CSE 166: Image Processing, Spring 2020 – Assignment 5

Instructor: Ben Ochoa

Due: Friday, May 29, 2020 11:59 PM

- Review the academic integrity and collaboration policies on the course website.
- This assignment must be completed individually.
- This assignment contains both math and programming problems.
- Programming aspects of this assignment must be completed using MATLAB.
- Unless specified below, you may not use MATLAB functions contained in toolboxes, including the image processing toolbox. Use the MATLAB `which` command to determine which toolbox a function is contained in. If you are unsure about using a specific function, then ask the instructor for clarification.
- You must prepare a report as a pdf file. The report must contain your solutions and results, and all of your MATLAB source code as a listing in the appendix of your report.
- Additionally, you must create a zip file containing all of your MATLAB source code.
- Your source code must contain a file `main.m` which runs all code necessary to produce results for your report. `main.m` should run start to finish without error. Use relative paths to read input data. For questions which require numerical output, `main.m` should print a message indicating what question is being answered followed by the numerical output for that question. Example: `display('Problem 2b')` followed by your answer to problem 2b. The instructors should be able to reproduce your report by running `main.m`
- You must submit both files (.pdf and .zip) on Gradescope. Further, you must mark each problem on Gradescope in the pdf file.
- It is highly recommended that you begin working on this assignment early.

Problems

1. **Textbook problems (15 points)**
   (a) Problem 8.5(a) (1 point)
   (b) Problem 8.9(a) (1 point)
   (c) Problem 9.8 (3 points)
   (d) Problem 9.9 (3 points)
   (e) Problem 9.10 (2 points)
   (f) Problem 9.11 (2 points)

(g) Problem 9.23 (3 points)

2. **Programming problems (40 points)**

   (a) **The discrete cosine transform and lossy block processing (15 points)**

   Develop a MATLAB script called `hw5_lossy_dct_block.m` that reads the input image `cameraman.tif` (included with MATLAB) and independently processes non-overlapping blocks (i.e., subimages) of size 8×8 as follows. For each block, compute the discrete cosine transform (DCT), retain the n greatest magnitude DCT coefficients (i.e., set the $64 - n$ remaining DCT coefficients to zero), and compute the inverse DCT. The 2D DCT and inverse DCT must be implemented using a set of 64 basis images, which must be calculated without the use of any MATLAB toolbox functions. Perform this for $n = 32, 16, 8, 4, 2$ and include the resulting images in your report.

   Additionally, include figures of the error images (with colorbars) and the root-mean-square error associated with each output image. Include a title with each figure. Briefly discuss the resulting images, including any differences between them.

   You may use the `blockproc` function. Use the function `imread` to read the input image in MATLAB. Use `imwrite` or `saveas` to write the output image in MATLAB.

   (b) **The discrete wavelet transform and lossy processing (15 points)**

   Develop a MATLAB script called `hw5_lossy_dwt.m` that reads the input image `cameraman.tif` (included with MATLAB), computes the 2-level discrete wavelet transform (DWT), sets to zero p percent of the smallest magnitude detail coefficients, and computes the inverse DWT. Perform this for $p = 25\%, 50\%, 90\%, 95\%$ using Haar, Daubechies, and biorthogonal wavelets. For each wavelet and value of p create a figure which contains 3 columns and 2 rows. The first row should show the reconstructed image using each wavelet type. The second row should include a figure of the scaled error image (with colorbars). Include the root-mean-square error as a caption in the image. Include a title with each figure and briefly discuss the resulting images, including how the choice of wavelet and n effects compression quality. Further, discuss any differences you observe with compression based on the DCT.

   You may use the built in wavelet tools in MATLAB. Use the function `imread` to read the input image in MATLAB. Use `imwrite` or `saveas` to write the output image in MATLAB.

   (c) **Improving Text Recognition Using Morphological Image Processing (10 points)**

   Develop a MATLAB script called `hw5_morphological.m` that reads the input image `text.tif` and performs morphological image processing on the image to remove noise in the text. Your results should visibly improve the quality of the letters. Include the original and resulting image in your report and describe your methodology. You may use the `imerode`, `imdilate`, `imopen`, `imclose`, and `strel` functions. Use the function `imread` to read the input image in MATLAB. Use `imwrite`/`saveas` to write the output image in MATLAB.