```
%Project Submission by: MANISH SONI

close all;
clear all;
%Lets load three differenet images for this project


Input_img1 = double(rgb2gray(imread('mountain-scene-2.jpg')));
Input_img2 = double(rgb2gray(imread('house_image-1.jpg')));
Input_img3 = double(rgb2gray(imread('Omega_in_Flight.jpg')));


[num_row1,num_col1] = size(Input_img1);
row_cen_1 = round(num_row1/2);
col_cen_1 = round(num_col1/2) +1;


[num_row2,num_col2] = size(Input_img2);
row_cen_2 = round(num_row2/2);
col_cen_2 = round(num_col2/2) +1;


[num_row3,num_col3] = size(Input_img3);
row_cen_3 = round(num_row3/2);
col_cen_3 = round(num_col3/2) +1;


% lets make the size of all the images same
halfwid = 160;
Original_img1 = Input_img1((row_cen_1-halfwid):(row_cen_1+halfwid),
(col_cen_1-halfwid):(col_cen_1+halfwid));
Original_img2 = Input_img2((row_cen_2-halfwid):(row_cen_2+halfwid),
(col_cen_2-halfwid):(col_cen_2+halfwid));
Original_img3 = Input_img3((row_cen_3-halfwid):(row_cen_3+halfwid),
(col_cen_3-halfwid):(col_cen_3+halfwid));


[length,length] = size(Original_img1);

figure
subplot(1,3,1)
imagesc(Original_img1)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Input Test Image Target 1'


subplot(1,3,2)
imagesc(Original_img2)
axis 'off'
axis 'image'
```

```
colormap(gray(256))
title 'Input Test Image Target 2'

subplot(1,3,3)
imagesc(Original_img3)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Input Test Image Target 3'




%Lets add gaussian noise
mean = 6;
std = 4;
probability_distribution_object =
 makedist('Normal','mu',mean,'sigma',std);
gaussian_noise = random(probability_distribution_object, [length,
 length]);

gaussian_noise_target_1 = Original_img1 + round(gaussian_noise);
[N,edges] = histcounts(gaussian_noise_target_1,256);
figure
subplot(2,1,1)
imagesc(gaussian_noise_target_1)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 1 with Gaussian Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 1 with Gaussian Noise'




gaussian_noise_target_2 = Original_img2 + round(gaussian_noise);
[N,edges] = histcounts(gaussian_noise_target_2,256);
figure
subplot(2,1,1)
imagesc(gaussian_noise_target_2)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 2 with Gaussian Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
```

```
title 'Histogram of Target 2 with Gaussian Noise'

gaussian_noise_target_3 = Original_img3 + round(gaussian_noise);
[N,edges] = histcounts(gaussian_noise_target_3,256);
figure
subplot(2,1,1)
imagesc(gaussian_noise_target_3)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 3 with Gaussian Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 3 with Gaussian Noise'



% Rayleigh noise
% lets take b value to 4
b=4;
probability_distribution_object = makedist('Rayleigh','b',b);
rayleigh_noise = random(probability_distribution_object,
[length,length]);
rayleigh_noise_target_1 = Original_img1 +
 round(rayleigh_noise); %makes integer valued so that we can use
 histcnts
[N,edges] = histcounts(rayleigh_noise_target_1,256);
figure
subplot(2,1,1)
imagesc(rayleigh_noise_target_1)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 1 with Rayleigh Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 1 with Rayleigh Noise'



rayleigh_noise_target_2 = Original_img2 +
 round(rayleigh_noise); %makes integer valued so that we can use
 histcnts
[N,edges] = histcounts(rayleigh_noise_target_2,256);
figure
subplot(2,1,1)
imagesc(rayleigh_noise_target_2)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 2 with Rayleigh Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 2 with Rayleigh Noise'


rayleigh_noise_target_3 = Original_img3 +
 round(rayleigh_noise); %makes integer valued so that we can use
 histcnts
[N,edges] = histcounts(rayleigh_noise_target_3,256);
figure
subplot(2,1,1)
imagesc(rayleigh_noise_target_3)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 3 with Rayleigh Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 3 with Rayleigh Noise'



%lets use imnoise function for salt and pepper noise with noise
 density d value as 0.4
% lets keep input to it normalized in the range [0,1]

d = 0.2; % Noise density
normalized_target1 = Original_img1/max(max(Original_img1));
normalized_target2 = Original_img2/max(max(Original_img2));
normalized_target3 = Original_img3/max(max(Original_img3));

saltpepper_noise_target1 =
 max(max(Original_img1))*imnoise(normalized_target1,'salt &
 pepper',d);
[N,edges] = histcounts(saltpepper_noise_target1,256);
figure
subplot(2,1,1)
imagesc(saltpepper_noise_target1)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 1 with Salt and Pepper Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
```

```
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 1 with Salt and Pepper Noise'



saltpepper_noise_target2 =
 max(max(Original_img2))*imnoise(normalized_target2,'salt &
 pepper',d);
[N,edges] = histcounts(saltpepper_noise_target2,256);
figure
subplot(2,1,1)
imagesc(saltpepper_noise_target2)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 2 with Salt and Pepper Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 2 with Salt and Pepper Noise'


saltpepper_noise_target3 =
 max(max(Original_img3))*imnoise(normalized_target3,'salt &
 pepper',d);
[N,edges] = histcounts(saltpepper_noise_target3,256);
figure
subplot(2,1,1)
imagesc(saltpepper_noise_target3)
axis 'off'
axis 'image'
colormap(gray(256))
title 'Target 3 with Salt and Pepper Noise'
subplot(2,1,2)
bar(N)
ylabel 'Counts of Pixels'
xlabel 'Varying Gray Scale'
axis([0 255 0 50*length])
title 'Histogram of Target 3 with Salt and Pepper Noise'

saltpepper_noise1=saltpepper_noise_target1-Original_img1;
saltpepper_noise2=saltpepper_noise_target2-Original_img2;
saltpepper_noise3=saltpepper_noise_target3-Original_img3;
```

```
% Harmonic mean filter here. I've created a separate function in file
% apply_harmonic_filter.m



% harmonic example
output_gaus_img1=apply_harmonic_filter(gaussian_noise_target_1,'Gaussian
 Target 1 inversed image','Output using harmonic filter');
ssim(output_gaus_img1,Original_img1)
output_gaus_img2=apply_harmonic_filter(gaussian_noise_target_2,'Gaussian
 Target 2 inversed image','Output using harmonic filter');
ssim(output_gaus_img2,Original_img2)
output_gaus_img3=apply_harmonic_filter(gaussian_noise_target_3,'Gaussian
 Target 3 inversed image','Output using harmonic filter');
ssim(output_gaus_img3,Original_img3)


['---------SSIM output of Harmonic median filtered Gaussian Noisy
 Image and original-------------------------']
['SSIM Output result between filtered Gaussian Target 1 and Original
 Image 1 = ',num2str(100*ssim(output_gaus_img1,Original_img1)),'%']
['SSIM Output result between filtered Gaussian Target 2 and Original
 Image 2 = ',num2str(100*ssim(output_gaus_img2,Original_img2)),'%']
['SSIM Output result between filtered Gaussian Target 3 and Original
 Image 3 = ',num2str(100*ssim(output_gaus_img3,Original_img3)),'%']
['-------end-SSIM output of Harmonic median filtered Gaussian Noisy
 Image and original-------------------------']




output_rayleigh_img1=apply_harmonic_filter(rayleigh_noise_target_1,'Rayleigh
 Target 1 inversed image','Output using harmonic filter');
ssim(output_rayleigh_img1,Original_img1)
output_rayleigh_img2=apply_harmonic_filter(rayleigh_noise_target_2,'Rayleigh
 Target 2 inversed image','Output using harmonic filter');
ssim(output_rayleigh_img2,Original_img2)
output_rayleigh_img3=apply_harmonic_filter(rayleigh_noise_target_3,'Rayleigh
 Target 3 inversed image','Output using harmonic filter');
ssim(output_rayleigh_img3,Original_img3)


['---------SSIM output of Harmonic median filtered Rayleigh Noisy
 Image and original-------------------------']
['SSIM Output result between filtered
 Rayleigh Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_rayleigh_img1,Original_img1)),'%']
['SSIM Output result between filtered
 Rayleigh Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_rayleigh_img2,Original_img2)),'%']
['SSIM Output result between filtered
 Rayleigh Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_rayleigh_img3,Original_img3)),'%']
```

```
['------end--SSIM output of Harmonic median filtered Rayleigh Noisy
 Image and original--------------------------']




output_saltpepper_img1=apply_harmonic_filter(saltpepper_noise_target1,'Salt
 and Pepper Target 1 inversed image','Output using harmonic filter');
ssim(output_saltpepper_img1,Original_img1)
output_saltpepper_img2=apply_harmonic_filter(saltpepper_noise_target2,'Salt
 and Pepper Target 2 inversed image','Output using harmonic filter');
ssim(output_saltpepper_img2,Original_img2)
output_saltpepper_img3=apply_harmonic_filter(saltpepper_noise_target3,'Salt
 and Pepper Target 3 inversed image','Output using harmonic filter');
ssim(output_saltpepper_img3,Original_img3)

['---------SSIM output of Harmonic median filtered Noisy Image and
 original--------------------------']
['SSIM Output result between filtered Salt
 and Pepper Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_saltpepper_img1,Original_img1)),'%']
['SSIM Output result between filtered Salt
 and Pepper Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_saltpepper_img2,Original_img2)),'%']
['SSIM Output result between filtered Salt
 and Pepper Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_saltpepper_img3,Original_img3)),'%']
['--------end-SSIM output of Harmonic median filtered Noisy Image and
 original--------------------------']


% adaptive local noise filter

% lets use adaptive local noise filter here. I've created a separate
 function in file
% adaptive_local_noise_filter.m



filterOutput='adaptive local filtered image';

output_gaus_img1=adaptive_local_noise_filter(gaussian_noise_target_1,gaussian_nois
 Target 1 noisy  image',filterOutput);
ssim(output_gaus_img1,Original_img1)
output_gaus_img2=adaptive_local_noise_filter(gaussian_noise_target_2,gaussian_nois
 Target 2 noisy  image',filterOutput);
ssim(output_gaus_img2,Original_img2)
output_gaus_img3=adaptive_local_noise_filter(gaussian_noise_target_3,gaussian_nois
 Target 3 noisy  image',filterOutput);
ssim(output_gaus_img3,Original_img3)

['---------SSIM output of Adaptive Local filtered Gaussian Noisy Image
 and original--------------------------']
```

```
['SSIM Output result between filtered Gaussian Target 1 and Original
 Image 1 = ',num2str(100*ssim(output_gaus_img1,Original_img1)),'%']
['SSIM Output result between filtered Gaussian Target 2 and Original
 Image 2 = ',num2str(100*ssim(output_gaus_img2,Original_img2)),'%']
['SSIM Output result between filtered Gaussian Target 3 and Original
 Image 3 = ',num2str(100*ssim(output_gaus_img3,Original_img3)),'%']
['------end--SSIM output of Adaptive Local filtered Gaussian Noisy
 Image and original-------------------------']




output_rayleigh_img1=adaptive_local_noise_filter(rayleigh_noise_target_1,rayleigh_
 Target 1 noisy  image',filterOutput);
ssim(output_rayleigh_img1,Original_img1)
output_rayleigh_img2=adaptive_local_noise_filter(rayleigh_noise_target_2,rayleigh_
 Target 2 noisy  image',filterOutput);
ssim(output_rayleigh_img2,Original_img2)
output_rayleigh_img3=adaptive_local_noise_filter(rayleigh_noise_target_3,rayleigh_
 Target 3 noisy  image',filterOutput);
ssim(output_rayleigh_img3,Original_img3)

['---------SSIM output of Adaptive Local filtered Rayleigh Noisy Image
 and original-------------------------']
['SSIM Output result between filtered
 Rayleigh Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_rayleigh_img1,Original_img1)),'%']
['SSIM Output result between filtered
 Rayleigh Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_rayleigh_img2,Original_img2)),'%']
['SSIM Output result between filtered
 Rayleigh Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_rayleigh_img3,Original_img3)),'%']
['-------end-SSIM output of Adaptive Local filtered Rayleigh Noisy
 Image and original-------------------------']



output_saltpepper_img1=adaptive_local_noise_filter(saltpepper_noise_target1,saltpe
 and Pepper Target 1 noisy  image',filterOutput);
ssim(output_saltpepper_img1,Original_img1)
output_saltpepper_img2=adaptive_local_noise_filter(saltpepper_noise_target2,saltpe
 and Pepper Target 2 noisy  image',filterOutput);
ssim(output_saltpepper_img2,Original_img2)
output_saltpepper_img3=adaptive_local_noise_filter(saltpepper_noise_target3,saltpe
 and Pepper Target 3 noisy  image',filterOutput);
ssim(output_saltpepper_img3,Original_img3)

['---------SSIM output of Adaptive Local filtered Noisy Image and
 original-------------------------']
['SSIM Output result between filtered Salt
 and Pepper Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_saltpepper_img1,Original_img1)),'%']
```

```
['SSIM Output result between filtered Salt
 and Pepper Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_saltpepper_img2,Original_img2)),'%']
['SSIM Output result between filtered Salt
 and Pepper Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_saltpepper_img3,Original_img3)),'%']
['--------end-SSIM output of Adaptive Local filtered Noisy Image and
 original-------------------------']


% adaptive median filter

% lets use adaptive median filter here. I've created a separate
 function in file
% apply_adaptic_median_filter.m




filterOutput='adaptive median filtered image';

output_gaus_img1=apply_adaptic_median_filter(gaussian_noise_target_1,'Gaussian
 Target 1 noisy  image',filterOutput);
ssim(output_gaus_img1,Original_img1)
output_gaus_img2=apply_adaptic_median_filter(gaussian_noise_target_2,'Gaussian
 Target 2 noisy  image',filterOutput);
ssim(output_gaus_img2,Original_img2)
output_gaus_img3=apply_adaptic_median_filter(gaussian_noise_target_3,'Gaussian
 Target 3 noisy  image',filterOutput);
ssim(output_gaus_img3,Original_img3)

['---------SSIM output of Adaptive median filtered Gaussian Noisy
 Image and original-------------------------']
['SSIM Output result between filtered Gaussian Target 1 and Original
 Image 1 = ',num2str(100*ssim(output_gaus_img1,Original_img1)),'%']
['SSIM Output result between filtered Gaussian Target 2 and Original
 Image 2 = ',num2str(100*ssim(output_gaus_img2,Original_img2)),'%']
['SSIM Output result between filtered Gaussian Target 3 and Original
 Image 3 = ',num2str(100*ssim(output_gaus_img3,Original_img3)),'%']
['-------end-SSIM output of Adaptive median filtered Gaussian Noisy
 Image and original-------------------------']



output_rayleigh_img1=apply_adaptic_median_filter(rayleigh_noise_target_1,'Rayleigh
 Target 1 noisy  image',filterOutput);
ssim(output_rayleigh_img1,Original_img1)
output_rayleigh_img2=apply_adaptic_median_filter(rayleigh_noise_target_2,'Rayleigh
 Target 2 noisy  image',filterOutput);
ssim(output_rayleigh_img2,Original_img2)
output_rayleigh_img3=apply_adaptic_median_filter(rayleigh_noise_target_3,'Rayleigh
 Target 3 noisy  image',filterOutput);
ssim(output_rayleigh_img3,Original_img3)
```

```
['---------SSIM output of Adaptive median filtered Rayleigh Noisy
 Image and original-------------------------']
['SSIM Output result between filtered
 Rayleigh Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_rayleigh_img1,Original_img1)),'%']
['SSIM Output result between filtered
 Rayleigh Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_rayleigh_img2,Original_img2)),'%']
['SSIM Output result between filtered
 Rayleigh Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_rayleigh_img3,Original_img3)),'%']
['------end--SSIM output of Adaptive median filtered Rayleigh Noisy
 Image and original-------------------------']
```

```
output_saltpepper_img1=apply_adaptic_median_filter(saltpepper_noise_target1,'Salt
 and Pepper Target 1 noisy  image',filterOutput);
ssim(output_saltpepper_img1,Original_img1)
output_saltpepper_img2=apply_adaptic_median_filter(saltpepper_noise_target2,'Salt
 and Pepper Target 2 noisy  image',filterOutput);
ssim(output_saltpepper_img2,Original_img2)
output_saltpepper_img3=apply_adaptic_median_filter(saltpepper_noise_target3,'Salt
 and Pepper Target 3 noisy  image',filterOutput);
ssim(output_saltpepper_img3,Original_img3)
['----- ssim
 output',num2str(100*ssim(output_saltpepper_img3,Original_img3)),'%']

['---------SSIM output of Adaptive median filtered Noisy Image and
 original-------------------------']
['SSIM Output result between filtered Salt
 and Pepper Target 1 and Original Image 1 =
 ',num2str(100*ssim(output_saltpepper_img1,Original_img1)),'%']
['SSIM Output result between filtered Salt
 and Pepper Target 2 and Original Image 2 =
 ',num2str(100*ssim(output_saltpepper_img2,Original_img2)),'%']
['SSIM Output result between filtered Salt
 and Pepper Target 3 and Original Image 3 =
 ',num2str(100*ssim(output_saltpepper_img3,Original_img3)),'%']
['--------end-SSIM output of Adaptive median filtered Noisy Image and
 original-------------------------']

% Conclusion: I've completed this Project and have commented relavant
% section of code and have prepared a report.
```

```
ans =

    0.6602
```

```
ans =

    0.6346


ans =

    0.3879


ans =

    '---------SSIM output of Harmonic median filtered Gaussian Noisy
 Image and original--------------------------'


ans =

    'SSIM Output result between filtered Gaussian Target 1 and
 Original Image 1 = 66.0224%'


ans =

    'SSIM Output result between filtered Gaussian Target 2 and
 Original Image 2 = 63.457%'


ans =

    'SSIM Output result between filtered Gaussian Target 3 and
 Original Image 3 = 38.7851%'


ans =

    '-------end-SSIM output of Harmonic median filtered Gaussian Noisy
 Image and original--------------------------'


ans =

    0.6768


ans =

    0.6523


ans =

    0.4814
```

```
ans =

    '---------SSIM output of Harmonic median filtered Rayleigh Noisy
 Image and original-------------------------'


ans =

    'SSIM Output result between filtered Rayleigh Target 1 and
 Original Image 1 = 67.6843%'


ans =

    'SSIM Output result between filtered Rayleigh Target 2 and
 Original Image 2 = 65.2261%'


ans =

    'SSIM Output result between filtered Rayleigh Target 3 and
 Original Image 3 = 48.1401%'


ans =

    '------end--SSIM output of Harmonic median filtered Rayleigh Noisy
 Image and original-------------------------'


ans =

    0.0339


ans =

    0.0455


ans =

    0.0105


ans =

    '---------SSIM output of Harmonic median filtered Noisy Image and
 original-------------------------'


ans =
    'SSIM Output result between filtered Salt and Pepper Target 1 and
 Original Image 1 = 3.3939%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 2 and
 Original Image 2 = 4.5464%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 3 and
 Original Image 3 = 1.0451%'


ans =

    '--------end-SSIM output of Harmonic median filtered Noisy Image
 and original-------------------------'


ans =

    0.6895


ans =

    0.6699


ans =

    0.3790


ans =

    '---------SSIM output of Adaptive Local filtered Gaussian Noisy
 Image and original-------------------------'


ans =

    'SSIM Output result between filtered Gaussian Target 1 and
 Original Image 1 = 68.9522%'


ans =

    'SSIM Output result between filtered Gaussian Target 2 and
 Original Image 2 = 66.9912%'
```

```
ans =

    'SSIM Output result between filtered Gaussian Target 3 and
     Original Image 3 = 37.9002%'


ans =

    '------end--SSIM output of Adaptive Local filtered Gaussian Noisy
     Image and original--------------------------'


ans =

    0.7120


ans =

    0.6868


ans =

    0.4722


ans =

    '---------SSIM output of Adaptive Local filtered Rayleigh Noisy
     Image and original--------------------------'


ans =

    'SSIM Output result between filtered Rayleigh Target 1 and
     Original Image 1 = 71.205%'


ans =

    'SSIM Output result between filtered Rayleigh Target 2 and
     Original Image 2 = 68.6768%'


ans =

    'SSIM Output result between filtered Rayleigh Target 3 and
     Original Image 3 = 47.2246%'


ans =
    '-------end-SSIM output of Adaptive Local filtered Rayleigh Noisy
 Image and original-------------------------'


ans =

    0.1978


ans =

    0.2502


ans =

    0.0801


ans =

    '---------SSIM output of Adaptive Local filtered Noisy Image and
 original-------------------------'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 1 and
 Original Image 1 = 19.7792%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 2 and
 Original Image 2 = 25.0168%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 3 and
 Original Image 3 = 8.0084%'


ans =

    '--------end-SSIM output of Adaptive Local filtered Noisy Image
 and original-------------------------'


ans =

    0.6391
```

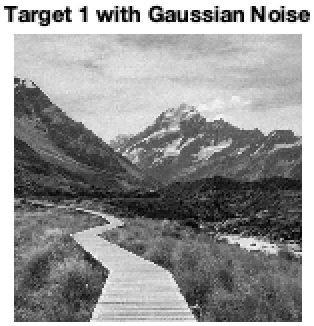
```
ans =

    0.6862


ans =

    0.3160


ans =

    '---------SSIM output of Adaptive median filtered Gaussian Noisy
 Image and original--------------------------'


ans =

    'SSIM Output result between filtered Gaussian Target 1 and
 Original Image 1 = 63.9082%'


ans =

    'SSIM Output result between filtered Gaussian Target 2 and
 Original Image 2 = 68.6235%'


ans =

    'SSIM Output result between filtered Gaussian Target 3 and
 Original Image 3 = 31.6017%'


ans =

    '-------end-SSIM output of Adaptive median filtered Gaussian Noisy
 Image and original--------------------------'


ans =

    0.6763


ans =

    0.7147


ans =

    0.3992
```

```
ans =

    '---------SSIM output of Adaptive median filtered Rayleigh Noisy
 Image and original-------------------------'


ans =

    'SSIM Output result between filtered Rayleigh Target 1 and
 Original Image 1 = 67.6308%'


ans =

    'SSIM Output result between filtered Rayleigh Target 2 and
 Original Image 2 = 71.4674%'


ans =

    'SSIM Output result between filtered Rayleigh Target 3 and
 Original Image 3 = 39.92%'


ans =

    '------end--SSIM output of Adaptive median filtered Rayleigh Noisy
 Image and original-------------------------'


ans =

    0.6733


ans =

    0.7201


ans =

    0.6350


ans =

    '----- ssim output63.5037%'


ans =
    '---------SSIM output of Adaptive median filtered Noisy Image and
 original--------------------------'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 1 and
 Original Image 1 = 67.3306%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 2 and
 Original Image 2 = 72.0122%'


ans =

    'SSIM Output result between filtered Salt and Pepper Target 3 and
 Original Image 3 = 63.5037%'


ans =

    '--------end-SSIM output of Adaptive median filtered Noisy Image
 and original--------------------------'
```

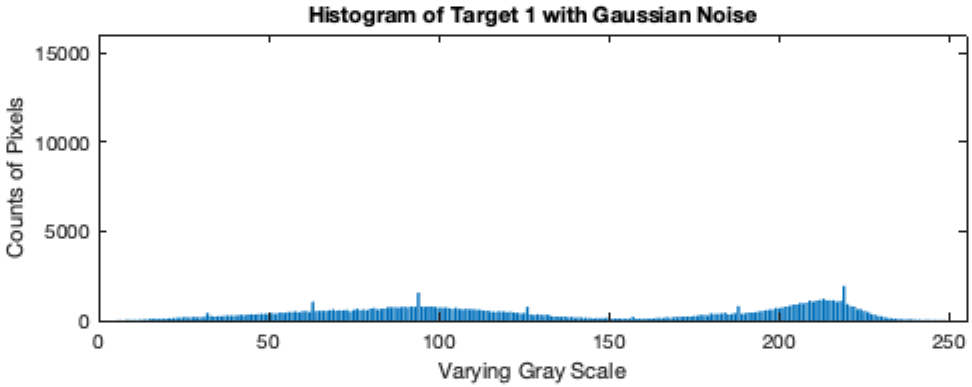
Input Test Image Target 1

Input Test Image Target 2

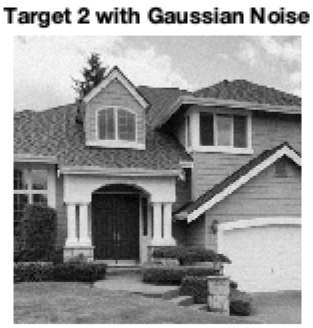
Input Test Image Target 3



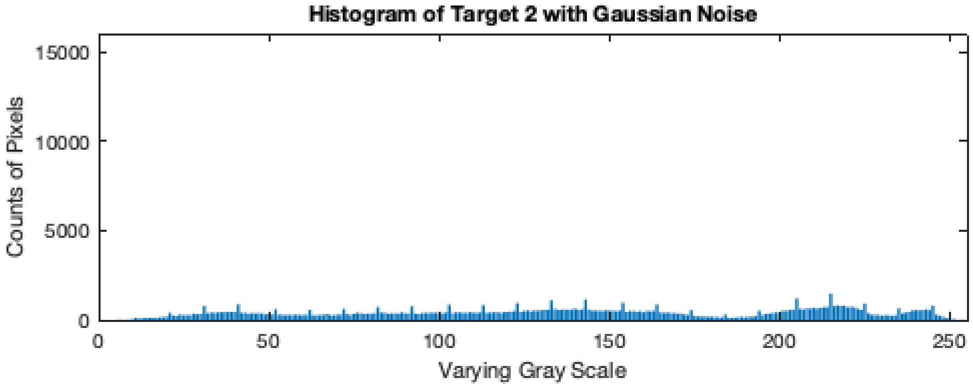
Target 1 with Gaussian Noise



Histogram of Target 1 with Gaussian Noise

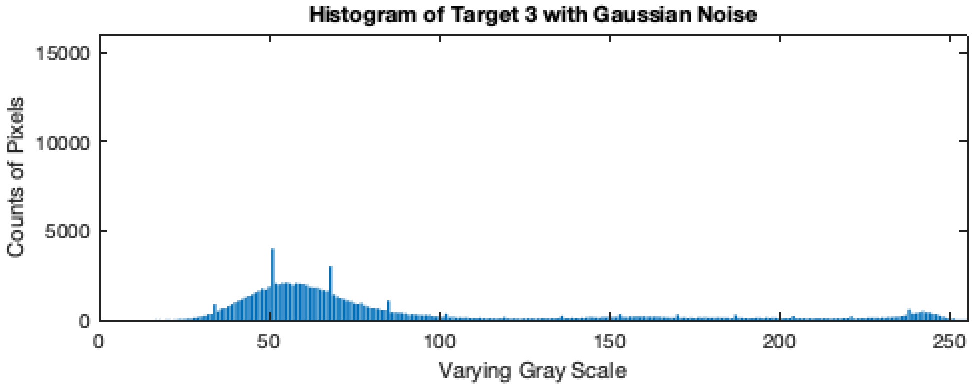


Target 2 with Gaussian Noise

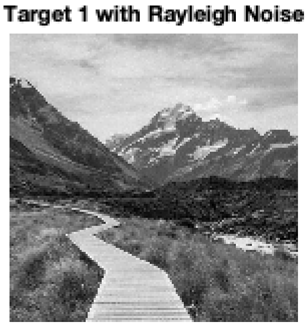


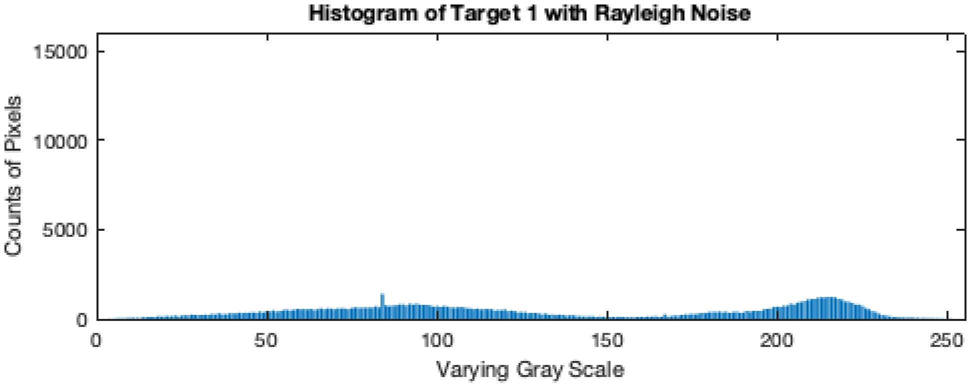
Histogram of Target 2 with Gaussian Noise

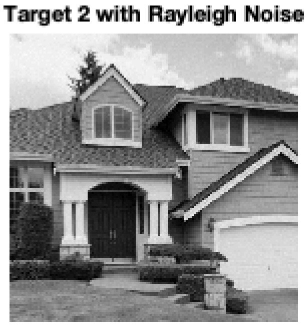
Target 3 with Gaussian Noise

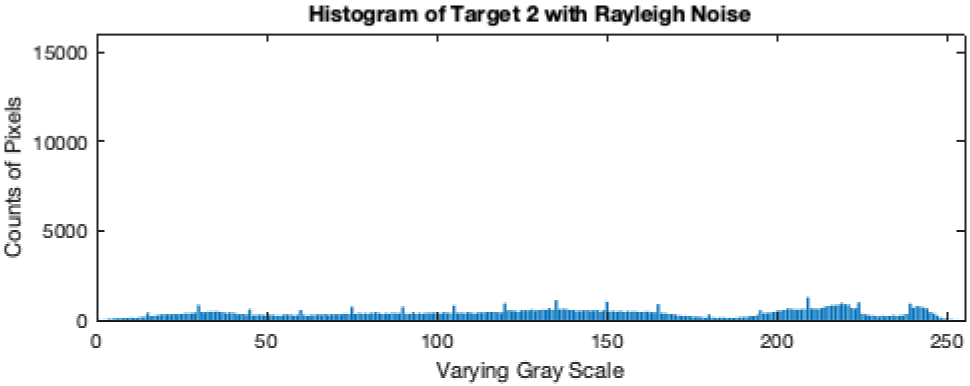


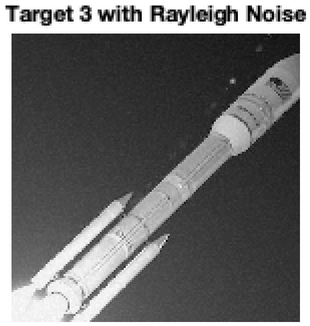
Histogram of Target 3 with Gaussian Noise

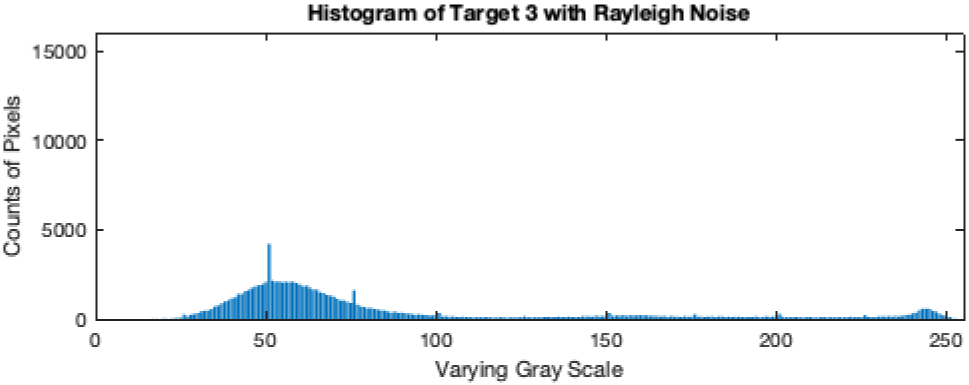

Target 1 with Rayleigh Noise

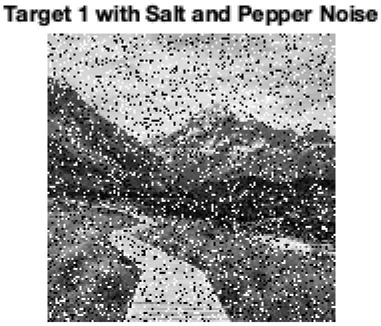

Histogram of Target 1 with Rayleigh Noise

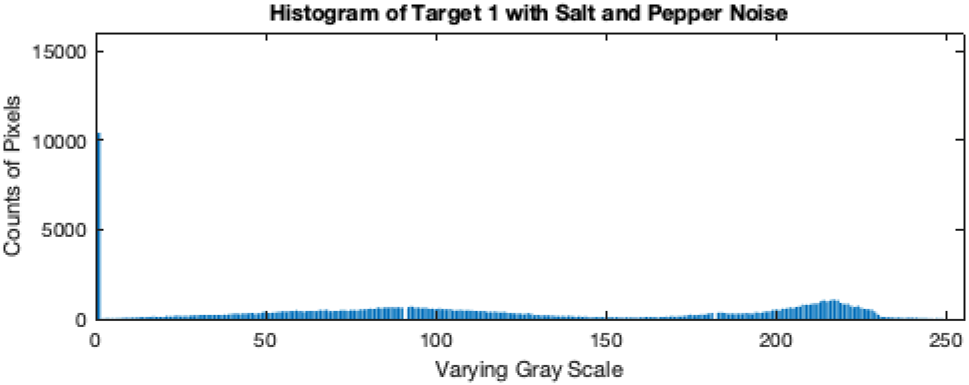

Target 2 with Rayleigh Noise

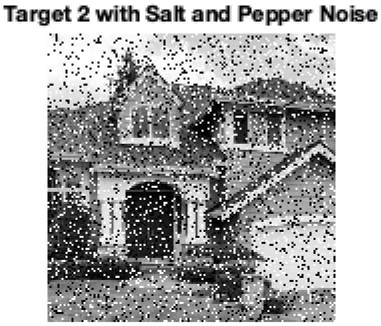

Histogram of Target 2 with Rayleigh Noise


Target 3 with Rayleigh Noise

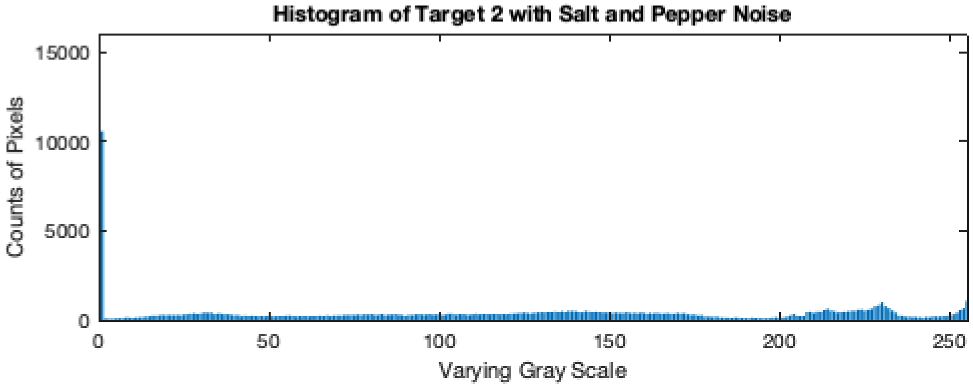

Histogram of Target 3 with Rayleigh Noise
Counts of Pixels
15000
10000
5000
0
0
50
100
150
200
250
Varying Gray Scale


Target 1 with Salt and Pepper Noise

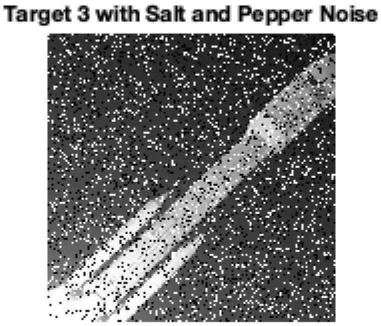

Histogram of Target 1 with Salt and Pepper Noise
Counts of Pixels
15000
10000
5000
0
0
50
100
150
200
250
Varying Gray Scale



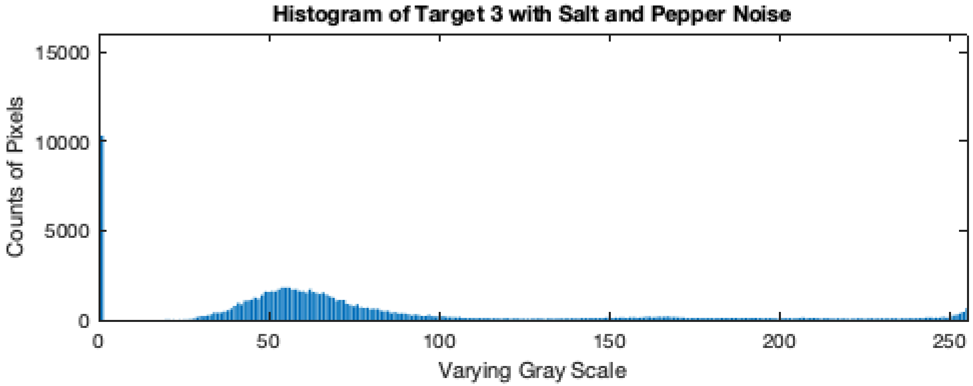
Target 2 with Salt and Pepper Noise



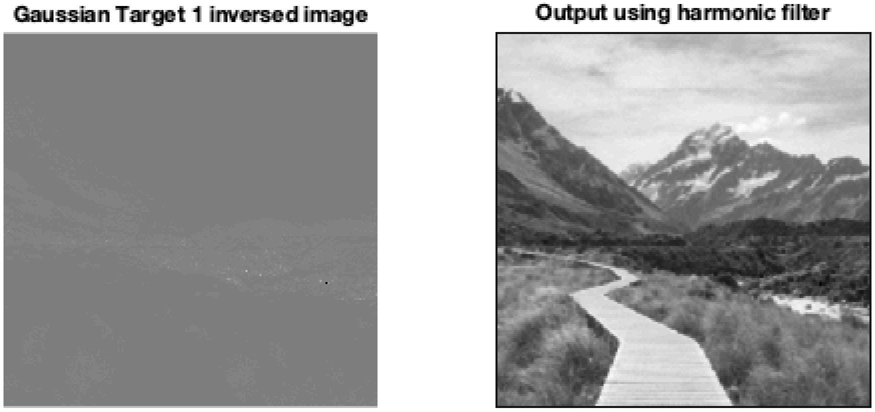
Histogram of Target 2 with Salt and Pepper Noise



Target 3 with Salt and Pepper Noise

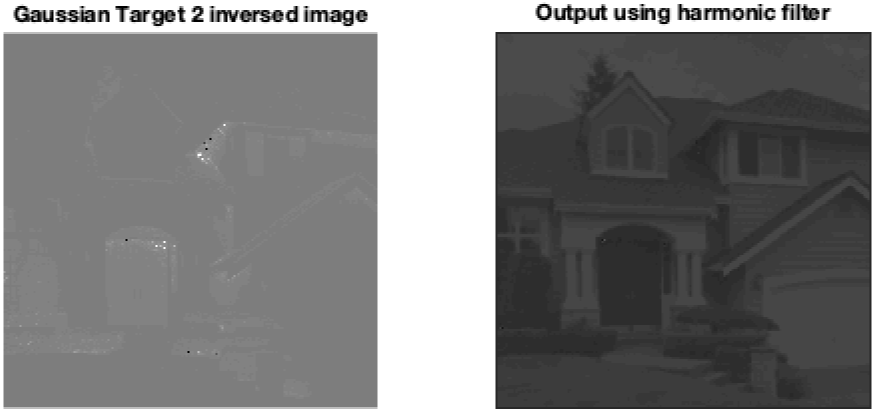


Histogram of Target 3 with Salt and Pepper Noise



Gaussian Target 1 inversed image

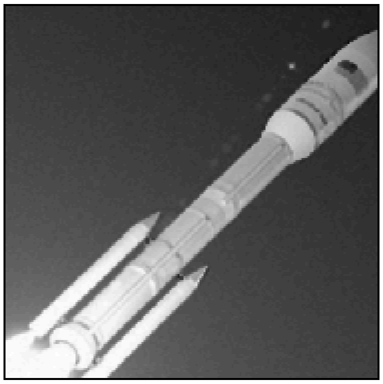
Output using harmonic filter



Gaussian Target 2 inversed image

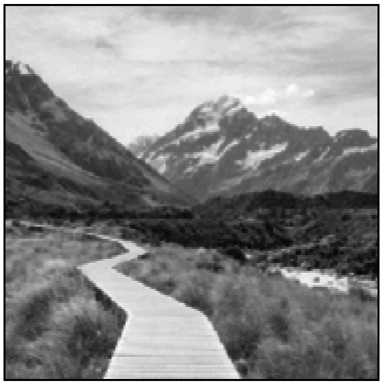
Output using harmonic filter

Gaussian Target 3 inversed image

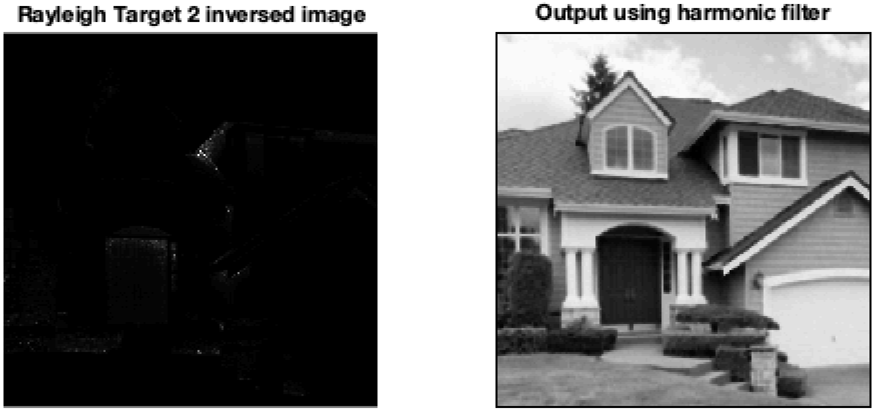


Output using harmonic filter

Rayleigh Target 1 inversed image

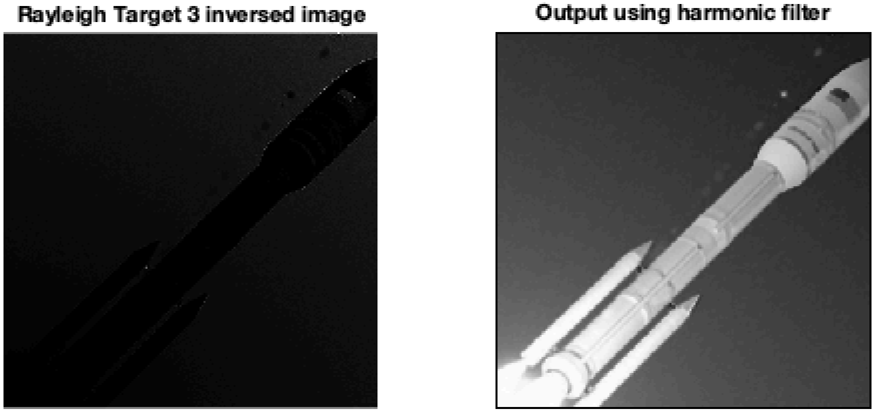


Output using harmonic filter



Rayleigh Target 2 inversed image

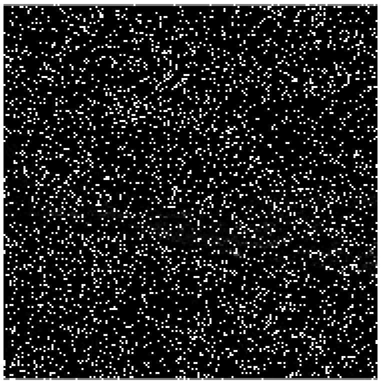
Output using harmonic filter



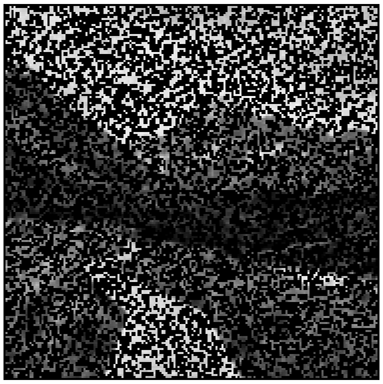
Rayleigh Target 3 inversed image

Output using harmonic filter

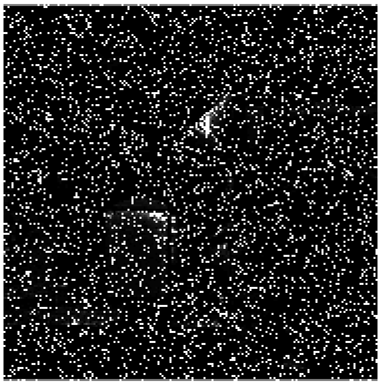
Salt and Pepper Target 1 inversed image



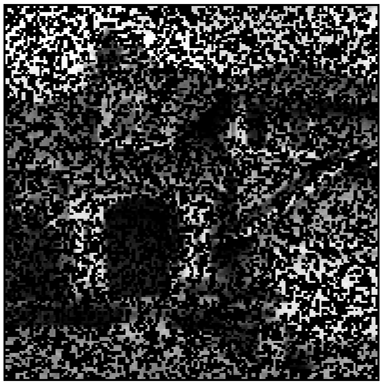
Output using harmonic filter

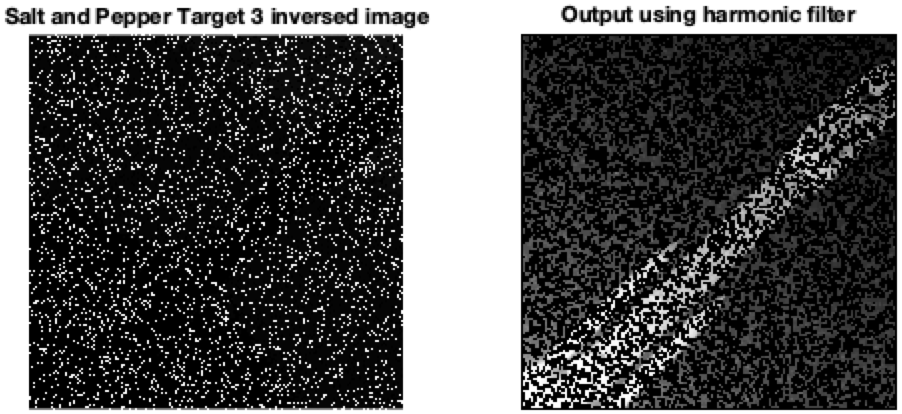


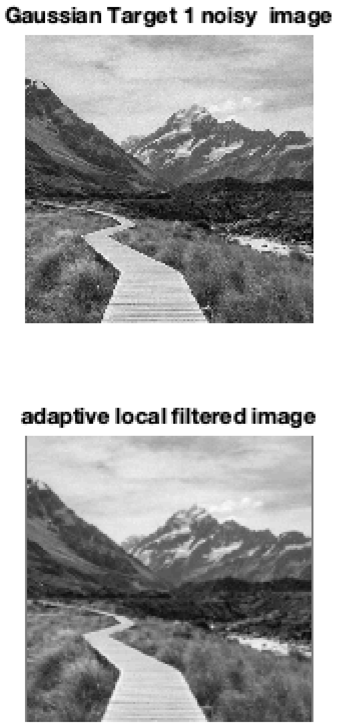
Salt and Pepper Target 2 inversed image



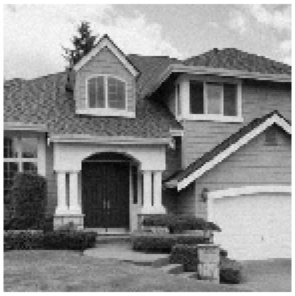
Output using harmonic filter



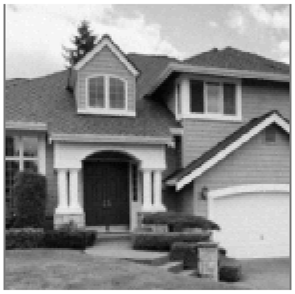

Salt and Pepper Target 3 inversed image
Output using harmonic filter


Gaussian Target 1 noisy image
adaptive local filtered image

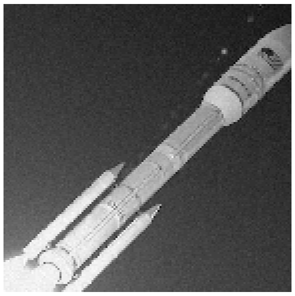
Gaussian Target 2 noisy image



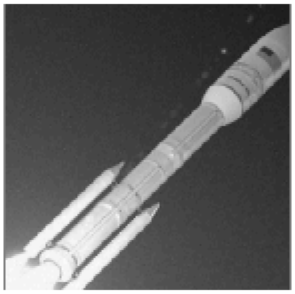
adaptive local filtered image

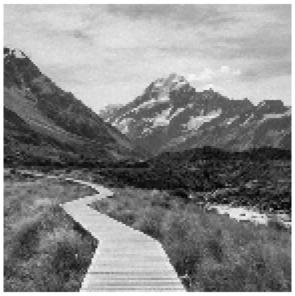


Gaussian Target 3 noisy image

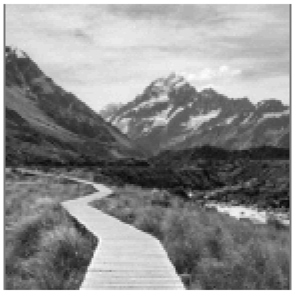


adaptive local filtered image

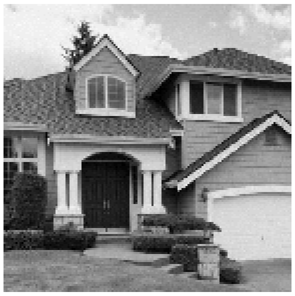




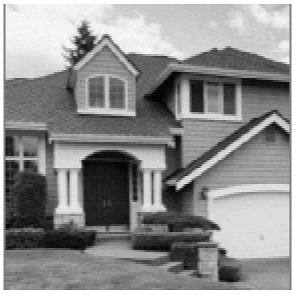
Rayleigh Target 1 noisy image



adaptive local filtered image

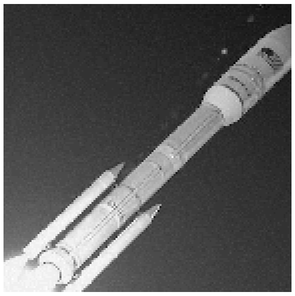


Rayleigh Target 2 noisy image

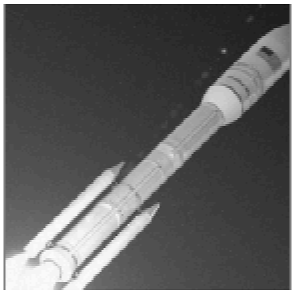


adaptive local filtered image

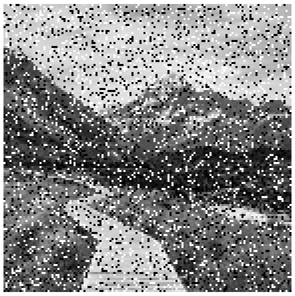
Rayleigh Target 3 noisy image



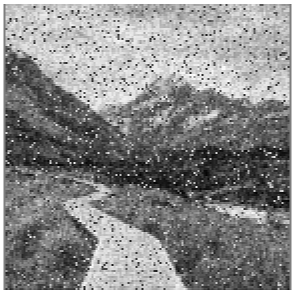
adaptive local filtered image



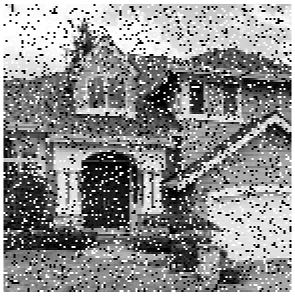
Salt and Pepper Target 1 noisy image



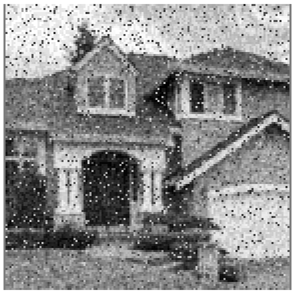
adaptive local filtered image



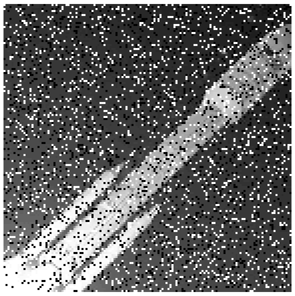
Salt and Pepper Target 2 noisy image



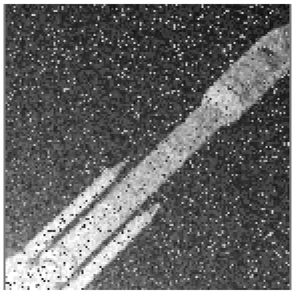
adaptive local filtered image

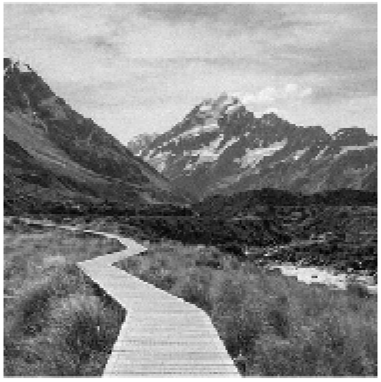


Salt and Pepper Target 3 noisy image

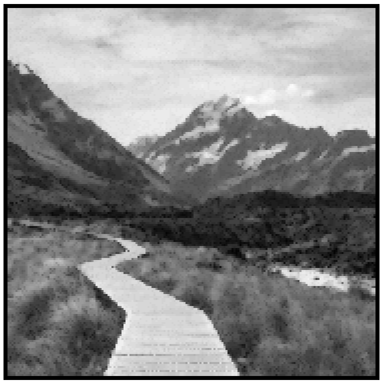


adaptive local filtered image

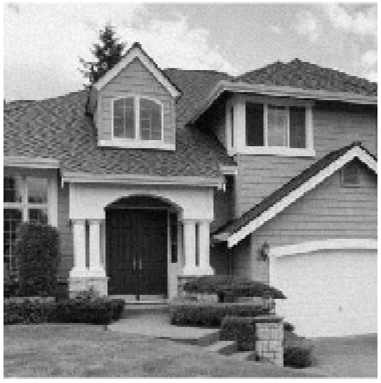




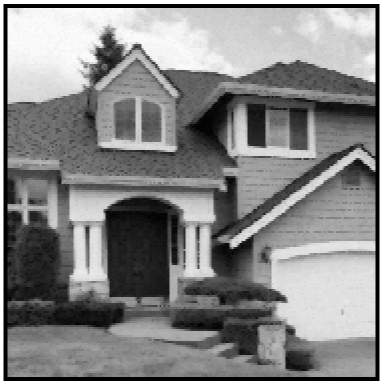
Gaussian Target 1 noisy image



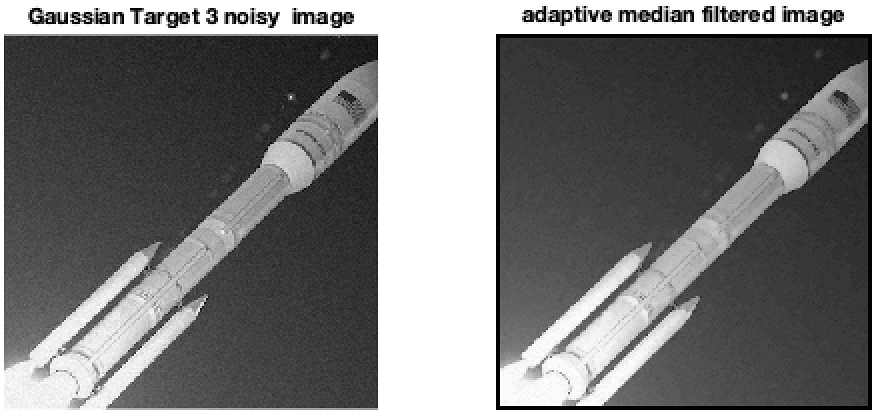
adaptive median filtered image

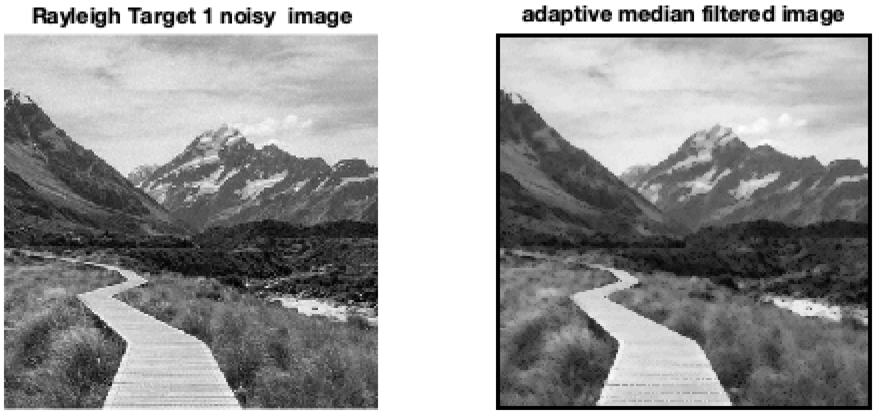


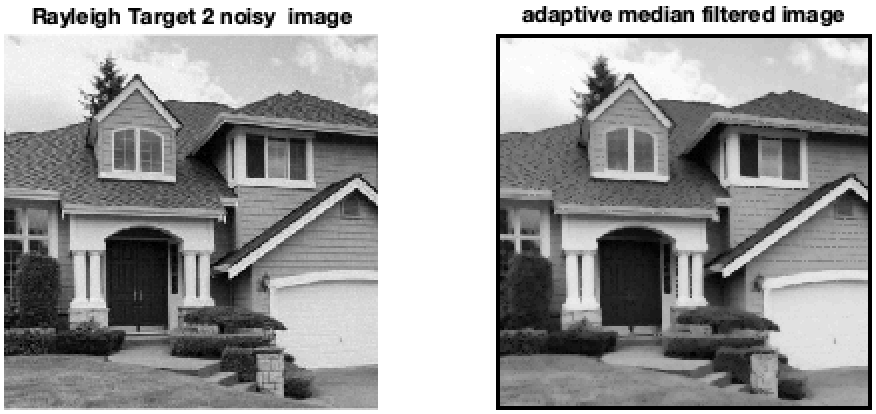
Gaussian Target 2 noisy image



adaptive median filtered image

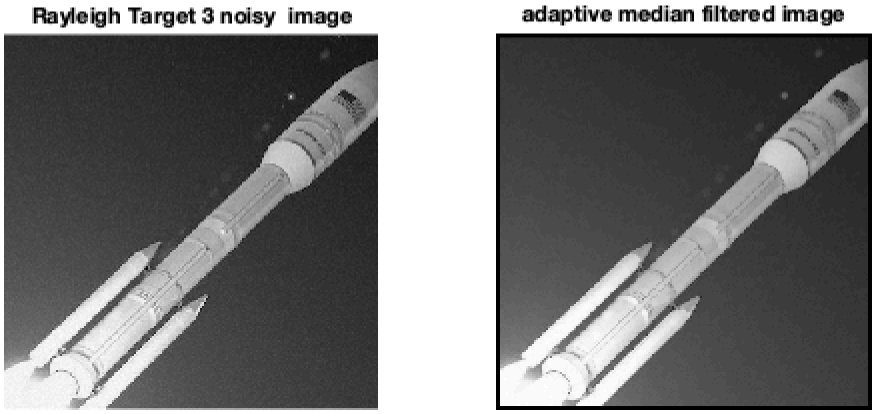

Gaussian Target 3 noisy image
adaptive median filtered image


Rayleigh Target 1 noisy image
adaptive median filtered image



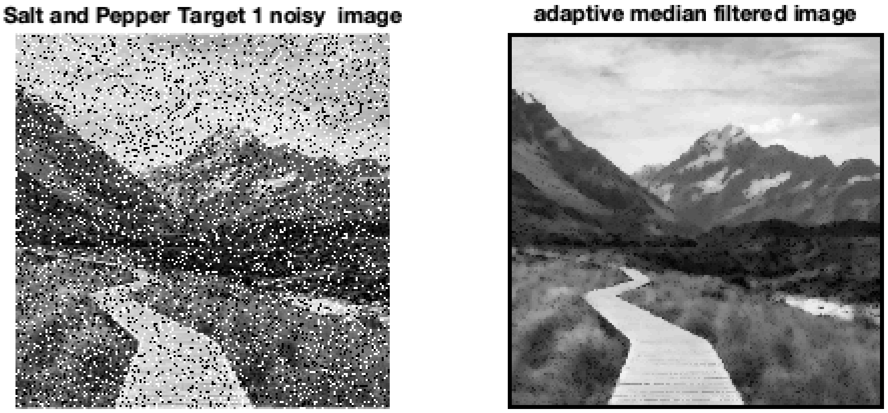
Rayleigh Target 2 noisy image

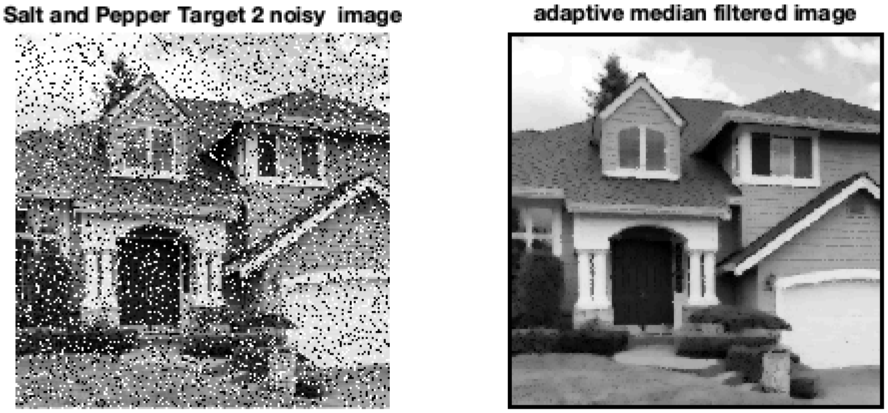
adaptive median filtered image

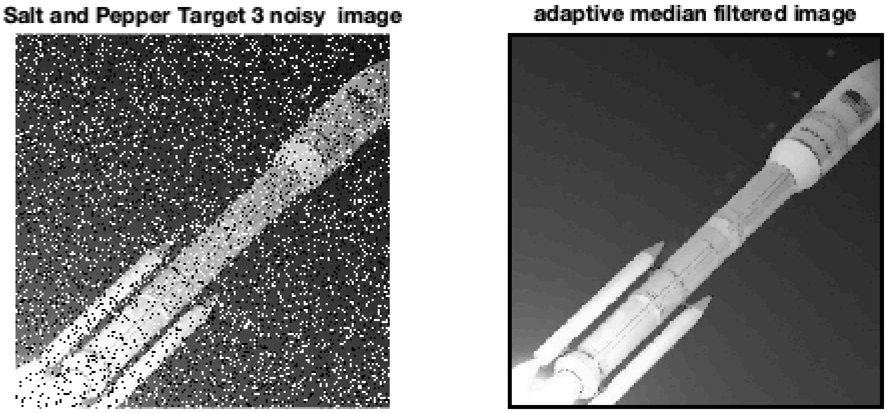


Rayleigh Target 3 noisy image

adaptive median filtered image


Salt and Pepper Target 1 noisy image
adaptive median filtered image


Salt and Pepper Target 2 noisy image
adaptive median filtered image


Salt and Pepper Target 3 noisy image
adaptive median filtered image

Published with MATLAB® R2018b