|  |  |
| --- | --- |
| **Homework #2**  *Digital Image Processing(EEE5320), 2019-2* | Due Date: 2019. 10. 21 |

1. Gaussian Filtering (25pt)



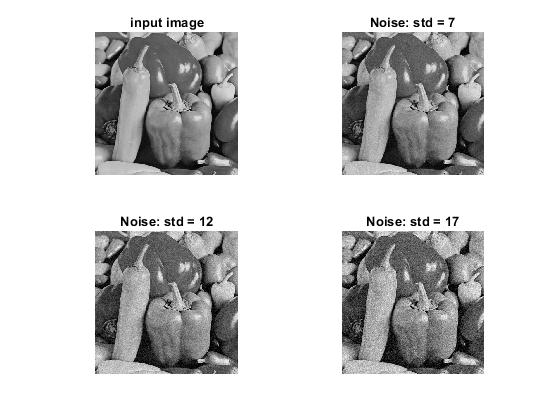
**Figure 1. input1.jpg**

Read the attached “input1.png” and please answer following questions.

* 1. Implement a Matlab function for Gaussian noise and Display the noise images with varying standard deviation (7, 12, 17)

**(You should not use ‘imnoise’ built-in function)**

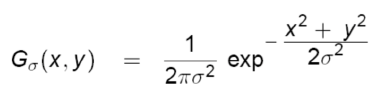
|  |
| --- |
| function output = GaussianNoise (I,SIGMA) %  % random number from SIGMA std Gaussian distrubution  output = I + SIGMA.\*randn(size(I));  end  ***% Complete the remaining part*** |
| **[GaussianNoise.m]** |



* 1. Implement a Matlab function for Gaussian filtering.

**(You should not use ‘imfilter’ built-in function)**

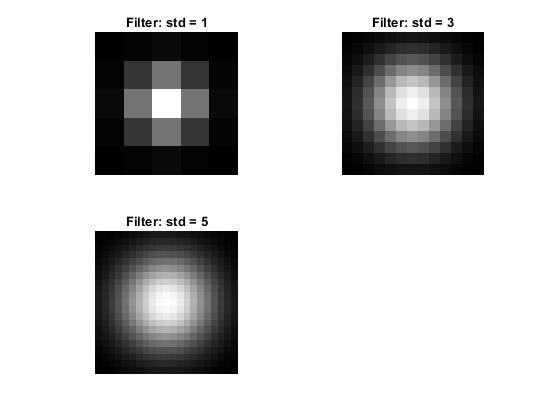
The Gaussian function is as follow:



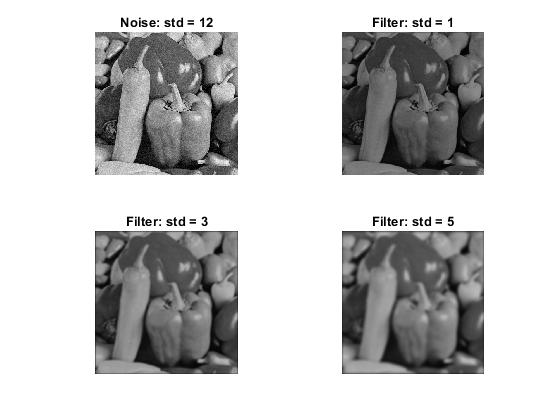
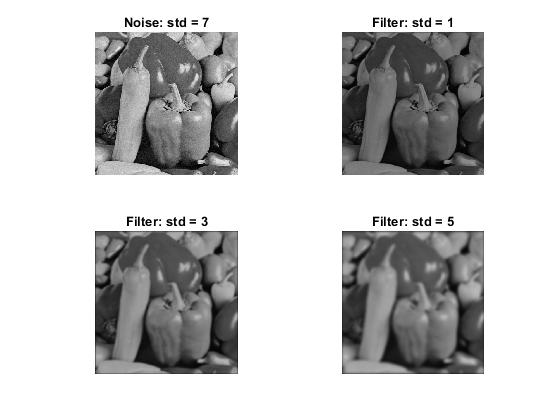
Where  are the location of the pixel and  is a standard deviation

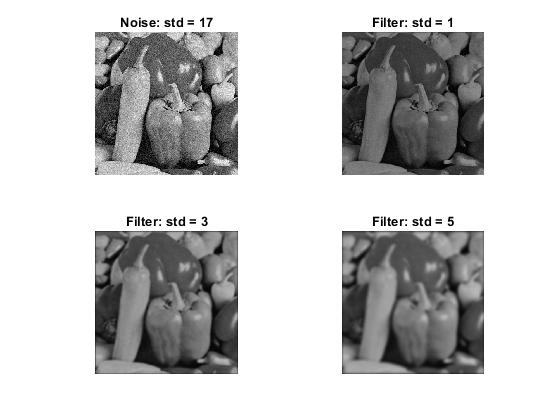
|  |
| --- |
| function output = GaussianFilter (I,SIGMA) %  % where I is and Gaussian noise images(std : 7,12,17),, SIGMA is standard deviation of Gaussian function  % Each dimension of filter is determined automatically by 2\*ceil(2\*sigma)+1  % filter dimension  G\_len = 2\*ceil(2\*SIGMA)+1;  % make zero matrix corresponding to filter size  G\_mat = zeros(G\_len,G\_len);  % x,y range  range = linspace(-G\_len/2,G\_len/2,G\_len);  % calculate filter matrix values  for x = 1:G\_len  for y = 1:G\_len  % Gaussian distribution  G\_sigma = 1./(2\*pi\*SIGMA\*SIGMA).\*exp(-1 .\* (range(x)\*range(x) + range(y)\*range(y))./(2\*SIGMA\*SIGMA));  G\_mat(x,y) = G\_sigma;  end  end  % return filter matrix  output = G\_mat;  end  ***% Complete the remaining part*** |
| **[GaussianFilter.m]** |

* 1. Visualize your gaussian filter with standard deviation (1, 3, 5), respectively.



* 1. Display the filtered images with varying standard deviation (1, 3, 5)   
     (input : the gaussian noise images with varying standard deviation (7, 12, 17))





* 1. Explain your implementation and discuss your results

1. Bilateral Filtering (25pt)



**Figure2. cat\_crop.png**

Read the attached “cat\_crop.png” and please answer following questions.

* 1. Implement a Matlab function for Gaussian noise and Display the noise images with varying standard deviation (7, 12, 17)

**(You should not use ‘imnoise’ built-in function)**

* 1. Implement a Matlab function for Bilateral filtering.

**(You should not use ‘imfilter’ built-in function)**

The output of bilateral filtering is as follow:

Where are neighboring pixels centered on in the kernel and is the intensity value of a pixel.  is a standard deviation. The normalization term .

|  |
| --- |
| function output = BilateralFilter (I,SIGMA\_d, SIGMA\_r, kernel\_size) %  % where I is and Gaussian noise images(std : 7,12,17), SIGMA\_d and SIGMA\_r are standard deviations of Bilateral function. Kernel size n is nxn box.  end  ***% Complete the remaining part*** |
| **[BilateralFilter.m]** |

* 1. display the filtered images with varying standard deviation = 20 and = (1, 2, 10) when all kernel sizes are 5.
  2. Explain your implementation and discuss your results
  3. Compare the results for the Gaussian filter with the bilateral filter and discuss the results.

1. Median Filtering (25pt)



**Figure 3. input1.jpg**

Load the attached “input1.png” and please answer following questions.

* 1. Implement a Matlab function for Median filtering.

**(You should not use ‘imfilter’ built-in function)**

|  |
| --- |
| function output = MedianFilter (I, kernel\_size) %  % where I is and input image, and kernel\_size n is nxn size of the filter  end  ***% Complete the remaining part*** |
| **[MedianFilter.m]** |

* 1. Add salt and pepper noise to the image with the percentage of spikes as from 10% to 50% by step of 10% **(You should not use ‘imnoise’ built-in function)**

|  |
| --- |
| function output = SaltAndPepper(img, ND)  % Add salt and pepper noise to image of certain density  % ND = noise density, if ND is 0.2, noisy image has 20% noise  end  ***% Complete the remaining part*** |
| **[SaltAndPepper.m]** |

* 1. Apply median filtering function to the corrupted images with varying the size of filter ([3,3], [5,5], [7,7]) and display the corrupted images and the filtered results.
  2. Explain your implementation and discuss your results.

1. Unsharp masking (25pt)



**Figure 4. eye.png**

Read the attached “eye.png” and please answer following questions.

* 1. Implement a Matlab function for unsharp mask filter. The output of unsharp mask filtering is as follow

where means blurred image.

|  |
| --- |
| function output = UnsharpMask (I, I\_blur, k) %  % where I is and input image, and I\_blur is blurred image and k is boosting factor  end  ***% Complete the remaining part*** |
| **[UnsharpMask.m]** |

* 1. Display the blurred image and sharpened images using unsharp mask filter with blurred image by average filter as average filter size = [3, 3], [11, 11] and k=1.

1. Set k = 3 and repeat ii.
2. Explain your implementation and discuss your results.