CS540/440 Digital Image Processing

Midterm

1. Problems with images, e.g., over and under exposure, brightness, contrast, dynamic range, can be identified on histogram. The image mid1.jpg is one with a low contrast.

(1) If the image is not a gray image, please convert it into a gray image. Display the histogram in figure 1 with an appropriate title. (10pts)

(2) Based on the above histogram, please find a way and write Matlab code to increase the contrast of this image. Display the original image and your enhanced image in figure 2 with appropriate titles. (20pts)

2. The goal is to process an image that shows a prominently featured object in the center. The processing should be such that the object – i.e., foreground – becomes sharper, and the other parts of the scene – i.e., background – become smoother, for better contrast. Write a MATLAB code for:

(1) Segmenting the image mid2.jpg or mid3.jpg (choose either) into foreground and background by using the following tool, which will give you an images A, called a saliency map. Reading the resulting image A, analyzing this image A, doing thresholding on A to generate a binary image C where the object is white and the background is black and displaying it in figure 3. (20pts)

# Instruction: To segment an image, we use the algorithm proposed in the paper “Global contrast based salient region detection”, which is published in the proceeding of IEEE conference on Computer Vision and Pattern Recognition (CVPR2011). To do segmentation, we run their code as following:

# OS: Windows 64 bit 1. open a terminal 2. cd to the folder path 3. run 'Saliency.exe data'

# (2) Convert the original image mid2.jpg or mid3.jpg to a gray image B. Using the above binary image C and B to generate one image D where the object is in the center but the background is black, and another image E where the object is black but the background does not change, and displaying two images D and E in figure 4 with appropriate titles. (20pts)

# (3) Applying Image Sharpening only on the foreground (object) and displaying the image in figure 5 with an appropriate title. (10pts)

(4) Applying Image Smoothing only on the background by and displaying the images in figure 6 with an appropriate title. (10pts)

(5) Fusing the resulting background and foreground images together into one image and saving the resulting image in a file. (10pts)

3. For graduate students, please segment the image mid4.jpg in your way without using the above algorithm. (10pts)

Matlab functions:

I = imread(image); %read an image

[row,col,dim]=size(I) % check dimention of the matrix or vector

A=ones(3); % D is a 3-by-3 matrix whose each element is 1

D=zeros(3); % D is a 3-by-3 matrix whose each element is 1

C=D.\*2; % elementwise multiply

C=A.\*D; % elementwise multiply of two matrices

figure, subplot(2,2,1), imshow(I), title('RGB color image');

subplot(2,2,2), imshow(I(:,:,1)), title('R channel');

subplot(2,2,3), imshow(I(:,:,2)), title('G channel');

subplot(2,2,4), imshow(I(:,:,3)), title('B channel');

I = rgb2gray(I); %% transfer the color image to gray image

I1 = I(207:207+50, 387:387+50);

figure, imshow(I1,[]); % show the Region of Interest(ROI)

I(207:207+50, 387:387+50) = 0; % change the ROI to black

I(207:207+50, 387:387+50) = 255; % change the ROI to white

%% do binary segmentation for the gray image

I2= im2bw(I, 0.7); % use a threshold in [0,1] for binary segmentation

imwrite(I2, 'dog\_binary.png'); % save the segmented image

MaximumValue=max(max(A)) % compute the maximum value of matrix A.

MinimumValue=min(min(A)) % compute the minimum value of matrix A.

MeanValue=mean(mean(A)) % compute the mean value of matrix A.

MedianValue=median(median(A)) % compute the median value of matrix A.

imhist(I) % calculate the histogram for the intensity image I and displays a plot of the histogram. The number of bins in the histogram is determined by the image type.

[equalizedFoodMat, transFuncMat] = histeq(food); % histogram equalization

imadjust(I,[double(originmin/255),double(originmax/255)],[double(newmin/255),double(newmax/255)]) %scale the image **I** into the equivalent range for [newMin newMax]

AverageFiltering(image, filter); % AverageFiltering function to perform an average filtering operation on image;

MedianFiltering(image, filter); % MedianFiltering function to perform an median filtering operation on image;

filteredMoon = imfilter(Image, h); % imfilter([A](https://www.mathworks.com/help/images/ref/imfilter.html" \l "inputarg_A),[h](https://www.mathworks.com/help/images/ref/imfilter.html#inputarg_h)) filters the multidimensional array A with the multidimensional filter h. The array A can be logical or a nonsparse numeric array of any class and dimension. The result B has the same size and class as A.

edge([I](https://www.mathworks.com/help/images/ref/edge.html" \l "inputarg_I),'Sobel',[threshold](https://www.mathworks.com/help/images/ref/edge.html#inputarg_threshold)) % return all edges that are stronger than threshold. If you do not specify threshold, or is you specify empty brackets, edge chooses the value automatically.