Application of Spatial and Frequency Domain Filters on Noisy Images for Image Enhancement (using Matlab).

ABSTRACT

Noise is unwanted information present in an image. Such unwanted information in an image can be removed with filters. In digital image processing, filters can be applied on an image in two ways, which include spatial and frequency domain. This report mainly deals with the application of predefined and user defined filters in both spatial and frequency domain on noisy images. This is done so to identify the efficiency of the filters in terms of enhancing the quality of the image thereby removing the noise present in it. The purpose of this paper is to compare between mean, Gaussian and median filters in spatial domain and Butterworth low Pass Filter (BLPF) and Ideal low Pass Filter (ILPF) within the frequency domain to enhance noisy image and obtain sharper image. The proposed enhanced models were applied on standard noisy pictures and simulated using MATLAB. Experimental results comparing the old and new filters design showed that BLPF outperformed ILPF. Moreover, the new designed filters outperformed the original ones.

KEYWORDS

Linear filter, Non-Linear Filter, Average, Mean, Median, Gaussian, Mean Square Error (MSE), spatial domain, frequency domain, Low pass filter, High pass filter, Butterworth filter, SNR(Signal to noise ratio),PSNR(Point signal to noise ratio).

INTRODUCTION

Noise present in an image affects the originality of the image and makes the process of interpretation of the image more difficult. Generally, an image gets affected by noise during its acquisition, transmission and storage. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:-

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise.
- Electronic transmission of image data can introduce noise.

Based on its nature, noise can be modelled as either an additive or a multiplicative process.

$$g\ x, = f\ x,y + n(x,y) \qquad\qquad g\ x,y = f\ x,y * n(x,y) \qquad (1)$$

Where f x,y - Original image, n(x,y) - Noise, g x,y -Noisy image.

The above notations (1) represent both the additive and multiplicative noise models. All such types of noise present in an image can be removed with filters.

A filter is a technique with which certain frequency components can be chosen or rejected. The spatial domain filter and frequency domain filters are the most important tools in an image processing. These filters refer to the image plane itself and operate directly on the image pixels and Fourier transformed image respectively. They can be characterized as linear (Average and Gaussian) and non-linear filters (Median) in spatial domain and Low pass and high pass filters in frequency domain.This study mainly deals with the application of filters which include Mean, Median, Gaussian, Low pass and high pass filters on noisy images, and the calculation of MSE (Mean square error) and PSNR(Point to signal Ratio) on the filtered images with a view to identifying the efficiency of these filters in terms of minimizing noises present in the image.

Noise Image

Image noise represents unwanted or undesired information that can occur during the image capture, transmission, processing or acquisition, and may be dependent or independent of the image content. In typical images, the noise can be modelled with either a Gaussian, uniform or salt-and-pepper distribution. If we had to introduce noise in the image simply add noise to the original image. The noise can be of various types like Salt & Pepper, Gaussian Noise, and Speckle Noise etc.

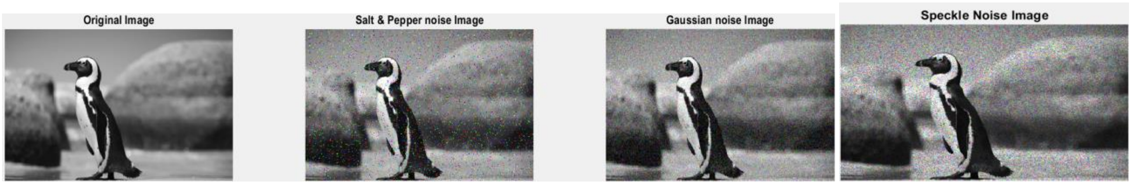


Fig 1: Original image, Salt and pepper noisy image, Gaussian noisy image, Speckle noisy image

Since we already have a noisy image “PenguinNoise.bmp” we would be using the same in the study.

SPATIAL DOMAIN

Application of Filters in Spatial domain

A filter is something that attenuates or enhances particular frequencies easiest to visualize in the domain. Image filters have a wide variety of uses such as noise removal and edge enhancement; It creates an embossed appearance making the image appear crisper and sharper.

The term **Filter** in “Digital image processing” is referred to the sub image. There are others term to call sub image such as mask, kernel, template or window .The value in a filter sub image is referred as coefficients rather than pixels.

Mean Filter

Spatial domain represents an important enhancement technique that can effectively be used to remove various types of noise in digital images. These spatial filters typically operate on small neighbourhood 3 x 3 to 11 x 11 . Mean filters are the most common spatial filters used as a simple method for reducing noise in an image, particularly Gaussian noise. The idea of mean filtering is simply to replace each pixel value in an image with the mean ‘average’ value of its neighbours, including itself. The extracted average values are the result of the convolution process, which is commonly based on specified fixed convolution mask (kernel). Differently sized kernels containing different patterns of number achieve different results under convolution. By increasing the size of the mean filter to 5 x 5, the obtained image will be characterized with less noise.

In general, linear filtering of an image f of size M x N with a filter mask of size m x n is given by the expression

$$g(x,y)= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x+s,y+t) \qquad (2)$$

where a=(m-1)/2 and b=(n-1)/2; m x n (odd numbers) for x=0,1,…,M-1 and y=0,1,…,N-1 . The process is also called convolution (used primarily in the frequency domain)

Figure (1) depicts the Noisy image and output image after applying Mean filter

Gaussian Filter

Gaussian filtering is used to blur images and remove noise and detail. The Gaussian function is used in numerous research areas: – It defines a probability distribution for noise or data.

- It is a smoothing operator.
- It is used in mathematics.

Gaussian filtering is more effective at smoothing images. It has its basis in the human visual perception system. It has been found that neurons create a similar filter when processing visual images.

In one dimension, the Gaussian function is:

$$g(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

In two dimensions, it is the product of two such Gaussians, one per direction:

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

Figure (1)depicts the Noisy image and output image after applying Gaussian filter

Median Filter

In average filters, according to a defined average criterion, the average value of the neighbouring pixels is calculated and this value is put to the center pixel location. Ordered filters are usually used to filter salt and pepper noises, negative exponential noises and Raleigh distributed noises. Average filters are used to remove Gauss distributed noises or uniform distributed noises, the most effective ordered filter is median filter. With this filter, gray-level values in a chosen neighbourhood are sorted in the ascending manner. Center pixel of the mask is replaced with the median value of the ordered gray-levels. With sorting operation, noise valued pixels will be either on the head of the array or in the end of the array with a greater probability and the median value will not be a noise point with high probability. If there is a noise point in the center of the mask and if it’s replaced with median value, then the noise could be filtered out.

Unlike the mean filter, the median filter is non-linear. This means that for two images A(x) and B(x):

$$median[A(x) + B(x)] \neq median[A(x)] + median[B(x)] \tag{5}$$

Figure (1) depicts the Noisy image and output image after applying Median filter

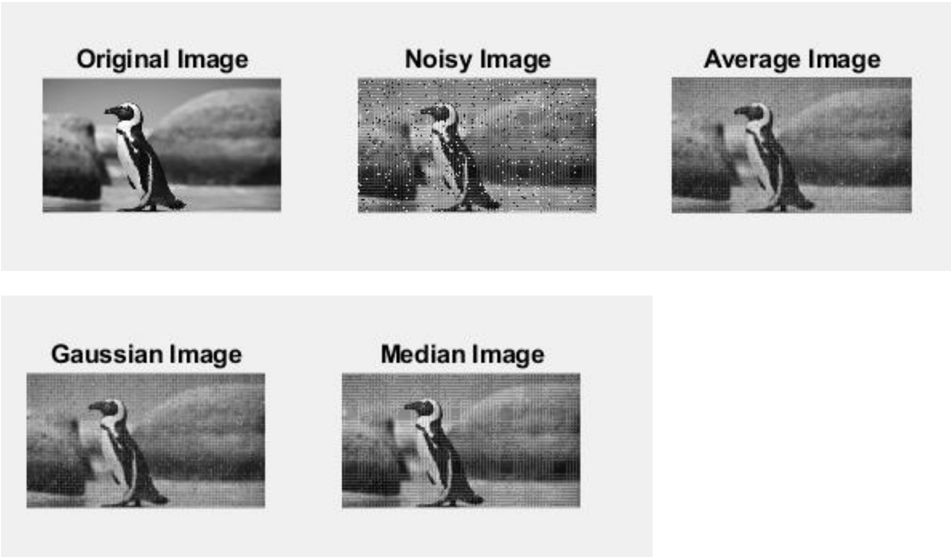


Figure (2) :- Original image ,Noisy image and Filtered image with Average/Mean, Gaussian and Median filter

Methodology

Neighbourhood averaging or Gaussian smoothing will tend to blur edges because the high frequencies in the image are attenuated. An alternative approach is to use *median filtering*. Here we set the grey level to be the median of the pixel values in the neighbourhood of that pixel. The median m of a set of values is such that half the values in the set are less than m and half are greater. For example, suppose the pixel values in a 3X3 neighbourhood are (10, 20, 20, 15, 20, 20, 20, 25, 100). If we sort the values we get (10, 15, 20, 20, |20|, 20, 20, 25, 100) and the median here is 20. The outcome of median filtering is that pixels with outlying values are forced to become more like their neighbours, but at the same time edges are preserved. Of course, median filters are non-linear.

Median filtering is in fact a morphological operation. When we erode an image, pixel values are replaced with the smallest value in the neighbourhood. Dilating an image corresponds to replacing pixel values with the largest value in the neighbourhood. Median filtering replaces pixels with the median value in the neighbourhood. It is the rank of the value of the pixel used in the neighbourhood that determines the type of morphological operation.

User defined Filter in spatial domain using median filter

Algorithm:

1. Consider any noisy image
2. Take the size of the image which is a matrix and put zeros
3. Form the neighbourhood of 3X3 matrix setting the boundaries across a pixel.
4. Apply filter on this sub image.
5. Iterate steps 3-5 till we cover the size of the image.
6. Output the original picture and the output picture to see the difference.

Figure (3) Original image, Noisy image and output image after applying User defined Median filter.

From fig (3) it is clear that there has been noise removal from the input image thereby enhancing the output.

Error Measures

In the development of image enhancement, restoration and coding techniques, it is useful to have some measure of the difference between a pair of similar images. The most common difference measure is the mean-square error.

Mean Square Error (MSE)

The mean-square error measure is popular because it correlates reasonably with subjective visual quality tests and it is mathematically tractable. The error metrics is used to compare the various image compression techniques .The MSE is the cumulative squared error between the compressed and the original image .

Consider a discrete I (x, y) for x = 1, 2,..., M and y = 1, 2,..., N, which is regarded as a reference image, and consider a second image of the same spatial dimensions as I'(x,y) that is to be compared to the reference image. Under the assumption that I (x,y) and I'(x,y) represent samples of a stochastic process, the mean-square error between the image pair is defined as

$$\text{MSE}= \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}\left[I(x,y) - I'(x,y)\right]^2 \qquad (6)$$

Using Matlab predefined function **immse** on each of the filtered image shown in fig(1)

The calculated mean-square error with Average filter is 420.8011
The calculated mean-square error with Gaussian filter is 692.7049
The calculated mean-square error with Median filter is 510.5344

Code in Matlab herewith attached.

Spatial Domain	**Code**
Built-in Filter	
User Defined - Median	

FREQUENCY DOMAIN

Application of Filters in Frequency Domain

Frequency domain is a term used to describe the analysis of mathematical functions or signals with respect to frequency, rather than time**.** Frequency domain methods are based on modifying the Fourier transformation of an image and the filters operate on the Fourier transform of an image.

• The Edges and sharp transitions (e.g. noise) in an image contribute significantly to high frequency content of Fourier transform.
• Low frequency contents in the Fourier transform are responsible to the general appearance of the image over smooth areas. By suppressing the noise, gradual changes can be seen that were invisible before. Therefore a low-pass filter can sometimes be used to bring out faint details that were smothered by noise.

$$G(x, y) = h(x, y)*f(x, y) \quad (7)$$

Where g(x, y) is enhanced image.

There are three basic steps to frequency domain filtering.

1. The image must be transformed from the spatial domain into the frequency domain using the Fast Fourier transform. In General, FT is utilized to represent the image in frequency domain, where it is easy to implement and has many advantages with regard to the image processing. The Fourier Transform is defined as:

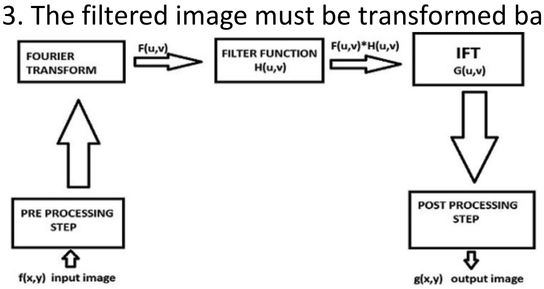
$$F(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(x\frac{m}{M}+y\frac{n}{N})} \quad (8)$$

While the inverse is defined as :

$$f(m,n) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(x,y) e^{j2\pi(x\frac{m}{M}+y\frac{n}{N})} \quad (9)$$

In a 2D image, the first dimension transform is performed for the first line of image F (u, y) and then it is kept in the memory. After that, the same calculation is performed column by column on the computed image. On the other hand, Fast Fourier Transform (FFT) is utilized instead of the Discrete Fourier Transform (DFT) because it is faster. However, FFT can only be used on images with second order width and height

2. The resulting complex image must be multiplied by a filter (that usually has only real values).
3. The filtered image must be transformed back to the spatial domain.



Types of frequency domain filters

1)Low pass filter

The most basic of filtering operations is called "low-pass". A low-pass filter, also called a "blurring" or "smoothing" filter, averages out rapid changes in intensity. The simplest low-pass filter just calculates the average of a pixel and all of its eight immediate neighbours. The result replaces the original value of the pixel. The process is repeated for every pixel in the image.

Filtering can be visualized by drawing a “convolution kernel”. A kernel is a small grid showing how a pixel's filtered value depends on its neighbours. To perform a low-pass filter by simply averaging adjacent pixels, the following kernel is used:

+1/9	+1/9	+1/9
+1/9	+1/9	+1/9
+1/9	+1/9	+1/9

When this kernel is applied, each pixel and its eight neighbours are multiplied by 1/9 and added together. The pixel in the middle is replaced by the sum. This is repeated for each pixel in the image. If we didn't want to filter so harshly, we could change the kernel to reduce the averaging, for example:

0	+1/8	0
+1/8	+1/2	+1/8
0	+1/8	0

The center pixel contributes half of its value to the result, and each of the four pixels above, below, left, and right of the center contribute 1/8 each. This will have a more subtle effect. By choosing different low-pass filters, we can pick the one that has enough noise smoothing, without blurring the image too much.We could also make the kernel larger. The examples above were 3x3 pixels for a total of nine. We could use 5x5 just as easily, or even more. The only problem with using larger kernels is the number of calculations required becomes very large.

Common low pass filters are :-Ideal Low pass filter ,Butterworth low pass filter, Gaussian low pass filter

2) High pass filter

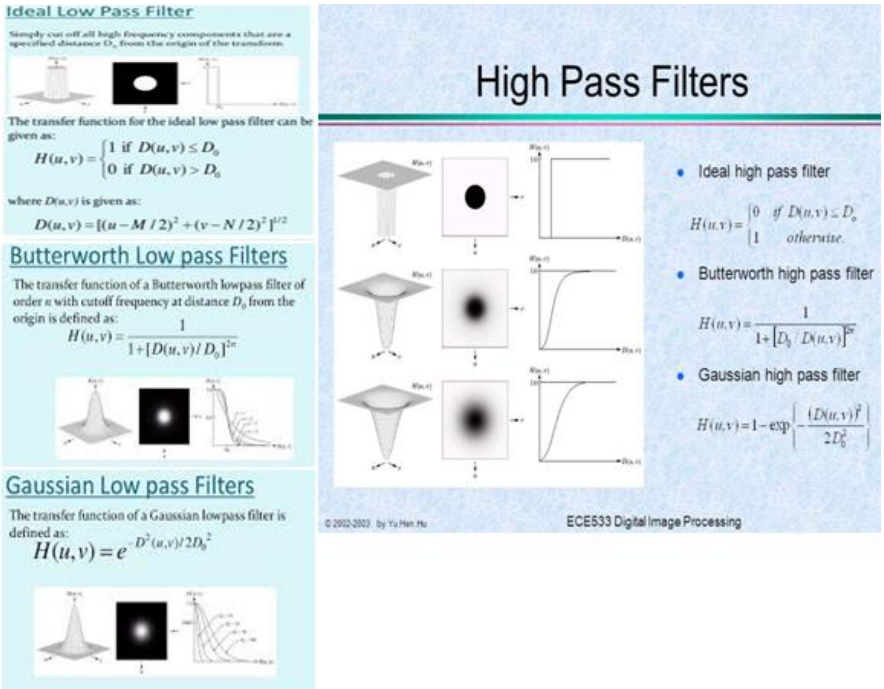
A high-pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image – exactly the opposite of the low-pass filter. High-pass filtering works in exactly the same way as low-pass filtering; it just uses a different convolution kernel. In the example below, notice the minus signs for the adjacent pixels. If there is no change in intensity, nothing happens. But if one pixel is brighter than its immediate neighbours, it gets boosted.

0	-1/4	0
-1/4	+2	-1/4
0	-1/4	0

Unfortunately, while low-pass filtering smoothes out noise, high-pass filtering does just the opposite: it amplifies noise. You can get away with this if the original image is not too noisy; otherwise the noise will overwhelm the image. High-pass filtering can also cause small, faint details to be greatly exaggerated. An over-processed image will look grainy and unnatural, and point sources will have dark donuts around them. So while high-pass filtering can often improve an image by sharpening detail, overdoing it can actually degrade the image quality significantly.

Common high pass filters are:-Ideal high pass filter, Butterworth high pass filter, Gaussian high pass filter

Comparison between the Low pass and High Pass filters



(9)

In all the cases the high pass filter would be **1-H$_L$** (Low pass filter function) and is exactly opposite of the low pass filter.

3) Band pass filter

A band pass attenuates very low and very high frequencies, but retains a middle range band of frequencies. Band pass filtering can be used to enhance edges (suppressing low frequencies) while reducing the noise at the same time (attenuating high frequencies).

Methodology

Noisy image have been processed to attain the objective of this paper. Where, MATLAB Software has been used to process the image. Taking Fourier transform of the above image, we get a fourier transformed image. Based on the cut-off freq (P) we design the filter function H , Here the cut-off frequency is nothing but radius of the white circle in the below image(fig4). The below image is usually referred as filter mask. Perform filtering by using G=H.*F i.e. Multiplying the Fourier transformed image with the filter mask H. Please note the convolution in time domain is equal to multiplication in frequency domain. To the filtered output, we take inverse Fourier transform for the above image and we get the enhanced image. This is the Ideal Low pass filtered image.

Algorithm for ideal low pass filter

Steps:

1. Read the noisy image.
2. Fourier transform the image by using predefined function fft2 (2D image).
3. Then generate a Circle filter.
4. Calculate the size of the image which is a matrix of m*n (height and width)
5. Form a sub image using the mesh grid function and consider the values from negative to positive.
6. Apply Euclidean's formula on the elements.
7. Consider a constant value for defining the cut-off.
8. If values are less than constant then it is 1 else 0 (1- white,0-black).
9. Thus forming the filter. Apply filter to the Fourier transformed image by multiplying both the filter and the Fourier image.
10. Post transformations display the changed image.

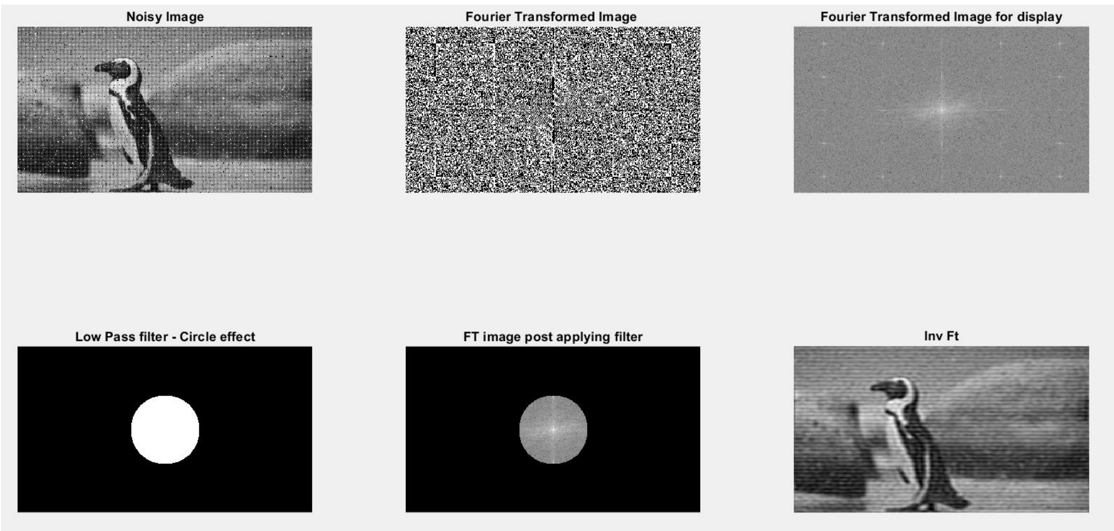


Fig 4: Noisy image, Fourier transformed image, Fourier transformed image for display, Low pass filter with circle effect, Fourier transformed image after applying filter, Inverse image.

Methodology

In frequency domain you can take 2D FFT of the image and then mask part of it based on low-pass or high-pass filter you desire, to get the 2D FFT of the image use fft2, then mask central parts for low pass filtering or corners for high-passing filtering.

A Butterworth high pass filter keeps frequencies outside radius D0 and discards values inside. It has BHPF) of order n and cut-off frequency D0 is defined as (9)

A Butterworth low pass filter keeps frequencies inside radius D0 and discard value outside it introduces a gradual transition from 1 to 0 to reduce ringing artifacts. The transfer function of a Butterworth low pass filter (BLPF) of order n, and with cut-off frequency at distance D0 from the origin, is defined as [2, 7]

Algorithm for Butterworth Low and High pass filter

Steps :

1. Read the noisy image.
2. Fourier transform the image by using predefined function fft2 (2D image).
3. Calculate the size of the image which is a matrix of m*n (height and width)
4. Form a sub image using the mesh grid function and consider the values from negative to positive.
5. Apply Butter worth low pass filter formula (9) .
6. Thus forming the filter. Apply filter to the Fourier transformed image by multiplying both the filter and the Fourier image.
7. Post transformations display the changed image i.e. inverse.
8. For High pass filter Repeat the same steps 1::5 and apply Butter worth high pass filter formula (9) and then repeat steps 6::7.

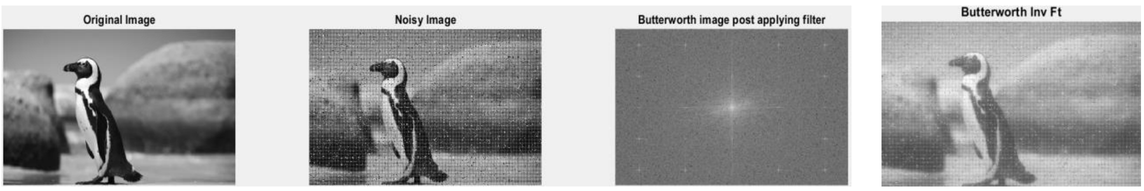


Fig 5: Original image, Noisy image,BW image after applying filter,BW inverse image.

Code in Matlab herewith attached

Frequency Domain	Code
Low Pass Filter	fftshow.m ifftshow.m FD_fft_circle_effect.m
Butterworth Low Pass	butterlp.m FD_Butterworth.m

Discussion and Results

The values of MSN ,PSNR,SNR have been used to perform a comparison between the filters. The computed values of MSE and PSNR for filters are shown in below table.

Images processed by Median filter are more visible and sharper than images processed by Mean and Gaussian filter in spatial domain, and Ideal Low pass filter processed image are less visible then image processed by Butterworth in frequency domain. There is no ringing effect in Butterworth.

The visibility of image increased with Lower value of PSNR and lower value of MSE.

	Spatial Domain				Frequency Domain	
Parameters	Mean	Gaussian	Median	User Defined - Median	Low Pass Filter	ButterWorth Low Pass
MSE	779.4502	148.2193	642.7511	643.6067	880.8223	2604.9373
PSNR	19.2129	26.4218	20.0504	20.0446	18.6819	13.9728
SNR	13.0346	20.2434	13.8721	13.8663	12.5036	7.7945

Tab:-1 Table shows the comparative study of the spatial and frequency domain filters.

Conclusion

This study has dealt with the application of spatial domain filters on 24-bit images and identified the following facts

- The fact that all filters behave differently with different type of noise, Median filter performs well in removing noise present in the image than Mean, Gaussian. This is evident from the calculated mean square error, psnr on all the filters applied in spatial domain and the figure(2). Neighbourhood averaging or Gaussian smoothing will tend to blur edges because the high frequencies in the image are attenuated. An alternative approach is to use median filtering
- The enhancement (Sharp edges) in frequency domain is much achieved by the Butterworth filter than the low pass filter as shown in the table and figure (5).The image blurriness that appears in the ideal filter is overcome by the Butterworth filter. An advantage with the Butterworth filter is that we can control the sharpness of the filter with the order.
- The ideal low pass filter is radically symmetric about the origin, which means that the filter is completely defined by radial cross section as shown in figure (4). Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that gives a clear cut-off between passed and filtered.

References

[1] Kavita, Saroha, R. Bala, R. and Siwach, S. Review Paper on Overview of Image Processing and Image Segmentation, Iinternational Journal of Research in Computer Applications and Robotics, Vol.1 Issue.7, P.p: 1-13, 2013.

[2] Joshi, M. Image Processing in Multimedia Applications, Journal of Information and Operations Management, Volume 3, Issue 1, pp-188-190, 2012. Available online at: http://www.bioinfo.in/contents.php?id=55 .

[3] Zhu, X. and Milanfar, P. Automatic Parameter Selection for Denoising Algorithms Using a No-Reference Measure of Image Content, IEEE Transactions on Image Processing, Vol. 19, No. 12, 2010.

[4] Dogra, A. and Patterh, M. S. Performance Comparison of Gaussian and Butterworth High Pass Filters, International Journal of Innovations in Engineering and Management, Vol. 2; No. 2, 2013.

[5] Prasad, S and Ganesan, N. An Image Sharpening Method by Suppressing the Noise, International Journal of Computer Applications, Volume 51– No.16, 2012.

[6] Makandar, A. and Halalli, Bh. Image Enhancement Techniques using High-pass and Low-pass Filters, International Journal of Computer Applications, Vol. 109 – No. 14, 2015.

[7] Naik, A., Barot, N., Brahmbhatt, R and Dahiya, V. Comparison between Spatial and Frequency Domain Methods, International Journal of Emerging Research in Management &Technology, Vol. 4, Issue. 12, 2015.

[8]Google.com

[9]You Tube

[10]Matlab-Mathwork.