Project Report

Submitted By: MANISH SONI

Given Objective:

Use your previously developed ability to add noise effects to images to create a test of images similar to those you have done in the past.

Code up and test the following filters on this data:

Harmonic mean filter (eq 5-25)

Adaptive local noise reduction filter (eq 5-32)

The adaptive median filter (algorithm on page 386 and lecture notes)

You must test on at least three different images, and you must use at least three different types of noise on each image, and then test and evaluate the noise reduction in every case.

For this assignment you are to write and submit a report on your approach, methods, and findings. This includes quantitatively assessing the results. Two ways to do this are the sum square errors between the known input image, and the noisy image and the filtered image. Another way to do this is with the "structural similarity index" (SSIM). The original paper on this is here:

wang03-reprint_ssim.pdf

Actions

(It has been cited on the order of 23,000 times, so they've got my best paper beat by maybe a factor of 50 to 100...)

You do not need to code the SSIM yourself however, it is a MATLAB subroutine called "ssim".

You submissions need to be typed, mathematical, well organized, and well thought out.

Your final submission should be a .zip file containing your report in .pdf format, your code, and your test images. The code should be organized so that if i put your code and your test images in a directory it will run for me.

Introduction:

As per the given objective we need to test out the performance of different filters for the different images with noise. We should have three different images and should introduce three kinds of noises for each image.

Once we get the noisy images we need to use different filter to test out the output received out of these filter with input images.

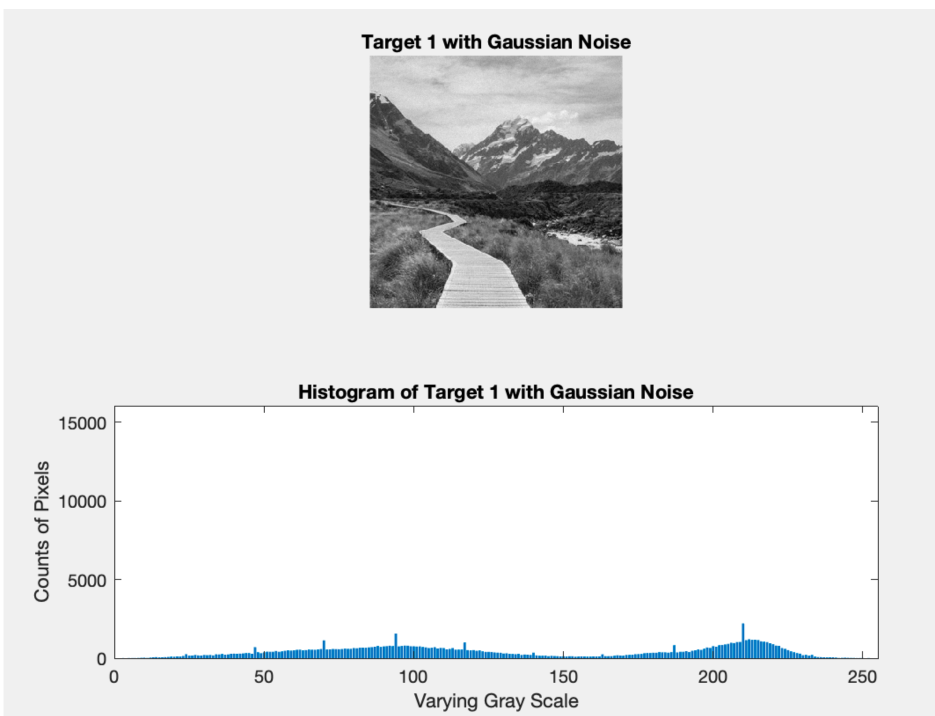
Approach Steps:

As per the given task we can divide all the steps in three main parts.

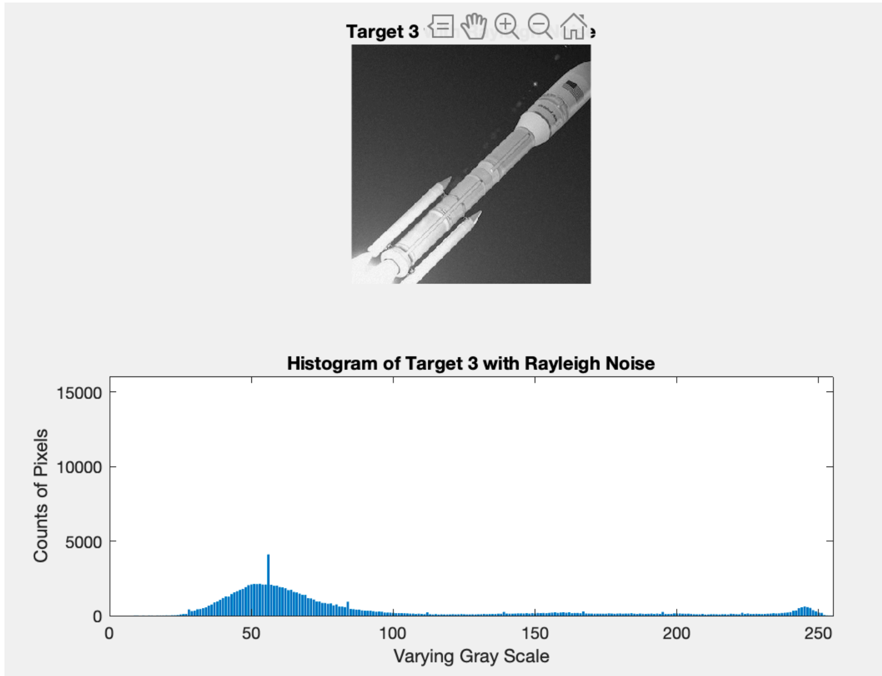
1. Noise introduction in images.
2. Noise reduction using Filter (Harmonic mean/ Adaptive Local Noise reduction/ Adaptive median)
3. Using SSIM evaluation to compare the input Original image and the output received at the end of Step 2.

Noise Types that we are using for three images:

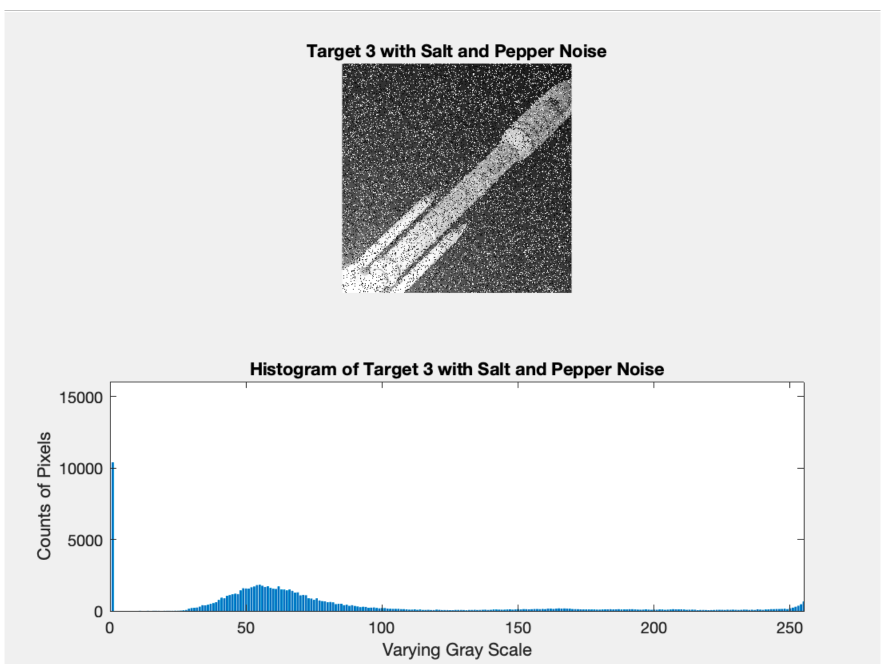
1. Gaussian Noise



2. Rayleigh Noise



3. Salt and pepper Noise



Filter Introduction:

Harmonic Median Filter:

The Harmonic Mean Filter work on the principle of calculating the regional Harmonic mean for the masked filter region and applying that on image with the below formula:

$$\hat{f}(x,y) = \frac{mn}{\sum_{(r,c)\in S_{xy}} \frac{1}{g(r,c)}}$$

We need to keep in mind that this approach filters out salt noise but not the pepper one when used on Salt and pepper noised image.

Adaptive Local Noise Reduction Filter:

Adaptive local noise reduction filter is basically trying to use a local spatial region for calculation the variance and mean for the masked region and then use below formula for (x,y) location.

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{S_{xy}}^2}\left[g(x,y) - \overline{Z_{S_{xy}}}\right]$$

Where $g(x,y)$ is the pixel intensity value at (x,y).
σ_η^2 is Variance of Noise in the image
$\sigma_{S_{xy}}^2$ is the variance of noise in the masked region
$Z_{S_{xy}}$ is the mean value of pixel intensity in the masked region.

Adaptive Median Filter:

Adaptive median filter according to book “Digital Image processing” discuss that adaptive median filtering can handle noise with probabilities larger than basic salt and pepper with less probability. An additional benefit of the adaptive median filter is that it seeks to preserve detail while simultaneously smoothing non-impulse noise, something that the “traditional” median filter does not do. The adaptive median filter also works in a rectangular neighbor- hood, S_{xy} . The adaptive median filter changes (increases) the size of S_{xy} during filtering, depending on certain conditions as per below.

$z_{\min}$ = minimum intensity value in S_{xy}

$z_{\max}$ = maximum intensity value in S_{xy}

z_{med} = median of intensity values in S_{xy}

z_{xy} = intensity at coordinates (x, y)

S_{max} = maximum allowed size of S_{xy}

The adaptive median filter uses below steps for implementing the filter.

Level A:

If$z_{min} < z_{med} < z_{max}$, gotoLevel$B$ Else, increase the size of S_{xy}
If $S_{xy} \leq S_{max}$, repeat level A
Else, output z_{med}.

Level B :

If $z_{min} < z_{xy} < z_{max}$, output z_{xy} Else output z_{med}.

Typical Code Structure:

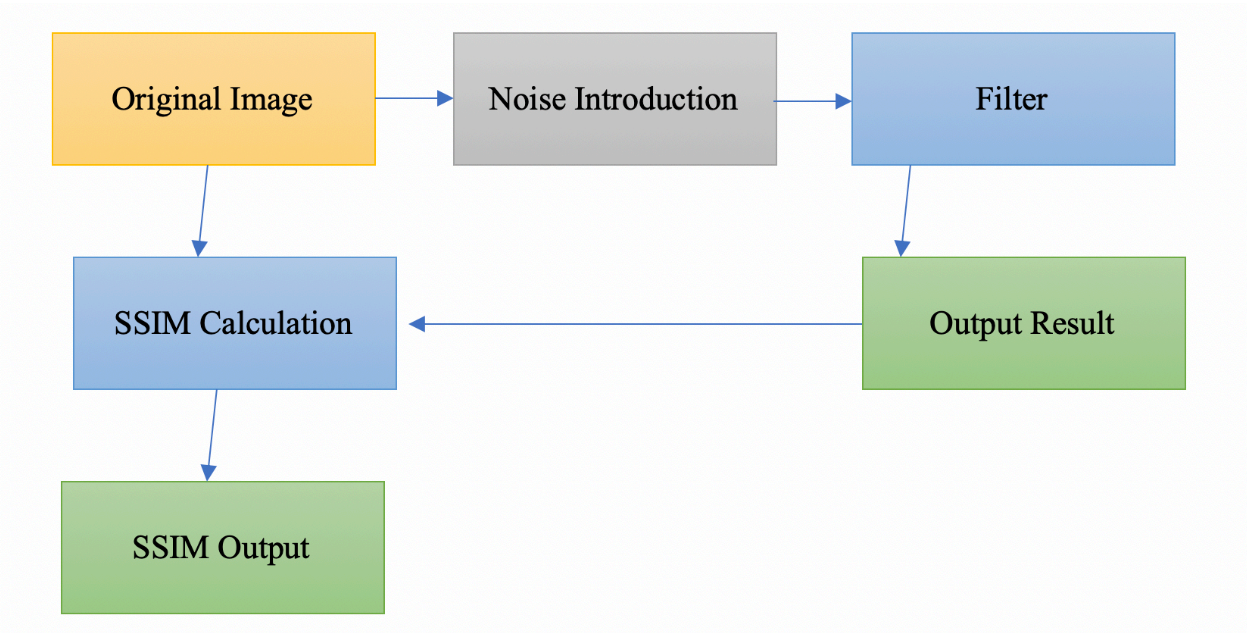
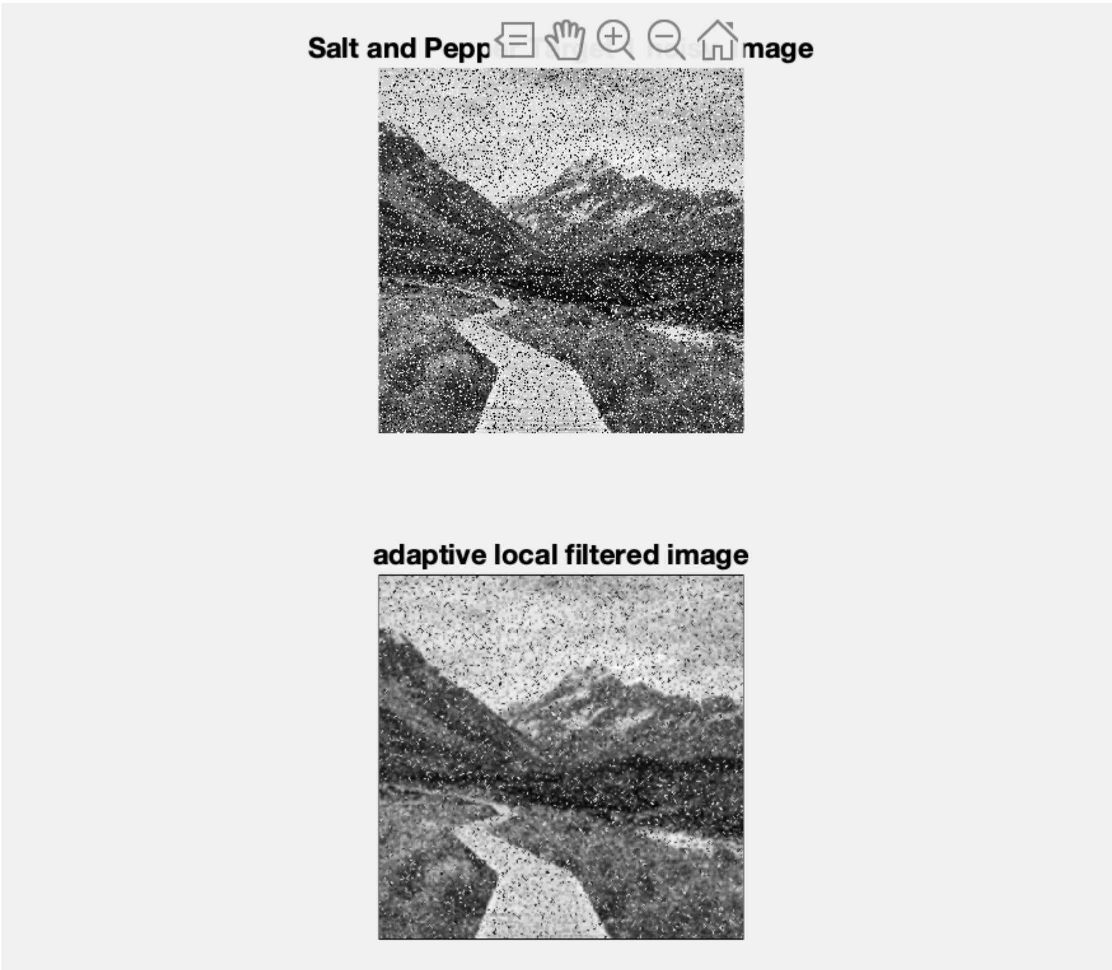


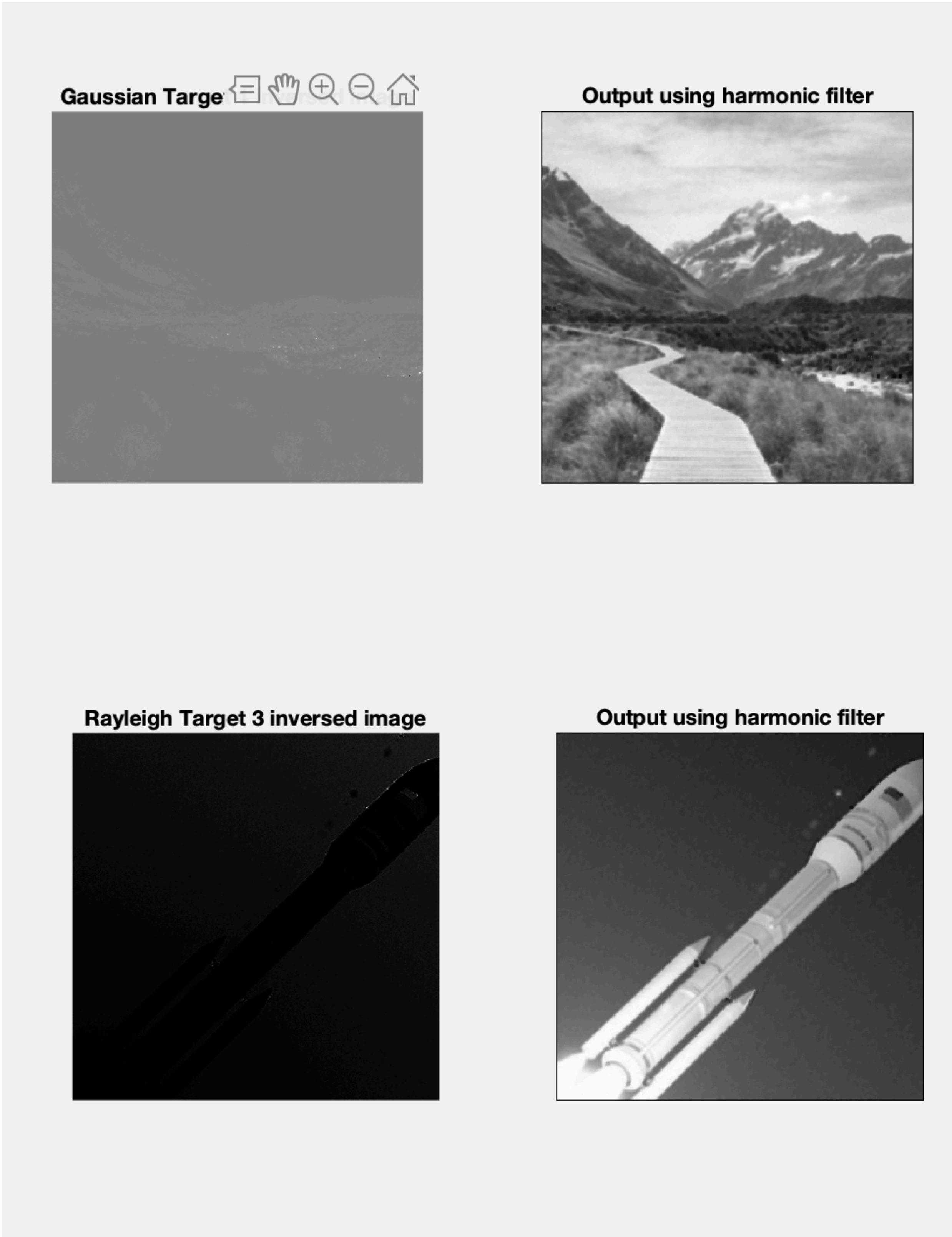
Figure 1: Block diagram representation of code flow for a single image

Function Arrangement:

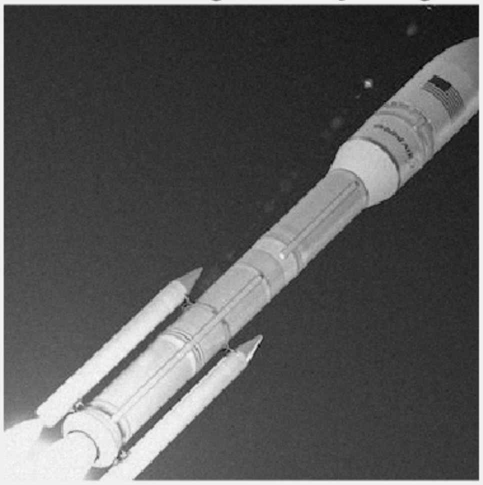
I've divided the complete code structure in 1 main file and 3 supporting functions for performing different filter operations. This make the code easy to reuse and is very structured so to understand it very easily.

Example Output:

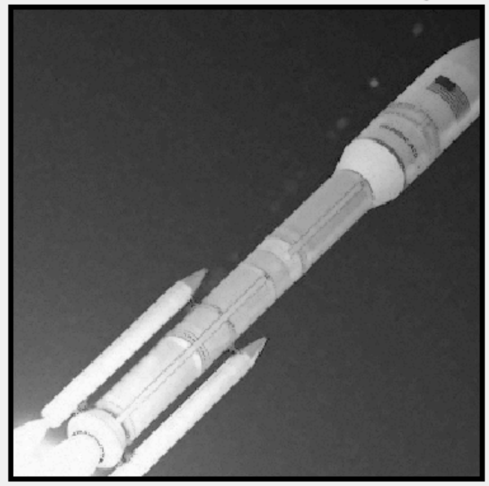



Gaussian Targe
Output using harmonic filter
Rayleigh Target 3 inversed image
Output using harmonic filter

Gaussian Target 3 noisy image



adaptive median filtered image



Salt and Pepper Target 2 noisy image

adaptive median filtered image

SSIM Observations:
For an observation:

Harmonic filter:

Input image with Gaussian noise

Filtered Image	Original image	SSIM Output
Gaussian Target 1	Original Image 1	65.7562%
Gaussian Target 2	Original Image 2	63.3438%
Gaussian Target 3	Original Image 3	38.9315%

Input image with Rayleigh noise

Filtered Image	Original image	SSIM Output
Rayleigh Target 1	Original Image 1	67.7084%
Rayleigh Target 2	Original Image 2	65.2986%
Rayleigh Target 3	Original Image 3	48.4932%

Input image with Salt and Pepper noise

Filtered Image	Original image	SSIM Output
Salt and Pepper Target 1	Original Image 1	3.2621%
Salt and Pepper Target 2	Original Image 2	4.5075%
Salt and Pepper Target 3	Original Image 3	0.9122%

Adaptive Local filter:

Input image with Gaussian noise

Filtered Image	Original image	SSIM Output
Gaussian Target 1	Original Image 1	68.9594%
Gaussian Target 2	Original Image 2	66.8692%
Gaussian Target 3	Original Image 3	38.0522%

Input image with Rayleigh noise

Filtered Image	Original image	SSIM Output
Rayleigh Target 1	Original Image 1	71.1526%
Rayleigh Target 2	Original Image 2	68.9519%
Rayleigh Target 3	Original Image 3	47.3764%

Input image with Salt and Pepper noise

Filtered Image	Original image	SSIM Output
Salt and Pepper Target 1	Original Image 1	20.0995%
Salt and Pepper Target 2	Original Image 2	25.4573%
Salt and Pepper Target 3	Original Image 3	7.8275%

Adaptive Median filter:

Input image with Gaussian noise

Filtered Image	Original image	SSIM Output
Gaussian Target 1	Original Image 1	64.1352%
Gaussian Target 2	Original Image 2	68.4716%
Gaussian Target 3	Original Image 3	31.9148%

Input image with Rayleigh noise

Filtered Image	Original image	SSIM Output
Rayleigh Target 1	Original Image 1	67.7087%
Rayleigh Target 2	Original Image 2	71.598%
Rayleigh Target 3	Original Image 3	40.1922%

Input image with Salt and Pepper noise

Filtered Image	Original image	SSIM Output
Salt and Pepper Target 1	Original Image 1	66.0709%
Salt and Pepper Target 2	Original Image 2	71.46%
Salt and Pepper Target 3	Original Image 3	63.7277%

Conclusion:

I have documented the basics of all the three filters Harmonic mean, Adaptive Local Noise reduction filter and Adaptive Median filter. Through this report, I've documented the output of different Noisy images. This report highlight Mathematical formula behind filters and Quantitatively discusses the output result.