Image Smoothening Filters:

Image noise is a random variation of brightness or color information in the captured image. Noise is basically the degradation in image signal caused by external sources such as cameras. Images containing multiplicative noise have the characteristic that the brighter the area the noisier it. We shall discuss various denoising filters in order to remove these noises from the digital images.

Types of Image noise

- Gaussian Noise- Within digital imaging, Gaussian noise occurs as a result of sensor limitations during image acquisition under low-light conditions, which make it difficult for the visible light sensors to efficiently capture details of the scene
- Salt and Pepper Noise - It presents itself as sparsely occurring white and black pixels.
- Poisson Noise- The appearance of this noise is seen due to the statistical nature of electromagnetic waves such as x-rays, visible lights and gamma rays.
- Speckle Noise- Speckle noise is the noise that arises due to the effect of environmental conditions on the imaging sensor during image acquisition.

Types of filters:
- Mean filter
- Median filter
- Gaussian filter
- Wiener filter

Mean Filter

It is also called as Box Averaging filtering technique.

It uses a kernel and is based on convolution.

It calculates the average of all a pixel and its surrounding pixels and the result is assigned to the central pixel.

It is a very effective technique for the removal of **Poisson noise.**

Median Filter

A Median filter is a non-linear filter. It sorts the pixels covered by the window and sorts them in ascending order then returns the median of them. For Median filter in the 2D image. We use medfit(). It takes 2 parameters. First is the noisy image, second is the window size used. By default window size is [3 3].

The median filter removes the **salt and pepper noise** completely but introduces blurriness to the image. It does not perform well with other noises.

Gaussian Filter

Gaussian filter relatively works better with **gaussian and poison noise.**

Wiener Filter

It filters the image pixel-wise.

Wiener2 performs little smoothing, wherever the variance is large. Wiener2 performs more smoothing, wherever the variance is small.

Steps:

1) Read image into the workspace and display it.

imread('eight.tif');

2) Add noise to the image using imnoise(). This type of noise consists of random pixels being set to black or white (the extremes of the data range).
J = imnoise(I,TYPE,...) Add noise of a given TYPE to the intensity image I. TYPE is a string or char vector that can have one of these values:

'gaussian'   Gaussian white noise with constant
               mean and variance

'localvar'   Zero-mean Gaussian white noise
               with an intensity-dependent variance

'poisson'   Poisson noise

'salt & pepper' "On and Off" pixels

'speckle'   Multiplicative noise

E.x: imnoise(I,'salt & pepper',0.02)

3) Filter the noisy image, J, with an averaging filter and display the results. The example uses a 3-by-3 neighborhood.
filter2(fspecial('average',3),J)/255;

4) Now use a median filter to filter the noisy image, J. The example also uses a 3-by-3 neighborhood.
Kmedian = medfilt2(J);
imshowpair(Kaverage,Kmedian,'montage')

Remove Noise By Adaptive Filtering

This example shows how to use the wiener2 function to apply a Wiener filter (a type of linear filter) to an image adaptively.

1) Read the image into the workspace.

   imread('saturn.png');

2) Convert the image from truecolor to grayscale.

im2gray(RGB);

3) Add Gaussian noise to the image

   J = imnoise(I,'gaussian',0,0.025);

4) Display the noisy image. Because the image is quite large, display only a portion of the image.

   imshow(J(600:1000,1:600));
   title('Portion of the Image with Added Gaussian Noise');

5) Remove the noise using the wiener2 function.

   K = wiener2(J,[5 5]);

6) Display the processed image. Because the image is quite large, display only a portion of the image.

   figure
   imshow(K(600:1000,1:600));
   title('Portion of the Image with Noise Removed by Wiener Filter');

Remove Noise by Gaussian filtering
1) fspecial
2) imgaussfilt

Low –pass filters:

img=imread('hestain.png');

gimg=rgb2gray(img);

subplot(331)

imshow(gimg)

title('f(x,y)')

%% Ideal low pass