CS270 Digital Image processing
Homework 2
Due date: Oct 23th, 2018
作业结果提交纸质版，课堂提交或送至信息学院 2 号楼 302G 办公室（请务必同时在名单上签名）
程序代码每题一个文件夹，请加入注释，最后压缩打包发送至助教刘宇婷邮箱：liuyt1@shanghaitech.edu.cn。请在邮件中注明姓名、学号。

Note: There are two functions you may refer for better display.

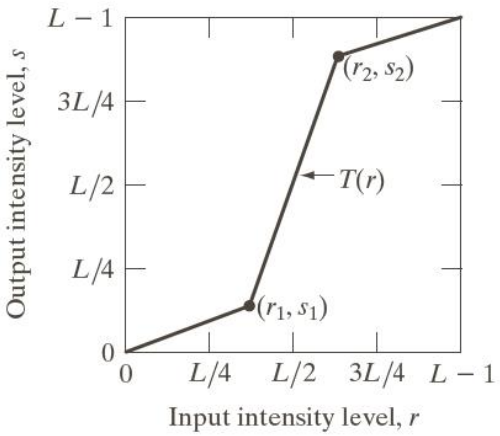
(1) “ImageDisplay.m”: to display the gray image properly.

(2) “scale2bytes.m”: to extend the intensity level to [0,255].

1. Contrast stretching

Please repeat the operation in Figure 3.10. Given low contrast image (“Image1.mat”, intensity level L=256), if the transformation function is as shown in the left figure. Let

$$r_1 = 96, \quad s_1 = 32, \quad r_2 = 160, \quad s_2 = 216$$



1) Derive the transformation function $s = T(r)$

For figure(1):

2) Display the transformation function curve in *subplot(2,2,1)*
3) Display the original low contrast image “Image1.mat” in *subplot(2,2,2)*
4) Display the processed image after contrast stretching with curve in 2) in *subplot(2,2,3)*
5) Display the binary image with $r_1 = r_2 = 120, s_1 = 0, \ s_2 = 255$ in *subplot(2,2,4)*

2. Histogram

1) Develop the function which computes the gray-level histogram of the uint8 (intensity level of 256) image. (Print out the function with comments in your homework solution)

For figure(2),

2) Find the histogram of "Image21.mat" using the function just developed and display in *subplot(1,2,1)*.
3) Display the histogram of "Image21.mat" using Matlab function "imhist" in *subplot(1,2,2)*.
   Compare the two histogram, and pay attention to the axis.

4) Design a function *h*=match(*f*,*g*) to modify the gray levels of an uint8 image *f* so that its gray-level histogram matches that of another uint8 image *g*. (Print out the function with comments in your homework solution)

For figure(3)

5) Display "Image21.mat" in *subplot(3,2,1)*
6) Display the histogram of "Image21.mat" in *subplot(3,2,2)*
7) Display high contrast image "Image22.mat" in *subplot(3,2,3)*
8) Display the histogram of "Image22.mat" in *subplot(3,2,4)*
9) Use the "match" function to match the histogram of "Image21.mat" to "Image22.mat". Display the new matched image in *subplot(3,2,5)*
10) Display the histogram of new image in *subplot(3,2,6)*

3. Spatial Filtering – Smoothing

Apply the 3*3 average filter (in the figure as below) and median filter to the image with salt and pepper noise ("Image3.mat", i.e. Fig3.35 in the textbook), and compare the results.

- Use "conv2" to do the average filtering, and set the 'shape' parameter to 'same' in "conv2".
- Develop your own median filter, and you can verify the result with Matlab function "medfilt2"
- Display original blurred image in *subplot(2,2,1)*, two images after average filtering in *subplot(2,2,2)* and *subplot(2,2,3)*, and the image after median filtering in *subplot(2,2,4).* Compare the difference between different filtering.

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \qquad \frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

4. Spatial Filtering – Sharpening

Enhance the blurred image ("Image4.mat") by applying a 3x3 Laplacian kernel in the spatial domain.

- Use "conv2" to do the convolution and set the 'shape' parameter to 'same' in "conv2".
- Apply all the filter masks in the figure as below to the blurred image.
- Use the transformation function to get the sharpened image.

$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)], \qquad c = \pm 1$$

- Display original blurred image in *subplot(3,2,1)*, and other 4 sharpened images in *subplot(3,2,3)- subplot(3,2,6)*. Compare the difference among the different masks.

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1