Assignment: 3

Mathematical Modelling in Industry

Name:	**Lokendra Kumar**
Roll No:	MA23M008
Submitted to:	Dr. Sundar S
Date of Sub:	30/09/2023

1 Assignment

Q.1 PFA the codes for implementing PMC model. Understand it completely and use the provided code to implement EED model.

Sol. For Edge Enhancing Diffusion We need to change only pmc.m function file. In this code function first i will define Gradient with the help of gD.m function and then for EED diffusion Tensor we need two orthonormal eigen vectors are constructed with one parallel to the gradient and the other perpendicular to the gradient. Now i am defining the corresponding eigen-values by the following choice ($m \in N, C_m > 0, \lambda > 0$)

$$\lambda_1(\mu_1) := g(\mu_1)$$
$$\lambda_2 := 1$$

with

$$g(s) = \begin{cases} 1 - exp(\frac{-C_m}{(s/\lambda)^m}) & \text{if } s > 0 \\ 1 & \text{if } s \leq 0. \end{cases}$$

```
function f = eed(f, timeStep, nosteps, verbose, W, m, K, cm)
% Edge Enhancing Diffusion
% f: The input image.
% timeStep: Time step size for the diffusion process.
% nosteps: Number of diffusion iterations to perform.
% verbose: A flag indicating whether to display intermediate results.
% W: A parameter sigma used for the Gaussian Kernel.
% m ,K, cm: A parameter used to calculate eigenvalue lambda 1.

if verbose
figure(verbose);
subplot(1, 2, 1);
imshow(f);
title('Original Image');
drawnow;
end
% Initialize Dimensions
[n, m] = size(f);
%nosteps=n*m;

for iter = 1:nosteps %Diffusion Loop

% g=gD(f,0.001,0,0); % Apply this for Catte et al model
% gx=gD(f,1,1,0);
% gy=gD(f,1,0,1);

%Gradient Calculation
gx=gD(f, W, 1, 0);
gy=gD(f, W, 0, 1);
g = gx+gy* 1i;
gu = g * 1i;

% Initialization of Matrices
d11 = zeros(n, m);
d12 = zeros(n, m);
d22 = zeros(n, m);

```

```
    % EED diffusion Tensor Calculation
    for j = 1:n
    for k = 1:m
    g1 = [real(g(j,k)); imag(g(j,k))];
    g2 = [real(gu(j,k)); imag(gu(j,k))];

    v1 = g1 / norm(g1);
    v2 = g2 / norm(g1);

    s = norm(g1)^2;
    lambda1 = 1 - exp(-cm ./ (s/K).^m);
    lambda2 = 1;

    d = [v1, v2] * diag([lambda1, lambda2]) * [v1'; v2'];

    d11(j,k) = d(1,1);
    d12(j,k) = d(1,2);
    d22(j,k) = d(2,2);

    end
    end
    % With  Weickert standard explicit scheme
    % f=f+stepsize*snldStep(f,c,W,ip);

    % Diffusion Step
    div_flux = tnldStep(f, d11, d12, d22, 1);
    f = f + timeStep * div_flux;

    % Display Updated Image
    if verbose
    figure(verbose);
    subplot(1,2,2);
    imshow(f);
    title('Edge␣Enhancing␣Diffusion');
    drawnow;
    end
    end
```

Listing 1: eed.m Matlab file function

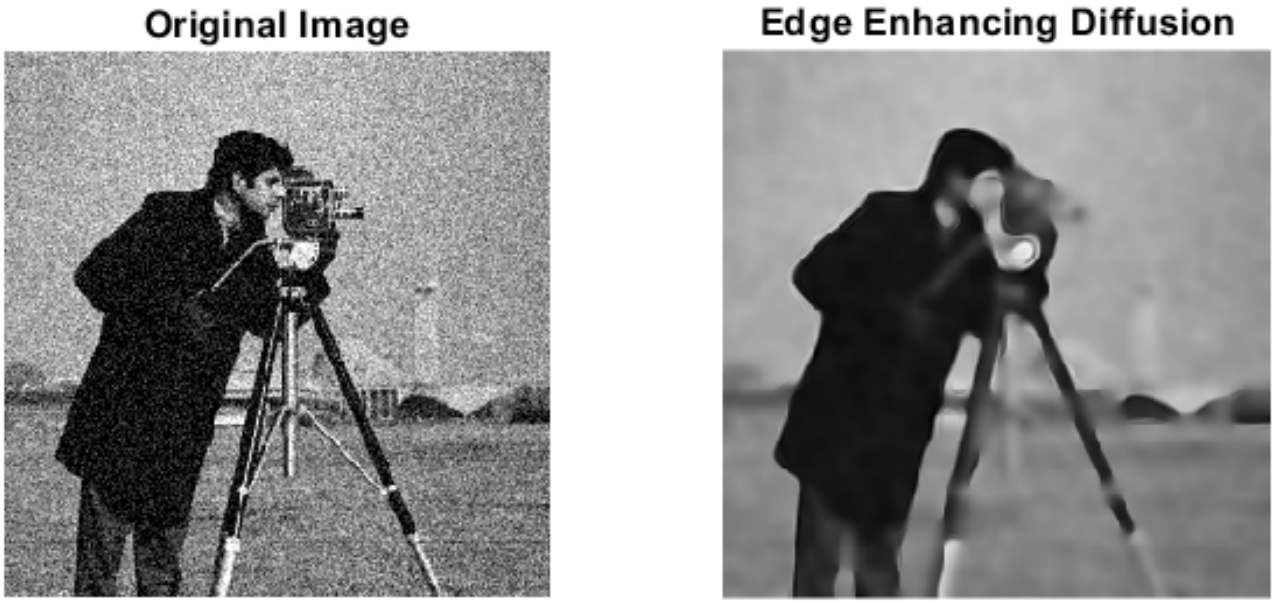
For above code eedtest.m file is following.

```
clc
clear all

a=imread('cameraman.tif'); % reading the image
%a=imread("tomo.jpg");
%a=imread("triangle.jpg");
a=im2double(a); % normalizing the instensity values to lie between o
   and 1

ref=a;
ad=imnoise(a,'gaussia',0.01); % adding Gaussian noise of mean zero
   and variance 0.01
```

```
%ad = imnoise(ref, 'poisson');
%ad = imnoise(ref, 'speckle');
%ad = imnoise(ref, 'salt & pepper');

Niter=20;
alpha=2.7; % Used in Numerical approximation
w= exp(4*alpha/9); % Used in Numerical approximation
k=0.001;
timestep = 0.2; % timestep size used in numerical approximation

[psnr, mse]=psnr_mse(ad, ref)
variance=w

b = eed(ad, timestep, Niter, 1, w, 3, k, 5.20);
% first argument is the noisy image, 2 is the reference image, 3 is
   the
% lambda value, 4 is the timestep size, 5 is the no of iterations, 6
   is the
% value to show the plot, 7 is the w value used in numerical
   approximation
% and last argument corresponding to choice of the numerical scheme.
```

Listing 2: eedtest.m Matlab file function

For above code output image is,



Q.2 Take a standard image(for example, cameraman image), perform Linear diffusion, PMC, EED and compare the respective PSNR values.

Sol. First we are calculating MSE and PSNR using the formula,

$$MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N}.$$

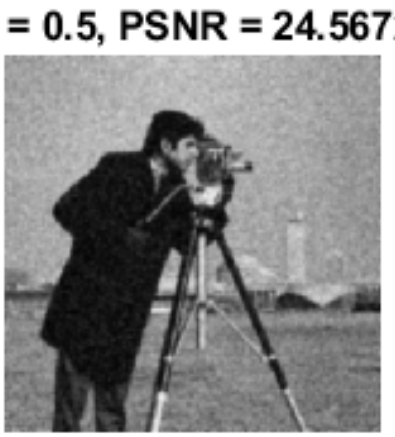
In the previous equation, M and N are the number of rows and columns in the input images, respectively. And

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right).$$

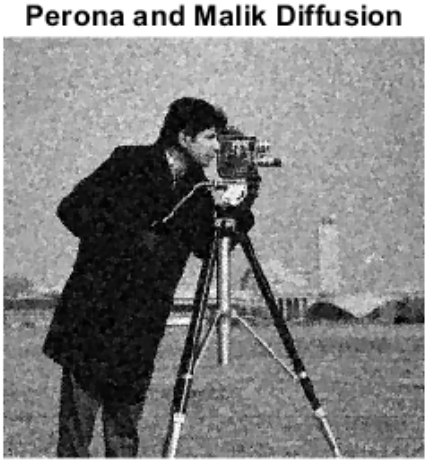
```
function [psnr, mse] = psnr_mse(image, image_ref)

[m, n] = size(image);
r=max (image, [],'all');
mse1 = 0;
for i = 1:m
for j = 1:n
mse1 = mean((double(image(:)) - double(image_ref(:))).^2);
end
end
mse = mse1;
psnr = 10 * log10(double(r) / double(mse));
end
```

Listing 3: psnr-mse.m Matlab file function

Now for Linear diffusion we have following results and output images. PSNR value:- 20.2772



Now for PMC diffusion we have following results and output images. PSNR value:- 20.3219



Now for EED diffusion we have following results and output images. PSNR value:- 20.2899

Q.3 Take 3 or 4 different images, add different noises to them(Gaussian noise, Salt and Pepper noise, speckle noise and poisson noise with different variances), treat each case as different input image, apply LD, PMC and EED, report respective PSNR values. For better interpretation of your outputs, compare the three models on variance vs PSNR plots for each noise and each image.

Sol. I am taking 3 different images, and adding 4 different noises(Gaussian noise, Salt and Pepper noise, speckle noise and poisson noise) to them. Now for different values of variances we have following images, variances a nd PSNR values.

Now for Linear diffusion we have following results and output images for (Gaussian noise, Salt and Pepper noise, speckle noise and poisson noise).

Image name	Noise	Variance	PSNR
cameraman	Gaussian	0.5	24.578
cameraman	Salt and Pepper	1.0	22.175
cameraman	speckle	1.5	20.775
cameraman	poisson	2.0	19.784
triangle	Gaussian	0.5	27.271
triangle	Salt and Pepper	1.0	28.8
triangle	speckle	1.5	26.94
triangle	poisson	2.0	25.7
tomo	Gaussian	0.5	26.481
tomo	Salt and Pepper	1.0	23.543
tomo	speckle	1.5	21.47
tomo	poisson	2.0	20.169

Listing 4: psnr-mse.m Matlab file function

Noisy Image

σ = 0.5, PSNR = 24.578, MSE = 223.069

Noisy Image

σ = 0.5, PSNR = 25.8309, MSE = 167.1674

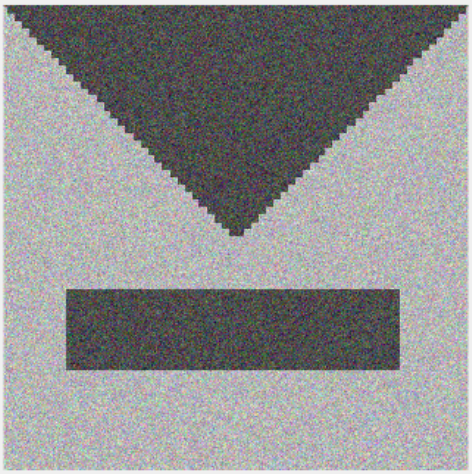
Noisy Image

σ = 1.5, PSNR = 20.7753, MSE = 535.4341

Noisy Image

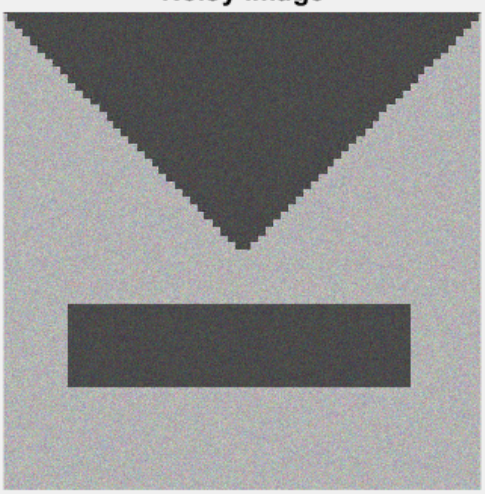
σ = 2, PSNR = 19.7839, MSE = 672.7484

Noisy Image



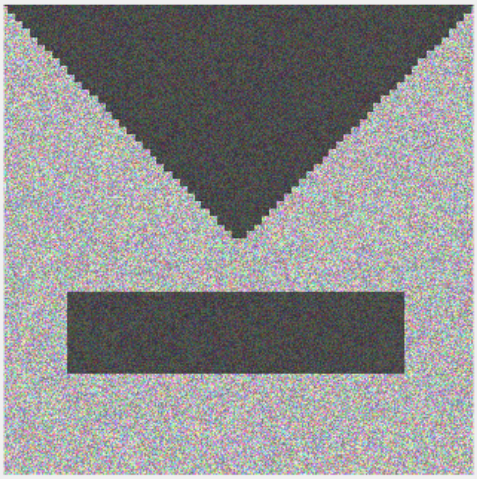
σ = 0.5, PSNR = 27.2706, MSE = 78.7853

Noisy Image



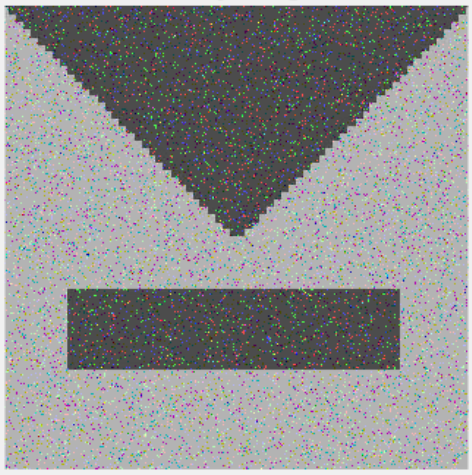
σ = 1, PSNR = 28.7999, MSE = 55.4009

Noisy Image



σ = 1.5, PSNR = 26.9399, MSE = 85.0201

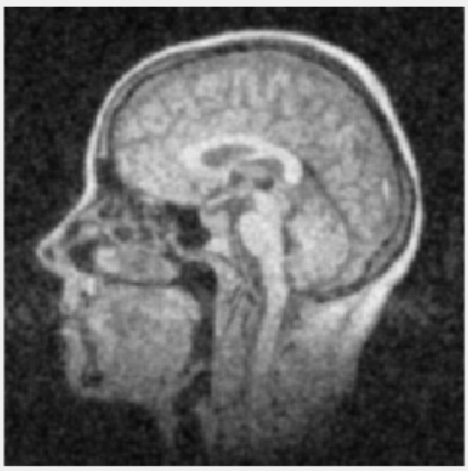
Noisy Image



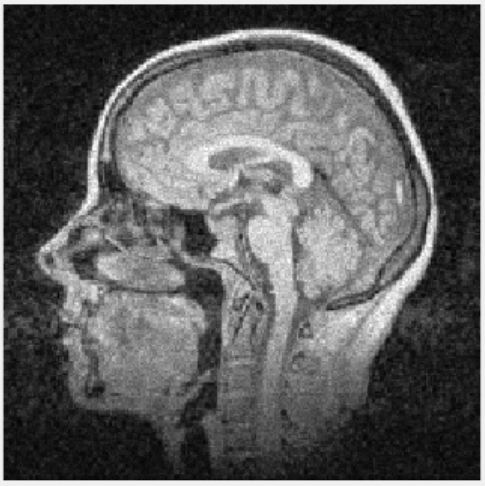
σ = 2, PSNR = 25.6996, MSE = 113.1209

Noisy Image

σ = 0.5, PSNR = 26.4811, MSE = 146.2089

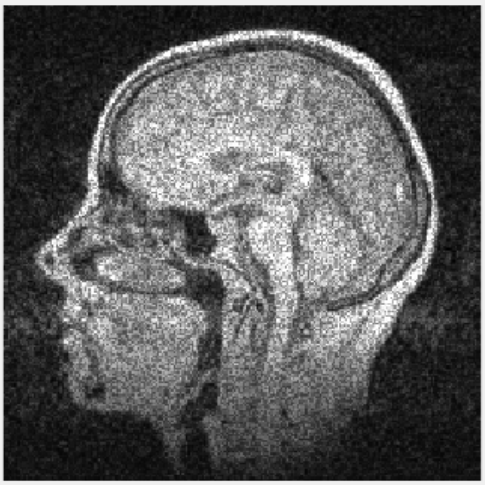


Noisy Image



σ = 1, PSNR = 23.5434, MSE = 287.5651

Noisy Image

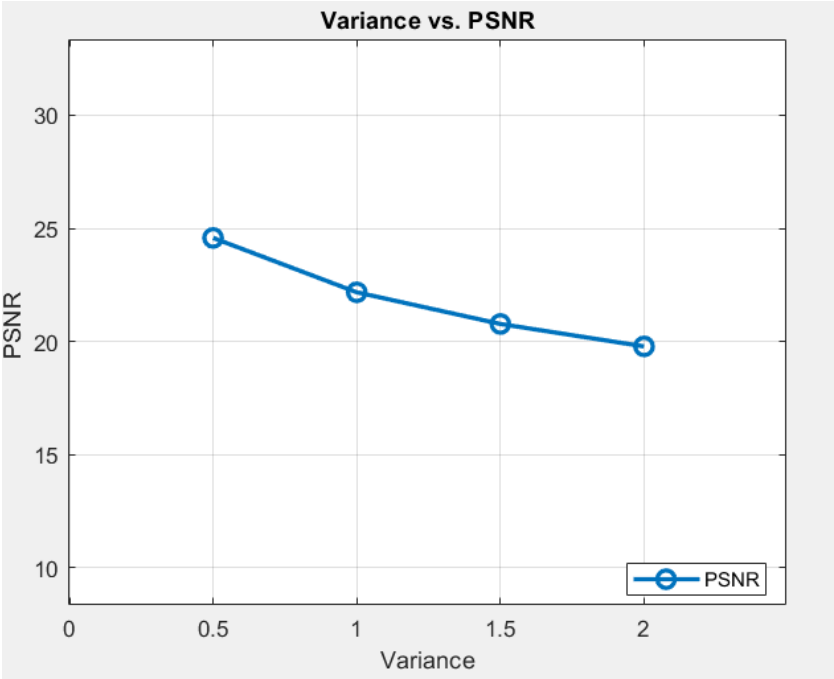


σ = 1.5, PSNR = 21.47, MSE = 463.5276

Noisy Image

σ = 2, PSNR = 20.1694, MSE = 625.3776

These is plot between variance and PSNR values for Linear diffusion method.



Now for PMC diffusion we have following results and output images for (Gaussian noise, Salt and Pepper noise, speckle noise and poisson noise).

Image name	Noise	Variance	PSNR
cameraman	Gaussian	0.5	20.3154
cameraman	Salt and Pepper	1.0	123.2395
cameraman	speckle	1.5	18.6432
cameraman	poisson	2.0	17.9337
triangle	Gaussian	0.5	19.9854
triangle	Salt and Pepper	1.0	121.6690
triangle	speckle	1.5	17.7167
triangle	poisson	2.0	18.3876
tomo	Gaussian	0.5	20.3722
tomo	Salt and Pepper	1.0	124.6422
tomo	speckle	1.5	20.6167
tomo	poisson	2.0	17.8687

Listing 5: psnr-mse.m Matlab file function

Original Image

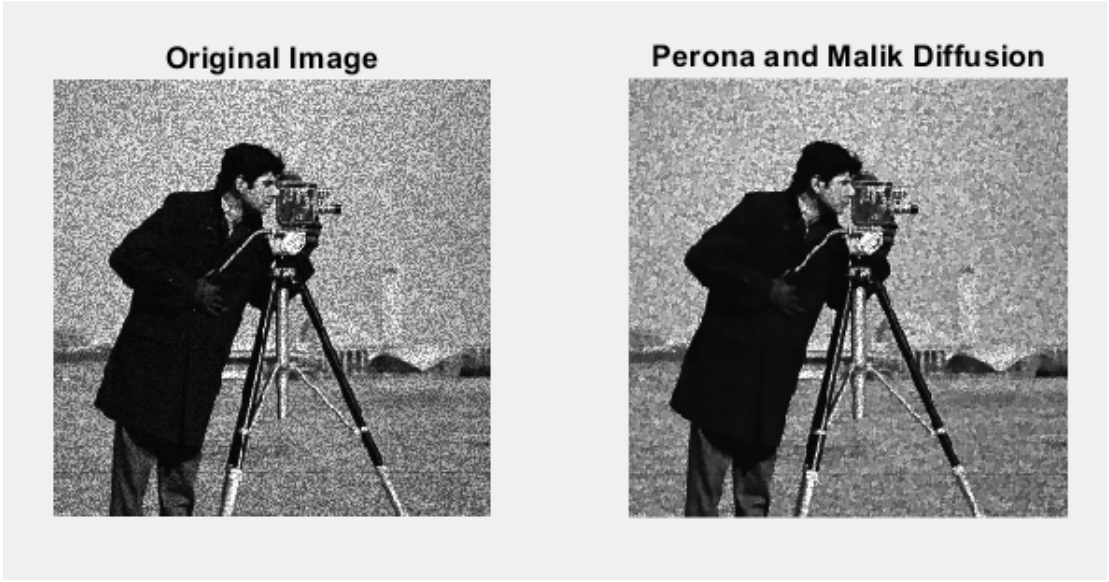
Perona and Malik Diffusion

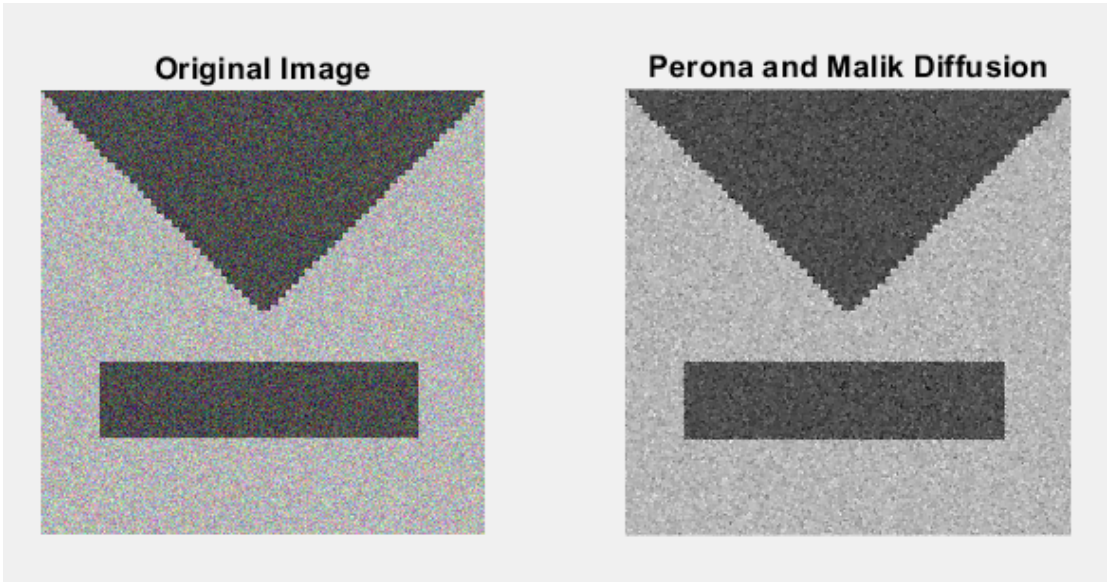
Original Image

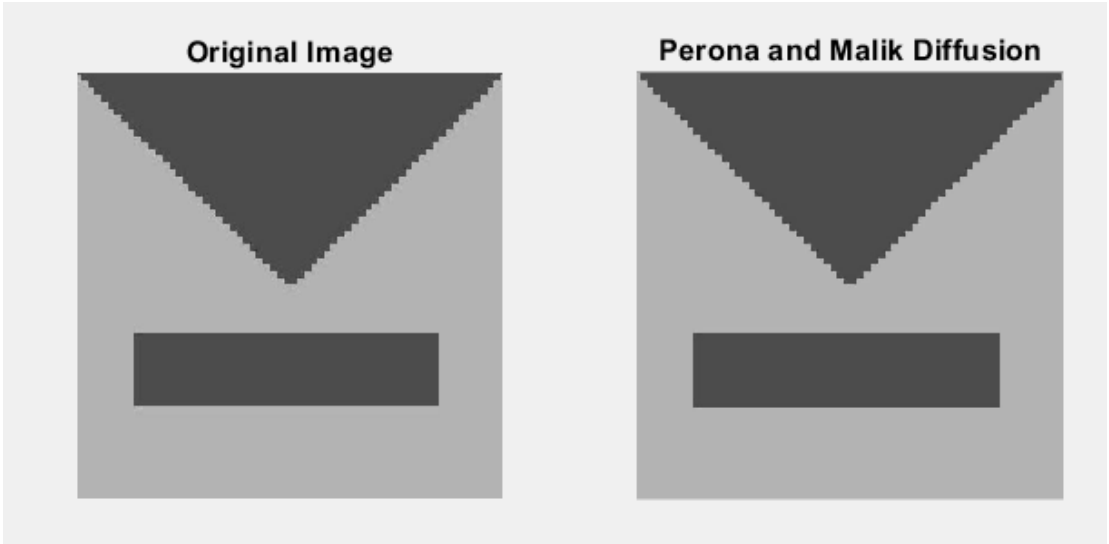
Perona and Malik Diffusion

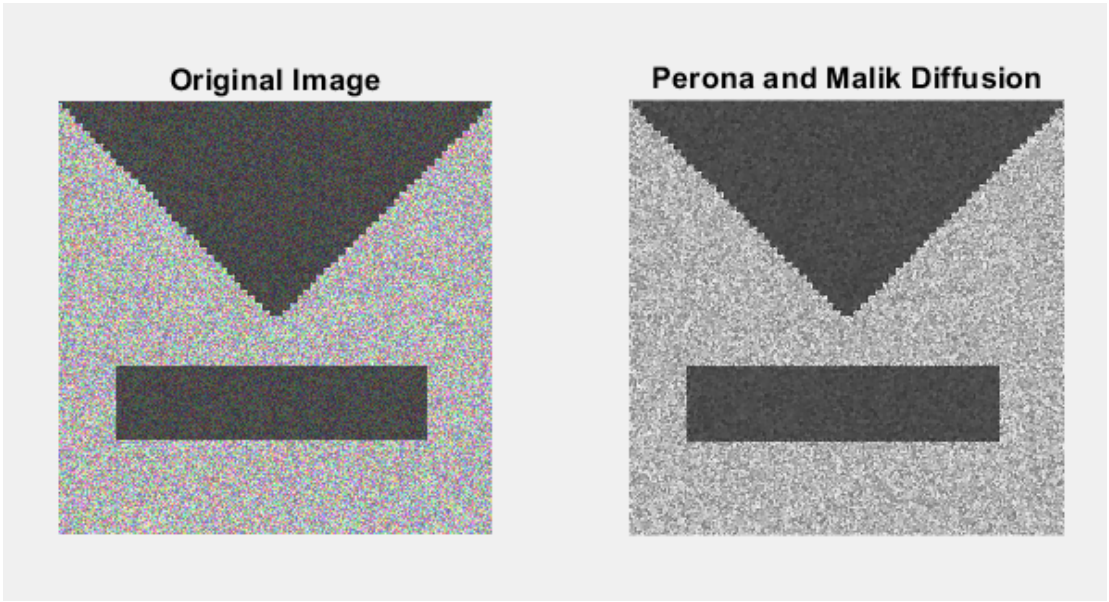
Original Image

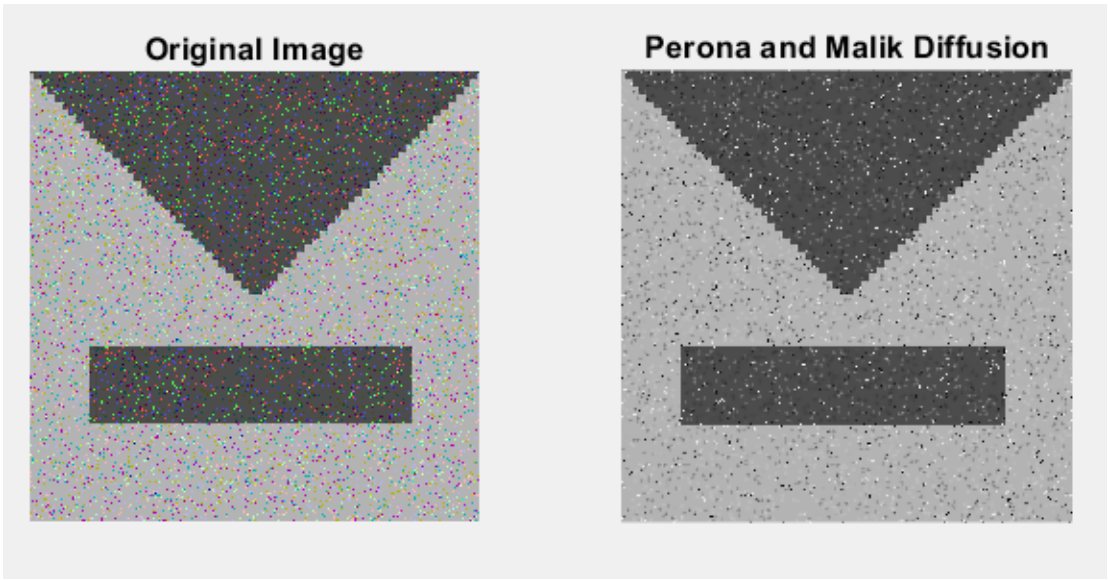
Perona and Malik Diffusion

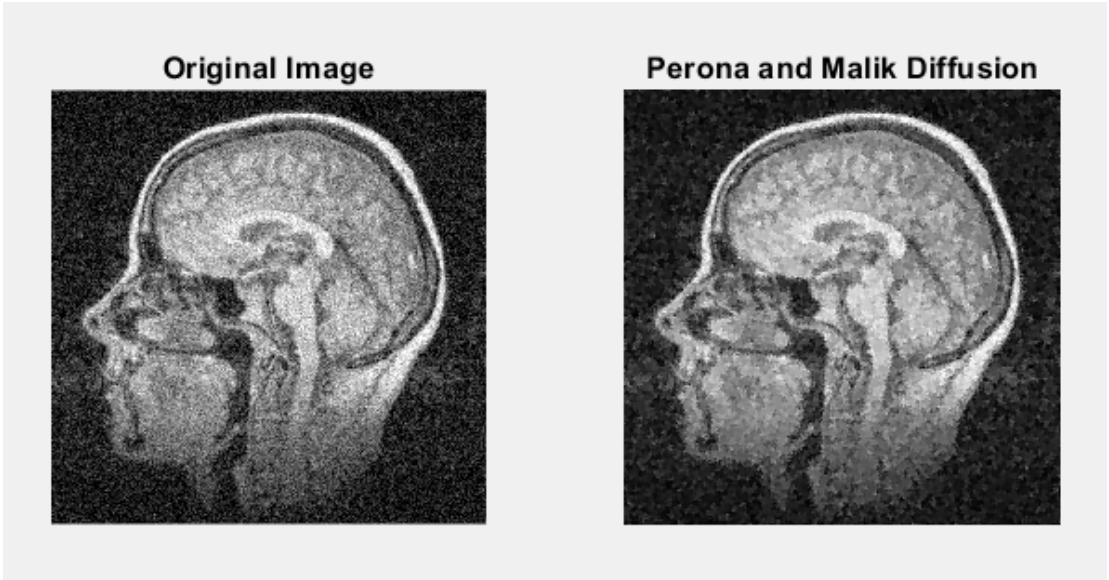

Original Image
Perona and Malik Diffusion

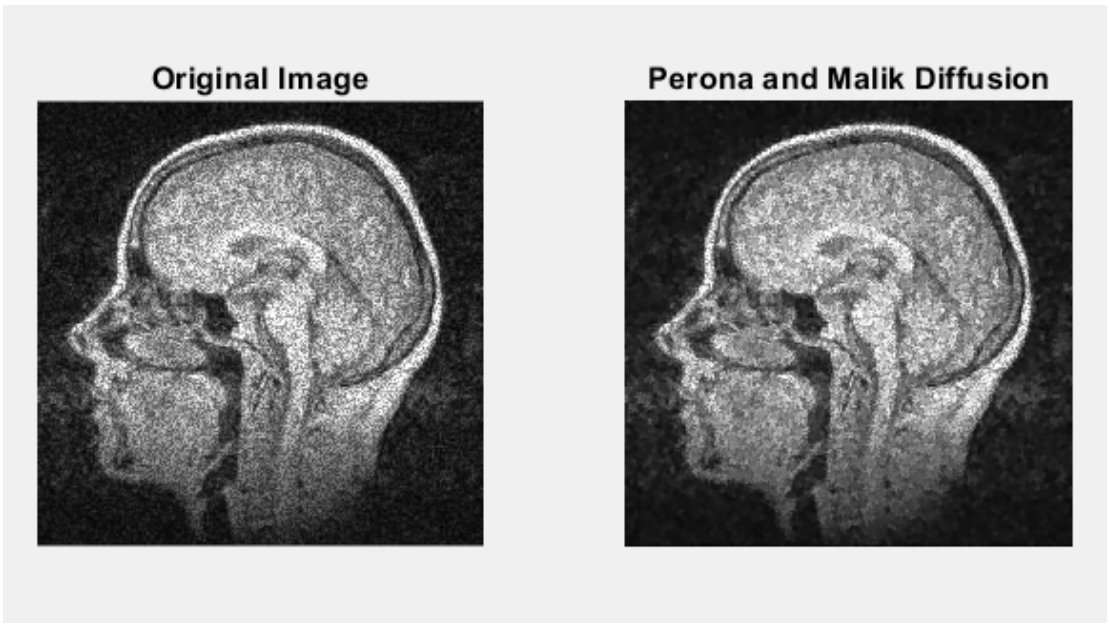

Original Image
Perona and Malik Diffusion

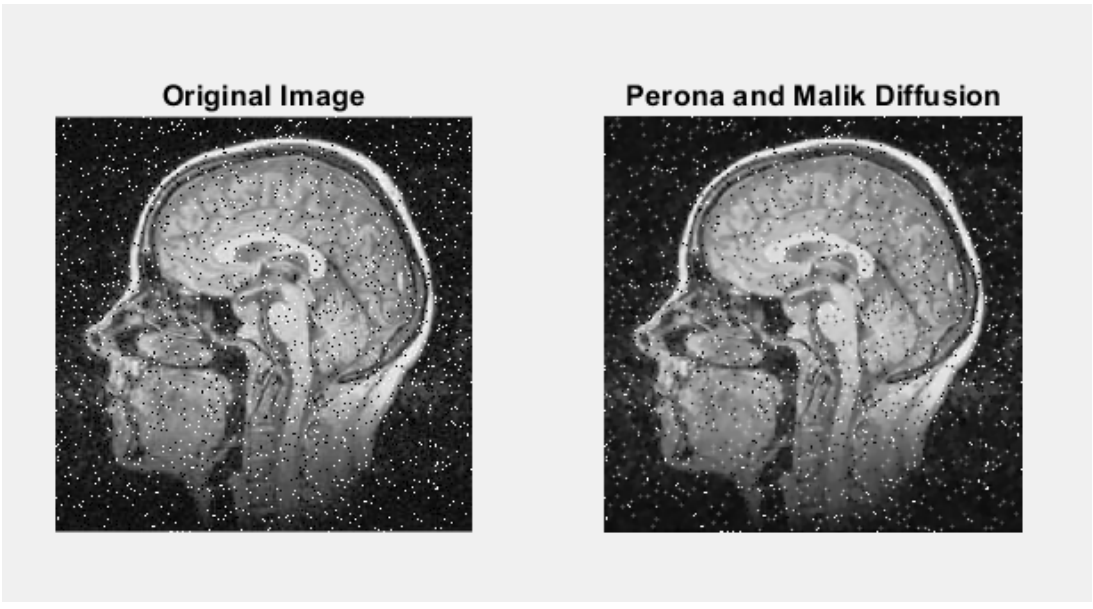

Original Image
Perona and Malik Diffusion


Original Image
Perona and Malik Diffusion

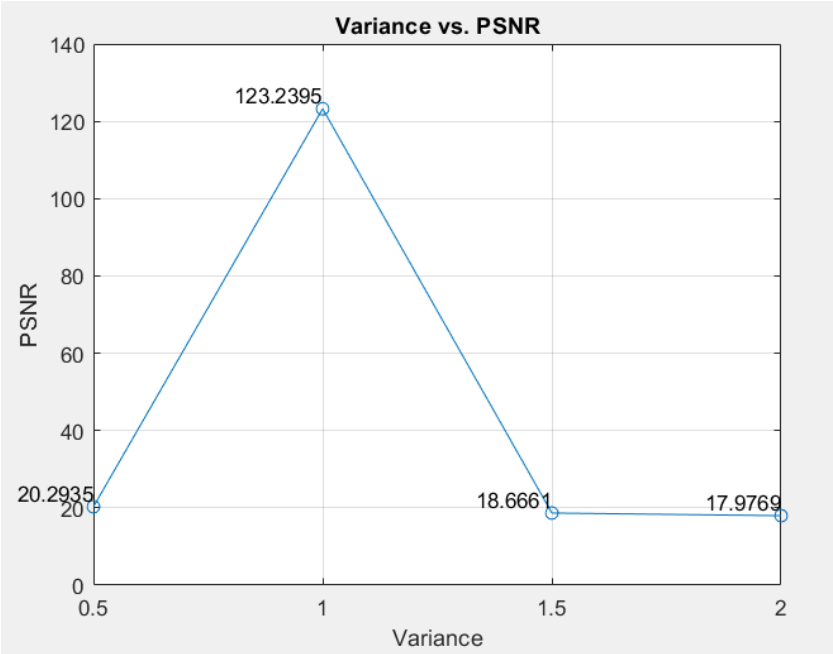

Original Image
Perona and Malik Diffusion


Original Image
Perona and Malik Diffusion





These is plot between variance and PSNR values for PMC method.



Now for EED diffusion we have following results and output images for (Gaussian noise, Salt and Pepper noise, speckle noise and poisson noise).

Image name	Noise	Variance	PSNR
cameraman	Gaussian	0.5	20.2935
cameraman	Salt and Pepper	1.0	123.2395
cameraman	speckle	1.5	18.6661
cameraman	poisson	2.0	17.9769
triangle	Gaussian	0.5	19.9542
triangle	Salt and Pepper	1.0	121.6728
triangle	speckle	1.5	17.7191
triangle	poisson	2.0	18.4082
tomo	Gaussian	0.5	20.3722
tomo	Salt and Pepper	1.0	124.6105
tomo	speckle	1.5	20.6167
tomo	poisson	2.0	17.8153

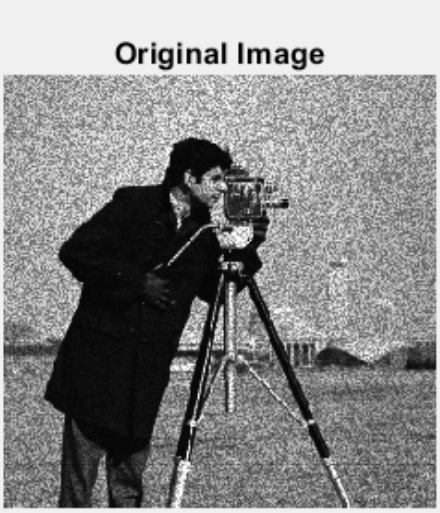
Listing 6: psnr-mse.m Matlab file function


Original Image


Edge Enhancing Diffusion


Original Image


Edge Enhancing Diffusion

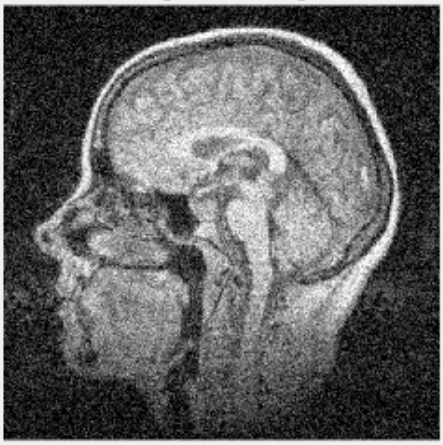

Original Image


Edge Enhancing Diffusion

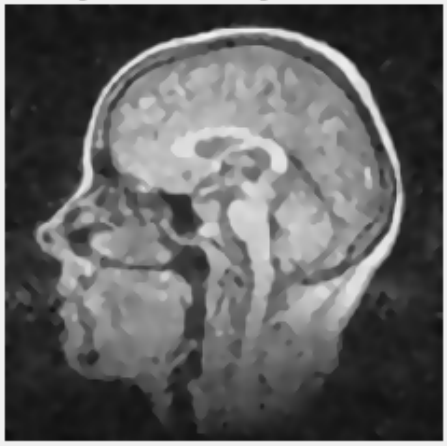
Original Image

Edge Enhancing Diffusion

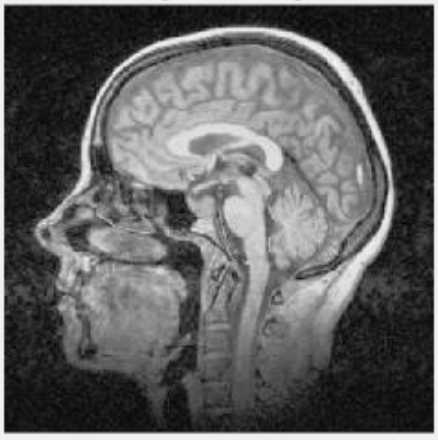
Original Image



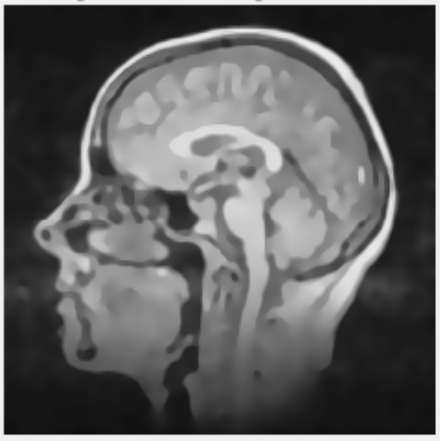
Edge Enhancing Diffusion

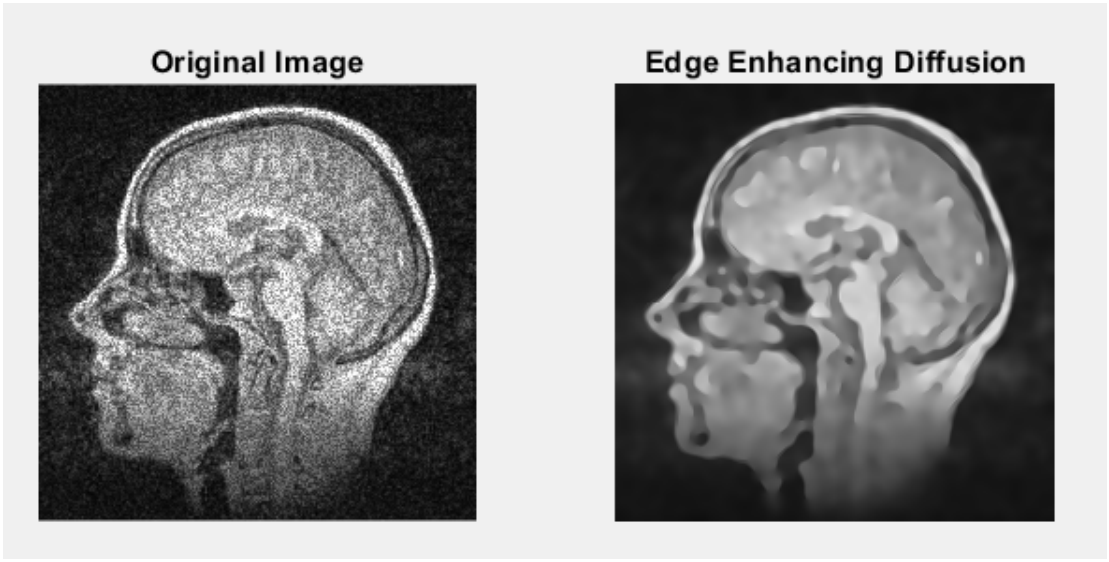


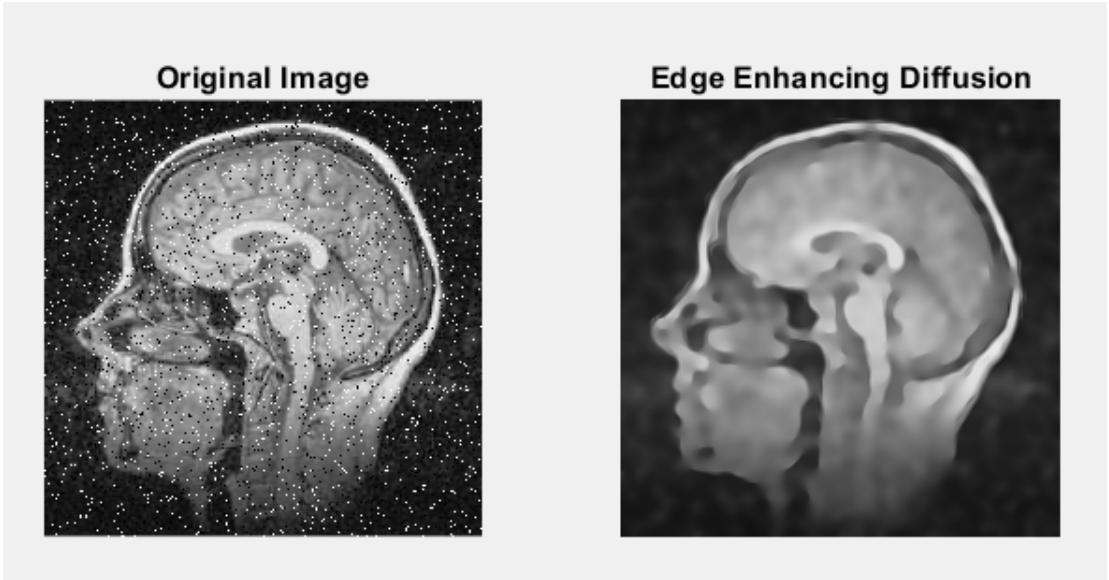
Original Image



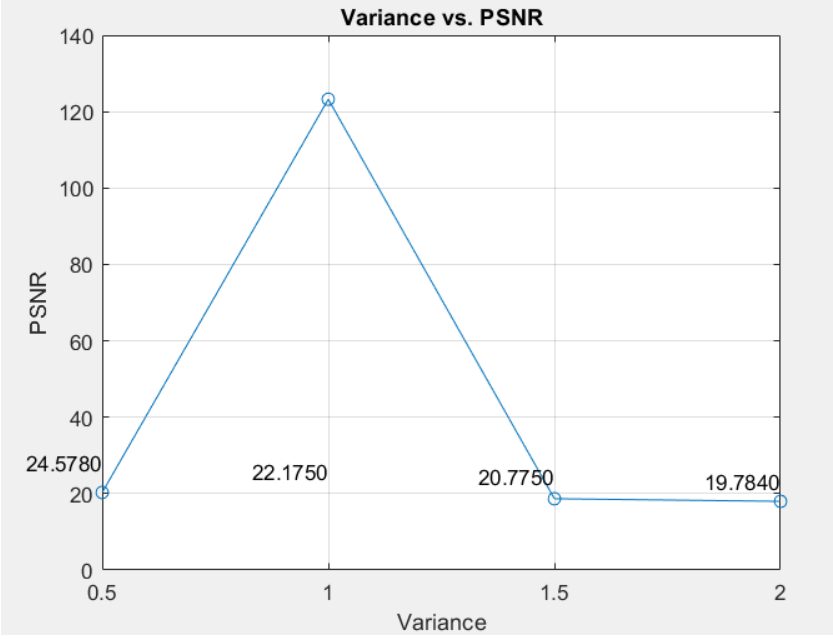
Edge Enhancing Diffusion







These is plot between variance and PSNR values for EED method.



Q.5 From the above findings, can you comment anything on the best stopping time for the respective models?

Sol. To find the optimal stopping time for the Perona-Malik diffusion, one can employ a criterion based on the total variation (TV) of the image, which quantifies the intensity changes within the image. First, estimate the TV of the original image by applying a linear diffusion equation with a small time step. Then, select the stopping time for the Perona-Malik model when the TV of the smoothed image closely matches that of the original image. This approach ensures that the diffusion process strikes a balance between preserving image details and reducing noise.

In the case of Edge Enhancing Diffusion (EED), one approach is to determine the stopping time by minimizing the correlation between signal and noise in the filtered image. This method is versatile, suitable for images with uncorrelated noise and signal components, and doesn't necessitate prior knowledge of noise variance or signal characteristics. It has been successfully applied to various filtering techniques, including Weickert's edge-enhancing anisotropic diffusion.

Alternatively, a second method involves using a rapid and automatic noise-adaptive estimator. This estimator halts the iterative process when the noise level in the image reaches a predefined threshold. Moreover, it identifies the optimal scale parameter (or edge strength) to effectively distinguish between edges and noise. This method has proven to be more effective than conventional anisotropic diffusion in noise reduction while preserving image structures.

Q.5 Figure out how sensitive are the model parameters?

Sol.

The sensitivity of the Perona-Malik diffusion model's contrast parameter on image quality, as measured by Peak Signal-to-Noise Ratio (PSNR), was assessed using provided data for different variance values (0.5, 1.0, and 1.5). PSNR is a metric used to evaluate image fidelity, with higher values indicating better quality.

For variance = 0.5, the contrast parameter showed a notable effect on PSNR. As the contrast parameter increased from 0.5 to 1.5, PSNR increased significantly from 20.775 to 27.271, implying that higher contrast values led to better image quality. However, at a contrast parameter of 2.0, there was a slight drop in PSNR to 26.481.

For variance = 1.0, the trend was similar. Increasing the contrast parameter from 0.5 to 1.5 resulted in a substantial PSNR improvement from 22.175 to 28.8. At 2.0, PSNR dropped to 23.543.

For variance = 1.5, increasing the contrast parameter from 0.5 to 1.5 increased PSNR from 21.47 to 25.7. However, at 2.0, PSNR decreased to 20.169.

Hence, the contrast parameter had a significant impact on image quality as measured by PSNR for all variance levels. Generally, higher contrast values improved PSNR, indicating better image quality. However, the relationship between contrast and PSNR was not strictly linear, and the optimal contrast parameter value varied with the variance level, suggesting the need for a balanced parameter choice to achieve the best image quality.

Similarly, the sensitivity analysis of the Linear diffusion model parameters, specifically the contrast parameter, reveals notable variations in Peak Signal-to-Noise Ratio (PSNR). Higher contrast values generally lead to lower PSNR, implying reduced image quality. However, exceptions exist, such as PSNR increasing from a contrast of 1.0 to 1.5 with variance 0.5. Overall, the contrast parameter significantly impacts model performance, with its influence on PSNR being influenced by other parameters like variance. Further investigation and data points are needed to fully comprehend the model's behavior with different parameter combinations.