Matlab hands-on 3

Make the Matlab program as instructed below.

1. Make a 2D matrix u by using matrix v as follows.

   v = [3 5 -2 5 -1 0]

   1) If a value in v is great than 0, move it to u at the same location. Otherwise, change the value to 0. You MUST use find() function, e.g. v = [5 4 -3] => u = [5 4 0].

   2) Using if and for (not using find() at this time), do the same thing as described in 1).

   3) Make a customized function thres() which runs in exactly same was as 1) and save it into a mfile as described below.

   -thres() MUST have one input parameter. The input parameter is a matrix like v in 1).

   -thres() MUST return the output matrix like u in 1)

   (hint: Before you start coding, you might need to run "help function" in the Matlab command window to remind the usage of 'function'.

   4) Run 1) again by using thres() function. (Note: Before using thres(), you need to make sure that your mfile of the function is on the same path as the current matlab work folder.)

   5) Make program in apply_thres.m where you call the function thres() with v2 as an input parameter to generate output matrix u2. Then, execute apply_thres().

   v2 = [-4 2 -5 3 -2 0 1000]

2. File IO

   1) Load 'lenna.txt' using load(). (hint: 'lenna.txt' is ASCII type then we MUST load ASCII type data as like matrix = load(filename, 'ASCII')
   2) Display the image on the screen. (hint: Use figure and imshow())
   3) Save the image into a mat file (lenna.mat).
   4) Do the same thing from 1) to 3) using 'lenna.bmp' by using imread() and imwrite(). (hint: imread(filename , datatype) , imwrite(loaddata, filename, datatype) ,datatype => 'bmp')

3. Simple image processing 1

1) Using imread() read 'chest_xray.jpg' into a matrix ori_image.

2) Display the image size, min value, and max value.

3) Make a matrix rev_image which has the same size as ori_image. (hint: zeros(), size())

4) For each pixel calculate (max value) – (pixel value) and put the results into rev_image array.

5) Display ori_image and rev_image within the same window. (subplot(), imshow())

6) Put rev_image into a bmp file. (imwrite())

4. Simple image processing 2

1) Using imread() read 'lenna.bmp' and store it into image1 array, and display it into subplot1.

2) Get the size and mean value of image1.

3) Make zero matrix image2 and image3.

4) Change the pixel values of image2 by the rule below.

   If a pixel in image1 is lower than the mean value, put 0 at the same pixel location. Otherwise, move the pixels from image1 to image2.

   (hint: review the thres() codes in the hands-on last week.)

5) Display image2 into subplot2

6) Make a function func1() that performs the same thing as 4). You need at least one input parameters: input image matrix.

7) Call func1() with image1 as an input parameter, store the results into image3, then display it into subplot3.