**EE 475 Fall 2018**

**HW3**

**MEAN vs MEDIAN**

In the 8x8 image X shown below, all pixels shown with an asterisk \* have gray level 41, all the other pixels have gray level 14.

1. Suppose we linearly smooth X using a 1-row by 3-column averaging mask whose origin is the center of the mask. How many of the 64 pixels in X will change value? What is the largest increase in brightness any pixel in the given image X will have?
2. Suppose instead we smooth X using a median filter with the same mask size and origin as in (a). How many of the 64 pixels in X will change value? What is the largest increase in brightness any pixel in the given image X will have due to this filtering?



**CONNECTIVITY and PATH LENGTH:** Consider the image segment below where lower left corner is pixel p and upper right corner is pixel q.

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 1 | 2 | 1 |
| 2 | 2 | 0 | 2 |
| 1 | 2 | 1 | 1 |
| 1 | 0 | 1 | 2 |

Compute the lengths of the shortest path 4-, 8- and m-paths for the two cases of: V = {0,1} and V = {1,2}. Here we are addressing gray-level connectivity, where connectedness is defined for the set of gray levels V.

**ESTIMATION of NOISE PARAMETERS**

Given the contaminated Lena image and the BBC image, estimate parameters of the Rayleigh pdf from the moments. Recall:

1. Plot the histograms of uncontaminated and contaminated images
2. Estimate the parameters (a, b) of the Rayleigh distribution. Does it check with the true parameters?

**GAUSSIAN NOISE**

Contaminate the Pentagon image with Gaussian noise N(0, 1000). Clip underflows and overflows. Plot the original and noisy images side by side. To denoise the images, apply the following filters, calculate and tabulate the PSNR and SSIM values, and plot filtered images.

a) 5x5 box filter, that is the mean filter,

b) Gaussian filter with , and (create its mask with the Matlab’s fspecial function)

c) Median filter with size 5x5,

d) 5x5 alpha-trimmed filter with trimming parameter p = 3.

e) Geometric mean filter in a 5x5 neighborhood

g) Harmonic mean filter in a 5x5 neighborhood

h) Contraharmonic filter with Q = 2.

1. Plot the images and write in their caption the PSNR and SSIM values.
2. Tabulate the results in 3x8 table where the first row is the name of the filter, the second and thirds rows correspond to PSNR and SSIM
3. Do you observe any discrepancy between your subjective quality evaluation and the scores, i.e., PSNR/SSIM is high but the image is lousy, blurred, or vice versa, the score is low but you prefer that restored image to another one with a higher score? Comment on the loss of detail while noise is being removed, e.g., are parked car or window details looking on the inside still distinguishable?



**SALT & PEPPER NOISE**

1. Consider the Rose image contaminated with 15% salt-and-pepper noise, that is, 7.5% of pixels are randomly affected by salt noise and 7.5% of pixels are randomly affected by pepper noise. Apply the median filter and alpha-trimmed mean filter with sizes 3x3, 5x5, 7x7, 9x9, and 17x17. Notice that while S&P noise is removed, hence image improved, there could also be loss of fidelity with increasing window size.
2. Plot 1D graphic with abscissa the window size, and the ordinate the percentage of S&P noise removed by the median filter. Plot these five images as well.
3. Plot 1D graphic with abscissa the window size, and the ordinate the PSNR figure. You must experiment to find the best alpha value for each window size. Plot the images as well.
4. Use a heat map to illustrate the performance of the median filter. The performance should be measured in terms of PSNR. The horizontal axis should be S&P noise as a percentage of pixels affected, from 5% to 50% in steps of 5. The vertical axis should denote the window size in steps 3x3, 5x5, 7x7, 9x9, 11x11, 17x17, 23x23. Enlarge the image on all sides by 11 pixels by reflection. The heat should be proportional to the PSNR. Set blue to high PSNR, red to low PSNR. For example, use colormap(flipud(jet)).

You can use xticklabels ve yticklabels function in order to correct for the axis ticks.

***Image Quality Metrics:***

There are several metrics to measure image quality and the performance of image processing algorithms.. The most common one is the Peak-Signal-to-Noise-Ratio:

.

where J and I are, respectively, the processed and the input images. SNR or PSNR is criticized for not reflecting subjective evaluations. A second index, more reflective of human judgement is SSIM: Structural Similarity Index Measure:

<https://en.wikipedia.org/wiki/Structural_similarity>

<https://www.mathworks.com/help/images/image-quality-metrics.html?requestedDomain=www.mathworks.com>

***Heat Maps***

In scientific research, we are often confronted with the problem of presenting a large, often huge quantity of numerical results. Just listing the thousands and thousands of numbers does not make a good presentation. Heat map is a visualization tool, such that one look at the map will reveal you the essence of all those numerical results (http://en.wikipedia.org/wiki/Heat\_map). I suggest you use the heat map to show the resulting improvements and deteriorations in your experiments. A suggested piece of code for this purpose is below: 's', and 'd' are vectors, and 'nofNoise' is a matrix. For example, you could choose in the plot the abscissa to represent one of the parameters, say the filter size; and the ordinate to represent another parameter, such as the percentage of S&P noise remaining in decreasing order. The color code represents the performance index.

d=0.05:0.05:0.4

s=3:2:15;

for i=1:length(d);

J = imnoise(I,'salt & pepper',d(i));

for k=1:length(s);

    f=medfilt2(J,[s(k) s(k)],'symmetric');

    nofNoise(i,k)=length(find(f==0 | f==255));

end

end

imagesc(s,d,nofNoise),colorbar