ECE 253, Homework 2

Due: Tuesday October 18, 2022 by 11:59pm

The first three problems should be done using Matlab, and the last one should be done by hand. Submit your homework electronically in Canvas on Gradescope. Everything can be uploaded as one PDF file– include your answers to each question and your Matlab code (cut and paste it in). Include your full name and PID.

1) **Histogram Equalization**

Use imread to read in the image lungs.jpeg.
This is a grayscale image but is stored as an RGB image with 3 color planes, so you can use:

```
a = lungs(:,:,1);
imshow(a)
ghe = histeq(a);
imshow(ghe)
```

to define a grayscale image ”a” and compute the global histogram equalized version (ghe) of it.

a) Read what the function ”cumsum” does. Carry out the following sequence of steps:

```
[n,y] = imhist(a,256);
map = cumsum(n);
map = map / max(map);
plot(y,map)
imshow(a, [map map map])
```

Explain what ”map” represents, what the plot shows, and what image is being shown with this imshow command.

b) The dark background on this image is hurting the histogram equalization of this image. Manipulate the histogram in some way which will help this problem, and explain what you did. Then use cumsum to generate a new map, and show your image with this new map. Plot the new map on the same plot as the previous one, and interpret the differences in your equalized images relative to the differences in these plots. Include your plot with 2 curves and your explanation of results.

2) **Noise Cleaning**

Load the files cleanbaby.mat (the baby image without added noise), babyS.mat (the baby image with salt noise added, in the form of little vertical streaks), and baby2.mat (the baby image with both salt noise and a low level of Gaussian noise).

a) Use medfilt2 with a 1x3 median filter to clean up the babyS image, and also to clean up the baby2 image. Likewise use a 3x3 median filter to clean them up. In all four cases, use the function immse to compute the mean squared error (MSE) between your filtered output and the cleanbaby image. You should find that the 3x3 filter does better (in terms of MSE) than the 1x3 filter for one of the noisy images but the 1x3 is better for the other. Explain why this occurs, and provide your matlab commands and your MSE values.

b) Try the sequential use of a median filter and a spatial averaging filter on the baby2 image that has both noise types. Provide your Matlab code, the MSE between your output image and the cleanbaby image, and a discussion of your results.

3) **Unsharp Masking**

a) You are to spatially enhance an image using unsharp masking. Consider the following MATLAB function:

```
function im_out = unsharp( im_in, maskA, weight )
[a,b] = size( maskA );
maskB = zeros( size( maskA)); maskB(ceil(a/2),ceil(b/2)) = 1;
maskC = maskB - maskA;
maskD = maskB + weight * maskC;
im_out = conv2(im_in,maskD,'valid');
```

Here im_in is the input image and im_out is the output image. Suppose maskA is a small odd-sized lowpass filter mask, and weight is a positive number. What kind of masks are masks B, C and D? Using the discussion from class on separating an image into lowpass and highpass components, explain how this function performs edge sharpening.

b) In unsharp masking you can choose which lowpass filter to use and how much weight to give the highpass part. We will investigate the effect of these on the resulting image. First, create a test image of size 128×128 that consists of a ramp and simple step function, as follows:

```
tst=ones(128,1)*[64*ones(1,32) (64:4:188) 192*ones(1,32) 64*ones(1,32)];
```

This has four equal sized areas (from left to right): first, 32 columns with value 64, then 32 columns of ramp going from 64 to 192, then 32 columns with value 192 and finally 32 columns with value 64. Try a few combinations of low pass filters and weights. Vary the size of the mask (e.g., 3×3, 5×5, maybe even 7×7) and the entries in the mask (e.g., unweighted averaging vs. strongly center-weighted averaging) and vary the extra weight given to the highpass part. Discuss the results in each case, and note any trends you see that arise from varying the parameters. Also look at a slice of the filtered images:

```
plot(tst(64,:));
```

Include in your homework a few of the plots of horizontal slices through the filtered images, with discussions of the trends.

c) For a real-world example at larger size, read in the image "blurry-moon.tif" and use highboost filtering on it starting with the following Gaussian lowpass filter:

$$f = 1/159 \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Try different boost values. Provide your code and your output image, and comment on your result.

4) **Order of Operations** (this is not a Matlab problem)

An image has been corrupted with a low level of additive Gaussian noise, and with 1% of salt and pepper noise. The noise is independent from pixel to pixel. We will process the image with 3 operations in some order: contrast manipulation, median filtering, and spatial averaging filtering. Consider that the image is stored as values ranging from 0 to 1.

- The contrast manipulation (CS) consists of remapping an input value s to an output value r according to $r = \sqrt{s}$
- The median filter (MF) is a 3x3 median filter
- The spatial averaging (SA) filter is the unweighted 3x3 mean filter

The 3 operations can be done in 6 possible orders:

A: MF $\rightarrow$ SA $\rightarrow$ CS
B: CS $\rightarrow$ MF $\rightarrow$ SA
C: SA $\rightarrow$ CS $\rightarrow$ MF
D: MF $\rightarrow$ CS $\rightarrow$ SA
E: SA $\rightarrow$ MF $\rightarrow$ CS
F: CS $\rightarrow$ SA $\rightarrow$ MF

a) Which of these systems, if any, will produce an output image that is exactly the same as the output from another system? Explain why?
b) Which of these systems, if any, do you expect to do better at noise removal? You are not being asked for a complete ranking of the systems, but rather just to point out (and explain) if certain placement orders clearly outperform some other ones for this noise removal task.