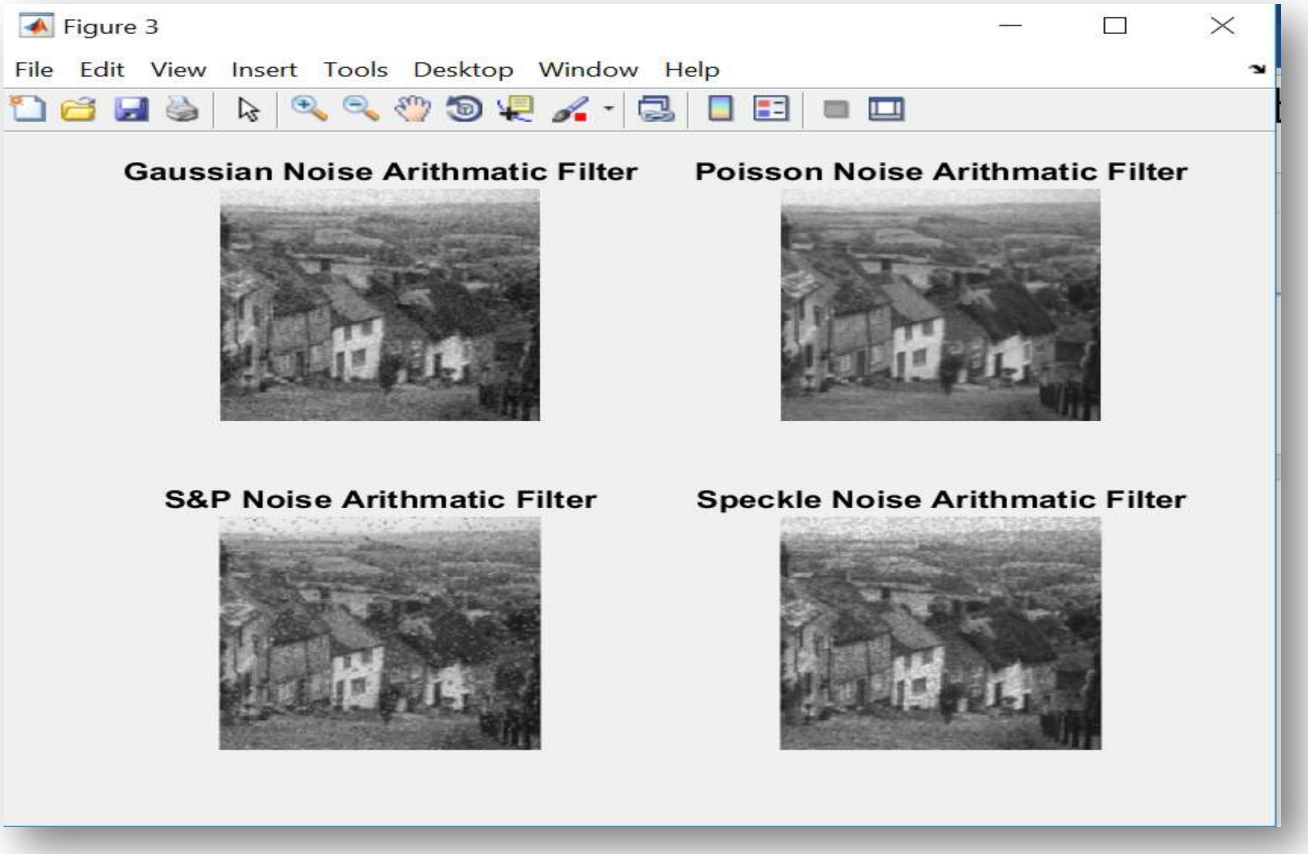


Figure 3
File Edit View Insert Tools Desktop Window Help
Gaussian Noise Arithmatic Filter
Poisson Noise Arithmatic Filter
S&P Noise Arithmatic Filter
Speckle Noise Arithmatic Filter

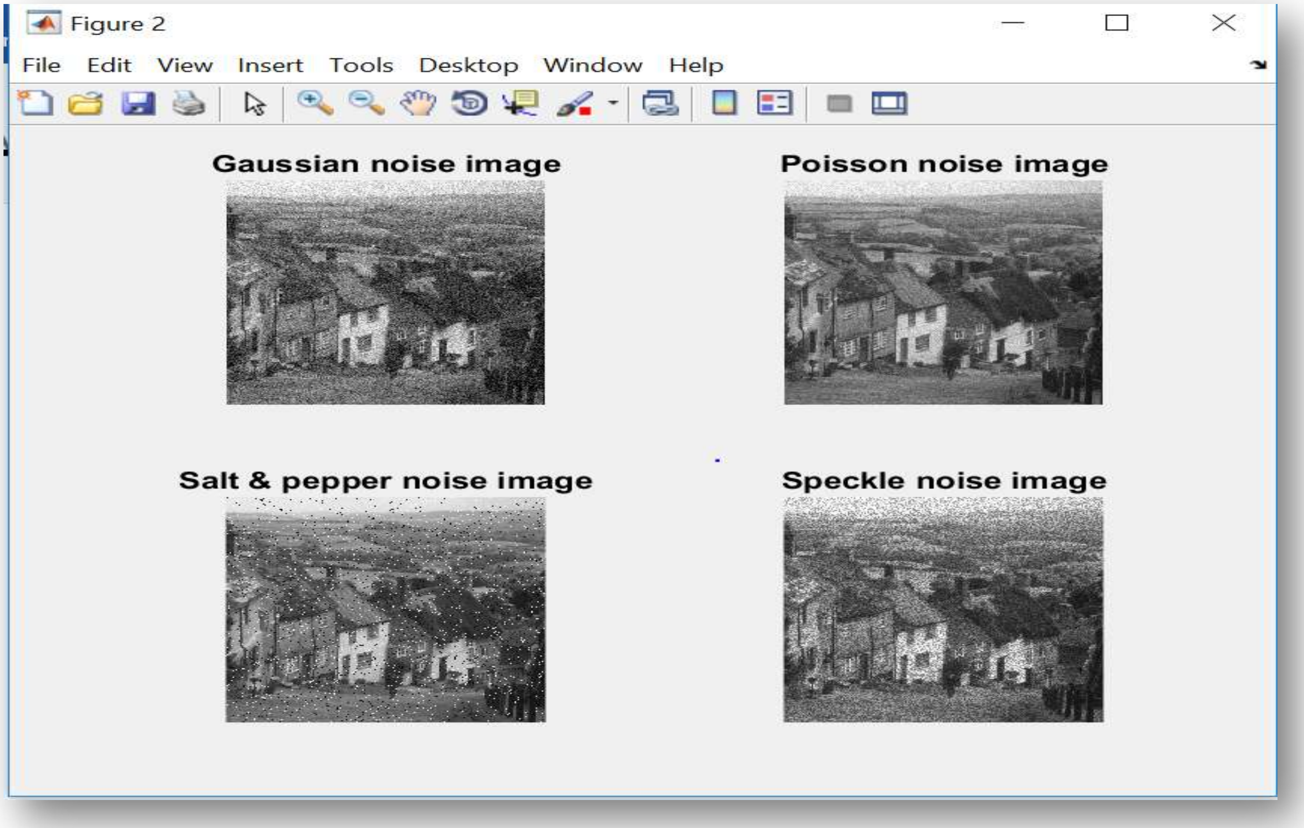


Figure 2
File Edit View Insert Tools Desktop Window Help
Gaussian noise image
Poisson noise image
Salt & pepper noise image
Speckle noise image

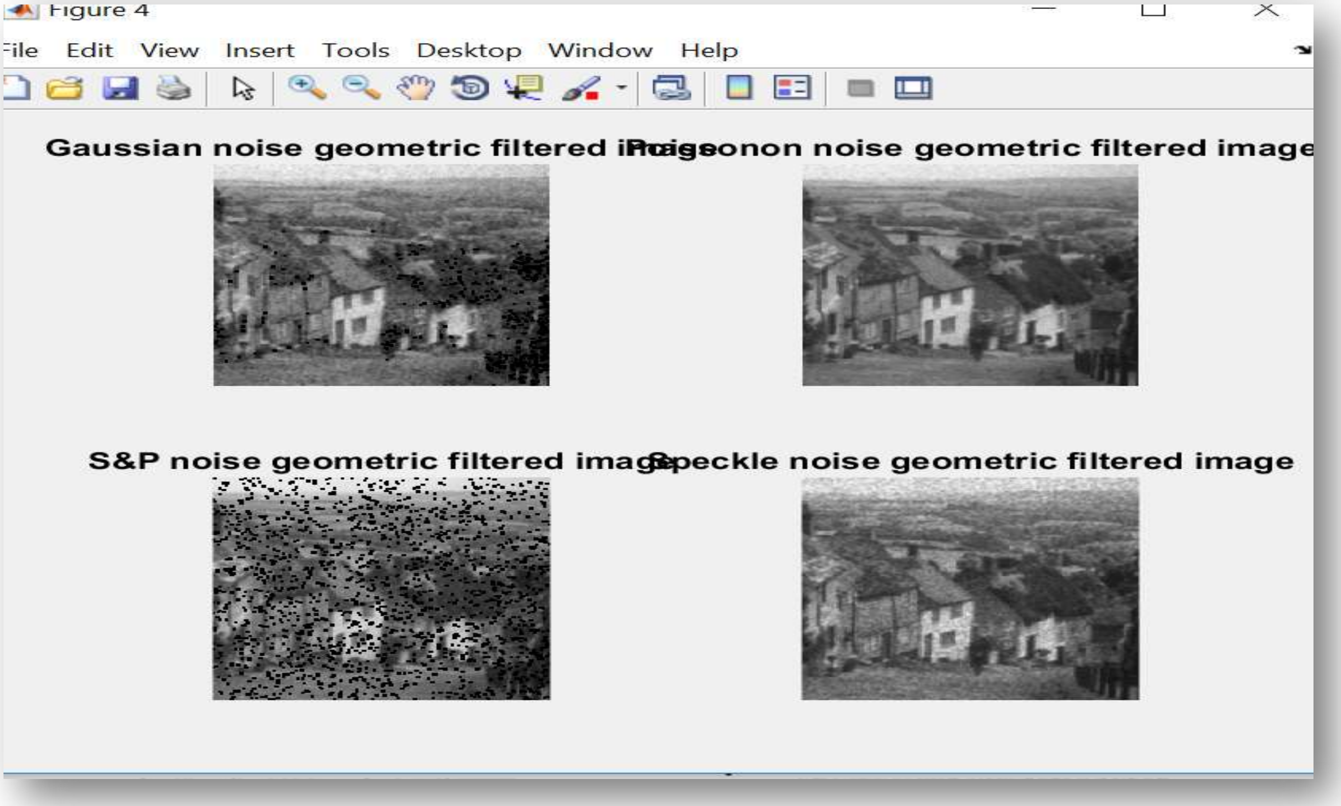


Figure 4
File Edit View Insert Tools Desktop Window Help
Gaussian noise geometric filtered iPoisgsonon noise geometric filtered image
S&P noise geometric filtered imagSpeckle noise geometric filtered image

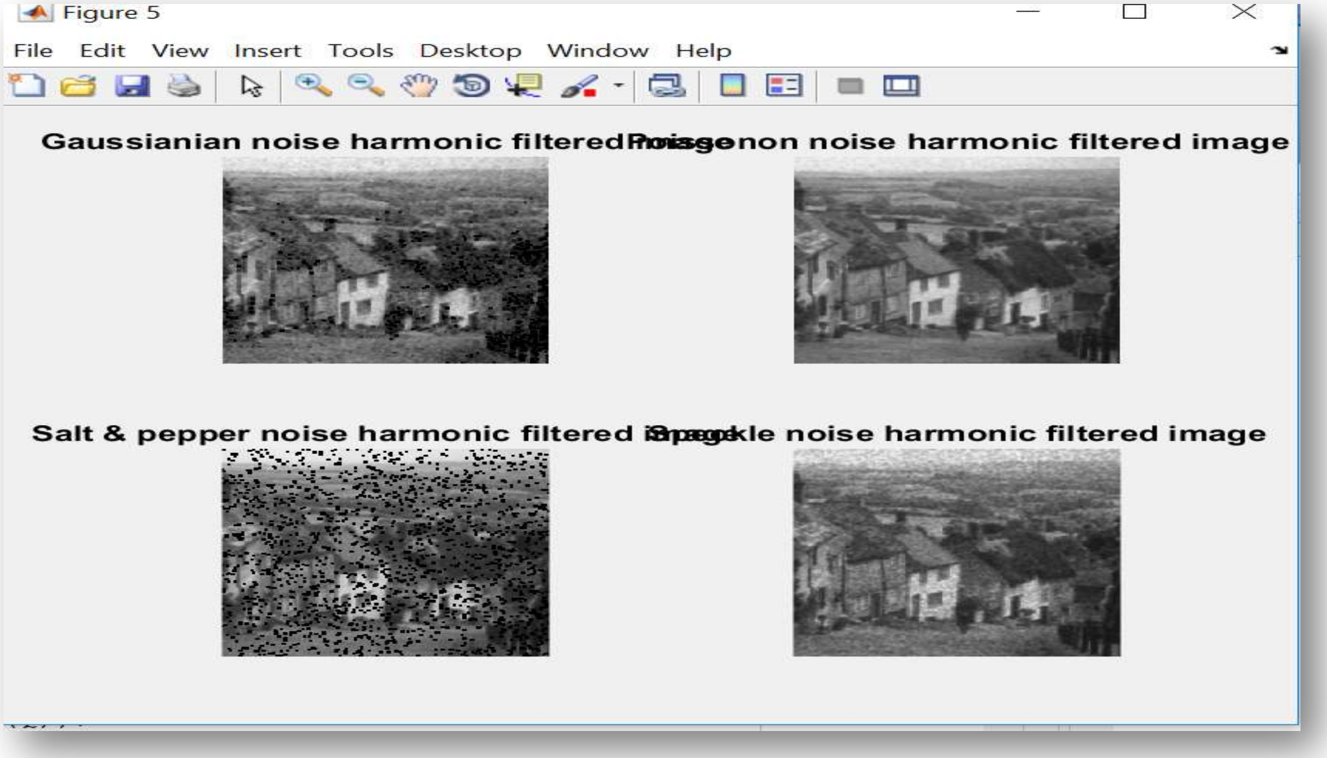


Figure 5
File Edit View Insert Tools Desktop Window Help
Gaussianian noise harmonic filtered image
Poissonon noise harmonic filtered image
Salt & pepper noise harmonic filtered image
Speckle noise harmonic filtered image

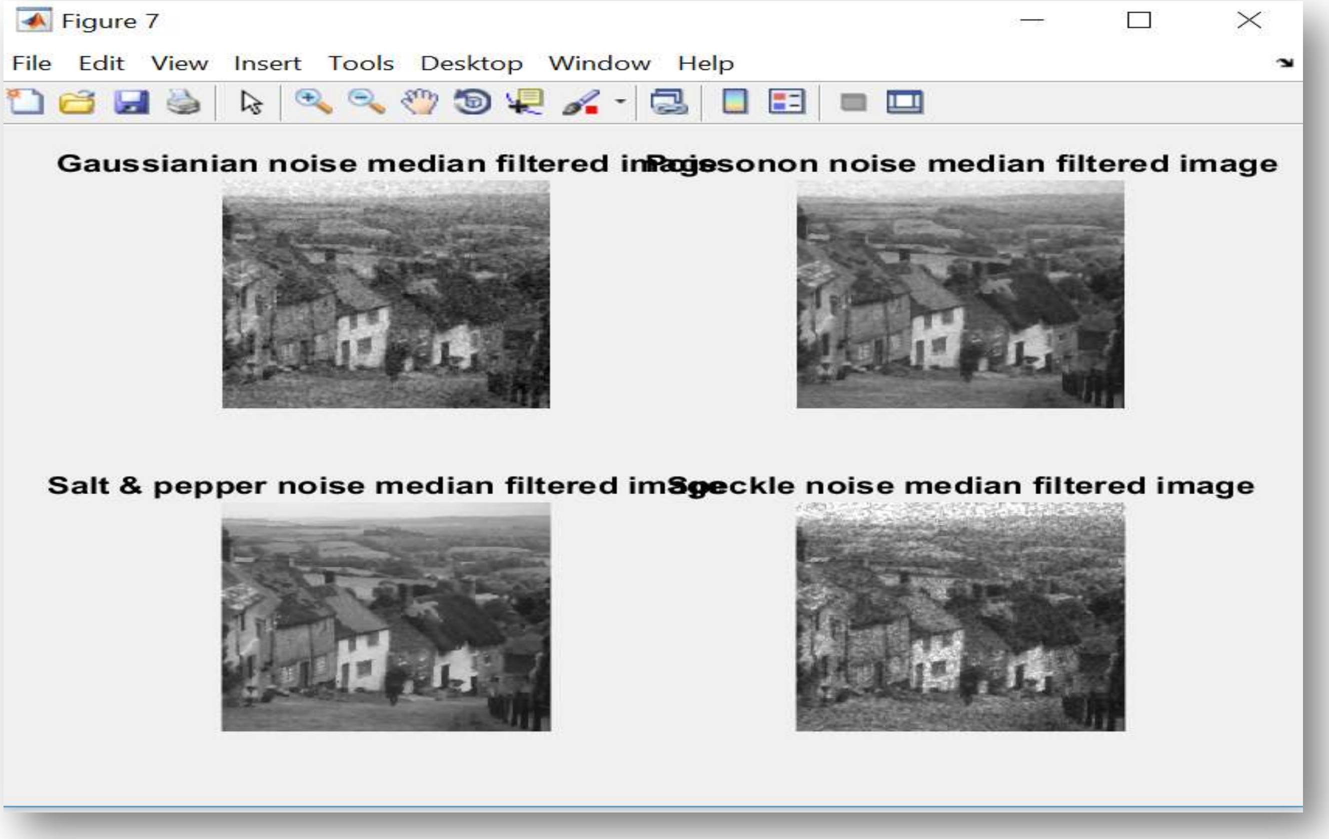


Figure 7
File Edit View Insert Tools Desktop Window Help
Gaussianian noise median filtered image
Poissonon noise median filtered image
Salt & pepper noise median filtered image
Speckle noise median filtered image

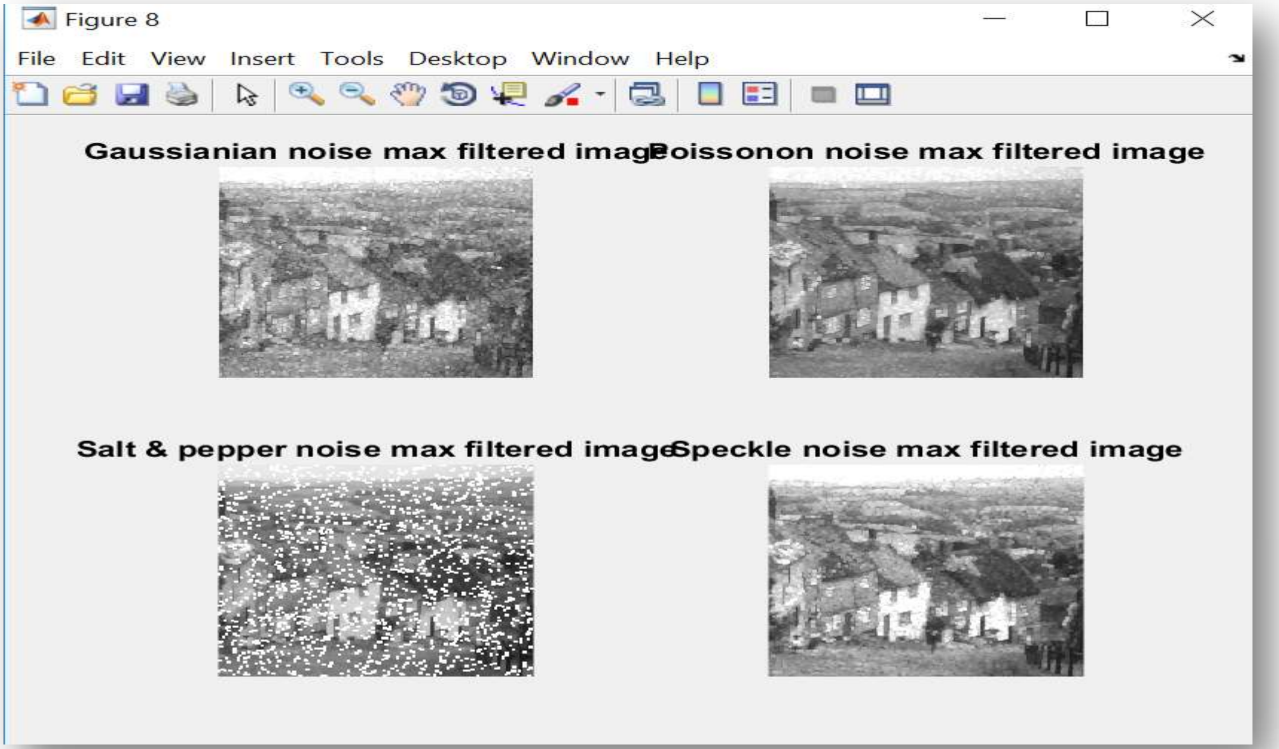


Figure 8
File Edit View Insert Tools Desktop Window Help
Gaussianian noise max filtered image
Poissonon noise max filtered image
Salt & pepper noise max filtered image
Speckle noise max filtered image

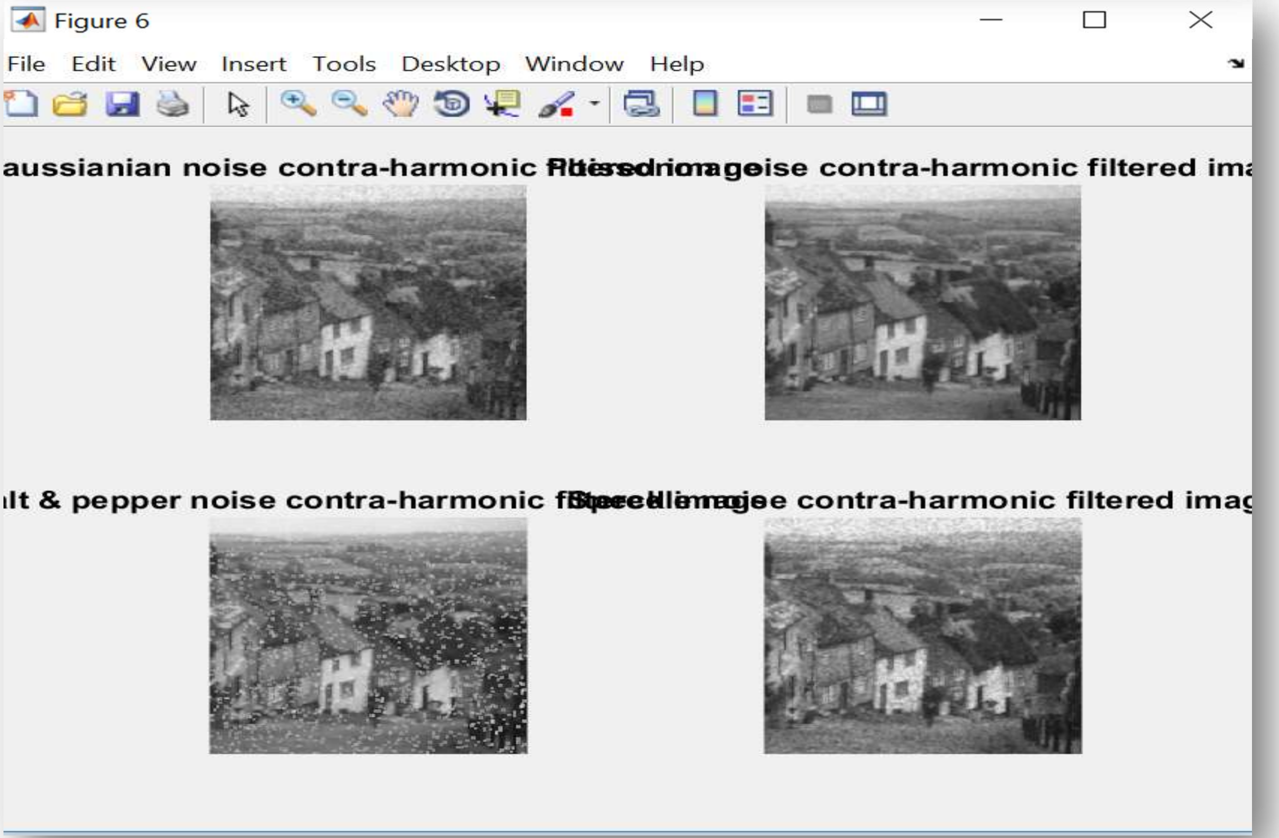
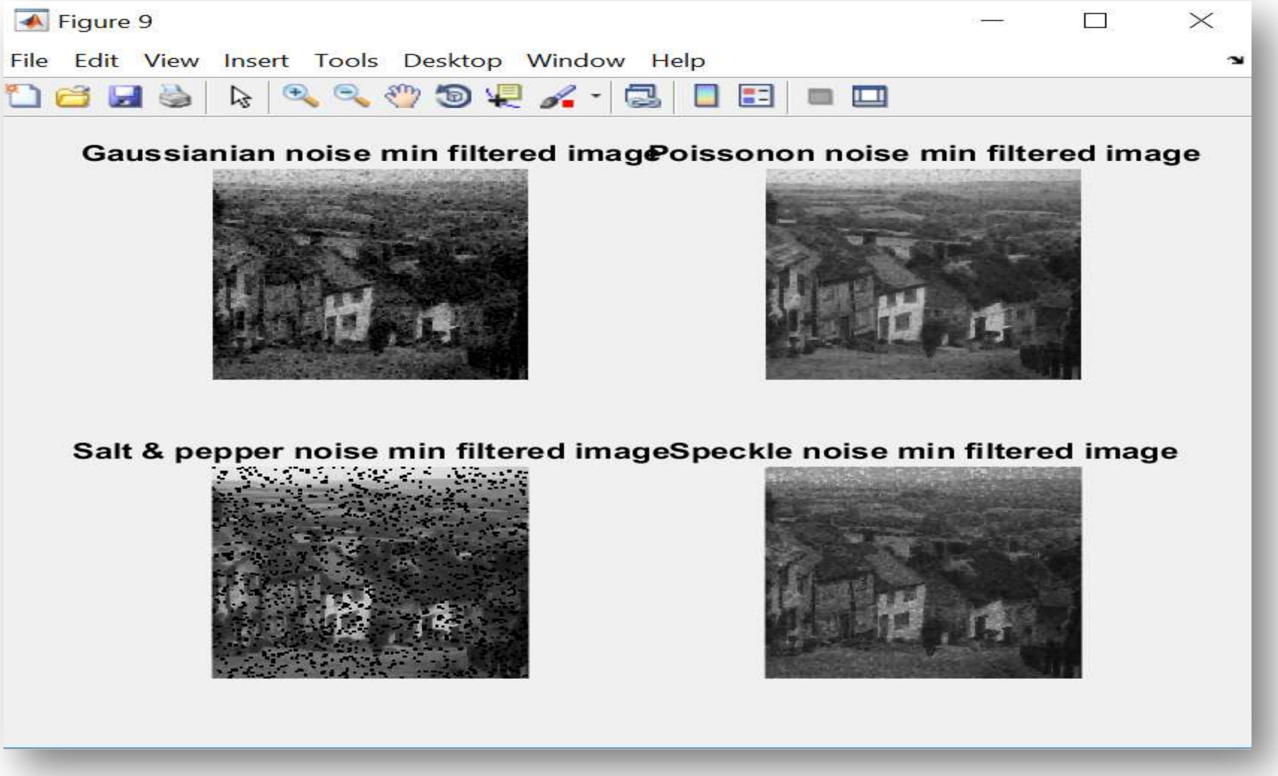


Figure 6
File Edit View Insert Tools Desktop Window Help
aussianian noise contra-harmonic filtered image
Poissonon noise contra-harmonic filtered ima
lt & pepper noise contra-harmonic filtered image
Speckle noise contra-harmonic filtered imag

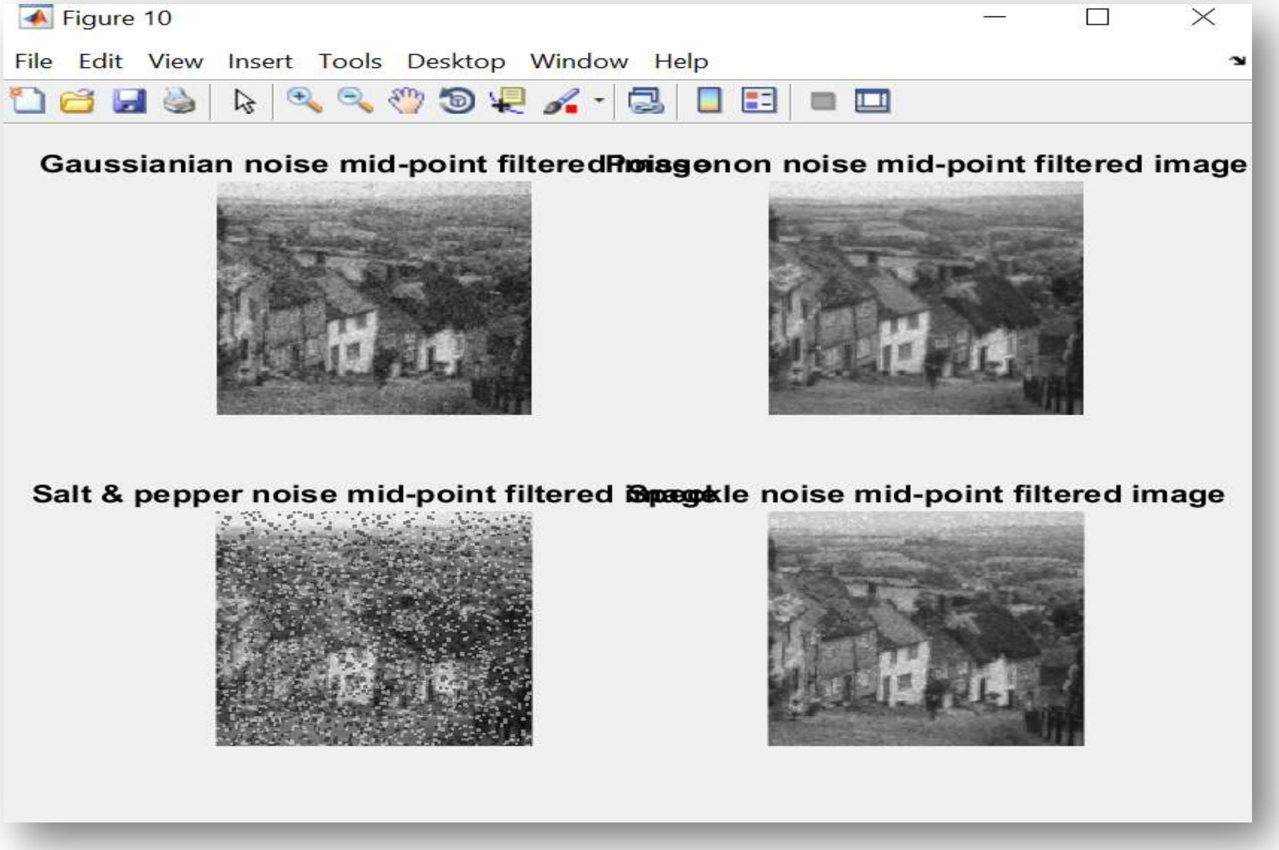


Gaussianian noise min filtered image

Poissonon noise min filtered image

Salt & pepper noise min filtered image

Speckle noise min filtered image



Gaussianian noise mid-point filtered image

Poissonon noise mid-point filtered image

Salt & pepper noise mid-point filtered image

Speckle noise mid-point filtered image

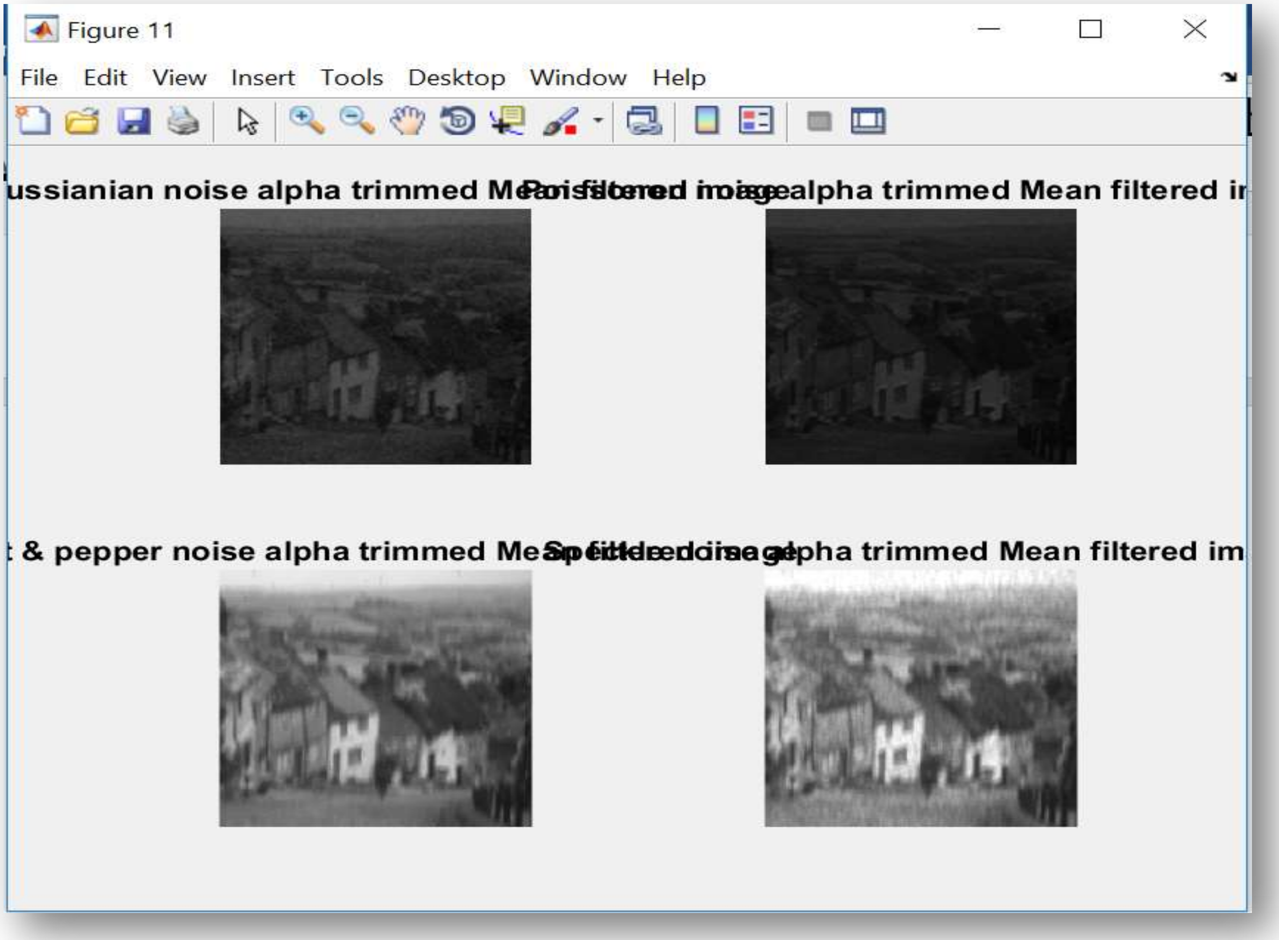


Figure 11
File Edit View Insert Tools Desktop Window Help
ussianian noise alpha trimmed Mean filtered image
Poisson noise alpha trimmed Mean filtered ir
t & pepper noise alpha trimmed Mean filtered image
Speckle noise alpha trimmed Mean filtered im

CONCLUSION:

For each of the above noise, the best filtering method to nullify the effect of that specific noise is listed below.

- **Gaussian noise**: Arithmetic filter seems to give the clearest output.
- **Poisson Noise:** Alpha trimmed, arithmetic, geometric, harmonic and contra-harmonic provide a better and clear output.
- **Salt & Pepper Noise:** Median filter and alpha mean trimmed filters seem to provide the best output
- **Speckle Noise:** Mid-point filter seems to remove most of the noise. After mid-point, arithmetic, geometric and harmonic do a decent job of removing the noise.