CS517 Programming Assignment 1

It is regarding implementing in Matlab some Image enhancement operations/methods example: · Image Resize using nearest neighbour and bilinear interpolation (to be compared with imresize), image rotate (compare later with imrotate), bitplane slicing, Image reconstruction given some tie points, Histogram equalization, Adaptive histogram equalization, Histogram matching, etc.

///Note – You have to implement these basic image processing operations by yourself and not use inbuilt functions like imresize, imrotate, imhist, histeq etc except for the comparison

Single program file to be submitted and programming language Matlab preferred. Report can be submitted by Thursday, 6 Feb 2020

The file name format to be as follows:

A1_<Yourfirstname>_<EntryNumber>_2019_CS517.m

For example: A1_Keshav_2018CSB9999_2019_CS517.m

Function input parameters: (qID, fname_inp1, fname_inp2, fname_out, prmts, toshow)

where q denote queryID, fname… denotes image filenames / string (likely .tif) for input(s)/output, and prmts denotes parameters input matrix. toshow will be either 0 or 1 and if it is non-zero the subplots/figures to be shown else not.

Example: A1_Keshav_2018CSB9999_2019_CS517(1, 'test_pattern.tif', ' ', 'temp_out', [788, 1200])

Below we describe the input/output parameters description for different functions needed to be implement in this assignment. Consider input1, input2 and output1 image sizes are denoted as M1xN1, M2xN2, and M3xN3 respectively. Assuming all input/output images to be 2D and grayscale, or you may convert to grayscale in case it is not so for some reasons. Also if there comes need to compute mse/rmse between 2D images, compute considering [0-255] intensity values, not [0-1], and note that if for some reasons, 2 images are not of same dimensions, use the dimensions that are smaller.

qID	prmts	functionality	Constraints	Remarks / Output
1	[M3, N3]	Resize input image using nearest neighbourhood interpolation	M3>=M1, N3>=N1 (i.e. always upscaling) Fname_img2 is need not be used. Null only	• Save output resized image as fname_out.tif. • Output rmse between your output image and system inbuilt command (imresize using nearest) based generated image, SImg. • If (toshow), Use Subplot(2,2,…) to show input image, your output image, SImg and system output image using bicubic. Have proper titles that can possibly include info of image size and mean grayscale value and mse for output images (your - SImg)
2	[M3, N3]	Resize input image using bilinear interpolation	M3>=M1, N3>=N1 (i.e. always upscaling) Fname_img2 is need not be used. Null only	• Save output resized image as fname_out.tif. • Output rmse between your output image and system inbuilt command (imresize using bilinear) based generated image, SImg. • If (toshow), Use Subplot(2,2,…) to show input image, your output image,

				SImg and system output image using bicubic. Have proper titles that can possibly include info of image size and mean grayscale value and mse for output images (your - SImg)
3	theta	Rotate image by angle theta anticlockwise	Theta is any value between 0 to 180 (denoting angle in degrees) and factor of 5. Fname_img2 is need not be used. Null only	• Save output resized image as fname_out.tif. • Output **the dimensions of output image** and the rmse between your output image and system inbuilt command imrotate based generated image, SImg. If (toshow), Use Subplot(1,3,…) to show input image, your output image, SImg and system output rotated image. Have proper titles that can possibly include info of image size, mean grayscale value and mse for output images (your - SImg)
4	bpx	Bit plane splicing	bpx is 1 to 3 elements row vector matrix where each element is unsigned int value btwn 0-255 e.g. bpx=[3 66 128] implying that output image1 reconstructed using 2nd and 3rd bit (counting LSB as 1st bit), image 2 using 7th and 2nd bit, and output 3rd image uses all bits. Fname_img2 is need not be used. Null only	• For each element in bpx, compute output image based on bitplane slicing of input imag and bpx. Whatever bits are on (i.e. 1) in bpx, use corresponding bitplanes only of input image to reconstruct • Save output image(s) as fname_out_1.tif, fname_out_2.tif, and/or fname_out_3.tif. • Output rmse(s) between input and your output image(s). If (toshow), Use Subplot(1,2,…) or Subplot(1,3,…) or Subplot(2,2,…) to show input image, your output image(s). For each image, have proper titles that can possibly include info of image size, mean grayscale value and mse for output images (your – input image)
5	TiePts points matrix 4x4	Reconstruct image from (affine) transformed image	Fname_inp2 is valid and denotes file name of (affine) transformed image. TiePts is 4x4 matrix where each row denote 1 tie point: [x1 y1 x2 y2] Implying (x1,y1) in input1 is mapped to (x2,y2) in input2	Without referring to intensity value of input image1, compute the estimated input image from input image 2(i.e. transformed matrix) and TiepPts based transformation, as accurately as possible. • Save output i.e reconstructed image imgR as fname_out.tif. • Output **the dimensions of output image** and the rmse between your output image and input imag1 If (toshow), Use Subplot(2,2,…) to show input image1, img2, your output image: imgR and error image between img1. Have proper titles that can possibly include info of image size and mean grayscale value and mse for output images (your - SImg)

6	Nil or say [ ]	Histogram Equalization	Fname_img2 is need not be used. Null only	• Compute histogram equalized image of same size of img1 using method as discussed in class • Save output image fname_out.tif. • Output rmse between your output image **outHeq** and image **sysHeq** generated using system inbuilt command histeq. If (toshow), Use Subplot(2,3,…) to show input image img1, your output image **outHeq** and image **sysHeq**. Show corresponding histograms in 2nd row of the plot for each these three images. Have proper titles that can possibly include info of image size and mean grayscale value etc
7	Nil or say [ ]	Histogram Matching	Fname_inp2 is valid and denotes file name of the image whose histogram to be matched.	Compute transformed version of input image1 s.t. transformed image (of same size) has histogram matching as best as possible with the histogram of input image 2 • Save output image fname_out.tif. • Output rmse between histogram of your output image **outHeq** and histogram of input image 2. Use Subplot(2,3,…) to show input image img1, img2 and your output image **outHeq** Show corresponding histograms in 2nd row of the plot for each these three images. Have proper titles that can possibly include info of image size and mean grayscale value etc
8	Nil or say [ ]	Adaptive Histogram Equalization	Fname_img2 is need not be used. Null only	• Compute histogram equalized image of same size using adaptive histogram equalization • Save output image fname_out.tif. • Output rmse between your output image **outHeq** and image **sysHeq** generated using system inbuilt command histeq. If (toshow), Use Subplot(2,3,…) to show input image img1, your output image **outHeq** and image **sysHeq**. Show corresponding histograms in 2nd row of the plot for each these three images. Have proper titles that can possibly include info of image size and mean grayscale value etc
Any other value	Nil or say [ ]	Nothing	--	--

Sample MatLab code (just to give you a reference):

```
function Outx = A1_Keshav_2018CSB9999_2019_CS517(qID, fname_inp1, fname_inp2, fname_out,
prmts, toshow)

Outx=[];
if (qID==1), Outx=interp_NN(fname_inp1,fname_out, prmts, toshow); end
if (qID==3), Outx=myimrotate(fname_inp1,fname_out, prmts, toshow); end

end

function xx = interp_NN(input_fname,fname_out, prmts, toshow)
A=imread(input_fname); B=[];
if (prmts(1) >= size(A,1) && prmts(2) >= size(A,2)),
    B = imresize(A,[prmts(1)  prmts(2)], 'nearest'); % CALL to YOUR CODE comes here
    B2=imresize(A,[prmts(1)  prmts(2)]);
    xx=compute_imgs_rmse(B,B2);
    imwrite(B,sprintf('%s.tif',fname_out));
    if (toshow), figure;
        subplot(2,2,1); imshow(A);
        subplot(2,2,2); imshow(B);
        subplot(2,2,3); imshow(imresize(A,[prmts(1)  prmts(2)], 'nearest'));
        subplot(2,2,4); B3=imresize(A,[prmts(1)  prmts(2)]); imshow(B3);
title(sprintf('Bicubic. Mean=%0.2f RMSE=%0.2f', mean(B3(:)),compute_imgs_rmse(B,B3)));
    end
end
end

function xr = myimrotate(input_fname,fname_out, prmts, toshow)
    A=imread(input_fname); B=[];
    B = imrotate(A,prmts, 'nearest'); % CALL to YOUR CODE comes here
    B2 = imrotate(A,prmts, 'bilinear');
    xr = [size(B) compute_imgs_rmse(B,B2) ];
    imwrite(B,sprintf('%s.tif',fname_out));
%     if (toshow), figure; imshow(B); end   %% Commented.. You have to complete
end


function xx = compute_imgs_rmse(A,B)
    t1=min(size(A,1), size(B,1));
    t2=min(size(A,2), size(B,2));
    err=double(A(1:t1,1:t2)-B(1:t1,1:t2));
    xx=sqrt( sum(sum(double(err.*err))) / (numel(err)) );
end
```