Machine Learning for Signal Processing and Pattern Classification.
IMAGE DENOISING – LEAST SQUARE APPROACH

SIMRAN.K

AIM:

To extend the least square solution for the below given problem formulation for signal denoising to image denoising.

$$\min_{x} \left(\|y - x\|_2^2 + \lambda \|Dx\|_2^2 \right)$$

SOLUTION:

Taking the image, adding the noise to it.
j is noisy image.
y1 is the desired output denoised image.
First column wise do column wise denoising so putting j(in position of y) into the least square solution for denoising:

$x = (I + \lambda D'D)^{-1} y$

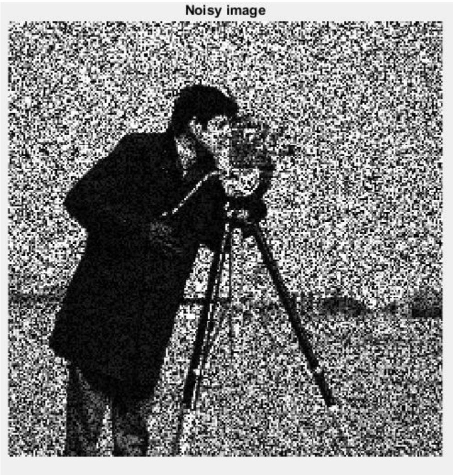
Now transposing x so that row wise denoising can be done. Putting it into the formula will give denoised image but transposed. So we transpose the image to get the output.
The output is a blurred image. Least Square regularization will remove noisy by estimating the original values.

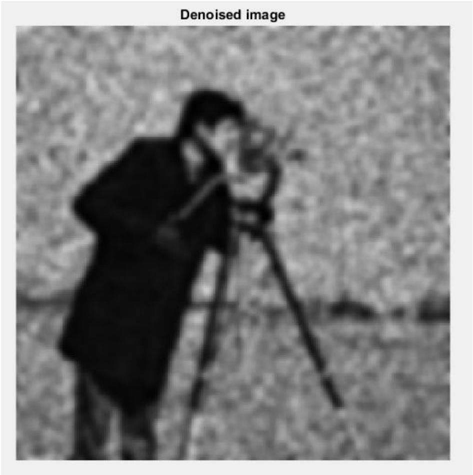
CODE:

```
% 1.
clc;
clear all;
i=imread('cameraman.tif'); % cameraman.tif is a gray built in
image
i=im2double(i);
% imshow(i);
j=imnoise(i,'speckle',0.5); %noise is added to the image
imshow(j) %showing the noisy image
title('Noisy image')
N=size(j,1);
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);%D is a diagonal matrix of
2nd derivation with size N-2 X N
lam = 10; % lamda taking as 50
F = speye(N) + lam * D' * D;  % F is a banded matrix
x = inv(F) * j; % column wise denoising
```

```
x=x'; % transposing to do row wise denoising
y1=inv(F) * x; % column wise denoising but as the image is
transposed, its row wise denoising.
y1=y1'; %Transposing the output to get the origial ouput.
figure;
imshow(y1);
title('Denoised image')
%image is denoised in row wise and column wise.
```

OUTPUT





AIM:

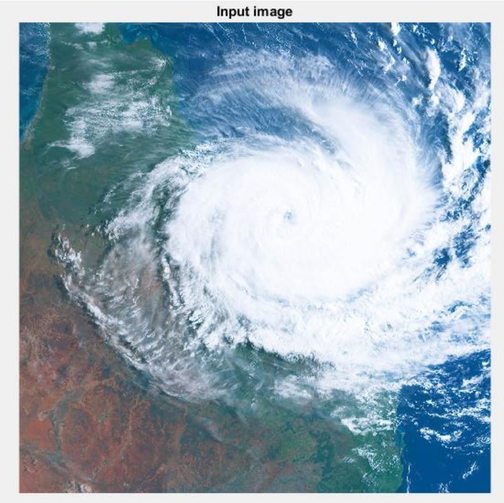
To denoise the satellite image (color) corrupted with Gaussian noise (mean=0 and variance =0.1). Compute the PSNR for the denoised image.

CODE:

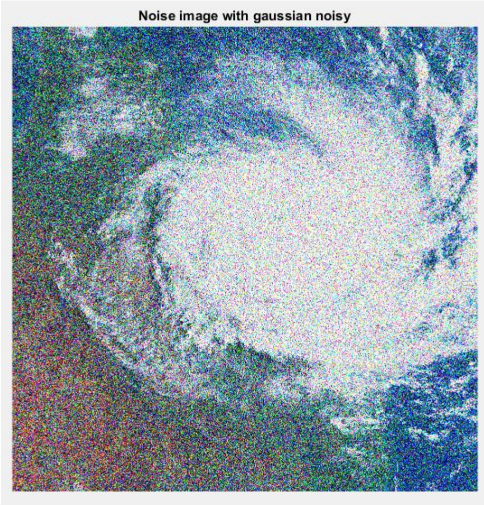
```
clc
clear all

im=imread('cyclone-debbie.jpg'); %Read the cyclone image
im=im2double(im); %the intensity of the image is converted to
double.
N=length(im);
figure
imshow(im); %showing the image
title('Input image');
y=imnoise(im,'gaussian',0,0.1); %Adding gaussian noise with
mean=0 variance=0.1
figure
```

```
imshow(y);
title('Noise image with gaussian noisy');%showing the noisy image
y1=y(:,:,1); % RGB layers of noisy image
y2=y(:,:,2);% RGB layers of noisy image
y3=y(:,:,3);% RGB layers of noisy image
e=ones(N,1);
D=spdiags([e -2*e e], 0:2, N-2, N);%D is a diagonal matrix of 2nd derivation with size N-2 X N
lambda=20;
F=speye(N)+lambda*D'*D;
F=full(F); %sparse double to double is converted
% Column wise
x1=(inv(F)*y1)';
x2=(inv(F)*y2)';
x3=(inv(F)*y3)';
% Row wise least
xr=(inv(F)*x1)';
xg=(inv(F)*x2)';
xb=(inv(F)*x3)';
%Concatenating all layers
yout(:,:,1)=xr;
yout(:,:,2)=xg;
yout(:,:,3)=xb;
figure
imshow(yout) %showing denoised image
title('Denoised Image');
psnr(y,im)  % 11.7297
psnr(yout,im) %21.0240
```

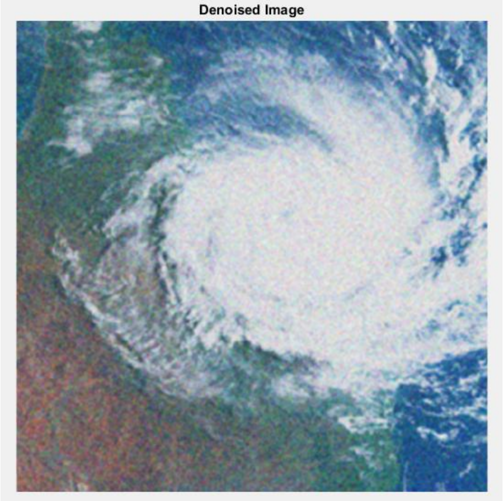
OUTPUT:



Input image



Noise image with gaussian noisy



Denoised Image