Homework 3

(Total 220 pts)

Due 5:00 pm, November 15, 2024 (Friday)

Canvas submission as a single PDF file (‘hw3_lastname.pdf’)

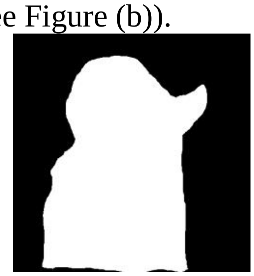
1. (30 pts) Using smoothing filter and thresholding for extraction of regions of interest. Clink the link: https://stsci-opo.org/STScI-01EVVDJAHZV7S6HKCP2K36217G.tif to download the image of the Hickson Compact Group 31, taken by NASA’s Hubble Space Telescope.

   (a) Convert the image to a grayscale image by using the function *rgb2gray*. Display the resulting grayscale image *f*.
   (b) Apply a 151×151 averaging filter on the image and show the filtered image g. Use zero padding for the boundary extension option.
   (c) Apply a thresholding operation on image g to better visualize of the regions of interest, where the threshold is chosen to be 20% of the highest intensity in g. Note: Do not use built-in functions such as *imbinarize*. Display the bi-level image after the thresholding operation.
   (d) Use two other thresholds (40% and 60%) and display the resulting bi-level images after the thresholding operation. Comment on your result.
   (e) Attach the Matlab script and the images displayed.

2. (50 pts) Mini-project on blending images to create a composite image by superimposing a portrait on a blurred background. Note: Do not use built-in functions such as *imfuse*, but instead write your own script to implement the image superimposition.
   Use your phone or camera to take two images: one is a portrait (see, e.g., Figure (a) below), and the other is a landscape image (see Figure (b)).


(a)


(b)


(c)


(d)

   Then crop out the foreground object from the portrait image using the Image Segmenter App in Matlab: >> imageSegmenter
   See a tutorial on segmenting an image by manually drawing regions of interest (ROI): https://www.mathworks.com/help/images/Segment-Image-By-Drawing-Regions.html
   The segmenter will generate a binary mask for the ROI (see, e.g., Figure (c) above). Next, apply a smoothing filter on the landscape image to create a blurred background. Then overlay the ROI on the blurred background based on the binary mask to generate Figure (d). The smoothing filter used in this example has a template of size 15×15. This size should change with varying image sizes, as well as with different blurring effect desired. Experiment with different template sizes and show the composite images generated using three different template sizes.
   Display the images similar to those shown above. Attach your Matlab scripts.

3. (20 pts) Read into Matlab the image f in Figure 4.38(a).
   The image can be downloaded from: http://www.ece.uah.edu/~dwpan/course/ee604/images/book/
   (a) Calculate x and y directional gradients using Sobel's gradient operator and display the images.
   (b) Calculate gradient magnitude using Sobel's gradient operator and display the image.
   (c) Attach the Matlab script and the images displayed.

4. (30 pts) Read into Matlab the image f in Figure 2.36(a).
   The image can be downloaded from: http://www.ece.uah.edu/~dwpan/course/ee604/images/book/
   (a) Display the centered DFT spectrum $|F(u,v)|$ (use log function to enhance the contrast).
   (b) Translate the image by $t_x = 50,$ and $t_y = 50$ and obtain a new image g. Display the centered DFT spectrum $|G(u,v)|$ (use log to enhance the contrast).
   (c) Then rotate the translated image g anti-clockwise by 45° (using bilinear interpolation) and obtain another new image h. Make the output image h the same size as the input image g by cropping the rotated image to fit.
   Display the centered DFT spectrum |H(u,v)| (use log function to enhance the contrast).
   (d) Attach the Matlab script and the images displayed.

5. (70 pts) 2D DFT on and image and basis images.
   (a) For a 4×4 image $f(x,y) = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$, where $x = 0,1,2,3;$ and $y = 0,1,2,3$. Apply 1D DFT on each of the rows, respectively. Determine the resulting transformed image $F_r(u,v)$ and show its content.
   (b) In Matlab, apply 1D DFT on each of the columns of the image $F_r(u,v)$ obtained in (a). Determine the resulting transformed image $F_c(u,v)$ and show its content..
   (c) In Matlab, apply 2D DFT directly on $f(x,y)$. Determine the resulting transformed image $F(u,v)$ and show its content.
   (d) Compare $F_c(u,v)$ and $F(u,v)$. Are they the same?
   (e) In the formula for the inverse 2D DFT,
   $f(x,y) = \frac{1}{16}\sum_{u=0}^{3}\sum_{v=0}^{3} F(u,v)\, B(u,v;x,y)$, where $B(x,y;u,v) = e^{j2\pi\left(\frac{ux}{4}+\frac{vy}{4}\right)}$ is the basis image for the frequency (u,v). Determine and show the content of the following 4×4 basis images $B(x,y;0,0)$, $B(x,y;1,1)$, $B(x,y;2,2)$, $B(x,y;3,3)$.
   (f) Write a Matlab script to calculate $f(x,y)$ using the formula in (e) based on the basis images. Show the result and the screenshot of running your script.
   (g) Attach the Matlab scripts.

6. (20 pts) A digital image has 8 bits per pixel. The image is transmitted over a binary communication link, with each pixel being transmitted in a bit-by-bit manner (starting with the most significant bit). Suppose this link is noisy, with probability of bit error being 0.2. Bit errors occur when a transmitted 0 is received as 1, or a transmitted 1 is received as 0. For a certain transmitted pixel with an intensity value of 128 (in decimal),

what is the probability that this pixel will be received as a pixel with intensity value being 8 (in decimal)? Round your answer to the fourth digit to the right of the decimal point.