EE371Q - Digital Image Processing

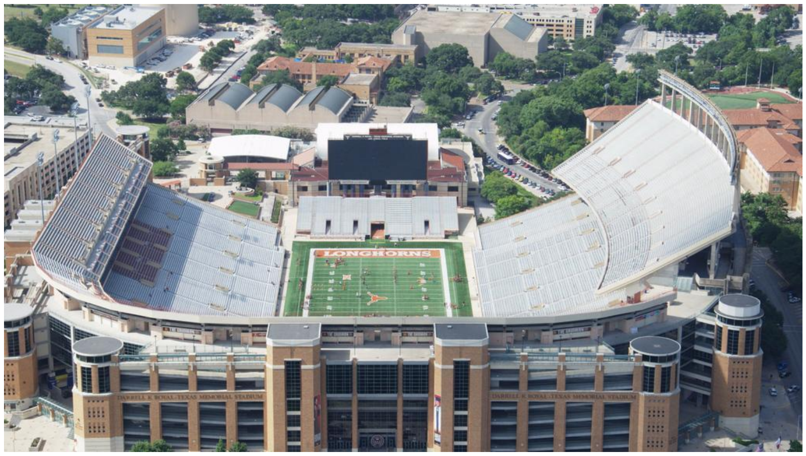
Homework #3

Due on Nov 10th 1:59pm (before class)

- You are encouraged to work with other students to understand the course material. However, sharing source code, images, or other answers from any homework is strictly forbidden. You are to submit your own work.
- Please include your written answer and/or output images for each problem.
- Please show all steps for full credit and attach the code to each problem.
- Images needed for this homework can be downloaded from Canvas under their respective homework folder.
- This homework will help you to familiarize yourself with Python if you have not worked with it in the past. The internet is a great resource for finding help with Python functions and will be available for you to use.
- MATLAB or Python can be used to solve these problems, it is your preference.
- Please provide screenshots and necessary code explanation for each question and show them in a single PDF document.

Question 1 : Gabor Filters (30 points)

The goal of this problem is to implement Gabor Filters and understand them, on the image stadium.jpg. You cannot use the imgaborfilt() function directly. Helpful functions: np.roll, np.repeat, imread, rgb2gray, abs, real.



stadium.jpg

a) (5 points) Read in the stadium.jpg file, convert it to grayscale. Display the image with appropriate labels.
b) (10 points) Write a function “gaborfilt()” which computes the magnitude and phase response of a Gabor filter for the given input grayscale image. The input parameters to this function should be the frequency in ‘cycles per image’ of the sinusoidal carrier, and the orientation of the filter in degrees.
c) (5 points) Write a function “gaborfilt_2()” which computes the real and imaginary parts of a Gabor filter for the given input grayscale image. The input parameters to this function should be the frequency in ‘cycles per image’ of the sinusoidal carrier, and the orientation of the filter in degrees.
d) (5 points) You should now use either of the filters from b) or c) on the grayscale version of stadium.jpg, for different values of “frequency” and “orientation”. By changing the orientation angle, we can search for features in a particular direction. For a frequency of your choice, generate 10 images with orientation varying from 18 degrees to 180 degrees (in strides of 18 degrees). Display all the 10 images in a 5x2 grid with appropriate labels. Comment on which filter you chose, from part b) or c), and why.
e) (5 points) Comment on why you chose that particular frequency for part d), and also on the observations on the differences between the 10 images. What direction do you think most features are present in stadium.jpg from the images, and does that make sense? Attach any one .jpg picture in which Gabor Filter will show up features significantly in one orientation, but not much in all other orientations. Also mention the angle (orientation) for the image you attach in which most features show up significantly.

Question 2 : Non Local (NL) Means (40 points)

Here we will be looking at the application of Non-Local Means in removing noise from the Google Chrome logo “logo.jpg” image, which is noise distorted. Helpful functions: skimage.util.view_as_windows, scipy.spatial.distance.cdist, padarray, im2col, pdist2, sum, reshape.

logo.jpg

a) (5 points) You need to read logo.jpg, convert it to grayscale, and resize the image so that it is a perfect square. Display this resulting image.
b) (10 points) You will now calculate the Luminance Similarity Measure “W(m,i)” between all the 3x3 square windows in the image, using the information given in Module 5 slide 14 onwards. Use Kw = 1 and σw = 1. Normalize “W(m,i)” so that the sum across each row is equal to 1.0.
c) (5 points) From all the windows, we will now compare the similarity measure images generated between choosing two different windows in the image. For this part, take the 3x3 window in the top-left corner of the image, and reshape it to a 3x3 matrix. Take the first row in W which corresponds to all of the pair-wise distances to that 3x3 window, and reshape it to the size of the original image “logo.jpg”. Display the resulting window and the original image side by side with appropriate labels.
d) (5 points) You will now need to repeat the same operation you did in part c), but by taking the 3x3 window in the middle of the left side of the image. Display the resulting window and the original image side by side with appropriate labels.
e) (5 points) We now have two resulting similarity measure images. What do the high and low values correspond to in the resulting similarity measure images? Why and where do the similarity measure images differ in part c) and d)?
f) (10 points) Using the luminance similarity measure, perform Non Local (NL) Means filtering on logo.jpg, and remove the noise. Display the resulting image and the original image in a 2x1 grid with appropriate results. Comment a few lines on the result achieved.

Question 3 : Block Truncation Coding (BTC) (25 points)

BTC is a fast and lossy compression technique. In this problem, we will understand how BTC is used, along with its advantages and disadvantages. **4x4 blocks** are used where applicable in the following sub-problems. You can use the following link to learn more about BTC: https://en.wikipedia.org/wiki/Block_Truncation_Coding#Encoder . Helpful functions: col2im, im2col, bi2de, de2bi.

ronaldo.jpg

a) (5 points) You should read ronaldo.jpg, and make it grayscale. Resize the image by a factor of ½ to make the compression faster. Resize the image further such that the height and width of the images are both multiples of 4; such that it is still roughly the same dimensions as ½ of the original image dimensions. Display the grayscale compressed image.
b) (5 points) Write a function to compute the mean and standard deviation of a 4x4 block with ‘B1’ bits to compute and store the mean, and ‘B2’ bits to compute and store the standard deviation; where ‘B1’ and ‘B2’ are input parameters to the function. Hint: you might need to compute the mean and standard deviation values, and then truncate in the end based on ‘B1’ and ‘B2’ parameter values.
c) (5 points) Write a function to compute the 4x4 BTC binary block by thresholding the input 4x4 image block at the mean. Pass ronaldo.jpg through this function for 3 different ‘B1’, ‘B2’ settings; {2,1}, {3,3}, and {7,5}. Compute the BTC binary blocks image for ronaldo.jpg, and display the 3 images in a 3x1 grid with appropriate labels.

d) (5 points) Write a function to decode a 4x4 BTC binary block given the mean and standard deviation as the input parameters, and get back the original 4x4 image block. Pass the 3 BTC binary block images generated in part c) through this function, for the aforementioned ‘B1’, ‘B2’ settings; and get back the 3 decompressed ronaldo.jpg images. Display these three images along with the original image in a 2x2 grid with appropriate labels.

e) (5 points) Compute and report the compression ratios for the 3 BTC encoded images generated.

Question 4 : Open Ended (5 marks)

Frame a question for me worth 5 marks based on any concept you learnt in Module 4 or 5 (that hasn’t already been covered in the homeworks). It should be complete with helpful functions, input images, and a small paragraph on how you would approach the problem (only text, no code required).