Indian Institute of Technology Tirupati

Image Processing Lab

Lab sheet. No: 04

Name: Pedasingu Rajesh **Roll. No:** EE21M019

Q1. <u>Smoothing Filters:</u> Write your own MATLAB function to smooth a given input image.

(For this purpose, write your own another function to perform the convolution operation that takes an image and a kernel matrix as inputs, and gives convolved image as the output.

Input Images: Lena_noisy.jpg, Cameraman.png

The inputs to the program are grayscale image and kernel. The output of your program should be a smoothed image. Perform the above method for the following filters of sizes 3x3, 5x5, and 7x7.

(a) Averaging Filter

(b) Gaussian Filter

Note: You can use the "fspecial" command to choose the appropriate filters.

Display the results with your own function and compare your results with inbuilt MATLAB functions "imboxfilt" and "imgaussfilt".

Mention the inferences about the effect of kernel size variation in each method and sigma variation in Gaussian filters.

Aim: To write MATLAB function to smooth the given input image, and To display and to compare the results with inbuilt MATLAB functions "imboxfilt" and "imgaussfilt".

Output:

given image

averaging filter window= 3

averaging filter window= 5

averaging filter window= 7

given image

averaging filter window= 3

averaging filter window= 5

averaging filter window= 7

given image

gaussian filter window= 3

gaussian filter window= 5

gaussian filter window= 7

given image

gaussian filter window= 3

gaussian filter window= 5

gaussian filter window= 7

Inferences:

1. We can observe that, both averaging and gaussian filter it **reduces the noise** in the image but it **blurs** the image.
2. Averaging filter smoothen the image and we **lost the edge information** due to blur.
3. We can observe that if the size of the window increases then smoothing is more and the image become more blur.
4. For larger variance value the image has good contrast, but for low variance images became low contrast.
5. smoothing is not the good method to reduce the noise in the image .

comparison with in-built command

gaussian filter window= 7

using imgaussfilt command withh window-7

averaging filter window= 7

using imboxfilt command withh window-7

Q2. Median Filter :

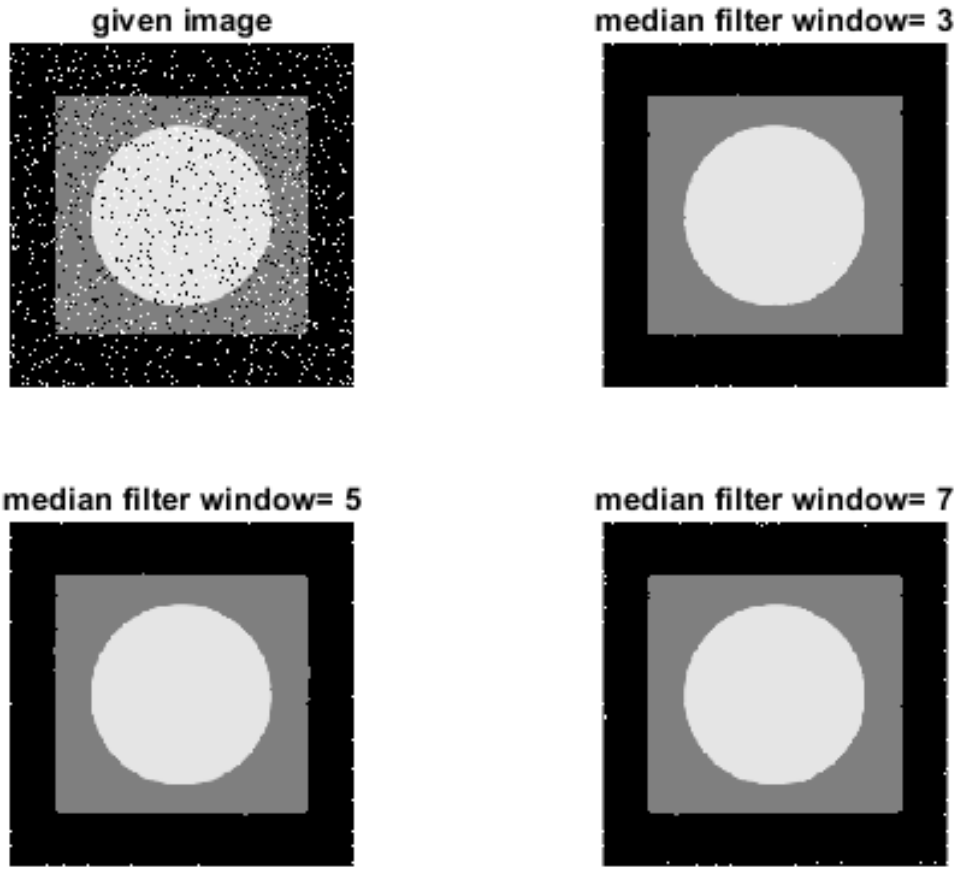
Write your own MATLAB function to perform median filtering on the given input image.

Input Images: salt-pepper-noise.tif, circuit-board.tif

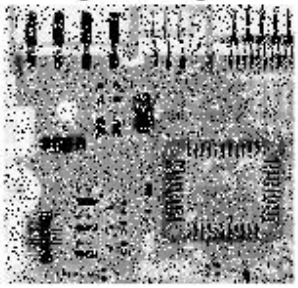
The inputs to the program are grayscale image and kernel size. The output of your program should be a filtered image. Perform this for the kernel sizes 3x3, 5x5, and 7x7. Display the results with your own function and compare your results with the inbuilt MATLAB function "**medfilt2**". Mention the inferences about the effect of kernel size variation.

Aim: To write MATLAB functionto perform median filtering on the given input image and to display and compare the results with the inbuilt MATLAB function "medfilt2".

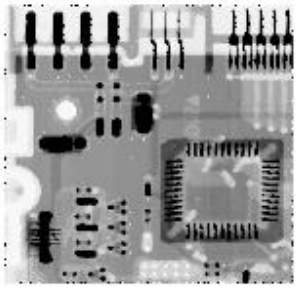
Output:



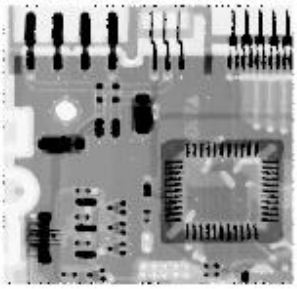
given image



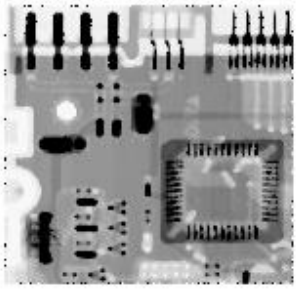
median filter window= 3



median filter window= 5



median filter window= 7



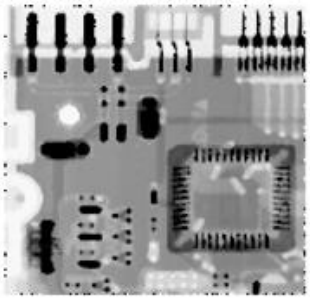
Inferences:

1. The given image contains salt pepper noise. Noise is impulsive type.
2. When we perform the medina filter on given image it removes the salt pepper noise very efficiently.
3. By performing median filter the resultant image Retain it edge information.
4. If the size of the window filter increases , it gives the better results when compared to small window size.
5. Median filter is perfect for removal of salt pepper noise, because median filter is **Non-linear filter**

Comparison with inbuilt command

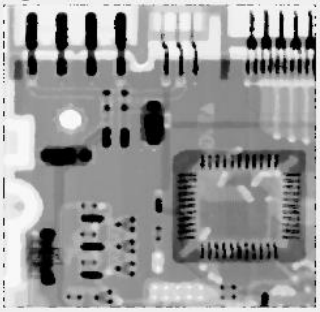
median filter window= 7

median filter window= 7



using medfilt2 command withh window-7

using medfilt2 command withh window-7



Q3. Edge Detection using Sobel Operators:

Write your own MATLAB function to find the horizontal and vertical edges in the given input images. You can use Sobel edges for this purpose.

Input Images: House.jpg, Bricks.jpg

The inputs to the program are grayscale image and appropriate kernel. The output of the program should display the edges.

Display the following outputs:

a) Horizontal edges

b) Vertical edges

c) Magnitude of gradient

d) Threshold the Gradient magnitude to highlight the edges.

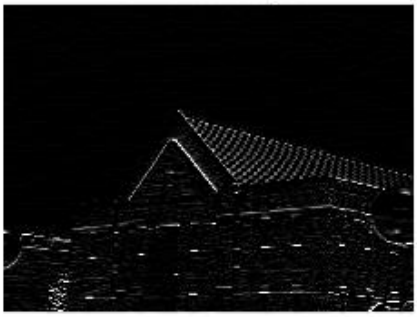
e) Display diagonal edges with ±45°, ±135° degrees after creating bins of 10 degrees (e.g., ±45° ±5°, ±135° ±5° will contribute to the diagonal edges). Write the observations from your results.

Aim: To write MATLAB function to find the horizontal and vertical edges for the given input images and To display: horizontal edges, vertical edges, magnitude of gradient, threshold the gradient magnitude to highlight the edges, and diagonal edges with ±45°, ±135° degrees after creating bins of 10 degrees.

Output:

given image

horizantal edges



vertical edges

magnitude of edges

given image

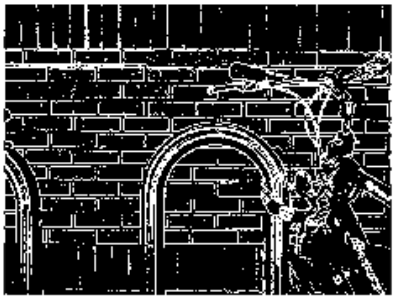
horizantal edges

vertical edges

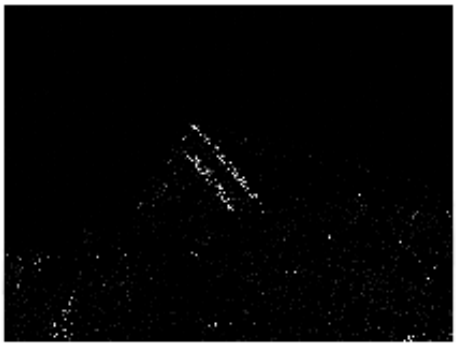
magnitude of edges

after thresholding magnitude of edges image-1

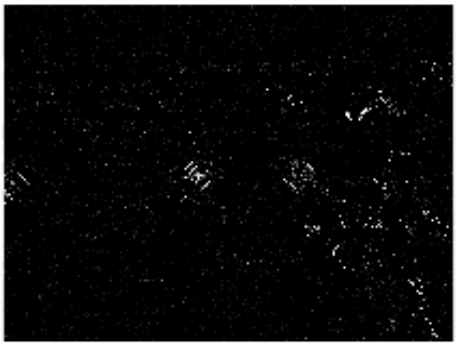
after thresholding magnitude ofedges image-2



diagonal edges(45 and 135 degrees)in image-1



diagonal edges(45 and 135 degrees)in image-2



Inferences:

1. By applying the Sobel **horizontal gradient** and **vertical gradient** we will get the **vertical edges**(Gx) and **horizontal edges**(Gy) respectively.
2. By performing the $G = \sqrt{Gx^2 + Gy^2}$ will give us the magnitude of edges. This will give the all the edges in the given image.
3. By applying the threshold on the magnitude of edges it will gives the perfect dominant edges in the given image and removes the non-dominant edges.
4. Finding the diagonal edges with a given range of angle (±45°±5°, ±135°±5°),
   a. First we find the all the angle of edges using $\tan \emptyset = \tan^{-1} \frac{Gy}{Gx}$.
   b. From the angle , we get take the magnitude of those angles (±45°±5°, ±135°±5°), and rest of the all images pixels are set as zero intensity value.
   c. When we display the resultant image it only shows the angle of edges in the bin range of (±45°±5°, ±135°±5°).