BIRLA INSTITUTE OF TECHNOLOGY

MESRA, RANCHI-835215 (INDIA)

Project Report

EE305- Digital Signal Processing

IMAGE PROCESSING

By:

ANKIT RAJ (BTECH/10228/21)

ASHISH BHARADWAJ (BTECH/10330/21)

MAYANK PRAKASH LOHANI (BTECH/10413/21)

PRAVEEN RAWAT(BTECH/10636/21)

CONTENT

PARAMETERS USED

Filters	Parameters
Adaptive Median Filter	$A_{ij}^{k} = k -$ Symmetric pad matrix t = max window size(9)
Adaptive Riesz Mean Filter	$[b_{ij}]_{m\times n}$ = Binary Matrix $A_{ij}^{k} = k -$ Symmetric pad matrix $ps(a_{ij}, a_{st})$=Pixel Singularity
Different Adaptive Modified Riesz Mean Filter	$[b_{ij}]_{m\times n}$ = Binary Matrix $A_{ij}^{k} = k -$ Symmetric pad matrix $pw(a_{ij}, a_{st})$=Pixel Weight
Alpha Trimmed Mean Filter	(d, α)=Item trimming parameters (α=d/2) d=(1-mn) (m,n)=Window dimensions
Geometric Mean Filter for Image Denoising	(m,n)=Window Dimensions
Sector Rotation Filter	m= Square window Size
Modified Median Filter	$[dij]_{mxn=}$Binary Matrix of noisy image
Recursive Spline Interpolation Filter	m=Square window size
Basic Mean Filter	No Such Parameters
Min/Max Filters	No Such Parameters
Basic Median Filter	No Such Parameters

Adaptive Median Filter

Non-linear spatial filter

INTRODUCTION

It is uses median filtering to remove impulsive noise. The window used for filtering here is adaptive i.e it starts from a window size of 3×3 (for k=1) and increases the window size if the adaptivity conditions are not met.

A window matrix is defined for filtering as A_{ij}^{k} of size (2k+1) × (2k+1), where i and j are the indices of the pixel a_{ij} being processed and k is the number of pixels taken around it.

$$A_{ij}^{k} = \begin{bmatrix} a_{(i-k)(j-k)} & \cdots & a_{(i-k)(j+k)} \\ \vdots & a_{ij} & \vdots \\ a_{(i+k)(j-k)} & \cdots & a_{(i+k)(j+k)} \end{bmatrix}_{(2k+1)\,\times\,(2k+1)}$$

The adaptivity condition used here checks if:

a) $a_{ij} = max(A_{ij}^{k})\ \boldsymbol{OR}\ a_{ij} = max(A_{ij}^{k})$
This checks if the center pixel is noisy and needs processing by comparing it to maximum and minimum of the window.
If it is impulsive the algorithm moves on to the next check.
If it is not impulsive then pixel does not need filtering and the moves on to the next pixel.

b) $min(A_{ij}^{k}) < median(A_{ij}^{k}) < max(A_{ij}^{k})$
This checks if the median value is impulsive or not.
If it is impulsive, then the filter increases the window size and checks again.
If it is not impulsive then the filter replaces the center pixel with the median.

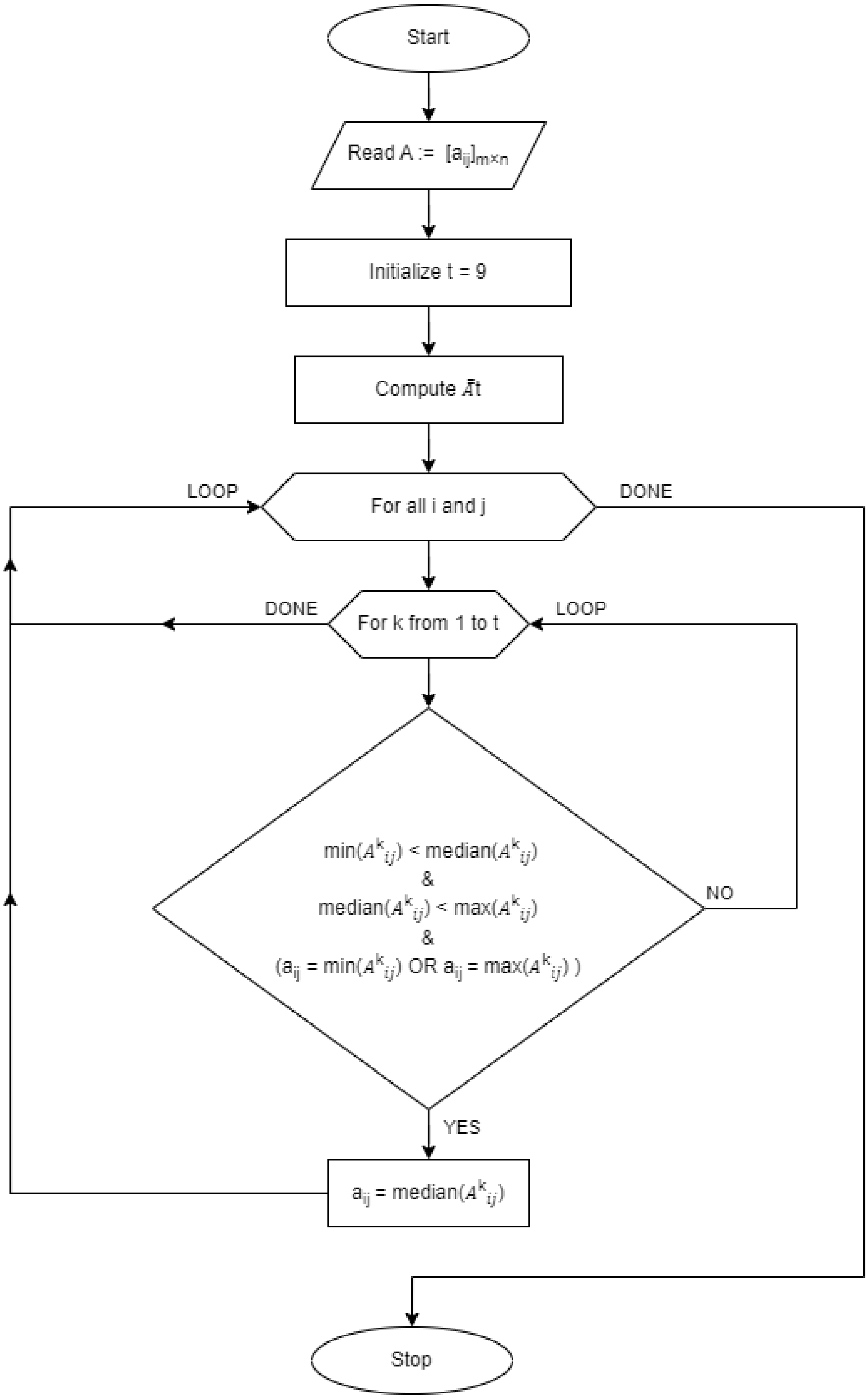
ALGORITHM: -

Input: noisy matrix, $A := [a_{ij}]_{m\times n}$ **Output:** Denoised matrix, $A := [a_{ij}]_{m\times n}$

Algorithm :-

```
Initialize t = 9
Compute $\bar{\bar{A}}_t$
For $all\ i\ and\ j$
    If $a_{ij} = max(A_{ij}^{k})\ \boldsymbol{OR}\ a_{ij} = max(A_{ij}^{k})$
        For $k\ from\ 1\ to\ t$
            If $min(A_{ij}^{k}) < median(A_{ij}^{k})\ \boldsymbol{AND}\ median(A_{ij}^{k}) < max(A_{ij}^{k})$
                $a_{ij} \leftarrow median(A_{ij}^{k})$
                Break
            End if
        End for
    End if
End for
```

FLOWCHART: -


Start
Read A := $[a_{ij}]_{m\times n}$
Initialize t = 9
Compute $\bar{A}$t
LOOP
For all i and j
DONE
DONE
For k from 1 to t
LOOP
$\min(A^{k}_{ij}) < \text{median}(A^{k}_{ij})$
&
$\text{median}(A^{k}_{ij}) < \max(A^{k}_{ij})$
&
$(a_{ij} = \min(A^{k}_{ij})$ OR $a_{ij} = \max(A^{k}_{ij}))$
NO
YES
$a_{ij} = \text{median}(A^{k}_{ij})$
Stop

$A^{k}_{ij} = k - \text{Symmetric pad matrix}$

Adaptive Riesz Mean Filter

Non-linear spatial filter

INTRODUCTION

In this section,two linear spatial window filters are introduced that are designed for high density impulse noise the Adaptive Riesz mean Filter(ARmF) and the Different Adaptive Modified Riesz mean Filter(DAMRmF). They work on pixels of the image that have extreme values (i.e. 0 or 255).

The ARmF calculates a riesz mean of the mask window based on the pixel similarity of each pixel and replaces the noisy pixel. It has an adaptivity condition for a completely noisy window to prevent error in the results.

The DAMRmF works similar to the ARmF but uses a modified version of the Riesz Mean in its calculations and employs an adaptivity condition based on the Adaptive Median Filter.

DETAIL

Adaptive Riesz mean Filter

To apply filtering to the pixels at the edge of the image, first the matrix were symmetrically padded with the required number of pixels. The padded image matrix for an image $A := [a_{ij}]_{m\times n}$ is represented by $\bar{\bar{A}}_t$ where t is the number of pixels padded around the image.

To process the image, the noisy pixels have to be determined . A binary matrix for that purpose was defined . The binary matrix of image A is defined as $B := \left[b_{ij}\right]_{m\times n}$where,

$$b_{ij} = \begin{cases} 0, & if\ a_{ij}\ is\ a\ noisy\ entry\ of\ A \\ 1, & otherwise \end{cases}$$

A window matrix was defined for filtering as A_{ij}^k of size (2k+1) × (2k+1), where i and j are the indices of the pixel a_{ij} being processed and k is the number of pixels taken around it.

$$A_{ij}^k = \begin{bmatrix} a_{(i-k)(j-k)} & \cdots & a_{(i-k)(j+k)} \\ \vdots & a_{ij} & \vdots \\ a_{(i+k)(j-k)} & \cdots & a_{(i+k)(j+k)} \end{bmatrix}_{(2k+1)\times(2k+1)}$$

Adaptive Riesz mean Filter (ARmF) works with adaptive windowing i.e. it starts with a window size of 3×3 (for k=1) and increases the window size if all the entries in the window are considered noisy before processing it.

The filter works on the idea of a weighted mean filter in which the weight of each pixel is determined by the pixel similarity of the pixel to the center pixel. The pixel similarity between two pixels is defined as

$$ps(a_{ij}, a_{st}) = \left(\frac{1}{1 + |i - s| + |j - t|}\right)^2$$

For pixels that are closer to each other, the value of pixel similarity will be closer to 1. The mean calculated using this method is called the Riesz mean of the window.

$$Riesz\ mean\ of\ \left(A_{ij}^{k}\right) = \frac{\sum_{(s,t)}\ a_{st} \times ps\left(a_{(k+1)(k+1)}, a_{st}\right)}{\sum_{(s,t)}\ ps\left(a_{(k+1)(k+1)}, a_{st}\right)}$$

Here, (s,t) is the set of all s and t such that a_{st} is a non-noisy entry of A_{ij}^{k} and $a_{(k+1)(k+1)}$ is the center pixel. The noisy center pixel is then replaced by the Riesz mean calculated this way.

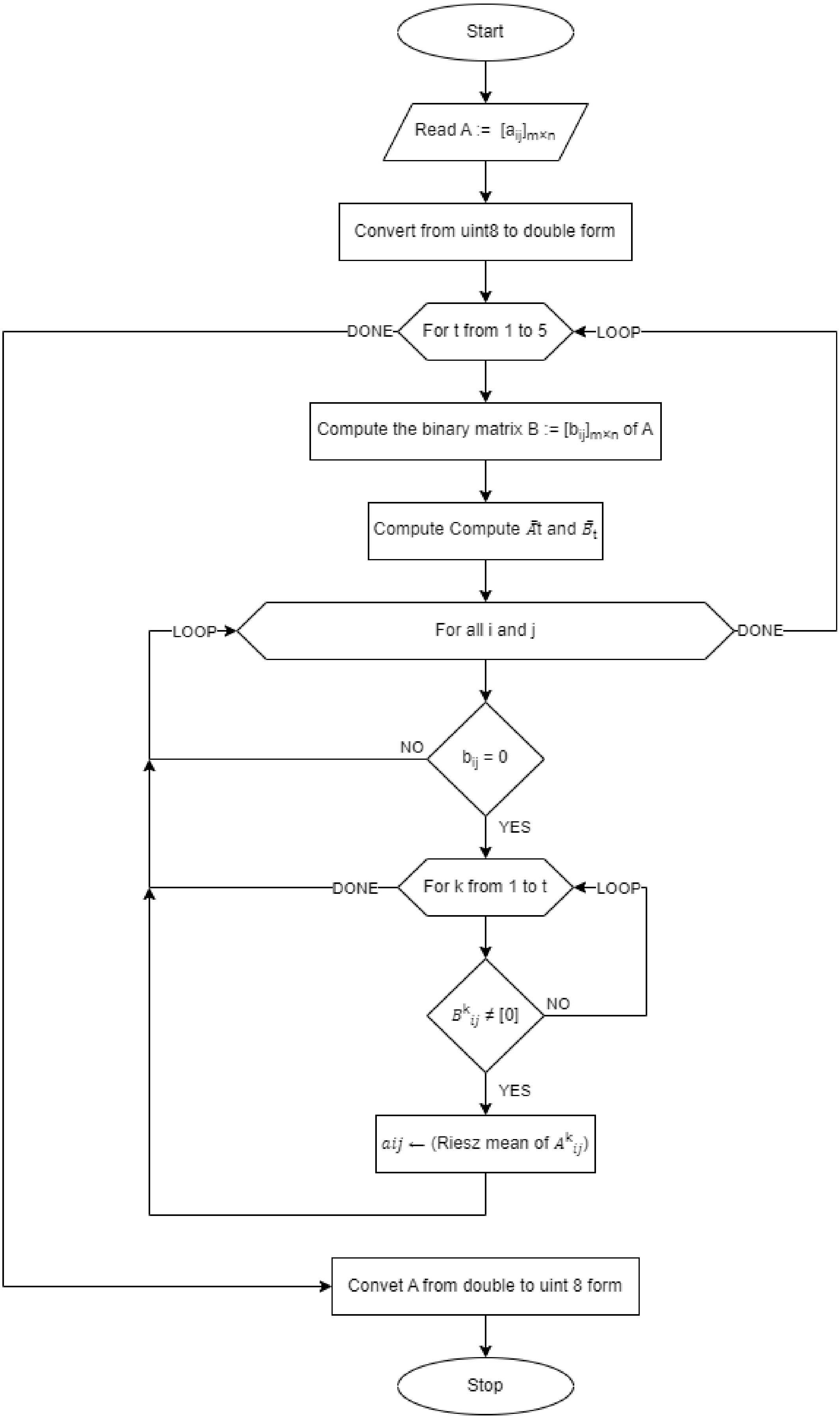
ALGORITHM:

Input: noisy matrix, $A := [a_{ij}]_{m\times n}$

Output: Denoised matrix, $A := [a_{ij}]_{m\times n}$

```
Convert A from uint8 to double form
For $t\ from\ 5\ to\ 1$
    Compute the binary matrix $B := [b_{ij}]_{m\times n}$
    Compute $\bar{\bar{A}}_t$ and $\bar{\bar{B}}_t$
    For $all\ i\ and\ j$
        If $b_{ij} = 0$
            For $k\ from\ 1\ to\ t$
                If $B_{ij}^{k}\ is\ not\ a\ zero\ matrix$
                    $a_{ij} \leftarrow \left(Riesz\ mean\ of\ A_{ij}^{k}\right)$
                    Break
                End if
            End for
        End if
    End for
End for
```

FLOWCHART: -


Start
Read A := $[a_{ij}]_{m\times n}$
Convert from uint8 to double form
For t from 1 to 5
DONE
LOOP
Compute the binary matrix B := $[b_{ij}]_{m\times n}$ of A
Compute Compute $\bar{A}$t and $\bar{B}_t$
For all i and j
LOOP
DONE
$b_{ij} = 0$
NO
YES
For k from 1 to t
DONE
LOOP
$B^k_{ij} \neq [0]$
NO
YES
$aij \leftarrow$ (Riesz mean of A^k_{ij})
Convet A from double to uint 8 form
Stop

$[b_{ij}]_{m\times n}$ = Binary Matrix A^k_{ij} = k − Symmetric pad matrix $ps(a_{ij}, a_{st})$=Pixel Singularity

Different Adaptive Modified Riesz mean Filter:-

Non-linear spatial filter

The Different Adaptive Modified Riesz mean Filter works similar to the Adaptive Riesz Mean Filter explained in the previous section. The initial padding and finding of the noisy pixels is the same as ARmF. The adaptive window condition in this filter first checks whether the median of the window is noisy or not. If median is noisy, the algorithm keeps moving on to a larger window around the noisy pixel until the median is found non-noisy or the maximum window size is reached. If median is not noisy, we replace the center pixel with the Modified Riesz mean of the window explained below.This filter also works on weighted mean principal but here the pixel similarity factor is changed with pixel weight defined below.

$$pw(a_{ij}, a_{st}) = \left(\frac{1}{1 + (i-s)^2 + (j-t)^2}\right)^2$$

So the weight of every pixel in the mean is decided by its pixel weight with respect to the center pixel of the window A_{ij}^k. This mean is called the Modified Riesz mean.

$$Modified\ Riesz\ mean\ of\ \left(A_{ij}^k\right) = \frac{\sum_{(s,t)}\ a_{st} \times pw\left(a_{(k+1)(k+1)}, a_{st}\right)}{\sum_{(s,t)}\ pw\left(a_{(k+1)(k+1)}, a_{st}\right)}$$

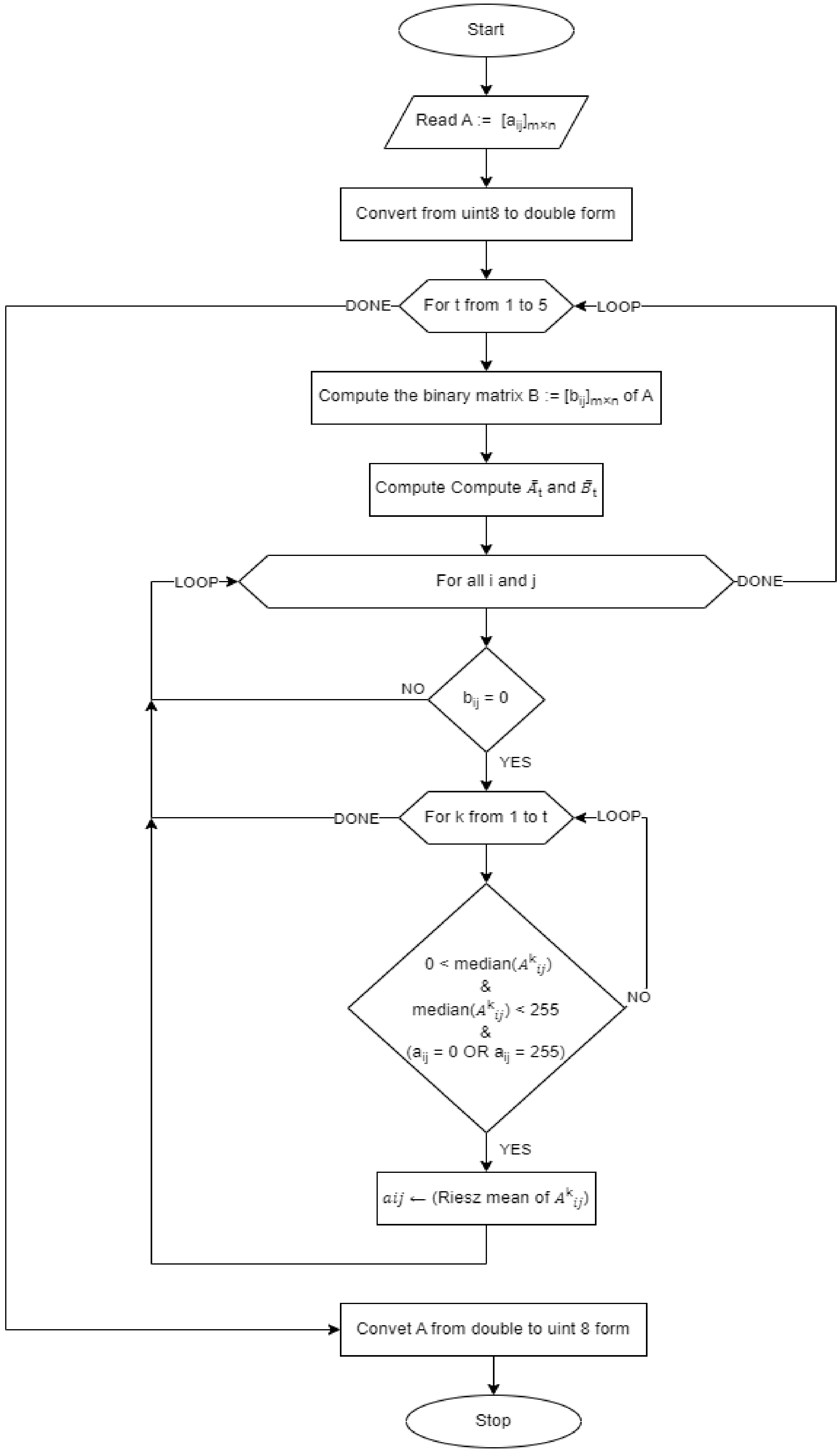
Here, (s,t) is the set of all s and t such that a_{st} is a non-noisy entry of A_{ij}^k and $a_{(k+1)(k+1)}$ is the center pixel. The noisy center pixel is then replaced by the Modified Riesz mean calculated this way.

ALGORITHM: -

Input: noisy matrix, $A := [a_{ij}]_{m\times n}$ **Output:** Denoised matrix, $A := [a_{ij}]_{m\times n}$

```
Convert A from uint8 to double form
For $t\ from\ 5\ to\ 1$
    Compute the binary matrix $B := [b_{ij}]_{m\times n}$
    Compute $\bar{\bar{A}}_t$ and $\bar{\bar{B}}_t$
    For $all\ i\ and\ j$
        If $b_{ij} = 0$
            For $k\ from\ 1\ to\ t$
If $0 < median(A_{ij}^k)\ \&\ median(A_{ij}^k) < 255\ \textbf{AND}\ (a_{ij} = 0\ \textbf{OR}\ a_{ij} = 255)$
                    $a_{ij} \leftarrow (Modified\ Riesz\ mean\ of\ A_{ij}^k)$
                    Break
                End if
            End for
        End if
    End for
```

FLOWCHART: -


Start
Read A := $[a_{ij}]_{m\times n}$
Convert from uint8 to double form
For t from 1 to 5
DONE
LOOP
Compute the binary matrix B := $[b_{ij}]_{m\times n}$ of A
Compute Compute $\bar{A}_t$ and $\bar{B}_t$
For all i and j
LOOP
DONE
$b_{ij} = 0$
NO
YES
For k from 1 to t
DONE
LOOP
0 < median(A^k_{ij}) & median(A^k_{ij}) < 255 & (a_{ij} = 0 OR a_{ij} = 255)
NO
YES
aij ← (Riesz mean of A^k_{ij})
Convet A from double to uint 8 form
Stop

$[b_{ij}]_{m\times n}$ = Binary Matrix A^k_{ij} = k − Symmetric pad matrix ps(a_{ij}, a_{st})=Pixel Singularity

Alpha Trimmed Mean Filter for Image Denoising

Non-linear spatial filter

Introduction:

It is nonlinear windowed filter that uses the median and mean filters. The idea behind this filter is to look at the area around each pixel and remove the pixels that has the highest and lowest value , and use the remaining elements to get the mean value. The alpha parameter controls how many elements are removed.

Background:

The idea is to sort the items, select the first and last element out of the sorted set, and then calculate the mean value using the remaining elements. For example below is the representation of how alpha trimmed mean is calculated

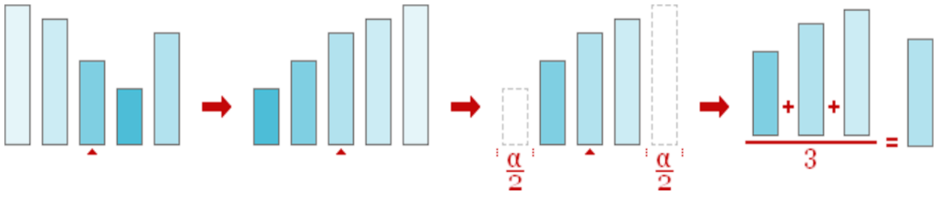


Fig. 1. Alpha-trimmed mean calculation.

To obtain an alpha-trimmed mean,items are arranged in the ascending/descending order and elements from first and last are removed from the collection, and then mean of the remaining elements is taken.

The formula used here to calculate alpha trimmed mean for a given noisy pixel which is to be replaced with the calculated value:

$$\mathrm{f(x,y)} = \sum\nolimits_{(s,t)\epsilon Sx,y} g(s,t)$$

Where:

m,n : Window dimensions

d: This parameter is used for calculating alpha trimmed mean which is equal to 2α

and α is the d/2 highest and d/2 lowest intensity values of the window array.

$\sum g_r(s,t)$: Summation of the remaining elements of the window array for finding the mean.

A few remarks concerning the **alpha parameter**, which controls item trimming. The number of elements that must be removed is referred to as alpha; in this filter**, d is 5** which means **α is 2.5** but here **α is taken as 2** . As the filter is symmetric, alpha should be an even nonnegative integer smaller than the size of the filter window.

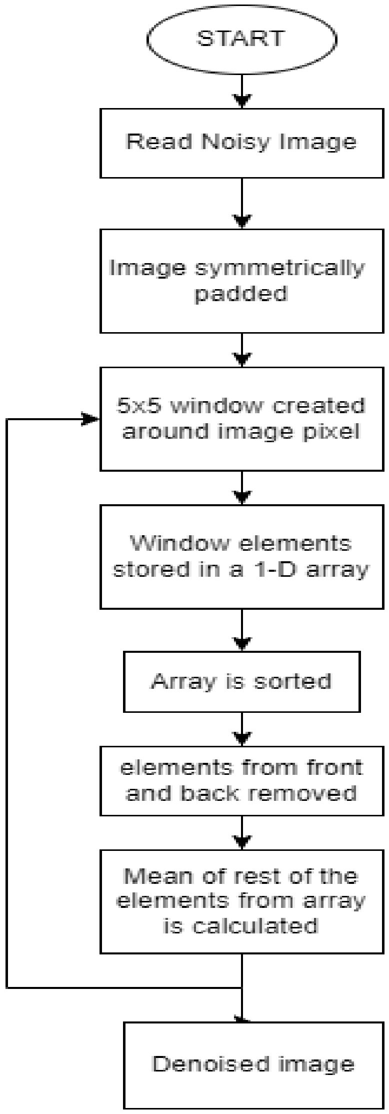
The alpha-trimmed mean filter degenerates into the mean filter when the alpha parameter's minimum value is 0. As alpha reaches its maximum value, which is equal to the filter window size minus one, the filter degenerates into a median filter.

Algorithm:

1. Window is placed over each pixel.
2. form a 1-D window array.
3. Sort the elements in the window array.
4. Elements are removed from front and at the back of the arranged set.
5. Calculate the mean of the remaining elements of the window array

Note: The widow dimensions can be 3x3,5x5,7x7 according to which value of d and α will change, but based on the observation 5X5 window gives the best result.

Flow chart:


START
Read Noisy Image
Image symmetrically padded
5x5 window created around image pixel
Window elements stored in a 1-D array
Array is sorted
elements from front and back removed
Mean of rest of the elements from array is calculated
Denoised image

Geometric Mean Filter for Image Denoising

Non-linear spatial filter

Introduction:

It is a type of image filter used to filter noises and smooth out images.The idea is to find the geometric mean of the window and replace the element with geometric mean. The geometric mean's output picture, G(x,y), is provided by

$$\mathrm{G(x,y)}=[\prod_{i,j\epsilon S} S(i,j)]^{\mathbf{1/mn}}$$

Where:

G(x,y): Calculated Geometric Mean which will be replaced with the center pixel of the selected window.

m,n : Window dimensions.

Π(S(i,j)): Product of Original Image Elements within

the selected window.

Background:

Geometric mean of the elements present in the geometric mean window is calculated and the center pixel of the window is replaced with result.The pixel (x,y) in the output image, for example, product of all 8 of the surrounding pixels raised to the 1/9 power is calculated when the window size is set to 3 by 3.

For instance, given below is a 3x3 window of the center pixel:

5	16	22
6	3	18
12	3	15

Gives the result: (5*16*22*6*3*18*12*3*15)$^{(1/9)}$ = 8.77

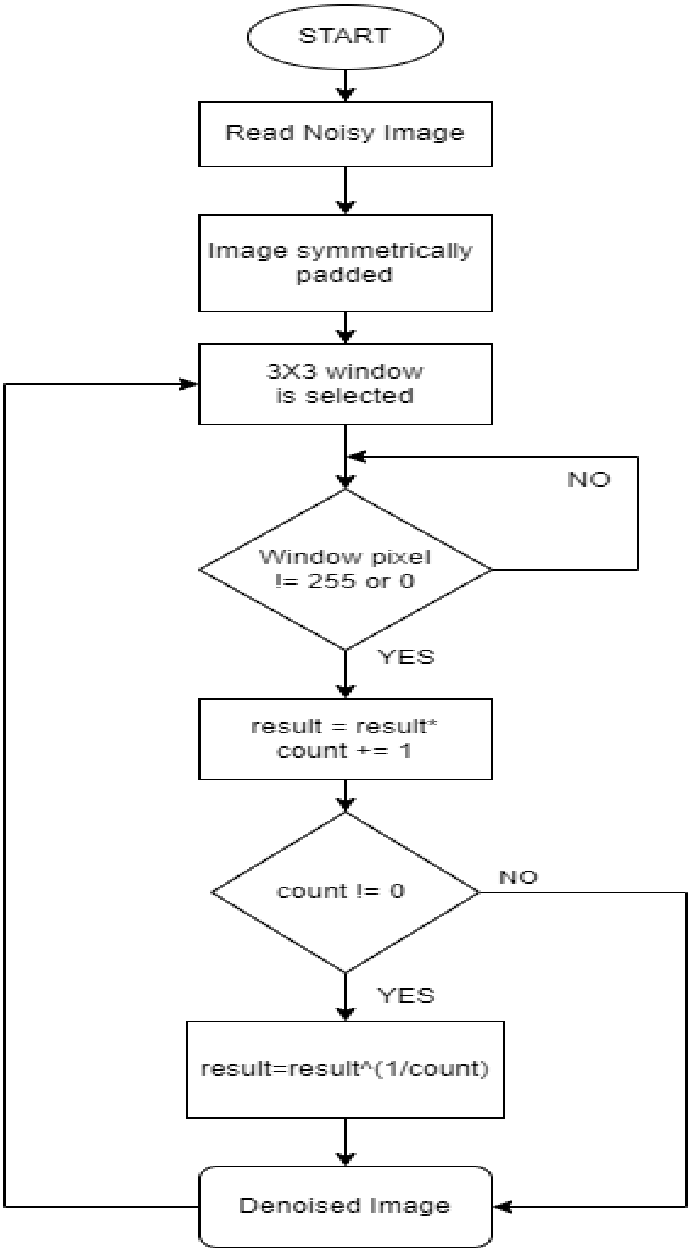
The centre element of the widow (i.e 3) will get replaced with the above calculated value i.e 8.77.

Algorithm:

1. Read the noisy image.
2. Symmetrically pad the noisy image matrix.
3. Select the window size.
4. If window pixel ≠ 255 or 0 product of pixel value is stored in result array and element count is incremented.
5. Find the geometric mean of result array.
6. Replace the centre element of the window with calculated geometric mean.
7. Do it for all elements of the noisy image matrix.

Note: Noisy Image has been symmetrically padded for the smoothening of the edges. For a 3x3 window, padding of 1 unit should be provided .It increases with the increase in the window size.

Flow chart:



Sector Rotation Filter

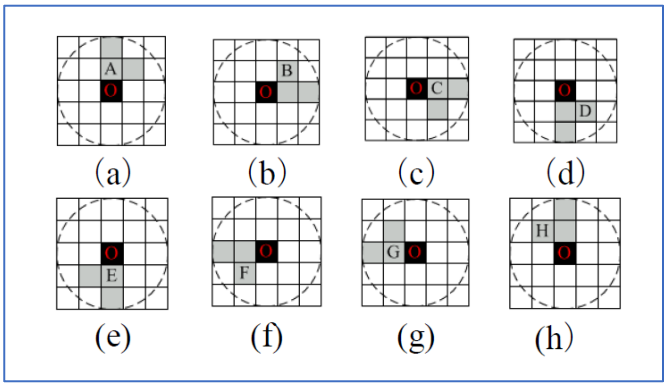
Non-linear spatial filter

INTRODUCTION: -

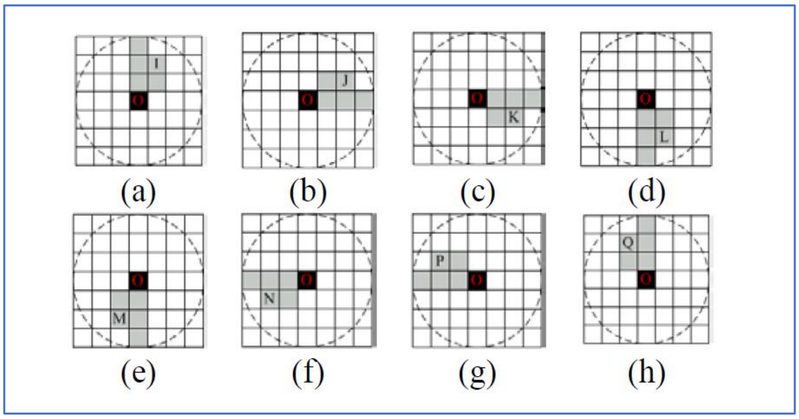
Image processing is the process of removing noise pollution which is caused due to several environmental and equipment factors of image acquisition. These noise pollution cause interference to image processing. Different noises are added to the image to check the working of the filter and comparison is being made with the help of certain parameters. Image denoising with the help of standard median filter somewhat fails to clarify the borders of a noisy image and thus resulting the image becoming blurrier. Adaptive median filtering technique, which is done from calculating and checking the weight of a pixel and perform certain mathematical operation on them, and then calculates the median and store it as the output. This methodology is a superior version of basic noise filtering technique, but the corners of certain change in weighted pixels remain in same noise condition with the inefficient time taken over the basic technique. Further on certain algorithms were developed and filtering effect on low density noise has been improved till some extent. Moreover, when there are more extreme points in the selected window, the value of blurred coefficient and fuzzy variables are still noisy, so the resultant picture obtained is still having relatively high noise. So, to fulfil this fault, an algorithm is developed to increase the window dimensions variably according to the noise found on the sector of the window selected of the image. Here, the value of center of the window in all possible dimensions is calculated to have the relation between the points with large difference and that are the ones taken in the result.

IMPROVEMENT IN THE ALGORITHM: -

For image with low noise window of smaller size would work significantly better than the one with large size. The rotation rule for a window of 5x5 window is shown below.



There are the 8 difference points considered according to the size of the window. The fan region with a small difference indicates a high difference relation with the pixel value of the non-noisy area. If all the difference points are found at corner or the higher value, then the noise percentage is higher, hence a window of significantly larger size is selected. The window of higher size is shown below.



(a) (b) (c) (d)

(e) (f) (g) (h)

If the same condition is found which lead to expansion of window size is the above condition, then the same would be applied here as well and a window of higher dimension will be selected.

Formulate the value of difference in the 8-difference point in the fan selected in the window.

(x-2,y-2)	(x-2,y-1)	(x-2,y)	(x-2,y+1)	(x-2,y+2)
(x-1,y-2)	(x-1,y-1)	(x-1,y)	(x-1,y+1)	(x-1,y+2)
(x,y-2)	(x,y-1)	O (x,y)	(x,y+1)	(x,y+2)
(x+1,y-2)	(x+1,y-1)	(x+1,y)	(x+1,y+1)	(x+1,y+2)
(x+2,y-2)	(x+2,y-1)	(x+2,y)	(x+2,y+1)	(x+2,y+2)

a(x-1, y) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

b(x-1, y+1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

c(x, y+1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

d(x+1, y+1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

e(x+1, y) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

f(x+1, y-1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

g(x, y-1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

h(x-1, y-1) = 2*f(x-1,y) – f(x-2,y) – f(x-1,y+1)

ALGORITHM: -

1. Convert the input image to grayscale, if necessary.
2. Choose a center point for the rotation. This could be the center of the image or any other point of interest.
3. Define the size of the filter to determine the size of the sectors to be rotated.

4. For each pixel in the image:

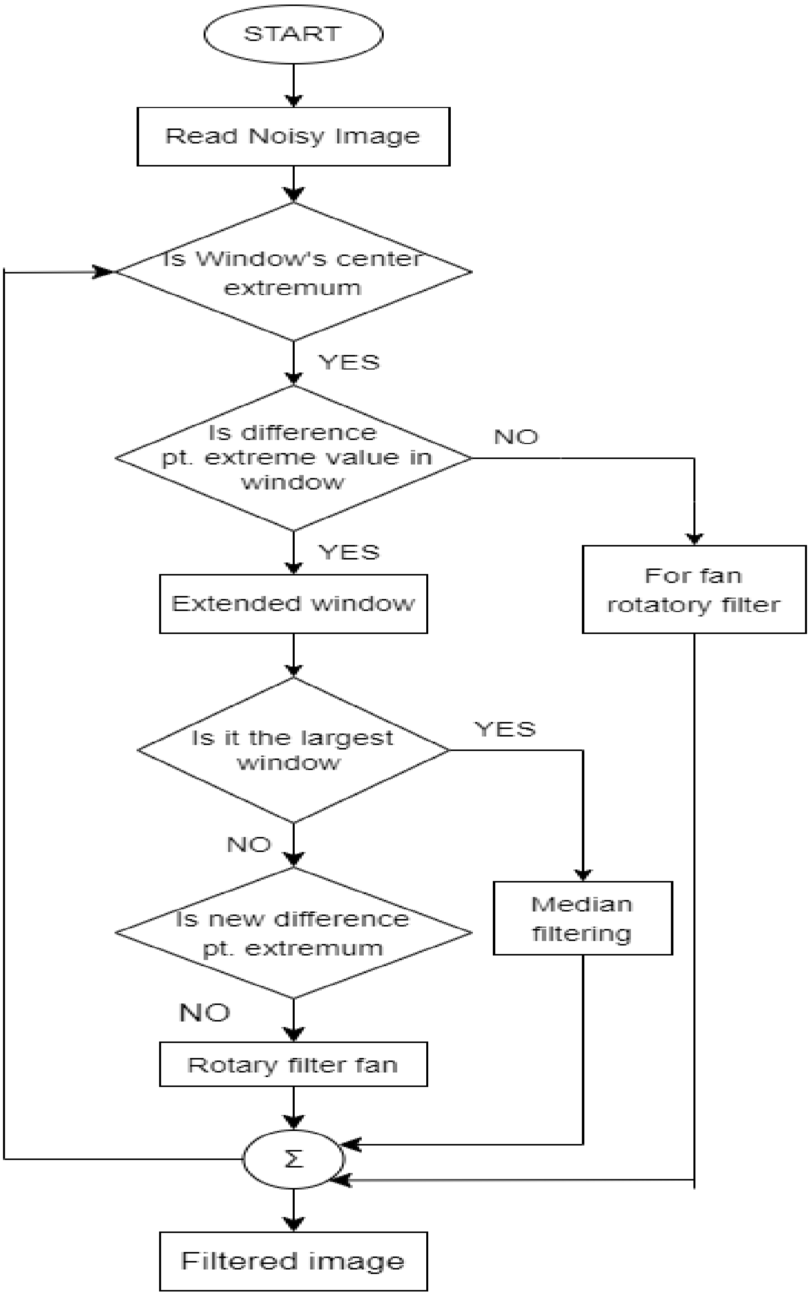
   a. Calculate the distance from the pixel to the center point.

   b. Calculate the angle between the pixel and the center point.

   c. Rotate the sector by an amount proportional to its distance from the center point.

   d. Assign the rotated pixel value to the original pixel location in the output image.
5. For the maximum size of the window (above 13×13 window), the median of the window is directly calculated for output.
6. Add the pixel without filtering and the pixel points after applying operation to obtain output image.

FLOWCHART: -



Modified Median Filter

Non-linear spatial filter

INTRODUCTION: -

An image is a composition of squared pixels arranged in row and column. It is captured by capturing devices like camera or scanner and stored in the mass storage of the computer system. A captured image is subjected to many kinds of distortion during the stages of processing, storing, compressing, transmitting and others. During the transfer of the image, noise may be added along the actual information. The unwanted information that added along the required information is due to certain environmental variations and faulty locations during transfer. It is an improvement on the standard median filter, which is a commonly used technique for noise reduction. The modified median filter extends the concept of the median filter by using additional information about the local characteristics of the signal to improve noise reduction performance.

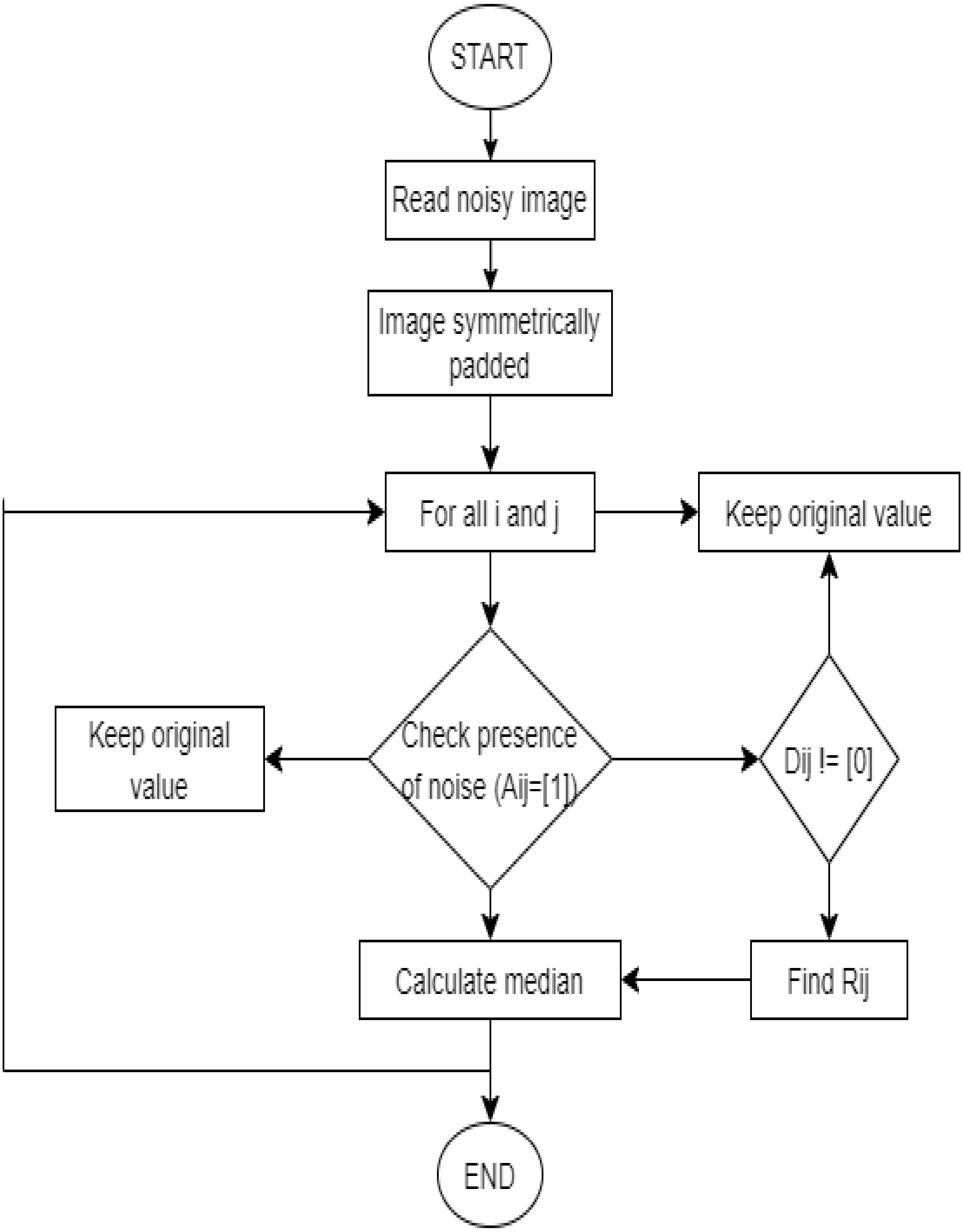
The modified median filter is particularly useful when dealing with non-Gaussian noise or when the signal is corrupted by large outliers. It works by examining the values of neighbouring pixels or samples in a signal and replacing the central pixel or sample with a value that is a function of the median of the neighbouring values. The exact function used to calculate the replacement value depends on the specific application and the nature of the noise present.

Overall, the modified median filter is a powerful tool for noise reduction in a wide range of applications, including image and audio processing, and can improve the quality and accuracy of signal analysis and interpretation

ALGORITHM: -

1. Read the noise image matrix img = [a_{ij}]XxY, where {x,y} >1.
2. Obtain the binary matrix B = [d_{ij}]$_{mxn}$ of img.
3. Calculate and obtain the padded matrix.
4. For all i and j. If d_{ij}=1, then keep value of a_{ij} else B_{ij}!=[0] then
    a) Find the value of R_{ij} for img_{ij}.
    b) Evaluate the median of R_{ij}.
    c) Replace this value to the c_{ij} else keep the value of a_{ij}.
5. Merge the median value and the original values together.

FLOWCHART: -


START
Read noisy image
Image symmetrically padded
For all i and j
Keep original value
Keep original value
Check presence of noise (Aij=[1])
Dij != [0]
Calculate median
Find Rij
END

Recursive Cubic Spline Interpolation Filter for Image Denoising

Non-linear spatial filter

Introduction:

Interpolation is defined as a technique to derive an equation with the help of given constant discrete points. For any higher order function defined over a range [x,y], this method can used to generate a lower order function. The more the data points are available to interpolate, the more accurate lower order function can be generated. The points/ data are connected through a smooth curve, which is being utilized by cubic spline interpolation filter.

Background:

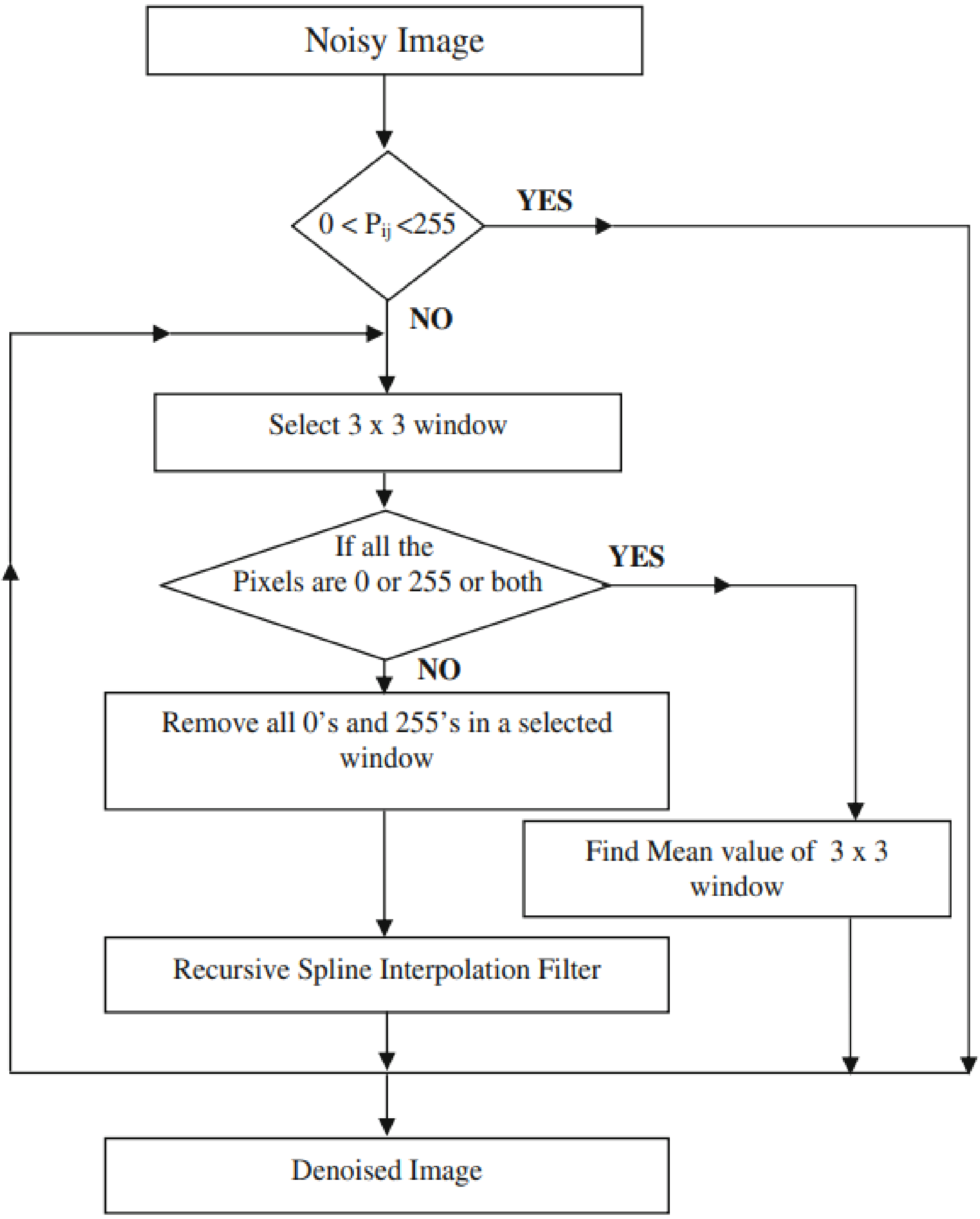
The filter is being used to remove Impulse noise. Basic idea behind this filter is to use the interpolation technique over the know noisy pixel. The filter would check each pixel to be either noise i.e. having a value of either 0 or 255, if it does it would take a 3x3 window around it, apply the interpolation method to give out the most accurate pixel value it is supposed to have. If the pixel is not found out to be noisy it would simply ignore it and move to next pixel.

Algorithm:

1. Read the noisy image.
2. Symmetrically pad the noisy image matrix.
3. If the pixel is not noisy move to next pixel
4. If the pixel is noisy, Select a 3x3 window around the noisy image
    a. If all the images are noisy in the 3x3 matrix take the mean value of the matrix
    b. if not, ignore all the other noisy pixel present in the 3x3 matrix and apply the cubic spline interpolation on it
    c. replace the generated value with the noisy pixel
5. Do it for all elements of the noisy image matrix.

Note: Noisy Image has been symmetrically padded for the smoothening of the edges. For a 3x3 window padding of 1 unit should be provided .It increases with the increase in the window size.

Flow Chart:


Noisy Image
0 < P_ij <255
YES
NO
Select 3 x 3 window
If all the Pixels are 0 or 255 or both
YES
NO
Remove all 0's and 255's in a selected window
Find Mean value of 3 x 3 window
Recursive Spline Interpolation Filter
Denoised Image

- $P_{i,j}$: It refers to the pixel value at location (i,j) of the matrix

Basic Mean Filter for Image Denoising

Linear spatial filter

Introduction:

The basic mean filter is a type of image filter used to smear out noise and smooth out images. It is based on the algebraic mean in mathematics. The algebraic mean's output picture, f(x,y), is provided by

$$f(x,y)=\frac{1}{mn}\sum_{(s,t)\epsilon Sx,y} g(s,t)$$

Where:

f(x,y): Calculated Algebraic Mean which will be replaced with the center pixel of the selected window.

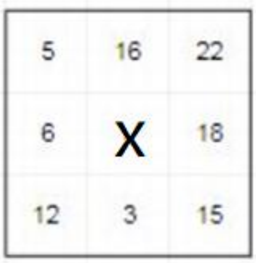
m,n : Window dimensions.

∑g(s,t): Sum of Original Image Elements within the selected window.

Background:

The filter is defined to remove impulse noise. The filter iterates over symmetrically padded matrix, to check for pixels having a value impulse noise. If it does it would take a 3x3 window across it, find the mean of the window and replace the noisy pixel with the generated mean value. If the pixel is not found out to be noisy to be noisy it would simply move to the next pixel.

For instance, given below is a 3x3 window of the center pixel:



5	16	22
6	X	18
12	3	15

Here the centre pixel is noisy.
Replacing the centre pixel with the mean for the window

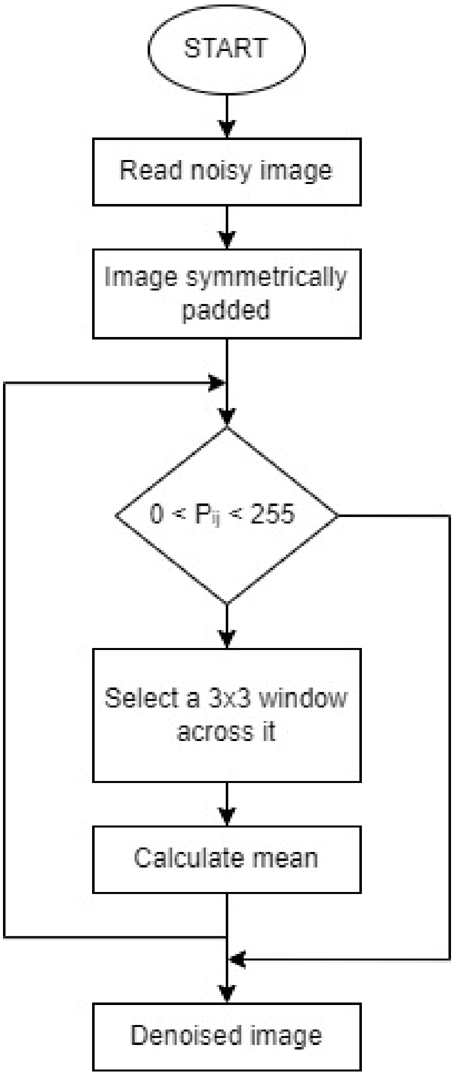
Gives the result: (5+16+22+6+3+18+12+3+15)/8 = 12.5 since the operation has been done in uint8, it would convert 12.5 to 13, and hence the centre pixel would be replaced by 13.

Algorithm:

1. Read the noisy image.
2. Symmetrically pad the noisy image matrix.
3. If the pixel is not noisy move to next pixel
4. If the pixel is noisy, Select a 3x3 window around the noisy image
    a. apply the mean filter on it and obtain algebraic mean of the 3x3 window
    b. replace the generated value with the noisy pixel
5. Do it for all elements of the noisy image matrix.

Note: Noisy Image has been symmetrically padded for the smoothening of the edges. For a 3x3 window padding of 1 unit should be provided .It increases with the increase in the window size.

Flow chart:



❖ $P_{i,j}$: It refers to the pixel value at location (i,j) of the matrix

Min/Max filters for Image Denoising

Non-linear spatial filter

Introduction:

The Min-Max filter is a morphological filter which can be utilised for image denoising. Max and Min filter are used to process image data in spatial domain. Furthermore, they belong among order-statistic filters along with filter. It is a non-linear filter that, within a predetermined window size, replaces each pixel in an image with the lowest or maximum value of its neighbors' pixels. The smooth borders of the picture are successfully tracked by the min-max filter, and the smooth regions are successfully tracked by their average.

Formula used for calculating above-described pixel values:

$$\mathrm{f(x,y)} = \max_{(s,t)\epsilon Sx,y}\{g(s,t)\}$$

$$\mathrm{f(x,y)} = \min_{(s,t)\epsilon Sx,y}\{g(s,t)\}$$

For Max filter:

f(x,y): Maximum value of the pixel present the selected window.

g(s,t): pixel intensity values of the selected window.

For Min filter:

f(x,y): Minimum value of the pixels present the selected window.

g(s,t): pixel intensity values of the selected window.

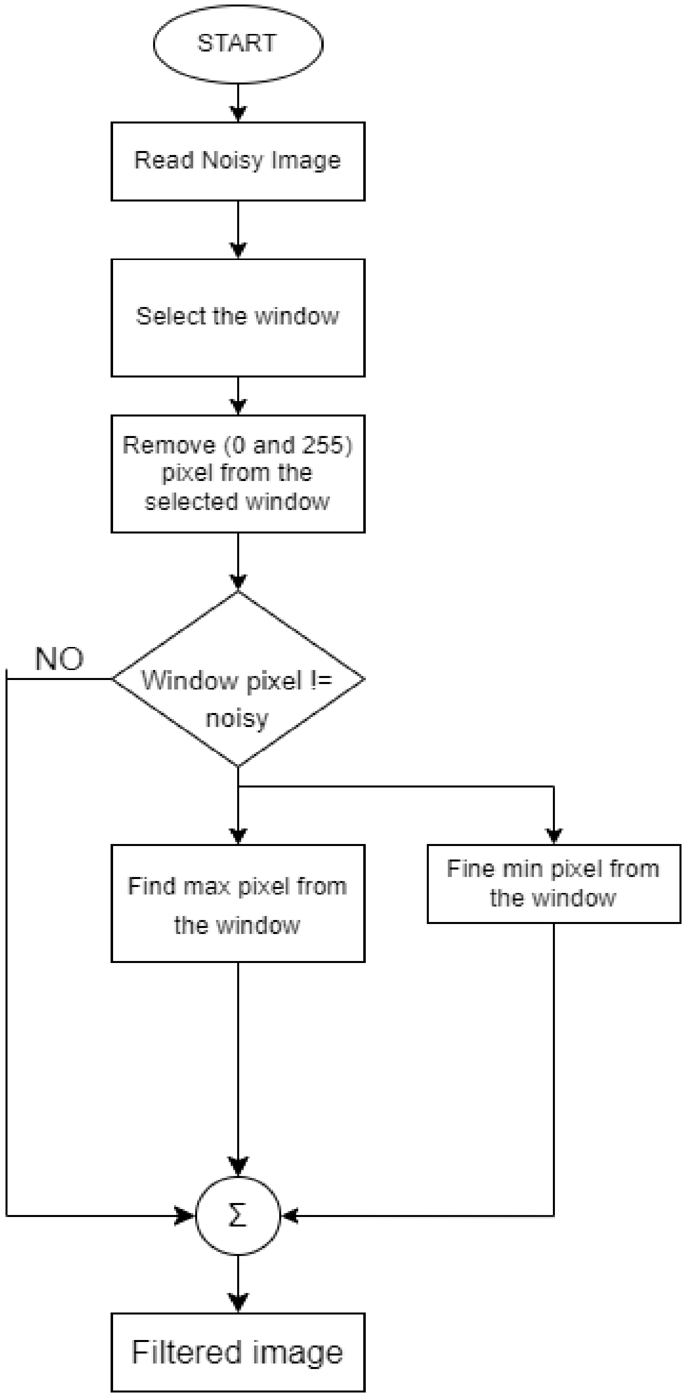
Background:

In Min/Max filter first a window is selected. Window size can be 3x3,5x5,7x7.. etc not necessarily the larger the window size the better the output cause it depends on how noisy the image is. In both Min/Max filter first 0 and 255 pixel values are eliminated from the window this made the filter to work on combination of salt & pepper noise rather than individual salt noise and pepper noise because in case of Max filter, the centre pixel will now get replaced with the maximum pixel value of the window which is not 255 thus preventing the image from becoming totally white, similarly for Min filter this prevents the image from becoming totally black

Algorithm:

1. Read the noisy image.
2. Pad the image symmetrically.
3. Select the window.
4. Remove the 0 and 255 pixel values from the selected window.
5. (A) For Max filter: Find the maximum pixel value from the window.
   (B) For Min filter: Find the minimum pixel value from the window.
6. Replace the center pixel value of the window from the calculated Min/Max value.

Flow chart:


START
Read Noisy Image
Select the window
Remove (0 and 255) pixel from the selected window
NO
Window pixel != noisy
Find max pixel from the window
Fine min pixel from the window
Σ
Filtered image

Basic Median Filter for Image Denoising

Non-linear spatial filter

Introduction:

The basic median filter is a type of image filter used to smear out noise while preserving the of the images. It is based on the algebraic median in mathematics. The algebraic median output picture, f(x,y), is provided by

$$f(x,y) = \underset{(s,t)\epsilon Sx,y}{\text{median}}\{g(s,t)\}$$

Where:

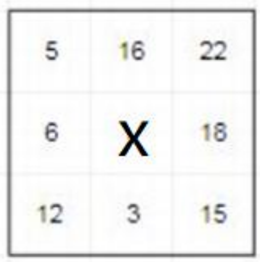
f(x,y): Calculated Algebraic Median which will be replaced with the center pixel of the selected window.

g(s,t): pixel value of Image Elements within the selected window.

Background:

The filter is defined to remove impulse noise. The filter iterates over symmetrically padded matrix, to check for pixels having a value impulse noise. If it does, it would take a 3x3 window across it, find the median value of the window and replace the noisy pixel with the generated mean value. If the pixel is not found out to be noisy to be noisy it would simply move to the next pixel.

For instance, given below is a 3x3 window of the center pixel:



5	16	22
6	X	18
12	3	15

Here the center pixel is noisy.
Replacing the center pixel with the median for the window

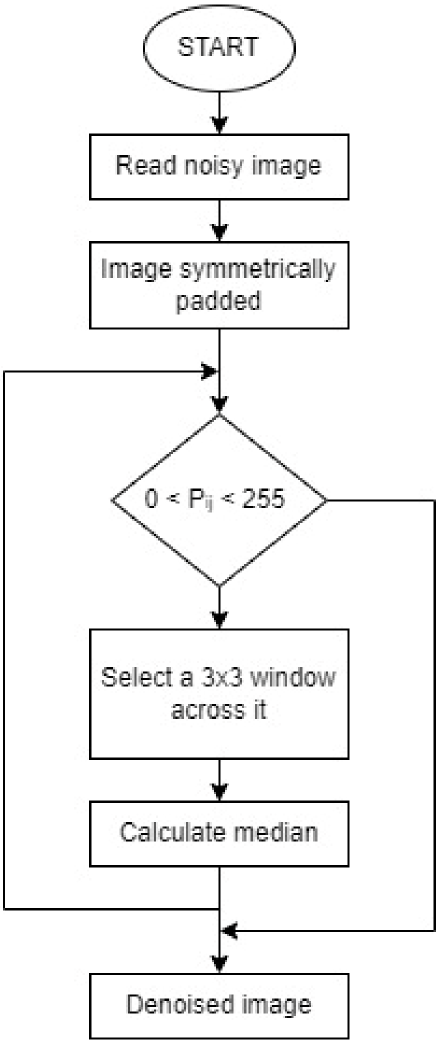
Gives the result: 13.5
since operation has been done uint8, it would convert 13.5 to 14, and hence the center pixel would be replaced by 14.

Algorithm:

1. Read the noisy image.
2. Symmetrically pad the noisy image matrix.
3. If the pixel is not noisy move to next pixel
4. If the pixel is noisy, Select a 3x3 window around the noisy image
    a. apply the median filter on it and obtain algebraic median of the 3x3 window
    b. replace the generated value with the noisy pixel
5. Do it for all elements of the noisy image matrix.

Note: Noisy Image has been symmetrically padded for the smoothening of the edges. For a 3x3 window padding of 1 unit should be provided .It increases with the increase in the window size.

Flow chart:

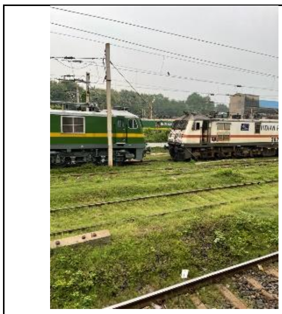
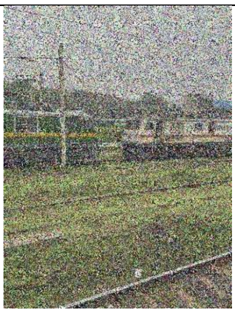
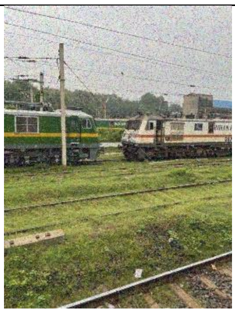
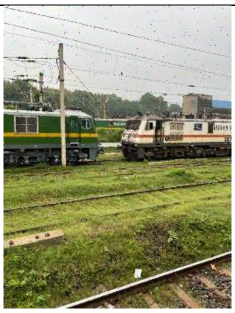
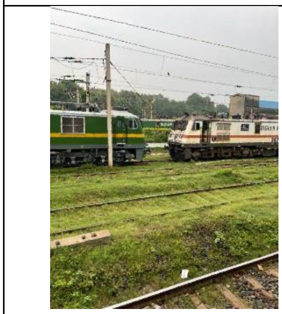
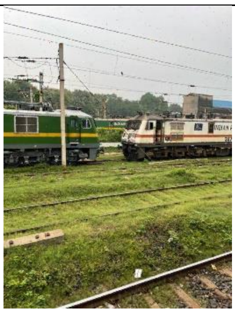
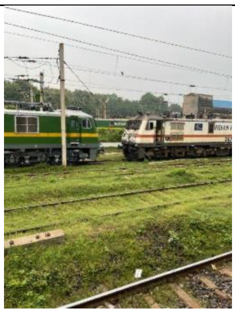
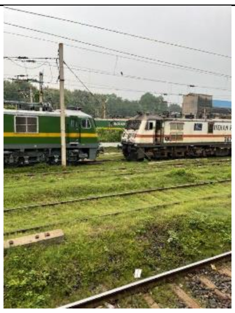
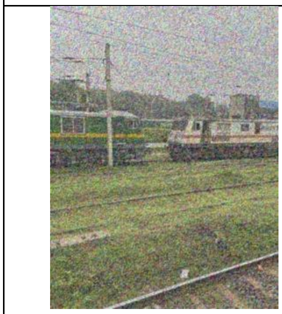
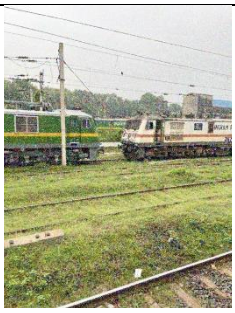
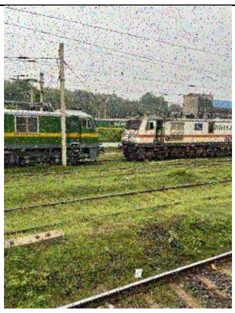
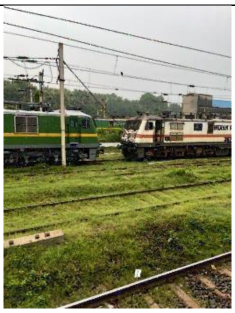
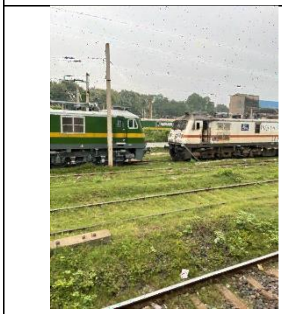
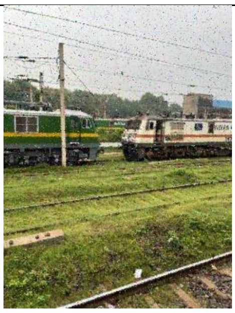
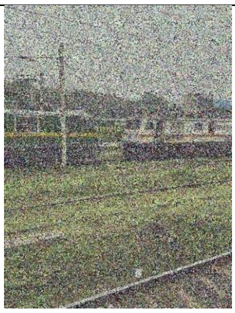


❖ $P_{i,j}$: It refers to the pixel value at location (i,j) of the matrix

RESULT :

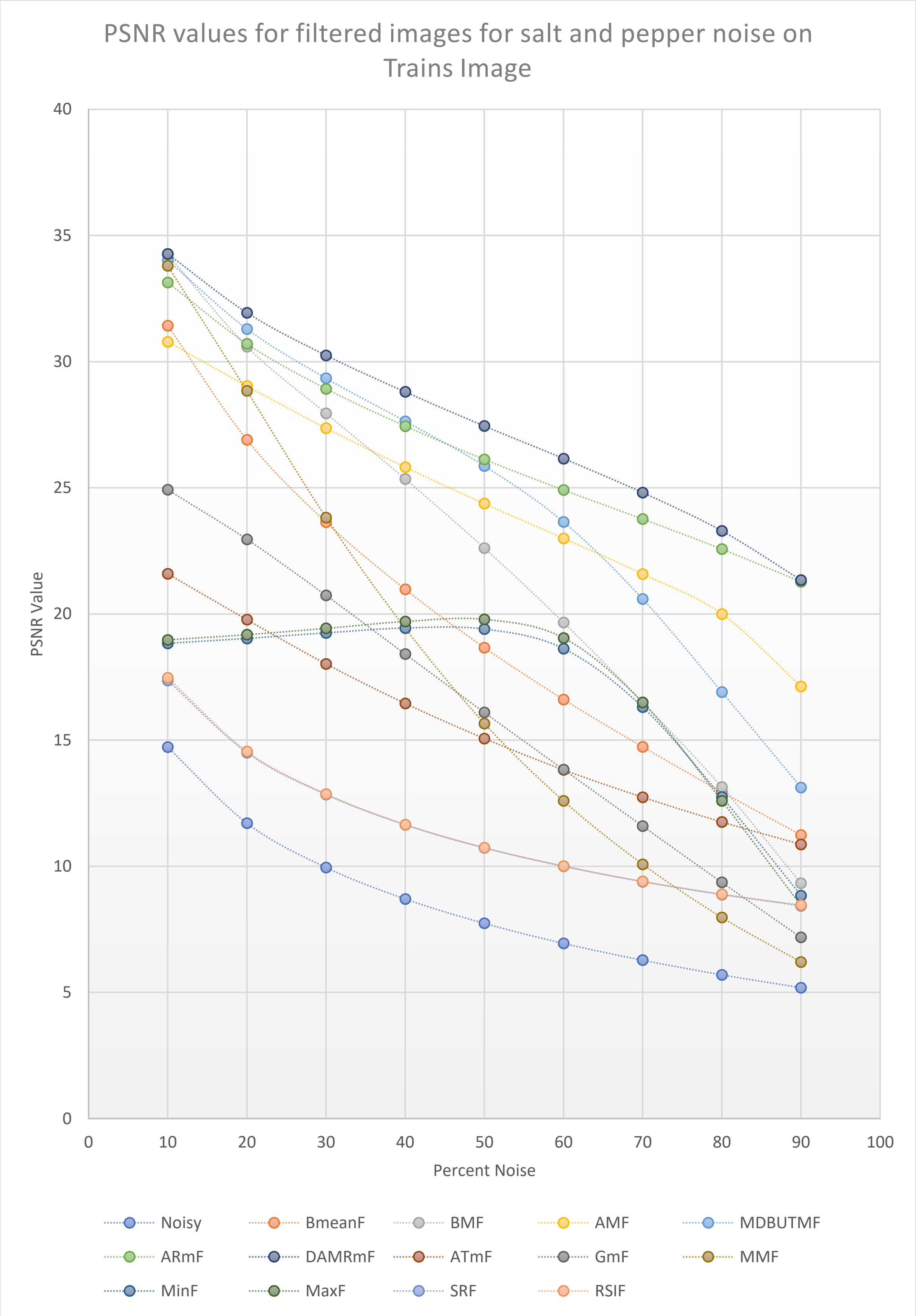
Below are the results of the performance of the various filters mentioned in this document:

50% salt and pepper noise was added to the given image and performance of the filter are as follows:

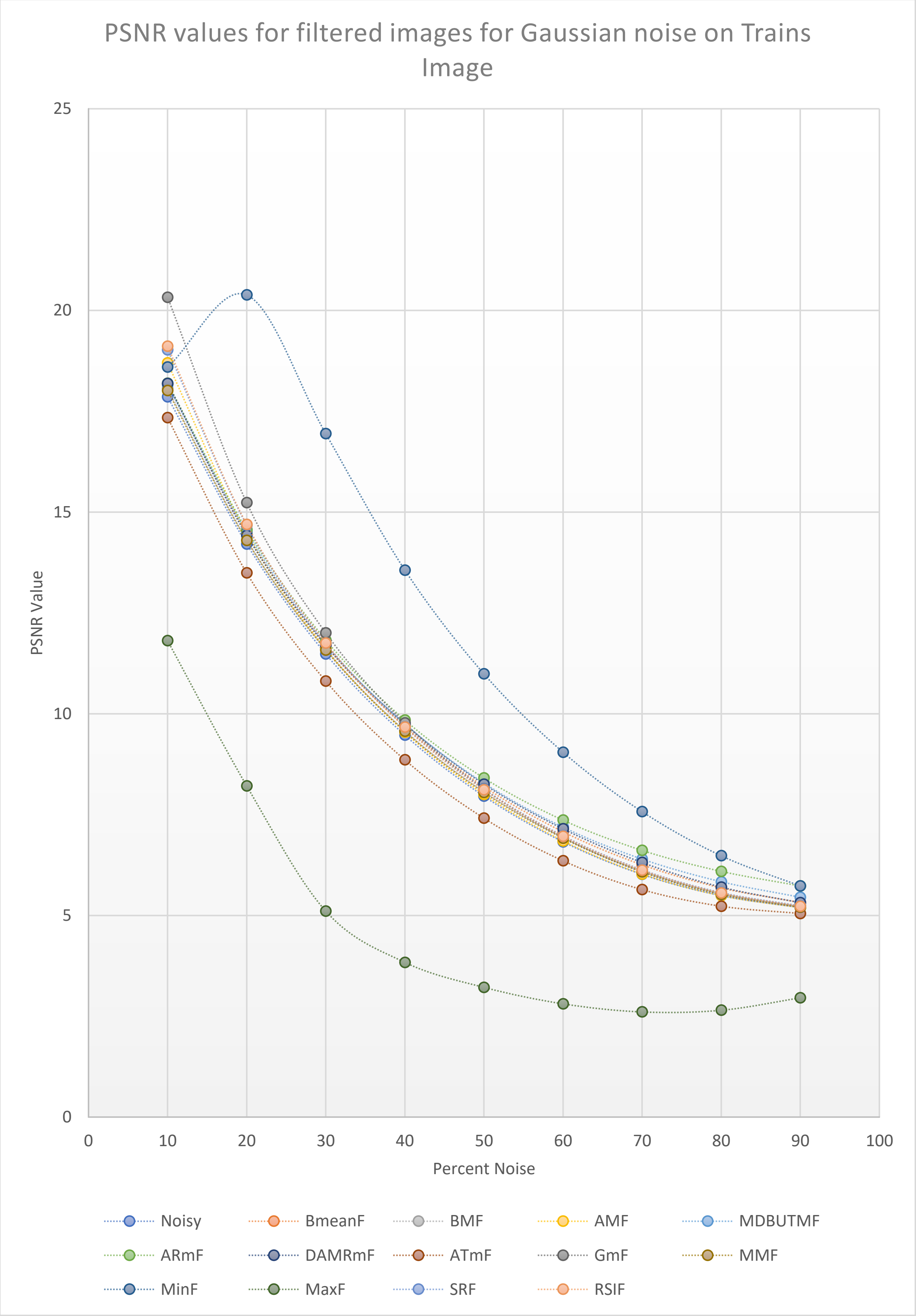
 Original image	 Noisy Image	 Basic Mean Filter	 Basic Median Filter
 Adaptive Median Filter	 Modified Decision Based Unsymmetrically Trimmed Median Filter	 Adaptive Riesz Mean Filter	 Different Adaptive Modified Riesz Mean Filter
 Alpha Trimmed Mean Filter	 Geometric Mean Filter	 Modified Mean Filter	 Min Filter
 Max Filter	 Sector Rotational Filter	 Recursive Spline Interpolation Filter	

Below are the performance of the filters based on PSNR values in graphical form:

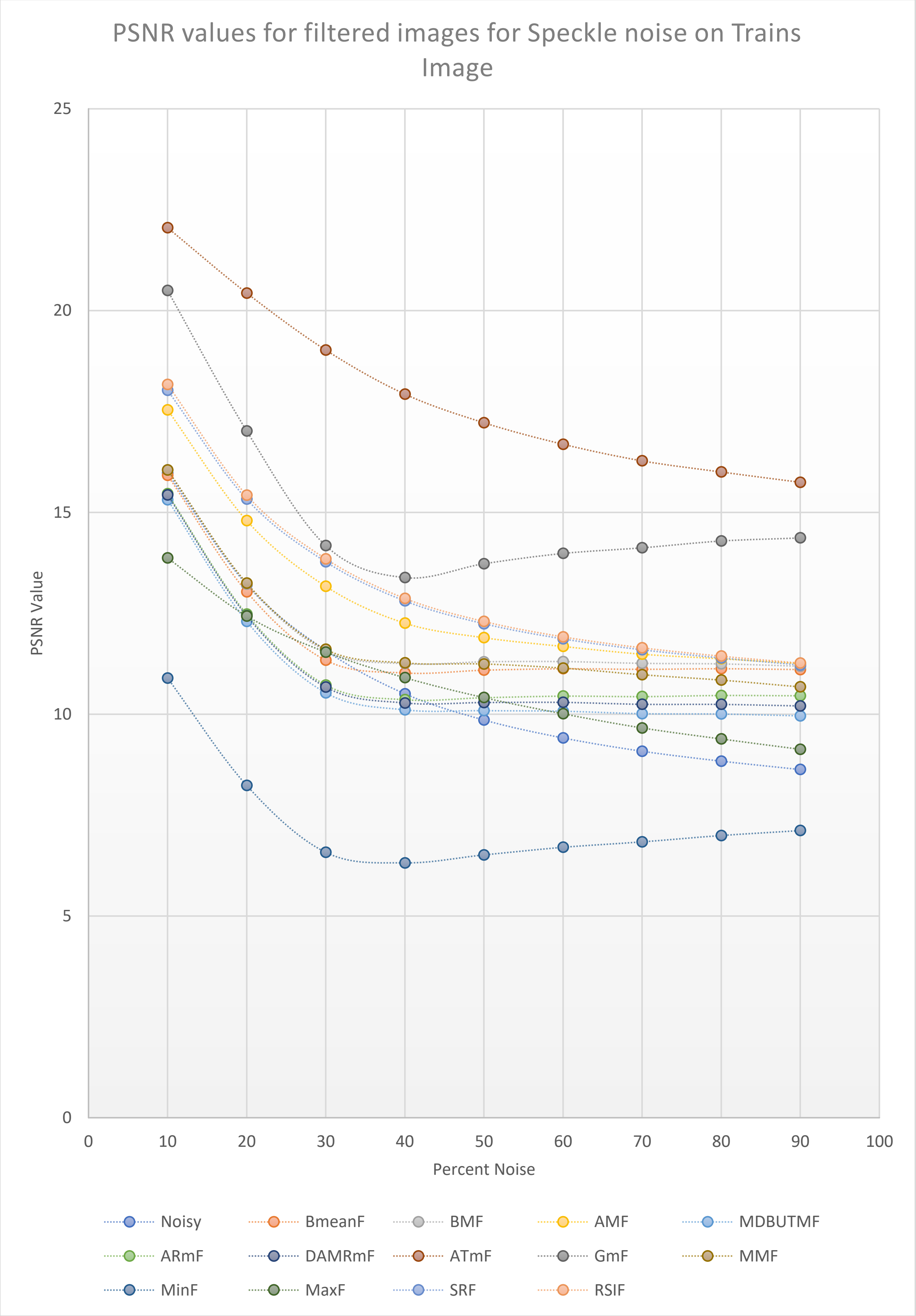
Noise: Salt and Pepper (From 10% to 90%);



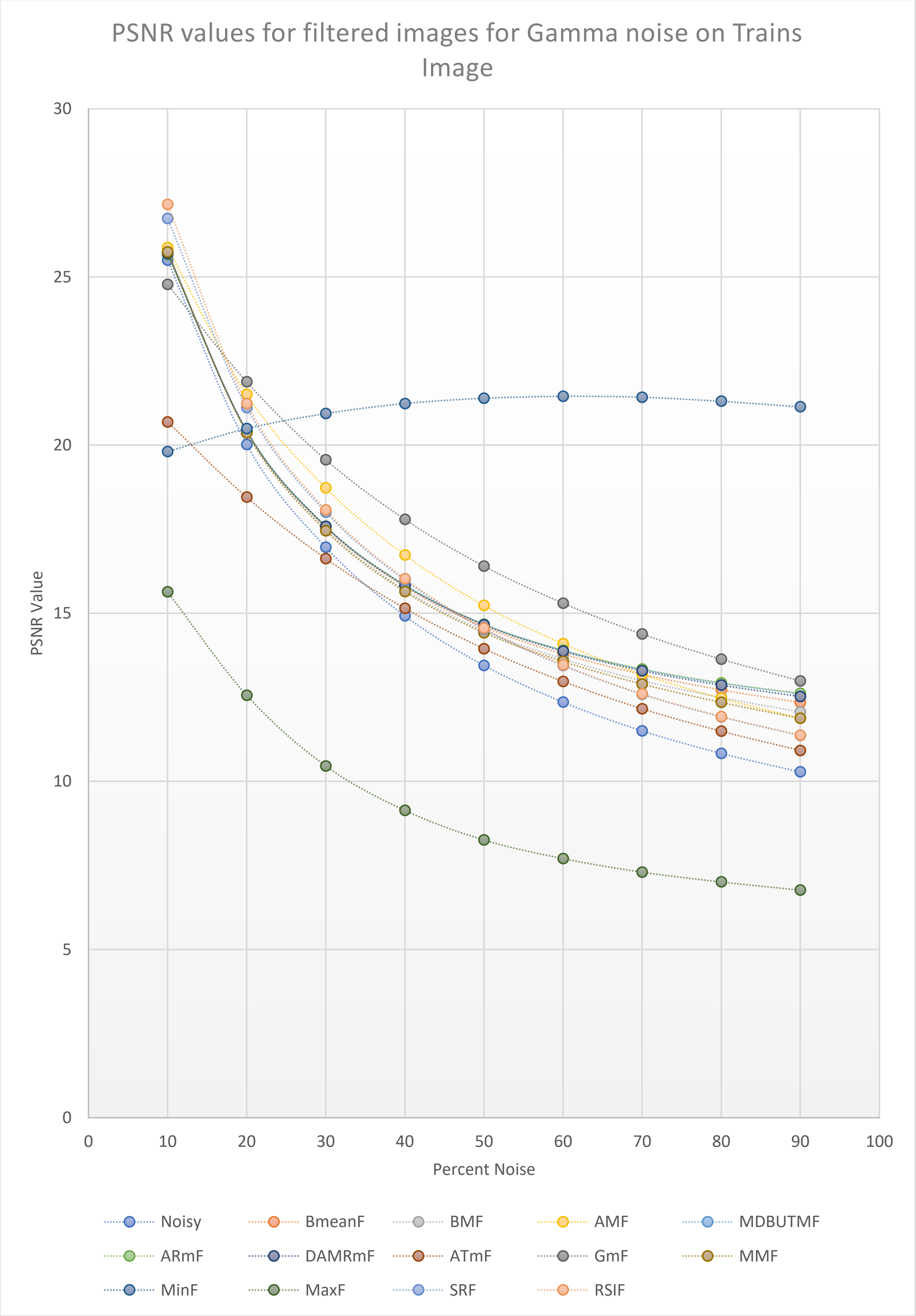
Noise: Gaussian (From 10% to 90%);



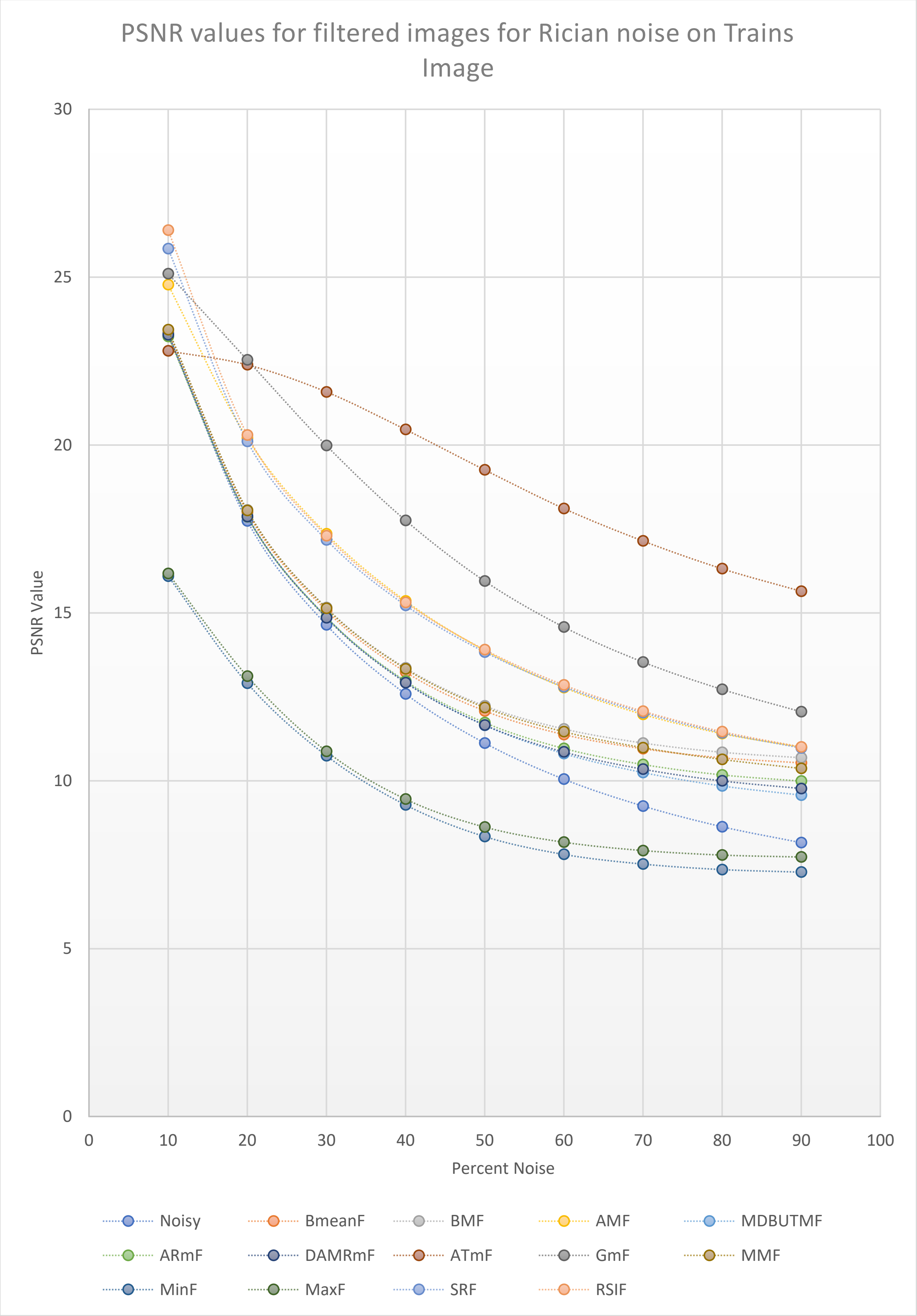
Noise: Speckle (From 10% to 90%);



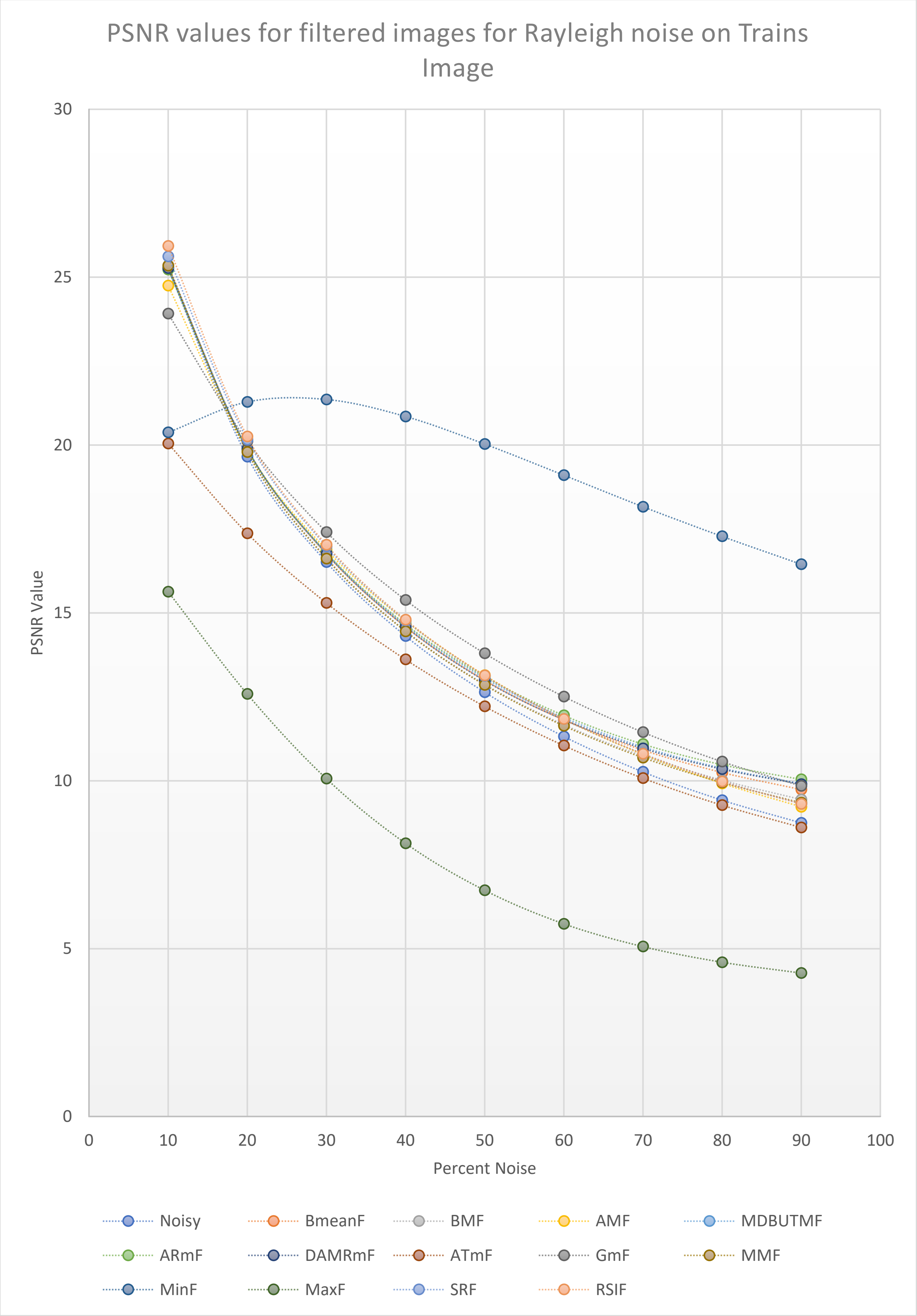
Noise: Gamma (From 10% to 90%);



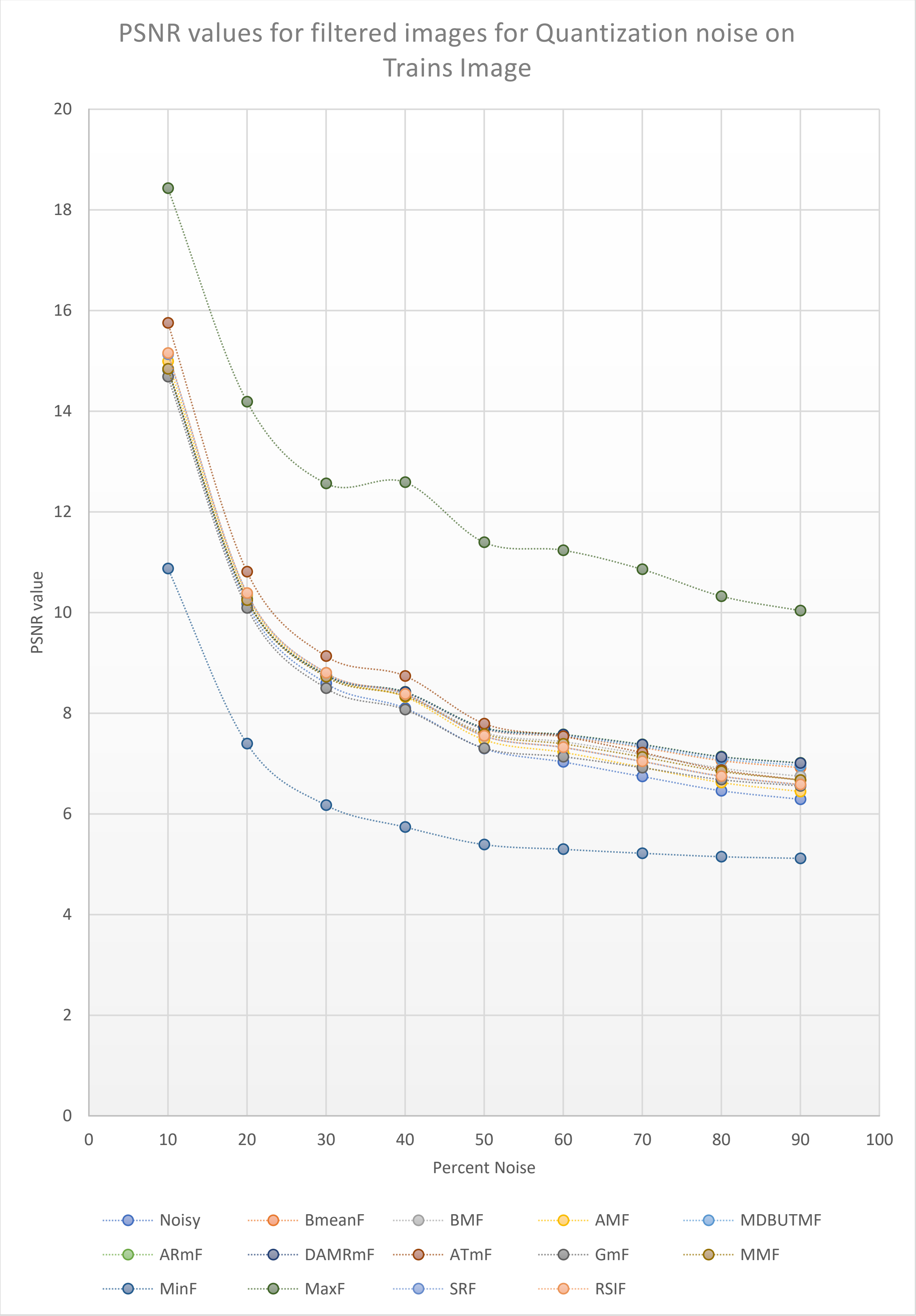
Noise: Rician (From 10% to 90%);



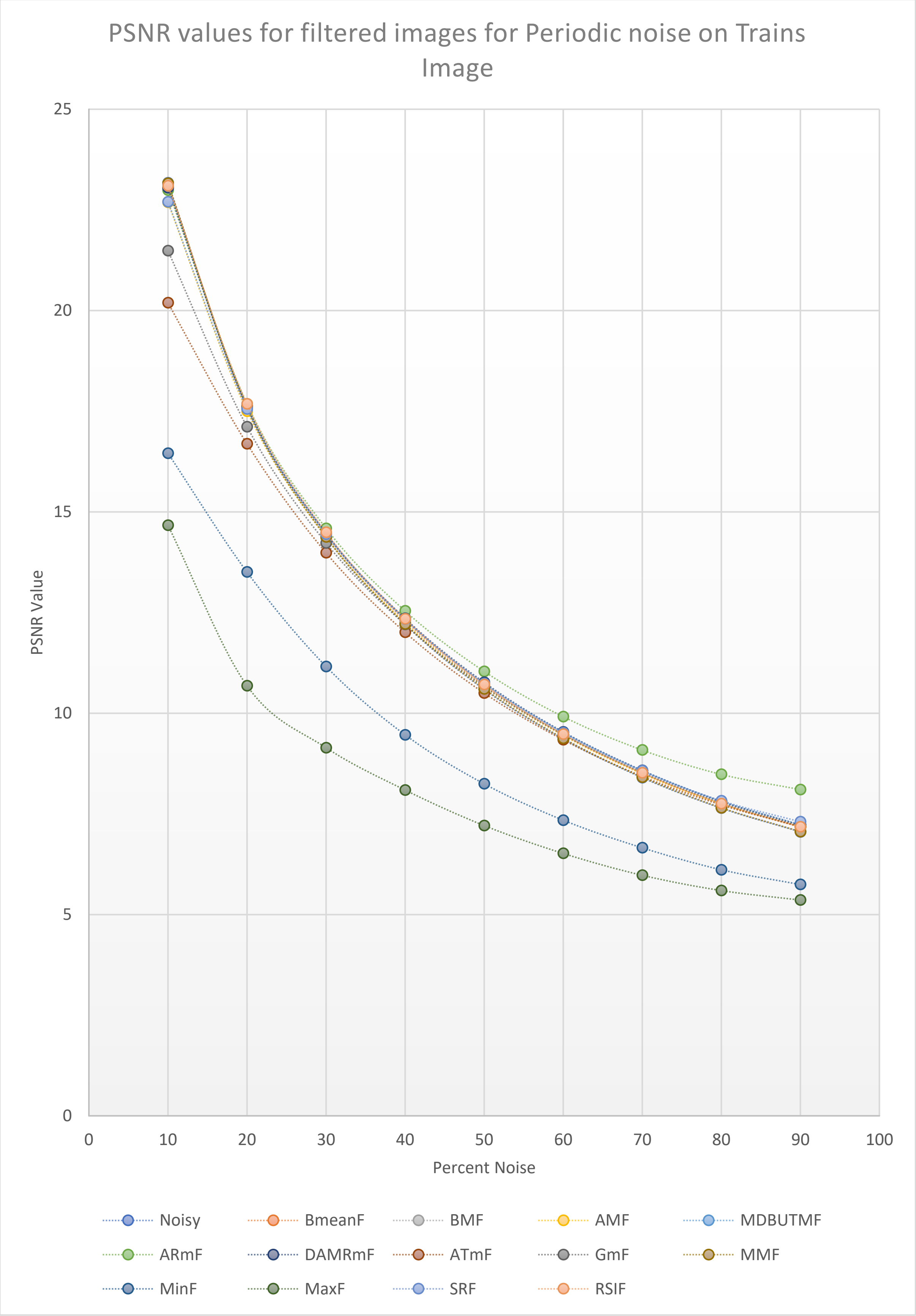
Noise: Rayleigh (From 10% to 90%):



Noise: Quantization (From 10% to 90%):



Noise: Periodic (From 10% to 90%):



CONCLUSION:

From the above shown graphs it can be concluded that most of the filters enhanced the PSNR value of the image.

Details for the specific noises are given below:

1. Impulse noise (salt and pepper):

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	14.72	31.43	34.26	30.79	34.02	33.14	34.27	21.59	24.92	33.80	18.84	18.97	17.37	17.47
20	11.71	26.90	30.60	29.04	31.30	30.71	31.94	19.77	22.96	28.84	19.03	19.18	14.50	14.55
30	9.95	23.63	27.95	27.36	29.34	28.93	30.25	18.02	20.74	23.82	19.25	19.43	12.84	12.86
40	8.70	20.98	25.35	25.82	27.64	27.45	28.80	16.45	18.41	19.44	19.44	19.70	11.64	11.65
50	7.74	18.67	22.61	24.37	25.87	26.12	27.45	15.06	16.09	15.66	19.40	19.79	10.73	10.73
60	6.94	16.61	19.66	23.00	23.65	24.91	26.16	13.82	13.83	12.59	18.63	19.05	10.00	10.01
70	6.28	14.73	16.50	21.58	20.59	23.76	24.81	12.74	11.59	10.08	16.31	16.50	9.39	9.40
80	5.70	12.95	13.14	20.00	16.91	22.57	23.29	11.76	9.36	7.97	12.74	12.59	8.88	8.89
90	5.19	11.24	9.33	17.13	13.12	21.28	21.35	10.87	7.18	6.21	8.84	8.44	8.43	8.46

- ❖ Adaptive Riesz Mean filter and Different Adaptive Modified Riesz Mean filter constantly performed well in all types of noise ranges for impulse noise.
- ❖ Modified Decision Based Unsymmetric Trimmed Median filter, Adaptive Median filter, Alpha trimmed mean filter and Basic Mean Filter failed to produce good PSNR values at high ranges of noise density, but performed well in all other noise values.
- ❖ Basic Median filter , Geometric mean filter, Modified Median filter, Min/ Max filter, Sector rotation filter and Recursive spline interpolation filter were able to produce high PSNR values only at low ranges of noise density.

Therefore, from this comparison it can be concluded that *Adaptive Riesz Mean filter and Different Adaptive Modified Riesz Mean filter* are the best filters to remove Impulse noises from a given image.

2. Gaussian noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	17.86	18.18	18.06	18.70	18.16	18.19	18.19	17.34	20.33	18.01	18.59	11.81	19.0	17.86
20	14.21	14.45	14.31	14.47	14.50	14.56	14.42	13.49	15.23	14.30	20.38	8.21	14.6	14.21
30	11.48	11.69	11.59	11.61	11.73	11.83	11.70	10.81	12.00	11.58	16.94	5.11	11.7	11.48
40	9.47	9.69	9.58	9.54	9.75	9.84	9.74	8.86	9.77	9.56	13.56	3.84	9.67	9.47
50	7.95	8.19	8.07	8.00	8.28	8.41	8.25	7.41	8.14	8.05	10.99	3.22	8.13	7.95
60	6.82	7.07	6.93	6.85	7.19	7.36	7.15	6.35	6.93	6.91	9.05	2.81	6.97	6.82
70	6.02	6.25	6.10	6.03	6.40	6.61	6.31	5.64	6.08	6.09	7.57	2.61	6.12	6.02
80	5.49	5.68	5.55	5.49	5.84	6.09	5.70	5.23	5.52	5.53	6.48	2.65	5.56	5.49
90	5.19	5.32	5.22	5.19	5.45	5.73	5.32	5.05	5.20	5.21	5.73	2.96	5.23	5.19

All of the filter designed are spatial filter hence they are not very effective against gaussian noise

From the PSNR table, it can be observed that none of the filters were able to enhance the image quality significantly and therefore it can be concluded that none of the designed filter is able to denoise the image having gaussian noise.

3. Speckle noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	15.97	15.92	16.05	17.54	15.31	15.46	15.43	22.05	20.50	16.05	10.89	13.87	18.0	15.97
20	13.21	13.03	13.22	14.80	12.30	12.48	12.43	20.44	17.02	13.25	8.23	12.43	15.3	13.21
30	11.58	11.34	11.57	13.17	10.53	10.72	10.67	19.02	14.17	11.60	6.58	11.54	13.7	11.58
40	10.50	11.02	11.25	12.25	10.11	10.36	10.28	17.93	13.38	11.27	6.31	10.90	12.8	10.50
50	9.85	11.09	11.30	11.89	10.09	10.41	10.29	17.22	13.73	11.24	6.51	10.41	12.2	9.85
60	9.41	11.12	11.30	11.68	10.07	10.45	10.29	16.69	13.98	11.14	6.70	10.01	11.8	9.41
70	9.08	11.11	11.26	11.49	10.01	10.43	10.24	16.28	14.12	10.98	6.84	9.66	11.5	9.08
80	8.84	11.12	11.24	11.37	10.00	10.46	10.24	16.00	14.29	10.84	6.99	9.39	11.3	8.84
90	8.63	11.11	11.19	11.25	9.96	10.46	10.20	15.75	14.37	10.68	7.12	9.13	11.2	8.63

- Alpha Trimmed mean filter and Geometric mean filter constantly performed well in all types of noise ranges for speckle noise.
- Basic Mean Filter, Adaptive Median filter, Basic Median filter Adaptive Riesz Mean filter, Different Adaptive Modified Riesz Mean filter, Sector rotation filter and Recursive spline interpolation filter failed to produce good PSNR values at high ranges of noise density, but performed relatively well in all other noise values.
- Modified Decision Based Unsymmetric Trimmed Median filter and Min/ Max filter were constantly unable to produce considerable PSNR values even at low noise density

Therefore, it can be concluded that, *Alpha Trimmed mean filter and Geometric mean filter* showed improved performance at high density noise and can be considered to remove high density speckle noises.

4. Gamma noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	25.50	25.71	25.75	25.88	25.72	25.66	25.69	20.69	24.78	25.75	19.81	15.64	26.7	25.50
20	20.02	20.42	20.40	21.51	20.43	20.41	20.42	18.45	21.88	20.37	20.49	12.56	21.1	20.02
30	16.96	17.56	17.50	18.72	17.58	17.58	17.58	16.63	19.56	17.45	20.94	10.46	18.0	16.96
40	14.92	15.79	15.70	16.73	15.83	15.83	15.83	15.15	17.79	15.65	21.24	9.14	15.9	14.92
50	13.45	14.60	14.48	15.23	14.67	14.67	14.65	13.95	16.40	14.42	21.39	8.26	14.5	13.45
60	12.36	13.79	13.64	14.09	13.89	13.90	13.87	12.97	15.30	13.57	21.45	7.70	13.4	12.36
70	11.50	13.18	13.00	13.19	13.33	13.34	13.29	12.16	14.38	12.89	21.42	7.30	12.5	11.50
80	10.83	12.72	12.49	12.47	12.92	12.93	12.86	11.49	13.64	12.35	21.30	7.01	11.9	10.83
90	10.28	12.35	12.07	11.88	12.60	12.62	12.52	10.92	12.99	11.87	21.14	6.77	11.3	10.28

- ❖ Minimum filter constantly performed well in all types of noise ranges for Gamma noise.
- ❖ Basic Mean Filter, Adaptive Median filter, Basic Median filter Adaptive Riesz Mean filter, Different Adaptive Modified Riesz Mean filter, Alpha trimmed mean filter, Sector rotation filter, Recursive spline interpolation filter and Modified Decision Based Unsymmetric Trimmed Median filter failed to produce good PSNR values at high ranges of noise density, but performed relatively well in all other noise values.
- ❖ Max filter was constantly unable to produce considerable PSNR values even at low noise density

From the comparison table it can be observed that only *<u>Minimum filter</u>* can denoise the image containing gamma noise and therefore is an ideal option for gamma noise denoising.

5. **Rician noise:**

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	23.31	23.33	23.43	24.77	23.30	23.23	23.28	22.80	25.10	23.43	16.10	16.17	25.84	26.40
20	17.74	17.99	18.06	20.20	17.89	17.86	17.87	22.39	22.54	18.05	12.90	13.11	20.10	20.29
30	14.64	15.07	15.16	17.35	14.87	14.87	14.85	21.58	19.98	15.14	10.75	10.88	17.17	17.29
40	12.59	13.24	13.36	15.35	12.94	12.97	12.92	20.46	17.76	13.33	9.28	9.45	15.22	15.31
50	11.12	12.08	12.23	13.88	11.65	11.73	11.66	19.26	15.95	12.18	8.34	8.62	13.83	13.90
60	10.05	11.38	11.54	12.78	10.81	10.96	10.86	18.11	14.58	11.46	7.81	8.17	12.79	12.85
70	9.25	10.95	11.12	11.97	10.24	10.48	10.34	17.14	13.54	10.99	7.52	7.92	12.02	12.07
80	8.63	10.68	10.85	11.40	9.84	10.17	10.00	16.31	12.72	10.63	7.35	7.79	11.42	11.46
90	8.16	10.53	10.68	11.00	9.57	9.99	9.77	15.64	12.05	10.37	7.28	7.73	10.97	11.00

- Alpha trimmed mean filter and geometric mean filter constantly performed relatively well in all types of noise ranges for Rician noise.
- Basic Mean Filter, Adaptive Median filter, Basic Median filter Adaptive Riesz Mean filter, Different Adaptive Modified Riesz Mean filter, Alpha trimmed mean filter, Sector rotation filter, Recursive spline interpolation filter and Modified Decision Based Unsymmetric Trimmed Median filter failed to produce good PSNR values at high ranges of noise density, but performed relatively well in all other noise values.
- Min /Max filter was constantly unable to produce considerable PSNR values even at low noise density

Alpha trimmed mean filter relatively denoised the image. *Geometric mean filter* was also able to recover the image to some extent. The rest of the filters did not work.

6. Rayleigh noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	25.26	25.32	25.35	24.75	25.30	25.23	25.27	20.05	23.91	25.34	20.37	15.63	25.62	25.92
20	19.65	19.89	19.82	19.84	19.89	19.87	19.88	17.37	20.12	19.79	21.28	12.58	20.15	20.25
30	16.51	16.75	16.64	16.88	16.79	16.80	16.76	15.29	17.41	16.61	21.36	10.06	16.98	17.02
40	14.31	14.58	14.46	14.76	14.63	14.66	14.58	13.61	15.38	14.45	20.85	8.14	14.77	14.79
50	12.64	12.99	12.87	13.14	13.05	13.10	12.98	12.21	13.79	12.85	20.03	6.74	13.12	13.13
60	11.32	11.81	11.67	11.84	11.88	11.94	11.82	11.05	12.51	11.64	19.10	5.74	11.83	11.83
70	10.26	10.91	10.74	10.78	11.01	11.09	10.96	10.07	11.44	10.69	18.16	5.06	10.80	10.80
80	9.42	10.25	10.02	9.92	10.37	10.48	10.35	9.27	10.57	9.95	17.28	4.59	9.97	9.97
90	8.75	9.74	9.44	9.23	9.91	10.04	9.90	8.61	9.84	9.34	16.45	4.27	9.31	9.31

All of the filter designed are spatial filter hence they are not very effective against Rayleigh noise

From the PSNR table, it can be observed that none of the filters were able to enhance the image quality and most of the even further deteriorated the image quality and therefore it can be concluded that none of the designed filter is able to denoise the image having Rayleigh noise.

7. Quantization noise

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	14.82	14.83	14.84	14.99	14.83	14.83	14.83	15.75	14.68	14.84	10.87	18.43	15.13	15.15
20	10.18	10.26	10.25	10.34	10.25	10.26	10.26	10.81	10.09	10.25	7.39	14.19	10.38	10.38
30	8.59	8.75	8.73	8.76	8.75	8.76	8.76	9.13	8.50	8.72	6.17	12.56	8.79	8.80
40	8.10	8.41	8.36	8.32	8.41	8.42	8.42	8.74	8.07	8.34	5.74	12.59	8.38	8.38
50	7.30	7.68	7.61	7.47	7.69	7.71	7.71	7.79	7.31	7.58	5.39	11.39	7.54	7.54
60	7.03	7.53	7.43	7.22	7.55	7.58	7.57	7.54	7.14	7.39	5.30	11.24	7.31	7.32
70	6.74	7.32	7.19	6.93	7.34	7.38	7.37	7.22	6.92	7.13	5.22	10.86	7.04	7.04
80	6.46	7.06	6.91	6.62	7.09	7.14	7.13	6.87	6.68	6.84	5.15	10.32	6.74	6.74
90	6.29	6.92	6.75	6.44	6.96	7.01	7.01	6.67	6.56	6.67	5.12	10.04	6.57	6.58

- ❖ Maximum filter was able to enhance the image quality up to some extent.
- ❖ All the other filter were unable to enhance the image quality rather most of them even further deteriorated the image.

From the PSNR table, it can be observed that only Maximum filters was able to enhance the image quality and rest of the filter deteriorated the image quality and therefore it can be concluded that only *‘Maximum filter’* of the designed filter can be used to denoise the image having Rayleigh noise.

8. Periodic noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	23.17	23.13	23.16	22.69	23.06	22.99	23.05	20.19	21.49	23.16	16.46	14.67	22.70	23.09
20	17.58	17.63	17.59	17.50	17.61	17.68	17.62	16.69	17.11	17.59	13.51	10.68	17.55	17.68
30	14.36	14.45	14.38	14.37	14.45	14.59	14.47	13.98	14.23	14.38	11.16	9.14	14.44	14.49
40	12.20	12.30	12.22	12.25	12.32	12.54	12.35	12.01	12.20	12.22	9.46	8.09	12.33	12.34
50	10.59	10.69	10.61	10.66	10.74	11.04	10.76	10.50	10.66	10.60	8.25	7.21	10.73	10.71
60	9.36	9.47	9.38	9.45	9.52	9.91	9.54	9.34	9.47	9.37	7.34	6.52	9.51	9.47
70	8.40	8.51	8.41	8.49	8.56	9.08	8.58	8.43	8.53	8.41	6.66	5.97	8.57	8.51
80	7.64	7.75	7.65	7.74	7.81	8.48	7.81	7.72	7.77	7.65	6.11	5.60	7.82	7.75
90	7.05	7.17	7.06	7.18	7.23	8.10	7.22	7.18	7.19	7.06	5.75	5.36	7.30	7.17

From the PSNR table, it can observed that none of the filters were able to enhance the image quality and a few of the even further deteriorated the image quality and therefore can be conclude that none of the designed filter is able to denoise the image having Periodic noise.

9. Poisson noise:

noise %	Noisy	BmeanF	BMF	AMF	MDBUTMF	ARmF	DAMRmF	ATmF	GmF	MMF	MinF	MaxF	SRF	RSIF
10	27.11	26.99	27.17	27.48	26.95	26.80	26.90	22.94	25.79	27.17	17.32	17.46	29.17	30.68

- The addition of poisson noise in the image nearly had no impact on the image quality as the PSNR of noisy image is about 27.11.
- Min/Max filter degraded the image quality significantly a deteriorated image.
- All the other filters gave a PSNR value nearly equal to the noisy PSNR value.
- Recursive spline interpolation filter and sector rotation filter marginally enhanced the PSNR value

Therefore *Sector Rotation filter and Recursive spline interpolation filter* can be used to denoise the image having Poisson noise.

REFERENCE: -

- Yuan-Bin Wang, Hai-Long Huang 2021 Image Denoising Based on Adaptive Sector Rotation Median Filter
- S. Zhang, X. Li and C. Zhang, "Modified Adaptive Median Filtering," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 2018, pp. 262-265.
- Enginoğlu, Serdar & Erkan, Uğur & Memiş, Samet. (2019). Pixel Similarity-Based Adaptive Riesz Mean Filter for Salt-and-Pepper Noise Removal. Multimedia Tools and Applications. 78. 35401-35418. 10.1007/s11042-019-08110-1.
- Memiş, S. and Erkan, U., 2021. Different adaptive modified Riesz mean filter for high-density salt-and-pepper noise removal in grayscale images. Avrupa Bilim ve Teknoloji Dergisi, (23), pp.359-367.
- R. Oten and R. J. P. de Figueiredo, "Alpha-trimmed mean filters under deviations from assumed noise model," in IEEE Transactions on Image Processing, vol. 13, no. 5, pp. 627-639, May 2004.
- B. Navya, J. Sridevi and K. Vasanth, "Geometric Mean Filter" 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 1095-1102.
- IMAGE DENOISING USING NEW ADAPTIVE BASED MEDIAN FILTER Suman Shrestha1, 2 1University of Massachusetts Medical School, Worcester, MA 01655 2 Department of Electrical and Computer Engineering, The University of Akron, Akron, OH 44325